



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE



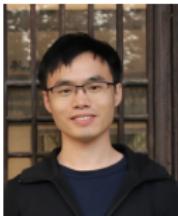
Matrix and Tensor Models for Spatiotemporal Traffic Data Imputation and Forecasting

Ph.D. Defense

Xinyu Chen

Polytechnique Montreal, Canada

December 11, 2023



Ph.D. Candidate
Xinyu Chen



Supervisor
Prof. Nicolas Saunier



Co-supervisor
Prof. Lijun Sun (McGill)

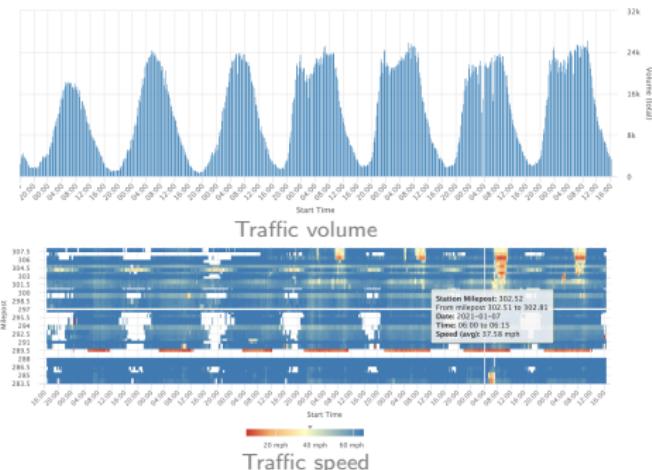
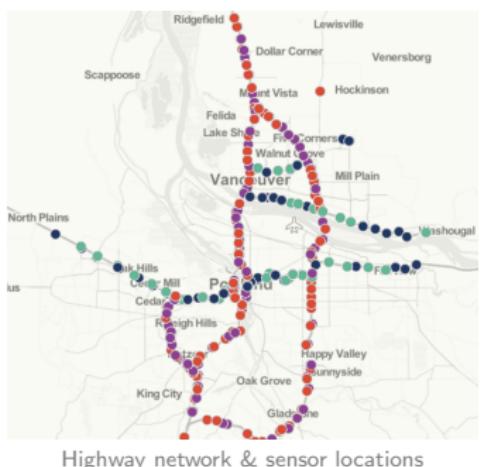
Outline

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

Multivariate Traffic Time Series

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

- Example: Portland highway traffic data¹.



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

¹<https://portal.its.pdx.edu/home>

Multiple Data Behaviors

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

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

In addition, spatiotemporal correlations are also very important.

Multiple Data Behaviors

Sparsity & high-dimensionality

- Uber (hourly) movement speed data²



NYC movement



Seattle movement

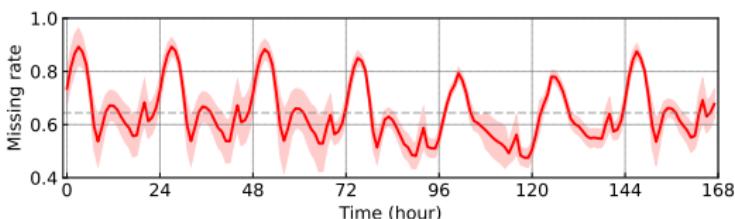
- The average speed on a given road segment for each hour of each day.
- Hourly speeds are computed when road segments have 5+ unique trips.
- **Issue:** insufficient sampling of ridesharing vehicles on the road network.

²<https://movement.uber.com/>

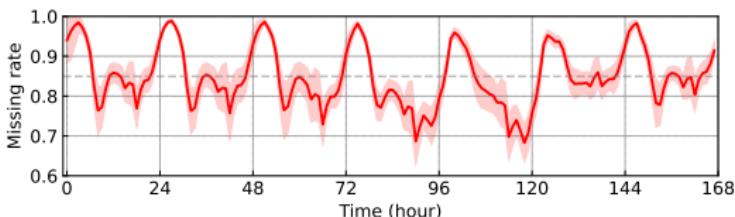
Multiple Data Behaviors

Sparsity & high-dimensionality

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

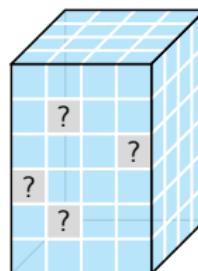
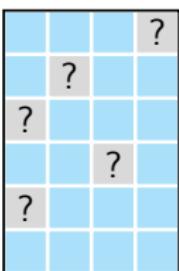


- **Seattle** movement speed data (2019)
 - 63,490 road segments & 8,760 time steps (hours)
 - Overall missing rate: 84.95%



Problem Formulation

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

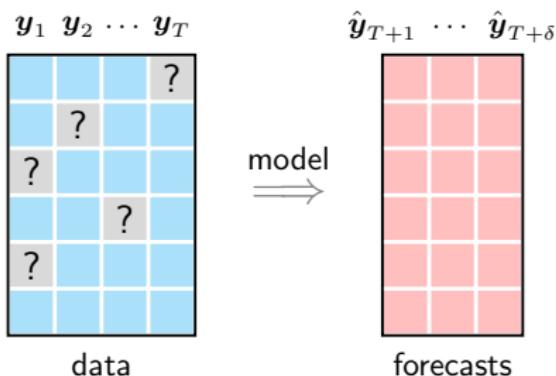


[Q]

- How to reconstruct missing values from observed data?
 - Matrix completion: From $\mathcal{P}_\Omega(\mathbf{Y})$ (observed) to $\mathcal{P}_\Omega^\perp(\mathbf{Y})$ (unobserved)
 - Tensor completion: From $\mathcal{P}_\Omega(\mathcal{Y})$ (observed) to $\mathcal{P}_\Omega^\perp(\mathcal{Y})$ (unobserved)
- How to make use of spatiotemporal correlations?
- How to make use of traffic time series dynamics?

Problem Formulation

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

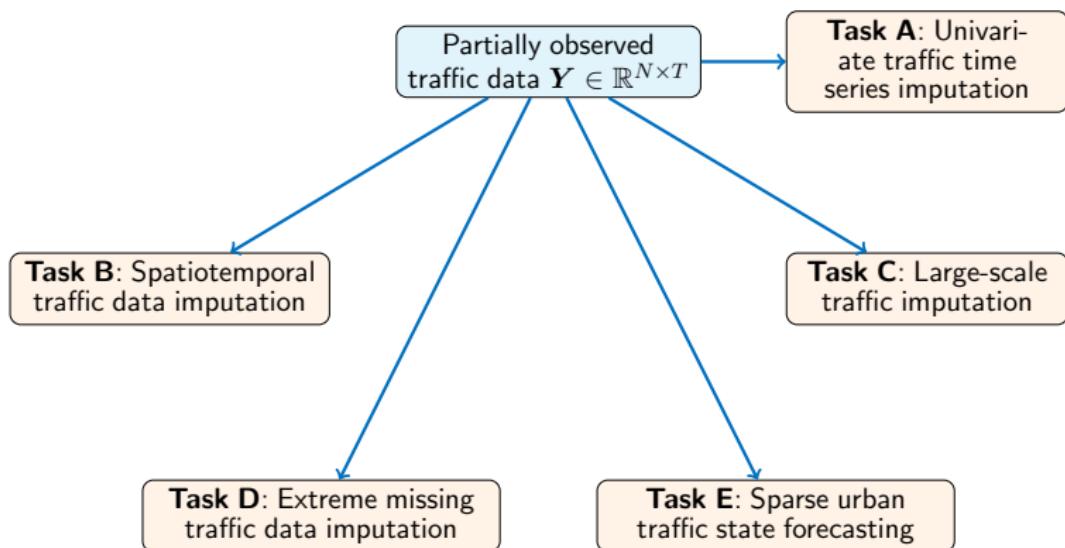


[Q]

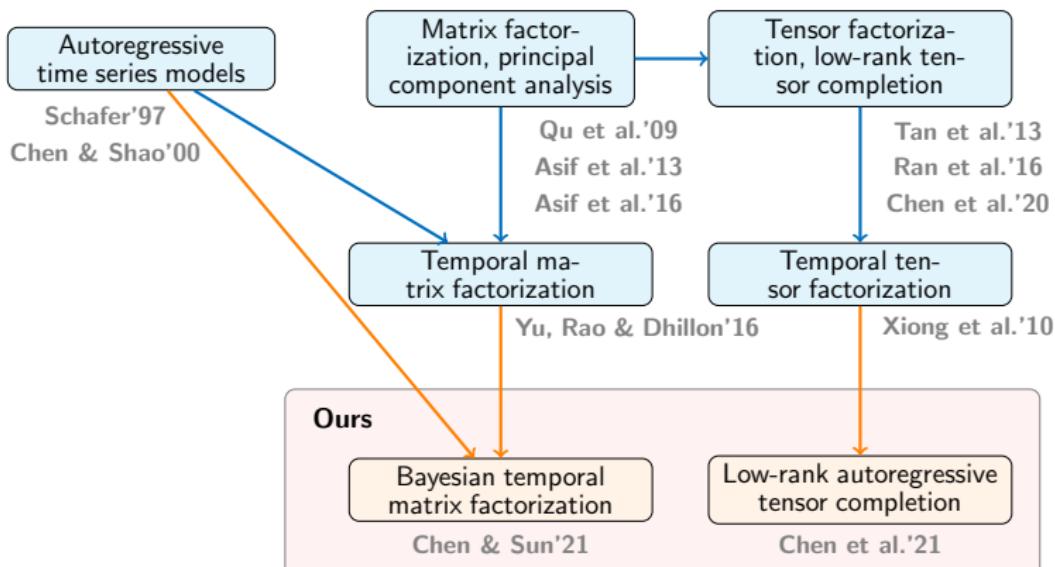
- How to learn from *high-dimensional* and *sparse* data?
- How to model *nonstationarity* in time series?
- How to perform forecasting on these time series?

Whole Picture

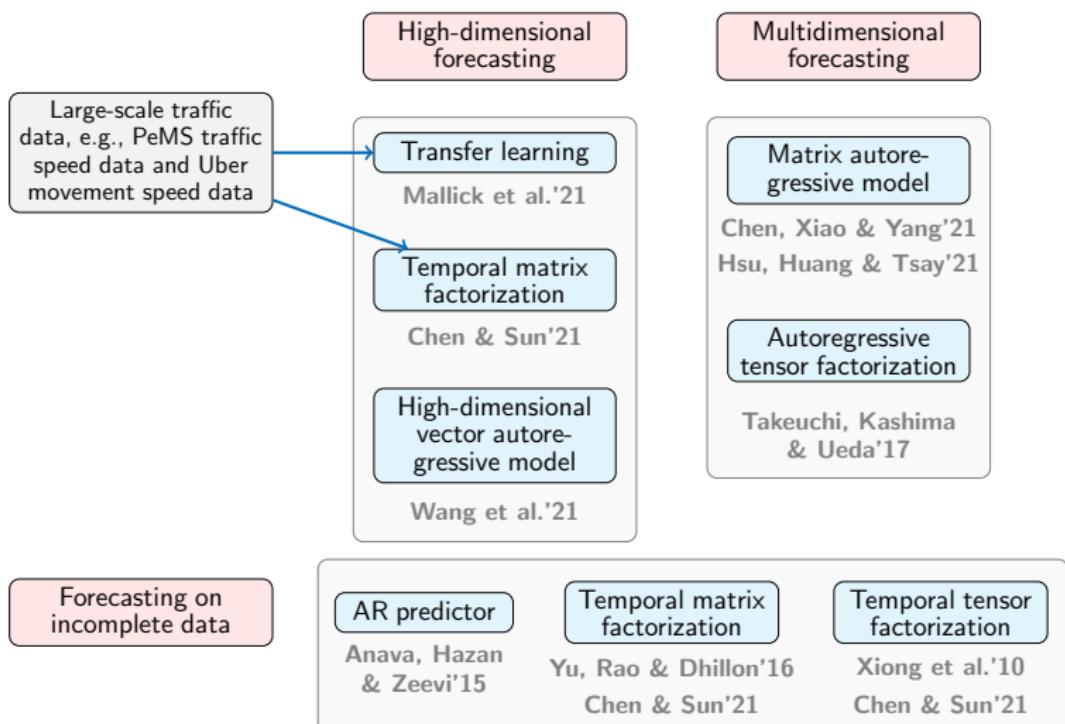
We are working on **spatiotemporal traffic data modeling**.



Spatiotemporal Traffic Data Imputation

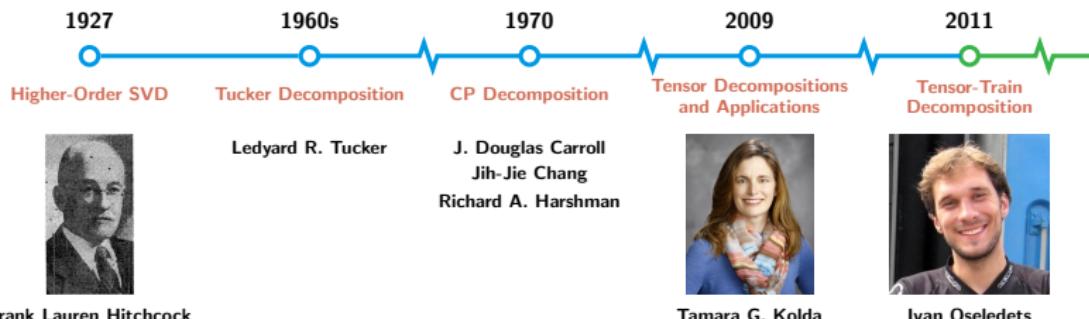


Spatiotemporal Traffic Forecasting



Tensor Factorization

- Revisit tensor factorization (TF)

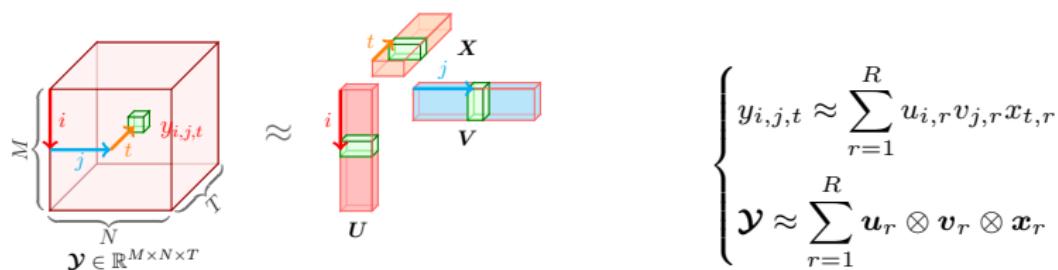


Frank Lauren Hitchcock

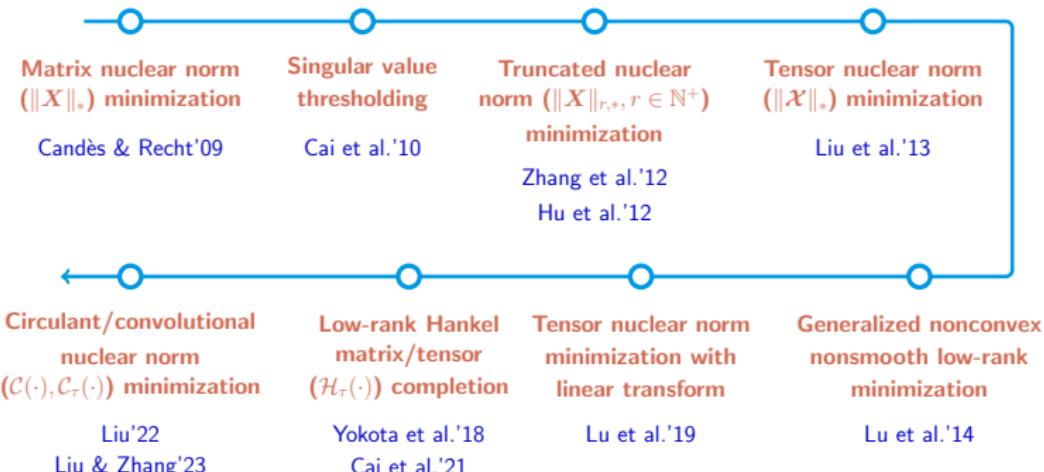
Tamara G. Kolda

Ivan Oseledets

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



Matrix/Tensor Completion

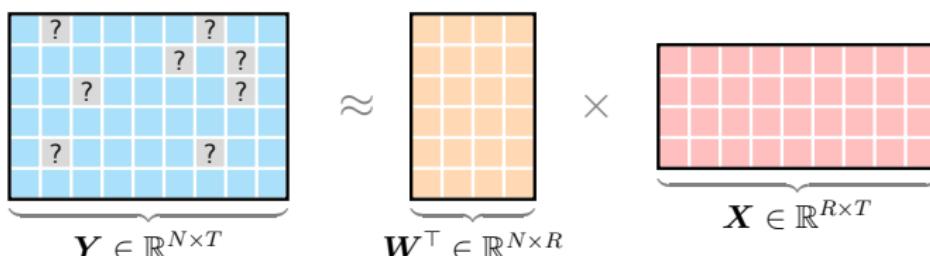


This research

- Integrate temporal modeling techniques (e.g., temporal smoothing and time series autoregression) into low-rank matrix and tensor methods
- Implement spatiotemporal traffic data imputation and forecasting on partially observed data

Matrix Factorization

A simple approach to reconstruct missing values.



MF (Koren et al.'09)

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

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

✓ Learn from sparse data

✗ Temporal correlations

✗ Time series forecasting

Temporal Matrix Factorization

Vector autoregression (VAR) on the temporal factor matrix.

$$\begin{matrix} \text{?} & & \text{?} \\ & \text{?} & \text{?} \\ \text{?} & & \text{?} \\ & \text{?} & \end{matrix} \underbrace{\quad}_{\mathbf{Y} \in \mathbb{R}^{N \times T}} \approx \begin{matrix} \text{?} & & \text{?} \\ & \text{?} & \text{?} \\ \text{?} & & \text{?} \\ & \text{?} & \end{matrix} \underbrace{\quad}_{\mathbf{W}^\top \in \mathbb{R}^{N \times R}} \times \begin{matrix} \text{?} & & \text{?} \\ & \text{?} & \text{?} \\ \text{?} & & \text{?} \\ & \text{?} & \end{matrix} \underbrace{\quad}_{\mathbf{X} \in \mathbb{R}^{R \times T}}$$

⇓ **X are time series?**

$$\begin{matrix} \text{?} & & \text{?} \\ & \text{?} & \text{?} \\ \text{?} & & \text{?} \\ & \text{?} & \end{matrix} \underbrace{\quad}_{\mathbf{Y} \in \mathbb{R}^{N \times T}} \approx \begin{matrix} \text{?} & & \text{?} \\ & \text{?} & \text{?} \\ \text{?} & & \text{?} \\ & \text{?} & \end{matrix} \underbrace{\quad}_{\mathbf{W}^\top \in \mathbb{R}^{N \times R}} \times \begin{matrix} \mathbf{x}_{t-2} & \mathbf{x}_{t-1} & \mathbf{x}_t & \mathbf{x}_{t+1} & \mathbf{x}_{t+2} \\ \dots & & & & \dots \end{matrix} \Bigg\}^R_{\substack{t-3 \\ t-2 \\ t-1 \\ t \\ t+1 \\ t+2 \\ t+3 \\ \text{time step}}}$$

Temporal Matrix Factorization

MF (Koren et al.'09)

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

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

dth-order VAR

$$\mathbf{x}_t = \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} + \boldsymbol{\epsilon}_t$$

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



Yu et al.'16
Chen & Sun'21

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

Nonstationary Temporal Matrix Factorization

Nonstationary temporal matrix factorization (NoTMF)

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

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

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

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

Nonstationary Temporal Matrix Factorization

Rewrite VAR in the form of matrix

Temporal operators

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

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

then

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

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

Nonstationary Temporal Matrix Factorization

Rewrite NoTMF:

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

Alternating minimization method:

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

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

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

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

This generalized Sylvester equation can be solved by **conjugate gradient**.

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

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

Nonstationary Temporal Matrix Factorization

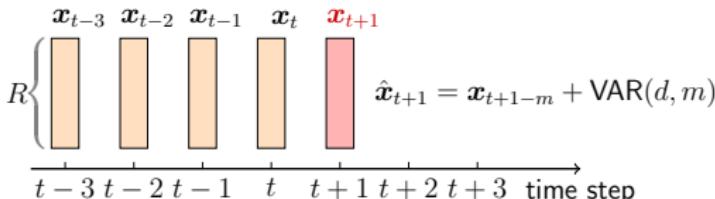
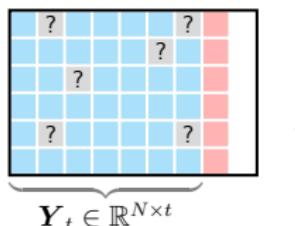
NoTMF forecasting on streaming data?

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

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
- ✓ Low-dimensional temporal factors
- ✓ Forecast in latent spaces



Nonstationary Temporal Matrix Factorization

NoTMF forecasting on streaming data?

- Online forecasting (Gultekin & Paisley'18): Fix \mathbf{W} and use \mathbf{Y}_{t+1} to update \mathbf{X}, \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}
- ✓ Fixed spatial factors \mathbf{W}
- ✓ Forecast in latent spaces

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

$$R \begin{cases} \mathbf{x}_{t-3} & \mathbf{x}_{t-2} & \mathbf{x}_{t-1} & \mathbf{x}_t & \mathbf{x}_{t+1} & \mathbf{x}_{t+2} \\ t-3 & t-2 & t-1 & t & t+1 & t+2 & t+3 \end{cases} \text{ time step}$$
$$\hat{\mathbf{x}}_{t+2} = \mathbf{x}_{t+2-m} + \text{VAR}(d, m)$$

Matrix/Tensor Completion

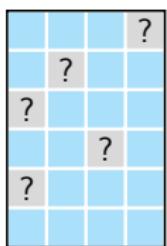
Cornerstone: Nuclear norm minimization in matrix/tensor completion

LRMC (Candès & Recht'09)

Estimating the matrix \mathbf{X} :

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

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



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

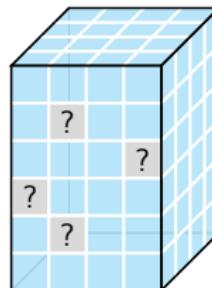
LRTC (Liu et al.'13)

Estimating the tensor \mathcal{X} :

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

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

vs.

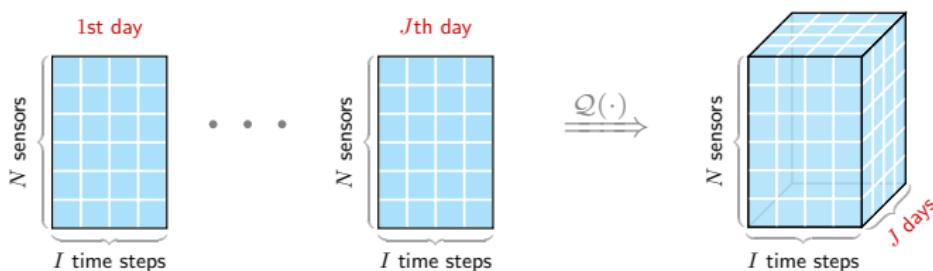


$$\mathcal{P}_\Omega(\mathcal{Y}) \in \mathbb{R}^{N \times I \times J}$$

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

Low-Rank Autoregressive Tensor Completion

- Introduce traffic tensors with day dimension⁴ (Tan et al.'13, Chen et al.'19, ...)



⁴There are $T = IJ$ time steps in total.

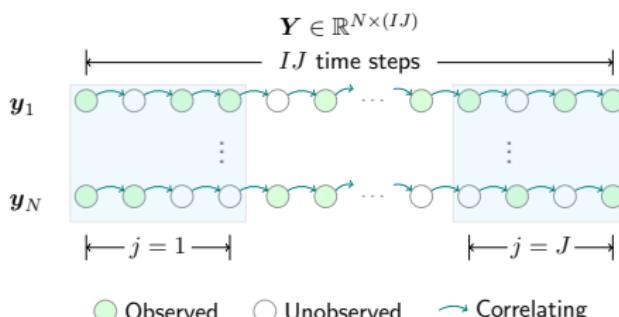
Low-Rank Autoregressive Tensor Completion

- Build temporal correlations with autoregression

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

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

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



Low-Rank Autoregressive Tensor Completion

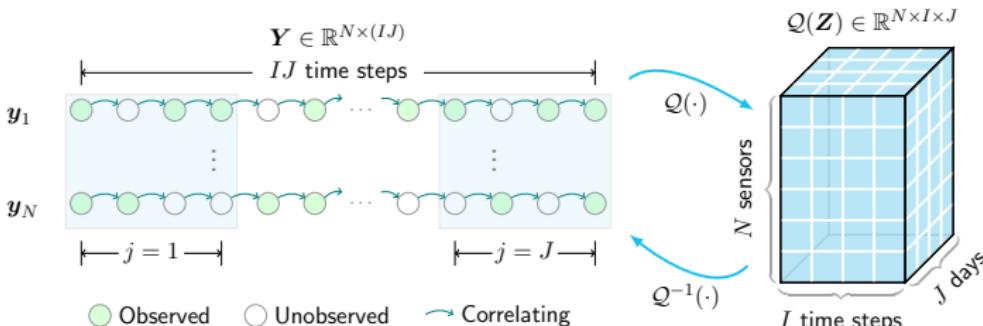
Low-Rank Autoregressive Tensor Completion (LATC)

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

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

Advantages:

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



Low-Rank Autoregressive Tensor Completion

- Optimization problem:

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

- Two subproblems:

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

Low-Rank Autoregressive Tensor Completion

How to solve the 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:

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

where $\mathbf{W} \in \mathbb{R}^{N \times I \times J}$ is the Lagrange multiplier, and $\langle \cdot, \cdot \rangle$ denotes the inner product. The indicator function is $\pi(\mathbf{Z}) = \begin{cases} 0, & \text{if } \mathcal{P}_\Omega(\mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{Y}), \\ +\infty, & \text{otherwise.} \end{cases}$

Implementation

Repeat

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



Implementation

Repeat

- Repeat
 - Compute \mathbf{X}
 - Compute \mathbf{Z}
 - Compute \mathbf{W}
- Compute \mathbf{A}

Low-Rank Autoregressive Tensor Completion

- Augmented Lagrangian function:

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

- The ADMM⁵ scheme:

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

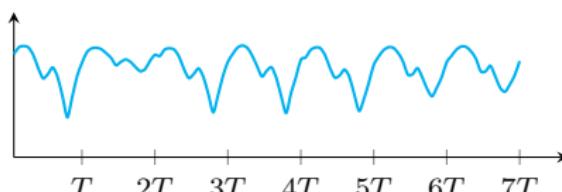
- ✓ Solution to \mathbf{X} : singular value thresholding
- ✓ Solution to \mathbf{Z} : conjugate gradient

⁵Alternating Direction Method of Multipliers.

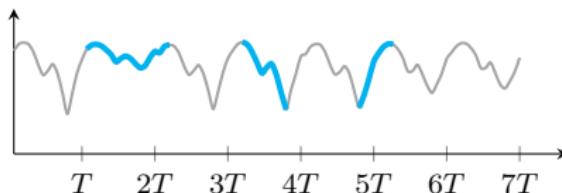
Laplacian Convolutional Representation

Motivation: Time series imputation

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



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

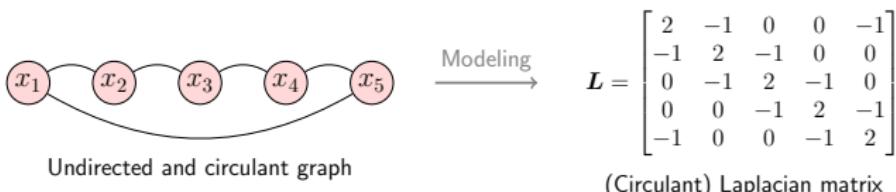


- [Question] How to characterize both global and local trends in sparse time series data?

Laplacian Convolutional Representation

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

- Intuition of (circulant) Laplacian matrix.



- Define Laplacian kernel:

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

↓

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

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

- (Laplacian) Temporal regularization:

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

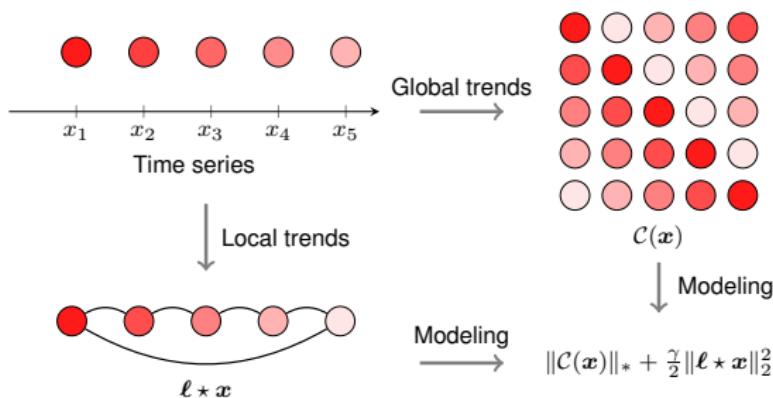
Laplacian Convolutional Representation

Laplacian Convolutional Representation (LCR)

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

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

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



Laplacian Convolutional Representation

- Augmented Lagrangian function:

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

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

- The ADMM scheme:

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

- Optimize \mathbf{x} ?

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

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

Laplacian Convolutional Representation

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

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

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

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

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

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

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

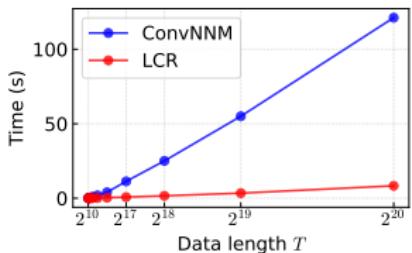
$$\hat{x}_t := \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - 1/\omega\}, t = 1, \dots, T.$$

Laplacian Convolutional Representation

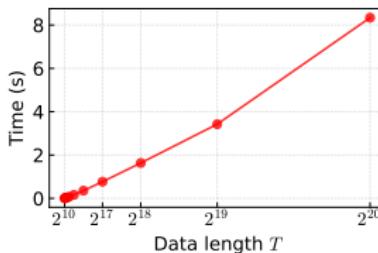
Empirical time complexity

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

- Ours: **LCR**
 - An FFT implementation in $\mathcal{O}(T \log T)$
 - The logarithmic factor $\log T$ makes the FFT highly efficient
- Baseline: **ConvNNM**⁶ ([Liu'22](#), [Liu & Zhang'23](#))
 - Convolution matrix $C_{\tilde{\tau}}(\mathbf{y}) \in \mathbb{R}^{T \times \tilde{\tau}}$ with kernel size $\tilde{\tau} \in \mathbb{N}^+$
 - Singular value thresholding in $\mathcal{O}(\tilde{\tau}^2 T)$



ConvNNM vs. LCR



LCR

⁶Convolution nuclear norm minimization.

Two-Dimensional LCR (LCR-2D)

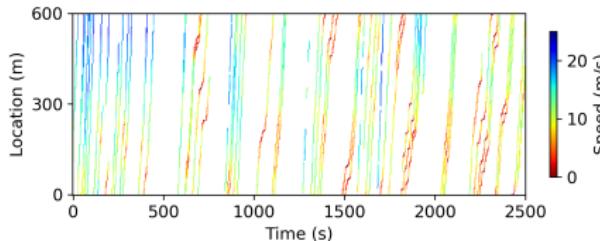
For any partially observed time series $\mathbf{Y} \in \mathbb{R}^{N \times T}$ with observed index set Ω , LCR can be formulated as follows,

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

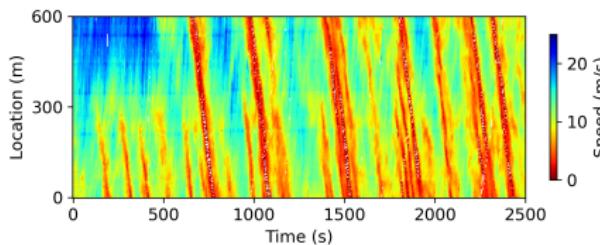
where $\mathcal{C} : \mathbb{R}^{N \times T} \rightarrow \mathbb{R}^{N \times N \times T \times T}$ denotes the circulant operator.

Motivation: Spatiotemporal data reconstruction

- Speed field reconstruction problem in vehicular traffic flow.



Reconstruct speed field from
5% sparse trajectories?



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

Hankel Tensor Factorization

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

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



Hankel matrix (Source: Twitter)

Hankel Tensor Factorization

- Hankel matrix

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

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

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

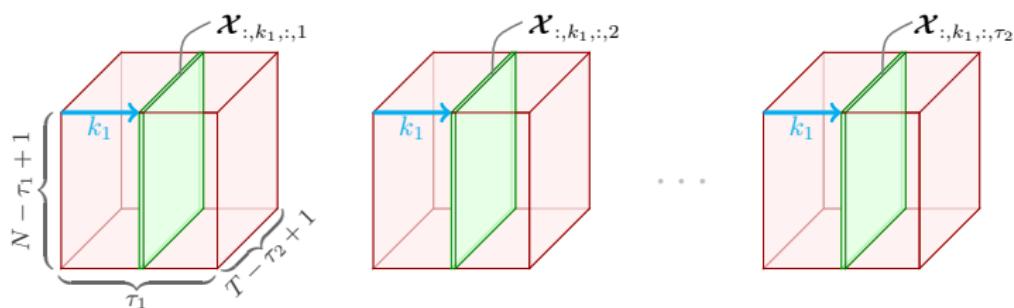
- Automatic temporal modeling.

Hankel Tensor Factorization

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

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

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



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

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

Hankel Tensor Factorization

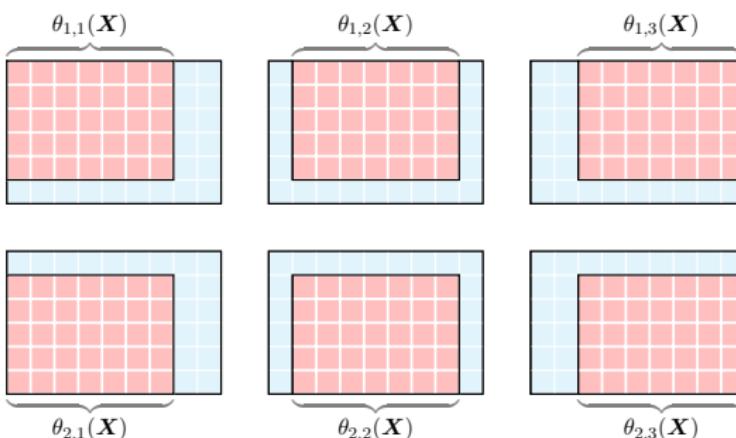
Hankel indexing:

- Sampling function for the Hankelization:

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

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

- [Importance] Developing memory-efficient algorithms.



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

Hankel Tensor Factorization

Ours:

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

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

Baselines:

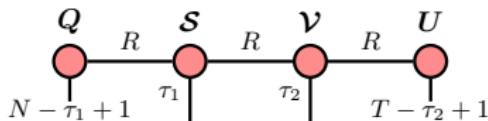
- CP tensor decomposition (Khatri-Rao product \odot):

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

- Tensor-train decomposition:

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

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



Hankel Tensor Factorization

HTF (convolutional decomposition)

- Optimization problem:

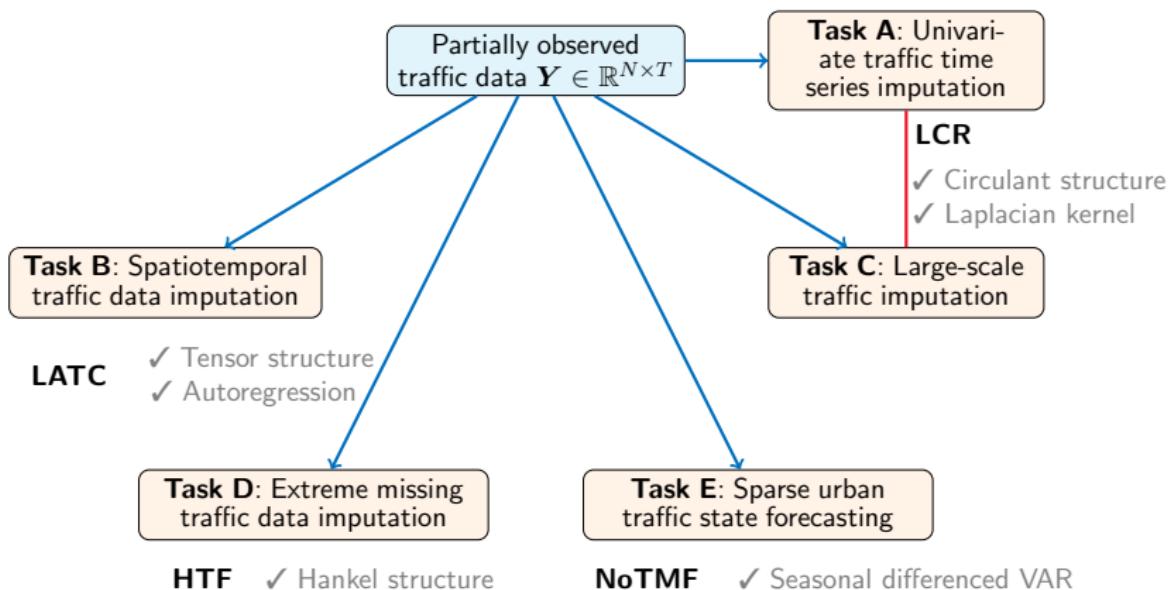
$$\begin{aligned} \min_{\mathbf{Q}, \mathbf{S}, \mathbf{U}, \mathbf{V}} \quad & \frac{1}{2} \sum_{k_1, k_2} \left\| \mathcal{P}_{\Omega_{k_1, k_2}} (\theta_{k_1, k_2}(\mathbf{Y}) - (\mathbf{Q} \star_{\text{row}} \mathbf{s}_{k_1})(\mathbf{U} \star_{\text{row}} \mathbf{v}_{k_2})^\top) \right\|_F^2 \\ &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

- Alternating minimization:

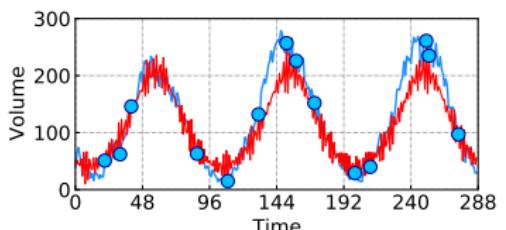
$$\left\{ \begin{array}{l} \mathbf{Q} := \{\mathbf{Q} \mid \frac{\partial f}{\partial \mathbf{Q}} = \mathbf{0}\} \\ \mathbf{s}_{k_1} := \{\mathbf{s}_{k_1} \mid \frac{\partial f}{\partial \mathbf{s}_{k_1}} = \mathbf{0}\}, k_1 \in \{1, 2, \dots, \tau_1\} \\ \mathbf{U} := \{\mathbf{U} \mid \frac{\partial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

Whole Picture

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



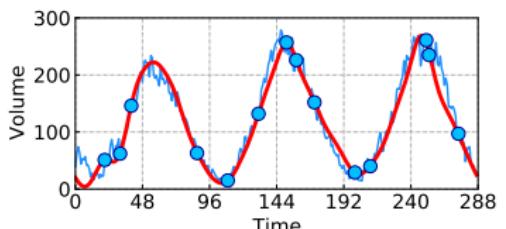
Univariate Traffic Time Series Imputation



CircNNM:

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

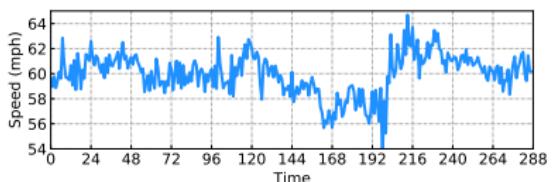
↓ Plus temporal regularization (TR)



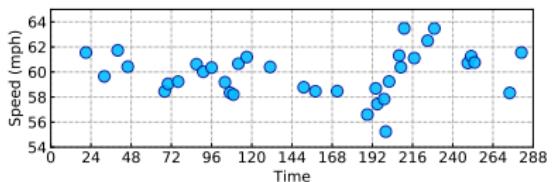
LCR:

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

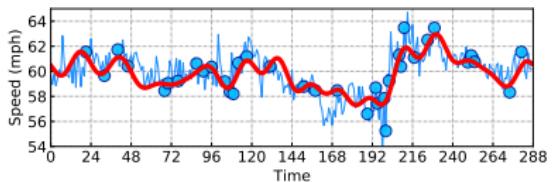
Univariate Traffic Time Series Imputation



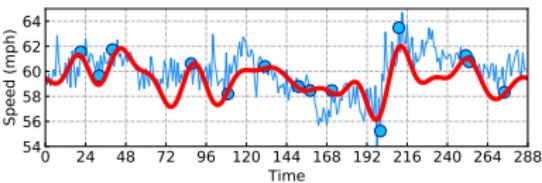
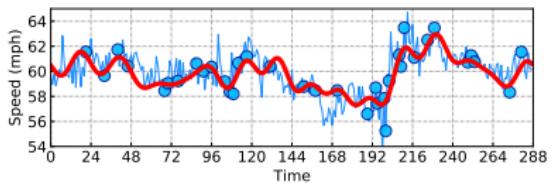
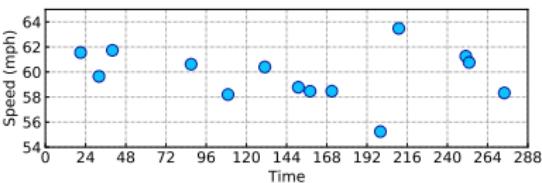
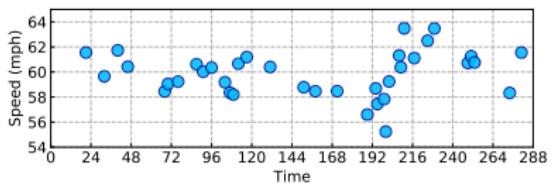
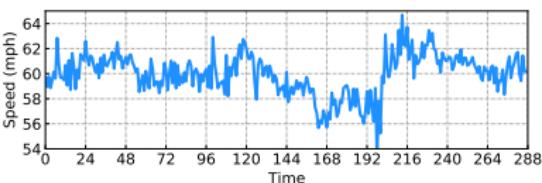
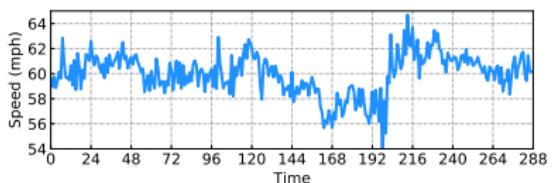
↓ Mask 90% observations



↓ Reconstruct time series



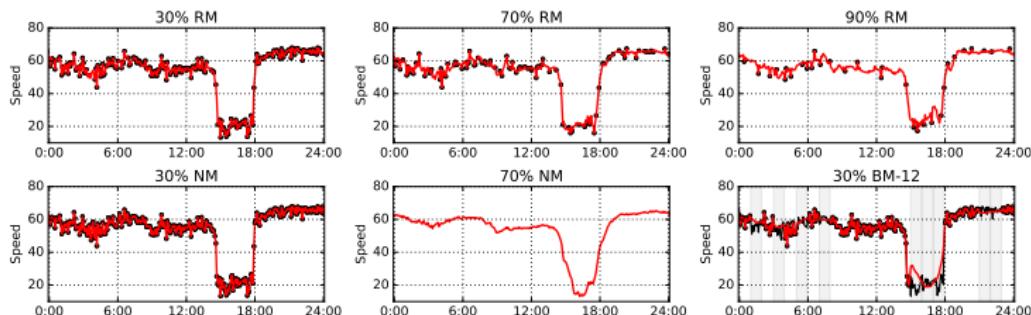
Univariate Traffic Time Series Imputation



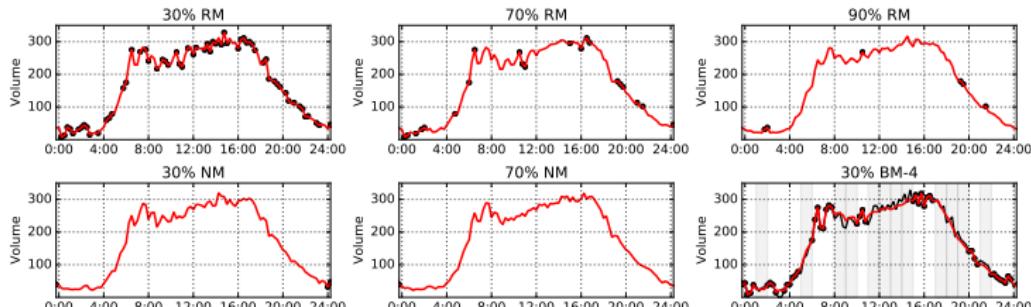
Spatiotemporal Traffic Data Imputation

LATC imputation

- Seattle freeway traffic speed data



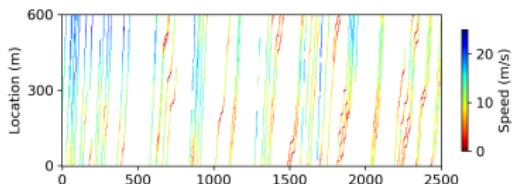
- Portland highway traffic volume data



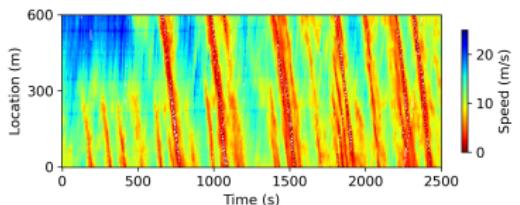
Large-Scale Traffic Data Imputation

LCR

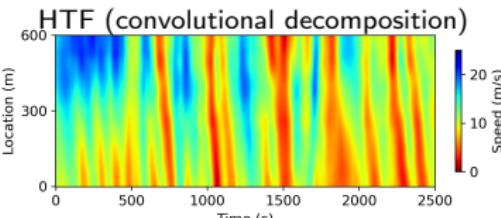
Extreme Missing Traffic Data Imputation



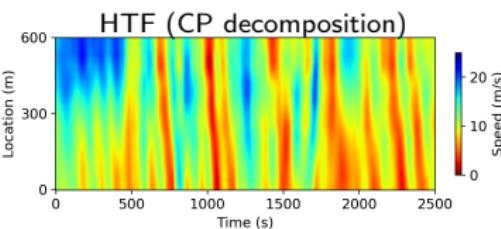
Sparse speed field



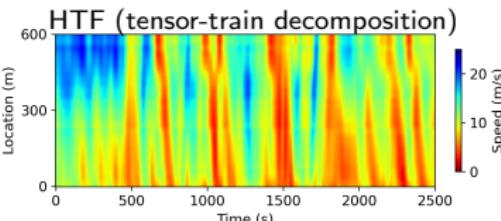
Ground truth speed field



MAPE = 51.92%



MAPE = 53.93%



MAPE = 56.48%

Sparse Urban Traffic State Forecasting

NoTMF

Conclusion

- Data (large-scale, high-dimensional, city-wide, sparse)
- Modeling (meaningfulness and importance of temporal correlations)

Conclusion

Time-Varying Autoregression:

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

Laplacian Convolutional Representation:

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

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

- Local time series trend modeling (temporal regularization):

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

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

Hankel Tensor Factorization:

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

Our studies:

- ② X. Chen, Z. Cheng, N. Saunier, L. Sun (2022). Laplacian convolutional representation for traffic time series imputation. arXiv preprint arXiv:2212.01529.
- ③ X. Chen, L. Sun (2022). Bayesian temporal factorization for multidimensional time series prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence. 44 (9): 4659-4673.

GitHub repository:

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



Dr. HanQin Cai



Xiaoxu Chen



Dr. Zhanhong Cheng



Chengyuan Zhang



Dr. Xi-Le Zhao

References

A short list:

- [Candès & Recht'09] Exact matrix completion via convex optimization.
Foundations of Computational Mathematics, 9 (6), 717-772.
- [Cai et al.'10] A singular value thresholding algorithm for matrix completion
- [Zhang et al.'12] Matrix completion by truncated nuclear norm regularization
- [Hu et al.'12] Fast and accurate matrix completion via truncated nuclear norm regularization
- [Lu et al.'14] Generalized Nonconvex Nonsmooth Low-Rank Minimization
- [Lu et al.'19] Tensor Robust Principal Component Analysis with A New Tensor Nuclear Norm
- [Yokota et al.'18] Missing Slice Recovery for Tensors Using a Low-rank Model in Embedded Space
- [Cai et al.'21] Accelerated Structured Alternating Projections for Robust Spectrally Sparse Signal Recovery
- [Liu'22]
- [Liu & Zhang'23]



Thanks for your attention!

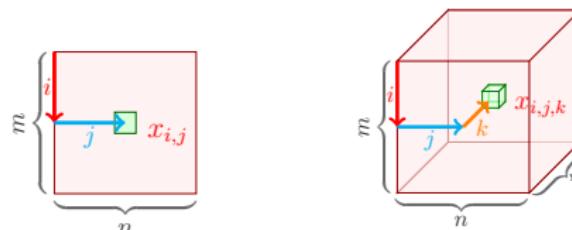
Any Questions?

About me:

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

What Is Tensors?

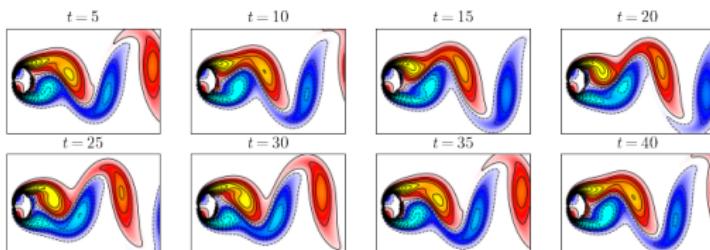
- What is tensor? $\mathbf{X} \in \mathbb{R}^{m \times n}$ vs. $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$



- Tensors are everywhere!



Color image with
RGB channels



Dynamical system (fluid flow)

Appendix