Attention-Guided Low-Rank Tensor Completion

Truong Thanh Nhat Mai, *Student Member, IEEE*, Edmund Y. Lam, *Fellow, IEEE*, and Chul Lee, *Member, IEEE*

***Abstract*—Low-rank tensor completion (LRTC) aims to recover missing data of high-dimensional structures from a limited set of observed entries. Despite recent significant successes, the original structures of data tensors are still not effectively preserved in LRTC algorithms, yielding less accurate restoration results. Moreover, LRTC algorithms often incur high computational costs, which hinder their applicability. In this work, we propose an attention-guided low-rank tensor completion (AGTC) algorithm, which can faithfully restore the original structures of data tensors using deep unfolding attention-guided tensor factorization. First, we formulate the LRTC task as a robust factorization problem based on low-rank and sparse error assumptions. Low-rank tensor recovery is guided by an attention mechanism to better preserve the structures of the original data. We also develop implicit regularizers to compensate for modeling inaccuracies. Then, we solve the optimization problem by employing an iterative technique. Finally, we design a multistage deep network by unfolding the iterative algorithm, where each stage corresponds to an iteration of the algorithm; at each stage, the optimization variables and regularizers are updated by closed-form solutions and learned deep networks, respectively. Experimental results for high dynamic range imaging and hyperspectral image restoration show that the proposed algorithm outperforms state-of-the-art algorithms.**

***Index Terms*—Low-rank tensor completion, robust tensor factorization, algorithm unrolling, high dynamic range (HDR) imaging, hyperspectral image (HSI) restoration.**

I. Introduction

TENSORS are multidimensional arrays of numerical values, which can be regarded as high-dimensional generalizations of vectors and matrices. In particular, tensors are a natural way to represent and store captured real-world signals. For instance, color images are 3rd-order tensors because they are combined 2D arrays, and color videos are 4th-order tensors because they are sets of coherent color images along the time dimension. However, the quality of signals is often affected by hardware limitations and disturbances, which lead to inaccuracies and measurement errors, resulting in data tensors being corrupted and damaged. Restoration of corrupted tensors occurs in a wide range of image processing and computer vision applications, such as image restoration [1], hyperspectral image (HSI) restoration [2], video completion [3], inpainting [4], and high dynamic range (HDR) imaging [5].

A common assumption in tensor recovery is that multidimensional signals, such as images and videos, often exhibit (approximately) low rank [6], [7]. Therefore, low-rank tensor completion (LRTC) algorithms have been developed to recover latent clean tensors from a small number of observed entries under the assumption of linear dependencies between their elements. Specifically, given an nth-order corrupted data tensor $\boldsymbol{\mathcal{D}}$ with a set of observed entries $\boldsymbol{\Omega}$, a generic LRTC problem that estimates missing elements can be formulated as

$$\begin{aligned} \underset{\boldsymbol{\mathcal{X}}}{\text{minimize}} \quad & \text{rank}(\boldsymbol{\mathcal{X}}) \\ \text{subject to} \quad & \boldsymbol{\mathcal{X}}(i_1,\ldots,i_n) = \boldsymbol{\mathcal{D}}(i_1,\ldots,i_n), \\ & \forall (i_1,\ldots,i_n) \in \boldsymbol{\Omega}. \end{aligned} \tag{1}$$

Note that the LRTC can be regarded as a generalization of low-rank matrix completion [8] because matrices are technically 2nd-order tensors. However, because the higher-dimensional structures of tensors lead to more complex intrinsic properties, different definitions of tensor rank [9], [10], [11], [12], [13] have been proposed to characterize the low-rankness of tensors from different aspects, resulting in different LRTC approaches for each application [14].

Although LRTC algorithms employing several tensor rank definitions have achieved significant success for image processing and computer vision tasks [15], they may fail to preserve the original structures of the data tensors because they focus on optimizing the convex surrogates of the tensor ranks, thereby overlooking the structural information and spatial dependencies in the latent tensors. In addition, because the true signal tensors are only approximately low rank in practice [14], the restoration results may be inaccurate. Several priors and constraints have been proposed to address this issue [2], [3], [16], [17]. However, since these priors were constructed by observing a small number of local samples, their generalizability is limited. For example, the priors are often in the form of norms, such as the ℓ_1-norm of gradient $\|\nabla \boldsymbol{\mathcal{X}}\|_1$ or the total variation norm $\|\boldsymbol{\mathcal{X}}\|_{\text{TV}}$, which are ineffective in capturing the spatial dependencies and high-level structures in the underlying data. Moreover, highly complex operators, such as singular value decomposition (SVD), are often employed to compute the convex surrogates of the tensor ranks, resulting in computationally expensive algorithms.

Recently, deep learning techniques using convolutional neural networks (CNNs) have been actively developed for image

Manuscript received 17 May 2023; revised 27 May 2024; accepted 11 July 2024. Date of publication 17 July 2024; date of current version 5 November 2024. This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) under Grant 2022R1F1A1074402 and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development under Grant IITP-2023-RS-2023-00254592 funded by MSIT. Recommended for acceptance by J. A. Schnabel. *(Corresponding author: Chul Lee.)*

Truong Thanh Nhat Mai and Chul Lee are with the Department of Multimedia Engineering, Dongguk University, Seoul 04620, South Korea (e-mail: mtntruong@mme.dongguk.edu; chullee@dongguk.edu).

Edmund Y. Lam is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong (e-mail: elam@eee.hku.hk).

Digital Object Identifier 10.1109/TPAMI.2024.3429498

processing and computer vision tasks [18] due to their powerful feature learning and inference. Motivated by these successes, CNN-aided LRTC algorithms [5], [19], [20], [21], [22], [23], [24], [25], [26] have been proposed to exploit the capability of CNNs to mitigate the aforementioned limitations of purely model-based LRTC algorithms. However, although CNNs have proven their effectiveness in enhancing tensor recovery performance, the desired structures of the restored tensors have not been fully considered and modeled in CNN-aided LRTC algorithms. Therefore, preserving and restoring the original structure of corrupted data remains an open challenge. Moreover, because conventional CNN-aided LRTC algorithms require complex operators such as SVD along with complicated CNNs, they are computationally expensive.

In this work, to address the aforementioned challenges in LRTC, we develop an attention-guided LRTC algorithm via deep unfolding robust tensor factorization, called attention-guided tensor completion (AGTC), which effectively preserves the original structures of data tensors. First, we formulate the LRTC task as a robust tensor factorization problem that decomposes an input data tensor into a target low-rank tensor and a sparse error component by integrating an attention mechanism. A CNN-based attention mechanism guides low-rank tensor recovery to effectively localize important regions, thereby preserving multidimensional spatial dependencies and ensuring a more faithful restoration of the original structures of tensors. Next, since real-world data may not accurately follow low-rank and sparse error assumptions, we compensate for the modeling inaccuracies by exploiting the capability of CNNs to learn visual features from real data. Specifically, we define CNN-based learnable regularizers instead of explicit mathematical expressions. We then iteratively solve the tensor factorization problem by employing an algorithm unrolling strategy [27]. To this end, we design a multistage deep network wherein the optimization variables are updated in each stage by either closed-form or CNN-based solutions. Integration of the learnable regularizers into LRTC enables superior generalizability to fully deep learning-based algorithms by combining the advantages of the theoretical foundation of mathematical models and the capability of CNNs to compensate for the modeling inaccuracies via learning from data.

In summary, we make the following contributions.

- We propose a new attention-guided LRTC algorithm that can effectively preserve the high-level structural information of data tensors. To this end, we formulate the LRTC task as a tensor factorization problem by integrating an attention mechanism, focusing on salient regions to better preserve and restore the original tensor structures. Due to the improved representation the attention provides, the proposed AGTC requires minimal learned information and training data, thus reducing the complexity. In addition, we define two learnable regularizers to compensate for the modeling inaccuracies. Then, we solve the attention-guided LRTC problem iteratively using the augmented Lagrange multiplier (ALM) method.
- We develop a deep neural network for the attention-guided LRTC using an ALM-based iterative algorithm by employing an algorithm unrolling strategy. Specifically, we unfold the iterations of the algorithm into a multistage network, where each stage in the network corresponds to an iteration. In each stage, the optimization variables are updated in two phases: reconstruction by closed-form solutions and regularization by CNN-based regularizers.
- We experimentally demonstrate that the proposed algorithm can overcome the limitations of state-of-the-art algorithms in preserving tensor structures while requiring fewer training samples to produce better restoration results in two practical applications: HDR imaging and HSI restoration. We also analyze the effectiveness of the attention on the performance. Furthermore, we show that the proposed algorithm is computationally more efficient than conventional CNN-aided LRTC algorithms.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed AGTC algorithm, and Section IV discusses the experimental results. Finally, Section V concludes the paper.

II. Related Work

A. Low-Rank Tensor Completion

The multidimensional structures of tensors lead to complex intrinsic properties. Therefore, different definitions of tensor ranks have been proposed to characterize the rank of tensors from different aspects. The most commonly used definitions of the tensor rank are the CANDECOMP/PARAFAC (CP) rank [9], Tucker rank [10], tensor train rank [11], tensor ring rank [12], and tensor tubal rank [13]. Different algorithms for the LRTC have been developed due to these different definitions.

The CP rank [9] is defined as the lowest number of rank-one tensors, i.e., the vectors that form the given tensor. Although the CP rank has achieved significant successes in both theoretical aspects [28], [29] and practical applications [30], [31], [32], LRTC employing the CP rank remains challenging because determining an optimal CP decomposition is an NP-hard problem [33]. Thus, several definitions of tensor rank have been proposed for efficient representation of the multidimensional dependencies of tensors.

The Tucker rank [10] was defined based on the ranks of all unfolded matrices of the given tensor along all dimensions. Because this definition is based on the matrix rank, convex surrogates for matrix rank can be used to form a convex surrogate of the Tucker rank [7]. Therefore, the Tucker rank has been adapted to several data completion applications [34], [35], [36]. However, the Tucker rank cannot capture the global correlations between the unfolding modes as the tensors are unfolded into matrices and thus lose their original structures [37].

The tensor train rank [11] and tensor ring rank [12], which are computed from a train or ring of interconnected core tensors decomposed from a given tensor, respectively, were proposed to overcome the limitations of the Tucker rank. The tensor train rank can better capture global correlations and thus improve the tensor restoration performance [37], [38], [39]. The tensor ring rank [12], which is a generalization of the tensor train rank, has also achieved significant success [17], [40], [41]. However, algorithms employing the tensor train rank and tensor ring

rank still suffer from heavy computations due to their complex decomposition.

The tubal rank [13] was proposed for 3rd-order tensors based on tensor SVD (t-SVD) [42], which can characterize low-rank properties better than other definitions while avoiding information loss [13], [42]. The tubal nuclear norm [43], a convex surrogate for tubal rank, has been widely adopted in LRTC models for various imaging and vision tasks because they can be represented by 3rd-order tensors [3], [44], [45]. Furthermore, attempts to extend the tubal rank for higher-dimensional tensors have been made [46], [47]. The tubal rank has superior modeling ability of low-rank structures and has shown state-of-the-art performance in LRTC applications [8], [44], [45], [48]. However, although LRTC algorithms employing the tubal rank have shown some advantages, they require SVD to compute the singular tensors, which is computationally expensive.

Although extensive research efforts have been made for LRTC algorithms using different definitions of tensor ranks, the original structures of the data tensors are still not fully considered in their mathematical models, yielding less accurate restoration results. Furthermore, a solution to the optimization problem may become inaccurate because there are cases in which the true signal tensors are only approximately low rank.

B. CNN-Aided Low-Rank Tensor Completion

CNN-aided LRTC algorithms have been developed to overcome the limitations of mathematical models by exploiting the capability of CNNs to learn diverse features from data and perform complex inferences. For example, CNNs were integrated into LRTC formulations to learn priors by extracting latent features from data [20], [23], [24], [25]. LRTC algorithms also transform data tensors into other domains using CNNs to represent low-rank structures more effectively. For example, Bragilevsky and Bajić [19] employed CNNs to transform the input image to the feature domain and then developed an LRTC algorithm to recover the deep feature tensor. Liu et al. [22] and Wang et al. [26] developed nonlinear transforms to model low-rank tensors using CNNs. Luo et al. [21] proposed an LRTC-based HSI recovery in a transform domain where the transformation was performed using a CNN via self-supervised learning. Mai et al. [5] proposed an explainable HDR imaging algorithm by unfolding an iterative LRTC algorithm into a deep network with CNN-based proximal operators.

Despite the significant success achieved by employing CNNs, mathematical models employed in conventional CNN-aided LRTC algorithms are still ineffective in capturing the desired structural information and spatial dependencies of the restored tensors due to their norm-based formulations, such as [5], [20], [45]; this establishes the problem of faithfully preserving and restoring the original structures of corrupted data as an open challenge. Moreover, conventional CNN-aided LRTC algorithms remain computationally expensive since they require complex operators, such as SVD in [20], and complicated deep neural networks as in [5].

C. Algorithm Unfolding

Algorithm unfolding [27] is a technique that turns a model-based iterative algorithm into a multistage deep network, where each stage corresponds to a single iteration of the algorithm. Therefore, algorithm unrolling benefits from the advantage of both approaches, i.e., better performance and interpretability, by combining the theoretical robustness of the iterative algorithms with the data-driven learning capability of deep networks. Because of its advantages, algorithm unfolding has achieved significant success in a wide range of imaging and computer vision applications.

Algorithm unfolding was first developed for sparse coding and compressive sensing [49], which are primarily iterative algorithms. Later, algorithm unfolding has been employed to learn unknown model parameters, such as diffusion filters [50], blur kernels [51], and measurement matrices [52], in various imaging applications. In addition, in low-level computer vision tasks, the prior and regularization terms, such as sparse prior [53], denoising prior [54], [55], Gaussian prior [56], and degradation priors [57], [58], [59], were learned using algorithm unfolding. In this work, we propose an attention-guided and computationally efficient LRTC algorithm via algorithm unfolding to address the aforementioned limitations of conventional LRTC algorithms.

III. Proposed Algorithm—AGTC

We first introduce the notations and definitions used in this paper. Then, we develop an attention-guided LRTC problem and an iterative algorithm to solve it. Finally, we construct a computationally efficient deep network by unfolding the iterative optimization algorithm.

A. Notations and Definitions

We denote tensors by calligraphic Latin or uppercase Greek letters in bold (e.g., $\boldsymbol{\mathcal{A}}$ or $\boldsymbol{\Gamma}$), while the non-bold calligraphic Latin letters are used to denote functions, e.g., $\mathcal{F}_{\text{att}}(\cdot)$. Matrices are denoted by uppercase letters in bold (e.g., $\mathbf{A}$) and scalars by Latin or Greek letters in italics (e.g., k, N, or λ).

In this work, we focus on the LRTC problem for 3rd-order tensors, because they are the most common data formats in image processing and computer vision. For a 3rd-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we denote the element at location (i_1, i_2, i_3) by $\boldsymbol{\mathcal{A}}(i_1, i_2, i_3)$, and its i-th horizontal, lateral, and frontal slices by $\boldsymbol{\mathcal{A}}(i, :, :)$, $\boldsymbol{\mathcal{A}}(:, i, :)$, and $\boldsymbol{\mathcal{A}}(:, :, i)$, respectively. As we focus on frontal slices in this work, we denote the i-th frontal slice by $\boldsymbol{\mathcal{A}}^{(i)}$ for simplicity.

Next, we introduce the essential tensor operations used throughout this paper. For unary operations, the ℓ_1-norm $\|\cdot\|_1$ and Frobenius norm $\|\cdot\|_F$ of a tensor are defined as

$$\|\boldsymbol{\mathcal{A}}\|_1 = \sum_{i_1, i_2, i_3} |\boldsymbol{\mathcal{A}}(i_1, i_2, i_3)|, \tag{2}$$

$$\|\boldsymbol{\mathcal{A}}\|_F = \sqrt{\sum_{i_1, i_2, i_3} |\boldsymbol{\mathcal{A}}(i_1, i_2, i_3)|^2}, \tag{3}$$

respectively. For binary operations, the inner product $\langle \cdot, \cdot \rangle$ of two tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \sum_{i_1, i_2, i_3} \boldsymbol{\mathcal{A}}(i_1, i_2, i_3) \boldsymbol{\mathcal{B}}(i_1, i_2, i_3). \tag{4}$$

The tensor-tensor product (t-product) [42] of two tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as

$$\boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}} = \texttt{fold}(\texttt{bcirc}(\boldsymbol{\mathcal{A}})\texttt{unfold}(\boldsymbol{\mathcal{B}})), \tag{5}$$

where $\texttt{bcirc}(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ is the block circulant matrix of $\boldsymbol{\mathcal{A}}$

$$\texttt{bcirc}(\boldsymbol{\mathcal{A}}) = \begin{bmatrix} \boldsymbol{\mathcal{A}}^{(1)} & \boldsymbol{\mathcal{A}}^{(n_3)} & \cdots & \boldsymbol{\mathcal{A}}^{(2)} \\ \boldsymbol{\mathcal{A}}^{(2)} & \boldsymbol{\mathcal{A}}^{(1)} & \cdots & \boldsymbol{\mathcal{A}}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{A}}^{(n_3)} & \boldsymbol{\mathcal{A}}^{(n_3-1)} & \cdots & \boldsymbol{\mathcal{A}}^{(1)} \end{bmatrix}, \tag{6}$$

$\texttt{unfold}(\boldsymbol{\mathcal{B}}) = [\boldsymbol{\mathcal{B}}^{(1)}; \boldsymbol{\mathcal{B}}^{(2)}; \ldots; \boldsymbol{\mathcal{B}}^{(n_3)}] \in \mathbb{R}^{n_1 n_3 \times n_2}$, and $\texttt{fold}$ is the reverse of $\texttt{unfold}$, i.e., $\texttt{fold}(\texttt{unfold}(\boldsymbol{\mathcal{B}})) = \boldsymbol{\mathcal{B}}$.

Finally, we denote $\mathcal{P}_{\boldsymbol{\Omega}} : \mathbb{R}^{n_1 \times n_2 \times n_3} \mapsto \mathbb{R}^{n_1 \times n_2 \times n_3}$ as the orthogonal projection of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ onto the subspace corresponding to the set of observed elements $\boldsymbol{\Omega}$ as

$$[\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{A}})](i_1, i_2, i_3) = \begin{cases} \boldsymbol{\mathcal{A}}(i_1, i_2, i_3), & \text{if } (i_1, i_2, i_3) \in \boldsymbol{\Omega}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

B. Problem Formulation

We consider a general robust LRTC problem for 3rd-order tensors, in which an input tensor is decomposed into a target low-rank tensor and a sparse error component. Specifically, given an observed data tensor $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ whose entries are incomplete and corrupted by noise, the robust LRTC problem for data recovery can be formulated as

$$\begin{aligned} \underset{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{E}}}{\text{minimize}} \quad & \text{rank}(\boldsymbol{\mathcal{X}}) + \|\boldsymbol{\mathcal{E}}\|_0 \\ \text{subject to} \quad & \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}) = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}), \end{aligned} \tag{8}$$

where $\boldsymbol{\mathcal{X}}$ is the low-rank tensor to be recovered, $\boldsymbol{\mathcal{E}}$ is the sparse error component, $\boldsymbol{\Omega} \subseteq [n_1] \times [n_2] \times [n_3]$ is the set of observed entries, and $\mathcal{P}_{\boldsymbol{\Omega}}$ is the projection defined in (7).

As mentioned in Section II, several definitions of tensor ranks have been proposed to characterize the low-rankness of tensors from different aspects. Among the different definitions, we employ the tubal rank, which can characterize low-rank more effectively than others while avoiding information loss [13], [42]. However, most of the existing LRTC algorithms, e.g., [5], [20], [45], employing tubal rank use t-SVD to compute the tensor norms, which requires high computational resources. To overcome this limitation, we employ a tensor factorization strategy [14] for a low-complexity LRTC. Specifically, the low-rank tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with tubal rank r can be factorized into the t-product $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}}$, where $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{r \times n_2 \times n_3}$. This strategy avoids the need for SVD to compute the tensor norms in the LRTC formulations, making it computationally more efficient. Furthermore, tensor factorization prevents changes in the shapes of the input data and preserves spatial dependencies, which are difficult to achieve in several conventional algorithms as they transform the tensors before performing optimization. For example, LRT-HDR [5] requires images vectorized for use as input, and HTNN-DCT [60] computes the tensor norm of flattened tensors. In contrast, the proposed algorithm can directly process images, making it applicable to a wider range of applications, e.g., hyperspectral imaging. By enforcing the tubal rank via tensor factorization and relaxing the ℓ_0-norm $\|\boldsymbol{\mathcal{E}}\|_0$ using the ℓ_1-norm $\|\boldsymbol{\mathcal{E}}\|_1$, the LRTC problem in (8) becomes

$$\begin{aligned} \underset{\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{E}}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{X}}\|_F^2 + \lambda \|\boldsymbol{\mathcal{E}}\|_1 \\ \text{subject to} \quad & \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}) = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}), \end{aligned} \tag{9}$$

where the parameter λ controls the relative importance between two terms.

Note that the optimization problem in (9) suffers from the same limitations as conventional LRTC algorithms. Specifically, the first term $\|\boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{X}}\|_F^2$, which determines the recovered tensors, does not consider the spatial properties and/or dependencies across multiple dimensions of the recovered tensors, degrading their quality. To address this issue, we incorporate an attention mechanism to improve the integrity of reconstructed tensors by localizing important regions to preserve multidimensional spatial dependencies. The tensor factorization in (9) enables the integration of attention mechanisms because it processes the original data tensors, whereas norm-based formulations, e.g., [5], [20], [45], process singular values of data tensors in the Fourier transform domain, making it intractable to analyze, interpret, and control the impact of attention on the reconstructed low-rank tensor. Then, the LRTC formulation in (9), guided by attention, is rewritten as

$$\begin{aligned} \underset{\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{E}}}{\text{minimize}} \quad & \frac{1}{2}\left\| \sqrt{\mathcal{F}_{\text{att}}(\boldsymbol{\mathcal{D}})} \odot (\boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{X}}) \right\|_F^2 + \lambda \|\boldsymbol{\mathcal{E}}\|_1 \\ \text{subject to} \quad & \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}) = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}), \end{aligned} \tag{10}$$

where $\mathcal{F}_{\text{att}}(\boldsymbol{\mathcal{D}}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the attention map computed from the input data tensors using a CNN, and $\sqrt{\cdot}$ and $\odot$ denote the element-wise square root and multiplication, respectively. The term $\|\boldsymbol{\mathcal{L}} * \boldsymbol{\mathcal{R}} - \boldsymbol{\mathcal{X}}\|_F^2$ is now guided by CNN-based attention to better preserve the structures of the original tensor, ensuring faithful tensor restoration. Thus, the CNN-based attention component mitigates the limitations of norm-based terms that ignore the structures of the desired tensors.

The tensor factorization model for LRTC in (10) is established based on certain assumptions, i.e., low-rankness and sparseness for $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{E}}$, respectively. However, this model may fail to represent real-world scenarios accurately, e.g., the original data are only approximately low-rank, or the errors are effectively dense; these modeling inaccuracies degrade the restoration accuracy. Therefore, to overcome the limitations of the models by compensating for the modeling inaccuracies, we employ two implicit regularization functions, $g_{\boldsymbol{\mathcal{X}}} : \mathbb{R}^{n_1 \times n_2 \times n_3} \mapsto \mathbb{R}$ and $h_{\boldsymbol{\mathcal{E}}} : \mathbb{R}^{n_1 \times n_2 \times n_3} \mapsto \mathbb{R}$, for $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{E}}$, respectively. Furthermore,

we consider a real-world data acquisition scenario as similarly done in [5], [61], i.e., only a set of entries corrupted by a small amount of noise is observed by $\mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}) = \mathcal{P}_{\mathbf{\Omega}}(\mathcal{X} + \mathcal{E} + \mathcal{N})$, where $\mathcal{N}$ is a noise tensor with a noise level $\|\mathcal{P}_{\mathbf{\Omega}}(\mathcal{N})\|_F \leq \delta$ for some $\delta > 0$. We employ a tensor $\mathcal{S}$ of slack variables to compensate for missing and noisy entries in $\mathcal{D}$. Then, the attention-guided LRTC problem in (10) can be rewritten as

$$\begin{aligned} \underset{\mathcal{L},\mathcal{R},\mathcal{X},\mathcal{E},\mathcal{S}}{\text{minimize}} \quad & \frac{1}{2}\left\|\sqrt{\mathcal{F}_{\text{att}}(\mathcal{D})} \odot (\mathcal{L} * \mathcal{R} - \mathcal{X})\right\|_F^2 \\ & + \lambda\|\mathcal{E}\|_1 + g_{\mathcal{X}}(\mathcal{X}) + h_{\mathcal{E}}(\mathcal{E}) \\ \text{subject to} \quad & \mathcal{X} + \mathcal{E} + \mathcal{S} = \mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}), \\ & \|\mathcal{P}_{\mathbf{\Omega}}(\mathcal{S})\|_F \leq \delta. \end{aligned} \tag{11}$$

C. Solution to the Optimization

We solve the optimization problem in (11) using an iterative algorithm employing the ALM method [62]. Specifically, we first introduce auxiliary variables $\mathcal{G}$ and $\mathcal{H}$ for variable splitting. The optimization in (11) can then be rewritten as

$$\begin{aligned} \underset{\mathcal{L},\mathcal{R},\mathcal{X},\mathcal{E},\mathcal{S},\mathcal{G},\mathcal{H}}{\text{minimize}} \quad & \frac{1}{2}\left\|\sqrt{\mathcal{F}_{\text{att}}(\mathcal{D})} \odot (\mathcal{L} * \mathcal{R} - \mathcal{X})\right\|_F^2 \\ & + \lambda\|\mathcal{E}\|_1 + g_{\mathcal{X}}(\mathcal{G}) + h_{\mathcal{E}}(\mathcal{H}) \\ \text{subject to} \quad & \mathcal{G} = \mathcal{X}, \mathcal{H} = \mathcal{E}, \\ & \mathcal{X} + \mathcal{E} + \mathcal{S} = \mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}), \\ & \|\mathcal{P}_{\mathbf{\Omega}}(\mathcal{S})\|_F \leq \delta. \end{aligned} \tag{12}$$

Then, we define the augmented Lagrangian function $\mathcal{L}$ for (12) as

$$\begin{aligned} & \mathcal{L}(\mathcal{L}, \mathcal{R}, \mathcal{X}, \mathcal{E}, \mathcal{S}, \mathcal{G}, \mathcal{H}, \mathbf{\Lambda}, \mathbf{\Gamma}, \mathbf{\Phi}) \\ & = \frac{1}{2}\left\|\sqrt{\mathcal{F}_{\text{att}}(\mathcal{D})} \odot (\mathcal{L} * \mathcal{R} - \mathcal{X})\right\|_F^2 + \lambda\|\mathcal{E}\|_1 \\ & \quad + \frac{\mu}{2}\|\mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}) - \mathcal{X} - \mathcal{E} - \mathcal{S}\|_F^2 \\ & \quad + \langle \mathbf{\Lambda}, \mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}) - \mathcal{X} - \mathcal{E} - \mathcal{S}\rangle \\ & \quad + g_{\mathcal{X}}(\mathcal{G}) + \frac{\alpha}{2}\|\mathcal{X} - \mathcal{G}\|_F^2 + \langle \mathbf{\Gamma}, \mathcal{X} - \mathcal{G}\rangle \\ & \quad + h_{\mathcal{E}}(\mathcal{H}) + \frac{\beta}{2}\|\mathcal{E} - \mathcal{H}\|_F^2 + \langle \mathbf{\Phi}, \mathcal{E} - \mathcal{H}\rangle, \end{aligned} \tag{13}$$

where $\mathbf{\Lambda}$, $\mathbf{\Gamma}$, and $\mathbf{\Phi} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are tensors of Lagrange multipliers; μ, α, and $\beta > 0$ are penalty parameters; and $\langle \cdot, \cdot \rangle$ is the tensor inner product defined in (4).

We can obtain the optimal solutions to (12) by minimizing the augmented Lagrangian function $\mathcal{L}$ in (13), i.e.,

$$\begin{aligned} & (\mathcal{L}^*, \mathcal{R}^*, \mathcal{X}^*, \mathcal{E}^*, \mathcal{S}^*, \mathcal{G}^*, \mathcal{H}^*) \\ & = \underset{\mathcal{L},\mathcal{R},\mathcal{X},\mathcal{E},\mathcal{S},\mathcal{G},\mathcal{H}}{\arg\min} \mathcal{L}(\mathcal{L}, \mathcal{R}, \mathcal{X}, \mathcal{E}, \mathcal{S}, \mathcal{G}, \mathcal{H}, \mathbf{\Lambda}, \mathbf{\Gamma}, \mathbf{\Phi}). \end{aligned} \tag{14}$$

As it is intractable to solve the joint optimization problem in (14) directly, we employ the alternating direction method of multipliers [63] to split it into subproblems corresponding to each variable and multiplier. We then solve the subproblems separately through iterations. In the following, we provide a solution for each subproblem at the k-th iteration.

$\mathcal{X}$-subproblem: We first update $\mathcal{X}$ as

$$\begin{aligned} \mathcal{X}_{k+1} &= \arg\min_{\mathcal{X}} \frac{1}{2}\left\|\sqrt{\mathcal{F}_{\text{att}}(\mathcal{D})} \odot (\mathcal{L}_k * \mathcal{R}_k - \mathcal{X})\right\|_F^2 \\ &\quad + \frac{\mu_k}{2}\|\mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}) - \mathcal{X} - \mathcal{E}_k - \mathcal{S}_k\|_F^2 \\ &\quad + \langle \mathbf{\Lambda}_k, \mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}) - \mathcal{X} - \mathcal{E}_k - \mathcal{S}_k\rangle \\ &\quad + \frac{\alpha_k}{2}\|\mathcal{X} - \mathcal{G}_k\|_F^2 + \langle \mathbf{\Gamma}_k, \mathcal{X} - \mathcal{G}_k\rangle \\ &= \arg\min_{\mathcal{X}} \frac{1}{2}\left\|\sqrt{\mathcal{F}_{\text{att}}(\mathcal{D})} \odot (\mathcal{L}_k * \mathcal{R}_k - \mathcal{X})\right\|_F^2 \\ &\quad + \frac{\mu_k + \alpha_k}{2}\|\mathcal{X} - \mathbf{\Psi}_{\mathcal{X},k}\|_F^2, \\ &= \frac{\mathcal{F}_{\text{att}}(\mathcal{D}) \odot (\mathcal{L}_k * \mathcal{R}_k) + (\mu_k + \alpha_k)\mathbf{\Psi}_{\mathcal{X},k}}{\mathcal{F}_{\text{att}}(\mathcal{D}) + \mu_k \mathbf{1} + \alpha_k \mathbf{1}}, \end{aligned} \tag{15}$$

where $\mathbf{\Psi}_{\mathcal{X},k} = (\mu_k + \alpha_k)^{-1}(\mathbf{\Lambda}_k + \mu_k \mathcal{P}_{\mathbf{\Omega}}(\mathcal{D}) - \mu_k \mathcal{E}_k - \mu_k \mathcal{S}_k + \alpha_k \mathcal{G}_k - \mathbf{\Gamma}_k)$, $\mathbf{1}$ is a tensor of all ones, and the division is element-wise. In Appendix A, available online, we derive the closed-form solution in (15).

$\mathcal{L}$- and $\mathcal{R}$-subproblems: Then, we update $\mathcal{L}$ and $\mathcal{R}$ simultaneously by solving

$$\underset{\mathcal{L},\mathcal{R}}{\text{minimize}} \frac{1}{2}\left\|\sqrt{\mathcal{F}_{\text{att}}(\mathcal{D})} \odot (\mathcal{L} * \mathcal{R} - \mathcal{X}_{k+1})\right\|_F^2. \tag{16}$$

The solutions can be computed efficiently in the Fourier domain due to the periodic property of $\texttt{bcirc}(\cdot)$ [14]. Specifically, let $\widehat{\mathcal{X}}_{k+1}$ denote a tensor of the Fourier transform coefficients of $\mathcal{X}_{k+1}$ along the third dimension, i.e., $\widehat{\mathcal{X}}_{k+1}(i,j,:) = \mathcal{F}_{\text{FFT}}\{\mathcal{X}_{k+1}(i,j,:)\}$. Then, the closed-form solutions in the Fourier domain for $\widehat{\mathcal{L}}_{k+1}$ and $\widehat{\mathcal{R}}_{k+1}$ are given by those of their frontal slices as

$$\begin{aligned} \widehat{\mathcal{L}}_{k+1}^{(i)} &= \arg\min_{\widehat{\mathcal{L}}^{(i)}} \frac{1}{2n_3}\left\|\sqrt{\mathcal{F}_{\text{att}}^{(i)}(\mathcal{D})} \odot \left(\widehat{\mathcal{L}}^{(i)} * \widehat{\mathcal{R}}_k^{(i)} - \widehat{\mathcal{X}}_{k+1}^{(i)}\right)\right\|_F^2 \\ &= \widehat{\mathcal{X}}_{k+1}^{(i)}\left(\widehat{\mathcal{R}}_k^{(i)}\right)^H \left(\widehat{\mathcal{R}}_k^{(i)}\left(\widehat{\mathcal{R}}_k^{(i)}\right)^H\right)^{\dagger}, \end{aligned} \tag{17}$$

$$\begin{aligned} \widehat{\mathcal{R}}_{k+1}^{(i)} &= \arg\min_{\widehat{\mathcal{R}}^{(i)}} \frac{1}{2n_3}\left\|\sqrt{\mathcal{F}_{\text{att}}^{(i)}(\mathcal{D})} \odot \left(\widehat{\mathcal{L}}_k^{(i)} * \widehat{\mathcal{R}}^{(i)} - \widehat{\mathcal{X}}_{k+1}^{(i)}\right)\right\|_F^2 \\ &= \left(\left(\widehat{\mathcal{L}}_k^{(i)}\right)^H \widehat{\mathcal{L}}_k^{(i)}\right)^{\dagger} \left(\widehat{\mathcal{L}}_k^{(i)}\right)^H \widehat{\mathcal{X}}_{k+1}^{(i)}, \end{aligned} \tag{18}$$

where $\mathbf{A}^H$ and $\mathbf{A}^{\dagger}$ denote the conjugate transpose and the pseudo-inverse of matrix $\mathbf{A}$, respectively. Finally, tensors $\mathcal{L}_{k+1}$ and $\mathcal{R}_{k+1}$ can be obtained by performing an inverse Fourier transform as

$$\begin{aligned} \mathcal{L}_{k+1}(i,j,:) &= \mathcal{F}_{\text{FFT}}^{-1}\{\widehat{\mathcal{L}}_{k+1}(i,j,:)\}, \\ \mathcal{R}_{k+1}(i,j,:) &= \mathcal{F}_{\text{FFT}}^{-1}\{\widehat{\mathcal{R}}_{k+1}(i,j,:)\}. \end{aligned} \tag{19}$$

$\mathcal{E}$-subproblem: Next, we update $\mathcal{E}$ as

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{k+1} &= \arg\min_{\boldsymbol{\mathcal{E}}} \lambda_k\|\boldsymbol{\mathcal{E}}\|_1 + \langle \boldsymbol{\Lambda}_k, \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}) - \boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{S}}_k\rangle \\ &\quad + \frac{\mu_k}{2}\|\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}) - \boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{S}}_k\|_F^2 \\ &\quad + \frac{\beta_k}{2}\|\boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{H}}_k\|_{\mathrm{F}}^2 + \langle \boldsymbol{\Phi}_k, \boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{H}}_k\rangle \\ &= \arg\min_{\boldsymbol{\mathcal{E}}} \frac{\lambda_k}{\mu_k+\beta_k}\|\boldsymbol{\mathcal{E}}\|_1 + \frac{1}{2}\|\boldsymbol{\mathcal{E}} - \boldsymbol{\Psi}_{\boldsymbol{\mathcal{E}},k}\|_F^2 \\ &= \mathcal{T}_{\frac{\lambda_k}{\mu_k+\beta_k}}(\boldsymbol{\Psi}_{\boldsymbol{\mathcal{E}},k}), \end{aligned} \tag{20}$$

where $\boldsymbol{\Psi}_{\boldsymbol{\mathcal{E}},k} = (\mu_k+\beta_k)^{-1}(\boldsymbol{\Lambda}_k + \mu_k\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}) - \mu_k\boldsymbol{\mathcal{X}}_{k+1} - \mu_k\boldsymbol{\mathcal{S}}_k + \beta_k\boldsymbol{\mathcal{H}}_k - \boldsymbol{\Phi}_k)$, and $\mathcal{T}_\tau(\boldsymbol{\mathcal{A}})$ denotes the element-wise soft-thresholding operator [66] with parameter $\tau > 0$, i.e., $[\mathcal{T}_\tau(\boldsymbol{\mathcal{A}})](i_1,i_2,i_3) = \text{sign}(\boldsymbol{\mathcal{A}}(i_1,i_2,i_3)) \cdot \max\{|\boldsymbol{\mathcal{A}}(i_1,i_2,i_3)| - \tau, 0\}$.

$\boldsymbol{\mathcal{S}}$-subproblem: We update the tensor of slack variables $\boldsymbol{\mathcal{S}}$ as similarly done in [5], [61]

$$\begin{aligned}\boldsymbol{\mathcal{S}}_{k+1} &= \arg\min_{\|\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{S}})\|_F \le \delta_k} \frac{\mu_k}{2}\|\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}) - \boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{E}}_{k+1} - \boldsymbol{\mathcal{S}}\|_F^2 \\ &\quad + \langle \boldsymbol{\Lambda}_k, \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}) - \boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{E}}_{k+1} - \boldsymbol{\mathcal{S}}\rangle \\ &= \arg\min_{\|\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{S}})\|_F \le \delta_k} \|\boldsymbol{\mathcal{S}} - \boldsymbol{\Psi}_{\boldsymbol{\mathcal{S}},k}\|_F^2 \\ &= \mathcal{P}_{\boldsymbol{\Omega}^{\mathrm{C}}}(\boldsymbol{\Psi}_{\boldsymbol{\mathcal{S}},k}) + \min\left\{\frac{\delta_k}{\|\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\Psi}_{\boldsymbol{\mathcal{S}},k})\|_F}, 1\right\}\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\Psi}_{\boldsymbol{\mathcal{S}},k}), \end{aligned} \tag{21}$$

where $\boldsymbol{\Psi}_{\boldsymbol{\mathcal{S}},k} = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}) - \boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{E}}_{k+1} + \mu_k^{-1}\boldsymbol{\Lambda}_k$. The derivation of the closed-form solution in (21) is provided in Appendix B, available online.

$\boldsymbol{\mathcal{G}}$- and $\boldsymbol{\mathcal{H}}$-subproblems: We then update the auxiliary variables $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{H}}$ as

$$\begin{aligned}\boldsymbol{\mathcal{G}}_{k+1} &= \arg\min_{\boldsymbol{\mathcal{G}}} g_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{G}}) + \frac{\alpha_k}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}}\|_F^2 + \langle \boldsymbol{\Gamma}, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}}\rangle \\ &= \arg\min_{\boldsymbol{\mathcal{G}}} g_{\boldsymbol{\mathcal{X}}}(\boldsymbol{\mathcal{G}}) + \frac{\alpha_k}{2}\|\boldsymbol{\mathcal{G}} - (\boldsymbol{\mathcal{X}}_{k+1} + \alpha_k^{-1}\boldsymbol{\Gamma}_k)\|_F^2 \\ &= \text{prox}_{g_{\boldsymbol{\mathcal{X}}}}(\boldsymbol{\mathcal{X}}_{k+1} + \alpha_k^{-1}\boldsymbol{\Gamma}_k), \end{aligned} \tag{22}$$

$$\begin{aligned}\boldsymbol{\mathcal{H}}_{k+1} &= \arg\min_{\boldsymbol{\mathcal{H}}} h_{\boldsymbol{\mathcal{E}}}(\boldsymbol{\mathcal{H}}) + \frac{\beta_k}{2}\|\boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{H}}\|_F^2 + \langle \boldsymbol{\Phi}, \boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{H}}\rangle, \\ &= \arg\min_{\boldsymbol{\mathcal{H}}} h_{\boldsymbol{\mathcal{E}}}(\boldsymbol{\mathcal{H}}) + \frac{\beta_k}{2}\|\boldsymbol{\mathcal{H}} - (\boldsymbol{\mathcal{E}}_{k+1} + \beta_k^{-1}\boldsymbol{\Phi}_k)\|_F^2 \\ &= \text{prox}_{h_{\boldsymbol{\mathcal{E}}}}(\boldsymbol{\mathcal{E}}_{k+1} + \beta_k^{-1}\boldsymbol{\Phi}_k), \end{aligned} \tag{23}$$

where $\text{prox}_{g_{\boldsymbol{\mathcal{X}}}}(\cdot)$ and $\text{prox}_{h_{\boldsymbol{\mathcal{E}}}}(\cdot)$ are the proximal operators [67] corresponding to regularization functions $g_{\boldsymbol{\mathcal{X}}}$ and $h_{\boldsymbol{\mathcal{E}}}$, respectively. The regularization functions constrain the variables with specific characteristics to improve the accuracy of solutions, e.g., sparsity [16], smoothness [2], or statistical priors [3], [17]. However, in pure model-based algorithms, regularization functions are constructed by observing a small number of samples, which limits their generalizability. To address this limitation, we design regularization functions $g_{\boldsymbol{\mathcal{X}}}$ and $h_{\boldsymbol{\mathcal{E}}}$ to model a wide range of characteristics and properties of real-world data. To this end, we use CNNs to implement $\text{prox}_{g_{\boldsymbol{\mathcal{X}}}}(\cdot)$ and $\text{prox}_{h_{\boldsymbol{\mathcal{E}}}}(\cdot)$ so that they can effectively learn complex visual features from diverse training data. Let $\mathcal{F}_{\boldsymbol{\mathcal{G}}}$ and $\mathcal{F}_{\boldsymbol{\mathcal{H}}}$ be the CNN-based proximal operators for $\text{prox}_{g_{\boldsymbol{\mathcal{X}}}}(\cdot)$ and $\text{prox}_{h_{\boldsymbol{\mathcal{E}}}}(\cdot)$ in (22) and (23), respectively. Then, the solutions of $\boldsymbol{\mathcal{G}}$- and $\boldsymbol{\mathcal{H}}$-subproblems can be respectively rewritten as

$$\boldsymbol{\mathcal{G}}_{k+1} = \mathcal{F}_{\boldsymbol{\mathcal{G}}}\big(\boldsymbol{\mathcal{X}}_{k+1} + \alpha_k^{-1}\boldsymbol{\Gamma}_k; \boldsymbol{\Theta}_{\boldsymbol{\mathcal{G}},k}\big), \tag{24}$$

$$\boldsymbol{\mathcal{H}}_{k+1} = \mathcal{F}_{\boldsymbol{\mathcal{H}}}\big(\boldsymbol{\mathcal{E}}_{k+1} + \beta_k^{-1}\boldsymbol{\Phi}_k; \boldsymbol{\Theta}_{\boldsymbol{\mathcal{H}},k}\big), \tag{25}$$

where $\boldsymbol{\Theta}_{\boldsymbol{\mathcal{G}},k}$ and $\boldsymbol{\Theta}_{\boldsymbol{\mathcal{H}},k}$ are the parameters of networks $\mathcal{F}_{\boldsymbol{\mathcal{G}}}$ and $\mathcal{F}_{\boldsymbol{\mathcal{H}}}$ at the k-th iteration.

Multipliers: Finally, the tensors of Lagrange multipliers are updated as

$$\begin{aligned}\boldsymbol{\Lambda}_{k+1} &= \boldsymbol{\Lambda}_k + \mu_k(\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}}) - \boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{E}}_{k+1} - \boldsymbol{\mathcal{S}}_{k+1}), \\ \boldsymbol{\Gamma}_{k+1} &= \boldsymbol{\Gamma}_k + \alpha_k(\boldsymbol{\mathcal{X}}_{k+1} - \boldsymbol{\mathcal{G}}_{k+1}), \\ \boldsymbol{\Phi}_{k+1} &= \boldsymbol{\Phi}_k + \beta_k(\boldsymbol{\mathcal{E}}_{k+1} - \boldsymbol{\mathcal{H}}_{k+1}). \end{aligned} \tag{26}$$

D. Deep Unfolded Network

Finally, we construct a computationally efficient deep network by unfolding the iterations of the iterative AGTC algorithm presented in the previous section. Fig. 1 illustrates the architecture of the deep network for the proposed AGTC. The observed data tensor $\boldsymbol{\mathcal{D}}$ is first fed into the attention module to estimate the attention maps $\mathcal{F}_{\text{att}}(\boldsymbol{\mathcal{D}})$ of the important regions. Because the proposed algorithm is structure-agnostic, any architecture can be employed for the attention module. In this work, we employ the attention module from AHDRNet [64] in Fig. 1(c) for simplicity, which is defined for a tensor $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ as

$$\mathcal{F}_{\text{att}}(\boldsymbol{\mathcal{D}}) = \texttt{Sigmoid}(\texttt{Conv}(\texttt{LeakyReLU}(\texttt{Conv}(\boldsymbol{\mathcal{D}})))), \tag{27}$$

where $\texttt{Sigmoid}(\cdot)$ is the sigmoid activation function, $\texttt{LeakyReLU}(\cdot)$ is the leaky rectified linear unit, and $\texttt{Conv}(\cdot)$ is a convolution layer with n_3 input and output channels, 3×3 kernels, a stride of 1, and padding of 1. Therefore, the output of $\mathcal{F}_{\text{att}}(\cdot)$ has the same dimension as that of $\boldsymbol{\mathcal{D}}$. The effects of different architectures for the attention module on the performance will be discussed in Section IV-C1. Then, the observed data tensor $\boldsymbol{\mathcal{D}}$ and attention maps $\mathcal{F}_{\text{att}}(\boldsymbol{\mathcal{D}})$ are fed into the deep unfolded network consisting of K consecutive stages. The operations in each stage in the network correspond to obtaining solutions in an iteration of the proposed iterative AGTC algorithm. Specifically, in each stage, the optimization variables, i.e., tensors, are updated in two phases: reconstruction and regularization.

First, variables $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{G}}$, and $\boldsymbol{\mathcal{H}}$ and the Lagrange multipliers $\boldsymbol{\Lambda}, \boldsymbol{\Gamma}$, and $\boldsymbol{\Phi}$ are initialized to zero tensors, while $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{R}}$ are initialized as tensors of 10^{-2}. Then, variables $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{E}}$, and $\boldsymbol{\mathcal{S}}$ are updated in the reconstruction phase by the closed-form solutions in (15), (17), (18), (20), and (21), respectively. The solution for $\boldsymbol{\mathcal{X}}_{k+1}$ in (15) is computationally expensive as it requires the t-product $\boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k$, which involves time- and memory-consuming operations, i.e., $\texttt{bcirc}(\cdot)$ and $\texttt{unfold}(\cdot)$. To overcome this issue, we exploit the properties of the Fourier transform to efficiently compute $\boldsymbol{\mathcal{X}}_{k+1}$ as in [14]. More specifically, let $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{L}}_k * \boldsymbol{\mathcal{R}}_k$, and first compute the Fourier transform along the third dimension of $\boldsymbol{\mathcal{L}}_k$ and $\boldsymbol{\mathcal{R}}_k$; then, the i-th slice of

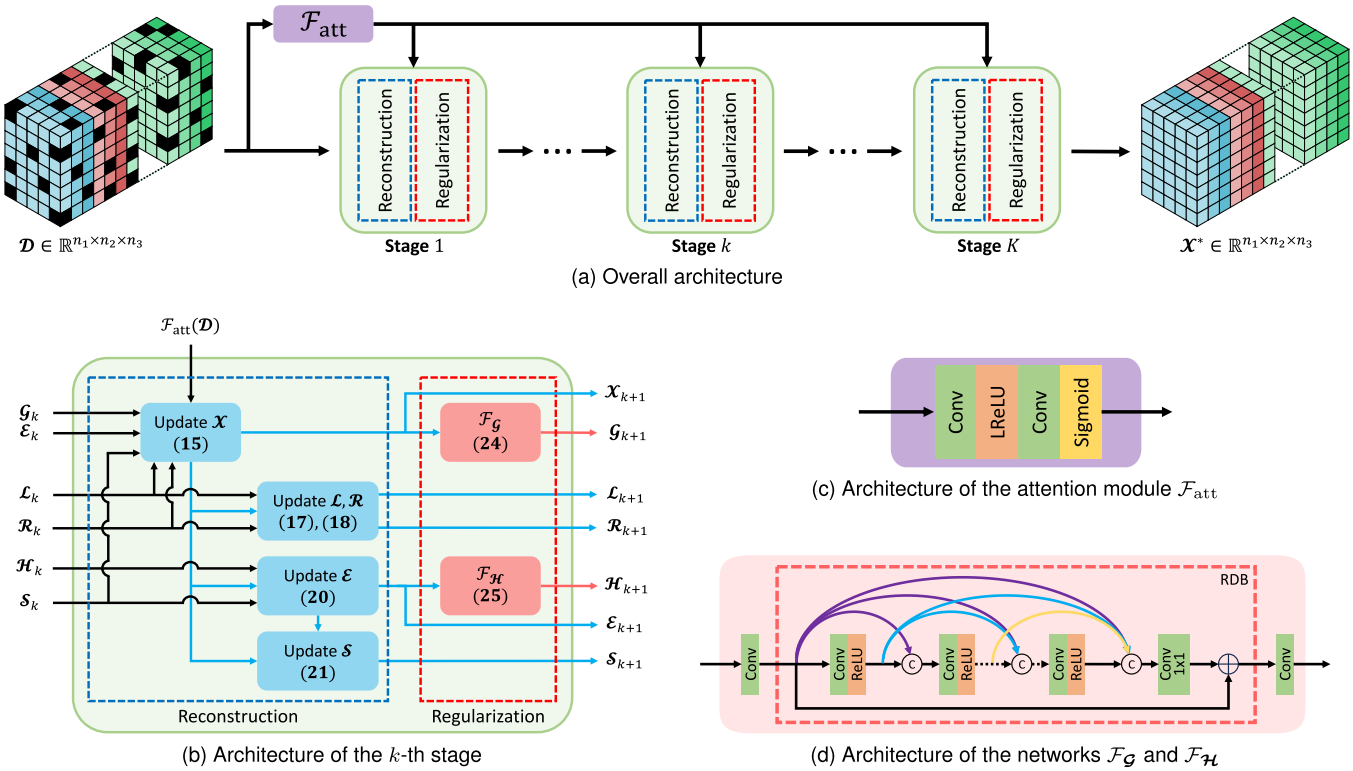


Fig. 1. Architecture of the AGTC network. (a) The network consists of K stages and an attention module for attention-guided restoration. (b) Each stage of the network corresponds to an iteration of the iterative LRTC algorithm. In each stage, the optimization variables are updated in two phases: the reconstruction phase by closed-form solutions and the regularization phase by CNNs. Architectures of (c) the attention module [64] and (d) the networks $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$ [65].

the t-product in the Fourier domain is given by

$$\widehat{\mathcal{Y}}^{(i)} = \widehat{\mathcal{L}}_k^{(i)} \widehat{\mathcal{R}}_k^{(i)}. \tag{28}$$

Because the slices can be computed independently, we perform parallel processing to reduce execution time. Finally, we obtain the t-product by performing an inverse Fourier transform on $\widehat{\mathcal{Y}}$, i.e.,

$$\mathcal{Y}(i,j,:) = \mathcal{F}_{\text{FFT}}^{-1}\{\widehat{\mathcal{Y}}(i,j,:)\}. \tag{29}$$

Next, auxiliary variables $\mathcal{G}$ and $\mathcal{H}$ are updated in the regularization phase by the CNNs in (24) and (25), respectively. Similar to $\mathcal{F}_{\text{att}}(\cdot)$, any architecture can be used because the CNNs $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$ are structure-agnostic. Also, note that, as the proposed AGTC takes tensors of the original shape as input, the CNNs $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$ can effectively learn the spatial dependencies. In this work, for simplicity, we employ a residual dense block (RDB) [65]. Fig. 1(d) shows the architecture of the networks $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$. Specifically, it comprises an RDB and two convolution layers. The RDB has 64 initial feature maps, a growth rate of 32, and N convolution layers. The first convolution layer has n_3 input and 64 out channels, 3×3 kernels, a stride of 1, and padding of 1, while the last convolutional layer has 64 input and n_3 output channels.

In addition to the CNNs' parameters $\boldsymbol{\Theta}_{\mathcal{G},k}$ and $\boldsymbol{\Theta}_{\mathcal{H},k}$, the penalty parameters μ_k in (15), (20), and (21), α_k in (15) and (22), β_k in (20) and (23), λ_k in (20), and the noise level δ_k in (21) are also learned from the data, i.e., implemented as trainable parameters and adjusted by backpropagation during training. Finally, the output of the last stage is the reconstructed low-rank tensor, i.e., $\mathcal{X}^* = \mathcal{X}_K$.

IV. Experimental Results

We evaluate the performance of the proposed AGTC algorithm in two applications—HDR imaging and HSI restoration. For reproducibility, we provide the source code and pretrained models on our project website.[1]

A. HDR Imaging

1) Settings: In this experiment, we apply the proposed AGTC algorithm to HDR imaging, in which multiple low dynamic range (LDR) images are fused to synthesize single HDR images without ghosting artifacts. We evaluate the performance of the AGTC and competing algorithms using the HDM-HDR dataset [73] preprocessed in [5], which contains 187 image sets—132 for training and 55 for testing. Each set contains three images with exposure biases of $\{-3, 0, +3\}$, of which the middle image is the reference.

Given a set of three LDR images $\{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3\}$, $\mathcal{I}_i \in \mathbb{R}^{n_1 \times n_2 \times 3}$, we first convert the pixel values of $\mathcal{I}_i$ to the irradiance

[1] https://github.com/mtntruong/AGTC

TABLE I
QUANTITATIVE EVALUATION OF HDR IMAGE SYNTHESIS PERFORMANCE ON THE HDM-HDR DATASET

	μ-PSNR (↑)	PU21-PSNR (↑)	PU21-MSSSIM (↑)	PU21-VSI (↑)	HDR-VDP (Q) (↑)	HDR-VDP (P) (↓)
TNNM-ALM [61]	33.68	28.30	0.9723	0.9851	57.28	0.3772
K. and R. [68]	38.21	30.19	0.9917	0.9819	57.58	0.2211
Wu *et al.* [69]	43.49	40.75	0.9984	0.9979	66.36	0.2520
AHDRNet [64]	40.10	38.27	0.9979	0.9970	64.18	0.4456
ADNet [70]	47.61	42.01	0.9991	0.9982	66.90	0.0457
Mai *et al.* [71]	48.94	41.08	0.9984	0.9981	66.84	0.0339
LRT-HDR [5]	49.54	41.98	0.9991	0.9984	68.11	0.0273
CA-ViT [72]	49.53	42.18	0.9992	0.9981	68.68	0.0252
Proposed (AGTC)	**50.09**	**42.86**	**0.9993**	**0.9987**	**69.20**	**0.0187**

The ↑ and ↓ symbols denote "higher is better" and "lower is better," respectively. For each metric, the **boldfaced** and underlined numbers denote the best and the second-best results, respectively.

values using the gamma correction function as

$$\boldsymbol{\mathcal{H}}_i = \frac{\boldsymbol{\mathcal{I}}_i^{\gamma}}{t_i}, \quad i = 1, 2, 3, \tag{30}$$

where t_i is the exposure time of the i-th LDR image, and we set $\gamma = 2.2$ for consistency with previous works [5], [64], [68], [69], [70], [71]. Then, we align $\boldsymbol{\mathcal{H}}_i$ with the reference image using SIFT-Flow [74] and concatenate them along the color channel dimension to construct the input data tensor $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1 \times n_2 \times 9}$. We define the set of observed entries as $\boldsymbol{\Omega} = \boldsymbol{\Omega}_1 \cap \boldsymbol{\Omega}_2$, where

$$\boldsymbol{\Omega}_1 = \{(j,k,l) | 0.01 \leq \boldsymbol{\mathcal{I}}_i(j,k,l) \leq 0.99\}, \tag{31}$$

$$\boldsymbol{\Omega}_2 = \{(j,k,l) | \texttt{SSIM}(\boldsymbol{\mathcal{H}}_i'(j,k,l), \boldsymbol{\mathcal{H}}_{\text{ref}}'(j,k,l)) \geq 0.90\} \tag{32}$$

are the sets of well-exposed and well-aligned pixels, respectively; $\texttt{SSIM}(\cdot)$ denotes the pixel-wise structural similarity index (SSIM) [75], and $\boldsymbol{\mathcal{H}}_i'$ denotes the aligned images.

Tensors $\boldsymbol{\mathcal{D}}$ and $\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}})$ are then fed into the proposed deep unfolded network with $K = 10$ stages. The CNN-based proximal operators, $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$, are both constructed using RDBs with $N = 8$ convolutional layers. We set the desired tubal rank to $r = 3$, the number of color channels. A 1×1 convolution layer with nine input channels and three output channels is used at the end of the network to synthesize the final HDR image $\boldsymbol{\mathcal{H}}^* \in \mathbb{R}^{n_1 \times n_2 \times 3}$ from the output tensor $\boldsymbol{\mathcal{X}}^* \in \mathbb{R}^{n_1 \times n_2 \times 9}$.

We use the settings in [5] to train the proposed AGTC algorithm for HDR imaging. Specifically, we employ the losses in rank fidelity and synthesis fidelity, which are computed in the perceptually uniform (PU) domain [76] as

$$\begin{aligned} L &= 0.5 L_{\text{rank}} + 0.5 L_{\text{syn}} \\ &= 0.5 \| l(\boldsymbol{\mathcal{X}}^*) - l(\texttt{Concat}(\boldsymbol{\mathcal{H}}_{\text{gt}}, \boldsymbol{\mathcal{H}}_{\text{gt}}, \boldsymbol{\mathcal{H}}_{\text{gt}})) \|_1 \\ &\quad + 0.5 \| l(\boldsymbol{\mathcal{H}}^*) - l(\boldsymbol{\mathcal{H}}_{\text{gt}}) \|_1, \end{aligned} \tag{33}$$

where $l(\cdot)$ denotes the luma function that maps the irradiance values to the PU domain [76]. The training samples are constructed by randomly cropping 256×256 patches from the images in the training dataset without augmentation. We use the Adam optimizer [77] for 50 epochs with an initial learning rate of $\eta = 10^{-5}$. We decrease the learning rate by a factor of 0.1 at every tenth epoch.

We compare the performance of AGTC in HDR synthesis with those of TNNM-ALM [61], Kalantari and Ramamoorthi's algorithm [68], Wu et al.'s algorithm [69], AHDRNet [64], ADNet [70], Mai et al.'s algorithm [71], LRT-HDR [5], and CA-ViT [72]. The results of the competing algorithms were obtained using the implementations provided by the respective authors with default settings. For qualitative comparison, the HDR images are tone-mapped using Reinhard and Devlin's algorithm [78]. For a quantitative comparison, we employ six metrics for HDR images: PSNR in the tone-mapped domain using μ-law [68] (μ-PSNR), PSNR, multi-scale SSIM, visual saliency-based index in the PU domain (PU21-PSNR, PU21-MSSSIM, PU21-VSI) [79], quality score Q and probability score P from the HDR visible difference predictor (HDR-VDP-Q and HDR-VDP-P) [80]. HDR-VDP scores are computed with the parameters of a 24-inch display and 0.5 m viewing distance.

2) Evaluation: Table I quantitatively compares the HDR synthesis performance of the algorithms on the HDM-HDR dataset. The proposed algorithm outperforms the conventional algorithms in all metrics. Specifically, the proposed algorithm achieves 0.55 and 0.68 dB higher μ-PSNR and PU21-PSNR scores than the second-best algorithms LRT-HDR [5] and CA-ViT [72], respectively, indicating that the HDR images synthesized by AGTC are the most faithful to the ground-truths. Furthermore, the proposed algorithm provides the best PU21-MSSSIM and PU21-VSI scores, indicating that the textures of the restored HDR images are most similar to the ground-truths with minimal differences. Finally, the proposed algorithm achieves the highest HDR-VDP (Q) and the lowest HDR-VDP (P) scores, 0.52 higher and lower by 0.0065, respectively, than the second-best algorithm CA-ViT [72], implying that the HDR images synthesized by the proposed algorithm show the lowest perceptual differences with the ground-truths.

Fig. 2 compares the synthesized HDR images obtained by each algorithm for the 13th image set, in which the over-exposed regions in the reference image are occluded, as shown in the red and blue rectangles in Fig. 2(a). In Fig. 2(b), TNNM-ALM [61] produces artifacts because its formulation does not contain regularization functions, yielding information loss in large regions. In Fig. 2(c), Kalantari and Ramamoorthi's algorithm [68] fails to restore over-exposed regions because their network learns the blending weights of aligned images, which may fail when

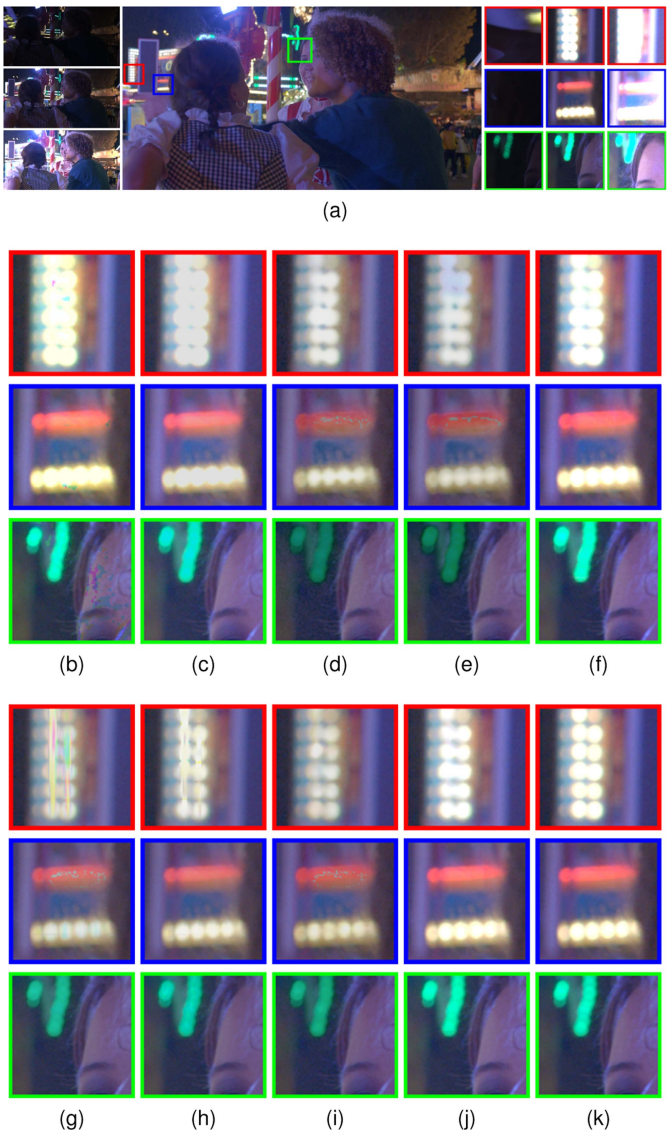


Fig. 2. Comparison of results for the 13th image of the HDM-HDR dataset. (a) LDR inputs and result of the proposed algorithm. The magnified regions marked by red, blue, and green rectangles of the synthesized results of (b) TNNM-ALM [61], (c) Kalantari and Ramamoorthi [68], (d) Wu et al. [69], (e) AHDRNet [64], (f) ADNet [70], (g) Mai et al. [71], (h) LRT-HDR [5], (i) CA-ViT [72], (j) the proposed algorithm, and (k) ground-truth.

the aligned images do not contain sufficient information. Wu et al.'s algorithm [69] and AHDRNet [64] in Fig. 2(d) and (e), respectively, lose the over-exposed details in the red and green rectangles and yield noticeable artifacts in the blue rectangles. ADNet [70] in Fig. 2(f) produces better results; however, the details of the light bulbs are lost because the CNNs fail to extract useful visual features for inference from the occluded regions. Mai et al.'s algorithm [71] and LRT-HDR [5] in Fig. 2(g) and (h), respectively, yield distinctive vertical patterns in the red rectangles because neither of their formulations considers the structure of the original data. In Fig. 2(i), CA-ViT [72] fails to restore the textures and colors of the light bulbs and produces visible artifacts because it relies heavily on the learned features, and the occluded regions prevent it from extracting useful

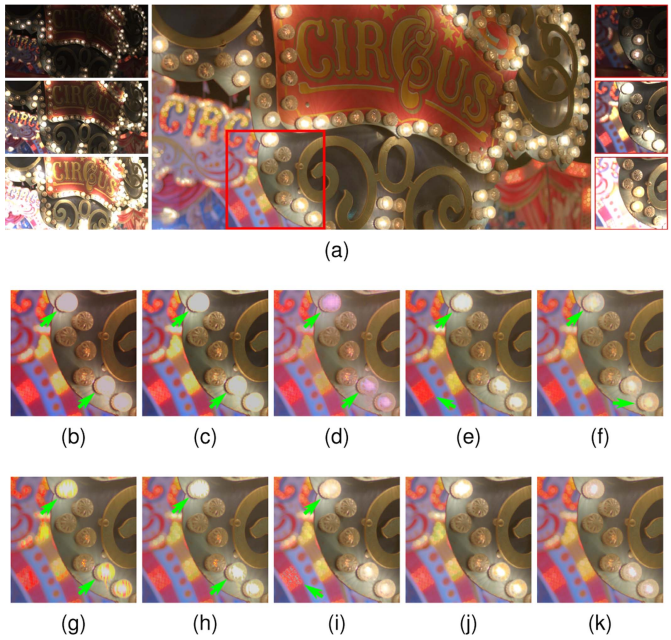


Fig. 3. Comparison of results for the 5th image of the HDM-HDR dataset. (a) LDR inputs and result of the proposed algorithm. The magnified regions marked by red rectangles of the synthesized results of (b) TNNM-ALM [61], (c) Kalantari and Ramamoorthi [68], (d) Wu et al. [69], (e) AHDRNet [64], (f) ADNet [70], (g) Mai et al. [71], (h) LRT-HDR [5], (i) CA-ViT [72], (j) the proposed algorithm, and (k) ground-truth.

features for accurate inference. In contrast, the proposed algorithm in Fig. 2(j) synthesizes an HDR image that is the most similar to the ground-truth; this is because the proposed attention-guided low-rank model effectively preserves the original structure of the data tensor and provides a robust constraint for handling occlusions.

Fig. 3 compares the synthesized HDR images for the 5th image set obtained by each algorithm, in which the states of the light bulbs change rapidly between exposures. The light bulbs are over-exposed or turned off in all exposures, resulting in complete information loss in large regions. Consequently, TNNM-ALM and Kalantari and Ramamoorthi's algorithm in Fig. 3(b) and (c), respectively, inaccurately restore the colors and textures of the light bulbs. This is because TNNM-ALM lacks regularizers to compensate for information loss, whereas the blending approach of Kalantari and Ramamoorthi's algorithm fails when the images contain insufficient information. The pure CNN-based algorithms fail to synthesize high-quality images because they cannot extract useful visual features from scenes with rapid changes. For example, Wu et al.'s algorithm in Fig. 3(d) produces severe artifacts. Although AHDRNet and ADNet provide better results, as shown in Fig. 3(e) and (f), respectively, the colors and textures still differ significantly from those of the ground-truth. The results of Mai et al.'s algorithm and LRT-HDR in Fig. 3(g) and (h), respectively, exhibit vertical patterns caused by the lack of consideration of the original tensor structures. CA-ViT in Fig. 3(i) yields severe artifacts in the red light bulbs because the transformer architecture fails to capture the spatial dependencies due to the rapid changes in the light bulbs. By contrast, the proposed algorithm successfully restores

TABLE II
QUANTITATIVE EVALUATION OF HSI RESTORATION PERFORMANCE ON THE PAVIA UNIVERSITY AND THE LANDSAT 7 ETM+ DATASETS

	Pavia University				Landsat 7 ETM+			
	PSNR (↑)	SSIM (↑)	ERGAS (↓)	SAM (↓)	PSNR (↑)	SSIM (↑)	ERGAS (↓)	SAM (↓)
DHP [81]	40.83	0.9718	39.4249	0.0449	40.25	0.9487	31.8698	0.0149
T3SC [82]	40.80	0.9886	38.3141	0.0464	38.91	0.9589	30.3982	0.0110
HTNN-DCT [60]	32.37	0.9059	93.9259	0.0999	42.07	0.9684	27.5952	**0.0058**
TV-WRT [83]	36.82	0.9593	58.6665	0.0533	38.44	0.9419	44.4469	0.0091
MGLRTA [84]	41.91	0.9844	34.1100	0.0302	41.85	0.9688	27.8182	**0.0058**
LNOP [85]	31.46	0.9229	109.4982	0.0872	39.25	0.9523	33.6259	0.0073
LPRN [86]	40.67	0.9794	41.8059	0.0318	40.85	0.9615	30.4398	0.0065
Proposed (AGTC)	**42.83**	**0.9923**	**31.3981**	**0.0285**	**42.26**	**0.9724**	**27.0021**	0.0063

the HDR image with the most faithful colors and textures, similar to the ground-truth.

B. Hyperspectral Image Restoration

1) Settings: In this experiment, we apply the proposed AGTC algorithm to HSI restoration, which aims to recover HSIs degraded by stripe noise or missing data. We evaluate the performance of AGTC and competing algorithms using the Pavia University [87] and Landsat 7 ETM+ [88] datasets.

- Pavia University [87] (synthetic data): It contains an HSI with a size of $610 \times 340 \times 103$. We use the settings in [84] to construct training and test samples. Specifically, we crop the top-left $256 \times 256 \times 103$ pixels of the original HSI; then, we randomly select and independently add stripe noise to 50% of the image columns in each band to synthesize a degraded test image. In addition, the 50th, 51st, 100th, and 200th columns of all the bands are discarded to simulate complete data loss. To construct training data, we randomly crop 64×64 patches from the remainder of the original HSI and then add stripe noise in the same way.
- Landsat 7 ETM+ [88] (real data): It contains two pairs of HSIs—one pair for training and one pair for testing with sizes of $1501 \times 1401 \times 8$ and $512 \times 512 \times 8$, respectively. Each pair contains a degraded HSI exhibiting oblique data gaps due to scan line corrector failure and a ground-truth HSI. As the degradation is due to a real-world failure, we regard this as a real-world dataset. We randomly crop 256×256 patches from the training pairs to construct the training data.

For both experiments, we augment training samples by randomly flipping and rotating them. The degraded tensors $\boldsymbol{\mathcal{D}}$ and $\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{D}})$ are fed into the deep unfolded network with $K = 10$ stages. We use RDBs with $N = 8$ and $N = 6$ convolutional layers for the CNN-based proximal operators $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$, respectively, to provide the best quantitative performance. We set the desired tubal rank to $r = 10$ and $r = 2$ for the Pavia University and Landsat 7 ETM+ datasets, respectively, as suggested by the ratio between the tubal rank and the number of spectral bands in [89]. We define the loss function as the ℓ_1-norm between the output tensor and the ground-truth. The Adam optimizer is used to train the network for 100 epochs with an initial learning rate of $\eta = 10^{-5}$, which is decreased by a factor of 0.1 at every twentieth epoch.

We compare the performance of AGTC in HSI restoration with those of DHP [81], T3SC [82], HTNN-DCT [60], TV-WRT [83], MGLRTA [84], LNOP [85], and LPRN [86]. The results of the competing algorithms were obtained using the implementations provided by the respective authors with default settings. For qualitative comparison, we show pseudocolor images composed of three representative bands. For a quantitative comparison, we employ four metrics for the HSI restoration: PSNR, SSIM [75], dimensionless global relative error of synthesis (ERGAS) [90], and spectral angle mapper (SAM) [91].

2) Evaluation: Table II quantitatively compares the HSI restoration performances of the algorithms on the Pavia University and Landsat 7 ETM+ datasets. The proposed algorithm outperforms conventional algorithms by large margins in terms of PSNR, SSIM, and ERGAS on both datasets while achieving the highest and second-highest SAM scores on the Pavia University and Landsat 7 ETM+ datasets, respectively. For both datasets, the proposed algorithm achieves the highest PSNR and SSIM scores, indicating that the HSIs recovered by the proposed algorithm are the most similar to the ground-truths. The ERGAS metric computes the normalized average error of each band in the restored images; thus, the highest ERGAS scores of the proposed algorithm for both datasets indicate that it can faithfully restore all HSI bands. The SAM metric computes the angle differences between spectra to measure their similarity. The proposed algorithm achieves the highest SAM on the Pavia University dataset and the second-highest SAM on the Landsat 7 ETM+ dataset; however, the difference is marginal. The quantitative evaluation confirms the effectiveness of the proposed algorithm over conventional algorithms for HSI restoration.

Fig. 4 compares the restored HSIs obtained by each algorithm for the Pavia University dataset. The pseudocolor images are synthesized from the 33rd, 75th, and 68th bands. DHP [81] and T3SC [82] in Fig. 4(b) and (c), respectively, produce better results than the conventional algorithms. The restored HSIs are close to the ground-truth without noticeable degradation. However, the restored details in the complete loss regions are still not faithful to the ground-truth, as shown in the error maps. HTNN-DCT [60] in Fig. 4(d) yields a blurry result with severe artifacts. TV-WRT [83], MGLRTA [84], LNOP [85], and

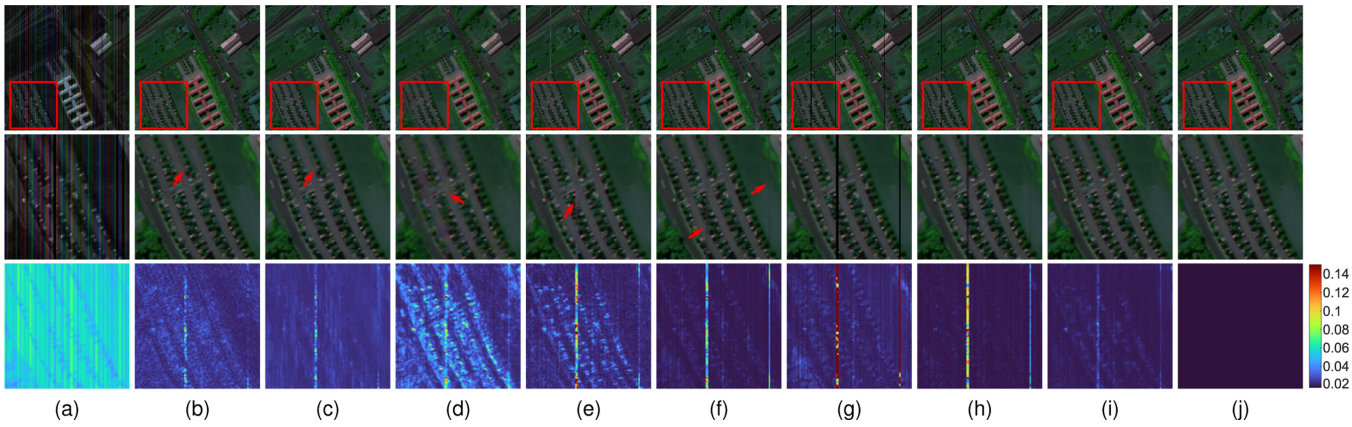


Fig. 4. Comparison of results on the Pavia University dataset. (a) Input and restored results of (b) DHP [81], (c) T3SC [82], (d) HTNN-DCT [60], (e) TV-WRT [83], (f) MGLRTA [84], (g) LNOP [85], (h) LPRN [86], (i) the proposed algorithm, and (j) ground-truth. The second and third rows show the magnified regions marked by the red rectangle in the first row and their error maps, respectively. The pseudocolor images are synthesized from the 33rd, 75th, and 68th bands.

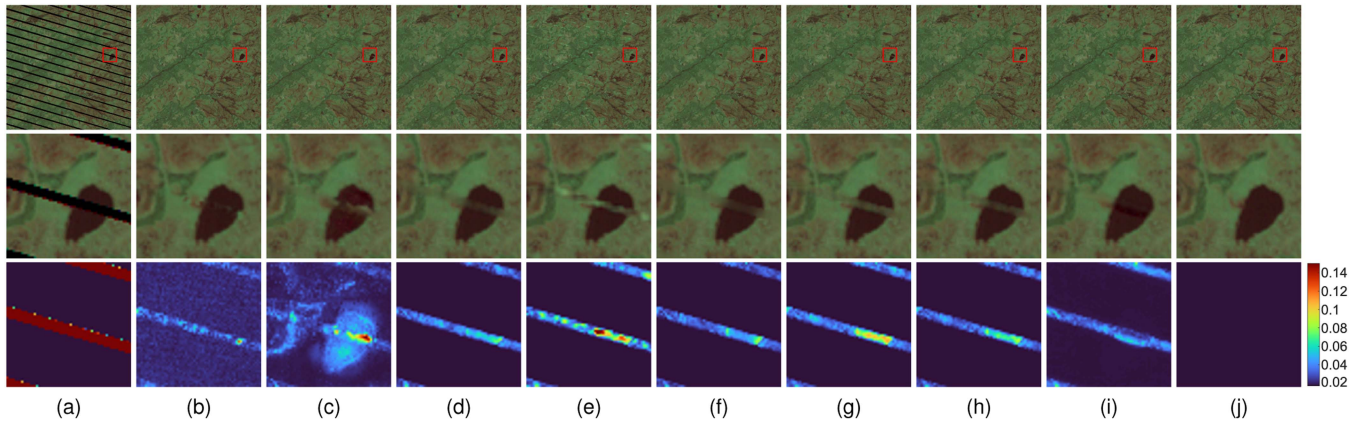


Fig. 5. Comparison of results on the Landsat 7 ETM+ dataset. (a) Input and restored results of (b) DHP [81], (c) T3SC [82], (d) HTNN-DCT [60], (e) TV-WRT [83], (f) MGLRTA [84], (g) LNOP [85], (h) LPRN [86], (i) the proposed algorithm, and (j) ground-truth. The second and third rows show the magnified regions marked by the red rectangle in the first row and their error maps, respectively. The pseudocolor images are synthesized from the 3rd, 5th, and 8th bands.

LPRN [86] fail to recover the information in the complete loss regions, resulting in artifacts in the form of strong vertical lines, as shown in Fig. 4(e)–(h). In contrast, the proposed algorithm in Fig. 4(i) successfully restores the HSI with the most faithful textures compared to the ground-truth, including information in the complete loss regions.

Fig. 5 compares the restored HSI obtained by each algorithm for the Landsat 7 ETM+ dataset. The pseudocolor images are synthesized from the 3rd, 5th, and 8th bands for illustration. T3SC, HTNN-DCT, TV-WRT, MGLRTA, LNOP, and LPRN in Fig. 5(c)–(h) fail to recover information in the data gap regions, resulting in severe texture degradation. DHP in Fig. 5(b) yields a better result, indicating that it is more similar to the ground-truth. However, the restored HSI still exhibits visibly distorted textures, which degrades its quality. By contrast, the HSI recovered by the proposed algorithm is the most natural and yields minimal visual differences from the ground-truth, as shown in Fig. 5(i). This comparison demonstrates the effectiveness of the proposed attention-guided formulation in preserving the original structures of the tensors.

C. Model Analysis

In this section, we perform ablation studies to analyze the effects of $\mathcal{F}_{\text{att}}$, the network structures of $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{G}}$, the number of unfolded iterations K, and the quantity of training data on the performance of the proposed algorithm in HDR imaging. We also analyze the computational complexity.

1) Learned Attention $\mathcal{F}_{att}$: We analyze the effects of the attention $\mathcal{F}_{\text{att}}$ on preserving the structure of the original data tensors. Table III quantitatively compares the HDR synthesis performances of the proposed algorithms with different modules for $\mathcal{F}_{\text{att}}$. First, the proposed AGTC algorithm with the attention module from AHDRNet [64] significantly improves scores by large margins in all metrics. Second, more sophisticated attention modules, CBAM [92], FFA [93], and SimAM [94], provide better performances at the expense of increased complexity. Third, instead of using the attention module, increasing the number of convolutional layers N in the RDB in $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$ to 10 and 12 (RDB-10 and RDB-12), respectively, improves the synthesis performance by increasing the amount of learned information, thus compensating for the lack of the attention module. However,

TABLE III
EFFECTS OF DIFFERENT MODULES FOR $\mathcal{F}_{\text{ATT}}$ ON THE SYNTHESIS PERFORMANCE

$\mathcal{F}_{\text{att}}$	μ-PSNR	PU-PSNR	PU-MSSSIM	PU-VSI	# Param.
None	48.71	41.83	0.9990	0.9984	8.73M
None (RDB-10)	49.58	42.14	0.9991	0.9985	12.69M
None (RDB-12)	49.43	42.41	0.9991	0.9986	17.38M
AHDRNet [64]	50.09	42.86	**0.9993**	0.9987	8.74M
CBAM [92]	51.22	43.64	**0.9993**	0.9988	8.74M
FFA [93]	50.75	43.07	0.9991	0.9986	8.74M
SimAM [94]	**51.62**	**43.97**	**0.9993**	**0.9989**	8.75M

Fig. 6. Effects of the attention-guided formulation for LRTC. (a) Ground-truth, and the synthesized results (b) without $\mathcal{F}_{\text{att}}$ and (c) with $\mathcal{F}_{\text{att}}$.

despite the significant increase in the number of parameters, the performance remains inferior to using the attention module. The results confirm the effectiveness of the attention mechanism for attention-guided LRTC and demonstrate that the proposed AGTC requires simple architectures to achieve state-of-the-art performance. In addition, the choice of the attention module significantly affects the performance. Therefore, the architecture of the attention module can be selected adaptively depending on applications or design choices, e.g., the trade-off between performance and computational cost. In this work, we employ the attention module from AHDRNet [64] as the baseline for its conceptual simplicity and to demonstrate the effectiveness of the attention mechanism for LRTC.

Fig. 6 qualitatively compares HDR synthesis results obtained using different settings. As shown in Fig. 6(b), the reconstructed HDR image without the attention mechanism exhibits severe color and texture artifacts, whereas the attention $\mathcal{F}_{\text{att}}$ improves the visual quality, producing more accurate results in Fig. 6(c). Both quantitative and qualitative comparisons confirm the effectiveness of the attention mechanism $\mathcal{F}_{\text{att}}$ in improving the performance of LRTC. Next, we analyze the effectiveness of the attention mechanism by visualizing attention maps. Fig. 7 compares the input LDR images and their corresponding attention maps estimated by $\mathcal{F}_{\text{att}}$. The attention maps exhibit higher values at well-exposed regions in each LDR image, e.g., the light bulbs in the short exposure in Fig. 7(a) and the background in the long exposure in Fig. 7(c). This implies that the optimization process focuses more on reconstructing those salient regions, thereby preserving textures and details across all exposures, leading to accurate synthesis.

2) Network Structures of $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$: As mentioned in Section III-D, any architecture can be used for the CNN-based

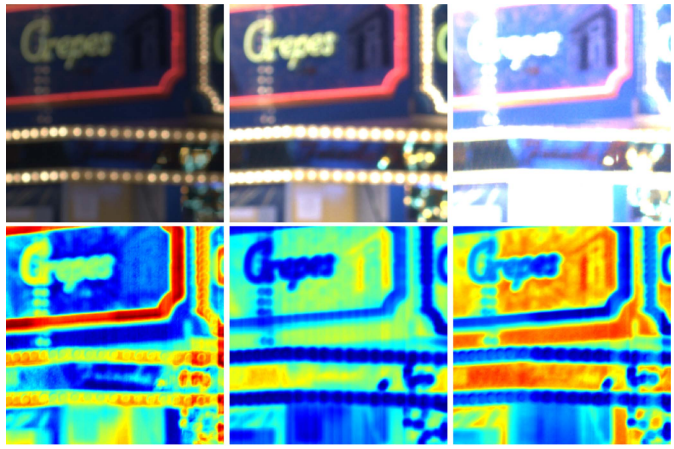


Fig. 7. Visualization of attention maps. LDR images (top row) and their corresponding attention maps (bottom row) of (a) short, (b) medium, and (c) long exposures.

TABLE IV
EFFECTS OF ARCHITECTURES FOR $\mathcal{F}_{\mathcal{G}}$ AND $\mathcal{F}_{\mathcal{H}}$ ON THE SYNTHESIS PERFORMANCE

	μ-PSNR	PU-PSNR	PU-MSSSIM	PU-VSI	# Param.
RDB [65]	50.09	42.86	**0.9993**	**0.9987**	8.74M
Conv+ReLU [5]	49.93	**43.18**	**0.9993**	**0.9987**	18.11M
UNetRes [55]	**50.28**	42.58	**0.9993**	0.9986	40.85M

TABLE V
EFFECTS OF THE NUMBERS OF CONVOLUTIONAL LAYERS IN THE RDB IN $\mathcal{F}_{\mathcal{G}}$ AND $\mathcal{F}_{\mathcal{H}}$ ON THE SYNTHESIS PERFORMANCE IN PU21-PSNR

$\mathcal{F}_{\mathcal{G}}$ \ $\mathcal{F}_{\mathcal{H}}$	6	7	8	9	10
6	42.11	42.19	41.12	41.97	41.94
7	41.37	42.32	41.66	42.69	42.45
8	42.52	42.18	42.86	41.48	42.35
9	41.85	41.46	41.90	42.84	41.55
10	41.52	**42.87**	40.57	40.67	41.93

regularizers $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$. We analyze the effects of their structures on performance by using three commonly used networks as regularizers: the proximal operator in LRT-HDR [5], which comprises a series of convolution and ReLU layers (Conv+ReLU), RDB [65], and UNetRes [55]. Table IV shows that the performance differences among different architectures are insignificant, despite differences in complexity. Specifically, although the RDB provides the best overall performance, albeit with marginal gains, it requires the smallest number of network parameters. Therefore, we use the RDB in this work for the best performance and complexity trade-off.

Next, we analyze the impact of the architectures of the RDB on the performance by training the network with different combinations of numbers of convolutional layers N in the RDB in $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$. Table V shows that the performance of the proposed network is significantly affected by the numbers of parameters in RDBs. In general, as the number of convolutional

TABLE VI
EFFECTS OF UNFOLDED ITERATION NUMBERS ON THE SYNTHESIS PERFORMANCE

K	μ-PSNR	PU21-PSNR	PU21-MSSSIM	PU21-VSI	HDR-VDP (Q)
5	48.14	40.77	0.9988	0.9980	67.63
10	**50.09**	42.86	**0.9993**	**0.9987**	**69.20**
15	49.75	**42.93**	0.9992	**0.9987**	69.16
20	49.16	42.55	0.9990	0.9986	69.00

TABLE VII
COMPARISON OF APPROXIMATE NUMBERS OF TRAINING SAMPLES AND AUGMENTATION

	# Training Samples	Aug.
Kalantari and Ramamoorthi [68]	2,000,000	✓
Wu *et al.* [69]	120,000	✓
AHDRNet [64]	120,000	✓
ADNet [70]	120,000	✓
Mai *et al.* [71]	66,000	✓
LRT-HDR [5]	13,000	
CA-ViT [72]	50,000	✓
Proposed (AGTC)	13,000	

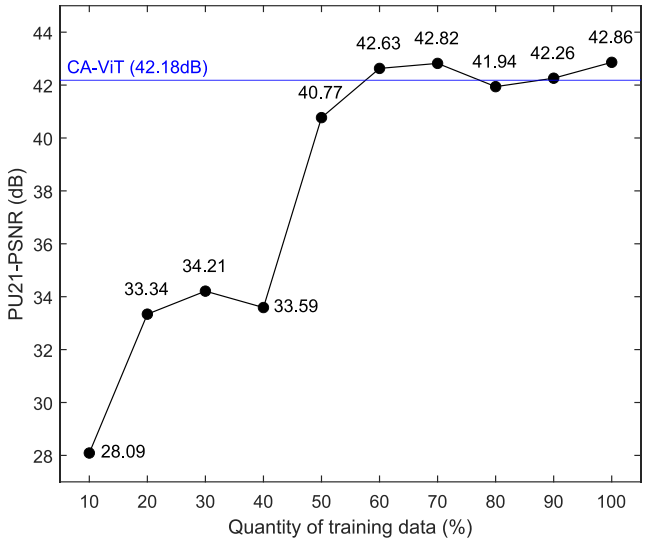


Fig. 8. Quantitative comparison for different quantities of training data. The blue line indicates the PU21-PSNR score of the second-best algorithm CA-ViT [72].

TABLE VIII
COMPARISON OF THE AVERAGE RUNTIMES IN SECONDS AND THE NUMBER OF PARAMETERS

	Runtime	# Param. (M)
TNNM-ALM [61]	20.07	-
Kalantari and Ramamoorthi [68]	0.25	0.38
Wu *et al.* [69]	0.22	16.61
AHDRNet [64]	0.76	1.51
ADNet [70]	1.19	2.96
Mai *et al.* [71]	59.27	17.75
LRT-HDR [5]	16.20	17.83
CA-ViT [72]	3.44	1.22
Proposed (AGTC)	3.56	8.74

layers in the RDB increases, the amount of learned information increases, which consequently improves performance. However, if we continue to increase the number of learnable parameters by increasing the number of convolutional layers in RDBs, the training process becomes difficult to converge, degrading the synthesis performance. Therefore, for HDR imaging, we construct both $\mathcal{F}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{H}}$ using both RDBs with eight convolutional layers to facilitate a graceful trade-off between performance and computational cost.

3) Unfolded Iteration: We analyze the effects of the number of stages K on performance. As shown in Table VI, when $K = 5$, the algorithm provides the worst performance because the amount of visual features learned by the CNN-based regularizers is insufficient for faithful synthesis. As K increases, the amount of learned visual information increases, thereby improving the synthesis performance. However, if we further increase K, the number of parameters in the CNN-based regularizers increases proportionally, hindering the convergence of the training process. As a result, the performance of the proposed algorithm gradually decreases. Therefore, we determined $K = 10$ to achieve the best trade-off between performance and computational and memory complexities.

4) Quantity of Training Data: Table VII compares the approximate numbers of training samples and whether augmentation is applied for the learning-based algorithms. All conventional algorithms, except for LRT-HDR [5], employ data augmentation, thereby constructing large numbers of training samples. Although the proposed AGTC uses the fewest number of training samples without augmentation, it achieves the best performance, as discussed in Section IV-A. This implies that since the proposed AGTC is based on the theoretical foundation of the low-rank model, it only requires a minimal amount of learned information.

We analyze the impact of the quantity of training data on the performance by training the network using different quantities of data. Fig. 8 quantitatively compares the PU21-PSNR performance. As the quantity of training data increases, the synthesis performance improves. Specifically, when only 50% of the training data is used, the proposed algorithm outperforms Kalantari and Ramamoorthi's algorithm [68], Wu et al.'s algorithm [69], and AHDRNet [64]. The proposed algorithm outperforms all the competing algorithms when using 60% or more of the training data except at 80%. This is because outlier samples are included in the training data when we increase the quantity from 70% to 80%; however, their effects gradually decrease as more training samples are added. This ablation study confirms that the proposed AGTC is less dependent on training data and thus necessitates minimal learned information because of its theoretical foundation from the low-rank model.

5) Computational Complexity: We evaluate the computational complexity of the algorithms by comparing the average runtimes for synthesizing 55 test HDR images of size 1820×980 from the HDM-HDR dataset. In this test, we use a PC with a 3.8 GHz CPU and an Nvidia RTX 3090 GPU. Table VIII compares the results. First, the proposed algorithm is

computationally more efficient than Mai et al.'s algorithm [71] and LRT-HDR [5], which are CNN-aided low-rank matrix and tensor completion algorithms, respectively; this is because the proposed algorithm employs tensor factorization instead of the computationally expensive SVD for the tensor norm. Second, compared with pure CNN-based algorithms [64], [68], [69], [70], [72], the proposed algorithm is slower because it still requires complex operations, such as the Fourier transform, which are slower than pure convolution operations. Nevertheless, the proposed algorithm is comparable to CA-ViT [72] in terms of runtime. Table VIII also compares the number of parameters of the learning-based algorithms. The proposed algorithm requires fewer parameters than the other rank minimization-based algorithms [5], [71] because of its theoretically more thorough formulation for the attention-guided LRTC, leading to simpler CNNs for regularizers. Although the number of parameters of the proposed AGTC is higher than those of Kalantari and Ramamoorthi's algorithm [68], AHDRNet [64], ADNet [70], and CA-ViT [72], we experimentally show that the proposed AGTC provides significantly better performance.

V. Conclusion

We proposed an attention-guided low-rank tensor completion (AGTC) algorithm to faithfully restore the original structures of data tensors via deep unfolding attention-guided tensor factorization. We first formulated the low-rank tensor completion task as a robust factorization problem based on low-rank and sparse error assumptions. Low-rank tensor recovery was guided by a CNN-based attention mechanism that localizes important regions to preserve multidimensional spatial dependencies, ensuring more faithful tensor restoration. We also developed implicit regularizers to compensate for the modeling inaccuracies. We then solved the optimization problem by employing an iterative technique to obtain closed-form solutions. Finally, we designed a multistage deep network by unfolding the iterative optimization algorithm, where each stage of the network corresponds to an iteration of the iterative algorithm. Experimental results demonstrated that the proposed algorithm provides state-of-the-art performance on HDR imaging and HSI restoration tasks. Some important directions for future work are to develop accurate low-rank tensor completion models for higher-order tensors [46], [47] and extend the attention-guiding strategy to other definitions of tensor rank for better generalization.

References

[1] L. Chen, X. Jiang, X. Liu, and M. Haardt, "Reweighted low-rank factorization with deep prior for image restoration," *IEEE Trans. Signal Process.*, vol. 70, pp. 3514–3529, 2022.

[2] H. Zhang, L. Liu, W. He, and L. Zhang, "Hyperspectral image denoising with total variation regularization and nonlocal low-rank tensor decomposition," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3071–3084, May 2020.

[3] X. Zhang and M. K. Ng, "Low rank tensor completion with Poisson observations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4239–4251, Aug. 2022.

[4] H. Xu, J. Zheng, X. Yao, Y. Feng, and S. Chen, "Fast tensor nuclear norm for structured low-rank visual inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 538–552, Feb. 2022.

[5] T. T. N. Mai, E. Y. Lam, and C. Lee, "Deep unrolled low-rank tensor completion for high dynamic range imaging," *IEEE Trans. Image Process.*, vol. 31, pp. 5774–5787, 2022.

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

[7] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[8] C. Lu, X. Peng, and Y. Wei, "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5996–6004.

[9] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, no. 1/4, pp. 164–189, Apr. 1927.

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

[11] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Sep. 2011.

[12] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*.

[13] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, Feb. 2013.

[14] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.

[15] Y. Panagakis et al., "Tensor methods in computer vision and deep learning," *Proc. IEEE*, vol. 109, no. 5, pp. 863–890, May 2021.

[16] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Logarithmic norm regularized low-rank factorization for matrix and tensor completion," *IEEE Trans. Image Process.*, vol. 30, pp. 3434–3449, 2021.

[17] Z. Long, C. Zhu, J. Liu, and Y. Liu, "Bayesian low rank tensor ring for image recovery," *IEEE Trans. Image Process.*, vol. 30, pp. 3568–3580, 2021.

[18] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Mach. Learn. Appl.*, vol. 6, pp. 100134:1–100134:13, Dec. 2021.

[19] L. Bragilevsky and I. V. Bajić, "Tensor completion methods for collaborative intelligence," *IEEE Access*, vol. 8, pp. 41162–41174, 2020.

[20] X.-L. Zhao, W.-H. Xu, T.-X. Jiang, Y. Wang, and M. K. Ng, "Deep plug-and-play prior for low-rank tensor completion," *Neurocomputing*, vol. 400, pp. 137–149, Aug. 2020.

[21] Y.-S. Luo, X.-L. Zhao, T.-X. Jiang, Y. Chang, M. K. Ng, and C. Li, "Self-supervised nonlinear transform-based tensor nuclear norm for multi-dimensional image recovery," *IEEE Trans. Image Process.*, vol. 31, pp. 3793–3808, 2022.

[22] H. Liu, Y. Li, M. Tsang, and Y. Liu, "CoSTCo: A neural tensor completion model for sparse tensors," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 324–334.

[23] Y. Li, D. Qiu, and X. Zhang, "Robust low transformed multi-rank tensor completion with deep prior regularization for multi-dimensional image recovery," *IEEE Trans. Big Data*, vol. 9, no. 5, pp. 1288–1301, Oct. 2023.

[24] H. Xu, J. Jiang, Y. Feng, Y. Jin, and J. Zheng, "Tensor completion via hybrid shallow-and-deep priors," *Appl. Intell.*, vol. 53, no. 13, pp. 17093–17114, Jul. 2023.

[25] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Tensor completion via complementary global, local, and nonlocal priors," *IEEE Trans. Image Process.*, vol. 31, pp. 984–999, 2021.

[26] J.-L. Wang, T.-Z. Huang, X.-L. Zhao, Y.-S. Luo, and T.-X. Jiang, "CoNoT: Coupled nonlinear transform-based low-rank tensor representation for multidimensional image completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8969–8983, Jul. 2024.

[27] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Feb. 2021.

[28] M. Ashraphijuo and X. Wang, "Fundamental conditions for low-CP-rank tensor completion," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2116–2145, Jan. 2017.

[29] P. Breiding and N. Vannieuwenhoven, "A Riemannian trust region method for the canonical tensor rank approximation problem," *SIAM J. Optim.*, vol. 28, no. 3, pp. 2435–2465, Jan. 2018.

[30] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sep. 2015.

[31] Q. Shi, H. Lu, and Y.-M. Cheung, "Tensor rank estimation and completion via CP-based nuclear norm," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 949–958.

[32] L. Zhang, W. Wei, Q. Shi, C. Shen, A. van den Hengel, and Y. Zhang, "Accurate tensor completion via adaptive low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4170–4184, Oct. 2020.

[33] C. Hillar and L. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, pp. 45:1–45:39, Nov. 2013.

[34] C. Da Silva and F. J. Herrmann, "Optimization on the Hierarchical Tucker manifold - Applications to tensor completion," *Linear Algebra Appl.*, vol. 481, pp. 131–173, Sep. 2015.

[35] Y. Liu, Z. Long, H. Huang, and C. Zhu, "Low CP rank and Tucker rank tensor completion for estimating missing components in image data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 944–954, Apr. 2020.

[36] Z. Long, Y. Liu, L. Chen, and C. Zhu, "Low rank tensor completion for multiway visual data," *Signal Process.*, vol. 155, pp. 301–316, Feb. 2019.

[37] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, May 2017.

[38] S. Holtz, T. Rohwedder, and R. Schneider, "On manifolds of tensors of fixed TT-rank," *Numer. Math.*, vol. 120, no. 4, pp. 701–731, Apr. 2012.

[39] Y. Liu, J. Liu, and C. Zhu, "Low-rank tensor train coefficient array estimation for tensor-on-tensor regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5402–5411, Dec. 2020.

[40] O. Mickelin and S. Karaman, "On algorithms for and computing with the tensor ring decomposition," *Numer. Linear Algebra Appl.*, vol. 27, no. 3, pp. e2289:1–e2289:24, May 2020.

[41] F. Sedighin, A. Cichocki, and A.-H. Phan, "Adaptive rank selection for tensor ring decomposition," *IEEE J. Sel. Top. Signal Process.*, vol. 15, no. 3, pp. 454–463, Apr. 2021.

[42] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[43] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, "Tensor-based formulation and nuclear norm regularization for multienergy computed tomography," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.

[44] Y. Zhou and Y.-M. Cheung, "Bayesian low-tubal-rank robust tensor factorization with multi-rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 62–76, Jan. 2021.

[45] J. Hou, F. Zhang, H. Qiu, J. Wang, Y. Wang, and D. Meng, "Robust low-tubal-rank tensor recovery from binary measurements," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4355–4373, Aug. 2022.

[46] C. D. Martin, R. Shafer, and B. LaRue, "An order-p tensor factorization with applications in imaging," *SIAM J. Sci. Comput.*, vol. 35, no. 1, pp. A474–A490, Jan. 2013.

[47] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-Y. Ji, and T.-H. Ma, "Tensor N-tubal rank and its convex relaxation for low-rank tensor recovery," *Inf. Sci.*, vol. 532, pp. 170–189, Sep. 2020.

[48] G.-J. Song, X.-Z. Wang, and M. K. Ng, "Riemannian conjugate gradient descent method for fixed multi rank third-order tensor completion," *J. Comput. Appl. Math.*, vol. 421, pp. 114866:1–114866:30, Mar. 2023.

[49] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[50] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[51] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 666–681, 2020.

[52] O. Solomon et al., "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, Apr. 2020.

[53] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 370–378.

[54] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, Oct. 2019.

[55] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6360–6376, Oct. 2022.

[56] T. Huang, X. Yuan, W. Dong, J. Wu, and G. Shi, "Deep Gaussian scale mixture prior for image reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10778–10794, Sep. 2023.

[57] H. Wang, Q. Xie, Q. Zhao, and D. Meng, "A model-driven deep neural network for single image rain removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3100–3109.

[58] X. Fu, M. Wang, X. Cao, X. Ding, and Z.-J. Zha, "A model-driven deep unfolding method for JPEG artifacts removal," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6802–6816, Nov. 2022.

[59] C. Mou, Q. Wang, and J. Zhang, "Deep generalized unfolding networks for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17399–17410.

[60] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, and T. Huang, "Low-rank high-order tensor completion with applications in visual data," *IEEE Trans. Image Process.*, vol. 31, pp. 2433–2448, Mar. 2022.

[61] C. Lee and E. Y. Lam, "Computationally efficient truncated nuclear norm minimization for high dynamic range imaging," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4145–4157, Sep. 2016.

[62] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," University of Illinois, Urbana-Champaign, Tech. Rep. UILU-ENG-09–2215, Nov. 2009.

[63] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[64] Q. Yan et al., "Attention-guided network for ghost-free high dynamic range imaging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1751–1760.

[65] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2472–2481.

[66] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1107–1130, Oct. 2008.

[67] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[68] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 144:1–144:12, Jul. 2017.

[69] S. Wu, J. Xu, Y.-W. Tai, and C.-K. Tang, "Deep high dynamic range imaging with large foreground motions," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 120–135.

[70] Z. Liu et al., "ADNet: Attention-guided deformable convolutional network for high dynamic range imaging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2021, pp. 463–470.

[71] T. T. N. Mai, E. Y. Lam, and C. Lee, "Ghost-free HDR imaging via unrolling low-rank matrix completion," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 2928–2932.

[72] Z. Liu, Y. Wang, B. Zeng, and S. Liu, "Ghost-free high dynamic range imaging with context-aware transformer," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 344–360.

[73] J. Froehlich, S. Grandinetti, B. Eberhardt, S. Walter, A. Schilling, and H. Brendel, "Creating cinematic wide gamut HDR-video for the evaluation of tone mapping operators and HDR-displays," in *Proc. Digit. Photography X*, 2014, pp. 279–288.

[74] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[75] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[76] R. Mantiuk, K. Myszkowski, and H.-P. Seidel, "Lossy compression of high dynamic range images and video," in *Proc. Digit. Photography X*, 2006, pp. 311–320.

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

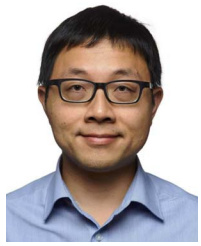
[78] E. Reinhard and K. Devlin, "Dynamic range reduction inspired by photoreceptor physiology," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 1, pp. 13–24, Jan./Feb. 2005.

[79] R. K. Mantiuk and M. Azimi, "PU21: A novel perceptually uniform encoding for adapting existing quality metrics for HDR," in *Proc. Pict. Coding Symp.*, 2021, pp. 1–5.

[80] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 40:1–40:14, Jul. 2011.

[81] O. Sidorov and J. Y. Hardeberg, "Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3844–3851.
[82] T. Bodrito, A. Zouaoui, J. Chanussot, and J. Mairal, "A trainable spectral-spatial sparse coding model for hyperspectral image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 5430–5442.
[83] M. Wang, Q. Wang, J. Chanussot, and D. Hong, "Total variation regularized weighted tensor ring decomposition for missing data recovery in high-dimensional optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6002505.
[84] N. Liu, W. Li, R. Tao, Q. Du, and J. Chanussot, "Multigraph-based low-rank tensor approximation for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5530314.
[85] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Robust low-rank tensor recovery via nonconvex singular value minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 9044–9059, 2020.
[86] Q. Yu and M. Yang, "Low-rank tensor recovery via non-convex regularization, structured factorization and spatio-temporal characteristics," *Pattern Recognit.*, vol. 137, pp. 109343:1–109343:14, May 2023.
[87] P. Gamba, The pavia university dataset. 2023. [Online]. Available: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
[88] EarthExplorer, The landsat 7 ETM dataset. 2023. [Online]. Available: https://earthexplorer.usgs.gov/
[89] H. Zhang, H. Chen, G. Yang, and L. Zhang, "LR-Net: Low-rank spatial-spectral network for hyperspectral image denoising," *IEEE Trans. Image Process.*, vol. 30, pp. 8743–8758, 2021.
[90] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogramm. Eng. Remote Sens.*, vol. 63, no. 6, pp. 691–699, Jun. 1997.
[91] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm," in *Proc. Summaries Annu. JPL Airborne Geosci. Workshop*, 1992, pp. 147–149.
[92] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[93] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia, "FFA-Net: Feature fusion attention network for single image dehazing," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11908–11915.
[94] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11863–11874.
[95] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Truong Thanh Nhat Mai (Student Member, IEEE) received the BS degree in mathematics and computer science from the Ho Chi Minh City University of Science, Ho Chi Minh City, Vietnam, in 2014, and the MS degree in electrical, electronic, and control engineering from Hankyong National University, Anseong, South Korea, in 2017. He is currently working toward the PhD degree with the Department of Multimedia Engineering, Dongguk University, Seoul, South Korea. His research interests include model-based deep learning for image processing, image restoration, and computational imaging.



Edmund Y. Lam (Fellow, IEEE) received the BS, MS, and PhD degrees in electrical engineering from Stanford University. He was a visiting associate professor with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. He is currently a professor of electrical and electronic engineering with The University of Hong Kong. He also the computer engineering program director and a research program coordinator with the AI Chip Center for Emerging Smart Systems. His research interest includes computational imaging algorithms, systems, and applications. He is a fellow of Optica, SPIE, IS&T, and HKIE; and a founding member of the Hong Kong Young Academy of Sciences.

Chul Lee (Member, IEEE) received the BS, MS, and PhD degrees in electrical engineering from Korea University, Seoul, South Korea, in 2003, 2008, and 2013, respectively. From 2002 to 2006, he was with Biospace Inc., Seoul, where he was involved in the development of medical equipment. From 2013 to 2014, he was a postdoctoral scholar with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA, USA. From 2014 to 2015, he was a Research Scientist with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong. From 2015 to 2019, he was an assistant professor with the Department of Computer Engineering, Pukyong National University, Busan, South Korea. In 2019, he joined the Department of Multimedia Engineering, Dongguk University, Seoul, where he is currently an associate professor. His research interests include image processing and computational imaging with an emphasis on restoration and high dynamic range imaging. Dr. Lee was the recipient of the Best Paper Award from the *Journal of Visual Communication and Image Representation,* in 2014. He is also an editorial board member of the *Journal of Visual Communication and Image Representation*.