THE UNIVERSITY OF UTAH

Bayesian Continuous-Time Tucker Decomposition

Shikai Fang, Akil Narayan, Robert M. Kirby, Shandian Zhe

Presenter: Shikai Fang

School of computing, The University of Utah

For ICML 2022

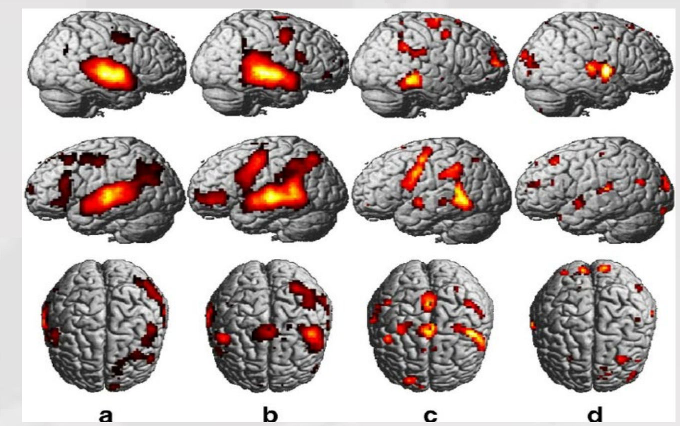
Outline

1. Background
2. Motivation
3. Dynamic Tucker-Core via SDE
4. Message-Passing Inference: SDE Discretization+ Conditional Moment Matching
5. Experiments on Real-world Data

Tensor Data: Widely Used High-Order Data Structures to Represent Interactions of Multiple Objects/Entities

(user, movie, episode)

(user, advertisement, page-section)



(subject, voxel, electrode)

(user, item, online-store)

(user, user, location, message-type)

(patient, gene, condition)

Tensor Decomposition: estimate latent factors to reconstruct tensor with observed entries

- Simple case:

  Collaborative Filtering (Matrix Factorization)

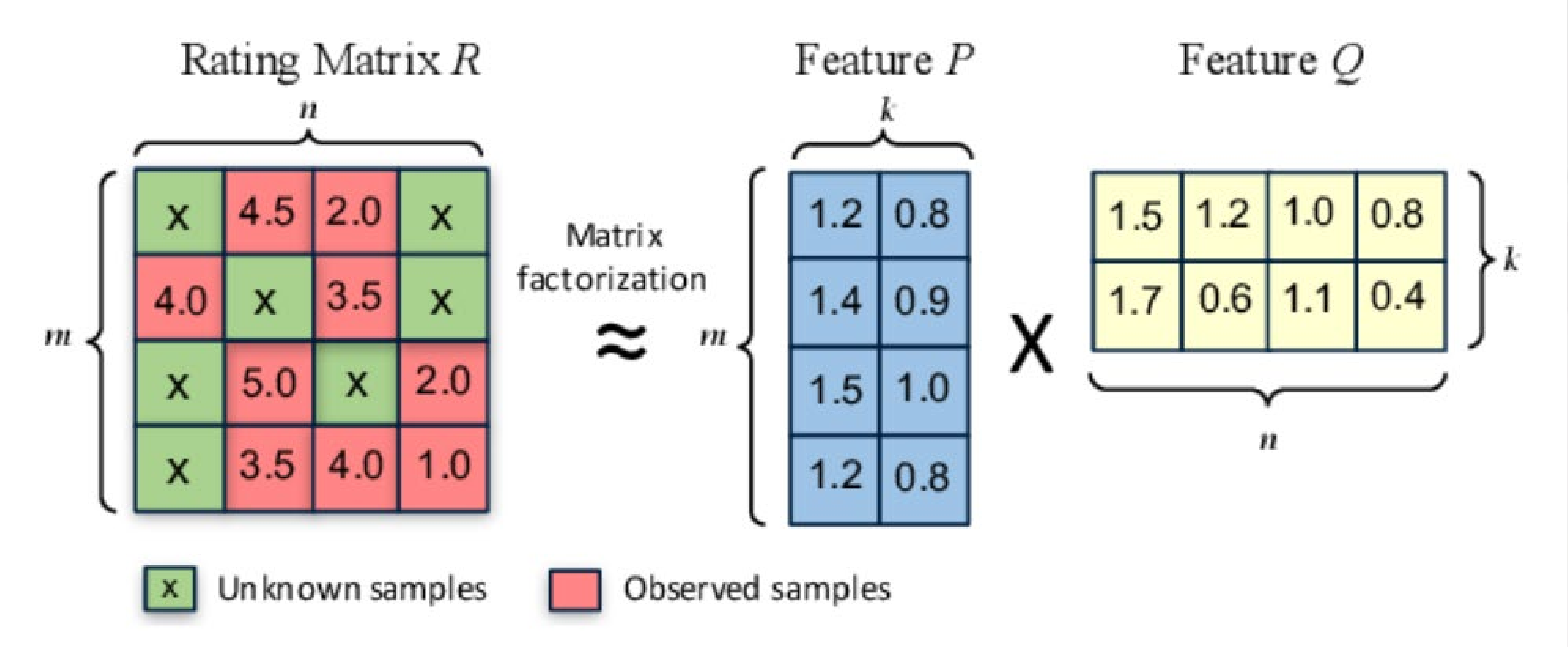


Image from https://www.slideshare.net/hontolab/matrix-factorization-192159058

<u>Tensor Decomposition</u>: estimate latent factors to reconstruct tensor with observed entries

- Simple case:
  Collaborative Filtering (Matrix Factorization)

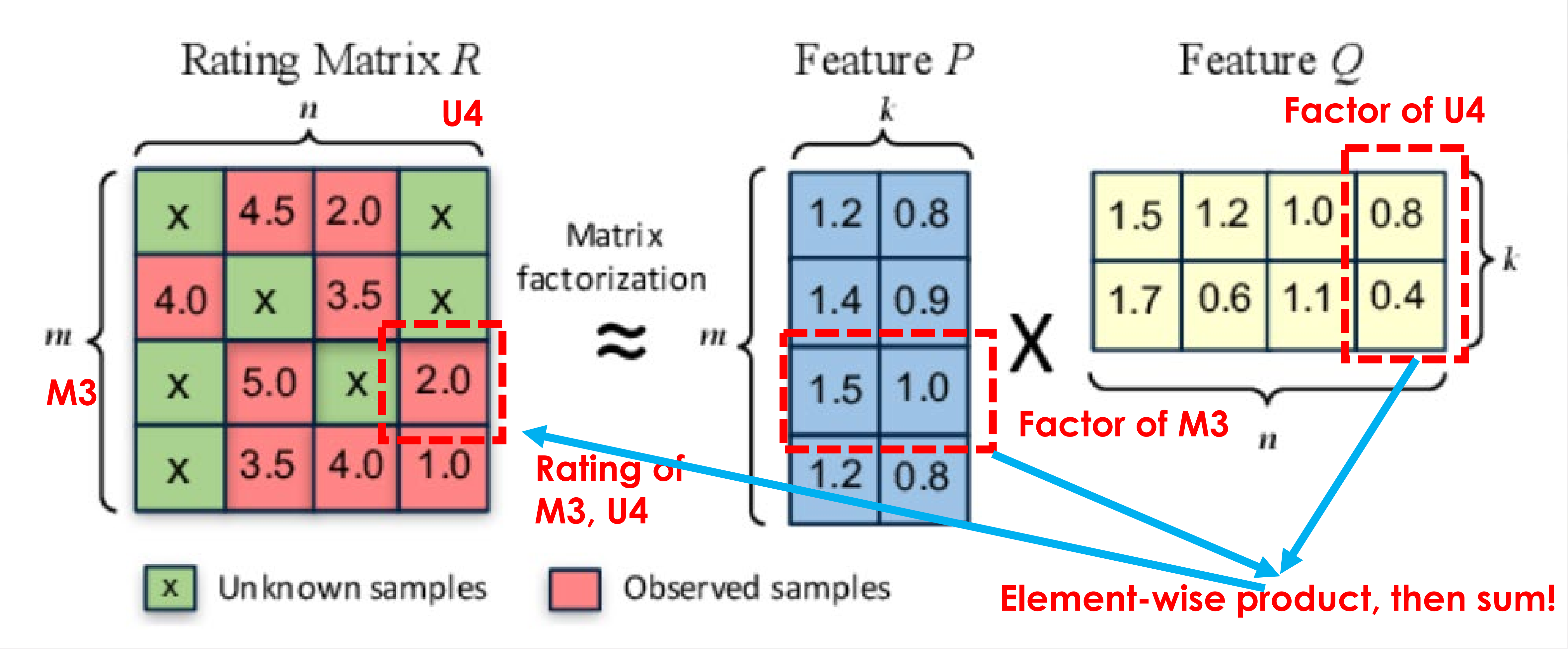
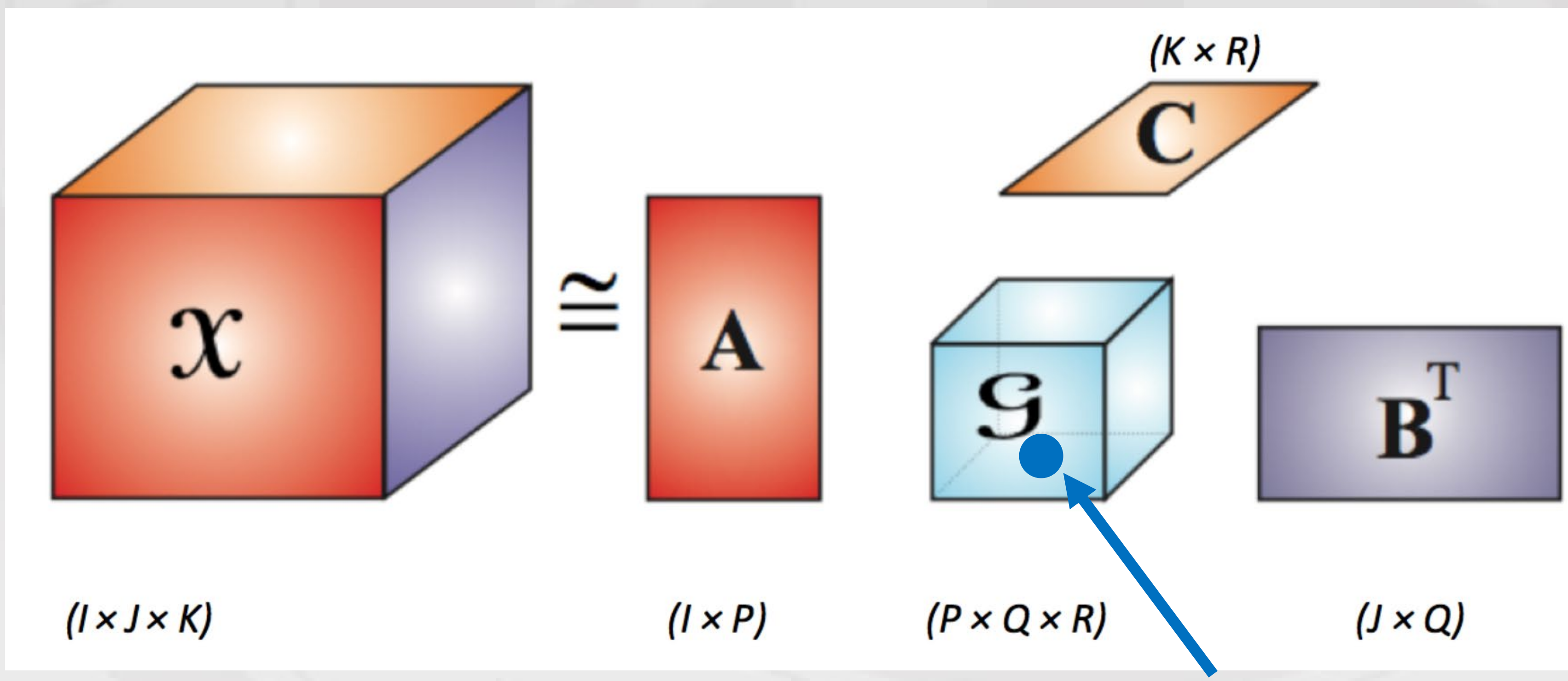


Image from https://www.slideshare.net/hontolab/matrix-factorization-192159058

Tucker Decomposition

- 2-D matrix => N-D tensor
- Element-wise interaction => all possible interactions

Tucker Decomposition



One interaction weight

Element-wise form for a K-mode tensor Y:

$$y_{\mathbf{i}} \approx \boxed{\mathrm{vec}(\mathcal{W})^\top} \left(\mathbf{u}^1_{i_1} \otimes \ldots \otimes \mathbf{u}^K_{i_K} \right)$$

$$= \sum_{r_1=1}^{R_1} \cdots \sum_{r_K=1}^{R_K} \left[\boxed{w_{(r_1,\ldots,r_K)}} \cdot \boxed{\prod_{k=1}^{K} u^k_{i_k,r_k}} \right]$$

One Interaction of latent factors

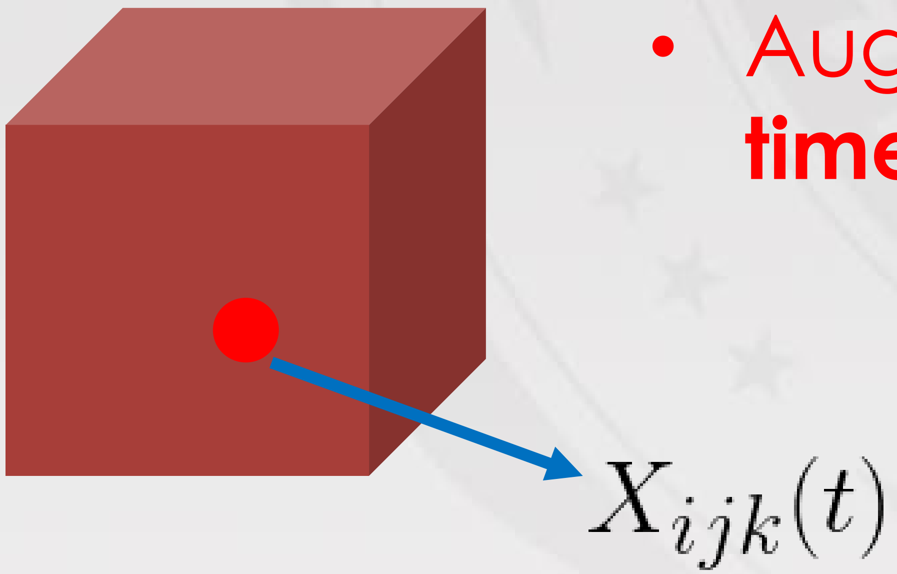
Image from https://iksinc.online/2018/05/02/understanding-tensors-and-tensor-decompositions-part-3/.

Challenge: Temporal info in Tensor

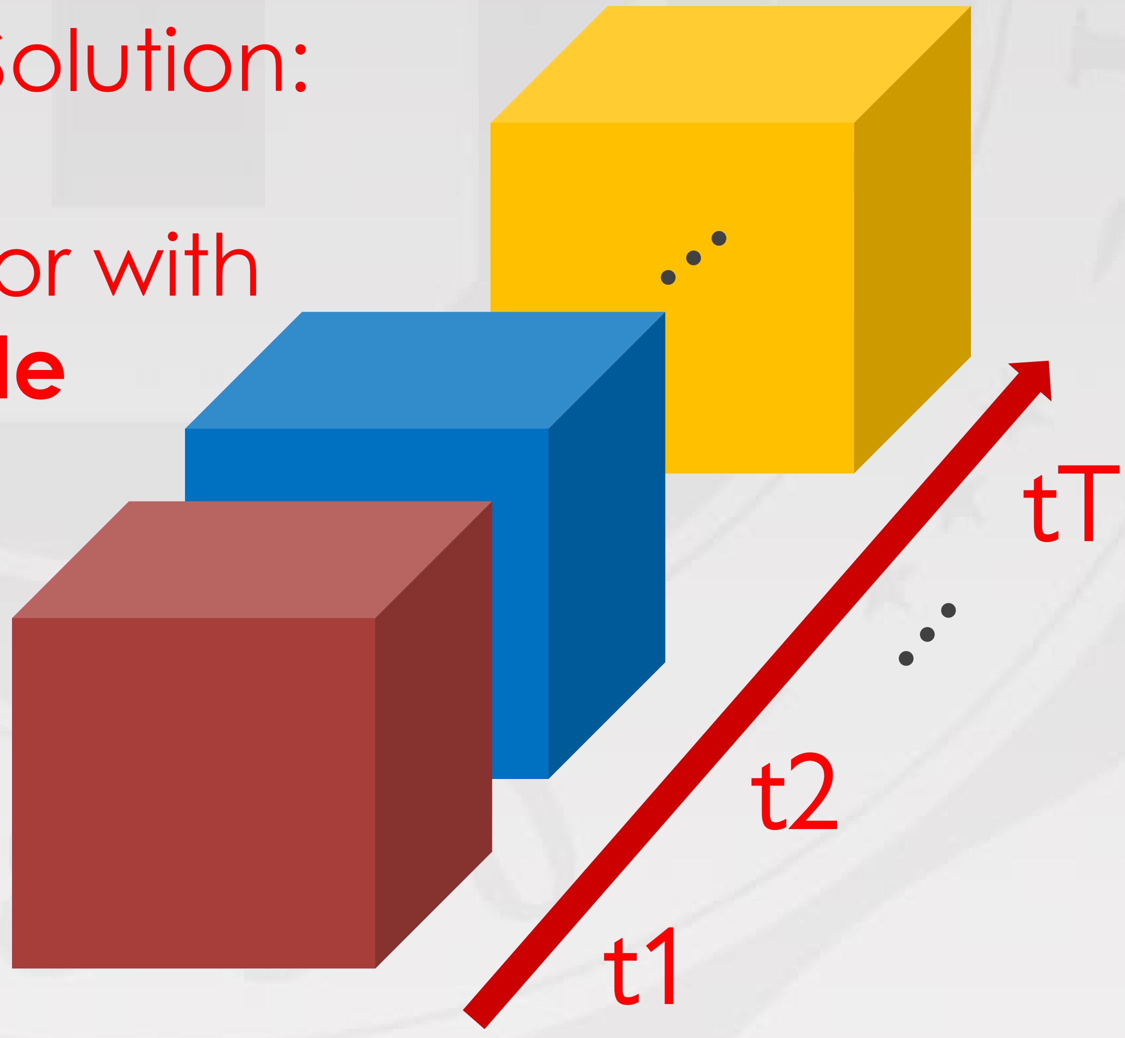
What about each entry is time-dependent?

Straightforward Solution:

- Drop time or
- Augment tensor with **time-step mode**



$(I \times J \times K)$

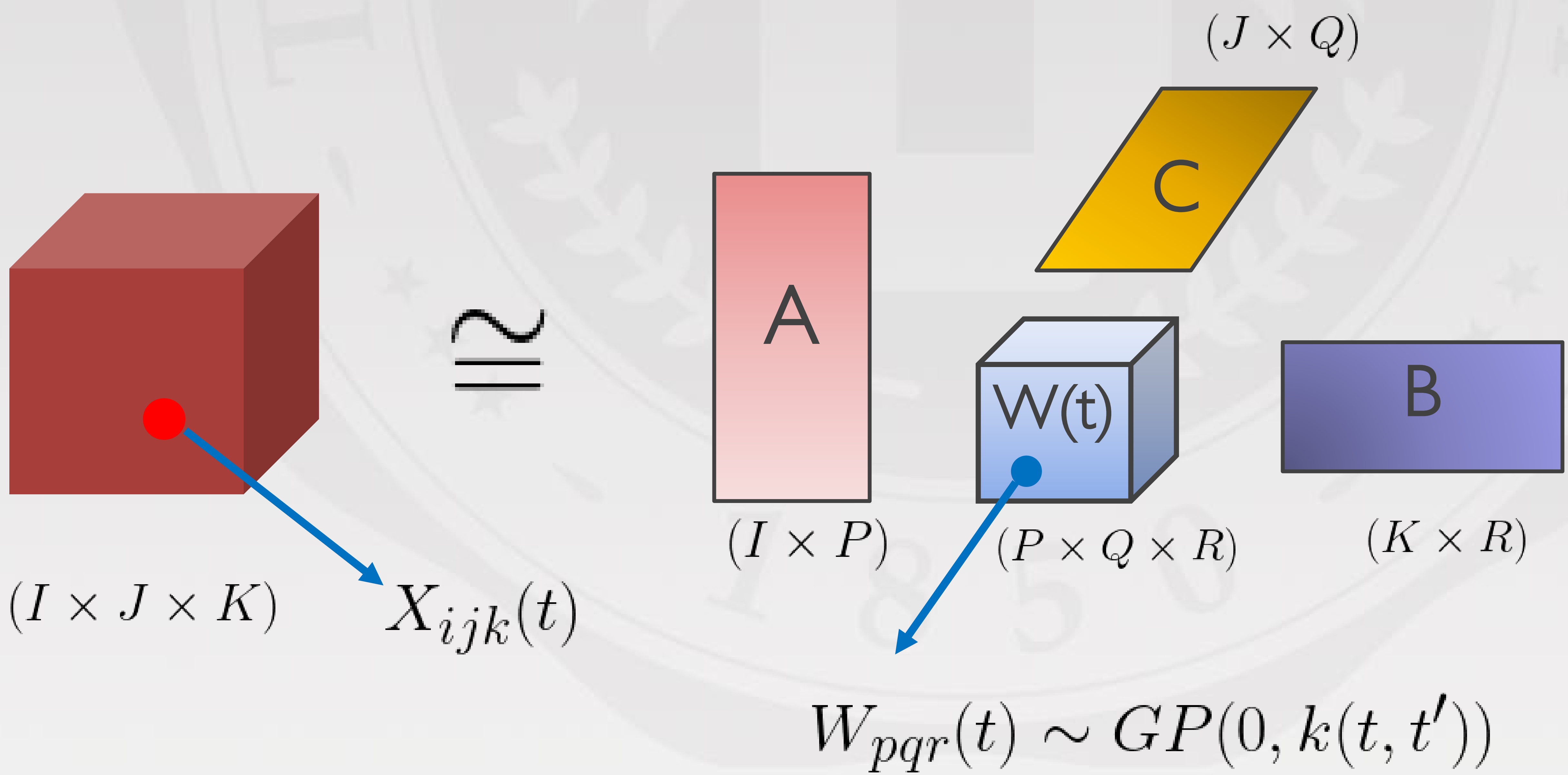


$(I \times J \times K \times T)$

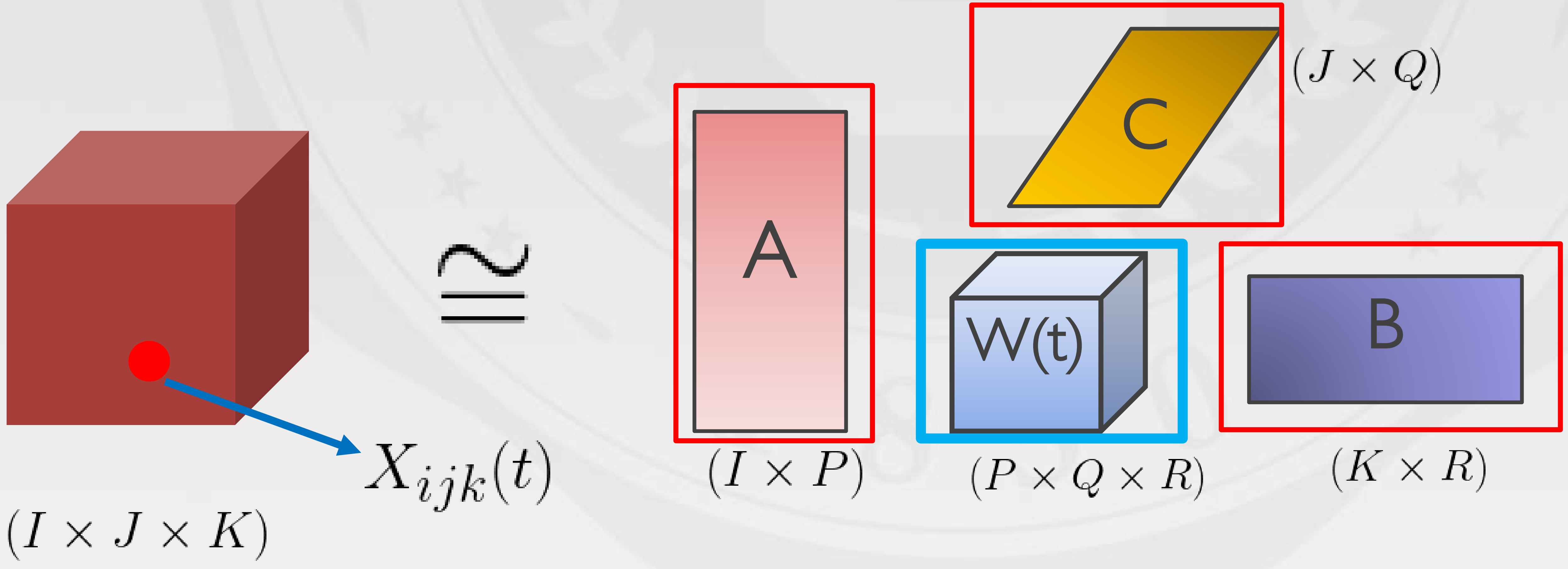
Problem:

1. Too Sparse
2. Ignore the temporary continuity

Our Solution: Modeling Dynamic Tucker Core by Temporal Gaussian Processes



High-level Motivation:
Decouple the representation learning of factors and the capture of dynamic pattern



Joint Probability:

$$p\left(\mathcal{U}, \{\mathbf{w_r}\}_{\mathbf{r}}, \tau, \mathbf{y}\right) =$$

$$\boxed{\mathrm{Gam}\left(\tau \mid b_0, c_0\right) \prod_{k=1}^{K} \prod_{j=1}^{d_k} \mathcal{N}\left(\mathbf{u}_j^k \mid \mathbf{0}, \mathbf{I}\right)} \times \boxed{\prod_{\mathbf{r}=(1,\ldots,1)}^{(R_1,\ldots,R_K)} \mathcal{N}\left(\mathbf{w_r} \mid \mathbf{0}, \mathbf{K_r}\right)} \times$$

Priors of factors and noise

Temporal GPs on Tucker Core

$$\boxed{\prod_{n=1}^{N} \mathcal{N}\left(y_n \mid \mathrm{vec}\left(\mathcal{W}\left(t_n\right)\right)^{\top}\left(\mathbf{u}_{i_{n_1}}^1 \otimes \ldots \otimes \mathbf{u}_{i_{n_K}}^K\right), \tau^{-1}\right)}$$

Gaussian Likelihood

Computational challenge: **O(N^3)** cost of full GPs

THE UNIVERSITY OF UTAH

To **avoid low-rank/sparse approx.** (low quality),
but enjoy **linear-cost inference of full GPs,**

We apply a crucial fact:

Temporal GPs

with stationary kernel

Linear Time-Invariant(LTI) SDE

discrete form on $(t_1, t_2 \ldots t_N)$

State Space Model (Gauss Markov Chain)

Can be solved by
Kalman filtering &
RTS Smoothing in O(N)*

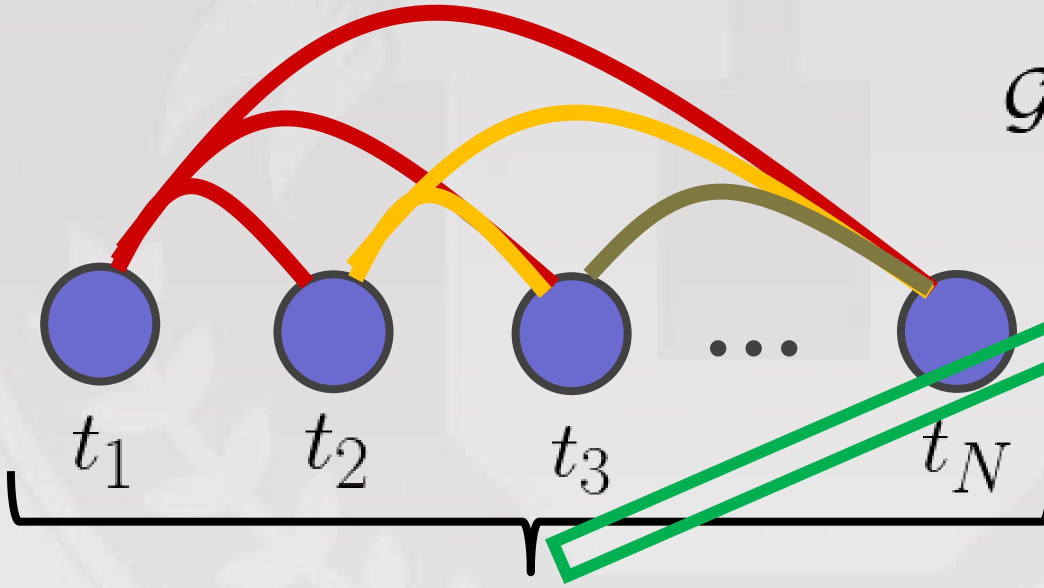
*: it holds with linear emission/observed likelihood, if with non-linear, we could apply non-linear filter and smoothing

Illustration of computation cost:

Temporal GPs

Temporal States:

$\mathcal{GP}\left(\mathbf{w}_{\mathbf{r}} \mid \mathbf{0}, \mathbf{K}_{\mathbf{r}}(t, t')\right)$



Space: $\mathcal{O}(N^2)$
Time: $\mathcal{O}(N^3)$

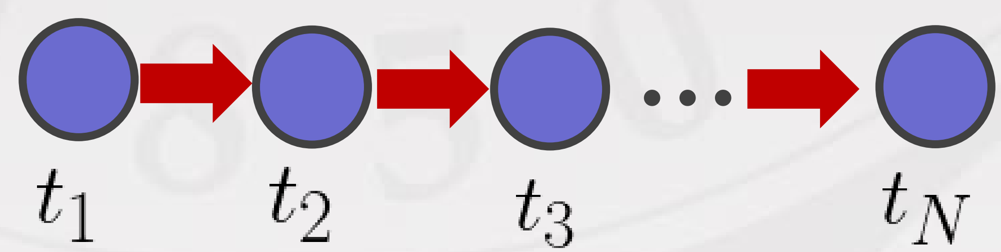
LTI-SDE

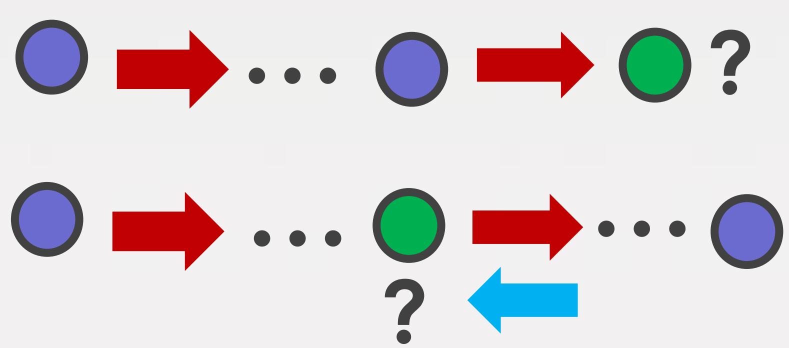
$$\frac{\mathrm{d}\gamma_{\mathbf{r}}(t)}{\mathrm{d}t} = \mathbf{F}\gamma_{\mathbf{r}} + \mathbf{L}\xi(t)$$

discrete form

$$p\left(\gamma_{\mathbf{r}}\left(t_{n+1}\right) \mid \gamma_{\mathbf{r}}\left(t_n\right)\right) = \mathcal{N}\left(\gamma_{\mathbf{r}}\left(t_{n+1}\right) \mid \mathbf{A}_n \gamma_{\mathbf{r}}\left(t_n\right), \mathbf{Q}_n\right)$$

State Space Model





Kalman Filter

RTS Smoothing

Space: $\mathcal{O}(N)$
Time: $\mathcal{O}(N)$

Specifically,

$$\prod_{\mathbf{r}=(1,\ldots,1)}^{(R_1,\ldots,R_K)} \mathcal{N}\left(\mathbf{w_r} \mid \mathbf{0}, \mathbf{K_r}\right)$$

Temporal GPs

with stationary kernel (e.g., Matern25)

$$\begin{cases} \gamma_{\mathbf{r}}(t) = \left(w_{\mathbf{r}}, \dfrac{\mathrm{d}w_{\mathbf{r}}}{\mathrm{d}t}\right)^{\top} \\ \dfrac{\mathrm{d}\gamma_{\mathbf{r}}(t)}{\mathrm{d}t} = \mathbf{F}\gamma_{\mathbf{r}} + \mathbf{L}\xi(t) \end{cases}$$

Linear Time-Invariant(LTI) SDE

discrete form on $(t_1, t_2 \ldots t_N)$

$$\begin{cases} p\left(\boldsymbol{\gamma}_{\mathbf{r}}\left(t_1\right)\right) = \mathcal{N}\left(\boldsymbol{\gamma}_{\mathbf{r}}\left(t_1\right) \mid \mathbf{0}, \mathbf{P}_{\infty}\right) \\ p\left(\boldsymbol{\gamma}_{\mathbf{r}}\left(t_{n+1}\right) \mid \boldsymbol{\gamma}_{\mathbf{r}}\left(t_n\right)\right) = \mathcal{N}\left(\boldsymbol{\gamma}_{\mathbf{r}}\left(t_{n+1}\right) \mid \mathbf{A}_n \boldsymbol{\gamma}_{\mathbf{r}}\left(t_n\right), \mathbf{Q}_n\right) \end{cases}$$

State Space Model (Gauss Markov Chain)

Recall: linear-cost solver - KF, RTS

Reformulate **Tucker core** with **State Space Priors**

$$p\left(\bar{\gamma}_1\right) \prod_{n=1}^{N-1} p\left(\bar{\gamma}_{n+1} \mid \bar{\gamma}_n\right)$$

We post **Gaussian-Gamma Approx**. to fit each data-llk

$$\mathcal{N}\left(y_n \mid (\mathbf{H}\bar{\gamma}_n)^\top \left(\mathbf{u}^1_{i_{n_1}} \otimes \ldots \otimes \mathbf{u}^K_{i_{nK}}\right), \tau^{-1}\right) \approx$$

$$Z_n \prod_{k=1}^{K} \boxed{\mathcal{N}\left(\mathbf{u}^k_{i_{n_k}} \mid \mathbf{m}^{k,n}_{i_{n_k}}, \mathbf{V}^{k,n}_{i_{n_k}}\right) \cdot \operatorname{Gam}\left(\tau \mid b_n, c_n\right)}$$

Approx. Msg of Factors & noise

$$\times \boxed{\mathcal{N}\left(\mathbf{H}\bar{\gamma}_n \mid \boldsymbol{\beta}_n, \mathbf{S}_n\right)}$$

Approx. Msg of SDE states /Tucker core

Substitute these into joint prob.

The proposed approx. posterior is:

$$q\left(\mathcal{U}, \{\bar{\gamma}_n\}, \tau\right) \propto \prod_{k=1}^{K} \prod_{j=1}^{d_k} \mathcal{N}\left(\mathbf{u}_j^k \mid \mathbf{0}, \mathbf{I}\right) \mathrm{Gam}\left(\tau \mid b_0, c_0\right)$$

Standard moment match? Infeasible!

$$\prod_{n=1}^{N} \prod_{k=1}^{K} \boxed{\mathcal{N}\left(\mathbf{u}_{i_{n_k}}^k \mid \mathbf{m}_{i_{n_k}}^{k,n}, \mathbf{V}_{i_{n_k}}^{k,n}\right) \mathrm{Gam}\left(\tau \mid b_n, c_n\right)}$$

$$p\left(\bar{\gamma}_1\right) \boxed{\mathcal{N}\left(\mathbf{H}\bar{\gamma}_1 \mid \boldsymbol{\beta}_1, \mathbf{S}_1\right)} \prod_{n=1}^{N-1} p\left(\bar{\gamma}_{n+1} \mid \bar{\gamma}_n\right) \boxed{\mathcal{N}\left(\mathbf{H}\bar{\gamma}_n \mid \boldsymbol{\beta}_n, \mathbf{S}_n\right)}$$

SDE states: Solve by KF and RTS

Apply conditional moment matching and delta method!

- **Conditional Moment Match**

$$\mathbb{E}_{\widetilde{p}}[\phi(\boldsymbol{\eta}_n)] = \mathbb{E}_{\widetilde{p}\left(\Theta_{\setminus \eta_n}\right)}\left[\mathbb{E}_{\widetilde{p}\left(\boldsymbol{\eta}_n \mid \Theta_{\setminus \eta_n}\right)}\left[\phi(\boldsymbol{\eta}) \mid \Theta_{\setminus \boldsymbol{\eta}_n}\right]\right]$$

- **Delta method:**

$$\mathbb{E}_{q\left(\Theta_{\setminus \eta_n}\right)}[\boldsymbol{\rho}_n] \approx \rho_n\left(\mathbb{E}_q\left[\boldsymbol{\Theta}_{\setminus \boldsymbol{\eta}_n}\right]\right)$$

Enable ***tractable moment matching***
to update approx. probability terms under
Expectation Propagation(EP) framework

Algorithm 1 BCTT

Input: $\mathcal{D} = \{(\mathbf{i}_1, t_1, y_1), \ldots, (\mathbf{i}_N, t_N, y_N)\}$, kernel hyper-parameters l, σ^2

Initialize approximation terms in (10) for each likelihood.

repeat

Run KF and RTS smoothing to compute each $q(\overline{\boldsymbol{\gamma}}_n)$

for $n = 1$ **to** N in parallel **do**

Simultaneously update $\mathcal{N}(\mathbf{H}\overline{\boldsymbol{\gamma}}_n | \boldsymbol{\beta}_n, \mathbf{S}_n)$, $\mathrm{Gam}(\tau | b_n, c_n)$ and $\left\{\mathcal{N}\left(\mathbf{u}^k_{i_{n_k}} | \mathbf{m}^{k,n}_{i_{n_k}}, \mathbf{V}^{k,n}_{i_{n_k}}\right)\right\}_k$ in (10) with conditional moment matching and multi-variate delta method.

end for

until Convergence

Return: $\{q(\mathcal{W}(t_n))\}_{n=1}^N, \{q(\mathbf{u}^k_j)\}_{1 \le k \le K, 1 \le j \le d_k}, q(\tau)$

Time cost: $\mathcal{O}(N\bar{R})$ **Space cost:** $\mathcal{O}\left(N\left(\bar{R}^2 + \sum_{k=1}^{K} R_k^2\right)\right)$

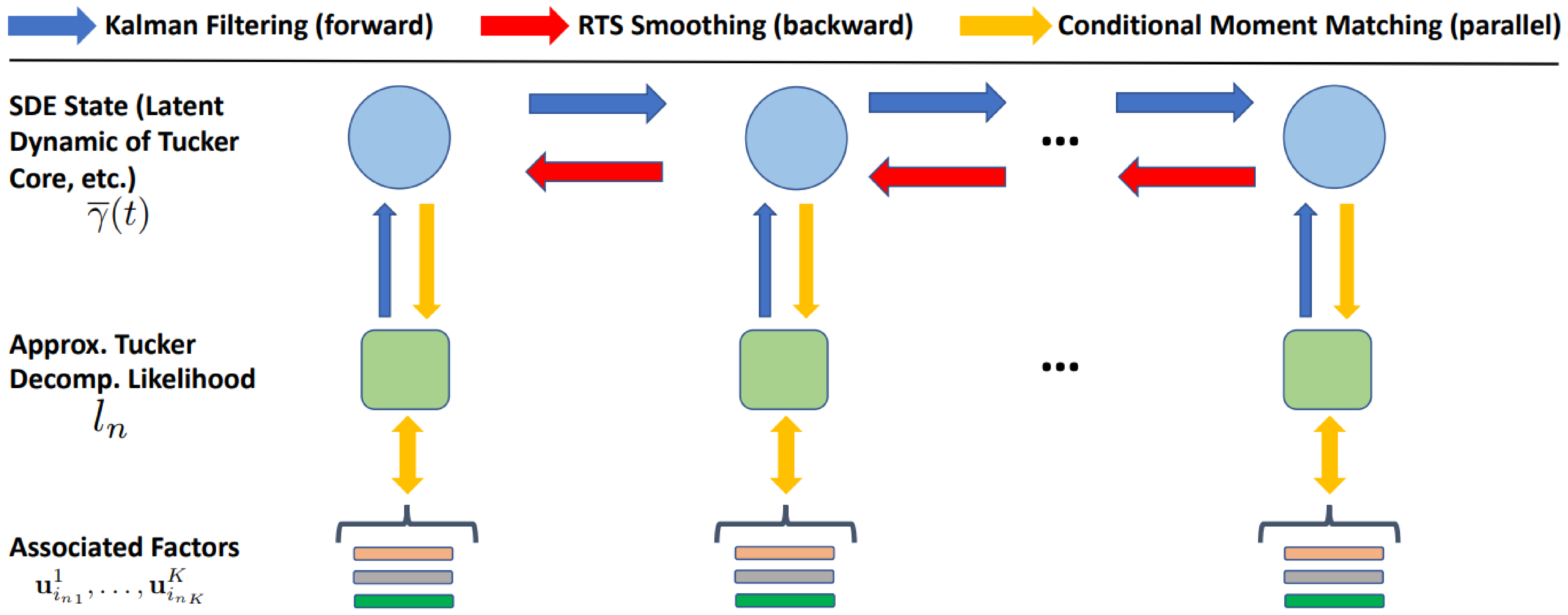
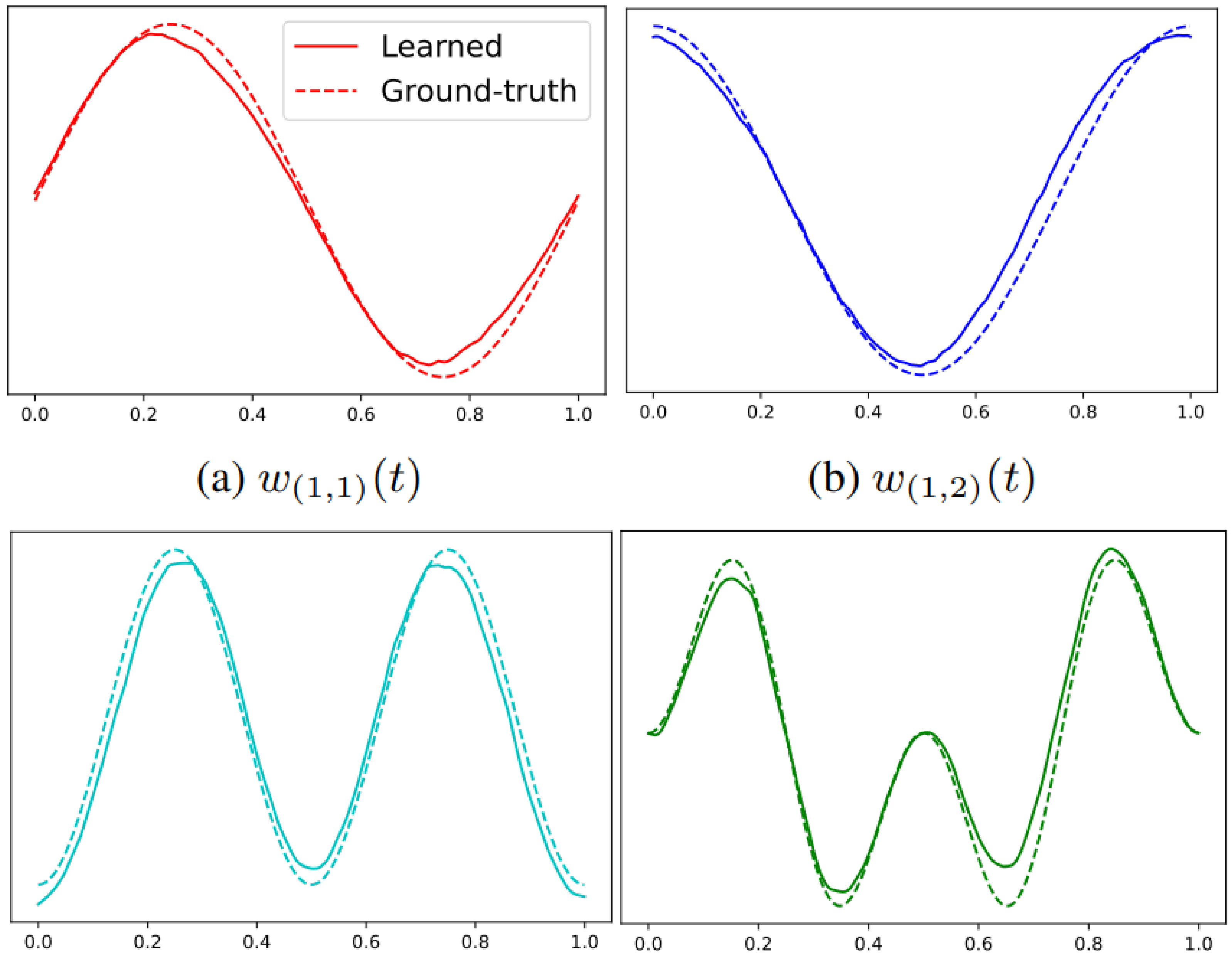


Figure 1. Graphical illustration of the message-passing inference algorithm.

THE UNIVERSITY OF UTAH

Can BCTT capture the temporal patterns in tensor?

- Exp on **simulation data**
- Plot the dynamics of Tucker core



(a) $w_{(1,1)}(t)$ (b) $w_{(1,2)}(t)$

(c) $w_{(2,1)}(t)$ (d) $w_{(2,2)}(t)$

Can BCTT capture the temporal patterns in tensor?

- Exp on **real-world data(DBLP dataset)**
- Scatter low-rank structures of Tucker core

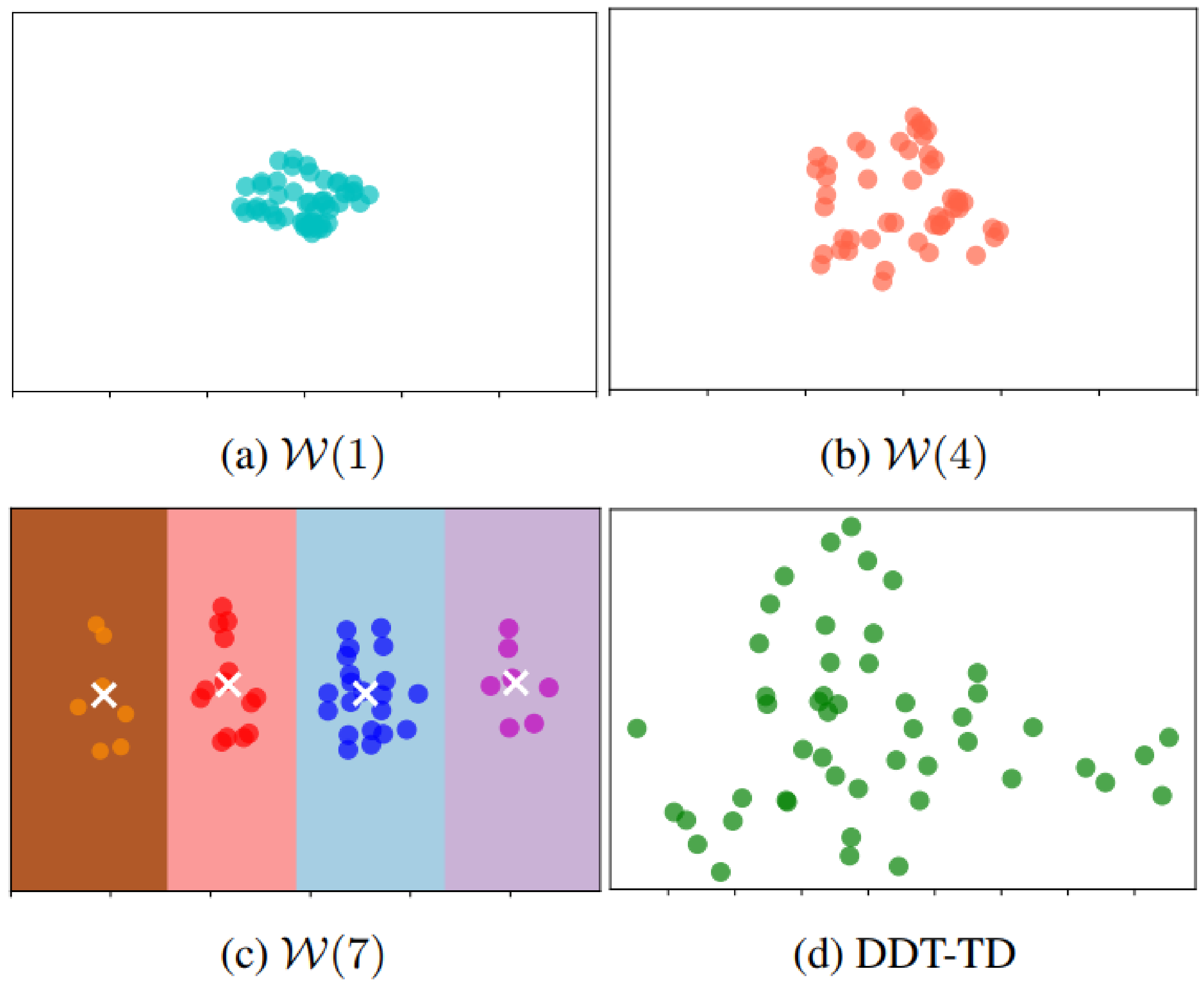


Figure 4. The structures of learned tensor-core at different time points by BCTT (a-c) and the static tensor-score learned by dynamic discrete-time Tucker decomposition (DDT-TD).

Prediction with BCTT

- Prediction performance of BCTT on 3 real-world data

RMSE	*MovieLens*	*AdsClicks*	*DBLP*
CT-CP	1.113 ± 0.004	1.337 ± 0.013	0.240 ± 0.007
CT-GP	0.949 ± 0.008	1.422 ± 0.008	0.227 ± 0.009
DT-GP	0.963 ± 0.008	1.436 ± 0.015	0.227 ± 0.007
DDT-GP	0.957 ± 0.008	1.437 ± 0.010	0.225 ± 0.006
DDT-CP	1.022 ± 0.003	1.420 ± 0.020	0.245 ± 0.004
DDT-TD	1.059 ± 0.006	1.401 ± 0.022	0.232 ± 0.09
BCTT	$\mathbf{0.922 \pm 0.002}$	$\mathbf{1.322 \pm 0.012}$	$\mathbf{0.214 \pm 0.009}$
MAE			
CT-CP	0.788 ± 0.004	0.787 ± 0.006	0.105 ± 0.001
CT-GP	0.714 ± 0.004	0.891 ± 0.011	0.092 ± 0.004
DT-GP	0.722 ± 0.008	0.893 ± 0.008	0.084 ± 0.003
DDT-GP	0.720 ± 0.003	0.894 ± 0.009	0.083 ± 0.001
DDT-CP	0.755 ± 0.002	0.901 ± 0.011	0.114 ± 0.002
DDT-TD	0.742 ± 0.006	0.866 ± 0.012	0.101 ± 0.001
BCTT	$\mathbf{0.698 \pm 0.002}$	$\mathbf{0.777 \pm 0.016}$	$\mathbf{0.084 \pm 0.001}$

(a) $R = 3$

RMSE	*MovieLens*	*AdsClicks*	*DBLP*
CT-CP	1.165 ± 0.008	1.324 ± 0.013	0.263 ± 0.006
CT-GP	0.965 ± 0.019	1.410 ± 0.015	0.227 ± 0.007
DT-GP	0.949 ± 0.007	1.425 ± 0.015	0.225 ± 0.008
DDT-GP	0.948 ± 0.005	1.421 ± 0.012	0.220 ± 0.006
DDT-CP	1.141 ± 0.007	1.623 ± 0.013	0.282 ± 0.011
DDT-TD	0.944 ± 0.003	1.453 ± 0.035	0.312 ± 0.072
BCTT	$\mathbf{0.895 \pm 0.007}$	$\mathbf{1.304 \pm 0.018}$	$\mathbf{0.202 \pm 0.009}$
MAE			
CT-CP	0.835 ± 0.006	0.792 ± 0.007	0.128 ± 0.001
CT-GP	0.717 ± 0.012	0.883 ± 0.016	0.092 ± 0.002
DT-GP	0.714 ± 0.005	0.886 ± 0.012	0.084 ± 0.001
DDT-GP	0.707 ± 0.004	0.882 ± 0.015	0.082 ± 0.003
DDT-CP	0.843 ± 0.003	1.082 ± 0.013	0.141 ± 0.004
DDT-TD	0.712 ± 0.002	0.903 ± 0.024	0.221 ± 0.047
BCTT	$\mathbf{0.679 \pm 0.001}$	$\mathbf{0.785 \pm 0.010}$	$\mathbf{0.080 \pm 0.001}$

(b) $R = 7$

THE UNIVERSITY OF UTAH

Thanks for attention
Q&A Time

Presenter' email: shikai.fang@utah.edu

Focus: Bayesian machine learning, tensor learning