Learning Unbalanced and Sparse Low-Order Tensors*

P. M. Hoang[1], H. D. Tuan[2], T. T. Son[1], H. V. Poor[3], and L. Hanzo[4]

***Abstract*—Efficient techniques are developed for completing unbalanced and sparse low-order tensors, which cannot be effectively completed by popular matrix-rank optimization based techniques such as compressed sensing and/or the ℓ_q-matrix-metric. We use our previously developed 2D-index encoding technique for tensor augmentation in order to represent these incomplete low-order tensors by high-order but low-dimensional tensors with their modes building up a coarse-grained hierachy of correlations among the incomplete tensor entries. The concept of tensor-trains is then exploited for decomposing these augmented tensors into trains of balanced and sparse matrices for efficient completion. More explicitly, we develop powerful algorithms exhibiting an excellent performance vs. complexity trade-off, which are supported by numerical examples by relying on matrix data and third-order tensor data constituted by color images.**

***Index Terms*—Matrix and/or low-order tensor completion, tensor train decomposition, tensor train rank, ℓ_q**

I. Introduction

Low-order tensors constitute components of representing multi-dimentional datasets. For instance, a DNA microarray is a matrix (2nd-order tensor) of gene expressions with the row index representing probes and the column index representing genes [1]. The Netflix movie rating dataset [2] is another matrix with the row index representing viewers and the column index representing the rated movies. A color image pixel is represented by a third-order tensor using the row and column indices for the pixels, while the third index represents the colors. To expound a little further, a vehicular traffic volume dataset may be modelled by a fifth-order tensor with three indices corresponding to the time stamp (weeks, days, hours) and two indices to the roads and road directions. Given these compelling examples as well as a range of further applications, matrix and and low-order tensor completion - which aims for completing them based on their partially observed entries - constitutes a fundamental problem in data learning and processing. The state-of-the-art in tensor completion techniques is dominated by matrix-rank optimization and singular value thresholding techniques [3]–[6]. Their extensions in [7], [8] use the ℓ_q penalty optimization technique for completing sparse matrices. However, the matrix rank is only useful for completing so-called balanced matrices having similar numbers of rows and columns. By contrast, unbalanced matrices having dissimilar numbers of rows and columns tend to be of low-rank, hence their rank optimization is not meaningful. In fact, as detailed in [9], the widely used compressed sensing technique [6] requires almost fully observed entries of unbalanced matrices for their successful completion. This is not surprising, because the singular values of unbalanced matrices tend to be well-conditioned by the law of large numbers, so it follows from the classical Eckart and Young theorem [10] that their best low-rank approximation is in fact quite rough. The information loss imposed by thresholding these well-conditioned singular values is almost the same as their von Neumann entropy [11].

Suffice to say that most matrices found in practical applications are unbalanced. For example, the aforementioned DNA microarray is unbalanced, because its number of rows - which is the total number of probes - is only on the order of a few dozens or hundreds. By contrast, the number of its columns - which is the number of genes - is tens of thousands. Similarly, the aforementioned Netflix movie rating dataset is unblanced, because its number of rows - which is the total number of viewers - is much higher than its number of columns, which is the total number of the rated movies. Interestingly, this Netflix data serves as the primary motivation of the matrix completion problem treated in [7], [8], but both the incomplete matrices and sub-matrices of the observable entries are square in all the numerical examples of [7], [8]. The size of the third index set of a color image pixel is as low as three for redness, blueness, and gree-nes, which is obviously very small compared to the cardinality of the other two modes (image height and width). The size of the fifth index of a vehicular traffic volume dataset is two for the pair of opposite road directions, which is also very small compared to that of the other four indices.

Analogously, the conceptual drawback of the tensor completion techniques treated in [12]–[16] is that they are based on matrix-rank optimization, which are constructed based on an unbalanced 'matricization' scheme (one mode versus the rest). The specific third-order tensor completion proposed in [17], [18] also requires that the tensors considered are well balanced having the same cardinality for all three modes. To overcome the issue of inherent low-rank under unbalanced matricization schemes, our previous work [19] is the first contribution exploiting tensor train (TT) decomposition [20],

*This work was supported in part by the Australian Research Council's Discovery Projects under Grant DP190102501, in part by the U.S. National Science Foundation under Grant CCF-1908308, and a grant from the C3.ai Digital Transformation Institute, and in part by the Engineering and Physical Sciences Research Council projects EP/W016605/1 and EP/P003990/1 (COALESCE), and the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028).

[1]Faculty of Information Technology, University of Science, VNU-HCM, Vietnam; (email: pmhoang@fit.hcmus.edu.vn, ttson@fit.hcmus.edu.vn); [2]School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia (email: tuan.hoang@uts.edu.au); [3]Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA (email: poor@princeton.edu); [4]School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, U.K (email: lh@ecs.soton.ac.uk).

[21][1] to form balanced matricization for facilitating both rank-optimization and singular value thresholding based methods for completing low-order tensors.[2] To maximize the capacity of TT decomposition leading to balanced matricizations, which capture both the global entry correlations and entanglements, it was proposed in [19] to represent the incomplete low-order tensors of large dimensions by higher-order tensors of low dimensions. More particularly, the quantum-state based ket-augmentation (KA) used in [24] has been shown to be very effective for tensor augmentation (TA) of representing $2^N \times 2^N$ color images (3rd-order $2^N \times 2^N \times 3$ tensors). By viewing TA as a problem of encoding two-dimensional (2D) indices by N-digit words, a new TA was proposed in [25] for representing low-order tensors of flexible sizes and structures by high-order tensors, with the latter providing a completely new coarse-grained hierarchy of the former's entries.

Against the above background, this paper is the first contribution addressing the problem of low-order unbalanced and sparse tensor completions. Both incomplete tensors and their sub-tensors of observable entries can be unbalanced. The paper's contributions can be summarized as follows:

- We lay the foundation for completing unbalanced and sparse low-order tensors, which is based on the TA concept [25] of representing low-order large-dimensional tensors by high-order low-dimensional tensors and then TT-decomposition for high-order tensors;
- Based on our new results on the ℓ_q-metric, we develop new high-performance algorithms for the completion of both unbalanced and sparse low-order tensors;
- We develop new computationally efficient algorithms for the completion of unbalanced low-order tensors, which do not require singular-value-decomposition (SVD).

The contributions of this work relative to previous related literature are shown in Table I.

The paper is organized as follows. Section II introduces a TA technique to transform the problem of completing unbalanced and low-order tensors to that of completing high-order and low-dimensional tensors. Section III then develops high-performance algorithm for completion of these tensors by exploiting their sparse structures. Its main ingredient is a new bounding technique for the ℓ_q-metric by exploitation of its partial convexities, which is relegated to the Appendix for maintaining the presentation flow. Section IV is based on Frobenius norm to develop an SVD-free algorithm for tensor completion. Computational experiments to support the development of Sections III and IV are provided in Section V, while Section VI concludes the paper.

Notation. Matrices are denoted as capital letters, e.g. $X \in \mathbb{R}^{I\times J}$, which is referred as a matrix of size $I \times J$ with entries $X(i,j)$, $i = 1,\dots,I$ and $j = 1,\dots,J$. Accordingly, $X^T \in \mathbb{R}^{J\times I}$ and $X^\dagger \in \mathbb{R}^{J\times I}$ stand for its transposed matrix and its pseudo-inversion. Also, for $\Omega \subset \{1,\dots,I\}\times\{1,\dots,J\}$, X_Ω is the matrix of the same size with X, which is defined by

$$X_\Omega(i,j) = \begin{cases} X(i,j) & \text{for} \quad (i,j)\in\Omega \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

$\text{diag}[d_1,\dots,d_r]$ stands for the diagonal matrix of size $r\times r$ with its diagonal entries $d_1,\dots,d_r$.

[1]The concept of TT was introduced in quantum physics as a "matrix-product" much earlier in 2003 (see e.g. [20]) and it is still a hot topic in quantum physics.

[2]This method was used in [22], [23] for completing images having some additional structural constraints imposed on the missing entries, and also for vehicular traffic volumes.

II. Tensor augmentation for unbalanced tensor completion

Let I_n, $n=1,\dots,N$ be positive integers. A Nth-order *tensor* $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is an N-dimensional array having entries $\mathcal{X}(i_1,i_2,\dots,i_N)$, $i_n\in\mathcal{I}_n \triangleq \{1,\dots,I_n\}$; $n\in\mathcal{N}\triangleq\{1,\dots,N\}$. Each $n\in\mathcal{N}$ is termed as its mode with size I_n. On one hand, such a tensor is said to be of high-order whenever N is large, and of low-order whenever N is low. On the other hand, such tensor is said to be large-dimensional whenever there are large I_n. It is also said to be unbalanced whenever the ratio $\max_{n\in\mathcal{N}} I_n/\min_{n\in\mathcal{N}} I_n$ is large. The lowest-order tensors are vectors (first-order), and matrices (second-order). Accordingly, the matrices of size $I_1\times I_2$ are unbalanced whenever we have $\max\{I_1,I_2\}/\min\{I_1,I_2\} >> 1$.

The Frobenius norm of $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is defined by

$$||\mathcal{X}|| = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N}\mathcal{X}^2(i_1,i_2,\dots,i_N)}.$$

Theorem 1. *Given $\alpha_k>0$, $k=1,\dots,K$, and tensors $\mathcal{U}_k$, $k=1,\dots,K$ of the same size, one has*

$$\frac{\sum_{k=1}^K\alpha_k\mathcal{U}_k}{\sum_{k=1}^K\alpha_k} = \arg\min_{\mathcal{X}}\sum_{k=1}^K\alpha_k||\mathcal{U}_k-\mathcal{X}||^2. \tag{2}$$

Mode-$(1,2,\dots,k)$ *matricization* of $\mathcal{X}$ [21] is defined as the matrix $\mathcal{X}_{[k]}\in\mathbb{R}^{(\prod_{\ell=1}^k I_\ell)\times(\prod_{\ell=k+1}^N I_\ell)}$ so that we have

$$\begin{aligned}\mathcal{X}_{[k]}(i_1+\sum_{\ell=2}^k(i_\ell-1)J_\ell, i_{k+1}+\sum_{\ell=k+2}^N(i_\ell-1)\hat{J}_\ell) &= \\ \mathcal{X}(i_1,i_2,\dots,i_N), \quad & \quad (3)\\ J_\ell = \prod_{m=1}^{\ell-1} I_m, \ell=2,\dots,k; &\\ \hat{J}_\ell = \prod_{m=k+1}^{\ell-1} I_m, \ell\geq k+2,\dots,N. &\end{aligned}$$

This is a *balanced matricization* scheme because $\mathcal{X}_{[k]}$ can essentially be a square matrix when $\prod_{\ell=1}^k I_\ell \approx \prod_{\ell=k+1}^N I_\ell$. Accordingly, the operator $\mathsf{fold}(\mathcal{X}_{[k]})$ recovers $\mathcal{X}$ so we can write $\mathcal{X}=\mathsf{fold}(\mathcal{X}_{[k]})$, i.e.

$$\begin{aligned}\mathsf{fold}(\mathcal{X}_{[k]})(i_1,i_2,\dots,i_N) &= \\ \mathcal{X}_{[k]}(i_1+\sum_{\ell=2}^k(i_\ell-1)J_\ell, i_{k+1}+\sum_{\ell=k+2}^N(i_\ell-1)\hat{J}_\ell). & \quad (4)\end{aligned}$$

The TT rank is defined as the vector $\mathbf{r}=(r_1,r_2,\dots,r_{N-1})$, where r_k is the matrix rank of $X_{[k]}$. One can see that r_k may

Literature / Contents	**This work**	[9], [25]	[19]	[12]–[17]	[7], [8]
Balanced matricization	√	√			
Sparse data	√				√
Unbalanced tensor completion	√	√			
Tensor augmentation based	√	√	√		
SVD-free algorithms	√				

Table I: Boldly contrasting our novel contributions to the related literature.

be high as $\mathcal{X}_{[k]}$ is more balanced, making the employment of matrix-rank optimization based completion more appropriate.

Since our objective is to complete low-order and sparse tensors, let us briefly consider the following problem of completing sparse matrices: *complete a matrix X of size $m\times n$ over $\Omega\subset\{1,\dots,m\}\times\{1,\dots,n\}$, which is the set of indices of its observed entries $\tilde{x}_{ij}$.*

As such it is conventionally formulated as

$$\min_{X\in\mathbb{R}^{m\times n}} f_q(X)\triangleq\frac{1}{2}||\tilde{X}_\Omega-X_\Omega||^2+\lambda\ell_q(X),\tag{5}$$

where in what follows, we have

$$\ell_q(X)\triangleq\sum_{i=1}^{L}\sigma_i^q(X)\tag{6}$$

where $\sigma_i(X)$, $i=1,\dots,L$ represent the singular values of X [26]–[28]. Still referring to (5), we have

$$\tilde{X}(i,j)=\tilde{x}_{ij}\quad\text{for}\quad(i,j)\in\Omega\tag{7}$$

which is called the matrix of projection onto the observed entries [7], and λ in (5) is a penalty parameter for incorporating $\ell_q(X)$ in the optimization objective function. The problem (5) was firstly proposed in [7] for $q=1$, and then in [8] for $0<q<1$. The motivation of the latter work is that the matrix-rank (ℓ_0) optimization based formulation may provide a beneficial technique of handling very sparse matrices but its optimization is challenging. Hence, (5) for $0<q<1$ may provide less biased and sparser solutions than its counterpart associated with $q=1$.

Define the set of indices of the missing entries by

$$\Omega^c\triangleq\{1,\dots,m\}\times\{1,\dots,n\}\setminus\Omega.\tag{8}$$

When initialized by the zero matrix $X^{(0)}\in\mathbb{R}^{m\times n}$, the κ-th iteration used for generating $X^{(\kappa+1)}$ is based on solving the following problem

$$\min_{X\in\mathbb{R}^{m\times n}} f_q^{(\kappa)}(X)\triangleq\frac{1}{2}||\tilde{X}_\Omega+X^{(\kappa)}_{\Omega^c}-X||^2+\lambda\ell_q(X),\tag{9}$$

which admits a closed-form solution:

$$X^{(\kappa+1)}=U\text{diag}[\varphi_\lambda(d_1^{(\kappa)}),\dots,\varphi_\lambda(d_r^{(\kappa)})]V^T,\tag{10}$$

under the SVD of

$$X^{(\kappa)}=UDV^T\tag{11}$$

with the orthogonal matrices U and V as well as the diagonal matrix $D=\text{diag}[d_1^{(\kappa)},\dots,d_r^{(\kappa)}]$, while

$$\varphi_\lambda(d)\triangleq\arg\min_x[\frac{1}{2}(d-x)^2+\lambda|x|^q],\tag{12}$$

which can be readily calculated. For instance, when $q=1$, we have

$$\varphi_\lambda(d)=(d-\lambda)_+\triangleq\max\{d-\lambda,0\},\tag{13}$$

which represents the popular soft-thresholding rule [3], [4] for solving (9). For other $0<q<1$ see [29, Proposition 2]. The main result of [7], [8] is that of proving the convergence of $\{X^{(\kappa)}\}$ in (11).

It is seen from (10)-(11) that their complexity is determined by the SVD (11) at each iteration. In the end, it is not $X^{(\kappa)}$ but $\tilde{X}_\Omega+X^{(\kappa)}_{\Omega^c}$ is accepted as the incumbent and then $\tilde{X}_\Omega+X^{(\infty)}_{\Omega^c}$ is accepted as the final solution. As will be shown by our simulations, the following trivial SVD-free and matrix-multiplication free iterations also perform similarly well to the solution in (10)-(11). Let us invoke the SVD of

$$\tilde{X}_\Omega=UDV^T\tag{14}$$

relying on the orthogonal matrices U and V as well as on the diagonal matrix $D=\text{diag}[d_1^{(0)},\dots,d_r^{(0)}]$. Then, for $\kappa=0,1,\dots$, we can generate $d_i^{(\kappa+1)}$ by

$$d_i^{(\kappa+1)}=\varphi_\lambda(d_i^{(\kappa)}),i=1,\dots,r\tag{15}$$

and accept the resultant solution in the form of

$$X^\infty=U\mathsf{diag}[d_1^\infty,\dots,d_r^\infty]V^T.\tag{16}$$

Observe that (5) particularly aims for minimizing the rank of X and as such it is only suitable for completing balanced matrices. For an unbalanced X, it is likely that its rank is $\min\{m,n\}$, so there is nothing to minimize. As shown in [9], this kind of rank-based compressed sensing is not suitable for unbalanced matrices, because it follows from a result in [6] that one needs almost as many as nm entries of X for successfully completing it. As shown in [19], the singular values of unbalanced matrices are very well conditioned so the information loss imposed by their least-square based completion is almost as high as their von Neumann entropy [11]. It is not a surprise that both [7] and [8] only provided simulation results for most balanced (square) matrices X and Ω.

To circumvent the issue of unbalanced matrices, which makes the matrix-rank optimization based completion and compressed sensing deficient, as a remedy, it was proposed in [9] and [25] to represent these matrices by high-order and low-dimensional tensors for tensor completion. For tensor completion, the TT-based tensor decomposition [20], [21] has been used for avoiding the creation of only unbalanced matrix factors by Tucker decomposition, also known as higher-order singular value decomposition (HOSVD) [30]. Naturally, the efficiency of this approach is heavily dependent on the capability of TA to capture all the correlations and entanglements among the matrix entries. We thus opt here for the most efficient known TA of [25], which works for matrices of flexible sizes, and it is capable of capturing the distinct correlations among coarse-grained textures. Let $m = \prod_{k=1}^{N} J_k^a$ and $n = \prod_{k=1}^{N} J_k^b$ with small J_k^a and J_k^b. Upon encoding the 2D indices $(i,j) \in \{1,\ldots,\prod_{k=1}^{N} J_k^a\} \times \{1,\ldots,\prod_{k=1}^{N} J_k^b\}$ by $N-$digit words $i_1 i_2 \ldots i_N$ with $i_k \in \{1,\ldots,J_k^a J_k^b\}$, $k=1,\ldots,N$ according to [25], the matrix X may be represented by an N-order tensor

$$\mathcal{X}^{\mathrm{A}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N} \tag{17}$$

with small $I_k = J_k^a J_k^b$, $k=1,\ldots,N$. Based on (3), $\mathcal{X}^{\mathrm{A}}$ can be unfolded to the matrix $\mathcal{X}_{[k]}^{\mathrm{A}}$, $k=1,\ldots,N$ of size $m_k \times n_k$

$$\mathcal{X}_{[k]}^{\mathrm{A}} \in \mathbb{R}^{m_k \times n_k}, \tag{18}$$

with

$$m_k = \prod_{j=1}^{k} I_j \tag{19}$$

and

$$n_k = \prod_{\ell=k+1}^{N} I_\ell, \tag{20}$$

which encapsulates the correlation among k modes $1,\ldots,k$ and the remaining $k+1,\ldots,N$. In parallel, the projection matrix $\tilde{X}$ defined by (7) is represented by the tensor $\tilde{\mathcal{X}}^{\mathrm{A}}$ of the same size with $\mathcal{X}^{\mathrm{A}}$ in (17).

Naturally, upon encoding the 2D indices $(i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\}$ by the $N-$digit words $i_1 i_2 \ldots i_N$ with $i_k \in \{1,\ldots,J_k^a J_k^b\}$, $k=1,\ldots,N$, $n = \prod_{k=1}^{N} J_k^a$ and $m = \prod_{k=1}^{N} J_k^b$, we can represent a third-order tensor $\mathcal{X}$ of size $m \times n \times p$ by the $(N+1)$-order tensor

$$\mathcal{X}^{\mathrm{A}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N \times p}. \tag{21}$$

for considering the following problem of completing unbalanced third-order tensors: *complete the third-order tensor $\mathcal{X}$ of size $m \times n \times p$ constructed over $\Omega \triangleq \bar{\Omega} \times \{1,\ldots,p\}$ with $\bar{\Omega} \subset \{1,\ldots,m\} \times \{1,\ldots,n\}$, which is the set of indices of the observed entries $\tilde{x}_{ijk}$.*

Based on (3), $\mathcal{X}^{\mathrm{A}}$ in (21) can be unfolded to the matrix $\mathcal{X}_{[k]}^{\mathrm{A}}$, $k=1,\ldots,N$ of size $m_k \times (n_k p)$ defined by (18) with m_k defined by (19), but n_k defined as:

$$n_k = p \prod_{\ell=k+1}^{N} I_\ell, \tag{22}$$

which encapsulates the correlation among the k modes $1,\ldots,k$ and the remaining $k+1,\ldots,N+1$.

Similarly to (7), we also define the third-order tensor of projection onto the observed entries as

$$\tilde{\mathcal{X}}_\Omega(i,j,k) = \begin{cases} \tilde{x}_{ijk} & \text{for} \quad (i,j,k) \in \Omega \\ 0 & \text{otherwise,} \end{cases} \tag{23}$$

which is also represented by the $(N+1)$-order tensor

$$\tilde{\mathcal{X}}^{\mathrm{A}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N \times p}, \tag{24}$$

of the same size with $\mathcal{X}^{\mathrm{A}}$ in (21).

In the sequel, instead of completing the matrix X or the third-order tensor $\mathcal{X}$, we will complete their high-order representations $\mathcal{X}^{\mathrm{A}}$ in (17) and (21), respectively.

III. ℓ_q-BASED TENSOR COMPLETION

The objective of this section is to complete both the high-order and low-dimensional tensors $\mathcal{X}^{\mathrm{A}}$ in (17) and (21) by exploiting their sparse structures. Indeed, $\mathcal{X}^{\mathrm{A}}$ is sparse if and only if so are its unfolding matrices $\mathcal{X}_{[k]}^{\mathrm{A}} \in \mathbb{R}^{m_k \times n_k}$ in (18) with m_k defined by (19) and n_k defined by (20) or (22). The first subsection thus develops completion algorithms seeking sparse $\mathcal{X}_{[k]}^{\mathrm{A}}$ while the second subsection develops completion algorithms seeking their sparse factor matrices in their matrix product factorizations.

Let $\Omega_{[k]}$ be the set of indices of observed entries of the unfolding matrices $\mathcal{X}_{[k]}^{\mathrm{A}}$ in (18). Accordingly, the set of indices of the missing entries of $\mathcal{X}_{[k]}^{\mathrm{A}}$ is defined by

$$\Omega_{[k]}^c \triangleq \{1,\ldots,m_k\} \times \{1,\ldots,n_k\} \setminus \Omega_{[k]}.$$

For notational convenience, we also use $\tilde{\mathcal{X}}_{\Omega_{[k]}}^{\mathrm{A}}$ to refer the mode-$(1,2,\ldots,k)$ matricization $(\tilde{\mathcal{X}}_\Omega^{\mathrm{A}})_{[k]}$ of $\tilde{\mathcal{X}}_\Omega^{\mathrm{A}}$.

A. Decomposition approach

To exploit the sparsity of the unfolding matrices $\mathcal{X}_{[k]}^{\mathrm{A}}$ in (18), we consider the problem

$$\min_{X_1,\ldots,X_{N-1}} \sum_{k=1}^{N-1} \left(\frac{\beta_k}{2} ||\tilde{\mathcal{X}}_{\Omega_{[k]}}^{\mathrm{A}} - (X_k)_{\Omega_{[k]}}||^2 + \alpha_k \ell_q(X_k) \right) \tag{25}$$

associated with the matrix $X_k \in \mathbb{R}^{m_k \times n_k}$ to learn $\mathcal{X}_{[k]}^{\mathrm{A}}$ and

$$\beta_k = \frac{\alpha_k}{\lambda}, \alpha_k = \frac{\delta_k}{\sum_{k'=1}^{N-1} \delta_{k'}}, \delta_{k'} = \min\{m_{k'}, n_{k'}\}, \quad k' = 1,\ldots,N-1, \tag{26}$$

for λ selected from one of the values in the set $\{100, 20, 10, 2, 1\}$, which assigns larger α_k to more balanced matrices. Note that in (25) we use the penalty term $\alpha_k \ell_q(X_k)$ to impose the sparse structure of X_k.

The problem (25) is thus decomposed into N independent subproblems

$$\min_{X_k} f_{q,k}(X_k) \triangleq \frac{\beta_k}{2} ||\tilde{\mathcal{X}}_{\Omega_{[k]}}^{\mathrm{A}} - (X_k)_{\Omega_{[k]}}||^2 + \alpha_k \ell_q(X_k). \tag{27}$$

When initialized by the zero matrix $X_k^{(0)} \in \mathbb{R}^{m_k \times n_k}$, at the κ-iteration, we solve the following problem for generating $X_k^{(\kappa+1)}$:

$$X_k^{(\kappa+1)} = \arg\min_{X_k} f_{q,k}^{(\kappa)}(X_k) \tag{28}$$

$$\triangleq \arg\min_{X_k} \left[\frac{1}{2} ||\tilde{\mathcal{X}}_{\Omega_{[k]}}^{\mathrm{A}} + (X_k^{(\kappa)})_{\Omega_{[k]}^c} - X_k||^2 + \frac{\alpha_k}{\beta_k} \ell_q(X_k) \right]$$

$$= \arg\min_{X_k} \left[\frac{1}{2} ||\tilde{\mathcal{X}}_{\Omega_{[k]}}^{\mathrm{A}} + (X_k^{(\kappa)})_{\Omega_{[k]}^c} - X_k||^2 + \lambda \ell_q(X_k) \right] \tag{29}$$

$$= U_k^{(\kappa)}\text{diag}[\varphi_\lambda(d_1^{(\kappa)}),\dots,\varphi_\lambda(d_r^{(\kappa)})](V_k^{(\kappa)})^T, \quad (30)$$

under the SVD of

$$X_k^{(\kappa)} = U^{(\kappa)}D(V_k^{(\kappa)})^T, \quad (31)$$

which is in the form of (10).

In Appendix I, we prove that

$$f_{q,k}(X_k^{(\kappa)}) < f_q(X_k^{(\kappa+1)}), \quad (32)$$

i.e. $X_k^{(\kappa+1)}$ is a better feasible point than $X_k^{(\kappa)}$ provided that $X_k^{(\kappa+1)} \neq X_k^{(\kappa)}$, and as such the sequence $\{X_k^{(\kappa)}\}$ converges to a local solution X_k^{∞} of (27) [31]. Under $q=1$, the problem (27) is convex so $\{X_k^{(\kappa)}\}$ converges to its global solution.

Then we use (2) to accept the final solution formulated as

$$\begin{aligned}\bar{\mathcal{X}} &= \arg\min_{\mathcal{X}} \frac{\beta_k}{2}||\mathcal{X} - \mathsf{fold}_k(X_k^{\infty})||^2\\ &= \frac{1}{\sum_{k=1}^{N-1}\beta_k}\sum_{k=1}^{N-1}\beta_k(\mathsf{fold}_k(X_k^{\infty}))\\ &= \sum_{k=1}^{N-1}\alpha_k \mathsf{fold}_k(X_k^{\infty}). \quad (33)\end{aligned}$$

Algorithm 1 represents the formal pseudo code of the above computational procedure.

Algorithm 1 TA+ℓ_q algorithm

1: **Do for** $k=1,\dots,N$**:**
 1.1 Initialize $(X_k^{(0)})_{\Omega_{[k]}} = 0$. Set $\kappa=0$.
 1.2 **Do until the convergence of** $X_k^{(\kappa)}$:
 Generate $X_k^{(\kappa+1)}$ by (30) under SVD (31).
 Reset $X_k^{(\kappa+1)} \to X_k^{(\kappa)}$ and $\kappa+1\to\kappa$.
2: **Output:** Accept $\bar{\mathcal{X}}$ by (33).

The main advantage of the formulation (25) is that it leads to Algorithm 1, which is a path-following procedure as it improves feasible points of each subproblem (27) at each iteration shown by (32), so the convergence is predictable, and the final solution (33) is defined only at the last step. It is different from the following formulation:

$$\min_{\mathcal{X},X_1,\dots,X_{N-1}} \sum_{k=1}^{N-1}\left(\frac{\beta_k}{2}||\mathcal{X}_{[k]} - X_k||^2 + \alpha_k\ell_q(X_k)\right) \quad \text{s.t.} \quad \tilde{\mathcal{X}}_\Omega = \mathcal{X}_\Omega, \quad (34)$$

which was proposed in [9, (26)] for $q=2$ to develop the so called simple low-rank tensor completion via tensor train (SiLRTC-TT) algorithm. Similarly to [9], (34) can be addressed by Algorithm 2, which is termed as LR+ℓ_q algorithm.

Algorithm 2 LR+ℓ_q algorithm

1: **Initialize** $(X_k^{(0)})_{\Omega_{[k]}} = 0$. Set $\kappa=0$.
2: **Do until the convergence of** $\mathcal{X}^{(\kappa)}$**:**

- For $k=1,\dots,N$, generate $X_k^{(\kappa+1)}$ by

$$\begin{aligned}X_k^{(\kappa+1)} &= \arg\min_{X_k}[\frac{\beta_k}{2}||\mathcal{X}_{[k]}^{(\kappa)} - X_k||^2 + \alpha_k\ell_q(X_k)]\\ &= \arg\min_{X_k}[\frac{1}{2}||\mathcal{X}_{[k]}^{(\kappa)} - X_k||^2 + \lambda\ell_q(X_k)]\\ &= U_k^{(\kappa)}\text{diag}[\varphi_\lambda(d_1^{(\kappa)}),\dots,\varphi_\lambda(d_r^{(\kappa)})](V_k^{(\kappa)})^T, \quad (35)\end{aligned}$$

 under the SVD (31).
- Use (2) to generate $\mathcal{X}^{(\kappa+1)}$ by

$$\mathcal{X}^{(\kappa+1)} = \arg\min_{\mathcal{X}_\Omega=\tilde{\mathcal{X}}_\Omega}\sum_{k=1}^{N-1}\frac{\beta_k}{2}||\mathcal{X} - \mathsf{fold}_k(X_k^{(\kappa+1)})||^2 \quad (36)$$

$$\begin{aligned}&= \frac{1}{\sum_{k=1}^{N-1}\beta_k}\sum_{k=1}^{N-1}\beta_k(\mathsf{fold}_k(X_k^{(\kappa+1)}))_{\Omega^c} + \tilde{\mathcal{X}}_\Omega\\ &= \sum_{k=1}^{N-1}\alpha_k(\mathsf{fold}_k(X_k^{(\kappa+1)}))_{\Omega^c} + \tilde{\mathcal{X}}_\Omega. \quad (37)\end{aligned}$$

- Reset $X_k^{(\kappa+1)} \to X_k^{(\kappa)}$, $k=1,\dots,N$, and $\mathcal{X}^{(\kappa+1)} \to \mathcal{X}^{(\kappa)}$, and $\kappa+1\to\kappa$.

3: **Output:** $\mathcal{X}^{(\kappa)}$.

B. Factorization approach

For

$$r_k = \text{rank}(\tilde{\mathcal{X}}^{\text{A}}_{\Omega_{[k]}}) \quad (38)$$

this subsection aims for learning $\mathcal{X}^{\text{A}}_{[k]}$ by U_kV_k with the aid of the sparse matrices

$$U_k \in \mathbb{R}^{m_k\times r_k} \quad \& \quad V_k \in \mathbb{R}^{r_k\times n_k}. \quad (39)$$

To this end, we consider the following optimization problem

$$\min_{\substack{\mathcal{X},U=(U_1,\dots,U_{N-1})\\ V=(V_1,\dots,V_{N-1})}} f(\mathcal{X},U,V) \triangleq \frac{1}{2\lambda}||\tilde{\mathcal{X}}_\Omega - \mathcal{X}_\Omega||^2 + \sum_{k=1}^{N-1}\alpha_k\left[||U_kV_k - \mathcal{X}_{[k]}||^2 + \lambda(\ell_q(U_k)+\ell_q(V_k))\right], \quad (40)$$

which uses the penalty term $\lambda(\ell_q(U_k)+\ell_q(V_k))$ to arrange for the sparse structure of U_k and V_k.

Initialized by $X^{(0)} = \tilde{X}_\Omega$ with $\tilde{\mathcal{X}}^{(0)}_{\Omega_{[k]}} = U_k^{(0)}V_k^{(0)}$, at the κ-th iteration we seek $U^{(\kappa+1)}$ so that

$$\begin{aligned}& f(\mathcal{X}^{(\kappa)},U^{(\kappa+1)},V^{(\kappa)}) < f(\mathcal{X}^{(\kappa)},U^{(\kappa)},V^{(\kappa)}) \quad (41)\\ \Leftrightarrow\quad & f_{1k}^{(\kappa)}(U_k^{(\kappa+1)}) < f_{1k}^{(\kappa)}(U_k^{(\kappa)}), \quad (42)\\ & k=1,\dots,N-1,\end{aligned}$$

for

$$f_{1k}^{(\kappa)}(U_k) \triangleq ||U_kV_k^{(\kappa)} - \mathcal{X}_{[k]}^{(\kappa)}||^2 + \lambda\ell_q(U_k). \quad (43)$$

The function $f_{1k}^{(\kappa)}$ is nonconvex and nonconcave. Applying the inequality (104) in Appendix II gives

$$\begin{aligned} f_{1k}^{(\kappa)}(U_k) \leq\ & ||U_k V_k^{(\kappa)} - \mathcal{X}_{[k]}^{(\kappa)}||^2 + \lambda\Big((1-\frac{q}{2})\ell_q(U_k^{(\kappa)}) \\ & + \frac{q}{2}\langle([(U_k^{(\kappa)})^T]^2)^{q/2-1}, (U_k)^T U_k\rangle\Big) \\ \triangleq\ & \tilde{f}_{1k}^{(\kappa)}(U_k), \end{aligned} \tag{44}$$

which together with $f_{1k}^{(\kappa)}(U_k^{(\kappa)}) = \tilde{f}_{1k}^{(\kappa)}(U_k^{(\kappa)})$ imply that $\tilde{f}_{1k}^{(\kappa)}$ is a tight majorant of $f_{1k}^{(\kappa)}$ at $U_k^{(\kappa)}$ [32]. Thus we generate $U_k^{(\kappa+1)}$ as

$$U_k^{(\kappa+1)} = \arg\min_{U_k} \tilde{f}_{1k}^{(\kappa)}(U_k) \tag{45}$$

$$= \mathcal{X}_{[k]}^{(\kappa)}(V_k^{(\kappa)})^T \left([V_k^{(\kappa)}]^2 + \frac{\lambda q}{2}([(U_k^{(\kappa)})^T]^2)^{q/2-1}\right)^{\dagger}. \tag{46}$$

Since we have $\tilde{f}_{1k}^{(\kappa)}(U_k^{(\kappa+1)}) < \tilde{f}_{1k}^{(\kappa)}(U_k^{(\kappa)}) = f_{1k}^{(\kappa)}(U_k^{(\kappa)})$ due to (45), we then have

$$f_{1k}^{(\kappa)}(U_k^{(\kappa+1)}) \leq \tilde{f}_{1k}^{(\kappa)}(U_k^{(\kappa+1)}) < f_{1k}^{(\kappa)}(U_k^{(\kappa)}) \tag{47}$$

verifying (41).

Next, we seek $V^{(\kappa+1)}$ so that

$$f(\mathcal{X}^{(\kappa)}, U^{(\kappa+1)}, V^{(\kappa+1)}) < f(\mathcal{X}^{(\kappa)}, U^{(\kappa+1)}, V^{(\kappa)}) \tag{48}$$

$$\Leftrightarrow \quad f_{2k}^{(\kappa)}(V_k^{(\kappa+1)}) < f_{2k}^{(\kappa)}(V_k^{(\kappa)}), \quad k = 1, \ldots, N-1, \tag{49}$$

for

$$f_{2k}^{(\kappa)}(V_k) \triangleq ||U_k^{(\kappa+1)} V_k - \mathcal{X}_{[k]}^{(\kappa)}||^2 + \lambda \ell_q(V_k). \tag{50}$$

Applying the inequality (103) in Appendix II gives

$$\begin{aligned} f_{2k}^{(\kappa)}(V_k) \leq\ & ||U_k^{(\kappa+1)} V_k - \mathcal{X}_{[k]}^{(\kappa)}||^2 + \lambda\Big((1-\frac{q}{2})\ell_q(V_k^{(\kappa)}) \\ & + \frac{q}{2}\langle([V_k^{(\kappa)}]^2)^{q/2-1}, [V^{(\kappa)}]^2\rangle\Big) \\ \triangleq\ & \tilde{f}_{2k}^{(\kappa)}(V_k), \end{aligned} \tag{51}$$

which together with $f_{2k}^{(\kappa)}(V_k^{(\kappa)}) = \tilde{f}_{2k}^{(\kappa)}(V_k^{(\kappa)})$ imply that $\tilde{f}_{2k}^{(\kappa)}$ is a tight majorant of $f_{2k}^{(\kappa)}$ at $V_k^{(\kappa)}$ [32]. Thus we generate $V_k^{(\kappa+1)}$ as

$$\begin{aligned} V_k^{(\kappa+1)} =\ & \arg\min_{V_k} \tilde{f}_{2k}^{(\kappa)}(V_k) \\ =\ & \left([(U_k^{(\kappa+1)})^T]^2 + \frac{\lambda q}{2}([V_k^{(\kappa)}]^2)^{q/2-1}\right)^{\dagger} \\ & (U_k^{(\kappa+1)})^T \mathcal{X}_{[k]}^{(\kappa)}, \end{aligned} \tag{52}$$

which like (46) verifies (48).

Lastly, we introduce

$$\begin{aligned} f^{(\kappa)}(\mathcal{X}) \triangleq\ & \frac{1}{2\lambda}||\tilde{\mathcal{X}}_\Omega + \mathcal{X}_{\Omega^c}^{(\kappa)} - \mathcal{X}||^2 \\ & + \sum_{k=1}^{N-1}\alpha_k\Big(||U_k^{(\kappa+1)}V_k^{(\kappa+1)} - \mathcal{X}_{[k]}||^2 \\ & + \lambda(\ell_q(U_k^{(\kappa+1)}) + \ell_q(V_k^{(\kappa+1)}))\Big), \end{aligned} \tag{53}$$

which is a tight majorant of $f(., U^{(\kappa+1)}, V^{(\kappa+1)})$ at $\mathcal{X}^{(\kappa)}$. Then we use (2) to generate $\mathcal{X}^{(\kappa+1)}$ by

$$\begin{aligned} \mathcal{X}^{(\kappa+1)} =\ & \arg\min_{\mathcal{X}} f^{(\kappa)}(\mathcal{X}) \\ =\ & \arg\min_{\mathcal{X}}\Big[\frac{1}{2\lambda}||\tilde{\mathcal{X}}_\Omega + \mathcal{X}_{\Omega^c}^{(\kappa)} - \mathcal{X}||^2 \\ & + \sum_{k=1}^{N-1}\alpha_k||\mathsf{fold}_k(U_k^{(\kappa+1)}V_k^{(\kappa+1)}) - \mathcal{X}||^2\Big] \\ =\ & \frac{2\lambda}{2\lambda+1}\Big[\frac{1}{2\lambda}(\tilde{\mathcal{X}}_\Omega + \mathcal{X}_{\Omega^c}^{(\kappa)}) \\ & + \sum_{k=1}^{N-1}\alpha_k \mathsf{fold}_k(U_k^{(\kappa+1)}V_k^{(\kappa+1)})\Big], \end{aligned} \tag{54}$$

which yields

$$\begin{aligned} & f(\mathcal{X}^{(\kappa+1)}, U^{(\kappa+1)}, V^{(\kappa+1)}) \leq f^{(\kappa)}(\mathcal{X}^{(\kappa+1)}) < \\ & f^{(\kappa)}(\mathcal{X}^{(\kappa)}) = f(\mathcal{X}^{(\kappa+1)}, U^{(\kappa+1)}, V^{(\kappa+1)}). \end{aligned} \tag{55}$$

Thus, based on (41), (48), and (55), $(U^{(\kappa+1)}, V^{(\kappa+1)}, \mathcal{X}^{(\kappa+1)})$ generated by (46), (52), and (54) is a better point than $(U^{(\kappa)}, V^{(\kappa)}, \mathcal{X}^{(\kappa)})$

$$f(\mathcal{X}^{(\kappa)}, U^{(\kappa)}, V^{(\kappa)}) < f(\mathcal{X}^{(\kappa+1)}, U^{(\kappa+1)}, V^{(\kappa+1)}). \tag{56}$$

As such, the sequence $\{(\mathcal{X}^{(\kappa)}, U^{(\kappa)}, V^{(\kappa)})\}$, which is of improved points, will converge.

Algorithm 3 represents the formal pseudo code of the above computational procedure, which is termed as the *sparse factorization algorithm (SFA)*, which needs at least two SVDs for the pseudo-inversions in (46) and (52).

Algorithm 3 Sparse factorization algorithm (SFA)

1: **Initialize** $X^{(0)} = \tilde{X}_\Omega$ with $\tilde{\mathcal{X}}_{\Omega_{[k]}}^{(0)} = U_k^{(0)}V_k^{(0)}$, $k = 1, \ldots, N$. Set $\kappa = 0$.

2: **Do until the convergence of** $\mathcal{X}^{(\kappa)}$**:**

- For $k = 1, \ldots, N$, generate $U_k^{(\kappa+1)}$ and $V_k^{(\kappa+1)}$ by (46) and (52) respectively, and then $\mathcal{X}^{(\kappa+1)}$ by (54).
- Reset $U_k^{(\kappa+1)} \to U_k^{(\kappa)}$ and $V_k^{(\kappa+1)} \to V_k^{(\kappa)}$ $k = 1, \ldots, N$, and $\mathcal{X}^{(\kappa+1)} \to \mathcal{X}^{(\kappa)}$, and $\kappa+1 \to \kappa$.

3: **Output:** $\mathcal{X}^{(\kappa)}$.

IV. Frobenius-norm-based SVD-free tensor completion

Upon recalling the definition (38) of r_k, we learn $\mathcal{X}_{[k]}$ by $U_k V_k$, with their size given by (39) with the aid of the following optimization problem

$$\begin{aligned} \min_{\substack{\mathcal{X}, U=(U_1,\ldots,U_{N-1}) \\ V=(V_1,\ldots,V_{N-1})}} f(\mathcal{X}, U, V) \triangleq\ & \frac{1}{2\lambda}||\tilde{\mathcal{X}}_\Omega - \mathcal{X}_\Omega||^2 \\ & + \sum_{k=1}^{N-1}\alpha_k||U_k V_k - \mathcal{X}_{[k]}||^2, \end{aligned} \tag{57}$$

which does not impose any sparsity-related objectives, unlike (40).

Note that (57) constitutes much more flexible and over-fitting free formulation than the following formulation used in [33], [34] and TMac-TT (parallel matrix factorization via tensor train) technique in [9]

$$\min_{U,V,\mathcal{X}} \sum_{k=1}^{K} \frac{\alpha_k}{2}||U_k V_k - \mathcal{X}_{[k]}||^2 \quad \text{s.t} \quad \mathcal{X}_\Omega = \tilde{\mathcal{X}}_\Omega. \tag{58}$$

The TMac-TT algorithm [9] is formally defined in Algorithm 4.

Algorithm 4 TMac-TT algorithm [9]

1: **Initialize** $X^{(0)} = \tilde{X}_\Omega$ with $\tilde{\mathcal{X}}^{(0)}_{\Omega_{[k]}} = U_k^{(0)} V_k^{(0)}$, $k = 1,\dots,N$. Set $\kappa = 0$.

2: **Do until the convergence of $\mathcal{X}^{(\kappa)}$:**

- For $k = 1,\dots,N$, generate $U_k^{(\kappa+1)}$ and $V_k^{(\kappa+1)}$ by

$$U_k^{(\kappa+1)} = \mathcal{X}^{(\kappa)}_{[k]} (V_k^{(\kappa)})^T \left(V_k^{(\kappa)} (V_k^{(\kappa)})^T\right)^\dagger, \tag{59}$$

and

$$V_k^{(\kappa+1)} = ((U_k^{(\kappa+1)})^T U_k^{(\kappa+1)})^\dagger (U_k^{(\kappa+1)})^T \mathcal{X}^{(\kappa)}_{[k]}, \tag{60}$$

and then $\mathcal{X}^{(\kappa+1)}$ by

$$\begin{aligned}\mathcal{X}^{(\kappa+1)} &= \arg\min_{\mathcal{X}_\Omega = \tilde{\mathcal{X}}_\Omega} \sum_{k=1}^{K} \alpha_k ||\mathsf{fold}_k(U_k^{(\kappa+1)} V_k^{(\kappa+1)}) \\ &\qquad - \mathcal{X}||^2 \qquad (61)\\ &= \sum_{k=1}^{N-1} \alpha_k (\mathsf{fold}_k((U_k^{(\kappa+1)} V_k^{(\kappa+1)})))_{\Omega^c} \\ &\qquad + \tilde{\mathcal{X}}_\Omega, \qquad (62)\end{aligned}$$

with α_k defined from (26).
- Reset $U_k^{(\kappa+1)} \to U_k^{(\kappa)}$ and $V_k^{(\kappa+1)} \to V_k^{(\kappa)}$, $k = 1,\dots,N$, and $\mathcal{X}^{(\kappa+1)} \to \mathcal{X}^{(\kappa)}$, and $\kappa + 1 \to \kappa$.

3: **Output:** $\mathcal{X}^{(\kappa)}$.

One can see that there are at least two SVDs made in (59) and (60) for pseudo-inversions.

Now we address (57) via SVD-free iterations as follows. Let $(\mathcal{X}^{(\kappa)}, U^{(\kappa)}, V^{(\kappa)})$ be the outcome of the $(\kappa-1)$-st iteration. At the κ-th iteration we seek $(U^{(\kappa+1)}, V^{(\kappa+1)}, \mathcal{X}^{(\kappa+1)})$ so that

$$\begin{aligned} g(\mathcal{X}^{(\kappa)}, U^{(\kappa)}, V^{(\kappa)}) &> g(\mathcal{X}^{(\kappa)}, U^{(\kappa+1)}, V^{(\kappa)}) && (63)\\ &> g(\mathcal{X}^{(\kappa)}, U^{(\kappa+1)}, V^{(\kappa+1)}) && (64)\\ &> g(\mathcal{X}^{(\kappa+1)}, U^{(\kappa+1)}, V^{(\kappa+1)}) && (65)\end{aligned}$$

Firstly, (63) is equivalent to

$$g_{1k}^{(\kappa)}(U_k^{(\kappa+1)}) < g_{1k}^{(\kappa)}(U_k^{(\kappa)}), \tag{66}$$

for

$$g_{1k}^{(\kappa)}(U_k) \triangleq ||U_k V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}||^2. \tag{67}$$

Now we can write

$$\begin{aligned} g_{1k}^{(\kappa)}(U_k^{(\kappa)} + \Delta_k) &= ||(U_k^{(\kappa)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}) + \Delta_k V_k^{(\kappa)}||^2 \\ &= g_{1k}^{(\kappa)}(U^{(\kappa)}) + 2\langle U_k^{(\kappa)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}, \Delta_k V_k^{(\kappa)}\rangle \\ &\quad + ||\Delta_k V_k^{(\kappa)}||^2. \end{aligned} \tag{68}$$

Here and after, $\langle A, B\rangle$ is the dot product of the matrices A and B, i.e. it is trace$(A^T B)$. For

$$\Delta_k \triangleq -t_k \left(U_k^{(\kappa)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}\right)(V_k^{(\kappa)})^T \tag{69}$$

one has

$$g_{1k}^{(\kappa)}(U^{(\kappa)} + \Delta_k) = g_{1k}^{(\kappa)}(U^{(\kappa)}) + \eta_{1k}(t_k) \tag{70}$$

where

$$\eta_{1k}(t_k) \triangleq -2a_k^{(\kappa)} t_k + b_k^{(k)} (t_k)^2 \tag{71}$$

with

$$\begin{aligned} 0 < a_k^{(\kappa)} &\triangleq ||\left(U_k^{(\kappa)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}\right)(V_k^{(\kappa)})^T||^2, \\ 0 < b_k^{(k)} &\triangleq ||\left(U_k^{(\kappa)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}\right)(V_k^{(\kappa)})^T V_k^{(\kappa)}||^2. \end{aligned} \tag{72}$$

Thus, choosing

$$t_k^{(\kappa)} \triangleq \arg\min_{\tau_k} \eta_{1k}(\tau_k) = a_k^{(\kappa)}/b_k^{(k)} \tag{73}$$

leads to

$$\eta_{1k}(t_k^{(\kappa)}) = -(a_k^{(\kappa)})^2/b_k^{(k)} < 0 \tag{74}$$

that results in (66). In short, we generate $U_k^{(\kappa+1)}$ satisfying (66)/(63) by

$$U_k^{(\kappa+1)} = U_k^{(\kappa)} - t_k^{(\kappa)} \left(U_k^{(\kappa)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}\right)(V_k^{(\kappa)})^T \tag{75}$$

for $t_k^{(\kappa)}$ defined from (73).

Analogously, (64) is equivalent to

$$g_{2k}^{(\kappa)}(V_k^{(\kappa+1)}) < g_{2k}^{(\kappa)}(V_k^{(\kappa)}) \tag{76}$$

for

$$g_{2k}^{(\kappa)}(V_k) \triangleq ||U_k^{(\kappa+1)} V_k - \mathcal{X}^{(\kappa)}_{[k]}||^2. \tag{77}$$

Now, we can write

$$\begin{aligned} g_{2k}^{(\kappa)}(V_k^{(\kappa)} + \Delta_k) &= \\ ||(U_k^{(\kappa+1)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}) + U_k^{(\kappa+1)}\Delta_k||^2 &= \\ g_{2k}^{(\kappa)}(V_k^{(\kappa)}) + ||U_k^{(\kappa+1)}\Delta_k||^2 & \\ +2\langle (U_k^{(\kappa+1)})^T (U_k^{(\kappa+1)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}), \Delta_k\rangle. & \end{aligned} \tag{78}$$

For

$$\Delta_k \triangleq -\tau_k (U_k^{(\kappa+1)})^T (U_k^{(\kappa+1)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}) \tag{79}$$

one has

$$g_{2k}^{(\kappa)}(V_k^{(\kappa)} + \Delta_k) = g_{2k}^{(\kappa)}(V_k^{(\kappa)}) + \eta_{2k}(\tau_k) \tag{80}$$

for

$$\eta_{2k}(\tau_k) = -2\tilde{a}_k^{(\kappa)} \tau_k + \tilde{b}_k^{(\kappa)} (\tau_k)^2 \tag{81}$$

with

$$\begin{aligned} 0 < \tilde{a}_k^{(\kappa)} &\triangleq ||(U_k^{(\kappa+1)})^T \left(U_k^{(\kappa+1)} V_k^{(\kappa)} - \mathcal{X}_k^{(\kappa)}\right)||^2, \\ 0 < \tilde{b}_k^{(\kappa)} &\triangleq ||U_k^{(\kappa+1)} (U_k^{(\kappa+1)})^T \left(U_k^{(\kappa+1)} V_k^{(\kappa)} - \mathcal{X}^{(\kappa)}_{[k]}\right)||^2. \end{aligned} \tag{82}$$

Thus, choosing

$$\tau_k^{(\kappa)} \triangleq \arg\min_{\tau_k} \eta_{2k}(\tau_k) = \tilde{a}_k^{(\kappa)}/\tilde{b}_k^{(\kappa)} \tag{83}$$

leads to

$$\eta_{2k}(\tau_k^{(\kappa)}) = -(\tilde{a}_k^{(\kappa)})^2/\tilde{b}_k^{(\kappa)} < 0 \tag{84}$$

that makes (76). In short, we generate $V^{(\kappa+1)}$ satisfying (76)/(64) by

$$V_k^{(\kappa+1)} = V_k^{(\kappa)} - \tau_k^{(\kappa)} (U_k^{(\kappa+1)})^T \left(U_k^{(\kappa+1)} V_k^{(\kappa)} - \mathcal{X}_{[k]}^{(\kappa)} \right) \tag{85}$$

for $\tau_k^{(\kappa)}$ defined in (83).

Finally, we generate $\mathcal{X}^{(\kappa+1)}$ satisfying (65) by

$$\begin{aligned}
\mathcal{X}^{(\kappa+1)} &= \arg\min_{\mathcal{X}} f(\mathcal{X}, U^{(\kappa+1)}, V^{(\kappa+1)}) \\
\Leftrightarrow \mathcal{X}^{(\kappa+1)} &= \arg\min_{\mathcal{X}} \left[\frac{1}{2\lambda} ||\tilde{\mathcal{X}}_\Omega + \mathcal{X}_{\Omega^c}^{(\kappa)} - \mathcal{X}||^2 \right. \\
&\quad \left. + \sum_{k=1}^{N-1} \alpha_k ||\mathsf{fold}_k \left(U_k^{(\kappa+1)} V_k^{(\kappa+1)} \right) - \mathcal{X}||^2 \right]
\end{aligned} \tag{86}$$

$$\begin{aligned}
\Leftrightarrow \mathcal{X}^{(\kappa+1)} = \frac{2\lambda}{2\lambda+1} &\left[\frac{1}{2\lambda} (\tilde{\mathcal{X}}_\Omega + \mathcal{X}_{\Omega^c}^{(\kappa)}) \right. \\
&\left. + \sum_{k=1}^{N-1} \alpha_k \mathsf{fold}_k \left(U_k^{(\kappa+1)} V_k^{(\kappa+1)} \right) \right].
\end{aligned} \tag{87}$$

Algorithm 5 represents the formal pseudo code of the above computational procedure, which is termed as the *SVD-free Algorithm*.

Algorithm 5 SVD-free Algorithm

1: **Initialize** $X^{(0)} = \tilde{X}_\Omega$ with $\tilde{\mathcal{X}}_{\Omega_{[k]}}^{(0)} = U_k^{(0)} V_k^{(0)}$, $k = 1, \dots, N$. Set $\kappa = 0$.
2: **Do until the convergence of $\mathcal{X}^{(\kappa)}$:**
- For $k = 1, \dots, N$, generate $U_k^{(\kappa+1)}$ and $V_k^{(\kappa+1)}$ by (75) and (85) respectively, and then $\mathcal{X}^{(\kappa+1)}$ by (87).
- Reset $U_k^{(\kappa+1)} \to U_k^{(\kappa)}$ and $V_k^{(\kappa+1)} \to V_k^{(\kappa)}$ $k = 1, \dots, N$, and $\mathcal{X}^{(\kappa+1)} \to \mathcal{X}^{(\kappa)}$, and $\kappa + 1 \to \kappa$.

3: **Output:** $\mathcal{X}^{(\kappa)}$.

V. Simulations

We use the following shorthands to specify the proposed implementations: "SoftImput [7], [35]" refers to the SoftImput algorithm of [7], [35], which is the state-of-the-art ℓ_1-norm based matrix completion; simple ℓ_p refers to that by iterating (15)-(16); "TMacTT [9]" refers to to TMacTT algorithm of [9], which is recaped in Algorithm 4; "TA+ℓ_q" refers to the TA+ℓ_q algorithm 1; "LR+ℓ_q" refers to the LR+ℓ_q algorithm 2, which is an ℓ_q-extension of SiLRTC-TT [9]; "SFA" refers to the SFA 3, and "SVD-free" refers to the SVD-free Algorithm 5. Note that both SiLRTC-TT [9] and TMacTT [9] outperform the completion algorithm [36], which deals with only a single matricization.

The simulations have been performed in Google Colab using the following hardware: (i) CPU: 1 × single core Xeon processor with hyper-threading at 2.3GHz; (ii) GPU: 1 × Tesla K80 having 2496 CUDA cores; (iii) RAM: 13 GB available, and (iv) hard disk: 40 GB available.

A. Matrix completion

For a 2D-index set of size $m \times n$, the unbalanced ratio is defined by

$$u_r = \frac{\max\{m,n\}}{\min\{m,n\}}, \tag{88}$$

while the missing ratio is defined by

$$m_r = 1 - \frac{|\Omega|}{mn}, \tag{89}$$

where $|\Omega|$ is the cardinality of Ω, which is the set of indices of the observed entries. The testing matrices are created from the original matrices by randomly generating missing entries according to this m_r. The algorithm performance is quantified by the following relative square error (RSE) between the completed matrix $\hat{X}$ and the original X

$$\text{RSE} = \frac{||\hat{X}_\Omega - X_\Omega||}{||X_\Omega||}. \tag{90}$$

We use the Netflix dataset for rating $17,770$ movies by $480,189$ viewers [7, Sec. 10], which relies on 1% of entries $(100,480,507)$ observed because each viewer can only rate a small fraction of the movies. We also use the Movilens dataset [37] containing the rating of $22,156$ movies by $10,533$ viewers. Tables II and III provide the size $m \times n$ of X taken from these datasets and that of the corresponding Nth-order tensor $\mathcal{X}$ by TA [25].

We run our simulations for $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\lambda \in \{1, 2, \dots, 10\}$ for TA+ℓ_q, LR+ℓ_q and SFA but $\lambda \in \{0.1, 0.2, \dots, 1\}$ for SVD-free and then the best achieved RSE is used for the RSE performance evaluation. Both Tables VI and V show that:

- The performances of SoftImpute [7], [35] and simple ℓ_q are similar but the computational complexity of the latter is extremely light as it is not only free from SVD but also from matrix-product calculations. As expected, they gradually perform worse as the unbalanced ratio increases because they aim for optimizing the matrix-rank;
- The performances of TMacTT [9] and SVD free are similar but the computational load of the latter is much lighter than that of the former. This also justifies the flexibility preference of the formulation (57) over (58). Since the matrix sparsity is not incorporated into these formulations, TTMacTT [9] and SVD free are outperformed by TA+ℓ_q, LR+ℓ_q and SFA;
- The performance of TA+ℓ_q is slightly better than that of LR+ℓ_q, justifying the flexibility preference of the formulation (25) over (34);
- The performance of SFA is consistently best among all the algorithms considered, justifying the formulation (40);
- In contrast to SoftImpute and simple ℓ_q, TA+ℓ_q, LR+ℓ_q and SFA perform much better, while TTMacTT [9] and SVD free perform indifferently as the unbalance ratio increases. This demonstrate the advantages of TA [25] and TT-decomposition in handling unbalanced matrices.

Table VI shows the computational time in seconds of SoftImpute, Simple l_q, TA+l_q, LR+l_q, SFA, TMacTT, SVD-free on Netflix dataset matrix of size 1296×256.

$m=\prod_{k=1}^N J_k^a$	$n=\prod_{k=1}^N J_k^b$	Size of the tensor $\mathcal{X}$ by TA
$1296=3^4\times2^4$	$256=2^8$	$6\times6\times6\times6\times4\times4\times4\times4$
$2592=3^4\times2^4$	$512=2^9$	$6\times6\times6\times6\times4\times4\times4\times4\times4$
$7776=3^5\times2^5$	$1024=2^{10}$	$6\times6\times6\times6\times6\times4\times4\times4\times4\times4$
$59049=3^{10}$	$1024=2^{10}$	$6\times6\times6\times6\times6\times6\times6\times6\times6\times6$
$82944=4^5\times3^4$	$512=2^9$	$8\times8\times8\times8\times8\times6\times6\times6\times6$

Table II: Size of X taken from the Netflix dataset and the corresponding Nth-order tensor $\mathcal{X}$ by TA [25]

$m=\prod_{k=1}^N J_k^a$	$n=\prod_{k=1}^N J_k^b$	Size of $\mathcal{X}$ by TA
$1296=3^4\times2^4$	$256=2^8$	$6\times6\times6\times6\times4\times4\times4\times4$
$1944=3^5\times2^3$	$256=2^8$	$6\times6\times6\times6\times6\times4\times4\times4$
$2592=3^4\times2^4$	$512=2^9$	$6\times6\times6\times6\times4\times4\times4\times4\times4$
$3888=3^5\times2^4$	$512=2^9$	$6\times6\times6\times6\times6\times4\times4\times4\times4$
$2592=3^6\times2^3$	$512=2^9$	$6\times6\times6\times6\times6\times6\times4\times4\times4$
$7776=3^5\times2^5$	$1024=2^10$	$6\times6\times6\times6\times6\times4\times4\times4\times4\times4$

Table III: Size of X taken from the Movielen dataset and the corresponding Nth-order tensor $\mathcal{X}$ by TA [25]

(m,n)	u_r	m_r	SoftImpute [7], [35]	simple ℓ_q	TA+ℓ_q	LR+ℓ_q	SFA	TMacTT [9]	SVD-free
(1296,256)	5.06	0.92	0.066	0.055	0.078	0.084	0.056	0.066	0.077
(2592,512)	5.06	0.91	0.089	0.067	0.052	0.069	0.024	0.052	0.058
(7776,1024)	7.59	0.91	0.124	0.092	0.037	0.032	0.016	0.052	0.056
(59049,1024)	57.67	0.91	0.136	0.087	0.023	0.025	0.012	0.047	0.051
(82944,512)	162	0.92	0.102	0.052	0.024	0.026	0.011	0.055	0.056

Table IV: RSE performance for Netflix completion

(m,n)	u_r	m_r	SoftImpute [7], [35]	simple ℓ_q	TA+ℓ_q	LR+ℓ_q	SFA	TMacTT [9]	SVD-free
(1296,256)	5.06	0.96	0.183	0.095	0.066	0.083	0.063	0.118	0.115
(1944,256)	7.59	0.96	0.187	0.094	0.062	0.076	0.050	0.122	0.100
(2592,512)	5.06	0.96	0.221	0.111	0.052	0.060	0.026	0.107	0.097
(3888,512)	7.59	0.97	0.220	0.110	0.053	0.062	0.034	0.105	0.088
(5832,512)	11.39	0.99	0.226	0.092	0.093	0.071	0.050	0.093	0.123
(7776,1024)	7.59	0.99	0.250	0.100	0.057	0.080	0.072	0.081	0.167

Table V: The RSE performance by Movielen completion

Table VI: The computational time in seconds for recovering Neflix dataset matrix of size 1296×256 with $(m_r,u_r)=(0.92,5.06)$

SoftImpute	simple ℓ_q	TA+ℓ_q	LR+ℓ_q	FSA	TMacTT	SVD-free
28.82	17.10	159	146	55.95	36.7	22.08

B. Third order tensor completion

Our objective in this subsection is to recover a colour image of the standard height $m=256$ and width $n=256$, which is represented by a third-order tensor $\mathcal{X}$ of size $m\times n\times3$. Thus $\mathcal{X}(i,j,1)$, $\mathcal{X}(i,j,2)$, and $\mathcal{X}(i,j,3)$ describe the redness, blueness, and greenness of the (i,j)-th pixel. The index set of its observed entries is $\Omega\triangleq\bar{\Omega}\times\{1,2,3\}$ with $\bar{\Omega}\triangleq\Omega_1\times\Omega_2\subset\{1,\dots,m\}\times\{1,\dots,n\}$. The unbalanced ratio u_r is still defined by (88) but the missing ratio m_r is defined by

$$m_r=1-\frac{|\bar{\Omega}|}{mn}, \tag{91}$$

instead of (89), while the RSE between the complete tensor $\hat{\mathcal{X}}$ and the original $\mathcal{X}$ is defined by

$$\text{RSE}=\frac{||\hat{\mathcal{X}}_\Omega-\mathcal{X}_\Omega||}{||\mathcal{X}_\Omega||}, \tag{92}$$

instead of (90). The test images are created from the original images by randomly generating missing entries according to the missing ratio m_r defined by (89). Upon encoding 2D indices $(i,j)\in\{1,\dots,256\}\times\{1,\dots,256\}$ by 8-digit words $i_1i_2\dots i_8$ associated with $i_n\in\{1,2,3,4\}$, $n=1,\dots8$, we thus cast the third-order tensor X of size $256\times256\times3$ into a ninth-order tensor $\mathcal{X}$ of size $4\times4\times4\times4\times4\times4\times4\times4\times3$, and then apply Algorithms 1-4 for completing the latter. In what follows, we define the unbalanced ratio of Ω by

$$u_\Omega\triangleq|\Omega_1|/|\Omega_2|. \tag{93}$$

1) Examples for the balanced ratio of $u_\Omega=1$: For these examples, TMacTT [9] has been shown to outperform all existing algorithms [9]. Tables VIII, IX, and X provide the recovery results for the popular Lena image by TA+ℓ_q, LR+ℓ_q, and SFA under different q and λ. Observe that TA+ℓ_q achieves its best RSE at $(q,\lambda)=(0.5,2)$, LR+ℓ_q achieves its best RSE at $(q,\lambda)=(0.5,10)$, while SFA achieves its best RSE at $(q,\lambda)=(0.3,5/4)$. Similar results and the optimal values (q,λ) for the particular algorithms are also observed when recovering the Pepper image.

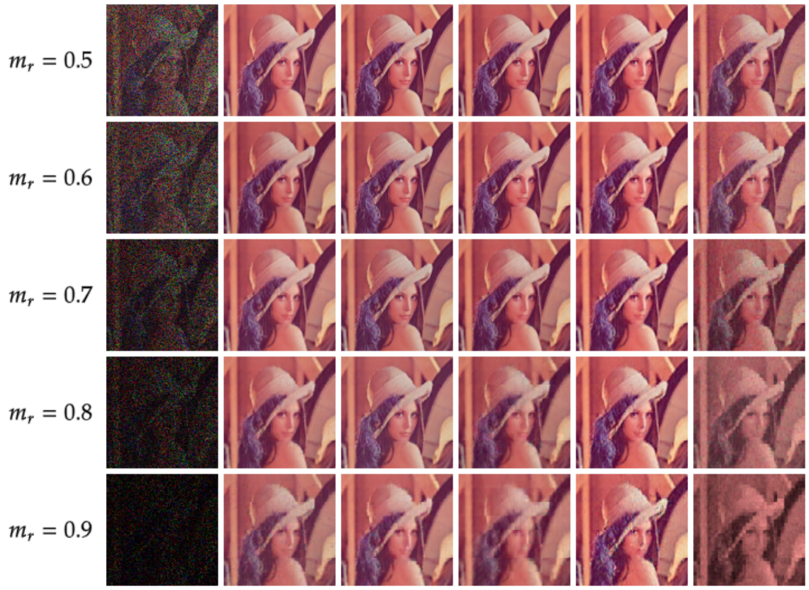

Figure 1: The images recovered from the Lena image having missing pixels. The first column top down to the bottom represent the incomplete images associated with $m_r \in [0.5, 0.9]$. The next columns are the images recovered by TA+ℓ_q, LR+ℓ_q, SFA, TMacTT [9], and SVD-free

Image	m_r	TA+ℓ_q	LR+ℓ_q	SFA	TMacTT [9]	SVD-free
Lena	0.5	0.057	0.036	0.038	0.061	0.081
	0.6	0.064	0.046	0.051	0.069	0.104
	0.7	0.076	0.060	0.065	0.077	0.142
	0.8	0.097	0.076	0.092	0.088	0.166
	0.9	0.129	0.108	0.134	0.109	0.186
Peppers	0.5	0.072	0.058	0.063	0.078	0.105
	0.6	0.086	0.069	0.075	0.088	0.129
	0.7	0.105	0.085	0.097	0.101	0.162
	0.8	0.131	0.108	0.132	0.117	0.191
	0.9	0.187	0.150	0.194	0.169	0.267

Table VII: The RSE performances in recovering the Lena and Peppers images under different values of m_r with $u_\Omega = 1$

Figure 2: The images recovered from the Pepper image having missing pixels. The first column top down to the bottom represent the incomplete images associated with $m_r \in [0.5, 0.9]$. The next columns are the images recovered by TA+ℓ_q, LR+ℓ_q, SFA, TMacTT [9], and SVD-free.

	$\lambda=1$	$\lambda=2$	$\lambda=10$	$\lambda=20$	$\lambda=100$
$q=0.1$	0.130	0.318	0.628	0.668	0.699
$q=0.3$	0.055	0.069	0.360	0.531	0.673
$q=0.5$	0.054	0.050	0.108	0.291	0.622
$q=0.7$	0.059	0.051	0.058	0.100	0.531
$q=0.9$	0.067	0.057	0.053	0.063	0.389

Table VIII: The RSE in recovering the Lena image by TA+ℓ_q

	$\lambda=1$	$\lambda=2$	$\lambda=10$	$\lambda=20$	$\lambda=100$
$q=0.1$	0.043	0.258	0.626	, 0.667	0.707
$q=0.3$	0.043	0.036	0.322	0.522	0.671
$q=0.5$	0.044	0.038	0.035	0.241	0.618
$q=0.7$	0.047	0.041	0.038	0.039	0.521
$q=0.9$	0.054	0.047	0.044	0.048	0.366

Table IX: The RSE in recovering the Lena image by LR+ℓ_q

	$\lambda=2.5$	$\lambda=2$	$\lambda=5/3$	$\lambda=10/7$	$\lambda=5/4$
$q=0.1$	0.069	0.233	0.338	0.408	0.458
$q=0.3$	0.048	0.045	0.043	0.042	0.040
$q=0.5$	0.051	0.049	0.048	0.046	0.045
$q=0.7$	0.059	0.058	0.047	0.048	0.045
$q=0.9$	0.109	0.077	0.052	0.052	0.052

Table X: The RSE in recovering the Lena image by SFA

Furthermore, Table VII provides the RSE performance of all implemented algorithms. LR+ℓ_q consistently outperforms the other algorithms, including the state-of-the-art TMacTT [9], so ℓ_q-optimization is indeed helpful. TMacTT [9] is also outperformed by TA+ℓ_q and SFA for $m_r \in \{0.5, 0.6, 0.7\}$.

The images with missing pixels and the recovered images are presented in Figs. 1 and 2.

2) Unbalanced examples with $u_\Omega < 1$: Tables XI, XII, and XIII, characterize the recovery results for the Lena image by TA+ℓ_q, LR+ℓ_q, and SFA for the missing ratio m_r and u_Ω fixed at 90% and 0.7 upon varying q and λ. TA+ℓ_q achieves its best RSE at $(q,\lambda) = (0.7, 1)$, LR+ℓ_q achieves its best RSE at $(q,\lambda) = (0.5, 1)$, while SFA achieves its best RSE at $(q,\lambda) = (0.7, 2)$. Similar results and the optimal values (q,λ) for the particular algorithms are observed in recovering the Pepper image.

Table XIV provides the RSE performance of all implemented algorithms, which is much worse than that for $u_\Omega = 1$ provided by Table VII. In fact, the RSE performance is monotonically degraded vs u_Ω. The RSE performance of TMacTT [9] is seen to quickly drop with u_Ω decreased and it is even outperformed by SVD-free. TA+ℓ_q and LR+ℓ_q consistently outperform the others, where the RSE performance of LR+ℓ_q is the best. Furthermore, it is not sensitive to u_Ω, demonstrating its efficiency in dealing with unbalanced sets of observable entries. The SFA performs relatively well for $u_\Omega \le 0.7$.

	$\lambda=1$	$\lambda=2$	$\lambda=10$	$\lambda=20$	$\lambda=100$
$q=0.1$	0.644	0.750	0.856	0.873	0.889
$q=0.3$	0.241	0.425	0.758	0.819	0.875
$q=0.5$	0.127	0.164	0.595	0.727	0.853
$q=0.7$	0.120	0.123	0.368	0.577	0.816
$q=0.9$	0.133	0.128	0.188	0.383	0.758

Table XI: The RSE in recovering the Lena image by TA+ℓ_q for the missing ratio $m_r = 90\%$ and $u_\Omega = 0.7$

	$\lambda=1$	$\lambda=2$	$\lambda=10$	$\lambda=20$	$\lambda=100$
$q=0.1$	0.689	0.783	0.866	0.878	0.893
$q=0.3$	0.302	0.482	0.793	0.839	0.882
$q=0.5$	0.101	0.188	0.663	0.771	0.867
$q=0.7$	0.105	0.102	0.450	0.653	0.839
$q=0.9$	0.116	0.109	0.234	0.485	0.799

Table XII: The RSE in recovering the Lena image by LR+ℓ_q for the missing ratio $m_r = 90\%$ and $u_\Omega = 0.7$

	$\lambda=2.5$	$\lambda=2$	$\lambda=5/3$	$\lambda=10/7$	$\lambda=5/4$
$q=0.1$	0.592	0.653	0.694	0.725	0.747
$q=0.3$	0.277	0.315	0.343	0.372	0.394
$q=0.5$	0.209	0.259	0.285	0.296	0.319
$q=0.7$	0.223	0.156	0.213	0.252	0.281
$q=0.9$	0.398	0.304	0.161	0.179	0.181

Table XIII: The RSE in recovering the Lena image by SFA for the missing ratio $m_r = 90\%$ and $u_\Omega = 0.7$

The images having missing pixels and the images recovered by TA+ℓ_q, LR+ℓ_q, SFA, TMacTT [9], and SVD-free are presented in Figs. 3 and 4, which are of much worse quality than their counterparts in Figs. 1 and 2 associated with $u_\Omega = 1$.

Table XV shows the computational time (in seconds) of TA+l_q, LR+l_q, FSA, TMacTT, SVD-free in recovering Lena image for $m_r = 0.8$ and $u_r \in \{0.3, \ldots, 0.9\}$.

Table XV: The computational time in seconds for recovering Lena image for $m_r = 0.8$ and different u_r

u_r	TA+ℓ_q	LR+ℓ_q	FSA	TMacTT	SVD-free
0.3	206	73.41	36.13	32.77	27.6
0.5	114	25.38	160	30.28	26.7
0.7	97.35	16.73	290	25.26	24.2
0.9	95.35	15.39	501	27.95	25.1

VI. Conclusions

We have developed efficient techniques for completing unbalanced and sparse matrices and third-order tensors with the index set of observable entries also of flexible structure, which could not be efficiently addressed by the state-of-the-art completion algorithms. Based on encoding the 2D-index set by an N-dimensional index set, the incomplete matrices and tensors are cast into high-order but low-dimensional tensors for carrying out their completion. We have proposed several novel algorithms for completing such matrices and tensors, which are efficient in terms of their completion performance or computational complexity. Its applications to data processing for cyber physical systems are under current study.

Appendix I: the proof of (32)

It is plausible that

$$\begin{aligned}
||\tilde{\mathcal{X}}_{\Omega_{[k]}} + (X_k^{(\kappa)})_{\Omega^c_{[k]}} - X_k|| &= \\
||\tilde{\mathcal{X}}_{\Omega_{[k]}} + (X_k^{(\kappa)})_{\Omega^c_{[k]}} - ((X_k)_\Omega + (X_k)_{\Omega^c})||^2 &= \\
||(\tilde{\mathcal{X}}_{\Omega_{[k]}} - (X_k)_\Omega) + ((X_k^{(\kappa)})_{\Omega^c_{[k]}} - (X_k)_{\Omega^c})||^2 &= \\
||\tilde{\mathcal{X}}_{\Omega_{[k]}} - (X_k)_\Omega||^2 + ||(X_k^{(\kappa)})_{\Omega^c_{[k]}} - (X_k)_{\Omega^c}||^2 &\geq \\
||\tilde{\mathcal{X}}_{\Omega_{[k]}} - (X_k)_\Omega||^2. & \qquad (94)
\end{aligned}$$

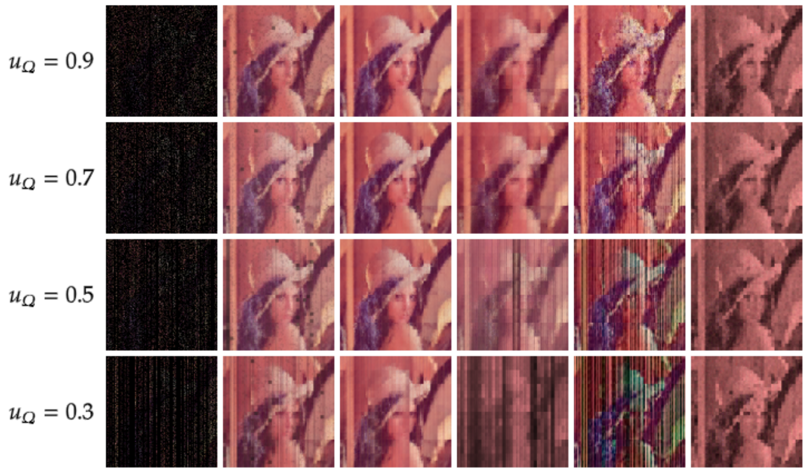

Figure 3: The images recovered from the Lena image having missing pixels at $m_r = 90\%$. The first column top down to the bottom represent the incomplete images associated with $u_\Omega \in [0.3, 0.9]$. The next columns are the images recovered by TA+ℓ_q, LR+ℓ_q, SFA, TMacTT [9], and SVD-free

Image	m_r	u_Ω	TA+ℓ_q	LR+ℓ_q	SFA	TMacTT [9]	SVD-free
Lena	0.8	0.9	0.115	0.112	0.119	0.115	0.185
		0.7	0.126	0.103	0.154	0.326	0.188
		0.5	0.142	0.112	0.234	0.480	0.201
		0.3	0.217	0.151	0.314	0.596	0.212
Peppers	0.8	0.9	0.153	0.131	0.134	0.154	0.235
		0.7	0.159	0.132	0.156	0.299	0.234
		0.5	0.176	0.147	0.225	0.407	0.249
		0.3	0.239	0.177	0.414	0.586	0.331
Lena	0.9	0.9	0.155	0.130	0.178	0.229	0.330
		0.7	0.164	0.129	0.182	0.300	0.326
		0.5	0.173	0.135	0.279	0.475	0.381
		0.3	0.207	0.159	0.414	0.654	0.498
Peppers	0.9	0.9	0.263	0.173	0.202	0.332	0.460
		0.7	0.269	0.177	0.211	0.355	0.470
		0.5	0.285	0.182	0.261	0.437	0.474
		0.3	0.442	0.206	0.465	0.610	0.599

Table XIV: The RSE performance in recovering the Lena and Peppers images.

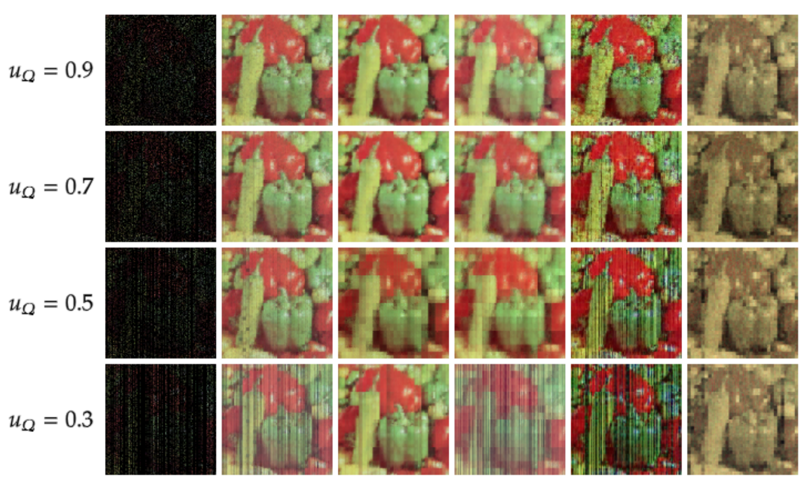

Figure 4: The images recovered from the Pepper image having missing pixels at $m_r = 90\%$. The first column top down to the bottom represent the incomplete images with $u_\Omega \in [0.3, 0.9]$. The next rows are the images recovered by TA+ℓ_q, LR+ℓ_q, SFA, TMacTT [9], and SVD-free

Therefore, we have

$$\begin{aligned} f_{q,k}(X_k) &= \frac{\beta_k}{2}||\tilde{\mathcal{X}}_{\Omega_{[k]}} - (X_k)_{\Omega_{[k]}}||^2 + \alpha_k \ell_q(X_k) \\ &\leq \frac{\beta_k}{2}||\tilde{\mathcal{X}}_{\Omega_{[k]}} + (X_k^{(\kappa)})_{\Omega^c_{[k]}} - X_k||^2 + \alpha_k \ell_q(X_k) \\ &= f^{(\kappa)}_{q,k}(X_k), \end{aligned} \quad (95)$$

which together with

$$f_{q,k}(X_k^{(\kappa)}) = f^{(\kappa)}_{q,k}(X_k^{(\kappa)}) \quad (96)$$

show that $f^{(\kappa)}_{q,k}$ is a tight majorant of $f_{q,k}$ at $X_k^{(\kappa)}$ [32]. Since $X_k^{(\kappa)}$ and $X_k^{(\kappa+1)}$ constitute a feasible point and the optimal solution of (28), it is true that $f^{(\kappa)}_{q,k}(X_k^{(\kappa)}) < f^{(\kappa)}_{q,k}(X_k^{(\kappa+1)}) \leq f_{q,k}(X_k^{(\kappa+1)})$. Hence we have

$$f_{q,k}(X_k^{(\kappa)}) = f^{(\kappa)}_{q,k}(X_k^{(\kappa)}) < f_q(X_k^{(\kappa+1)}), \quad (97)$$

i.e. (32).

Appendix II: ℓ_q as a spectral function

Let $\bar{X}$ be a positive semi-definite matrix with the eigenvalues $\lambda_i(\bar{X}) \geq 0$, $i = 1, \dots, N$, which admits the SVD $\bar{X} = U_{\bar{X}} \mathsf{diag}[\lambda_i(\bar{X})]_{i=1,\dots,N} U^H_{\bar{X}}$ with U_X unitary. For arbitrary $q > 0$ we define $\bar{X}^q$ as

$$0 \preceq \bar{X}^q \triangleq U_{\bar{X}} \mathsf{diag}[\lambda_i^q(\bar{X})]_{i=1,\dots,} U^H_{\bar{X}}, \quad (98)$$

under the agreement

$$\lambda_i^q(\bar{X}) \equiv 0 \quad \text{for} \quad \lambda_i(\bar{X}) = 0. \quad (99)$$

Let $\sigma_i(X)$, $i = 1, \dots, N$ be the singular values of X, which are actually $\sqrt{\lambda_i([X]^2)}$, where $\lambda_i([X]^2)$ are the eigenvalues of $[X]^2 \triangleq XX^T$. Then $\ell_q(X)$ defined by (6) is represented by $\ell_q(X) = \sum_{i=1}^N \lambda_i^{q/2}([X]^2)$. Thus $\ell_2(X)$ is the square Frobenius norm $||X||^2 = \text{trace}([X]^2)$. The function $\ell_1(X)$ is still a convex function [6], but $\ell_q(X)$ for $0 < q < 1$ is not. However, based on the fact that $\ell_q(X)$ is a spectral function [38] with $\ell_q(X) = \sum_i \lambda_i^q(X)$ for all positive semi-definite matrix X, where $\lambda_i(X) \geq 0$ are eigenvalues of X, we can obtain the following properties

- The function $\sum_i \lambda_i^q(X)$ of the variables $\lambda_i(X) \geq 0$ is concave so $\ell_q(X)$ is a concave function for $0 < q \leq 1$ in the domain of positive definite matrices X [38], while $\ell_1(X)$ is both convex and concave so it is a linear function, which is plausible because $\ell_1(X) = \text{trace}(X)$.
- In the domain of positive definite matrices, we have

$$\begin{aligned} \ell_q(X) \leq\ & \ell_q(\bar{X}) \\ & + q\langle U_{\bar{X}} \mathsf{diag}[\lambda_i^{q-1}(\bar{X})]_{i=1,\dots,N} U^H_{\bar{X}}, X - \bar{X}\rangle \quad (100) \\ =\ & (1-q)\ell_q(\bar{X}) \\ & + q\langle U_{\bar{X}} \mathsf{diag}[\lambda_i^{q-1}(\bar{X})]_{i=1,\dots,N} U^H_{\bar{X}}, X\rangle \quad (101) \\ =\ & (1-q)\ell_q(\bar{X}) \quad (102) \\ & q\langle \bar{X}^{q-1}, X\rangle \quad \forall\ X \succeq 0, \bar{X} \succeq 0 \end{aligned}$$

where $\bar{X}$ admits the SVD $U_{\bar{X}} \mathsf{diag}[\lambda_i]_{i=1,\dots,N} U^H_{\bar{X}}$.

- In the domain of arbitrary matrices, applying the inequality (102) for $q \to q/2$ and $X \to [X]^2$ while $\bar{X} \to [\bar{X}]^2$ yields

$$\ell_q(X) \leq (1-q/2)\ell_q(\bar{X}) + \frac{q}{2}\langle([\bar{X}]^2)^{q/2-1}, [X]^2\rangle \quad \forall\ X, \bar{X}, (103)$$

or

$$\ell_q(X) \leq (1-q/2)\ell_q(\bar{X}) + \frac{q}{2}\langle([\bar{X}^T]^2)^{q/2-1}, [X^T]^2\rangle \quad \forall\ X, \bar{X} (104)$$

Particularly,

$$\ell_1(X) \leq \frac{1}{2}\ell_1(\bar{X}) + \frac{1}{2}\langle([\bar{X}]^2)^{-1/2}, [X]^2\rangle \quad \forall\ X, \bar{X}, \quad (105)$$

and

$$\ell_1(X) \leq \frac{1}{2}\ell_1(\bar{X}) + \frac{1}{2}\langle([\bar{X}^T]^2)^{-1/2}, [X^T]^2\rangle \quad \forall\ X, \bar{X}. \quad (106)$$

Thus, similarly to [39]–[41], we can obtain useful bounds for the function $\ell_q(X)$, which is nonconvex and nonconcave.

References

[1] A. Y. Yakovlev, L. Klebanov, and D. Gaile (Eds.), *Statistical Methods for Microarray Data Analysis: Methods and Protocols*. Humana Press, 2013.

[2] J. Bennett and S. Lanning, “The Netflix prize,” in *Proc. KDD Cup & Workshop, San Jose, CA*, Sept. 2007, pp. 1–4.

[3] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan 2010.

[4] S. Ma, D. Goldfarb, and L. Chen, “Fixed point and Bregman iterative methods for matrix rank minimization,” *Math. Programm.*, vol. 128, no. 1-2, pp. 321–353, Sep 2009.

[5] F. R. Bach, “Consistency of trace norm minimization,” *J. Mach. Learn. Res.*, vol. 9, pp. 1019–1048, Jun. 2008.

[6] E. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[7] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral regularization algorithms for learning large incomplete matrices,” *J. Machine Learning Res.*, vol. 11, p. 2287âĂŞ2322, 2010.

[8] G. Marjanovic and V. Solo, “On l_p optimization and matrix completion,” *IEEE Trans. Signal Process.*, vol. 60, p. 5714âĂŞ5724, 2012.

[9] 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.

[10] C. Eckart and G. Young, “The approximation of one matrix by another of lower rank,” *Psychometrika*, vol. 1, pp. 211–218, 1936.

[11] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press (CUP), 2009.

[12] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 35, no. 1, pp. 208–220, Jan 2013.

[13] Y. Liu, L. Chen, and C. Zhu, “Improved robust tensor principal commponent analysis via low-rank core matrix,” *IEEE J. Select. Topics Signal Process.*, vol. 12, pp. 1378–1389, Dec. 2018.

[14] B. Jiang, S. Ma, and S. Zhang, “Low-M-rank tensor completion and robust tensor PCA,” *IEEE J. Select. Topics Signal Process.*, vol. 12, pp. 1390–1402, Dec. 2018.

[15] W. Dong, T. Huang, G. Shi, Y. Ma, and X. Li, “Robust tensor approximation with Laplacian scale mixture modeling for multiframe image and video denoising,” *IEEE J. Select. Topics Signal Process.*, vol. 12, pp. 1435–1448, Dec. 2018.

[16] H. Kong, X. Xie, and Z. Lin, "t-Schatten-p norm for low-rank tensor recovery," *IEEE J. Select. Topics Signal Process.*, vol. 12, pp. 1405–1419, Dec. 2018.

[17] Y. Yang, L. Han, Y. Liu, J. Zhu, and H. Yan, "A novel regularized model for third-order tensor completion," *IEEE Trans. Signal Process.*, vol. 69, pp. 3473–3483, 2021.

[18] C. Cai, G. Li, H. V. Poor, and Y. Chen, "Nonconvex low-rank tensor completion from noisy data," *Operation Research*, vol. 70, pp. 1219–1237, Mar.-Apr. 2022.

[19] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Matrix product state for higher-order tensor compression and classification," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4019–4030, Aug. 2017.

[20] G. Vidal, "Efficient classical simulation of slightly entangled quantum computations," *Phys. Rev. Lett.*, vol. 91, p. 147902, Oct. 2003.

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

[22] J. Yang, Y. Zhu, K. Li, J. Yang, and C. Hou, "Tensor completion from structurally-missing entries by low-tt-rankness and fiber-wise sparsity," *IEEE J. Select. Topics Signal Process.*, vol. 12, pp. 1420–1434, Dec. 2018.

[23] G. Pastor, "A low-rank tensor model for imputation of missing vehicular traffic volume," *IEEE Trans. Vehic. Tech.*, vol. 67, no. 8, pp. 8934–8938, Sept. 2018.

[24] J. I. Latorre, "Image compression and entanglement," *arxiv*, vol. abs/quant-ph/0510031, 2005.

[25] P. M. Hoang, H. D. Tuan, T. T. Son, and H. V. Poor, "Qualitative HD image and video recovery via high-order tensor augmentation and completion," *IEEE J. Select. Topics Signal Process.*, vol. 15, pp. 688–701, Mar. 2021.

[26] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex optimization," *IEEE Signal Process. Lett.*, vol. 14, pp. 707–710, Oct. 2007.

[27] R. Chartrand and V. Staneva, "Restricted isometry properties and non-convex compressive sensing," *Inverse Prob.*, vol. 24, pp. 1–14, 2008.

[28] X. Chen, F. Xu, and Y. Ye, "Lower bound theory of nozero entries in solutions of l_2-l_p minimization," *SIAM J. Sci. Comput.*, vol. 32, p. 2832âĂŞ2851, 2010.

[29] R.Chartrand, "Nonconvex splitting for regularized low-rank + sparse decomposition," *IEEE Trans. Signal Process.*, vol. 60, p. 5810âĂŞ5819, Nov. 2012.

[30] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. App.*, vol. 21, no. 4, 2000.

[31] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, 1978.

[32] H. Tuy, *Convex Analysis and Global Optimization (second edition)*. Springer International, 2017.

[33] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, and B. Ran, "Tensor completion via a multi-linear low-n-rank factorization model," *Neurocomputing*, vol. 133, pp. 161–169, Jun 2014.

[34] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *Inverse Problems & Imaging*, vol. 9, no. 2, pp. 601–624, Mar 2015.

[35] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, "Matrix completion and low-rank SVD via fast alternating least squares," *J. Machine Learning Res.*, vol. 16, pp. 3367–3402, 2015.

[36] C. Mu, B. Huang, J. Wright, and D. Goldfarb, "Square deal: Lower bounds and improved relaxations for tensor recovery," in *Proc. 31th Int'l Conf. Machine Learning, ICML 2014*, vol. 32, 2014, pp. 73–81.

[37] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, Dec. 2015. [Online]. Available: https://doi.org/10.1145/2827872

[38] A. S. Lewis, "Convex analysis on the Hermitian matrices," *SIAM J. Optimiz.*, vol. 6, pp. 164–177, Feb. 1996.

[39] H. H. M. Tam, H. D. Tuan, and D. T. Ngo, "Successive convex quadratic programming for quality-of-service management in full-duplex MU-MIMO multicell networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2340–2353, Jun. 2016.

[40] H. Yu, H. D. Tuan, T. Q. Duong, and L. Hanzo, "Improper Gaussian signaling for integrated data and energy networking," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3922–3934, Jun. 2020.

[41] H. Yu, H. D. Tuan, A. A. Nasir, T. Q. Duong, and H. V. Poor, "Joint design of reconfigurable intelligent surfaces and transmit beamforming under proper and improper Gaussian signaling," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2589–2603, Nov. 2020.

Pham Minh Hoang received the B.S. and M. S. degrees in Computer Science from the University of Science, VNU-HCM, Vietnam, in 2011 and 2016, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW, Australia, and with University of Science, VNU-HCM, Vietnam. His research interests include optimization methods for image processing, computer vision, machine learning, and deep learning.

Hoang Duong Tuan received the Diploma (Hons.) and Ph.D. degrees in applied mathematics from Odessa State University, Ukraine, in 1987 and 1991, respectively. He spent nine academic years in Japan as an Assistant Professor in the Department of Electronic-Mechanical Engineering, Nagoya University, from 1994 to 1999, and then as an Associate Professor in the Department of Electrical and Computer Engineering, Toyota Technological Institute, Nagoya, from 1999 to 2003. He was a Professor with the School of Electrical Engineering and Telecommunications, University of New South Wales, from 2003 to 2011. He is currently a Professor with the School of Electrical and Data Engineering, University of Technology Sydney. He has been involved in research with the areas of optimization, control, signal processing, wireless communication, and biomedical engineering for more than 25 years.

Tran Thai Son received the Bachelor's degree in Science from the Faculty of Information Technology, the University of Science, VNU-HCM, Vietnam in 1997, and the Ph.D. degree in Engineering from the Department of Electrical and Computer Engineering, Toyota Technological Institute, Japan, in 2005. He was a researcher of smart vehicle projects at Toyota Technological Institute, Japan, from 2005 to 2010. He is currently a lecturer of the Faculty of Information Technology, the University of Science, VNU-HCM, Vietnam. His research interests lie in the areas of machine learning, image processing, and computer vision.

H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. During 2006 to 2016, he served as the dean of PrincetonŠs School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the recent book *Machine Learning and Wireless Communications* (Cambridge University Press, 2022). Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.

Lajos Hanzo (Life Fellow, IEEE) received his Master degree and Doctorate in 1976 and 1983, respectively from the Technical University (TU) of Budapest. He was also awarded the Doctor of Sciences (DSc) degree by the University of Southampton (2004) and Honorary Doctorates by the TU of Budapest (2009) and by the University of Edinburgh (2015). He is a Foreign Member of the Hungarian Academy of Sciences and a former Editor-in-Chief of the IEEE Press. He has served several terms as Governor of both IEEE ComSoc and of VTS. He has published 2000+ contributions at IEEE Xplore, 19 Wiley-IEEE Press books and has helped the fast-track career of 123 PhD students. Over 40 of them are Professors at various stages of their careers in academia and many of them are leading scientists in the wireless industry. He is also a Fellow of the Royal Academy of Engineering (FREng), of the IET and of EURASIP. He holds the Eric Sumner Field Award.