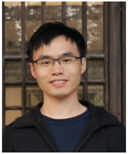
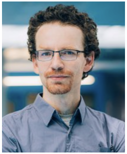
Low-Rank Matrix and Tensor Methods for Spatiotemporal Data Modeling

Xinyu Chen

University of Montreal, Canada

July 28, 2023





Ph.D. Candidate
Xinyu Chen

Supervisor
Prof. Nicolas Saunier

Co-supervisor
Prof. Lijun Sun (McGill)

Our studies: (Low-rank modeling + temporal modeling)

1. X. Chen, C. Zhang, X. Chen, N. Saunier, L. Sun (2023). Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression. IEEE Transactions on Knowledge and Data Engineering. Early access.
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

Slides:

- https://xinychen.github.io/slides/tensors_in10min.pdf

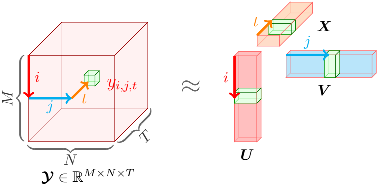
Tensor Factorization

- Revisit tensor factorization (TF)

1927	1960s	1970	2009	2011
Higher-Order SVD	Tucker Decomposition	CP Decomposition	Tensor Decompositions and Applications	Tensor-Train Decomposition
Frank Lauren Hitchcock	Ledyard R. Tucker	J. Douglas Carroll Jih-Jie Chang Richard A. Harshman	Tamara G. Kolda	Ivan Oseledets



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



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

Time-Varying Autoregression

- **Motivation: Identify the dynamic patterns of time-varying systems.**
  (e.g., fluid flow, climate variables, and human mobility)

- Given a sequence of time series snapshots $\boldsymbol{y}_t \in \mathbb{R}^N$, $t = 1, 2, \ldots, T$,

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

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

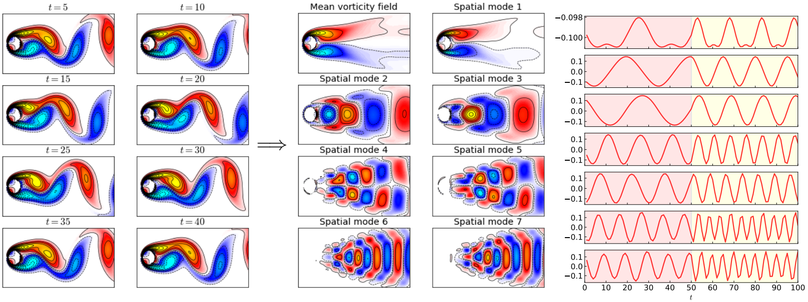
- **(Ours)** Parameterize coefficients via TF:

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

- Time-varying autoregression with TF

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

- **Multi-resolution fluid flow dataset** (the first 50 snapshots + 50 snapshots randomly selected from the last 100 snapshots)



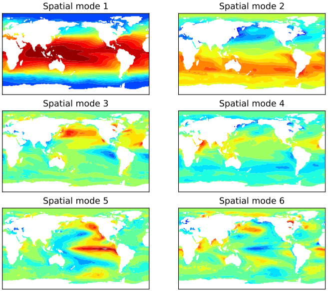
Vectorize snapshots as $\{\boldsymbol{y}_t\}$ — Spatial modes in $\boldsymbol{W}$ — Temporal modes in $\boldsymbol{X}$

- Produce interpretable patterns and identify the system of different frequencies.

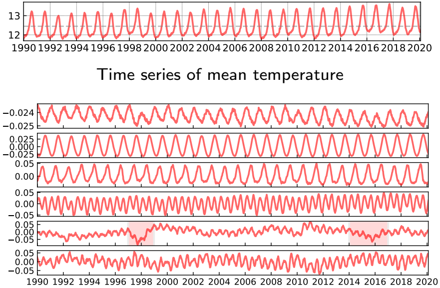
- Time-varying autoregression with TF

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

- Sea surface temperature (**SST**) dataset



Spatial modes in $\boldsymbol{W}$

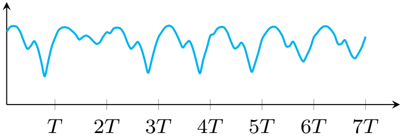


Temporal modes in $\boldsymbol{X}$

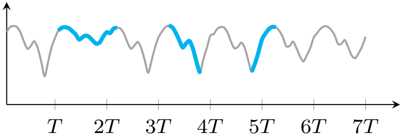
- Identify two strongest El Nino events (on 1997-98 & 2014-16).

Motivation: Time series imputation

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



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

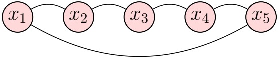


- **[Question]** How to characterize both global and local trends in sparse time series data?

Laplacian Convolutional Representation

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

- Intuition of (circulant) Laplacian matrix.



Undirected and circulant graph

$$\xrightarrow{\text{Modeling}} \quad \boldsymbol{L}=\begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

(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:

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

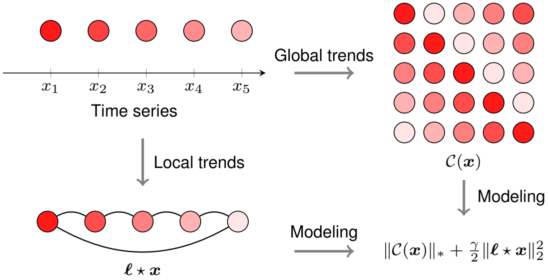
Laplacian Convolutional Representation

Laplacian Convolutional Representation (LCR)

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

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

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



Laplacian Convolutional Representation

- Augmented Lagrangian function:

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

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

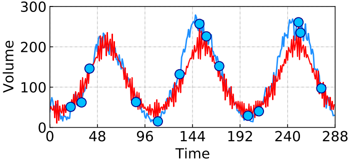
- The ADMM scheme:

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

- Optimize $\boldsymbol{x}$?

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

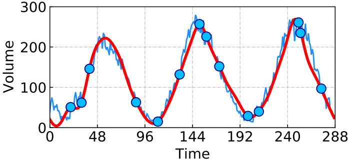
  Nuclear norm minimization $\Rightarrow$ ℓ_1-norm minimization with FFT.



CircNNM:

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

$\Downarrow$ Plus temporal regularization (TR)

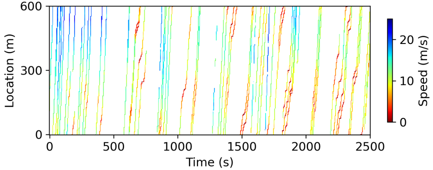


LCR:

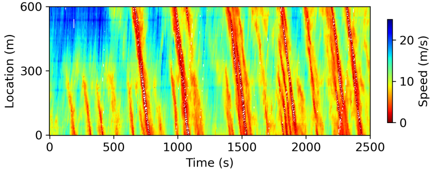
$$\min_{\boldsymbol{x}} \ \|\mathcal{C}(\boldsymbol{x})\|_* + \frac{\gamma}{2}\|\boldsymbol{\ell} \star \boldsymbol{x}\|_2^2 \quad \text{s. t.} \ \|\mathcal{P}_\Omega(\boldsymbol{x}-\boldsymbol{y})\|_2 \leq \epsilon$$

Motivation: Spatiotemporal data reconstruction

- Speed field reconstruction problem in vehicular traffic flow.



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



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

Hankel Tensor Factorization

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

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

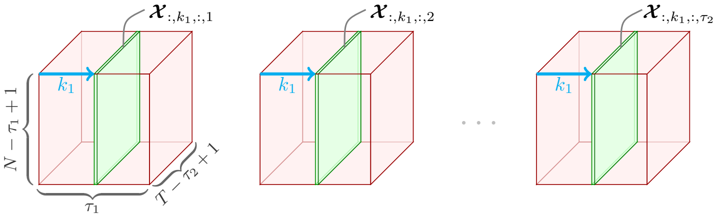
Hankel matrix (Source: Twitter)

Hankel Tensor Factorization

- Hankel 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

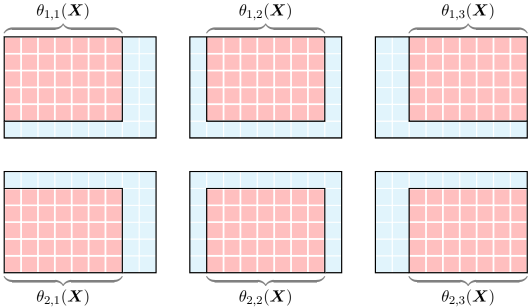
Hankel indexing:

- Sampling function for the Hankelization:

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

  referring to 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

Our model:

- 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:

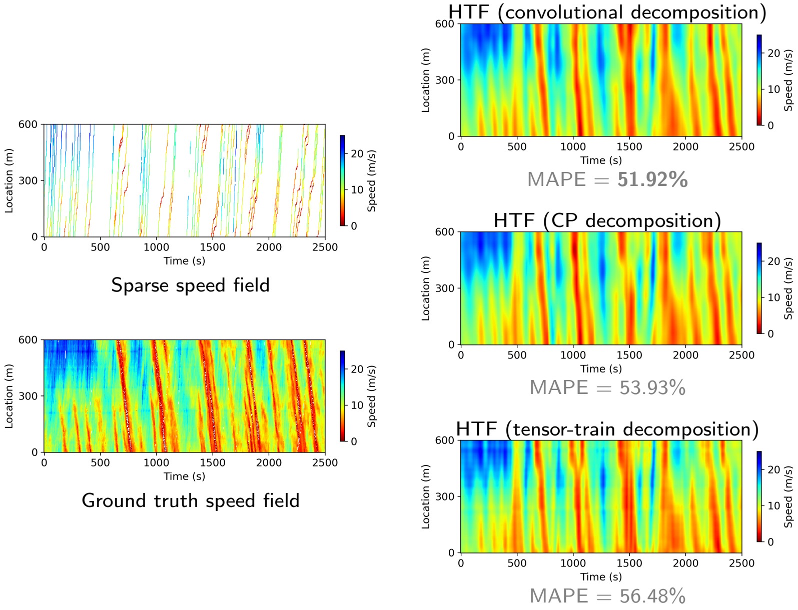
- CP tensor decomposition (Khatri-Rao product $\odot$):

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

- 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



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%
Location (m)
Time (s)
Speed (m/s)

Future Directions

•

Thanks for your attention!

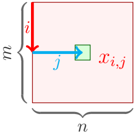
Any Questions?

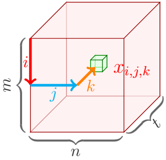
About me:

- Homepage: https://xinychen.github.io
- GitHub: https://github.com/xinychen (**3.4k+ stars**)
- Blog: https://medium.com/@xinyu.chen (**70k+ views**)
- How to reach me: chenxy346@gmail.com

Tensors

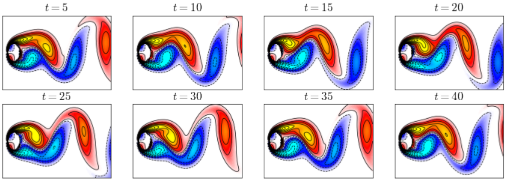
- What is tensor? $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ vs. $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{m \times n \times t}$





- Tensors are everywhere!

Color image with RGB channels



Dynamical system (fluid flow)

Appendix

LCR:

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

$$\begin{aligned}\boldsymbol{x} &:= \arg\min_{\boldsymbol{x}}\ \|\mathcal{C}(\boldsymbol{x})\|_* + \frac{\gamma}{2}\|\boldsymbol{\ell}\star\boldsymbol{x}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{w}/\lambda\|_2^2 \\ \Longrightarrow \hat{\boldsymbol{x}} &:= \arg\min_{\hat{\boldsymbol{x}}}\ \|\hat{\boldsymbol{x}}\|_1 + \frac{\gamma}{2T}\|\hat{\boldsymbol{\ell}}\circ\hat{\boldsymbol{x}}\|_2^2 + \frac{\lambda}{2T}\|\hat{\boldsymbol{x}}-\hat{\boldsymbol{z}}+\hat{\boldsymbol{w}}/\lambda\|_2^2 \\ &= \arg\min_{\hat{\boldsymbol{x}}}\ \|\hat{\boldsymbol{x}}\|_1 + \frac{\lambda}{2T}\|\hat{\boldsymbol{x}}-(\lambda\hat{\boldsymbol{z}}-\hat{\boldsymbol{w}})\oslash(\gamma\hat{\boldsymbol{\ell}}^*\circ\hat{\boldsymbol{\ell}}+\lambda\mathbb{1}_T)\|_2^2\end{aligned}$$

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).

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:

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

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

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

Appendix

- Hankel matrix
  - On time series $\boldsymbol{y} = (y_1, y_2, \ldots, y_5)^\top$ with $\tau = 2$:

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

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

  - Automatic temporal modeling.