THE UNIVERSITY OF UTAH

Bayesian Tensor Decomposition: Dynamics and Sparsity

Presenter: 方樹楷 Shikai Fang

School of computing, The University of Utah

For ZJU talk

Outline

1. **Background**

2. Tensor learning via Bayesian Inference

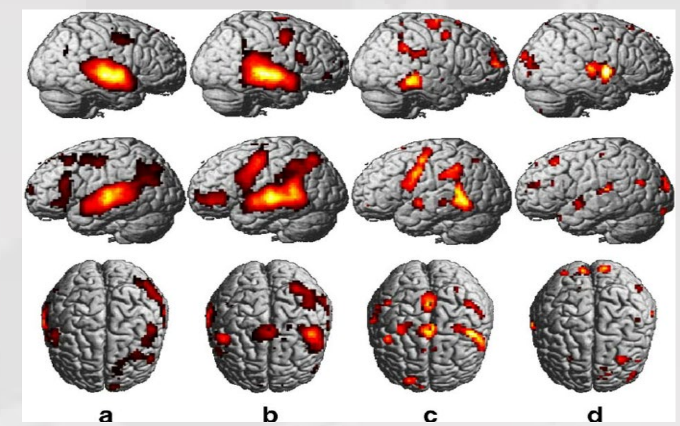
3. Dynamics in Tensor (ICML 2022 oral paper)

4. Sparsity in Tensor(UAI & ICML 2021 paper)

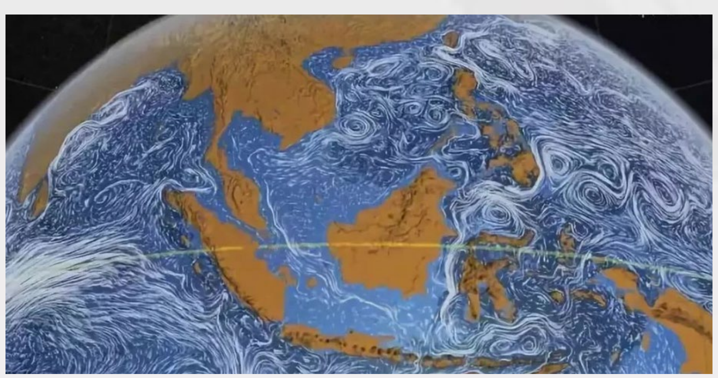
<u>Tensor Data</u>: Widely Used High-Order Data Structures to Represent **Interactions** of **Multiple Objects/Entities**

(user, movie, episode)

(user, advertisement, page-section)



(subject, voxel, electrode)



(location, region, time, climate)

(user, location, message-type)

(patient, gene, condition)

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

- Simple case:
  Collaborative Filtering (Matrix Factorization)

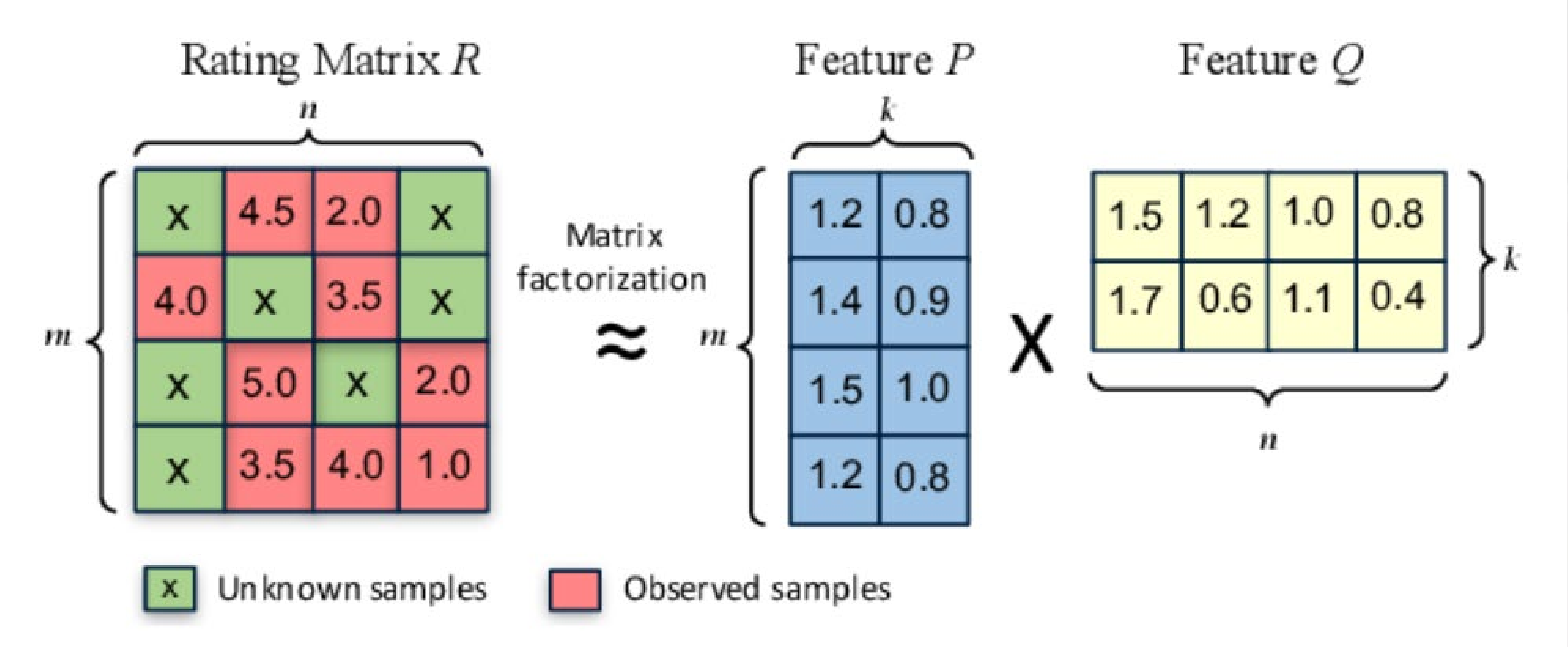


Image from https://www.slideshare.net/hontolab/matrix-factorization-192159058

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

- Simple case:
  Collaborative Filtering (Matrix Factorization)

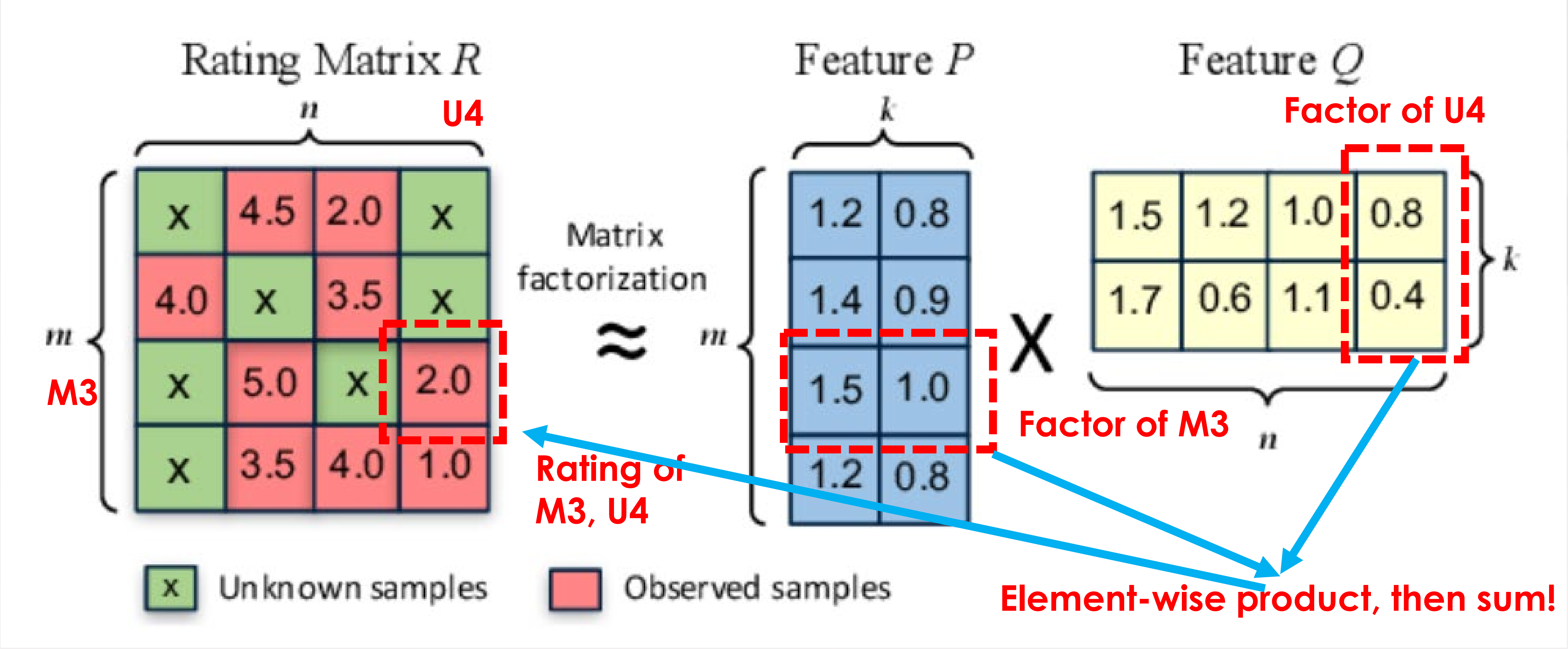


Image from https://www.slideshare.net/hontolab/matrix-factorization-192159058

From matrix to tensor: **CP decomposition**

- Example of three-mode tensor:

$$\mathcal{X} \in \mathbb{R}^{I \times J \times K}, A \in \mathbb{R}^{I \times R}, B \in \mathbb{R}^{J \times R}, C \in \mathbb{R}^{K \times R}$$

$$x_{ijk} \approx \underbrace{\sum_{r=1}^{R} A_{ir} B_{jr} C_{kr}}_{\text{Element-wise product, then sum!}} \text{ for } i = 1, \ldots, I, j = 1, \ldots, J, k = 1, \ldots, K$$

Interaction Records

user	item	page	purchase
100	25	35	1
23	21	56	0
100	25	35	1
..	..		..
32	33	46	0

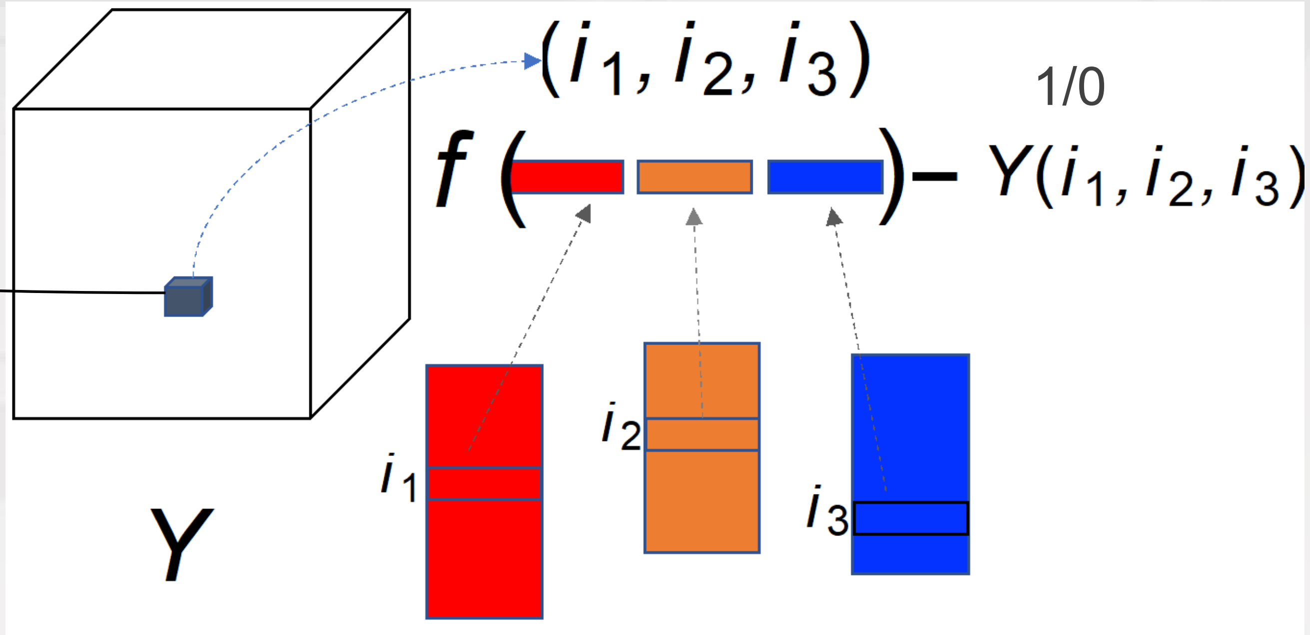


Image from Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, *51*(3), 455-500.

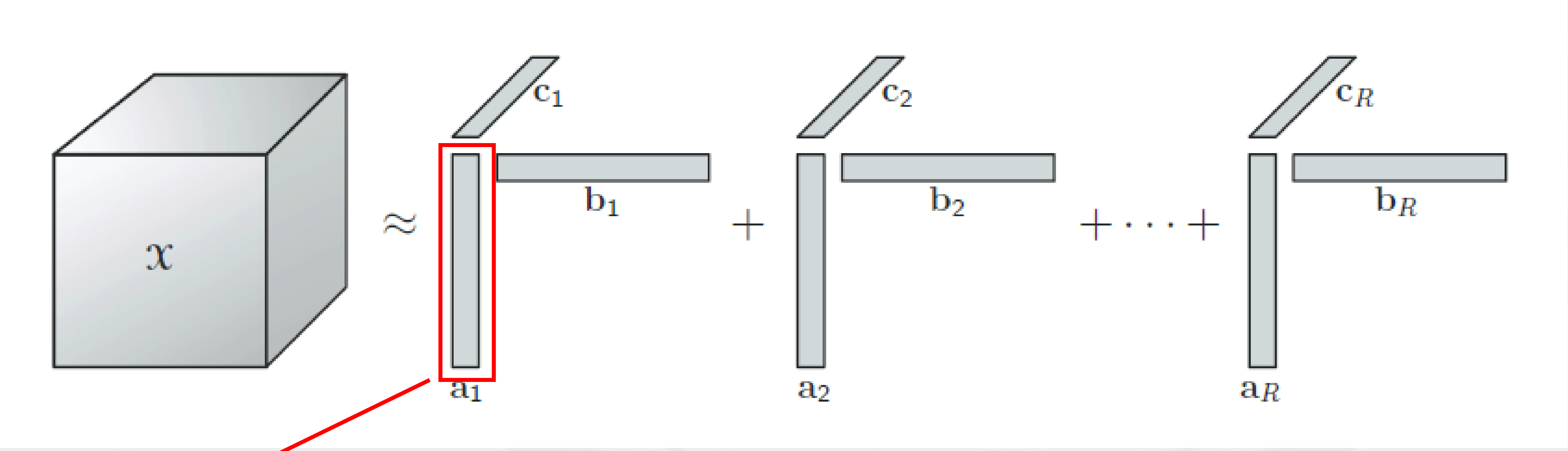
THE UNIVERSITY OF UTAH

Outline

1. Background
2. **Tensor learning via Bayesian Inference**
3. Dynamics in Tensor (ICML 2022 oral paper)
4. Sparsity in Tensor(UAI & ICML 2021 paper)

- Make latent factor from variable to a **distribution**
- Turns Deterministic model -> **Bayesian model**

Uncertainty really counts!



Deterministic version: $a_k \in \mathbb{R}^{d_k}$

Probabilistic version: $a_k \sim \mathcal{N}\left(\mathbf{u}_s^k \mid \mathbf{u}_s^k, v\mathbf{I}\right), \mathbf{u}_s^k \in \mathbb{R}^{d_k}$

Image from Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, *51*(3), 455-500.

- Bayesian version for CP decomposition,
- the **joint distribution** is:

Observed data, factors, noise

Prior of the noise

Gauss Prior of latent factors

$$p\left(\boxed{\{y_{\mathbf{i}}\}_{\mathbf{i}\in S}}, \boxed{\mathcal{U}}, \boxed{\tau}\right) = \boxed{\mathrm{Gam}\left(\tau \mid a_0, b_0\right)} \prod_{k=1}^{K} \prod_{s=1}^{d_k} \boxed{\mathcal{N}\left(\mathbf{u}_s^k \mid \mathbf{m}_s^k, v\mathbf{I}\right)}$$

$$\prod_{i\in S} \boxed{\mathcal{N}\left(y_{\mathbf{i}} \mid \mathbf{1}^\top \left(\mathbf{u}_{i_1}^1 \circ \ldots \circ \mathbf{u}_{i_K}^K\right), \tau^{-1}\right)}$$

Gauss likelihood of the prediction

Final goal : Get the **exact posterior distribution :**

$$p\left(\mathcal{U}, \tau \mid \{y_{\mathbf{i}}\}_{\mathbf{i}\in S}\right) = \frac{p\left(\{y_{\mathbf{i}}\}_{\mathbf{i}\in S}, \mathcal{U}, \tau\right)}{\boxed{p\left(\{y_{\mathbf{i}}\}_{\mathbf{i}\in S}\right)}}$$

Distribution of data, Constant, we never know!!

- Exact posterior distribution p is always intractable
- Approximation it by a **tractable distribution: q**

$$p\left(\mathcal{U}, \tau \mid \{y_{\mathbf{i}}\}_{\mathbf{i} \in S}\right) \approx q(\mathcal{U}, \tau) = q(\tau) \prod_{k=1}^{K} \prod_{s=1}^{d_k} q\left(\mathbf{u}_s^k\right)$$

$$= \operatorname{Gamma}(\tau \mid \boxed{a^\star, b^\star}) \prod_{k=1}^{K} \prod_{s=1}^{d_k} \mathcal{N}\left(\mathbf{u}_s^k \mid \boxed{\boldsymbol{\mu}_s^{k\,\star}, \boldsymbol{\Sigma}_s^{k\,\star}}\right)$$

Parameters needed to be optimized
to make the approximation accurate !!

Become a problem of **distribution match !**

- Key point in Bayesian machine learning

$$p\left(\{y_{\mathbf{i}}\}_{\mathbf{i}\in S}, \mathcal{U}, \tau\right) = \mathrm{Gam}\left(\tau \mid a_0, b_0\right) \prod_{k=1}^{K} \prod_{s=1}^{d_k} \mathcal{N}\left(\mathbf{u}_s^k \mid \mathbf{m}_s^k, v\mathbf{I}\right)$$

$$\prod_{\mathbf{i}\in S} \mathcal{N}\left(y_{\mathbf{i}} \mid \boxed{f\left(\mathbf{u}_{i_1}^1, \ldots, \mathbf{u}_{i_K}^K\right)}, \tau^{-1}\right)$$

$$p\left(\mathcal{U}, \tau \mid \{y_{\mathbf{i}}\}_{\mathbf{i}\in S}\right) \approx q(\mathcal{U}, \tau) = q(\tau) \prod_{k=1}^{K} \prod_{s=1}^{d_k} q\left(\mathbf{u}_s^k\right)$$

$$= \mathrm{Gamma}(\tau \mid a^\star, b^\star) \prod_{k=1}^{K} \prod_{s=1}^{d_k} \mathcal{N}\left(\mathbf{u}_s^k \mid \boldsymbol{\mu}_s^{k\,\star}, \boldsymbol{\Sigma}_s^{k\,\star}\right)$$

- Complete toolbox, but no silver bullet.
- Depend on the model **f** picked for the likelihood

Most-frequently used tool:
Variational inference(VI): minimize the **KL divergence** of two distributions

$$\mathcal{L} = \int q^*(\mathcal{U}, \tau) \log \frac{p\left(\{y_i\}_{i \in S_t} \mid \mathcal{U}, \tau\right) q(\mathcal{U}, \tau)}{q^*(\mathcal{U}, \tau)} \mathrm{d}\mathcal{U} \mathrm{d}\tau$$

- **Expectation propagation(EP):**

moment match when L(reverse order) has **closed form solution**

- **Assumed density filtering(ADF):**

moment match when having **tractable normalization term**

- **Reparameterization trick (SVI):**

Probabilistic version SGD when **L is totally intractable**

…(sampling based methods)

THE UNIVERSITY OF UTAH

Series of work from our group under such architecture.

Short Names	Likelihood Model & Prior	Inference Method	Publication
POST[1]	CP decomposition/multi-linear	EP	ICDM 2018
POND[2]	Neural-kernel Gaussian process	SVI	ICDM 2020
SBTD[3]	Bayesian neural network + **Sparse**	ADF (**Streaming**)	ICML 2021
BASS[4]	Tucker decomposition + **Sparse**	ADF (**Streaming**)	UAI 2021
BCTT[5]	Tucker decomposition + **Dynamics**	CEP + KF/RTS	ICML 2022

[1]Du, Yishuai, et al. "Probabilistic streaming tensor decomposition." *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018.
[2]Conor Tillinghast, **Shikai Fang**, Kai Zheng, and Shandian Zhe, "Probabilistic Neural-Kernel Tensor Decomposition", IEEE International Conference on Data Mining (ICDM), 2020.
[3] **Fang, Shikai**, et al. "Streaming Probabilistic Deep Tensor Factorization." The Thirty-eighth International Conference on Machine Learning (ICML), 2021
[4] **Fang, Shikai**, et al. "Bayesian Streaming Sparse Tucker decomposition."*Conference on Uncertainty in Artificial Intelligence*. UAI, 2021.
[5] **Fang, Shikai**, et al. "Bayesian Continues-Time Tucker Decomposition." The Thirty-ninth International Conference on Machine Learning (ICML), 2022

THE UNIVERSITY OF UTAH

Outline

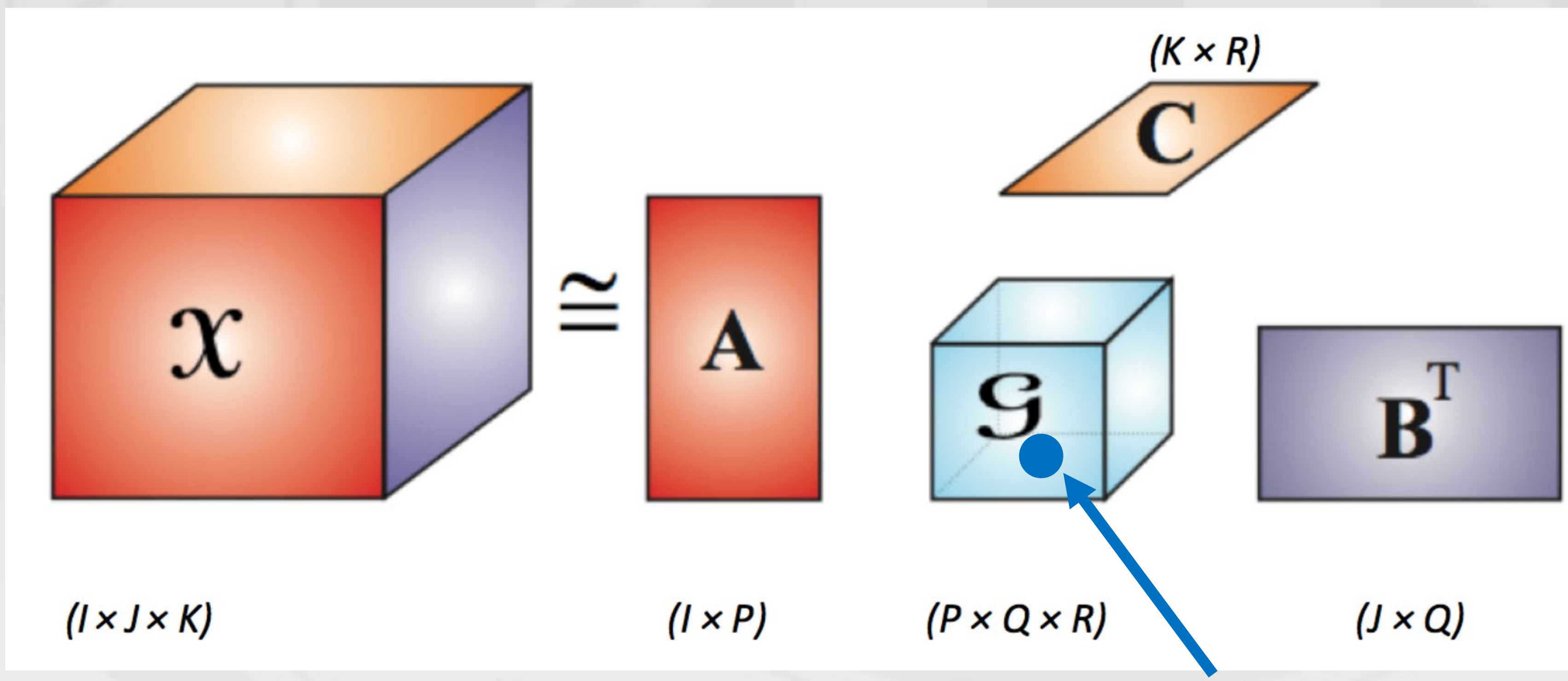
1. Background
2. Tensor learning via Bayesian Inference
3. **Dynamics in Tensor (ICML 2022 oral paper)**
4. Sparsity in Tensor(UAI & ICML 2021 paper)

Shikai Fang, Akil Narayan, Robert M. Kirby, and Shandian Zhe, "Bayesian Continuous-Time Tucker Decomposition" (Oral), The 39th International Conference on Machine Learning (ICML), 2022

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:

$$y_{\mathbf{i}} \approx \boxed{\mathrm{vec}(\mathcal{W})^\top}\left(\mathbf{u}^1_{i_1}\otimes\dots\otimes\mathbf{u}^K_{i_K}\right)$$
$$= \sum_{r_1=1}^{R_1}\cdots\sum_{r_K=1}^{R_K}\left[\boxed{w_{(r_1,\dots,r_K)}}\cdot\boxed{\prod_{k=1}^{K} u^k_{i_k,r_k}}\right]$$

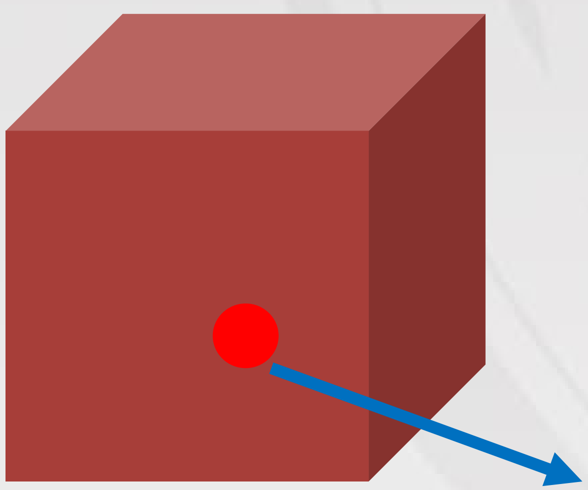
One Interaction of latent factors

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

Challenge: Temporal info in Tensor

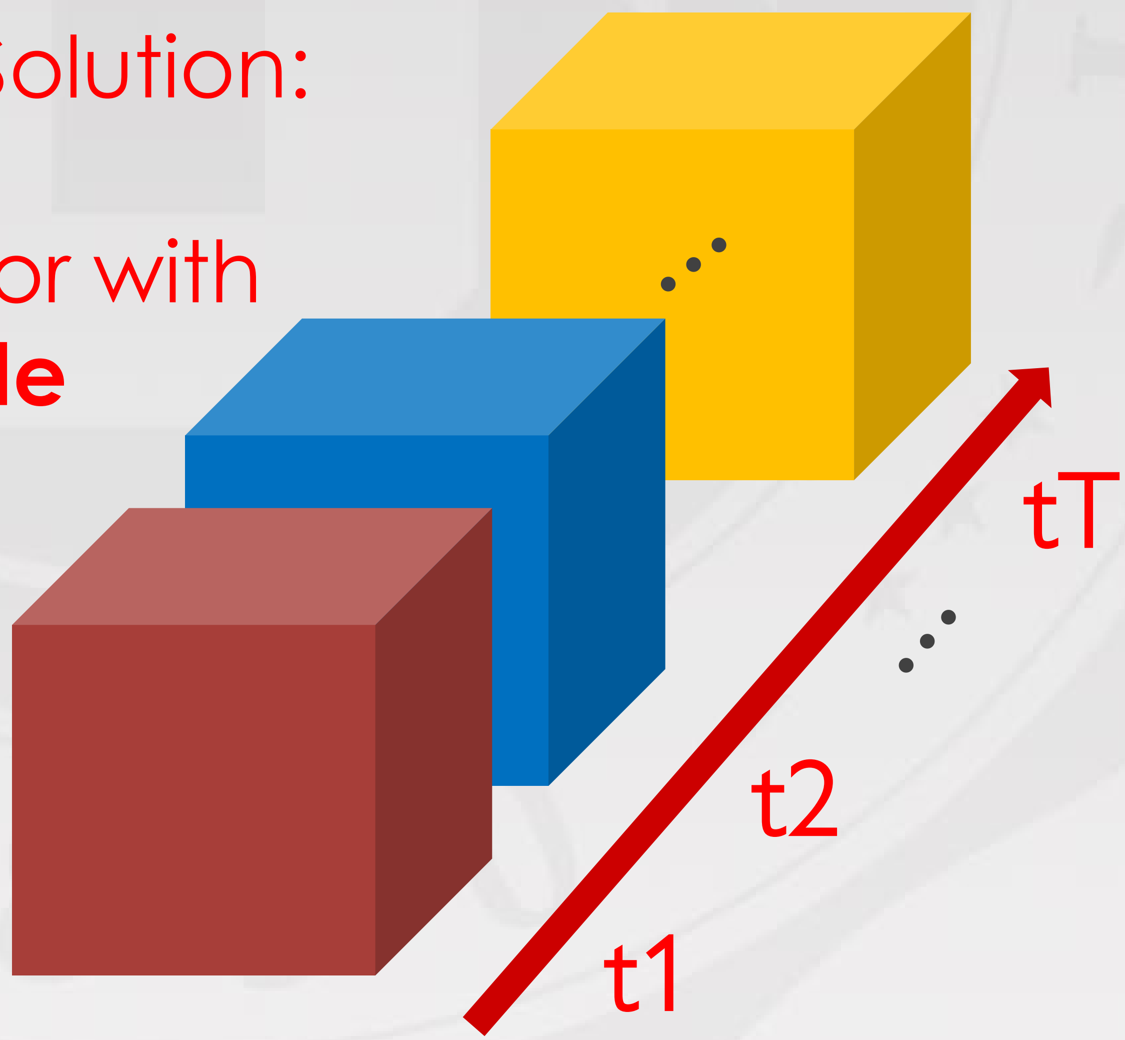
What about each entry is time-dependent?

Straightforward Solution:
- Drop time or
- Augment tensor with **time-step mode**



$X_{ijk}(t)$

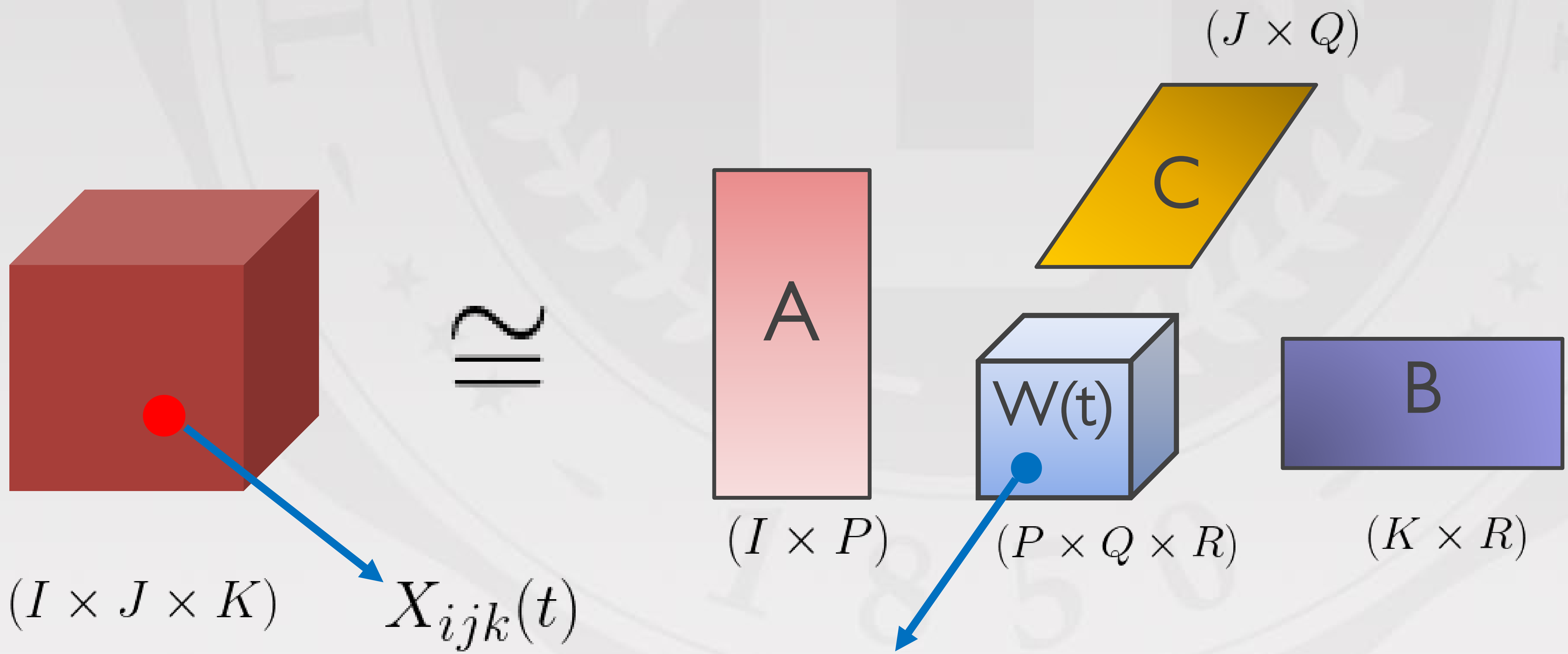
$(I \times J \times K)$



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

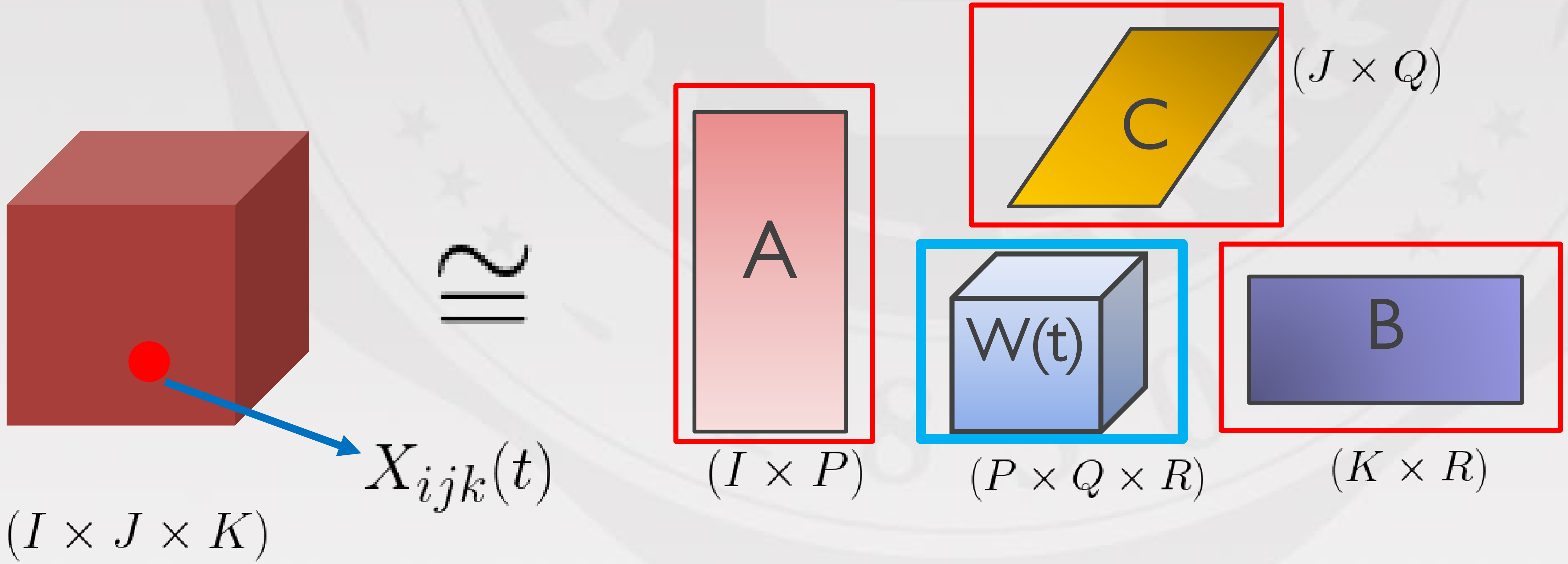
Problem:
1. Too Sparse
2. Ignore the temporary continuity

Our Solution: Modeling Dynamic Tucker Core by Temporal Gaussian Processes



$$W_{pqr}(t) \sim GP(0, k(t, t'))$$

High-level Motivation: <u>Decouple</u> 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

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

We apply a crucial fact:

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

Temporal GPs

with stationary kernel

Linear Time-Invariant(LTI) SDE

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

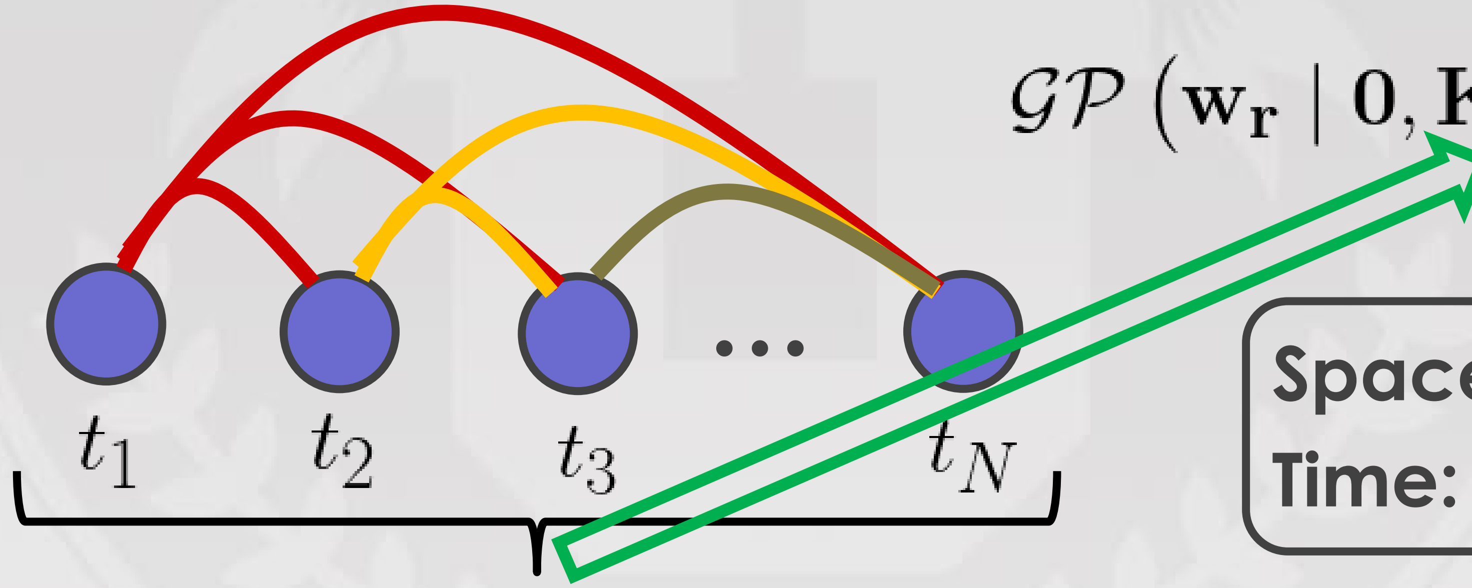
State Space Model (Gauss Markov Chain)

*: 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:



$\mathcal{GP}\left(\mathbf{w_r} \mid \mathbf{0}, \mathbf{K_r}(t, t')\right)$

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

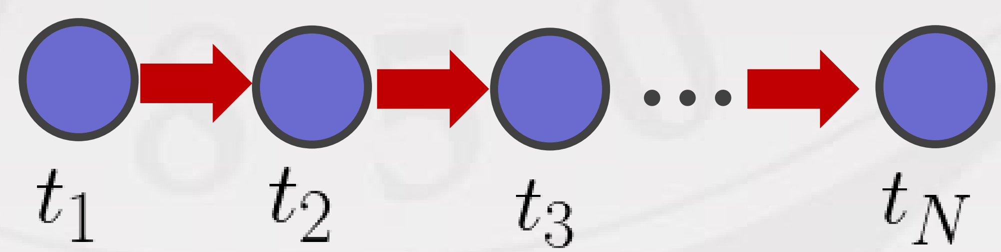
LTI-SDE

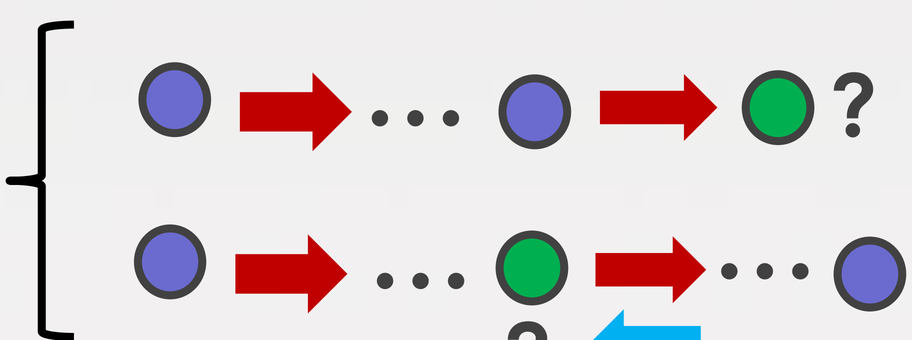
$$\frac{\mathrm{d}\gamma_{\mathbf{r}}(t)}{\mathrm{d}t} = \mathbf{F}\gamma_{\mathbf{r}} + \mathbf{L}\xi(t)$$

discrete form

State Space Model

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





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)

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

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**

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

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

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

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

Approx. Msg of Factors & noise

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

Approx. Msg of SDE states /Tucker core

Substitute these into joint prob.

THE UNIVERSITY OF UTAH

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!

THE UNIVERSITY OF UTAH

- **Conditional Moment Match**

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

- **Delta method:**

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

Enable ***<u>tractable moment matching</u>***
to update approx. probability terms under
<u>Expectation Propagation(EP)</u> 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:** $\mathcal{O}\left(N\left(\bar{R}^2 + \sum_{k=1}^{K} R_k^2\right)\right)$

THE UNIVERSITY OF UTAH

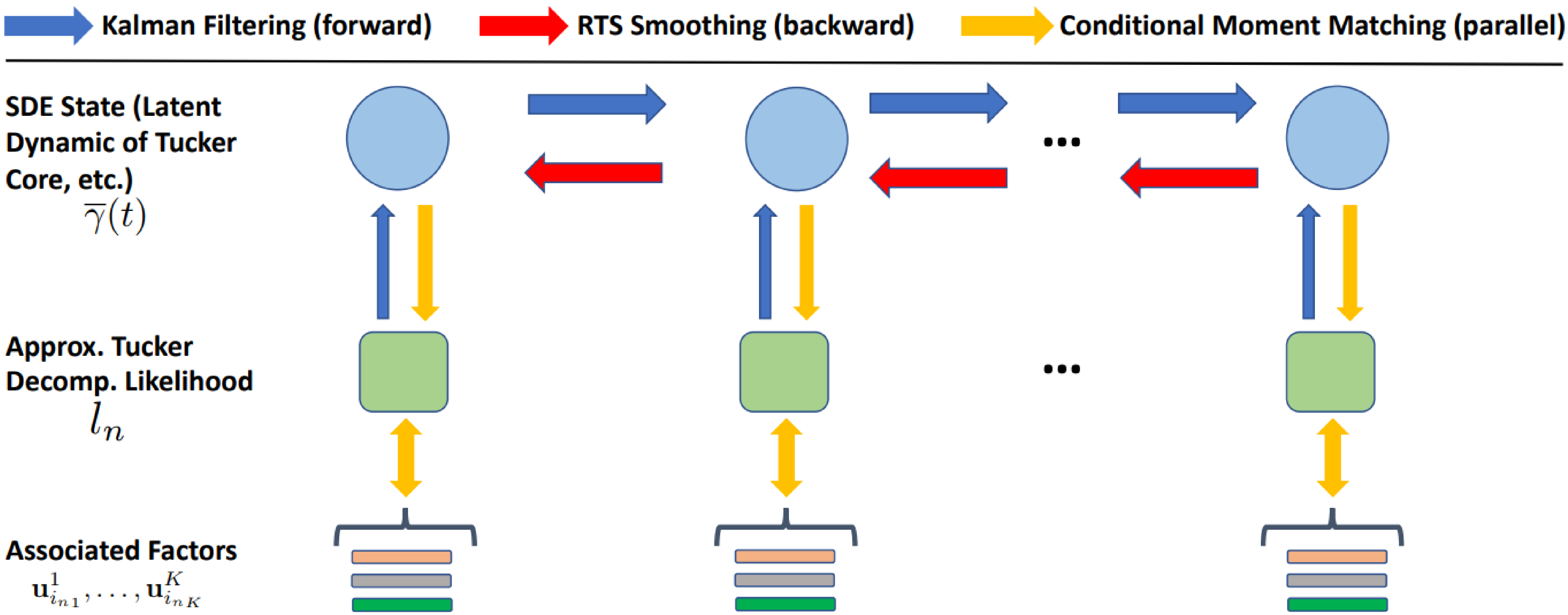
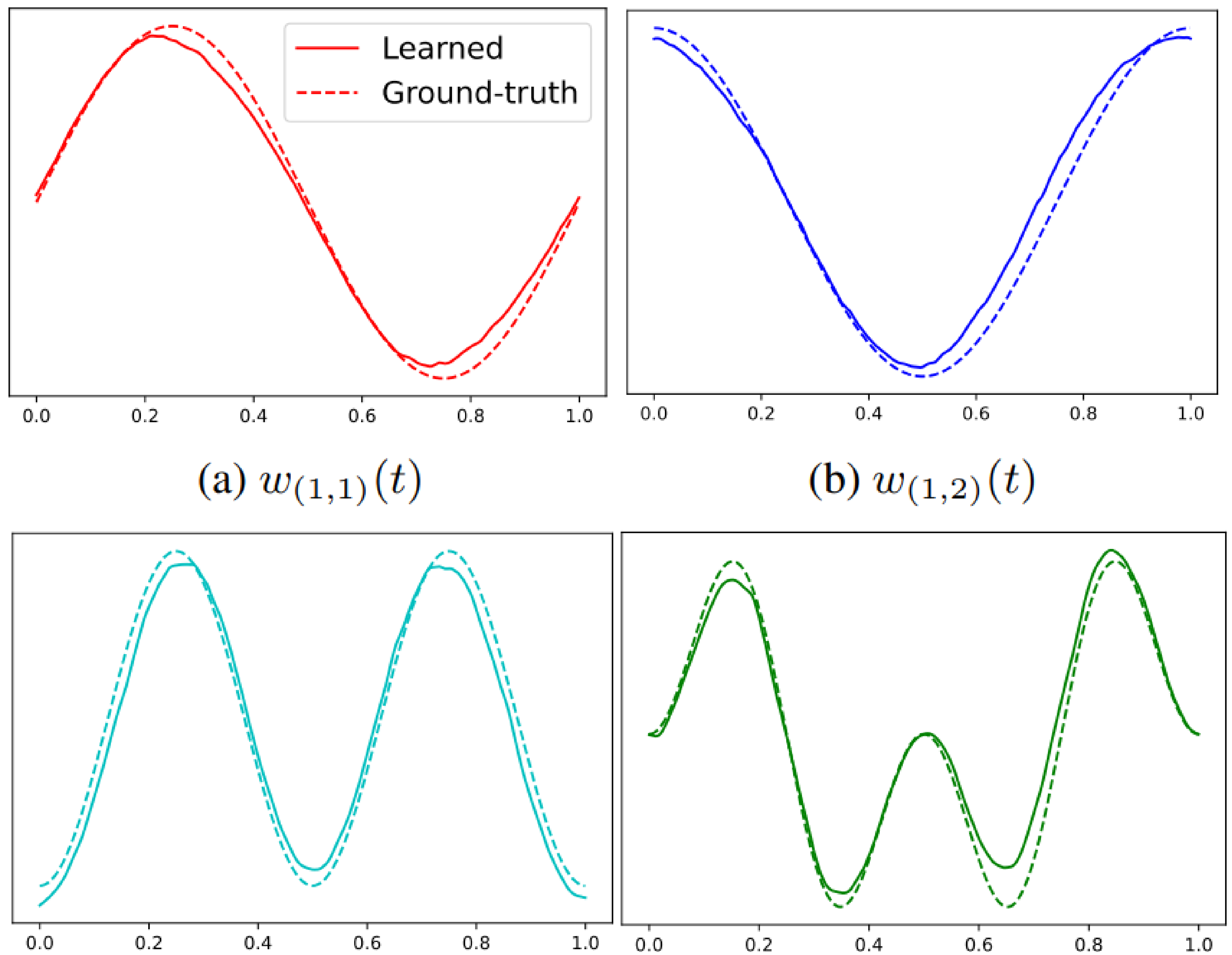


Figure 1. Graphical illustration of the message-passing inference algorithm.

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) $w_{(2,2)}(t)$

THE UNIVERSITY OF UTAH

Can BCTT capture the temporal patterns in tensor?

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

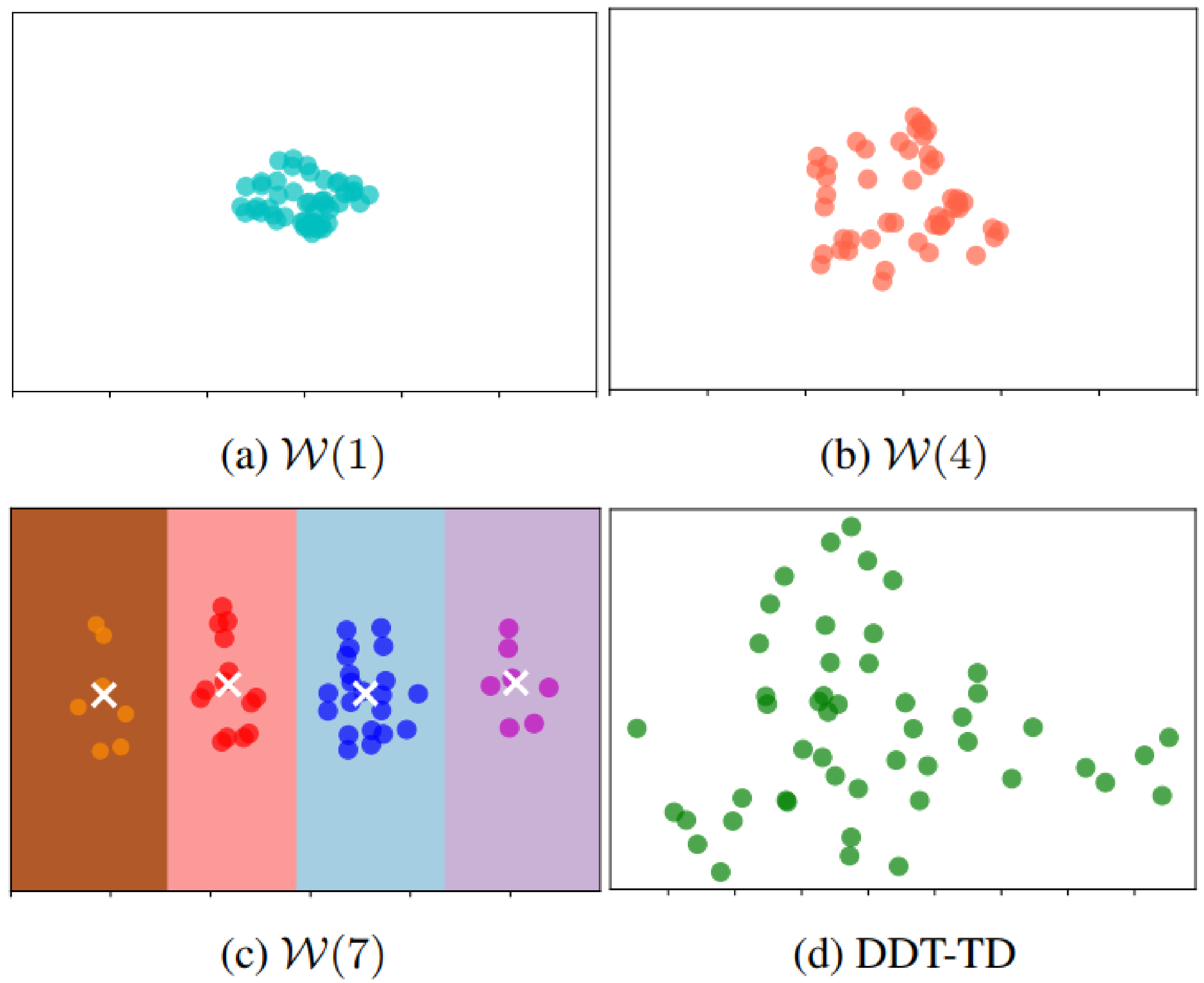


Figure 4. The structures of learned tensor-core at different time points by BCTT (a-c) and the static tensor-score learned by dynamic discrete-time Tucker decomposition (DDT-TD).

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

Outline

1. Background
2. Tensor learning via Bayesian Inference
3. Dynamics in Tensor (ICML 2022 oral paper)
4. **Sparsity in Tensor(UAI & ICML 2021 paper)**

Shikai Fang, Zheng Wang, Zhimeng Pan, Ji Liu, and Shandian Zhe, "Streaming Bayesian Deep Tensor Factorization", The Thirty-eighth International Conference on Machine Learning (ICML), 2021

Shikai Fang, Robert. M. Kirby, and Shandian Zhe, "Bayesian Streaming Sparse Tucker Decomposition", The 37th Conference on Uncertainty in Artificial Intelligence (UAI), 2021

- Sparsity in tensor data requires Sparsity in model

Tensor-Datasets	Size	#Observed entries	Observed Ratio
Gowalla	18737*1000*32510	821,931	0.0001%
SG	2321*5596*1600	105,764	0.0005%
ACC	3000*150*30000	1,220,000	0.1%
Movielens1M	6000*3700	1,000,000	4%

- otherwise, **overfitting** risk, especially for complex model like NN

- How non-Bayesian people get sparsity?

--L1 regular terms

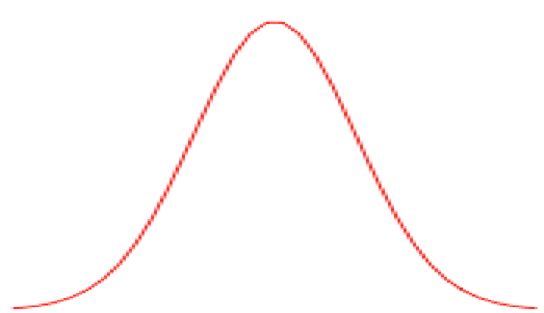
--overall sparse, not accurate enough

How Bayesian people get sparsity?

- **Spike and Slab Priors**

Introduce binary selection indicators $s_1, s_2, \ldots, s_d$
on each parameter ! -- **Element-wise sparse control!**

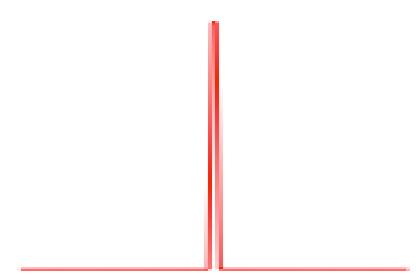
$s_j = 1$



Mild Regularize
(**slab**)

$s_{\mathbf{j}} \mathcal{N}\left(w_{\mathbf{j}} \mid 0, \sigma_0^2\right)$

$s_j = 0$

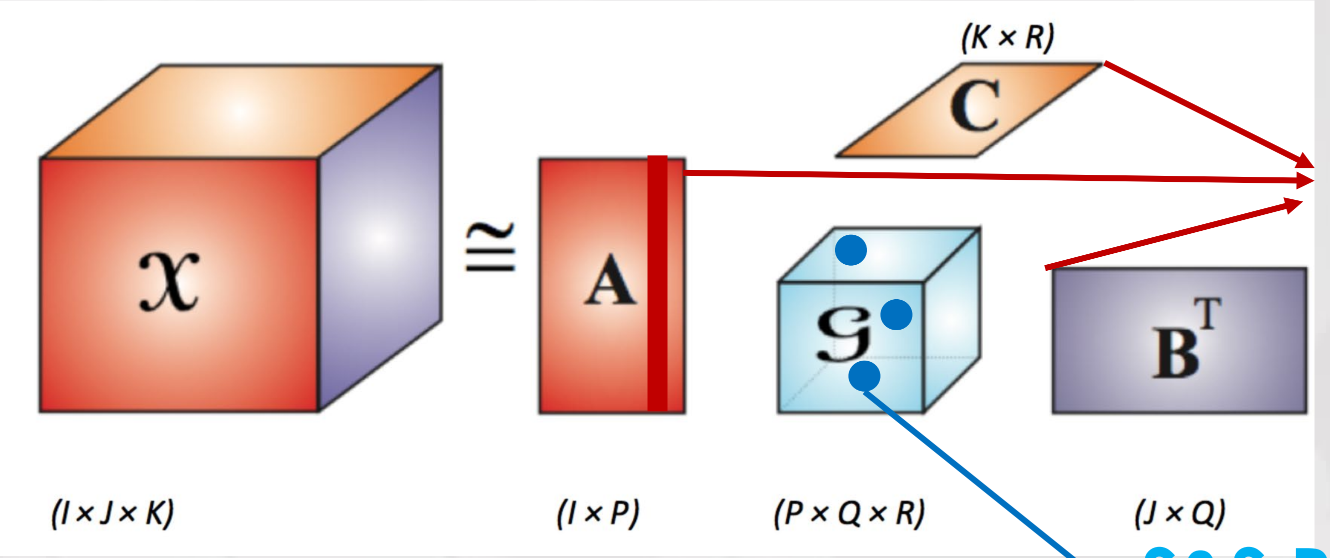


Strong shrinkage
(**spike**)

$\left(1 - s_{\mathbf{j}}\right) \delta\left(w_{\mathbf{j}}\right)$

Where to put such sparsity?

- **On Tucker Core– build a sparse core (BASS[4])**



Standard Gaussian prior

S&S Prior over each core tensor element

$$p(\mathcal{S},\mathcal{W},\mathcal{U},\mathcal{Y},\tau)=\prod_k\prod_j\mathcal{N}\left(\mathbf{u}_j^k \mid \mathbf{0},\mathbf{I}\right)\operatorname{Gam}\left(\tau \mid a_0,b_0\right)$$

$$\cdot\prod_{\mathbf{j}}\boxed{\operatorname{Bern}\left(s_{\mathbf{j}} \mid \rho_0\right)}\left(\boxed{s_{\mathbf{j}}\mathcal{N}\left(w_{\mathbf{j}} \mid 0,\sigma_0^2\right)+\left(1-s_{\mathbf{j}}\right)\delta\left(w_{\mathbf{j}}\right)}\right)$$

Binary indicators

Selective shrinkage priors on core based on indicator

$$\cdot\prod_{\mathbf{i}\in\mathcal{F}}\mathcal{N}\left(y_{\mathbf{i}} \mid \mathcal{W}\times_1\left(\mathbf{u}_{i_1}^1\right)^\top\times_2\ldots\times_K\left(\mathbf{u}_{i_K}^K\right)^\top,\tau^{-1}\right)$$

[5] **Fang, Shikai**, et al. "Bayesian Streaming Sparse Tucker decomposition."*Conference on Uncertainty in Artificial Intelligence*. UAI, 2021

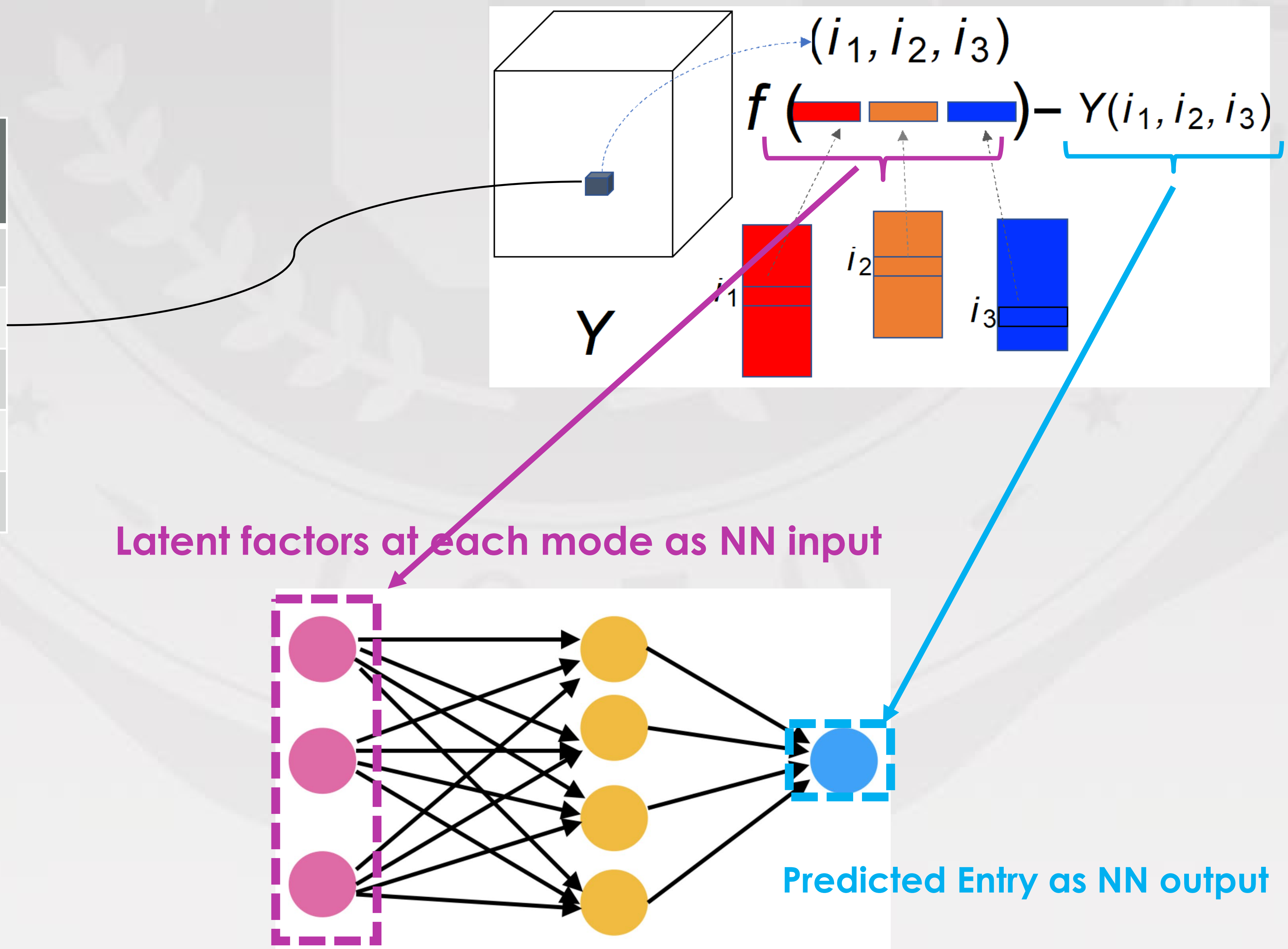
THE UNIVERSITY OF UTAH

Where to put such sparsity?

- **On NN weights – build a sparse BNN (SBDT[4])**

Interaction Records

user	item	page	purchase
100	25	35	1
23	21	56	0
100	25	35	1
..	..		..
32	33	46	0



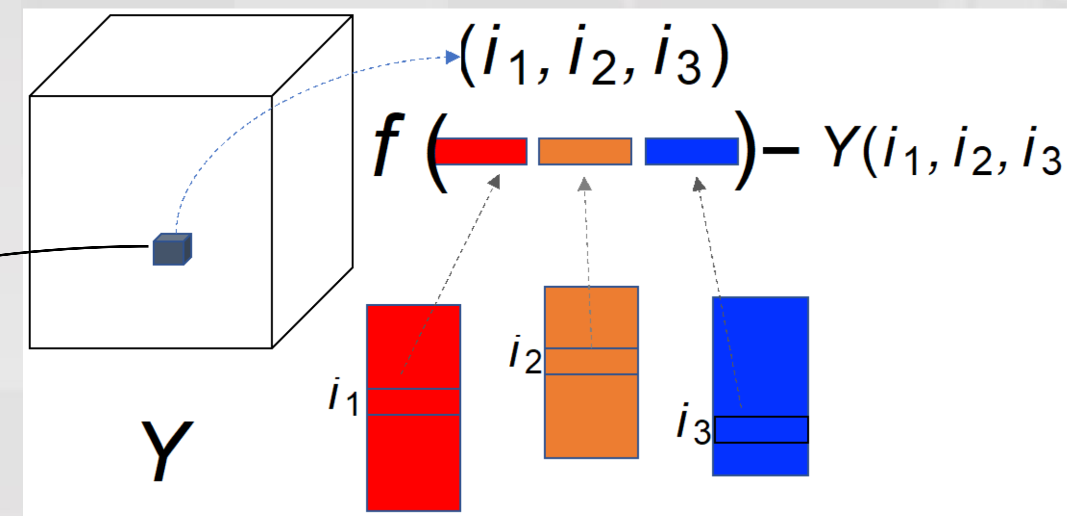
[4] **Fang, Shikai**, et al. "Streaming Probabilistic Deep Tensor Factorization." The Thirty-eighth International Conference on Machine Learning (ICML), 2021

Where to put such sparsity?

- **On NN weights – build a sparse BNN (SBDT[4])**

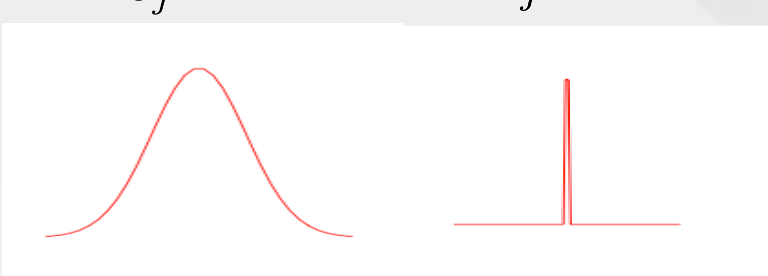
Interaction Records

user	item	page	purchase
100	25	35	1
23	21	56	0
100	25	35	1
..	..		..
32	33	46	0

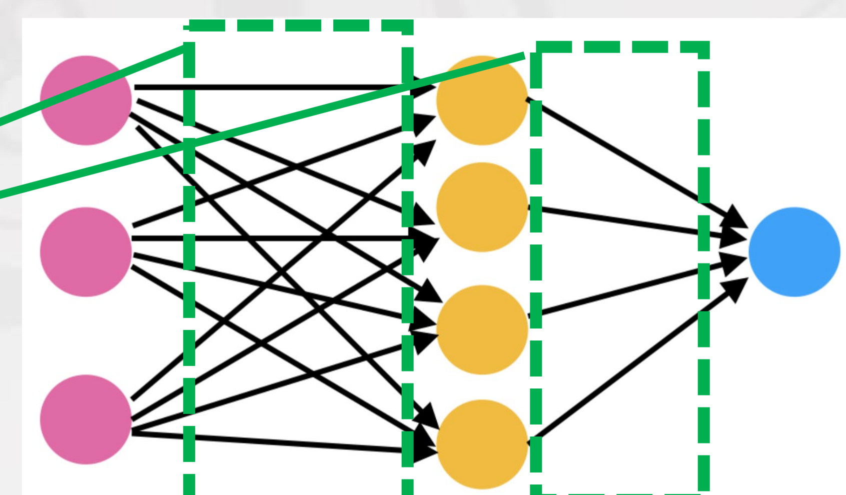


Assign s&s priors over each NN weights !

$s_j = 1$ $\quad$ $s_j = 0$



$s_{\mathbf{j}}\mathcal{N}\left(w_{\mathbf{j}} \mid 0, \sigma_0^2\right)$ $\quad$ $\left(1 - s_{\mathbf{j}}\right)\delta\left(w_{\mathbf{j}}\right)$



[4] **Fang, Shikai**, et al. "Streaming Probabilistic Deep Tensor Factorization." The Thirty-eighth International Conference on Machine Learning (ICML), 2021

How the final sparsity exactly look like?

- Approx. of S&S Priors in exponential family:
  Gaussian + Bernoulli

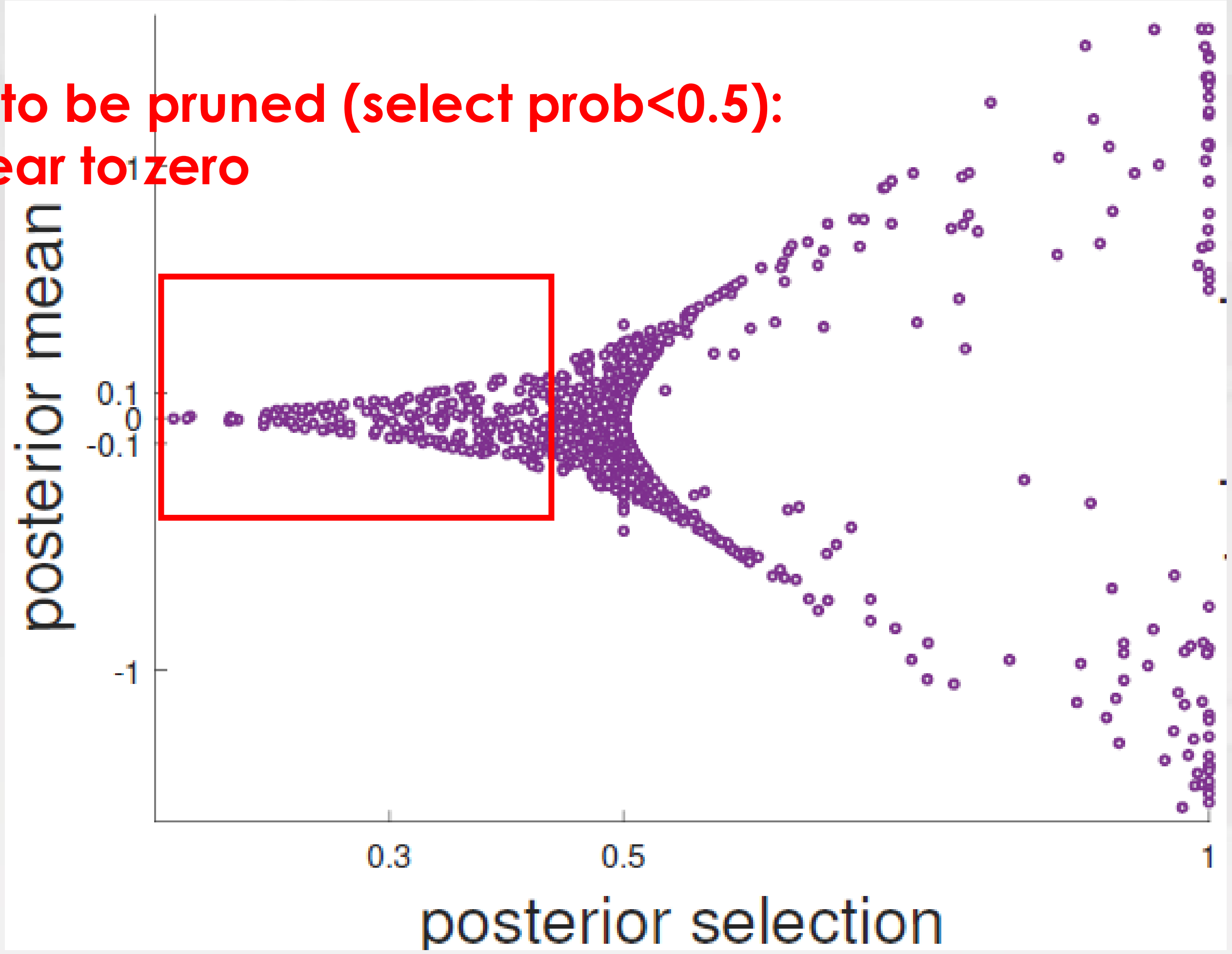
$$p\left(w_{mjt} \mid s_{mjt}\right) \propto A\left(w_{mjt}, s_{mjt}\right)$$
$$= \mathrm{Bern}\left(s_{mjt} \mid c\boxed{\left(\rho_{mjt}\right)}\right) \mathcal{N}\left(w_{mjt} \mid \mu_{mjt}^{0}, v_{mjt}^{0}\right)$$

Select posterior prob of each weight
<0.5: unselected <=> sparse

- **How sparse can model get? – Light model**

For SBDT work (Sparse BNN as factorization model)

- Plot of sparse-BNN weights after training
- Each weight has its posterior mean, var and selection prob.



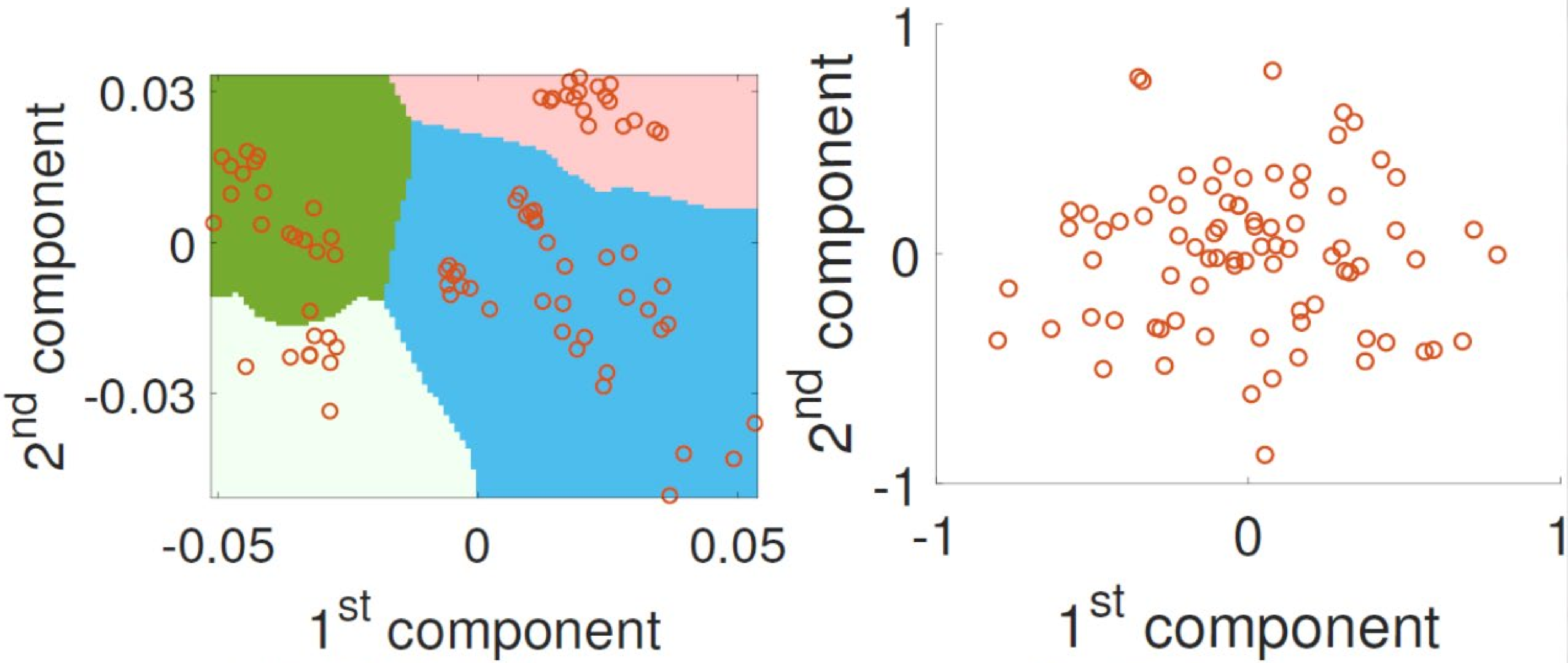
Image" **Fang, Shikai**, et al. "Streaming Probabilistic Deep Tensor Factorization." The Thirty-eighth International Conference on Machine Learning (ICML), 2021

THE UNIVERSITY OF UTAH

- **How sparse can model get? – Interpretability**

For BASS work (Tucker with Sparse core as factorization model)

- Plot of projected Tucker core elements with sparsity
- Significant structure of interactions



(a) BASS-Tucker (b) P-Tucker

Image: Fang, Shikai, et al. "Bayesian Streaming Sparse Tucker decomposition."*Conference on Uncertainty in Artificial Intelligence*. UAI, 2021.

How to make use of quantized uncertainty?

An example in ad-recommend system

- plots of model predictions on **CTR(click-through-rate) tensor dataset**

- **exploration and exploitation**, optimization policy for down steaming tasks

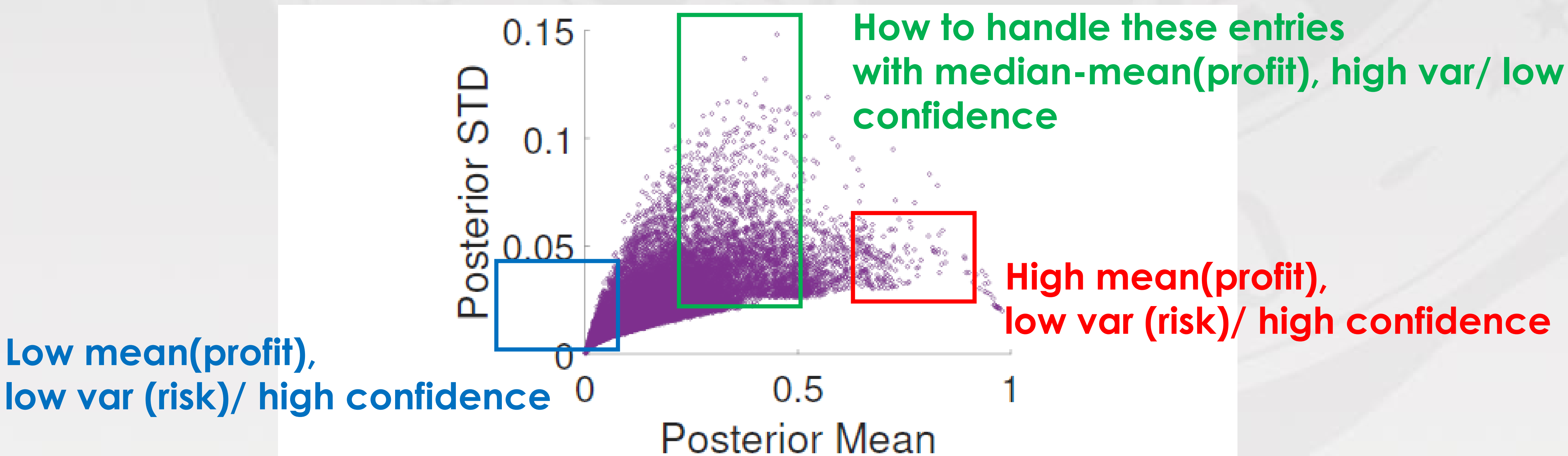


Figure 4: the posterior mean *v.s.* the posterior standard deviation (STD) of the click probability prediction.

Image from Conor Tillinghast, Shikai Fang, Kai Zheng, and Shandian Zhe, "Probabilistic Neural-Kernel Tensor Decomposition", IEEE International Conference on Data Mining (ICDM), 2020.

THE UNIVERSITY OF UTAH

Open questions for Cooperation

- Domain knowledge embedded in prior
- Make good use of the uncertainty measure
- Challenges and inspiration from real world
- …

Domain model(PDE/SDE) + AI4Science + new algos…

Thanks for attention
Q&A Time

Presenter' email: shikai.fang@utah.edu

Webpage: https://www.cs.utah.edu/~shikai/

Focus: Bayesian machine learning, tensor learning

知乎: 方轩固