



MENS
MANUS AND
MACHINA

Modeling Temporal Correlations and Dynamics in Spatiotemporal Data Systems

Xinyu Chen

Postdoctoral Associate, MIT

May 16, 2024

Outline

A quick look:

- Motivation (data, task, and models)
- Traffic data imputation with global/local trend modeling
 - Traffic flow imputation, speed field reconstruction, and network traffic state estimation
- Unsupervised pattern discovery from spatiotemporal systems
 - Time-varying autoregression & tensor factorization
 - Applications to fluid flow (benchmark), sea surface temperature, USA climate, NYC taxi, international trade, and Chicago ridesharing
- Possible directions & collaboration:
 - Clifford product & convolution
 - Graphs, hypergraphs, and higher-order motifs

Motivation

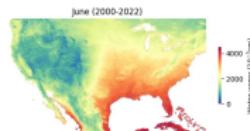
- Spatiotemporal systems & data scenarios



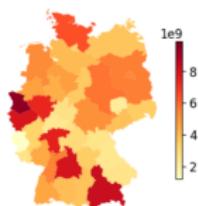
Transportation



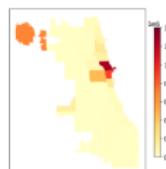
International trade



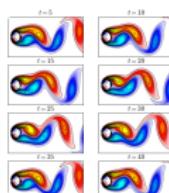
Climate



Energy



Mobility



Fluid flow

- Challenges: Sparsity, time-varying system, multidimensional system (e.g., human mobility)

Prior Art

- Sequence models: Time series autoregression, LSTM, attention-based sequence models, etc.
- Machine learning problems:
 - **Imputation/Interpolation:** Time series models, sparse learning (e.g., matrix/tensor factorization), deep learning (e.g., generative models), etc.
 - **Unsupervised pattern discovery:** Dynamic mode decomposition in dynamical systems, matrix/tensor factorization, etc.
 - **Prediction:** Almost deep learning, but depending on scenarios

Laplacian Convolutional Representation for Traffic Time Series Imputation

IEEE Transactions on Knowledge and Data Engineering, 2024

<https://doi.org/10.1109/TKDE.2024.3419698>



Xinyu Chen



Zhanhong Cheng



HanQin Cai



Nicolas Saunier



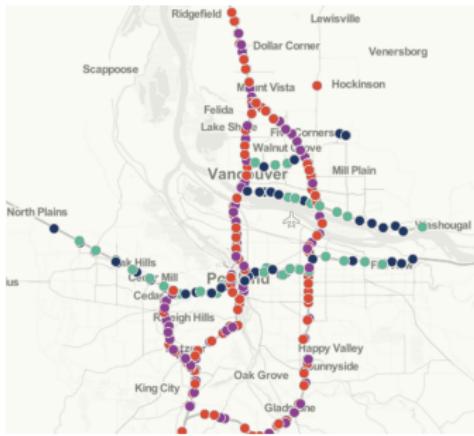
Lijun Sun

Materials:

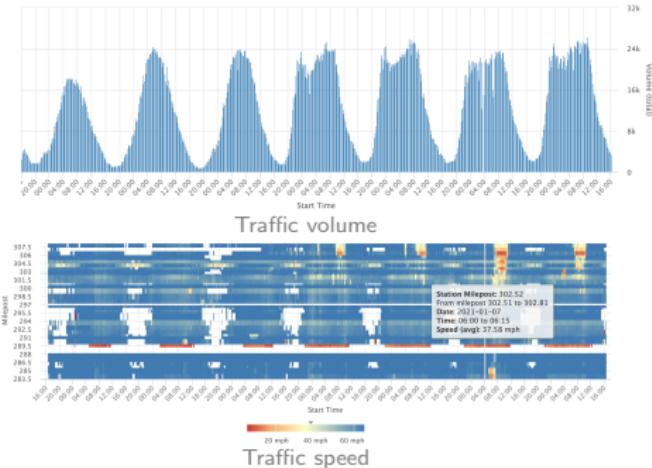
- PDF: https://xinyuchen.github.io/papers/Laplacian_convolution.pdf
- GitHub: <https://github.com/xinyuchen/transdim> (1.1k+ stars)
- Blog: <https://spatiotemporal-data.github.io/posts/LCR/> (coming soon)

Traffic Flow Data

- Portland highway traffic data¹



Highway network & sensor locations



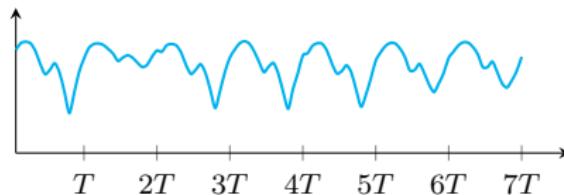
- $X \in \mathbb{R}^{N \times T}$ with N spatial locations $\times T$ time steps
 - Traffic volume/speed shows strong spatial/temporal dependencies

¹<https://portal.its.pdx.edu/home>

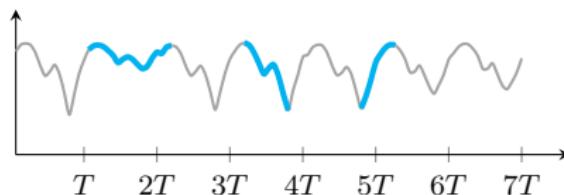
Time Series Imputation

Motivation: Traffic imputation

- Global trends (e.g., long-term quasi-seasonality & daily/weekly rhythm):



- Local trends (e.g., short-term time series trends):



How to characterize both global and local trends in sparse time series?

Local Trend Modeling

- Intuition of (circulant) Laplacian matrix

Undirected and circulant graph

Modeling

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

(Circulant) Laplacian matrix

- Define Laplacian kernel:

$$\boldsymbol{\ell} \triangleq (2, -1, 0, 0, -1)^\top$$

⇓

$$\boldsymbol{\ell} \triangleq (\underbrace{2\tau}_{\text{degree}}, \underbrace{-1, \dots, -1}_\tau, 0, \dots, 0, \underbrace{-1, \dots, -1}_\tau)^\top \in \mathbb{R}^T$$

for any time series $\mathbf{x} = (x_1, \dots, x_T)^\top \in \mathbb{R}^T$.

- Temporal regularization (w/ circular convolution \star):

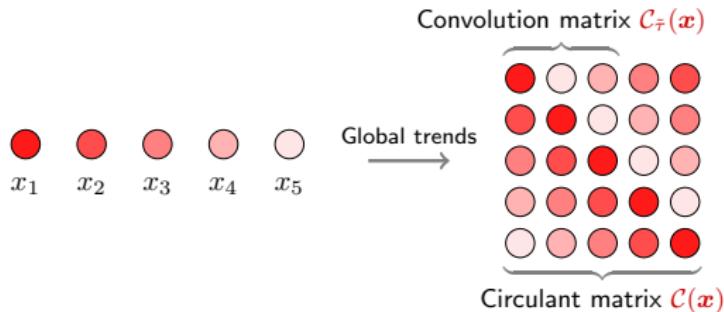
$$\mathcal{R}_\tau(\mathbf{x}) = \frac{1}{2} \|\mathbf{L}\mathbf{x}\|_2^2 = \frac{1}{2} \|\boldsymbol{\ell} \star \mathbf{x}\|_2^2$$

“... The circulant graph has an adjacency matrix that is a circulant matrix.”

— Circulant graph on Wikipedia

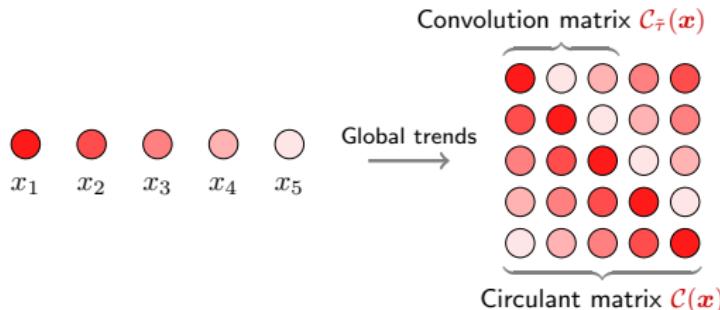
Global Trend Modeling

Circulant matrix $\mathcal{C}(\mathbf{x})$ vs. convolution matrix $\mathcal{C}_{\tilde{\tau}}(\mathbf{x})$



Global Trend Modeling

Circulant matrix $\mathcal{C}(\mathbf{x})$ vs. convolution matrix $\mathcal{C}_{\tilde{\tau}}(\mathbf{x})$



- Circulant/Convolution nuclear norm minimization
 - A balance between global and local trends modeling?

CircNNM (Liu'22, Liu & Zhang'23)

Estimating \mathbf{x} :

$$\begin{aligned}\min_{\mathbf{x}} \quad & \|\mathcal{C}(\mathbf{x})\|_* \\ \text{s.t. } & \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon\end{aligned}$$

on data \mathbf{y} w/ observed index set Ω .

ConvNNM (Liu'22, Liu & Zhang'23)

Estimating \mathbf{x} :

$$\begin{aligned}\min_{\mathbf{x}} \quad & \|\mathcal{C}_{\tilde{\tau}}(\mathbf{x})\|_* \\ \text{s.t. } & \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon\end{aligned}$$

on data \mathbf{y} w/ observed index set Ω .

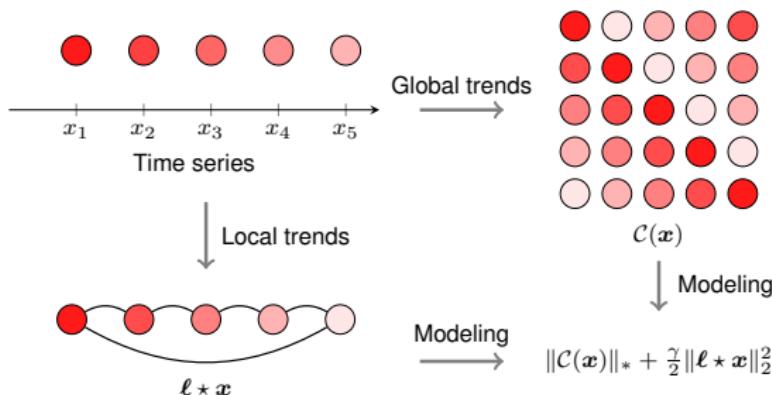
Global + Local Trends?

Laplacian Convolutional Representation (LCR)

For any partially observed time series $\mathbf{y} \in \mathbb{R}^T$ with observed index set Ω , LCR utilizes **circulant matrix** and **Laplacian kernel** to characterize global and local trends in time series, respectively, i.e.,

$$\min_{\mathbf{x}} \underbrace{\|\mathcal{C}(\mathbf{x})\|_*}_{\text{global}} + \frac{\gamma}{2} \underbrace{\|\ell * \mathbf{x}\|_2^2}_{\text{local}}$$

s.t. $\|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon$



Laplacian Convolutional Representation

- Augmented Lagrangian function:²

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) = \|\mathcal{C}(\mathbf{x})\|_* + \frac{\gamma}{2} \|\ell * \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \langle \mathbf{w}, \mathbf{x} - \mathbf{z} \rangle + \frac{\eta}{2} \|\mathcal{P}_\Omega(\mathbf{z} - \mathbf{y})\|_2^2$$

- The ADMM scheme:

$$\begin{cases} \mathbf{x} := \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) & \text{(Nuclear norm minimization)} \\ \mathbf{z} := \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) & \text{(Closed-form solution)} \\ \mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z}) & \text{(Standard update)} \end{cases}$$

- Optimize \mathbf{x} ?

$$\underbrace{\|\mathcal{C}(\mathbf{x})\|_* = \|\mathcal{F}(\mathbf{x})\|_1}_{\text{property of circulant matrix}} \quad \& \quad \underbrace{\frac{1}{2} \|\ell * \mathbf{x}\|_2^2 = \frac{1}{2T} \|\mathcal{F}(\ell) \circ \mathcal{F}(\mathbf{x})\|_2^2}_{\text{property of circular convolution}}$$

Nuclear norm minimization \Rightarrow **ℓ_1 -norm minimization with FFT** in $\mathcal{O}(T \log T)$ time.

² $\mathbf{w} \in \mathbb{R}^T$ (Lagrange multiplier); $\langle \cdot, \cdot \rangle$ (inner product).

Laplacian Convolutional Representation

- Optimize \mathbf{x} via FFT (in $\mathcal{O}(T \log T)$ time):

$$\begin{aligned}\mathbf{x} &:= \arg \min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_* + \frac{\gamma}{2} \|\ell * \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{w}/\lambda\|_2^2 \\ \implies \hat{\mathbf{x}} &:= \arg \min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1 + \frac{\gamma}{2T} \|\hat{\ell} \circ \hat{\mathbf{x}}\|_2^2 + \frac{\lambda}{2T} \|\hat{\mathbf{x}} - \hat{\mathbf{z}} + \hat{\mathbf{w}}/\lambda\|_2^2\end{aligned}$$

where we introduce $\{\hat{\ell}, \hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{w}}\} \triangleq \mathcal{F}\{\ell, \mathbf{x}, \mathbf{z}, \mathbf{w}\}$ (i.e., FFT).

ℓ_1 -norm Minimization in Complex Space (Liu & Zhang'23)

For any optimization problem in the form of ℓ_1 -norm minimization in complex space:

$$\min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1 + \frac{\delta}{2} \|\hat{\mathbf{x}} - \hat{\mathbf{h}}\|_2^2$$

with complex-valued $\hat{\mathbf{x}}, \hat{\mathbf{h}} \in \mathbb{C}^T$ and weight parameter δ , element-wise, the solution is given by

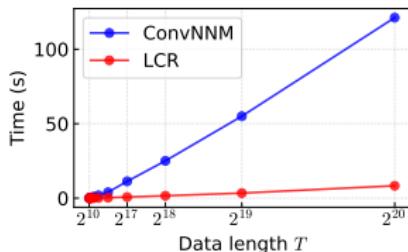
$$\hat{x}_t := \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - 1/\delta\}, t \in [T].$$

Laplacian Convolutional Representation

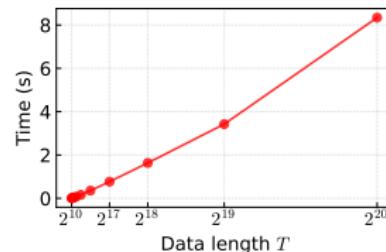
Empirical time complexity

On the synthetic data $\mathbf{y} \in \mathbb{R}^T$ with $T \in \{2^{10}, 2^{11}, \dots, 2^{20}\}$

- Ours: **LCR**
 - An FFT implementation in $\mathcal{O}(T \log T)$
 - The logarithmic factor $\log T$ makes the FFT highly efficient
- Baseline: **ConvNNM** (Liu'22, Liu & Zhang'23)
 - Convolution matrix $\mathcal{C}_{\tilde{\tau}}(\mathbf{y}) \in \mathbb{R}^{T \times \tilde{\tau}}$ with kernel size $\tilde{\tau} = 2^4$
 - Singular value thresholding in $\mathcal{O}(\tilde{\tau}^2 T)$

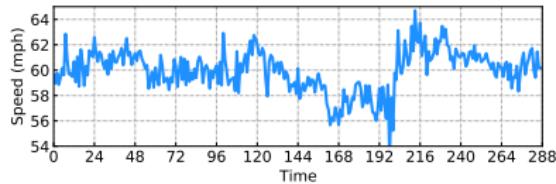
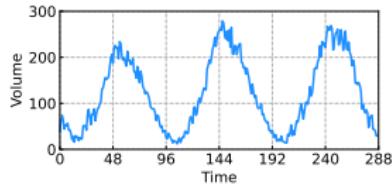
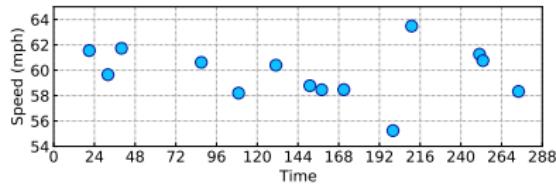
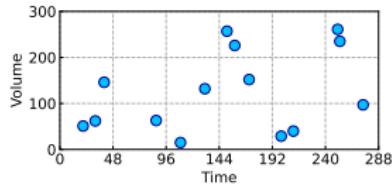


ConvNNM vs. LCR



LCR

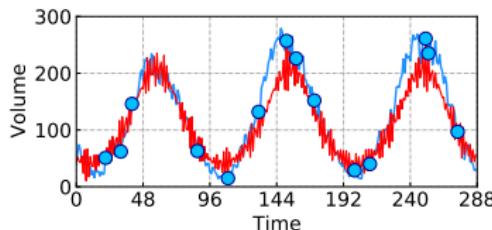
Experiments



- How to utilize the global trends of traffic time series?
- How to produce local consistency of traffic data?

Experiments

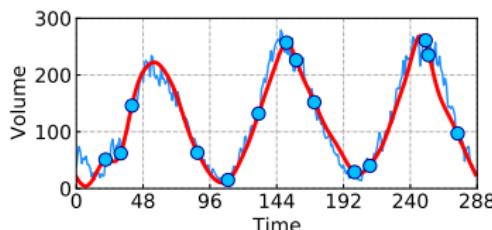
- Substantial performance gains?



CircNNM:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\mathcal{C}(\boldsymbol{x})\|_* \\ \text{s. t. } \quad & \|\mathcal{P}_\Omega(\boldsymbol{x} - \boldsymbol{y})\|_2 \leq \epsilon \end{aligned}$$

↓ Plus **local** time series trends



LCR:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\mathcal{C}(\boldsymbol{x})\|_* + \frac{\gamma}{2} \|\boldsymbol{\ell} * \boldsymbol{x}\|_2^2 \\ \text{s. t. } \quad & \|\mathcal{P}_\Omega(\boldsymbol{x} - \boldsymbol{y})\|_2 \leq \epsilon \end{aligned}$$

Experiments

- The start data points and end data points are connected?

Undirected and circulant graph

Modeling →

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

(Circulant) Laplacian matrix

- Flipping operation on $\mathbf{x} \in \mathbb{R}^5$:

$$\mathbf{x}_{\text{new}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{Jx} \end{bmatrix} = (\underbrace{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5}_{\text{original time series}}, \underbrace{\mathbf{x}_5, \mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1}_{\text{flipped time series}})^{\top} \in \mathbb{R}^{10}$$

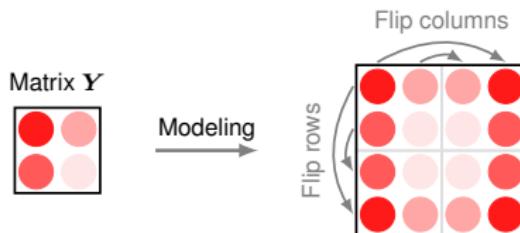
where $\mathbf{J} \in \mathbb{R}^{5 \times 5}$ is the exchange matrix.

- Potential applications: Passenger flow prediction with strong global/local trends

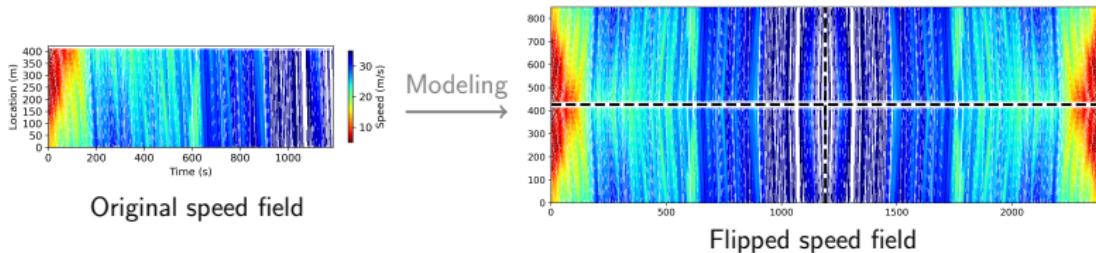
Experiments

Speed field reconstruction³

- Flipping operation on a matrix:



- Flipping operation on a speed field of vehicular traffic flow:



³Highway Drone (HighD) dataset at <https://www.hightd-dataset.com/>

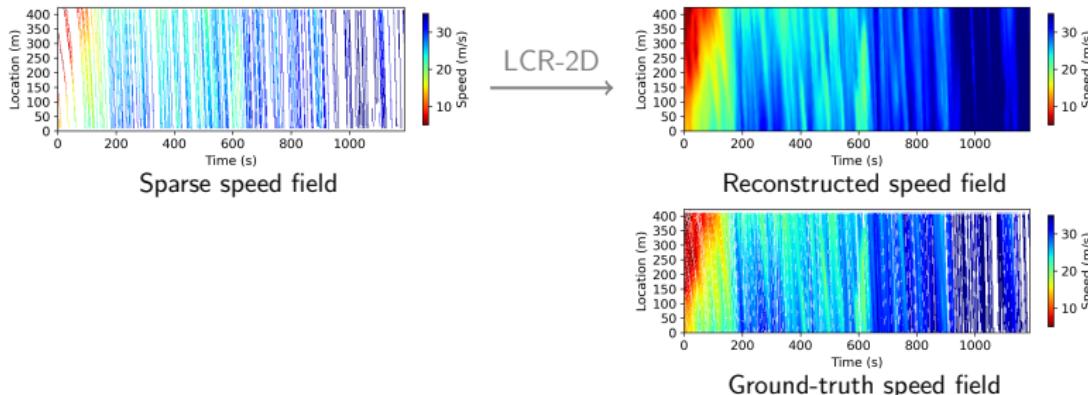
Experiments

Speed field reconstruction⁴

- Scenario: Mask trajectories of 70% vehicles
- LCR-2D on partially observed $\mathbf{Y} \in \mathbb{R}^{N \times T}$:

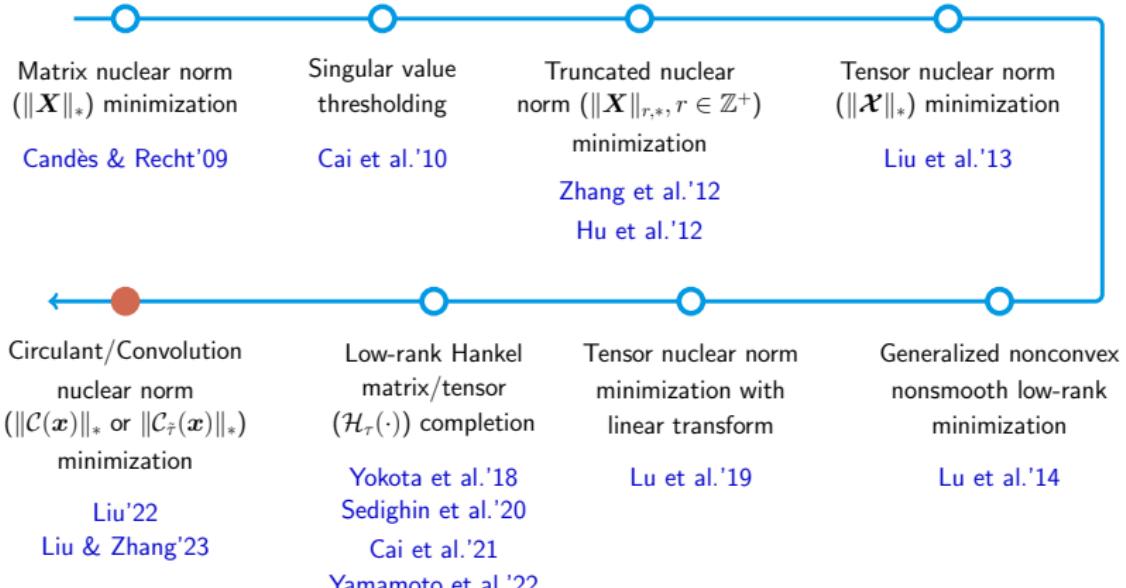
$$\min_{\mathbf{X}} \underbrace{\|\mathcal{C}(\mathbf{X})\|_*}_{\text{global trend}} + \frac{\gamma}{2} \underbrace{\|(\ell_s \ell^\top) * \mathbf{X}\|_F^2}_{\text{local trend}}$$

s.t. $\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{Y})\|_F \leq \epsilon$



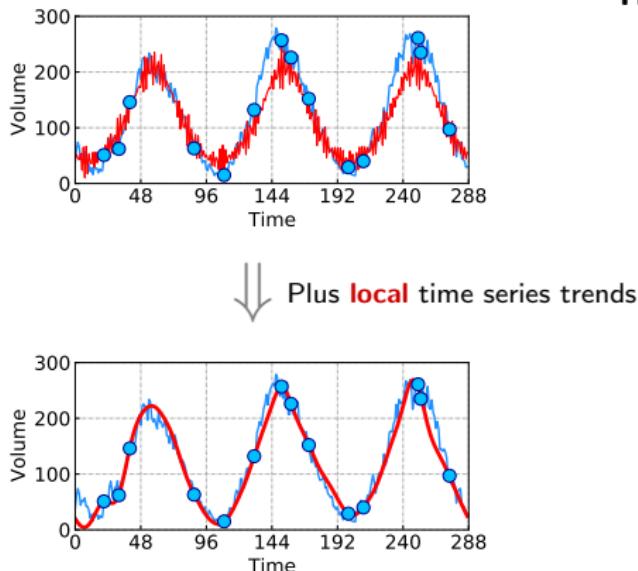
⁴Highway Drone (HighD) dataset at <https://www.hightd-dataset.com/>

Contributions



(Ours) LCR:

- ✓ Local trend modeling
- ✓ An FFT implementation



Highlights:

- Rethinking the importance of local trend modeling in traffic data imputation tasks.
- Finding a unified **global and local trend** modeling framework whose optimization can be efficiently solved by **FFT**:

$$\min_{\mathbf{x}} \underbrace{\|\mathcal{C}(\mathbf{x})\|_*}_{\text{global}} + \frac{\gamma}{2} \underbrace{\|\ell * \mathbf{x}\|_2^2}_{\text{local}}$$

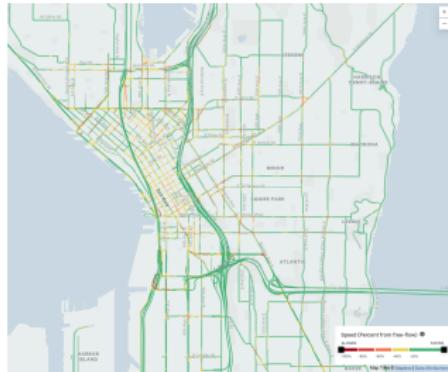
$$\text{s. t. } \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon$$

Vision & Insight

- Uber (hourly) movement speed data⁵



NYC movement



Seattle movement

- {road segment, time slot (hour), average speed}
- Computing hourly speed: Road segments have 5+ unique trips.
- Estimating network-wide traffic states for traffic planning & management.
(Data/model biases/fairness concerns for imputation, interpolation, and prediction.)

⁵<https://movement.uber.com/> (not available now)

Discovering Dynamic Patterns from Spatiotemporal Data with Time-Varying Low-Rank Autoregression

IEEE Transactions on Knowledge and Data Engineering, 2024

<https://doi.org/10.1109/TKDE.2023.3294440>



Xinyu Chen



Chengyuan Zhang*



Xiaoxu Chen



Nicolas Saunier



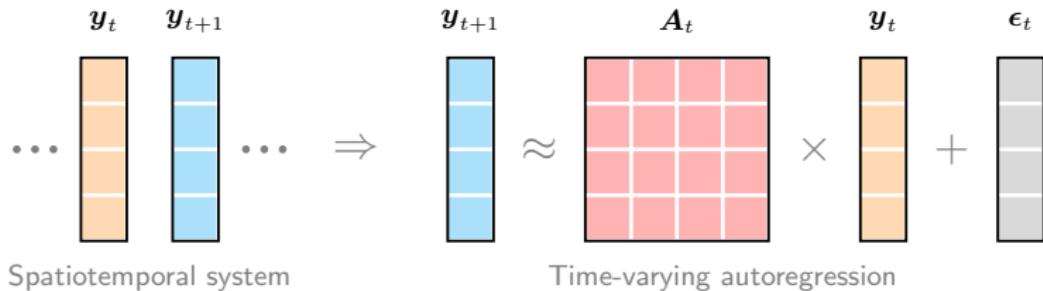
Lijun Sun

Materials:

- PDF: https://xinyuchen.github.io/papers/time_varying_model.pdf
- GitHub: <https://github.com/xinyuchen/vars>
- Blog:
https://spatiotemporal-data.github.io/posts/time_varying_model

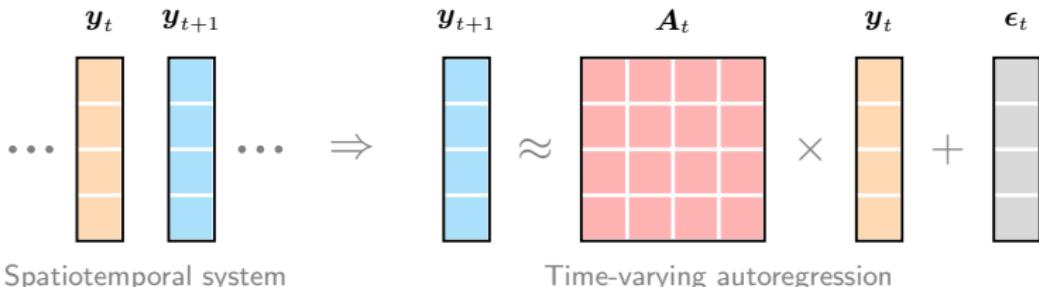
Autoregression

- How to characterize dynamical systems?



Autoregression

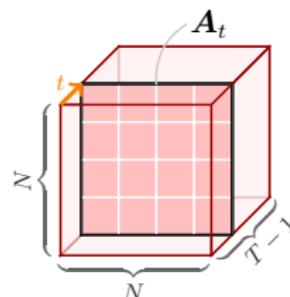
- How to characterize dynamical systems?

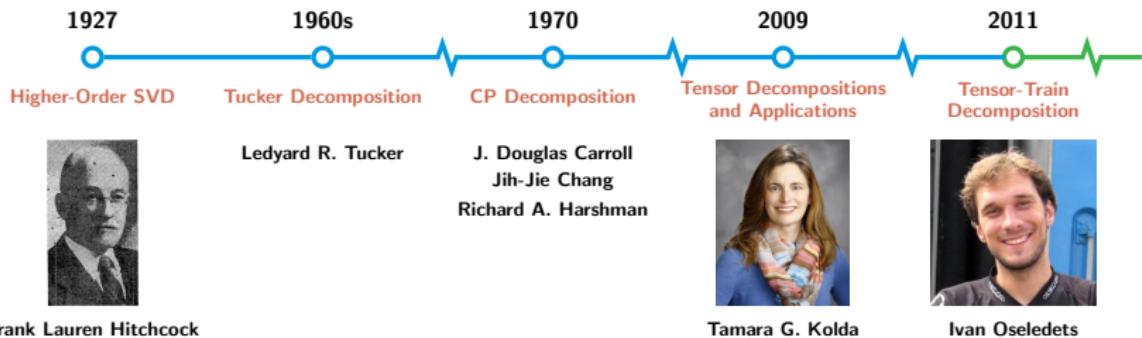


- On spatiotemporal systems $\mathbf{Y} \in \mathbb{R}^{N \times T}$:

$$\underbrace{\mathbf{y}_{t+1} = \mathbf{A}\mathbf{y}_t + \epsilon_t}_{\text{time-invariant (e.g., DMD)}} \quad \text{v.s.} \quad \underbrace{\mathbf{y}_{t+1} = \mathbf{A}_t \mathbf{y}_t + \epsilon_t}_{\text{time-varying}}$$

- How to discover spatial/temporal modes (patterns) from the tensor $\mathcal{A} \triangleq \{\mathbf{A}_t\}_{t \in [T-1]}$?

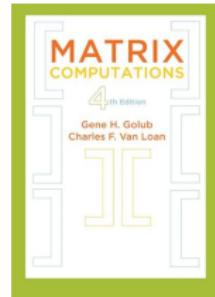




Time-Varying Autoregression

- Tensor factorization⁶:

$$\mathcal{A} = \underbrace{\mathcal{G} \times_1 \mathbf{W} \times_2 \mathbf{V} \times_3 \mathbf{X}}_{\text{Tucker decomposition}}$$
$$\Updownarrow$$
$$\mathbf{A}_t = \mathcal{G} \times_1 \underbrace{\mathbf{W}}_{\text{spatial modes}} \times_2 \mathbf{V} \times_3 \underbrace{\mathbf{x}_t^\top}_{\text{temporal modes}}$$



- (Ours) Time-varying low-rank autoregression:

$$\min_{\mathcal{G}, \mathbf{W}, \mathbf{V}, \mathbf{x}} \frac{1}{2} \sum_{t \in [T-1]} \| \mathbf{y}_{t+1} - (\mathcal{G} \times_1 \mathbf{W} \times_2 \mathbf{V} \times_3 \mathbf{x}_t^\top) \mathbf{y}_t \|_2^2$$

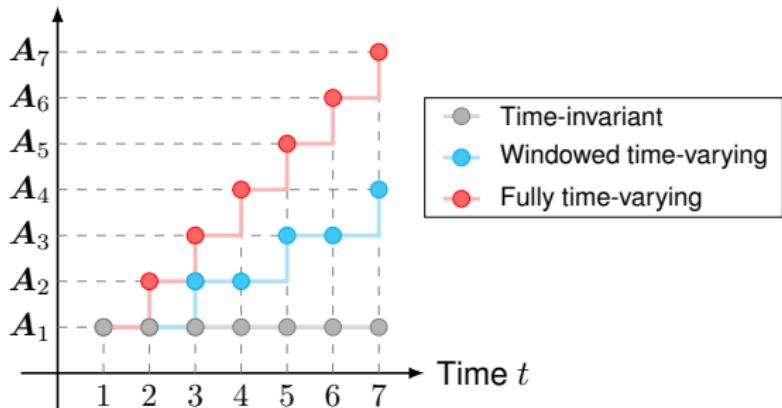
- Alternating minimization: \mathcal{G} (LS) \rightarrow \mathbf{W} (LS) \rightarrow \mathbf{V} (CG) \rightarrow \mathbf{x}_t (LS)

⁶ \times_k , $\forall k$ is the mode- k product between tensor and matrix/vector.

- On the data $\mathbf{Y} \in \mathbb{R}^{N \times T}$:

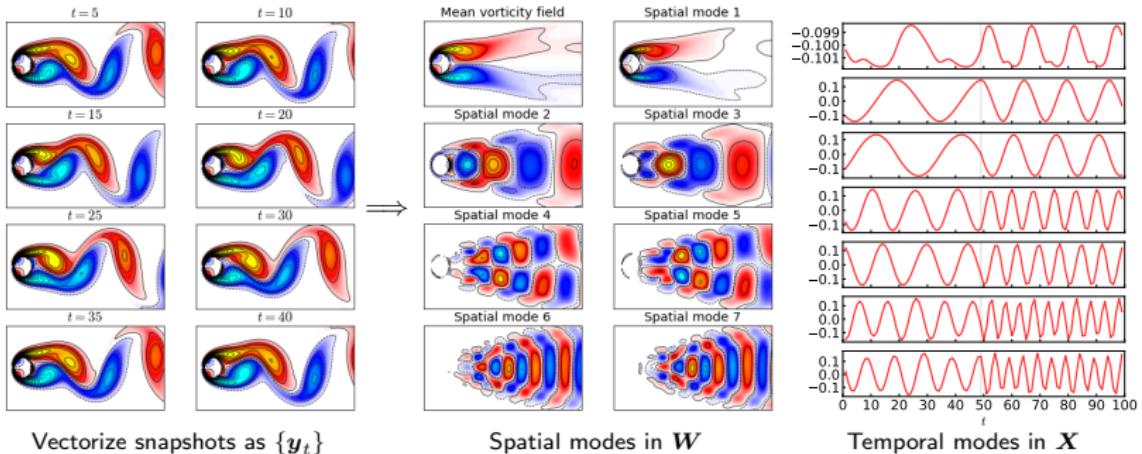
$$\underbrace{\mathbf{y}_{t+1} = \mathbf{A}\mathbf{y}_t + \epsilon_t}_{\text{time-invariant (e.g., DMD)}} \quad \text{v.s.} \quad \underbrace{\mathbf{y}_{t+1} = \mathbf{A}_t \mathbf{y}_t + \epsilon_t}_{\text{fully time-varying (ours)}}$$

Coefficients



Fluid Flow

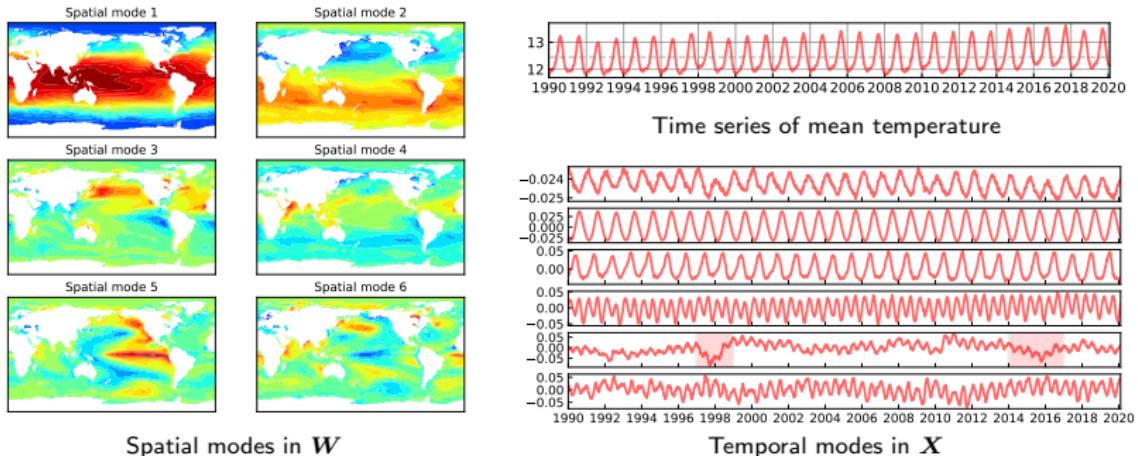
- **Multiresolution fluid flow dataset** (the first 50 snapshots + 50 snapshots randomly selected from the last 100 snapshots)



- Produce interpretable patterns and identify the system of different frequencies.

Sea Surface Temperature

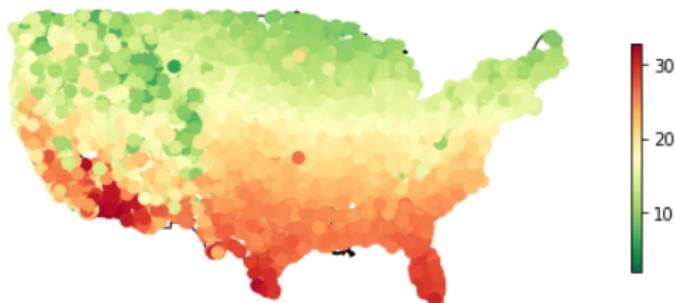
- Sea surface temperature (**SST**) dataset



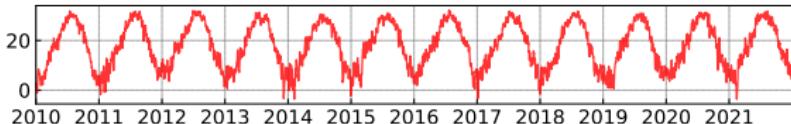
- Identify two strongest El Nino events (on 1997-98 & 2014-16).

USA Temperature

- Data⁷: 5,380 stations & 12 years with the day resolution
- Spatial distribution of mean temperature

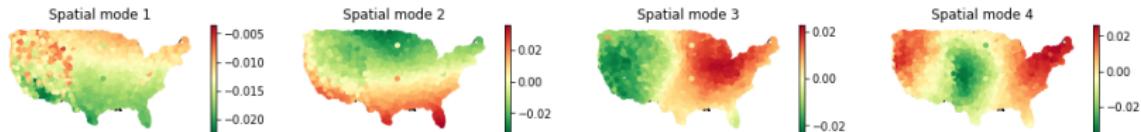


- Time series of mean temperatures

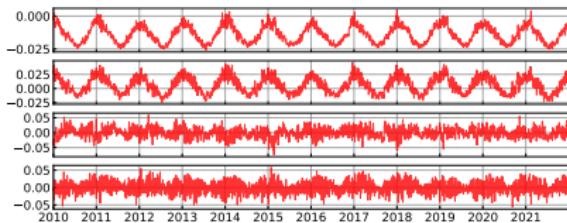


⁷<https://daac.ornl.gov/DAYMET>

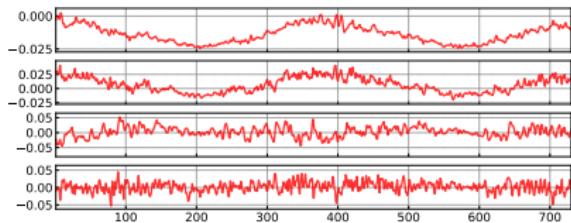
USA Temperature



Spatial patterns in W



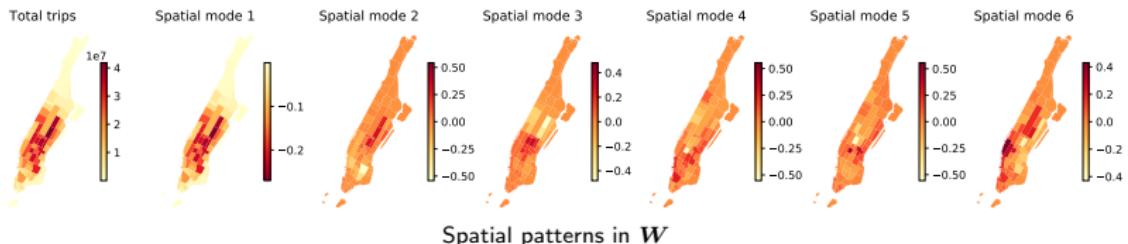
Temporal patterns over 12 years



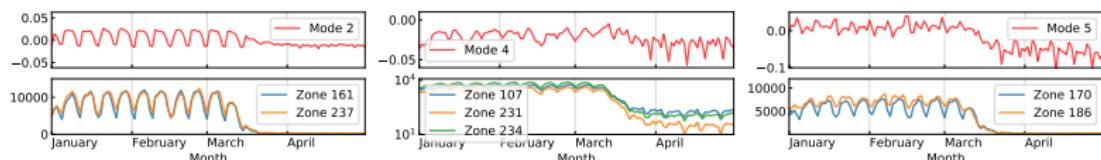
Temporal patterns over 2 years

NYC Taxi Data

- NYC taxi dataset (pickup)



Spatial patterns in W



Pattern #2 & taxi trips (2020)

Pattern #4 & taxi trips (2020)

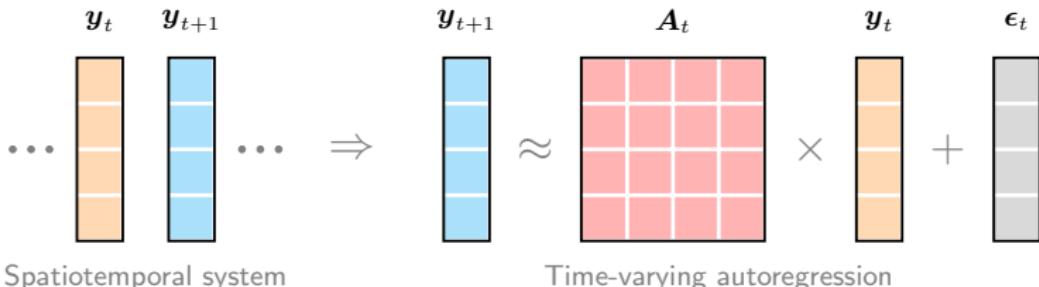
Pattern #5 & taxi trips (2020)

Dynamic Autoregressive Tensor Factorization for Pattern Discovery of Spatiotemporal Systems

International Trade & Ridesharing Mobility

Autoregression

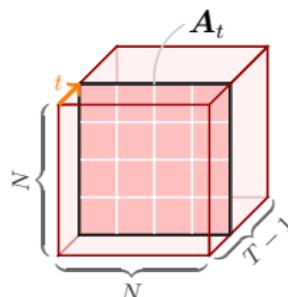
- How to characterize dynamical systems?



- On spatiotemporal systems $\mathbf{Y} \in \mathbb{R}^{N \times T}$:

$$\underbrace{y_{t+1} = \mathbf{A}y_t + \epsilon_t}_{\text{time-invariant (e.g., DMD)}} \quad \text{v.s.} \quad \underbrace{y_{t+1} = \mathbf{A}_t y_t + \epsilon_t}_{\text{time-varying}}$$

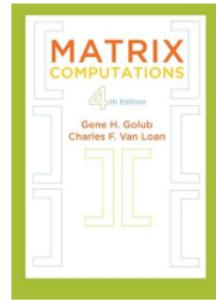
- How to discover spatial/temporal modes (patterns) from the tensor $\mathcal{A} \triangleq \{\mathbf{A}_t\}_{t \in [T-1]}$?



DATF

- Tensor factorization:

$$\mathcal{A} = \underbrace{\mathcal{G} \times_1 \mathbf{W} \times_2 \mathbf{V} \times_3 \mathbf{X}}_{\text{Tucker decomposition}}$$
$$\Downarrow$$
$$\mathbf{A}_t = \mathcal{G} \times_1 \underbrace{\mathbf{W}}_{\text{spatial modes}} \times_2 \mathbf{V} \times_3 \underbrace{\mathbf{x}_t^\top}_{\text{temporal modes}}$$



- (Ours) Dynamic autoregressive tensor factorization (DATF):

$$\min_{\mathcal{G}, \mathbf{W}, \mathbf{V}, \mathbf{X}} \frac{1}{2} \sum_{t \in [T-1]} \|\mathbf{y}_{t+1} - (\mathcal{G} \times_1 \mathbf{W} \times_2 \mathbf{V} \times_3 \mathbf{x}_t^\top) \mathbf{y}_t\|_2^2$$

s.t. $\underbrace{\mathbf{W}^\top \mathbf{W} = \mathbf{I}_R}_{\text{orthogonal spatial modes}}$

- Solution: \mathcal{G} (LS) \rightarrow \mathbf{W} (OPP) \rightarrow \mathbf{V} (CG) \rightarrow \mathbf{x}_t (LS)

- **Orthogonal Procrustes problem**

(OPP): For any $\mathbf{Q} \in \mathbb{R}^{m \times r}$, $m \geq r$,
the solution to

$$\min_{\mathbf{F}} \|\mathbf{F} - \mathbf{Q}\|_F^2$$

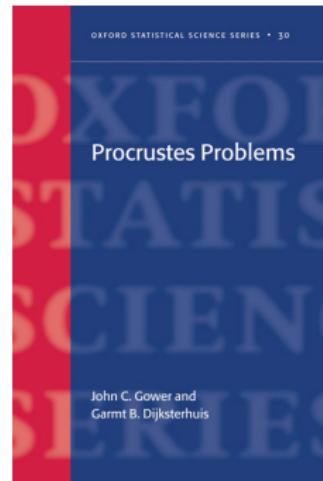
$$\text{s. t. } \underbrace{\mathbf{F}^\top \mathbf{F} = \mathbf{I}_r}_{\text{orthogonal}}$$

is

$$\mathbf{F} := \mathbf{U}\mathbf{V}^\top$$

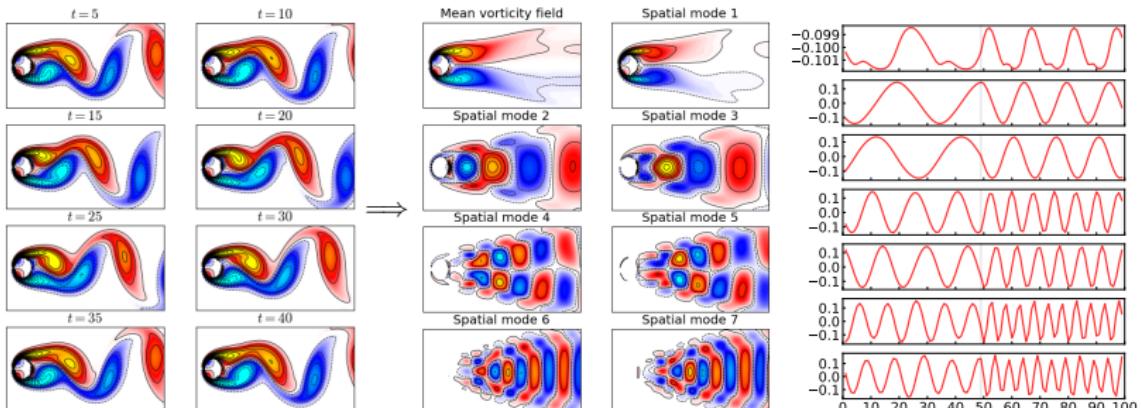
where

$$\underbrace{\mathbf{Q} = \mathbf{U}\Sigma\mathbf{V}^\top}_{\text{singular value decomposition}}$$



Benchmark Evaluation

- **Multi-resolution fluid flow dataset** (the first 50 snapshots + 50 snapshots randomly selected from the last 100 snapshots)
 - Produce interpretable patterns: Low-frequency modes (dominant patterns) & high-frequency modes (e.g., secondary patterns, outliers)
 - Identify the system of different frequencies (i.e., at $t = 50$)



Vectorize snapshots as $\{y_t\}_t$

Spatial modes in W

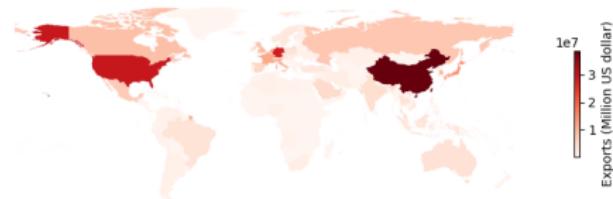
Temporal modes in X

International Trade

- Import/Export merchandise trade values (annual)⁸ (215 countries/regions & period of 2000-2022)
 - Total merchandise trade values
 - Represent import/export trade data as a 215-by-23 matrix



Imports from 2000 to 2022



Exports from 2000 to 2022

⁸The dataset is available at <https://stats.wto.org>.



Import pattern 1



Import pattern 2



Import pattern 3



Import pattern 4



Export pattern 1



Export pattern 2



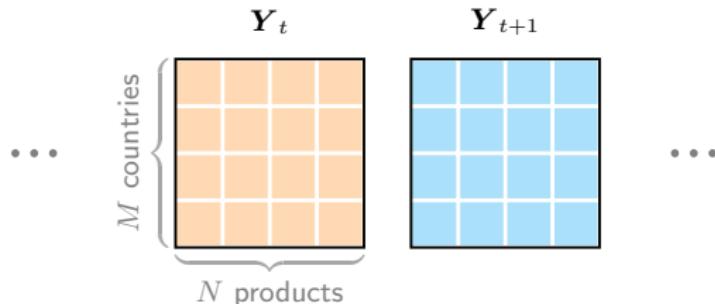
Export pattern 3



Export pattern 4

International Trade

- Three-dimensional trade (Economy, Product, Year)



- On spatiotemporal systems $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$:

$$\mathbf{y}_{n,t+1} = \underbrace{\mathbf{A}_{n,t} \mathbf{y}_{n,t} + \boldsymbol{\epsilon}_{n,t}}_{\text{time-varying \& product-varying}}$$

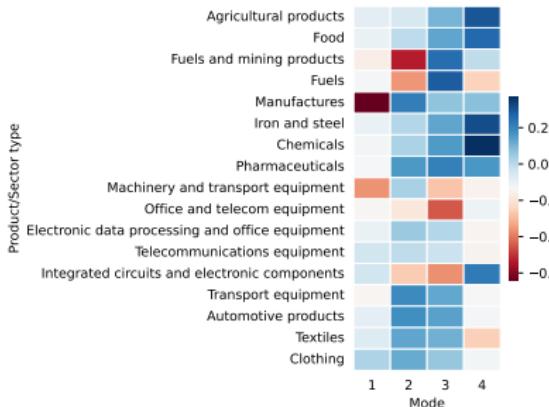
- Optimization problem of DATEF:

$$\min_{\mathcal{G}, \mathbf{W}, \mathbf{U}, \mathbf{V}, \mathbf{x}} \frac{1}{2} \sum_{n \in [N]} \sum_{t \in [T-1]} \|\mathbf{y}_{n,t+1} - (\mathcal{G} \times_1 \mathbf{W} \times_2 \mathbf{U} \times_3 \mathbf{V} \times_4 \mathbf{x}_t^\top) \mathbf{y}_{n,t}\|_2^2$$

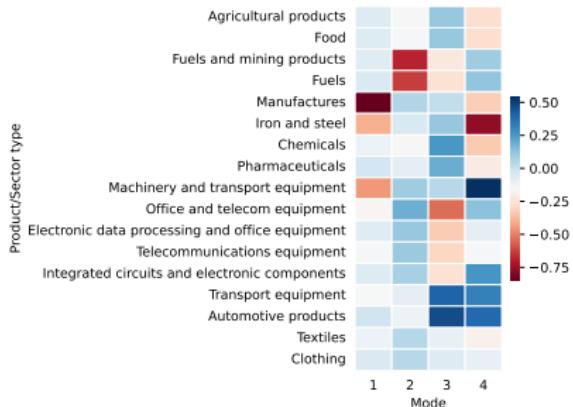
s.t.
$$\underbrace{\mathbf{W}^\top \mathbf{W} = \mathbf{I}_R}_{\text{orthogonal country patterns}}$$

Product Patterns

- On 17 merchandise types



Imports



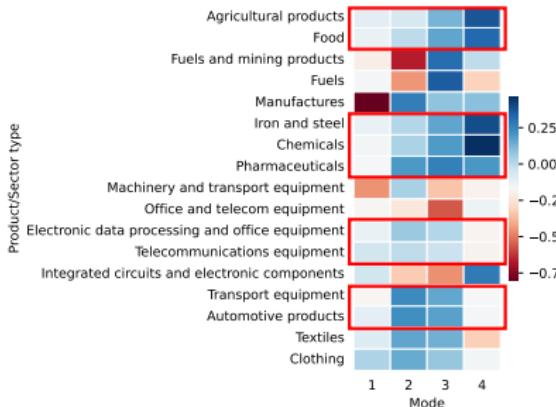
Exports

- Classify import/export merchandise according to product patterns
- Basic principle:

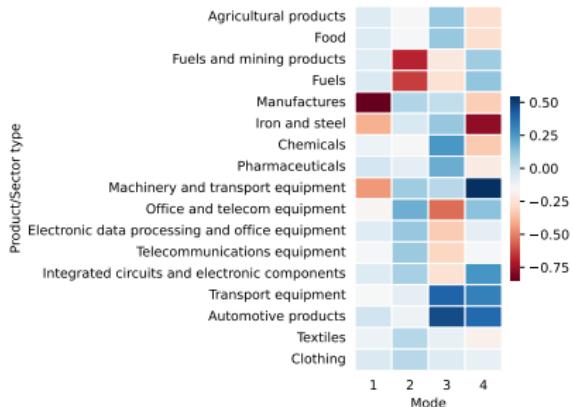
Import: What we buy? (demand) vs. Export: What we sell? (supply)

Product Patterns

- On 17 merchandise types



Imports



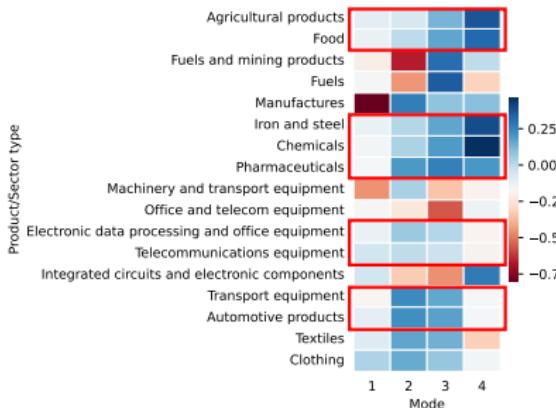
Exports

- Classify import/export merchandise according to product patterns
- Basic principle:

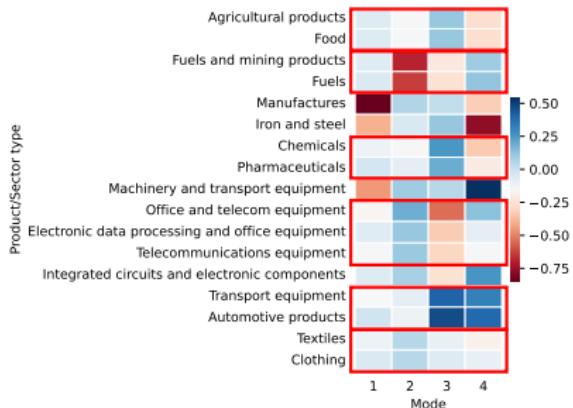
Import: What we buy? (demand) vs. Export: What we sell? (supply)

Product Patterns

- On 17 merchandise types



Imports



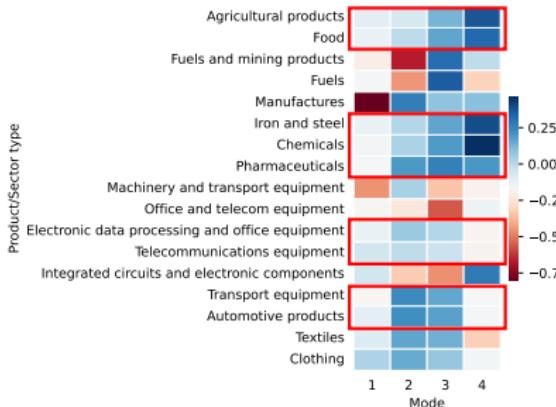
Exports

- Classify import/export merchandise according to product patterns
- Basic principle:

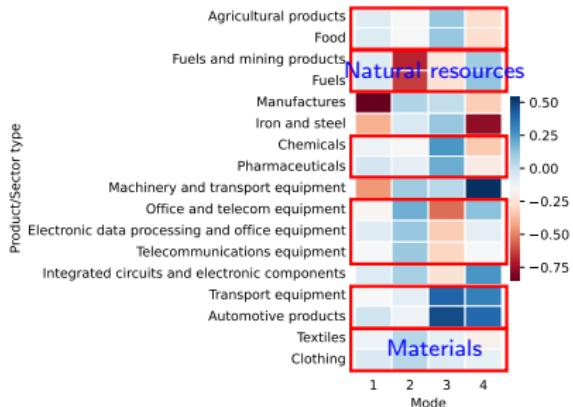
Import: What we buy? (demand) vs. Export: What we sell? (supply)

Product Patterns

- On 17 merchandise types



Imports



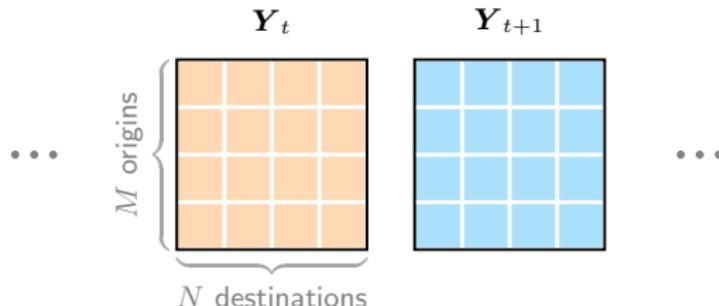
Exports

- Classify import/export merchandise according to product patterns
- Basic principle:

Import: What we buy? (demand) vs. Export: What we sell? (supply)

Human Mobility

- Origin-Destination (OD) matrices



- On spatiotemporal systems $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$:

$$\mathbf{y}_{n,t+1} = \underbrace{\mathbf{A}_{n,t} \mathbf{y}_{n,t} + \boldsymbol{\epsilon}_{n,t}}_{\text{time-varying \& destination-varying}}$$

- Optimization problem of DATF:

$$\min_{\mathcal{G}, \mathbf{W}, \mathbf{U}, \mathbf{V}, \mathbf{x}} \frac{1}{2} \sum_{n \in [N]} \sum_{t \in [T-1]} \left\| \mathbf{y}_{n,t+1} - (\mathcal{G} \times_1 \mathbf{W} \times_2 \mathbf{U} \times_3 \mathbf{V} \times_4 \mathbf{x}_t^\top) \mathbf{y}_{n,t} \right\|_2^2$$

s.t.
$$\underbrace{\mathbf{W}^\top \mathbf{W} = \mathbf{I}_R}_{\text{orthogonal origin patterns}}$$

Human Mobility

• Chicago taxi/ridesharing data

Matching Taxi Trips with Community Areas

There are three basic steps to follow for processing taxi trip data:

- Download taxi trips in 2022 in the `.csv` format, e.g., `Taxi_Trips_-_2022.csv`.
- Use the `pandas` package in Python to process the raw trip data.
- Match trip pickup/dropoff locations with boundaries of the community area.

```
import pandas as pd
data = pd.read_csv('Taxi_Trips_-_2022.csv')
data.head()
```

For each taxi trip, one can select some important information:

- **Trip Start Timestamp:** When the trip started, rounded to the nearest 15 minutes.
- **Trip Seconds:** Time of the trip in seconds.
- **Trip Miles:** Distance of the trip in miles.
- **Pickup Community Area:** The Community Area where the trip began. This column will be blank for locations outside Chicago.
- **Dropoff Community Area:** The Community Area where the trip ended. This column will be blank for locations outside Chicago.

```
# df['Trip Start Timestamp'] = data['Trip Start timestamp']
# df['Trip Seconds'] = data['Trip Seconds']
# df['Trip Miles'] = data['Trip Miles']
# df['Pickup Community Area'] = data['Pickup Community Area']
# df['Dropoff Community Area'] = data['Dropoff Community Area']
# df = data
df
```

Figure 2 shows taxi pickup and dropoff trips (2022) on 77 community areas in the City of Chicago. Note that the average trip duration is 1207.75 seconds and the average trip distance is 8.16 miles.

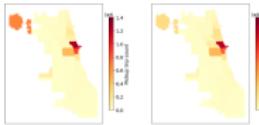


Figure 2. Taxi pickup and dropoff trips (2022) in the City of Chicago, USA. There are 4,763,961 remaining trips after the data processing.

For comparison, Figure 3 shows taxi pickup and dropoff trips (2019) on 77 community areas in the City of Chicago. Note that the average trip duration is 915.62 seconds and the average trip distance is 3.93 miles.

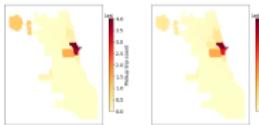


Figure 3. Taxi pickup and dropoff trips (2019) in the City of Chicago, USA. There are 12,484,572 remaining trips after the data processing. See the data processing codes.

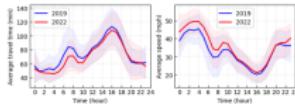


Figure 4. Average travel time and speed from area 32 (i.e., Downtown) to area 76 (i.e., Airport) in both 2019 and 2022.

```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(4, 2.5))
ax = fig.add_subplot(1, 2, 1)
# Average travel time in 2019
st = df.groupby(['Hour'])['Trip Seconds'].mean().values / 30
sr = df.groupby(['Hour'])['Trip Seconds'].std().values / 30
plt.plot(st, color='blue', linewidth=1.0, label='2019')
upper = st + sr
lower = st - sr
x_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
y_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
plt.fill(y_bound, x_bound, value = 0.1, alpha = 0.2)
plt.title('Average travel time in 2019')

st = df.groupby(['Hour'])['Trip Seconds'].mean().values / 30
sr = df.groupby(['Hour'])['Trip Seconds'].std().values / 30
plt.plot(sr, color='red', linewidth=1.0, label='2022')
upper = st + sr
lower = st - sr
x_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
y_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
plt.fill(y_bound, x_bound, value = 0.1, alpha = 0.2)
plt.title('Average travel time in 2022')

# Average speed in 2019
st = df.groupby(['Hour'])['Trip Miles'].mean().values / 30
sr = df.groupby(['Hour'])['Trip Miles'].std().values / 30
plt.plot(st, color='blue', linewidth=1.0, label='2019')
upper = st + sr
lower = st - sr
x_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
y_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
plt.fill(y_bound, x_bound, value = 0.1, alpha = 0.2)
plt.title('Average speed in 2019')

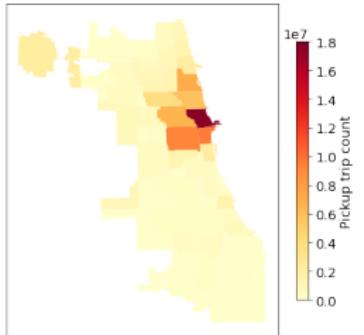
st = df.groupby(['Hour'])['Trip Miles'].mean().values / 30
sr = df.groupby(['Hour'])['Trip Miles'].std().values / 30
plt.plot(sr, color='red', linewidth=1.0, label='2022')
upper = st + sr
lower = st - sr
x_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
y_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)), np.array([-1, -1, -1])), np.arange(24, 25, -1))
plt.fill(y_bound, x_bound, value = 0.1, alpha = 0.2)
plt.title('Average speed in 2022')
```

Source: <https://spatiotemporal-data.github.io/Chicago-mobility/taxi-data>

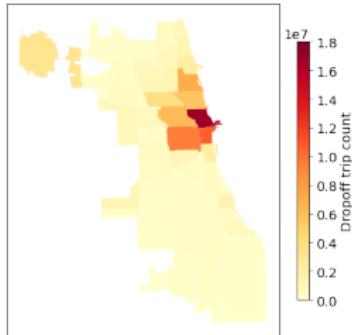
Human Mobility

- Ridesharing: 96,642,881 trips in 2019 vs. 57,290,954 trips in 2022

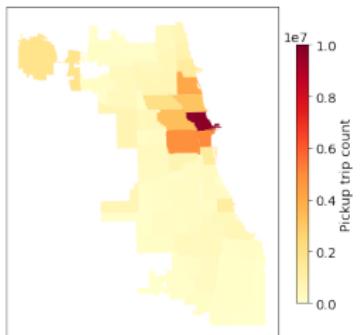
Pickup trips (2019)



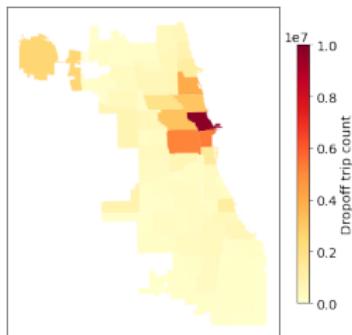
Dropoff trips (2019)



Pickup trips (2022)



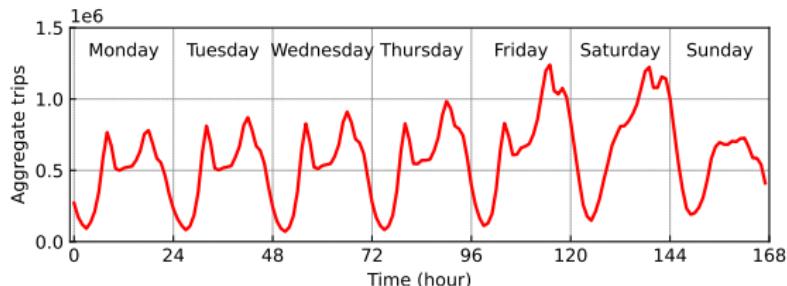
Dropoff trips (2022)



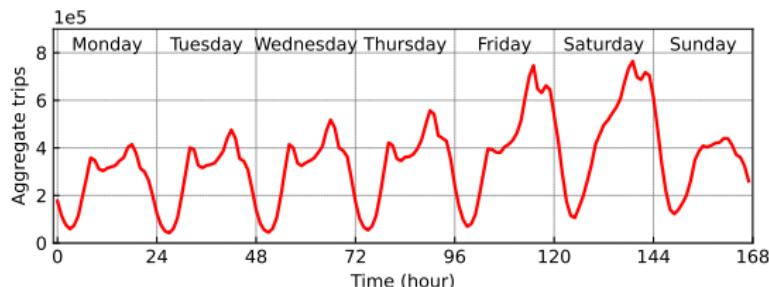
Human Mobility

- Ridesharing: 96,642,881 trips in 2019 vs. 57,290,954 trips in 2022

Pickup trips aggregated over 52 weeks in 2019

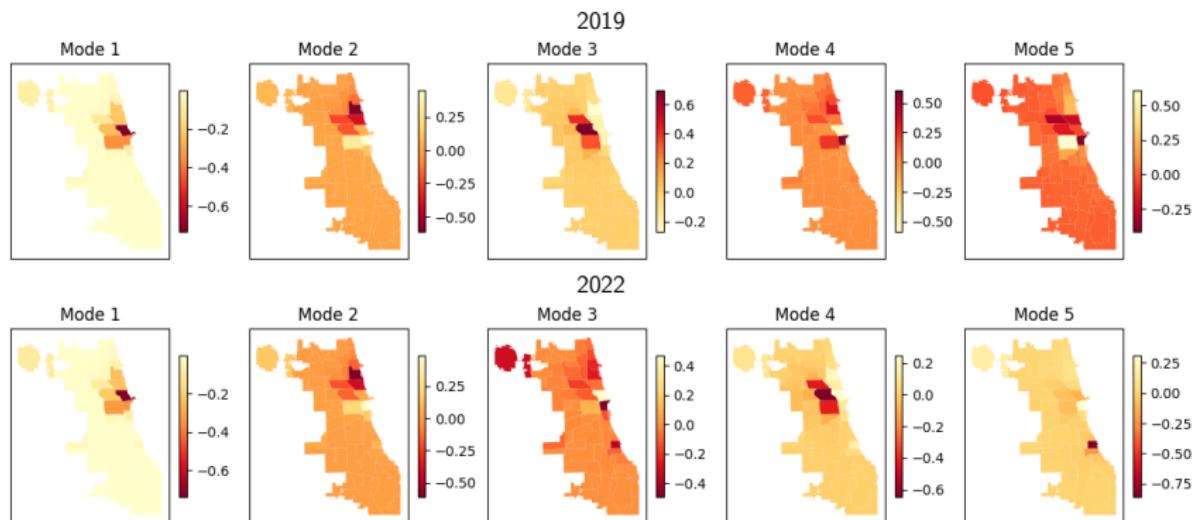


Pickup trips aggregated over 52 weeks in 2022



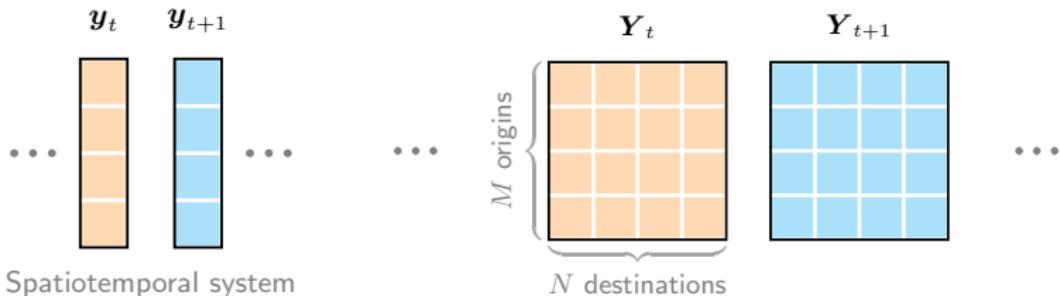
Human Mobility

- Ridesharing trip data: $77 \text{ origins} \times 77 \text{ destinations} \times 168 \text{ hours}$
- Our model Identifies the changes in pickup zones before and after COVID-19



Concluding Remark

- Discovering **spatial/temporal patterns** from 2D and 3D spatiotemporal systems with unsupervised learning:
 - Time-varying autoregression **on the data**
 - Tensor factorization **on the coefficients**



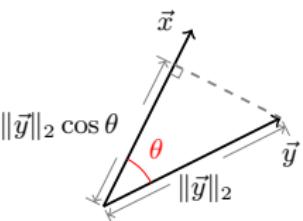
Next Stage?

1st Direction: Clifford Product

- Grassmann algebra⁹:

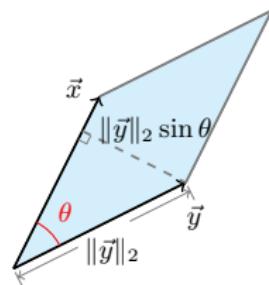
- Inner product

$$\langle \vec{x}, \vec{y} \rangle = \|\vec{x}\|_2 \underbrace{\|\vec{y}\|_2 \cos \theta}_{\text{projection}}$$



- Wedge product

$$\vec{x} \wedge \vec{y} = \underbrace{(\vec{e}_1 \wedge \vec{e}_2)}_{\text{orientation } (\pm 1)} \underbrace{\|\vec{x}\|_2 \|\vec{y}\|_2 \sin \theta}_{\text{area/determinant}}$$



- Clifford algebra (w/ Clifford product)

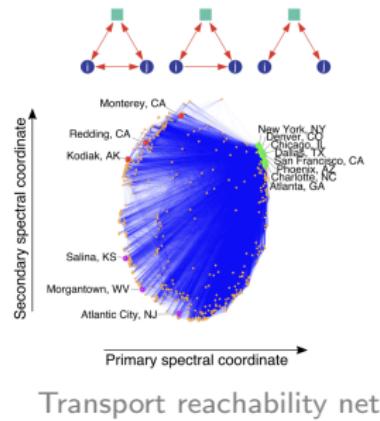
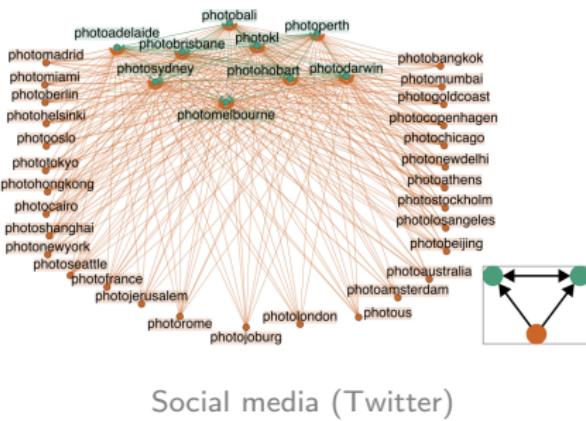
$$\vec{x} \cdot \vec{y} = \underbrace{\langle \vec{x}, \vec{y} \rangle}_{\text{inner prod.}} + \underbrace{\vec{x} \wedge \vec{y}}_{\text{Wedge prod.}}$$

- On PDE: Vector fields such as wind field and fluid dynamics

⁹On vectors $\vec{x} = x_1 \vec{e}_1 + x_2 \vec{e}_2$ and $\vec{y} = y_1 \vec{e}_1 + y_2 \vec{e}_2$.

2nd Direction: Learning from Graphs

- Graph **topology**: Graph & higher-order graph (e.g., Twitter)^{10, 11}
- Graph **community**: Hypergraph & higher-order motif (e.g., transport net)



- Connecting anything? Graph theory, signal processing, network science, data science, machine learning, ...
- Application: System evolution analysis on dynamic graphs

¹⁰ Picture source: <https://snap.stanford.edu/higher-order/>

¹¹ Notes: <https://spatiotemporal-data.github.io/bib/>



MENS
MANUS AND
MACHINA

Thanks for your attention!

Any Questions?

Slides: https://xinychen.github.io/slides/temporal_modeling.pdf

About me:

- 🏠 Homepage: <https://xinychen.github.io>
- ✉️ How to reach me: chenxy346@gmail.com
- ✉️ Or send to: xinychen@mit.edu