



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE

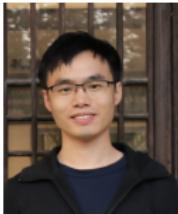


# Low-Rank Matrix and Tensor Methods for Spatiotemporal Traffic Data Modeling

**Xinyu Chen**

University of Montreal, Canada

May 11, 2023



**Ph.D. Candidate**  
Xinyu Chen



**Supervisor**  
Prof. Nicolas Saunier



**Co-supervisor**  
Prof. Lijun Sun

# Outline

---

- **Spatiotemporal Traffic Data**

- Preliminaries: What are matrix & tensor?

- Traffic data & representation & problems

- **Spatiotemporal Traffic Data Imputation**

- Laplacian convolutional representation

- Hankel tensor factorization

- **Sparse Traffic Flow Forecasting**

- Uber movement speed data

- Temporal matrix factorization

- **Dynamic Pattern Discovery**

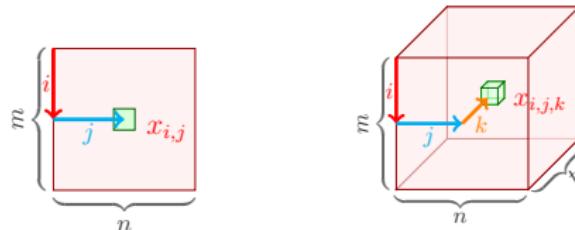
- Human mobility changes over COVID-19

- Time-varying low-rank autoregression

- **Conclusion**

# What are Matrix & Tensor?

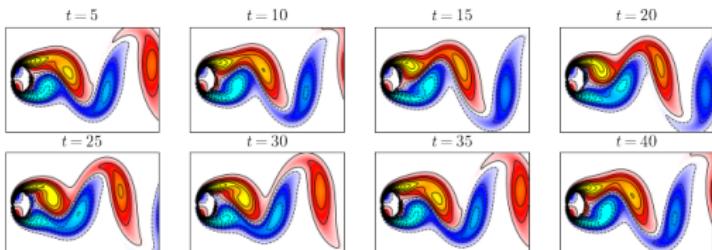
- What is tensor?  $\mathbf{X} \in \mathbb{R}^{m \times n}$  vs.  $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$



- Tensors are everywhere!



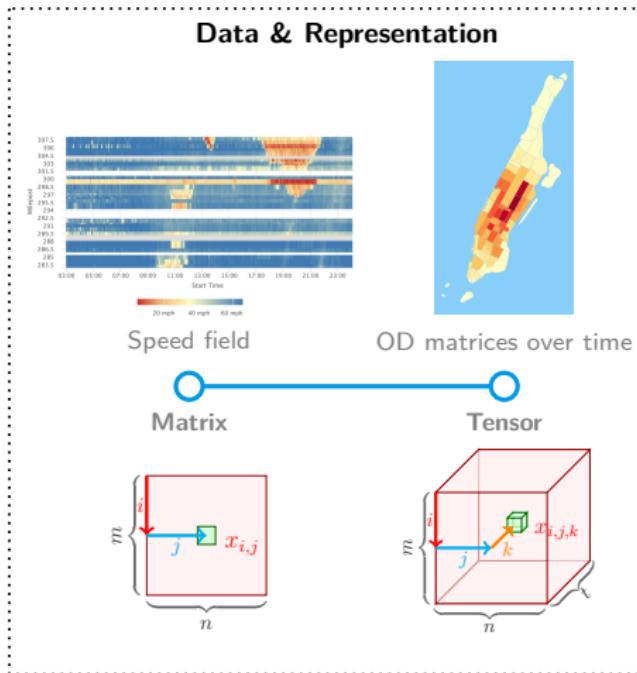
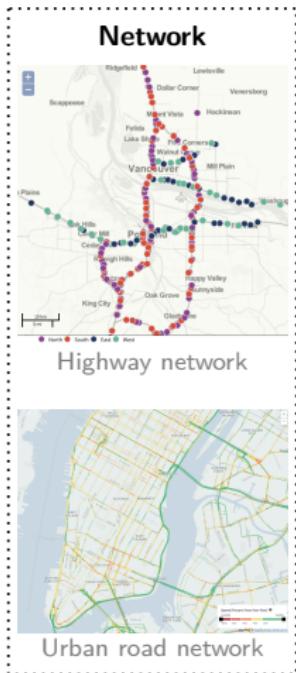
Color image with  
RGB channels



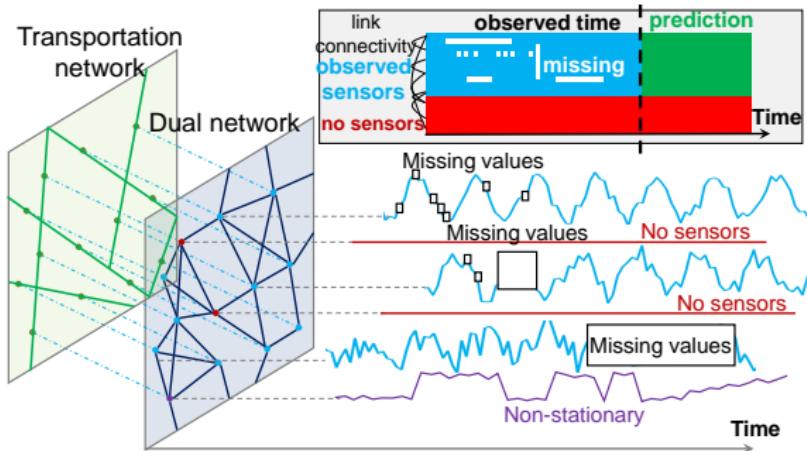
Dynamical system (fluid flow)

# Spatiotemporal Traffic Data

- Spatiotemporal traffic data are indeed matrices or tensors.



# Spatiotemporal Traffic Data



## Spatiotemporal Traffic Data Imputation

---

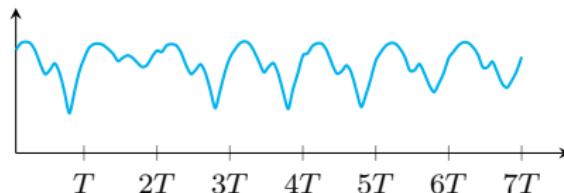
- ① X. Chen, Z. Cheng, N. Saunier, L. Sun (2022). Laplacian convolutional representation for traffic time series imputation. arXiv preprint arXiv:2212.01529.  
(Under 1st review at IEEE Transactions on Signal Processing)
- ② X. Chen Z. Cheng, L. Sun, N. Saunier (2023). Memory-efficient Hankel tensor factorization for extreme missing traffic data imputation. (coming soon)

# Laplacian Convolutional Representation

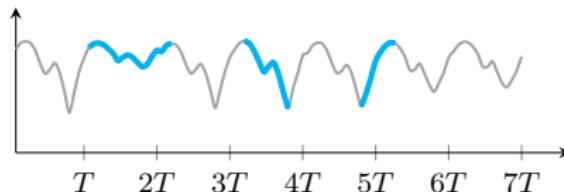
---

## Motivation:

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



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



- **Question:** How to characterize both global and local trends in sparse traffic data?

# Laplacian Convolutional Representation

Reformulate Laplacian regularization with circular convolution.

- 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

- Laplacian kernel:  $\ell = (2, -1, 0, 0, -1)^\top$ .
- Define Laplacian kernel:

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

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

- Local trend modeling via (Laplacian) temporal regularization:

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

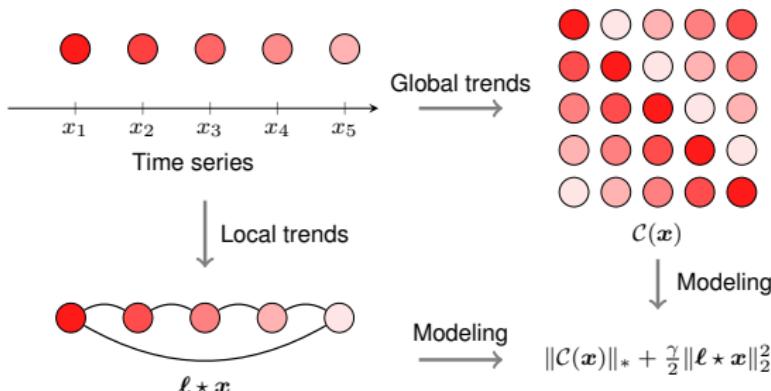
# Laplacian Convolutional Representation

## Laplacian Convolutional Representation (LCR)

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

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

where  $\mathcal{C} : \mathbb{R}^T \rightarrow \mathbb{R}^{T \times T}$  denotes the circulant operator.  $\|\cdot\|_*$  denotes the nuclear norm of matrix, namely, the sum of singular values.



## Laplacian Convolutional Representation

---

- LCR model:

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

- Augmented Lagrangian function:

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

where  $\mathbf{w} \in \mathbb{R}^T$  is the Lagrange multiplier, and  $\langle \cdot, \cdot \rangle$  denotes the inner product.

- The ADMM scheme:

$$\left\{ \begin{array}{l} \mathbf{x} := \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) \\ \mathbf{z} := \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) \\ = \frac{1}{\lambda + \eta} \mathcal{P}_\Omega(\lambda \mathbf{x} + \mathbf{w} + \eta \mathbf{y}) + \frac{1}{\lambda} \mathcal{P}_\Omega^\perp(\lambda \mathbf{x} + \mathbf{w}) \\ \mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z}) \end{array} \right.$$

## Laplacian Convolutional Representation

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

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

where  $\{\hat{\ell}, \hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{w}}\}$  refers to  $\{\ell, \mathbf{x}, \mathbf{z}, \mathbf{w}\}$  in the frequency domain.

### $\ell_1$ -norm Minimization in Complex Space (Liu & Zhang'22)

For any  $\ell_1$ -norm minimization problem in complex space:

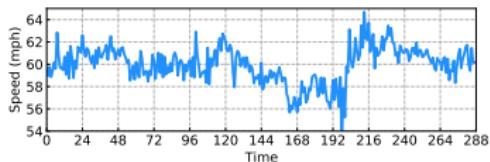
$$\min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1 + \frac{\lambda}{2T} \|\hat{\mathbf{x}} - \hat{\mathbf{h}}\|_2^2$$

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

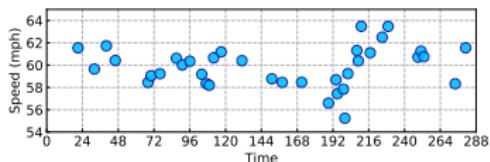
$$\hat{x}_t := \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - T/\lambda\}, t = 1, \dots, T.$$

# Laplacian Convolutional Representation

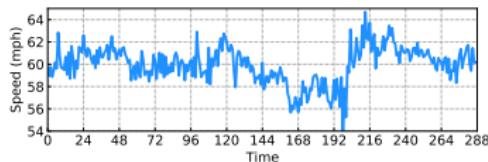
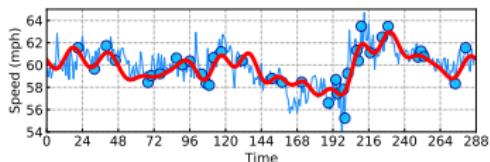
- On traffic speed time series:



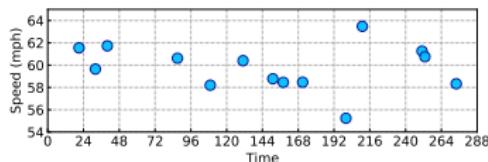
↓ Mask 90% observations



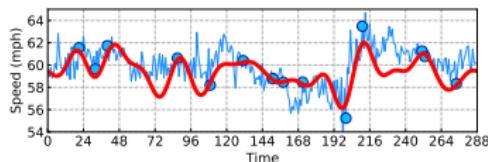
↓ Reconstruct time series



↓ Mask 95% observations

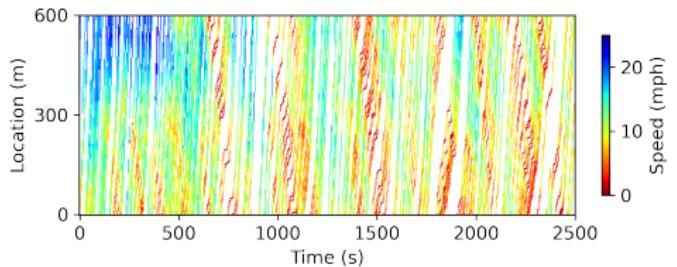


↓ Reconstruct time series

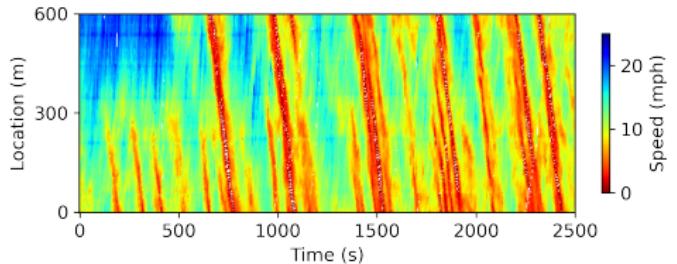


- LCR can characterize both global and local trends and produce accurate results.

# Hankel Tensor Factorization



200-by-500 matrix  
(NGSIM)  $\Downarrow$  Reconstruct speed field from  
20% sparse trajectories?



## Motivation:

- How to learn from sparse spatiotemporal data?
- How to characterize spatial/temporal local dependencies?

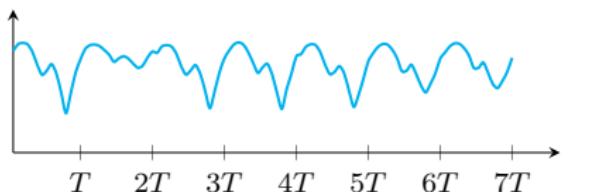
# Hankel Tensor Factorization

- Hankel matrix

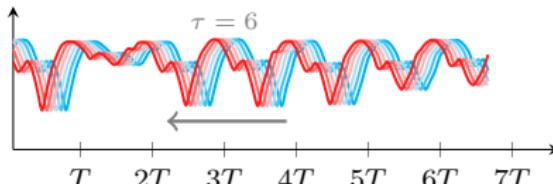
- Given  $\mathbf{y} = (1, 2, 3, 4, 5)^\top$  and window length  $\tau = 2$ , we have

$$\mathcal{H}_\tau(\mathbf{y}) = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix} \in \mathbb{R}^{4 \times 2}$$

- Hankel matrix  $\mathcal{H}_\tau(\mathbf{y})$  on time series  $\mathbf{y}$ :



↓ Construct Hankel matrix



# Hankel Tensor Factorization

---

- Hankel matrix

- Given  $\mathbf{y} = (1, 2, 3, 4, 5)^\top$  and window length  $\tau = 2$ , we have

$$\mathcal{H}_\tau(\mathbf{y}) = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix} \in \mathbb{R}^{4 \times 2}$$

- On time series  $\mathbf{y} = (y_1, y_2, \dots, y_5)^\top$  with  $\tau = 2$ :

$$\mathcal{H}_\tau(\mathbf{y}) = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_3 \\ y_3 & y_4 \\ y_4 & y_5 \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

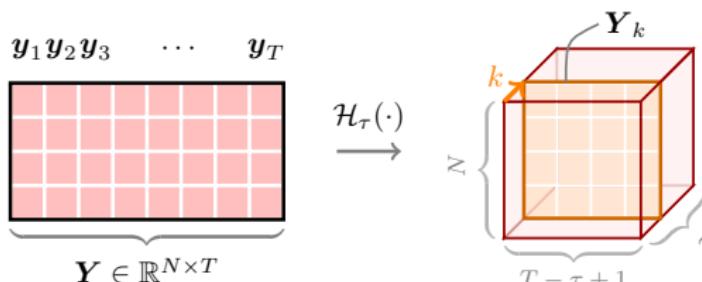
$$\implies \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \\ \hat{y}_5 \end{bmatrix} = \mathcal{H}_\tau^{-1} \left( \begin{bmatrix} v_1 x_1 & v_1 x_2 \\ v_2 x_1 & v_2 x_2 \\ v_3 x_1 & v_3 x_2 \\ v_4 x_1 & v_4 x_2 \end{bmatrix} \right) = \begin{bmatrix} v_1 x_1 \\ (v_1 x_2 + v_2 x_1)/2 \\ (v_2 x_2 + v_3 x_1)/2 \\ (v_3 x_2 + v_4 x_1)/2 \\ v_4 x_2 \end{bmatrix}$$

- Automatic **temporal** modeling

# Hankel Tensor Factorization

- Hankelization from  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  to Hankel tensor  $\mathcal{H}_\tau(\mathbf{Y})$ .

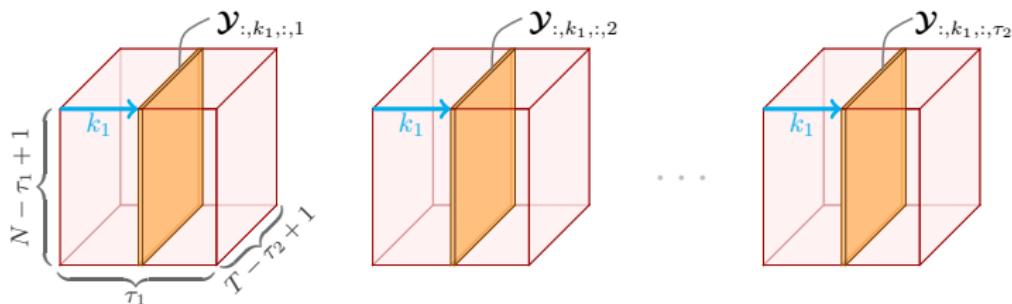
- Tensor size:  $N \times (T - \tau + 1) \times \tau$ ;
- Slices:  $\mathbf{Y}_k = \begin{bmatrix} | & | & | \\ \mathbf{y}_k & \mathbf{y}_{k+1} & \cdots & \mathbf{y}_{T-\tau+k} \\ | & | & | \end{bmatrix}, k = 1, 2, \dots, \tau$ ;
- Slice size:  $N \times (T - \tau + 1)$ .



- Automatic **temporal** modeling

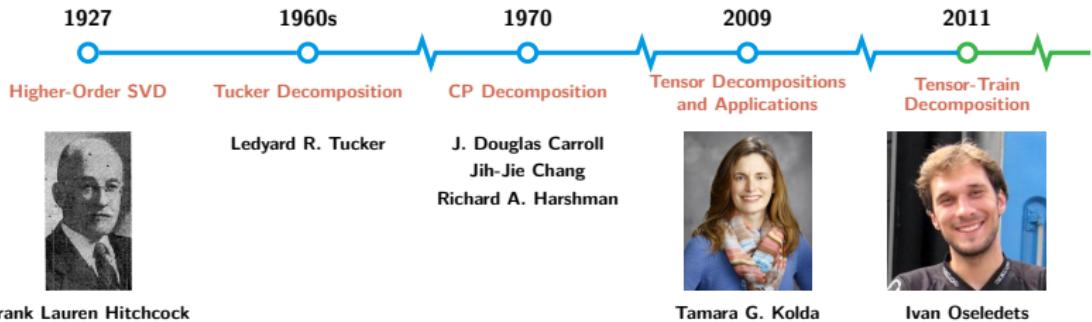
# Hankel Tensor Factorization

- Hankelization from  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  to  $\mathcal{Y} \triangleq \mathcal{H}_{\tau_1, \tau_2}(\mathbf{Y})$  (Hankel tensor).
  - Tensor size:  $(N - \tau_1 + 1) \times \tau_1 \times (T - \tau_2 + 1) \times \tau_2$ ;
  - Slice:  $\mathcal{Y}_{:, k_1, :, k_2}, \forall k_1, k_2$ ;
  - Slice size:  $(N - \tau_1 + 1) \times (T - \tau_2 + 1)$ .

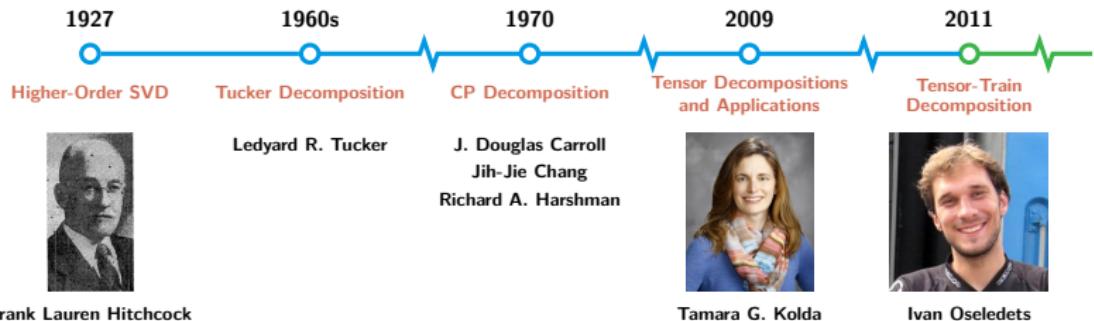


- Automatic **spatial** and **temporal** modeling

- Revisit tensor factorization (TF)



- Revisit tensor factorization (TF)



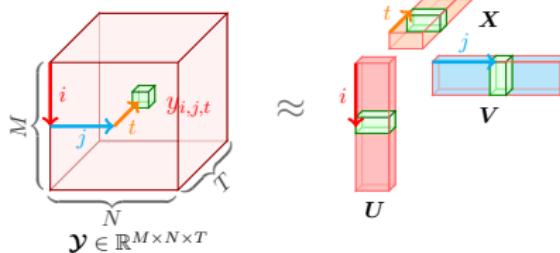
Frank Lauren Hitchcock

Tamara G. Kolda

Ivan Oseledets

- **CP decomposition:** Factorize  $\mathcal{Y}$  into the combination of rank- $R$  factor

$$\text{matrices, i.e., } \mathcal{Y} \approx \sum_{r=1}^R \mathbf{u}_r \otimes \mathbf{v}_r \otimes \mathbf{x}_r.$$



# Hankel Tensor Factorization

---

- Hankel tensor factorization:

$$\min_{\mathbf{Q}, \mathbf{S}, \mathbf{U}, \mathbf{V}} \frac{1}{2} \left\| \mathcal{P}_{\dot{\Omega}} \left( \mathcal{H}_{\tau_1, \tau_2}(\mathbf{Y}) - \sum_{r=1}^R \mathbf{q}_r \otimes \mathbf{s}_r \otimes \mathbf{u}_r \otimes \mathbf{v}_r \right) \right\|_F^2$$

- Advantage/disadvantage:

- ✓ Automatic spatial and temporal modeling
- ✗ High memory consumption

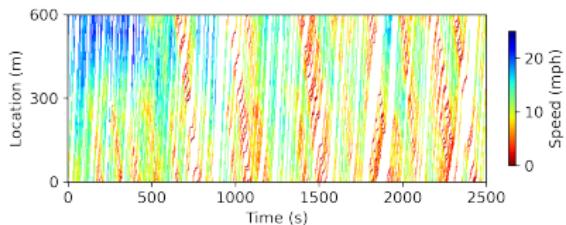
- Space complexity:

$$\mathcal{O}(\tau_1 \tau_2 (N - \tau_1 + 1)(T - \tau_2 + 1))$$

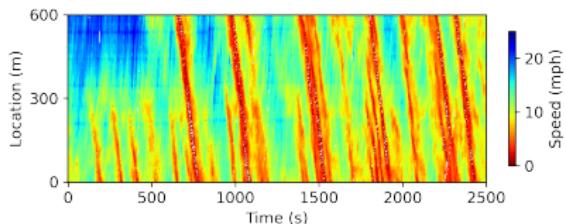
- **(Contribution)** Reduce the space complexity to  $\mathcal{O}(NT)$ .

# Hankel Tensor Factorization

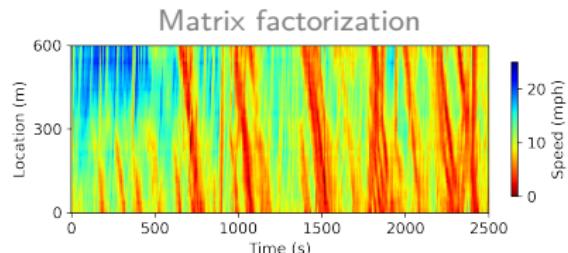
Which Model Is Better?



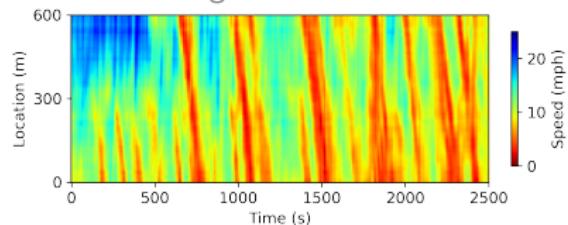
Sparse speed field



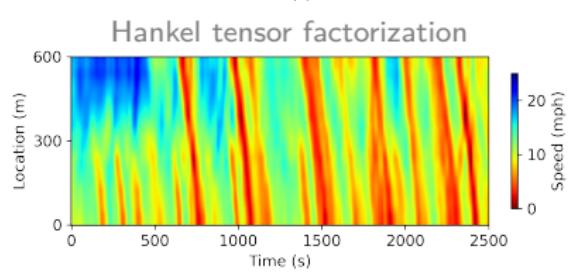
Ground truth speed field



Matrix factorization



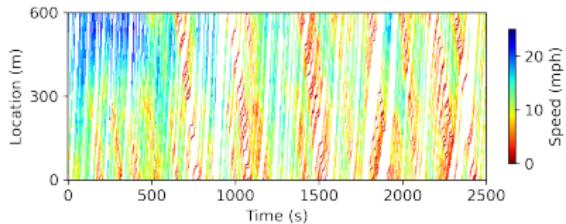
Smoothing matrix factorization



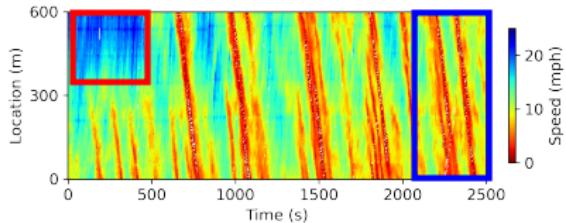
Hankel tensor factorization

# Hankel Tensor Factorization

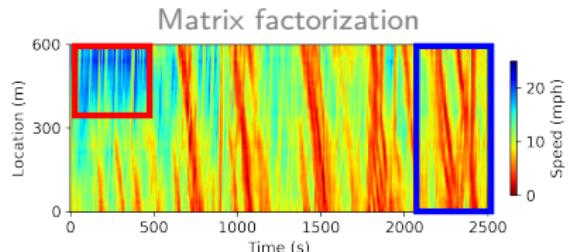
Which Model Is Better?



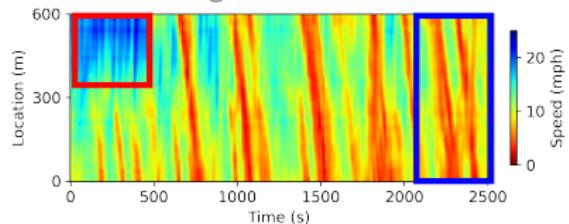
Sparse speed field



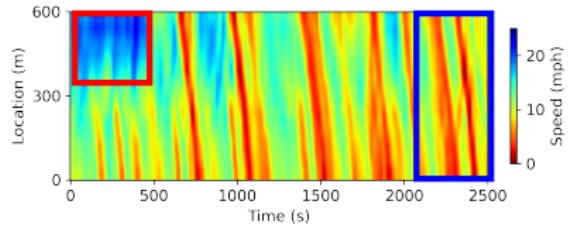
Ground truth speed field



Matrix factorization



Smoothing matrix factorization



Hankel tensor factorization

# Sparse Traffic Flow Forecasting

---

- ③ X. Chen, L. Sun (2022). Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (9): 4659–4673.
  - 100+ citations on Google Scholar
  - ESI highly cited paper (top 1%)
  - ESI hot paper (top 0.1%)
- ④ X. Chen, C. Zhang, X.-L. Zhao, N. Saunier, L. Sun (2022). Nonstationary temporal matrix factorization for sparse traffic time series forecasting. *arXiv preprint arXiv:2203.10651*.  
(Under 2nd review at *Transportation Research Part C: Emerging Technologies*)

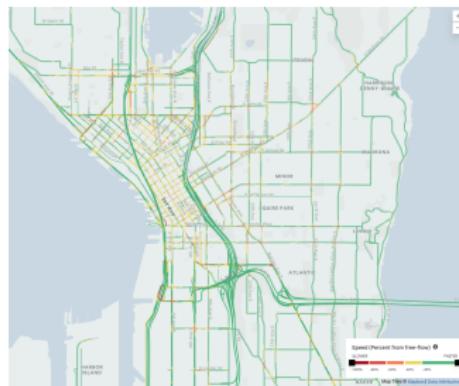
# Sparse Traffic Flow Forecasting

## Motivation:

- Uber (hourly) movement speed data<sup>1</sup>



NYC movement



Seattle movement

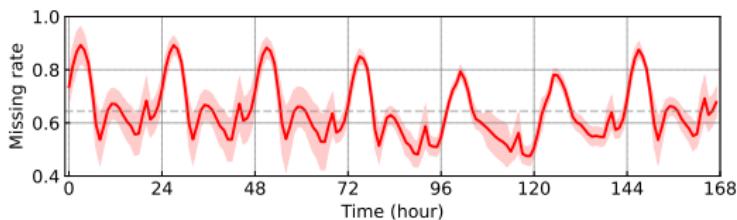
- The average speed on a given road segment for each hour of each day.
- Hourly speeds are computed when road segments have 5+ unique trips.
- **Issue:** insufficient sampling of ridesharing vehicles on the road network.

<sup>1</sup><https://movement.uber.com/>

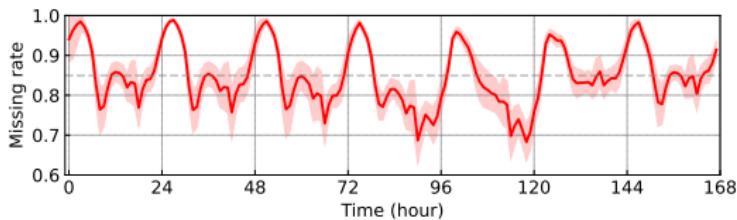
# Sparse Traffic Flow Forecasting

## High-dimensionality & Sparsity

- **NYC** movement speed data (2019)
  - **98,210** road segments & 8,760 time steps (hours)
  - Whole missing rate: **64.43%**

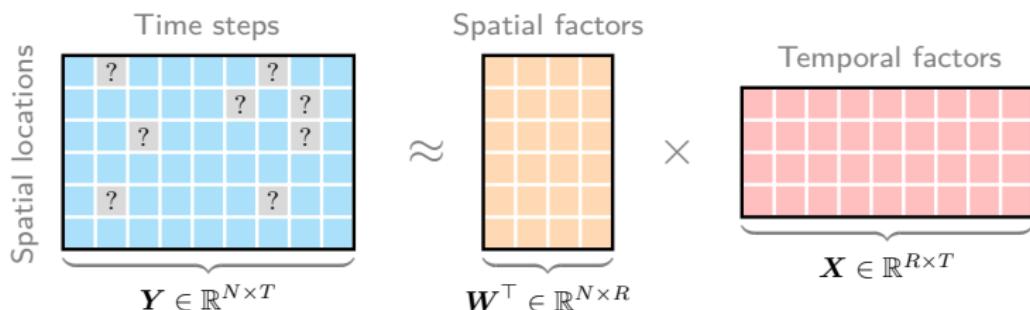


- **Seattle** movement speed data (2019)
  - **63,490** road segments & 8,760 time steps (hours)
  - Whole missing rate: **84.95%**



# Sparse Traffic Flow Forecasting

- Spatiotemporal data can be reconstructed by low-dimensional latent factors!



- MF optimization problem

$$\min_{\mathbf{W}, \mathbf{X}} \frac{1}{2} \left\| \mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) \right\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2)$$

with factor matrices  $\mathbf{W}$  and  $\mathbf{X}$ .

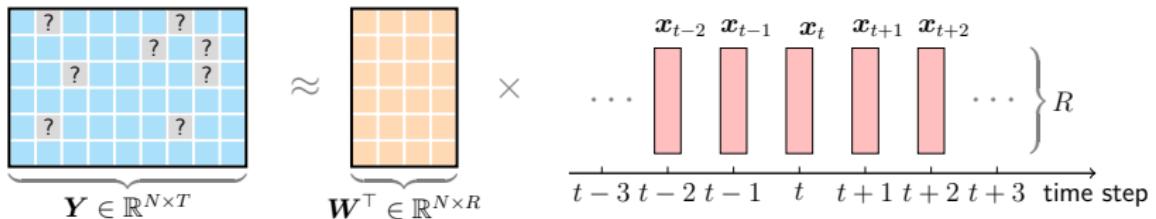
- Disadvantages:
  - Cannot capture temporal correlations.
  - Cannot perform time series forecasting.

# Sparse Traffic Flow Forecasting

## Temporal matrix factorization (Yu et al.'16; Chen & Sun'22)

Given any partially observed time series data  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  with observed index set  $\Omega$ , then temporal matrix factorization assumes a  $d$ th-order vector autoregressive (VAR) process on the temporal factor matrix:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k=1}^d} \quad & \frac{1}{2} \left\| \mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) \right\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) \\ & + \frac{\lambda}{2} \sum_{t=d+1}^T \left\| \mathbf{x}_t - \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} \right\|_2^2 \end{aligned}$$



## GitHub repositories:

- **transdim**: Machine learning for spatiotemporal traffic data imputation and forecasting. (970+ stars & 270+ forks)  
<https://github.com/xinychen/transdim>
- **tracebase**: Multivariate time series forecasting on high-dimensional and sparse Uber movement speed data. (30+ stars)  
<https://github.com/xinychen/tracebase>
- **awesome-latex-drawing**: Academic drawing examples in LaTeX. (1,000+ stars & 140+ forks)  
<https://github.com/xinychen/awesome-latex-drawing>

## Dynamic Pattern Discovery

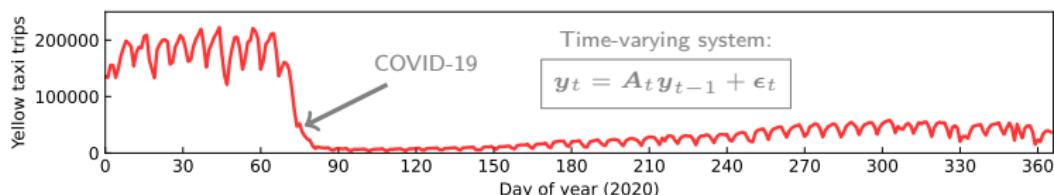
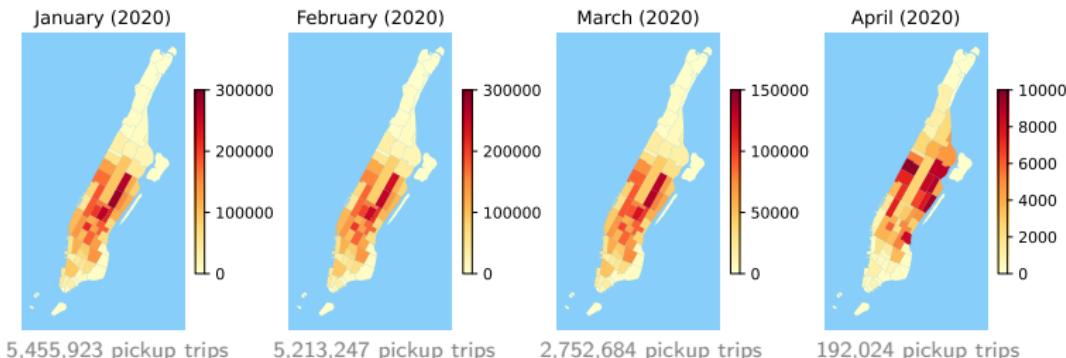
---

- ⑤ X. Chen, C. Zhang<sup>†</sup>, X. Chen, N. Saunier, L. Sun (2022). Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression. arXiv preprint arXiv:2211.15482.  
(Under 2nd review at IEEE Transactions on Knowledge and Data Engineering)

# Dynamic Pattern Discovery

## Motivation:

- NYC (yellow) taxi data<sup>2</sup>



- How to characterize the dynamic patterns?

<sup>2</sup><https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

# Dynamic Pattern Discovery

- Given a sequence of spatiotemporal measurements  
 $\mathbf{y}_t \in \mathbb{R}^N$ ,  $t = 1, 2, \dots, T$

$$\min_{\{\mathbf{A}_t\}} \underbrace{\frac{1}{2} \sum_t \|\mathbf{y}_t - \mathbf{A}_t \mathbf{y}_{t-1}\|_2^2}_{\text{Time-varying autoregression}}$$

[Over-parameterization]  $\mathcal{O}(N^2(T-1))$  parameters vs.  $\mathcal{O}(NT)$  data.

- (Ours)** Parameterize coefficients via tensor factorization (TF):

$$\min_{\mathbf{W}, \mathbf{G}, \mathbf{V}, \mathbf{x}} \underbrace{\frac{1}{2} \sum_t \left\| \mathbf{y}_t - \mathbf{W} \mathbf{G} (\mathbf{x}_t^\top \otimes \mathbf{V})^\top \mathbf{y}_{t-1} \right\|_2^2}_{\text{Let } \mathbf{A}_t = \mathbf{G} \times_1 \mathbf{W} \times_2 \mathbf{V} \times_3 \mathbf{x}_t^\top \text{ be the TF}}$$

- Alternating minimization (Let  $f$  be the obj.)

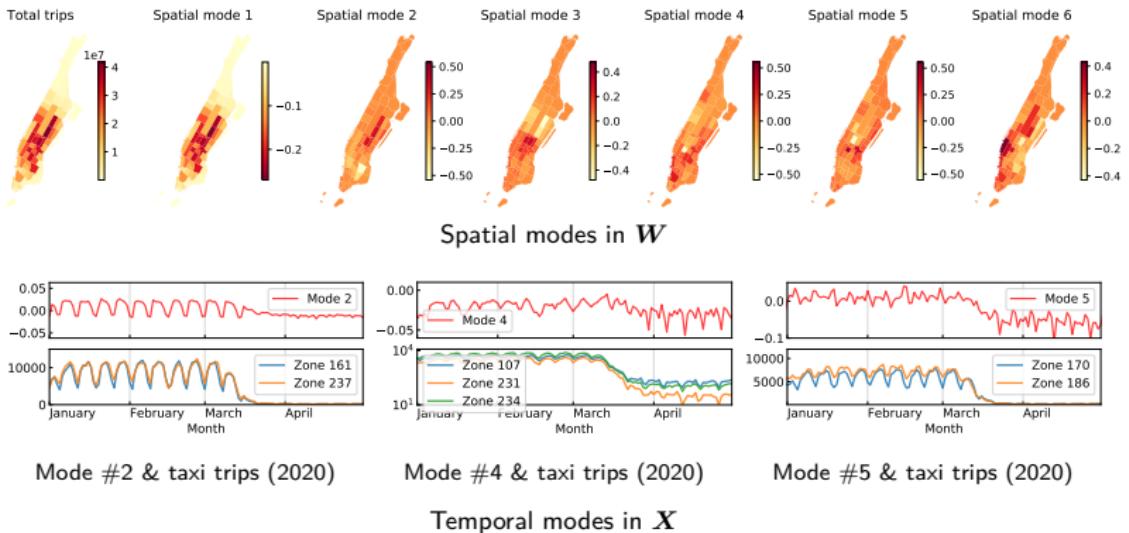
$$\begin{aligned} \mathbf{W} &:= \{\mathbf{W} \mid \frac{\partial f}{\partial \mathbf{W}} = \mathbf{0}\} & \mathbf{G} &:= \{\mathbf{G} \mid \frac{\partial f}{\partial \mathbf{G}} = \mathbf{0}\} \\ \mathbf{V} &:= \{\mathbf{V} \mid \frac{\partial f}{\partial \mathbf{V}} = \mathbf{0}\} & \mathbf{x}_t &:= \{\mathbf{x}_t \mid \frac{\partial f}{\partial \mathbf{x}_t} = \mathbf{0}\} \end{aligned}$$

- Solve each subproblem by **conjugate gradient** and **least squares**.

- Time-varying autoregression with TF

$$\min_{\mathbf{W}, \mathbf{G}, \mathbf{V}, \mathbf{x}} \frac{1}{2} \sum_t \left\| \mathbf{y}_t - \mathbf{W} \mathbf{G} (\mathbf{x}_t^\top \otimes \mathbf{V})^\top \mathbf{y}_{t-1} \right\|_2^2$$

- NYC taxi dataset (pickup)

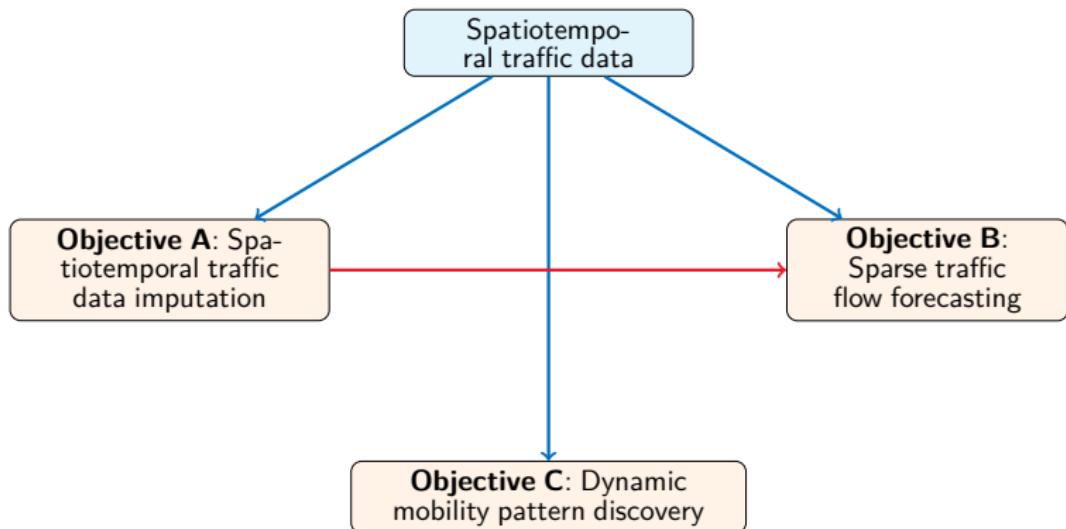


- Produce interpretable patterns and identify the changing point of system (mainly due to COVID-19).

# Conclusion

---

We give some studies on **spatiotemporal traffic data modeling**.



## Prior Works

---

Other studies about spatiotemporal data modeling:

- ⑥ X. Chen, M. Lei, N. Saunier, L. Sun (2022). Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems*, 23 (8): 12301–12310.
- ⑦ X. Chen, Y. Chen, N. Saunier, L. Sun (2021). Scalable low-rank tensor learning for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 129: 103226.
- ⑧ X. Chen, J. Yang, L. Sun (2020). A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 117: 102673.
- ⑨ X. Chen, Z. He, Y. Chen, Y. Lu, J. Wang (2019). Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model. *Transportation Research Part C: Emerging Technologies*, 104: 66-77.
- ⑩ X. Chen, Z. He, L. Sun (2019). A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 98: 73-84.

## References

---

### A short list:

- [Liu & Zhang'22] G. Liu and W. Zhang (2022). Recovery of future data via convolution nuclear norm minimization. *IEEE Transactions on Information Theory*, 69(1), 650–665.
- [Yu et al.'16] H.-F. Yu, N. Rao, and I. S. Dhillon (2016). Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems (NIPS)*.



POLYTECHNIQUE  
MONTRÉAL

UNIVERSITÉ  
D'INGÉNIERIE



# Thanks for your attention!

## Any Questions?

**Slides:** [https://xinychen.github.io/slides/traffic\\_data\\_modeling.pdf](https://xinychen.github.io/slides/traffic_data_modeling.pdf)

### About me:

- 🏠 Homepage: <https://xinychen.github.io>
- ✉️ Google Scholar: [user=mCrW04wAAAAJhl](#) (690 citations)
- ⌚ GitHub: <https://github.com/xinychen> (3.2k+ stars)
- 💻 Blog: <https://medium.com/@xinyu.chen> (70k+ views)
- ✉️ How to reach me: [chenxy346@gmail.com](mailto:chenxy346@gmail.com)