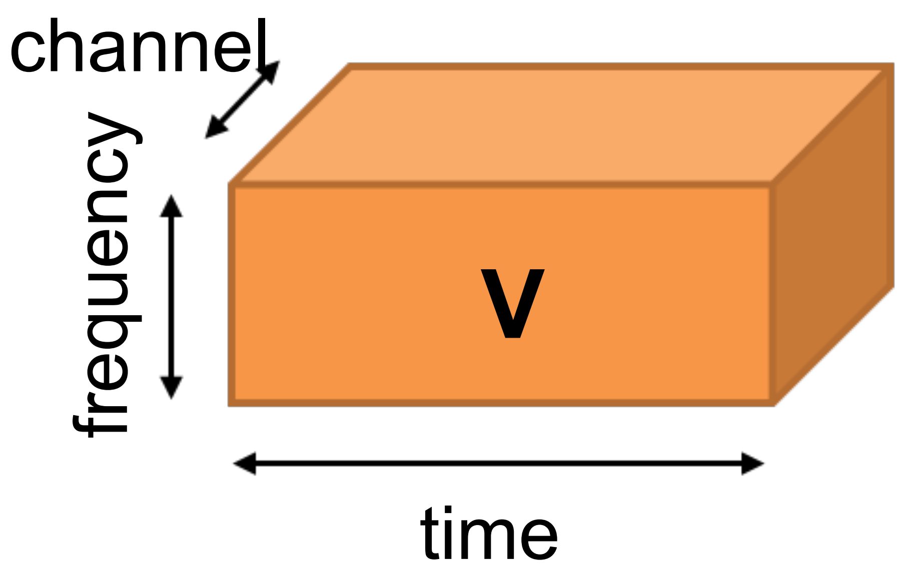
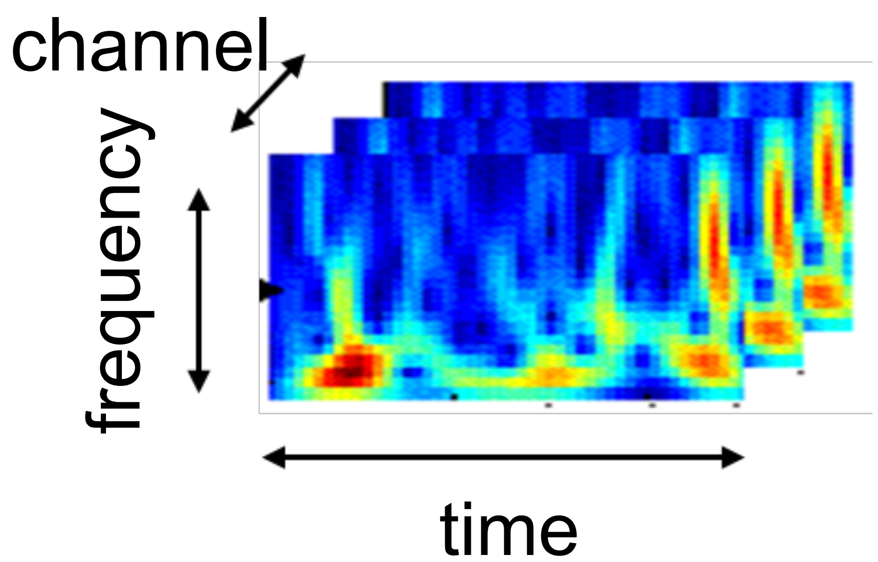
What is tensor?

Tensor is multidimensional array of numerical values and has more **meaningful representation**

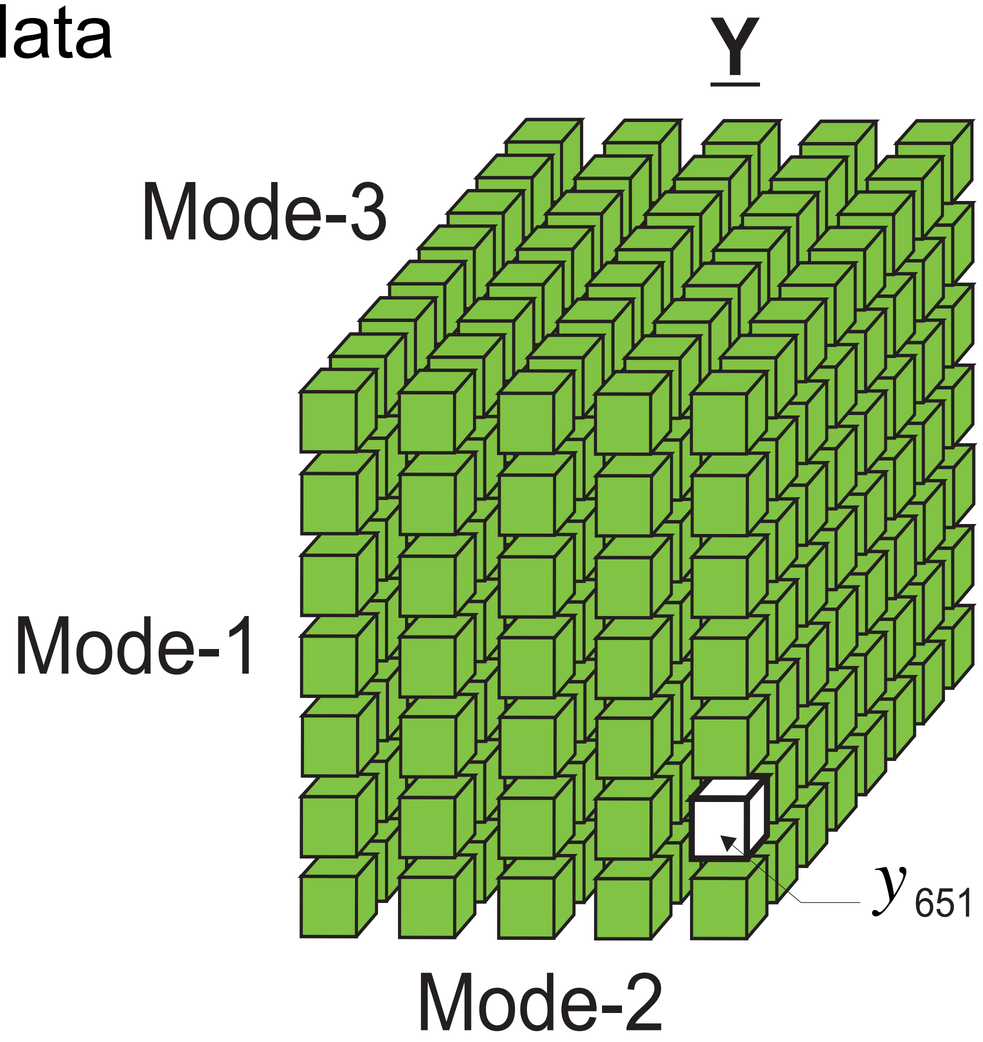
Electroencephalography (EEG) data



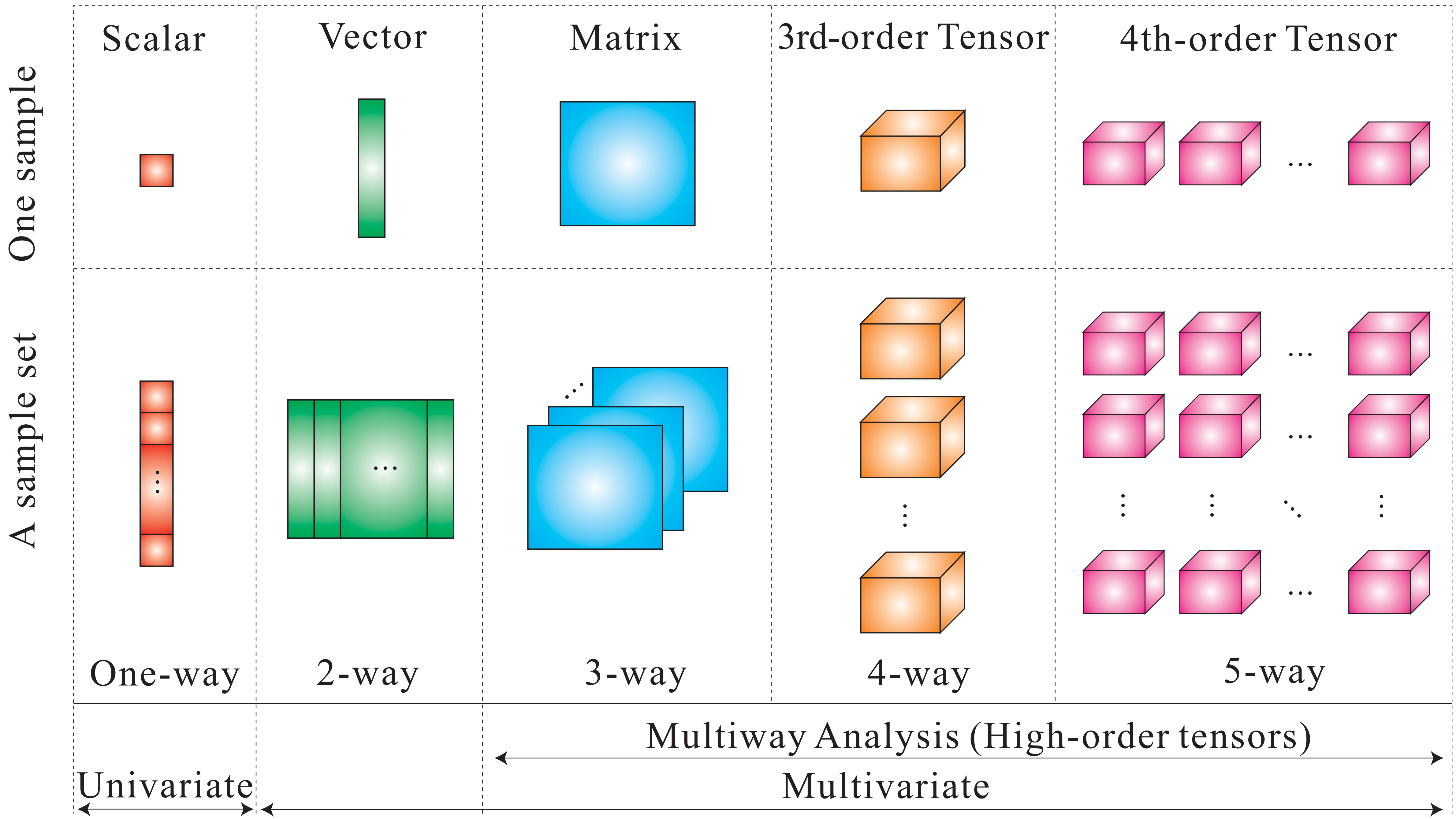


What is tensor?

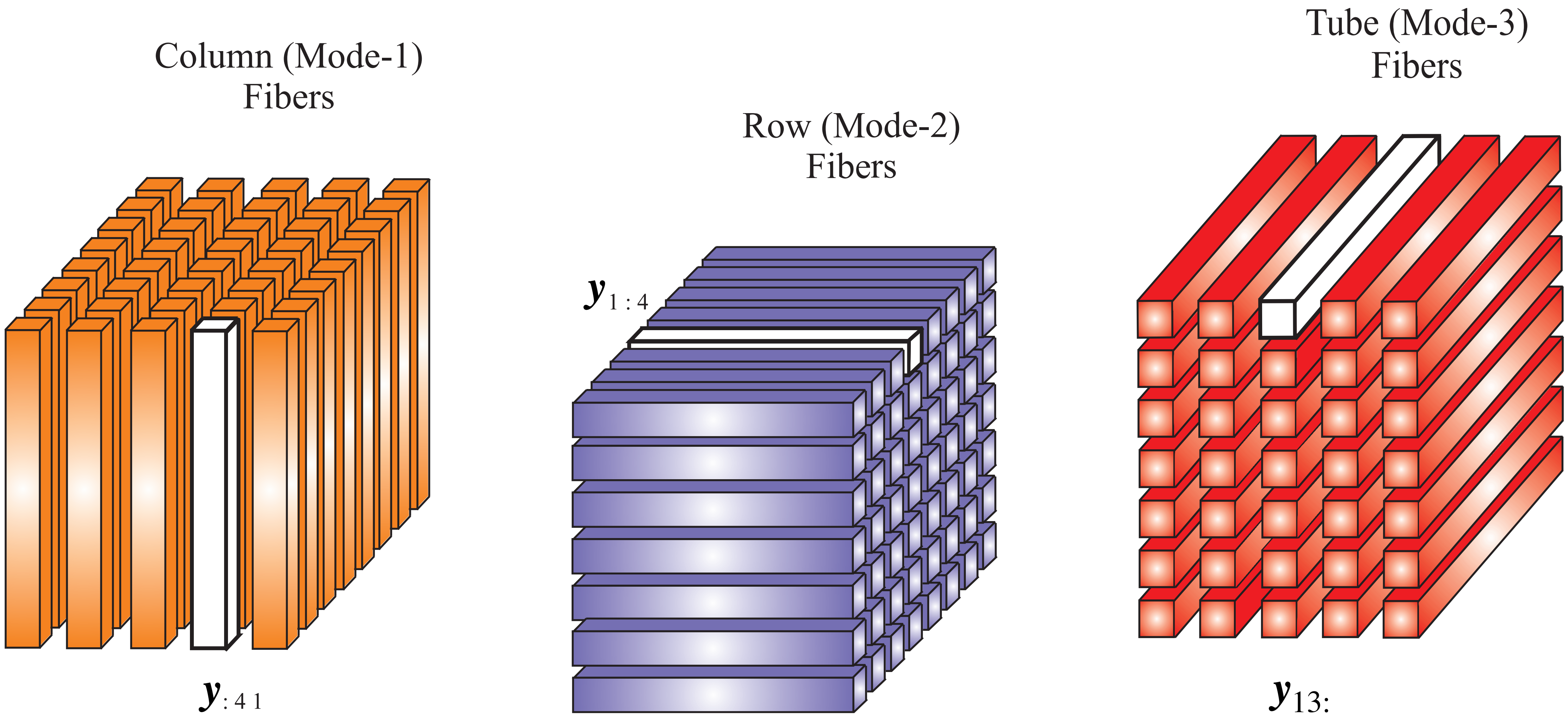
3rd-order tensor data



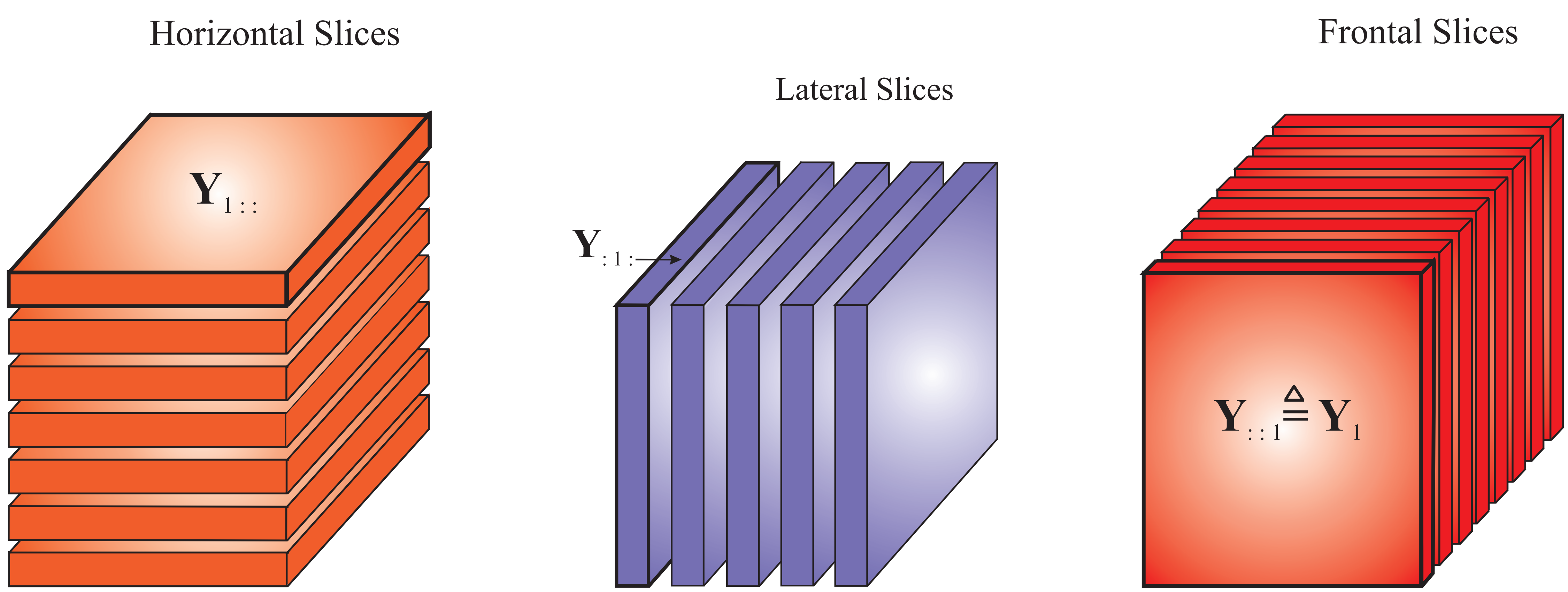
What is tensor?



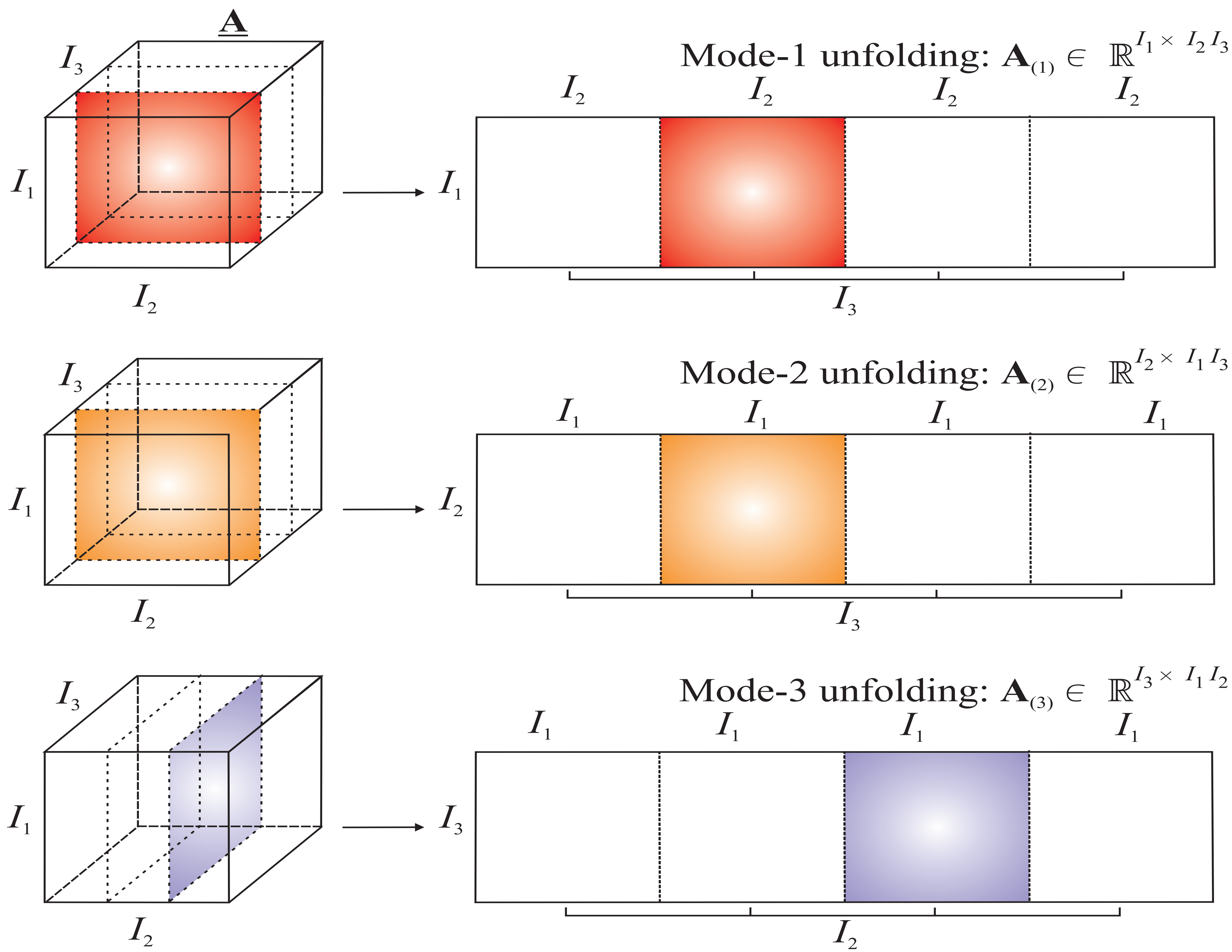
Tensor Fibers



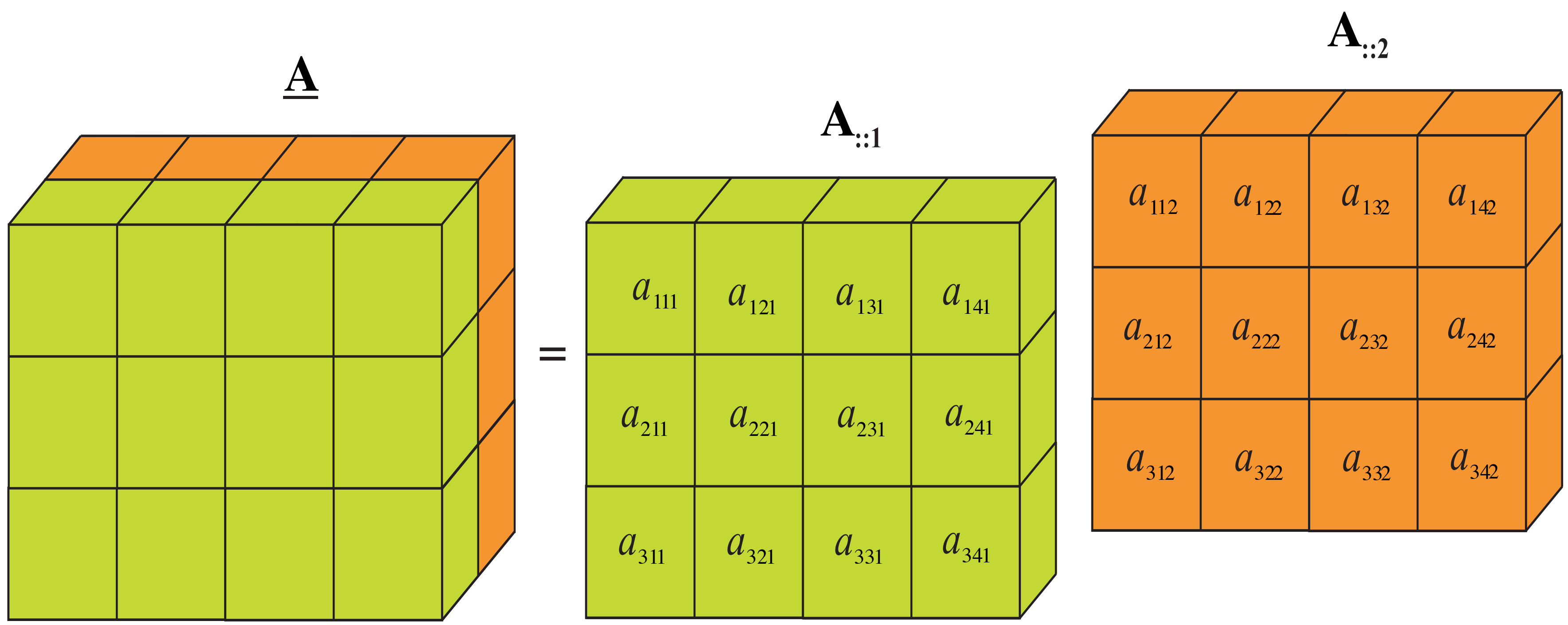
Tensor slices



Tensor Unfolding



An Example of Tensor Unfolding



$$\mathbf{A}_{(1)} = \left[\begin{array}{cccc|cccc} a_{111} & a_{121} & a_{131} & a_{141} & a_{112} & a_{122} & a_{132} & a_{142} \\ a_{211} & a_{221} & a_{231} & a_{241} & a_{212} & a_{222} & a_{232} & a_{242} \\ a_{311} & a_{321} & a_{331} & a_{341} & a_{312} & a_{322} & a_{332} & a_{342} \end{array}\right]$$

$$\mathbf{A}_{(2)} = \left[\begin{array}{ccc|ccc} a_{111} & a_{211} & a_{311} & a_{112} & a_{212} & a_{312} \\ a_{121} & a_{221} & a_{321} & a_{122} & a_{222} & a_{322} \\ a_{131} & a_{231} & a_{331} & a_{132} & a_{232} & a_{332} \\ a_{141} & a_{241} & a_{341} & a_{142} & a_{242} & a_{342} \end{array}\right]$$

$$\mathbf{A}_{(3)} = \left[\begin{array}{ccc|ccc|ccc|ccc} a_{111} & a_{211} & a_{311} & a_{121} & a_{221} & a_{321} & a_{131} & a_{231} & a_{331} & a_{141} & a_{241} & a_{341} \\ a_{112} & a_{212} & a_{312} & a_{122} & a_{222} & a_{322} & a_{132} & a_{232} & a_{332} & a_{142} & a_{242} & a_{342} \end{array}\right]$$

Matrix Products

Matrix Outer Product:

The outer product of the tensors $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\underline{\mathbf{X}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M}$ is given by

$$\underline{\mathbf{Z}} = \mathbf{Y} \circ \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times J_1 \times J_2 \times \cdots \times J_M},$$

where

$$z_{i_1, i_2, \ldots, i_N, j_1, j_2, \ldots, j_M} = y_{i_1, i_2, \ldots, i_N} \; x_{j_1, j_2, \ldots, j_M}.$$

Matrix Kronecker Product:

The Kronecker product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{T \times R}$ is a matrix denoted as $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{IT \times JR}$ and defined as (see the MATLAB function `kron`):

$$\begin{aligned}
\mathbf{A} \otimes \mathbf{B} &= \begin{bmatrix} a_{11}\,\mathbf{B} & a_{12}\,\mathbf{B} & \cdots & a_{1J}\,\mathbf{B} \\ a_{21}\,\mathbf{B} & a_{22}\,\mathbf{B} & \cdots & a_{2J}\,\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\,\mathbf{B} & a_{I2}\,\mathbf{B} & \cdots & a_{IJ}\,\mathbf{B} \end{bmatrix} \\
&= \left[\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \;\; \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 \;\; \boldsymbol{a}_1 \otimes \boldsymbol{b}_3 \;\; \cdots \;\; \boldsymbol{a}_J \otimes \boldsymbol{b}_{R-1} \;\; \boldsymbol{a}_J \otimes \boldsymbol{b}_R \right].
\end{aligned}$$

Matrix Products

Matrix Hadamard Product:

The Hadamard product of two equal-size matrices is the element-wise product denoted by $\circledast$ and defined as

$$\mathbf{A} \circledast \mathbf{B} = \begin{bmatrix} a_{11}\, b_{11} & a_{12}\, b_{12} & \cdots & a_{1J}\, b_{1J} \\ a_{21}\, b_{21} & a_{22}\, b_{22} & \cdots & a_{2J}\, b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\, b_{I1} & a_{I2}\, b_{I2} & \cdots & a_{IJ}\, b_{IJ} \end{bmatrix}.$$

Matrix Khatri-Rao Product:

For two matrices $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_J] \in \mathbb{R}^{I\times J}$ and $\mathbf{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_J] \in \mathbb{R}^{T\times J}$ with the same number of columns J, their Khatri-Rao product, denoted by $\odot$, performs the following operation:

$$\begin{aligned} \mathbf{A} \odot \mathbf{B} &= [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \;\; \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \;\; \cdots \;\; \boldsymbol{a}_J \otimes \boldsymbol{b}_J] \\ &= \left[\mathrm{vec}(\boldsymbol{b}_1 \boldsymbol{a}_1^T) \;\; \mathrm{vec}(\boldsymbol{b}_2 \boldsymbol{a}_2^T) \;\; \cdots \;\; \mathrm{vec}(\boldsymbol{b}_J \boldsymbol{a}_J^T)\right] \in \mathbb{R}^{IT\times J}. \end{aligned}$$

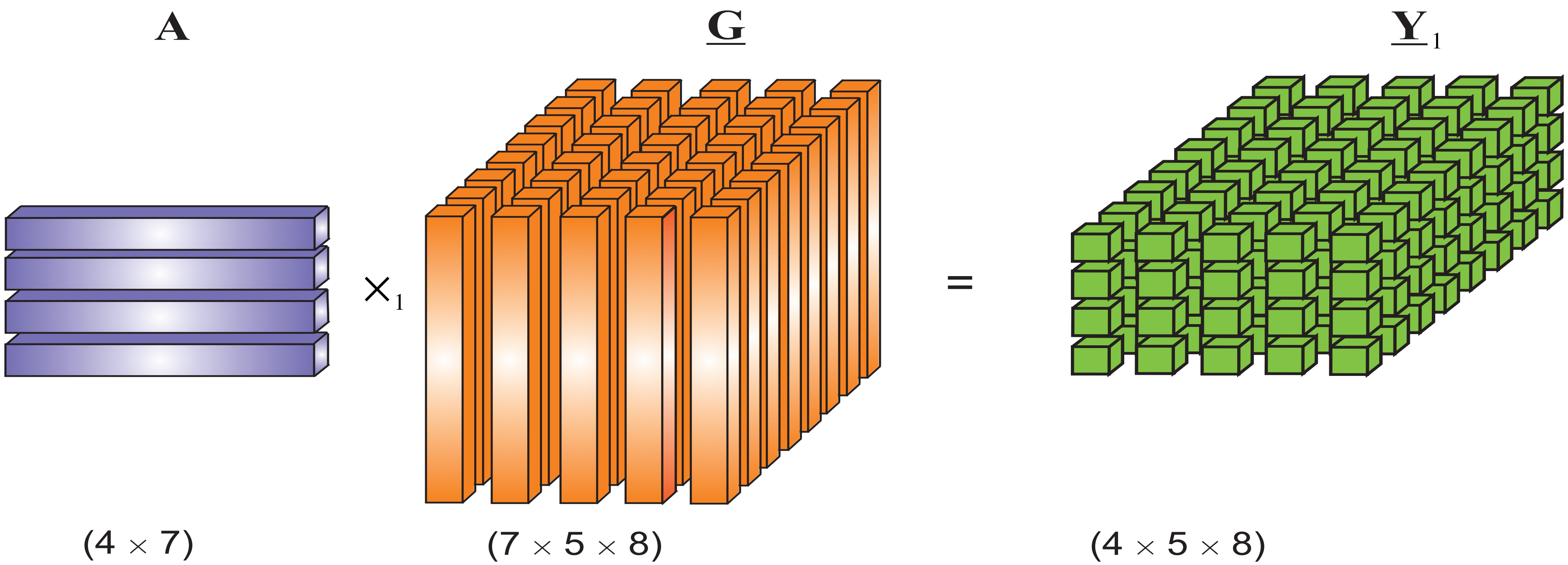
Tensor Matrix Product

Mode-n Product:

The Tensor matrix product is the mode-n unfolding matrix product denoted by $\times_n$ and defined as

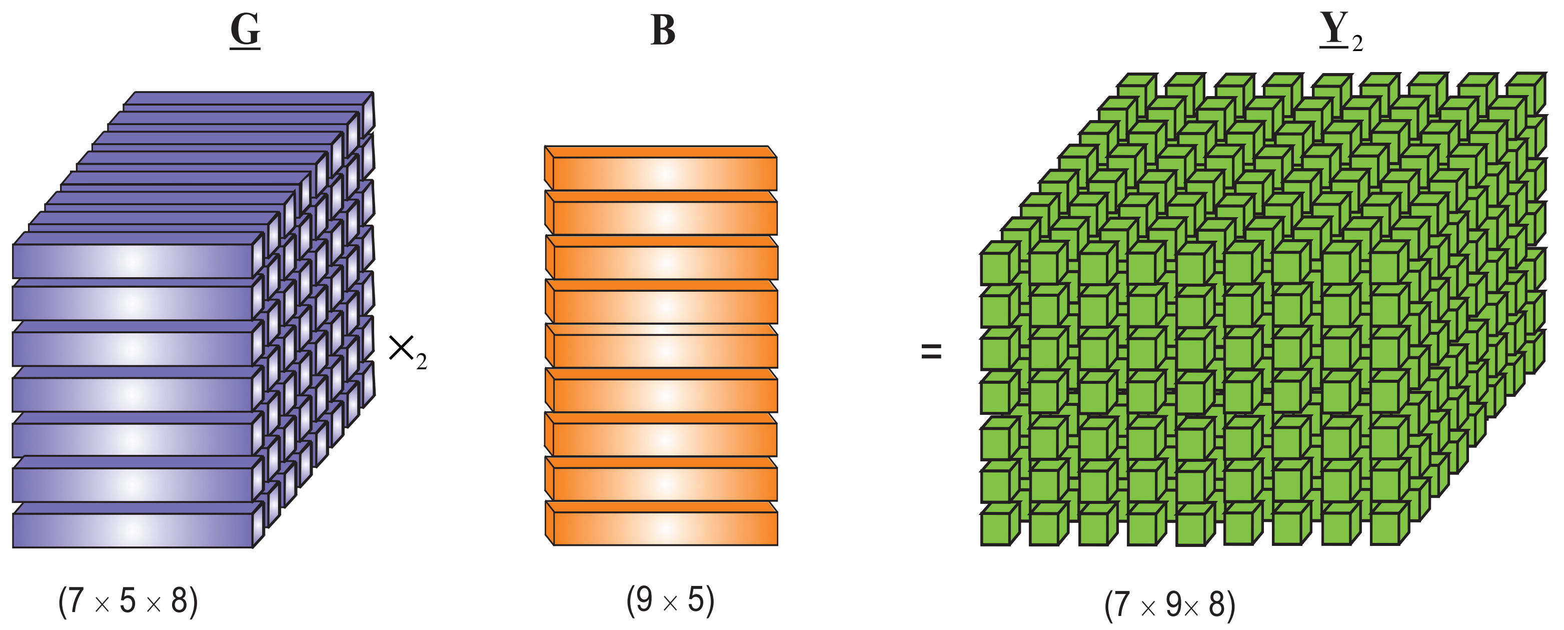
$$y_{j_1,j_2,\ldots,j_{n-1},i_n,j_{n+1},\ldots,j_N} = \sum_{j_n=1}^{J_n} g_{j_1,j_2,\ldots,J_N}\ a_{i_n,j_n}.$$

(a)

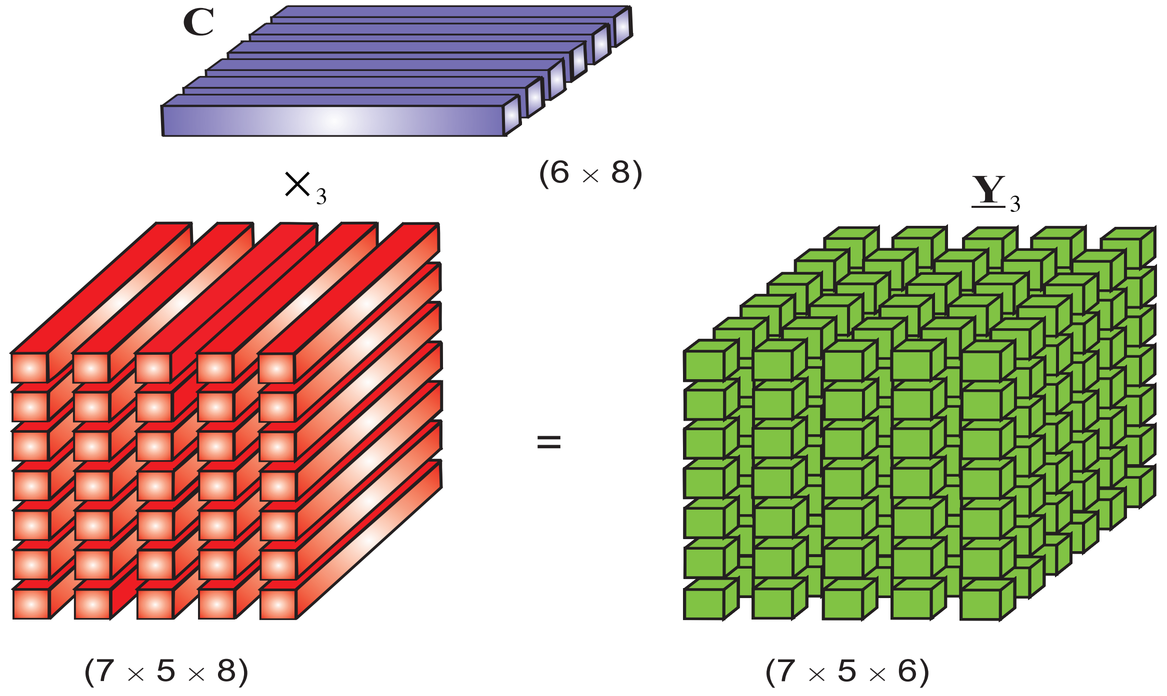


Tensor Matrix Product

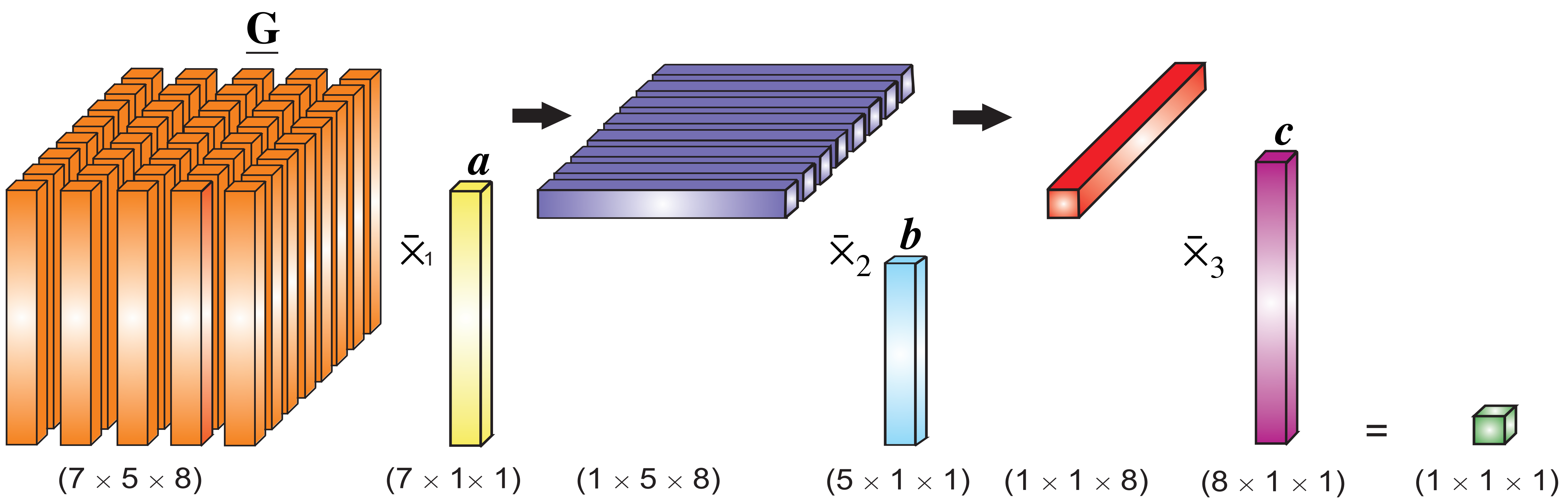
(b)



(c)

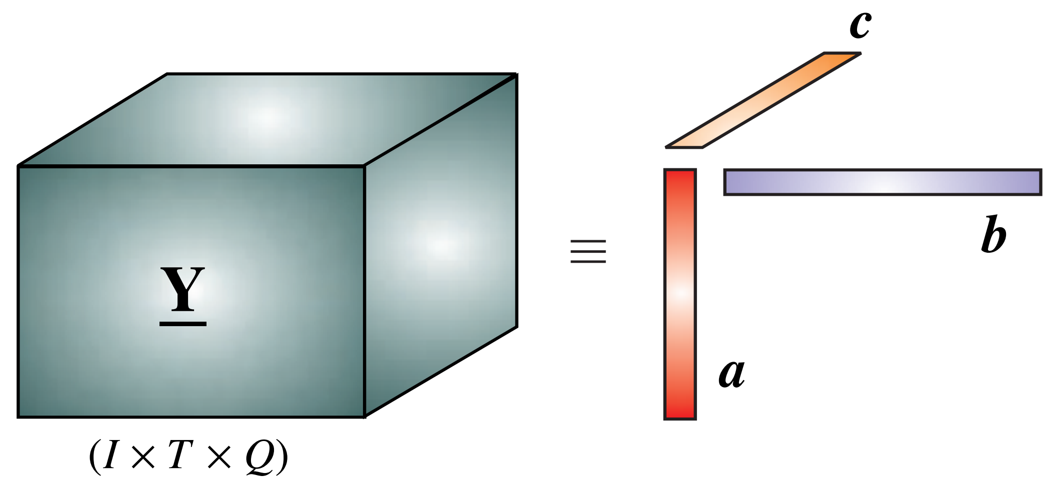


Tensor Vector Contracted Product

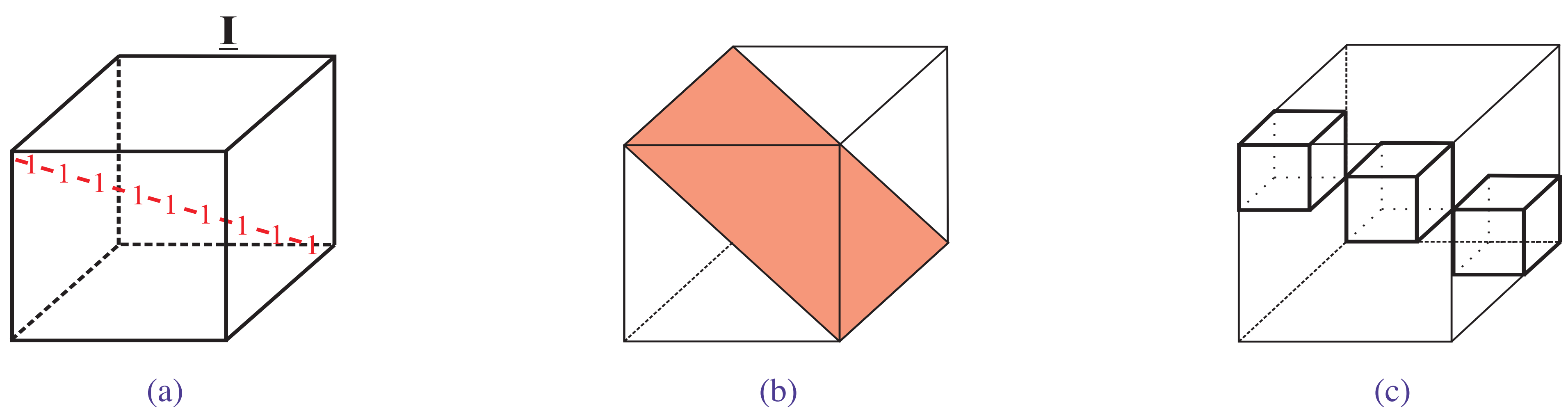


Special Form of Tensors

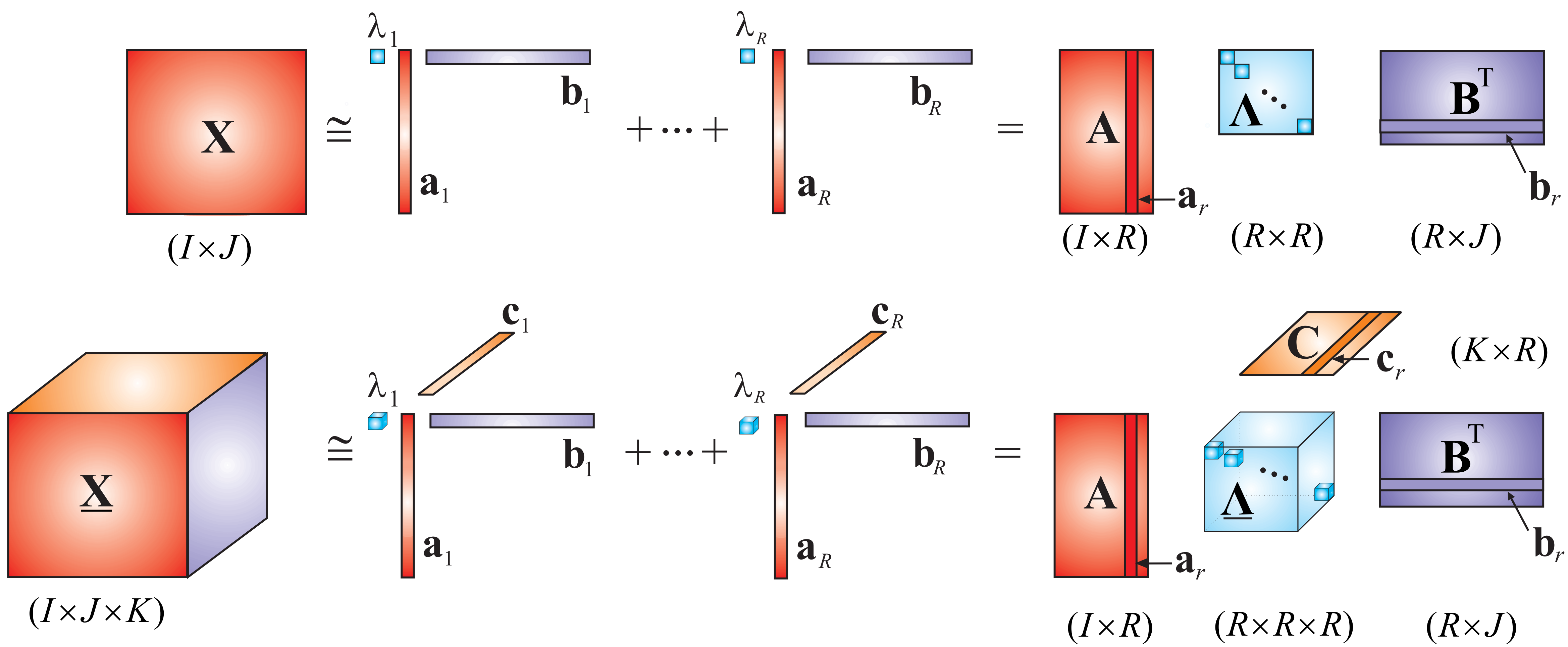
Rank-one tensor:



Examples of tensors with special forms



CP Approximation



$$\underline{\mathbf{X}} \cong \sum_{r=1}^{R} \lambda_r \, \mathbf{b}_r^{(1)} \circ \mathbf{b}_r^{(2)} \circ \cdots \circ \mathbf{b}_r^{(N)}$$

$$= \underline{\mathbf{\Lambda}} \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times_N \mathbf{B}^{(N)}$$

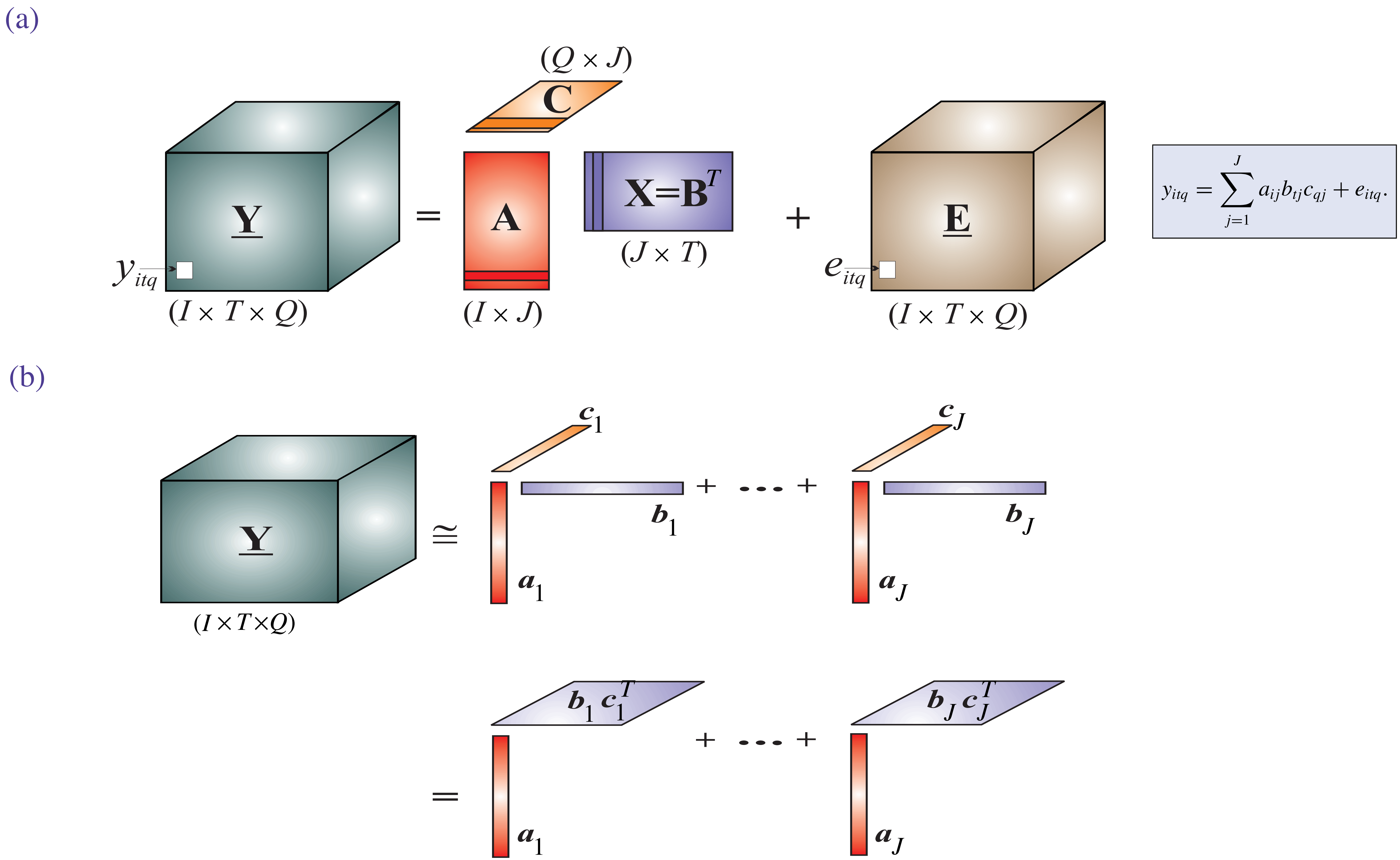
$$= [\![\underline{\mathbf{\Lambda}}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!],$$

$$\mathbf{X}_{(1)} = \mathbf{A}\mathbf{\Lambda}\,(\mathbf{C} \odot \mathbf{B})^T + \mathbf{E}_{(1)}$$

$$\mathbf{X}_{(2)} = \mathbf{B}\mathbf{\Lambda}\,(\mathbf{C} \odot \mathbf{A})^T + \mathbf{E}_{(2)}$$

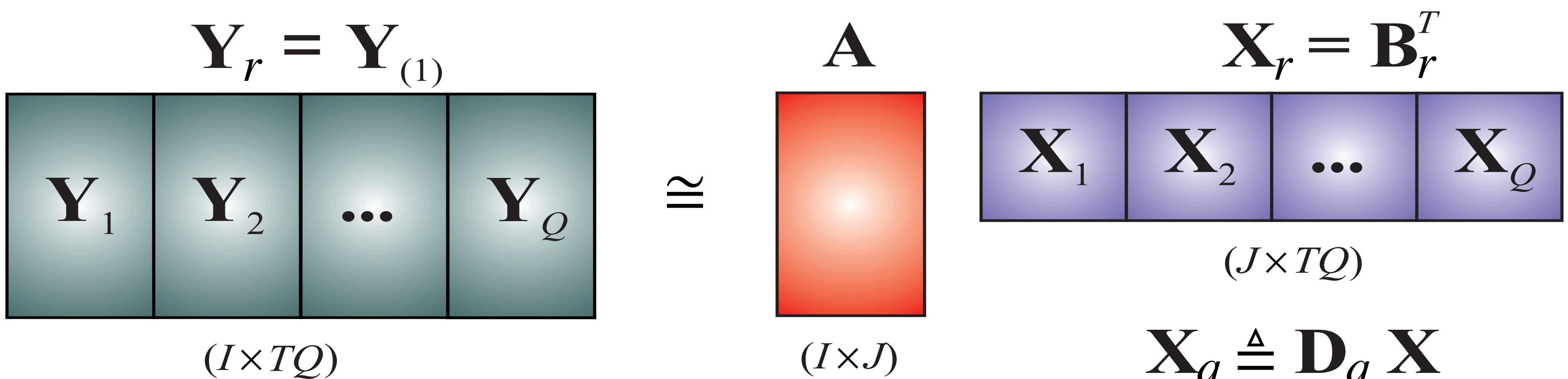
$$\mathbf{X}_{(3)} = \mathbf{C}\mathbf{\Lambda}\,(\mathbf{B} \odot \mathbf{A})^T + \mathbf{E}_{(3)}$$

Alternative Representations



Alternative Representations

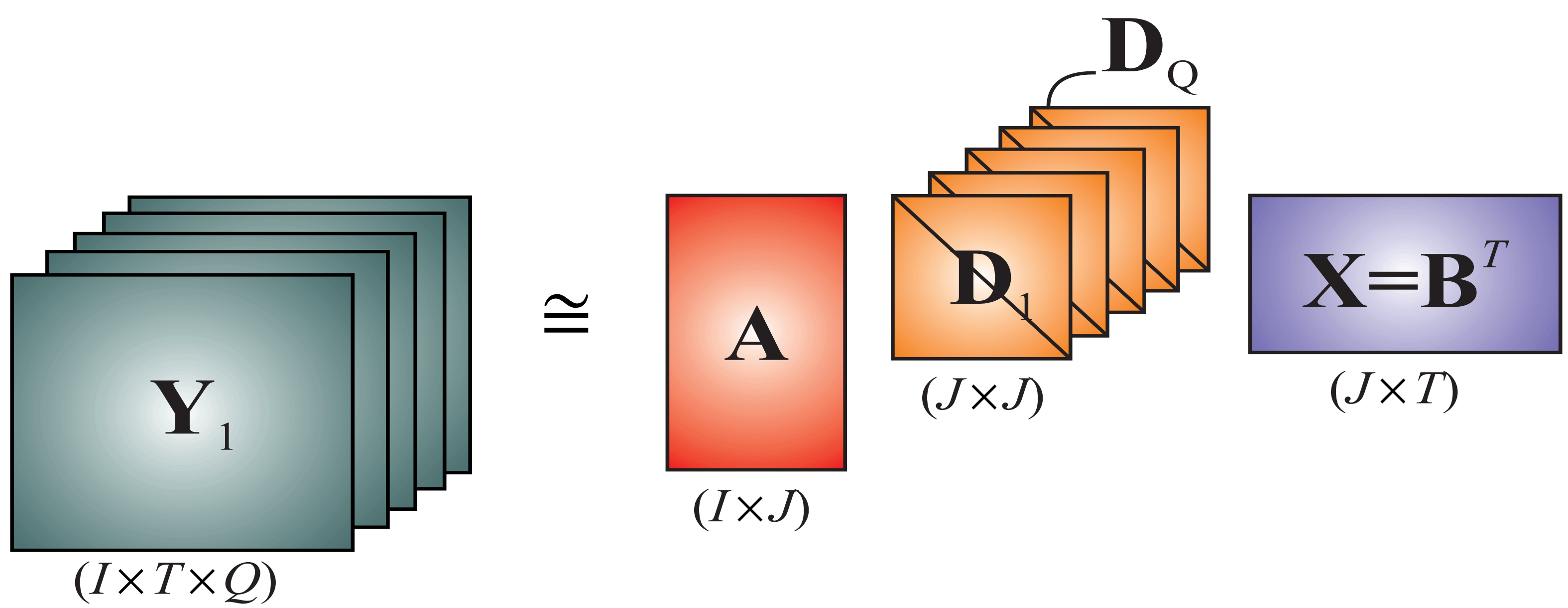
(c)



$$\mathbf{X}_q \triangleq \mathbf{D}_q \, \mathbf{X}$$

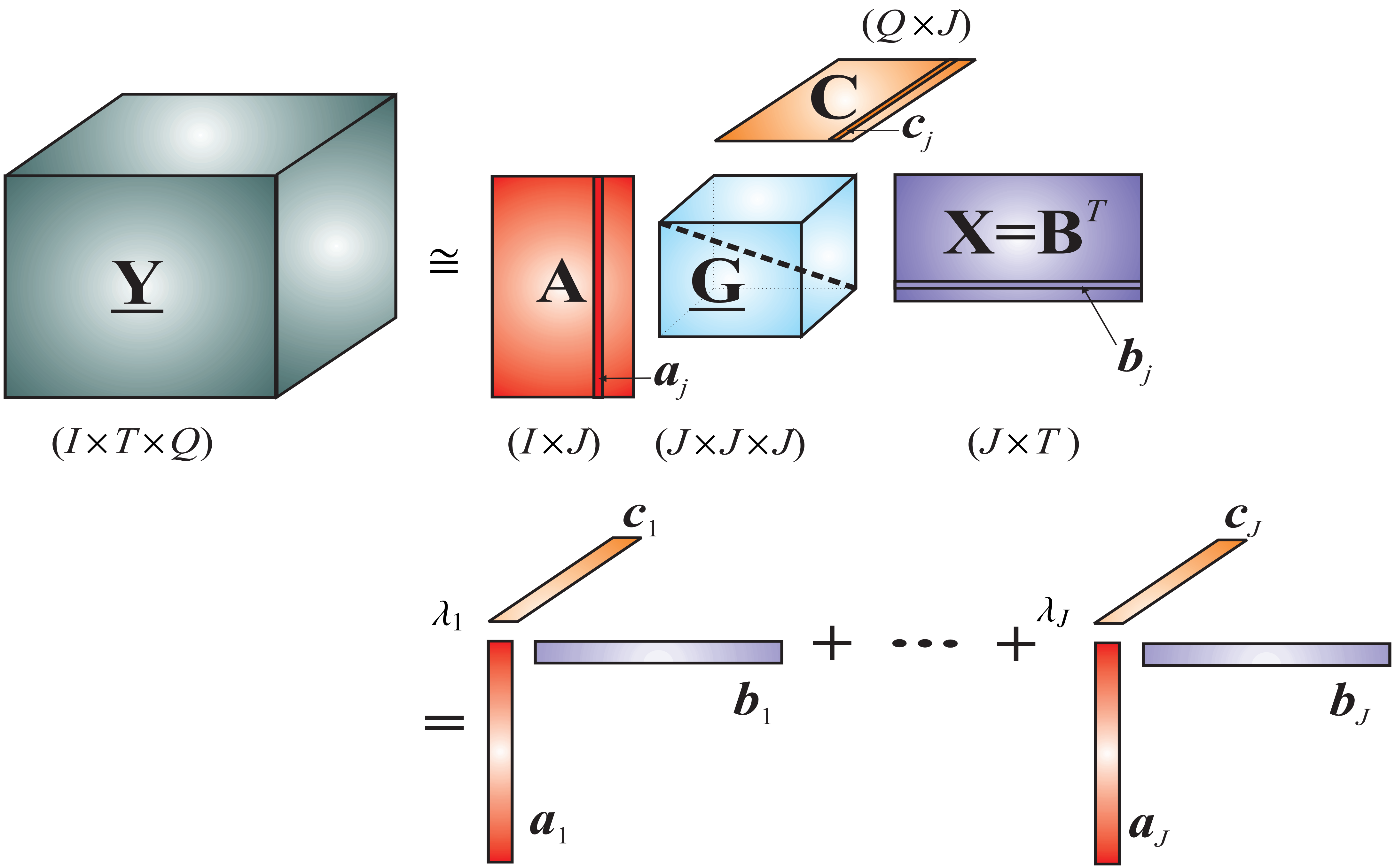
$$(q = 1, 2, \ldots, Q)$$

(d)



$$\mathbf{Y}_q = \mathbf{A}\mathbf{D}_q\mathbf{X}, \qquad (q = 1, 2, \ldots, Q)$$

Alternative Representations



ALS

Algorithm 1: Basic ALS for the CP decomposition of a 3rd-order tensor

Input: Data tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ and rank R

Output: Factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$, and scaling vector $\boldsymbol{\lambda} \in \mathbb{R}^{R}$

Initialize $\mathbf{A}, \mathbf{B}, \mathbf{C}$

while not converged or iteration limit is not reached **do**

  $\mathbf{A} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^{\mathrm{T}}\mathbf{C} \circledast \mathbf{B}^{\mathrm{T}}\mathbf{B})^{\dagger}$

  Normalize column vectors of $\mathbf{A}$ to unit length (by computing the norm of each column vector and dividing each element of a vector by its norm)

  $\mathbf{B} \leftarrow \mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})(\mathbf{C}^{\mathrm{T}}\mathbf{C} \circledast \mathbf{A}^{\mathrm{T}}\mathbf{A})^{\dagger}$

  Normalize column vectors of $\mathbf{B}$ to unit length

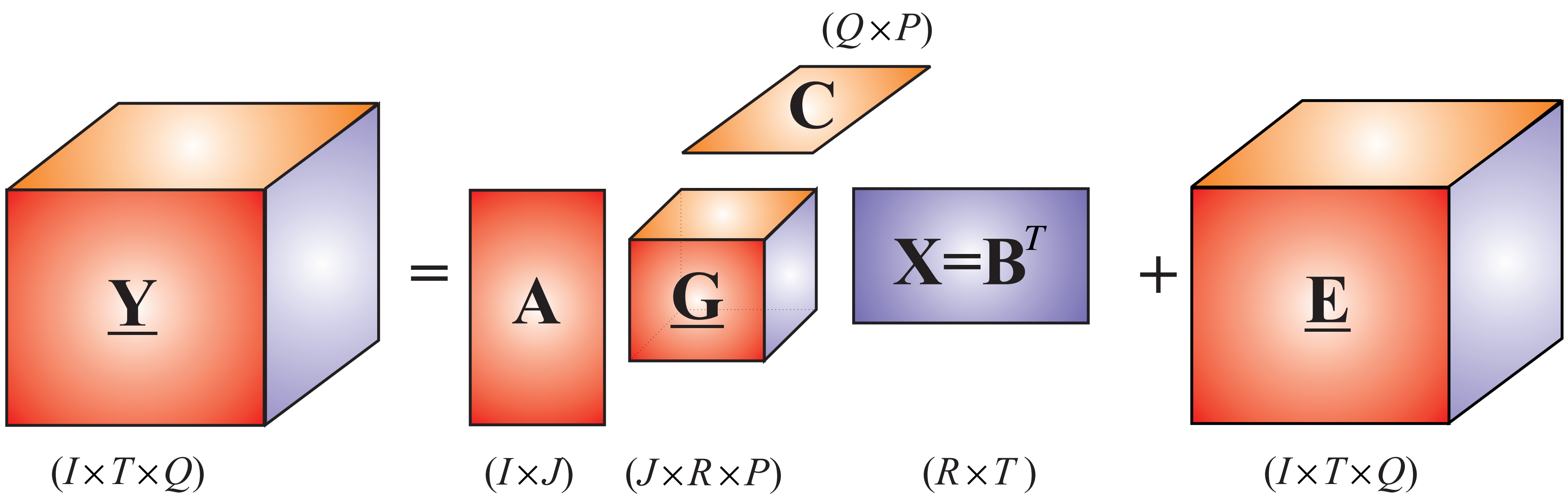
  $\mathbf{C} \leftarrow \mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})(\mathbf{B}^{\mathrm{T}}\mathbf{B} \circledast \mathbf{C}^{\mathrm{T}}\mathbf{C})^{\dagger}$

  Normalize column vectors of $\mathbf{C}$ to unit length, store the norms in vector $\boldsymbol{\lambda}$

end while

return $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\boldsymbol{\lambda}$.

Tucker Approximation



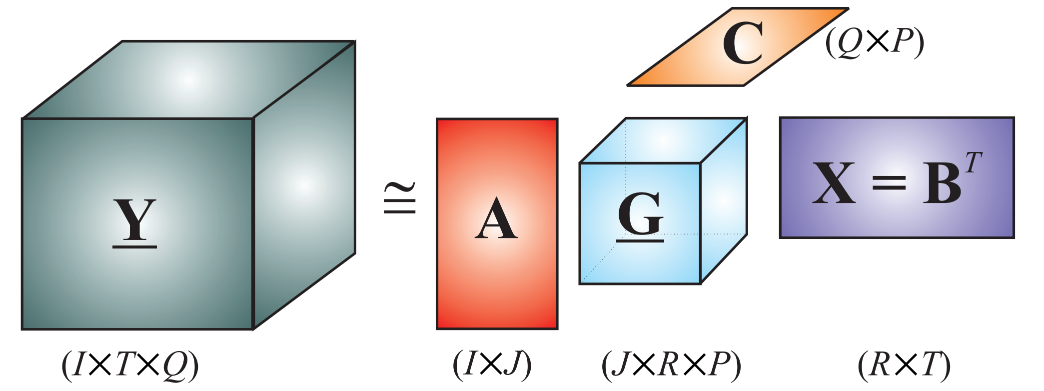
$$\underline{\mathbf{Y}} = \underline{\mathbf{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} + \underline{\mathbf{E}} = [\![\underline{\mathbf{G}}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] + \underline{\mathbf{E}},$$

Matrix Form of Tucker Decomposition

$$\mathbf{X}_{(1)} \approx \mathbf{A}\mathbf{G}_{(1)}(\mathbf{C} \otimes \mathbf{B})^{\mathsf{T}},$$
$$\mathbf{X}_{(2)} \approx \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^{\mathsf{T}},$$
$$\mathbf{X}_{(3)} \approx \mathbf{C}\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^{\mathsf{T}}.$$

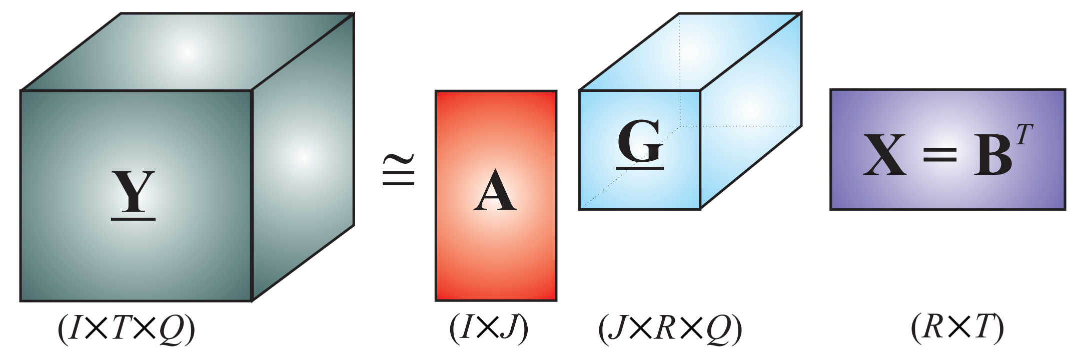
Alternative Representations

$$y_{itq} = \sum_{j=1}^{J} \sum_{r=1}^{R} \sum_{p=1}^{P} g_{jrp}\, a_{ij}\, b_{tr}\, c_{qp}$$



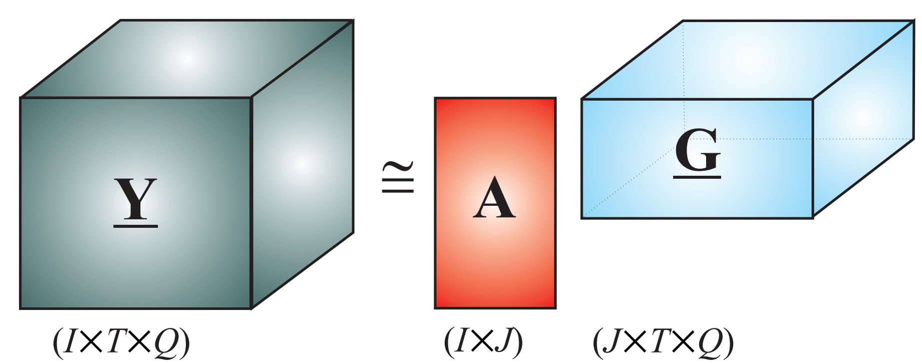
(a) Tucker3

$$y_{itq} = \sum_{j=1}^{J} \sum_{r=1}^{R} g_{jrq}\, a_{ij}\, b_{tr}$$



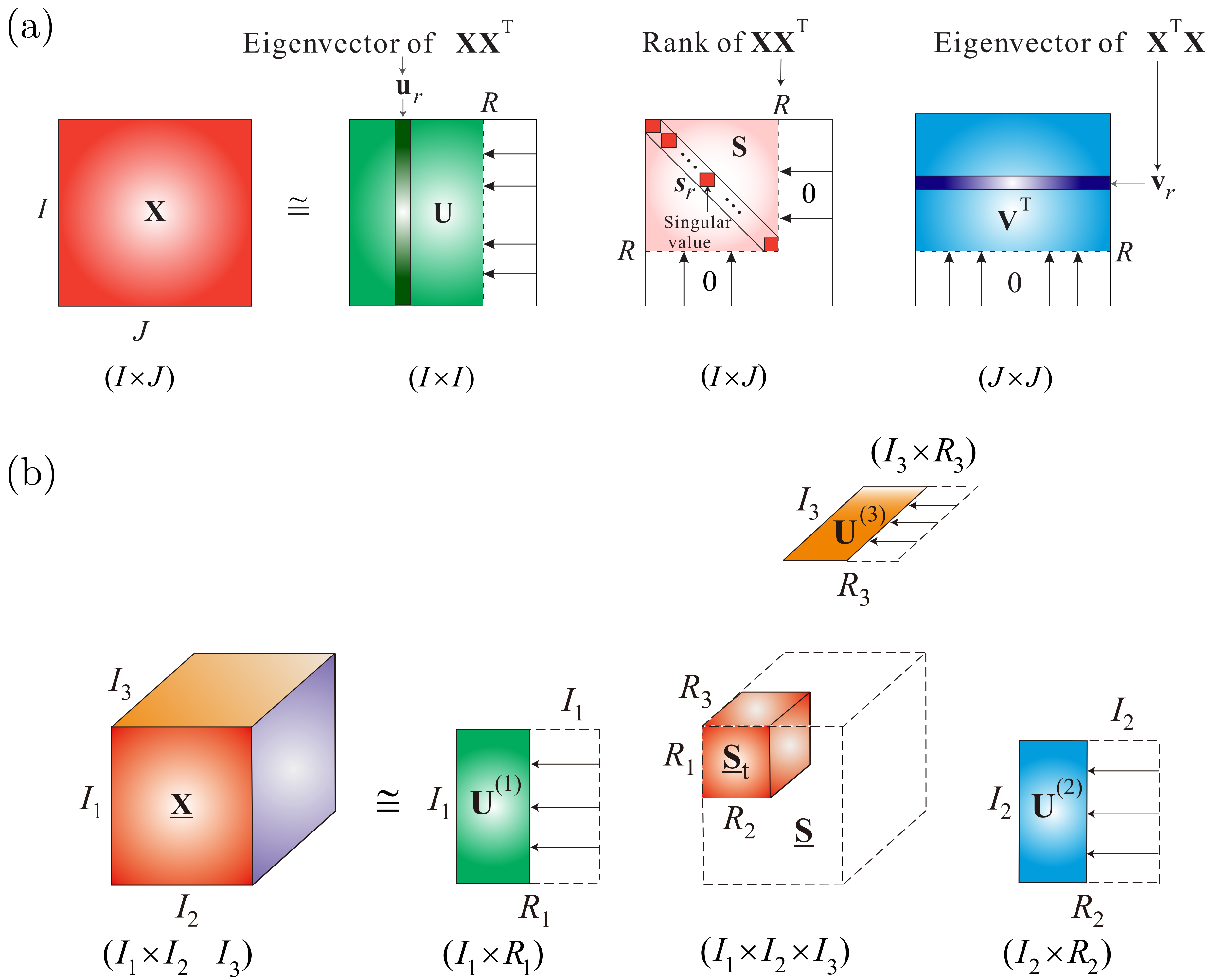
(b) Tucker2

$$y_{itq} = \sum_{j=1}^{J} g_{jtq}\, a_{ij}$$



(c) Tucker1

SVD Higher-order Case



HOSVD

Algorithm 2: Sequentially Truncated HOSVD (Vannieuwenhoven *et al.*, 2012)

Input: Nth-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and approximation accuracy ε

Output: HOSVD in the Tucker format $\hat{\underline{\mathbf{X}}} = [\![\underline{\mathbf{S}}; \mathbf{U}^{(1)}, \ldots, \underline{\mathbf{U}}^{(N)}]\!]$, such that $\|\underline{\mathbf{X}} - \hat{\underline{\mathbf{X}}}\|_F \leqslant \varepsilon$

1: $\underline{\mathbf{S}} \leftarrow \underline{\mathbf{X}}$
2: **for** $n = 1$ to N **do**
3: $\quad [\mathbf{U}^{(n)}, \mathbf{S}, \mathbf{V}] = \texttt{truncated\_svd}(\mathbf{S}_{(n)}, \frac{\varepsilon}{\sqrt{N}})$
4: $\quad \underline{\mathbf{S}} \leftarrow \mathbf{VS}$
5: **end for**
6: $\underline{\mathbf{S}} \leftarrow \texttt{reshape}(\underline{\mathbf{S}}, [R_1, \ldots, R_N])$
7: **return** Core tensor $\underline{\mathbf{S}}$ and orthogonal factor matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$.

rSVD

Algorithm 3: Randomized SVD (rSVD) for large-scale and low-rank matrices with single sketch (Halko *et al.*, 2011)

Input: A matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$, desired or estimated rank R, and oversampling parameter P or overestimated rank $\widetilde{R} = R + P$, exponent of the power method q ($q = 0$ or $q = 1$)

Output: An approximate rank-$\widetilde{R}$ SVD, $\mathbf{X} \cong \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$, i.e., orthogonal matrices $\mathbf{U} \in \mathbb{R}^{I \times \widetilde{R}}$, $\mathbf{V} \in \mathbb{R}^{J \times \widetilde{R}}$ and diagonal matrix $\mathbf{S} \in \mathbb{R}^{\widetilde{R} \times \widetilde{R}}$ with singular values

1: Draw a random Gaussian matrix $\boldsymbol{\Omega} \in \mathbb{R}^{J \times \widetilde{R}}$,
2: Form the sample matrix $\mathbf{Y} = (\mathbf{X}\mathbf{X}^{\mathrm{T}})^q \, \mathbf{X}\boldsymbol{\Omega} \in \mathbb{R}^{I \times \widetilde{R}}$
3: Compute a QR decomposition $\mathbf{Y} = \mathbf{Q}\mathbf{R}$
4: Form the matrix $\mathbf{A} = \mathbf{Q}^{\mathrm{T}}\mathbf{X} \in \mathbb{R}^{\widetilde{R} \times J}$
5: Compute the SVD of the small matrix $\mathbf{A}$ as $\mathbf{A} = \widehat{\mathbf{U}}\mathbf{S}\mathbf{V}^{\mathrm{T}}$
6: Form the matrix $\mathbf{U} = \mathbf{Q}\widehat{\mathbf{U}}$.

HOOI

Algorithm 4: Higher Order Orthogonal Iteration (HOOI) (De Lathauwer *et al.*, 2000b; Austin *et al.*, 2015)

Input: Nth-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ (usually in Tucker/HOSVD format)

Output: Improved Tucker approximation using ALS approach, with orthogonal factor matrices $\mathbf{U}^{(n)}$

```
Initialization via the standard HOSVD (see Algorithm 2)
repeat
    for n = 1 to N do
        Z ← X ×_{p≠n} {U^(p)T}
        C ← Z_(n) Z_(n)^T ∈ R^{R×R}
        U^(n) ← leading R_n eigenvectors of C
    end for
    G ← Z ×_N U^(N)T
until the cost function (‖X‖_F^2 − ‖G‖_F^2) ceases to decrease
return ⟦G; U^(1), U^(2), ..., U^(N)⟧
```

Initialization via the standard HOSVD (see Algorithm 2)

repeat

  for $n = 1$ to N **do**

    $\underline{\mathbf{Z}} \leftarrow \underline{\mathbf{X}} \times_{p \neq n} \{\mathbf{U}^{(p)\,\mathrm{T}}\}$

    $\mathbf{C} \leftarrow \mathbf{Z}_{(n)} \mathbf{Z}_{(n)}^{\mathrm{T}} \in \mathbb{R}^{R \times R}$

    $\mathbf{U}^{(n)} \leftarrow$ leading R_n eigenvectors of $\mathbf{C}$

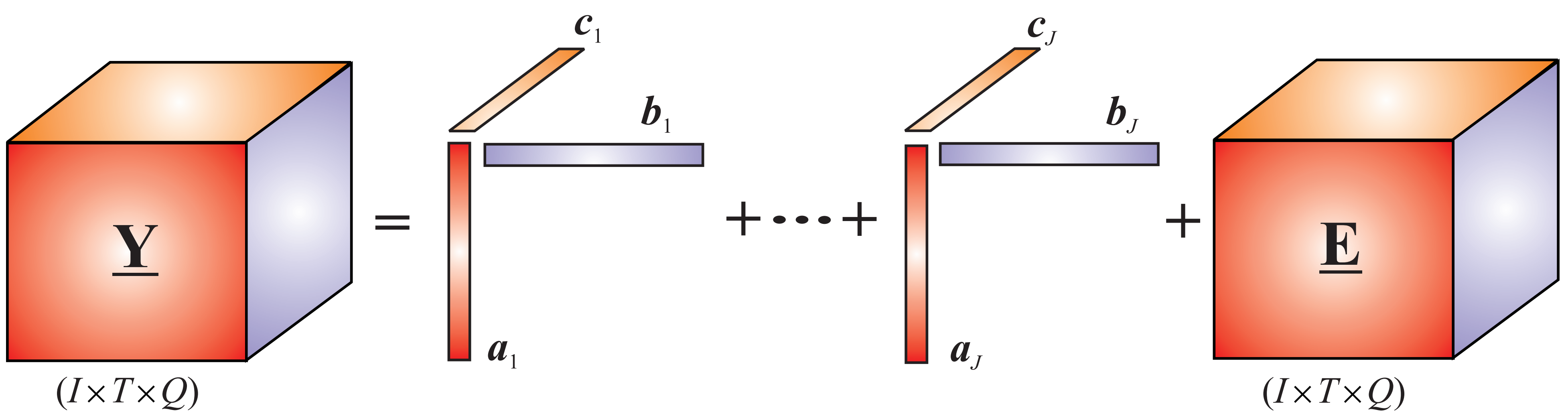
  end for

  $\underline{\mathbf{G}} \leftarrow \underline{\mathbf{Z}} \times_N \mathbf{U}^{(N)\,\mathrm{T}}$

until the cost function $(\|\underline{\mathbf{X}}\|_F^2 - \|\underline{\mathbf{G}}\|_F^2)$ ceases to decrease

return $[\![\underline{\mathbf{G}}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$

Nonnegative Tensor Factorization



Definition (NTF). *Given an N-th order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and a positive integer J, factorize $\underline{\mathbf{Y}}$ into a set of N nonnegative component matrices $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \ldots, \mathbf{a}_J^{(n)}] \in \mathbb{R}^{I_n \times J}$, $(n = 1, 2, \ldots, N)$ representing the common (loading) factors, that is,*

$$\underline{\mathbf{Y}} = \hat{\underline{\mathbf{Y}}} + \underline{\mathbf{E}} = \sum_{j=1}^{J} \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \ldots \circ \mathbf{a}_j^{(N)} + \underline{\mathbf{E}} =$$

$$\underline{\mathbf{I}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} + \underline{\mathbf{E}} = [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] + \underline{\mathbf{E}}$$

with $||\mathbf{a}_j^{(n)}||_2 = 1$ for $n = 1, 2, \ldots N-1$ and $j = 1, 2, \ldots, J$.