Illustration by Chris Brigman

MathSci.ai

Tensor Decomposition meets Reproducing Kernel Hilbert Spaces

Tammy Kolda

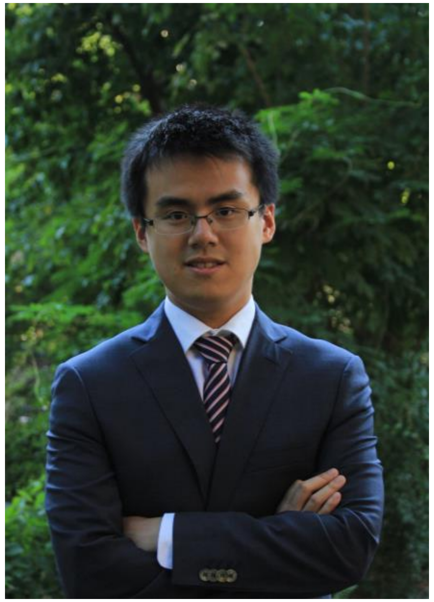
Model Reduction and Surrogate Modeling
La Jolla, CA

September 10, 2024

Collaborators (RKHS work)

Brett Larsen
Databricks Mosaic Research

Alex Williams
Flatiron/NYU



Anru Zhang
Duke

Not pictured: Runshi Tang (Duke)

Goal:
Find Low-Rank Structure in Data

Tensors
- Tensor Decompositions

Quasi-Tensors
- Quasi-Tensor Decomposition
- RKHS

Aligned versus Unaligned Data
- Experimental Results

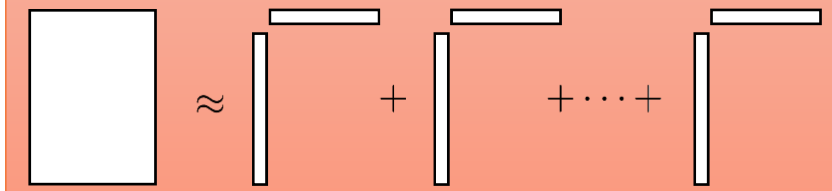
Tensors

And tensor decomposition

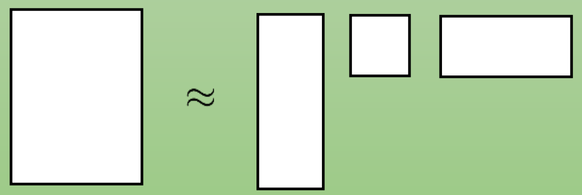
From Matrix to Tensor Decomposition

Singular value decomposition (SVD), nonnegative matrix factorization (NMF), plus connections to Proper Orthogonal Decomposition (POD)

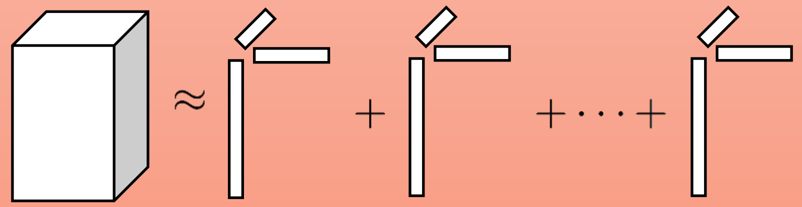
Viewpoint 1: Sum of outer products, useful for interpretation



Viewpoint 2: High-variance subspaces, useful for compression

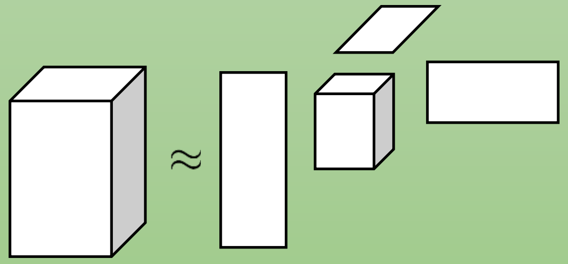


CP Model: Sum of d-way outer products, useful for interpretation



CANDECOMP, PARAFAC, Canonical Polyadic

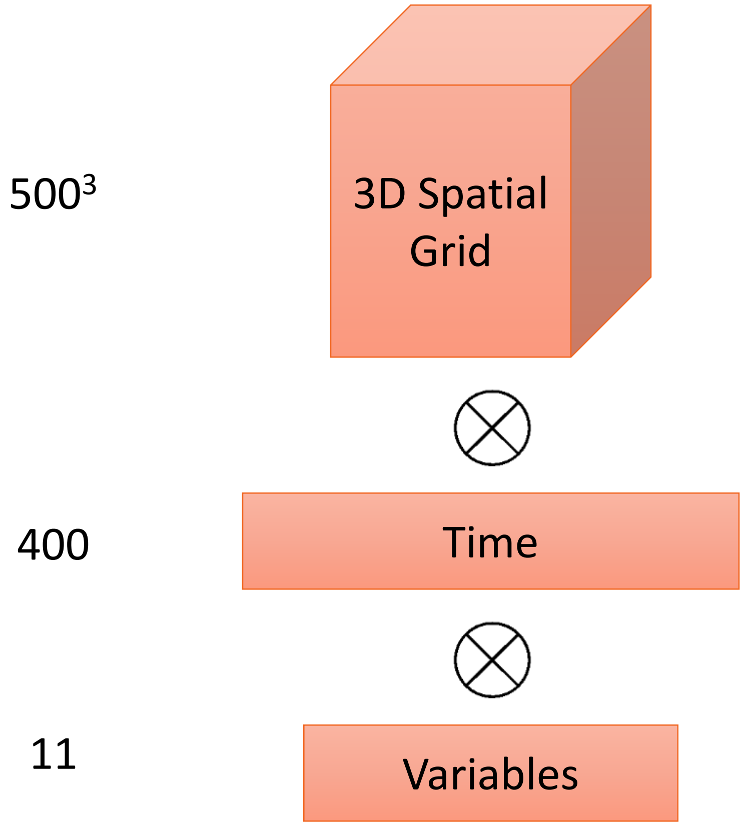
Tucker Model: Project onto high-variance subspaces to reduce dimensionality



HO-SVD, Best Rank-(R_1,R_2,…,R_d) decomposition

Other models for compression include t-SVD, tensor train, etc.

Simulations Produce Tensors!



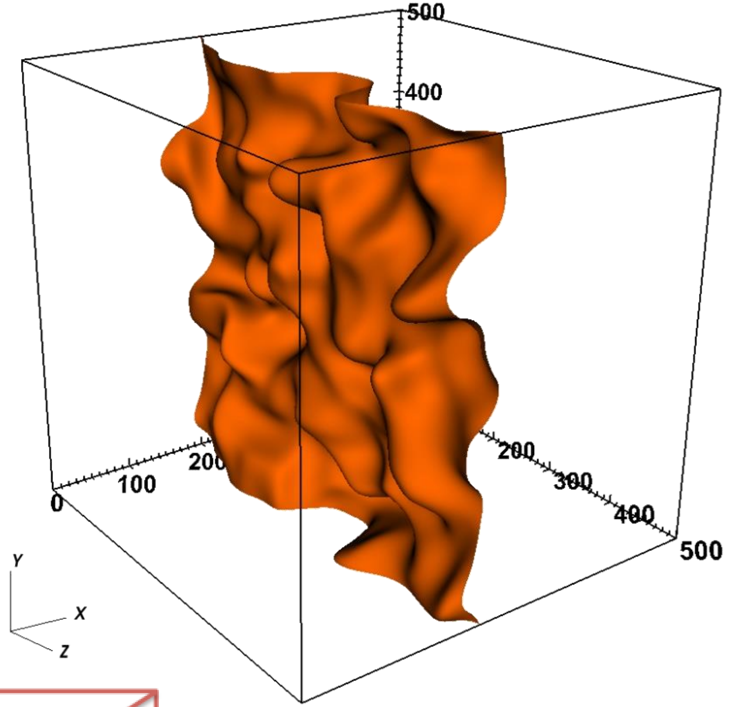
5.5×10^{11} elements

4 TB (double precision)

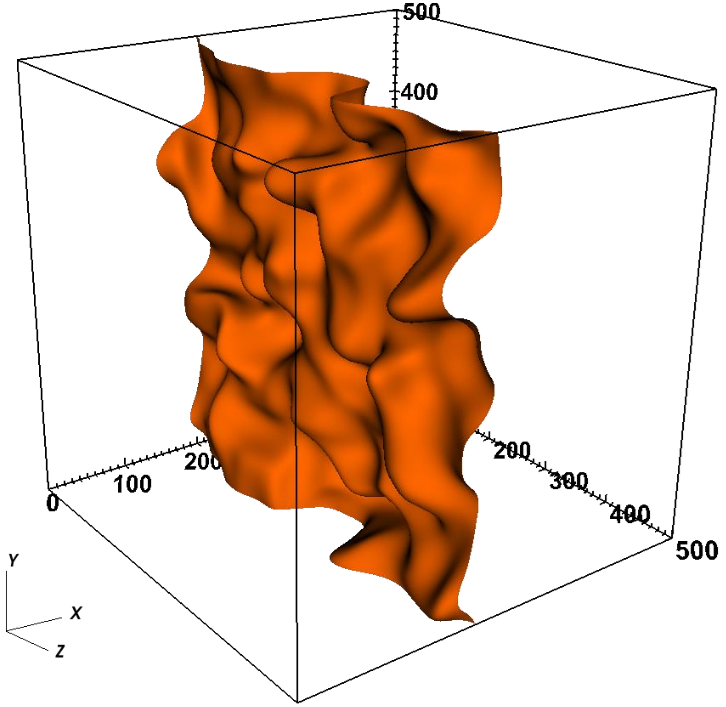
- Fluid flow DNS
  - Single computational experiment produces terabytes of data
  - Storage limits spatial, temporal resolutions
  - Difficult to analyze or transfer data
- Other applications
  - Electron Microscopy Experiments
  - Telemetry Experiments
  - Cosmology Simulations
  - Climate Modeling
- Can be compressed using tensor decompositions

4 TB Combustion Simulation Compression

Original
$500^3 \times 400 \times 11$

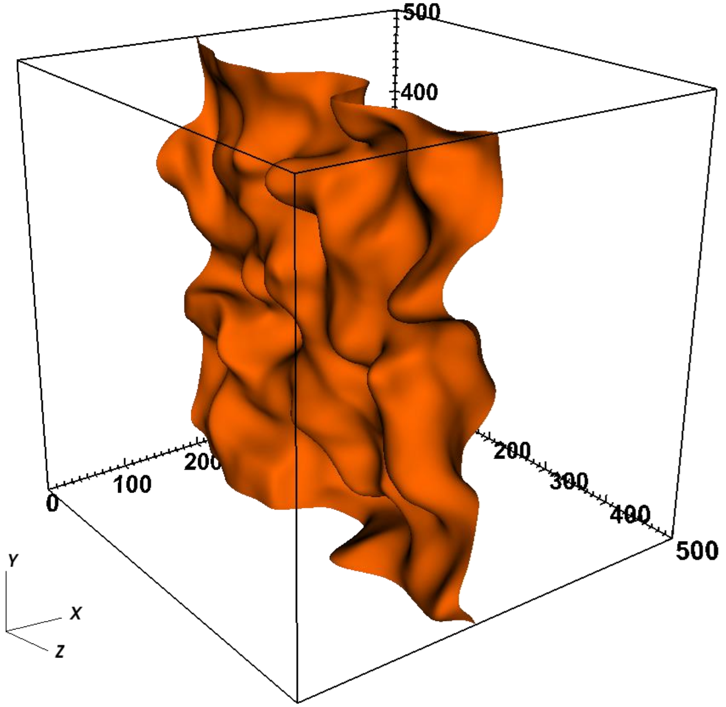


$\epsilon^2 = 10^{-4}$



410X Tucker Compression
13 sec. compute time
on 400-node supercomputer (2019)

$\epsilon^2 = 10^{-2}$

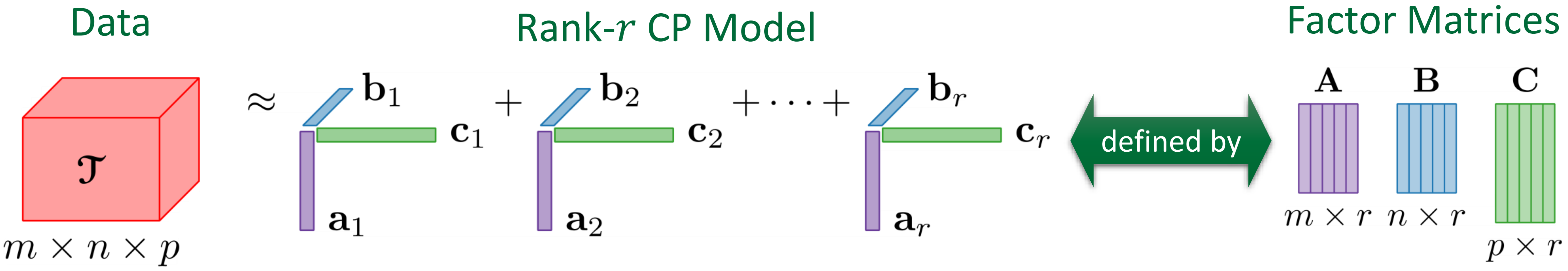


200,000X Tucker Compression
6 sec. compute time
on 400-node supercomputer (2019)

TUCKERMPI

sandialabs/TuckerMPI (github.com)

CP Tensor Decomposition



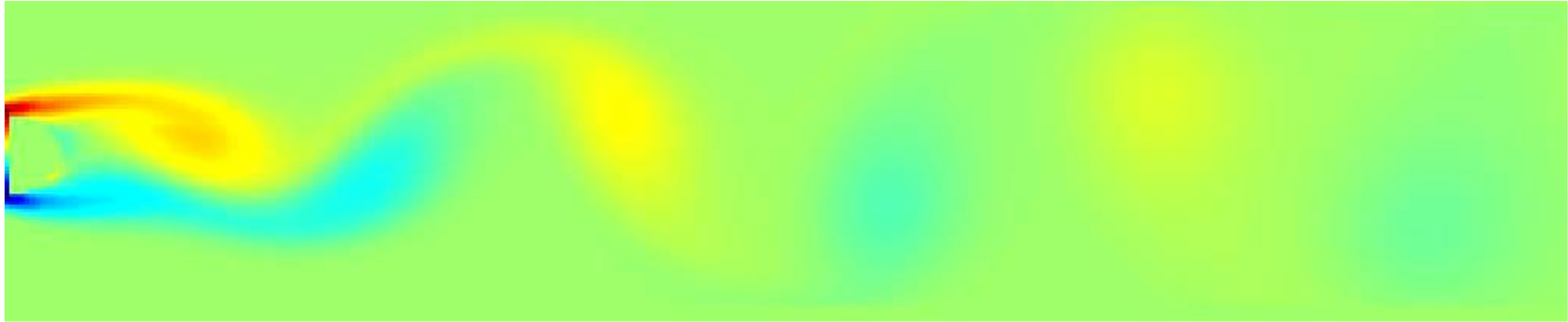
$$\mathfrak{T}(i,j,k) \approx \sum_{\ell=1}^{r} \mathbf{a}_\ell(i)\mathbf{b}_\ell(j)\mathbf{c}_\ell(k)$$

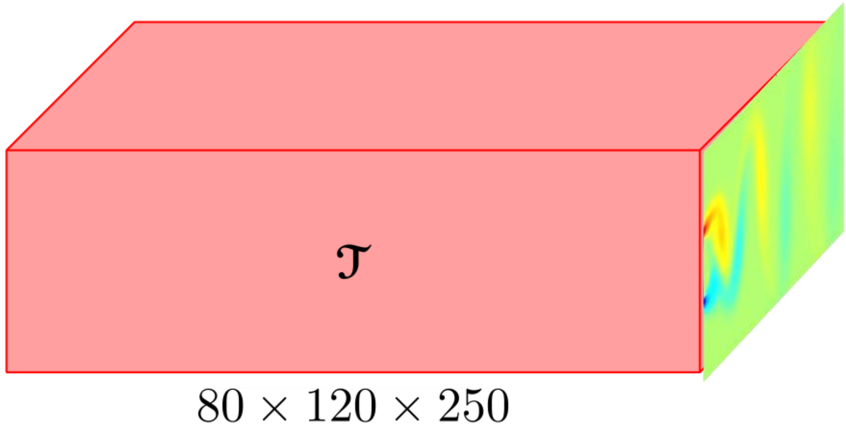
$$\mathfrak{T} \approx [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]$$

Tensor Compressed Fluid Flow Simulation

Bi-dimensional flow past a circular cylinder at Reynolds' number Re = 100

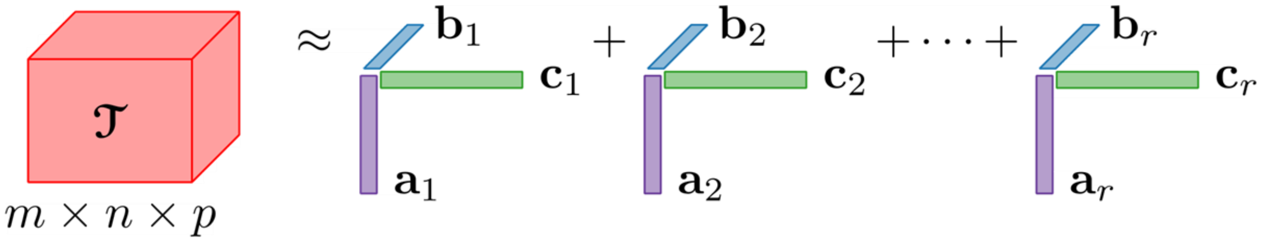
Fluid Flow Data
80 x 120 grid
250 timesteps
Storage: 6×10^6 entries





Alternating Least Squares (CP-ALS)

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \left\| \mathcal{T} - [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!] \right\|^2$$



repeat

$\quad \mathbf{A} \leftarrow \arg\min_{\mathbf{A}} \left\| (\mathbf{C} \odot \mathbf{B})\mathbf{A}^\intercal - \mathbf{T}_{(1)}^\intercal \right\|^2$

$\quad \mathbf{B} \leftarrow \arg\min_{\mathbf{B}} \left\| (\mathbf{C} \odot \mathbf{A})\mathbf{B}^\intercal - \mathbf{T}_{(2)}^\intercal \right\|^2$

$\quad \mathbf{C} \leftarrow \arg\min_{\mathbf{C}} \left\| (\mathbf{B} \odot \mathbf{A})\mathbf{C}^\intercal - \mathbf{T}_{(3)}^\intercal \right\|^2$

until converged

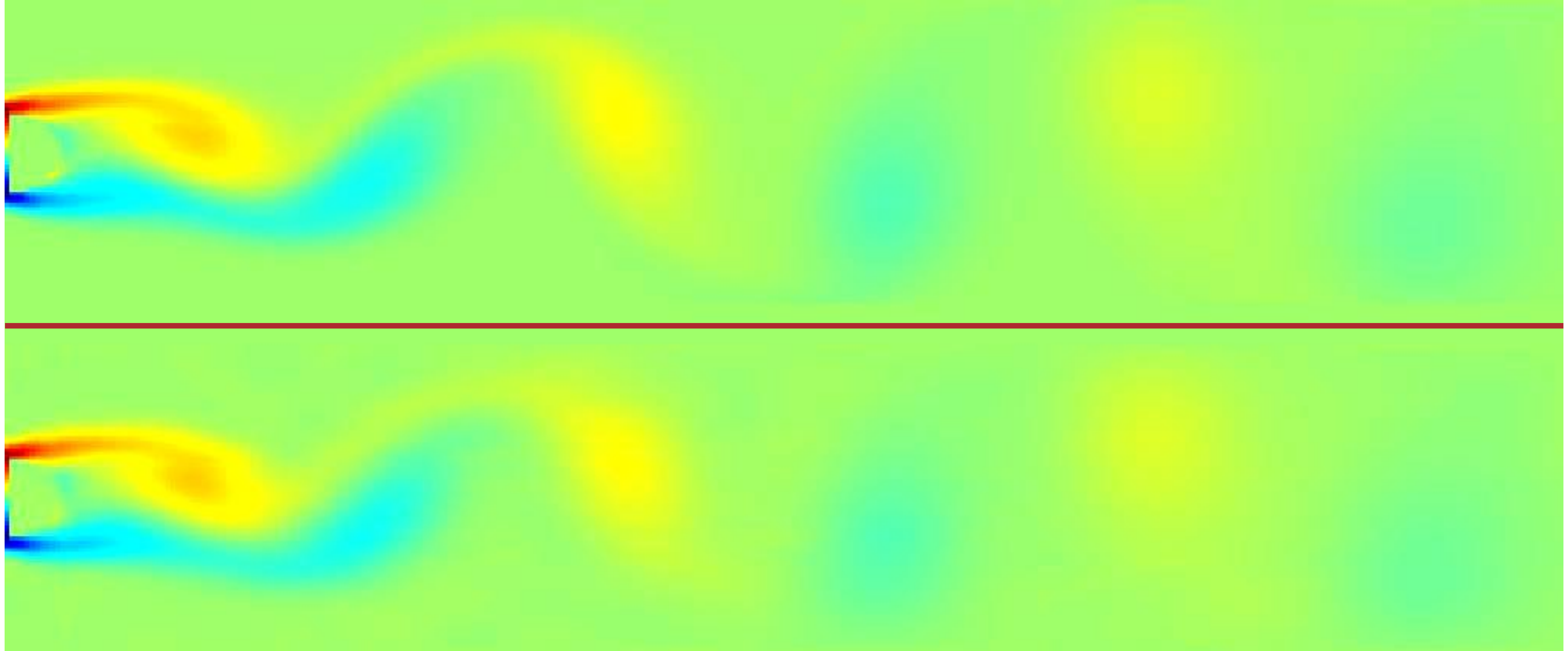
Tensor Compressed Fluid Flow Simulation

Bi-dimensional flow past a circular cylinder at Reynolds' number Re = 100

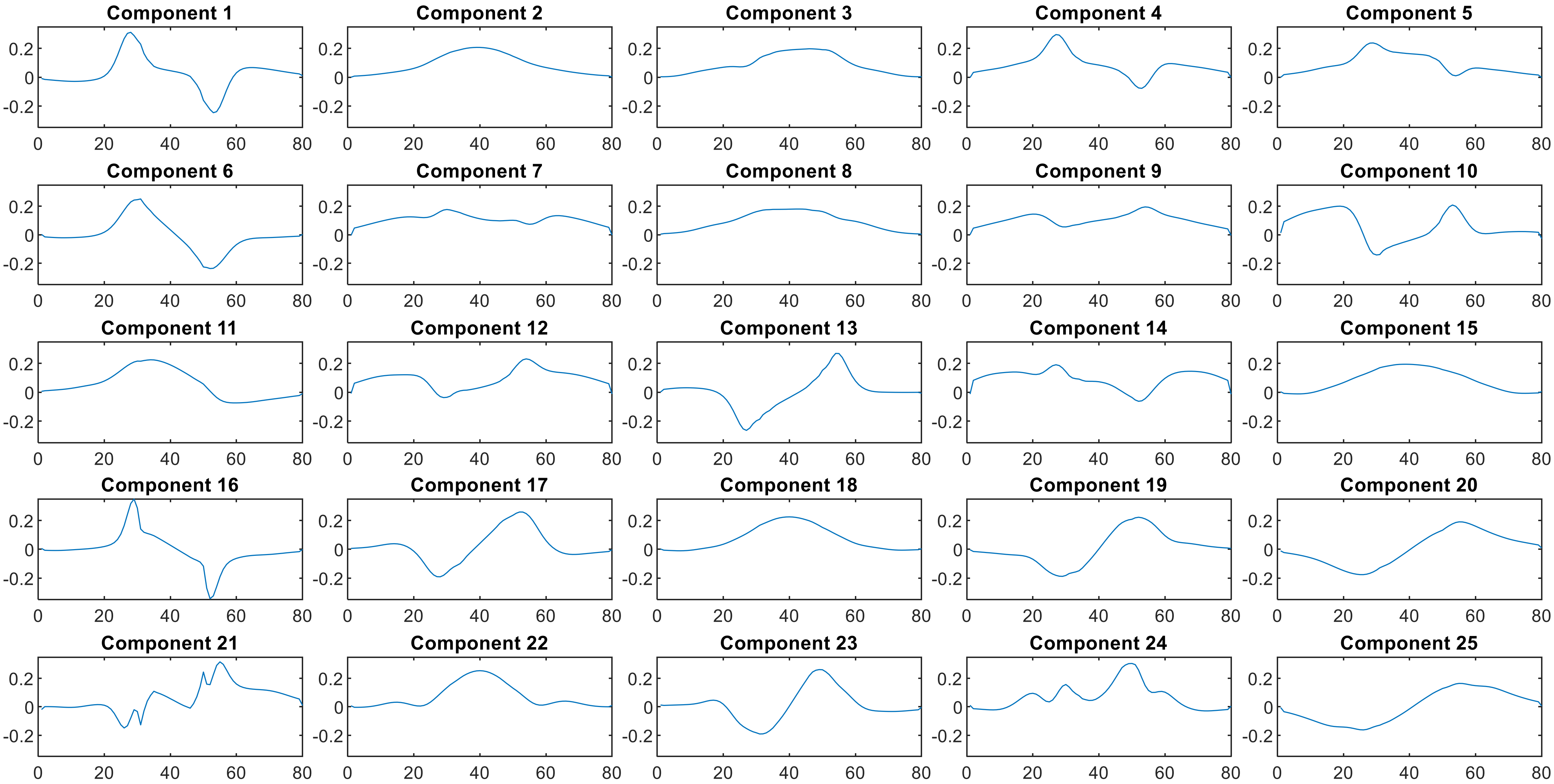
Fluid Flow Data
80 x 120 grid
250 timesteps
Storage: 6×10^6 entries

Tensor Compressed
CP components: 25
Storage: 2×10^4 entries

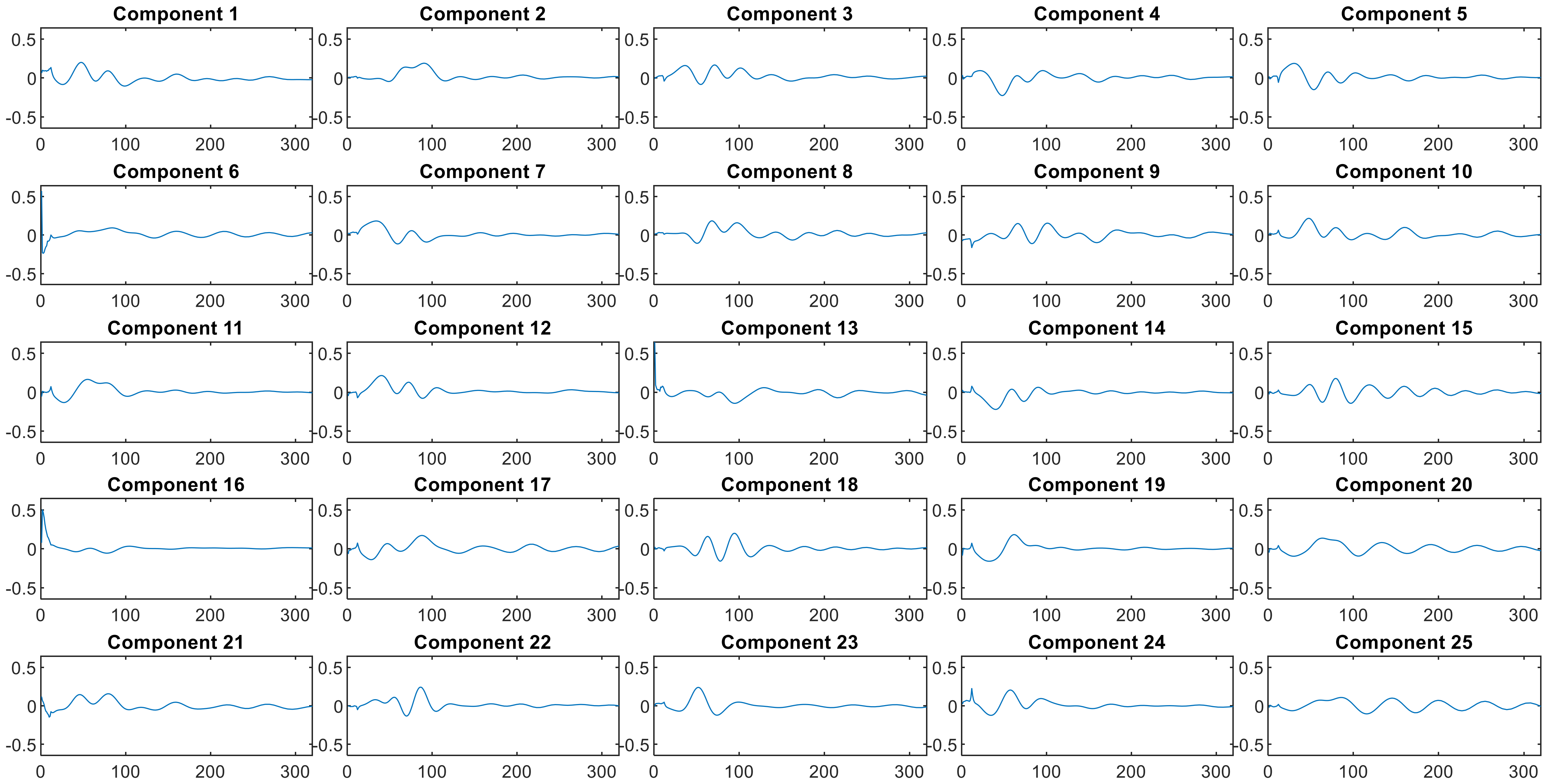
Compression ratio: 400



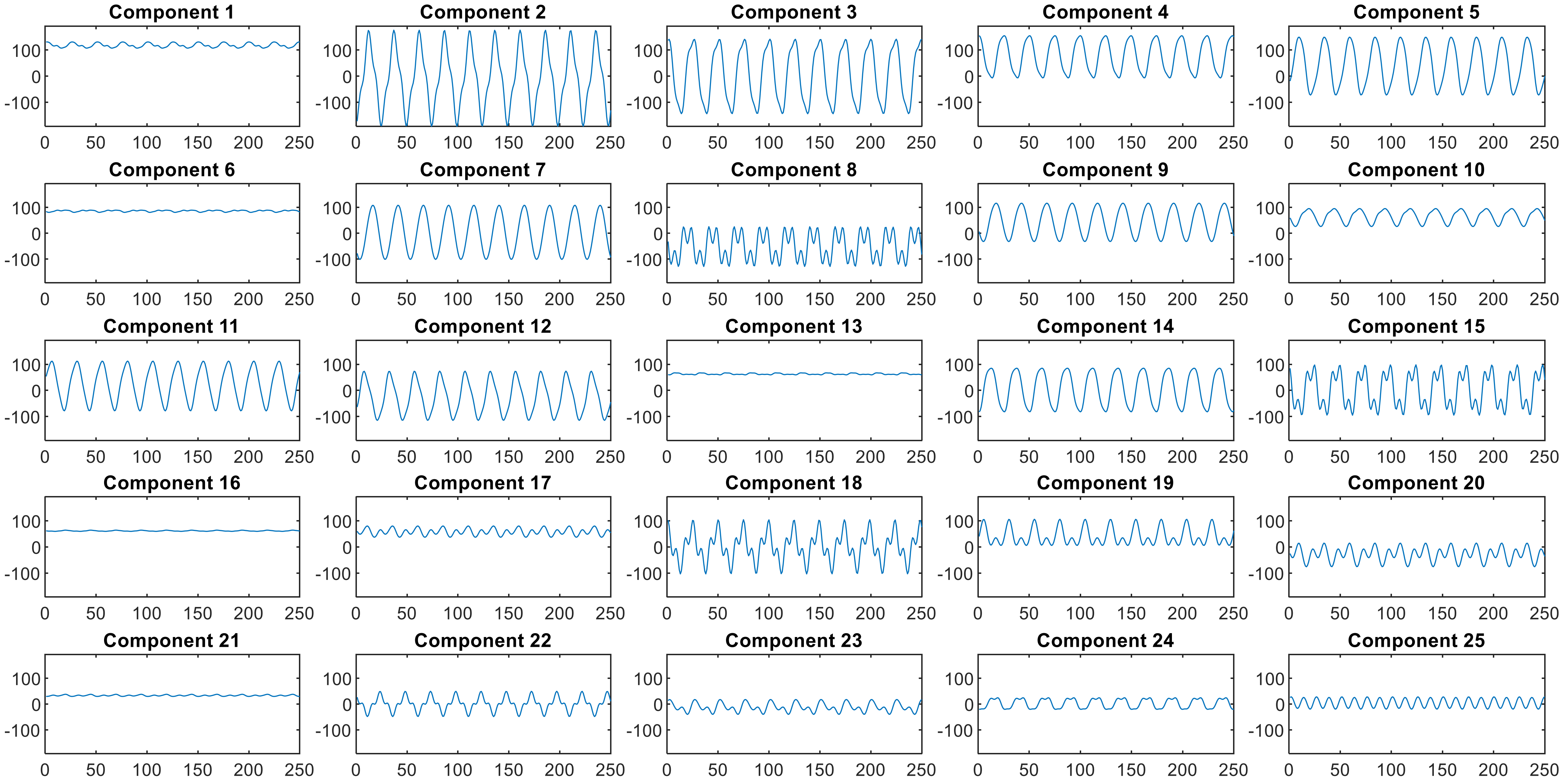
1st Spatial Mode Components



2^{nd} Spatial Mode Components



Temporal Mode Components

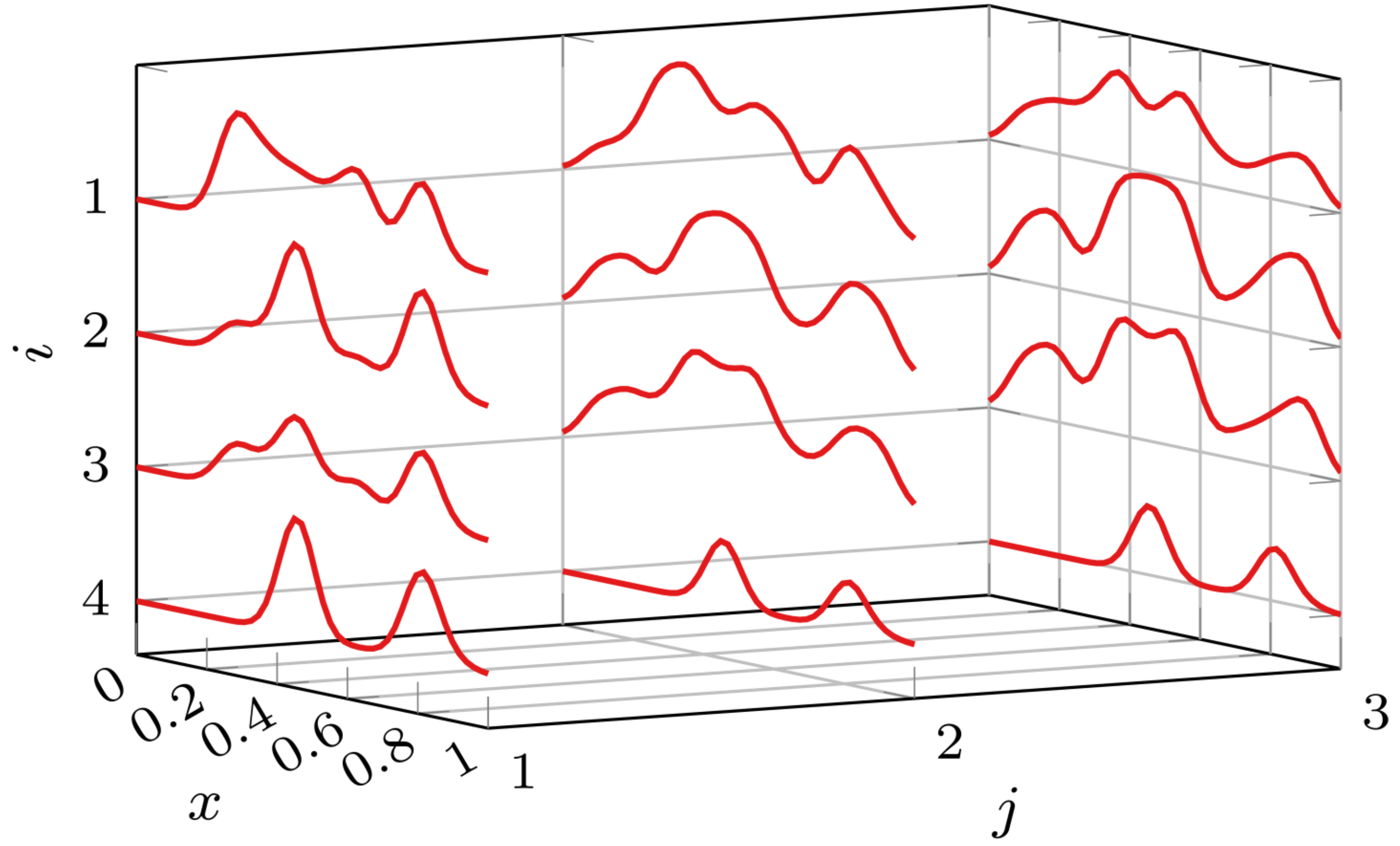


Quasi-Tensors

RKHS and quasi-tensor decomposition

Quasi-Tensor: $m \times n \times \infty$

MathSci.ai

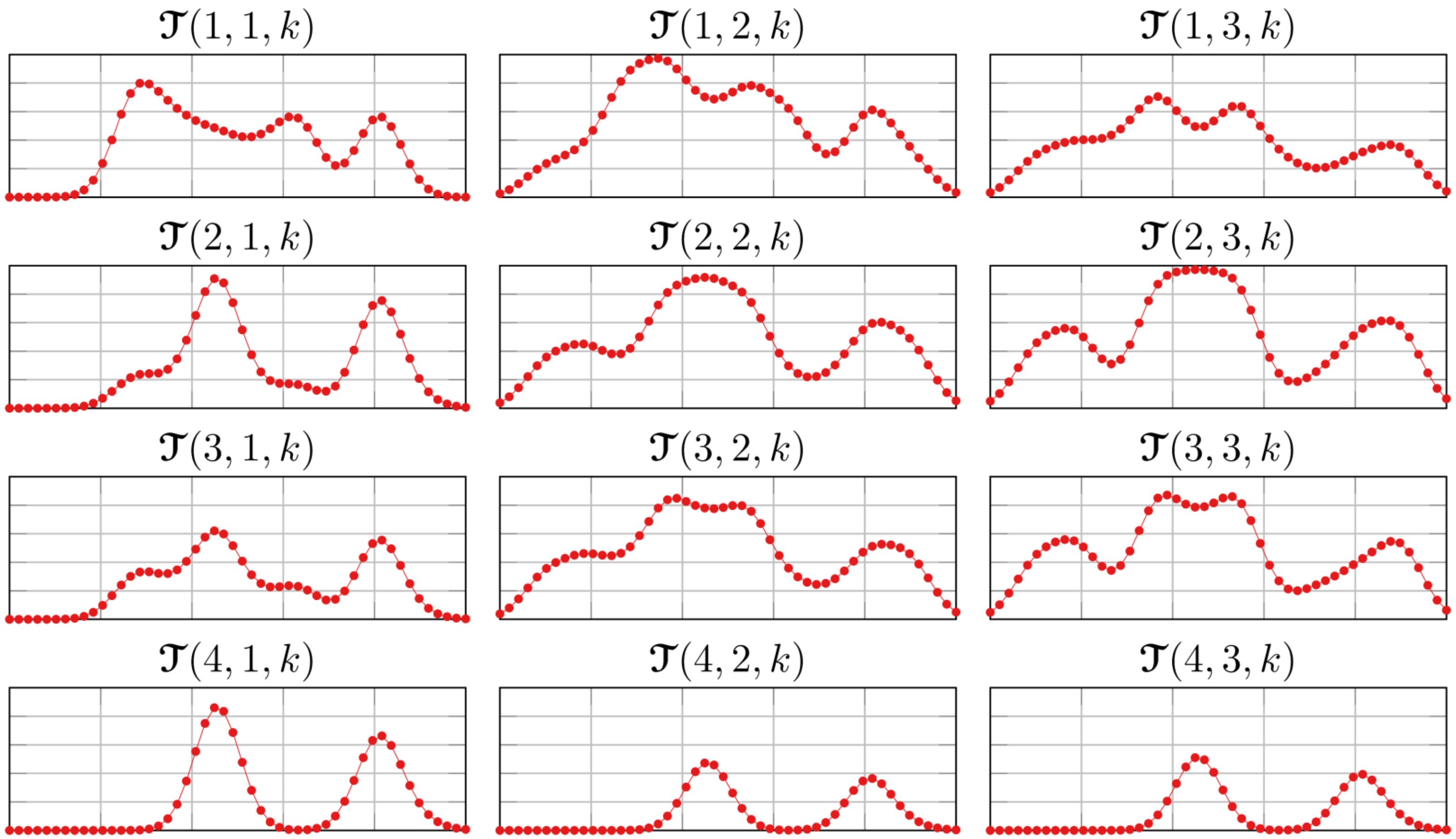


- **Quasi-tensor:** One or more modes of tensor are continuous
- Borrowed from **quasi-matrix:** $n \times \infty$ array of functions (Townsend & Trefethen, 2014)
- Can be converted to a tensor by evaluating the functions at finitely many observations (generally only have this)
- Dynamics/smoothness can be important and may want to preserve it in some fashion

Quasi-Tensor → Tensor

In practice, can only observe real-world quasi-tensor at discrete vales of x

MathSci.ai



$$\boldsymbol{T} \in \mathbb{R}^{4\times 3\times \infty}$$

$$\boldsymbol{T}(i,j,x)$$

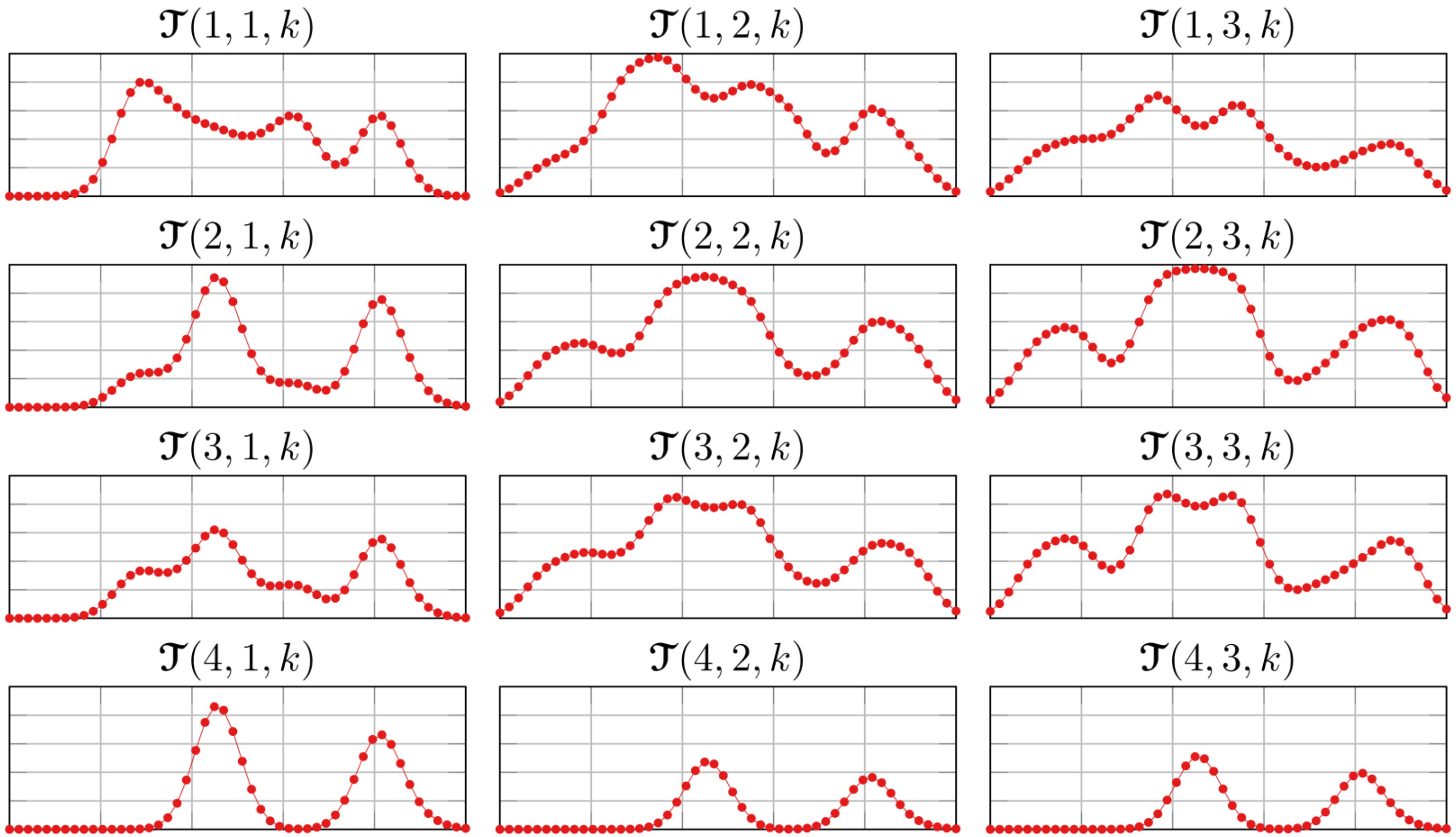
$$\mathbf{v} = [0, 0.02, \ldots, 1]^\intercal \in \mathbb{R}^{50}$$

$$\bar{\mathfrak{T}} = \boldsymbol{T}(\mathbf{v})$$

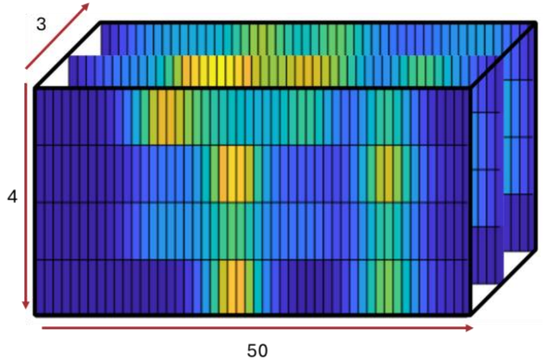
$$\bar{\mathfrak{T}} \in \mathbb{R}^{4\times 3\times 50}$$

$$\bar{\mathfrak{T}}(i,j,k) = \boldsymbol{T}(i,j,v_k)$$

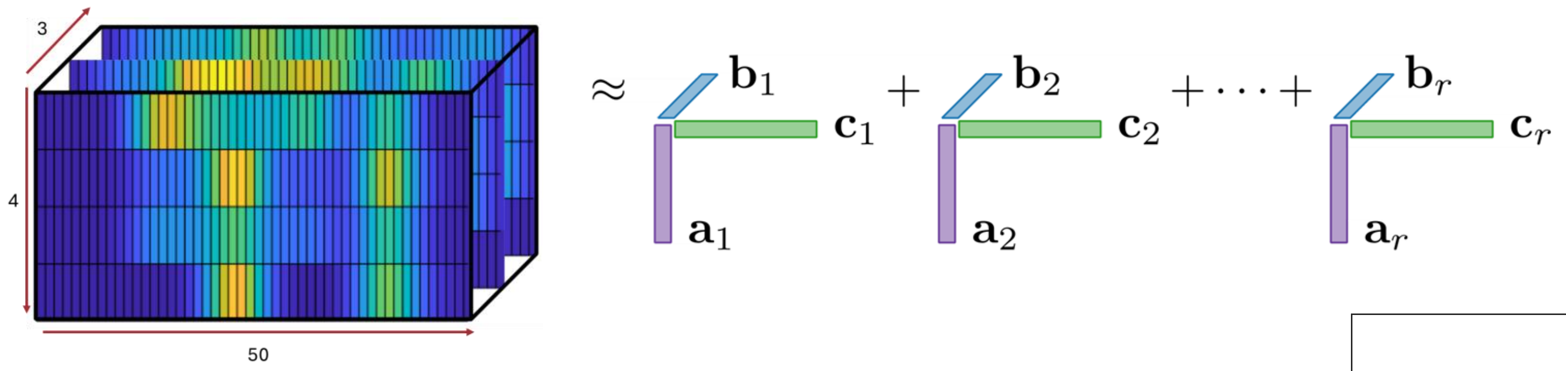
Synthetic Data Example



$$\mathfrak{T} \in \mathbb{R}^{4\times3\times50}$$

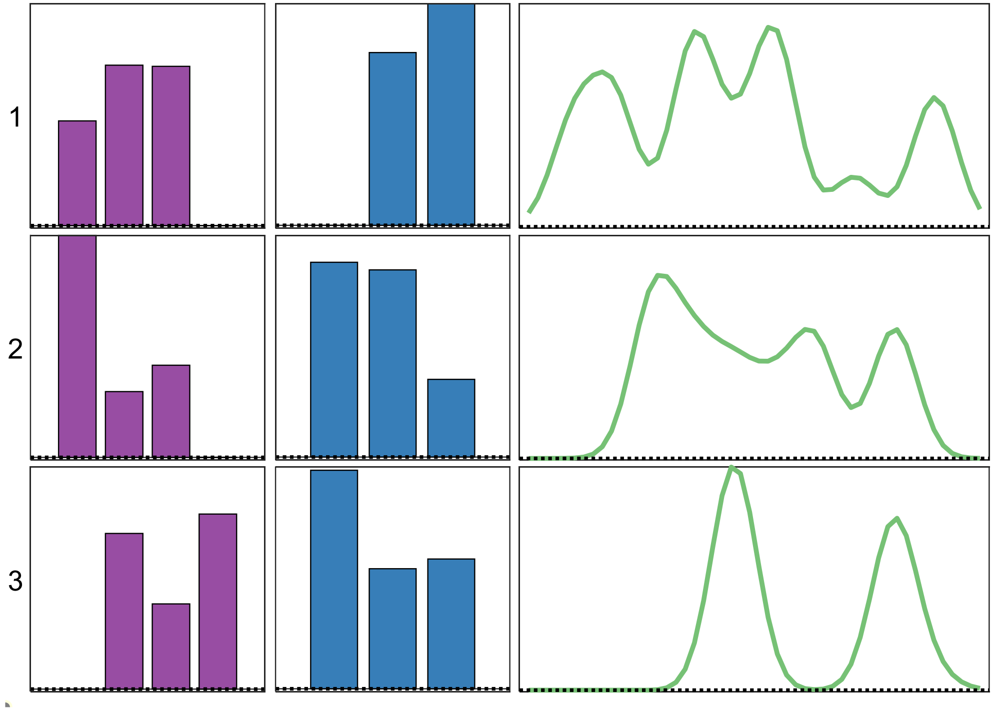


Rank-3 Factorization by CP-ALS



$$\bar{\mathfrak{T}} \in \mathbb{R}^{4\times3\times50}$$

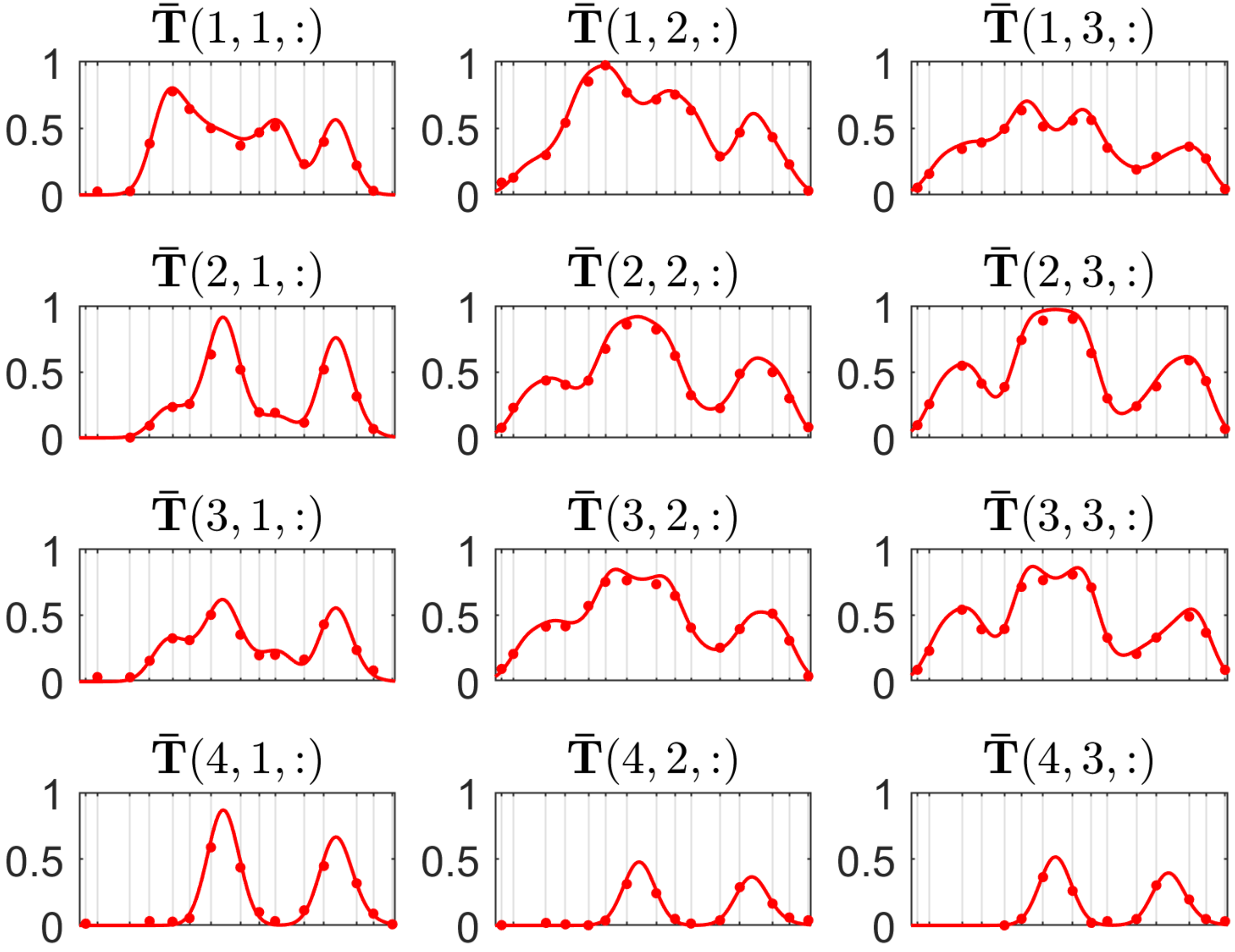
CP-ALS (nonnegative)



If sampling frequency is high enough in continuous mode, CP will generally pick up the smoothness without help!

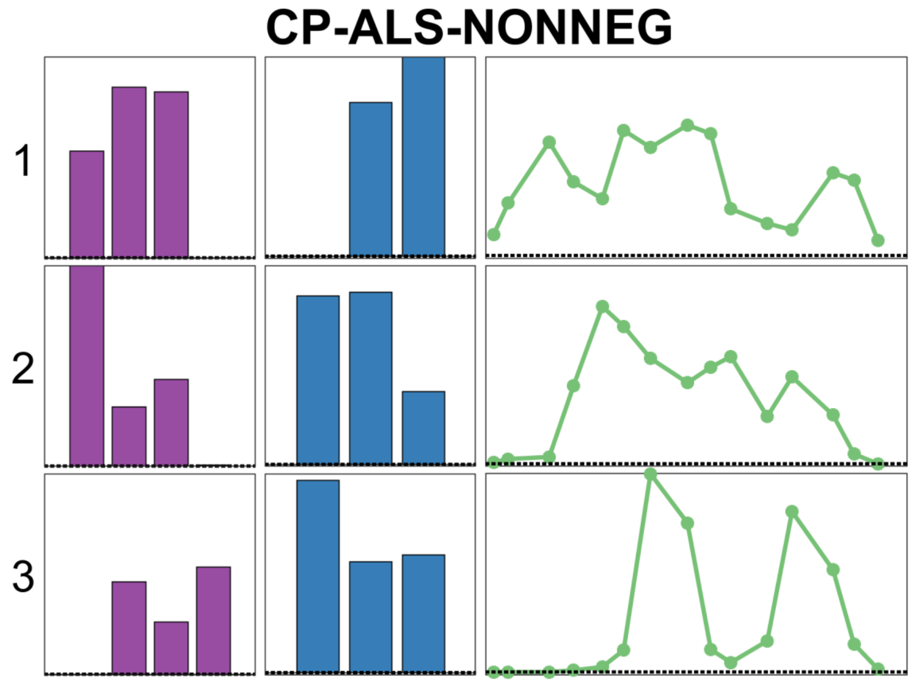
But what if the frequency is lower?

What happens with fewer observations?

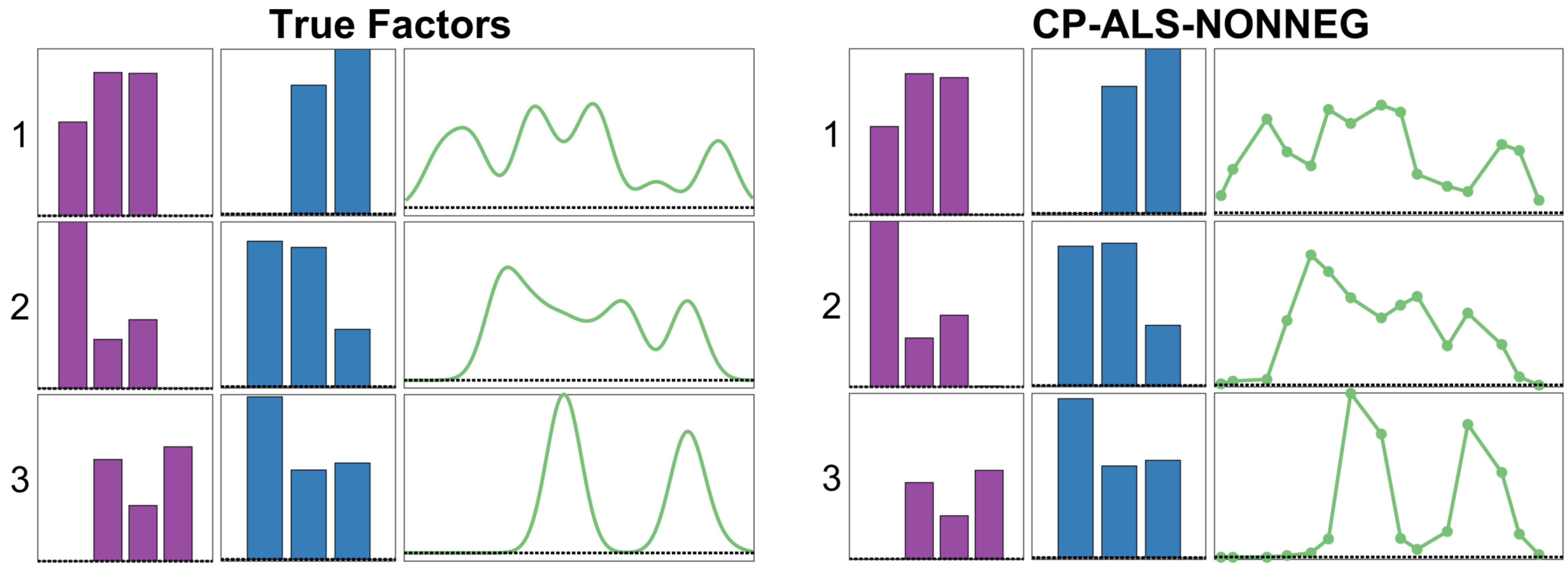


$\bar{\mathfrak{T}} \in \mathbb{R}^{4\times3\times15}$

15 aligned, unevenly-spaced noisy samples per function



Few Observations ⇒ Jagged Components



Handing Smooth Modes

CP Tensor Decomposition with Hybrid Infinite and Finite Dimensional Modes

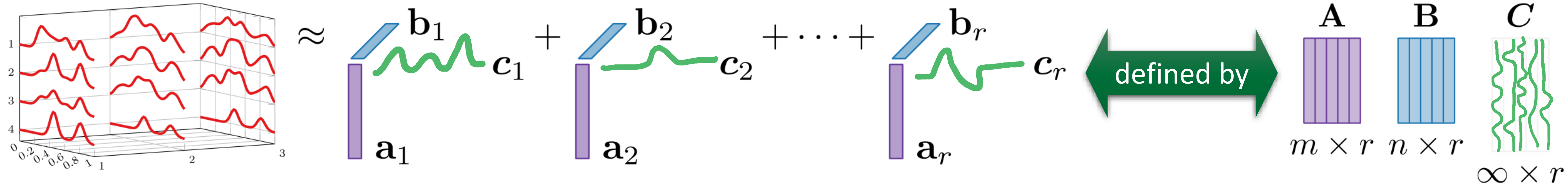
CP Hybrid Infinite & Finite (HIFI) Decomposition

MathSci.ai

Quasi-Tensor

Rank-r CP Model

Factor Matrices



$m \times n \times \infty$

$$\boldsymbol{T}(i,j,x) \approx \sum_{\ell=1}^{r} \mathbf{a}_\ell(i)\mathbf{b}_\ell(j)\boldsymbol{c}_\ell(x)$$

$$\boldsymbol{T} \approx [\![\mathbf{A}, \mathbf{B}, \boldsymbol{C}]\!]$$

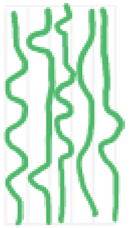
Factors in infinite-dimensional modes are smooth functions!

Quasi-Matrix

MathSci.ai

$$\boldsymbol{C} \in \mathbb{R}^{\infty \times r}$$

$\boldsymbol{C}$



$\infty \times r$

Collection of r functions:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_r \end{bmatrix}$$

Evaluation at a vector to create matrix:

$$\mathbf{v} \in \mathbb{R}^p$$

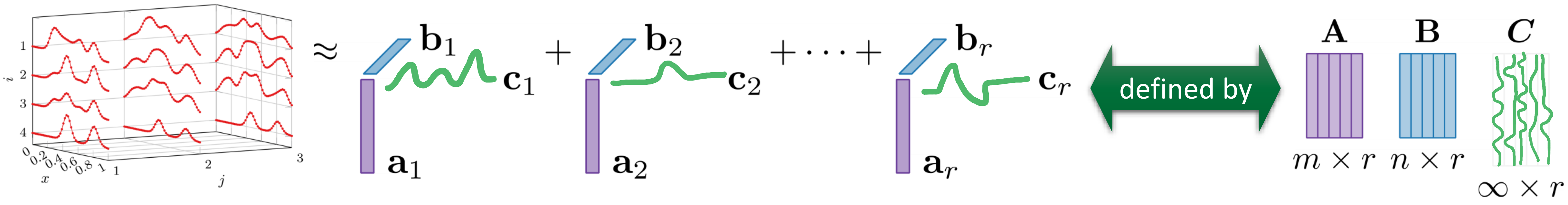
$$\boldsymbol{C}(\mathbf{v}) = \begin{bmatrix} \boldsymbol{c}_1(v_1) & \boldsymbol{c}_2(v_1) & \cdots & \boldsymbol{c}_r(v_1) \\ \boldsymbol{c}_1(v_2) & \boldsymbol{c}_2(v_2) & \cdots & \boldsymbol{c}_r(v_2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{c}_1(v_p) & \boldsymbol{c}_2(v_p) & \cdots & \boldsymbol{c}_r(v_p) \end{bmatrix} \in \mathbb{R}^{p \times r}$$

Fitting CP-HIFI Decomposition

Data

Rank-r CP Model

Factor Matrices



$$\boldsymbol{T}(i,j,x) \approx \sum_{\ell=1}^{r} \mathbf{a}_\ell(i)\mathbf{b}_\ell(j)\boldsymbol{c}_\ell(x)$$

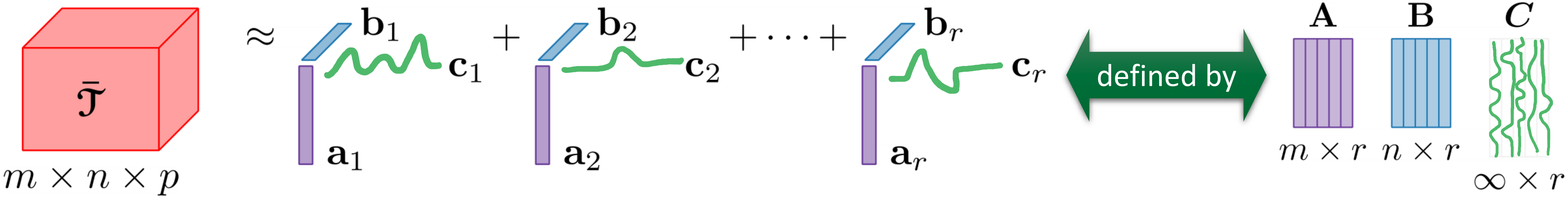
$$\boldsymbol{T} \approx [\![\mathbf{A}, \mathbf{B}, \boldsymbol{C}]\!]$$

Finite-Dimensional Data

Data

Rank-r CP Model

Factor Matrices



grid sampled
at design points

$$\mathbf{v} = [v_1, v_2, \cdots, v_p]^T$$

$$\bar{\mathfrak{T}} = \boldsymbol{T}(\mathbf{v})$$

$$\bar{\mathfrak{T}}(i,j,k) = \boldsymbol{T}(i,j,v_k)$$

quasi-tensor — function

$$\boldsymbol{T}(i,j,v_k) \approx \sum_{\ell=1}^{r} \mathbf{a}_\ell(i)\mathbf{b}_\ell(j)\boldsymbol{c}_\ell(v_k)$$

tensor — vector

$$\bar{\mathfrak{T}}(i,j,k) \approx \sum_{\ell=1}^{r} \mathbf{a}_\ell(i)\mathbf{b}_\ell(j)\mathbf{c}_\ell(k)$$

quasi-matrix

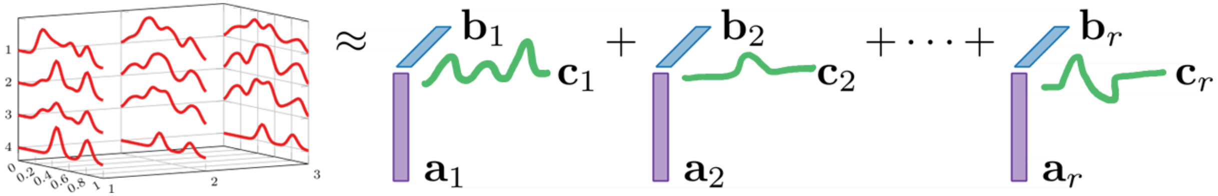
$$\boldsymbol{T}(\mathbf{v}) \approx [\![\mathbf{A}, \mathbf{B}, \boldsymbol{C}(\mathbf{v})]\!]$$

matrix

$$\bar{\mathfrak{T}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \text{ with } \mathbf{C} = \boldsymbol{C}(\mathbf{v})$$

Options for Imposing Smoothness

MathSci.ai



- $\mathbf{C}$ is a matrix, but enforce smoothness
  - **Linear Combination of Splines:** $\mathbf{C} = \boldsymbol{SW}$ where $\boldsymbol{S} \in \mathbb{R}^{p \times q}$ is matrix of smooth b-splines and $\mathbf{W} \in \mathbb{R}^{q \times r}$ is weight matrix (Timmerman & Kiers, 2002)
  - **Second-difference Operator Regularization:** $\|\mathbf{L}_2\mathbf{C}\|^2$ where $\mathbf{L}_2$ is second-difference operator of size $(p-1) \times p$ (Martinez-Montes, Sanchez-Bornot, Valdes-Sosa, 2008); aka Whittaker smoothing
- $\boldsymbol{C}$ is a quasi-matrix
  - **Gaussian Process Factor Analysis** (GPFA, matrix factorization): Draws factors from an associated Gaussian process
  - **Functional Principal Component Analysis** (FPCA, matrix factorization): Uses Karhunen–Loeve Decomposition (i.e., continuous version of SVD)
  - **Chebfun**: linear combination of Chebyshev polynomials (matrix decomposition: Townsend & Trefethen, 2013 & 2015; Tucker decomposition: Hashemi & Trefethen, 2017)
  - **Reproducing Kernel Hilbert Space** (RKHS): Used in Tucker tensor decomposition (Han, Shi, Zhang, 2023)

Reproducing Kernel Hilbert Space (RKHS)

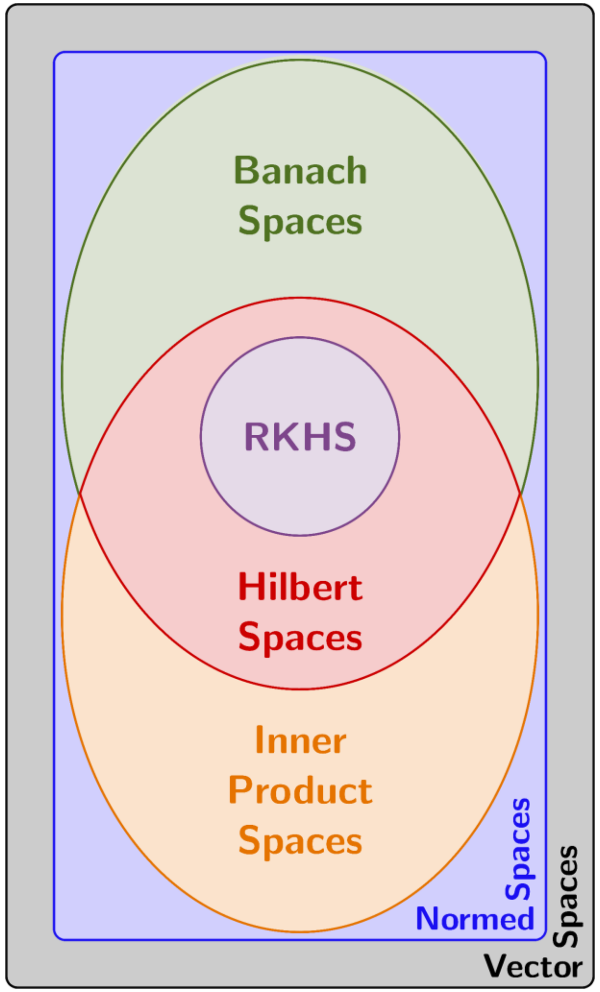
- Want to find smooth functions <u>but</u> impractical to optimize over all functions

- Restrict to an infinite-dimensional **Reproducing Kernel Hilbert Space** or **RKHS** denoted as $\mathcal{H}_K$ with kernel

$$\boldsymbol{K}(\cdot,\cdot):\mathbb{R}\times\mathbb{R}\rightarrow\mathbb{R}$$

- Kernel is positive semidefinite (PSD) which means that for any $\mathbf{v}=[v_1, v_2, \dots, v_p]^T$, the matrix $\mathbf{K}\equiv\boldsymbol{K}(\mathbf{v},\mathbf{v})$ is positive semidefinite

$$\mathbf{K}(i,j)=\boldsymbol{K}(v_i,v_j)$$



Kernel Quasi-Matrix and Matrix

Given a kernel: $\boldsymbol{K}(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$

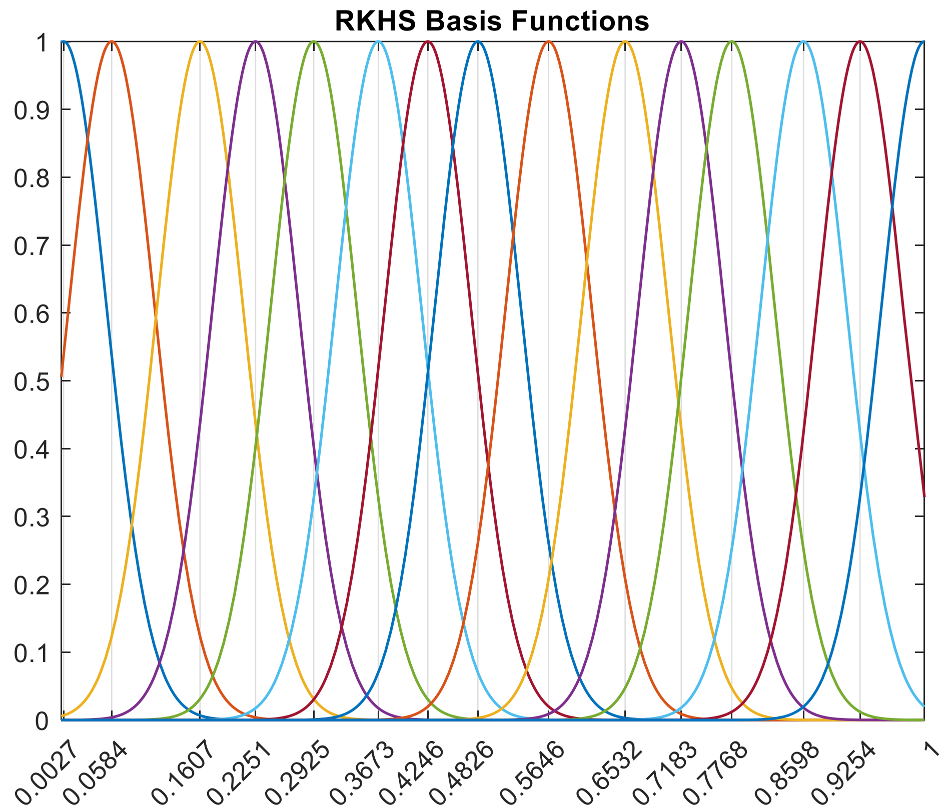
Design points: $\mathbf{v} \in \mathbb{R}^p$

Kernel Quasi-matrix: $\hat{\boldsymbol{K}}(\cdot) = \boldsymbol{K}(\cdot, \mathbf{v}) \in \mathbb{R}^{\infty \times p}$

$$\hat{\boldsymbol{K}}(x) = \begin{bmatrix} \boldsymbol{K}(x,v_1) & \boldsymbol{K}(x,v_2) & \cdots & \boldsymbol{K}(x,v_p) \end{bmatrix} \in \mathbb{R}^{1 \times p}$$

Kernel Matrix: $\mathbf{K} = \hat{\boldsymbol{K}}(\mathbf{v}) = \boldsymbol{K}(\mathbf{v}, \mathbf{v}) \in \mathbb{R}^{p \times p}$

$$\mathbf{K} = \begin{bmatrix} \boldsymbol{K}(v_1,v_1) & \boldsymbol{K}(v_1,v_2) & \cdots & \boldsymbol{K}(v_1,v_p) \\ \boldsymbol{K}(v_2,v_1) & \boldsymbol{K}(v_2,v_2) & \cdots & \boldsymbol{K}(v_2,v_p) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{K}(v_p,v_1) & \boldsymbol{K}(v_p,v_2) & \cdots & \boldsymbol{K}(v_p,v_p) \end{bmatrix} \in \mathbb{R}^{p \times p}$$



Representer Theorem Translates RKHS Problem to Finite-Dimensional Space

MathSci.ai

For a set of p observations of the form $\{v_i, y_i \equiv f(v_i)\}$ and $\lambda > 0$, we can consider the following regularized regression problem:

$$\min_{\hat{\boldsymbol{f}} \in \mathcal{H}_{\boldsymbol{K}}} \underbrace{\sum_{i=1}^{p} \left(y_i - \hat{\boldsymbol{f}}(v_i)\right)^2}_{\text{empirical risk}} + \underbrace{\lambda \|\hat{\boldsymbol{f}}\|_{\mathcal{H}_{\boldsymbol{K}}}}_{\text{regularization}}$$

The ***representer theorem*** tells us that the optimal solution in the *infinite-dimensional Hilbert space* has the following *finite* form:

$$\hat{\boldsymbol{f}}(\cdot) = \sum_{j=1}^{p} w_j \boldsymbol{K}(\cdot, v_j)$$

$$\hat{\boldsymbol{f}} = \hat{\boldsymbol{K}} \mathbf{w}$$

RKHS Problem in Practice

MathSci.ai

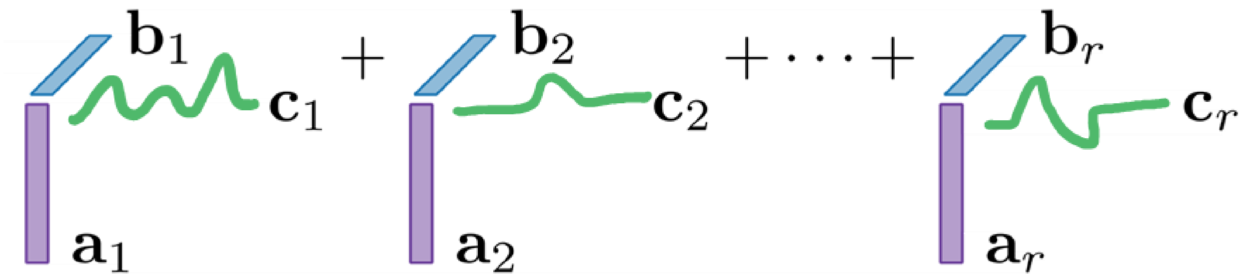
- Given p observations $\{v_i, y_i \equiv f(v_i)\}$
- Choose p.s.d. kernel $\boldsymbol{K}$ and regularization parameter $\lambda > 0$
- Compute $\mathbf{K}(i,j) \equiv \boldsymbol{K}(v_i, v_j)$
- Solve the following problem for $\mathbf{w} \in \mathbb{R}^p$:

$$\min_{\mathbf{w}\in\mathbb{R}^p} \underbrace{\|\mathbf{K}\mathbf{w} - \mathbf{y}\|^2}_{\text{empirical risk}} + \underbrace{\lambda \mathbf{w}^\intercal \mathbf{K}\mathbf{w}}_{\text{regularization}}$$

- Final solution is

$$\hat{f} = \hat{\boldsymbol{K}}\mathbf{w} \quad \text{with} \quad \mathbf{w} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}$$

Representer Theorem Enables Us to Optimize in Finite-Dimensional Space

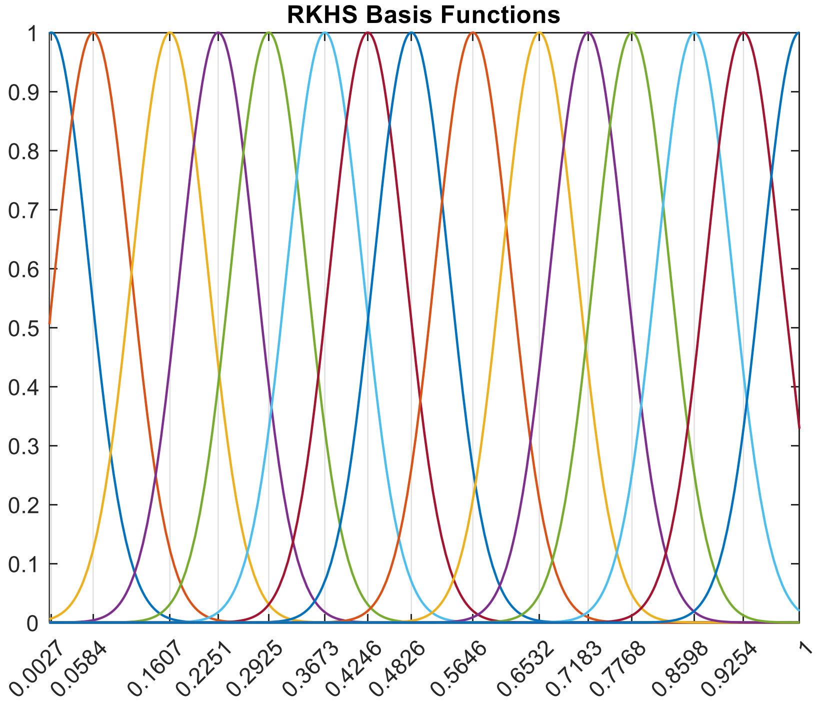


$$\boldsymbol{c}_1 = \hat{\boldsymbol{K}}\mathbf{w}_1 \qquad \boldsymbol{c}_2 = \hat{\boldsymbol{K}}\mathbf{w}_2 \qquad \cdots \qquad \boldsymbol{c}_r = \hat{\boldsymbol{K}}\mathbf{w}_r$$

$$\boldsymbol{C} = \hat{\boldsymbol{K}}\mathbf{W} \qquad \mathbf{W} \in \mathbb{R}^{p \times r}$$

$$\boldsymbol{T} \approx [\![\mathbf{A}, \mathbf{B}, \hat{\boldsymbol{K}}\mathbf{W}]\!]$$

$$\hat{\boldsymbol{K}}(\cdot) = \boldsymbol{K}(\cdot, \mathbf{v}) \in \mathbb{R}^{\infty \times p}$$

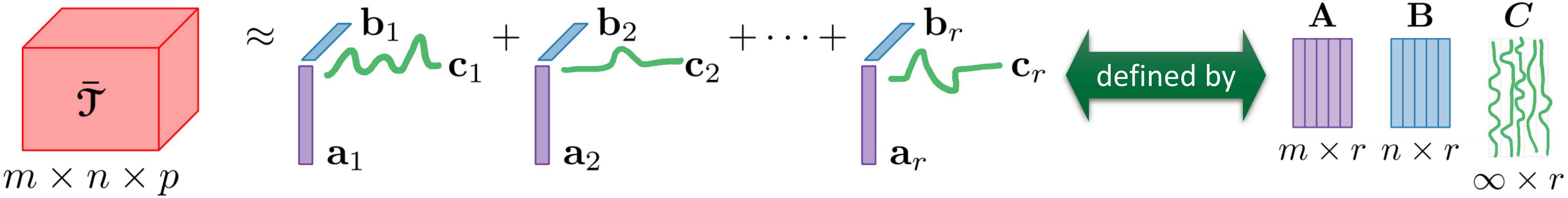


Finite-Dimensional Data

Data

Rank-r CP Model

Factor Matrices



grid sampled
at design points
$\mathbf{v} = [v_1, v_2, \cdots, v_p]^T$

$$\bar{\mathfrak{T}} = T(\mathbf{v})$$

$$T \approx [\![\mathbf{A}, \mathbf{B}, \hat{K}\mathbf{W}]\!]$$

$$T(\mathbf{v}) \approx [\![\mathbf{A}, \mathbf{B}, \hat{K}(\mathbf{v})\mathbf{W}]\!]$$

$$\bar{\mathfrak{T}} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{K}\mathbf{W}]\!]$$

Alternating Least Squares (CP-HIFI-ALS)

MathSci.ai

$$\min_{\mathbf{A},\mathbf{B},\mathbf{W}} \left\| \bar{\mathfrak{T}} - [\![\mathbf{A}, \mathbf{B}, \mathbf{K}\mathbf{W}]\!] \right\|^2 + \lambda \|\mathbf{W}\|_{\mathbf{K}}^2$$

$$\|\mathbf{W}\|_{\mathbf{K}}^2 = \sum_{j=1}^{r} \mathbf{w}_j^\intercal \mathbf{K} \mathbf{w}_j$$

repeat

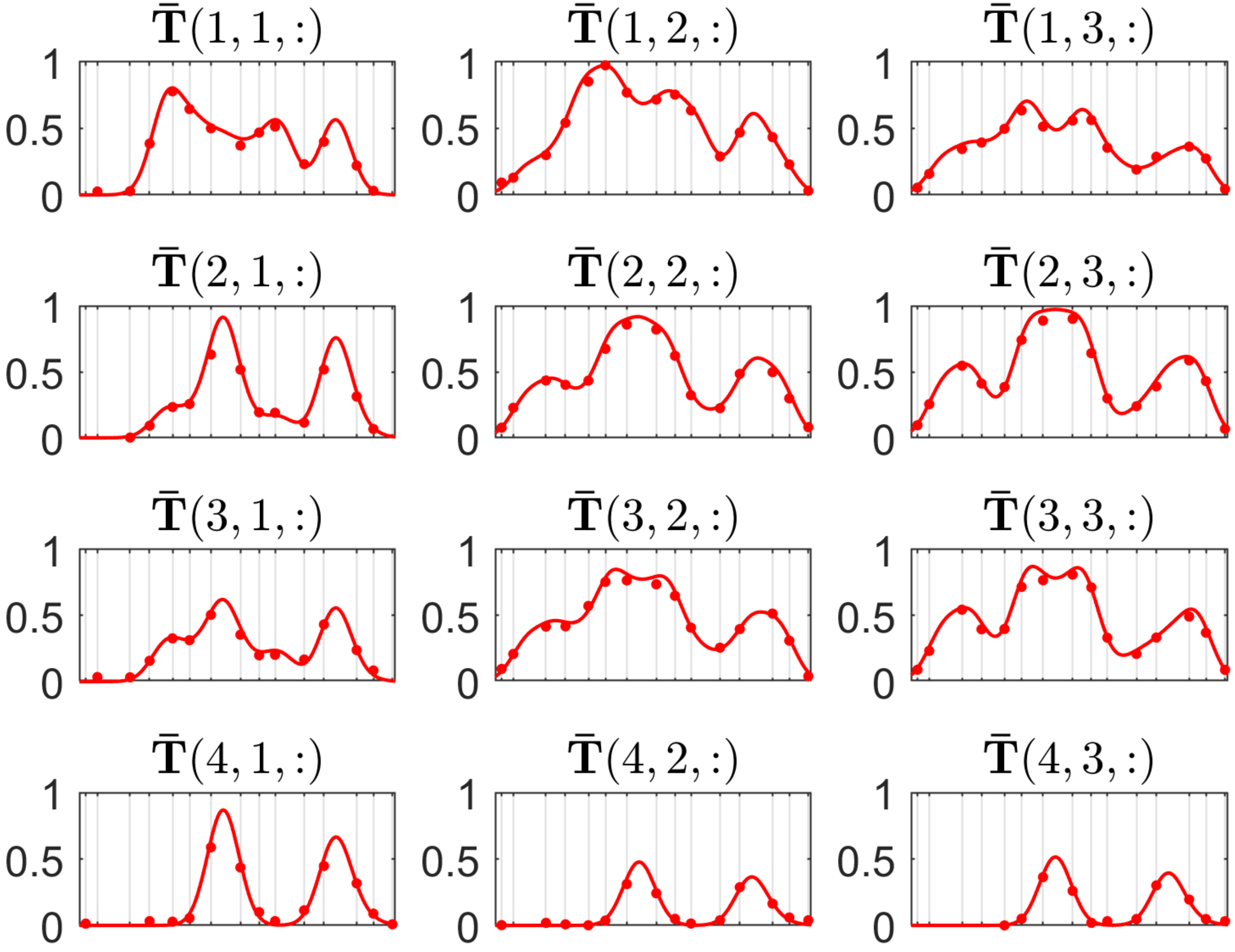
$\quad \mathbf{A} \leftarrow \arg\min_{\mathbf{A}} \left\| (\mathbf{K}\mathbf{W} \odot \mathbf{B})\mathbf{A}^\intercal - \bar{\mathbf{T}}_{(1)}^\intercal \right\|^2$

$\quad \mathbf{B} \leftarrow \arg\min_{\mathbf{B}} \left\| (\mathbf{K}\mathbf{W} \odot \mathbf{A})\mathbf{B}^\intercal - \bar{\mathbf{T}}_{(2)}^\intercal \right\|^2$

$\quad \mathbf{W} \leftarrow \arg\min_{\mathbf{W}} \left\| (\mathbf{B} \odot \mathbf{A})\mathbf{K}\mathbf{W}^\intercal - \bar{\mathbf{T}}_{(3)}^\intercal \right\|^2 + \frac{\lambda}{2} \|\mathbf{W}\|_{\mathbf{K}}^2$

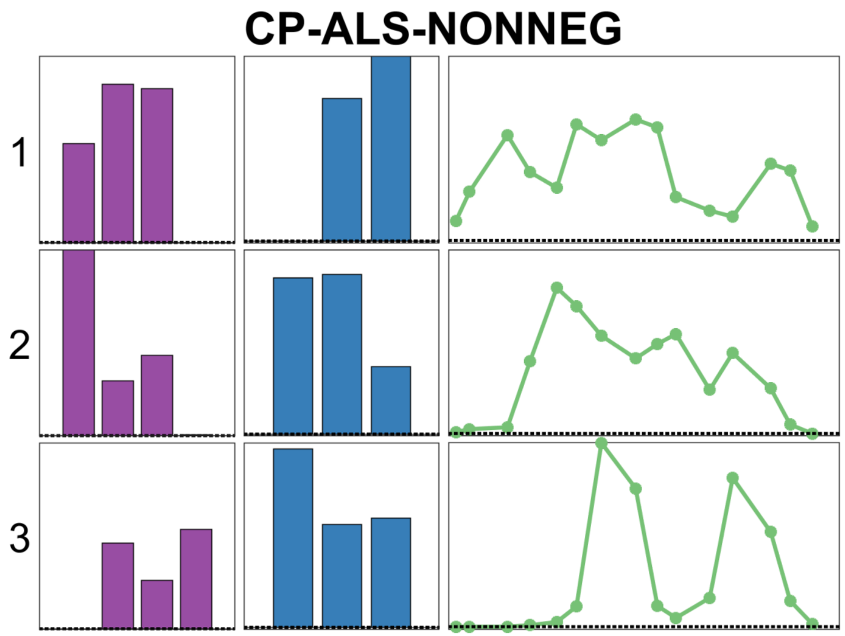
until converged

15 Aligned Observations per Function



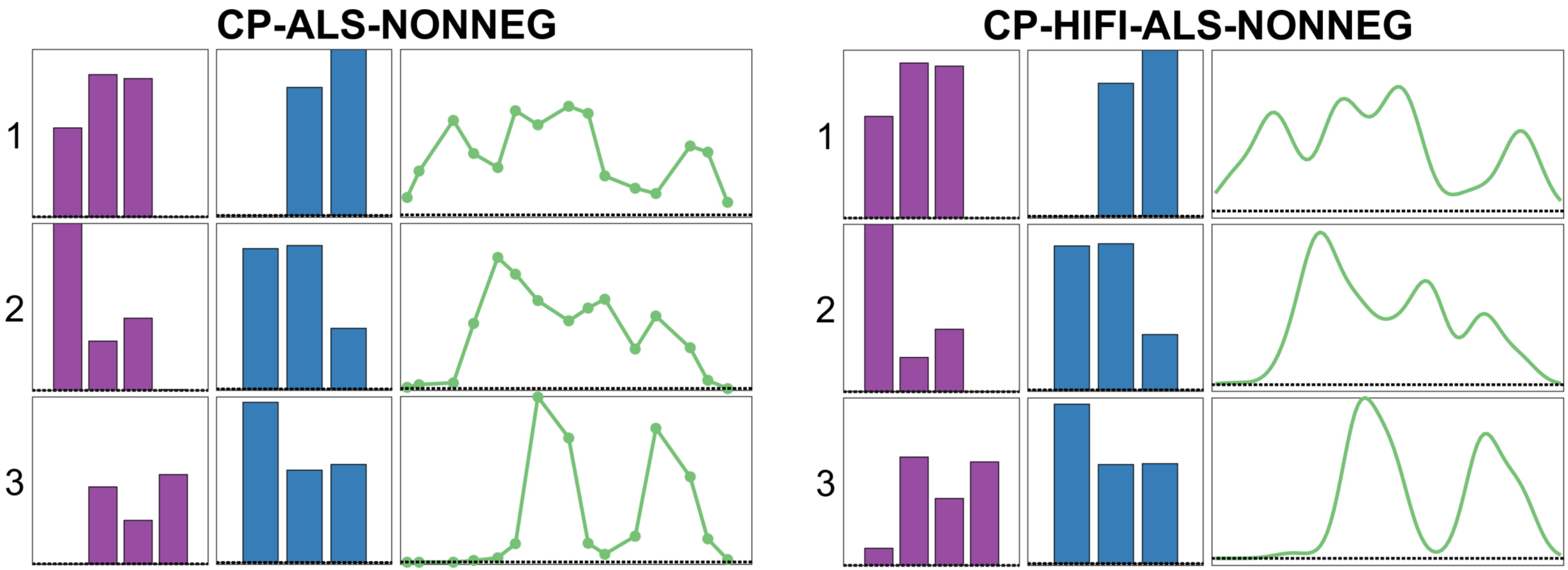
$$\bar{\mathfrak{T}} \in \mathbb{R}^{4\times3\times15}$$

15 aligned, unevenly-spaced noisy samples per function



15 Aligned Observations: CP vs CP-HIFI

MathSci.ai



Unaligned Observations

Data Need Not Always Aligned

$$\boldsymbol{T}(1,1,0.4472) = 0.4892$$
$$\boldsymbol{T}(4,1,0.5879) = 0.0262$$
$$\boldsymbol{T}(3,3,0.7482) = 0.2921$$
$$\boldsymbol{T}(3,1,0.2173) = 0.1404$$
$$\boldsymbol{T}(1,3,0.9970) = 0.0465$$
$$\boldsymbol{T}(3,2,0.8192) = 0.5155$$
$$\boldsymbol{T}(4,1,0.6799) = 0.0209$$
$$\boldsymbol{T}(1,2,0.8192) = 0.6120$$
$$\boldsymbol{T}(1,1,0.5879) = 0.5188$$
$$\boldsymbol{T}(4,2,0.2936) = 0.0025$$
$$\vdots$$

Measurements may be taken at different times for different subjects, e.g., patients coming in for blood work

An irregular grid!

Experimental setup may vary by site, e.g., equipment to measure weather settings might not all use same interval

Unaligned Observations

MathSci.ai

Design Points

$\mathbf{v} = [v_1, v_2, \ldots, v_p] \equiv$ distinct x values in samples $T(i,j,x)$

Observed Points

$\Omega = \{\, (i,j,k) \in [m] \otimes [n] \otimes [p] \mid \boldsymbol{T}(i,j,v_k) \text{ is known} \,\}$

(Partially) Observed $m \times n \times p$ Tensor

$$\bar{\mathfrak{T}}(i,j,k) \equiv \begin{cases} \boldsymbol{T}(i,j,v_k) & \text{if } (i,j,k) \in \Omega \\ 0 & \text{if } (i,j,k) \notin \Omega \end{cases}$$

Norm on Only Observed Points

$\|\mathfrak{T}\|_\Omega^2 \equiv \sum_{(i,j,k)\in\Omega} \mathfrak{T}(i,j,k)^2$

$\boldsymbol{T}(1,1,0.4472) = 0.4892$
$\boldsymbol{T}(4,1,0.5879) = 0.0262$
$\boldsymbol{T}(3,3,0.7482) = 0.2921$
$\boldsymbol{T}(3,1,0.2173) = 0.1404$
$\boldsymbol{T}(1,3,0.9970) = 0.0465$
$\boldsymbol{T}(3,2,0.8192) = 0.5155$
$\boldsymbol{T}(4,1,0.6799) = 0.0209$
$\boldsymbol{T}(1,2,0.8192) = 0.6120$
$\boldsymbol{T}(1,1,0.5879) = 0.5188$
$\boldsymbol{T}(4,2,0.2936) = 0.0025$

Alternating Least Squares (CP-HIFI-ALS)

MathSci.ai

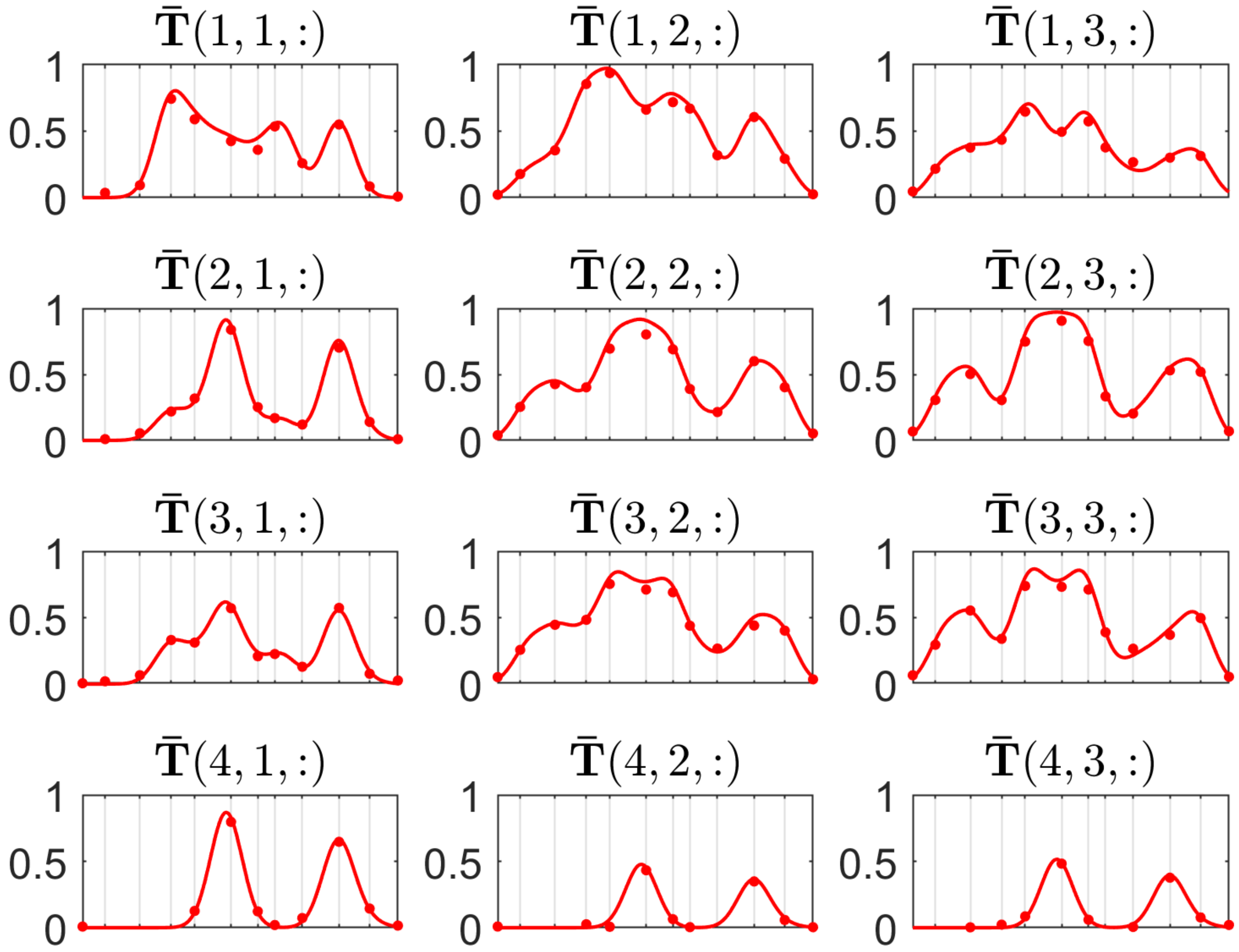
$$\min_{\mathbf{A},\mathbf{B},\mathbf{W}} \|\bar{\mathfrak{T}} - [\![\mathbf{A},\mathbf{B},\mathbf{K}\mathbf{W}]\!]\|_{\Omega}^2 + \lambda \|\mathbf{W}\|_{\mathbf{K}}^2$$

$$\|\mathbf{W}\|_{\mathbf{K}}^2 = \sum_{j=1}^{r} \mathbf{w}_j^{\intercal}\mathbf{K}\mathbf{w}_j$$

$$\|\mathfrak{T}\|_{\Omega}^2 \equiv \sum_{(i,j,k)\in\Omega} \mathfrak{T}(i,j,k)^2$$

1: **repeat**
2: $\quad \mathbf{A} \leftarrow \arg\min_{\mathbf{A}} \|(\mathbf{K}\mathbf{W} \odot \mathbf{B})\mathbf{A}^{\intercal} - \bar{\mathbf{T}}_{(1)}^{\intercal}\|_{\Omega}^2$
3: $\quad \mathbf{B} \leftarrow \arg\min_{\mathbf{B}} \|(\mathbf{K}\mathbf{W} \odot \mathbf{A})\mathbf{B}^{\intercal} - \bar{\mathbf{T}}_{(2)}^{\intercal}\|_{\Omega}^2$
4: $\quad \mathbf{W} \leftarrow \arg\min_{\mathbf{W}} \|(\mathbf{B} \odot \mathbf{A})\mathbf{K}\mathbf{W}^{\intercal} - \bar{\mathbf{T}}_{(3)}^{\intercal}\|_{\Omega}^2 + \frac{\lambda}{2}\|\mathbf{W}\|_{\mathbf{K}}^2$
5: **until** converged

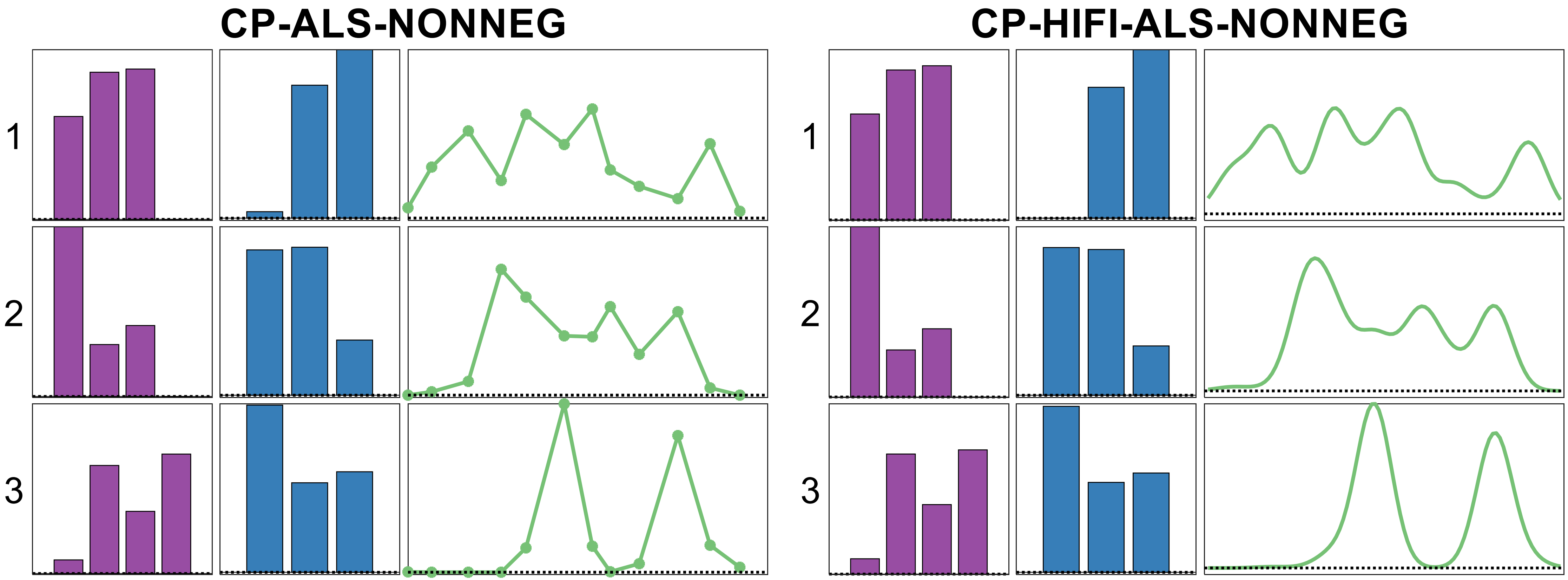
12 Aligned Observations per Function



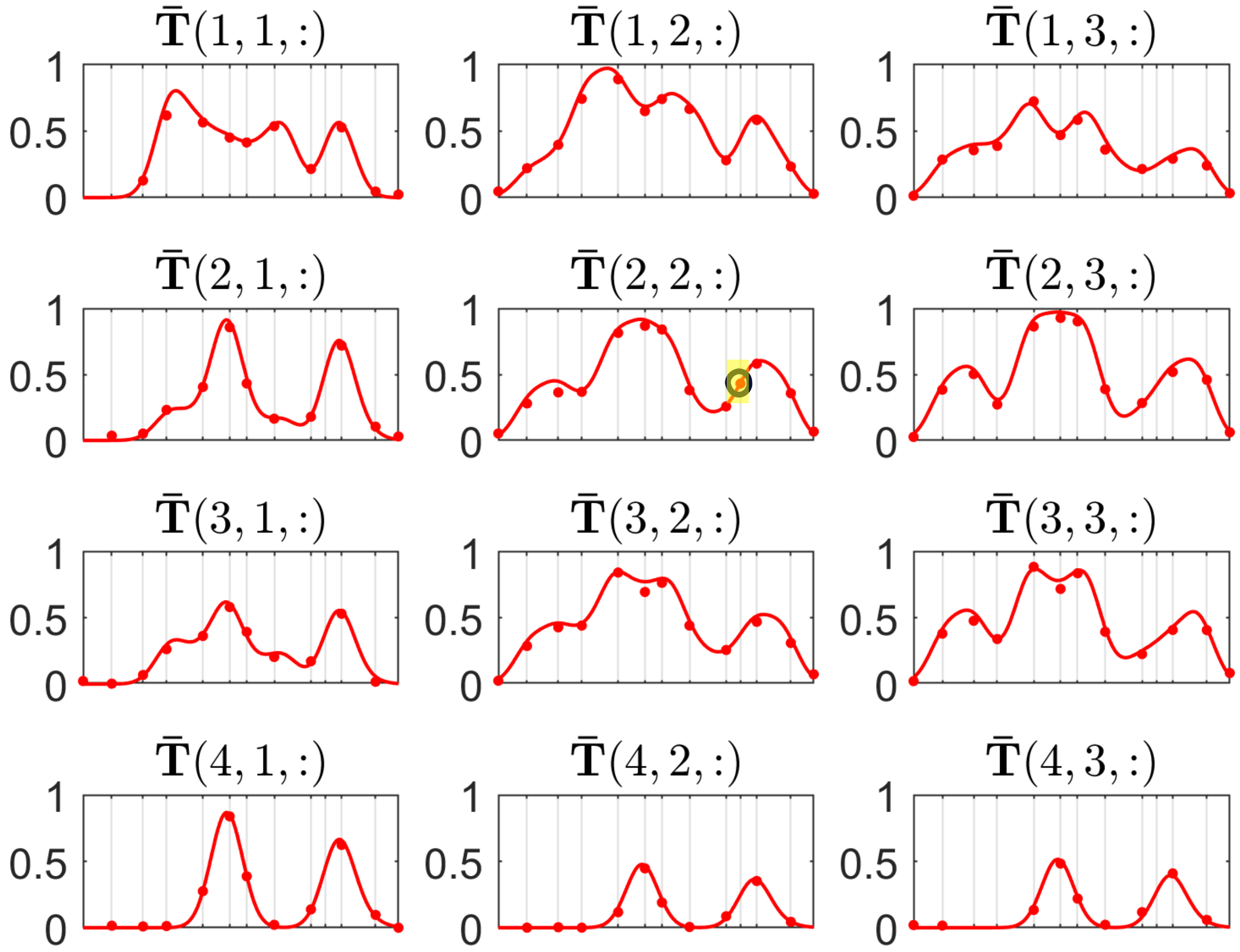
$\bar{\mathfrak{T}} \in \mathbb{R}^{4\times 3\times 12}$

12 aligned, unevenly-spaced

12 Aligned Observations: CP vs CP-HIFI



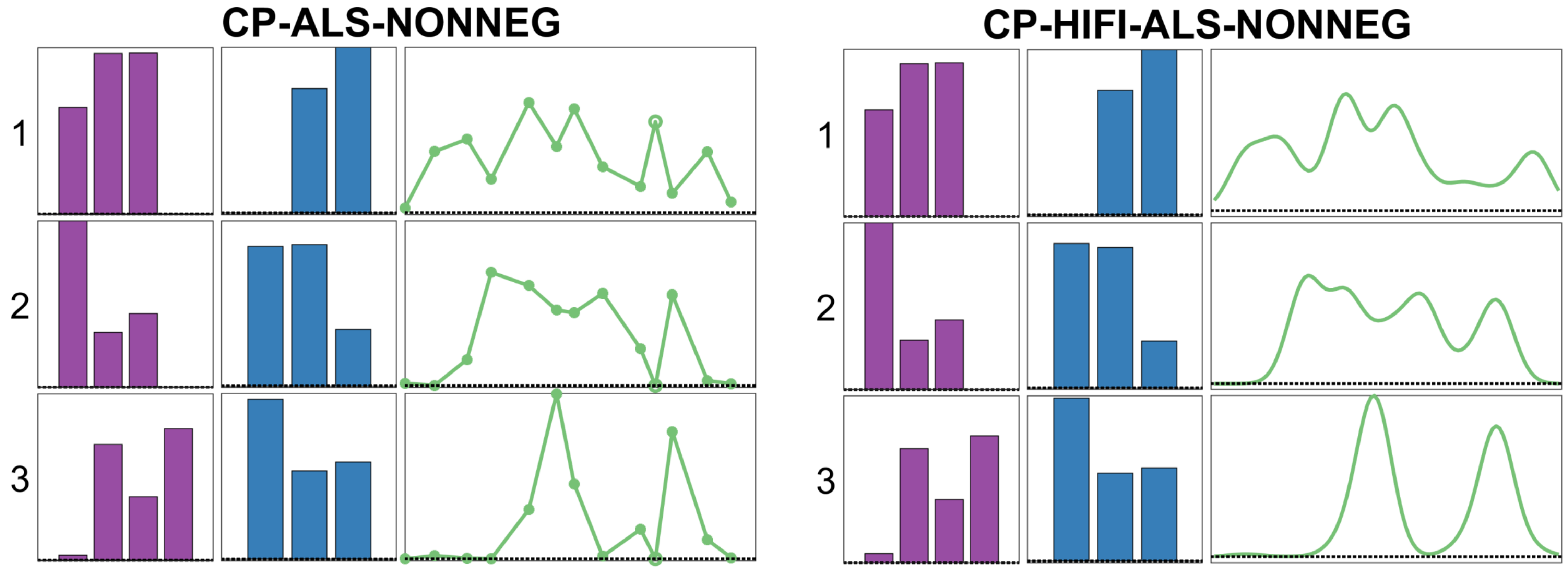
12 Aligned Per Function + 1 Extra Data Point



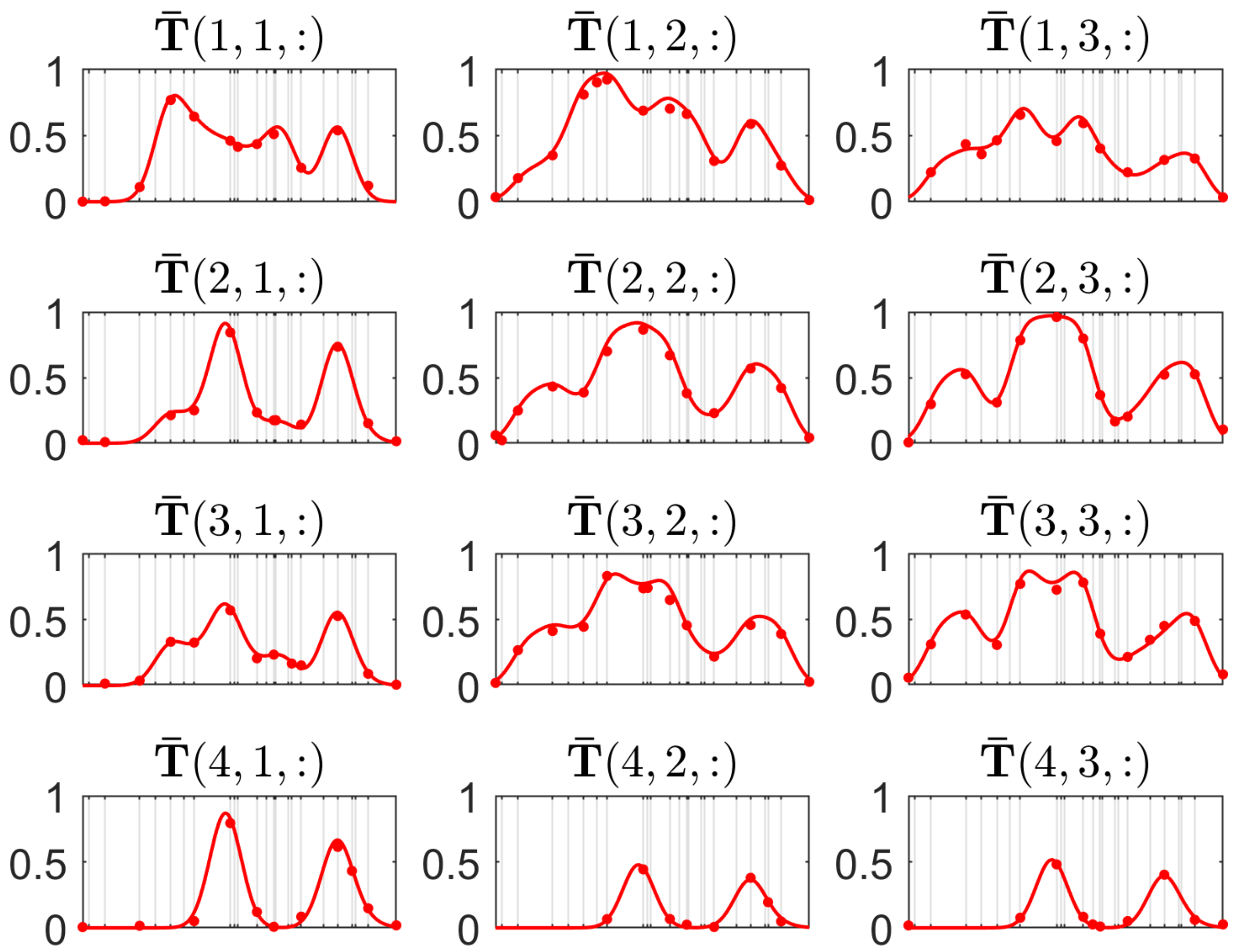
$\bar{\mathcal{T}} \in \mathbb{R}^{4\times 3\times 13}$

12 aligned, unevenly-spaced
plus 1 extra point for (2,2)

12 Aligned + 1 Extra: CP vs CP-HIFI



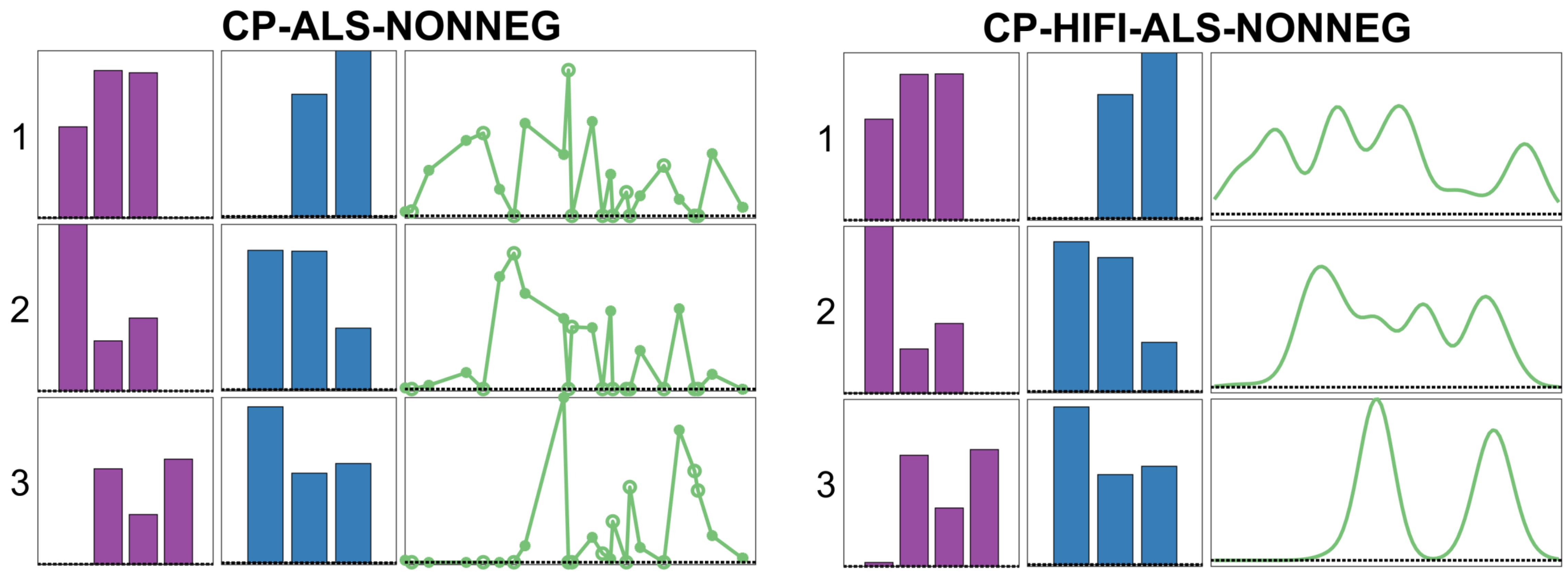
12 Aligned + 1 Unaligned per Function



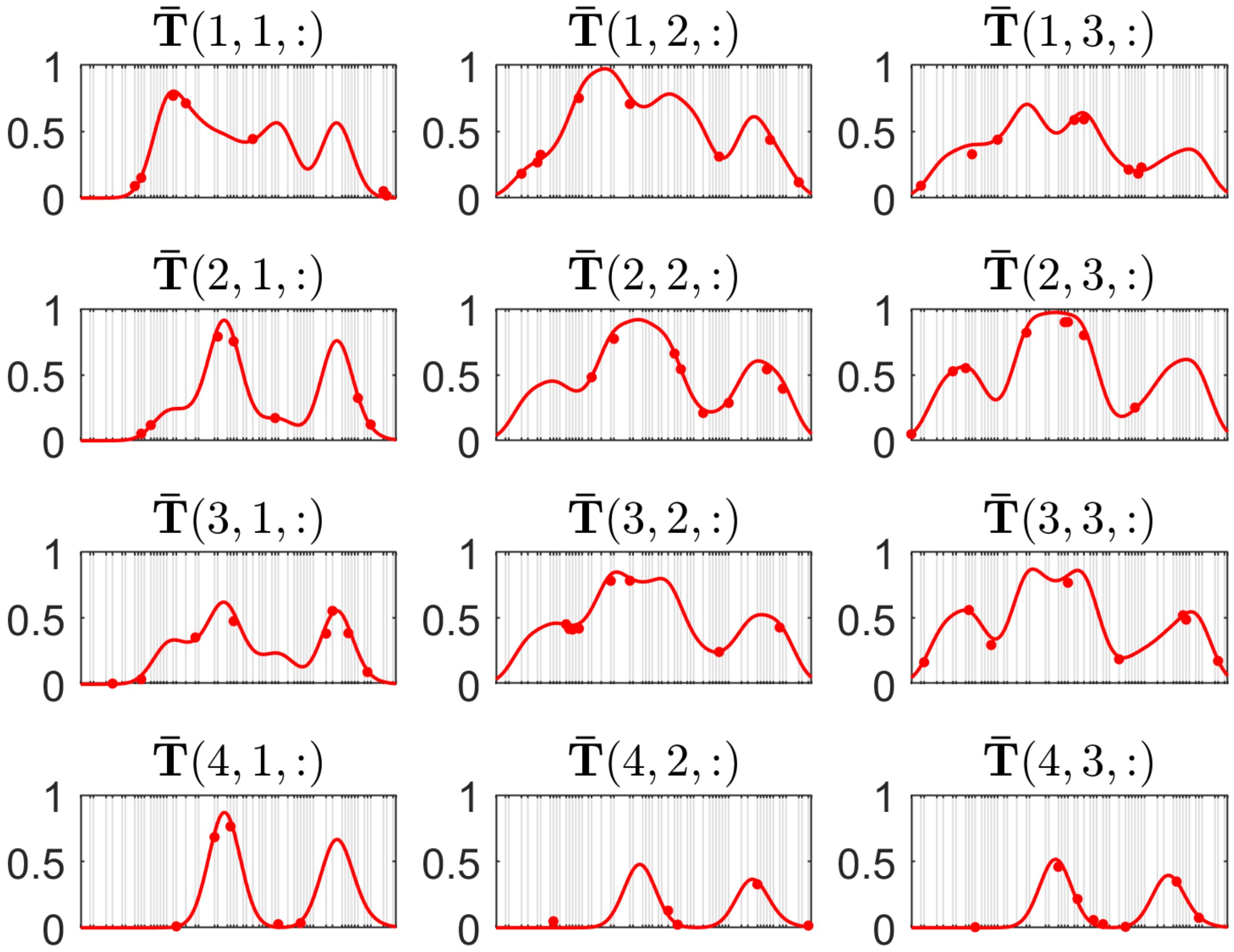
$\bar{\mathcal{T}} \in \mathbb{R}^{4\times 3\times 24}$

12 aligned, unevenly-spaced
plus 1 extra point for each (i,j)

12 Aligned + 1 Unaligned: CP vs CP-HIFI



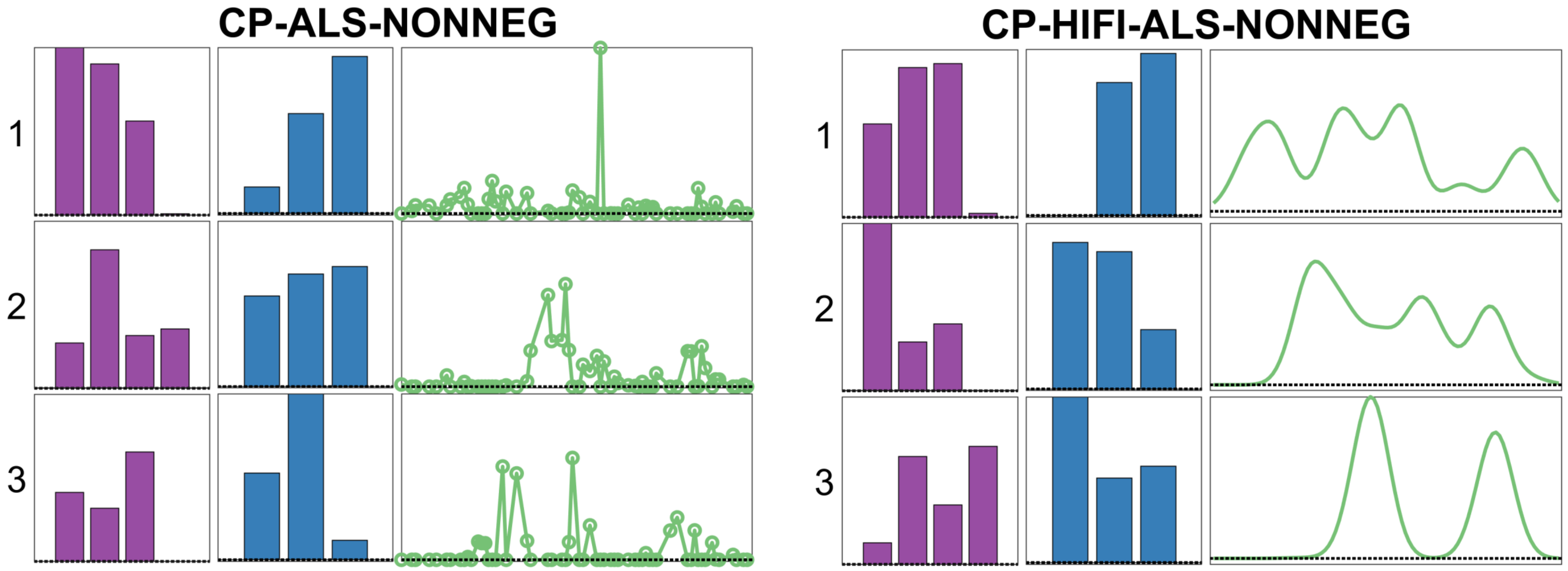
12 Unaligned Points Per Function



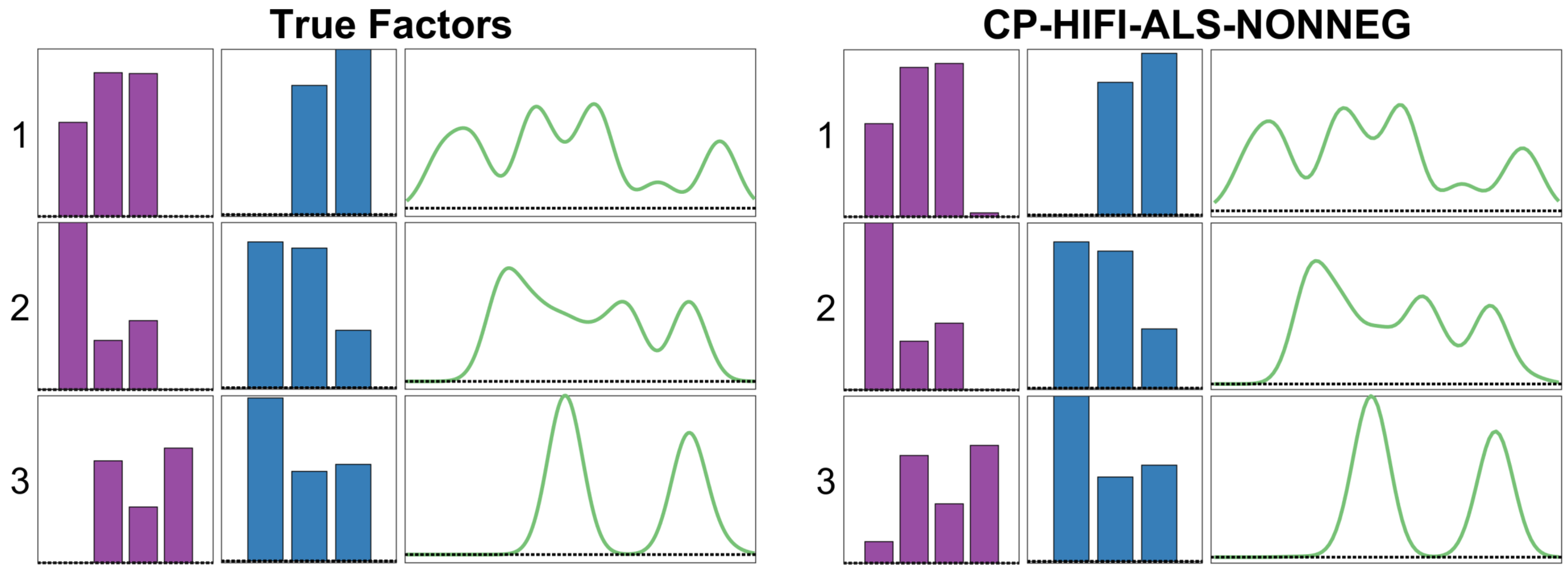
$$\bar{\mathcal{T}} \in \mathbb{R}^{4\times 3\times 60}$$

12 unaligned, unevenly-spaced points per (i, j)

12 Unaligned: CP vs CP-HIFI



12 Unaligned: True vs CP-HIFI



Conclusions

Conclusions & Future Work

MathSci.ai

Tensors & Tensor Decomposition

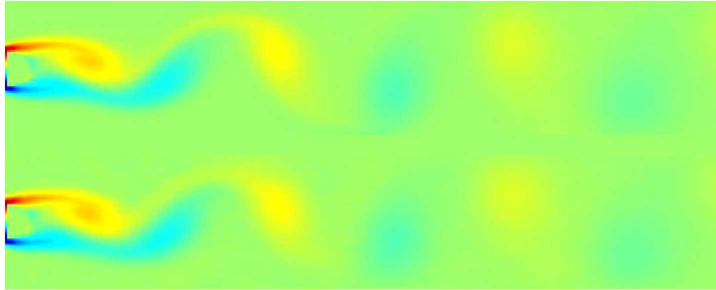
- Tensor data ubiquitous in modeling
- Tensor decomposition yields orders-of-magnitude reduction

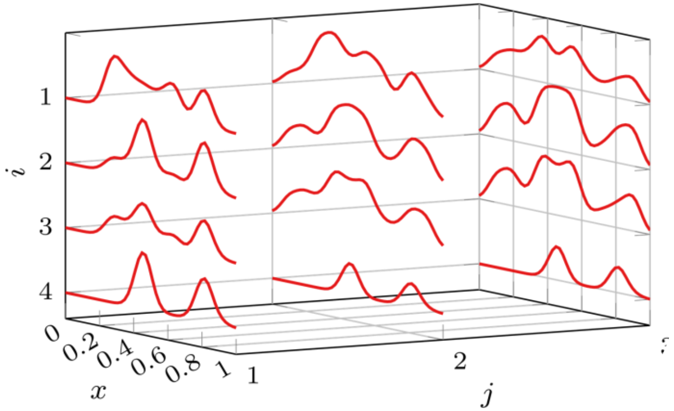
Quasi-Tensors & Decomposition

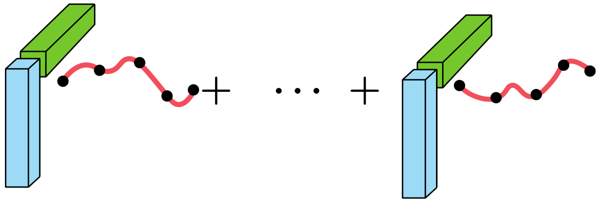
- Quasi-Tensors have one or more “continuous” modes
- Decomposed with functions rather than vectors
- Variety of methods to yield functions (or function-like vectors)
- RKHS = principled way to learn smooth functions
- Aligned versus unaligned data

Connecting with ROM: Some Ideas…

- POD with tensor rather than matrix decomposition?
- Learning functions rather than vectors (in any decomposition)?
- Assimilating unstructured grid data?

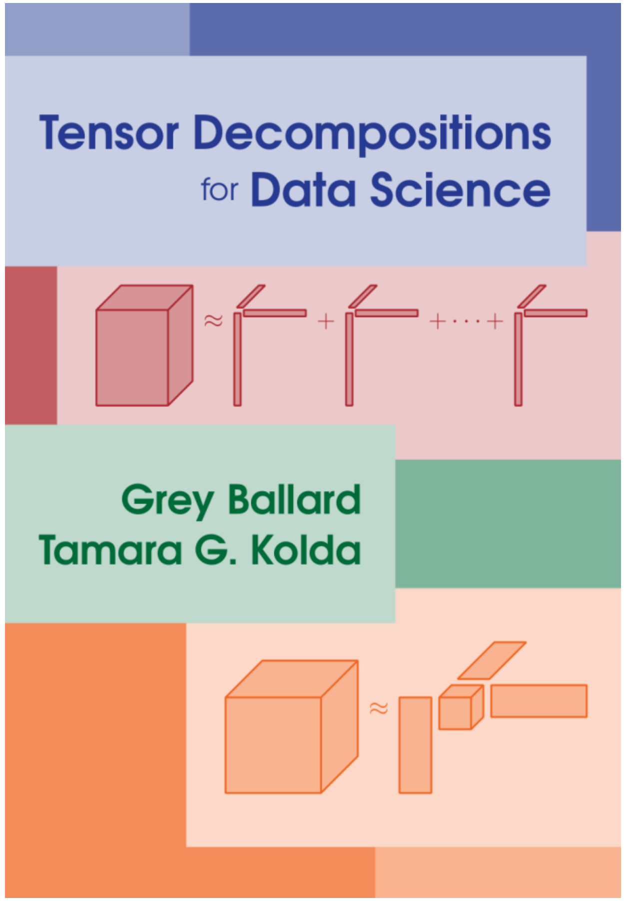




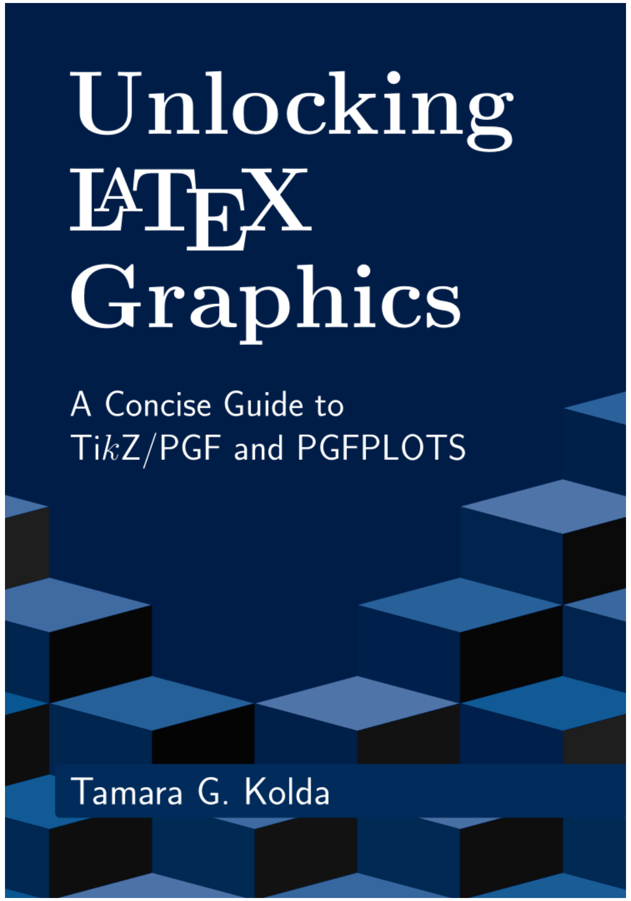


Tensor Decomposition Meets RKHS: Efficient Algorithms for Smooth and Misaligned Data, http://arxiv.org/abs/2408.05677

Two Books and a Channel

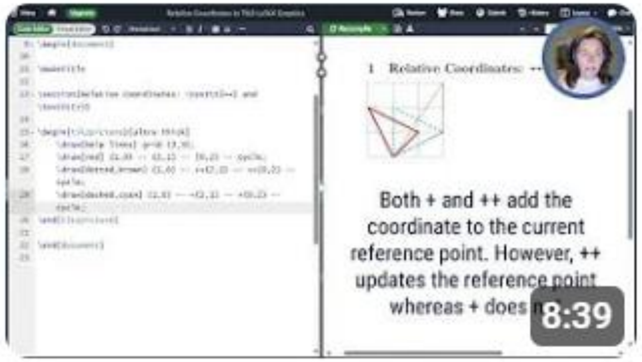


mathsci.ai/tensor-textbook
PDF free online and coming soon from Cambridge University Press



latex-graphics.com
Print-on-demand now available and coming soon to Amazon.com

https://www.youtube.com/@UnlockingLaTeXGraphics



Relative Coordinates (++ and +) and Turns in TikZ LaTeX...