

SUPPLEMENTARY MATERIAL FOR ‘ROBUST HIGH-ORDER TENSOR RECOVERY VIA NONCONVEX LOW-RANK APPROXIMATION’

Wenjin Qin¹, Hailin Wang², Weijun Ma³, Jianjun Wang^{1,*}

¹School of Mathematics and Statistics, Southwest University, China

² School of Mathematics and Statistics, Xi’an Jiaotong University, China

³ School of Information Engineering, Ningxia University, China

In this document, we present the algebraic framework associated with order- d ($d \geq 4$) tensor Singular Value Decomposition (t-SVD). To improve readability, the basic symbols and operators utilized for each section are summarized in Table 1, some of which originate from [1–7].

1. BASIC NOTIONS AND PRELIMINARIES

In this section, we introduce main notions and preliminaries concerning order- d tensor that are necessary for the whole paper.

Table 1: The main notions and preliminaries for order- d tensor.

| Notations | Descriptions | Notations | Descriptions |
|--|--|--|--|
| $\lfloor t \rfloor$ | nearest integer less than or equal to t | $\lceil t \rceil$ | the one greater than or equal to t |
| $[d] = \{1, 2, \dots, d-1, d\}$ | set of the first d natural numbers | $\mathbf{A} \in \mathbb{C}^{n_1 \times n_2}$ | matrix |
| \mathbf{A}^* | conjugate transpose | $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ | $n \times n$ identity matrix |
| $\text{tr}(\mathbf{A})$ | matrix trace | $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}^* \times \mathbf{B})$ | matrix inner product |
| $\sigma_i(\mathbf{A})$ | i -th singular value | $\mathbf{A} \otimes \mathbf{B}$ | Kronecker product of \mathbf{A} and \mathbf{B} |
| $\ \mathbf{A}\ _q = (\sum_{i,j} \mathbf{A}_{ij} ^q)^{\frac{1}{q}}$ | matrix ℓ_q -norm | $\ \mathbf{A}\ _F = (\sum_{i,j} \mathbf{A}_{ij} ^2)^{1/2}$ | matrix Frobenius norm |
| $\ \mathbf{A}\ _\infty = \max_{i,j} \mathbf{A}_{ij} $ | matrix infinity norm | $\ \mathbf{A}\ = \max_i \sigma_i(\mathbf{A})$ | matrix spectral norm |
| $\ \mathbf{A}\ _w = \sum_i w_i \sigma_i(\mathbf{A})$ | matrix weighted nuclear norm | $\ \mathbf{A}\ _{w, \mathcal{S}_p} = (\sum_i w_i \sigma_i(\mathbf{A}) ^p)^{1/p}$ | matrix weighted Schatten- p norm |
| $\ \mathbf{A}\ _* = \sum_i \sigma_i(\mathbf{A})$ | matrix nuclear norm | $\mathbf{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ | order- d tensor |
| (i_1, \dots, i_d) | tensor multiple indices | $\mathcal{A}_{i_1 \dots i_d}$ or $\mathcal{A}(i_1, \dots, i_d)$ | (i_1, \dots, i_d) -th entry |
| $\mathcal{A}_{(k)} \in \mathbb{R}^{n_k \times \prod_{j \neq k} n_j}$ | mode- k unfolding of \mathcal{A} | $\ \mathcal{A}\ _\infty = \max_{i_1 \dots i_d} \mathcal{A}_{i_1 \dots i_d} $ | tensor infinity norm |
| $\ \mathcal{A}\ _q = (\sum_{i_1 \dots i_d} \mathcal{A}_{i_1 \dots i_d} ^q)^{\frac{1}{q}}$ | tensor ℓ_q -norm | $\mathcal{A}(:, :, i_3, \dots, i_d)$ or $\mathcal{A}^{(i_3, \dots, i_d)}$ | slice along mode-1 and mode-2 |
| $\ \mathcal{A}\ _F = (\sum_{i_1 \dots i_d} \mathcal{A}_{i_1 \dots i_d} ^2)^{1/2}$ | tensor Frobenius norm | $\langle \mathcal{A}, \mathcal{B} \rangle = \sum \langle \mathcal{A}^{(i_3, \dots, i_d)}, \mathcal{B}^{(i_3, \dots, i_d)} \rangle$ | tensor inner product |
| $\mathcal{A}_i = \text{fft}(\mathcal{A}, [i], i)$ for $i = 3, \dots, d$ | FFT along mode-3 to mode- d | $\mathcal{A} = \text{ifft}(\mathcal{A}_i, [i], j)$ for $j = d, \dots, 3$ | inverse Fast Fourier Transform |
| $L(\mathcal{A}) : \mathbb{C}^{n_1 \times \dots \times n_d} \rightarrow \mathbb{C}^{n_1 \times \dots \times n_d}$ | generic invertible linear transform | $\mathcal{A} *_L \mathcal{B}$ | linear transform L based t-product |
| $\mathcal{A}_i \in \mathbb{R}^{n_1 \times \dots \times n_{d-1}}$ | order- $(d-1)$ tensor constructed by keeping the d -th index of \mathcal{A} fixed at i , $\mathcal{A}_i := \mathcal{A}(:, \dots, :, i)$. | | |
| $\mathcal{A}^j \in \mathbb{R}^{n_1 \times n_2}$ | $\mathcal{A}^j = \mathcal{A}(:, :, i_3, \dots, i_d)$, $j = (i_d - 1)n_3 \dots n_{d-1} + \dots + (i_4 - 1)n_3 + i_3$, $i_d \in \{1, \dots, n_d\}$. | | |
| $\mathcal{A} \times_n \mathbf{U}$ | the mode- n product of tensor \mathcal{A} with matrix \mathbf{U} , $\mathcal{B} = \mathcal{A} \times_n \mathbf{U} \iff \mathcal{B}_{(n)} = \mathbf{U} \cdot \mathcal{A}_{(n)}$. | | |
| $\mathcal{A}_L \triangleq L(\mathcal{A})$ | $L(\mathcal{A}) = \mathcal{A} \times_3 \mathbf{U}_{n_3} \times_4 \mathbf{U}_{n_4} \dots \times_d \mathbf{U}_{n_d}$, $\mathbf{U}_{n_i} \in \mathbb{C}^{n_i \times n_i}$ denotes an invertible transform matrix. | | |
| $L^{-1}(\mathcal{A})$ | $L^{-1}(\mathcal{A}) = \mathcal{A} \times_d \mathbf{U}_{n_d}^{-1} \times_{d-1} \mathbf{U}_{n_{d-1}}^{-1} \dots \times_3 \mathbf{U}_{n_3}^{-1}$, $L^{-1}(L(\mathcal{A})) = \mathcal{A}$. | | |
| $\text{circ}(\mathcal{A})$ | $\text{circ}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_{n_d} & \mathcal{A}_{n_d-1} & \dots & \mathcal{A}_2 \\ \mathcal{A}_2 & \mathcal{A}_1 & \mathcal{A}_{n_d} & \dots & \mathcal{A}_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_{n_d} & \mathcal{A}_{n_d-1} & \mathcal{A}_{n_d-2} & \dots & \mathcal{A}_1 \end{bmatrix} \in \mathbb{R}^{n_1 n_d \times \dots \times n_{d-2} n_d \times n_{d-1}}$. | | |
| $\text{bcirc}(\mathcal{A})$ | a $(n_1 n_3 \dots n_d \times n_2 n_3 \dots n_d)$ block circulant matrix at the base level of the operator $\text{circ}(\mathcal{A})$. | | |
| $\text{unfold}(\mathcal{A})$ | $\text{unfold}(\mathcal{A}) = [\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{n_d-1}, \mathcal{A}_{n_d}]^T \in \mathbb{R}^{n_1 n_d \times n_2 \times \dots \times n_{d-1}}$. | | |
| $\text{fold}(\mathcal{A})$ | the operation takes $\text{unfold}(\mathcal{A})$ back to order- d tensor form, i.e., $\text{fold}(\text{unfold}(\mathcal{A}), n_d) = \mathcal{A}$. | | |
| $\text{bunf}(\mathcal{A})$ | a $(n_1 n_3 \dots n_d \times n_2)$ matrix formed by applying $\text{unfold}(\cdot)$ repeatedly until a block matrix result. | | |
| $\text{bfold}(\mathcal{A})$ | the operation takes $\text{bunf}(\mathcal{A})$ back to order- d tensor form, i.e., $\text{bfold}(\text{bunf}(\mathcal{A})) = \mathcal{A}$. | | |
| $\text{bdiag}(\mathcal{A})$ | $\text{bdiag}(\mathcal{A}) = \text{diag}(\mathcal{A}^1, \mathcal{A}^2, \dots, \mathcal{A}^{J-1}, \mathcal{A}^J) \in \mathbb{R}^{n_1 n_3 \dots n_d \times n_2 n_3 n_d}$, $J = n_3 \dots n_d$. | | |
| $\mathcal{A} \triangle \mathcal{B}$ | face-wise product of two order- d tensor, $\mathcal{C} = \mathcal{A} \triangle \mathcal{B} \iff \text{bdiag}(\mathcal{C}) = \text{bdiag}(\mathcal{A}) \cdot \text{bdiag}(\mathcal{B})$. | | |

*✉ wjj@swu.edu.cn.

Algorithm 1: Generic order- d t-product [6], tproduct-L($\mathcal{A}, \mathcal{B}, L$).

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}, \mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3 \times \dots \times n_d}$,
and corresponding matrices $\{U_{n_i}\}_{i=3}^d$ of linear transforms L .
Output: $\mathcal{C} = \mathcal{A} *_L \mathcal{B} \in \mathbb{R}^{n_1 \times l \times n_3 \times \dots \times n_d}$.

- 1 Compute the result of linear transform on \mathcal{A} and \mathcal{B}
- 2 $\mathcal{A}_L \leftarrow L(\mathcal{A}), \mathcal{B}_L \leftarrow L(\mathcal{B})$;
- 3 Compute each matrix slice of \mathcal{C}_L by
- 4 **for** $i_3 \in \{1, \dots, n_3\}, \dots, i_d \in \{1, \dots, n_d\}$ **do**
- 5 $\mathcal{C}_L(:, :, i_3, \dots, i_d) = \mathcal{A}_L(:, :, i_3, \dots, i_d) \cdot \mathcal{B}_L(:, :, i_3, \dots, i_d)$;
- 6 **end**
- 7 Compute the result of inverse linear transform on \mathcal{C}_L
- 8 $\mathcal{C} \leftarrow L^{-1}(\mathcal{C}_L)$.

Definition 1.1. (Order- d t-product [4, 6]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3 \times \dots \times n_d}$, then the tensor-tensor product (t-product) $\mathcal{C} = \mathcal{A} * \mathcal{B}$ is an $n_1 \times l \times n_3 \times \dots \times n_d$ tensor defined as follows:

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = \text{bfold}(\text{bcirc}(\mathcal{A}) \cdot \text{bunfoid}(\mathcal{B})). \quad (1)$$

The order- d t-product in (1) can be converted to the matrix-matrix multiplication in the transform domain. That is to say,

$$\mathcal{C} = \mathcal{A} *_L \mathcal{B} = L^{-1}(\mathcal{A}_L \triangle \mathcal{B}_L). \quad (2)$$

The computational procedure of generic order- d t-product is shown in Algorithm 1.

Definition 1.2. (Order- d conjugate transpose [6]) The conjugate transpose of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$ is the tensor $\mathcal{A}^* \in \mathbb{C}^{n_2 \times n_1 \times n_3 \times \dots \times n_d}$ if $\mathcal{A}^*_L(:, :, i_3, \dots, i_d) = (\mathcal{A}_L(:, :, i_3, \dots, i_d))^*$, for all $i_j \in [n_j], j \in \{3, \dots, d\}$.

Definition 1.3. (order- d identity tensor [6]) The order- d identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3 \times \dots \times n_d}$ is the tensor such that $\mathcal{I}_L(:, :, i_3, \dots, i_d) = \mathbf{I}_n$ for all $i_j \in [n_j], j \in \{3, \dots, d\}$.

Definition 1.4. (Order- d orthogonal tensor [6]) An order- d tensor $\mathcal{Q} \in \mathbb{C}^{n \times n \times n_3 \times \dots \times n_d}$ is orthogonal if it satisfies $\mathcal{Q}^* *_L \mathcal{Q} = \mathcal{Q} *_L \mathcal{Q}^* = \mathcal{I}$.

Definition 1.5. (Order- d f-diagonal tensor [6]) An order- d tensor $\mathcal{A} \in \mathbb{R}^{n \times n \times n_3 \times \dots \times n_d}$ is called f-diagonal if $\mathcal{A}_L(:, :, i_3, \dots, i_d)$ is a diagonal matrix for all $i_j \in [n_j], j \in \{3, \dots, d\}$.

Definition 1.6. (Order- d f-upper triangular tensor) An order- d tensor $\mathcal{A} \in \mathbb{R}^{n \times n \times n_3 \times \dots \times n_d}$ is called f-upper triangular if $\mathcal{A}_L(:, :, i_3, \dots, i_d)$ is an upper triangular matrix for all $i_j \in [n_j], j \in \{3, \dots, d\}$.

Definition 1.7. (Order- d Gaussian random tensor) $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$ is called a Gaussian random tensor, if $\mathcal{A}_L(:, :, i_3, \dots, i_d)$ satisfy the standard normal distribution for all $i_j \in [n_j], j \in \{3, \dots, d\}$.

Definition 1.8. (Order- d t-QR decomposition [4]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$, then it can be decomposed as

$$\mathcal{A} = \mathcal{Q} *_L \mathcal{R}, \quad (3)$$

where $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \dots \times n_d}$ is an orthogonal tensor, $\mathcal{R} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$ is f-upper triangular.

The computational procedure of generic order- d t-QR decomposition is presented in Algorithm 2.

Definition 1.9. (Order- d t-SVD [6]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$, then it can be factorized as

$$\mathcal{A} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^*, \quad (4)$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \dots \times n_d}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3 \times \dots \times n_d}$ are orthogonal, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$ is a f-diagonal tensor which has the property that $\mathcal{S}_{i_1 \dots i_d} = 0$ unless $n_1 = n_2$.

The computational procedure of generic order- d t-SVD is presented in Algorithm 3.

Algorithm 2: Generic order- d t-QR decomposition, $\text{tqr-L}(\mathcal{A}, L)$.

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and corresponding matrices $\{U_{n_i}\}_{i=3}^d$ of linear transforms L .
Output: t-QR components \mathcal{Q} and \mathcal{R} of \mathcal{A} .
1 Compute the result of linear transform on \mathcal{A}
2 $\mathcal{A}_L \leftarrow L(\mathcal{A})$;
3 Compute the matrix slice of \mathcal{Q}_L and \mathcal{R}_L from \mathcal{A}_L by
4 **for** $i_3 \in \{1, \dots, n_3\}, \dots, i_d \in \{1, \dots, n_d\}$ **do**
5 $[\mathcal{Q}, \mathcal{R}] = \text{qr}(\mathcal{A}_L(:, :, i_3, \dots, i_d), 0)$;
6 $\mathcal{Q}_L(:, :, i_3, \dots, i_d) = \mathcal{Q}, \mathcal{R}_L(:, :, i_3, \dots, i_d) = \mathcal{R}$;
7 **end**
8 Compute the result of inverse linear transform on \mathcal{Q}_L and \mathcal{R}_L
9 $\mathcal{Q} \leftarrow L^{-1}(\mathcal{Q}_L)$ and $\mathcal{R} \leftarrow L^{-1}(\mathcal{R}_L)$.

Algorithm 3: Generic order- d t-SVD [6], $\text{tsvd-L}(\mathcal{A}, L)$.

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and corresponding matrices $\{U_{n_i}\}_{i=3}^d$ of linear transforms L .
Output: t-SVD components \mathcal{U}, \mathcal{S} and \mathcal{V} of \mathcal{A} .
1 Compute the result of linear transform on \mathcal{A}
2 $\mathcal{A}_L \leftarrow L(\mathcal{A})$;
3 Compute the matrix slice of $\mathcal{U}_L, \mathcal{S}_L$ and \mathcal{V}_L from \mathcal{A}_L by
4 **for** $i_3 \in \{1, \dots, n_3\}, \dots, i_d \in \{1, \dots, n_d\}$ **do**
5 $[\mathcal{U}, \mathcal{S}, \mathcal{V}] = \text{svd}(\mathcal{A}_L(:, :, i_3, \dots, i_d))$;
6 $\mathcal{U}_L(:, :, i_3, \dots, i_d) = \mathcal{U}, \mathcal{S}_L(:, :, i_3, \dots, i_d) = \mathcal{S}, \mathcal{V}_L(:, :, i_3, \dots, i_d) = \mathcal{V}$;
7 **end**
8 Compute the result of inverse linear transform on $\mathcal{U}_L, \mathcal{S}_L$ and \mathcal{V}_L
9 $\mathcal{U} \leftarrow L^{-1}(\mathcal{U}_L), \mathcal{S} \leftarrow L^{-1}(\mathcal{S}_L)$ and $\mathcal{V} \leftarrow L^{-1}(\mathcal{V}_L)$.

Definition 1.10. (Order- d t-SVD rank [6]) For any $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, let \mathcal{S} be from the t-SVD component of $\mathcal{A} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^*$. Then the t-SVD rank of \mathcal{A} is defined as

$$\begin{aligned} \text{rank}_{\text{tsvd}}(\mathcal{A}) &= \#\{i : \mathcal{S}(i, i, :, \dots, :) \neq \mathbf{0}\}, \\ &= \max_{i_3 \in [n_3], \dots, i_d \in [n_d]} \text{rank}(\mathcal{A}_L(:, :, i_3, \dots, i_d)), \end{aligned}$$

where $\#$ denotes the cardinality of a set.

Remark 1.1. Let $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ with $\text{rank}_{\text{tsvd}}(\mathcal{X}) = r$, the skinny t-SVD of tensor \mathcal{X} is $\mathcal{X} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^*$, where $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3 \times \dots \times n_d}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3 \times \dots \times n_d}$ satisfy $\mathcal{U}^* *_L \mathcal{U} = \mathcal{I}$ and $\mathcal{V}^* *_L \mathcal{V} = \mathcal{I}$, and $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3 \times \dots \times n_d}$ is a f -diagonal tensor which has the property that $\mathcal{S}_{i_1 \dots i_d} = 0$ unless $n_1 = n_2$.

Definition 1.11. (Order- d tensor spectral norm [6]) The spectral norm of $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ is defined as $\|\mathcal{A}\| := \|\text{bdiag}(\mathcal{A}_L)\|$.

Definition 1.12. (Order- d TNN [6]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and $m = \min(n_1, n_2)$, then the tensor nuclear norm (TNN) of \mathcal{A} is defined as

$$\|\mathcal{A}\|_{*,L} := \frac{1}{\rho} \|\text{bdiag}(\mathcal{A}_L)\|_* \quad (5)$$

$$= \frac{1}{\rho} \sum_{i=1}^m \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \mathcal{S}_L(i, i, i_3, \dots, i_d), \quad (6)$$

where $\rho > 0$ is a positive constant determined by the invertible linear transforms L , and the entries on the diagonal of $\mathcal{S}_L(:, :, i_3, \dots, i_d)$ denote the singular values of $\mathcal{A}_L(:, :, i_3, \dots, i_d)$.

Definition 1.13. (Order- d WTNN [6]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and $m = \min(n_1, n_2)$, then the weighted tensor nuclear norm (WTNN) of \mathcal{A} is defined as

$$\|\mathcal{A}\|_{\mathcal{W},L} := \frac{1}{\rho} \|\text{bdiag}(\mathcal{A}_L)\|_w \quad (7)$$

$$= \frac{1}{\rho} \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \|\mathcal{A}_L(:, :, i_3, \dots, i_d)\|_w \quad (8)$$

$$= \frac{1}{\rho} \sum_{i=1}^m \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \mathcal{W}(i, i, i_3, \dots, i_d) \mathcal{S}_L(i, i, i_3, \dots, i_d), \quad (9)$$

where ρ is a positive constant, \mathcal{W} represents the weight parameters composed of f -diagonal tensor, $\mathcal{W}(i, i, i_3, \dots, i_p) > 0$, and the entries on the diagonal of $\mathcal{S}_L(:, :, i_3, \dots, i_d)$ denote the singular values of $\mathcal{A}_L(:, :, i_3, \dots, i_d)$.

Definition 1.14. (Order- d WTSN) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and $m = \min(n_1, n_2)$, then the weighted tensor Schatten- p norm (WTSN) of \mathcal{A} is defined as

$$\|\mathcal{A}\|_{\mathcal{W},\mathcal{S}_p} := \left(\frac{1}{\rho} \|\text{bdiag}(\mathcal{A}_L)\|_{w,\mathcal{S}_p}^p \right)^{1/p} \quad (10)$$

$$= \left(\frac{1}{\rho} \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \|\mathcal{A}_L(:, :, i_3, \dots, i_d)\|_{w,\mathcal{S}_p}^p \right)^{\frac{1}{p}} \quad (11)$$

$$= \left(\frac{1}{\rho} \sum_{i=1}^m \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \mathcal{W}(i, i, i_3, \dots, i_p) |\mathcal{S}_L(i, i, i_3, \dots, i_p)|^p \right)^{\frac{1}{p}}, \quad (12)$$

where ρ is a positive constant, \mathcal{W} represents the weight parameters composed of f -diagonal tensor, $\mathcal{W}(i, i, i_3, \dots, i_p) > 0$, and the entries on the diagonal of $\mathcal{S}_L(:, :, i_3, \dots, i_d)$ denote the singular values of $\mathcal{A}_L(:, :, i_3, \dots, i_d)$.

The WTSN defined in (10) can be equivalently reformulated as follows:

$$\|\mathcal{A}\|_{\mathcal{W},\mathcal{S}_p}^p = \frac{1}{\rho} \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \text{tr} \left(\mathcal{W}^{(i_3, \dots, i_p)} |\mathcal{S}_L^{(i_3, \dots, i_p)}|^p \right). \quad (13)$$

Remark 1.2. Throughout the article, the constant ρ appeared in the key definition and theorem is the constant ρ obtained when corresponding matrices of the invertible linear transform L satisfy the equation:

$$\begin{aligned} & (\mathbf{U}_{n_p} \otimes \mathbf{U}_{n_{p-1}} \otimes \dots \otimes \mathbf{U}_{n_3})(\mathbf{U}_{n_p}^* \otimes \mathbf{U}_{n_{p-1}}^* \otimes \dots \otimes \mathbf{U}_{n_3}^*) \\ &= (\mathbf{U}_{n_p}^* \otimes \mathbf{U}_{n_{p-1}}^* \otimes \dots \otimes \mathbf{U}_{n_3}^*)(\mathbf{U}_{n_p} \otimes \mathbf{U}_{n_{p-1}} \otimes \dots \otimes \mathbf{U}_{n_3}) \\ &= \rho \mathbf{I}_{n_3 n_4 \dots n_p}. \end{aligned} \quad (14)$$

The constant ρ is related to the invertible linear transform L . For instance, if the invertible transform matrices $\{\mathbf{U}_{n_i}\}_{i=3}^d$ satisfy: $\mathbf{U}_{n_3} \times \mathbf{U}_{n_3}^* = \mathbf{U}_{n_3}^* \times \mathbf{U}_{n_3} = n_3 \mathbf{I}_{n_3}, \dots, \mathbf{U}_{n_d} \times \mathbf{U}_{n_d}^* = \mathbf{U}_{n_d}^* \times \mathbf{U}_{n_d} = n_d \mathbf{I}_{n_d}$, then ρ equals to $n_3 \times \dots \times n_d$.

2. ORDER- D RT-SVD SCHEME

In this section, we propose the generic randomized tensor Singular Value Decomposition (rt-SVD) method, which can be viewed as a flexible extension of the matrix randomized SVD (r-SVD) [8]. The generic order- d rt-SVD approach mainly incorporates the randomized technique into the generic order- d t-SVD as a result of better efficiency.

The simplified version of generic order- d rt-SVD is presented in Algorithm 4. Following a similar acceleration technique to matrix r-SVD, the core of generic order- d rt-SVD method is to find a good approximate factorization of tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, $\mathcal{U}^* \mathcal{S} \mathcal{V}^*$. This approximation can be implemented by multiplying \mathcal{A} with a random tensor \mathcal{G} on its right side, and then obtaining an orthogonal subspace basis tensor \mathcal{Q} such that $\mathcal{A} \approx \mathcal{Q}^* \mathcal{Q}^* \mathcal{A}$. Specifically, one can capture the main actions for the column space of $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ via $\mathcal{Y} = \mathcal{A}^* \mathcal{G}$ with \mathcal{G} an $n_2 \times (k+q) \times n_3 \times \dots \times n_d$ Gaussian random tensor.

Algorithm 4: Generic order- d rt-SVD (simplified version).

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, truncation term $k < \min(n_1, n_2)$, oversampling parameter: $q > 0, l = k + q$ and corresponding matrices $\{U_{n_i}\}_{i=3}^d$ of linear transforms L .

Output: $\mathcal{U} \in \mathbb{R}^{n_1 \times k \times n_3 \times \dots \times n_d}, \mathcal{S} \in \mathbb{R}^{k \times k \times n_3 \times \dots \times n_d}, \mathcal{V} \in \mathbb{R}^{n_2 \times k \times n_3 \times \dots \times n_d}$.

- 1 Generate a Gaussian random tensor $\mathcal{G} \in \mathbb{R}^{n_2 \times l \times n_3 \times \dots \times n_d}$;
 - 2 Construct a random projection of tensor \mathcal{A} as $\mathcal{Y} = \mathcal{A} *_L \mathcal{G}$;
 - 3 Form the tensor $\mathcal{Q} \in \mathbb{R}^{n_1 \times l \times n_3 \times \dots \times n_d}$ by using t-QR decomposition of \mathcal{Y} ;
 - 4 Construct a tensor $\mathcal{B} = \mathcal{Q}^* *_L \mathcal{A}$, whose size is $l \times n_2 \times n_3 \times \dots \times n_d$;
 - 5 Compute t-SVD of \mathcal{B} , truncate it with target truncation term k , and obtain $\mathcal{U}_k, \mathcal{S}_k$, and \mathcal{V}_k ;
 - 6 Form the rt-SVD components of \mathcal{A} , $\mathcal{U} = (\mathcal{Q} *_L \mathcal{U}_k), \mathcal{S} = \mathcal{S}_k, \mathcal{V} = \mathcal{V}_k$.
-

Algorithm 5: PowerMethod ($\mathcal{A}, \mathcal{W}, \eta$) [8, 9].

Input: $\mathcal{A} \in \mathbb{R}^{m \times n}, \mathcal{W} \in \mathbb{R}^{n \times k}$, and the number of power iterations η .

- 1 **Initialize:** $\mathcal{Y}_1 = \mathcal{A}\mathcal{W}$;
 - 2 **for** $j = 1, 2, \dots, \eta$ **do**
 - 3 $\mathcal{Q}_j = \text{qr}(\mathcal{Y}_j, 0)$;
 - 4 $\mathcal{Y}_{j+1} = \mathcal{A}(\mathcal{A}^\top \mathcal{Q}_j)$;
 - 5 **end**
 - 6 **return:** PowerMethod($\mathcal{A}, \mathcal{W}, \eta$) = \mathcal{Q}_j .
-

Algorithm 6: Randomized SVD method using power iteration, r-svd($\mathcal{A}, \mathcal{G}, k, q, p$).

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2}$, truncation term $k < \min(n_1, n_2)$, oversampling parameter: $q > 0, p \geq 0, l = k + q$ and Gaussian random matrix $\mathcal{G} \in \mathbb{R}^{n_2 \times l}$.

Output: r-SVD components $\mathcal{U} \in \mathbb{R}^{n_1 \times k}, \mathcal{S} \in \mathbb{R}^{k \times k}, \mathcal{V} \in \mathbb{R}^{n_2 \times k}$ of \mathcal{A} .

- 1 $\mathcal{Q}_1 = \text{PowerMethod}(\mathcal{A}, \mathcal{G}, p)$;
 - 2 Obtain the small matrix \mathcal{R} using the QR factorization $[\mathcal{Q}_2, \mathcal{R}] = \text{qr}(\mathcal{A}^* \cdot \mathcal{Q}_1, 0)$;
 - 3 Take the SVD of matrix \mathcal{R} , i.e., $[\mathcal{U}_1, \mathcal{\Lambda}_1, \mathcal{V}_1] = \text{svd}(\mathcal{R})$;
 - 4 Approximate the SVD components of \mathcal{A} using the results of the SVD of \mathcal{R}
 - 5 $\mathcal{U}_k = \mathcal{Q}_1 \mathcal{V}_1, \mathcal{\Lambda}_k = \mathcal{\Lambda}_1, \mathcal{V}_k = \mathcal{Q}_2 \mathcal{U}_1$;
 - 6 Extract components corresponding to the k largest singular values
 - 7 $\mathcal{U} = \mathcal{U}_k(:, 1:k), \mathcal{S} = \mathcal{\Lambda}_k(1:k, 1:k), \mathcal{V} = \mathcal{V}_k(:, 1:k)$.
-

Algorithm 7: Generic order- d rt-SVD using power iteration, rtsvd-L(\mathcal{A}, L, k, q, p).

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, truncation term $k < \min(n_1, n_2)$, oversampling parameter: $q > 0, p \geq 0$ and corresponding matrices $\{U_{n_i}\}_{i=3}^d$ of linear transforms L .

Output: $\mathcal{U} \in \mathbb{R}^{n_1 \times k \times n_3 \times \dots \times n_d}, \mathcal{S} \in \mathbb{R}^{k \times k \times n_3 \times \dots \times n_d}, \mathcal{V} \in \mathbb{R}^{n_2 \times k \times n_3 \times \dots \times n_d}$.

- 1 Set $l = k + q$ and initialize a Gaussian random tensor $\mathcal{G} \in \mathbb{R}^{n_2 \times l \times n_3 \times \dots \times n_d}$;
 - 2 Compute the result of linear transform on \mathcal{A} and \mathcal{G}
 - 3 $\mathcal{A}_L \leftarrow L(\mathcal{A}), \mathcal{G}_L \leftarrow L(\mathcal{G})$;
 - 4 Compute the matrix slice of $\mathcal{U}_L, \mathcal{S}_L$ and \mathcal{V}_L from \mathcal{A}_L by
 - 5 **for** $i_3 \in \{1, \dots, n_3\}, \dots, i_d \in \{1, \dots, n_d\}$ **do**
 - 6 $[\mathcal{U}, \mathcal{S}, \mathcal{V}] = \text{r-svd}(\mathcal{A}_L(:, :, i_3, \dots, i_d), \mathcal{G}_L(:, :, i_3, \dots, i_d), k, q, p)$;
 - 7 $\mathcal{U}_L(:, :, i_3, \dots, i_d) = \mathcal{U}, \mathcal{S}_L(:, :, i_3, \dots, i_d) = \mathcal{S}, \mathcal{V}_L(:, :, i_3, \dots, i_d) = \mathcal{V}$.
 - 8 **end**
 - 9 Compute the result of inverse linear transform on $\mathcal{U}_L, \mathcal{S}_L$ and \mathcal{V}_L
 - 10 $\mathcal{U} \leftarrow L^{-1}(\mathcal{U}_L), \mathcal{S} \leftarrow L^{-1}(\mathcal{S}_L)$ and $\mathcal{V} \leftarrow L^{-1}(\mathcal{V}_L)$.
-

Algorithm 8: Generalized Soft-Thresholding (GST) [10, 11].

Input: s, w, p, J .

Output: $\text{GST}(s, w, p, J)$.

```

1  $\delta_p^{GST}(w) = [2w(1-p)]^{\frac{1}{2-p}} + wp[2w(1-p)]^{\frac{p-1}{2-p}};$ 
2 if  $|s| \leq \delta_p^{GST}(w)$  then
3    $\text{GST}(s, w, p, J) = 0;$ 
4 else
5    $j = 0, x^{(j)} = |s|;$ 
6   for  $j = 0, 1, \dots, J$  do
7      $x^{(j+1)} = |s| - wp(x^{(j)})^{p-1};$ 
8      $j = j + 1;$ 
9   end
10   $\text{GST}(s, w, p, J) = \text{sgn}(s)x^{(j)};$ 
11 end

```

As a result, \mathcal{Y} is the tensor of size $n_1 \times (k + q) \times n_3 \times \dots \times n_d$, in which k is the desired t-SVD rank and q denotes a small oversampling parameter. Eventually, \mathcal{Q} can be achieved from the tensor QR decomposition (t-QR) of \mathcal{Y} .

To further improve the accuracy of randomized approximation of \mathcal{A} , we can additionally apply the power iteration scheme [8], which multiplies alternately with \mathcal{A} and \mathcal{A}^* , i.e., $(\mathcal{A} *_L \mathcal{A}^*)^p *_L \mathcal{A}$, where p is a nonnegative integer. In other word, we replace the Step 2 of Algorithm 4 with

$$\mathcal{Y} = (\mathcal{A} *_L \mathcal{A}^*)^p *_L \mathcal{A} *_L \mathcal{G}.$$

Order- d rt-SVD using power iteration scheme works efficiently when the underlying t-SVD rank of \mathcal{A} is low, i.e., the singular values of $\mathcal{A}_L(:, :, i_3, \dots, i_d), \forall i_j \in [n_j], j \in \{3, \dots, d\}$ decay at a comparable rate. The generic order- d rt-SVD using power iteration scheme is presented in Algorithm 7, which is built on Algorithm 5 and 6.

3. ORDER- D WTSN MINIMIZATION PROBLEM

In this section, in virtue of generalized soft-thresholding (GST) algorithm [10, 11], we provide the calculation method of order- d WTSN minimization problem

$$\arg \min_{\mathcal{X}} \tau \|\mathcal{X}\|_{\mathcal{W}, \mathcal{S}_p}^p + \frac{1}{2} \|\mathcal{X} - \mathcal{Z}\|_F^2. \quad (15)$$

Definition 3.1. (Order- d GT-SVT operator) Let $\mathcal{A} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^*$ be the t-SVD of $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$. For any $\tau > 0, 0 < p < 1$, the order- d Generalized Tensor Singular Value Thresholding (GT-SVT) operator is defined as

$$\mathcal{D}_{\mathcal{W}, p, \tau}(\mathcal{A}) = \mathcal{U} *_L \mathcal{S}_{\mathcal{W}, p, \tau} *_L \mathcal{V}^*, \quad (16)$$

where

$$\mathcal{S}_{\mathcal{W}, p, \tau} = L^{-1}(\text{GST}(\mathcal{S}_L, \tau \mathcal{W}, p, J)),$$

in which J is the number of iterations of GST algorithm, and $\mathcal{W} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ denotes the weight parameter composed of an order- d f-diagonal tensor.

The computational procedure of generic order- d GT-SVT operator is presented in Algorithm 9.

Lemma 3.1. [11] Let the SVD of $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$ be $\mathbf{Y} = \mathbf{U} \Sigma \mathbf{V}^\top$ with $\Sigma = \text{diag}\{\sigma_1, \dots, \sigma_m\}$. For any $\tau \geq 0$ and $0 \leq w_1 \leq w_2 \leq \dots \leq w_m$ ($m = \min\{n_1, n_2\}$), a global optimal solution to the optimization problem

$$\min_{\mathbf{X}} \tau \|\mathbf{X}\|_{\mathbf{w}, \mathcal{S}_p}^p + \frac{1}{2} \|\mathbf{X} - \mathbf{Y}\|_F^2$$

is given by

$$\mathcal{D}_{\mathbf{w}, \tau, p}(\mathbf{Y}) = \mathbf{U} \cdot \mathbf{S}_{\mathbf{w}, \tau, p}(\mathbf{Y}) \cdot \mathbf{V}^T,$$

where $\mathbf{S}_{\mathbf{w}, \tau, p}(\mathbf{Y}) = \text{diag}\{\text{GST}(\sigma_i(\mathbf{Y}), \tau w_i, p, J), i = 1, \dots, m\}$.

Algorithm 9: Order- D Generalized Tensor Singular Value Thresholding (GT-SVT).

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, weight parameter: $\mathcal{W} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, truncation term: $k < \min(n_1, n_2)$, $\tau > 0$, $0 < p < 1$, the number of GST and PowerMethod iteration: α, β , oversampling parameter: $q > 0$, and corresponding matrices $\{\mathbf{U}_{n_i}\}_{i=3}^d$ of invertible linear transforms L .

Output: $\mathcal{D}_{\mathcal{W},p,\tau}(\mathcal{A}) = \mathbf{U}_{*L}^* \mathcal{S}_{\mathcal{W},p,\tau} \mathbf{V}_{*L}^*$.

```

1 Compute the rt-SVD or t-SVD components  $\mathcal{U}, \mathcal{S}$  and  $\mathcal{V}$  of  $\mathcal{A}$ 
2 if utilize the rt-SVD scheme then
3    $[\mathcal{U}, \mathcal{S}, \mathcal{V}] = \text{rtsvd-L}(\mathcal{A}, L, k, q, \beta)$ ;
4 end
5 else if utilize the t-SVD scheme then
6    $[\mathcal{U}, \mathcal{S}, \mathcal{V}] = \text{tsvd-L}(\mathcal{A}, L)$ ;
7 end
8 Compute the matrix slice of  $\mathcal{C}_L$  by
9 for  $i_3 \in \{1, \dots, n_3\}, \dots, i_d \in \{1, \dots, n_d\}$  do
10   $\text{diagS} = \text{GST}(\text{diag}(\mathcal{S}_L(:, :, i_3, \dots, i_d)), \tau \text{diag}(\mathcal{W}(:, :, i_3, \dots, i_d)), p, \alpha)$ ;
11   $\mathcal{C}_L(:, :, i_3, \dots, i_d) = \text{diag}(\text{diagS})$ ;
12 end
13 Compute the result of inverse linear transform on  $\mathcal{C}_L$ 
14  $\mathcal{S}_{\mathcal{W},p,\tau} \leftarrow L^{-1}(\mathcal{C}_L)$ .
15 Compute  $\mathcal{D}_{\mathcal{W},p,\tau}(\mathcal{A}) = \text{tproduct-L}(\text{tproduct-L}(\mathcal{U}, \mathcal{S}_{\mathcal{W},p,\tau}), \mathcal{V}^*)$ .

```

Theorem 3.1. Let $m = \min(n_1, n_2)$. For any $\tau > 0$, $0 < p < 1$ and $\mathcal{Z} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, then the order- d GT-SVT operator (16) obeys

$$\mathcal{D}_{\mathcal{W},p,\tau}(\mathcal{Z}) = \arg \min_{\mathcal{X}} \tau \|\mathcal{X}\|_{\mathcal{W},\mathcal{S}_p}^p + \frac{1}{2} \|\mathcal{X} - \mathcal{Z}\|_F^2, \quad (17)$$

if the weight parameter satisfies $0 \leq \mathcal{W}(1, 1, i_3, \dots, i_d) \leq \dots \leq \mathcal{W}(m, m, i_3, \dots, i_d)$, for all $i_j \in [n_j], j \in \{3, \dots, d\}$.

Proof. Based on the definition of order- d WTSN, on the one hand, we have

$$\tau \|\mathcal{X}\|_{\mathcal{W},\mathcal{S}_p}^p = \frac{1}{\rho} \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \tau \|\mathcal{X}_L(:, :, i_3, \dots, i_d)\|_{w,\mathcal{S}_p}^p, \quad (18)$$

where $w = \text{diag}(\mathcal{W}(:, :, i_3, \dots, i_d))$. Utilizing the property (14), on the other hand, we have the following important equations:

$$\begin{aligned} \|\mathcal{A}\|_F &= \frac{1}{\sqrt{\rho}} \|\text{bdiag}(\mathcal{A}_L)\|_F, \\ \langle \mathcal{A}, \mathcal{B} \rangle &= \frac{1}{\rho} \langle \text{bdiag}(\mathcal{A}_L), \text{bdiag}(\mathcal{B}_L) \rangle, \end{aligned}$$

thus

$$\begin{aligned} \frac{1}{2} \|\mathcal{X} - \mathcal{Z}\|_F^2 &= \frac{1}{2\rho} \|\text{bdiag}(\mathcal{X}_L) - \text{bdiag}(\mathcal{Z}_L)\|_F^2 \\ &= \frac{1}{2\rho} \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \|\mathcal{X}_L(:, :, i_3, \dots, i_d) - \mathcal{Z}_L(:, :, i_3, \dots, i_d)\|_F^2. \end{aligned} \quad (19)$$

Then, the problem (17) is equivalent to

$$\begin{aligned} \arg \min_{\mathcal{X}} \frac{1}{\rho} \sum_{i_3=1}^{n_3} \dots \sum_{i_d=1}^{n_d} \left(\tau \|\mathcal{X}_L(:, :, i_3, \dots, i_d)\|_{w,\mathcal{S}_p}^p \right. \\ \left. + \frac{1}{2} \|\mathcal{X}_L(:, :, i_3, \dots, i_d) - \mathcal{Z}_L(:, :, i_3, \dots, i_d)\|_F^2 \right). \end{aligned} \quad (20)$$

Since $0 \leq \mathcal{W}(1, 1, i_3, \dots, i_d) \leq \dots \leq \mathcal{W}(m, m, i_3, \dots, i_d)$, the (i_3, \dots, i_d) -th matrix slice of $L(\mathcal{D}_{\mathcal{W},p,\tau}(\mathcal{Z}))$ solves the (i_3, \dots, i_d) -th subproblem of (20) in virtue of Lemma 3.1. Hence, $\mathcal{D}_{\mathcal{W},p,\tau}(\mathcal{Z})$ solves the problem (17). \square

Algorithm 10: FFT based order- d t-product, tproduct-fft(\mathcal{A}, \mathcal{B}).

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_d}$, $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3 \times \dots \times n_d}$.
Output: $\mathcal{C} = \mathcal{A} * \mathcal{B} \in \mathbb{R}^{n_1 \times l \times n_3 \times \dots \times n_d}$.
1 Compute the result of FFT on \mathcal{A} and \mathcal{B}
2 **for** $i = 3, 4, \dots, d$ **do**
3 $\mathcal{A}_f \leftarrow \text{fft}(\mathcal{A}, [], i)$, $\mathcal{B}_f \leftarrow \text{fft}(\mathcal{B}, [], i)$;
4 **end**
5 Compute the matrix slice of \mathcal{C}_f for given index by
6 **for** $i_3 \in \{2, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i_d \in \{2, \dots, \lceil \frac{n_d+1}{2} \rceil\}$,
7 $i'_3 \in \{1, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i'_{d-1} \in \{1, \dots, \lceil \frac{n_{d-1}+1}{2} \rceil\}$ **do**
8
$$\begin{cases} \mathcal{C}_f^{(1,1,1,\dots,1,1)} = \mathbf{A}_f^{(1,1,1,\dots,1,1)} \cdot \mathbf{B}_f^{(1,1,1,\dots,1,1)}, \\ \mathcal{C}_f^{(i_3,1,1,\dots,1,1)} = \mathbf{A}_f^{(i_3,1,1,\dots,1,1)} \cdot \mathbf{B}_f^{(i_3,1,1,\dots,1,1)}, \\ \mathcal{C}_f^{(i'_3,i_4,1,\dots,1,1)} = \mathbf{A}_f^{(i'_3,i_4,1,\dots,1,1)} \cdot \mathbf{B}_f^{(i'_3,i_4,1,\dots,1,1)}, \\ \mathcal{C}_f^{(i'_3,i'_4,i_5,\dots,1,1)} = \mathbf{A}_f^{(i'_3,i'_4,i_5,\dots,1,1)} \cdot \mathbf{B}_f^{(i'_3,i'_4,i_5,\dots,1,1)}, \\ \dots\dots\dots \\ \mathcal{C}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)} = \mathbf{A}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)} \cdot \mathbf{B}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}. \end{cases}$$

9 **end**
10 Compute the remaining matrix slice of \mathcal{C}_f by conjugate symmetry (22);
11 Compute the result of inverse FFT on \mathcal{C}_f
12 **for** $i = d, d-1, \dots, 3$ **do**
13 $\mathcal{C} \leftarrow \text{ifft}(\mathcal{C}_f, [], i)$.
14 **end**

4. FFT BASED RELEVANT ALGORITHMS

In this section, we present a series of commonly-used algorithms that implement in the Fourier domain. These algorithms can resort to the conjugate symmetry of FFT to reduce the computational cost.

Remark 4.1. When we utilize the Fast Fourier Transform (FFT) as the invertible linear transform L , the block circulant matrix of \mathcal{A} and the block diagonal matrix of \mathcal{A}_f have the following relationship:

$$(\tilde{\mathbf{F}} \otimes \mathbf{I}_{n_1}) \cdot \text{bcirc}(\mathcal{A}) \cdot (\tilde{\mathbf{F}}^{-1} \otimes \mathbf{I}_{n_2}) = \text{bdiag}(\mathcal{A}_f), \quad (21)$$

where $\tilde{\mathbf{F}} = \mathbf{F}_{n_d} \otimes \mathbf{F}_{n_{d-1}} \otimes \dots \otimes \mathbf{F}_{n_3}$, $\tilde{\mathbf{F}}^{-1} = \mathbf{F}_{n_d}^{-1} \otimes \mathbf{F}_{n_{d-1}}^{-1} \otimes \dots \otimes \mathbf{F}_{n_3}^{-1}$, $\mathbf{F}_{n_d} \in \mathbb{C}^{n_d \times n_d}$ denotes the DFT matrix, and $(\mathbf{F}_{n_d} \otimes \mathbf{F}_{n_{d-1}} \otimes \dots \otimes \mathbf{F}_{n_3}) / \sqrt{n_d n_{d-1} \dots n_3}$ is orthogonal. By using the property of real symmetric circulant matrix (see the Definition 1 in [12]), we have

$$\begin{cases} \mathbf{A}_f^{(1,1,1,\dots,1,1)} \in \mathbb{R}^{n_1 \times n_2}, \\ \text{conj}(\mathbf{A}_f^{(i_3,1,\dots,1,1)}) = \mathbf{A}_f^{(n_3-i_3+2,1,\dots,1,1)}, \\ \text{conj}(\mathbf{A}_f^{(i'_3,i_4,1,\dots,1,1)}) = \mathbf{A}_f^{(n'_3,n_4-i_4+2,1,\dots,1,1)}, \\ \text{conj}(\mathbf{A}_f^{(i'_3,i'_4,i_5,1,\dots,1,1)}) = \mathbf{A}_f^{(n'_3,n'_4,n_5-i_5+2,1,\dots,1,1)}, \\ \dots\dots\dots \\ \text{conj}(\mathbf{A}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}) = \mathbf{A}_f^{(n'_3,n'_4,n'_5,\dots,n'_{d-1},n_d-i_d+2)}, \end{cases} \quad (22)$$

for $i_3 \in \{2, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i_d \in \{2, \dots, \lceil \frac{n_d+1}{2} \rceil\}$; $i'_3 \in \{1, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i'_{d-1} \in \{1, \dots, \lceil \frac{n_{d-1}+1}{2} \rceil\}$. Here, $\text{conj}(\cdot)$ is the conjugate operator. The explicit expressions of n'_j ($j = 3, 4, 5, \dots, d$) can be written as follow:

$$n'_j = \begin{cases} n'_j - i'_j + 2, & i'_j \neq 1 \\ 1, & i'_j = 1 \end{cases}.$$

On the contrary, for any given $\mathcal{A}_f \in \mathbb{C}^{n_1 \times \dots \times n_d}$ satisfying (22), there exists a real tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ such that (21) holds. Leveraging on conjugate symmetry (22), the computational cost for order- d t-product, order- d t-SVD, order- d rt-SVD, and order- d GT-SVT operator can be further reduced.

Algorithm 11: FFT based order- d t-SVD, $\text{tsvd-fft}(\mathcal{A})$.

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_{d-1} \times n_d}$.**Output:** t-SVD components \mathcal{U}, \mathcal{S} and \mathcal{V} of \mathcal{A} .

```
1 Compute the result of FFT on  $\mathcal{A}$ 
2 for  $i = 3, 4, \dots, d$  do
3    $\mathcal{A}_f \leftarrow \text{fft}(\mathcal{A}, [ ], i)$ ;
4 end
5 Compute the matrix slice of  $\mathcal{U}_f, \mathcal{S}_f$  and  $\mathcal{V}_f$  from  $\mathcal{A}_f$ 
6 for  $i_3 \in \{2, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i_d \in \{2, \dots, \lceil \frac{n_d+1}{2} \rceil\},$ 
7  $i'_3 \in \{1, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i'_{d-1} \in \{1, \dots, \lceil \frac{n_{d-1}+1}{2} \rceil\}$  do
8   
$$\begin{cases} [U_f^{(1,1,1,\dots,1,1)}, S_f^{(1,1,1,\dots,1,1)}, V_f^{(1,1,1,\dots,1,1)}] = \text{svd}(A_f^{(1,1,1,\dots,1,1)}), \\ [U_f^{(i_3,1,1,\dots,1,1)}, S_f^{(i_3,1,1,\dots,1,1)}, V_f^{(i_3,1,1,\dots,1,1)}] = \text{svd}(A_f^{(i_3,1,1,\dots,1,1)}), \\ [U_f^{(i'_3,i_4,1,\dots,1,1)}, S_f^{(i'_3,i_4,1,\dots,1,1)}, V_f^{(i'_3,i_4,1,\dots,1,1)}] = \text{svd}(A_f^{(i'_3,i_4,1,\dots,1,1)}), \\ [U_f^{(i'_3,i'_4,i_5,\dots,1,1)}, S_f^{(i'_3,i'_4,i_5,\dots,1,1)}, V_f^{(i'_3,i'_4,i_5,\dots,1,1)}] = \text{svd}(A_f^{(i'_3,i'_4,i_5,\dots,1,1)}), \\ \dots\dots\dots \\ [U_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, S_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, V_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}] = \text{svd}(A_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}); \end{cases}$$

9 end
10 Compute the remaining matrix slice of  $\mathcal{U}_f, \mathcal{S}_f$  and  $\mathcal{V}_f$  via (24)-(26);
11 Compute the result of inverse FFT on  $\mathcal{U}_f, \mathcal{S}_f$  and  $\mathcal{V}_f$ 
12 for  $i = d, d-1, \dots, 3$  do
13    $\mathcal{U} \leftarrow \text{ifft}(\mathcal{U}_f, [ ], i), \mathcal{S} \leftarrow \text{ifft}(\mathcal{S}_f, [ ], i), \mathcal{V} \leftarrow \text{ifft}(\mathcal{V}_f, [ ], i).$ 
14 end
```

Algorithm 12: FFT based order- d rt-SVD, $\text{rtsvd-fft}(\mathcal{A}, k, q, p)$.

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_{d-1} \times n_d}$, truncation term $k < \min(n_1, n_2)$, oversampling parameter: $q > 0, p \geq 0$ **Output:** rt-SVD components \mathcal{U}, \mathcal{S} and \mathcal{V} of \mathcal{A} .

```
1 Set  $l = k + q$  and initialize a Gaussian random tensor  $\mathcal{G} \in \mathbb{R}^{n_2 \times l \times n_3 \times \dots \times n_d}$ ;
2 Compute the result of FFT on  $\mathcal{A}$  and  $\mathcal{G}$ 
3 for  $i = 3, 4, \dots, d$  do
4    $\mathcal{A}_f \leftarrow \text{fft}(\mathcal{A}, [ ], i), \mathcal{G}_f \leftarrow \text{fft}(\mathcal{G}, [ ], i)$ ;
5 end
6 Compute the matrix slice of  $\mathcal{U}_f, \mathcal{S}_f$  and  $\mathcal{V}_f$  from  $\mathcal{A}_f$ 
7 for  $i_3 \in \{2, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i_d \in \{2, \dots, \lceil \frac{n_d+1}{2} \rceil\},$ 
8  $i'_3 \in \{1, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i'_{d-1} \in \{1, \dots, \lceil \frac{n_{d-1}+1}{2} \rceil\}$  do
9   
$$\begin{cases} [U_f^{(1,1,1,\dots,1,1)}, S_f^{(1,1,1,\dots,1,1)}, V_f^{(1,1,1,\dots,1,1)}] = \text{r-svd}(A_f^{(1,1,1,\dots,1,1)}, G_f^{(1,1,1,\dots,1,1)}, k, q, p), \\ [U_f^{(i_3,1,1,\dots,1,1)}, S_f^{(i_3,1,1,\dots,1,1)}, V_f^{(i_3,1,1,\dots,1,1)}] = \text{r-svd}(A_f^{(i_3,1,1,\dots,1,1)}, G_f^{(i_3,1,1,\dots,1,1)}, k, q, p), \\ [U_f^{(i'_3,i_4,1,\dots,1,1)}, S_f^{(i'_3,i_4,1,\dots,1,1)}, V_f^{(i'_3,i_4,1,\dots,1,1)}] = \text{r-svd}(A_f^{(i'_3,i_4,1,\dots,1,1)}, G_f^{(i'_3,i_4,1,\dots,1,1)}, k, q, p), \\ [U_f^{(i'_3,i'_4,i_5,\dots,1,1)}, S_f^{(i'_3,i'_4,i_5,\dots,1,1)}, V_f^{(i'_3,i'_4,i_5,\dots,1,1)}] = \text{r-svd}(A_f^{(i'_3,i'_4,i_5,\dots,1,1)}, G_f^{(i'_3,i'_4,i_5,\dots,1,1)}, k, q, p), \\ \dots\dots\dots \\ [U_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, S_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, V_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}] = \text{r-svd}(A_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, \\ G_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, k, q, p); \end{cases}$$

10 end
11 Compute the remaining matrix slice of  $\mathcal{U}_f, \mathcal{S}_f$  and  $\mathcal{V}_f$  via (24)-(26);
12 Compute the result of inverse FFT on  $\mathcal{U}_f, \mathcal{S}_f$  and  $\mathcal{V}_f$ 
13 for  $i = d, d-1, \dots, 3$  do
14    $\mathcal{U} \leftarrow \text{ifft}(\mathcal{U}_f, [ ], i), \mathcal{S} \leftarrow \text{ifft}(\mathcal{S}_f, [ ], i), \mathcal{V} \leftarrow \text{ifft}(\mathcal{V}_f, [ ], i).$ 
15 end
```

Remark 4.2. We let the SVD or randomized SVD (r-SVD) of $\mathbf{A}_f^{(i_3, i_4, \dots, i_{d-1}, i_d)}$ be

$$\begin{cases} [\mathbf{U}_f^{(1,1,1,\dots,1,1)}, \mathbf{S}_f^{(1,1,1,\dots,1,1)}, \mathbf{V}_f^{(1,1,1,\dots,1,1)}] = \Phi(\mathbf{A}_f^{(1,1,1,\dots,1,1)}), \\ [\mathbf{U}_f^{(i_3,1,1,\dots,1,1)}, \mathbf{S}_f^{(i_3,1,1,\dots,1,1)}, \mathbf{V}_f^{(i_3,1,1,\dots,1,1)}] = \Phi(\mathbf{A}_f^{(i_3,1,1,\dots,1,1)}), \\ [\mathbf{U}_f^{(i'_3,i_4,1,\dots,1,1)}, \mathbf{S}_f^{(i'_3,i_4,1,\dots,1,1)}, \mathbf{V}_f^{(i'_3,i_4,1,\dots,1,1)}] = \Phi(\mathbf{A}_f^{(i'_3,i_4,1,\dots,1,1)}), \\ [\mathbf{U}_f^{(i'_3,i'_4,i_5,\dots,1,1)}, \mathbf{S}_f^{(i'_3,i'_4,i_5,\dots,1,1)}, \mathbf{V}_f^{(i'_3,i'_4,i_5,\dots,1,1)}] = \Phi(\mathbf{A}_f^{(i'_3,i'_4,i_5,\dots,1,1)}), \\ \vdots \\ [\mathbf{U}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, \mathbf{S}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, \mathbf{V}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}] = \Phi(\mathbf{A}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}), \end{cases} \quad (23)$$

for $i_3 \in \{2, \dots, \lceil \frac{n_3+1}{2} \rceil\}$, \dots , $i_d \in \{2, \dots, \lceil \frac{n_d+1}{2} \rceil\}$, $i'_3 \in \{1, \dots, \lceil \frac{n_3+1}{2} \rceil\}$, \dots , $i'_{d-1} \in \{1, \dots, \lceil \frac{n_{d-1}+1}{2} \rceil\}$, where Φ denotes decomposition scheme. Here, the singular values in $\mathbf{S}_f^{(i_3, i_4, \dots, i_d)}$ are real. Besides, we let

$$\begin{cases} \mathbf{U}_f^{(1,1,1,\dots,1,1)} \in \mathbb{R}^{n_1 \times n_2}, \\ \mathbf{U}_f^{(i_3,1,1,\dots,1,1)} = \text{conj}(\mathbf{U}_f^{(n_3-i_3+2,1,1,\dots,1,1)}), \\ \mathbf{U}_f^{(i_3,i_4,1,\dots,1,1)} = \text{conj}(\mathbf{U}_f^{(n'_3, n_4-i_4+2,1,\dots,1,1)}), \\ \mathbf{U}_f^{(i_3,i_4,i_5,1,\dots,1)} = \text{conj}(\mathbf{U}_f^{(n'_3, n'_4, n_5-i_5+2,1,\dots,1)}), \\ \vdots \\ \mathbf{U}_f^{(i_3,i_4,i_5,\dots,i_{d-1},i_d)} = \text{conj}(\mathbf{U}_f^{(n'_3, n'_4, n'_5, \dots, n'_{d-1}, n_d-i_d+2)}), \end{cases} \quad (24)$$

$$\begin{cases} \mathbf{S}_f^{(1,1,1,\dots,1,1)} \in \mathbb{R}^{n_1 \times n_2}, \\ \mathbf{S}_f^{(i_3,1,1,\dots,1,1)} = (\mathbf{S}_f^{(n_3-i_3+2,1,1,\dots,1,1)}), \\ \mathbf{S}_f^{(i_3,i_4,1,\dots,1,1)} = (\mathbf{S}_f^{(n'_3, n_4-i_4+2,1,\dots,1,1)}), \\ \mathbf{S}_f^{(i_3,i_4,i_5,1,\dots,1)} = (\mathbf{S}_f^{(n'_3, n'_4, n_5-i_5+2,1,\dots,1)}), \\ \vdots \\ \mathbf{S}_f^{(i_3,i_4,i_5,\dots,i_{d-1},i_d)} = (\mathbf{S}_f^{(n'_3, n'_4, n'_5, \dots, n'_{d-1}, n_d-i_d+2)}), \end{cases} \quad (25)$$

$$\begin{cases} \mathbf{V}_f^{(1,1,1,\dots,1,1)} \in \mathbb{R}^{n_1 \times n_2}, \\ \mathbf{V}_f^{(i_3,1,1,\dots,1,1)} = \text{conj}(\mathbf{V}_f^{(n_3-i_3+2,1,1,\dots,1,1)}), \\ \mathbf{V}_f^{(i_3,i_4,1,\dots,1,1)} = \text{conj}(\mathbf{V}_f^{(n'_3, n_4-i_4+2,1,\dots,1,1)}), \\ \mathbf{V}_f^{(i_3,i_4,i_5,1,\dots,1)} = \text{conj}(\mathbf{V}_f^{(n'_3, n'_4, n_5-i_5+2,1,\dots,1)}), \\ \vdots \\ \mathbf{V}_f^{(i_3,i_4,i_5,\dots,i_{d-1},i_d)} = \text{conj}(\mathbf{V}_f^{(n'_3, n'_4, n'_5, \dots, n'_{d-1}, n_d-i_d+2)}), \end{cases} \quad (26)$$

for $i_3 \in \{\lceil \frac{n_3+1}{2} \rceil + 1, \dots, n_3\}$, \dots , $i_d \in \{\lceil \frac{n_d+1}{2} \rceil + 1, \dots, n_d\}$.

In virtue of (22)-(26), we present FFT based order- d t-product, FFT based order- d t-SVD, FFT based order- d rt-SVD, FFT based order- d t-QR, and FFT based order- d GT-SVT operator in Algorithm 10-14, respectively.

5. REFERENCES

- [1] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [2] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Alg. Appl.*, vol. 435, no. 3, pp. 641–658, 2011.
- [3] 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, 2013.

Algorithm 13: FFT based order- d t-QR decomposition, $\text{tqr-fft}(\mathcal{A})$.

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_{d-1} \times n_d}$.

Output: t-QR decomposition components \mathcal{Q} and \mathcal{R} of \mathcal{A} .

```
1 Compute the result of FFT on  $\mathcal{A}$ 
2 for  $i = 3, 4, \dots, d$  do
3    $\mathcal{A}_f \leftarrow \text{fft}(\mathcal{A}, [], i)$ ;
4 end
5 Compute the matrix slice of  $\mathcal{Q}_f$  and  $\mathcal{R}_f$  from  $\mathcal{A}_f$ 
6 for  $i_3 \in \{2, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i_d \in \{2, \dots, \lceil \frac{n_d+1}{2} \rceil\}$ ,
7  $i'_3 \in \{1, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i'_{d-1} \in \{1, \dots, \lceil \frac{n_{d-1}+1}{2} \rceil\}$  do
8   
$$\begin{cases} [\mathcal{Q}_f^{(1,1,1,\dots,1,1)}, \mathcal{R}_f^{(1,1,1,\dots,1,1)}] = \text{qr}(\mathcal{A}_f^{(1,1,1,\dots,1,1)}, 0), \\ [\mathcal{Q}_f^{(i_3,1,1,\dots,1,1)}, \mathcal{R}_f^{(i_3,1,1,\dots,1,1)}] = \text{qr}(\mathcal{A}_f^{(i_3,1,1,\dots,1,1)}, 0), \\ [\mathcal{Q}_f^{(i'_3,i_4,1,\dots,1,1)}, \mathcal{R}_f^{(i'_3,i_4,1,\dots,1,1)}] = \text{qr}(\mathcal{A}_f^{(i'_3,i_4,1,\dots,1,1)}, 0), \\ [\mathcal{Q}_f^{(i'_3,i'_4,i_5,\dots,1,1)}, \mathcal{R}_f^{(i'_3,i'_4,i_5,\dots,1,1)}] = \text{qr}(\mathcal{A}_f^{(i'_3,i'_4,i_5,\dots,1,1)}, 0), \\ \dots\dots\dots \\ [\mathcal{Q}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, \mathcal{R}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}] = \text{qr}(\mathcal{A}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}, 0); \end{cases}$$

9 end
10 Compute the remaining matrix slice of  $\mathcal{Q}_f$  and  $\mathcal{R}_f$  via by conjugate symmetry (22);
11 Compute the result of inverse FFT on  $\mathcal{Q}_f$  and  $\mathcal{R}_f$ 
12 for  $i = d, d-1, \dots, 3$  do
13    $\mathcal{Q} \leftarrow \text{ifft}(\mathcal{Q}_f, [], i)$ ,  $\mathcal{R} \leftarrow \text{ifft}(\mathcal{R}_f, [], i)$ .
14 end
```

Algorithm 14: FFT based order- d GT-SVT operator.

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, weight parameter: $\mathcal{W} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, truncation term: $k < \min(n_1, n_2)$, $\tau > 0$, $0 < p < 1$, the number of GST and PowerMethod iteration: J, I , oversampling parameter: $q > 0$.

Output: $\mathcal{D}_{\mathcal{W},p,\tau}(\mathcal{A}) = \mathcal{U} *_L \mathcal{S}_{\mathcal{W},p,\tau} *_L \mathcal{V}^*$.

```
1 Compute the rt-SVD or t-SVD components  $\mathcal{U}, \mathcal{S}$  and  $\mathcal{V}$  of  $\mathcal{A}$ 
2 if utilize the rt-SVD scheme then
3    $[\mathcal{U}, \mathcal{S}, \mathcal{V}] = \text{rtsvd-fft}(\mathcal{A}, k, q, I)$ ;
4 end
5 else if utilize the t-SVD scheme then
6    $[\mathcal{U}, \mathcal{S}, \mathcal{V}] = \text{tsvd-fft}(\mathcal{A})$ ;
7 end
8 Compute the matrix slice of  $\mathcal{Z}_L$  for given index by
9 for  $i_3 \in \{2, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i_d \in \{2, \dots, \lceil \frac{n_d+1}{2} \rceil\}$ 
10  $i'_3 \in \{1, \dots, \lceil \frac{n_3+1}{2} \rceil\}, \dots, i'_{d-1} \in \{1, \dots, \lceil \frac{n_{d-1}+1}{2} \rceil\}$  do
11   
$$\begin{cases} \mathcal{Z}_f^{(1,1,1,\dots,1,1)} = \text{diag} \left\{ \text{GST} \left( \text{diag}(\mathcal{S}_f^{(1,1,1,\dots,1,1)}), \tau \text{diag}(\mathcal{W}^{(1,1,1,\dots,1,1)}), p, J \right) \right\}, \\ \mathcal{Z}_f^{(i_3,1,1,\dots,1,1)} = \text{diag} \left\{ \text{GST} \left( \text{diag}(\mathcal{S}_f^{(i_3,1,1,\dots,1,1)}), \tau \text{diag}(\mathcal{W}^{(i_3,1,1,\dots,1,1)}), p, J \right) \right\}, \\ \mathcal{Z}_f^{(i'_3,i_4,1,\dots,1,1)} = \text{diag} \left\{ \text{GST} \left( \text{diag}(\mathcal{S}_f^{(i'_3,i_4,1,\dots,1,1)}), \tau \text{diag}(\mathcal{W}^{(i'_3,i_4,1,\dots,1,1)}), p, J \right) \right\}, \\ \dots\dots\dots \\ \mathcal{Z}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)} = \text{diag} \left\{ \text{GST} \left( \text{diag}(\mathcal{S}_f^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}), \tau \text{diag}(\mathcal{W}^{(i'_3,i'_4,i'_5,\dots,i'_{d-1},i_d)}), p, J \right) \right\}; \end{cases}$$

12 end
13 Compute the remaining matrix slice of  $\mathcal{Z}_f$  by conjugate symmetry (22);
14 Compute the result of inverse FFT on  $\mathcal{Z}_f$ 
15 for  $i = d, d-1, \dots, 3$  do
16    $\mathcal{S}_{\mathcal{W},p,\tau} \leftarrow \text{ifft}(\mathcal{Z}_f, [], i)$ ;
17 end
18 Compute  $\mathcal{D}_{\mathcal{W},p,\tau}(\mathcal{A}) = \text{tproduct-fft}(\text{tproduct-fft}(\mathcal{U}, \mathcal{S}_{\mathcal{W},p,\tau}), \mathcal{V}^*)$ .
```

- [4] 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, 2013.
- [5] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor–tensor products with invertible linear transforms," *Linear Alg. Appl.*, vol. 485, pp. 545–570, 2015.
- [6] W. Qin, H. Wang, F. Zhang, M. Dai, and J. Wang, "Robust low-rank tensor reconstruction using high-order t-SVD," *Journal of Electronic Imaging*, vol. 30, no. 6, p. 063016, 2021.
- [7] 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, 2022.
- [8] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.
- [9] Q. Yao and J. T. Kwok, "Accelerated and inexact soft-impute for large-scale matrix and tensor completion," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1665–1679, 2018.
- [10] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, "A generalized iterated shrinkage algorithm for non-convex sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 217–224.
- [11] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted Schatten p -norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4842–4857, 2016.
- [12] O. Rojo and H. Rojo, "Some results on symmetric circulant matrices and on symmetric centrosymmetric matrices," *Linear algebra and its applications*, vol. 392, pp. 211–233, 2004.