Decomposing Temporal High-Order Interactions via Latent ODEs

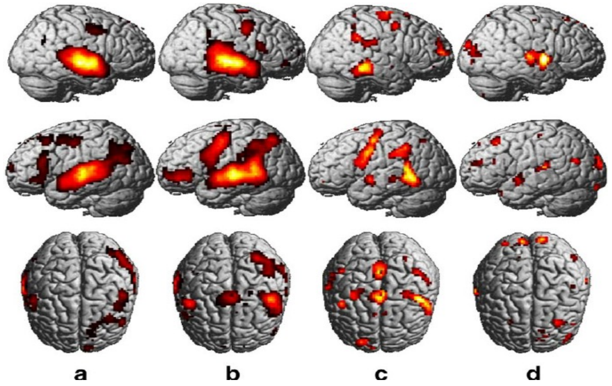
Shibo Li, Robert M. Kirby *and* Shandian Zhe
School of Computing, University of Utah
Presenter: Shibo Li
shibo@cs.utah.edu

Background

- (High-order) Interactions in Real-world

(user, movie, episode)

(user, advertisement, page-section)



(subject, voxel, electrode)

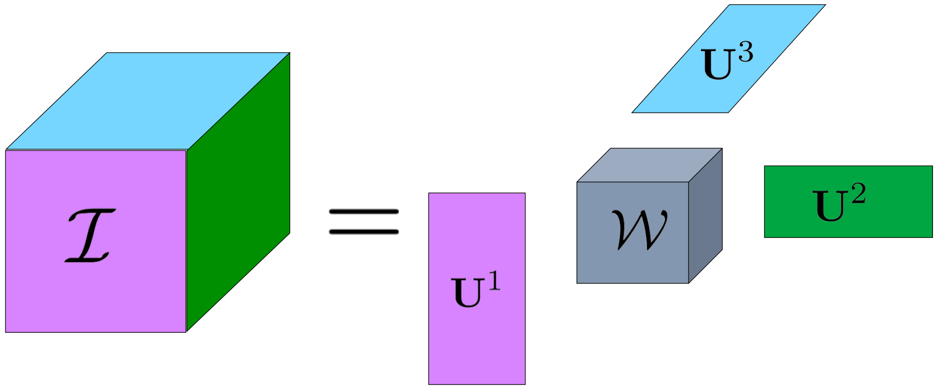
(user, item, online-store)

(user, user, location, message-type)

(patient, gene, condition)

Background

- Tensor Decomposition

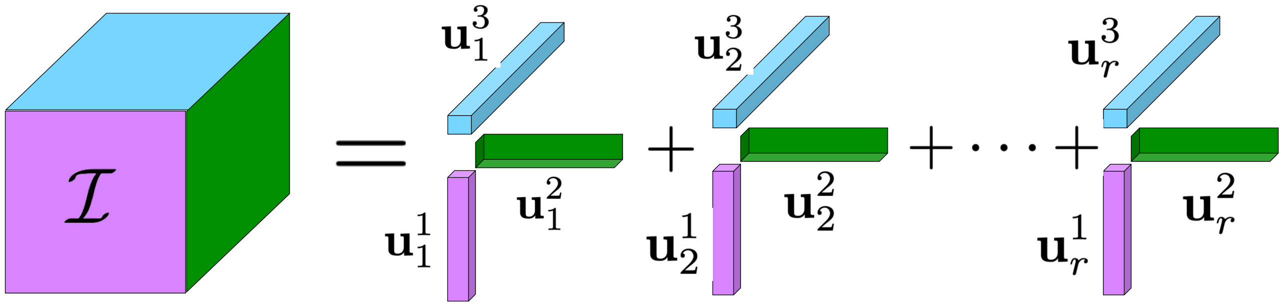


$d_1 \times \cdots \times d_K$

$$\mathcal{I} = \mathcal{W} \times_1 \mathbf{U}^1 \times_2 \cdots \times_K \mathbf{U}^K$$

$r_1 \times \cdots \times r_K$ $d_1 \times r_1$ $d_K \times r_K$

Tucker, 1966



$$\mathcal{I} = \sum_{j=1}^{r} \lambda_j \cdot \mathbf{U}^1[:, j] \circ \ldots \circ \mathbf{U}^K[:, j]$$

CANDECOMP/PARAFAC (CP), Harshman, 1970

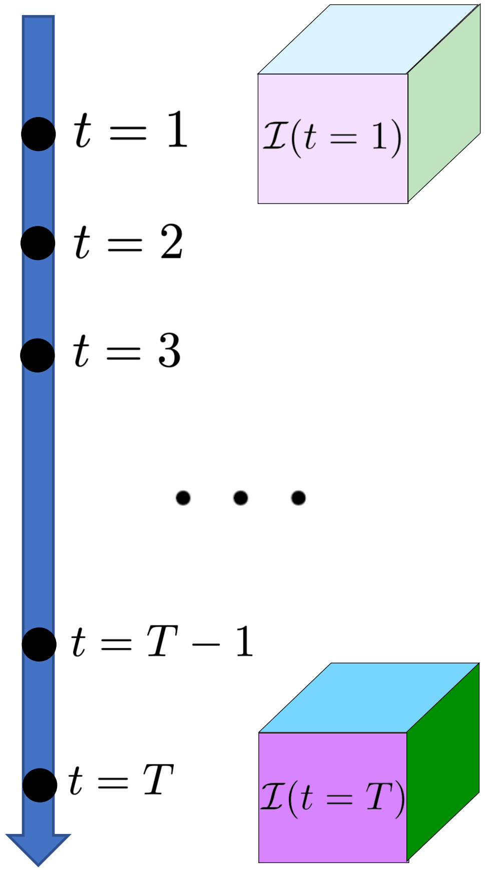
Motivation

- Temporal Interactions $\mathcal{I}(t)$

Our Contribution:

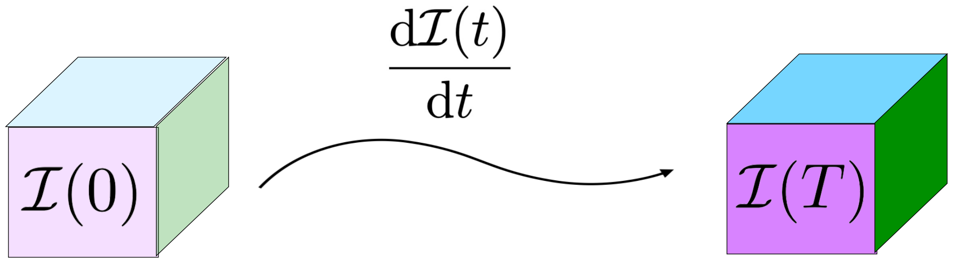
- ***THIS-ODE***: A novel decomposing model of temporal high-order interactions.
- Leverage continuous timestamps, capture all kinds of complex temporal dynamics within interactions.
- Tractable inference with forward sensitivity analysis and time alignment/integral transform tricks.

Discretized time



vs.

Continuous time



Ours: THIS-ODE

Method

- Temporal interaction as parametric latent ODE

$$\begin{cases} \frac{\mathrm{d}m_{\mathbf{i}}(t)}{\mathrm{d}t} = f\left(m_{\mathbf{i}}(t), \mathbf{v}_{\mathbf{i}}, t\right) & \text{Dynamics} \\ m_{\mathbf{i}}(0) = \beta(\mathbf{v}_i) & \text{I.C.} \end{cases} \xrightarrow{\text{ODESolve}} m_{\mathbf{i}}(t) = m_{\mathbf{i}}(0) + \int_0^t f_{\boldsymbol{\theta}}(m_{\mathbf{i}}(s), \mathbf{v}_{\mathbf{i}}, s)\mathrm{d}s$$

- Joint probability given $\mathcal{D} = \{(\mathbf{i}_1, t_1, y_1), \ldots, (\mathbf{i}_N, t_N, y_N)\}$

$$p(\mathcal{U}, \nu, \mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{K} \prod_{j=1}^{d_k} \mathcal{N}(\mathbf{u}_j^k|\mathbf{0}, \mathbf{I}) \cdot \mathrm{Gam}(\nu|a_0, b_0) \cdot \prod_{n=1}^{N} \mathcal{N}(y_n|m_{\mathbf{i}_n}(t_n), \nu^{-1})$$

Prior of representations — Prior noise precision — Probability of noisy observations

Note: the ODE parameters are implicitly located in the probability of observations

Method

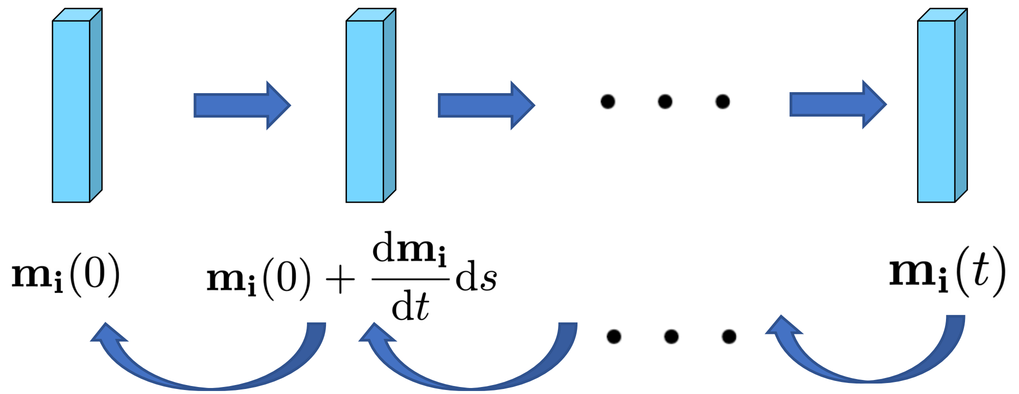
- Why not auto-differentiation?

$$p(\mathcal{U}, \nu, \mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{K} \prod_{j=1}^{d_k} \mathcal{N}(\mathbf{u}_j^k|\mathbf{0}, \mathbf{I}) \cdot \mathrm{Gam}(\nu|a_0, b_0) \cdot \prod_{n=1}^{N} \mathcal{N}(y_n|m_{\mathbf{i}_n}(t_n), \nu^{-1})$$

$$m_{\mathbf{i}}(t) = m_{\mathbf{i}}(0) + \int_0^t f_{\boldsymbol{\theta}}(m_{\mathbf{i}}(s), \mathbf{v}_{\mathbf{i}}, s)\mathrm{d}s$$

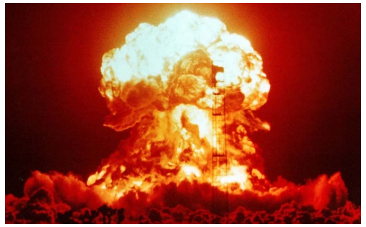
Given by ODE solvers

Computational graph constructed by ODE solvers



Auto-differentiation

All the intermedia states needs to be saved to compute the gradients



Algorithm

- Efficient computation of gradients with ***Forward Sensitivity***

ODE of system sensitivity

$$\mathbf{z_i} = [m_\mathbf{i}^0; \boldsymbol{\eta}]$$

Then $\mathbf{s_i}(t) = \frac{\partial m_\mathbf{i}(t)}{\partial \mathbf{z_i}} = [\frac{\partial m_\mathbf{i}(t)}{\partial m_\mathbf{i}^0}; \frac{\partial m_\mathbf{i}(t)}{\partial \boldsymbol{\eta}}]$

$$\begin{cases} \frac{\mathrm{d}\mathbf{s_i}(t)}{\mathrm{d}t} = \frac{\partial f}{\partial m_\mathbf{i}(t)}\mathbf{s_i}(t) + \frac{\partial f}{\partial \mathbf{z_i}} \\ \mathbf{s_i}(0) = \frac{\mathrm{d}m_\mathbf{i}^0}{\mathrm{d}\mathbf{z_i}}, \end{cases}$$

- ***Time alignment*** for efficient stochastic mini-batch optimization

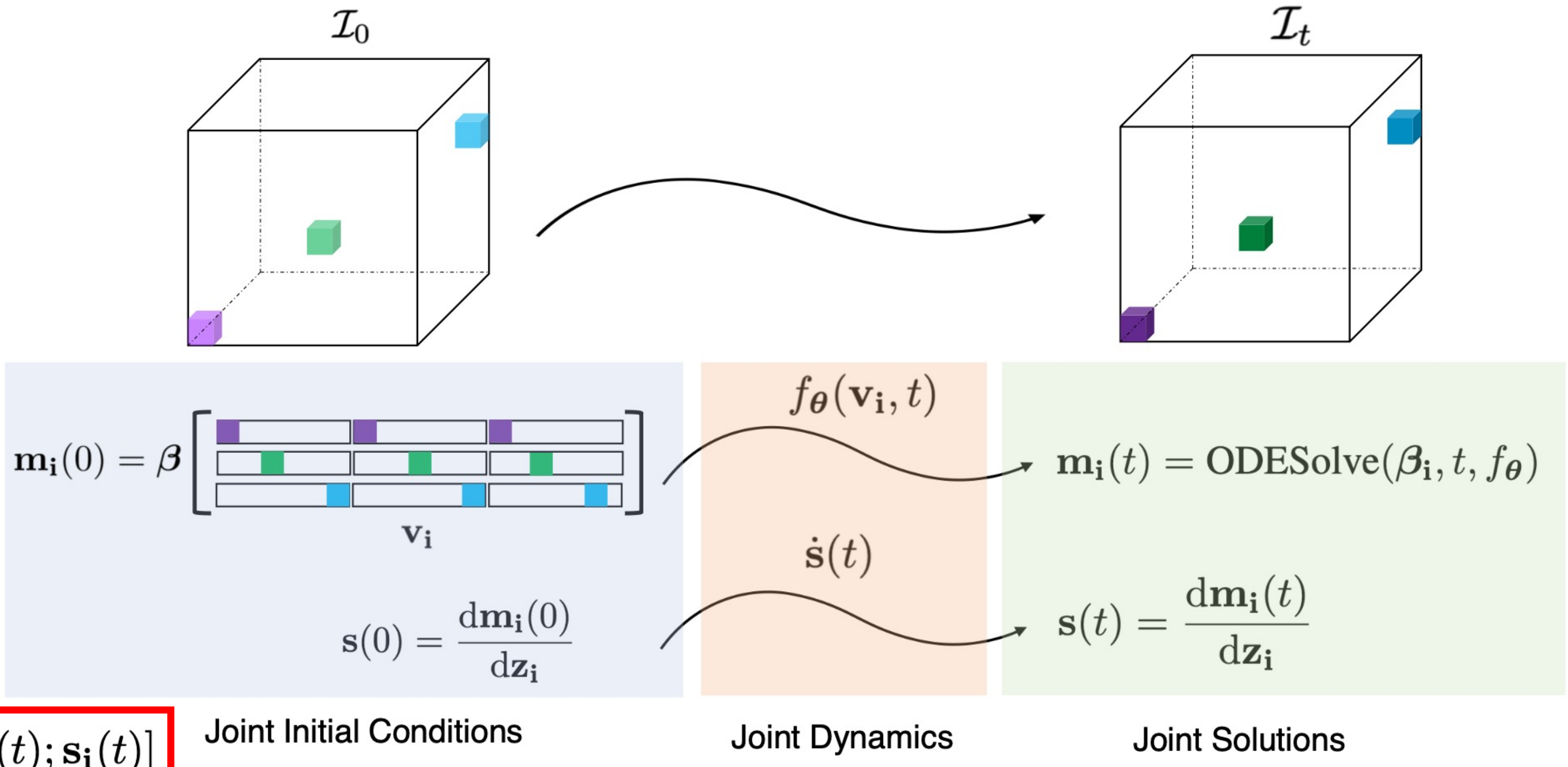
$$\mathbf{h}_{\mathbf{i}_l}(t_l) = \mathbf{h}_{\mathbf{i}_l}(0) + \int_0^{t_l} \boldsymbol{\alpha}\left(\mathbf{h}_{\mathbf{i}_l}(\tau), \tau\right) \mathrm{d}\tau = \mathbf{h}_{\mathbf{i}_l}(0) + \int_0^{t_e} \frac{t_l}{t_e} \boldsymbol{\alpha}\left(\mathbf{h}_{\mathbf{i}_l}(\frac{t_l}{t_e}s), \frac{t_l}{t_e}s\right) \mathrm{d}s$$

$$\mathbf{h_i}(t) = [m_\mathbf{i}(t); \mathbf{s_i}(t)]$$

Algorithm

- The bird view

- Depends on current state solution only
- Jointly solved with the system state with **only one ODE solver** and **forward pass**



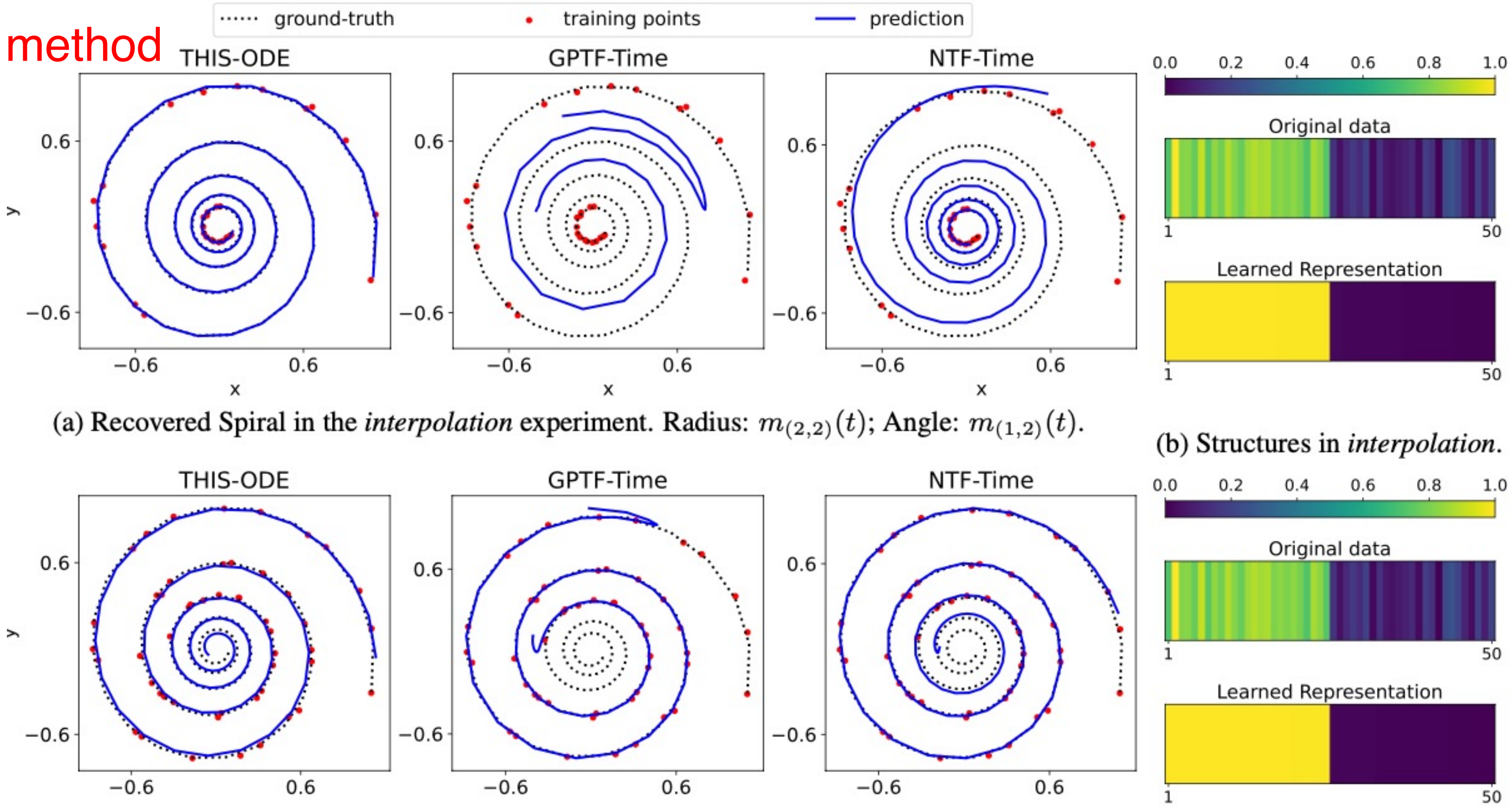
$$\mathbf{h_i}(t) = [m_\mathbf{i}(t); \mathbf{s_i}(t)]$$

Experiment

- Ablation Study: Spiral Interactions

$$m_{\mathbf{i}}(t) = \left(u^1_{i_1} \exp(-0.5t)\right)^{\mathbb{1}(i_1+i_2 \mod 2=0)} \cdot \left(u^2_{i_2} + 2\pi t\right)^{\mathbb{1}(i_1+i_2 \mod 2=1)}$$

Our method



(a) Recovered Spiral in the *interpolation* experiment. Radius: $m_{(2,2)}(t)$; Angle: $m_{(1,2)}(t)$.

(b) Structures in *interpolation*.

(c) Recovered Spiral in the *extrapolation* experiment. Radius: $m_{(2,2)}(t)$; Angle: $m_{(1,2)}(t)$.

(d) Structures in *extrapolation*.

	Interpolation	Extrapolation
GPTF-Time	0.5557	0.9032
NTF-Time	0.1004	0.3656
THIS-ODE	**0.0148**	**0.0746**

Our method

Experiment

- Real-world applications: *Beijing Air*, *Indoor Condition*, *Server Room*, *Fit Record*

Name	Description	Size	NNZ	Granularity in Time
Beijing Air Quality	time x locations x pollutants	35064 x 12 x 6	2454305	hourly
Indoor Condition	time x locations x sensor	19735 x 9 x 2	241201	every 10 minutes

Interpolation	*Beijing Air*	*Indoor Condition*	…
CP-Time	0.897 ± 0.012	0.780 ± 0.012	
CP-DTL	0.898 ± 0.015	0.842 ± 0.003	
CP-DTN	0.833 ± 0.003	0.889 ± 0.005	
GPTF-Time	0.711 ± 0.011	0.849 ± 0.005	
GPTF-DTL	0.686 ± 0.045	0.852 ± 0.004	
GPTF-DTN	0.670 ± 0.062	0.713 ± 0.104	
NTF-Time	0.745 ± 0.095	0.800 ± 0.009	
NTF-DTL	0.757 ± 0.006	0.777 ± 0.018	
NTF-DTN	0.686 ± 0.011	0.665 ± 0.004	
PTucker	0.959 ± 0.015	0.806 ± 0.027	
THIS-ODE	$\mathbf{0.624 \pm 0.008}$	$\mathbf{0.618 \pm 0.007}$	

Extrapolation			…
CP-Time	0.863 ± 0.022	0.867 ± 0.010	
CP-DTL	0.553 ± 0.005	0.527 ± 0.006	
CP-DTN	0.557 ± 0.004	0.584 ± 0.009	
GPTF-Time	0.527 ± 0.018	0.489 ± 0.011	
GPTF-DTL	0.577 ± 0.035	0.506 ± 0.013	
GPTF-DTN	0.511 ± 0.002	0.489 ± 0.003	
NTF-Time	0.537 ± 0.002	0.510 ± 0.027	
NTF-DTL	0.512 ± 0.009	0.593 ± 0.079	
NTF-DTN	0.513 ± 0.003	0.484 ± 0.011	
PTucker	0.522 ± 0.022	0.749 ± 0.006	
THIS-ODE	$\mathbf{0.498 \pm 0.013}$	$\mathbf{0.460 \pm 0.004}$	

Our method

Welcome to our poster!

Shibo Li (shibo@cs.utah.edu)