



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE



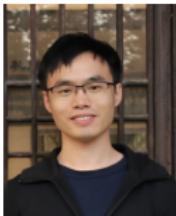
# Matrix and Tensor Models for Spatiotemporal Traffic Data Imputation and Forecasting

Ph.D. Defense

**Xinyu Chen**

Polytechnique Montreal, Canada

December 11, 2023



**Ph.D. Candidate**  
Xinyu Chen



**Supervisor**  
Prof. Nicolas Saunier



**Co-supervisor**  
Prof. Lijun Sun (McGill)

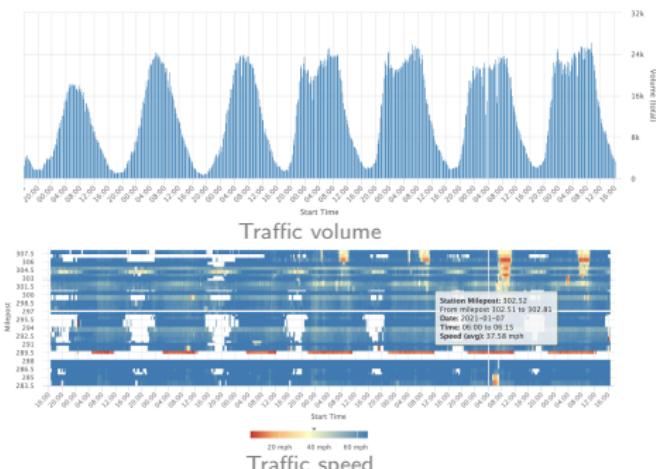
# Outline

- **Background**
- **Literature Review**
  - Tensor Factorization
  - Tensor Factorization (TF)
- **Nonstationary Temporal Matrix Factorization**
- **Low-Rank Autoregressive Tensor Completion**
- **Laplacian Convolutional Representation**
  - Motivation
  - Reformulate Laplacian Regularization
  - Traffic Time Series Imputation
- **Hankel Tensor Factorization**
  - Motivation
  - Hankel Structure
- **Experiments**
- **Conclusion**

# Multivariate Traffic Time Series

Many spatiotemporal traffic time series data are in the form of **matrix**.

- Example: Portland highway traffic data<sup>1</sup>.



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

<sup>1</sup><https://portal.its.pdx.edu/home>

## Multiple Data Behaviors

Spatiotemporal traffic data are time series, but they involve multiple data behaviors.

- Incompleteness & sparsity
- High-dimensionality
- Multidimensionality
- Noises & outliers
- Time-varying behavior
- Nonstationarity
- .....

In addition, spatiotemporal correlations are also very important.

# Multiple Data Behaviors

## Sparsity & high-dimensionality

- Uber (hourly) movement speed data<sup>2</sup>



NYC movement



Seattle movement

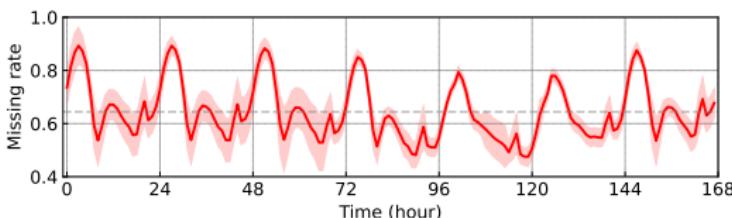
- The average speed on a given road segment for each hour of each day.
- Hourly speeds are computed when road segments have 5+ unique trips.
- **Issue:** insufficient sampling of ridesharing vehicles on the road network.

<sup>2</sup><https://movement.uber.com/>

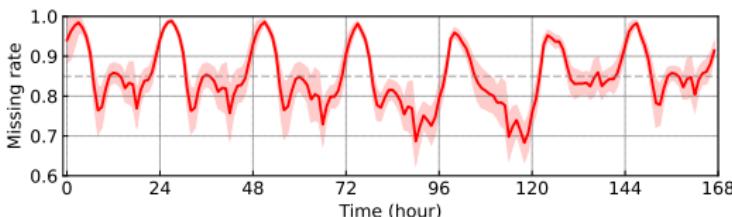
# Multiple Data Behaviors

## Sparsity & high-dimensionality

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

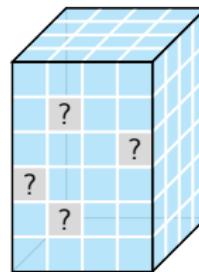
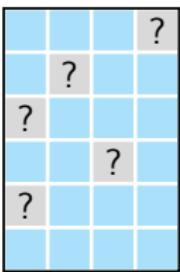


- **Seattle** movement speed data (2019)
  - 63,490 road segments & 8,760 time steps (hours)
  - Overall missing rate: 84.95%



## Problem Formulation

- **Objective A:** Given a multivariate time series data like  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  or a multidimensional time series data like  $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$ , impute the missing values of the data.

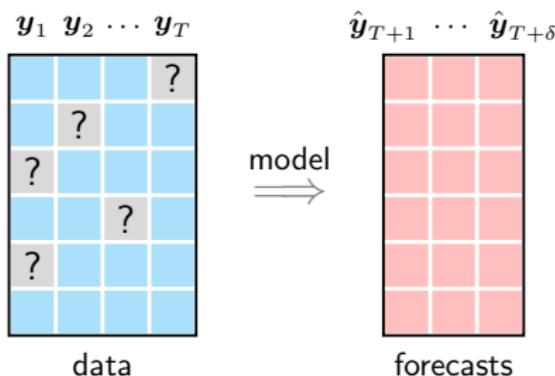


[Q]

- How to reconstruct missing values from observed data?
- How to make use of spatiotemporal correlations?
- How to make use of traffic time series dynamics?

## Problem Formulation

- **Objective B:** Given a partially observed data  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  consisting of time series  $\mathbf{y}_1, \dots, \mathbf{y}_T \in \mathbb{R}^N$ , forecast data points  $\hat{\mathbf{y}}_{T+\delta}, \delta \in \mathbb{N}^+$ .

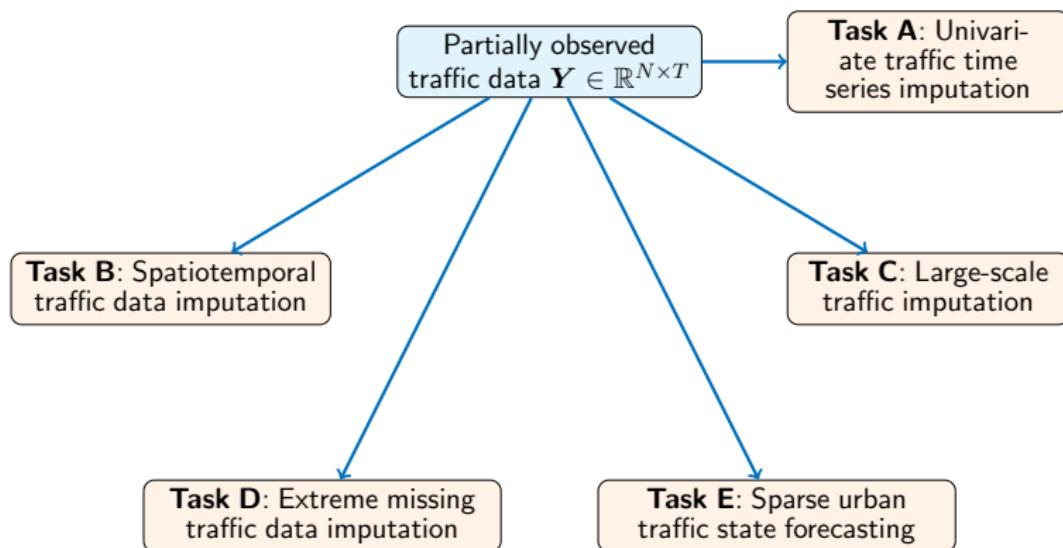


[Q]

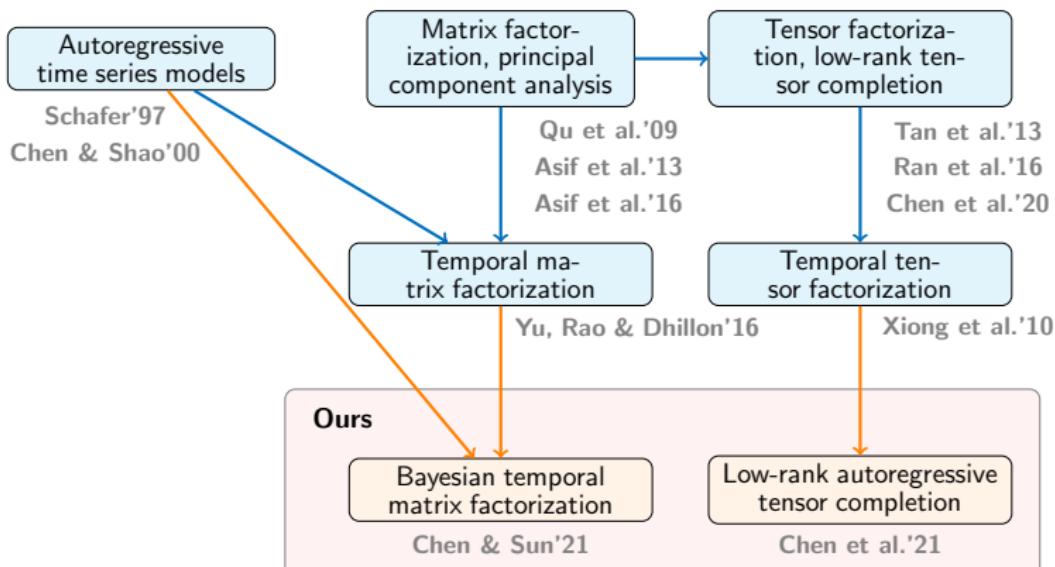
- How to learn from *high-dimensional* and *sparse* data?
- How to model *nonstationarity* in time series?
- How to perform forecasting on these time series?

# Whole Picture

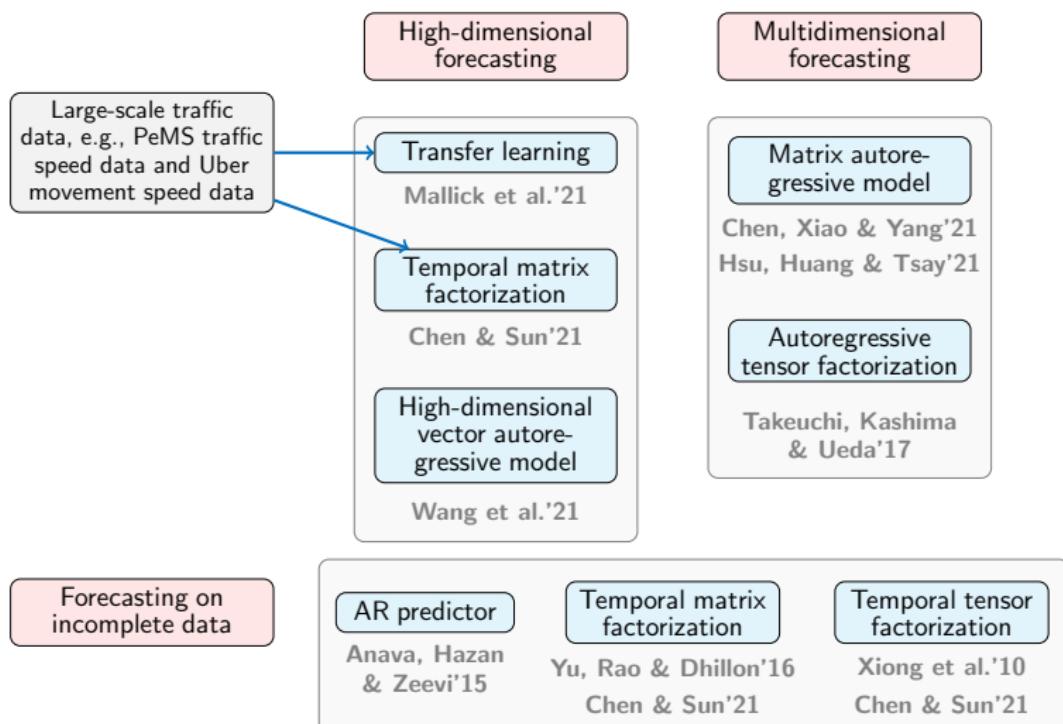
We are working on **spatiotemporal traffic data modeling**.



# Spatiotemporal Traffic Data Imputation



# Spatiotemporal Traffic Forecasting



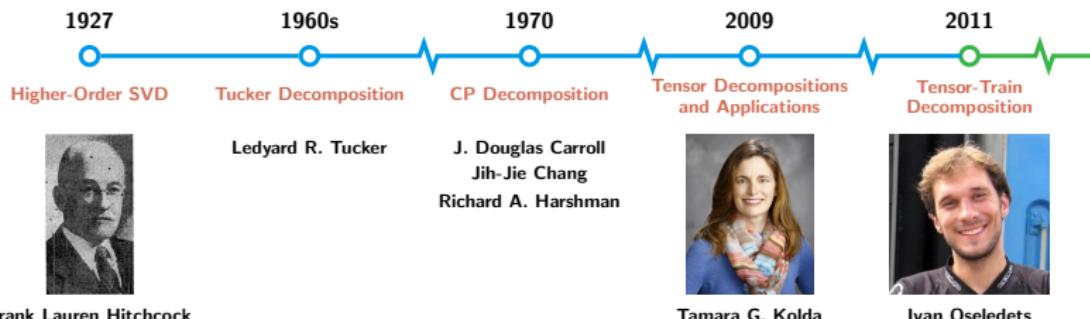
# Low-Rank Tensor Models

- Low-rank matrix/tensor completion

- 
- Candès & Recht'09:** Convex nuclear norm minimization for matrix completion.
$$\begin{aligned} & \min_{\mathbf{X}} \|\mathbf{X}\|_* \\ & \text{s.t. } \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{Y}) \end{aligned}$$
  - Cai, Candès & Shen'10:** Singular value thresholding algorithm.
$$\begin{cases} \mathbf{X}^\ell = \mathcal{D}_\tau(\mathbf{Z}^{\ell-1}) \\ \mathbf{Z}^\ell = \mathbf{Z}^{\ell-1} + \delta_\ell \mathcal{P}_\Omega(\mathbf{Y} - \mathbf{X}^\ell) \end{cases}$$
  - Zhang et al.'12:** Nonconvex truncated nuclear norm minimization.
  - Liu et al.'13:** Convex nuclear norm minimization for tensor completion.
$$\begin{aligned} & \min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{X}}\|_* \\ & \text{s.t. } \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{Y}}) \end{aligned}$$
  - Lu, Peng & Wei'19:** Tensor nuclear norm induced by linear transform.

# Tensor Factorization

- Revisit tensor factorization (TF)

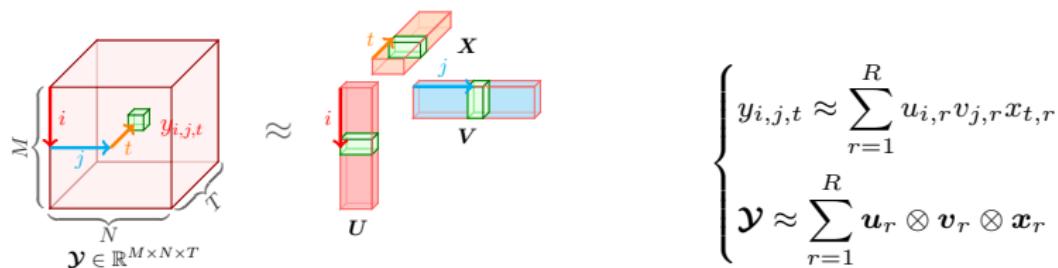


Frank Lauren Hitchcock

Tamara G. Kolda

Ivan Oseledets

- CP tensor factorization:** Factorize  $\mathcal{Y}$  into the combination of three rank- $R$  factor matrices (i.e., low-dimensional latent factors).

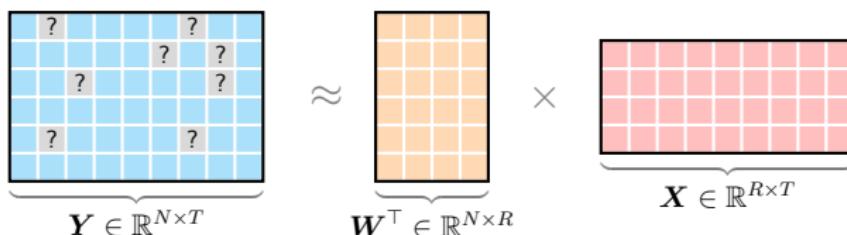


# Matrix Factorization

## Low-Rank Matrix Factorization (Koren et al.'09)

Given any partially observed data matrix  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  with observed index set  $\Omega$ , then the optimization problem of matrix factorization with rank  $R$  can be formulated as

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



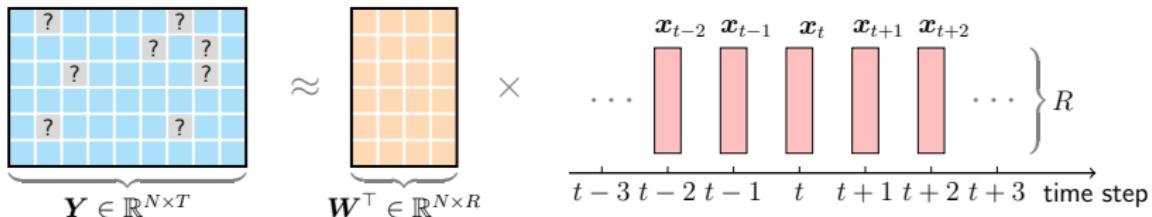
- ✗ Cannot capture temporal correlations.
- ✗ Cannot perform time series forecasting.

# Temporal Matrix Factorization

Temporal Matrix Factorization (Yu et al.'16; Chen & Sun'21)

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:

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



$\times$  VAR is usually built on stationary time series (temporal factors).

# Nonstationary Temporal Matrix Factorization

Rewrite VAR in the form of matrix

## Temporal operators

For any multivariate time series  $\mathbf{X} \in \mathbb{R}^{R \times T}$  with  $m, d \in \mathbb{N}^+$ , if we define temporal operators as

$$\begin{aligned}\Psi_k &\triangleq \begin{bmatrix} \mathbf{0}_{(T-d-m) \times (d-k)} & -\mathbf{I}_{T-d-m} & \mathbf{0}_{(T-d-m) \times (k+m)} \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0}_{(T-d-m) \times (d+m-k)} & \mathbf{I}_{T-d-m} & \mathbf{0}_{(T-d-m) \times k} \end{bmatrix} \\ &\in \mathbb{R}^{(T-d-m) \times T}, \quad k = 0, 1, \dots, d\end{aligned}$$

then

$$\begin{aligned}&\sum_{t=d+m+1}^T \|(\mathbf{x}_t - \mathbf{x}_{t-m}) - \sum_{k=1}^d \mathbf{A}_k (\mathbf{x}_{t-k} - \mathbf{x}_{t-m-k})\|_2^2 \\ &\equiv \|\mathbf{X} \Psi_0^\top - \sum_{k=1}^d \mathbf{A}_k \mathbf{X} \Psi_k^\top\|_F^2 \triangleq \|\mathbf{X} \Psi_0^\top - \mathbf{A} (\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top\|_F^2\end{aligned}$$

where  $\mathbf{A} \triangleq [\mathbf{A}_1 \quad \cdots \quad \mathbf{A}_d]$  and  $\Psi \triangleq [\Psi_1 \quad \cdots \quad \Psi_d]$ .

# Nonstationary Temporal Matrix Factorization

## Nonstationary temporal matrix factorization (NoTMF)

Given any partially observed time series data  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  with observed index set  $\Omega$ , then we assume a season- $m$  differencing on the latent temporal factors:

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

- First-order differencing  $\mathbf{x}'_t = \mathbf{x}_t - \mathbf{x}_{t-1}$ .
  - Second-order differencing  $\mathbf{x}''_t = (\mathbf{x}_t - \mathbf{x}_{t-1}) - (\mathbf{x}_{t-1} - \mathbf{x}_{t-2})$ .
  - Twice-differenced series  $\mathbf{x}'''_t = (\mathbf{x}_t - \mathbf{x}_{t-m}) - (\mathbf{x}_{t-1} - \mathbf{x}_{t-m-1})$ .
- ✓ Stationarizing a time series with differencing can improve the prediction.<sup>3</sup>

<sup>3</sup>Stationarity and differencing: <https://otexts.com/fpp2/stationarity.html>

# Nonstationary Temporal Matrix Factorization

Rewrite NoTMF:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{X}, \mathbf{A}} \quad & \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) \\ & + \frac{\lambda}{2} \|\mathbf{X} \Psi_0^\top - \mathbf{A} (\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top\|_F^2 \end{aligned}$$

Alternating minimization method:

- w.r.t.  $\mathbf{W}$ :

$$\frac{\partial f}{\partial \mathbf{W}} = -\mathbf{X} \mathcal{P}_\Omega^\top (\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{W} = \mathbf{0}$$

- w.r.t.  $\mathbf{X}$ :

$$\frac{\partial f}{\partial \mathbf{X}} = -\mathbf{W} \mathcal{P}_\Omega (\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{X} + \lambda \sum_{k=0}^d \mathbf{A}_k^\top \left( \sum_{h=0}^d \mathbf{A}_h \mathbf{X} \Psi_h^\top \right) \Psi_k = \mathbf{0}$$

- w.r.t.  $\mathbf{A}$ :

$$\mathbf{A} = \mathbf{X} \Psi_0^\top [(\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top]^\dagger$$

# Nonstationary Temporal Matrix Factorization

NoTMF forecasting on streaming data?

- NoTMF: Use  $\mathbf{Y}_t$  to estimate  $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$ .

$$\underbrace{\mathbf{Y}_t}_{\in \mathbb{R}^{N \times t}} = \begin{matrix} ? & & & ? \\ & ? & & ? \\ ? & & ? & \end{matrix}$$

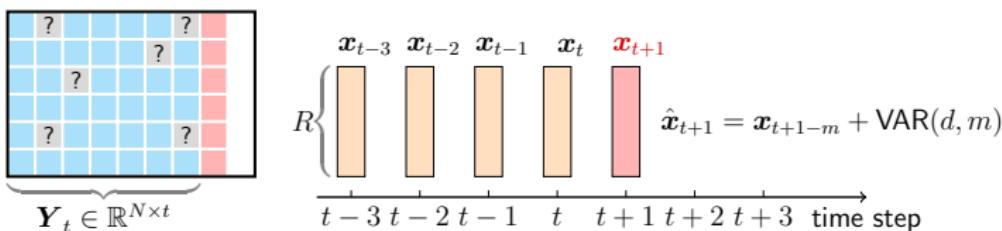
$$R \left\{ \begin{matrix} \mathbf{x}_{t-3} & \mathbf{x}_{t-2} & \mathbf{x}_{t-1} & \mathbf{x}_t & \mathbf{x}_{t+1} \\ t-3 & t-2 & t-1 & t & t+1 \end{matrix} \right. \quad \hat{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1-m} + \text{VAR}(d, m)$$

time step

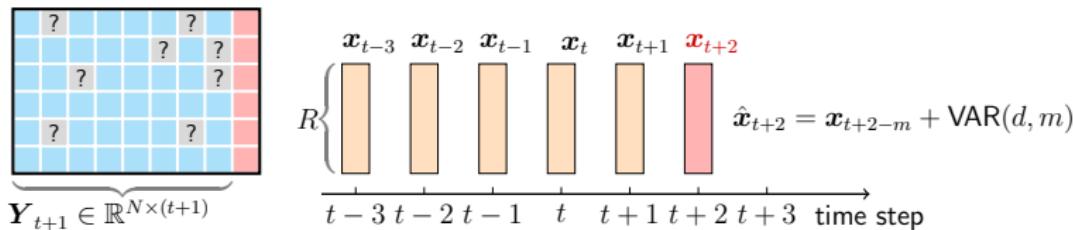
# Nonstationary Temporal Matrix Factorization

NoTMF forecasting on streaming data?

- NoTMF: Use  $\mathbf{Y}_t$  to estimate  $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$ .



- Online forecasting (Gultekin & Paisley'18): Fix  $\mathbf{W}$  and use  $\mathbf{Y}_{t+1}$  to update  $\{\mathbf{X}, \mathbf{A}\}$ .



# Low-Rank Autoregressive Tensor Completion

**Cornerstone:** Nuclear norm minimization in matrix/tensor completion

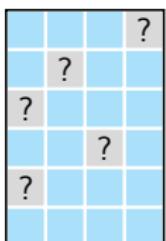
LRMC (Candès & Recht'09)

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{X}\|_* \\ \text{s.t. } \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{Y}) \end{aligned}$$

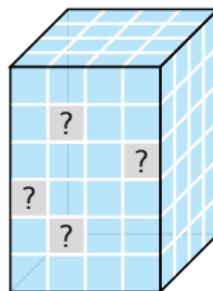
vs.

LRTC (Liu et al.'13)

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}}} \quad & \|\boldsymbol{\mathcal{X}}\|_* \\ \text{s.t. } \mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}) = \mathcal{P}_\Omega(\boldsymbol{\mathcal{Y}}) \end{aligned}$$



$$\mathcal{P}_\Omega(\mathbf{Y}) \in \mathbb{R}^{N \times T}$$



$$\mathcal{P}_\Omega(\boldsymbol{\mathcal{Y}}) \in \mathbb{R}^{N \times I \times J}$$

- **Limitation:** Only cover global consistency
- **Highlight:** Introduce local consistency (e.g., temporal correlations)

Background  
oooooooo

Literature Review  
oooo

NoTMF  
oooooo

LATC  
o●oooo

LCR  
oooooooo

HTF  
ooooooooo

Experiments  
ooooooo

Conclusion  
oooo

# Low-Rank Autoregressive Tensor Completion

# Low-Rank Autoregressive Tensor Completion

- Tensorization
- Autoregression

# Low-Rank Autoregressive Tensor Completion

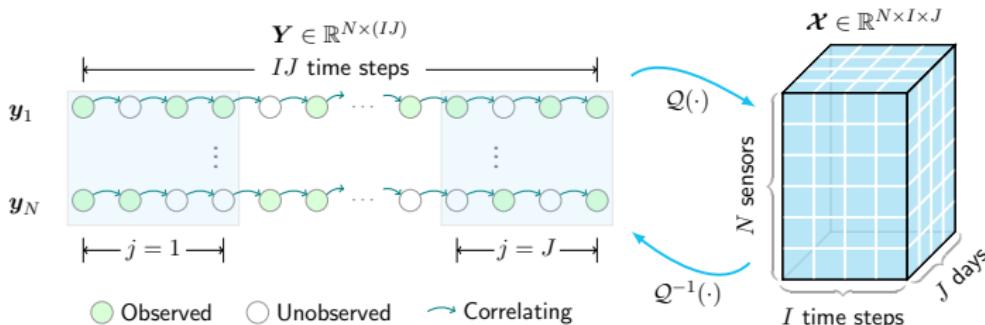
## Low-Rank Autoregressive Tensor Completion (LATC)

For any partially observed traffic data  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  with observed index set  $\Omega$ , LATC takes

$$\begin{aligned} & \min_{\mathbf{Z}, \mathbf{A}} \|\mathcal{Q}(\mathbf{Z})\|_{r,*} + \frac{\gamma}{2} \|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}} \\ & \text{s.t. } \mathcal{P}_{\Omega}(\mathbf{Z}) = \mathcal{P}_{\Omega}(\mathbf{Y}) \end{aligned}$$

Advantages:

- ✓ Global consistency (w/ tensor representation)
- ✓ Local consistency (w/ temporal autoregression)



**Background**  
oooooooo

**Literature Review**  
oooo

**NoTMF**  
oooooo

**LATC**  
oooo●○

**LCR**  
oooooooo

**HTF**  
ooooooooo

**Experiments**  
ooooooo

**Conclusion**  
oooo

# Low-Rank Autoregressive Tensor Completion

- Optimization problem:

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{A}} \quad & \|\mathcal{Q}(\mathbf{Z})\|_{r,*} + \frac{\gamma}{2} \|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}} \\ \text{s.t. } \mathcal{P}_{\Omega}(\mathbf{Z}) = \mathcal{P}_{\Omega}(\mathbf{Y}) \end{aligned}$$

- Two subproblems:

$$\begin{cases} \mathbf{Z} := \underset{\mathcal{P}_{\Omega}(\mathbf{Z})=\mathcal{P}_{\Omega}(\mathbf{Y})}{\arg \min} \|\mathcal{Q}(\mathbf{Z})\|_{r,*} + \frac{\gamma}{2} \|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}} \\ \mathbf{A} := \frac{1}{2} \|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}} \quad (\text{Least squares}) \end{cases}$$

- Optimize  $\mathbf{Z}$ ? Write down the augmented Lagrangian function:

$$\mathcal{L}(\mathbf{X}, \mathbf{Z}, \mathbf{W}) = \|\mathbf{X}\|_{r,*} + \frac{\gamma}{2} \|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}} + \frac{\lambda}{2} \|\mathbf{X} - \mathcal{Q}(\mathbf{Z})\|_F^2 + \langle \mathbf{W}, \mathbf{X} - \mathcal{Q}(\mathbf{Z}) \rangle + \pi(\mathbf{Z})$$

where  $\mathbf{W} \in \mathbb{R}^{N \times I \times J}$  is the Lagrange multiplier, and  $\langle \cdot, \cdot \rangle$  denotes the inner product.<sup>4</sup>

---

<sup>4</sup>The indicator function is  $\pi(\mathbf{Z}) = \begin{cases} 0, & \text{if } \mathcal{P}_{\Omega}(\mathbf{Z}) = \mathcal{P}_{\Omega}(\mathbf{Y}), \\ +\infty, & \text{otherwise.} \end{cases}$

# Low-Rank Autoregressive Tensor Completion

- Augmented Lagrangian function:

$$\mathcal{L}(\mathbf{X}, \mathbf{Z}, \mathbf{W}) = \|\mathbf{X}\|_{r,*} + \frac{\gamma}{2} \|\mathbf{Z}\|_{A,\mathcal{H}} + \frac{\lambda}{2} \|\mathbf{X} - \mathcal{Q}(\mathbf{Z})\|_F^2 + \langle \mathbf{W}, \mathbf{X} - \mathcal{Q}(\mathbf{Z}) \rangle + \pi(\mathbf{Z})$$

- The ADMM scheme:

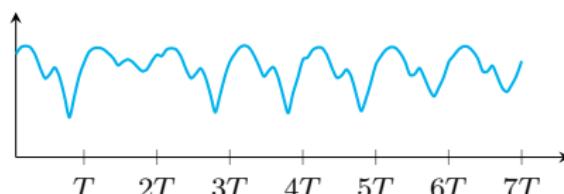
$$\begin{cases} \mathbf{X} := \arg \min_{\mathbf{X}} \mathcal{L}(\mathbf{X}, \mathbf{Z}, \mathbf{W}) & \text{(Truncated nuclear norm minimization)} \\ \mathbf{Z} := \arg \min_{\mathbf{Z}} \mathcal{L}(\mathbf{X}, \mathbf{Z}, \mathbf{W}) & \text{(Generalized Sylvester equation)} \\ \mathbf{W} := \mathbf{W} + \lambda(\mathbf{X} - \mathcal{Q}(\mathbf{Z})) & \text{(Standard update)} \end{cases}$$

- ✓ Solution to  $\mathbf{X}$ : singular value thresholding
- ✓ Solution to  $\mathbf{Z}$ : conjugate gradient

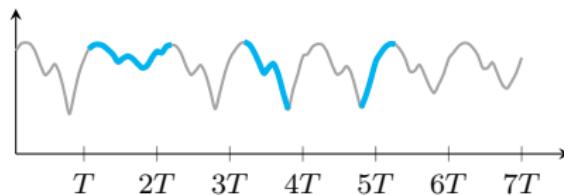
# Laplacian Convolutional Representation

## Motivation: Time series imputation

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



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

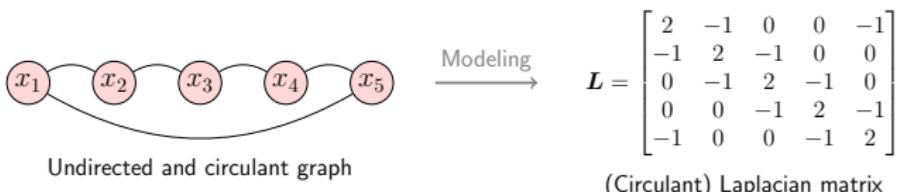


- [Question] How to characterize both global and local trends in sparse time series data?

# Laplacian Convolutional Representation

**[Local trend modeling]** Reformulate temporal regularization with circular convolution.

- Intuition of (circulant) Laplacian matrix.



- Define Laplacian kernel:

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

$\Downarrow$

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

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

- (Laplacian) Temporal regularization:

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

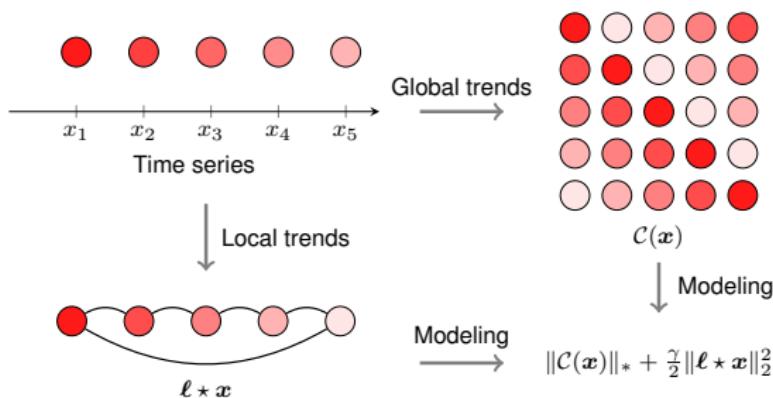
# Laplacian Convolutional Representation

## 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.,

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

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

- Augmented Lagrangian function:

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

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

- The ADMM scheme:

$$\begin{cases} \mathbf{x} := \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) & \text{(Nuclear norm minimization)} \\ \mathbf{z} := \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) & \text{(Closed-form solution)} \\ = \frac{1}{\lambda + \eta} \mathcal{P}_\Omega(\lambda \mathbf{x} + \mathbf{w} + \eta \mathbf{y}) + \frac{1}{\lambda} \mathcal{P}_\Omega^\perp(\lambda \mathbf{x} + \mathbf{w}) \\ \mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z}) & \text{(Standard update)} \end{cases}$$

- Optimize  $\mathbf{x}$ ?

$$\|\mathcal{C}(\mathbf{x})\|_* = \|\mathcal{F}(\mathbf{x})\|_1 \quad \& \quad \frac{1}{2} \|\ell * \mathbf{x}\|_2^2 = \frac{1}{2T} \|\mathcal{F}(\ell) \circ \mathcal{F}(\mathbf{x})\|_2^2$$

Nuclear norm minimization  $\Rightarrow \ell_1$ -norm minimization with FFT (in  $\mathcal{O}(T \log T)$  time).

## 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).

### $\ell_1$ -norm Minimization in Complex Space (Liu & Zhang'23)

For any optimization problem in the form of  $\ell_1$ -norm minimization in complex space:

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

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

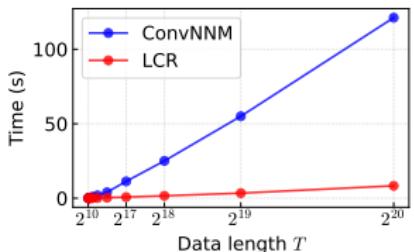
$$\hat{x}_t := \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - 1/\omega\}, t = 1, \dots, T.$$

# Laplacian Convolutional Representation

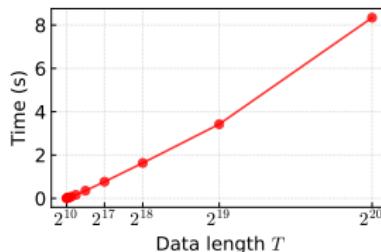
## Empirical time complexity

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**<sup>5</sup> (Liu & Zhang'23)
  - Convolution matrix  $C_{\tilde{\tau}}(\mathbf{y}) \in \mathbb{R}^{T \times \tilde{\tau}}$  with kernel size  $\tilde{\tau} \in \mathbb{N}^+$
  - Singular value thresholding in  $\mathcal{O}(\tilde{\tau}^2 T)$



ConvNNM vs. LCR



LCR

<sup>5</sup>Convolution nuclear norm minimization.

## Two-Dimensional LCR (LCR-2D)

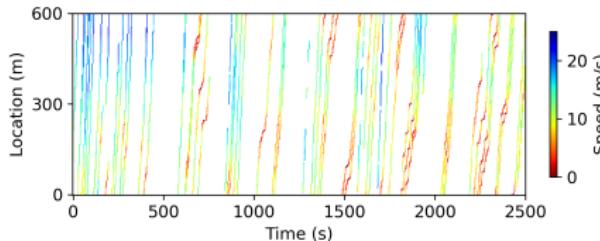
For any partially observed time series  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  with observed index set  $\Omega$ , LCR can be formulated as follows,

$$\begin{aligned} & \min_{\mathbf{X}} \|\mathcal{C}(\mathbf{X})\|_* + \frac{\gamma}{2} \|(\boldsymbol{\ell}_s \boldsymbol{\ell}^\top) \star \mathbf{X}\|_F^2 \\ & \text{s.t. } \|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{Y})\|_F \leq \epsilon \end{aligned}$$

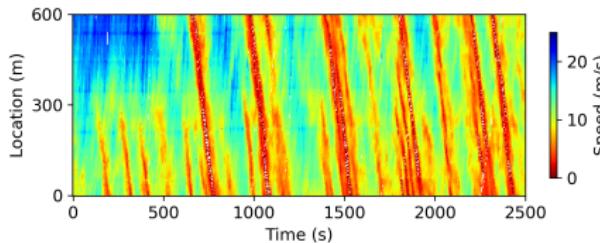
where  $\mathcal{C} : \mathbb{R}^{N \times T} \rightarrow \mathbb{R}^{N \times N \times T \times T}$  denotes the circulant operator.

## Motivation: Spatiotemporal data reconstruction

- Speed field reconstruction problem in vehicular traffic flow.



200-by-500 matrix  
(NGSIM)  $\Downarrow$  Reconstruct speed field from  
5% sparse trajectories?



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

# Hankel Tensor Factorization

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

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



Hankel matrix (Source: Twitter)

# Hankel Tensor Factorization

- Hankel matrix

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

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

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

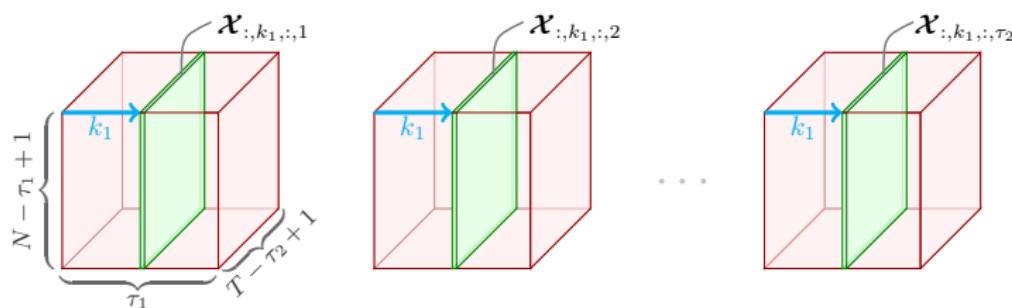
- Automatic temporal modeling.

# Hankel Tensor Factorization

- Hankel tensor: Given any matrix  $\mathbf{X} \in \mathbb{R}^{N \times T}$ , we have

$$\mathcal{X} \triangleq \mathcal{H}_{\tau_1, \tau_2}(\mathbf{X})$$

- Window lengths:  $\tau_1, \tau_2 \in \mathbb{N}^+$ ;
- Tensor size:  $(N - \tau_1 + 1) \times \tau_1 \times (T - \tau_2 + 1) \times \tau_2$ ;



(Figure) 4th order Hankel tensor: A sequence of third-order tensors.

- Slice:  $\mathcal{X}_{:,k_1,:,k_2}, \forall k_1, k_2$ ;
- Slice size:  $(N - \tau_1 + 1) \times (T - \tau_2 + 1)$ .

# Hankel Tensor Factorization

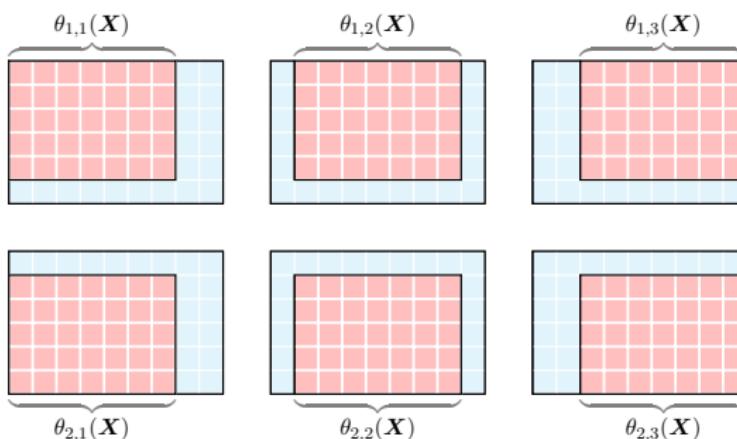
## Hankel indexing:

- Sampling function for the Hankelization:

$$\theta_{k_1, k_2}(\mathbf{X}) \triangleq [\mathcal{H}_{\tau_1, \tau_2}(\mathbf{X})]_{:, k_1, :, k_2},$$

referring to the tensor slice with  $k_1 \in \{1, \dots, \tau_1\}$ ,  $k_2 \in \{1, \dots, \tau_2\}$ .

- [Importance] Developing memory-efficient algorithms.



- Tensor slices  $\theta_{k_1, k_2}(\mathbf{X})$  vs. data matrix  $\mathbf{X}$

# Hankel Tensor Factorization

## Ours:

- Convolutional tensor decomposition (circular convolution  $\star_{\text{row}}$ ):

$$\theta_{k_1, k_2}(\mathbf{Y}) \approx (\mathbf{Q} \star_{\text{row}} \mathbf{s}_{k_1}^{\top})(\mathbf{U} \star_{\text{row}} \mathbf{v}_{k_2}^{\top})^{\top}$$

## Baselines:

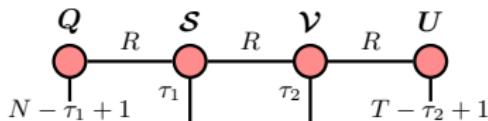
- CP tensor decomposition (Khatri-Rao product  $\odot$ ):

$$\theta_{k_1, k_2}(\mathbf{Y}) \approx (\mathbf{Q} \odot \mathbf{s}_{k_1}^{\top})(\mathbf{U} \odot \mathbf{v}_{k_2}^{\top})^{\top}$$

- Tensor-train decomposition:

$$\theta_{k_1, k_2}(\mathbf{Y}) \approx (\mathbf{Q} \mathbf{S}_{k_1})(\mathbf{U} \mathbf{V}_{k_2})^{\top}$$

- $\{\mathbf{S}_{k_1}, \mathbf{V}_{k_2}\}$  are **circulant matrices**  $\Rightarrow$  convolutional decomposition
- $\{\mathbf{S}_{k_1}, \mathbf{V}_{k_2}\}$  are **diagonal matrices**  $\Rightarrow$  CP decomposition



# Hankel Tensor Factorization

## HTF (convolutional decomposition)

- Optimization problem:

$$\begin{aligned} \min_{\mathbf{Q}, \mathbf{S}, \mathbf{U}, \mathbf{V}} \quad & \frac{1}{2} \sum_{k_1, k_2} \left\| \mathcal{P}_{\Omega_{k_1, k_2}} (\theta_{k_1, k_2}(\mathbf{Y}) - (\mathbf{Q} \star_{\text{row}} \mathbf{s}_{k_1})(\mathbf{U} \star_{\text{row}} \mathbf{v}_{k_2})^\top) \right\|_F^2 \\ & + \frac{\rho}{2} (\|\mathbf{Q}\|_F^2 + \|\mathbf{S}\|_F^2 + \|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \end{aligned}$$

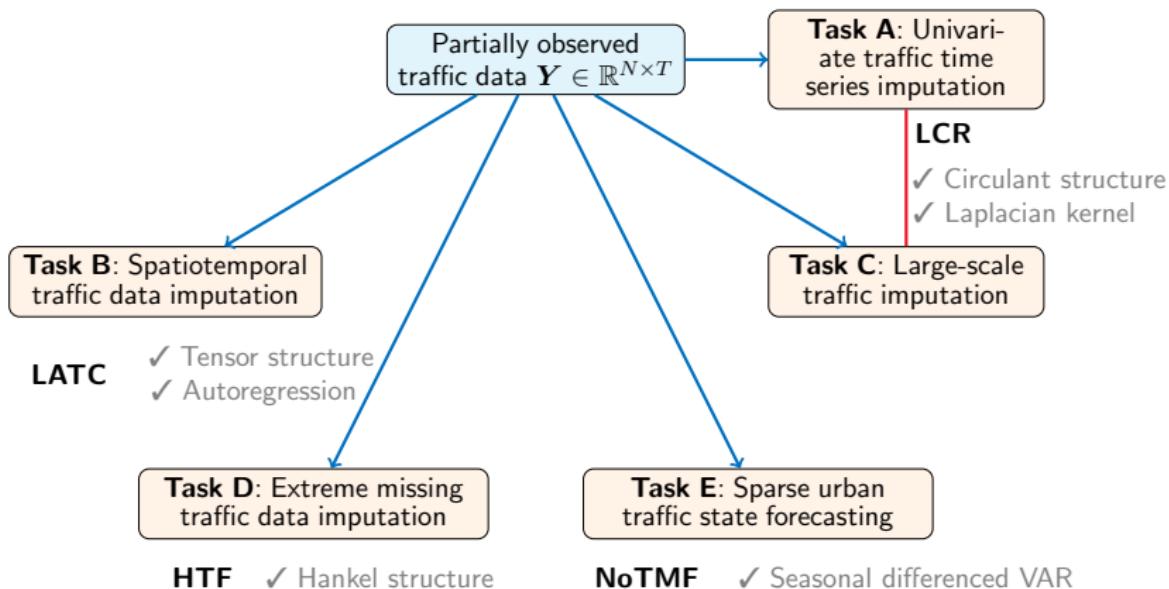
- Alternating minimization:

$$\left\{ \begin{array}{l} \mathbf{Q} := \{ \mathbf{Q} \mid \frac{\partial f}{\partial \mathbf{Q}} = \mathbf{0} \} \\ \mathbf{s}_{k_1} := \{ \mathbf{s}_{k_1} \mid \frac{\partial f}{\partial \mathbf{s}_{k_1}} = \mathbf{0} \}, k_1 \in \{1, 2, \dots, \tau_1\} \\ \mathbf{U} := \{ \mathbf{U} \mid \frac{\partial f}{\partial \mathbf{U}} = \mathbf{0} \} \\ \mathbf{v}_{k_2} := \{ \mathbf{v}_{k_2} \mid \frac{\partial f}{\partial \mathbf{v}_{k_2}} = \mathbf{0} \}, k_2 \in \{1, 2, \dots, \tau_2\} \end{array} \right.$$

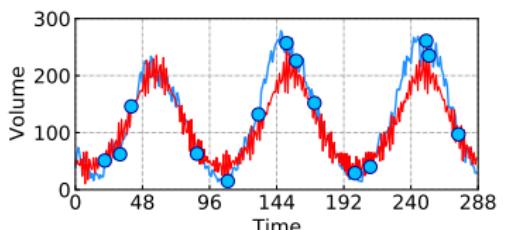


# Whole Picture

We are working on **spatiotemporal traffic data modeling**.



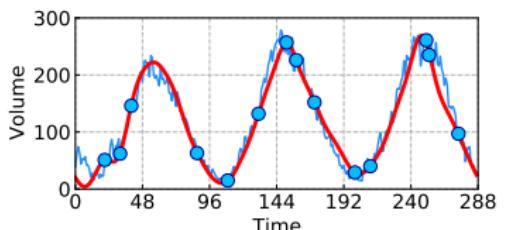
# Univariate Traffic Time Series Imputation



**CircNNM:**

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

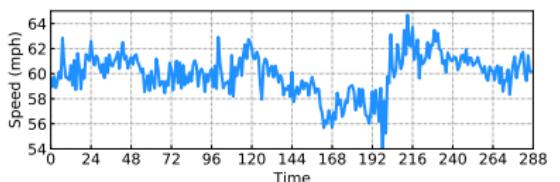
↓ Plus temporal regularization (TR)



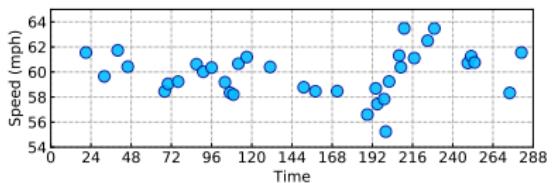
**LCR:**

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

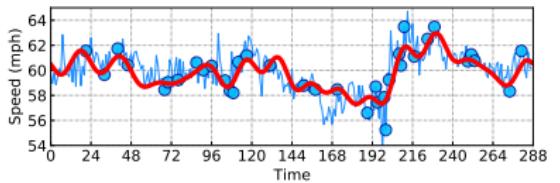
# Univariate Traffic Time Series Imputation



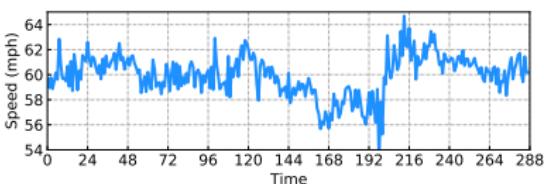
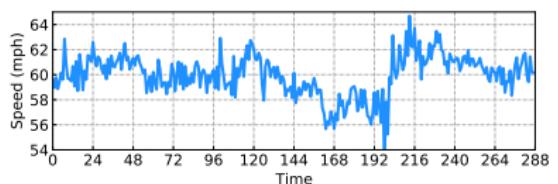
↓ Mask 90% observations



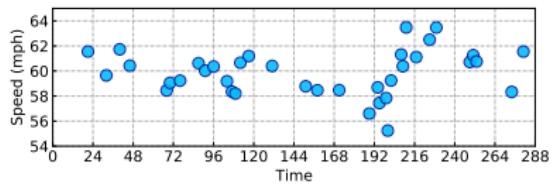
↓ Reconstruct time series



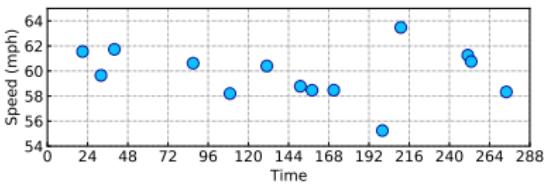
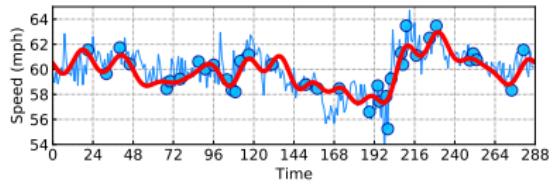
# Univariate Traffic Time Series Imputation



↓ Mask 90% observations

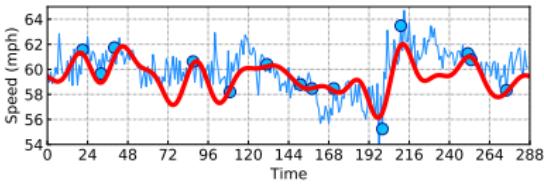


↓ Reconstruct time series



↓ Mask 95% observations

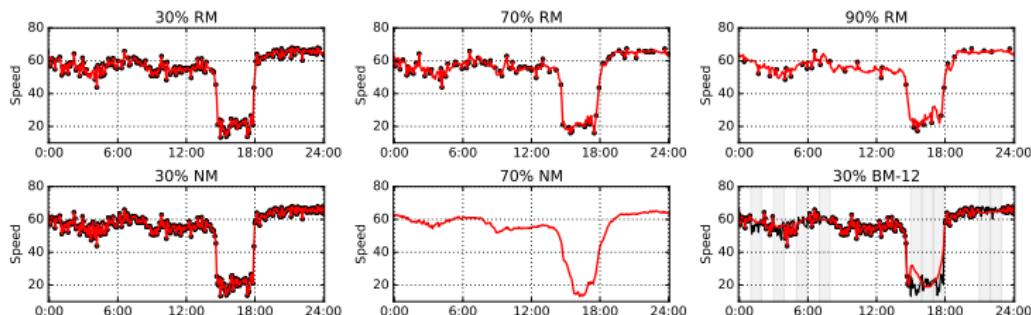
↓ Reconstruct time series



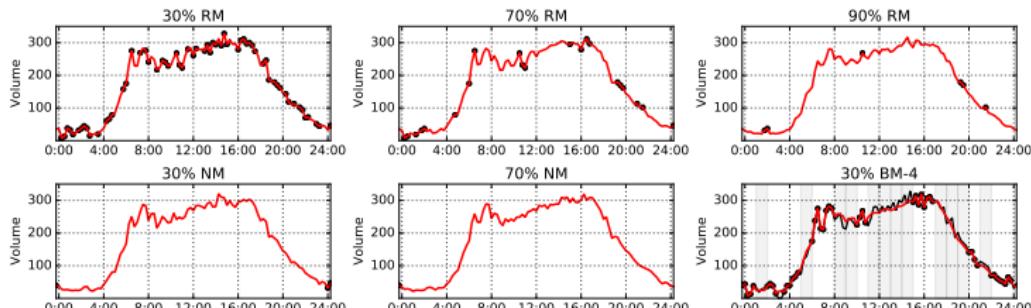
# Spatiotemporal Traffic Data Imputation

## LATC imputation

- Seattle freeway traffic speed data



- Portland highway traffic volume data



Background  
oooooooo

Literature Review  
oooo

NoTMF  
oooooo

LATC  
ooooooo

LCR  
oooooooo

HTF  
ooooooooo

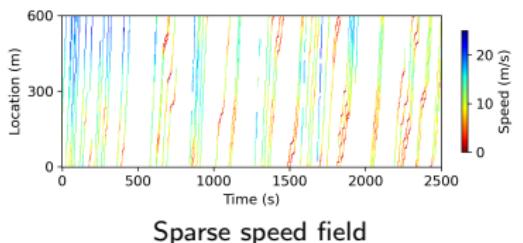
Experiments  
oooo●oo

Conclusion  
oooo

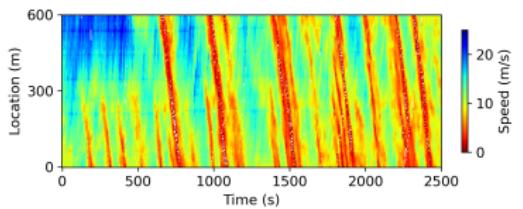
# Large-Scale Traffic Data Imputation

LCR

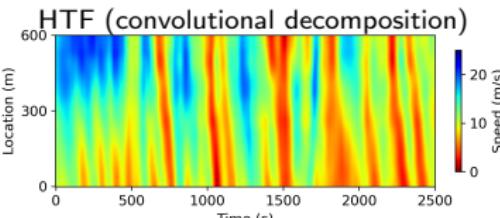
# Extreme Missing Traffic Data Imputation



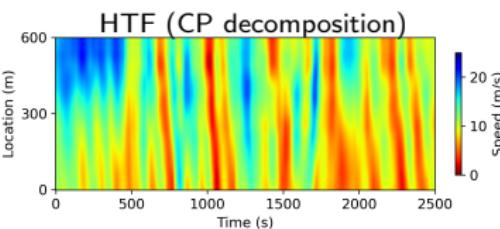
Sparse speed field



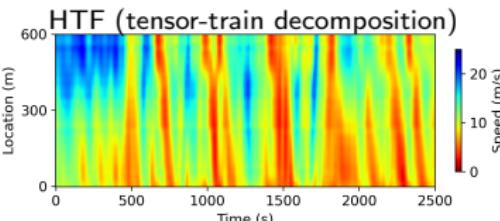
Ground truth speed field



MAPE = 51.92%



MAPE = 53.93%



MAPE = 56.48%

# Sparse Urban Traffic State Forecasting

NoTMF

# Conclusion

## Time-Varying Autoregression:

- (Highlight) Interpretable model with tensor factorization.
  - ✓ Parameter compression ✓ Pattern discovery

## Laplacian Convolutional Representation:

- (Solution) Time series trend modeling in the low-rank framework?
  - Global time series trend modeling (low-rank model):

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

- Local time series trend modeling (temporal regularization):

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

- (Highlight) A unified framework with the **FFT** implementation.

## Hankel Tensor Factorization:

- (Highlight) Memory-efficient **Hankel indexing & convolutional parameterization**.

## **Our studies:**

- ② X. Chen, Z. Cheng, N. Saunier, L. Sun (2022). Laplacian convolutional representation for traffic time series imputation. arXiv preprint arXiv:2212.01529.
- 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.

## **GitHub repository:**

- **transdim**: Machine learning for spatiotemporal traffic data imputation and forecasting. (1,000 stars & 270 forks on GitHub)  
<https://github.com/xinychen/transdim>



Dr. HanQin Cai



Xiaoxu Chen



Dr. Zhanhong Cheng



Chengyuan Zhang



# Thanks for your attention!

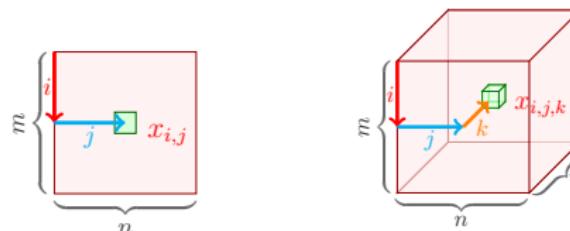
Any Questions?

## About me:

- Homepage: <https://xinychen.github.io>
- GitHub: <https://github.com/xinychen>
- How to reach me: [chenxy346@gmail.com](mailto:chenxy346@gmail.com)

# What Is Tensors?

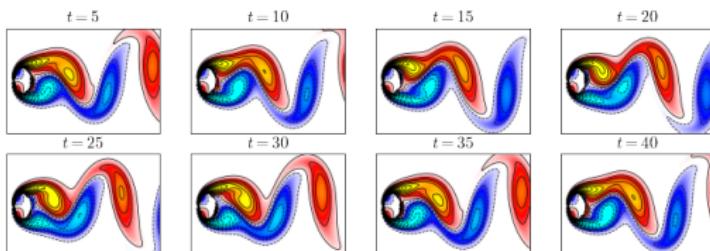
- What is tensor?  $\mathbf{X} \in \mathbb{R}^{m \times n}$  vs.  $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$



- Tensors are everywhere!



Color image with  
RGB channels



Dynamical system (fluid flow)

# Appendix