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

**Xinyu Chen**

Ph.D., University of Montreal, Canada

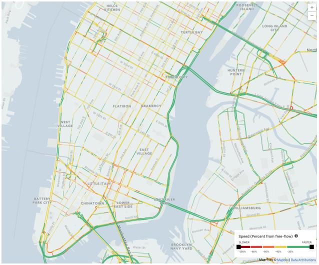
December 28, 2023

# Outline

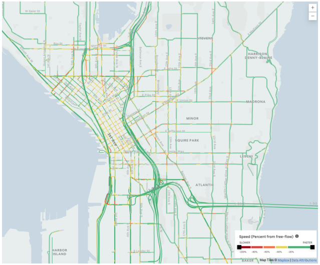
1. **Background**
   - Urban movement data
   - Problem formulation
2. **Literature Review**
   - Tensor factorization
   - Matrix/Tensor completion
3. **Nonstationary Temporal Matrix Factorization (NoTMF)**
   - Model description
   - Sparse urban traffic state forecasting
4. **Laplacian Convolutional Representation (LCR)**
   - Model description
   - Univariate traffic time series imputation
   - Large-scale traffic data imputation
5. **Hankel Tensor Factorization (HTF)**
   - Model description
   - Extreme missing traffic data imputation
6. **Conclusion**

# Urban Movement Data

- Uber (hourly) movement speed data




NYC movement




Seattle movement

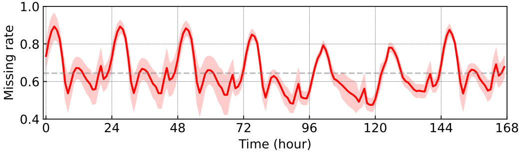
- {road segment, time step (hour), average speed}
- $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ with $N$ spatial locations $\times$ $T$ time steps
- Computing hourly speed: Road segments have 5+ unique trips.

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

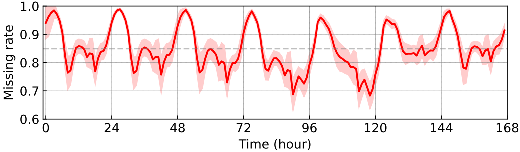
# Urban Movement Data

**High-dimensional & sparse**

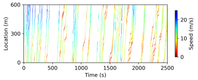
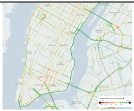
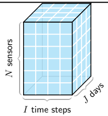
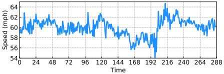
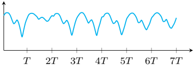
- **NYC** movement speed data (2019)
  - **98,210** road segments & 8,760 time steps (hours)
  - Overall missing rate: **64.43%**




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



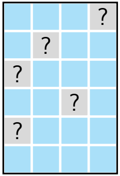
# Spatiotemporal Traffic Data

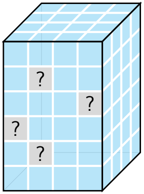
| | |
|---|---|
| Sparse |  |
| High-dimensional |  |
| Multidimensional |  |
| Noises & outliers |  |
| Nonstationary |  |

Traffic data show complicated spatiotemporal patterns and correlations.

# Problem Formulation

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





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

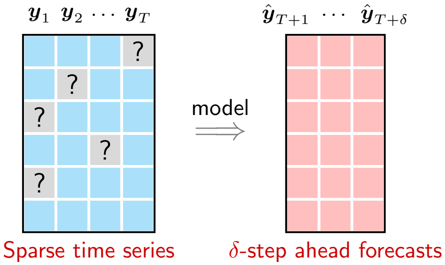
$$\underbrace{\mathcal{P}_{\Omega}(\boldsymbol{Y})}_{\text{Partially observed}} \xrightarrow{\text{Estimate}} \underbrace{\mathcal{P}_{\Omega}^{\perp}(\boldsymbol{Y})}_{\text{Unobserved}}$$

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 $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ consisting of time series $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T \in \mathbb{R}^N$, forecast data points $\boldsymbol{y}_{T+\delta}, \delta \in \mathbb{N}^+$.




Modeling process:

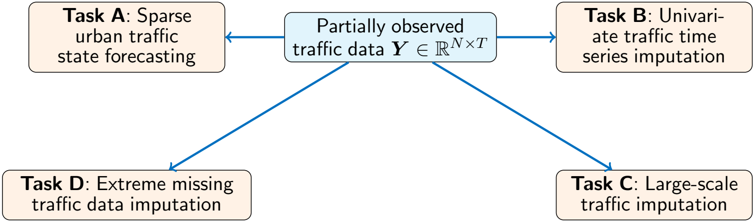
- How to characterize time series dynamics in high-dimensional and sparse traffic data?

Real-world applications:

- Forecasting urban traffic states with sparse data.

# Tasks

Focus: **spatiotemporal traffic data imputation and forecasting**.



# Tensor Factorization

- Revisit tensor factorization

**1927** — Higher-Order SVD

Frank Lauren Hitchcock

**1960s** — Tucker Decomposition

Ledyard R. Tucker

**1970** — CP Decomposition

J. Douglas Carroll
Jih-Jie Chang
Richard A. Harshman

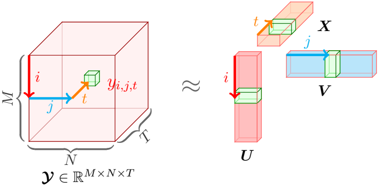
**2009** — Tensor Decompositions and Applications

Tamara G. Kolda

**2011** — Tensor-Train Decomposition

Ivan Oseledets

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




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

# Matrix/Tensor Completion

**Matrix nuclear norm ($\|\boldsymbol{X}\|_*$) minimization**

Candès & Recht'09

**Singular value thresholding**

Cai et al.'10

**Truncated nuclear norm ($\|\boldsymbol{X}\|_{r,*}, r \in \mathbb{N}^+$) minimization**

Zhang et al.'12
Hu et al.'12

**Tensor nuclear norm ($\|\boldsymbol{\mathcal{X}}\|_*$) minimization**

Liu et al.'13

**Generalized nonconvex nonsmooth low-rank minimization**

Lu et al.'14

**Tensor nuclear norm minimization with linear transform**

Lu et al.'19

**Low-rank Hankel matrix/tensor ($\mathcal{H}_\tau(\cdot)$) completion**

Yokota et al.'18
Sedighin et al.'20
Cai et al.'21
Yamamoto et al.'22

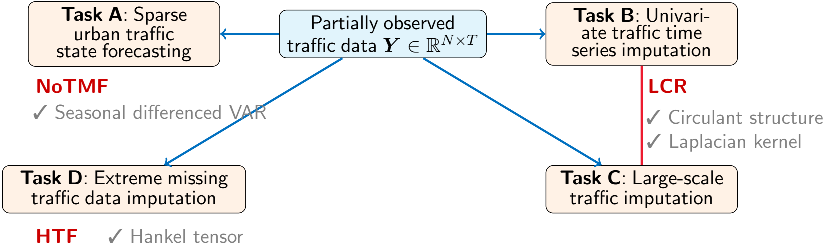
**Circulant/Convolution nuclear norm ($\|\mathcal{C}(\boldsymbol{X})\|_*$ or $\|\mathcal{C}_{\tilde{\tau}}(\boldsymbol{X})\|_*$) minimization**

Liu'22
Liu & Zhang'23

# Overview

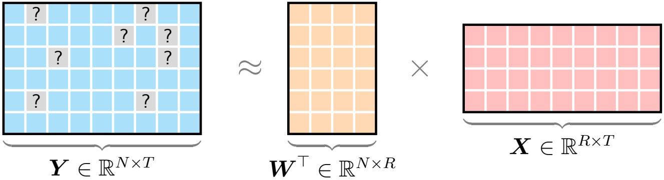
**Machine learning framework**

- Matrix and tensor methods: Learning from sparse data.
- Temporal modeling: Building and reinforcing temporal dependencies for time series.



# Matrix Factorization

A simple approach to reconstruct missing values.




**MF (Koren et al.'09)**

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

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

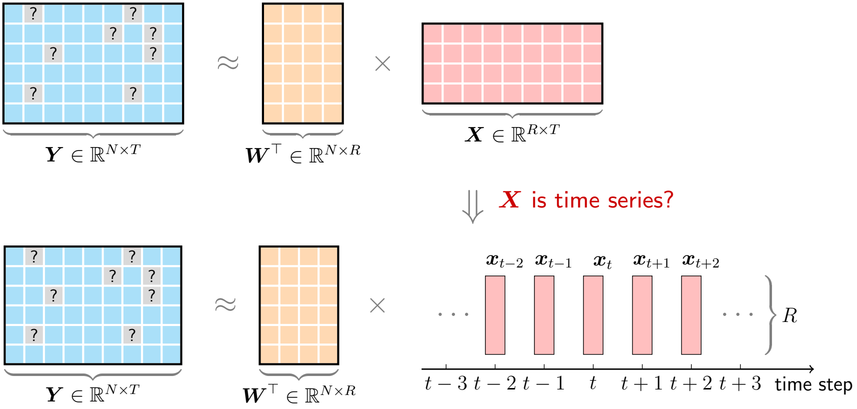
on data $\boldsymbol{Y}$ w/ observed index set $\Omega$.

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

How to build temporal correlations on MF?

# Temporal Matrix Factorization

Vector autoregression (VAR) on the temporal factor matrix.




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

# Temporal Matrix Factorization

Vector autoregression (VAR) on the temporal factor matrix.

**MF (Koren et al.'09)**

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

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

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

$+$

**$d$th-order VAR**

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

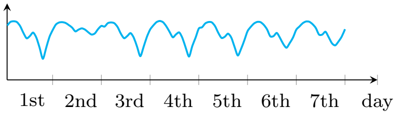
w/ coefficients $\{\boldsymbol{A}_k\}$.

$\Downarrow$ Yu et al.'16, Chen & Sun'21

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

# 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):

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

- **(Ours)** Optimization problem:

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

# Nonstationary Temporal Matrix Factorization

## Rewrite NoTMF

- Optimization problem:[1]

$$\min_{\boldsymbol{W},\boldsymbol{X},\boldsymbol{A}} \; \frac{1}{2}\underbrace{\|\mathcal{P}_{\Omega}(\boldsymbol{Y}-\boldsymbol{W}^\top\boldsymbol{X})\|_F^2}_{\text{MF on data } \boldsymbol{Y}}+\frac{\rho}{2}\underbrace{(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{X}\|_F^2)}_{\text{Regularization}}$$
$$+\frac{\gamma}{2}\underbrace{\|\boldsymbol{X}\boldsymbol{\Psi}_0^\top-\boldsymbol{A}(\boldsymbol{I}_d\otimes\boldsymbol{X})\boldsymbol{\Psi}^\top\|_F^2}_{\text{VAR on } \boldsymbol{X}}$$

  where $\boldsymbol{\Psi}_0\in\mathbb{R}^{(T-d-m)\times T}$, $\boldsymbol{\Psi}\in\mathbb{R}^{(T-d-m)\times(dT)}$ are temporal operators.

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

$$\begin{cases} \text{Spatial factors} & \boldsymbol{W}:=\{\boldsymbol{W}\mid\frac{\partial f}{\partial\boldsymbol{W}}=\boldsymbol{0}\} & \text{(least squares)} \\ \text{Temporal factors} & \boldsymbol{X}:=\{\boldsymbol{X}\mid\frac{\partial f}{\partial\boldsymbol{X}}=\boldsymbol{0}\} & \text{(conjugate gradient)} \\ \text{VAR coefficients} & \boldsymbol{A}:=\{\boldsymbol{A}\mid\frac{\partial f}{\partial\boldsymbol{A}}=\boldsymbol{0}\} & \text{(least squares)} \end{cases}$$

[1] $\boldsymbol{A}\triangleq\begin{bmatrix}\boldsymbol{A}_1 & \cdots & \boldsymbol{A}_d\end{bmatrix}\in\mathbb{R}^{R\times(dR)}$ (coefficient matrix).

# Nonstationary Temporal Matrix Factorization

Optimize $\boldsymbol{X}$?

- Solve the generalized Sylvester equation (i.e., $\frac{\partial f}{\partial \boldsymbol{X}}=\boldsymbol{0}$):

$$\underbrace{\boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{W}^\top\boldsymbol{X})+\rho\boldsymbol{X}+\gamma\sum_{k=0}^{d}\boldsymbol{A}_k^\top\left(\sum_{h=0}^{d}\boldsymbol{A}_h\boldsymbol{X}\boldsymbol{\Psi}_h^\top\right)\boldsymbol{\Psi}_k}_{\text{Left-hand side:}\quad \mathcal{L}_x(\boldsymbol{X})\triangleq\text{vec}(\cdot)}=\boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{Y})$$

- Conjugate gradient for estimating $\boldsymbol{X}$:
  - Initialize $\boldsymbol{x}$ as $\boldsymbol{x}_0$, and compute the residual $r_0:=\text{vec}(\boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{Y}))-\mathcal{L}_x(\boldsymbol{X}_0)$. Let $\boldsymbol{q}_0:=\boldsymbol{r}_0$.
  - Repeat:
    - Convert $\boldsymbol{q}_\ell$ into matrix $\boldsymbol{Q}_\ell$.
    - Compute $\alpha_\ell:=\frac{\boldsymbol{r}_\ell^\top\boldsymbol{r}_\ell}{\boldsymbol{q}_\ell^\top\mathcal{L}_x(\boldsymbol{Q}_\ell)}$, and update
    $$\begin{cases}\boldsymbol{x}_{\ell+1}:=\boldsymbol{x}_\ell+\alpha_\ell\boldsymbol{q}_\ell\\ \boldsymbol{r}_{\ell+1}:=\boldsymbol{r}_\ell-\alpha_\ell\mathcal{L}_x(\boldsymbol{Q}_\ell)\end{cases}.$$
    - Compute $\beta_\ell:=\frac{\boldsymbol{r}_{\ell+1}^\top\boldsymbol{r}_{\ell+1}}{\boldsymbol{r}_\ell^\top\boldsymbol{r}_\ell}$, and update $\boldsymbol{q}_{\ell+1}:=\boldsymbol{r}_{\ell+1}+\beta_\ell\boldsymbol{q}_\ell$.
    - Update $\ell:=\ell+1$.

# Nonstationary Temporal Matrix Factorization

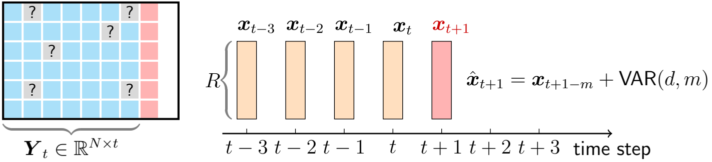
## NoTMF forecasting?

**Implementation**

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

✓ Sparse input $\boldsymbol{Y}_t$

✓ Forecast in latent spaces



# Nonstationary Temporal Matrix Factorization

## NoTMF forecasting on streaming data?

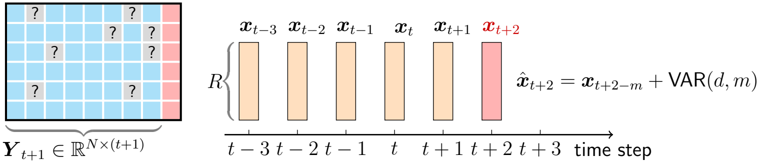
- Online forecasting (Gultekin & Paisley'18):
  - Fix the spatial factor matrix $\boldsymbol{W}$
  - Use input data $\boldsymbol{Y}_{t+1}$ to update the temporal factor matrix $\boldsymbol{X}$ and the coefficient matrix $\boldsymbol{A}$

**Implementation**

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

✓ Sparse input $\boldsymbol{Y}_{t+1}$

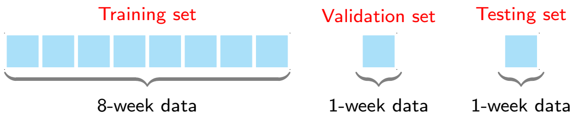
✓ Forecast in latent spaces



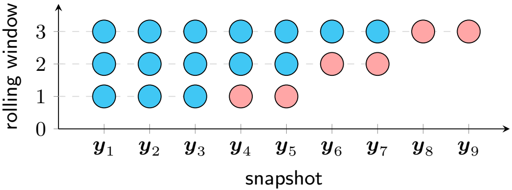
# Sparse Urban Traffic State Forecasting

## NoTMF forecasting

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




  - Weight parameter $\gamma \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$
  - Weight parameter $\rho = \{10^{-1}\gamma, 5 \times 10^{-1}\gamma, \gamma, 5\gamma, 10\gamma\}$
- Rolling forecasting illustration ($\delta = 2$):



# Sparse Urban Traffic State Forecasting

## NoTMF vs. baseline (in MAPE/RMSE)

- On the NYC Uber movement speed dataset

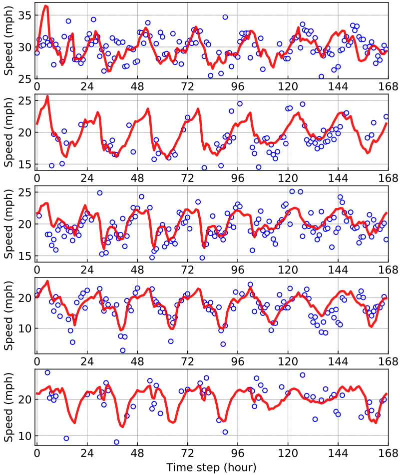
| $\delta$ | $d$ | NoTMF ($m = 24$) | NoTMF ($m = 168$) | NoTMF-1st ($m = 168$) | TRMF | BTMF | BTRMF |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 13.63/2.88 | 13.53/2.86 | **13.45**/**2.85** | 14.50/3.12 | 14.94/3.13 | 15.93/3.33 |
| | 2 | 13.47/**2.84** | **13.41**/**2.84** | 13.42/**2.84** | 14.14/3.05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

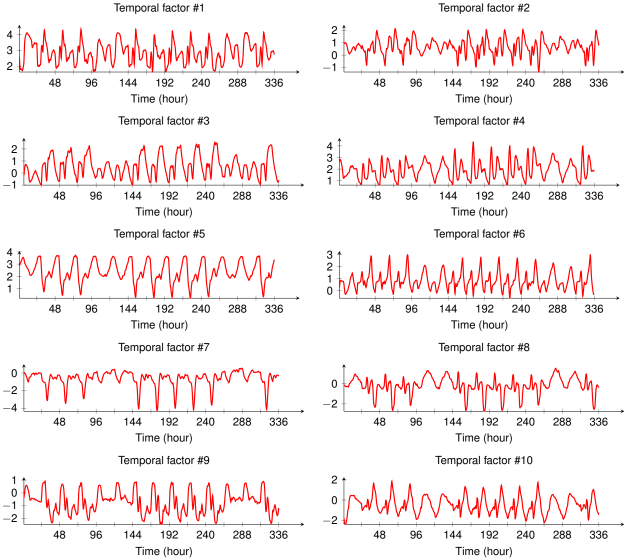
# Sparse Urban Traffic State Forecasting

**NoTMF forecasting** $(\delta = 6)$

- On the NYC Uber movement speed dataset



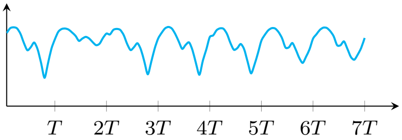
# Sparse Urban Traffic State Forecasting



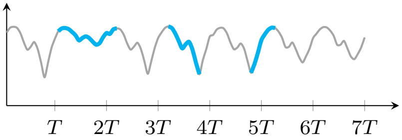
# Laplacian Convolutional Representation

**Motivation: Time series imputation**

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




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

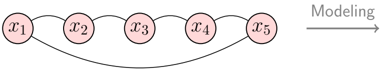



How to characterize both global and local trends in sparse time series?

# Laplacian Convolutional Representation

## Local trend modeling

- Intuition of (circulant) Laplacian matrix




Undirected and circulant graph

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

(Circulant) Laplacian matrix

- Define Laplacian kernel:

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

$$\Downarrow$$

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

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

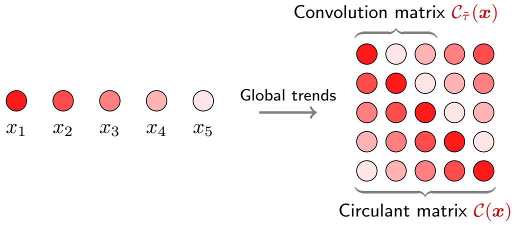
- (Laplacian) Temporal regularization:

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

Reformulate temporal regularization with circular convolution.

# Laplacian Convolutional Representation

**Global trend modeling**: Circulant matrix $\mathcal{C}(\boldsymbol{x})$ vs. convolution matrix $\mathcal{C}_{\tilde{\tau}}(\boldsymbol{x})$




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

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

Estimating $\boldsymbol{x}$:

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

on data $\boldsymbol{y}$ w/ observed index set $\Omega$.

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

Estimating $\boldsymbol{x}$:

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

on data $\boldsymbol{y}$ w/ observed index set $\Omega$.

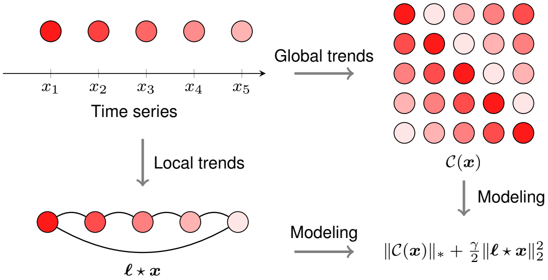
# Laplacian Convolutional Representation

## Laplacian Convolutional Representation (LCR)

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

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

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



## Laplacian Convolutional Representation

- Augmented Lagrangian function:[2]

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

- The ADMM scheme:

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

- Optimize $\boldsymbol{x}$?

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

Nuclear norm minimization $\Rightarrow$ **$\ell_1$-norm minimization with FFT**

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

# Laplacian Convolutional Representation

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

$$\boldsymbol{x} := \arg\min_{\boldsymbol{x}} \ \|\mathcal{C}(\boldsymbol{x})\|_* + \frac{\gamma}{2}\|\boldsymbol{\ell}\star\boldsymbol{x}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{w}/\lambda\|_2^2$$

$$\Longrightarrow \hat{\boldsymbol{x}} := \arg\min_{\hat{\boldsymbol{x}}} \ \|\hat{\boldsymbol{x}}\|_1 + \frac{\gamma}{2T}\|\hat{\boldsymbol{\ell}}\circ\hat{\boldsymbol{x}}\|_2^2 + \frac{\lambda}{2T}\|\hat{\boldsymbol{x}}-\hat{\boldsymbol{z}}+\hat{\boldsymbol{w}}/\lambda\|_2^2$$

where we introduce $\{\hat{\boldsymbol{\ell}},\hat{\boldsymbol{x}},\hat{\boldsymbol{z}},\hat{\boldsymbol{w}}\}\triangleq\mathcal{F}\{\boldsymbol{\ell},\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}\}$ (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:

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

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

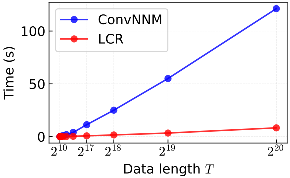
$$\hat{x}_t := \frac{\hat{h}_t}{|\hat{h}_t|}\cdot\max\{0,|\hat{h}_t|-1/\delta\}, t=1,\ldots,T.$$

# Laplacian Convolutional Representation

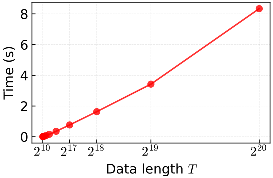
## Empirical time complexity

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

- Ours: **LCR**
  - An FFT implementation in $\mathcal{O}(T \log T)$
  - The logarithmic factor $\log T$ makes the FFT highly efficient
- Baseline: **ConvNNM**[3] (Liu'22, Liu & Zhang'23)
  - Convolution matrix $\mathcal{C}_{\tilde{\tau}}(\boldsymbol{y}) \in \mathbb{R}^{T \times \tilde{\tau}}$ with kernel size $\tilde{\tau} = 2^4$
  - Singular value thresholding in $\mathcal{O}(\tilde{\tau}^2 T)$



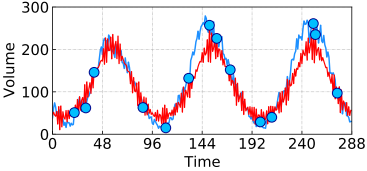

ConvNNM vs. LCR




LCR

[3] Convolution nuclear norm minimization.

# Univariate Traffic Time Series Imputation

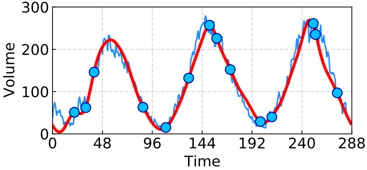



**CircNNM**:

$$\min_{\boldsymbol{x}} \ \underbrace{\|\mathcal{C}(\boldsymbol{x})\|_*}_{\text{global}}$$

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

$\Downarrow$ Plus temporal regularization

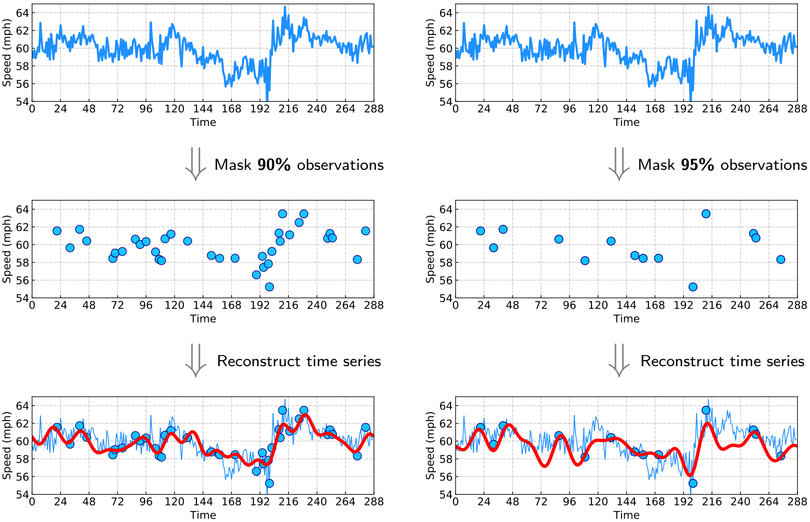



**LCR**:

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

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

# Univariate Traffic Time Series Imputation




LCR can reconstruct traffic time series from very sparse data.

# Large-Scale Traffic Data Imputation

**LCR vs. baseline** (in MAPE/RMSE)

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

| Model | Missing rate | | | |
|---|---|---|---|---|
| | 30% | 50% | 70% | 90% |
| LCR-2D | 1.50/**1.49** | 1.76/**1.69** | **2.07**/**2.06** | **3.19**/**3.05** |
| LCR$_N$ | **1.48**/1.50 | **1.73**/1.73 | **2.07**/2.12 | 3.24/3.22 |
| LCR | 1.50/**1.49** | 1.76/**1.69** | 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**

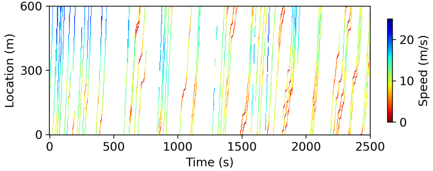
- LCR-2D $>$ CTNNM: The importance of temporal regularization.
- CTNNM $\geq$ CircNNM: Ciculant tensor is superior to circulant matrix.
- LCR $>$ LRMC/LRTC: The importance of global/local modeling.

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

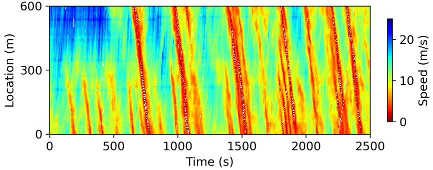
# Hankel Tensor Factorization

## Motivation: Spatiotemporal data reconstruction

- Sparse speed field reconstruction problem in vehicular traffic flow.




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



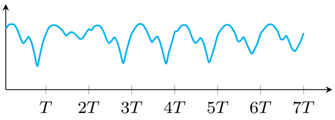

> How to characterize both spatial and temporal dependencies?

# Hankel Tensor Factorization

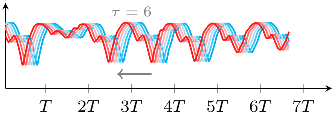
- Hankel matrix
  - Given $\boldsymbol{x} = (1,2,3,4,5)^\top$ and window length $\tau = 2$, we have

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

  - Automatic temporal modeling




Traffic time series



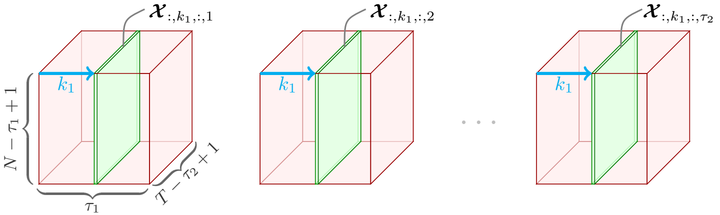

Hankel matrix

# Hankel Tensor Factorization

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

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

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

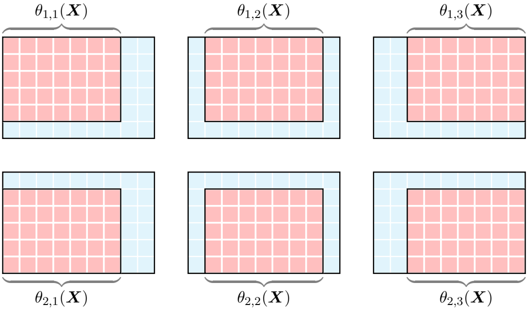
# Hankel Tensor Factorization

## Hankel indexing

- Sampling function for the Hankel tensor:

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

  referring to as the tensor slice with $k_1 \in \{1,\ldots,\tau_1\}$, $k_2 \in \{1,\ldots,\tau_2\}$.
- **[Importance]** Developing memory-efficient algorithms




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

# Hankel Tensor Factorization

**Ours:**

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

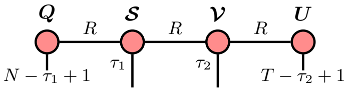
$$\theta_{k_1,k_2}(\boldsymbol{Y}) \approx (\boldsymbol{Q} \star_{\text{row}} \boldsymbol{s}_{k_1}^\top)(\boldsymbol{U} \star_{\text{row}} \boldsymbol{v}_{k_2}^\top)^\top$$

**Baselines:**

- Tensor-train decomposition:

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

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




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

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

# Hankel Tensor Factorization

## HTF (convolutional decomposition)

- Optimization problem:

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

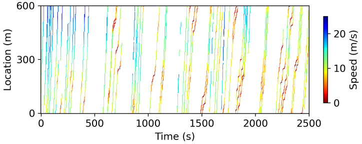
$$+\frac{\rho}{2}(\|\boldsymbol{Q}\|_F^2+\|\boldsymbol{S}\|_F^2+\|\boldsymbol{U}\|_F^2+\|\boldsymbol{V}\|_F^2)$$

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

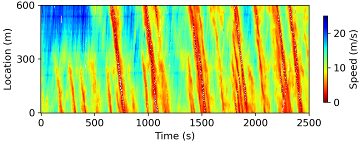
$$\begin{cases}\boldsymbol{Q}:=\{\boldsymbol{Q}\mid\frac{\partial f}{\partial\boldsymbol{Q}}=\boldsymbol{0}\} & \text{(conjugate gradient)}\\ \boldsymbol{s}_{k_1}:=\{\boldsymbol{s}_{k_1}\mid\frac{\partial f}{\partial\boldsymbol{s}_{k_1}}=\boldsymbol{0}\},\forall k_1 & \text{(conjugate gradient)}\\ \boldsymbol{U}:=\{\boldsymbol{U}\mid\frac{\partial f}{\partial\boldsymbol{U}}=\boldsymbol{0}\} & \text{(conjugate gradient)}\\ \boldsymbol{v}_{k_2}:=\{\boldsymbol{v}_{k_2}\mid\frac{\partial f}{\partial\boldsymbol{v}_{k_2}}=\boldsymbol{0}\},\forall k_2 & \text{(conjugate gradient)}\end{cases}$$

- Memory-efficient but still computationally costly!

# Extreme Missing Traffic Data Imputation

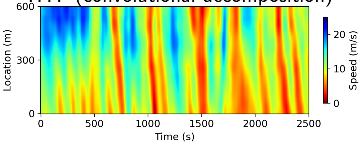



Sparse speed field



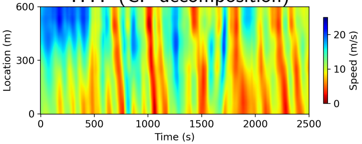

Ground truth speed field

HTF (convolutional decomposition)



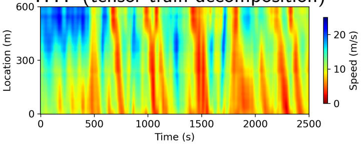

MAPE = **51.92%**

HTF (CP decomposition)




MAPE = 53.93%

HTF (tensor-train decomposition)




MAPE = 56.48%

# Extreme Missing Traffic Data Imputation

## HTF vs. baseline (in MAPE/RMSE)

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

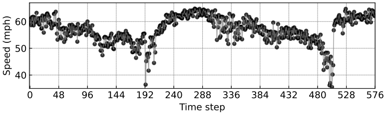
| Model | Random missing rate | | | |
|---|---|---|---|---|
| | 80% | 85% | 90% | 95% |
| HTF (`convolution`) | **6.21/3.88** | **6.51/4.06** | **6.98/4.30** | **8.02/4.84** |
| HTF (`tensor-train`) | 8.75/5.16 | 9.86/5.76 | 9.24/5.36 | 9.89/5.70 |
| HTF (`CP`) | 7.25/4.33 | 7.93/4.66 | 8.61/4.96 | 9.25/5.20 |
| 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-2D | 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**

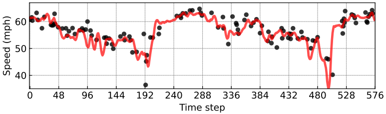
- Convolutional tensor decomposition outperforms both tensor-train and CP tensor decomposition.
- Our HTF model performs better than state-of-the-art baseline models.

# Extreme Missing Traffic Data Imputation

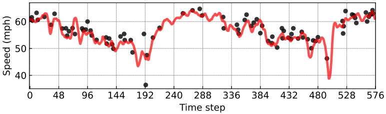
- Example: 1st time series within two days.



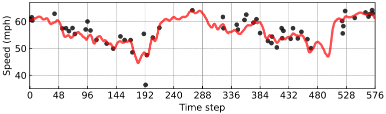

Original time series



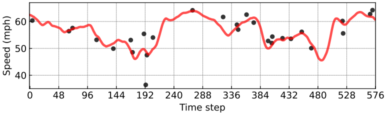

80% missing rate




85% missing rate

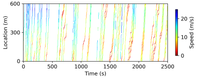
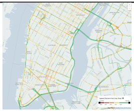
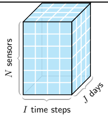
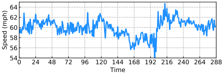
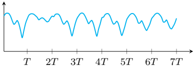



90% missing rate




95% missing rate

# Conclusion

| | |
|---|---|
| Sparse |   |
| High-dimensional |  |
| Multidimensional |   |
| Noises & outliers |   |
| Nonstationary |   |

$\Rightarrow$

**Low-rank framework:**

- NoTMF: matrix factorization
- LCR: circulant matrix nuclear norm minimization
- HTF: tensor factorization

**Temporal modeling:**

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

# Highlights & Contributions

Matrix nuclear norm ($\|\boldsymbol{X}\|_*$) minimization

Candès & Recht'09

Singular value thresholding

Cai et al.'10

Truncated nuclear norm ($\|\boldsymbol{X}\|_{r,*}, r \in \mathbb{N}^+$) minimization

Zhang et al.'12

Hu et al.'12

Tensor nuclear norm ($\|\boldsymbol{\mathcal{X}}\|_*$) minimization

Liu et al.'13

Generalized nonconvex nonsmooth low-rank minimization

Lu et al.'14

Tensor nuclear norm minimization with linear transform

Lu et al.'19

Low-rank Hankel matrix/tensor ($\mathcal{H}_\tau(\cdot)$) completion

Yokota et al.'18

Sedighin et al.'20

Cai et al.'21

Yamamoto et al.'22

(Ours) **HTF**:

- ✓ Memory-efficient
- ✓ Conv. para.

Circulant/Convolution nuclear norm ($\|\mathcal{C}(\boldsymbol{X})\|_*$ or $\|\mathcal{C}_{\tilde{\tau}}(\boldsymbol{X})\|_*$) minimization

Liu'22

Liu & Zhang'23

(Ours) **LCR**:

- ✓ Local trend modeling
- ✓ An FFT implementation

# 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.
- (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.
- (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.
- (Cai et al.'21) "Accelerated structured alternating projections for robust spectrally sparse signal recovery." IEEE Transactions on Signal Processing. 2021, 69: 809-821.
- (Chen & Sun'22) "Bayesian temporal factorization for multidimensional time series prediction." IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022, 44(9): 4659-4673.
- (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.

# Thanks for your attention!

## Any Questions?

Slides: https://xinychen.github.io/slides/sustech23.pdf

**About me**:

- Homepage: https://xinychen.github.io
- How to reach me: chenxy346@gmail.com