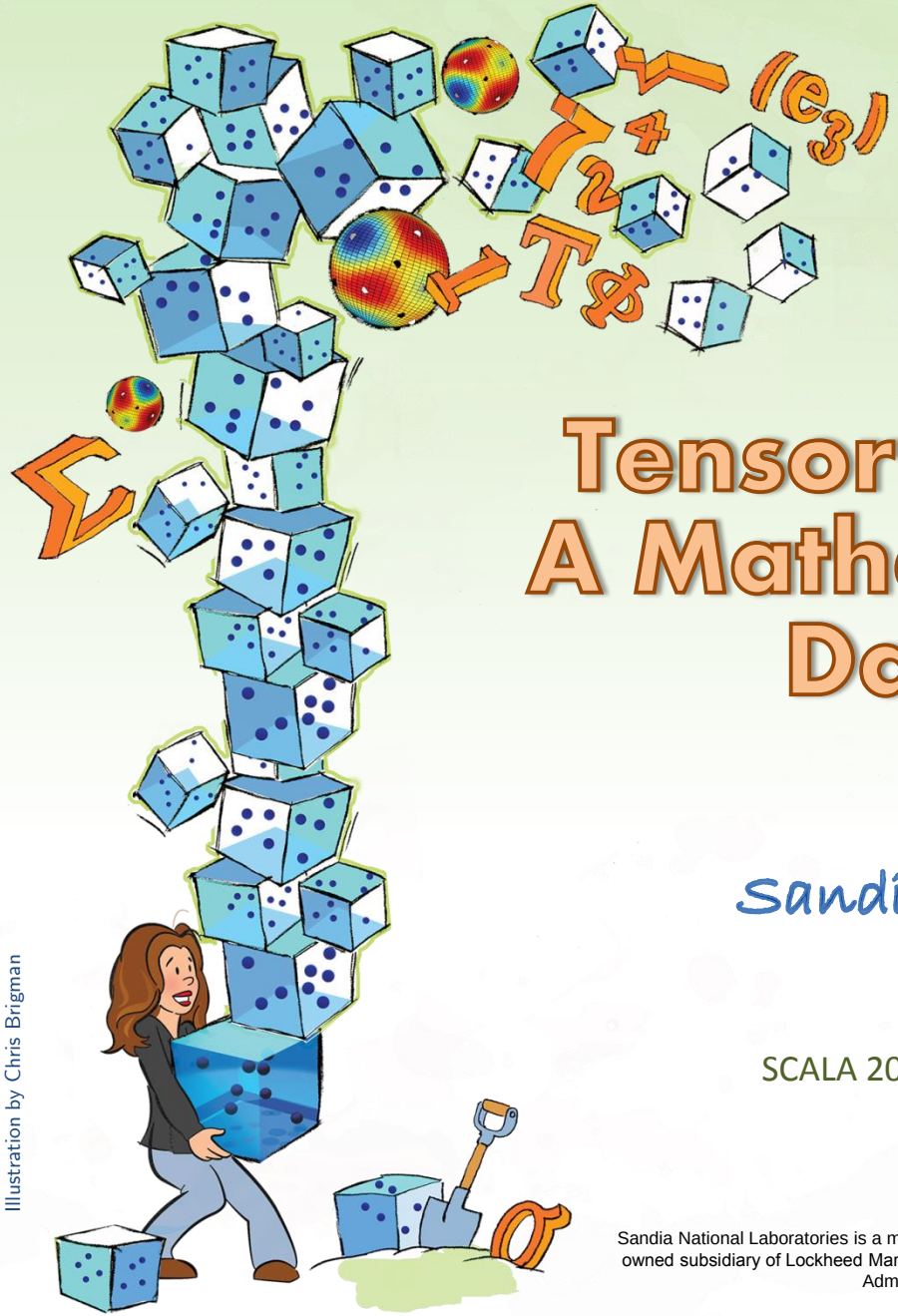


Illustration by Chris Brigman

Sandia National Laboratories

SAND2017-2987C

Tensor Decomposition: A Mathematical Tool for Data Analysis

Tamara G. Kolda
Sandia National Laboratories
Livermore, CA

SCALA 2017 : Scientific Computing Around Louisiana
Tulane University, New Orleans
March 17, 2017

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

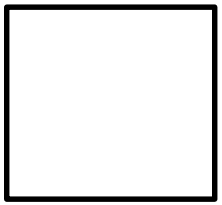
U.S. DEPARTMENT OF ENERGY

A Tensor is an d-Way Array

Vector
d = 1

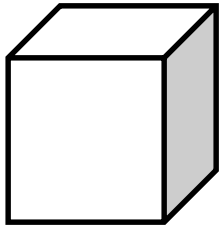
$\mathbf{a}$

Matrix
d = 2



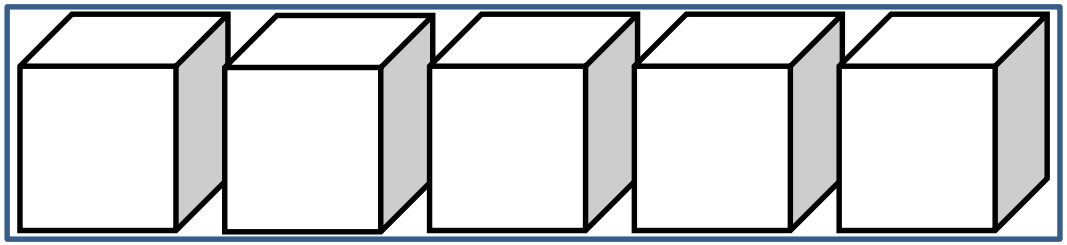
$\mathbf{A}$

3rd-Order Tensor
d = 3



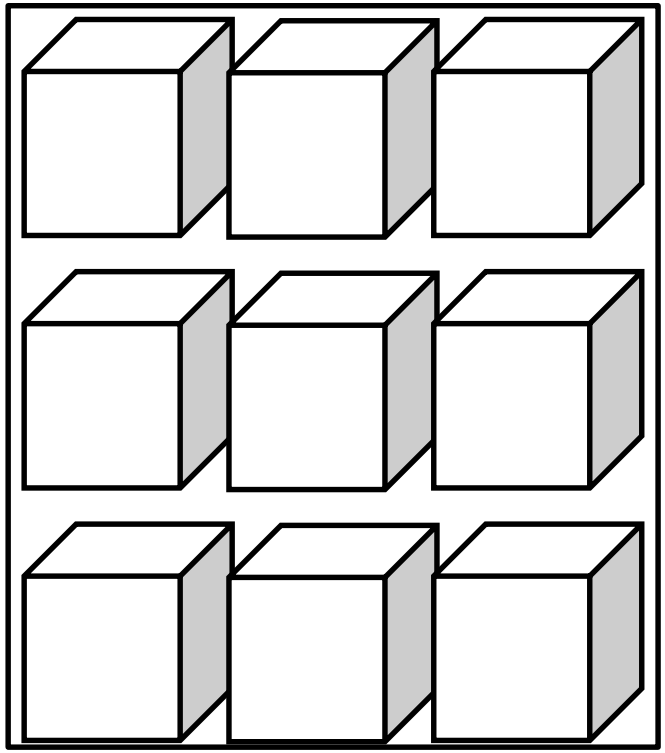
$\mathcal{A}$

4th-Order Tensor
d = 4



$\mathcal{A}$

5th-Order Tensor
d = 5

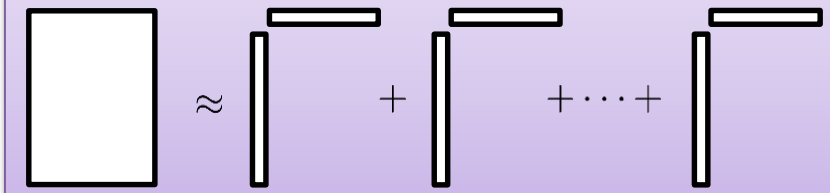


$\mathcal{A}$

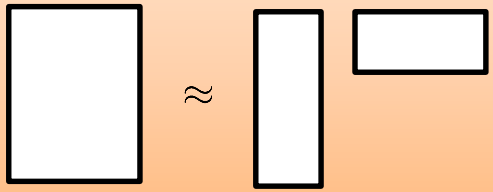
From Matrices to Tensors: Two Points of View

Singular value decomposition (SVD), eigendecomposition (EVD), nonnegative matrix factorization (NMF), sparse SVD, etc.

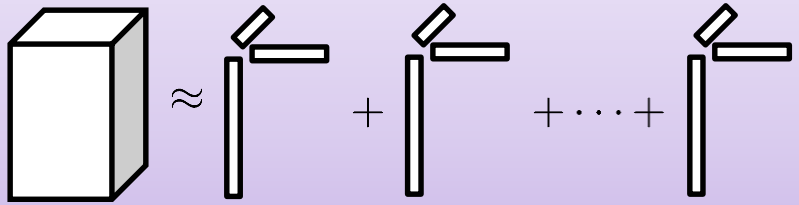
Viewpoint 1: Sum of outer products, useful for interpretation



Viewpoint 2: High-variance subspaces, useful for compression

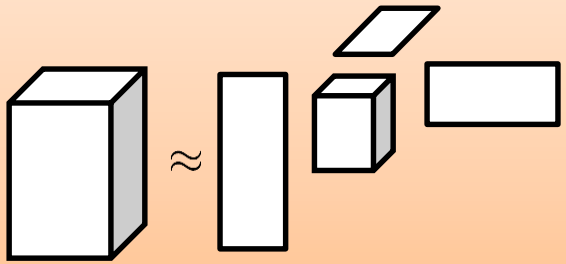


CP Model: Sum of d-way outer products, useful for interpretation



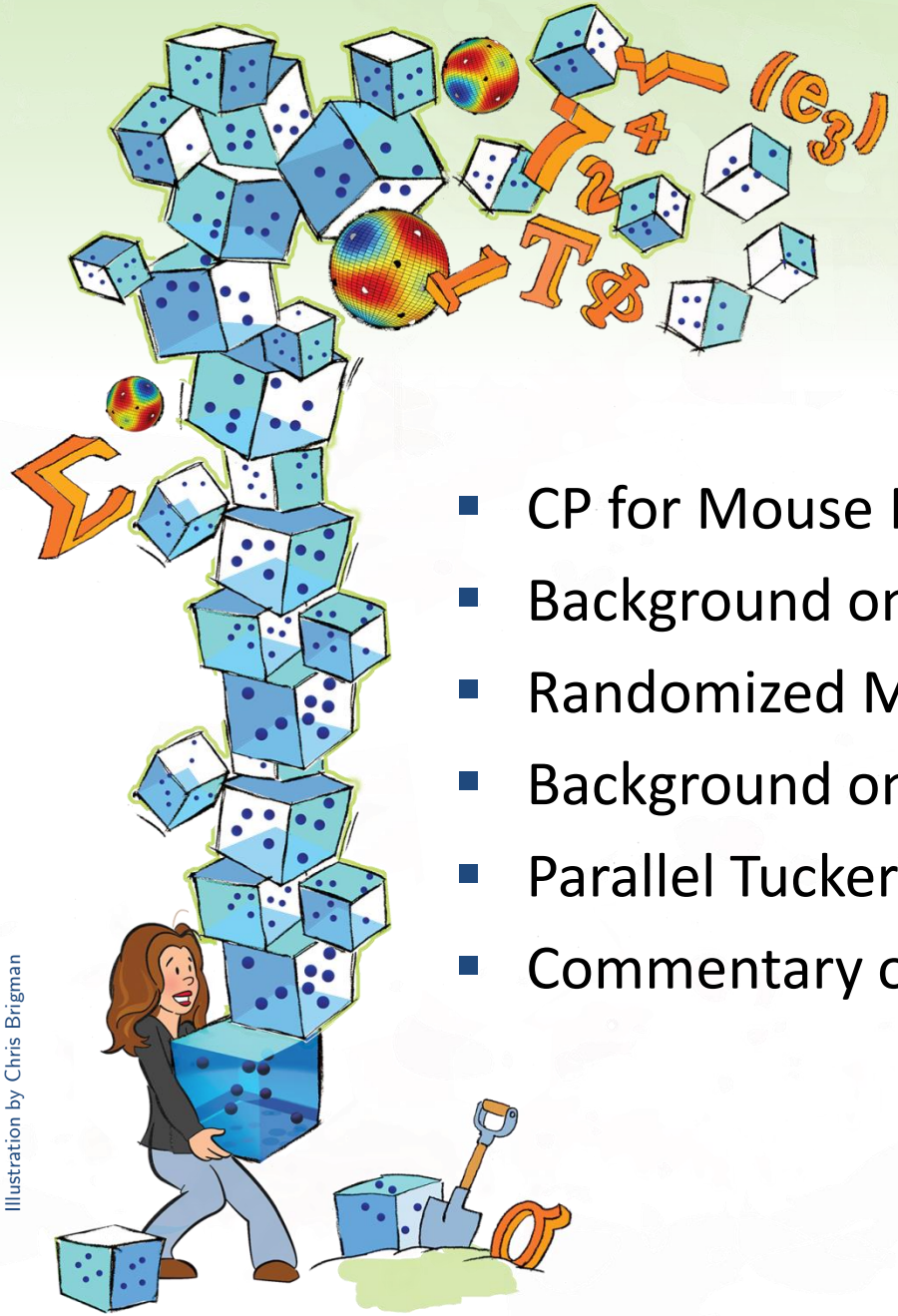
CANDECOMP, PARAFAC, Canonical Polyadic, CP

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



HO-SVD, Best Rank-(R1,R2,…,RN) decomposition

Other models for compression include hierarchical Tucker and tensor train.



Sandia National Laboratories

Outline

- CP for Mouse Neural Activity
- Background on Computing CP
- Randomized Method for Computing CP
- Background on Computing Tucker
- Parallel Tucker for Data Compression
- Commentary on Model Fitting, ex. Binary Data

Sandia National Laboratories

Motivation: CP for Mouse Neural Activity

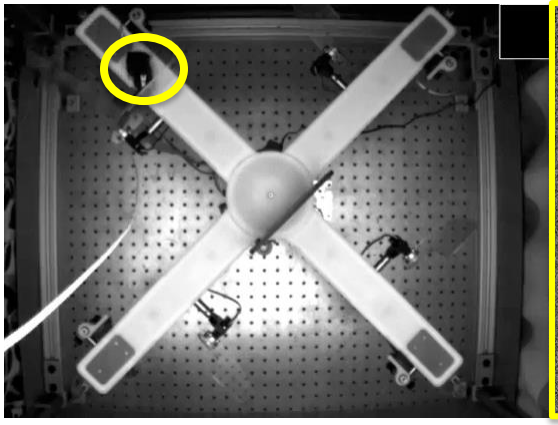
Featuring work of Alex Williams

Motivating Example: Neuron Activity in Learning

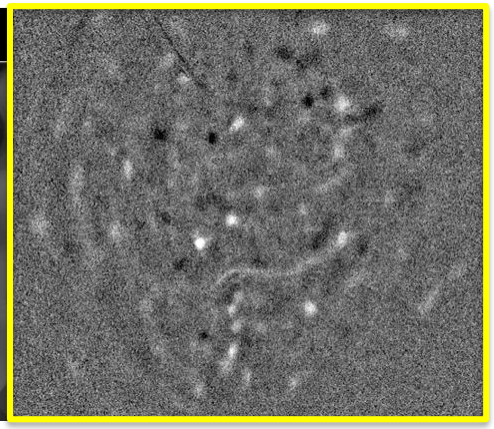
Thanks to Schnitzer Group @ Stanford
Mark Schnitzer, Fori Wang, Tony Kim

Microscope by Inscopix

mouse in "maze"



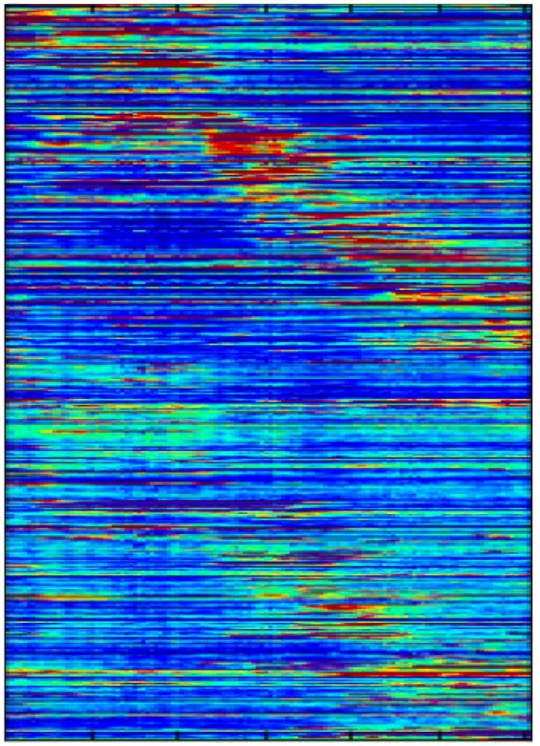
neural activity



One Column of Neuron x Time Matrix

One Trial

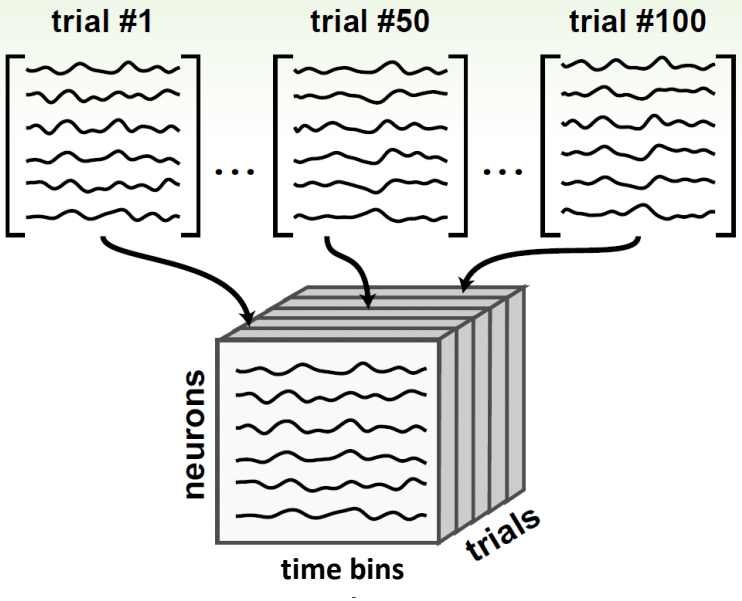
300 neurons × 120 time bins

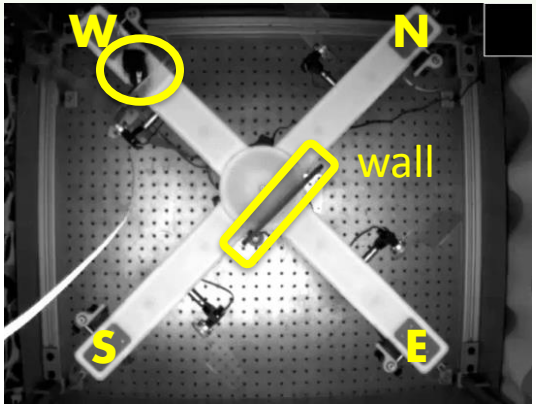


× 600 trials (over 5 days)

Coming soon: Williams, Wang, Kim, Schnitzer, Ganguli, Kolda, 2016

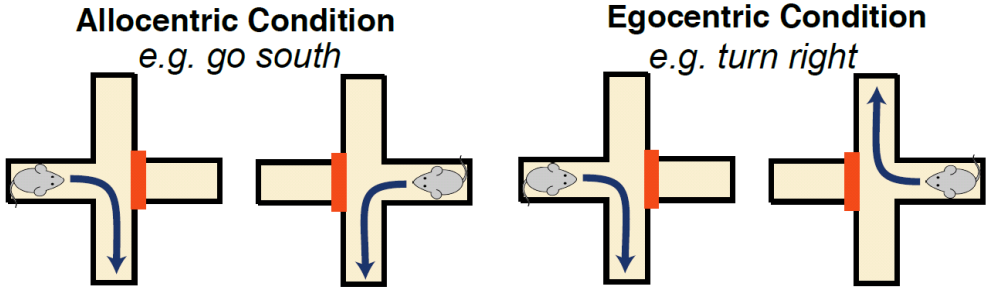
Trials Vary Start Position and Strategies





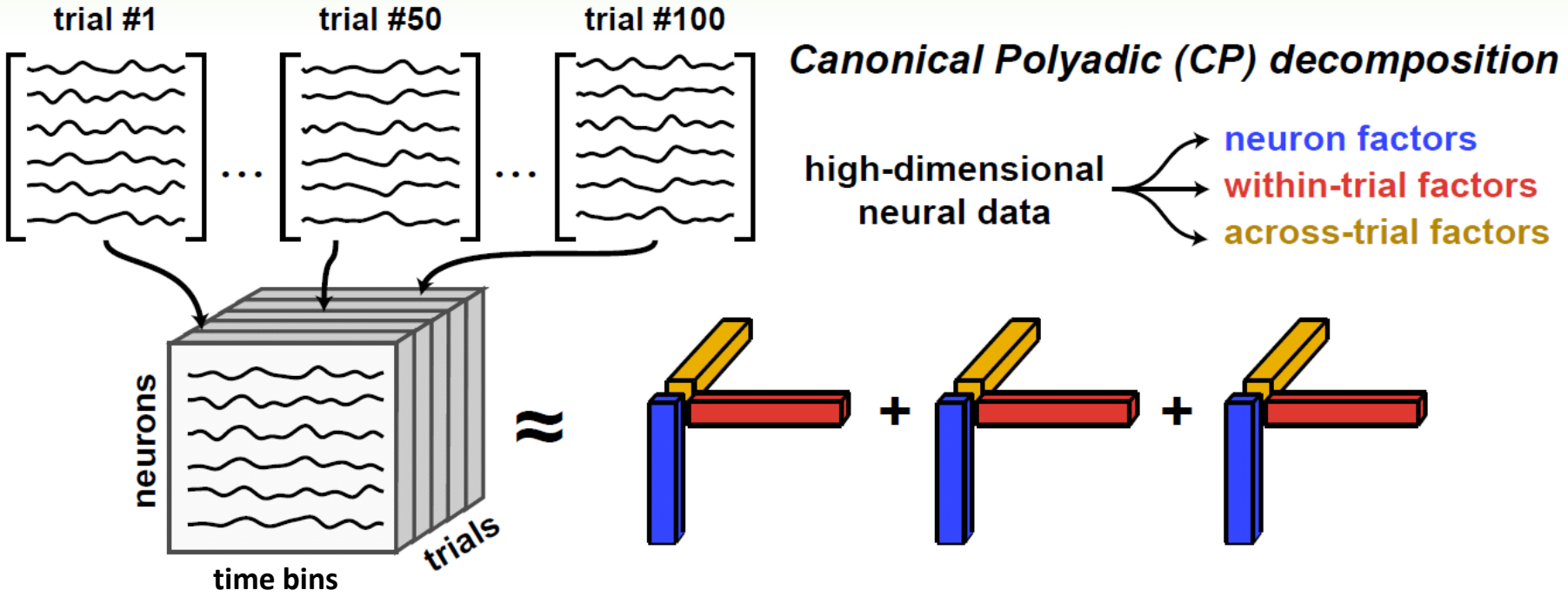
note different patterns on curtains

- 600 Trials over 5 Days
- Start West or East
- Conditions Swap Twice
  - ❖ Always Turn South
  - ❖ Always Turn Right
  - ❖ Always Turn South



Coming soon: Williams, Wang, Kim, Schnitzer, Ganguli, Kolda, 2016

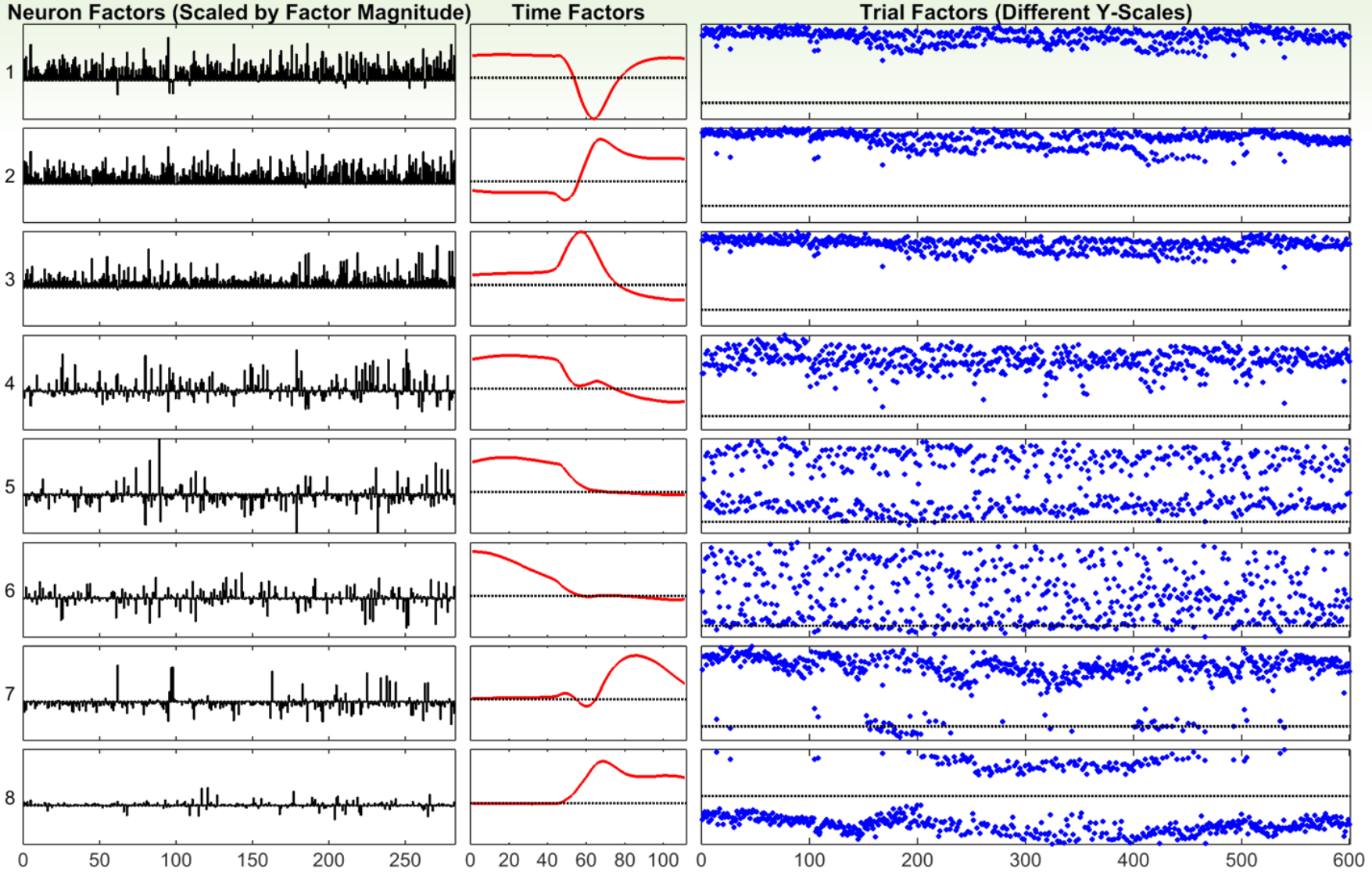
CP for Simultaneous Analysis of Neurons, Time, and Trial



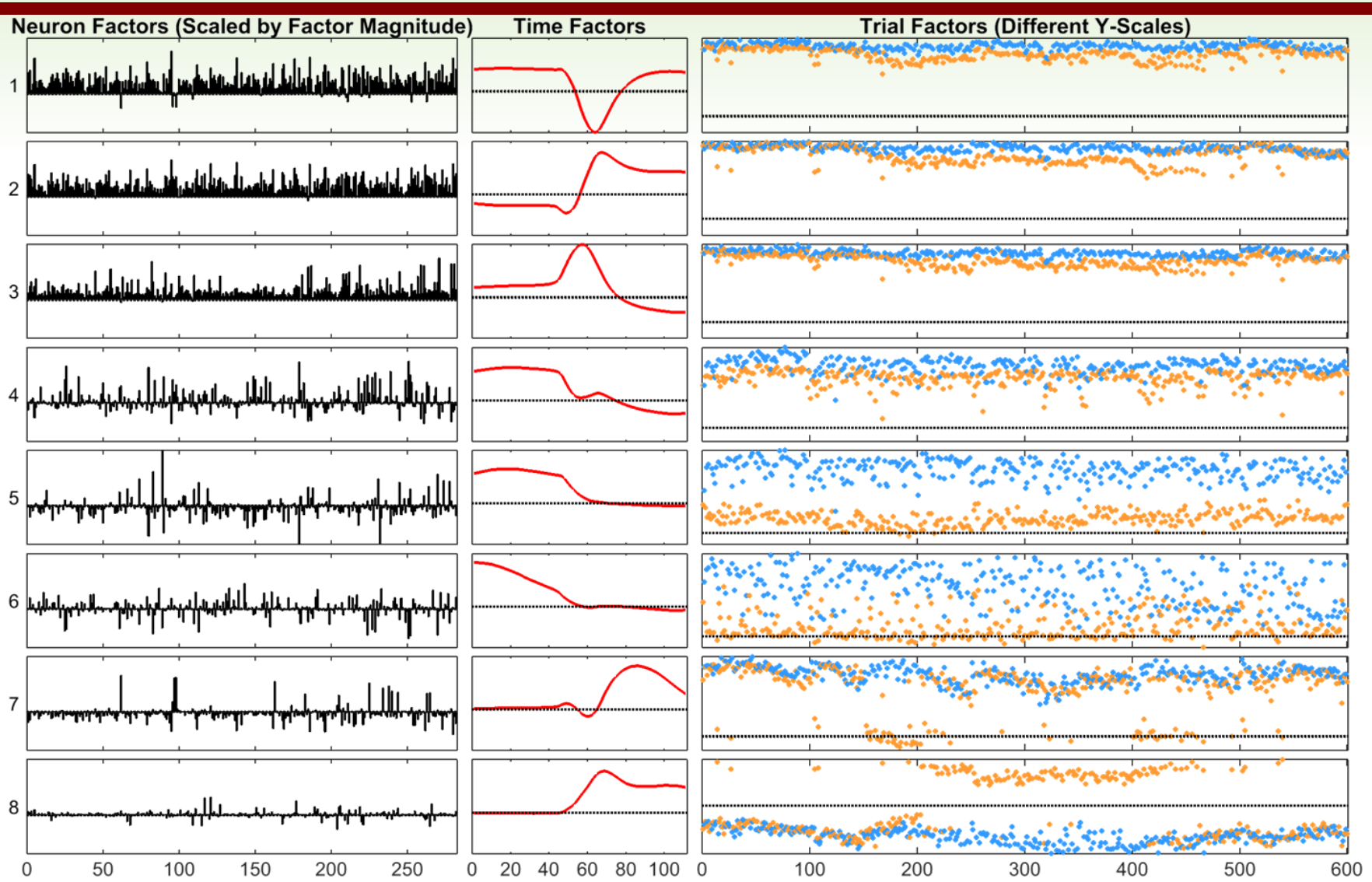
Past work could only look at 2 factors at once: Time x Neuron, Trial x Neuron, etc.

Coming soon: Williams, Wang, Kim, Schnitzer, Ganguli, Kolda, 2016

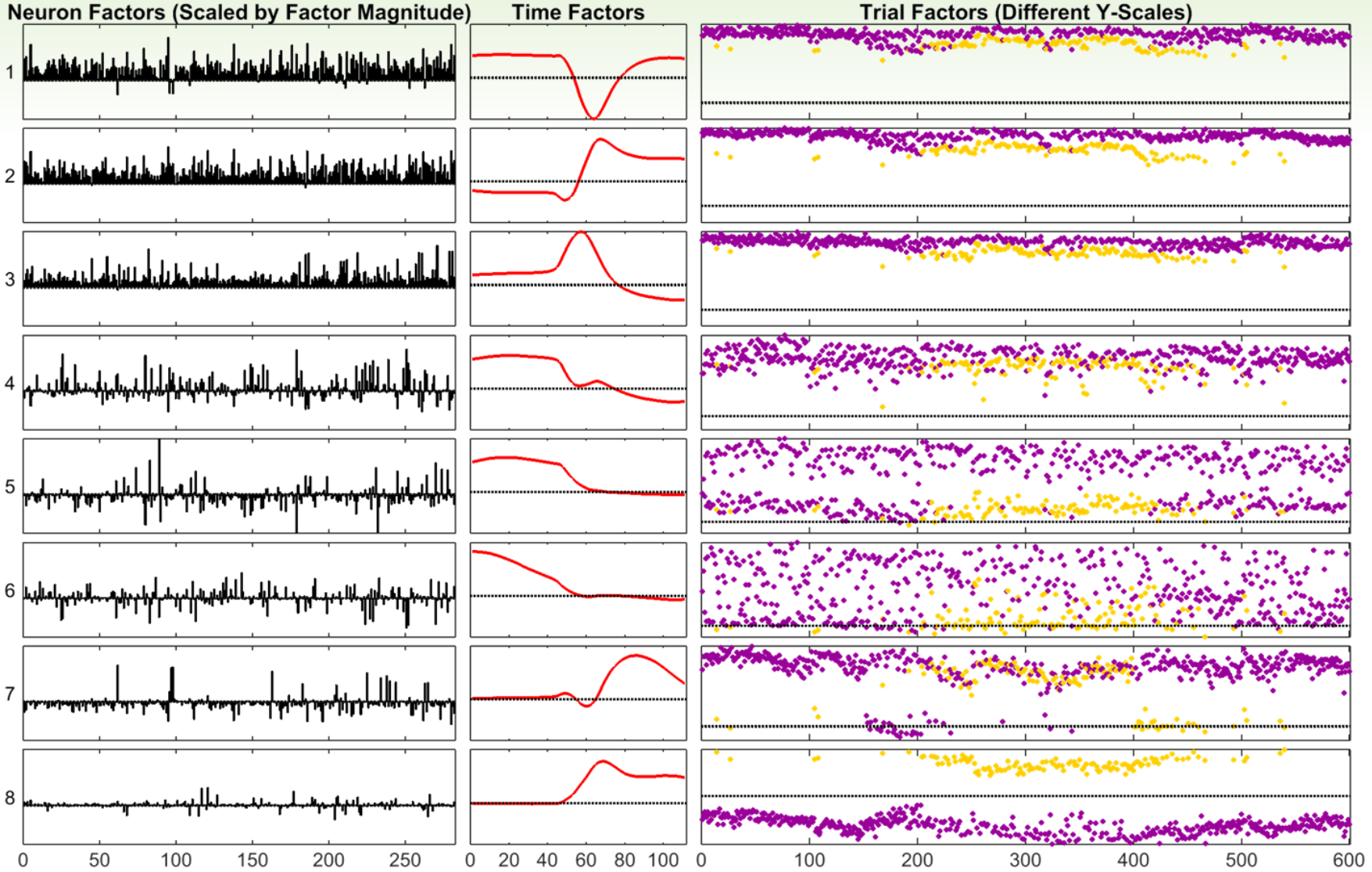
8-Component CP Tensor Decomposition



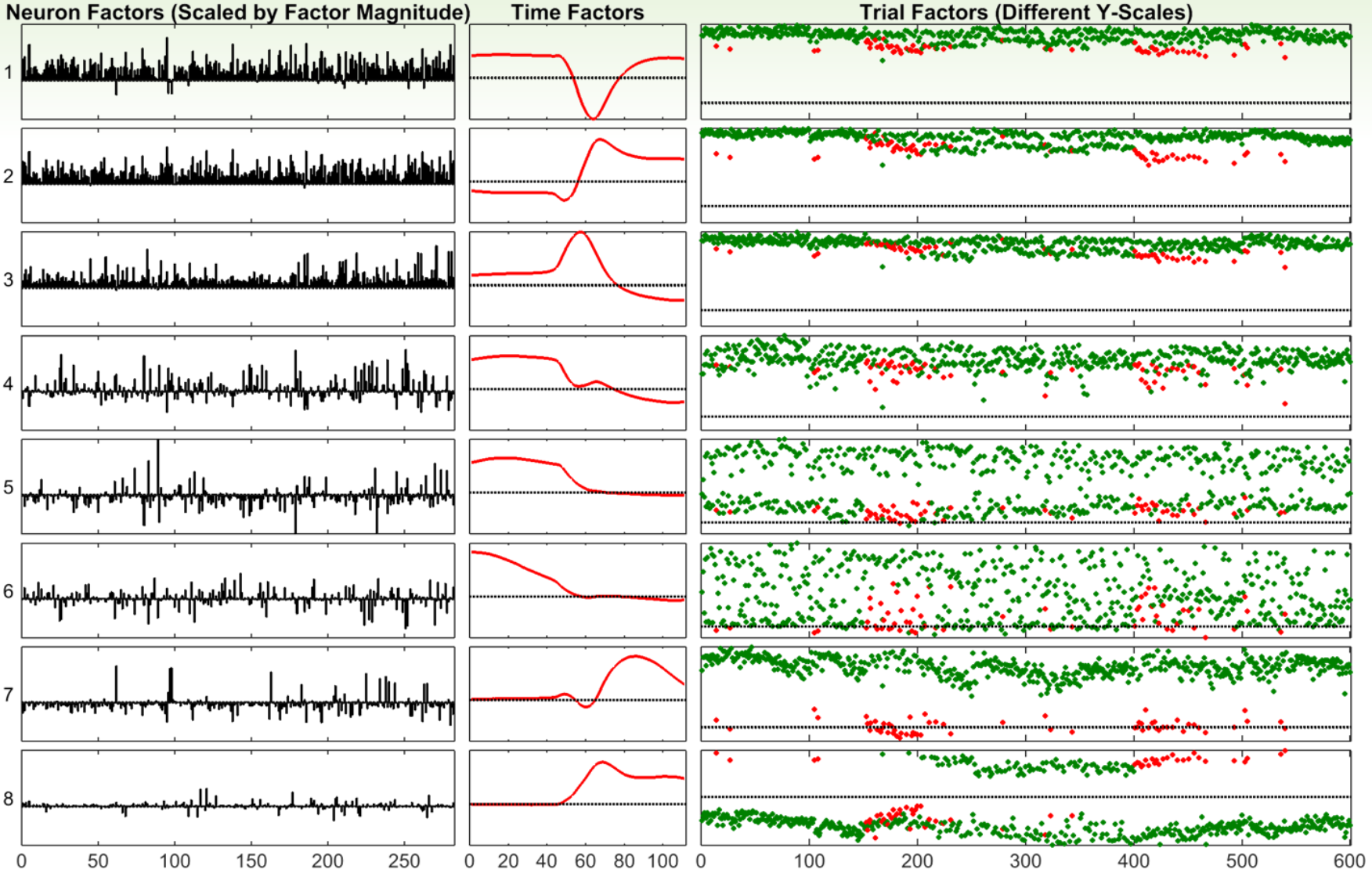
Component 5 Separates Starting Point (Orange = East, Blue = West)

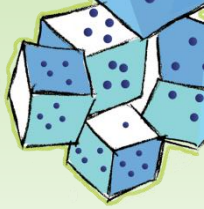


Component 8 Separates End Point (Purple = South, Orange = North)



Component 7 Separates Reward (Green = Yes, Red = No)

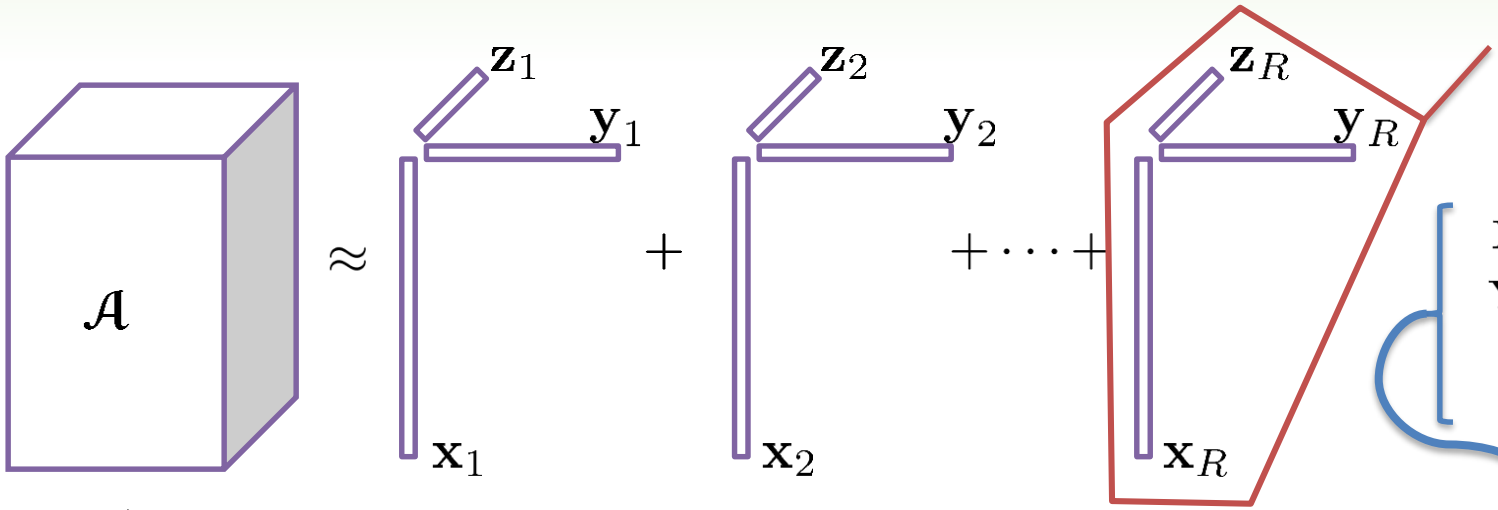




Background on Computing CP

CP Tensor Decomposition: Sum of Outer Products

CANDECOMP/PARAFAC or canonical polyadic (CP) Model

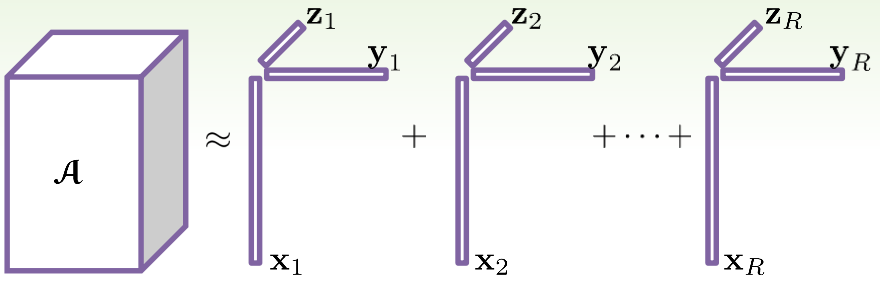


Model: $\mathcal{M} = \sum_r \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$

$$\min_{\mathcal{M}} \sum_{ijk} (a_{ijk} - m_{ijk})^2 \quad \text{subject to} \quad m_{ijk} = \sum_r x_{ir}\, y_{jr}\, z_{kr}$$

Key references: Hitchcock, 1927; Harshman, 1970; Carroll and Chang, 1970

CP-ALS: Fitting CP Model via Alternating Least Squares



$$f(\mathbf{X},\mathbf{Y},\mathbf{Z}) = \sum_{ijk}\left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr}\right)^2$$

Repeat until convergence:

Step 1: $\min_{\mathbf{X}} \sum_{ijk}\left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr}\right)^2$

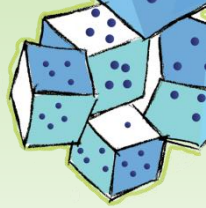
Step 2: $\min_{\mathbf{Y}} \sum_{ijk}\left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr}\right)^2$

Step 3: $\min_{\mathbf{Z}} \sum_{ijk}\left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr}\right)^2$

- **Rank (R) NP-Hard:** . Even best low-rank solution may not exist (Håstad 1990, Silva & Lim 2006, Hillar & Lim 2009)
- **Not nested:** Best rank-(R-1) factorization may not be part of best rank-R factorization (Kolda 2001)
- **Nonconvex:** But convex linear least squares problems
- **Not orthogonal:** Factor matrices are not orthogonal and may even have linearly dependent columns
- **Objective imperfect:** Ideally want to detect true structure, which is usually essentially unique (Kruskal 1977) but cannot measure directly
- ***Need to solve nonconvex optimization problem for many values of R.***

Harshman, 1970; Carroll & Chang, 1970

Sandia National Laboratories



A Randomized Method for Fitting the CP Decomposition

Featuring work of Casey Battaglino, Grey Ballard

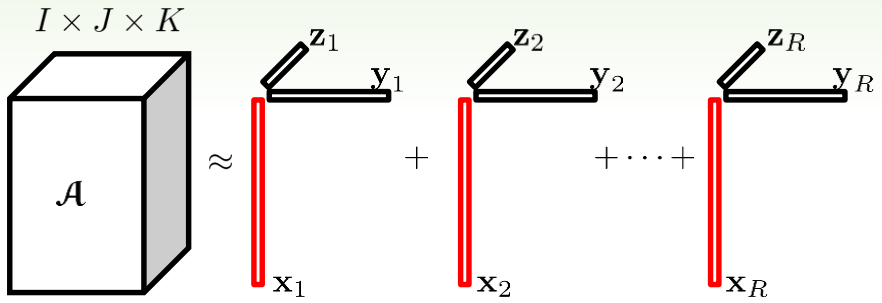
Rewriting CP-ALS Subproblem as Matrix Least Squares

Repeat until convergence:

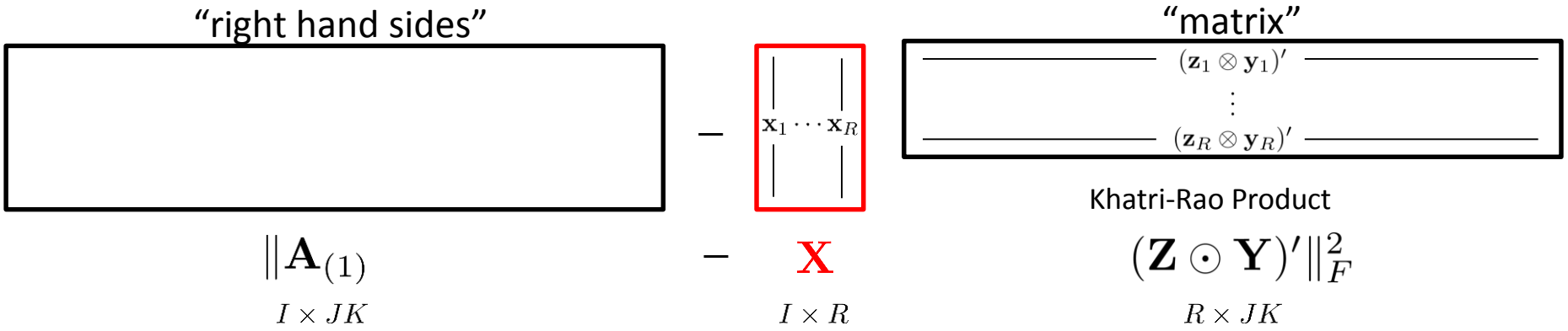
Step 1: $\min_{\mathbf{X}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$

Step 2: $\min_{\mathbf{Y}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$

Step 3: $\min_{\mathbf{Z}} \sum_{ijk} \left(a_{ijk} - \sum_r x_{ir}\, y_{jr}\, z_{kr} \right)^2$

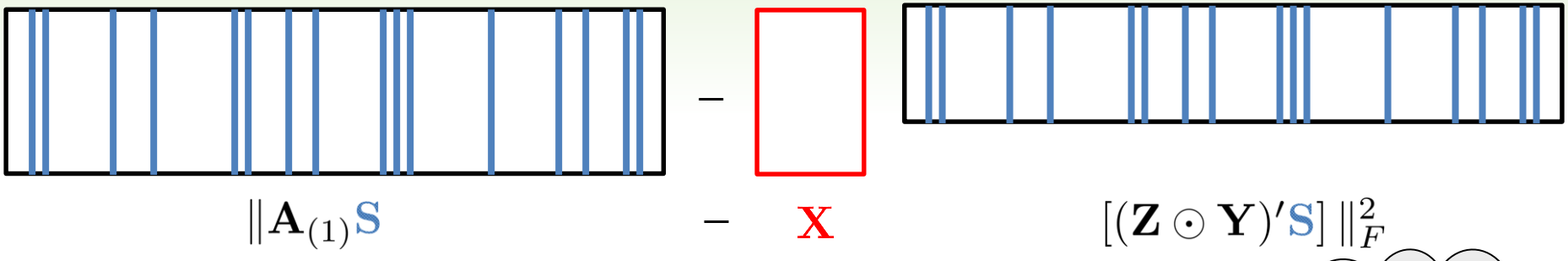


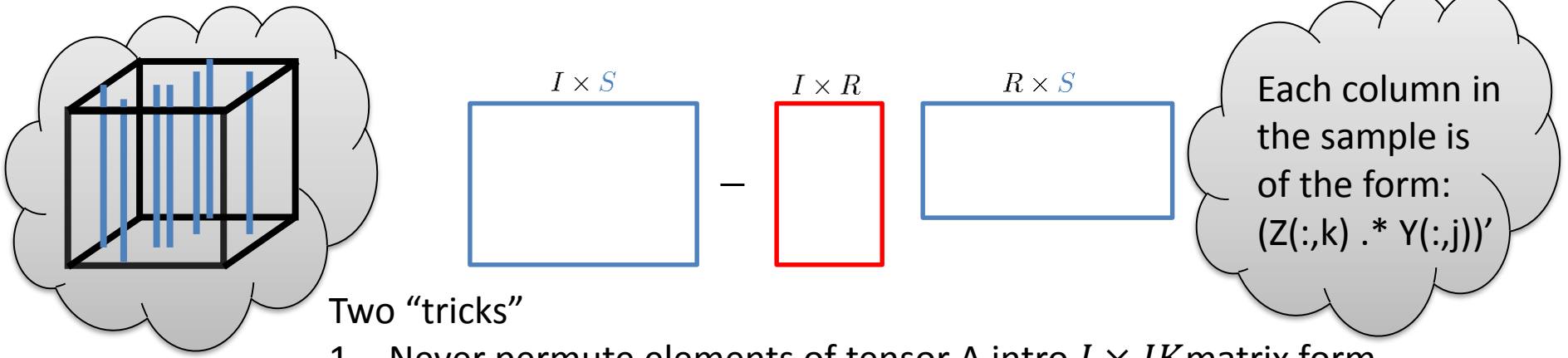
Rewrite as matrix least squares problem
(backwards and transposed matrix version!)



Coming soon: Battaglino, Ballard, Kolda, 2016

CPRAND: Randomizing Matrix Least Squares Subproblem





Two "tricks"

1. Never permute elements of tensor A intro $I \times JK$ matrix form
2. Never form full Khatri-Rao product of size $R \times JK$

CPRAND-MIX: Apply fast Johnson-Lindenstrauss Transform to mix the data in each direction to ensure "incoherence" – introduces some preprocessing cost

Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

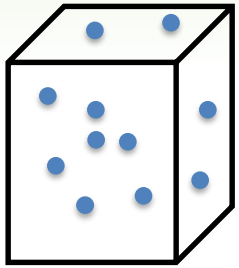
Randomizing the Convergence Check

Repeat until convergence:

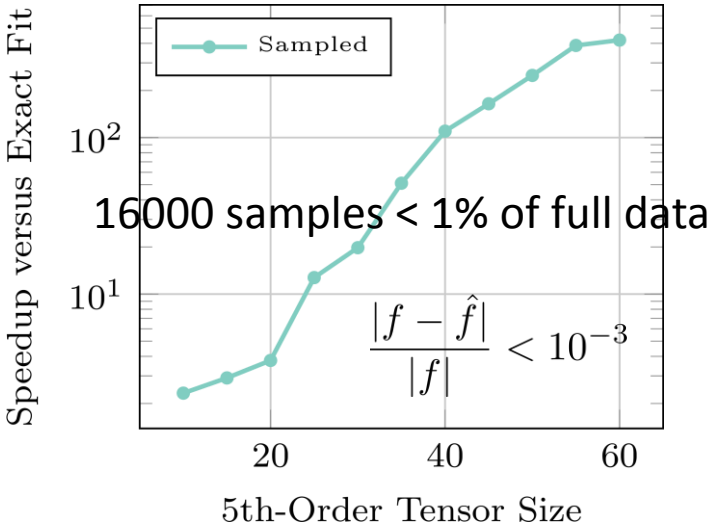
Step 1: $\min_{\mathbf{X}} \sum_{ijk} \left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr} \right)^2$

Step 2: $\min_{\mathbf{Y}} \sum_{ijk} \left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr} \right)^2$

Step 3: $\min_{\mathbf{Z}} \sum_{ijk} \left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr} \right)^2$



Estimate convergence of function values using small random subset of elements in function evaluation (use Chernoff-Hoeffding to bound accuracy)

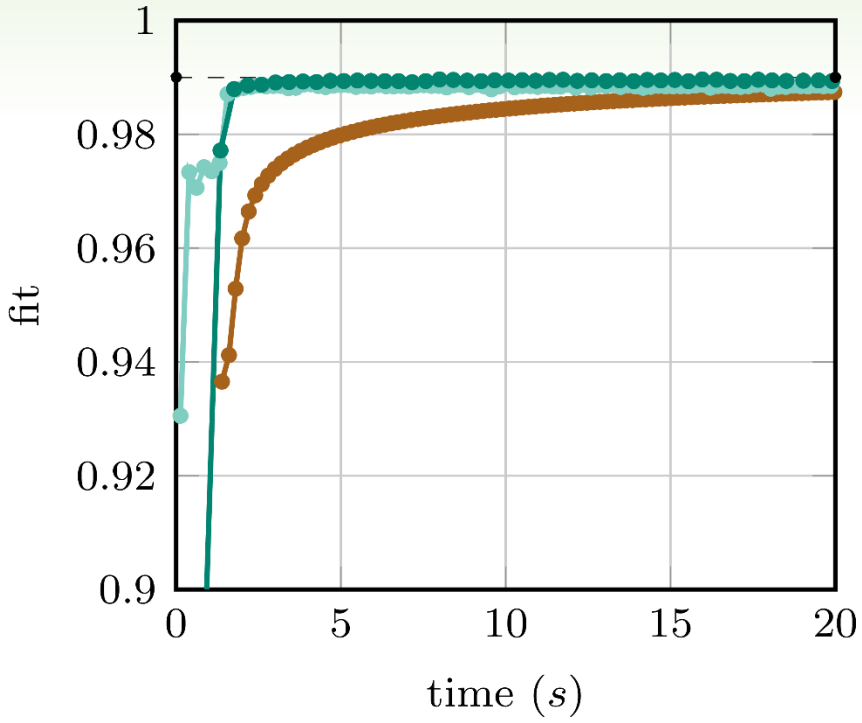


$$f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \sum_{ijk} \left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr} \right)^2$$

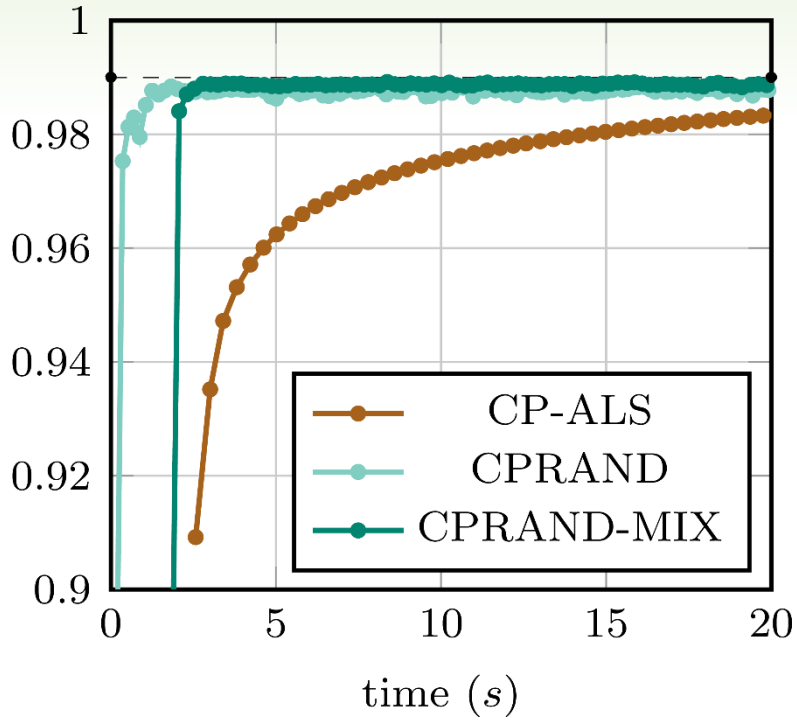
$$\hat{f}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \approx \frac{IJK}{|\Omega|} \sum_{ijk \in \Omega} \left(a_{ijk} - \sum_{r} x_{ir}\, y_{jr}\, z_{kr} \right)^2$$

Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

CPRAND faster than CP-ALS



300 x 300 x 300 Random Rank-5 Tensor with 1% Noise



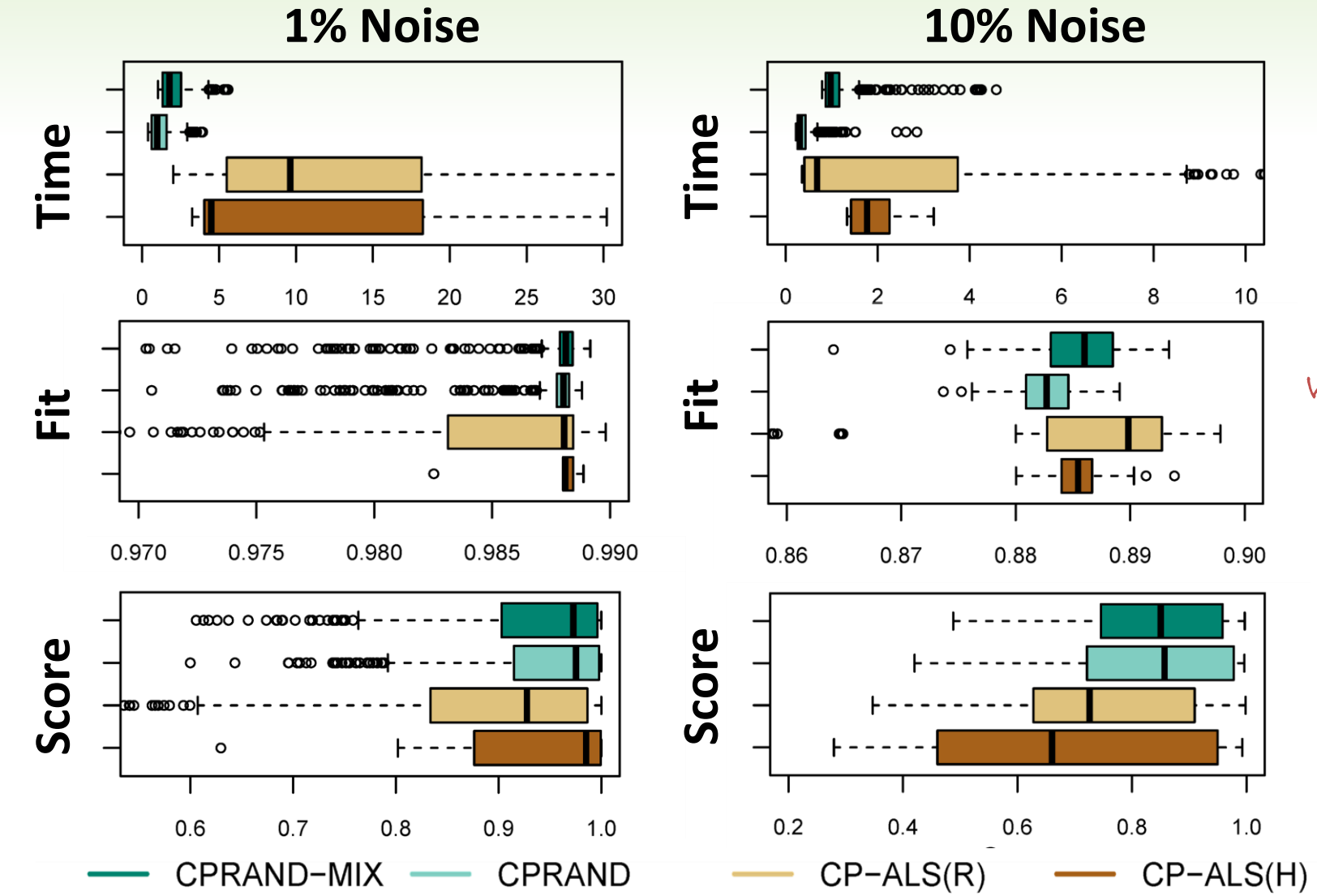
80 x 80 x 80 x 80 Random Rank-5 Tensor with 1% Noise

Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600

CPRAND Faster & More Robust than CP-ALS

200 Tensors: 400 x 400 x 400, R = 5 or 6

Sandia National Laboratories



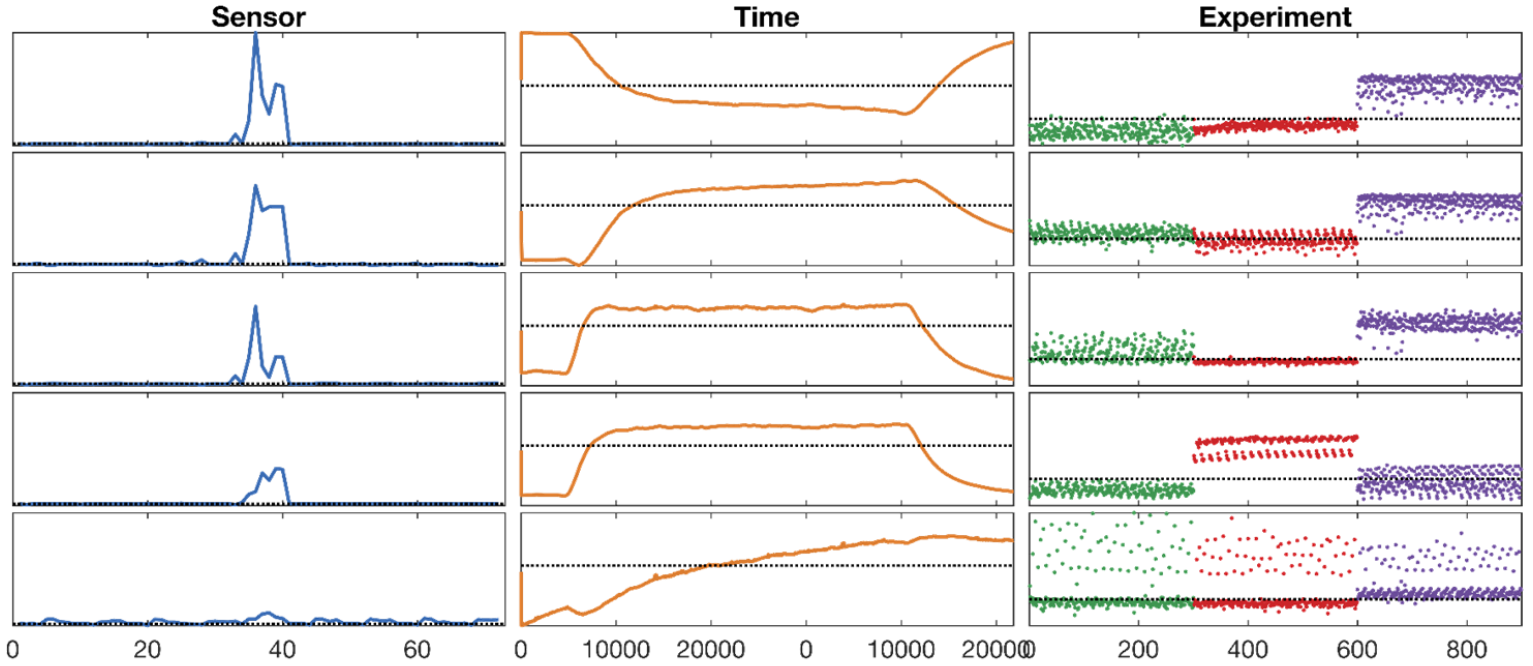
What we can measure

What we really want!

Analysis of Hazardous Gas Data

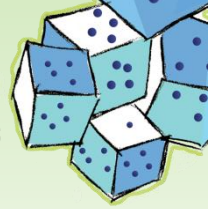
900 experiments (with three different gas types) x 72 sensors x 25,900 time steps (13 GB)

Method	Median Time (s)	Median Fit	Median Classification Error
CPRAND	53.6	0.715	0.61%
CP-ALS (H)	578.4	0.724	0.67%
CP-ALS (R)	204.7	0.724	0.67%



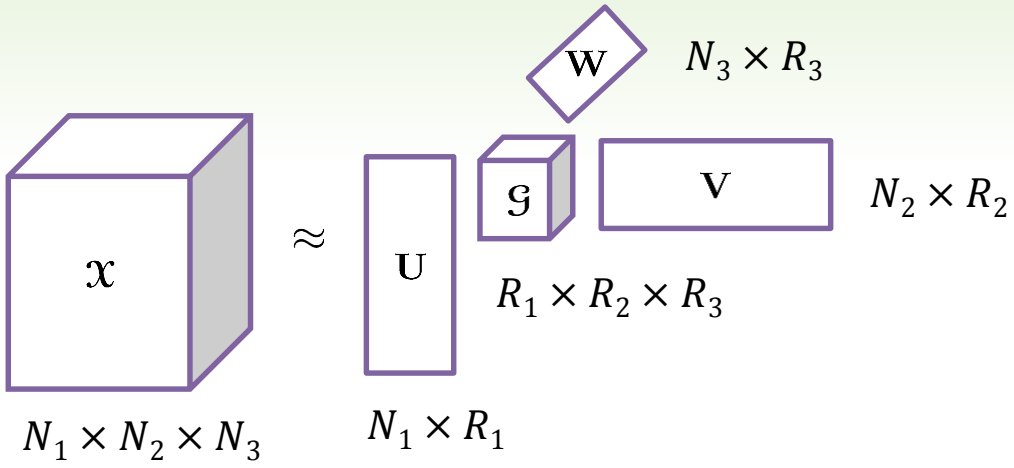
Data from Vergara et al. 2013; see also Vervliet and De Lathauwer (2016)

Battaglino, Ballard, Kolda, *A Practical Randomized CP Tensor Decomposition*, Jan 2017, arXiv:1701.06600



Background on Tucker

Tucker Compression (3-way)



$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

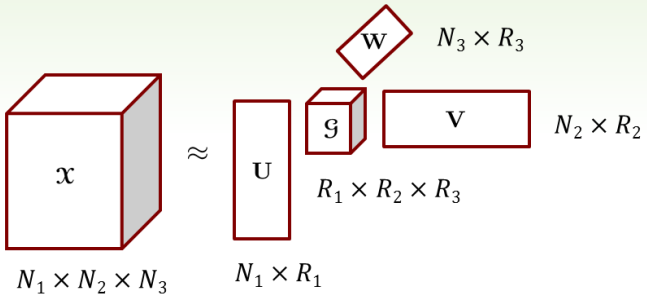
$\mathcal{G}$ = "Core Tensor" = Reduced representation, determined by factor matrices

$\mathbf{U}, \mathbf{V}, \mathbf{W}$ = "Factor Matrices" = Orthogonal matrices spanning high-variance subspaces

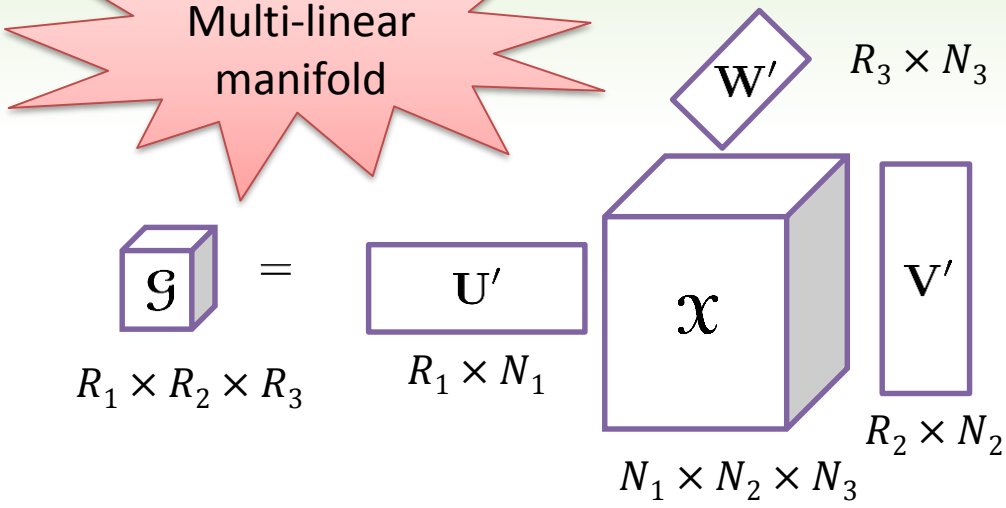
$\times_k$ = Tensor-times-matrix in mode k

- Hitchcock (1927) – Mathematical definition
- Tucker (1966) – Algorithms and applications for 3-way data
- Kapteyn, Neudecker, Wansbeek (1986) – Algorithms for N-way data
- Vannieuwenhoven, Vandebril, and Meerbergen (2012) – Sequentially-truncated algorithm

Choosing Tucker Ranks to Retain Accuracy



Multi-linear manifold



Find orthogonal matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ that reduce the size of tensor but retain its "mass"

For a given relative error ϵ, choose projection ranks R_1, R_2, and R_3 such that:

$$\|\mathcal{X} - (\mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W})\| \leq \epsilon \|\mathcal{X}\|$$

Core tensor satisfies: $\mathcal{G} = \mathcal{X} \times_1 \mathbf{U}' \times_2 \mathbf{V}' \times_3 \mathbf{W}'$

$$\|\mathcal{X}\|^2 - \|\mathcal{G}\|^2 \leq \epsilon^2 \|\mathcal{X}\|^2$$

Tucker Compression (d-way)

$$\underbrace{\mathcal{X}}_{N_1 \times N_2 \cdots \times N_d} \approx \underbrace{\mathcal{G}}_{R_1 \times R_2 \cdots \times R_d} \times_1 \underbrace{\mathbf{U}^{(1)}}_{N_1 \times R_1} \times_2 \underbrace{\mathbf{U}^{(2)}}_{N_2 \times R_2} \cdots \times_d \underbrace{\mathbf{U}^{(d)}}_{N_d \times R_d}$$

$\mathcal{G}$ = "Core Tensor" = Reduced representation, determined by factor matrices

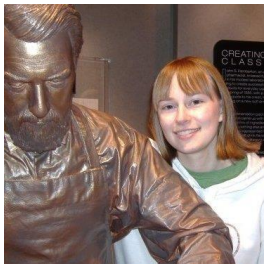
$\mathbf{U}^{(k)}$ = kth "Factor Matrix" = Orthogonal matrix spanning high-variance subspaces

$\times_k$ = Tensor-times-matrix in mode k

Compression Ratio: $C \approx \prod_{k=1}^{d} \frac{N_k}{R_k}$

Sandia National Laboratories

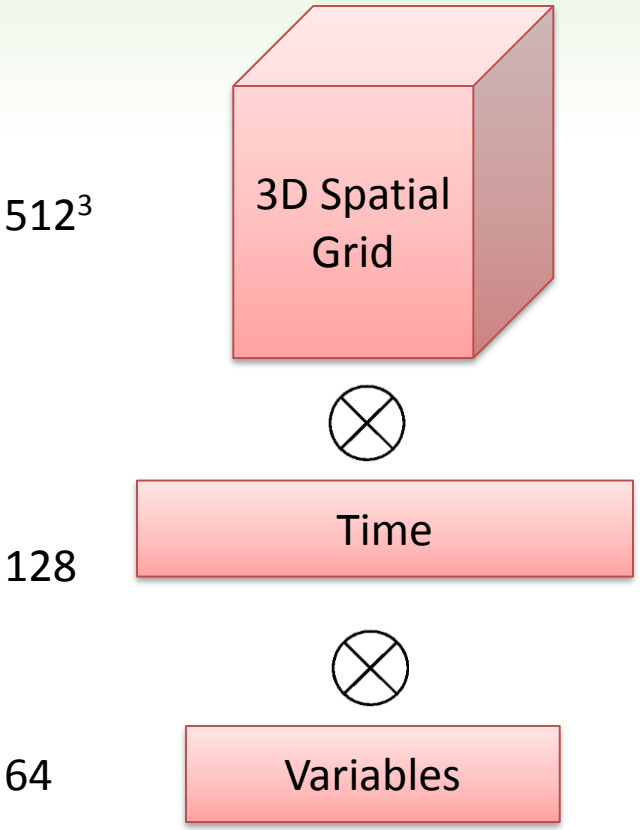





Parallel Tucker Decomposition for Data Compression

Featuring work of Woody Austin, Grey Ballard, Alicia Klinvex, Hemanth Kolla

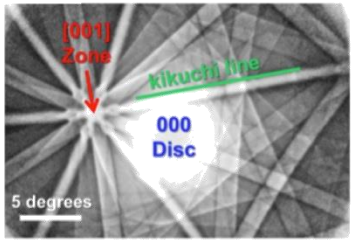
DOE Advanced Simulations and Experiments Deluged by Data

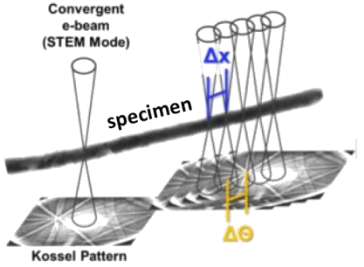


2^{40} elements

8TB (double precision)

- Combustion simulations
  - S3D used direct numerical simulation
  - Gold standard for comparisons, but…
  - Single experiment produces terabytes of data
  - Storage limits spatial, temporal resolutions
  - Difficult to analyze or transfer data
- Electron microscopy
  - New technology produces 1-D spectra and 2-D diffraction patterns *at each pixel*
  - Single experiment produces terabytes of data
  - Usually 10-100 experiments per
  - Limited hardware utilization due to storage limits



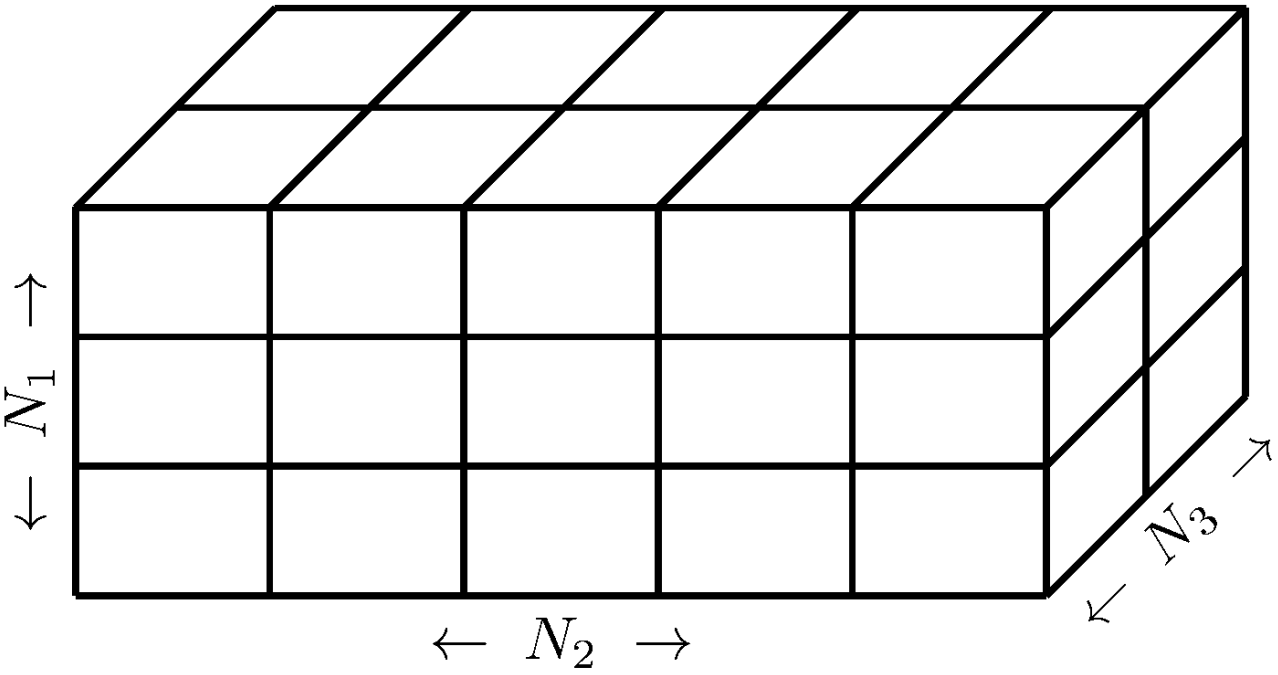


- Other applications
  - Telemetry
  - Cosmology Simulations (with LBL)
  - Climate Modeling (with ORNL)

New Contribution: Parallel Tucker Decomposition

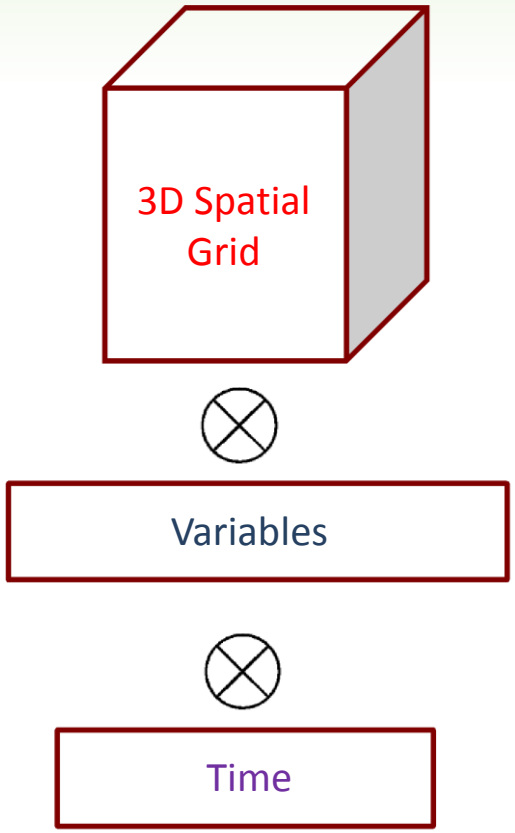
For N-way tensor, Cartesian block distribution on N-way processor grid

$$P_1 \times P_2 \times P_3 = 3 \times 5 \times 2$$



Local block size: $\frac{N_1}{P_1} \times \frac{N_2}{P_2} \times \frac{N_3}{P_3}$

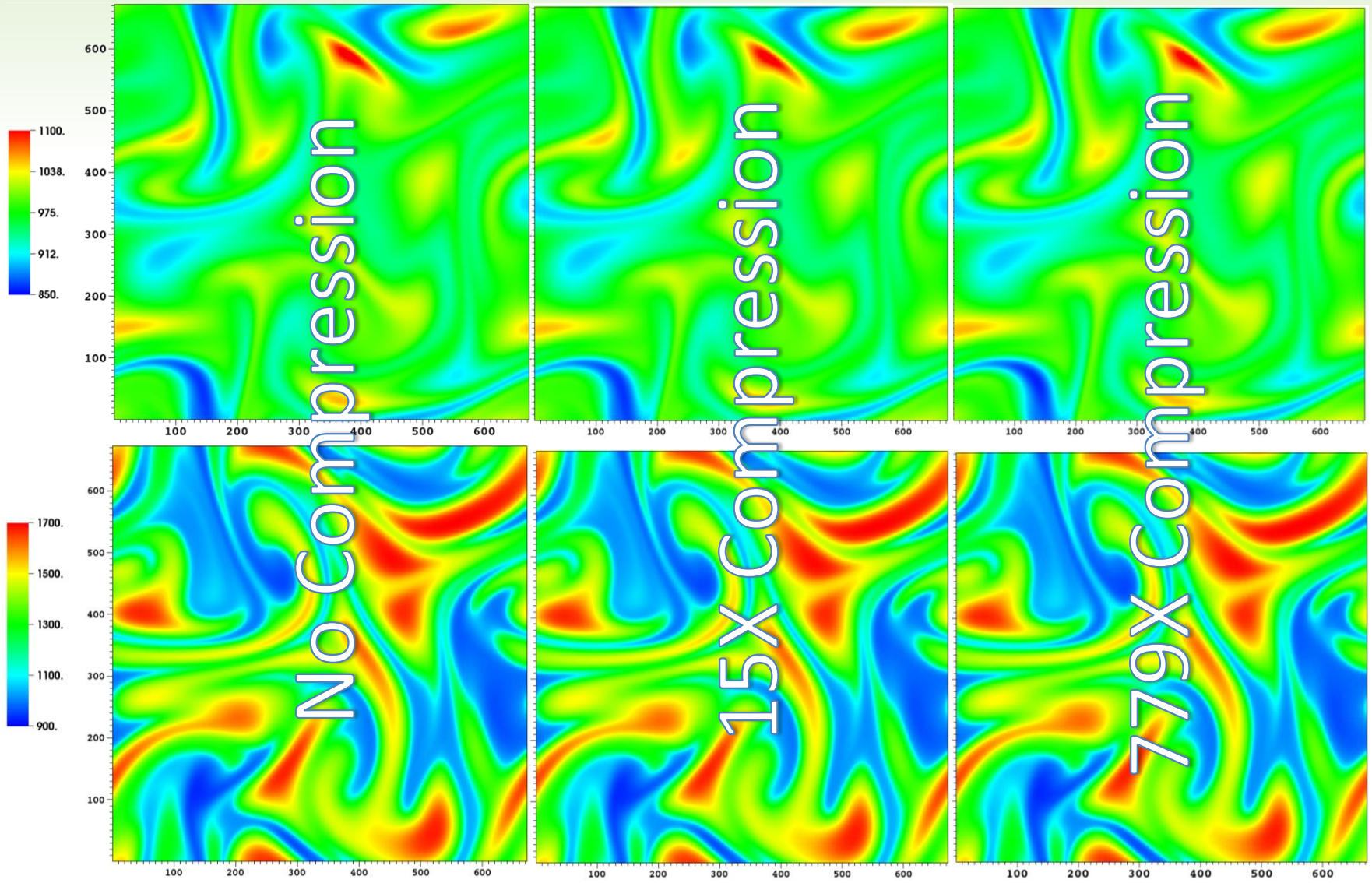
Up to 200000X compression



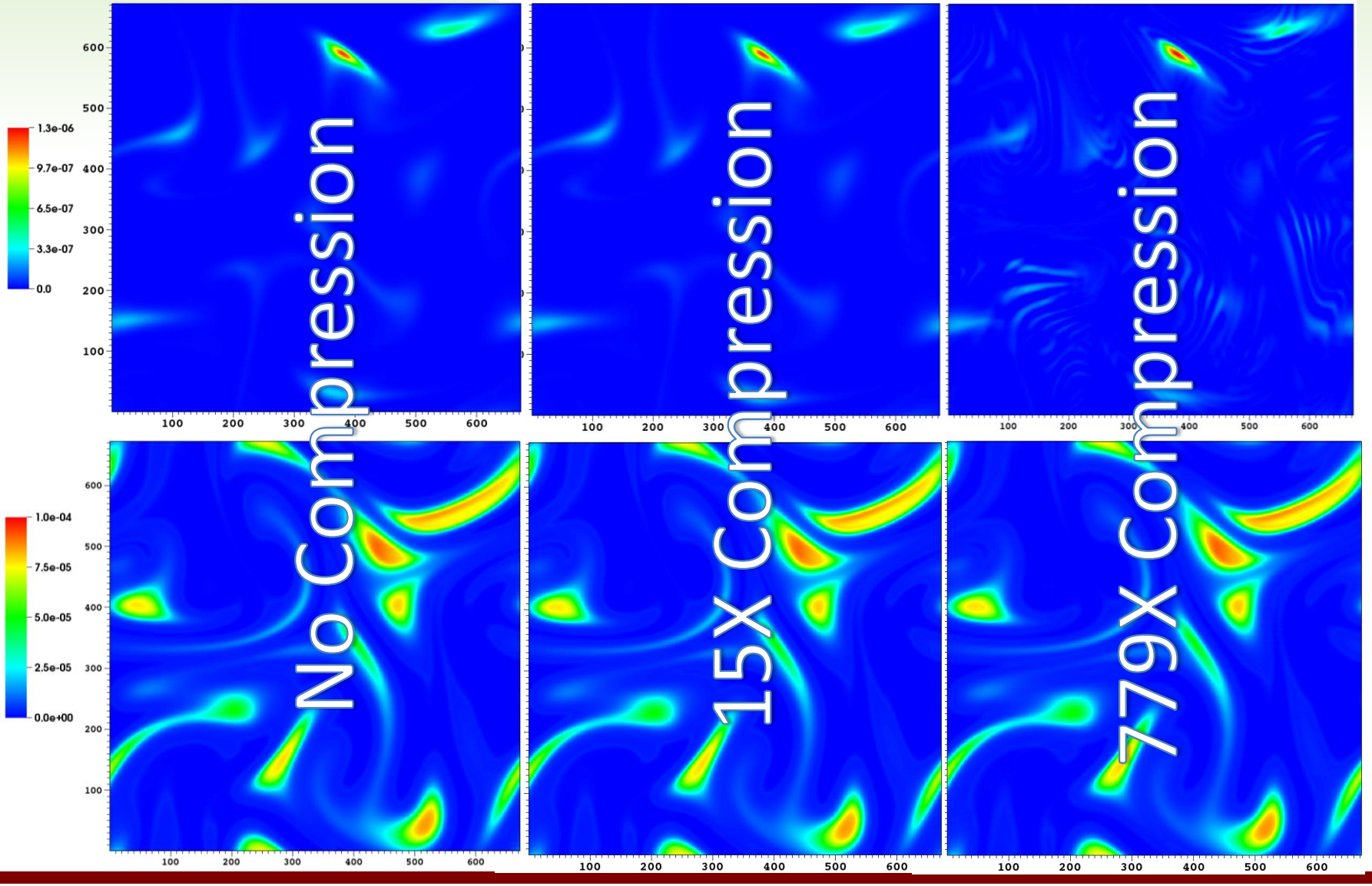
	Original	$\epsilon = 10^{-4}$	$\epsilon = 10^{-2}$
HCCI	672 x 672 x 32 x 626 **67 GB**	330 x 310 x 31 x 199 **(14 X)**	111 x 105 x 22 x 46 **(760 X)**
SP	500 x 500 x 500 x 11 x 400 **4 TB**	95 x 129 x 125 x 7 x 125 **(410 X)**	30 x 38 x 35 x 6 x 11 **(200,000 X)**
JICF	1500 x 2080 x 1500 x 18 x 10 **6 TB**	424 x 387 x 261 x 18 x 10 **(110 X)**	90 x 61 x 48 x 13 x 6 **(40,000 X)**

HCCI Combustion Simulation Results: Temperature @ 2 Times

Sandia National Laboratories

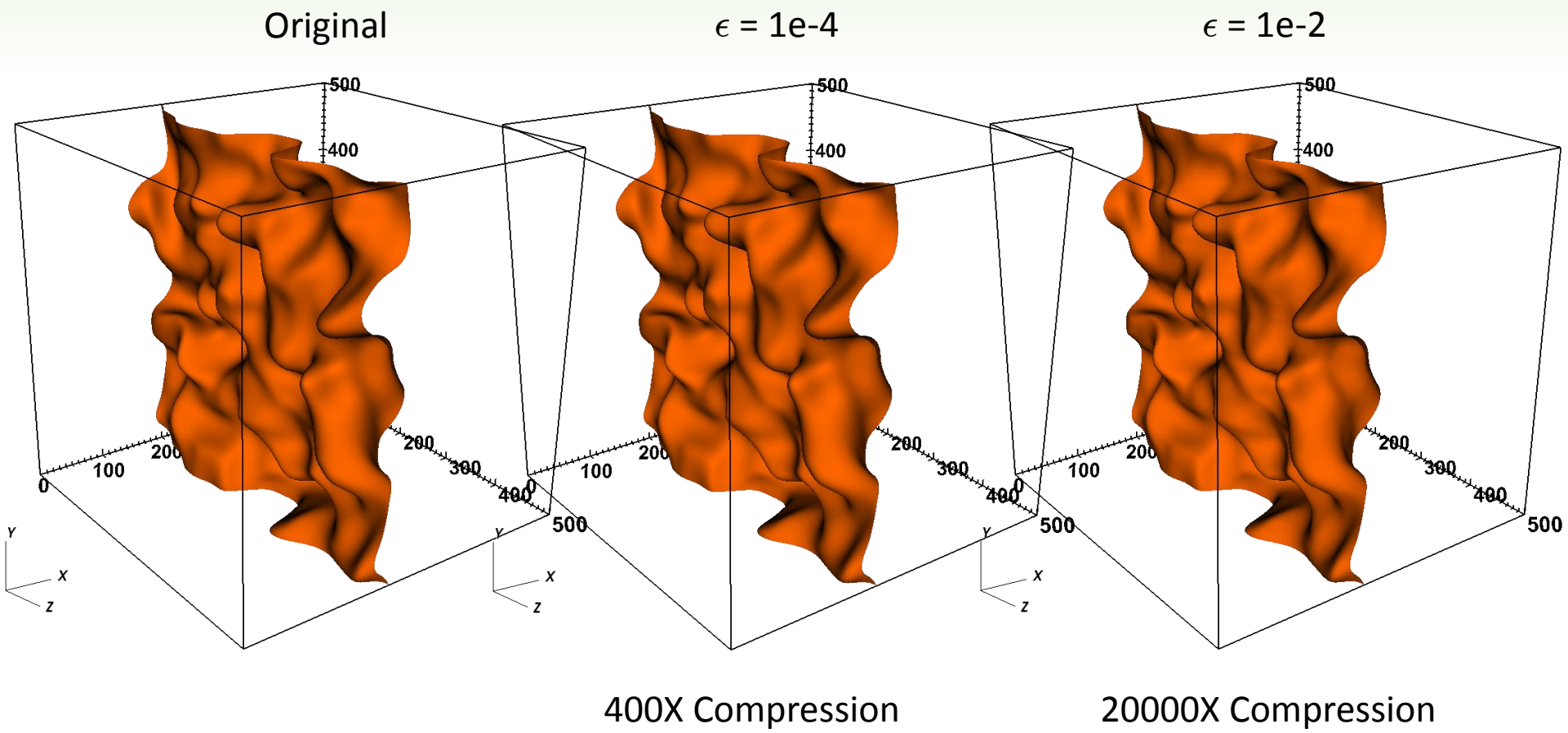


HCCI Combustion Simulation Results: OH @ 2 Times



SP-400 4.4TB Dataset: Can you guess which is the original?

Sandia National Laboratories



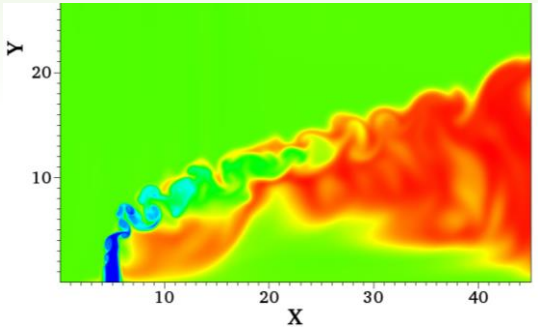
Flame surface at single time step. Using temperature variable (iso-value is 2/3 of max).

JICF: Original vs. Reconstructed

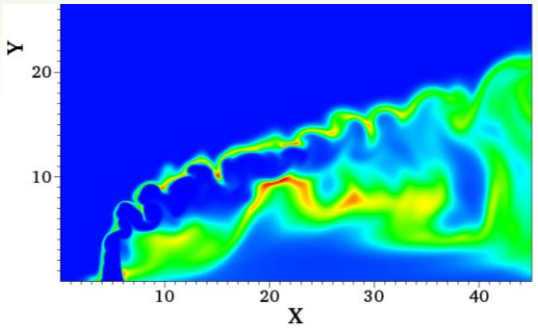
Sandia National Laboratories

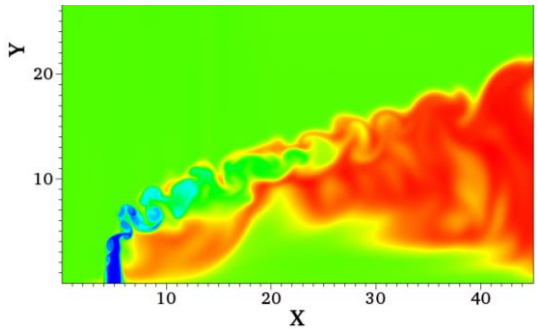
Temperature

OH concentration

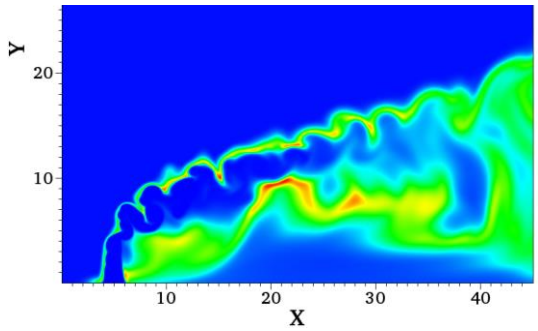


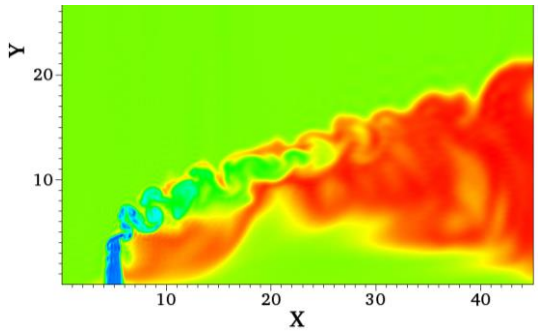
Original



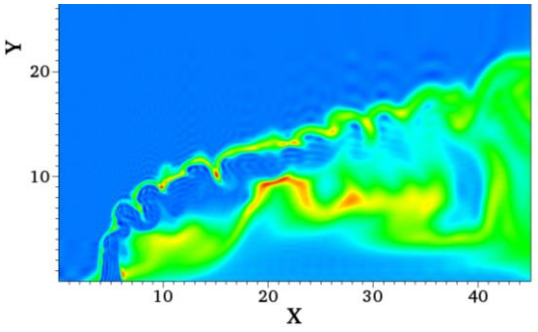


$\epsilon = 10^{-4}$
(110X)



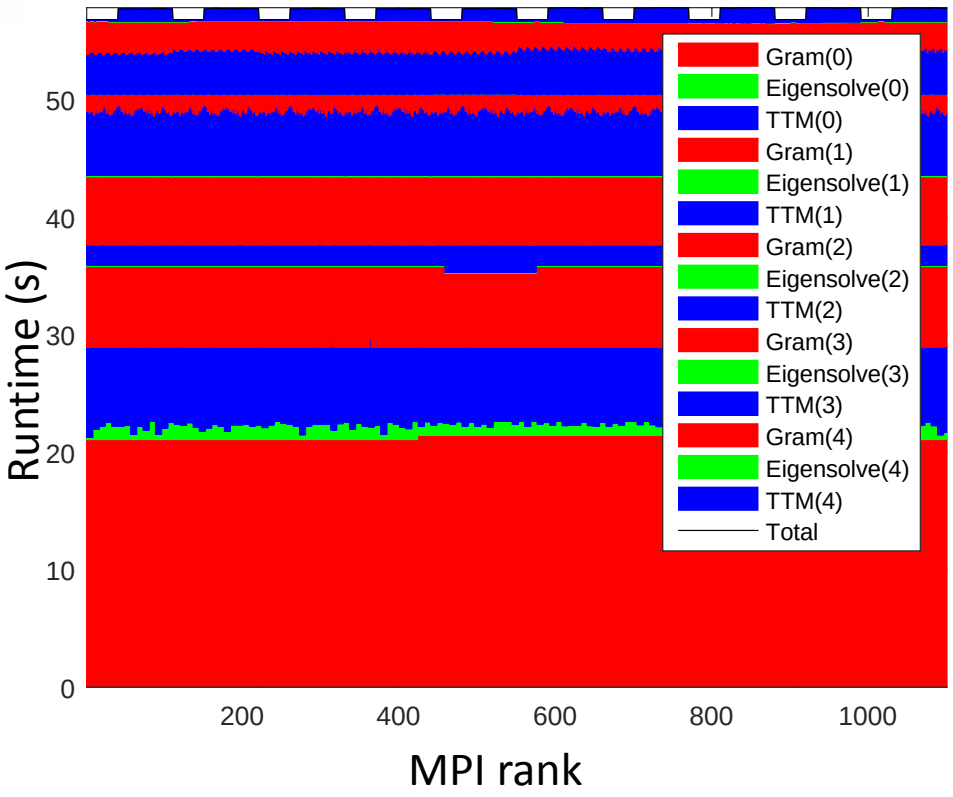


$\epsilon = 10^{-2}$
(400,000X)



- 6 TB (JICF) dataset
- single timestep
- slice along z direction

Parallel TuckerMPI performance



- Total of 55s; 1100 cores.
- 4.4TB -> 10GB (410X).
- Bulk of time is in first mode (GRAM computation).
- **Time for I/O is order of magnitude greater (~450s)**

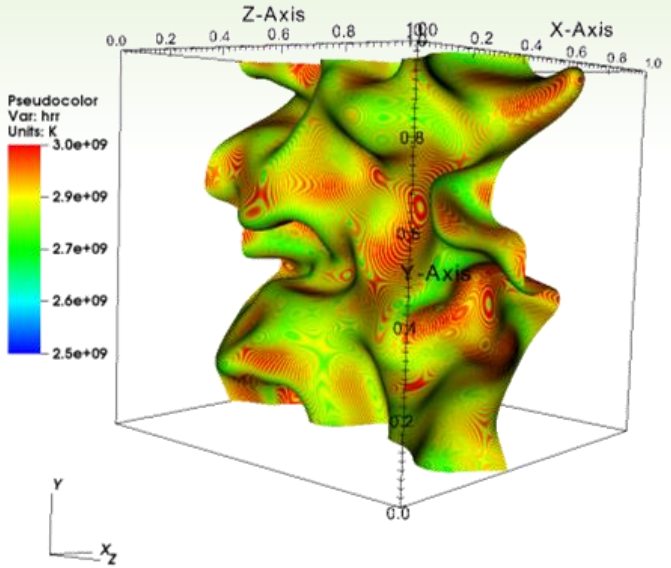
Key Feature: Need Only Do Partial Reconstruction on Laptops, etc.

Sandia National Laboratories

Reconstruction requires as much space as the original data!

$$\hat{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)} \times_5 \mathbf{U}^{(5)}$$

$$N_1 \times N_2 \times N_3 \times N_4 \times N_5$$



But we can just reconstruct the portion that we need at the moment:

$$\bar{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{C}^{(3)}\mathbf{U}^{(3)} \times_4 \mathbf{C}^{(4)}\mathbf{U}^{(4)} \times_5 \mathbf{C}^{(5)}\mathbf{U}^{(5)}$$

$$N_1 \times N_2 \times \frac{N_3}{2} \times 1 \times 1$$

$$\mathbf{C}^{(3)} = \begin{bmatrix} 1/2 & 0 & \cdots & 0 \\ 1/2 & 0 & \cdots & 0 \\ 0 & 1/2 & \cdots & 0 \\ 0 & 1/2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

Downsample

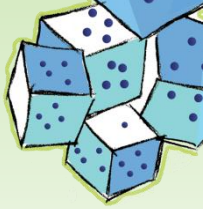
$$\mathbf{C}^{(4)} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

Pick single variable

$$\mathbf{C}^{(5)} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

Pick single time step

JICF Partial Reconstruction



- Reconstructing a single variable at a single timestep along a slice of the z axis (*24 MB*)
- Ran on a single node with 128 GB of RAM

Original data size	6 TB	
Error	1e-2	1e-4
Core tensor size	156 MB	57 GB
Read time	.1 s	139 s
Reconstruct time	.1 s	21 s
Reconstruct max memory	185 MB	58 GB
Reconstructed data size	24 MB	

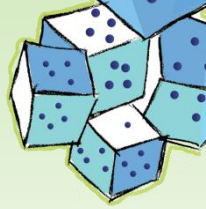
Software: TuckerMPI

`git@gitlab.com:tensors/TuckerMPI.git`

Alicia Klinvex, Woody Austin, Grey Ballard, Hemanth Kolla, Tammy Kolda

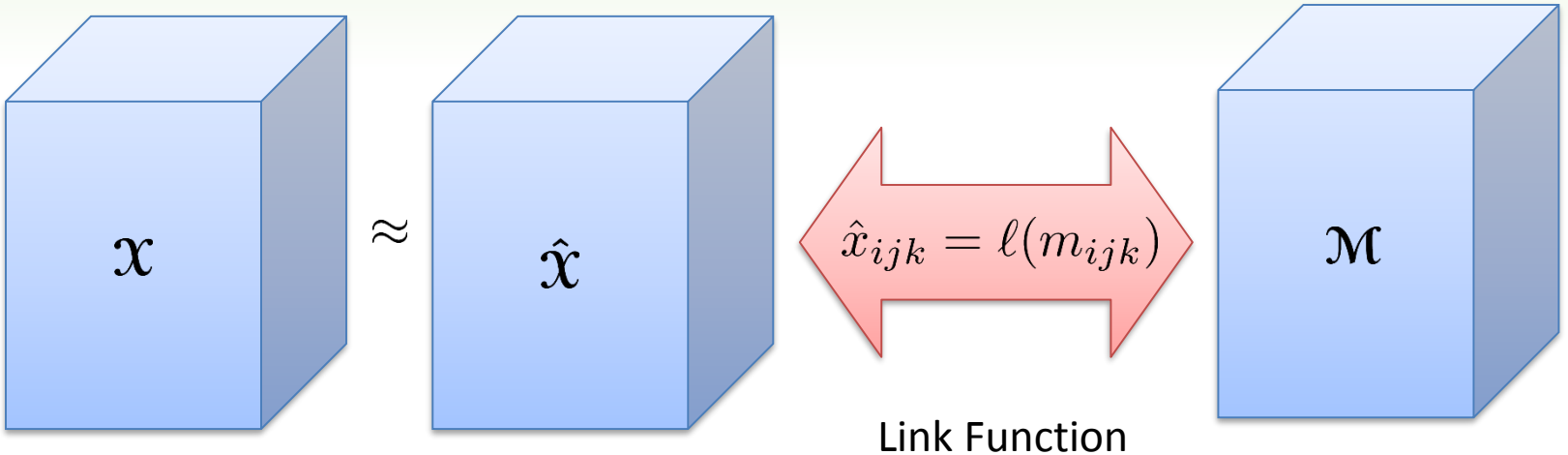
- Open source code for computing Tucker compression
- MPI/BLAS/LAPACK/C++11
- Still in development but available for testing
- Looking for new applications and users
- Interested in partnering
- Reference paper: W. Austin, G. Ballard, and T. G. Kolda, ***Parallel Tensor Compression for Large-Scale Scientific Data***, IPDPS'16 (arXiv:1510.06689)

Sandia National Laboratories



Model Fitting & Binary Data

Data, Estimates, Models, and Loss Functions



Data

Estimate
(expected value)

Link Function

Model
(low rank)

Loss function: $F(\mathcal{X}, \mathcal{M}) = \sum_{ijk} f(x_{ijk}, m_{ijk})$ (sum of elementwise functions)

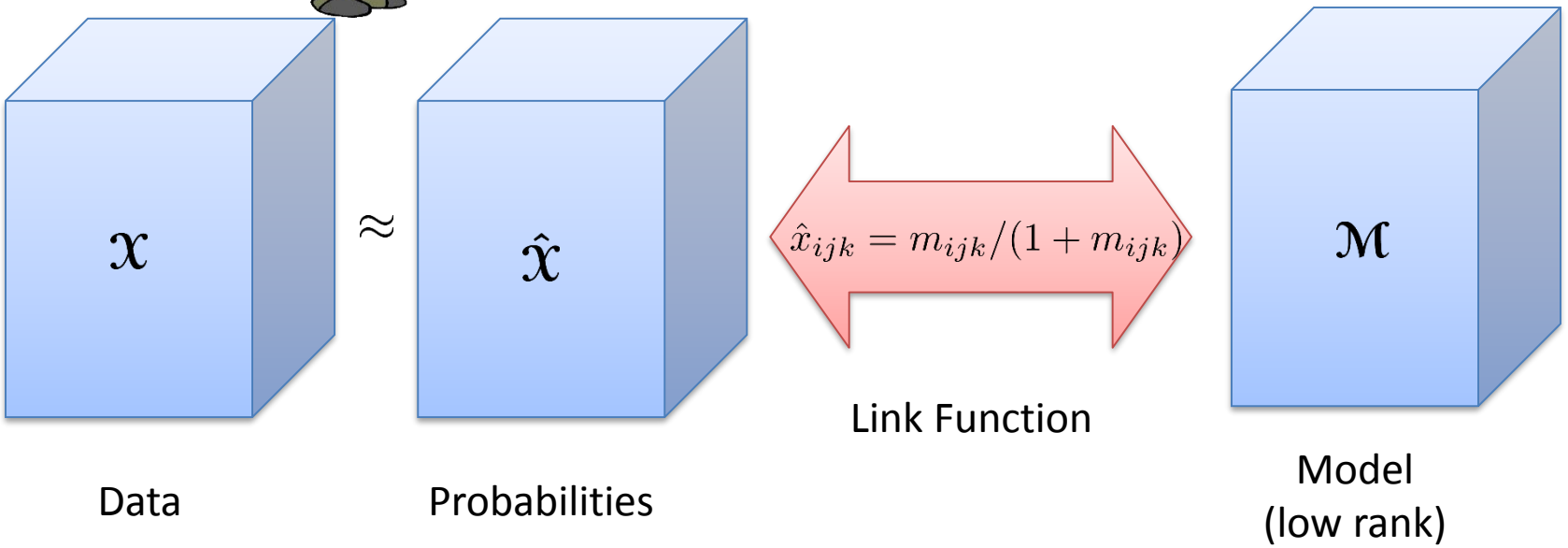
$$x_{ijk} \sim \mathcal{N}(m_{ijk}, \sigma)$$

$$f(x, m) = (x - m)^2$$

Loss Function for Bernoulli Data

Bernoulli Probability Mass Function (PMF)

$$p^x(1-p)^{(1-x)},\ p \in (0,1)$$



Data

Probabilities

Link Function

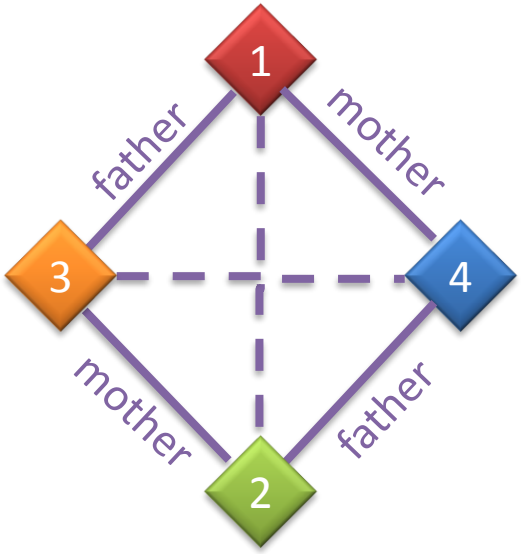
Model
(low rank)

$$x_{ijk} \sim \text{Bernoulli}(m_{ijk}/(1+m_{ijk})),\ m_{ijk} \geq 0$$

$$f(x,m) = \log(m+1) - x\log m$$

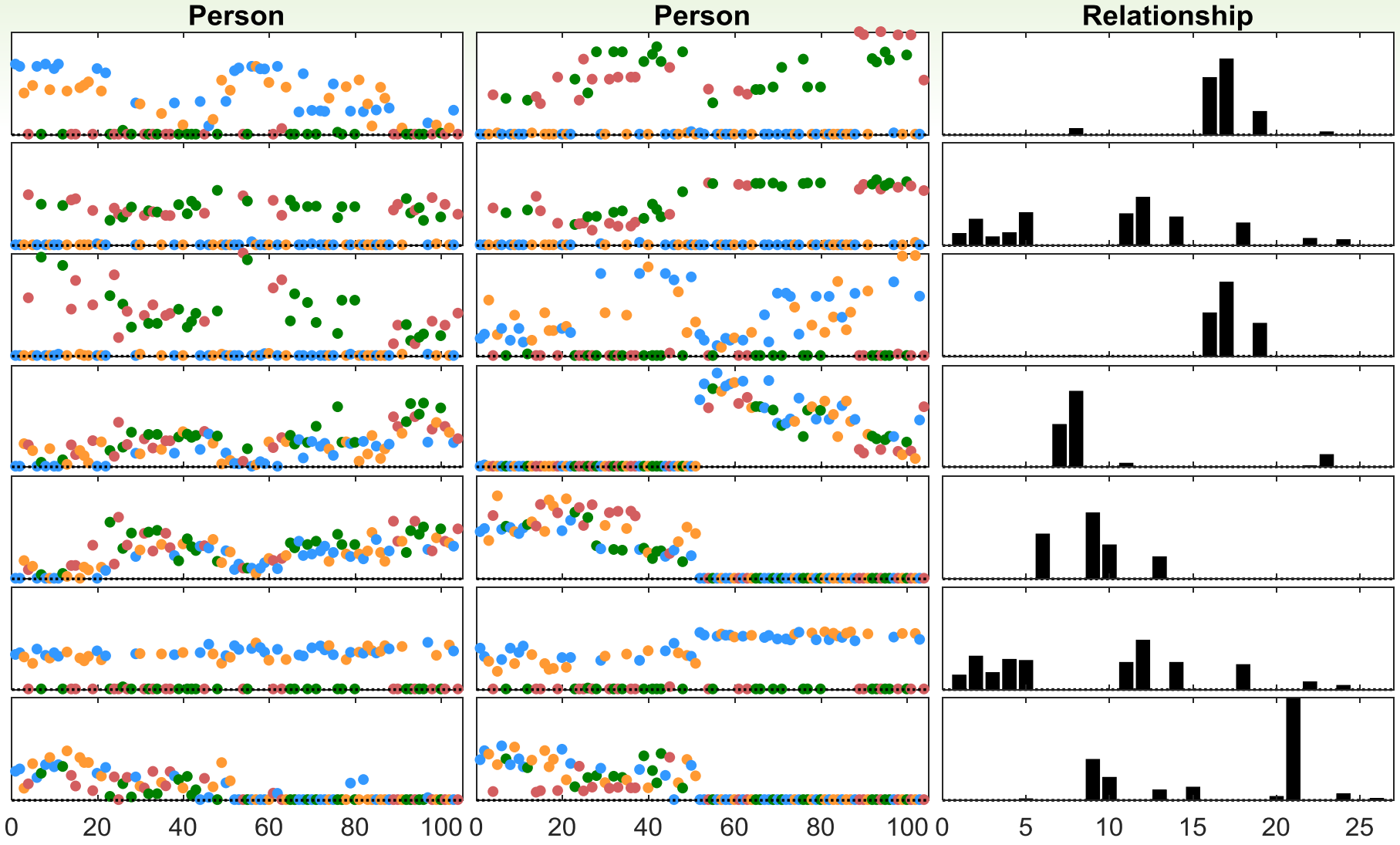
Preliminary Analysis: Kinship Data

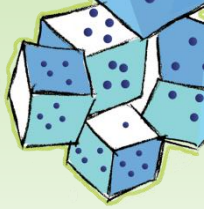
- Australian tribe
- 104 persons
- 4 sections
- 26 kinship terms



- Kinship Terms
  - Complex relationships having to do with sections, gender, and age
  - Example: *Adiadya* – Younger person in same section
- Citations
  - Denham, PhD Thesis, 1973
  - Kemp, Tenenbaum, Griffiths, Yamada, Ueda, *Learning Systems of Concepts with an Infinite Relational Model*, AAAI-06, 2006
  - Nickel, Tresp and Kriegel, *A three-way model for collective learning on multi-relational data,* ICML-11, 2011

7-Component Results

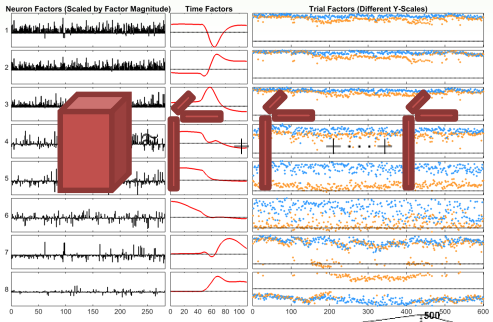


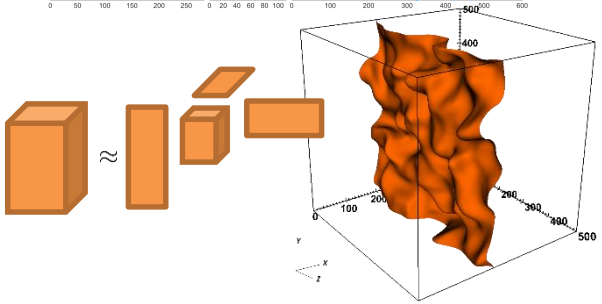


Wrapping Up…

Tensors are a Foundational Tool for Data Analysis

- Tensor decomposition extends matrix factorization
  - SVD, PCA, NMF, etc.
- Useful for data analysis
  - Mouse neuron activity
  - CFD simulation data compression
  - Kinship data
- Choice of decompositions
  - Canonical polyadic (CP) decomposition
  - Tucker decomposition
  - More!
- Computational & algorithmic advances
  - Randomization
  - Parallelization
  - Choice of fitting function





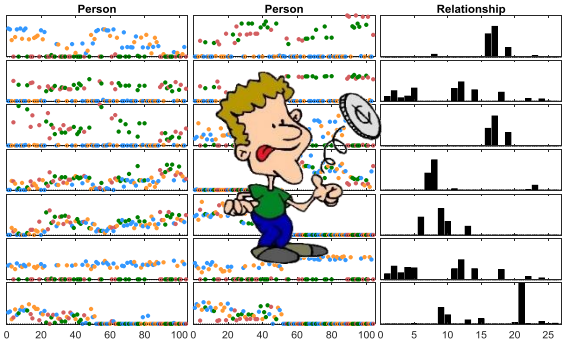
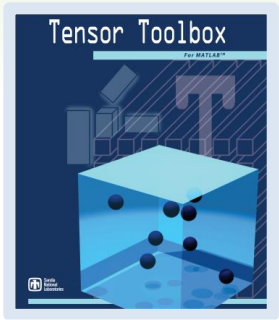


Illustration by Chris Brigman

References, Codes, Etc.

- **Overview**
  - T. G. Kolda and B. W. Bader, ***Tensor Decompositions and Applications***, SIAM Review, Vol. 51, No. 3, pp. 455-500, September 2009, doi:10.1137/07070111X
  - B. W. Bader and T. G. Kolda, ***Efficient MATLAB Computations with Sparse and Factored Tensors***, SIAM Journal on Scientific Computing, Vol. 30, No. 1, pp. 205-231, December 2007, doi:10.1137/060676489
- **Randomization for CP**
  - C. Battaglino, G. Ballard, and T. G. Kolda, ***A Practical Randomized CP Tensor Decomposition***, January 2017, arXiv:1701.06600
- **Parallel Tucker for Compression**
  - W. Austin, G. Ballard and T. G. Kolda, ***Parallel Tensor Compression for Large-Scale Scientific Data***, IPDPS'16: Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium, pp. 912-922, May 2016, doi:10.1109/IPDPS.2016.67, arXiv:1510.06689
- **Codes**
  - Tensor Toolbox for MATLAB: http://www.sandia.gov/~tgkolda/TensorToolbox/
  - TuckerMPI: https://gitlab.com/tensors/TuckerMPI
- **POC**
  - Tammy Kolda, tgkolda@sandia.gov

Kolda and Bader, Tensor Decompositions and Applications, *SIAM Review* '09



Tensor Toolbox for MATLAB: Bader, Kolda, Acar, Dunlavy, and others

TUCKERMPI

https://gitlab.com/tensors/TuckerMPI
Klinvex, Austin, Ballard, Kolda