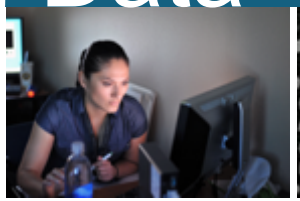
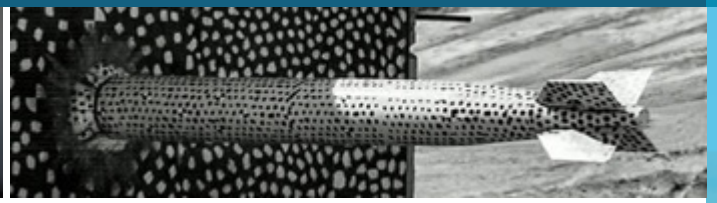
SAND2023-02697C

Sandia National Laboratories

Generalized Canonical Polyadic Tensor Decompositions for Streaming Data





PASC 23, June 26-28, 2023

Eric Phipps, Nick Johnson, Tammy Kolda

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2023-02697C

Multiway “Tensor” Data is Ubiquitous

Neuron activity:
Neuron x Time x Trial

Travel data: Start Location
x Finish Location
x Departure Hour
x Departure Date

Crime data: Crime x Location
x Hour x Date

Signal processing:
Sensor x Frequency x Time

Social interaction data:
Person A x Person B
x Venue x Time

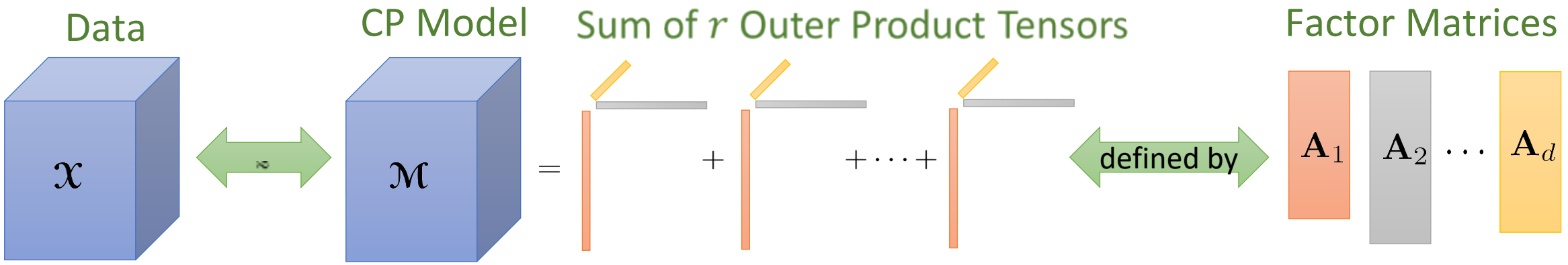
Cyber data: Src IP x Dst IP
x Dst Port x Time

Twitter co-occurrence:
Term A x Term B x Time

Host data: Host
x Action/Library x Time

Tensor Decomposition Finds Patterns in Multiway Data

Canonical Polyadic (CP) Tensor Decomposition



$$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \qquad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!] \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \qquad \mathbf{A}_k \in \mathbb{R}^{n_k \times r}$$

$$x(i_1, \ldots, i_d) \approx m(i_1, \ldots, i_d) = \sum_{j=1}^{r} \prod_{k=1}^{d} a_k(i_k, j) = \sum_{j=1}^{r} \lambda_j \prod_{k=1}^{d} \hat{a}_k(i_k, j)$$

Model Rank

$$\hat{a}_k(:, j) = a_k(:, j) / \|a_k(:, j)\|, \quad \lambda_j = \prod_{k=1}^{d} \|a_k(:, j)\|$$

Order sum with decreasing λ

Vectors comprising each factor matrix are interpretable!

- They each contribute to model *M* in a simple way
- Thus they describe patterns in the data

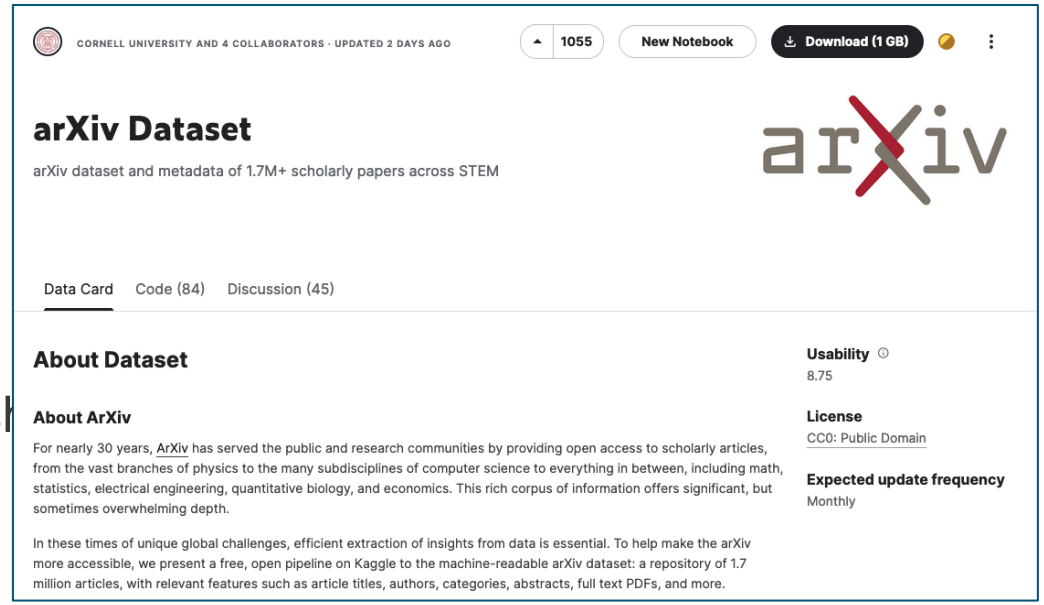
ArXiv Abstract Data

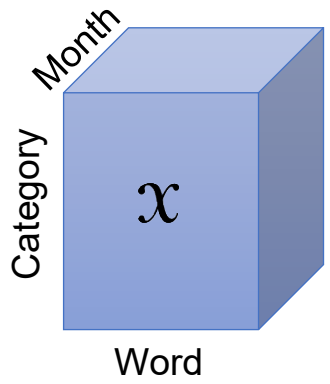
Kaggle provides metadata for ArXiv abstracts that is regularly updated
- https://www.kaggle.com/datasets/Cornell-University/arxiv
- Publication date, going back to first paper backdated to 1986
- Paper category (e.g., cs.LG:Machine Learning)
- Abstracts text

Process abstracts into unique words using Spacy's English core parser
- https://spacy.io/
- Resulted in ~25K unique words

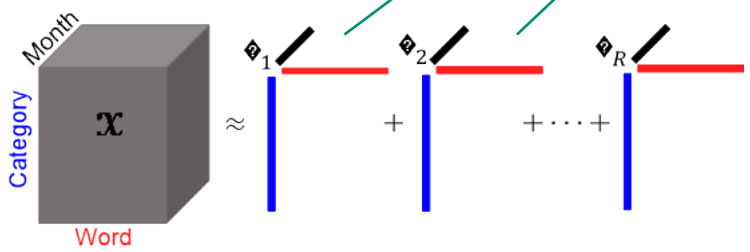
For each month, construct 2-way tensor (matrix):
- Mode 1: Paper category
- Mode 2: Word
- Value: Number of abstracts of the given category containing the given word

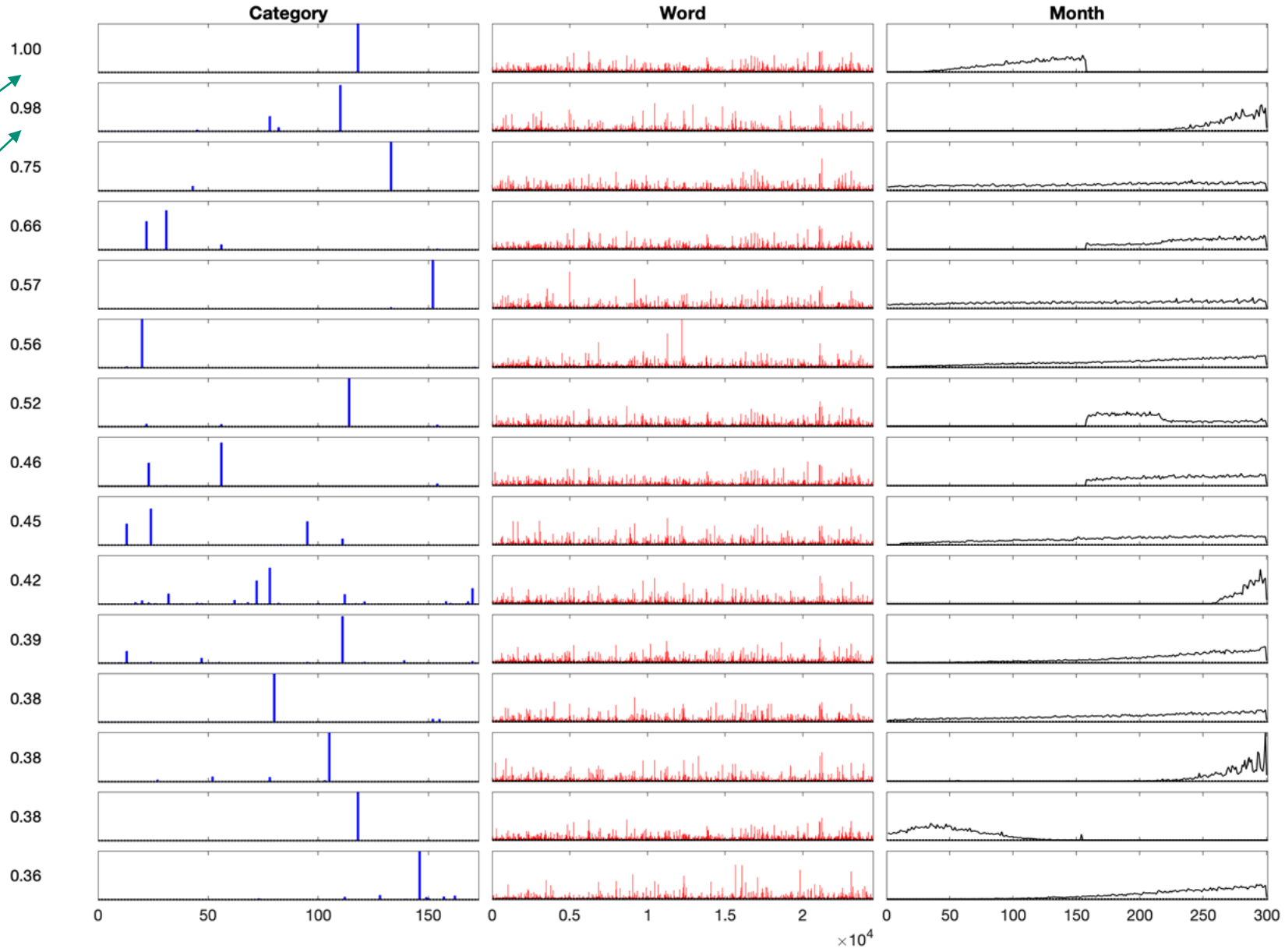
Collection of matrices for each month since start-date forms a sparse 3-way tensor
- Start date is 10 years after first paper (1986) to allow for sufficient volume
- 172 × 24558 × 300, ~2.4% sparsity





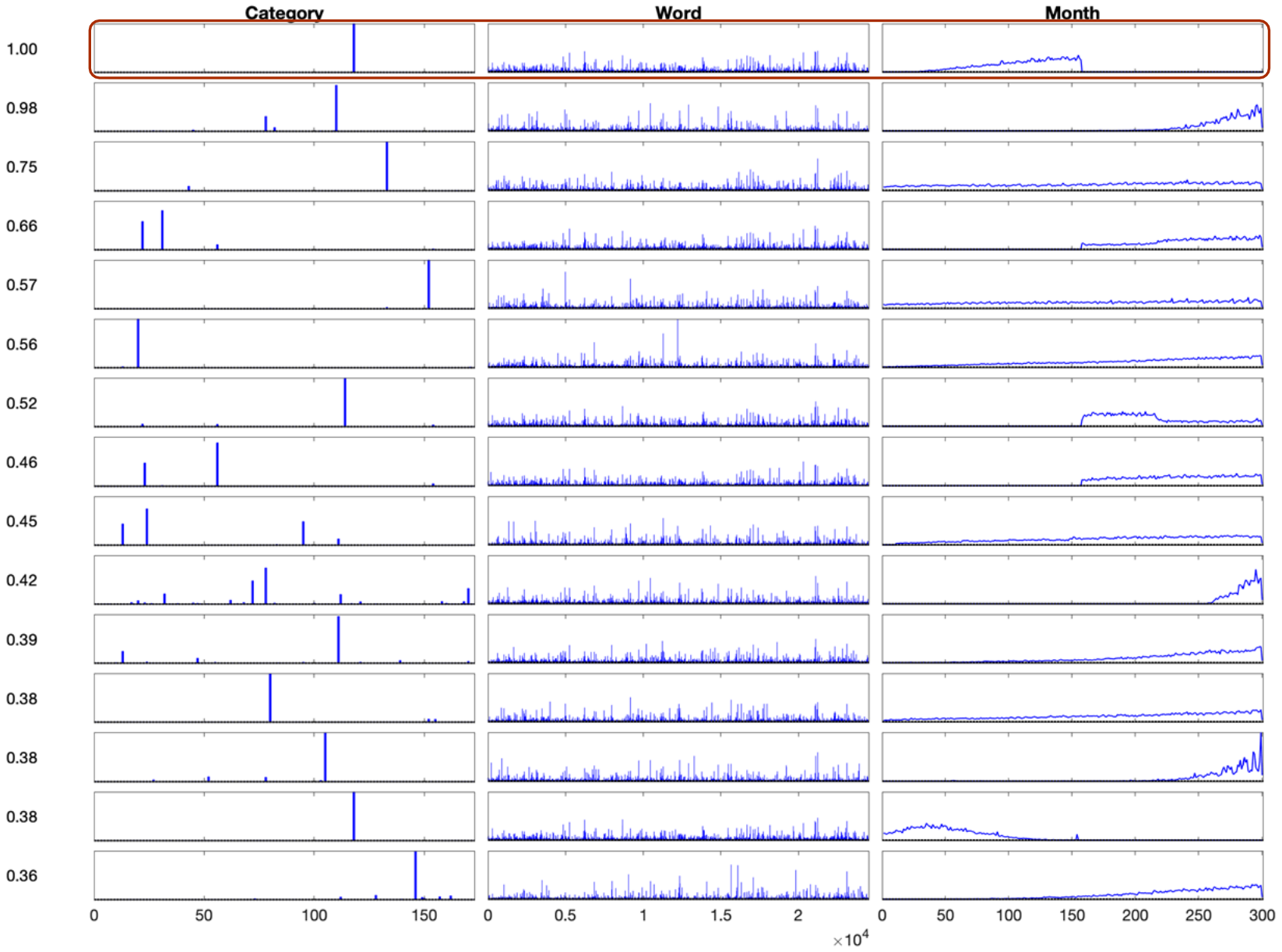
Top 15 (of 50) ArXiv CP Factors





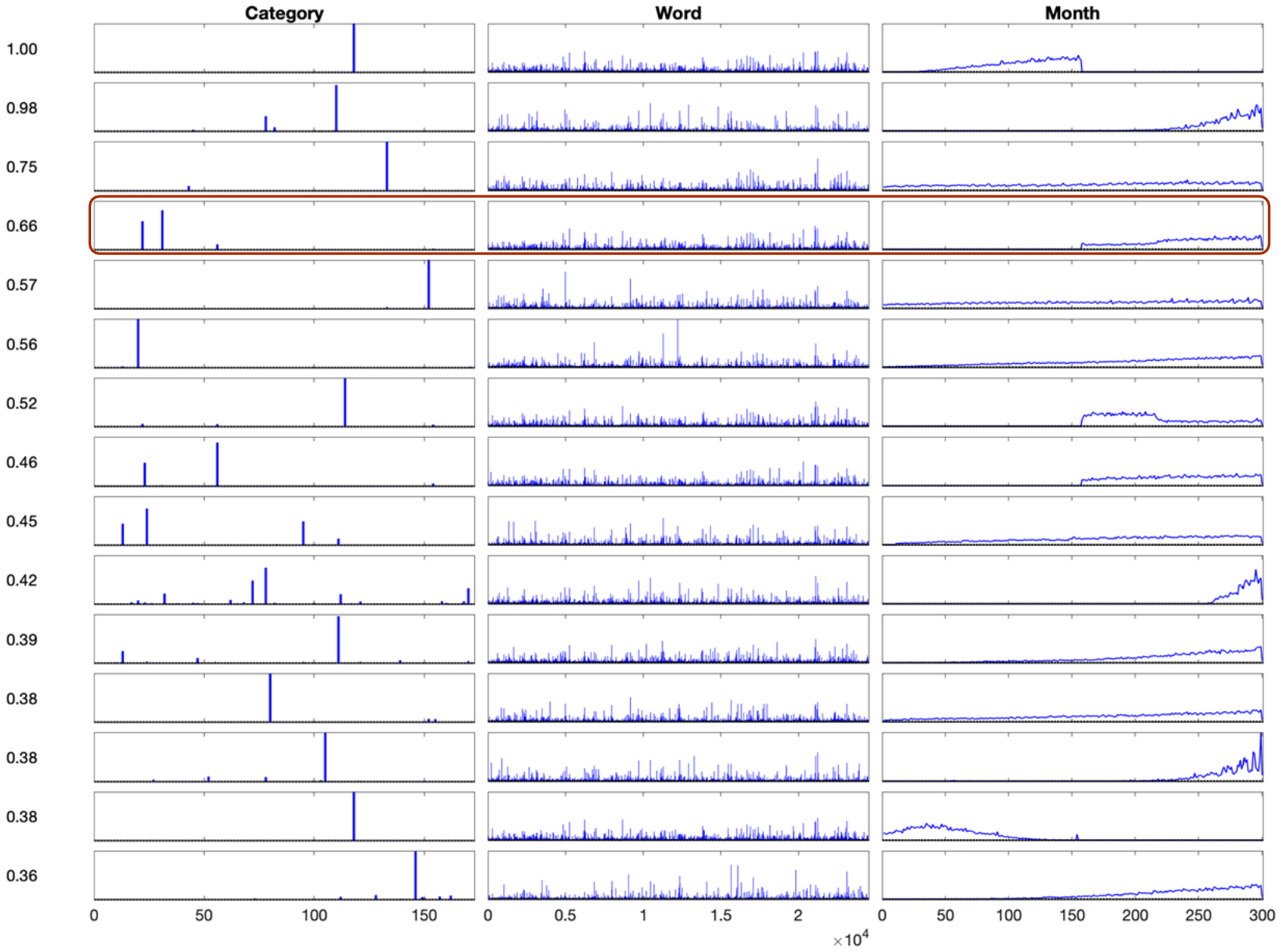
Component #1 (of 50): Astrophysics

Top Categories	Category Weight	Top Words	Word Weight
astro-ph	1.00	model	1.00
		present	0.98
		find	0.97
		use	0.94
		result	0.92
		star	0.92
		high	0.91
		observation	0.79
		galaxy	0.78
		large	0.77
		observe	0.73
		mass	0.72
		datum	0.70
		low	0.69
		study	0.67
		ray	0.64
		spectrum	0.62
		time	0.62
		field	0.60
		emission	0.60



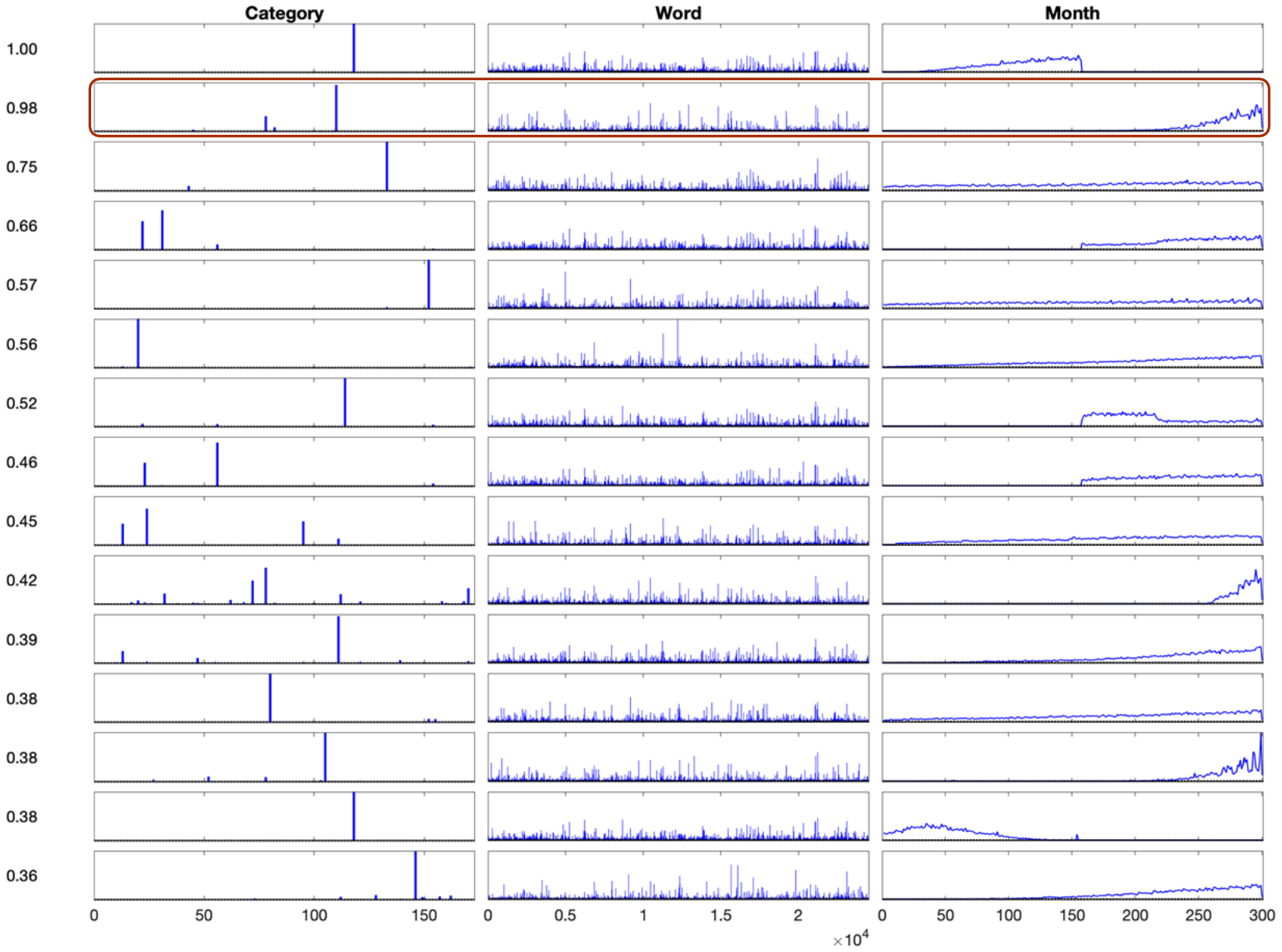
Component #4 (of 50): Astrophysics Reorganized

Top Categories	Category Weight	Top Words	Word Weight
astro-ph.GA	1.00	find	1.00
astro-ph.HE	0.72	model	0.88
astro-ph.SR	0.13	high	0.88
astro-ph.IM	0.02	star	0.85
		use	0.83
		galaxy	0.78
		result	0.74
		present	0.73
		observation	0.72
		mass	0.71
		observe	0.68
		large	0.68
		emission	0.67
		ray	0.67
		study	0.67
		low	0.64
		datum	0.60
		source	0.56
		energy	0.54
		time	0.54



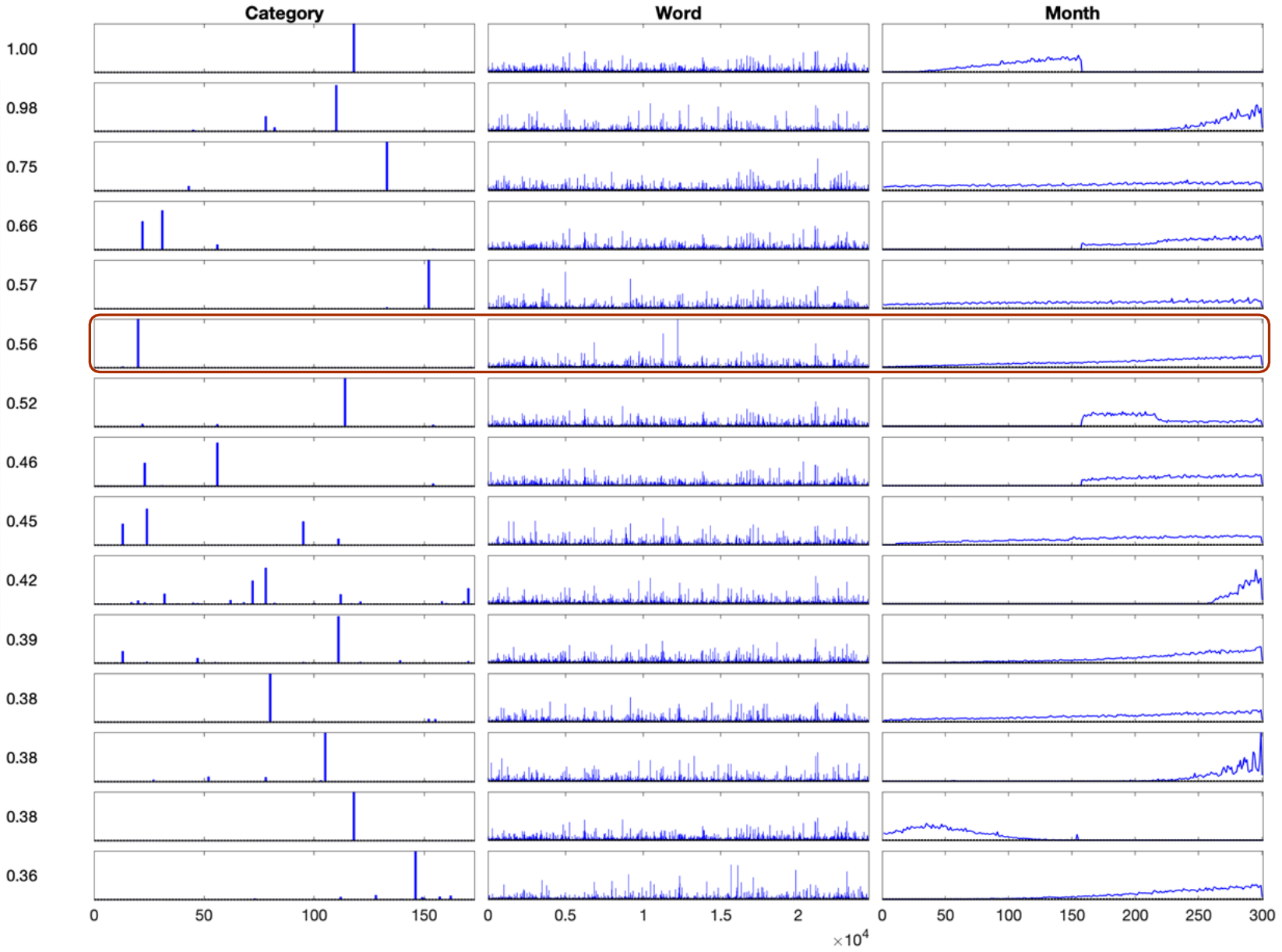
Component #2 (of 50): Computer Vision/Machine Learning

Top Categories	Category Weight	Top Words	Word Weight
cs.CV	1.00	propose	1.00
cs.LG	0.33	learn	0.95
stat.ML	0.08	use	0.92
cs.NE	0.03	method	0.88
cs.SD	0.02	model	0.82
cs.RO	0.01	network	0.77
q-bio.NC	0.01	paper	0.73
cs.GR	0.01	base	0.73
cs.SI	0.01	dataset	0.72
cs.IR	0.01	train	0.71
cs.MM	0.01	result	0.71
		image	0.70
		approach	0.63
		state	0.63
		datum	0.62
		art	0.62
		neural	0.61
		performance	0.61
		task	0.58
		deep	0.55



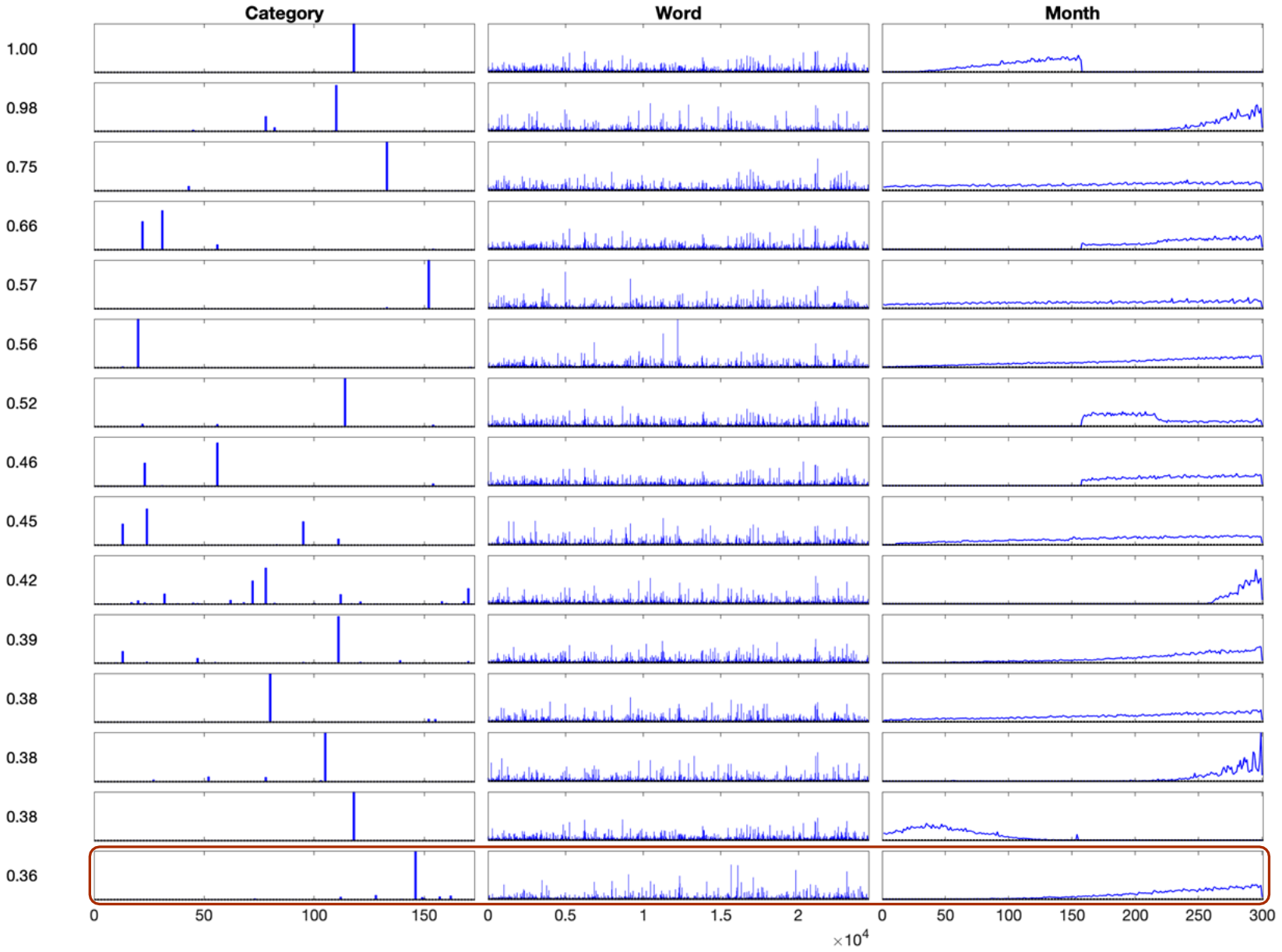
Component #6 (of 50): Quantum Physics

Top Categories	Category Weight	Top Words	Word Weight
quant-ph	1.00	quantum	1.00
cond-mat.mes-hall	0.02	state	0.70
cond-mat.stat-mech	0.01	system	0.52
		use	0.49
		result	0.35
		classical	0.33
		time	0.32
		entanglement	0.30
		qubit	0.29
		measurement	0.29
		present	0.28
		information	0.28
		non	0.28
		study	0.26
		single	0.25
		base	0.24
		find	0.24
		photon	0.24
		model	0.24
		field	0.23



Component #15 (of 50): Analysis of PDEs

Top Categories	Category Weight	Top Words	Word Weight
math.AP	1.00	equation	1.00
math.DS	0.09	solution	0.98
math.NA	0.08	prove	0.84
math.PR	0.06	result	0.80
math.OC	0.05	problem	0.72
math.DG	0.05	paper	0.67
math-ph	0.03	study	0.63
math.CA	0.02	consider	0.55
math.SP	0.01	existence	0.52
		condition	0.52
		space	0.50
		system	0.49
		time	0.49
		case	0.45
		function	0.43
		boundary	0.43
		obtain	0.42
		use	0.41
		estimate	0.40
		bound	0.37

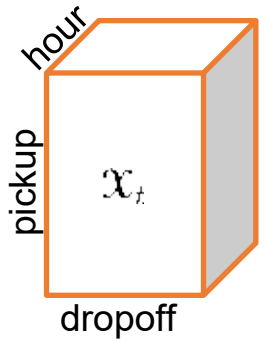


NYC Taxi Dataset – 4-way Tensor

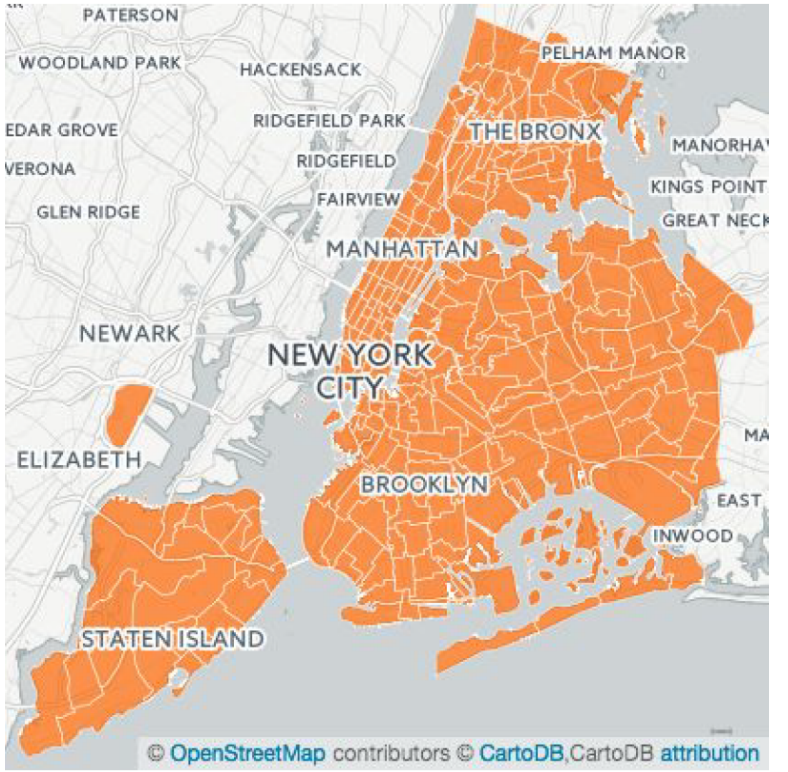
- **Data from NYC public records**
  https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

- **10+ Years of Data**

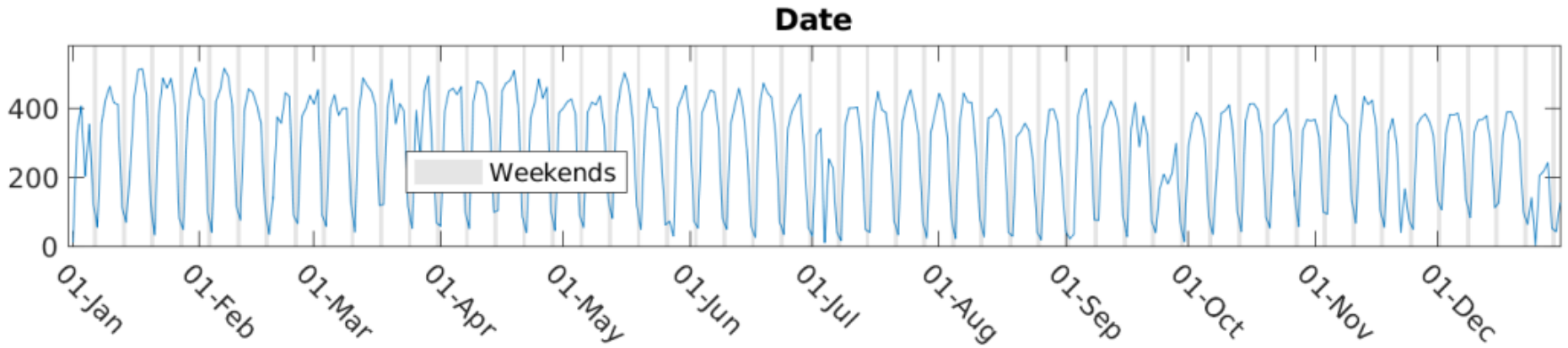
- **4-way Tensor, Updated Daily**
  - Pickup Zone
  - Dropoff Zone
  - Pickup Hour
  - *New 3-way sparse tensor each day*

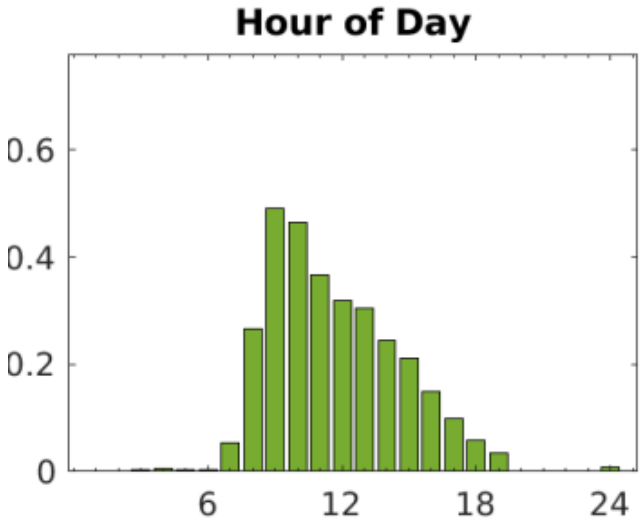


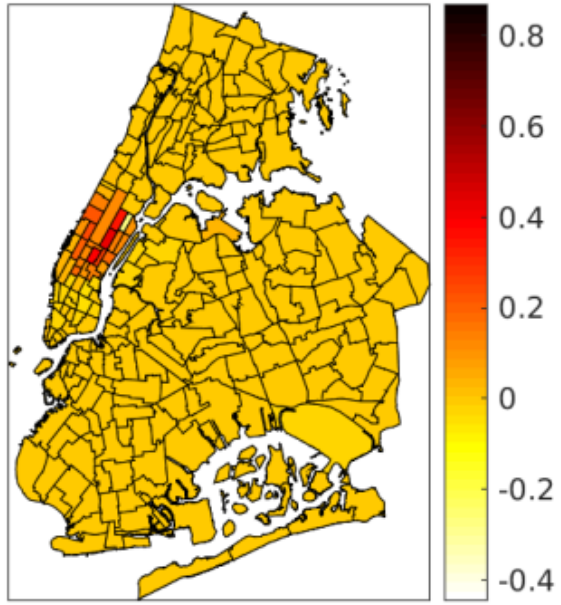
- **265 Taxi Zones**
  https://catalog.data.gov/dataset/nyc-taxi-zones

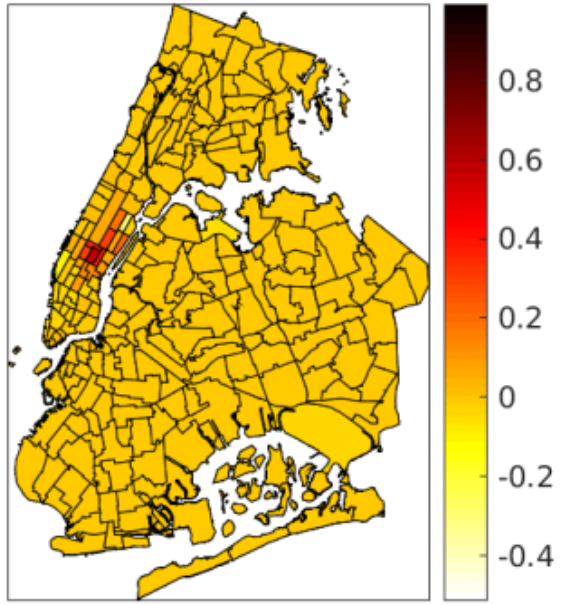


Component #1 (of 50) – Standard Mid-morning Weekday Traffic

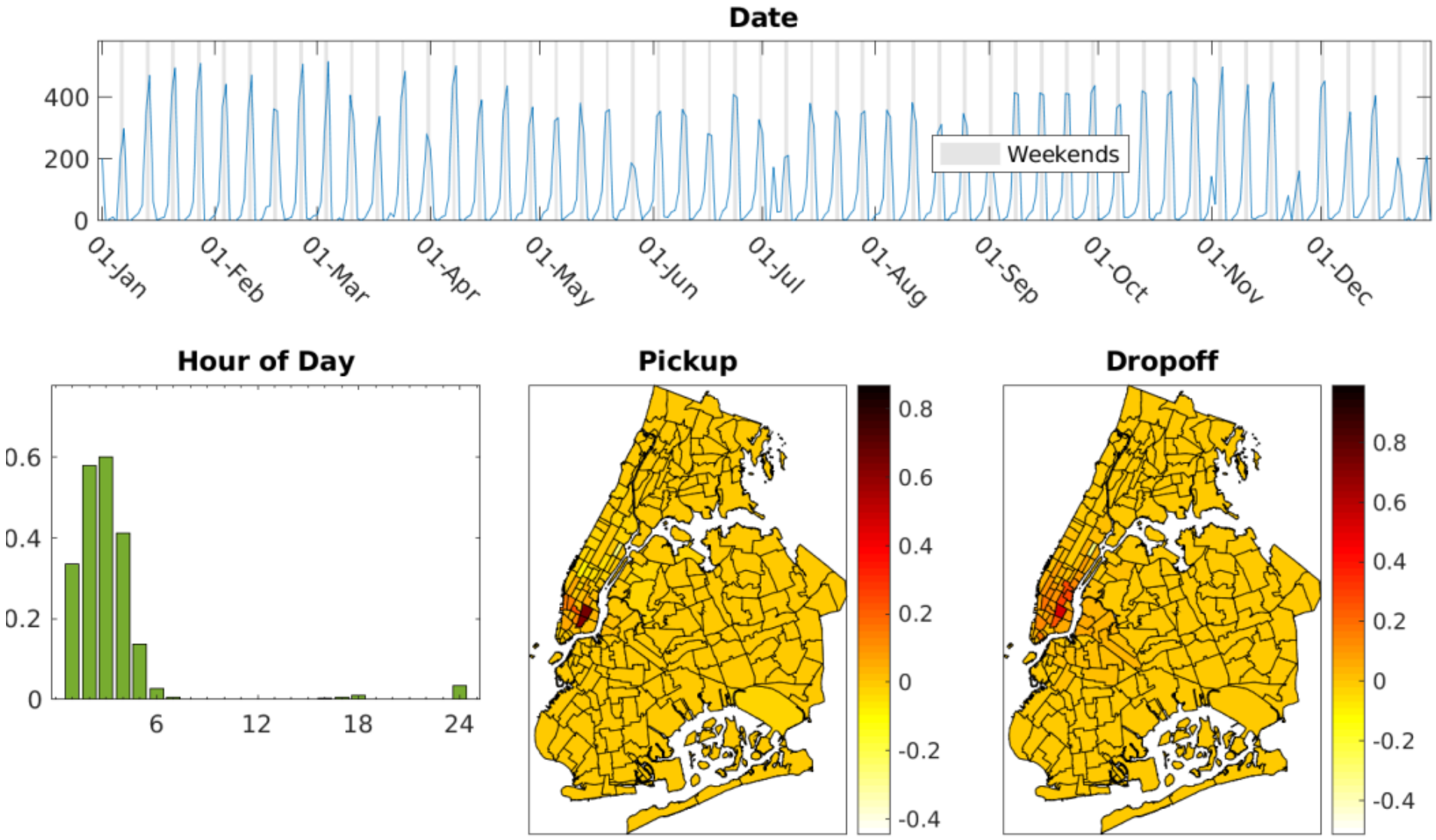




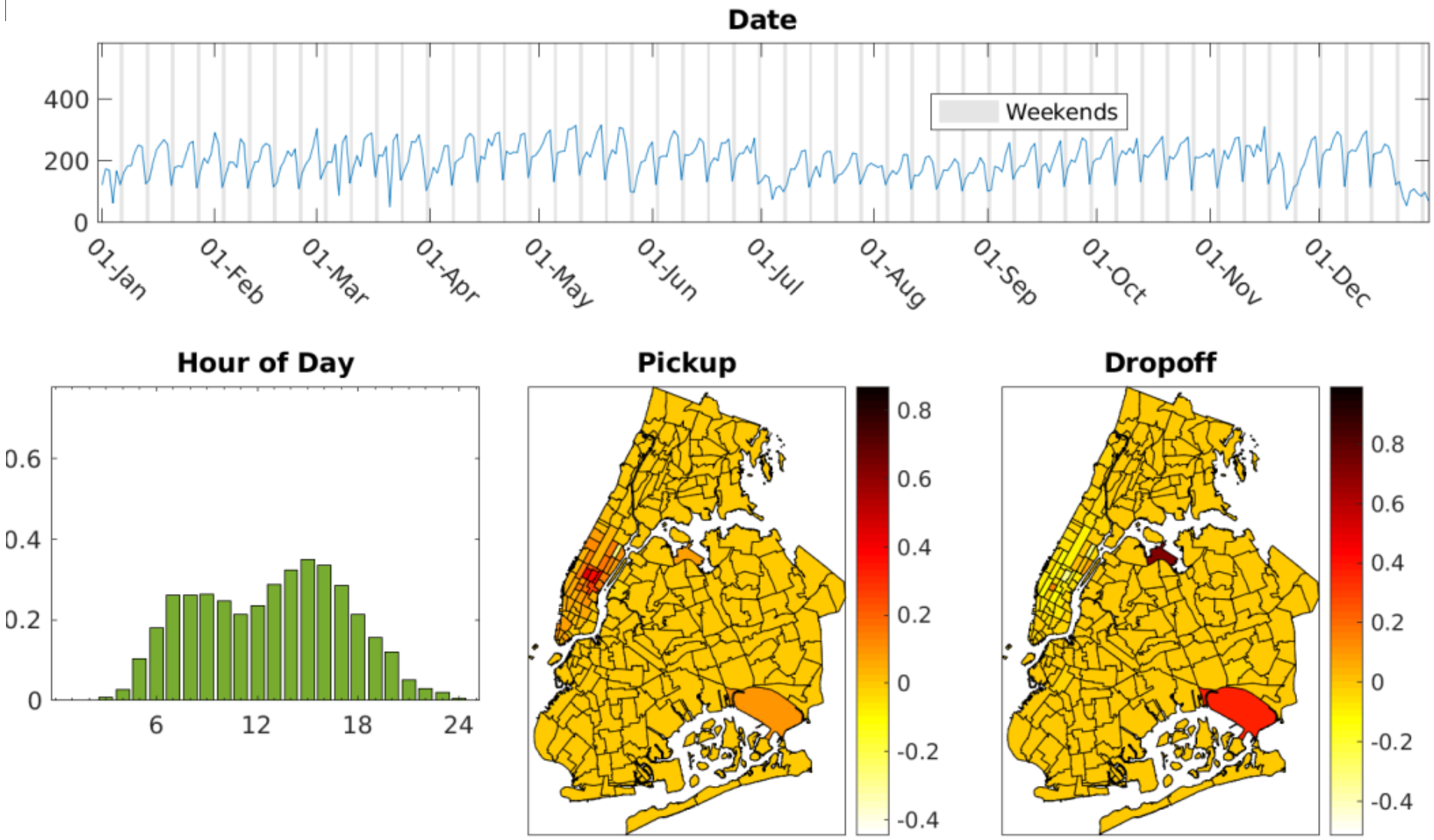




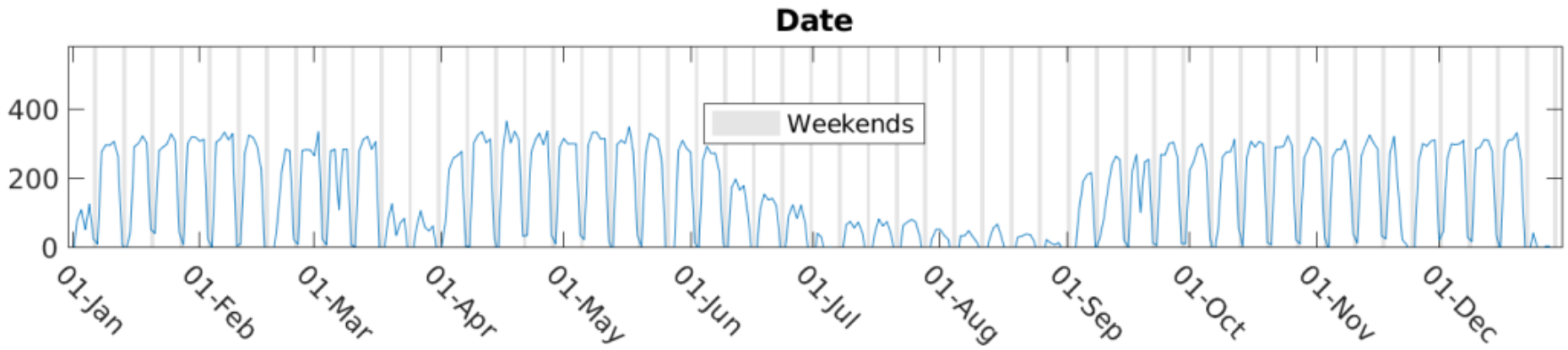
Component #21 (of 50): Weekend Nightlife

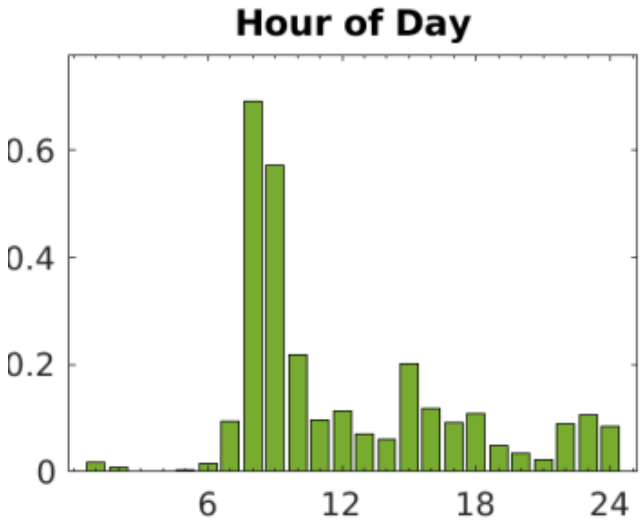


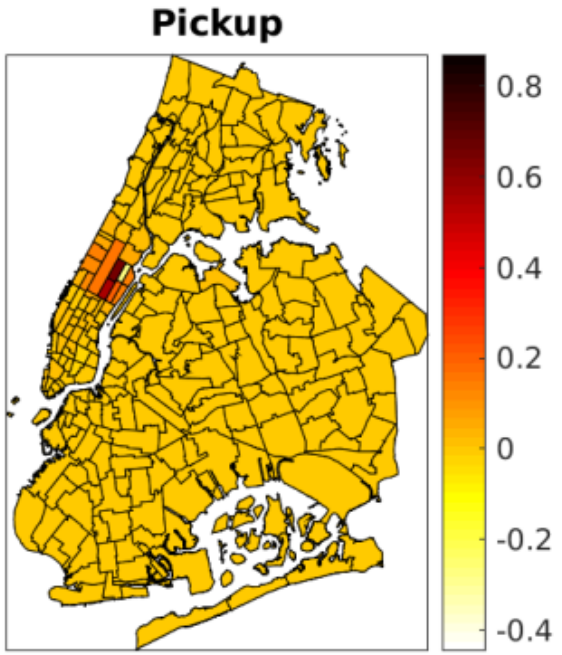
Component #17 (of 50): Travel to JFK and La Guardia Airports

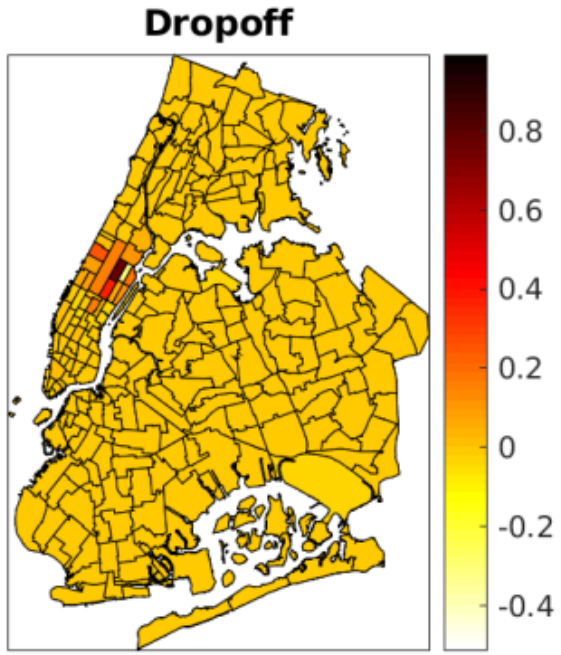


Component #20 (of 50): School Morning Dropoff

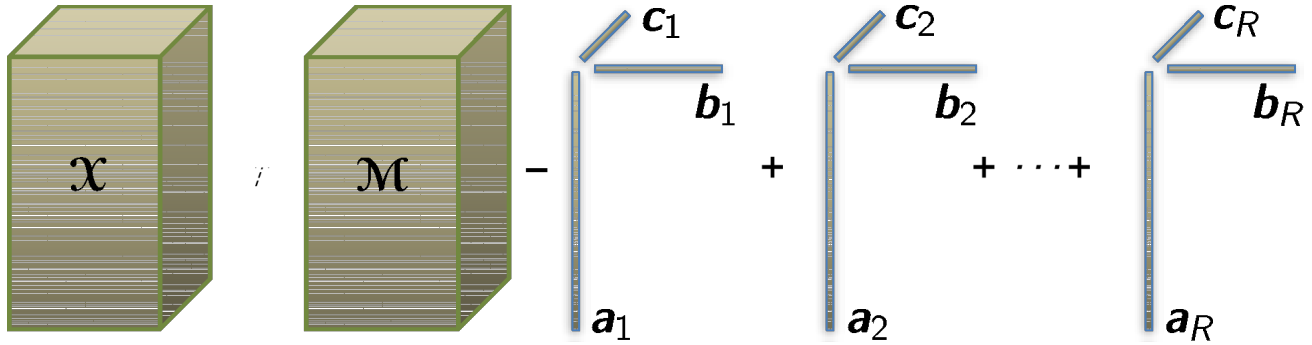








Standard CP and ALS Solution Method



$$\min_{\mathcal{M}} \ \|\mathcal{X}-\mathcal{M}\|_F^2 \ \text{ s.t. } \ \mathcal{M} = \boldsymbol{a}_1\circ\boldsymbol{b}_1\circ\boldsymbol{c}_1+\cdots+\boldsymbol{a}_R\circ\boldsymbol{b}_R\circ\boldsymbol{c}_R = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$$

$$\boldsymbol{A} = [\boldsymbol{a}_1 \ \ldots \ \boldsymbol{a}_R]$$
$$\boldsymbol{B} = [\boldsymbol{b}_1 \ \ldots \ \boldsymbol{b}_R]$$
$$\boldsymbol{C} = [\boldsymbol{c}_1 \ \ldots \ \boldsymbol{c}_R]$$

Using…

$$\mathcal{M}_{(1)} = \boldsymbol{A}(\boldsymbol{C}\odot\boldsymbol{B})^T$$
$$\mathcal{M}_{(2)} = \boldsymbol{B}(\boldsymbol{C}\odot\boldsymbol{A})^T$$
$$\mathcal{M}_{(3)} = \boldsymbol{C}(\boldsymbol{B}\odot\boldsymbol{A})^T$$

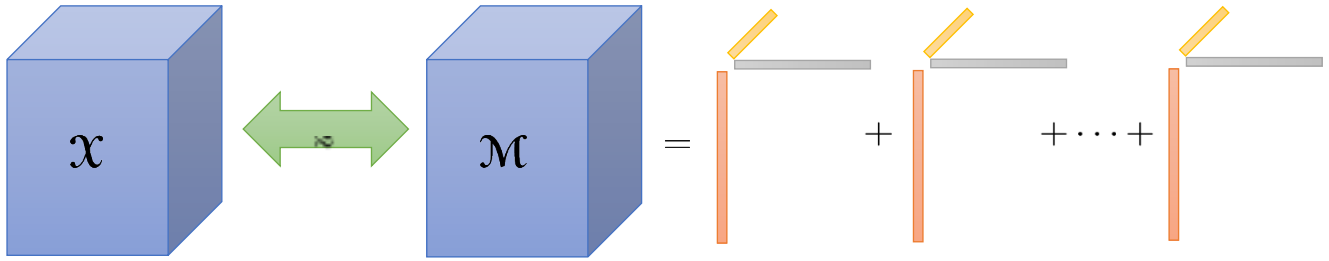
Repeat until convergence…

$$\min_{\boldsymbol{A}} \|\mathcal{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C}\odot\boldsymbol{B})^T\|_F^2$$
$$\min_{\boldsymbol{B}} \|\mathcal{X}_{(2)} - \boldsymbol{B}(\boldsymbol{C}\odot\boldsymbol{A})^T\|_F^2$$
$$\min_{\boldsymbol{C}} \|\mathcal{X}_{(3)} - \boldsymbol{C}(\boldsymbol{B}\odot\boldsymbol{A})^T\|_F^2$$

Repeat until convergence…

$$\boldsymbol{A} = \mathcal{X}_{(1)}(\boldsymbol{C}\odot\boldsymbol{B})(\boldsymbol{C}^T\boldsymbol{C} * \boldsymbol{B}^T\boldsymbol{B})^\dagger$$
$$\boldsymbol{B} = \mathcal{X}_{(2)}(\boldsymbol{C}\odot\boldsymbol{A})(\boldsymbol{C}^T\boldsymbol{C} * \boldsymbol{A}^T\boldsymbol{A})^\dagger$$
$$\boldsymbol{C} = \underbrace{\mathcal{X}_{(3)}(\boldsymbol{B}\odot\boldsymbol{A})}_{\text{MTTKRP}}(\boldsymbol{B}^T\boldsymbol{B} * \boldsymbol{A}^T\boldsymbol{A})^\dagger$$

Generalized CP (GCP) Tensor Decomposition Allows Flexible Loss Function



Generalized CP (GCP)

$$\min_{\mathbf{A}_1,\ldots,\mathbf{A}_d} \quad F(\mathcal{X},\mathcal{M}) = \sum_i f(x_i, m_i)$$

$$\text{s.t.} \quad \mathcal{M} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_d]\!]$$

$$\mathbf{A}_k \in \mathbb{R}^{n_k \times r} \text{ for } k = 1, \ldots, d$$

Hong, Kolda, Duersch, Generalized Canonical Polyadic Tensor Decomposition, SIREV 2020.

Example Loss Functions

Normal ($x, m \in \mathbb{R}$)

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

Poisson ($x \in \mathbb{N}, m > 0$)

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

Bernoulli ($x \in \{0,1\}, m > 0$)

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

β-divergence ($x > 0, m > 0, \beta = \frac{1}{2}$)

$$f(x,m) = x/\sqrt{m} + \sqrt{m}$$

Fitting GCP Model[1]

$$\min_{\mathcal{M}} \; F(\mathcal{X}, \mathcal{M}) = \sum_i f(x_i, m_i) \quad \text{s.t.} \quad \mathcal{M} = [\![\boldsymbol{A}_1, \ldots, \boldsymbol{A}_d]\!]$$

Lose the least-squares structure underlying ALS-type algorithms. Instead pursue gradient-based optimization approach.

Define tensor $\boldsymbol{Y}$ such that

$$y(i_1, \ldots, i_d) = y_i = \frac{\partial f}{\partial m}(x_i, m_i)$$

Then gradient of objective function given by

$$\boldsymbol{G}_k = \frac{\partial F}{\partial \boldsymbol{A}_k} = \mathcal{Y}_{(k)}(\boldsymbol{A}_d \odot \cdots \odot \boldsymbol{A}_{k+1} \odot \boldsymbol{A}_{k-1} \odot \cdots \odot \boldsymbol{A}_1)$$

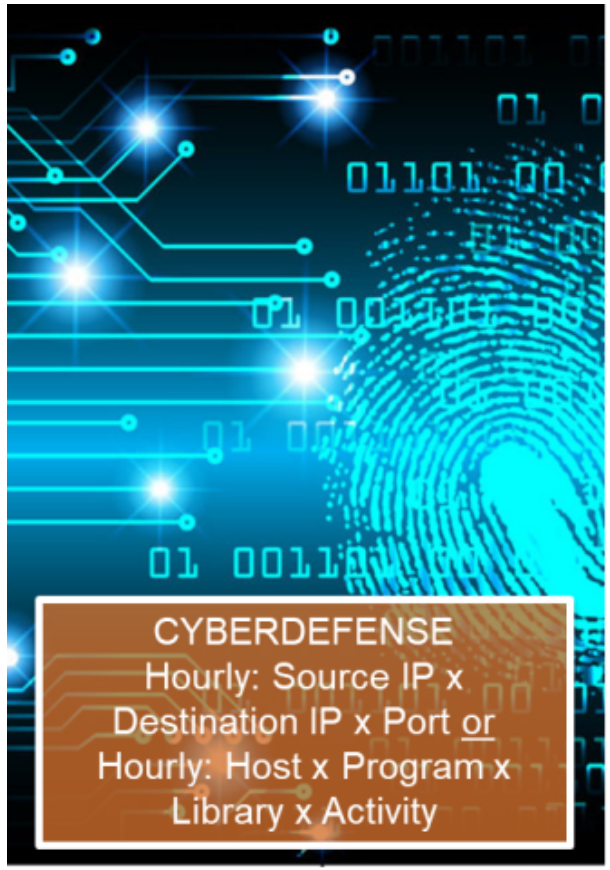
← MTTKRP!

Unfortunately, $\boldsymbol{Y}$ is in general dense, even when $\boldsymbol{X}$ is sparse, making standard optimization infeasible.

Instead, employ Stochastic Gradient Descent (SGD) where $\boldsymbol{Y}$ is only randomly sampled

- Stratified: sample zeros and nonzeros separately (requires tensor search)
- Semi-stratified: skip tensor search and adjust for "zeros" that are really nonzeros

[1]Kolda and Hong Stochastic Gradients for Large-Scale Tensor Decomposition, SIMODS, 2020.



Examples of Streaming Data

Streaming Tensor Decompositions

Desire algorithms that can compute CP decompositions as data streams in

Many formulations of this problem and assumptions on the type of data and processes generating it

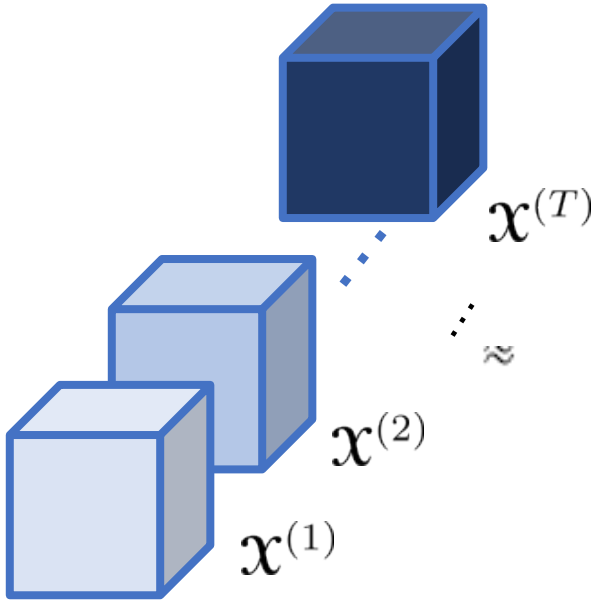
Assumptions in this work:

- Updates are processed in discrete batches indexed by time $t = 1,2, \ldots$
- A complete *d*-way tensor is observed at each time step
- The number of time steps is unbounded (infinite streaming)
- The dimensions of each update are fixed throughout time
- The CP rank is known and fixed

Requires algorithms that can incrementally update the decomposition without revisiting sufficiently old data

Streaming Tensors – Two Points-of-View

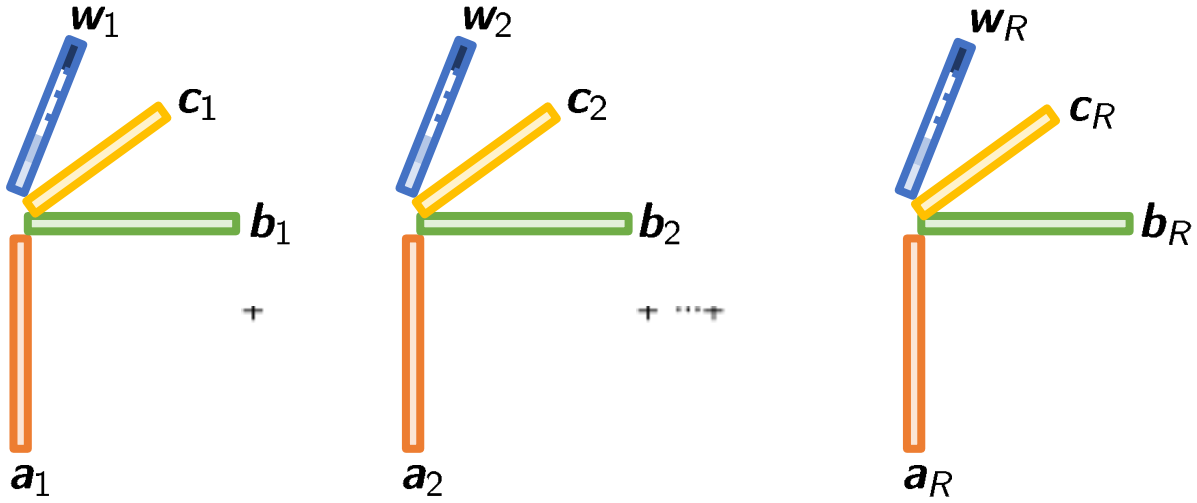
At each time step t, new 3-way hyperslice added



$$\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3 \times T}$$

$$\mathcal{X}^{(t)} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$$

If we assume factor matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ *fixed through time*, then the ideal factorization looks like…



$$\mathcal{X} \approx \sum_{j=1}^{R} \boldsymbol{a}_j \circ \boldsymbol{b}_j \circ \boldsymbol{c}_j \circ \boldsymbol{w}_j = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{W}]\!]$$

$$\boldsymbol{A} = [\boldsymbol{a}_1 \ \boldsymbol{a}_2 \ \ldots \ \boldsymbol{a}_R] \in \mathbb{R}^{N_1 \times R}$$

$$\boldsymbol{B} = [\boldsymbol{b}_1 \ \boldsymbol{b}_2 \ \ldots \ \boldsymbol{b}_R] \in \mathbb{R}^{N_2 \times R}$$

$$\boldsymbol{C} = [\boldsymbol{c}_1 \ \boldsymbol{c}_2 \ \ldots \ \boldsymbol{c}_R] \in \mathbb{R}^{N_3 \times R}$$

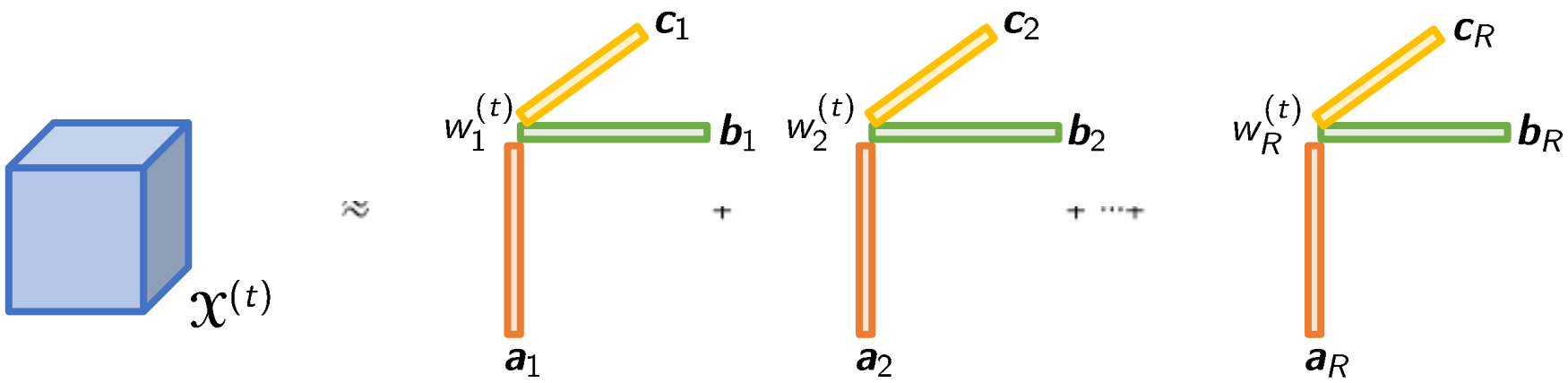
$$\boldsymbol{W} = [\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \ldots \ \boldsymbol{w}_R] \in \mathbb{R}^{T \times R}$$

Generic d-way setup is similar.

Streaming Tensors – Two Points-of-View

At each time step t, new 3-way hyperslice added

If we assume factor matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ *fixed through time*, then the ideal factorization looks like…



$$\mathcal{X}^{(t)} \approx \sum_{j=1}^{R} w_j^{(t)} \boldsymbol{a}_j \circ \boldsymbol{b}_j \circ \boldsymbol{c}_j = [\![\boldsymbol{w}^{(t)}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$$

$$\boldsymbol{A} = [\boldsymbol{a}_1 \ \boldsymbol{a}_2 \ \ldots \ \boldsymbol{a}_R] \in \mathbb{R}^{N_1 \times R}$$

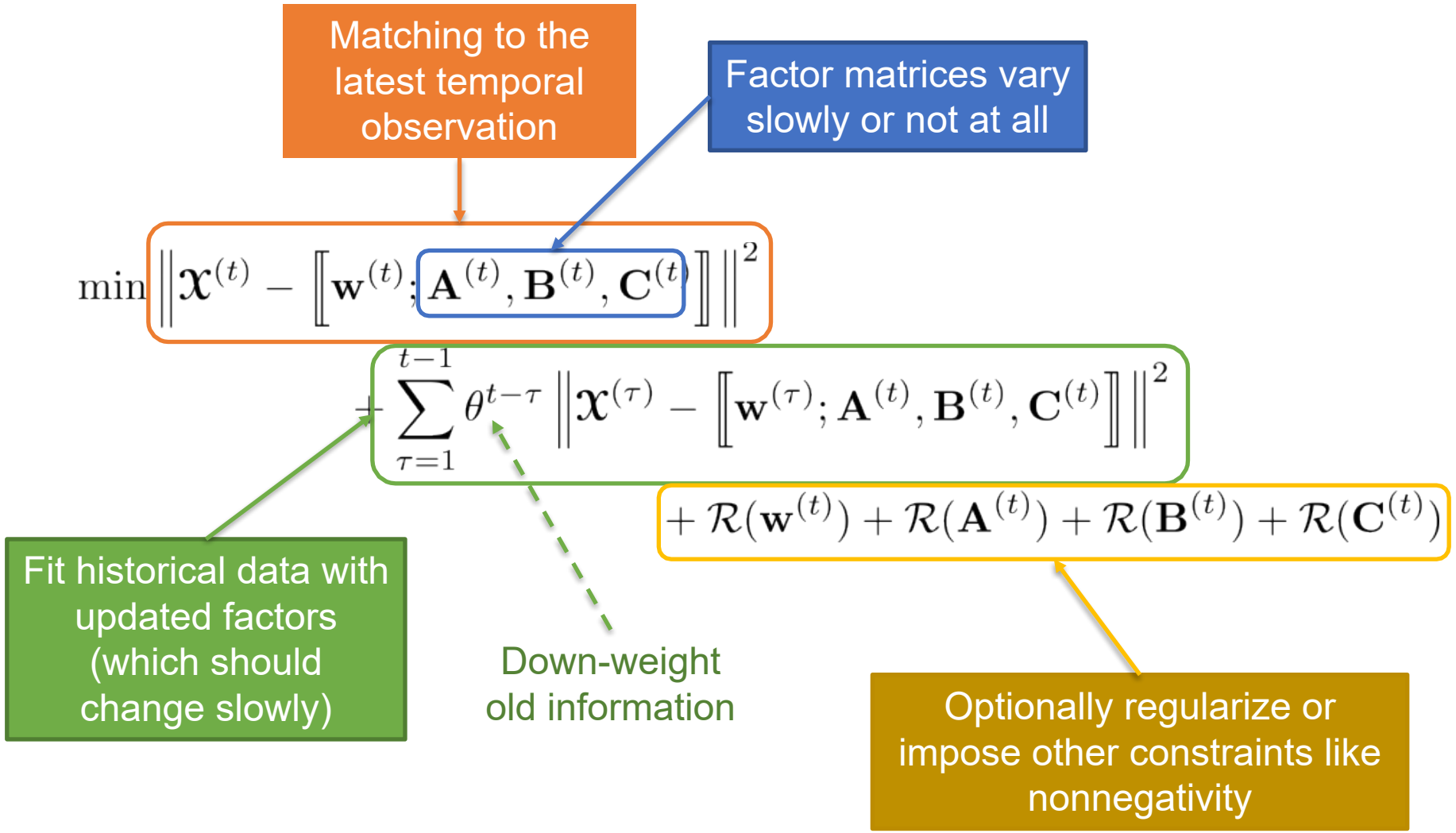
$$\boldsymbol{B} = [\boldsymbol{b}_1 \ \boldsymbol{b}_2 \ \ldots \ \boldsymbol{b}_R] \in \mathbb{R}^{N_2 \times R}$$

$$\boldsymbol{C} = [\boldsymbol{c}_1 \ \boldsymbol{c}_2 \ \ldots \ \boldsymbol{c}_R] \in \mathbb{R}^{N_3 \times R}$$

$$\mathcal{X}^{(t)} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$$

Generic d-way setup is similar.

Generic Streaming Formulation (For Gaussian Data)



A Few (of Many) Related Methods

Definitions

- CP model at current time step: $\mathcal{M}^{(t)} = [\![\boldsymbol{w}^{(t)}; \boldsymbol{A}_1, \ldots, \boldsymbol{A}_d]\!]$
- CP model with factor matrices from prior time step: $\bar{\mathcal{M}}^{(t)} = [\![\boldsymbol{w}^{(t)}; \bar{\boldsymbol{A}}_1, \ldots, \bar{\boldsymbol{A}}_d]\!]$

Online SGD[1]

- Formulation: $\min_{\boldsymbol{w}^{(t)}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_d} \sum_{h=1}^{t} \theta^{t-h} \| \mathcal{U}^{(h)} * (\mathcal{X}^{(h)} - \mathcal{M}^{(h)}) \|^2 + \bar{\lambda}_t \sum_{j=1}^{d} \| \boldsymbol{A}_j \|_F^2 + \lambda \| \boldsymbol{w}^{(t)} \|_2^2$
- Two-step solution process at each time step:
  - Solve for temporal weights $\boldsymbol{w}^{(t)}$ with factor matrices $\boldsymbol{A}_j$ held fixed (linear least squares)
  - Apply 1 step of gradient descent to update factor matrices $\boldsymbol{A}_j$ (called stochastic as mask may vary randomly from step to step)

CP-Stream[2]

- Formulation: $\min_{\boldsymbol{w}^{(t)}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_d} \| \mathcal{X}^{(t)} - \mathcal{M}^{(t)} \|^2 + \sum_{h=1}^{t-1} \theta^{t-h} \| \bar{\mathcal{M}}^{(h)} - \mathcal{M}^{(h)} \|^2 + \lambda \| \boldsymbol{w}^{(t)} \|_2^2$
- Two-step solution process at each time step:
  - Solve for temporal weights $\boldsymbol{w}^{(t)}$ with factor matrices $\boldsymbol{A}_j$ held fixed (linear least squares)
  - Apply ADMM algorithm for updating factor matrices $\boldsymbol{A}_j$

OnlineCP[3]

- Formulation: $\min_{\boldsymbol{w}^{(t)}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_d} \sum_{h=1}^{t} \| \mathcal{X}^{(h)} - \mathcal{M}^{(h)} \|^2$
- Applies one step of ALS each time step for temporal weights $\boldsymbol{w}^{(t)}$ and factor matrices $\boldsymbol{A}_j$
- Leverages least-squares structure to incorporate prior tensor data into MTTKRP and Gram computations without explicitly storing prior slices

[1]Mardani et al, Subspace Learning and Imputation for Streaming Big Data Matrices and Tensors, 2015
[2]Smith et al, Streaming Tensor Factorization for Infinite Data Sources, 2018
[3]Zhou et al, Accelerating Online CP Decompositions for Higher Order Tensors, 2016

Streaming GCP ("OnlineGCP")

History regularization penalizing large changes in CP model

$$\min_{\boldsymbol{w}^{(t)},\boldsymbol{A}_1,\ldots,\boldsymbol{A}_d} \sum_{i\in\mathcal{I}} f(x_i^{(t)}, m_i^{(t)}) + \frac{1}{2}\sum_{h\subset\mathcal{H}^{(t)}} s^{(h)} \|\bar{\mathcal{M}}^{(h)} - \mathcal{M}^{(h)}\|^2 + \frac{\lambda}{2}\sum_{k=1}^{d} |\boldsymbol{A}_k|_F^2 - \frac{\mu}{2}\|\boldsymbol{w}^{(t)}\|_2^2$$

GCP Loss for current time step

History window

Factor matrix regularization to encourage low-rank

Two step solution process

- Solve for temporal weights $\boldsymbol{w}^{(t)}$ with factor matrices $\boldsymbol{A}_j$ held fixed using GCP-SGD
- Solve for updated factor matrices $\boldsymbol{A}_j$ using GCP-SGD

Leverage ADAM SGD solvers

- Start new ADAM solver each time step for temporal weights
- Keep momentum terms from ADAM across factor matrix solves
- Sampled gradient approximations each ADAM epoch (stratified sampling)

History window

- Many possibilities depending on the problem
- We used *reservoir sampling*, which provides a uniform sample of [1,t], by randomly evicting previous entries with diminishing probability

GenTen : Software for Generalized Canonical Polyadic Tensor Decompositions

GenTen : Software for Generalized Canonical Polyadic Tensor Decompositions[1]

- HPC-oriented toolbox for CP decompositions
- CP-ALS, CP-OPT, GCP-OPT, GCP-SGD, OnlineGCP
- Supports both sparse and dense tensors

Targeted at extreme-scale architectures

- Distributed memory parallelism via MPI
- Shared memory parallelism via Kokkos
  - Multicore CPUs (OpenMP, pThreads)
  - NVIDIA (CUDA), AMD (HIP) and Intel (SYCL) GPUs

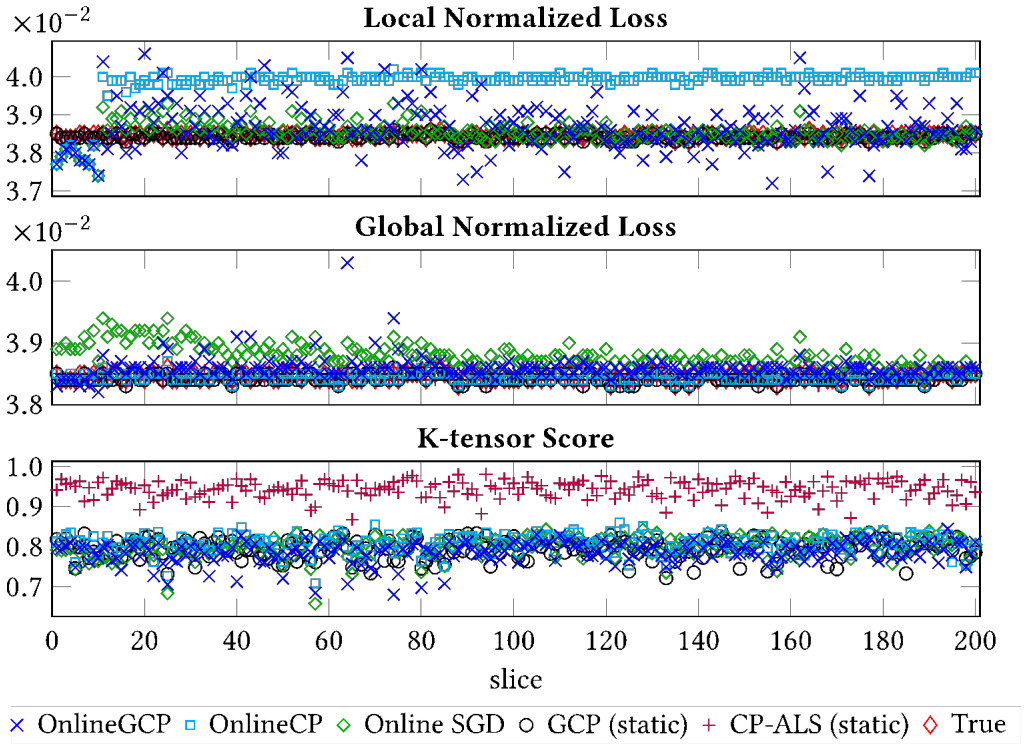
Callable from Matlab and Python

- Integrates with Matlab[2] and Python[3] Tensor Toolboxes
- Manipulate/analyze tensor data but call GenTen HPC decomposition algorithms

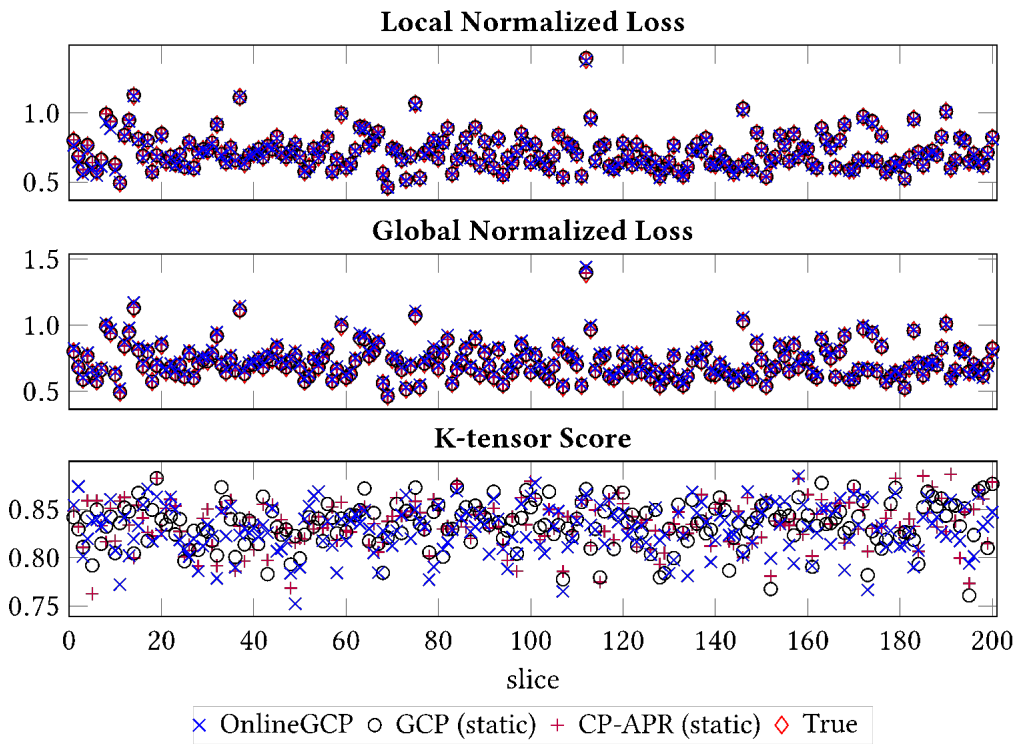
[1] https://gitlab.com/tensors/genten
[2] https://www.tensortoolbox.org/
[3] https://github.com/sandialabs/pyttb

Synthetic Data Experiments

Gaussian 300x300x200, dense, R=20



Poisson 300x300x200, 3.2% sparsity, R=20

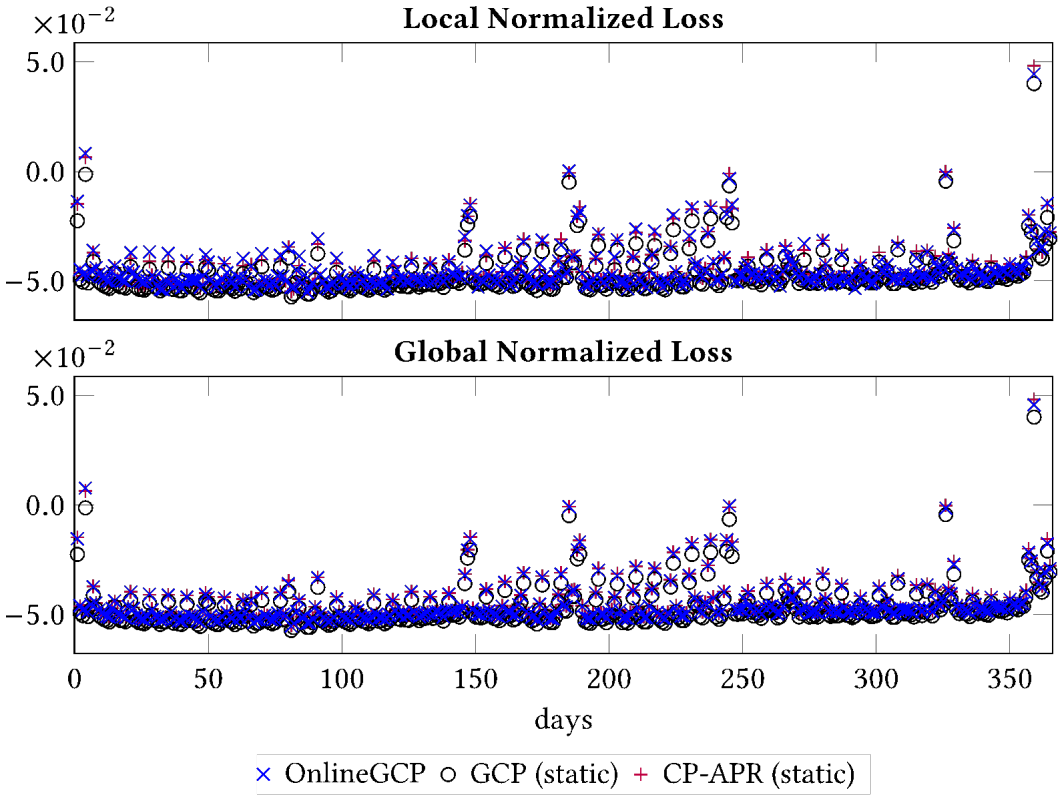


Local Normalized Loss: $F_{local}(\mathcal{X}^{(t)}, \mathcal{M}^{(t)}) = \frac{1}{\|\mathcal{X}^{(t)}\|_F^2} \sum_{i \in \mathcal{I}} f(x_i^{(t)}, m_i^{(t)}),; \; \mathcal{M}^{(t)} = [\![\boldsymbol{w}^{(t)}; \boldsymbol{A}_1^{(t)}, \ldots, \boldsymbol{A}_d^{(t)}]\!]$

Global Normalized Loss: $F_{global}(\mathcal{X}^{(t)}, \tilde{\mathcal{M}}^{(t)}) = \frac{1}{\|\mathcal{X}^{(t)}\|_F^2} \sum_{i \in \mathcal{I}} f(x_i^{(t)}, \tilde{m}_i^{(t)}),; \; \tilde{\mathcal{M}}^{(t)} = [\![\boldsymbol{w}^{(t)}; \boldsymbol{A}_1^{(T)}, \ldots, \boldsymbol{A}_d^{(T)}]\!]$

Real Data Experiments (Poisson)

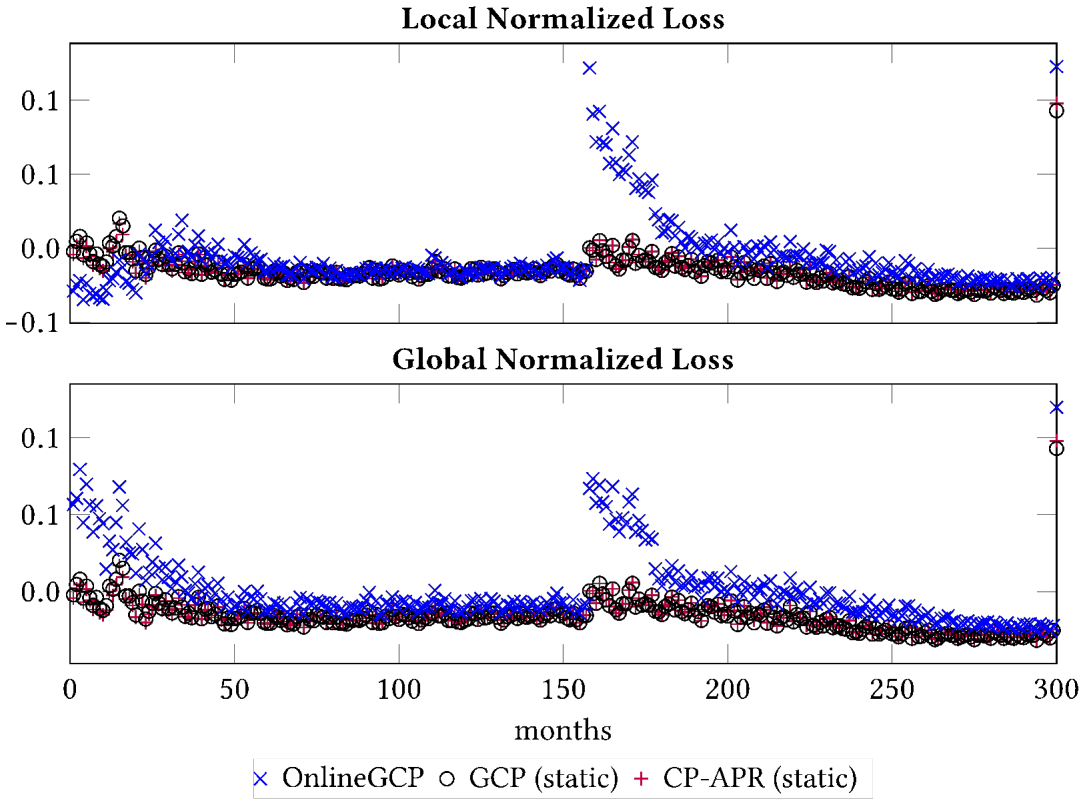
NYC Taxicab



Four-way tensor of counts of taxi rides in NYC in 2018

- Pickup zone (263)
- Dropoff zone (263)
- Pickup hour (24)
- Day (265)
- 3.8% sparsity

ArXiv Abstracts



Three-way tensor of counts of words in abstracts published on arxiv.org

- Category (172)
- Word (24558)
- Month since 10 years after first abstract (300)
- 2.4% sparsity

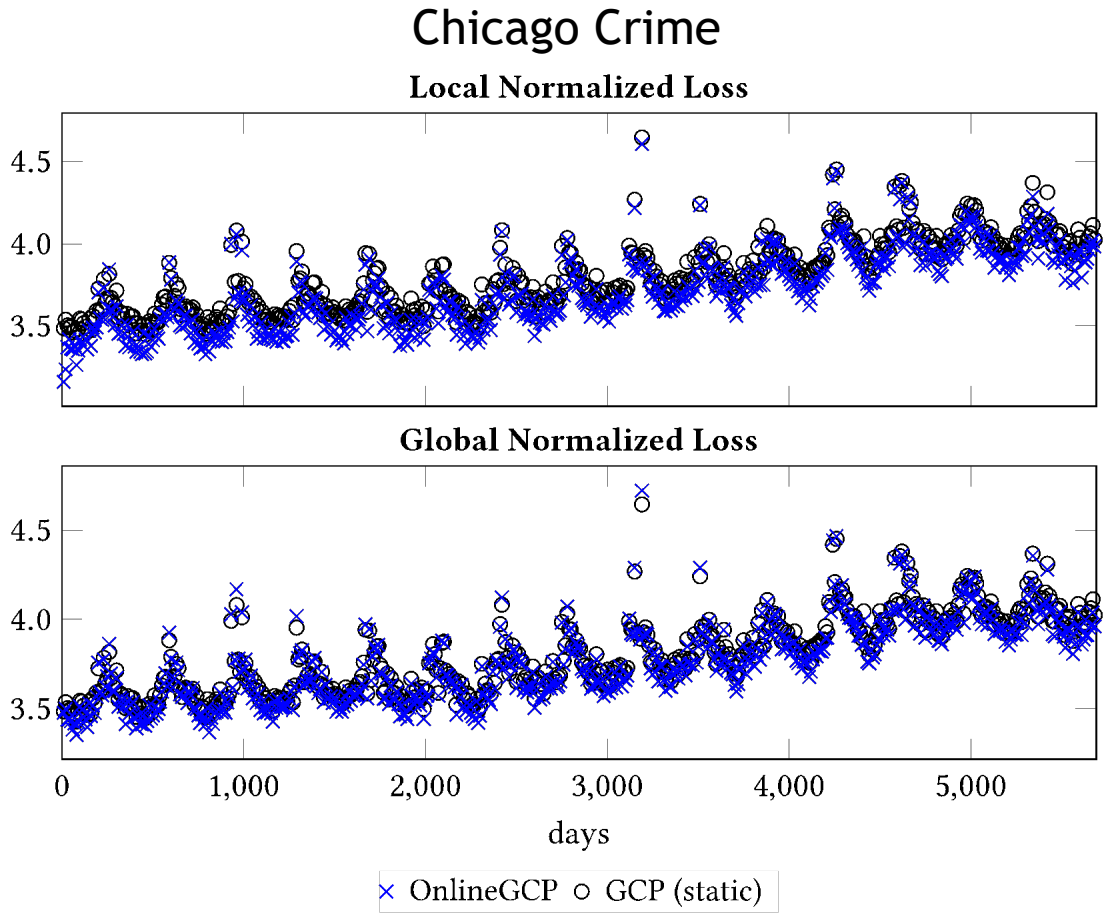
Real Data Experiments (Bernoulli)

Count tensor from FROSTT[1]

Four-way tensor:

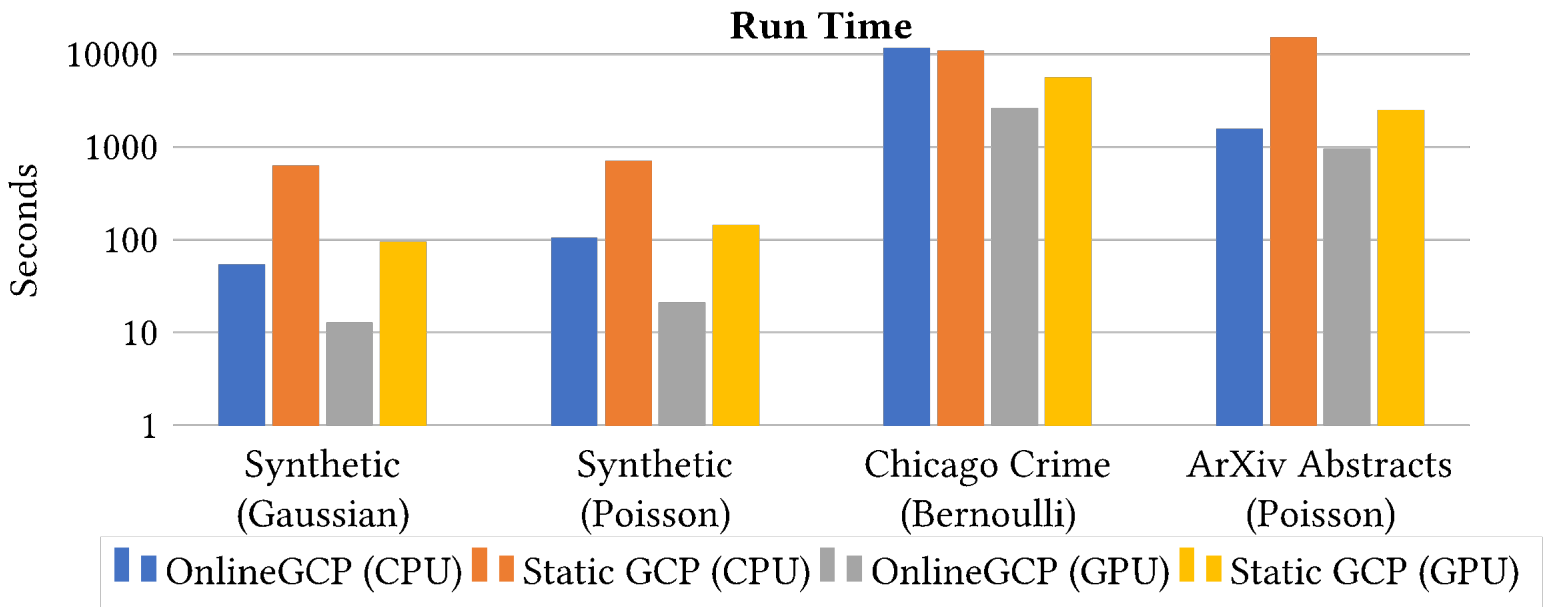
- Hour of the day (24)
- Crime type (77)
- Neighborhood (32)
- Day since start date (5687)
- 1.6% sparsity

Most counts are just 1, so use Bernoulli instead of Poisson



[FROSTT: The Formidable Repository of Open Sparse Tensors and Tools]

Computational Performance



CPU:
- Dual-socket Intel Xeon Platinum 8260, 24 cores/socket
- OpenMP Kokkos backend

GPU:
- NVIDIA Volta V100
- CUDA Kokkos backend

Same total number of samples between Static GCP and OnlineGCP

Summary and Conclusions

Summary

- Streaming tensor decomposition method for general statistical data types (continuous, count, binary, …)
- Provides accuracy comparable to static/batch methods
- HPC software implementation in GenTen with Matlab/Python usability front-ends

Challenges

- Online (and static) GCP require tuning of many hyper-parameters (number of samples, learning rates, …) particularly when balancing cost versus accuracy
- SGD solver often converges slowly increasing computational cost

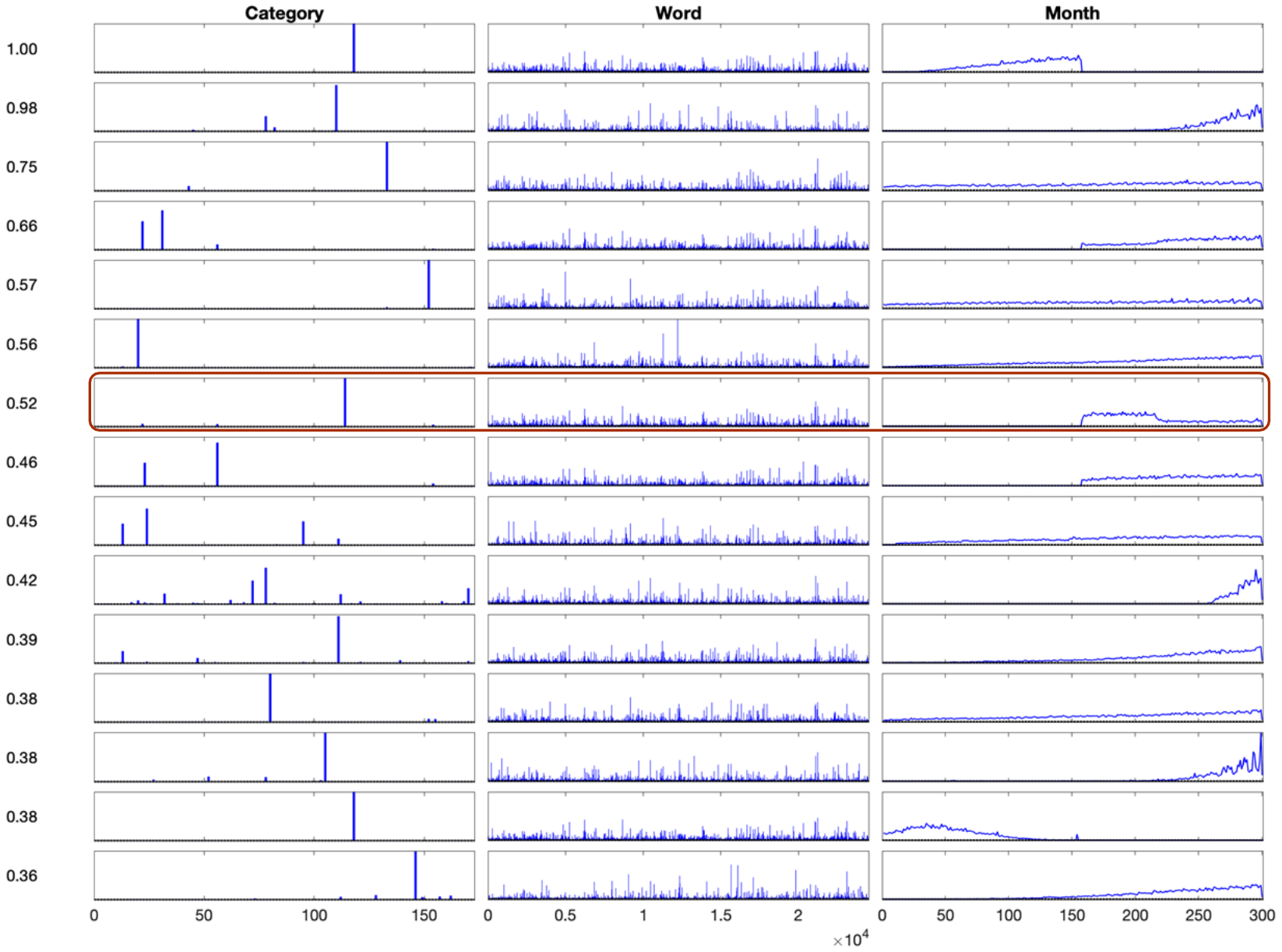
Moving forward

- Address online and static GCP solver challenges
- Incorporating distributed (synchronous and asynchronous parallelism)

Backup Slides

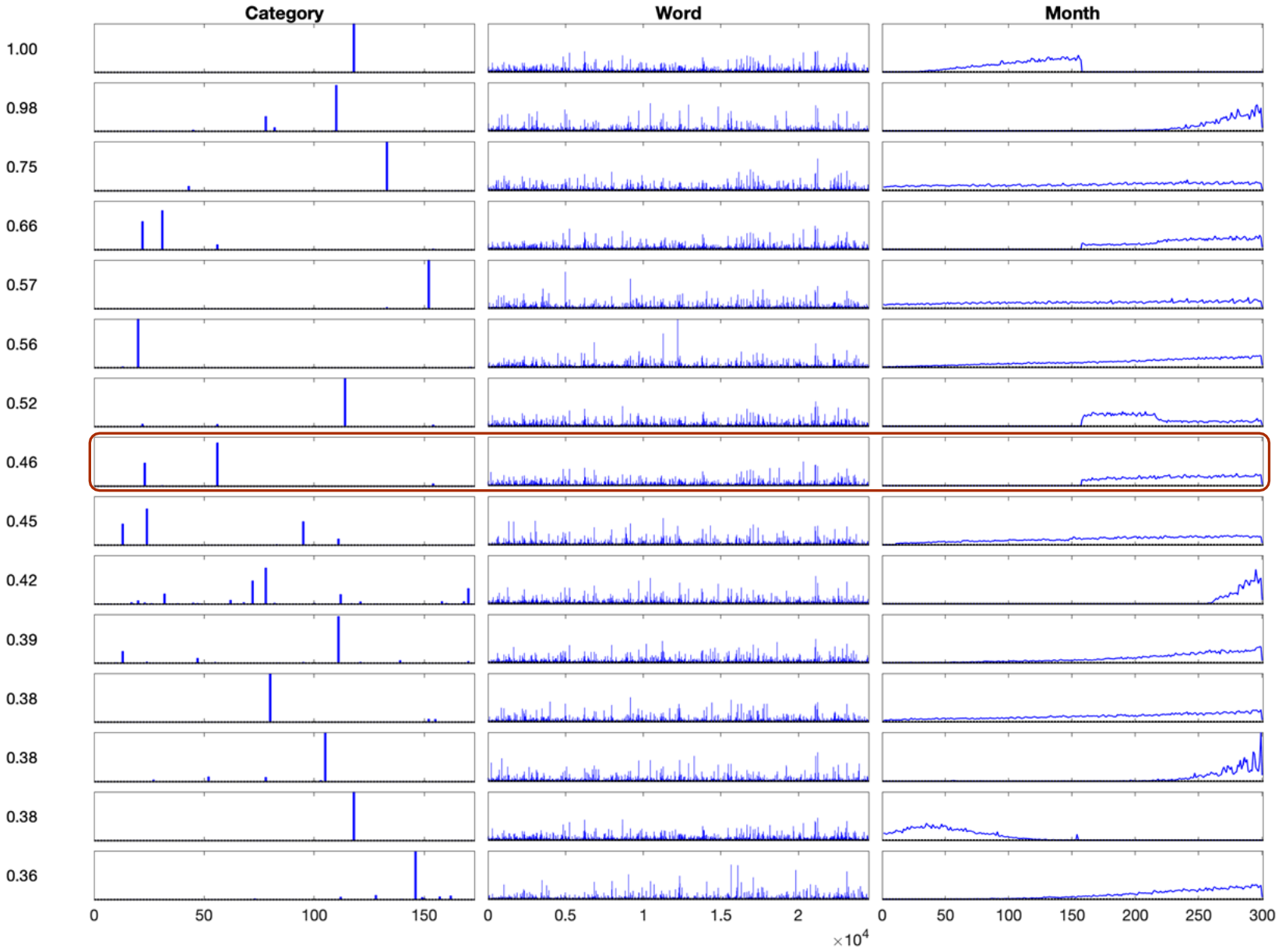
Component #5 (of 50): Astrophysics Reorganized

Top Categories	Category Weight	Top Words	Word Weight
astro-ph.CO	1.00	use	1.00
astro-ph.HE	0.05	galaxy	0.81
astro-ph.SR	0.05	model	0.80
astro-ph.IM	0.04	find	0.77
		datum	0.75
		result	0.70
		large	0.70
		present	0.70
		high	0.63
		scale	0.63
		redshift	0.58
		study	0.55
		mass	0.55
		cosmological	0.52
		parameter	0.52
		dark	0.51
		field	0.50
		observation	0.50
		good	0.50
		density	0.49



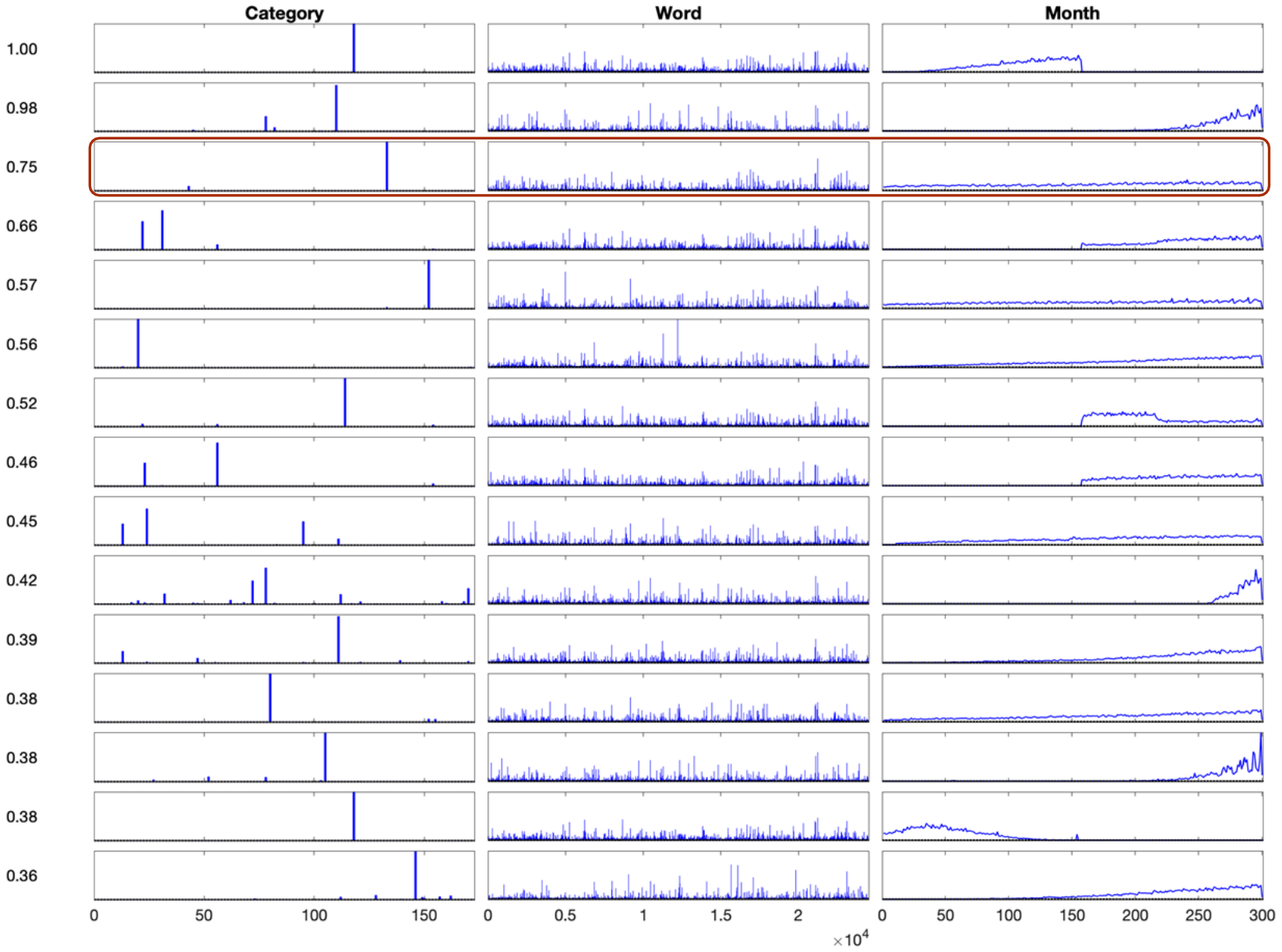
Component #6 (of 50): Astrophysics Reorganized

Top Categories	Category Weight	Top Words	Word Weight
astro-ph.SR	1.00	star	1.00
astro-ph.EP	0.54	use	0.87
astro-ph.IM	0.06	find	0.85
astro-ph.GA	0.02	model	0.81
physics.space-ph	0.01	present	0.74
		solar	0.73
		observation	0.73
		result	0.71
		high	0.69
		observe	0.69
		mass	0.67
		system	0.65
		large	0.62
		study	0.62
		low	0.59
		time	0.58
		stellar	0.57
		planet	0.56
		datum	0.53
		good	0.48



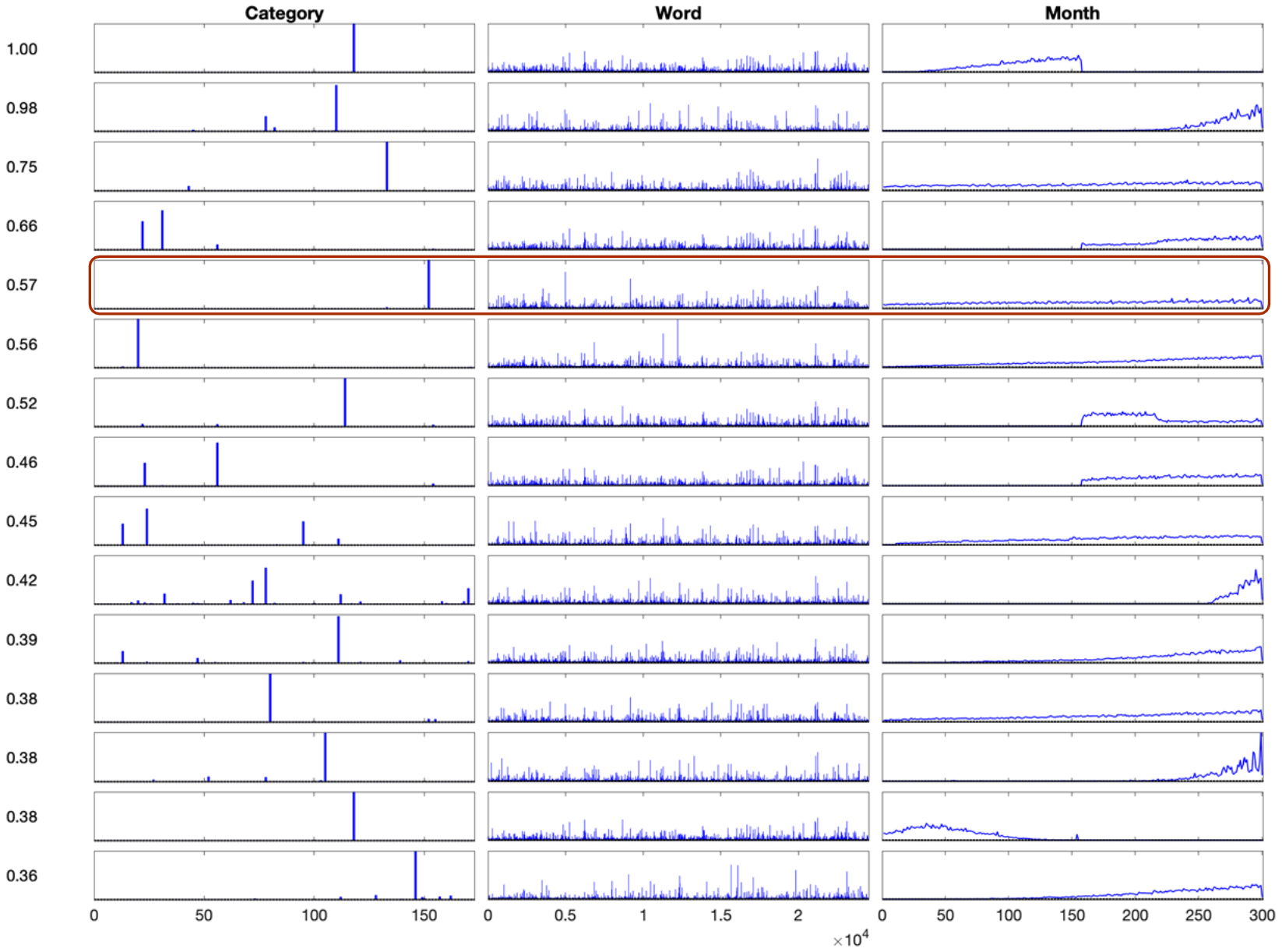
Component #3 (of 50): High Energy Physics

Top Categories	Category Weight	Top Words	Word Weight
hep-ph	1.00	model	1.00
nucl-th	0.10	mass	0.66
		result	0.61
		energy	0.58
		study	0.58
		quark	0.57
		use	0.54
		decay	0.54
		find	0.52
		present	0.51
		datum	0.48
		discuss	0.47
		large	0.47
		order	0.46
		standard	0.46
		parameter	0.44
		new	0.40
		effect	0.40
		lead	0.39
		high	0.39

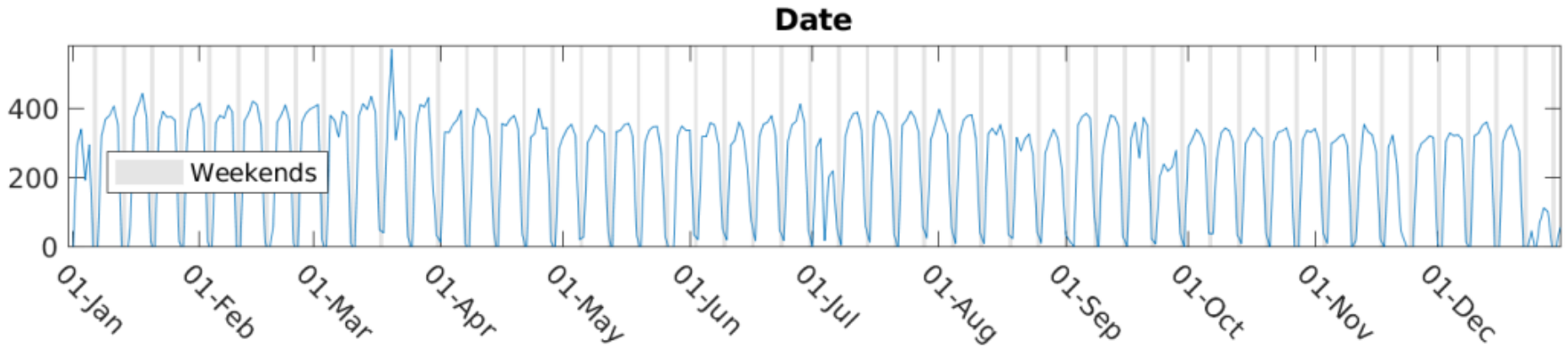


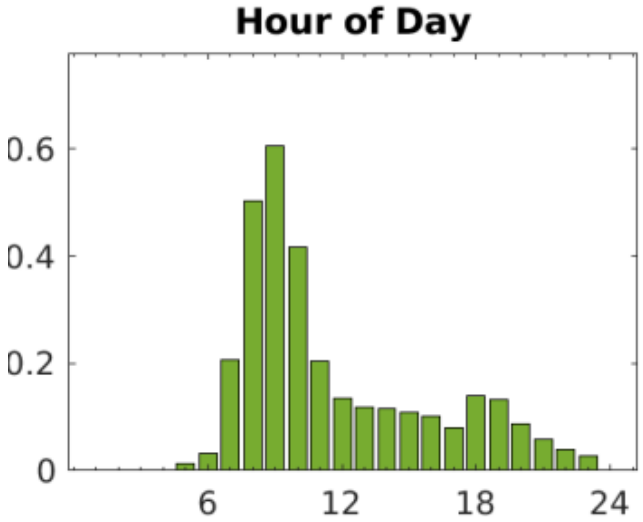
Component #5 (of 50): Theoretical High Energy Physics

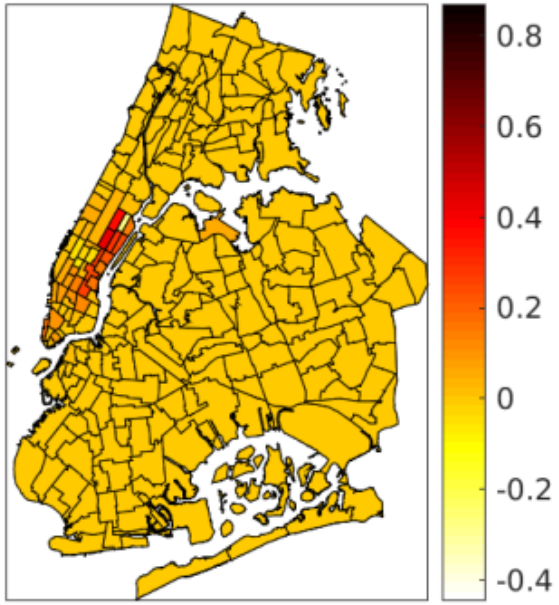
Top Categories	Category Weight	Top Words	Word Weight
hep-th	1.00	theory	1.00
hep-ph	0.03	field	0.81
gr-qc	0.01	model	0.61
		gauge	0.54
		non	0.52
		study	0.51
		find	0.49
		result	0.47
		space	0.47
		use	0.43
		dimensional	0.41
		term	0.40
		string	0.40
		solution	0.37
		case	0.37
		dimension	0.37
		quantum	0.36
		discuss	0.36
		obtain	0.35
		symmetry	0.35

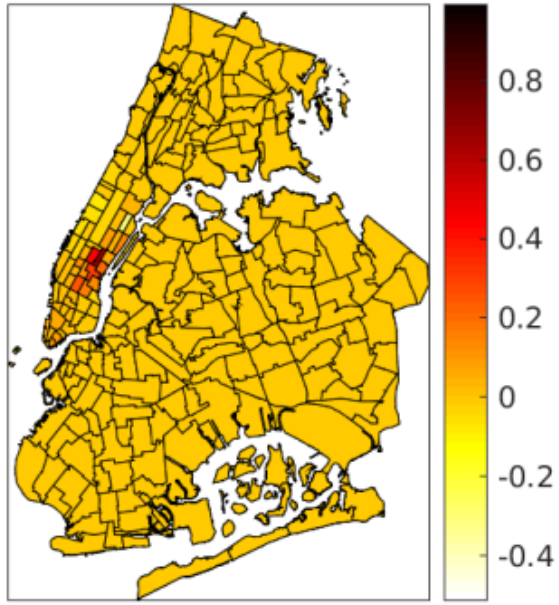


Component #4 (of 50): Morning Commute to Rockefeller Center

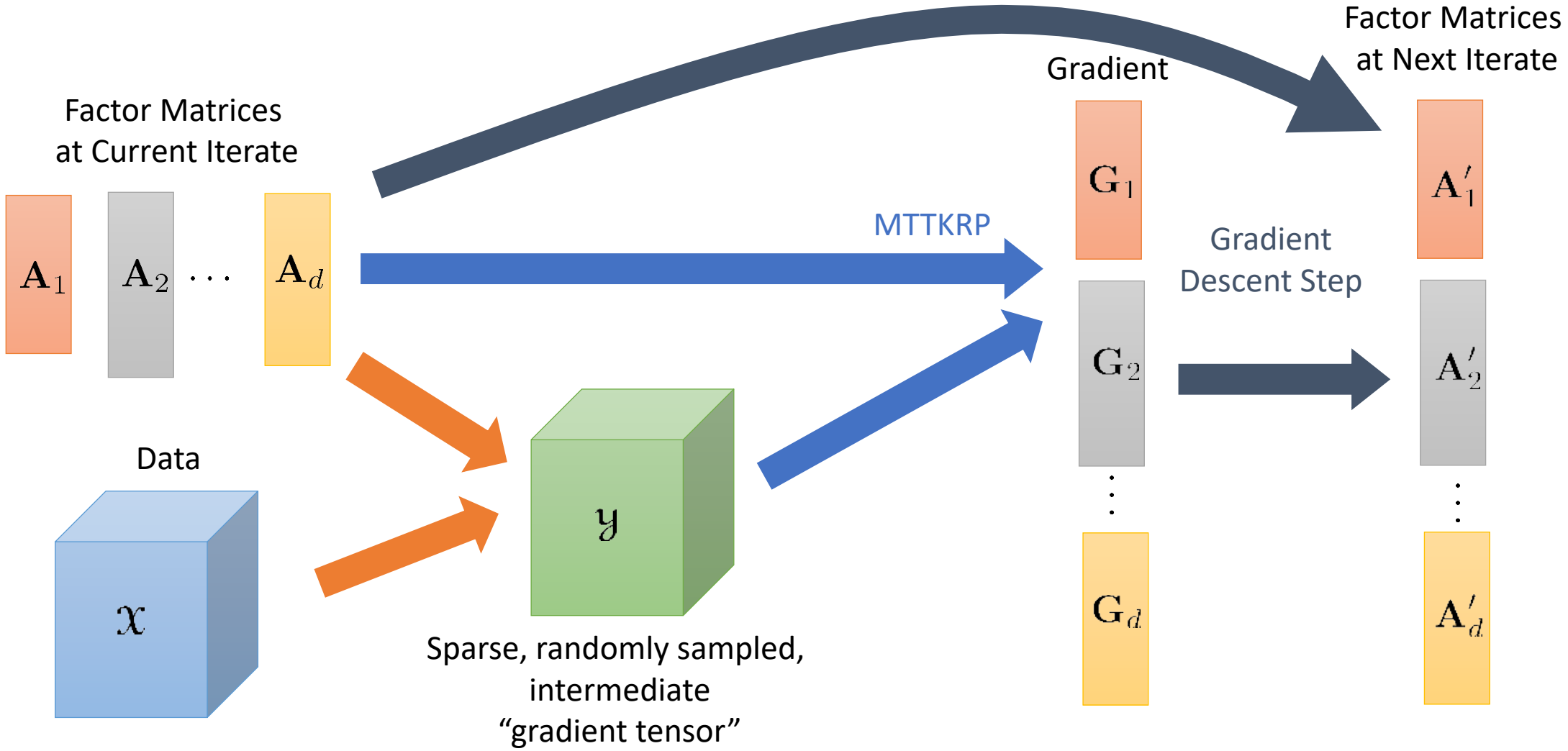




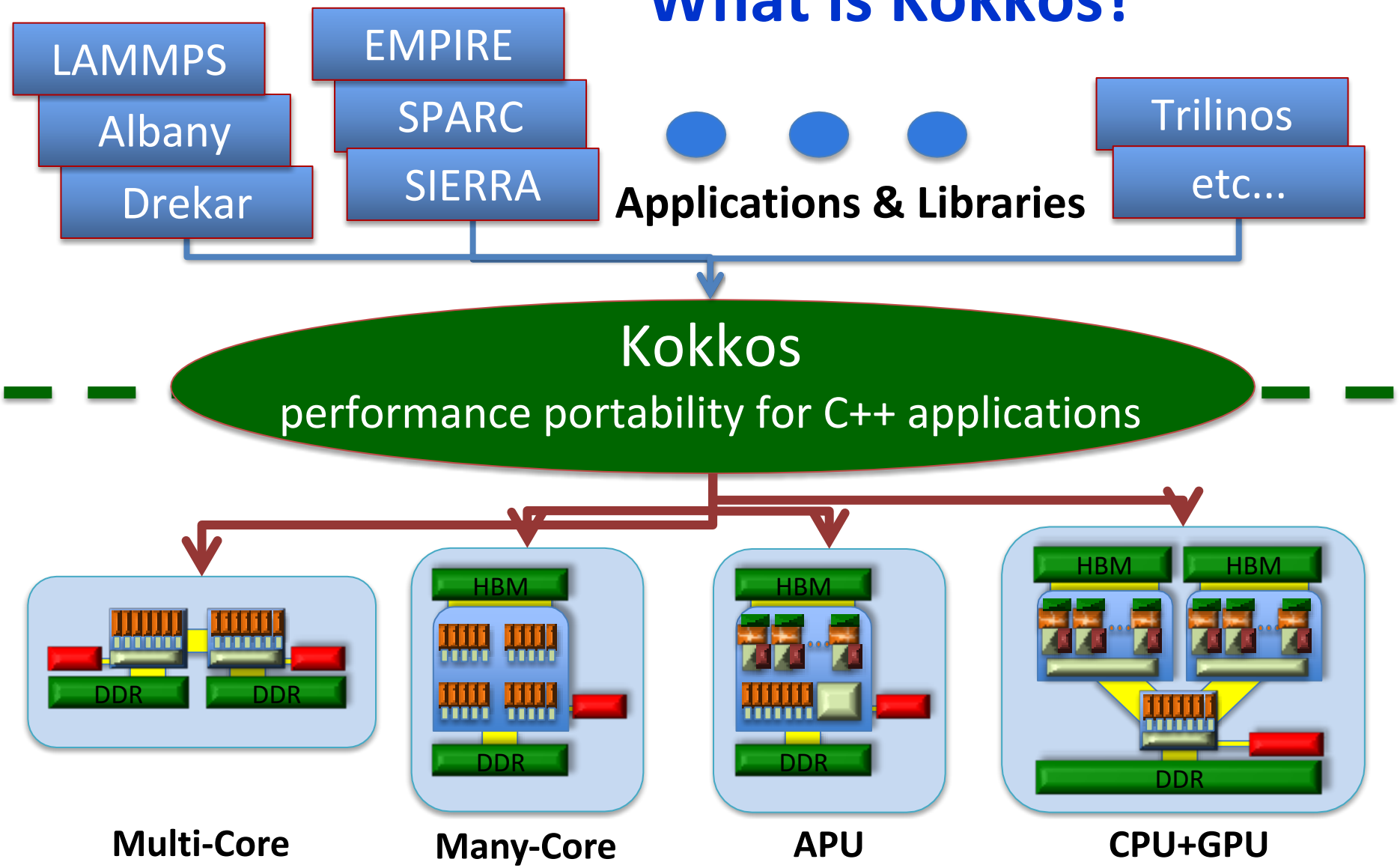




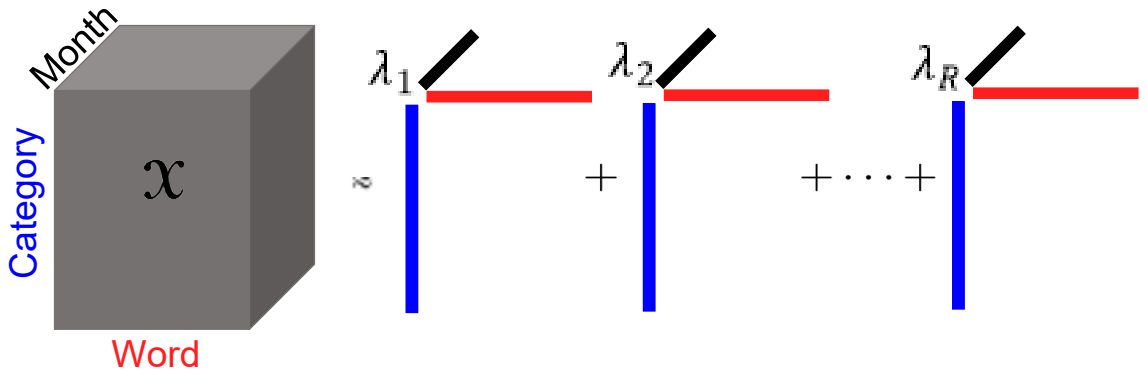
High-level View of GCP Optimization & Dependencies



What is Kokkos?



C. Trott, *et al.*, https://github.com/kokkos/kokkos


Month
Category
Word
$\mathcal{X}$
$\approx$
λ_1
$+$
λ_2
$+ \cdots +$
λ_R