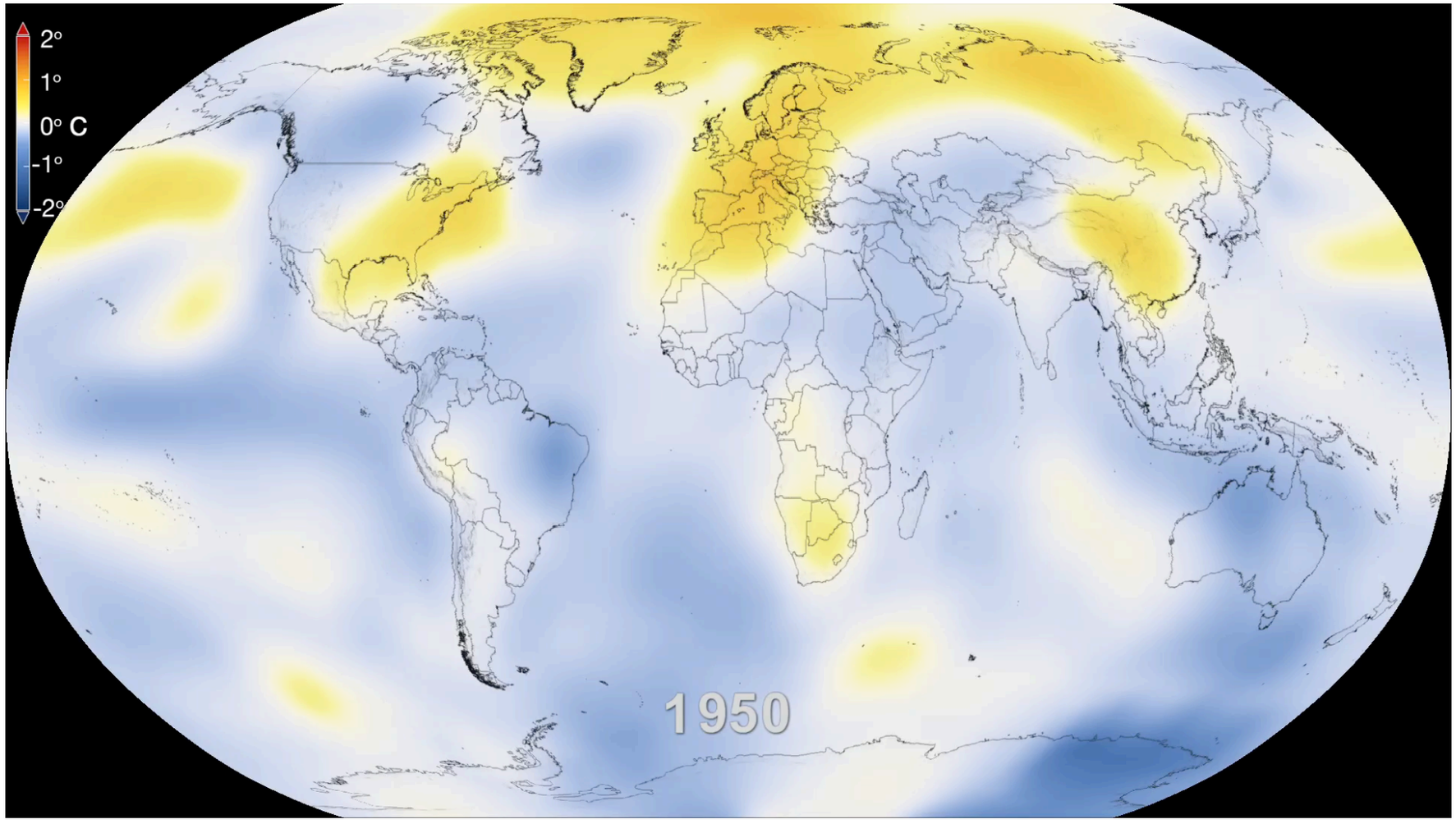
Tensor Methods for Large-Scale Spatiotemporal Learning

Rose Yu
Assistant Professor
Northeastern University

Predicting Global Climate

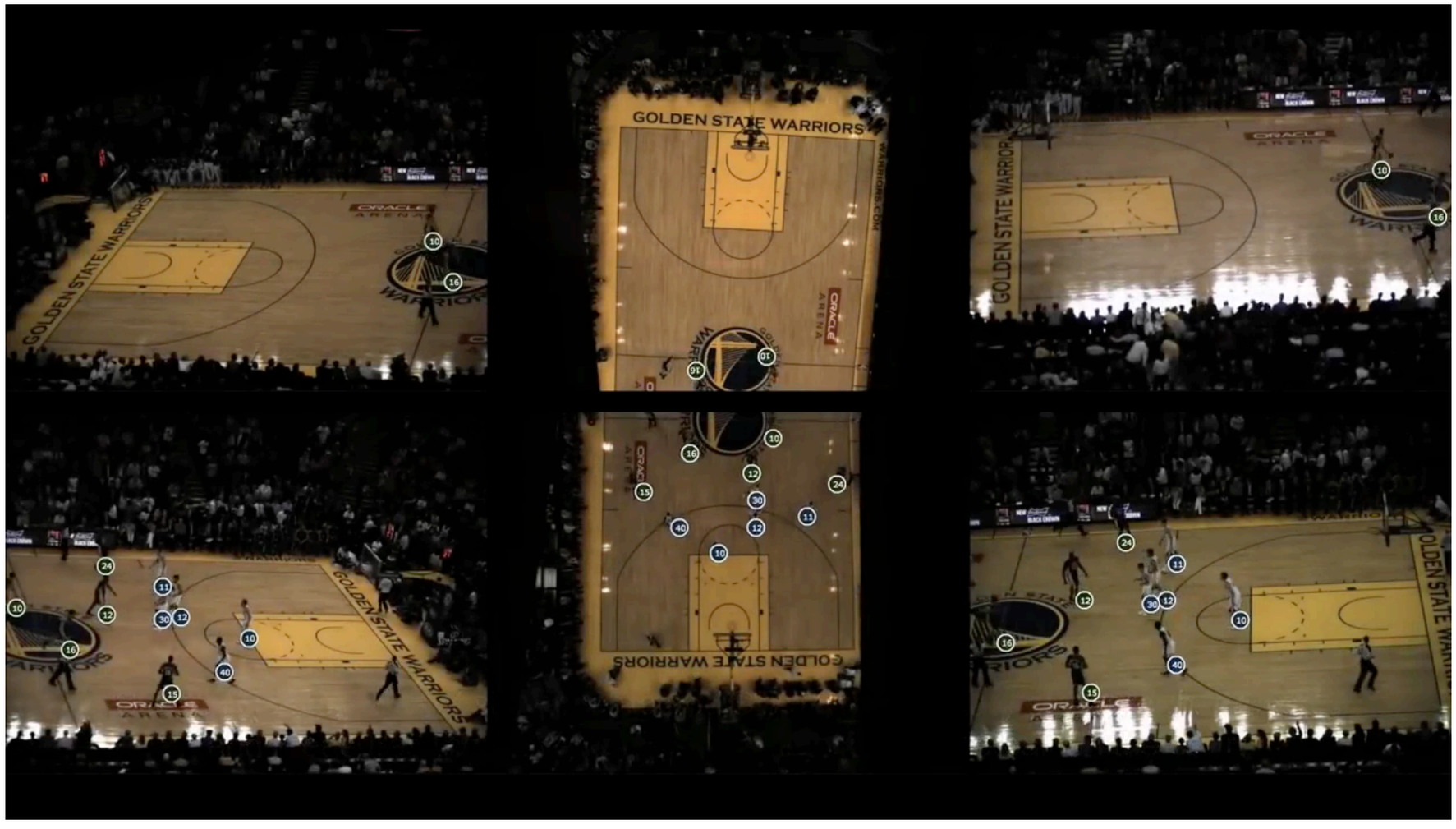
100,000 stations, 180 countries



credit: NASA

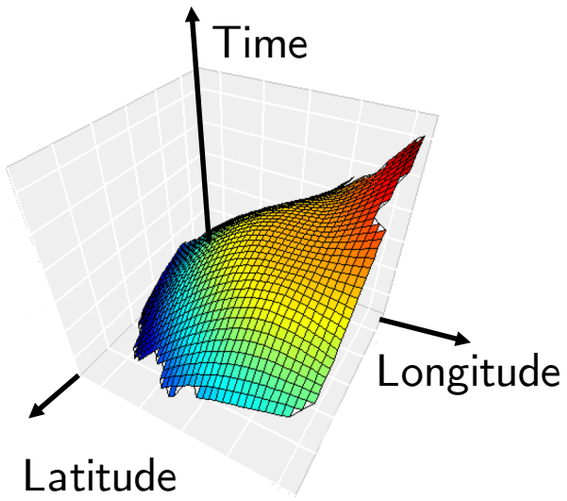
Modeling Basketball Play

2,000 events, 1.5 million data points

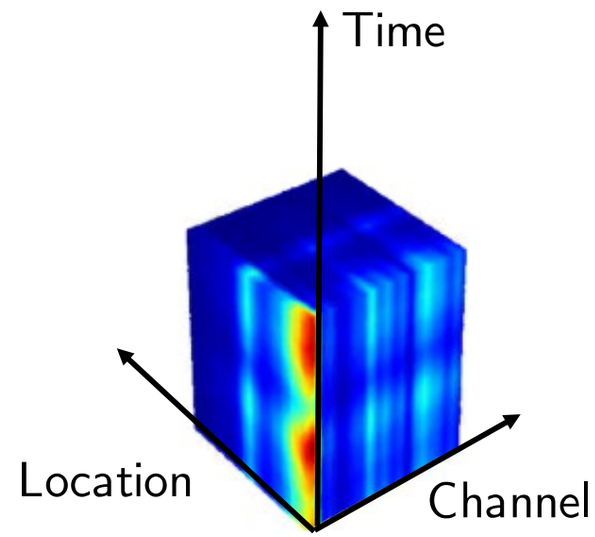


credit: STATS

Spatiotemporal Learning



climate measurements

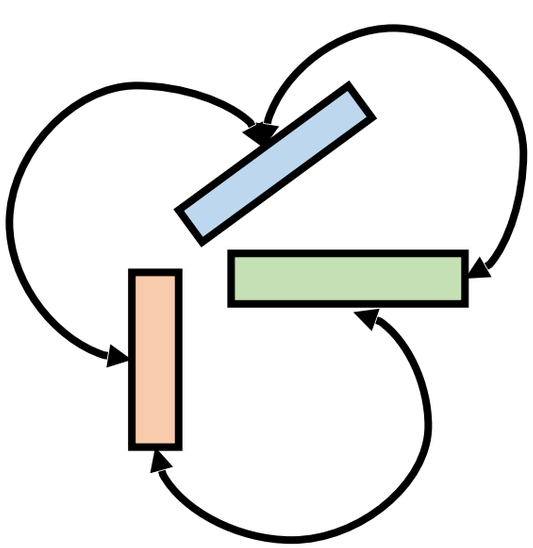


brain activities

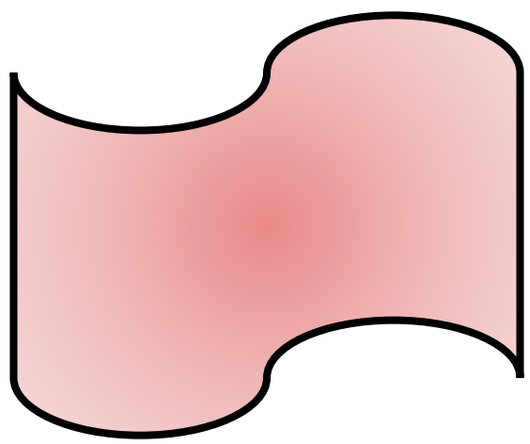
sports signals

- Make sense of large amount of data collected over **space** and **time**
- Enable **AI systems** to understand and reason in space and time
- Critical to **real-time decision making** in science and engineering.

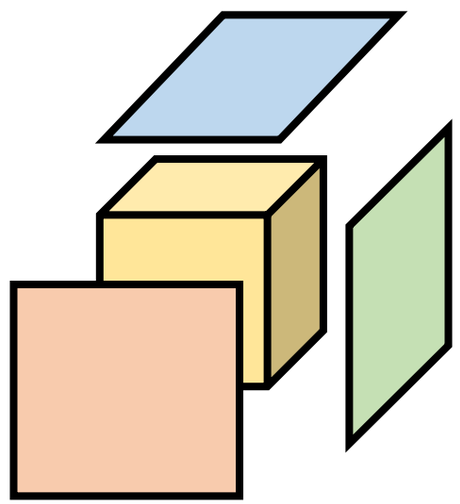
Tensor Methods



High-order Model



Nonlinear Map



Dimension Reduction

- Tensors can encode high-order dependency
- Multi-linear models are non-linear by nature
- Rich family of tensor models for dimension reduction

Tensor Methods for Machine Learning

Latent Variable Model

- Topic models [Anandkumar et al. 2014, Kuleshov et al 2015, Arabshahi et al. 2016]
- Hidden Markov models [Song et al. 2013, Huang et al. 2013, Kuznetsov et al. 2018]

Multi-task / Graph Learning

- Multi-relation/task learning [Nickle & Tresp 2013, Hoff 2015, Romera-Paredes 2013, Tomioka et al. 2015, Yu et al. 2016, Rabusseau et al, 2017]
- Hyper-graph partition [Ghoshdastidar 2015b, 2017, Chang et al. 2016]

Deep Learning

- Learning complexity analysis [Cohen et al. 2016, 2018, Sharir et al. 2016]
- Tensor neural network [Stoudenmire et al. 2016, Yu et al. 2017,Yang et al. 2017, Khrulkov 2019]

Reinforcement Learning

- POMDP [Azizzadenesheli et al. 2016, 2018]
- Stochastic optimal control [Gorodetsky et al. 2015, 2018]

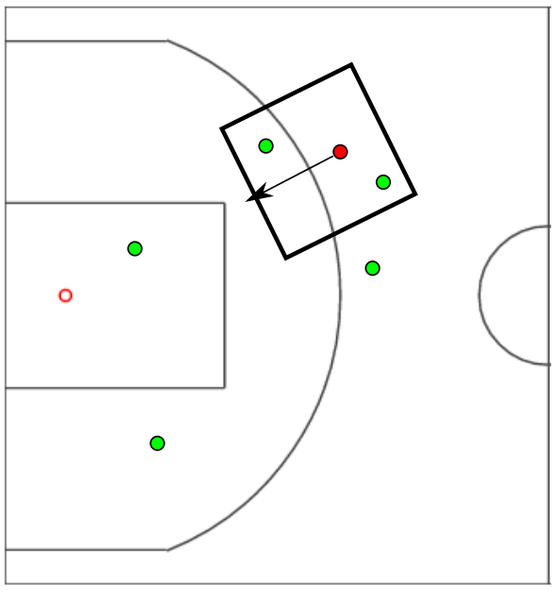
Multi-Resolution Tensor Learning

Stephan Zheng
Salesforce research

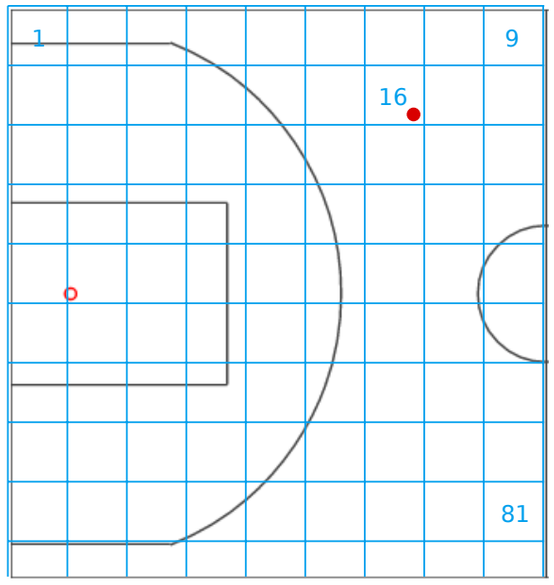
Yisong Yue
Caltech

Multi-resolution Tensor Learning for Large-Scale Spatial Data
Stephan Zheng, Rose Yu, Yisong Yue
Arxiv Preprint: https://arxiv.org/abs/1802.06825

Example: Field Goal Prediction



spatial field $\mathbf{x}$ and one-hot encoding of discretized grid $\phi(\mathbf{x})$



$$\phi(x) = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} 1 \\ \\ 16 \\ \\ \end{matrix}$$

- $x = (34.1, 40.7)$

continuous discrete

- Input: positions of the ball handler and defenders
- Output: likelihood of shooting
- Goal: learn the players shooting profiles in the presence of defenders

Tensor Latent Factor Model

- Tensor latent factor model assumes a multi-linear regression from positions to shooting likelihood

$$P(y_a = 1|\mathbf{x}) = f_a(\mathbf{x}) = \sum_{bc} \mathcal{W}_{abc}\phi_b(\mathbf{x})\psi_c(\mathbf{x}) + \mathbf{b}_a$$

- The weight tensor factorizes

$$\mathcal{W}_{abc} = \sum_k A_{ak}B_{bk}C_{ck}$$

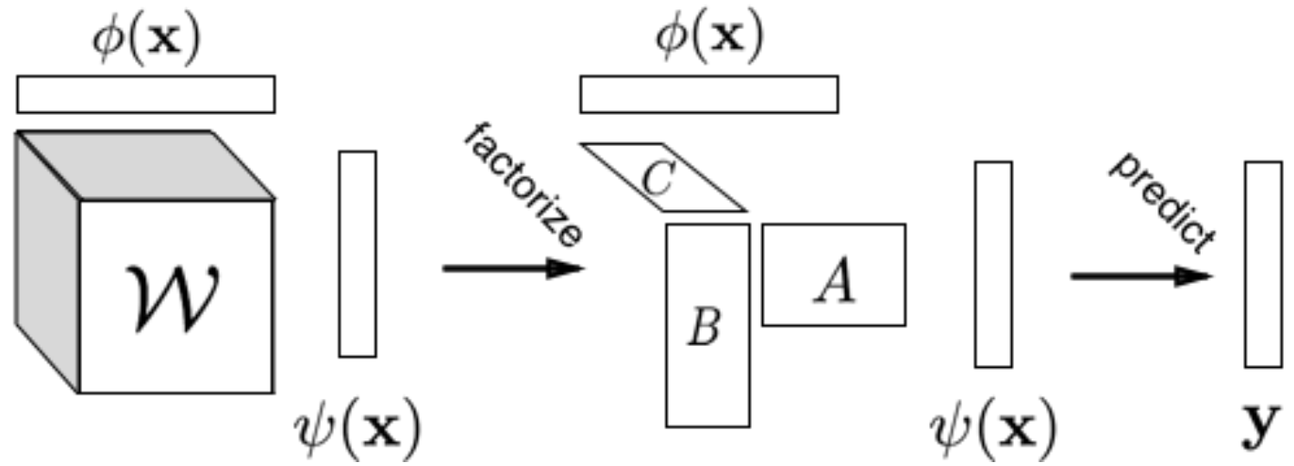
player shooting profiles | ball handler profiles | defender profiles

- Low-rank (spatially dense) plus sparse (spatially peaky)

$$\mathcal{W}_{abc} = \sum_k A_{ak}B_{bk}C_{ck} + \sum_k U_{ak}V_{bk}Z_{ck}$$

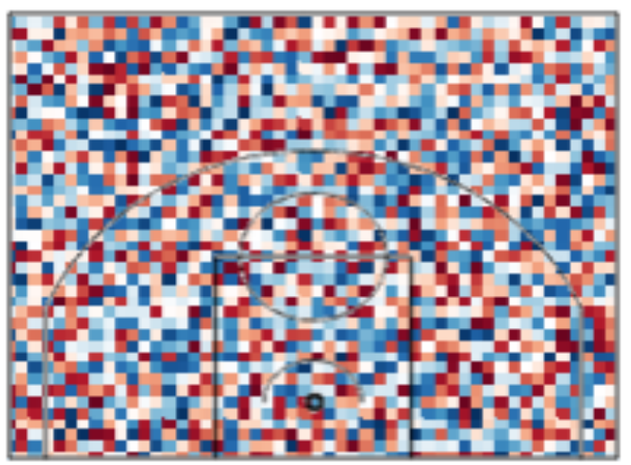
Difficulty of Training

$$\min_{W} -\sum_{a} y_a \log(f_a(x)) \qquad \text{s.t.} \quad f_a(\mathbf{x}) = \sum_{bc} \mathcal{W}_{abc}\phi_b(\mathbf{x})\psi_c(\mathbf{x}) + \mathbf{b}_a$$

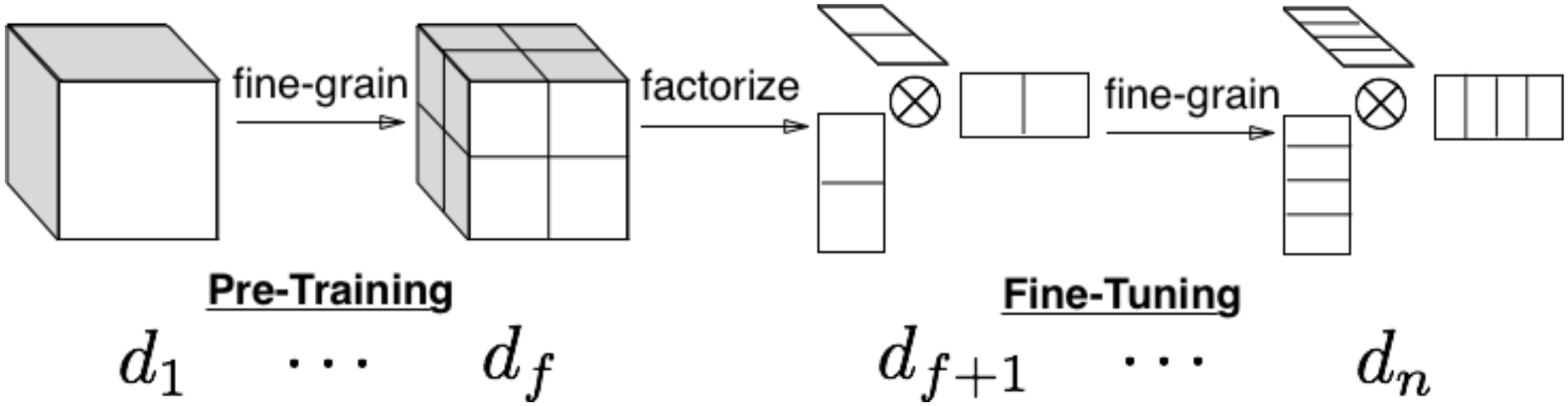


- High-Dimensional tensor models are computational expensive to train

- Non-convex objective: sensitive to initialization [Anandkumar et al. 2014], can yield uninterpretable latent factors



Multi-Resolution Learning



- **Initialize from factorization**: factorize a full-rank tensor and use the factors as initialization

- **Iterative fine-graining**: train a model at coarse resolution and iteratively increase the spatial resolution

Theoretical Analysis

- Assumption: the gradient-based algorithm is a contraction map

$$\mathcal{W}^{t+1} \leftarrow F(\mathcal{W}^t) \qquad \|F(\mathcal{W}) - F(\mathcal{W}')\| \leq \alpha \|\mathcal{W} - \mathcal{W}'\|$$

- Fine-graining criteria

$$\|\mathcal{W}^{t(d)} - \mathcal{W}^{t(d)-1}\| \leq \frac{C_0 d}{\alpha(1-\alpha)}$$

Lemma 1 (Nash, 2000) For each resolution level $[d_0, d_1, \cdots, d_n]$, there exists a constant C_1 and C_2 , such that the fixed point iteration with discretization size d has an estimation error:

$$\|F(\mathcal{W}) - F_d(\mathcal{W})\| \leq (C_1 + \alpha C_2 \|\mathcal{W}\|)d.$$

Theoretical Analysis

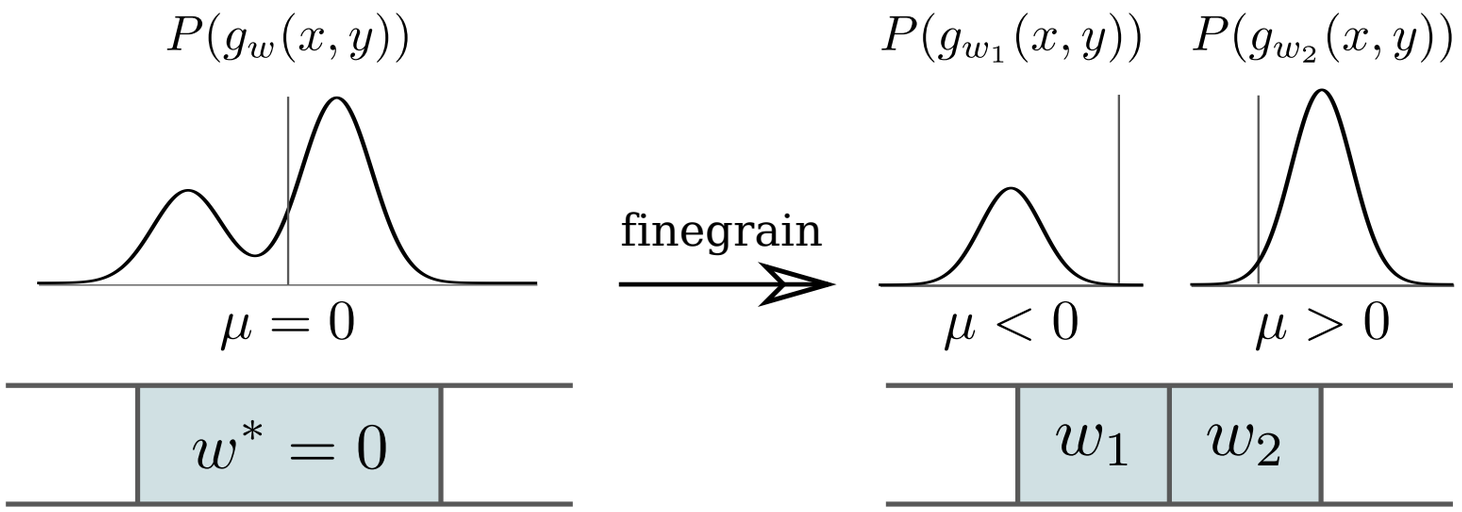
- Optimize with (stochastic) gradient descent during both **pre-training** and **fine-tuning** stages

- Multi-resolution training **speed-up** w.r.t. the contraction factor α and the terminal estimation error ϵ

Theorem 1 [Yu et al. 2018] If the gradient descent operator has a contraction factor of α, multi-resolution learning is faster than that of the fixed resolution algorithm by a factor of $\log\left(\frac{1}{(1-\alpha)\epsilon}\right)$, with ϵ as the terminal estimation error.

Fine-Graining Criteria

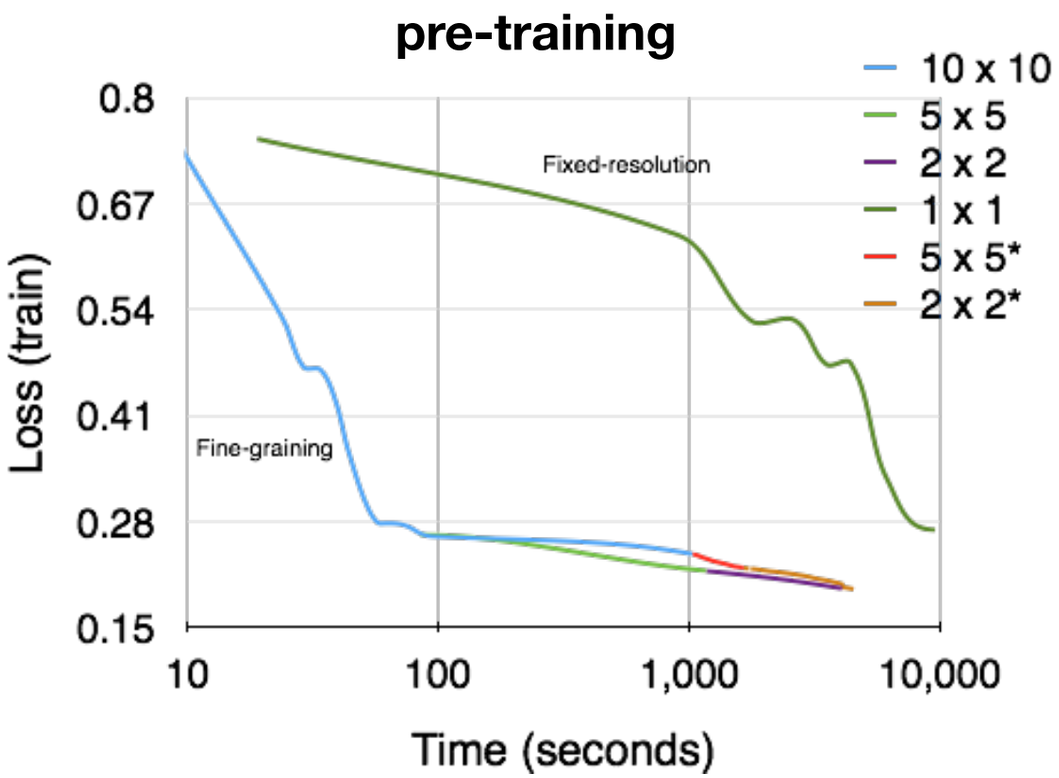
- Loss convergence: $\left|\mathcal{L}_t - \mathcal{L}_{t-1}\right| \leq \tau$

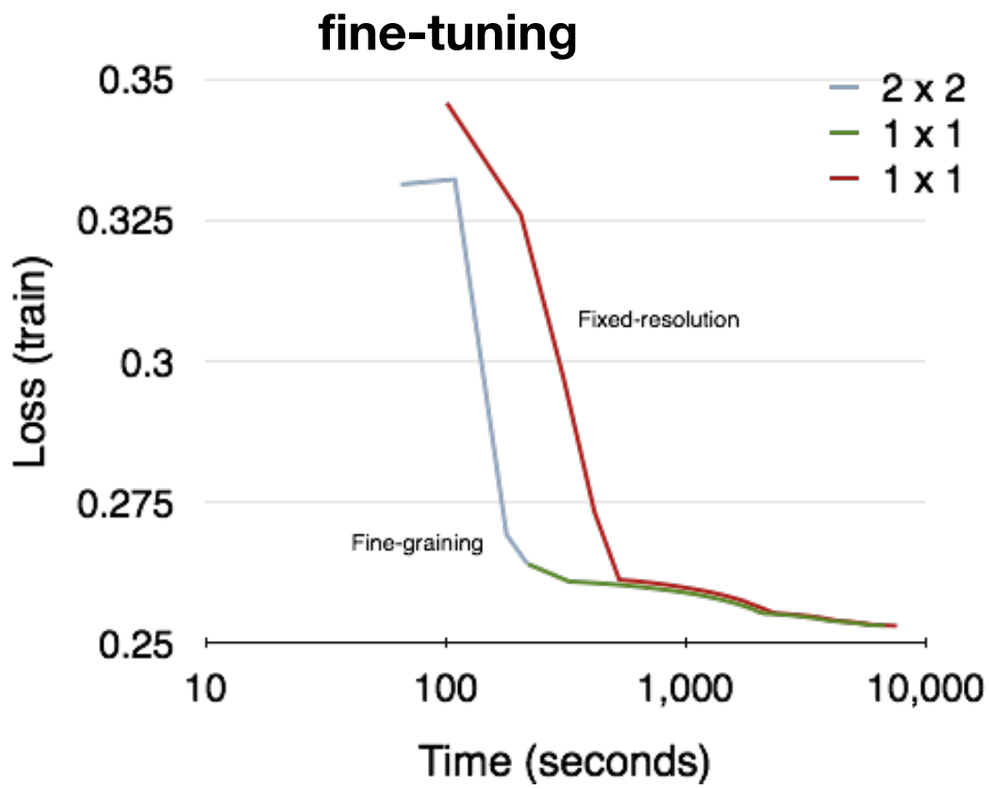


- Gradient statistics $g_w^{(i)} = \nabla_w \mathcal{L}(x^{(i)}, y^{(i)}, f(x^{(i)}, w))$

  - entropy $\mathbb{E}[\log p(g_w)] > \tau$
  - σ-divergence $\sigma > \tau$
  - μ, σ-divergence $\sigma > \tau, \mu < \tau'$

Experiments

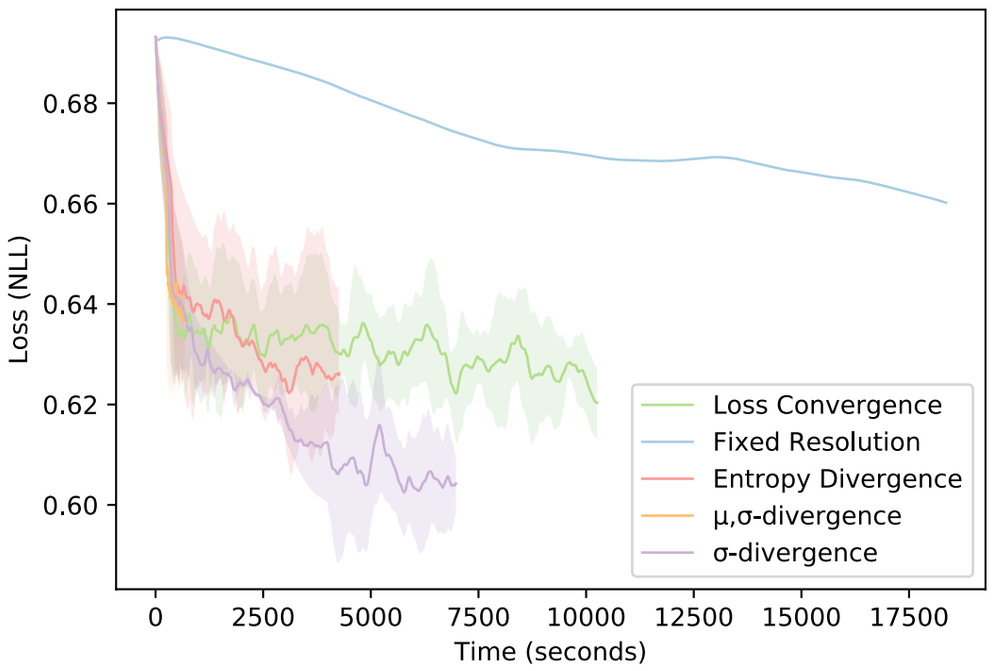




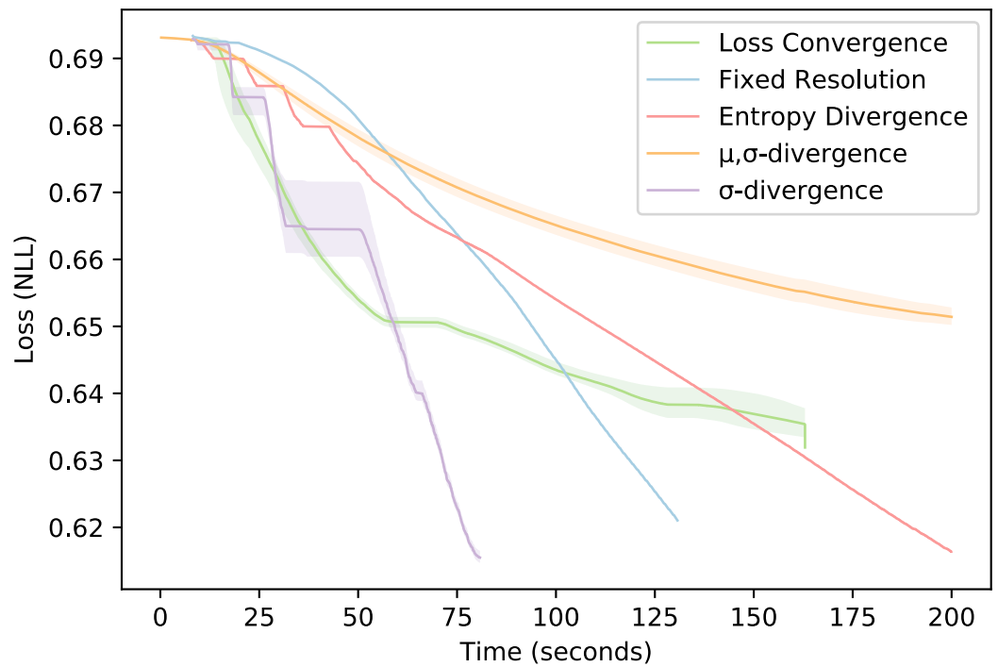
- Basketball shots: 50x40 full resolution, millions of game frames
- Multi-resolution outperforms fixed resolution
- Entropy (green, purple) control outperforms the loss convergence criterion (red, orange)

Sensitivity to fine-graining criteria

pre-training



fine-tuning



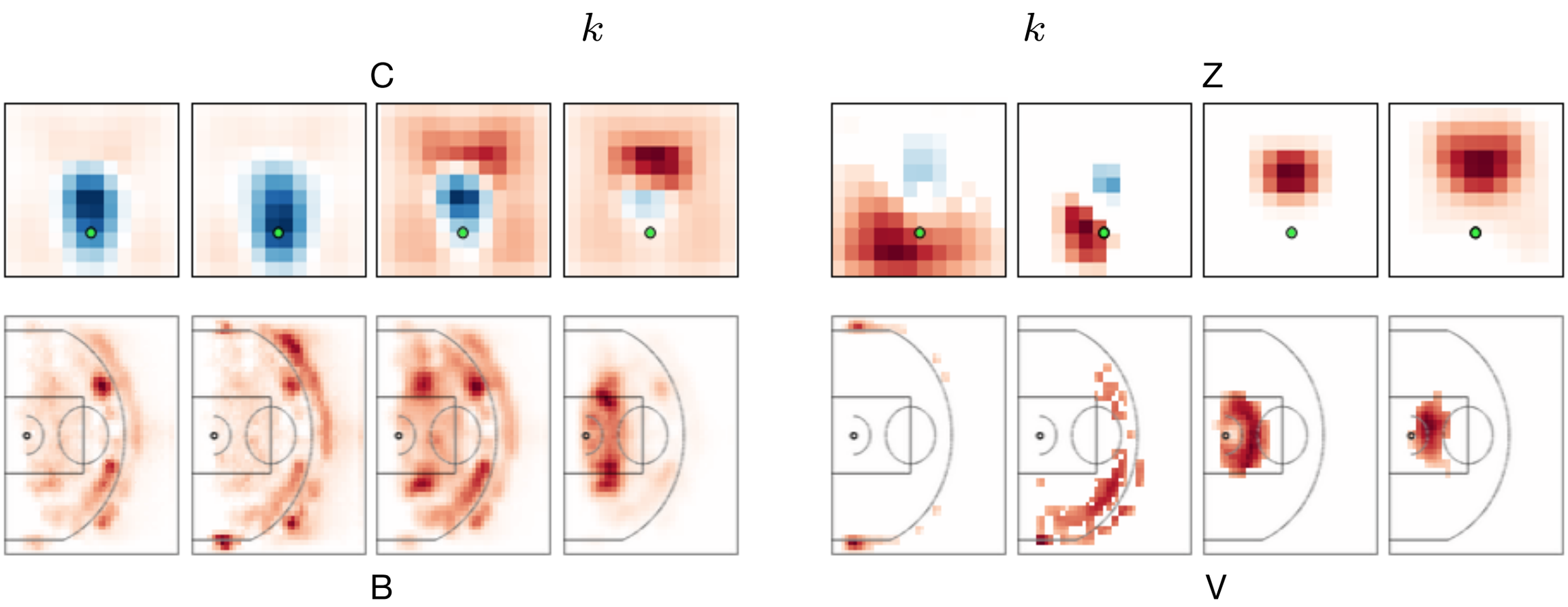
- Randomly sample different stopping thresholds
- Gradient statistics consistently outperforms loss convergence in the short-term (σ-threshold) and long-term (entropy threshold)

Interpretable Shooting Profiles

- C,Z: defenders around the ball handler influence the probability of a shot at the basket

- B,V: influence of ball handler's location

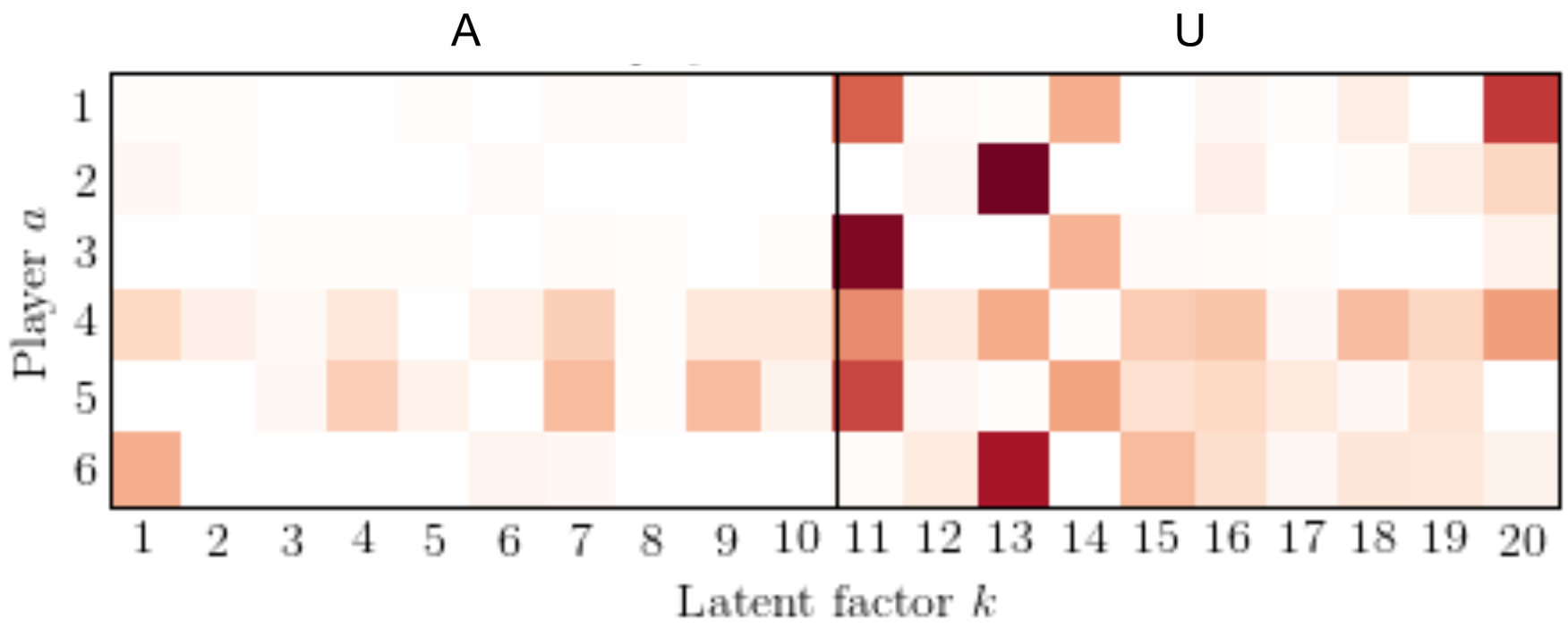
$$\mathcal{W}_{abc} = \sum_k A_{ak} B_{bk} C_{ck} + \sum_k U_{ak} V_{bk} Z_{ck}$$



Individual Player Profile

- A,U: Basketball play latent factor activation weights
- 1,2,3 consistent players, 4,5,6 inconsistent players

$$\mathcal{W}_{abc} = \sum_k A_{ak} B_{bk} C_{ck} + \sum_k U_{ak} V_{bk} Z_{ck}$$



Discussion

- Theoretical analysis on fine-graining criteria
  - When to fine-grain
  - How to interpolate from coarse to fine
- Uniqueness of the solution
  - Sensitivity to initial conditions
- Adaptive spatial grid based on density

Tensor-Train Recurrent Neural Network

Stephan Zheng
Salesforce

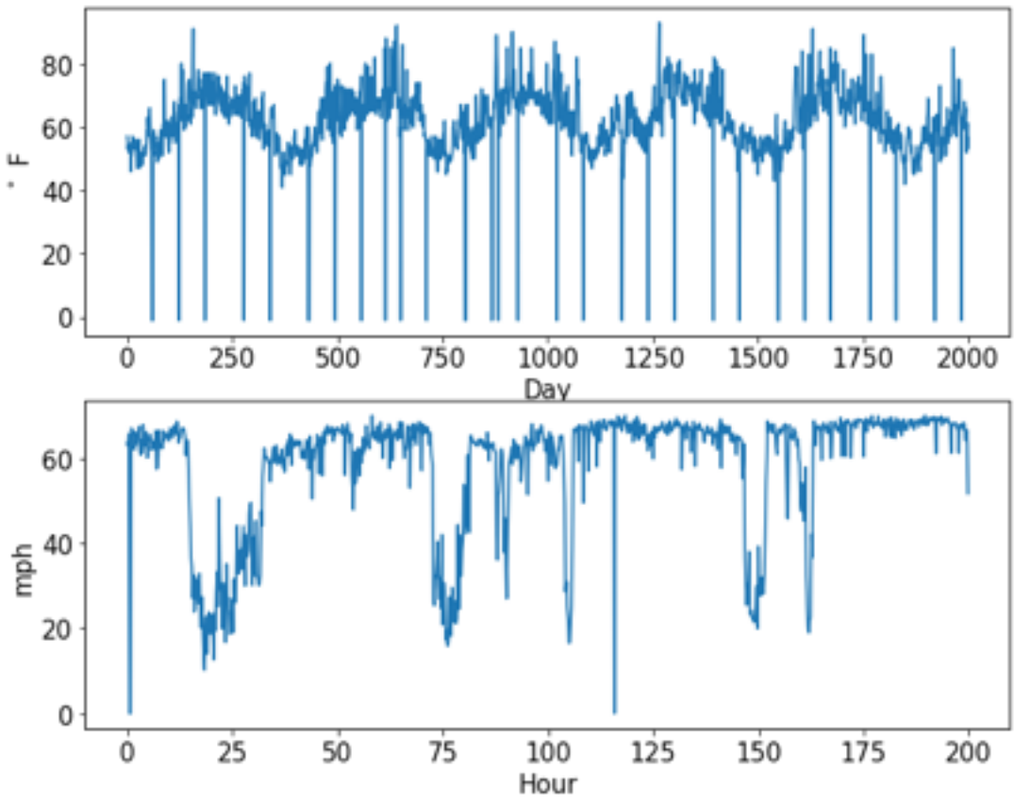
Anima Anandkumar
Caltech/NVIDIA

Anima Anandkumar
Caltech/NVIDIA

Long-term Forecasting using Tensor-Train RNNs
Rose Yu, Stephan Zheng, Anima Anandkumar, Yisong Yue
Best Paper, NIPS Time Series Workshop, Preprint arXiv:1711.00073

Difficulty in Nonlinear Environments

- Long-term dependencies
- High-order correlations
- Error propagation



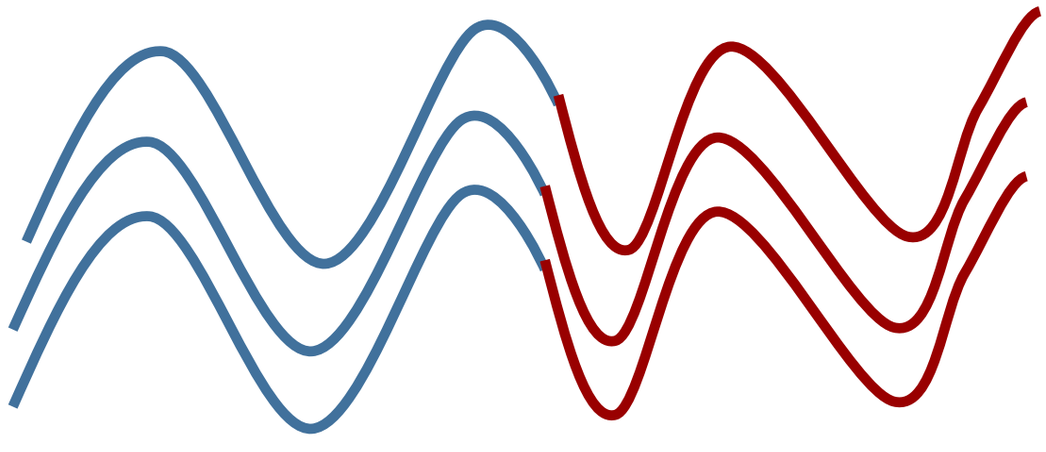
Long-Term Forecasting

- Given a system state x_t , parameters ϕ

$$\{\xi^i(x_t, \frac{\partial x_t}{\partial t}, \frac{\partial^2 x_t}{\partial t^2}, \cdots; \phi) = 0\}$$

- Learn a model f

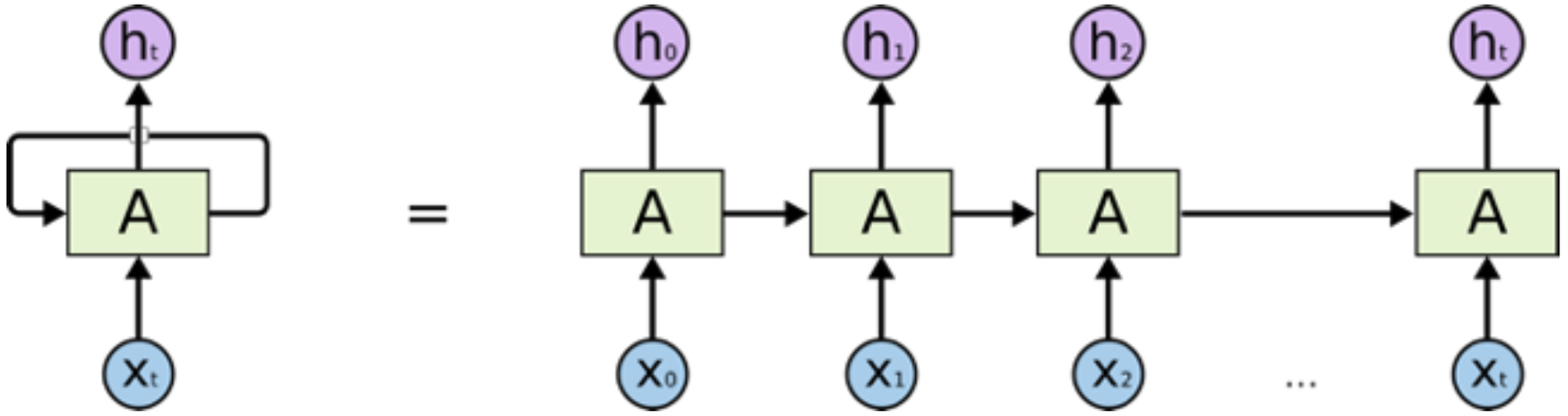
$$f : (x_0, \cdots, x_t) \rightarrow (x_{t+1}, \cdots, x_T)$$



First-Order Markov Models

- Input state x_t, hidden state h_t, output x_{t+1}

$$h_t = f(x_t, h_{t-1}; \theta) \qquad x_{t+1} = g(h_t; \theta)$$



- Linear interactions

$$h_t = f(W^{hx} x_t + W^{hh} h_{t-1}; \theta)$$

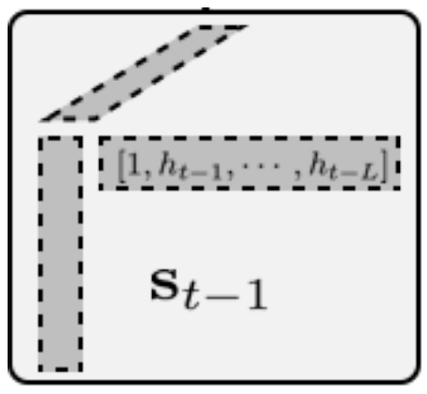
High-Order Non-Markovian

- L-order Markov

$$h_t = f(x_t, h_{t-1}, h_{t-1}, \cdots, h_{t-L}; \theta) \qquad x_{t+1} = g(h_t; \theta)$$

- Polynomial interactions

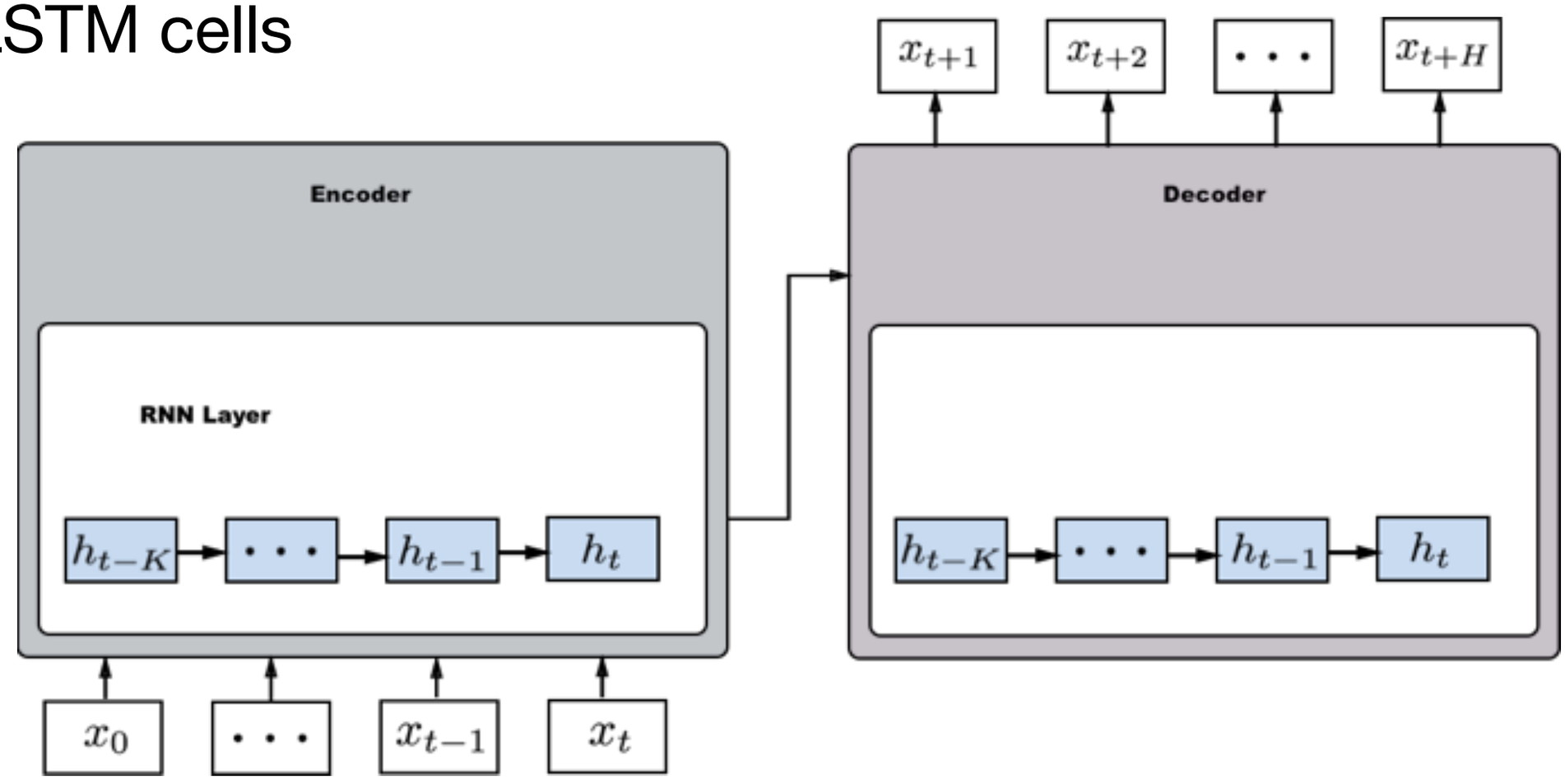
$$s_{t-1}^\top = [1, h_{t-1}^\top, \cdots, h_{t-L}^\top]$$



$$h_t = f(W^{hx} x_t + \mathcal{W}^{hh} \underbrace{s_{t-1} \otimes \cdots \otimes s_{t-1}}_{p}; \theta)$$

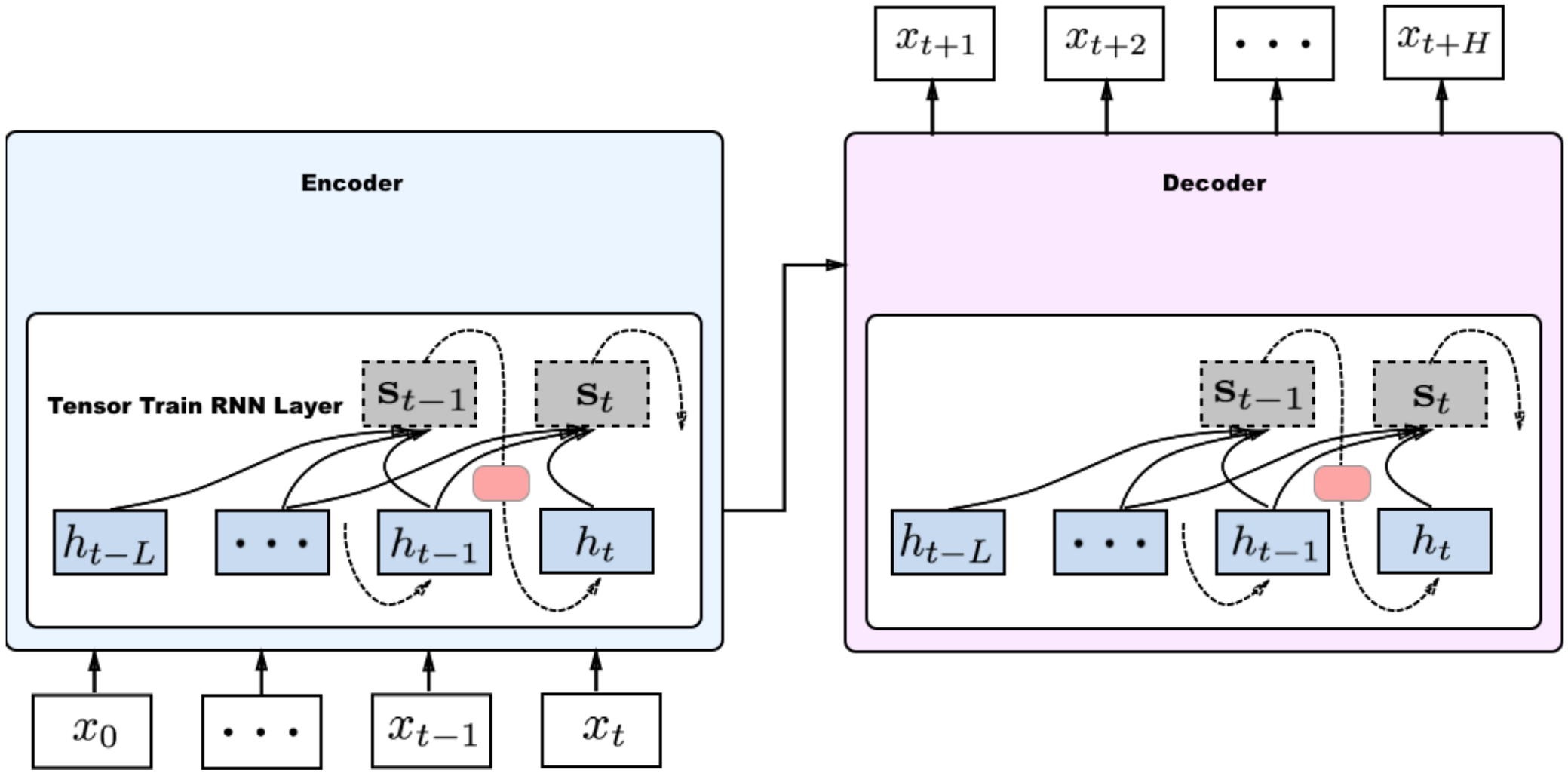
Forecasting with RNNs

- Seq2Seq architecture
- LSTM cells



High-Order Tensor RNNs

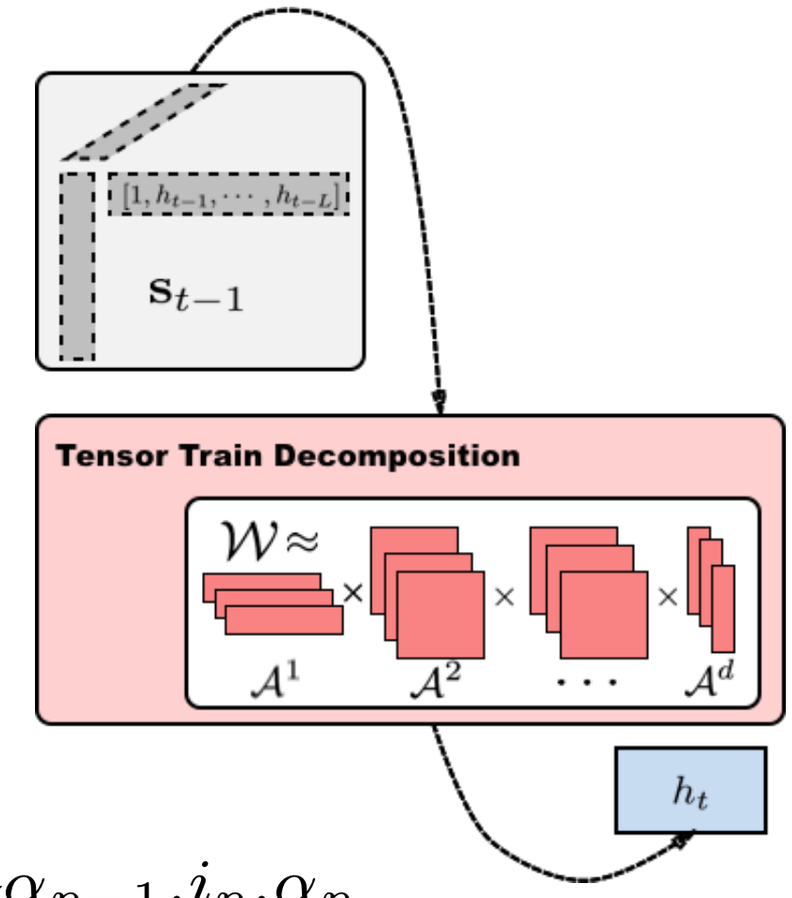
- Seq2Seq architecture
- HOT-LSTM cells



Tensor Train Decomposition

$$h_t = f(W^{hx}x_t + \mathcal{W}^{hh}\underbrace{s_{t-1} \otimes \cdots \otimes s_{t-1}}_{p}; \theta)$$

- Reduce # of parameters
- Linear tensor network
- Dimension-free decomposition



$$\mathcal{W}^{hh}_{i_1,\cdots,i_p} = \sum_{\alpha_0,\cdots,\alpha_p} \mathcal{A}_{\alpha_0,i_1,\alpha_1} \cdots \mathcal{A}_{\alpha_{p-1},i_p,\alpha_p}$$

Approximation Guarantee

Theorem 1 [Yu et al 2017]: *Let the state transition function $f \in \mathcal{H}_{\mu}^{k}$ be a Hölder continuous, with bounded derivatives up to order k and finite Fourier magnitude distribution C_f. Then a single layer HOT-RNN can approximate f with an estimation error of ε using with h hidden units:*

$$\varepsilon \leq \frac{C_f^2}{h} \frac{(d-1)}{(k-1)(r+1)^{(k-1)}} + \frac{C(k)}{h} p^{-k}$$

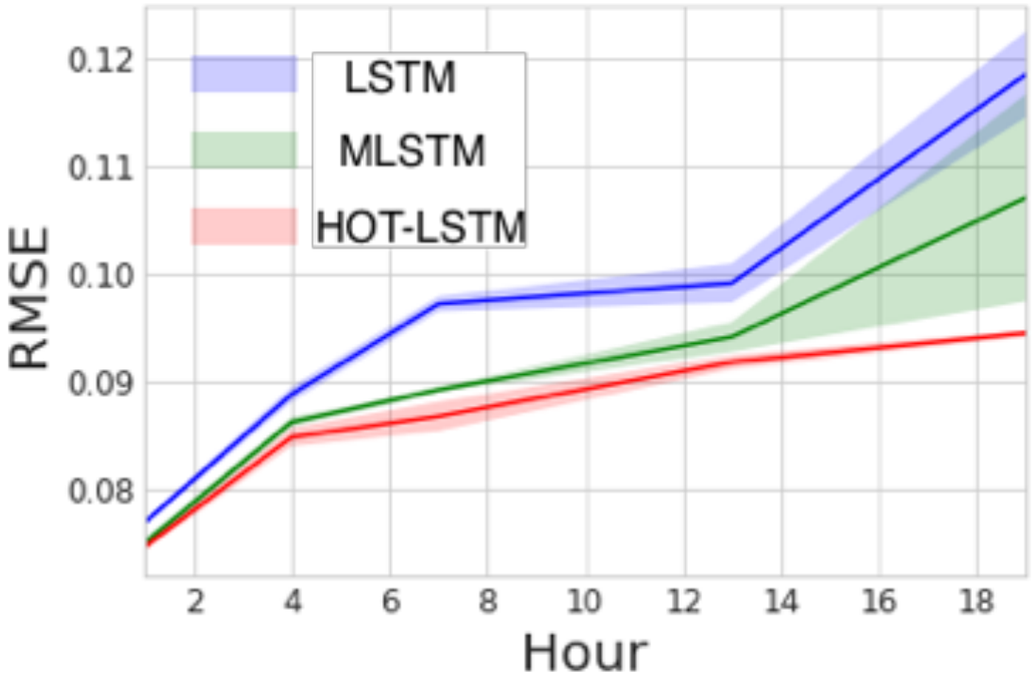
Where d is the size of the state space, r is the tensor-train rank and p is the degree of high-order polynomials i.e., the order of tensor.

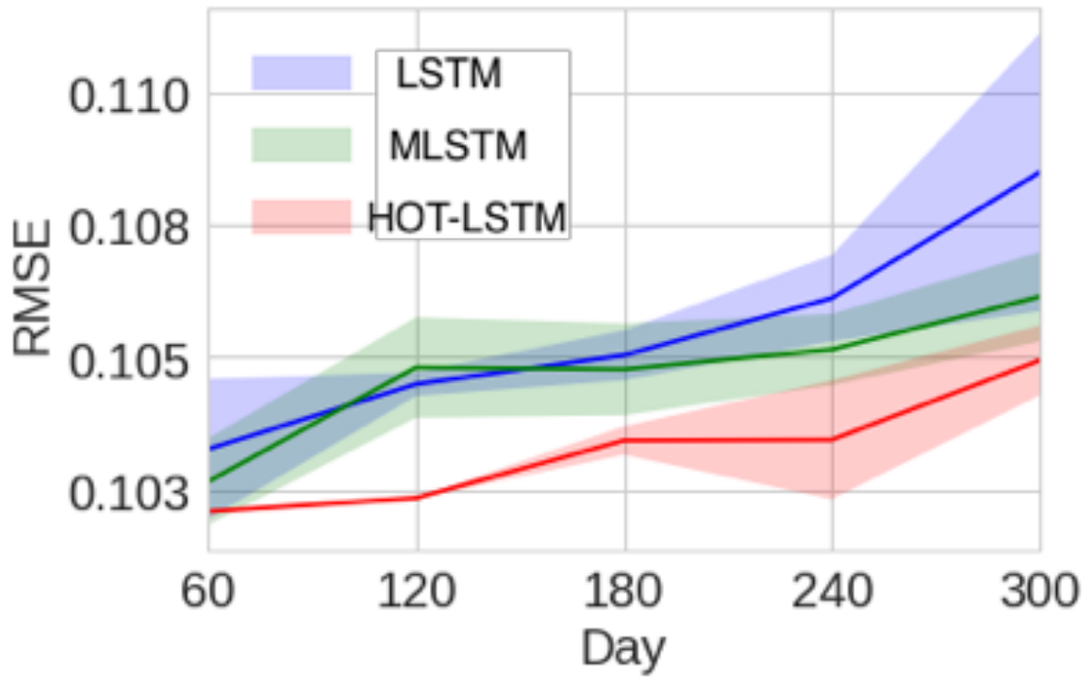
- Number of weights dictated by its regularity *k*
- Expressiveness is driven by the selection of the rank *r* and the polynomial degree *p*
- Improves for functions with increasing regularity

Experiments

Traffic: Multi-sensor 8,700 traffic speed sequences, forecast 18 hour ahead given 5 hour observations

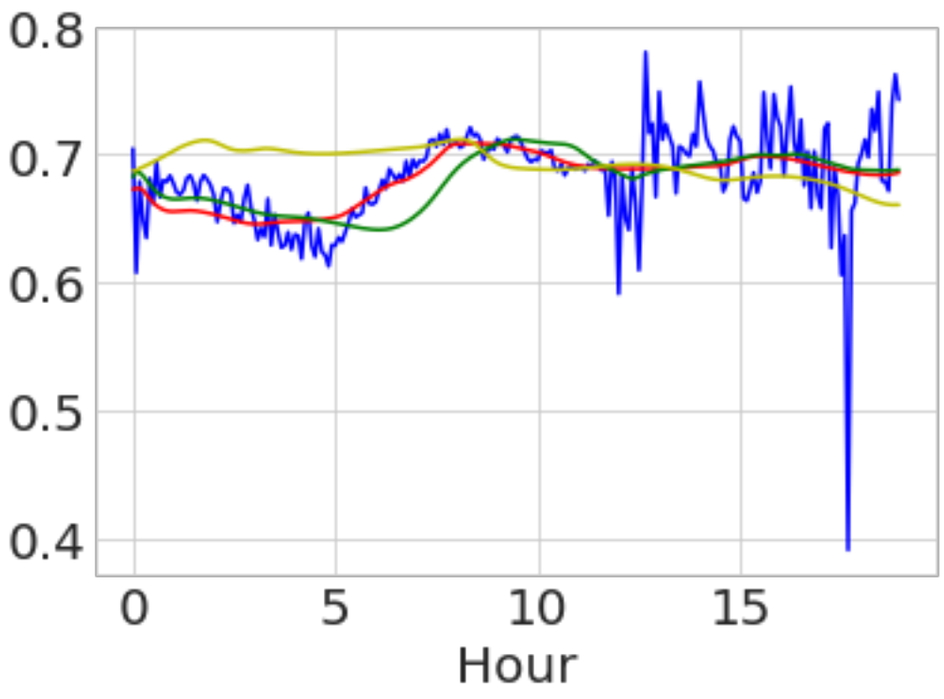
Climate: Multi-station 6,900 temperature sequences, forecast 300 days ahead given 60 days observations

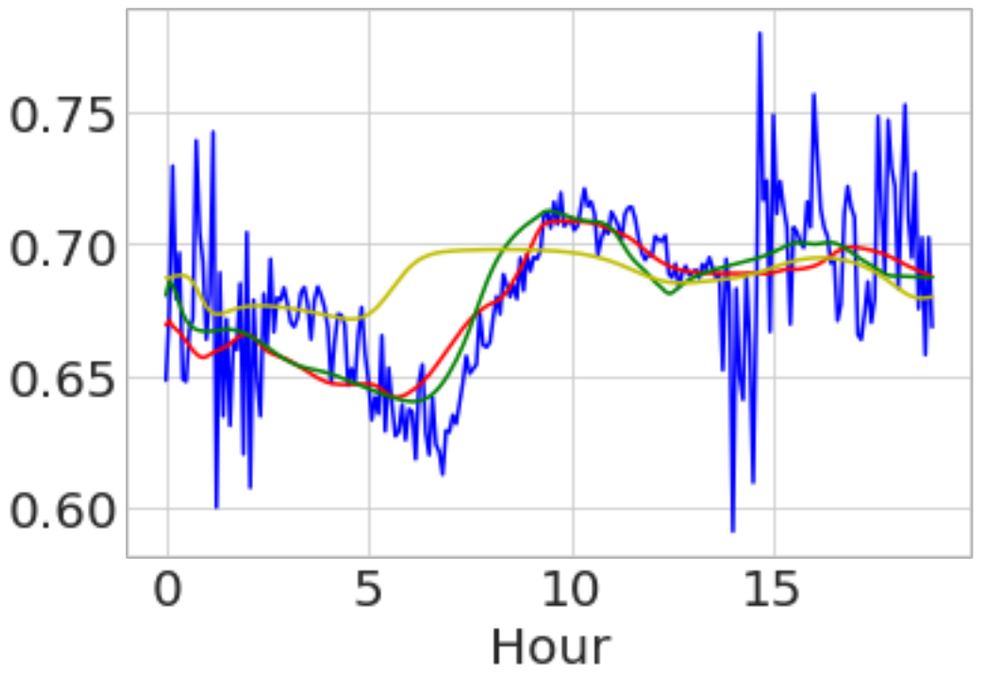




Prediction Visualization

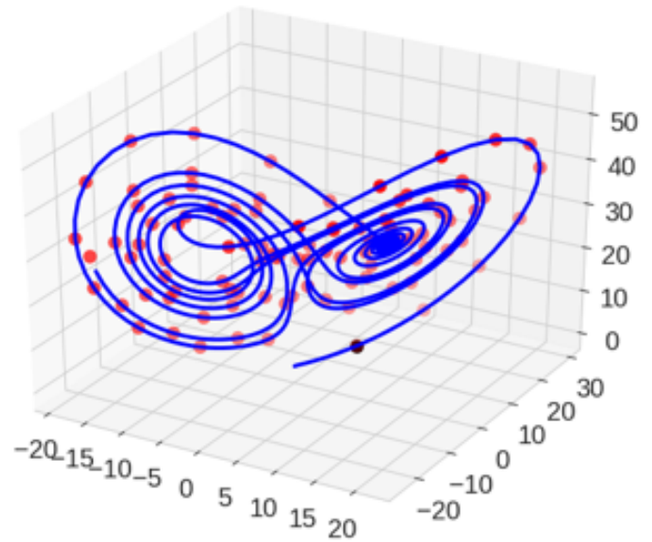
- 18 hour ahead traffic forecasting
- HOT-LSTM predictions have more variability
- Can learn high-order structure in time series

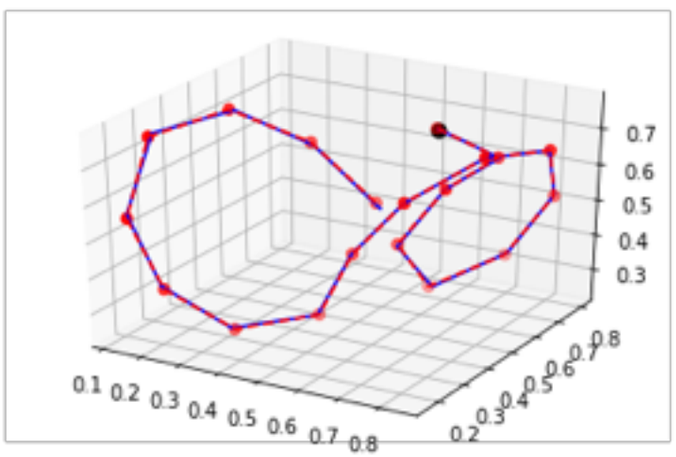


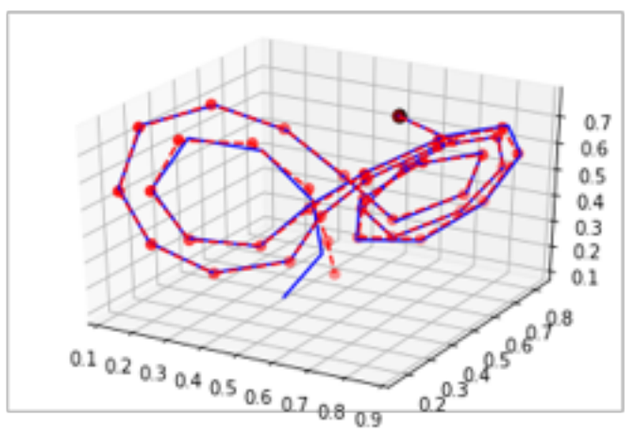


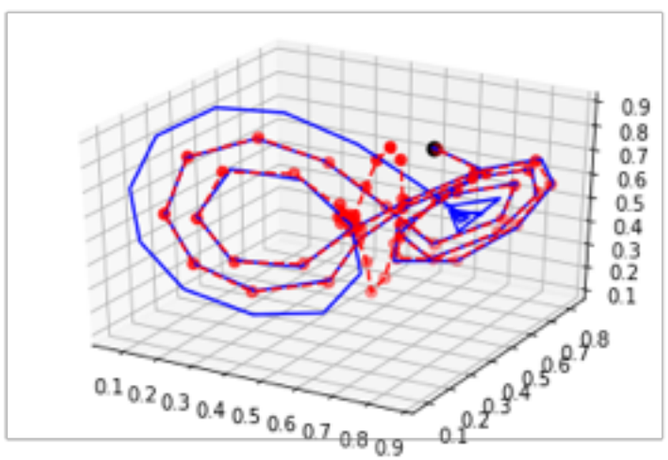
Open Problem

- Chaotic dynamics
- Small initial perturbation leads to different behavior
- E.g. Lorenz attractor









Discussion

- Probabilistic counterpart of tensor-train RNNs
  - Forecasting with uncertainty
    - Efficient contraction algorithms within the neural network
- Model selection
  - Optimal rank combination during training
  - Effect of rank in generalization error

Thank you!

Email: roseyu@northeastern.edu