



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE



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

Ph.D. Defense

**Xinyu Chen**

Polytechnique Montréal, Canada

December 11, 2023



**President**  
Prof. Francesco Ciari  
Polytechnique Montréal



**Supervisor**  
Prof. Nicolas Saunier  
Polytechnique Montréal



**Co-supervisor**  
Prof. Lijun Sun  
McGill University



**Member**  
Prof. James Goulet  
Polytechnique Montréal



**External member**  
Prof. Guillaume Rabusseau  
Université de Montréal

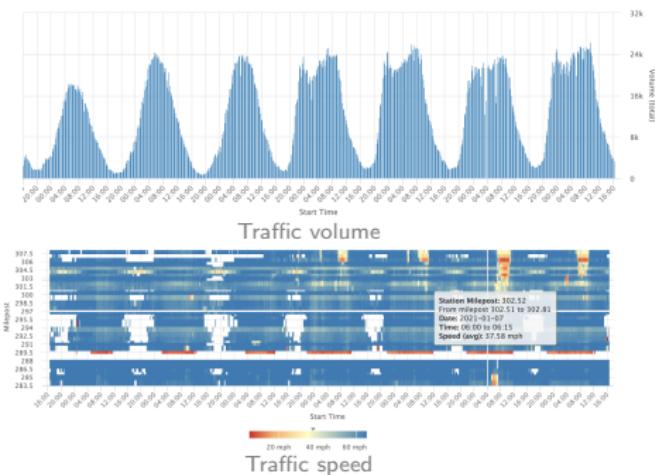
# Outline

1. Background
2. Literature Review
3. Nonstationary Temporal Matrix Factorization (NoTMF)
4. Low-Rank Autoregressive Tensor Completion (LATC)
5. Laplacian Convolutional Representation (LCR)
6. Hankel Tensor Factorization (HTF)
7. Experiments
8. Conclusion

## Traffic Flow Data

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

- Portland highway traffic data<sup>1</sup>



- $\mathbf{X} \in \mathbb{R}^{N \times T}$  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>

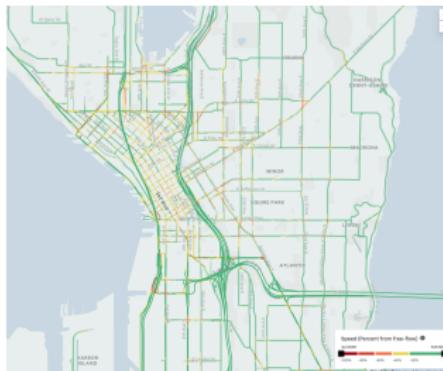
## Urban Movement Data

## High-dimensional & sparse

- Uber (hourly) movement speed data



## NYC movement



## Seattle movement

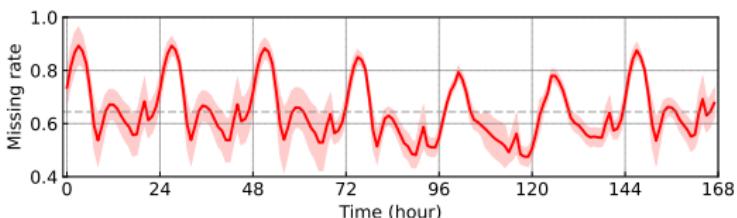
- $\{($ road segment, time slot (hour) $\}),$  average speed
  - Computing hourly speed: Road segments have 5+ unique trips.

**Issue:** Insufficient sampling of ridesharing vehicles on the road network!

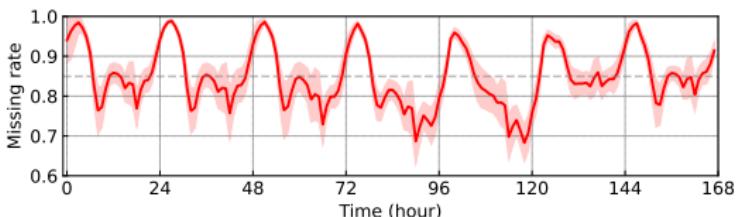
Urban Movement Data

## High-dimensional & sparse

- **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%



## Spatiotemporal Traffic Data

Sparse	
High-dimensional	
Multidimensional	
Noises & outliers	
Nonstationary	

Traffic data show complicated spatiotemporal patterns and correlations.

## Problem Formulation

**Objective A:** Impute missing values in the data matrix  $\mathbf{Y} \in \mathbb{R}^{N \times T}$  (or tensor  $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$ ).



- Matrix completion (Observed index set  $\Omega$ )

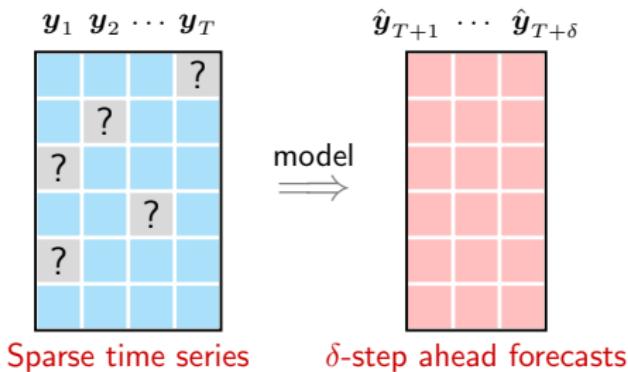
$$\underbrace{\mathcal{P}_\Omega(\mathbf{Y})}_{\text{Partially observed}} \xrightarrow{\text{Estimate}} \underbrace{\mathcal{P}_\Omega^\perp(\mathbf{Y})}_{\text{Unobserved}}$$

## Modeling process:

- How to make use of spatiotemporal traffic patterns?
  - 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  $\mathbf{y}_{T+\delta}, \delta \in \mathbb{N}^+$ .

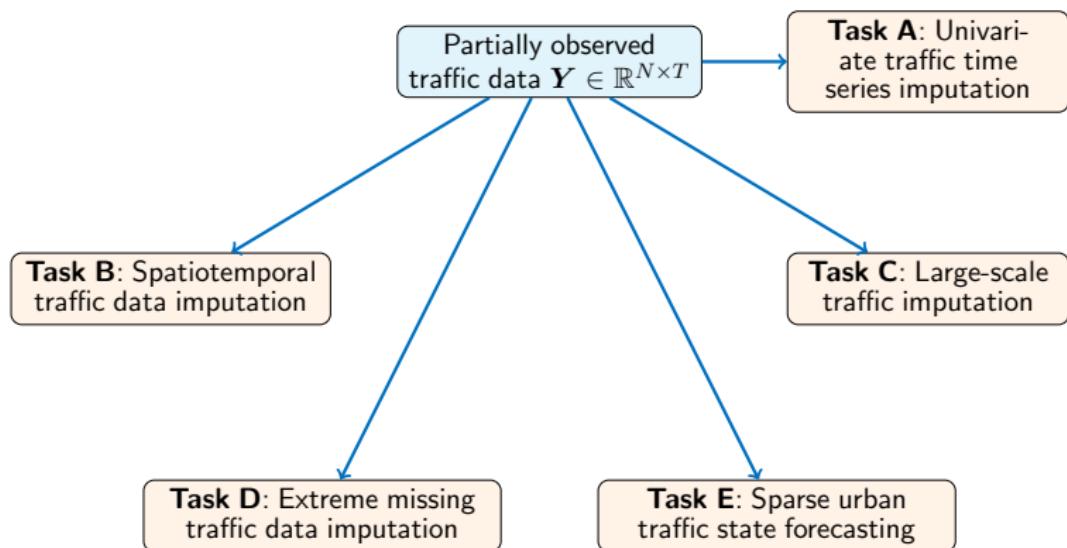


Modeling process:

- How to characterize time series dynamics in high-dimensional and sparse traffic data?

# Tasks

We are working on **spatiotemporal traffic data imputation and forecasting**.



## Imputation & Forecasting

Traffic data imputation

- Time series autoregression  
(Schafer'97, Chen & Shao'00)
  - Principal component analysis  
(Qu et al.'09, Li et al.'13)
  - Matrix factorization  
(Asif et al.'13, Asif et al.'16)
  - Tensor factorization  
(Tan et al.'13, Chen et al.'19)
  - Low-rank tensor completion  
(Ran et al.'16, Chen et al.'20)
  - Temporal matrix/tensor factorization  
(Chen & Sun'22)

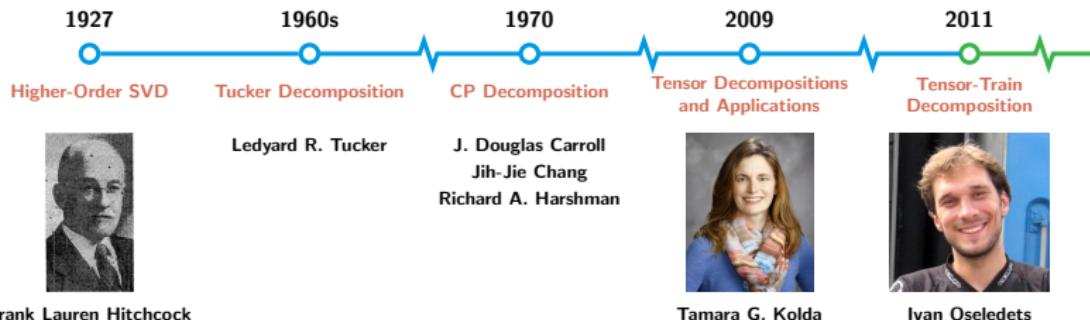
## Time series forecasting on sparse data

- Autoregression predictor  
(Anava et al.'15)
  - Prediction on the imputed data  
(e.g., Che et al.'18)
  - Dynamic tensor completion  
(Tan et al.'16)
  - Temporal matrix factorization  
(Yu et al.'16, Chen & Sun'22)
  - Online matrix factorization  
(Gultekin & Paisley'18)

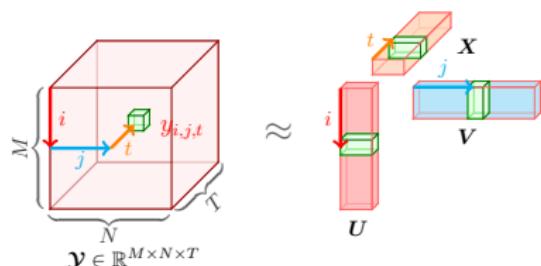
## Autoregression, matrix/tensor factorization/completion, ...

# Tensor Factorization

- Revisit tensor factorization

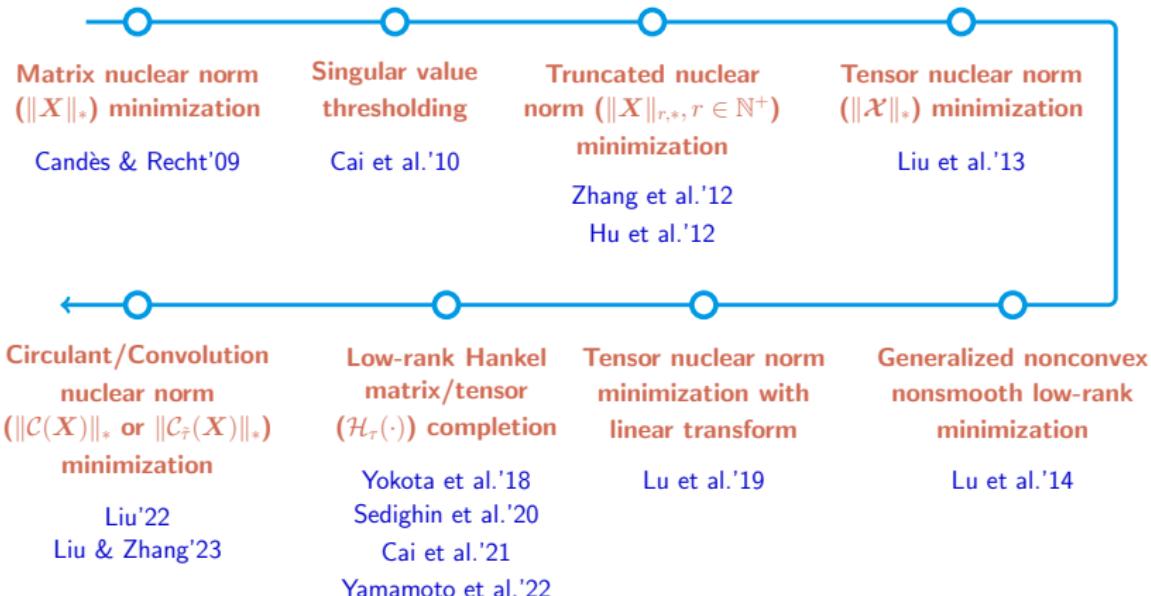


- **CP tensor factorization:** Factorize  $\mathcal{Y}$  into the combination of three rank- $R$  factor matrices (i.e., low-dimensional latent factors).



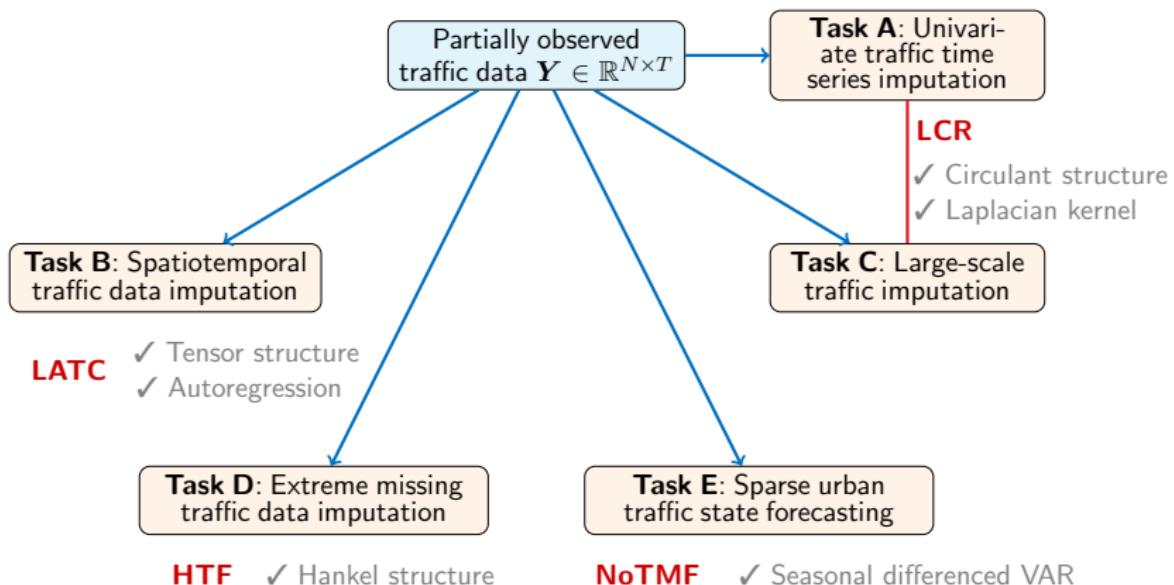
$$\begin{cases} y_{i,j,t} \approx \sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r} \\ \mathbf{y} \approx \sum_{r=1}^R \mathbf{u}_r \otimes \mathbf{v}_r \otimes \mathbf{x}_r \end{cases}$$

## Matrix/Tensor Completion



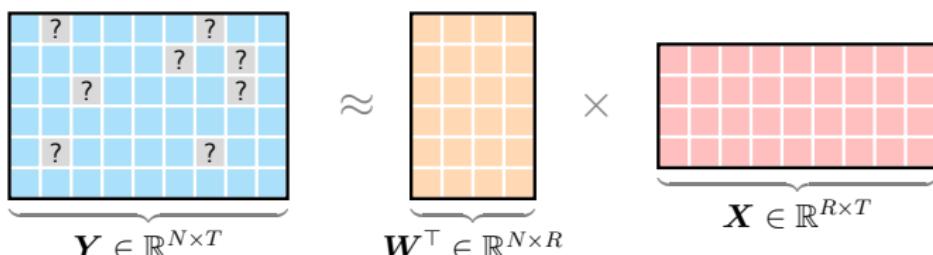
# Overview

Matrix/Tensor methods + temporal modeling (e.g., smoothing & autoregression)



# Matrix Factorization

A simple approach to reconstruct missing values.



## MF (Koren et al.'09)

Estimating low-dimensional  $\mathbf{W}, \mathbf{X}$ :

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

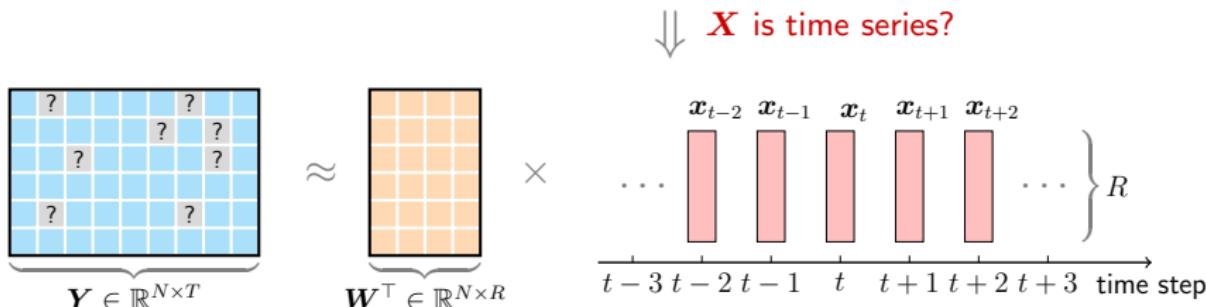
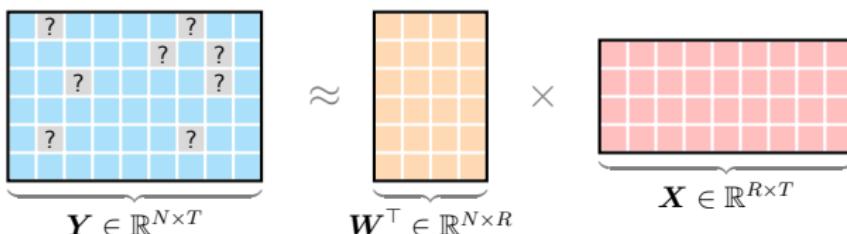
on data  $\mathbf{Y}$  w/ observed index set  $\Omega$ .

- ✓ Learn from sparse data
- ✓ Spatial factor matrix  $\mathbf{W}$
- ✓ Temporal factor matrix  $\mathbf{X}$

How to build temporal correlations on MF?

# Temporal Matrix Factorization

Vector autoregression (VAR) on the temporal factor matrix.



↓  **$\mathbf{X}$  is time series?**

**Why?** Temporal factor matrix  $\mathbf{X} \in \mathbb{R}^{R \times T}$  is the low-dimensional representation of time series dynamics of  $\mathbf{Y} \in \mathbb{R}^{N \times T}$ .

# Temporal Matrix Factorization

Vector autoregression (VAR) on the temporal factor matrix.

## MF (Koren et al.'09)

Estimating low-dimensional  $\mathbf{W}, \mathbf{X}$ :

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

on data  $\mathbf{Y}$  w/ observed index set  $\Omega$ .

## dth-order VAR

$$\mathbf{x}_t = \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} + \underbrace{\boldsymbol{\epsilon}_t}_{\mathcal{N}(\mathbf{0}, \mathbf{I})}$$

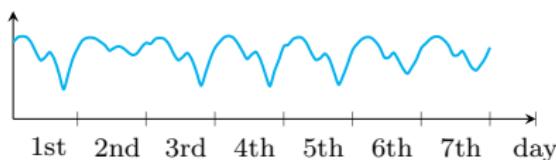
w/ coefficients  $\{\mathbf{A}_k\}$ .

↓ Yu et al.'16  
Chen & Sun'21

$$\min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k=1}^d} \frac{1}{2} \underbrace{\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2}_{\text{MF on data } \mathbf{Y}} + \frac{\gamma}{2} \underbrace{\sum_{t=d+1}^T \left\| \mathbf{x}_t - \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} \right\|_2^2}_{\text{VAR on temporal factors } \mathbf{X}}$$

# Nonstationary Temporal Matrix Factorization

Traffic data are nonstationary due to daily patterns of traffic flow.



- Season- $m$  differencing ( $m \in \mathbb{N}^+$ , e.g., daily/weekly):

$$\mathbf{x}_t \approx \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} \quad \Rightarrow \quad \mathbf{x}_t - \mathbf{x}_{t-m} \approx \sum_{k=1}^d \mathbf{A}_k (\mathbf{x}_{t-k} - \mathbf{x}_{t-m-k})$$

- (Ours) Optimization problem:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k=1}^d} & \frac{1}{2} \underbrace{\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2}_{\text{MF on data } \mathbf{Y}} + \frac{\rho}{2} \underbrace{(\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2)}_{\text{Regularization}} \\ & + \frac{\gamma}{2} \underbrace{\sum_{t=d+m+1}^T \left\| (\mathbf{x}_t - \mathbf{x}_{t-m}) - \sum_{k=1}^d \mathbf{A}_k (\mathbf{x}_{t-k} - \mathbf{x}_{t-m-k}) \right\|_2^2}_{\text{VAR on seasonal differenced temporal factors}} \end{aligned}$$

# Nonstationary Temporal Matrix Factorization

## Rewrite NoTMF

- Optimization problem:

$$\min_{\mathbf{W}, \mathbf{X}, \mathbf{A}} \frac{1}{2} \underbrace{\|\mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^T \mathbf{X})\|_F^2}_{\text{MF on data } \mathbf{Y}} + \frac{\rho}{2} \underbrace{(\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2)}_{\text{Regularization}} + \frac{\gamma}{2} \underbrace{\|\mathbf{X} \Psi_0^T - \mathbf{A} (\mathbf{I}_d \otimes \mathbf{X}) \Psi^T\|_F^2}_{\text{Temporal modeling on } \mathbf{X}}$$

where  $\Psi_0 \in \mathbb{R}^{(T-d-m) \times T}$  and  $\Psi \in \mathbb{R}^{(T-d-m) \times (dT)}$  are temporal operators.

- Alternating minimization (let  $f$  be the obj.):

$$\left\{ \begin{array}{ll} \text{Spatial factors} & \mathbf{W} := \{\mathbf{W} \mid \frac{\partial f}{\partial \mathbf{W}} = \mathbf{0}\} \quad (\text{least squares}) \\ \text{Temporal factors} & \mathbf{X} := \{\mathbf{X} \mid \frac{\partial f}{\partial \mathbf{X}} = \mathbf{0}\} \quad (\text{conjugate gradient}) \\ \text{VAR coefficients} & \mathbf{A} := \{\mathbf{A} \mid \frac{\partial f}{\partial \mathbf{A}} = \mathbf{0}\} \quad (\text{least squares}) \end{array} \right.$$

# Nonstationary Temporal Matrix Factorization

## NoTMF forecasting?

### Implementation

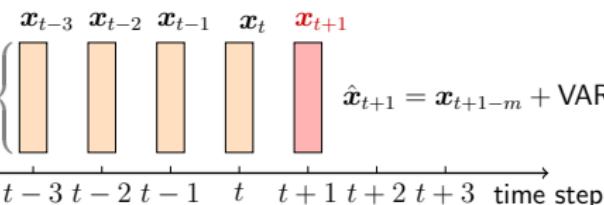
- Estimate  $\mathbf{W}, \mathbf{X}, \mathbf{A}$
- Forecast  $\hat{\mathbf{x}}_{t+1}$  with VAR
- Return  $\hat{\mathbf{y}}_{t+1} = \mathbf{W}^\top \hat{\mathbf{x}}_{t+1}$

- ✓ Sparse input  $\mathbf{Y}_t$
- ✓ Forecast in latent spaces

$$\underbrace{\mathbf{Y}_t \in \mathbb{R}^{N \times t}}_{\text{Matrix } \mathbf{Y}_t \text{ with } N \text{ rows and } t \text{ columns}} \quad \begin{matrix} ? & & & ? & ? \\ & ? & & & ? \\ ? & & & ? & ? \end{matrix}$$

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

time step



# Nonstationary Temporal Matrix Factorization

## NoTMF forecasting on streaming data?

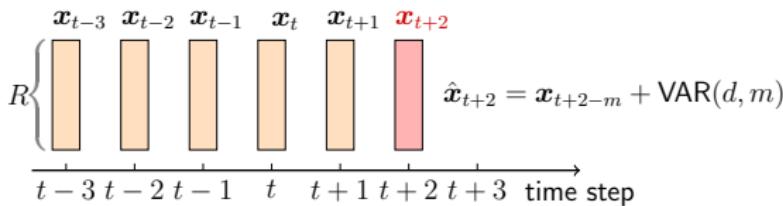
- Online forecasting (Gultekin & Paisley'18):
  - Fix the spatial factor matrix  $\mathbf{W}$
  - Use input data  $\mathbf{Y}_{t+1}$  to update the temporal factor matrix  $\mathbf{X}$  and the coefficient matrix  $\mathbf{A}$

### Implementation

- Estimate  $\mathbf{X}, \mathbf{A}$
- Forecast  $\hat{\mathbf{x}}_{t+2}$  with VAR
- Return  $\hat{\mathbf{y}}_{t+2} = \mathbf{W}^\top \hat{\mathbf{x}}_{t+2}$

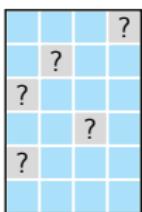
- ✓ Sparse input  $\mathbf{Y}_{t+1}$
- ✓ Forecast in latent spaces

$$\mathbf{Y}_{t+1} \in \mathbb{R}^{N \times (t+1)}$$

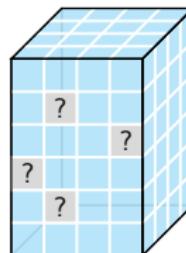


# Matrix/Tensor Completion

Problem? Impute missing values in matrices/tensors.



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



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

Cornerstone: Nuclear norm minimization

## LRMC (Candès & Recht'09)

Estimating the matrix  $\mathbf{X}$ :

$$\min_{\mathbf{X}} \|\mathbf{X}\|_*$$

$$\text{s.t. } \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{Y})$$

on data  $\mathbf{Y}$  w/ observed index set  $\Omega$ .

vs.

## LRTC (Liu et al.'13)

Estimating the tensor  $\mathcal{X}$ :

$$\min_{\mathcal{X}} \|\mathcal{X}\|_*$$

$$\text{s.t. } \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathbf{Y})$$

on data  $\mathbf{Y}$  w/ observed index set  $\Omega$ .

**Limitation:** Nuclear norm minimization only covers global consistency.

# Low-Rank Autoregressive Tensor Completion

- Introduce traffic tensors with day dimension<sup>2</sup> (Tan et al.'13, Chen et al.'19, ...).



- Build temporal correlations with univariate autoregression.

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

$$\|\mathbf{Y}\|_{\mathbf{A}, \mathcal{H}} \triangleq \sum_{n,t} \left( y_{n,t} - \sum_k \mathbf{a}_{n,k} y_{n,t-h_k} \right)^2$$

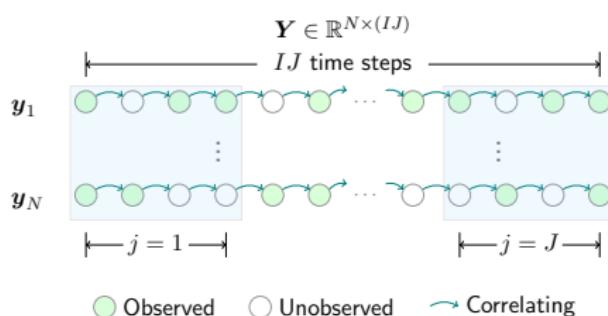
w/ the time lag set  $\mathcal{H} = \{h_1, \dots, h_d\}$  and the coefficient matrix  $\mathbf{A} \in \mathbb{R}^{N \times d}$ .

---

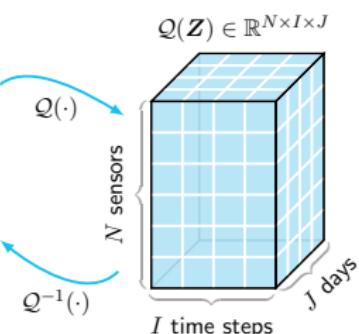
<sup>2</sup>There are  $T = IJ$  time steps in total.

# Low-Rank Autoregressive Tensor Completion

**Local consistency** w/ autoregression



**Global consistency** w/ tensor structure



## LATC

Optimization problem:

$$\min_{\mathbf{Z}, \mathbf{A}} \underbrace{\|Q(\mathbf{Z})\|_{r,*}}_{\text{global}} + \frac{\gamma}{2} \underbrace{\|\mathbf{Z}\|_{\mathbf{A}, \mathcal{H}}}_{\text{local}}$$

$$\text{s.t. } \mathcal{P}_\Omega(\mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{Y})$$

on data  $\mathbf{Y}$  w/ observed index set  $\Omega$ .

## Two subproblems

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

# Low-Rank Autoregressive Tensor Completion

$Z$ -subproblem:

$$\mathbf{Z} := \arg \min_{\mathcal{P}_\Omega(\mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{Y})} \|\mathcal{Q}(\mathbf{Z})\|_{r,*} + \frac{\gamma}{2} \|\mathbf{Z}\|_{\mathbf{A},\mathcal{H}}$$

- Augmented Lagrangian function:<sup>3</sup>

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

## Implementation

Repeat

- Compute  $\mathbf{Z}$
- Compute  $\mathbf{A}$



## Implementation

Repeat

- Repeat
  - # Alternating Direction Method of Multipliers (ADMM)
  - Compute  $\mathbf{X}$
  - Compute  $\mathbf{Z}$
  - Compute  $\mathbf{W}$
- Compute  $\mathbf{A}$

---

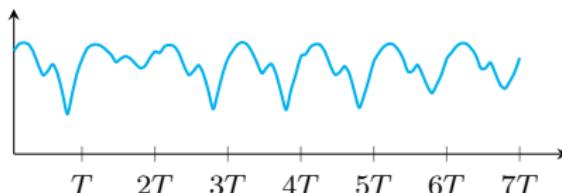
<sup>3</sup> $\mathbf{W} \in \mathbb{R}^{N \times I \times J}$  (Lagrange multiplier);  $\langle \cdot, \cdot \rangle$  (inner product). The indicator function:

$$\pi(\mathbf{Z}) = \begin{cases} 0, & \text{if } \mathcal{P}_\Omega(\mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{Y}), \\ +\infty, & \text{otherwise.} \end{cases}$$

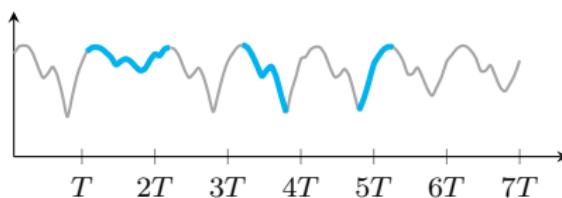
# Laplacian Convolutional Representation

## Motivation: Time series imputation

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



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

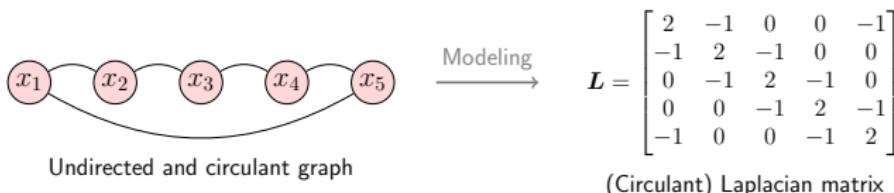


How to characterize both global and local trends in sparse time series?

# Laplacian Convolutional Representation

## Local trend modeling

- Intuition of (circulant) Laplacian matrix



- Define Laplacian kernel:

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

↓

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

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

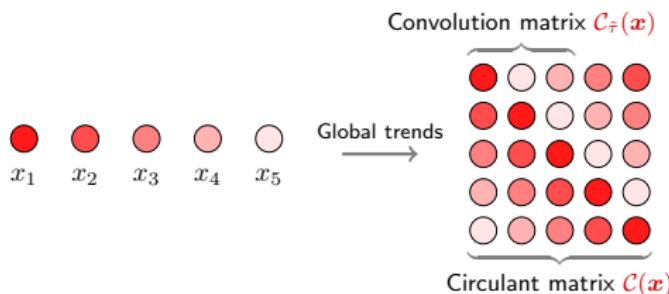
- (Laplacian) Temporal regularization:

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

Reformulate temporal regularization with circular convolution.

# Laplacian Convolutional Representation

**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  $\Omega$ .

## 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  $\Omega$ .

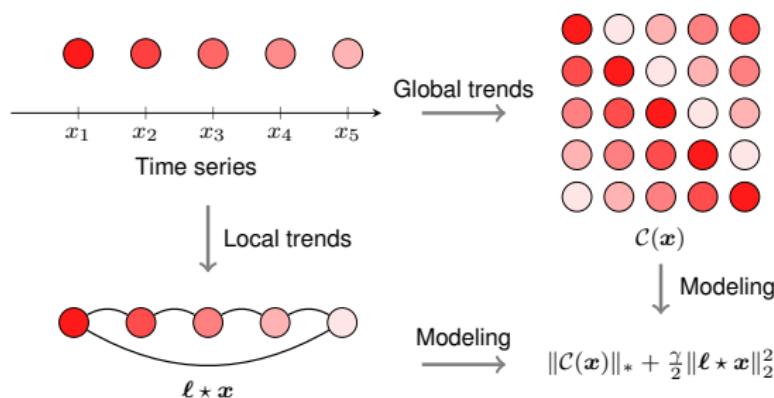
# Laplacian Convolutional Representation

## 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.,

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

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



## Laplacian Convolutional Representation

- Augmented Lagrangian function:<sup>4</sup>

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

- 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  $\mathbf{x}$ ?

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

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

---

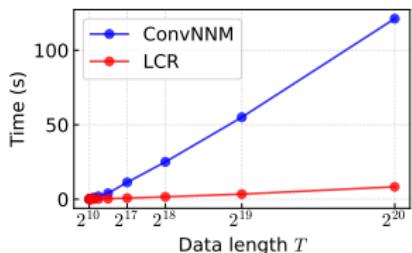
<sup>4</sup> $\mathbf{w} \in \mathbb{R}^T$  (Lagrange multiplier);  $\langle \cdot, \cdot \rangle$  (inner product).

# Laplacian Convolutional Representation

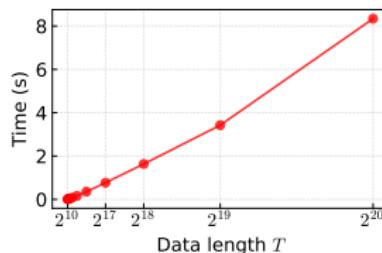
## Empirical time complexity

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'22](#), [Liu & Zhang'23](#))
  - Convolution matrix  $C_{\tilde{\tau}}(\mathbf{y}) \in \mathbb{R}^{T \times \tilde{\tau}}$  with kernel size  $\tilde{\tau} = 2^4$
  - Singular value thresholding in  $\mathcal{O}(\tilde{\tau}^2 T)$



ConvNNM vs. LCR



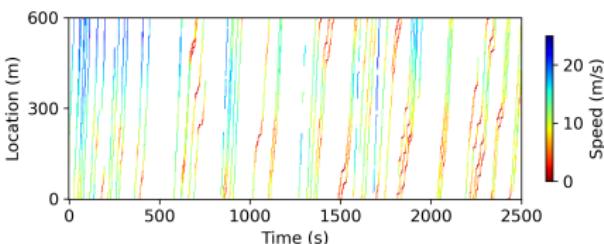
LCR

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

# Hankel Tensor Factorization

## Motivation: Spatiotemporal data reconstruction

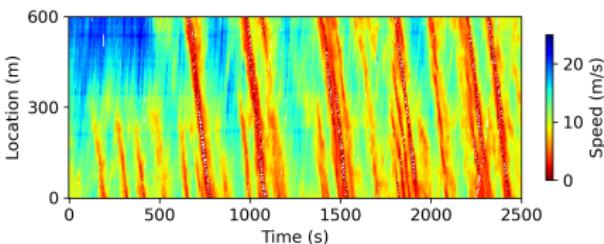
- Sparse speed field reconstruction problem in vehicular traffic flow.



200-by-500 matrix  
(NGSIM)



Reconstruct speed field from  
5% sparse trajectories?



How to characterize both spatial and temporal dependencies?

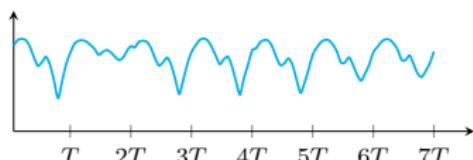
# Hankel Tensor Factorization

- Hankel matrix

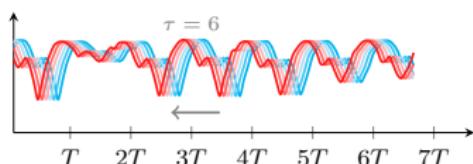
- Given  $\mathbf{x} = (1, 2, 3, 4, 5)^\top$  and window length  $\tau = 2$ , we have

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

- Automatic temporal modeling



Traffic time series



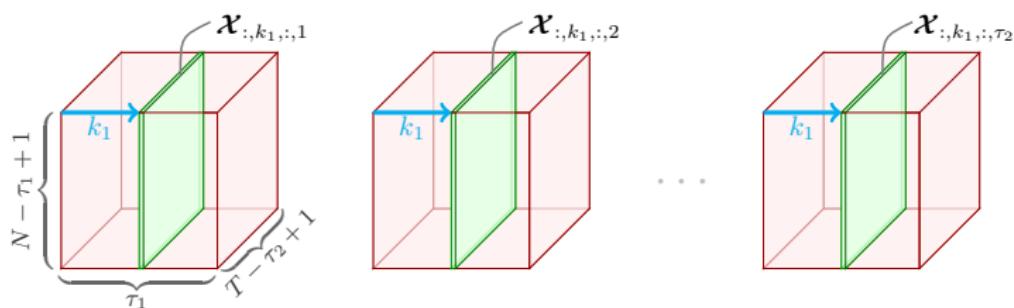
Hankel matrix

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

# Hankel Tensor Factorization

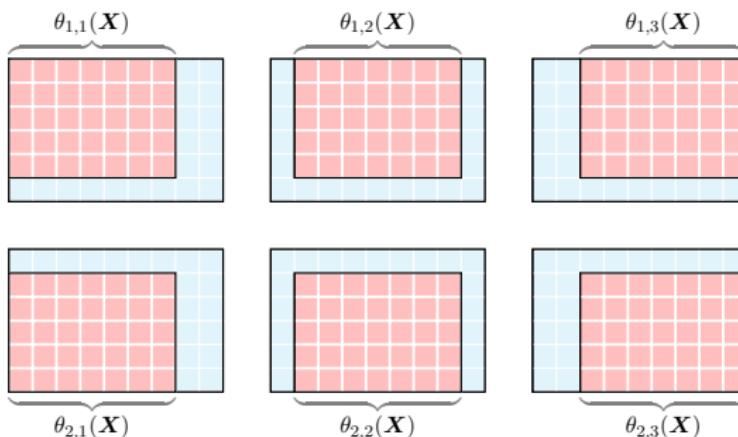
## Hankel indexing

- Sampling function for the Hankel tensor:

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

referring to as 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  $\star_{\text{row}}$ ):

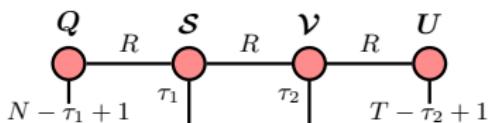
$$\theta_{k_1, k_2}(\mathbf{Y}) \approx (\mathbf{Q} \star_{\text{row}} \mathbf{s}_{k_1}^{\top})(\mathbf{U} \star_{\text{row}} \mathbf{v}_{k_2}^{\top})^{\top}$$

## Baselines:

- 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



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

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

# Hankel Tensor Factorization

## HTF (convolutional decomposition)

- Optimization problem:

$$\min_{Q, S, U, V} \frac{1}{2} \underbrace{\sum_{k_1, k_2} \left\| \mathcal{P}_{\Omega_{k_1, k_2}} (\theta_{k_1, k_2}(Y) - (Q \star_{\text{row}} s_{k_1})(U \star_{\text{row}} v_{k_2})^T) \right\|_F^2}_{\text{Tensor decomposition on Hankel tensor slices}} + \frac{\rho}{2} (\|Q\|_F^2 + \|S\|_F^2 + \|U\|_F^2 + \|V\|_F^2)$$

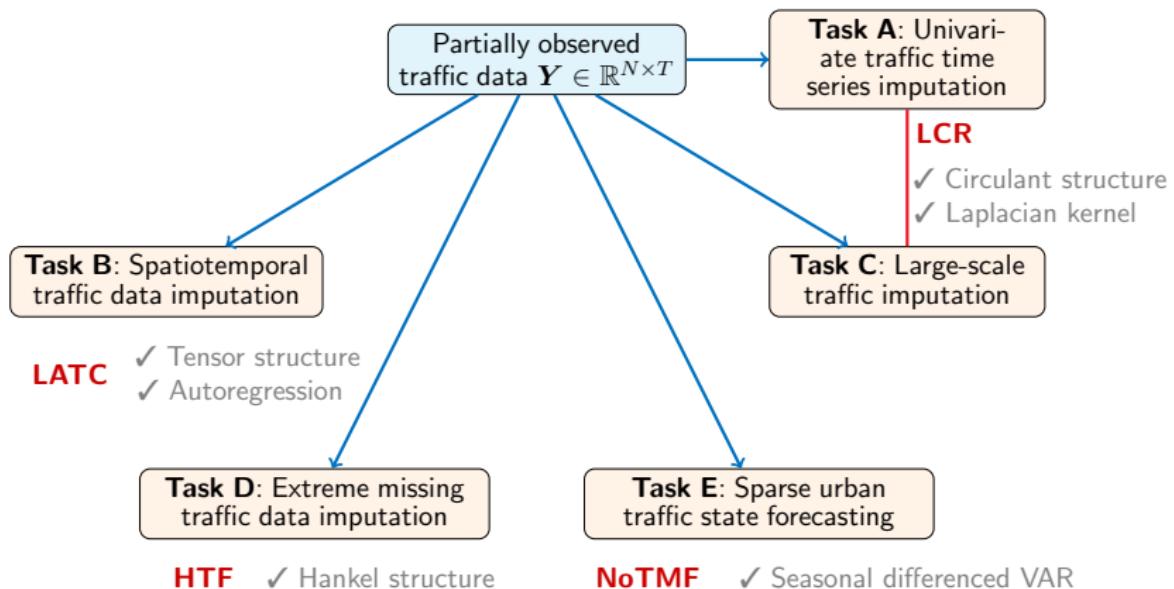
- Alternating minimization (let  $f$  be the obj.):

$$\begin{cases} Q := \{Q \mid \frac{\partial f}{\partial Q} = \mathbf{0}\} & \text{(conjugate gradient)} \\ s_{k_1} := \{s_{k_1} \mid \frac{\partial f}{\partial s_{k_1}} = \mathbf{0}\}, \forall k_1 & \text{(conjugate gradient)} \\ U := \{U \mid \frac{\partial f}{\partial U} = \mathbf{0}\} & \text{(conjugate gradient)} \\ v_{k_2} := \{v_{k_2} \mid \frac{\partial f}{\partial v_{k_2}} = \mathbf{0}\}, \forall k_2 & \text{(conjugate gradient)} \end{cases}$$

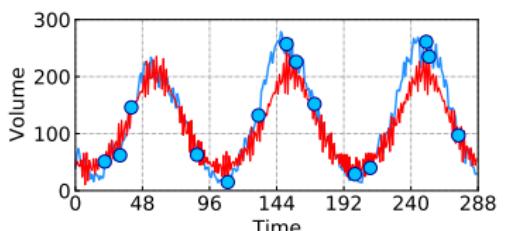
- Memory-efficient but still computationally costly!

# Overview

We are working on **spatiotemporal traffic data imputation and forecasting**.



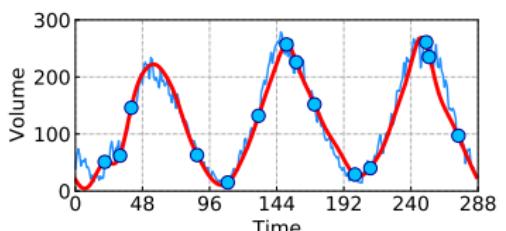
## Task A: Univariate Traffic Time Series Imputation



**CircNNM:**

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

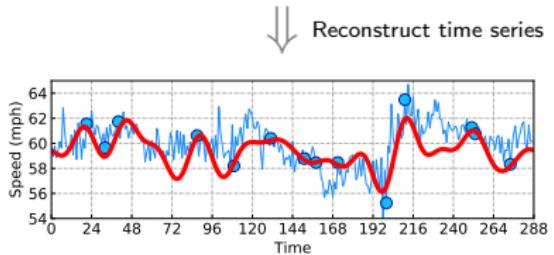
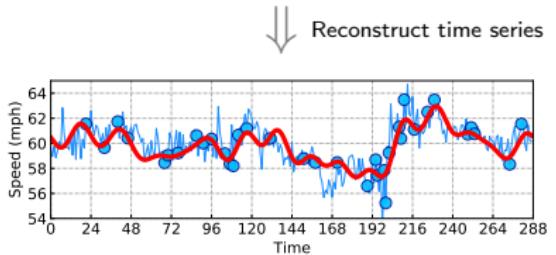
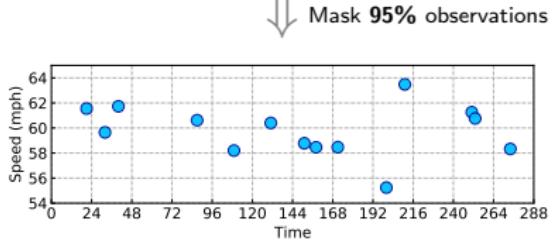
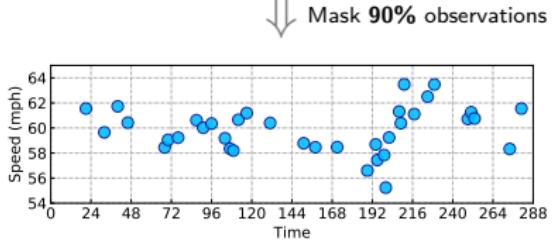
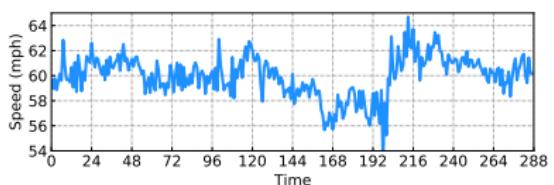
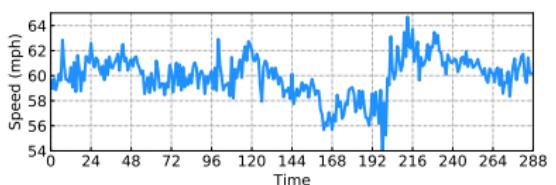
↓ Plus temporal regularization



**LCR:**

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

## Task A: Univariate Traffic Time Series Imputation



## Task B: Spatiotemporal Traffic Data Imputation

LATC vs. baseline models (in MAPE/RMSE)

- On the Seattle freeway traffic speed dataset ( $\mathbf{Y} \in \mathbb{R}^{323 \times 8064}$ )

Missing rate	LATC	LAMC	LRTC-TNN	BTMF	SPC
30%, Random Missing	<b>4.90/3.16</b>	5.98/3.73	4.99/3.20	5.91/3.72	5.92/3.62
70%, Random Missing	<b>5.96/3.71</b>	8.02/4.70	6.10/3.77	6.47/3.98	7.38/4.30
90%, Random Missing	<b>7.46/4.50</b>	10.56/5.91	8.08/4.80	8.17/4.81	9.75/5.31
30%, Nonrandom Missing	7.10/4.33	6.99/4.25	<b>6.85/4.21</b>	9.26/5.36	8.87/4.99
70%, Nonrandom Missing	9.40/5.40	9.75/5.60	<b>9.23/5.35</b>	10.47/6.15	11.32/5.92
30%, Block-out Missing	<b>9.43/5.36</b>	27.05/13.66	9.52/5.41	14.33/13.60	11.30/5.84

- On the Portland highway traffic volume dataset ( $\mathbf{Y} \in \mathbb{R}^{1156 \times 2976}$ )

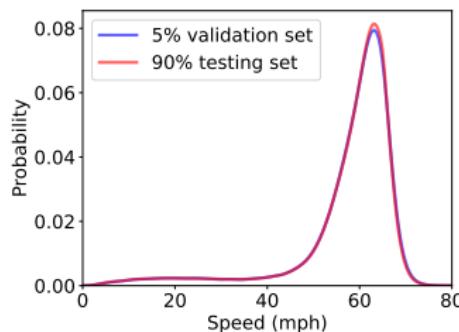
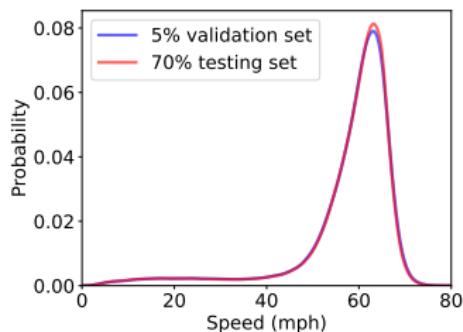
Missing rate	LATC	LAMC	LRTC-TNN	BTMF	SPC
30%, Random Missing	<b>16.95/15.99</b>	17.93/16.03	17.27/16.08	18.22/19.14	21.29/56.73
70%, Random Missing	<b>19.59/18.70</b>	21.26/19.37	19.99/18.73	19.96/22.21	24.35/43.32
90%, Random Missing	23.15/22.83	25.64/23.75	<b>22.90/22.68</b>	23.90/25.71	28.45/39.65
30%, Nonrandom Missing	<b>19.48/19.14</b>	19.93/19.69	19.59/ <b>18.91</b>	19.55/20.38	26.96/60.33
70%, Nonrandom Missing	27.67/45.03	25.75/28.25	30.26/60.85	<b>23.86/26.74</b>	33.42/47.34
30%, Block-out Missing	<b>24.01/23.50</b>	29.21/27.60	31.74/74.42	27.85/25.68	31.01/60.33

- LATC vs. LAMC: The significance of tensor representation
- LATC vs. LRTC-TNN: The significance of temporal autoregression

## Task B: Spatiotemporal Traffic Data Imputation

**Parameter tuning process:** Training set, validation set, and testing set?

- Random missing on the Seattle freeway traffic speed dataset



- Imputation performance (e.g., 70% missing rate)

On the validation set (5% data)

$\gamma/\lambda$	Truncation parameter				
	$r = 5$	$r = 10$	$r = 15$	$r = 20$	$r = 25$
1/10	7.84/4.52	7.20/4.25	6.82/4.08	6.60/3.98	6.41/3.92
1/5	7.84/4.52	7.20/4.25	6.82/4.08	6.59/3.97	6.41/3.92
1	7.81/4.51	7.18/4.24	6.80/4.07	6.58/3.97	6.39/3.91
5	7.70/4.45	7.09/4.20	6.72/4.04	6.49/3.93	6.29/3.87
10	7.59/4.39	7.00/4.16	6.64/4.00	6.41/3.89	<b>6.22/3.83</b>

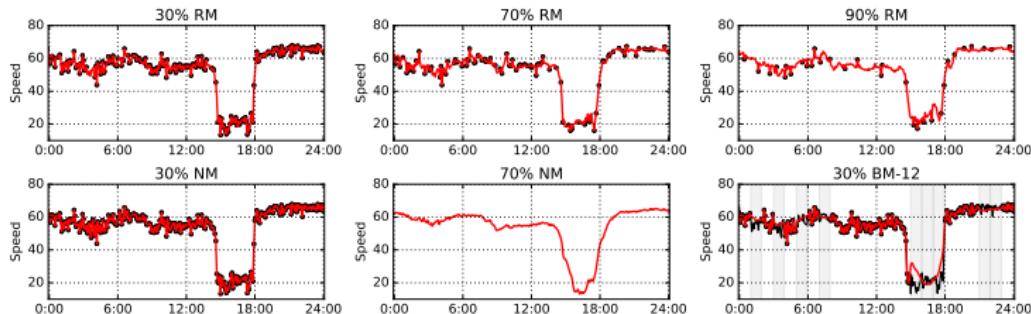
On the testing set (70% data)

$\gamma/\lambda$	Truncation parameter				
	$r = 5$	$r = 10$	$r = 15$	$r = 20$	$r = 25$
1/10	7.83/4.53	7.18/4.27	6.80/4.09	6.58/3.99	6.41/3.92
1/5	7.83/4.53	7.18/4.26	6.80/4.09	6.57/3.98	6.40/3.92
1	7.80/4.52	7.16/4.25	6.78/4.08	6.55/3.98	6.40/3.92
5	7.70/4.47	7.08/4.21	6.70/4.04	6.46/3.94	6.29/3.87
10	7.58/4.41	6.99/4.17	6.62/4.01	6.39/3.90	<b>6.21/3.84</b>

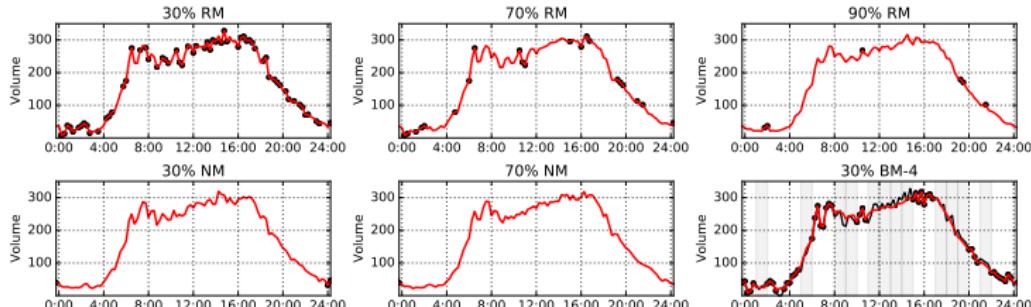
# Task B: Spatiotemporal Traffic Data Imputation

## LATC imputation

- Seattle freeway traffic speed data



- Portland highway traffic volume data



## Task C: Large-Scale Traffic Data Imputation

LCR vs. baseline models (in MAPE/RMSE)

- PeMS-4W: California freeway traffic speed dataset ( $Y \in \mathbb{R}^{11160 \times 8064}$ )

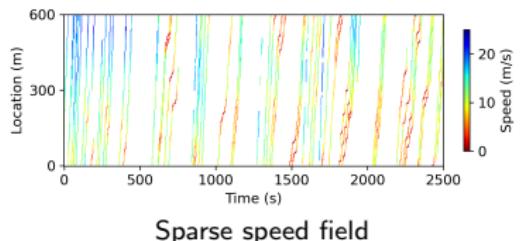
Model	Missing rate			
	30%	50%	70%	90%
LCR-2D	<b>1.50/1.49</b>	<b>1.76/1.69</b>	<b>2.07/2.06</b>	<b>3.19/3.05</b>
$LCR_N$	<b>1.48/1.50</b>	<b>1.73/1.73</b>	<b>2.07/2.12</b>	3.24/3.22
LCR	<b>1.50/1.49</b>	<b>1.76/1.69</b>	2.08/2.07	3.21/3.06
CTNNM	2.26/1.84	2.67/2.14	3.40/2.66	5.22/3.90
CircNNM	2.26/1.84	2.69/2.15	3.43/2.67	5.34/3.96
LRMC	2.04/1.80	2.43/2.12	3.08/2.66	6.05/4.43
HaLRTC	1.98/1.73	2.22/1.98	2.84/2.49	4.39/3.66
LRTC-TNN	1.68/1.55	1.93/1.77	2.33/2.14	3.40/3.10
NoTMF	2.95/2.65	3.05/2.73	3.33/2.97	5.22/4.71

### Results

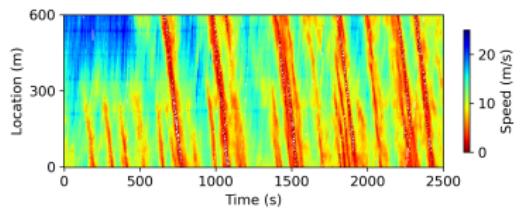
- LCR-2D > CTNNM: The importance of temporal regularization.
- CTNNM  $\geq$  CircNNM: Cyclic tensor is superior to circulant matrix.
- LCR > LRMC/LRTC: The importance of global/local modeling.

$$\mathcal{O}(NT \log(NT)) \text{ (FFT)} \quad \text{vs.} \quad \mathcal{O}(\min\{N^2T, NT^2\}) \text{ (SVD)}$$

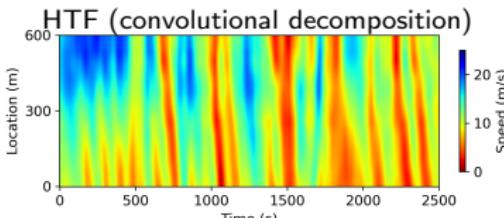
## Task D: Extreme Missing Traffic Data Imputation



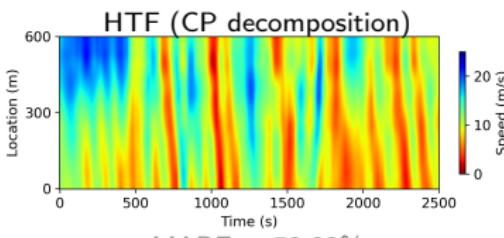
Sparse speed field



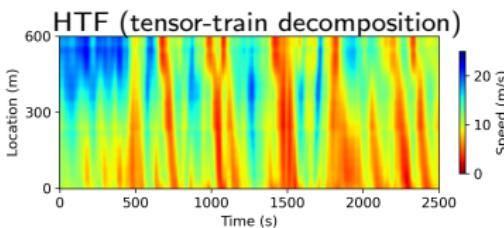
Ground truth speed field



MAPE = 51.92%



MAPE = 53.93%



MAPE = 56.48%

## Task D: Extreme Missing Traffic Data Imputation

HTF vs. baseline models (in MAPE/RMSE)

- On the Seattle freeway traffic speed dataset ( $Y \in \mathbb{R}^{323 \times 8064}$ )

Model	Random missing rate			
	80%	85%	90%	95%
HTF (convolution)	<b>6.21/3.88</b>	<b>6.51/4.06</b>	<b>6.98/4.30</b>	<b>8.02/4.84</b>
HTF (tensor-train)	8.75/5.16	9.86/5.76	9.24/5.36	9.89/5.70
HTF (CP)	7.25/4.33	7.93/4.66	8.61/4.96	9.25/5.20
LATC	6.50/4.00	6.90/4.21	7.47/4.51	8.75/5.05
LRTC-TNN	6.97/4.24	7.43/4.43	8.19/4.81	9.60/5.55
LCR	6.75/4.15	7.31/4.38	7.96/4.71	9.78/5.39
BTMF	6.85/4.17	7.36/4.42	8.13/4.79	9.63/5.48

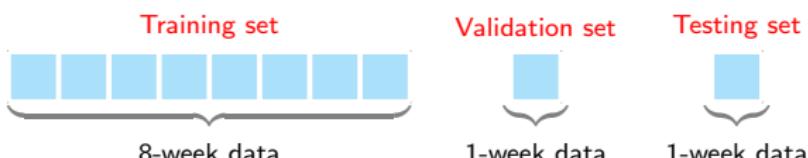
### Results

- Convolutional tensor decomposition outperforms both tensor-train and CP tensor decomposition.
- Our HTF model performs better than state-of-the-art baseline models.

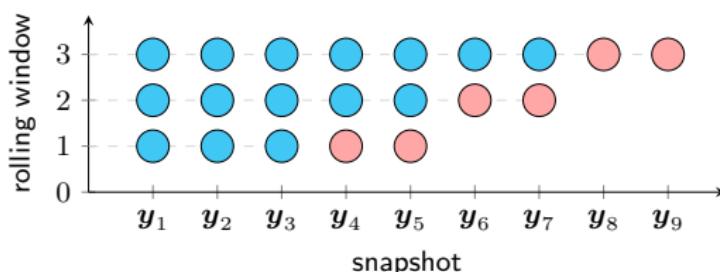
## Task E: Sparse Urban Traffic State Forecasting

### NoTMF forecasting

- NYC movement speed dataset:
  - **10-week data** of size  $98210 \times 1680$
  - Contain **66.56%** missing values
- Rolling forecasting setup (Time horizon  $\delta = 1, 2, 3, 6$ ):



- Rolling forecasting illustration ( $\delta = 2$ ):



## Task E: Sparse Urban Traffic State Forecasting

NoTMF vs. baseline models (in MAPE/RMSE)

- On the NYC Uber movement speed dataset

$\delta$	$d$	NoTMF ( $m = 24$ )	NoTMF ( $m = 168$ )	NoTMF-1st ( $m = 168$ )	TRMF	BTMF	BTRMF
1	1	13.63/2.88	13.53/2.86	<b>13.45/2.85</b>	14.50/3.12	14.94/3.13	15.93/3.33
	2	<b>13.47/2.84</b>	<b>13.41/2.84</b>	<b>13.42/2.84</b>	14.14/3.05	15.70/3.41	15.90/3.35
	3	13.46/2.84	<b>13.39/2.83</b>	13.43/2.84	13.87/2.96	15.80/3.34	16.08/3.43
	6	<b>13.41/2.83</b>	<b>13.39/2.83</b>	<b>13.41/2.83</b>	14.00/2.98	15.45/3.27	16.26/3.48
2	1	13.91/2.96	13.76/2.94	<b>13.70/2.92</b>	15.85/3.43	15.33/3.21	16.85/3.56
	2	13.77/2.92	<b>13.63/2.89</b>	13.72/2.92	15.04/3.31	15.87/3.32	17.27/3.71
	3	13.72/2.91	<b>13.61/2.89</b>	13.73/2.92	15.25/3.36	15.69/3.33	17.24/3.74
	6	<b>13.59/2.87</b>	<b>13.57/2.88</b>	13.68/2.91	14.92/3.24	15.91/3.39	18.18/3.97
3	1	14.30/3.05	14.06/3.02	<b>14.02/3.00</b>	17.52/3.83	15.86/3.32	18.61/3.91
	2	14.01/2.98	<b>13.84/2.94</b>	13.96/2.98	17.32/4.00	16.30/3.40	18.90/4.10
	3	13.95/2.97	<b>13.79/2.93</b>	13.98/2.98	16.91/3.71	16.56/3.49	18.68/4.05
	6	<b>13.78/2.92</b>	<b>13.73/2.92</b>	13.91/2.96	16.72/3.65	15.49/3.27	20.45/4.66
6	1	<b>14.61/3.11</b>	14.67/3.20	14.98/3.32	21.20/4.70	15.99/3.32	22.40/4.69
	2	<b>14.30/3.03</b>	14.33/3.09	14.90/3.28	20.87/5.01	16.04/3.33	23.56/5.63
	3	<b>14.26/3.03</b>	14.28/3.09	14.86/3.26	20.08/4.65	15.67/3.28	24.27/5.72
	6	<b>14.06/2.97</b>	14.16/3.06	14.80/3.23	20.40/4.35	16.38/3.50	26.34/6.60

NoTMF performs better than other TMF models (e.g., TRMF, BTMF & BTRMF).

## Task E: Sparse Urban Traffic State Forecasting

NoTMF vs. baseline models (in MAPE/RMSE)

- On the Seattle Uber movement speed dataset

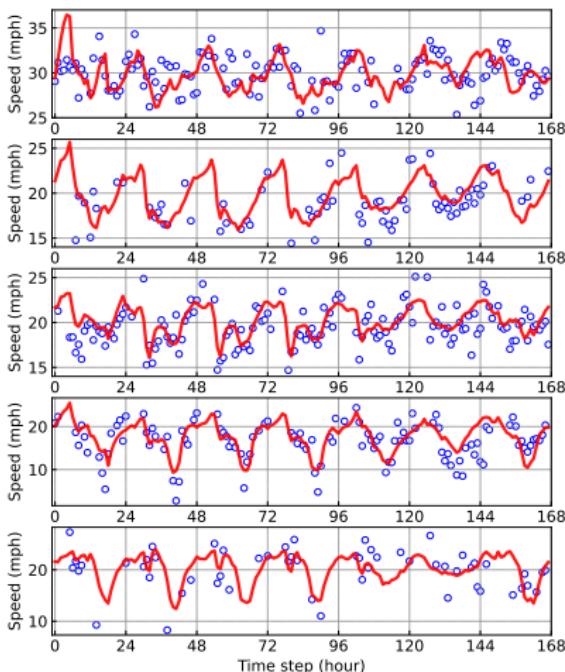
$\delta$	$d$	NoTMF ( $m = 24$ )	NoTMF ( $m = 168$ )	NoTMF-1st ( $m = 168$ )	TRMF	BTMF	BTRMF
1	1	10.45/3.32	<b>10.26</b> /3.22	<b>10.26</b> / <b>3.21</b>	11.58/3.79	12.23/3.89	12.52/4.01
	2	10.53/3.34	10.29/3.23	<b>10.23</b> / <b>3.21</b>	10.92/3.51	12.95/4.18	13.16/4.31
	3	10.42/3.30	10.30/3.22	<b>10.25</b> / <b>3.21</b>	10.86/3.47	12.96/4.22	13.89/4.64
	6	10.50/3.32	<b>10.21</b> / <b>3.21</b>	10.27/3.22	10.99/3.51	12.91/4.18	13.90/4.67
2	1	10.90/3.55	10.32/3.25	<b>10.25</b> / <b>3.23</b>	12.07/4.02	12.74/4.06	13.31/4.32
	2	10.90/3.52	10.31/3.24	<b>10.25</b> / <b>3.23</b>	12.59/4.24	13.68/4.45	13.44/4.43
	3	10.81/3.49	10.31/3.24	<b>10.27</b> / <b>3.23</b>	12.01/3.96	13.55/4.46	13.66/4.56
	6	10.57/3.38	<b>10.25</b> / <b>3.23</b>	10.27/ <b>3.23</b>	12.18/3.98	13.56/4.42	14.67/4.92
3	1	11.27/3.71	<b>10.41</b> / <b>3.29</b>	<b>10.41</b> / <b>3.29</b>	13.47/4.62	13.16/4.15	14.01/4.52
	2	11.26/3.71	<b>10.30</b> / <b>3.27</b>	10.34/ <b>3.27</b>	14.48/5.19	13.63/4.37	14.39/4.76
	3	11.11/3.62	<b>10.35</b> / <b>3.28</b>	10.38/ <b>3.28</b>	14.04/4.83	13.76/4.42	14.67/4.84
	6	10.96/3.55	<b>10.30</b> / <b>3.26</b>	<b>10.30</b> / <b>3.26</b>	13.32/4.51	13.28/4.29	15.64/5.31
6	1	11.88/3.97	10.63/3.43	<b>10.60</b> / <b>3.42</b>	15.59/5.32	13.63/4.30	16.39/5.28
	2	11.58/3.83	<b>10.55</b> / <b>3.40</b>	10.56/ <b>3.40</b>	18.66/7.20	13.27/4.19	16.77/5.58
	3	11.54/3.81	10.57/3.39	<b>10.53</b> / <b>3.38</b>	17.94/6.32	13.88/4.36	17.35/5.70
	6	11.27/3.70	10.53/ <b>3.35</b>	<b>10.50</b> / <b>3.35</b>	15.12/5.24	13.30/4.24	16.63/5.62

NoTMF performs better than other TMF models (e.g., TRMF, BTMF & BTRMF).

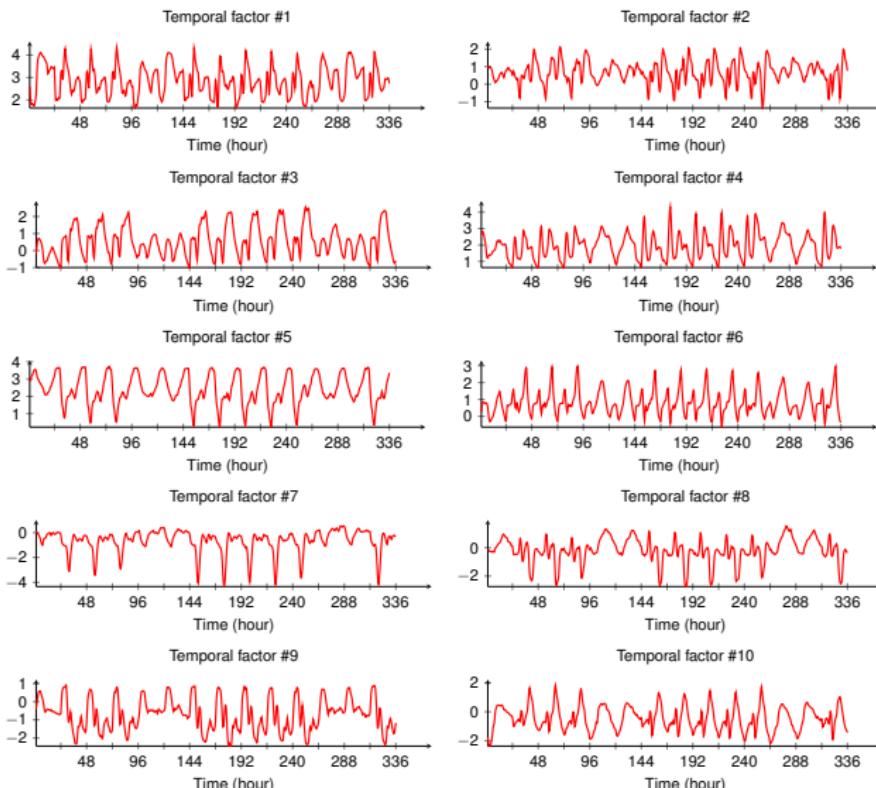
## Task E: Sparse Urban Traffic State Forecasting

NoTMF forecasting ( $\delta = 6$ )

- On the NYC Uber movement speed dataset



## Task E: Sparse Urban Traffic State Forecasting



# Conclusion

Sparse	
High-dimensional	
Multidimensional	
Noises & outliers	
Nonstationary	

## Low-rank framework:

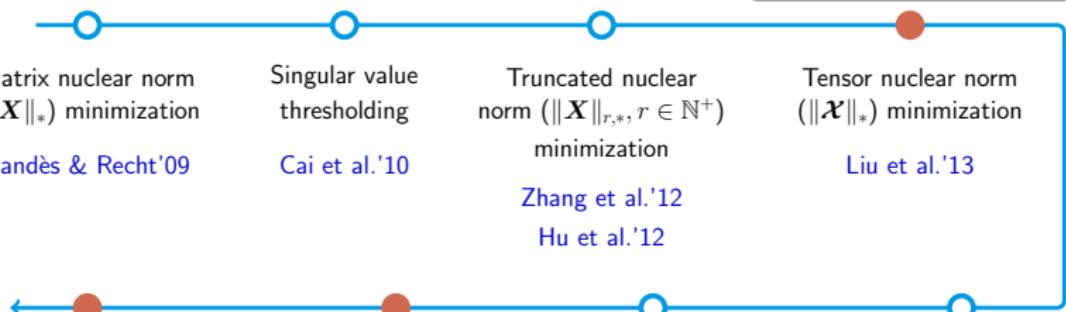
- NoTMF: matrix factorization
- LATC: low-rank tensor completion
- LCR: circulant matrix nuclear norm minimization
- HTF: tensor factorization

⇒ **Temporal modeling:**

- NoTMF: seasonal differenced vector autoregression
- LATC: univariate autoregression
- LCR: temporal smoothing
- HTF: automatic temporal modeling with Hankel tensor

# Highlights & Contributions

(Ours) LATC:  
 ✓ Temporal autoregression



(Ours) LCR:  
 ✓ Local trend modeling  
 ✓ An FFT implementation

(Ours) HTF:  
 ✓ Memory-efficient  
 ✓ Conv. para.

## Collaborators



Dr. HanQin Cai



Xiaoxu Chen



Dr. Zhanhong Cheng



Chengyuan Zhang



Dr. Xi-Le Zhao

# References

A short list:

- (Candès & Recht'09) "Exact matrix completion via convex optimization." Foundations of Computational Mathematics. 2009, 9(6): 717-772.
- (Cai et al.'10) "A singular value thresholding algorithm for matrix completion." SIAM Journal on optimization. 2010, 20(4): 1956-1982.
- (Zhang et al.'12) "Matrix completion by truncated nuclear norm regularization." IEEE Conference on computer vision and pattern recognition. 2012.
- (Hu et al.'12) "Fast and accurate matrix completion via truncated nuclear norm regularization." IEEE transactions on pattern analysis and machine intelligence. 2012, 35(9): 2117-2130.
- (Lu et al.'14) "Generalized nonconvex nonsmooth low-rank minimization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
- (Lu et al.'19) "Tensor robust principal component analysis with a new tensor nuclear norm." IEEE transactions on pattern analysis and machine intelligence. 2019, 42(4): 925-938.
- (Gultekin & Paisley'18) "Online forecasting matrix factorization." IEEE Transactions on Signal Processing. 2018, 67(5): 1223-1236.
- (Yokota et al.'18) "Missing slice recovery for tensors using a low-rank model in embedded space." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- (Cai et al.'21) "Accelerated structured alternating projections for robust spectrally sparse signal recovery." IEEE Transactions on Signal Processing. 2021, 69: 809-821.
- (Liu'22) "Time series forecasting via learning convolutionally low-rank models." IEEE Transactions on Information Theory. 2022, 68(5): 3362-3380.
- (Liu & Zhang'23) "Recovery of future data via convolution nuclear norm minimization." IEEE Transactions on Information Theory. 2023, 69(1): 650-665.



POLYTECHNIQUE  
MONTRÉAL

UNIVERSITÉ  
D'INGÉNIERIE



# Thanks for your attention!

Any Questions?

<https://xinychen.github.io/papers/thesis.pdf>

## About me:

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

## Supplementary Material

- Another way to tune models' hyperparameters in the imputation task:  
[https://xinychen.github.io/papers/model\\_settings\\_v1.pdf](https://xinychen.github.io/papers/model_settings_v1.pdf)
- NoTMF slides: <https://xinychen.github.io/slides/notmf.pdf>
- LCR slides: <https://xinychen.github.io/slides/LCR.pdf>
- MF & TF slides: [https://xinychen.github.io/slides/MF\\_TF\\_SFR.pdf](https://xinychen.github.io/slides/MF_TF_SFR.pdf)
- transdim project: <https://github.com/xinychen/transdim>

## Flipping Operation in LCR

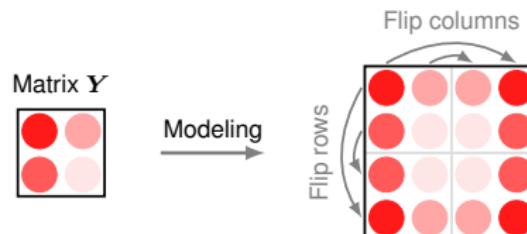
Circulant operation: Start and end data points are connected!

- On  $\mathbf{y} \in \mathbb{R}^T$ , we have

$$\mathbf{y}_{\text{new}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{J}_T \mathbf{x} \end{bmatrix} = (y_1, \dots, y_T, y_T, \dots, y_1)^\top \in \mathbb{R}^{2T}$$

w/ exchange matrix  $\mathbf{J}_T \in \mathbb{R}^{T \times T}$ .

- On  $\mathbf{Y} \in \mathbb{R}^{N \times T}$ , we have



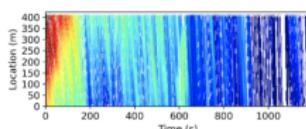
flipping  $\mathbf{Y}$  along rows and columns.

**[Significance]** Preventing LCR-2D from misleading values on the border rows and columns.

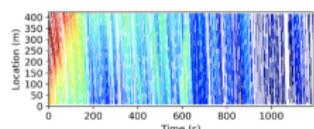
# Flipping Operation in LCR

Speed field reconstruction:

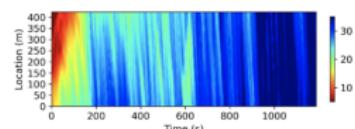
- LCR-2D on 70%-masked trajectories of the HighD dataset.
- Length: 426 meters; time duration: 1,190 seconds.



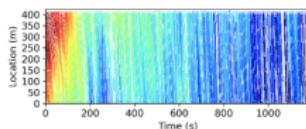
(a) Original speed field (lane #1).



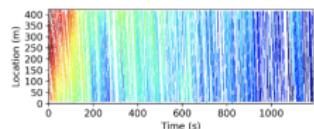
(b) Incomplete speed field (lane #1).



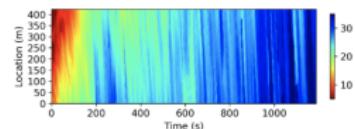
(c) Reconstructed speed field (lane #1).



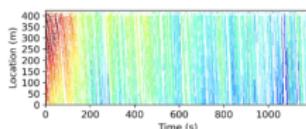
(d) Original speed field (lane #2).



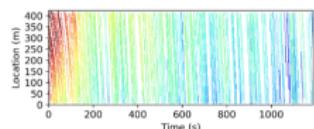
(e) Incomplete speed field (lane #2).



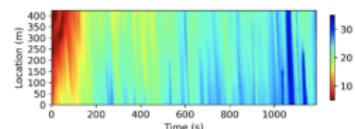
(f) Reconstructed speed field (lane #2).



(g) Original speed field (lane #3).



(h) Incomplete speed field (lane #3).



(i) Reconstructed speed field (lane #3).