Artificial Intelligence for Smart Transportation

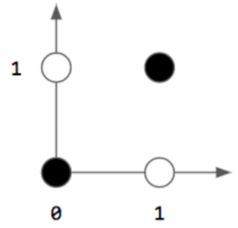
Yan Liu

Associate Professor

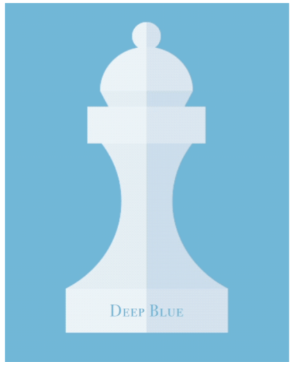
Computer Science Department

University of Southern California

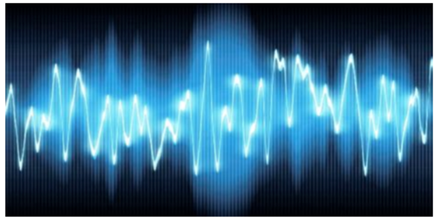
AI and Machine Learning



Neural Networks



Machine Learning: supervised, unsupervised



Deep Learning

Reinforcement Learning

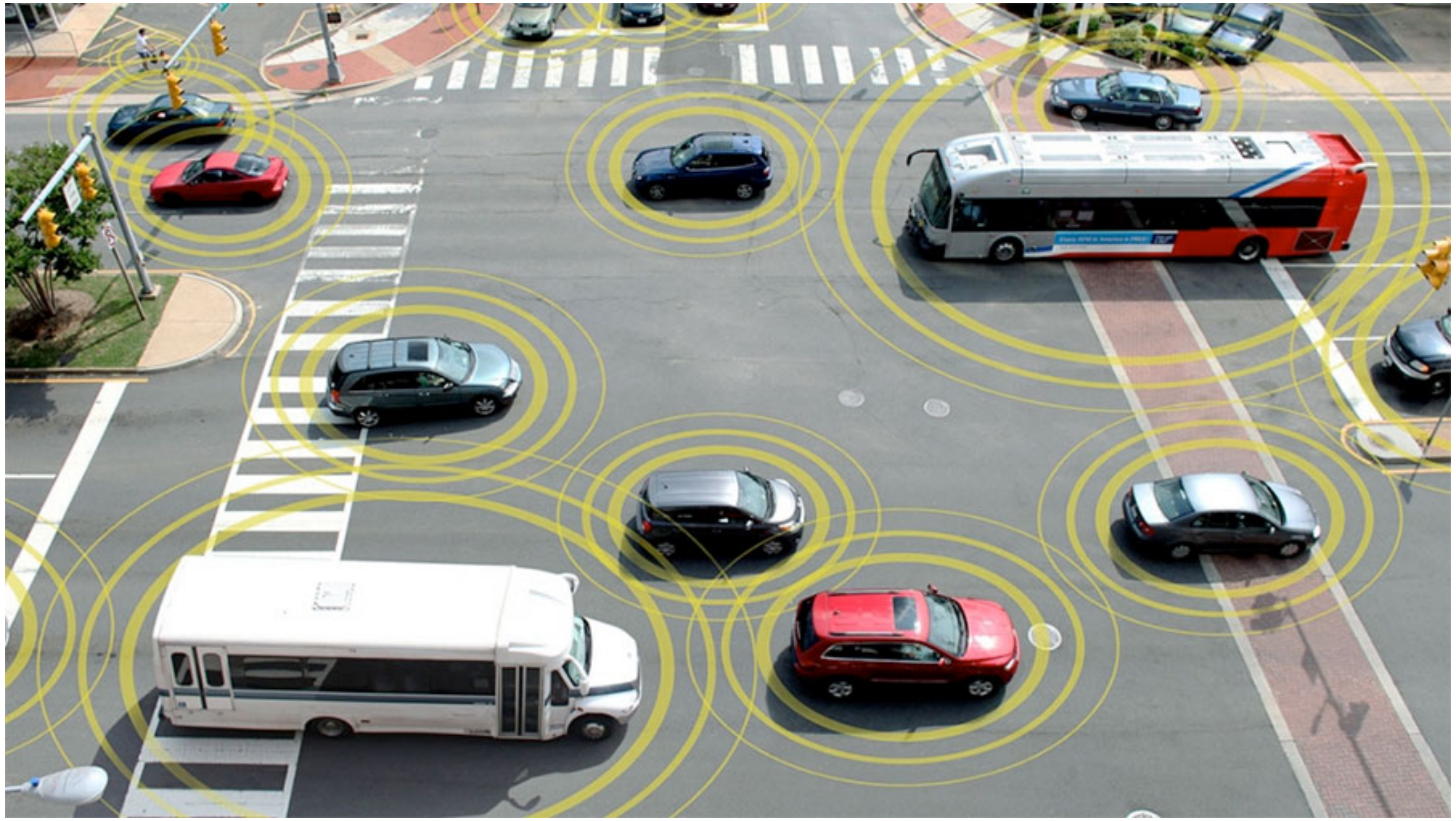
AlphaGo

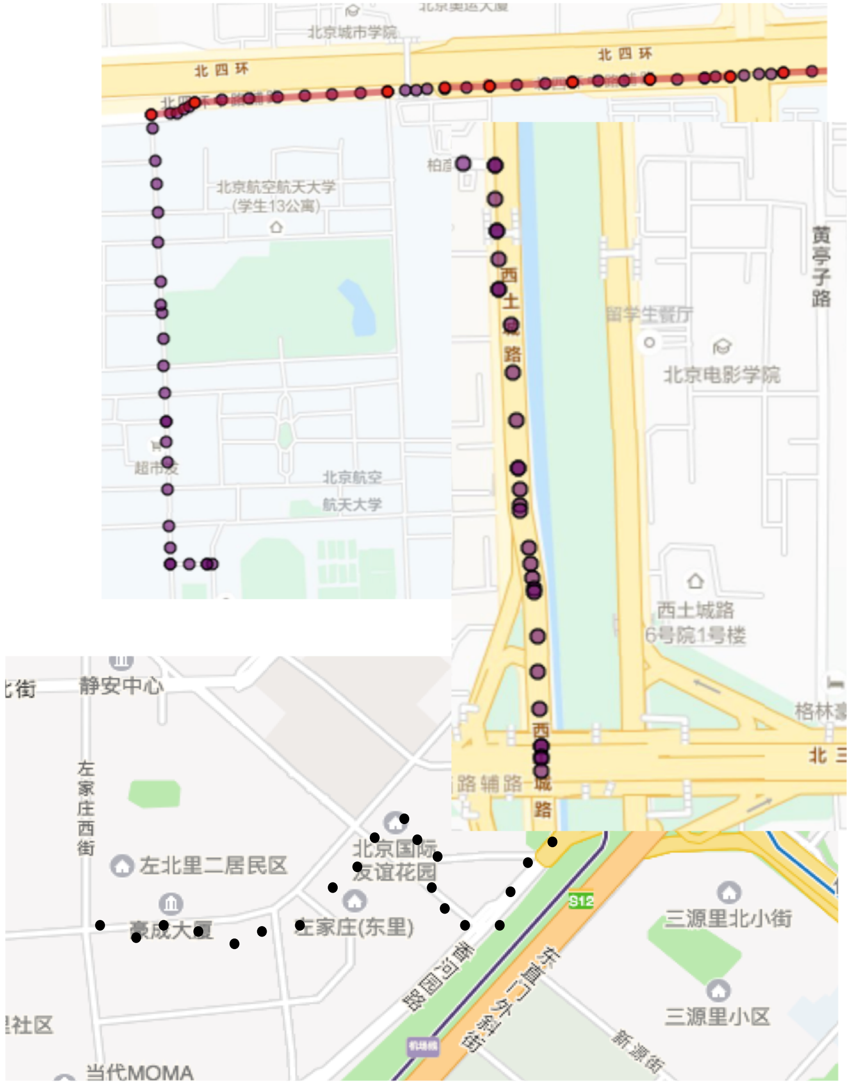
nlp

?

GPS Data

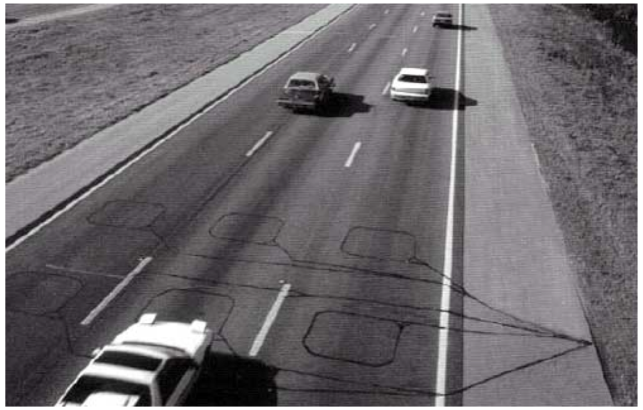
Location Data and Floating-Car Trajectory

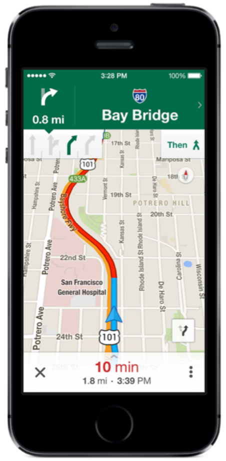




Sensors

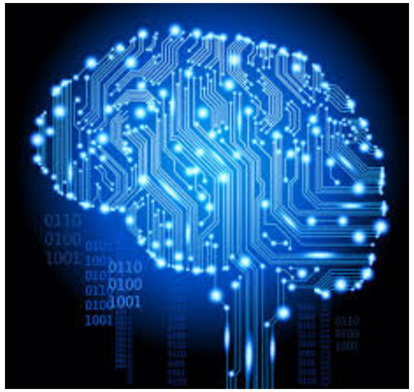
Loop detector, camera, microphone, mobile sensors …



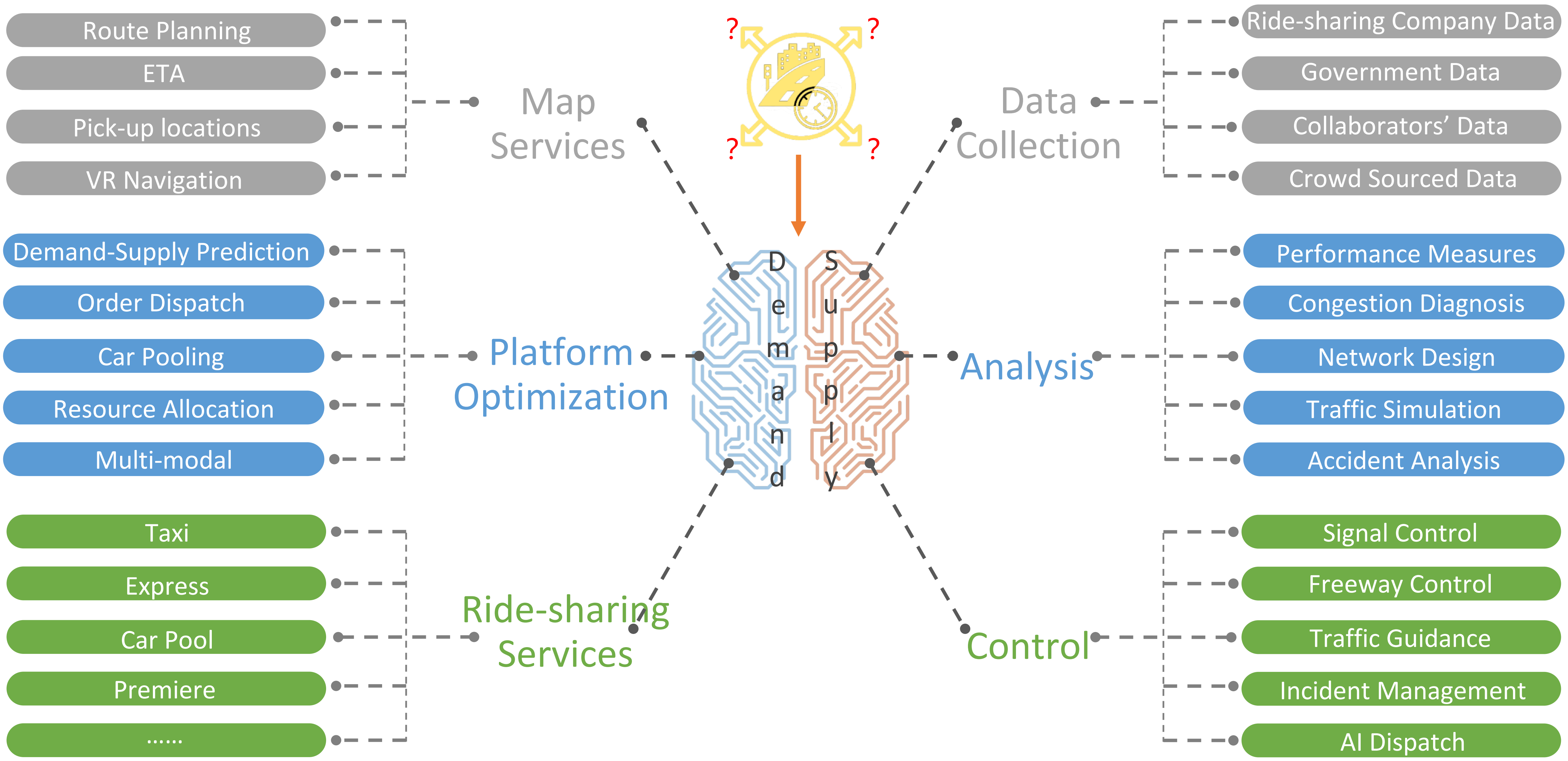



Transportation AI

Big data makes AI possible for transportation.



Smart Transportation Brain



Outline

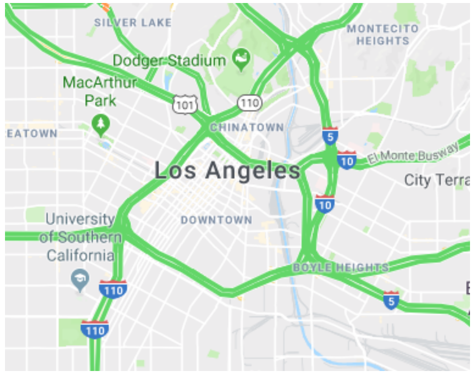
- Traffic estimation and forecasting
  - Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018
- Demand forecasting
  - Li et al, Spatiotemporal Multi-Graph Convolution for Ride-hailing Demand Forecasting, AAAI 2019
- Multi-rate multi-resolution forecasting/interpolation
  - Che et al, Hierarchical Deep Generative Models for Multi-Rate Multivariate Time Series, ICML 2018

Traffic Prediction

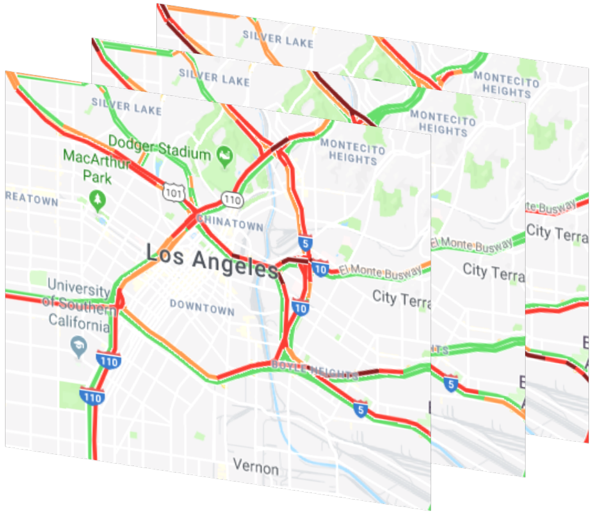
- Input: road network and past T’ traffic speed observed at sensors
- Output: traffic speed for the next T steps

Output: Predictions

Input: Observations



...



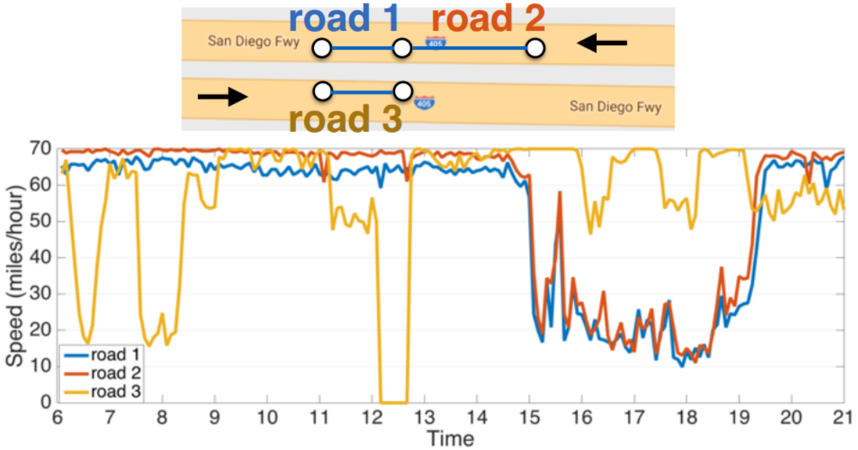
7:00 AM ... 8:00 AM 8:10AM, 8:20AM, …, 9:00 AM

Existing Work

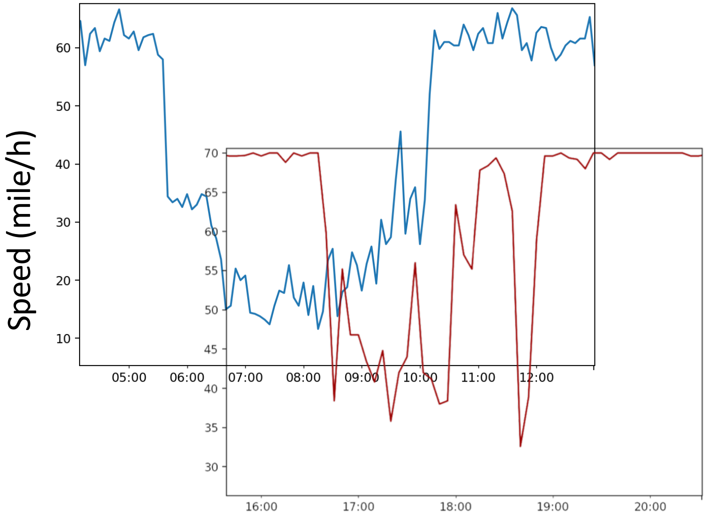
- KNN-based models
- Time series models
  - Seasonal Autoregressive Integrated Moving Average (S-ARIMA)
- Support vector regression
- Our prior work:
  - Latent space models: Dingxiong Deng et al, Latent Space Model for Road Networks to Predict Time-Varying Traffic. KDD, 2016
  - Mixture LSTM: Y. Qi et al, Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. SDM 2016

Challenges for Traffic Forecasting

Complex
Spatial Dependency

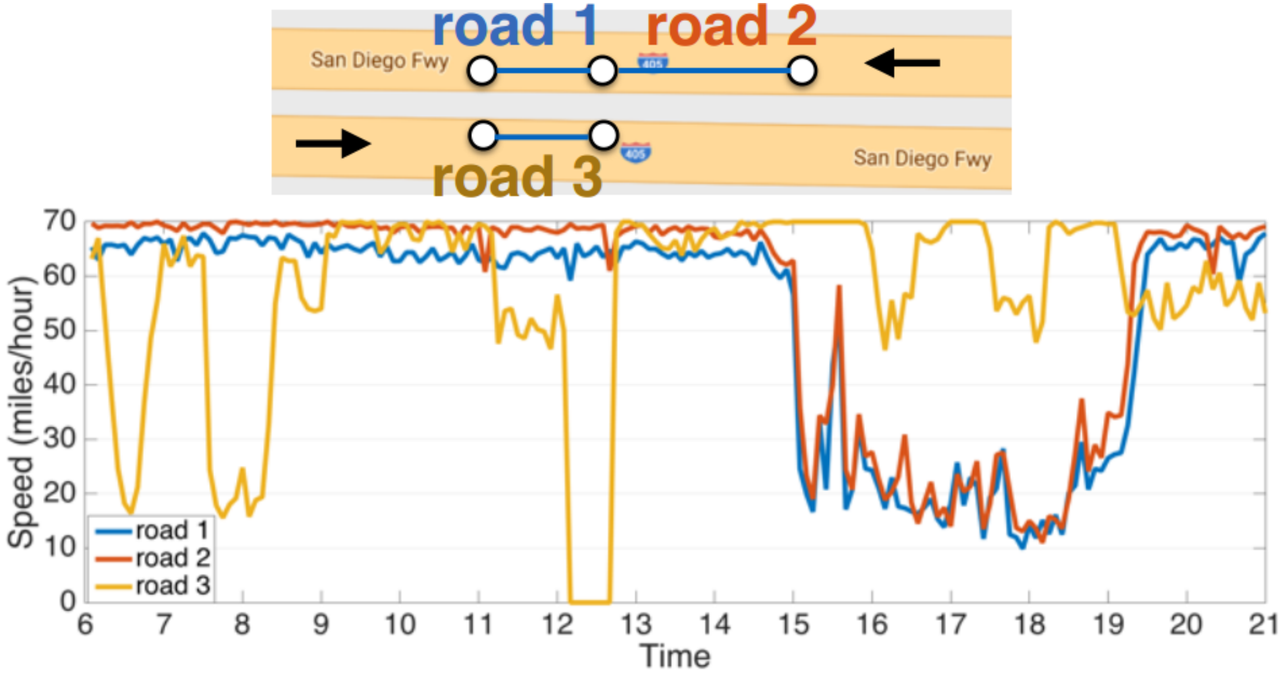


Non-linear, non-stationary
Temporal Dynamic



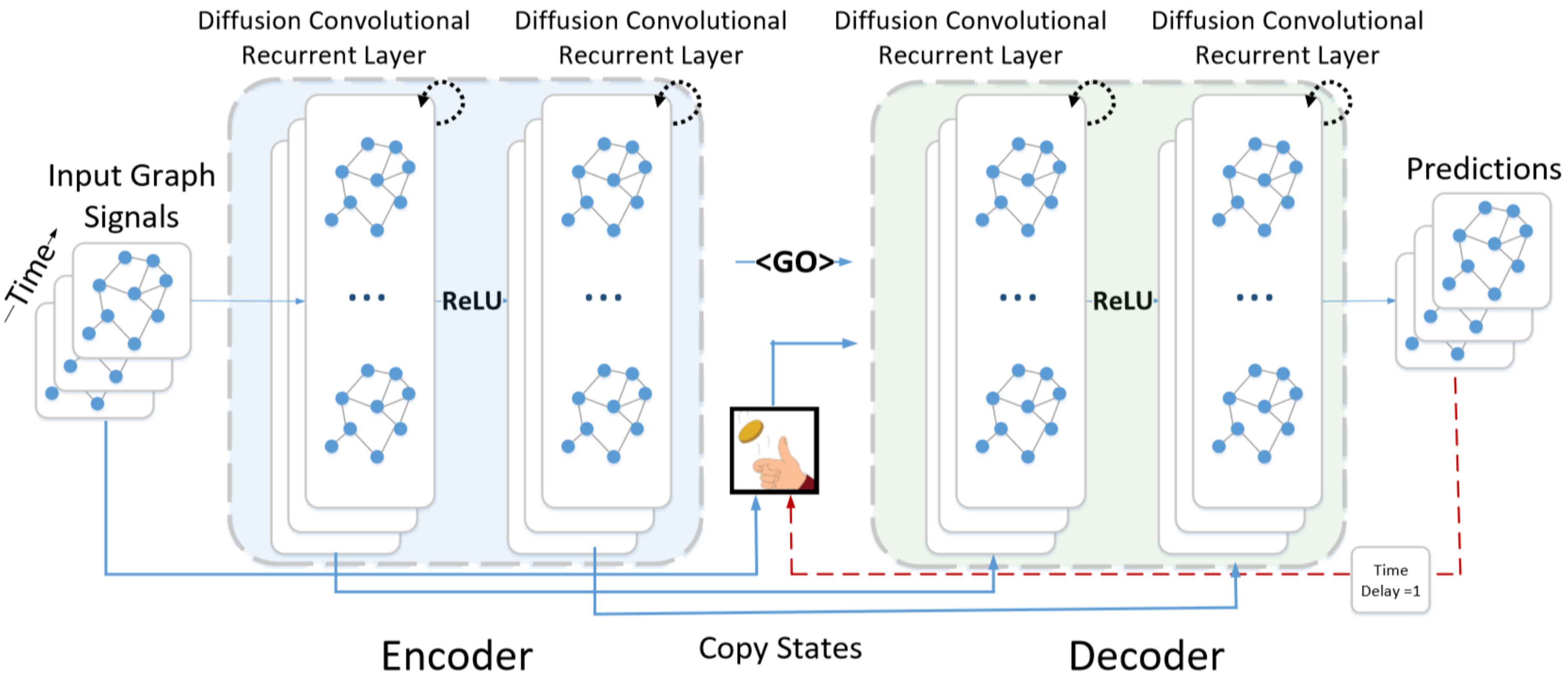
Challenges for Traffic Forecasting

- Spatial relationship among traffic flow is ***non-Euclidean*** and ***directed***



Traffic Forecasting with Convolution on Graph

- Model spatial dependency with proposed ***diffusion convolution on graph***



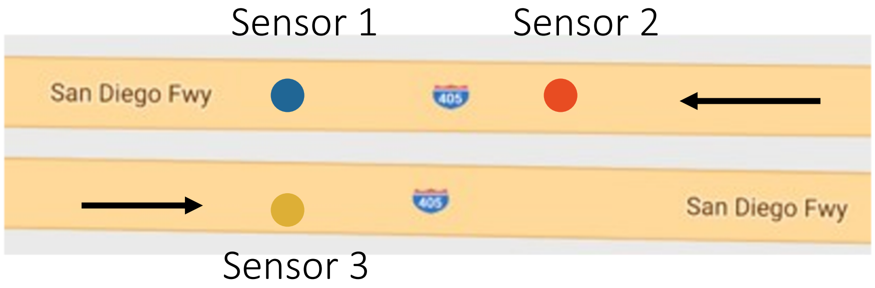
* Yaguang Li et al, Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting. ICLR, 2018

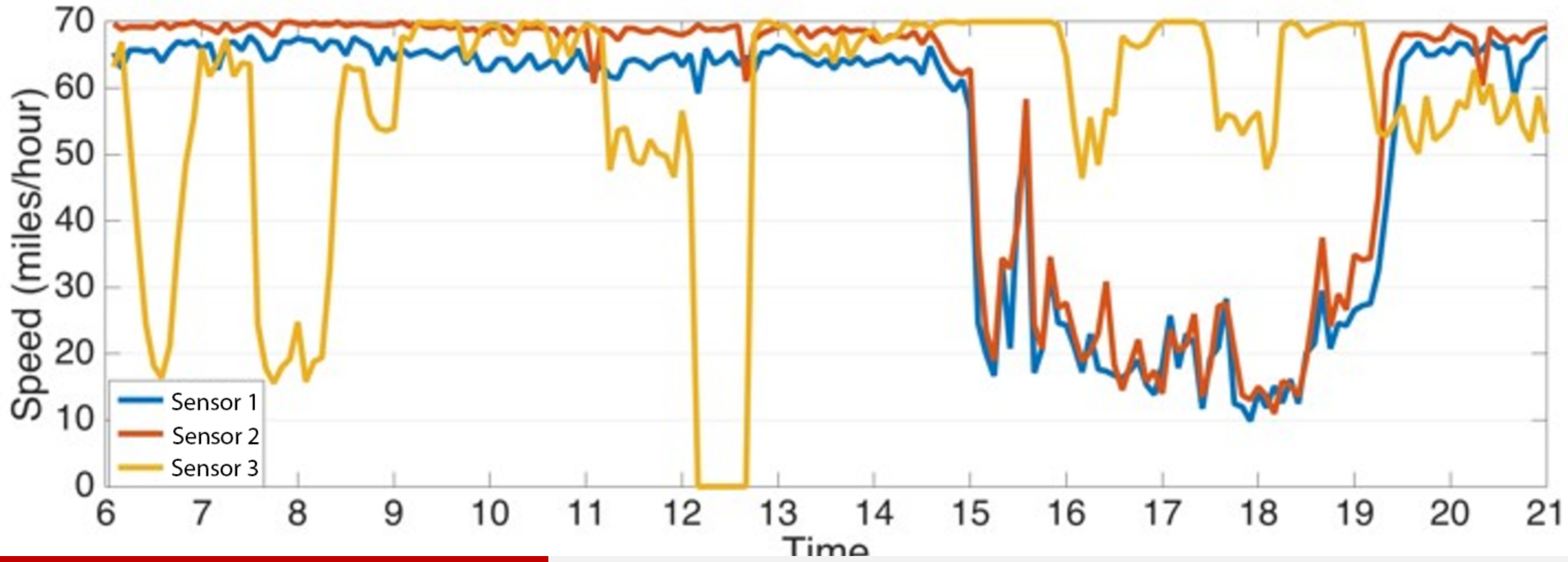
Spatial Dependency in Traffic Prediction

- Spatial dependency among traffic flow is ***non-Euclidean*** and ***directed***

Close in Euclidean space → Similar traffic speed

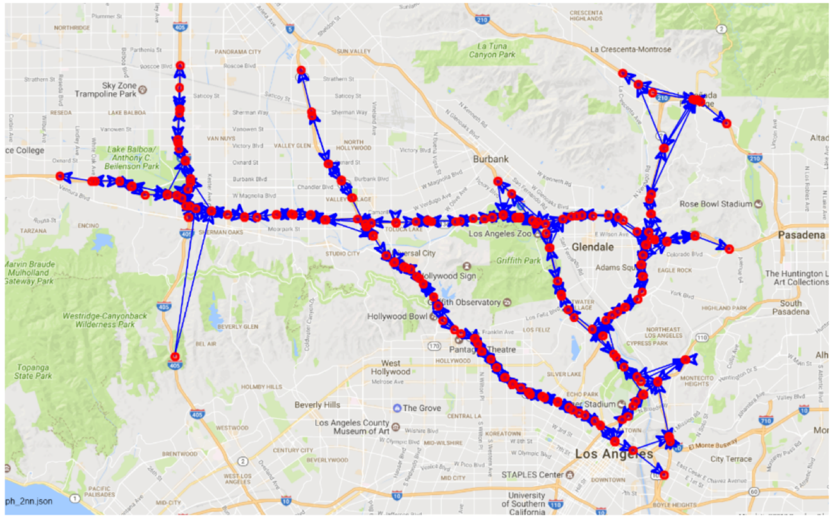
$$dist_{net}(v_i \rightarrow v_j) \neq dist_{net}(v_i \rightarrow v_j)$$





Spatial Dependency Modeling

- Model the network of traffic sensors, i.e., loop detectors, as a ***directed graph***
  - Graph $\mathcal{G} = (\mathbf{V}, \boldsymbol{A})$
  - Vertices $\boldsymbol{V}$: **o** sensors
  - Adjacency matrix $\boldsymbol{A}$: $\rightarrow$ weight between vertices

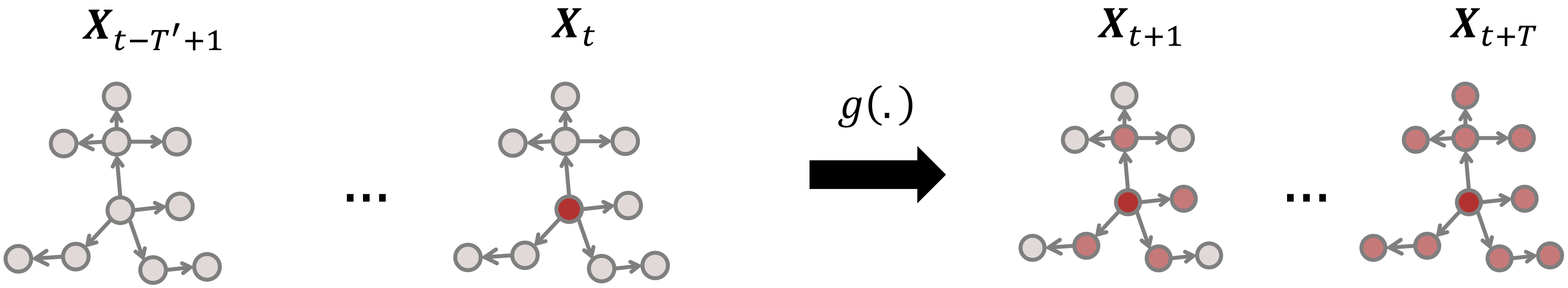


$$A_{ij} = \exp\left(-\frac{\mathrm{dist}_{\mathrm{net}}(v_i, v_j)^2}{\sigma^2}\right) \text{ if } \mathrm{dist}_{\mathrm{net}}(v_i, v_j) \leq \kappa$$

$\mathrm{dist}_{\mathrm{net}}(v_i, v_j)$: road network distance from v_i to v_j,
κ: threshold to ensure sparsity, σ^2 variance of all pairwise road network distances

Problem Statement

- Graph signal: $\boldsymbol{X}_{\mathbf{t}} \in \mathbb{R}^{|V| \times P}$, observation on $\boldsymbol{\mathcal{G}}$ at time t
  - $|\boldsymbol{V}|$: number of vertices
  - P : feature dimension of each vertex.
- **Problem Statement**: Learn a function $g(\cdot)$ to map T' historical graph signals to future T graph signals



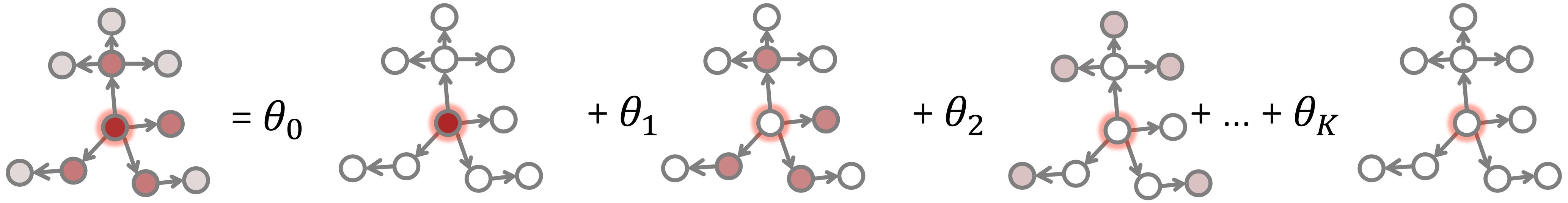
Generalize Convolution to Graph

- Diffusion convolution filter: combination of ***diffusion processes*** with different steps on the graph.

Transition matrices of the diffusion process

$$\boldsymbol{X}_{:,p} \star_{\mathcal{G}} f_{\theta} = \sum_{k=0}^{K-1} \left(\theta_k \left(\boldsymbol{D}_{\boldsymbol{O}}^{-1} \boldsymbol{A} \right)^k \right) \boldsymbol{X}_{:,p}$$

Learning complexity: $O(K)$



Example diffusion filter Centered at

0 Step Diffusion

1 Step Diffusion

2 Step Diffusion

K Step Diffusion

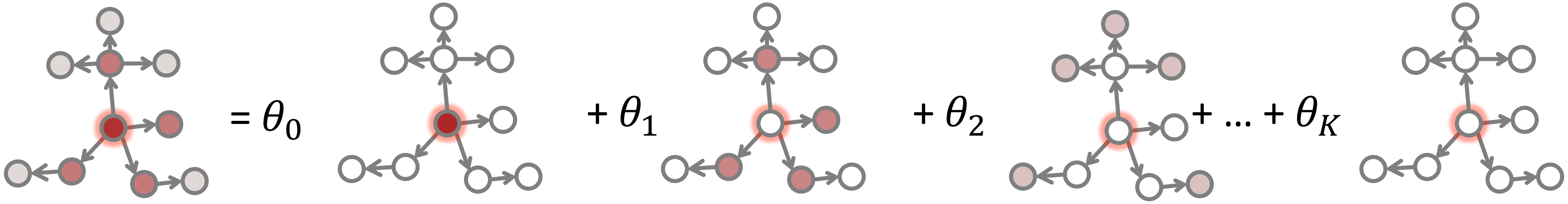
$\star_{\mathcal{G}}$: diffusion convolution, D_O: diagonal out-degree matrix.

Generalize Convolution to Graph

- Diffusion convolution filter: combination of ***diffusion processes*** with different steps on the graph.

Dual directional diffusion to model upstream and downstream separately

$$\boldsymbol{X}_{:,p} \star_{\mathcal{G}} f_{\theta} = \sum_{k=0}^{K-1} \left(\theta_{k,1} \left(\boldsymbol{D}_{\boldsymbol{O}}^{-1} \boldsymbol{A} \right)^{k} + \theta_{k,2} \left(\boldsymbol{D}_{\boldsymbol{I}}^{-1} \boldsymbol{A}^{\intercal} \right)^{k} \right) \boldsymbol{X}_{:,p}$$



Example diffusion filter Centered at

0 Step Diffusion

1 Step Diffusion

2 Step Diffusion

K Step Diffusion

$\star_{\mathcal{G}}$: diffusion convolution, D_o: diagonal out-degree matrix, D_I: diagonal in-degree matrix

Advantage of Diffusion Convolution

$$\boldsymbol{X}_{:,p} \star_{\mathcal{G}} f_\theta = \sum_{k=0}^{K-1} \left(\theta_{k,1} \left(\boldsymbol{D}_{\boldsymbol{O}}^{-1} \boldsymbol{A} \right)^k + \theta_{k,2} \left(\boldsymbol{D}_{\boldsymbol{I}}^{-1} \boldsymbol{A}^\intercal \right)^k \right) \boldsymbol{X}_{:,p}$$

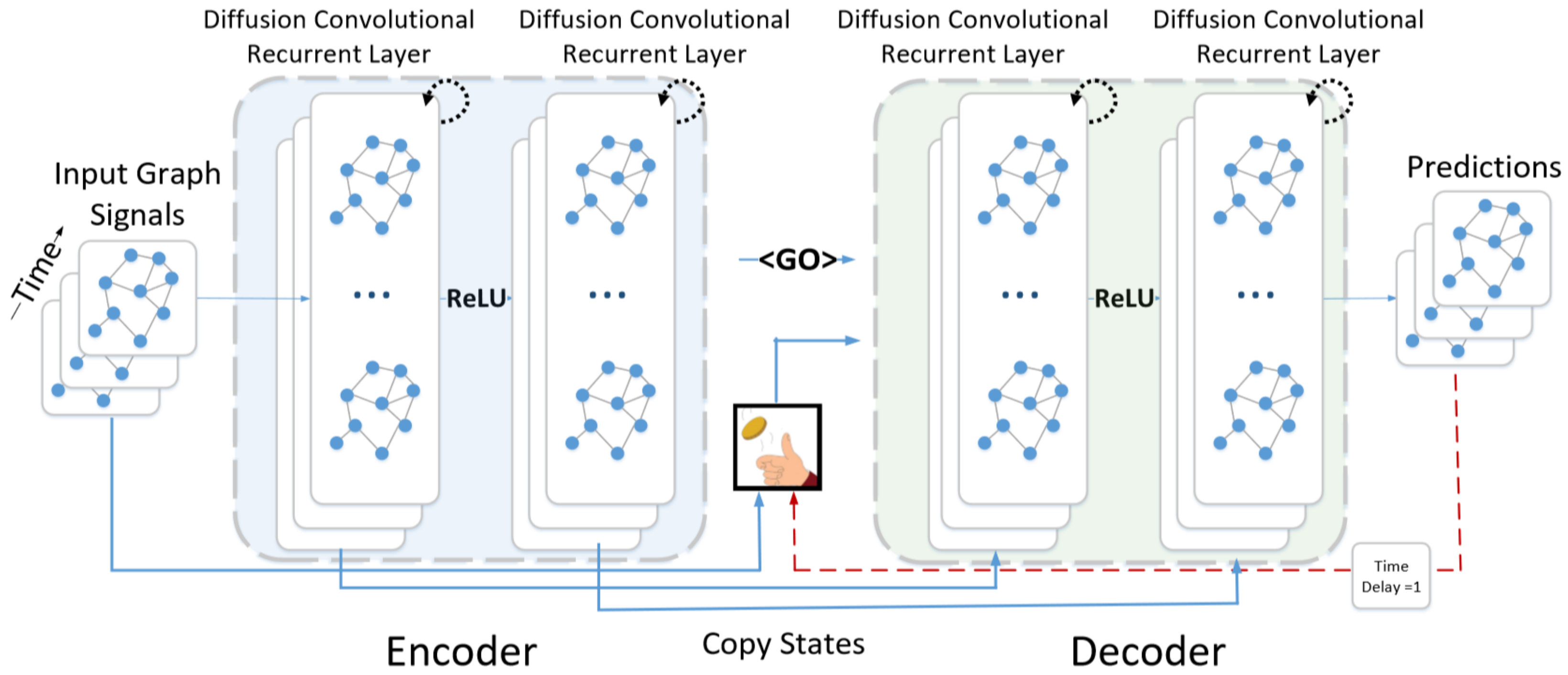
- Efficient
  - Learning complexity: $O(K)$
  - Time complexity: $O(K|E|)$, $|E|$ number of edges
- Expressive
  - Many popular convolution operations, including the ChebNet [Defferrard et al., NIPS '16], can be seen as special cases of the diffusion convolution [Li et al. ICLR '18].

$\star_{\mathcal{G}}$: diffusion convolution, D_O: diagonal out-degree matrix, D_I: diagonal in-degree matrix

* Defferrard, M et al, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, NIPS, 2016
* Yaguang Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR, 2018

Diffusion Convolutional Recurrent Neural Network

- Diffusion Convolutional Recurrent Neural Network (DCRNN)
  - Model spatial dependency with ***diffusion convolution***
  - Sequence to sequence learning with ***encoder-decoder*** framework



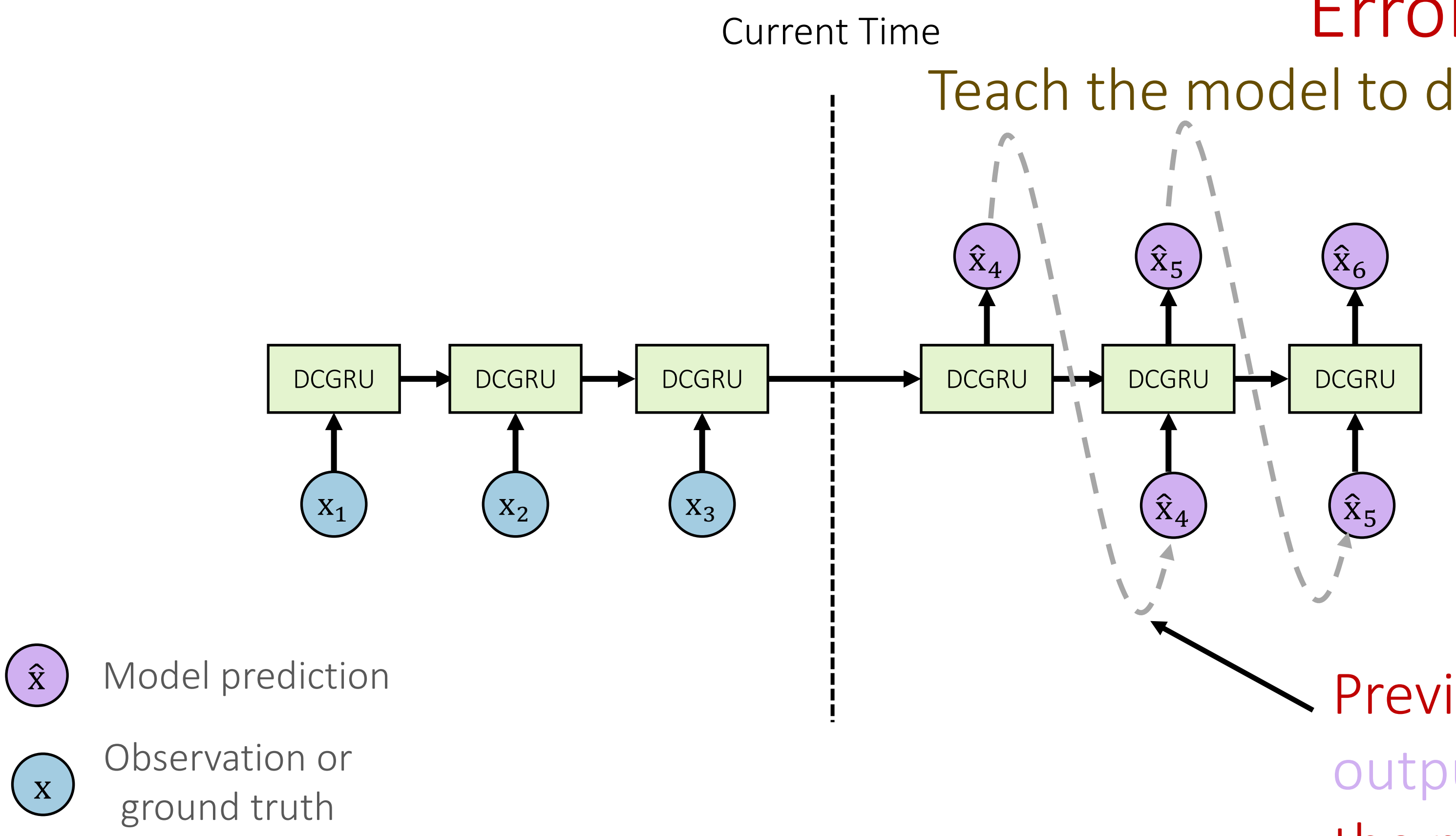
* Yaguang Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018

Model Temporal Dynamics using Recurrent Neural Network

Multi-step ahead prediction with RNN

Error Propagation

Teach the model to deal with its own error.



Previous model output is fed into the network

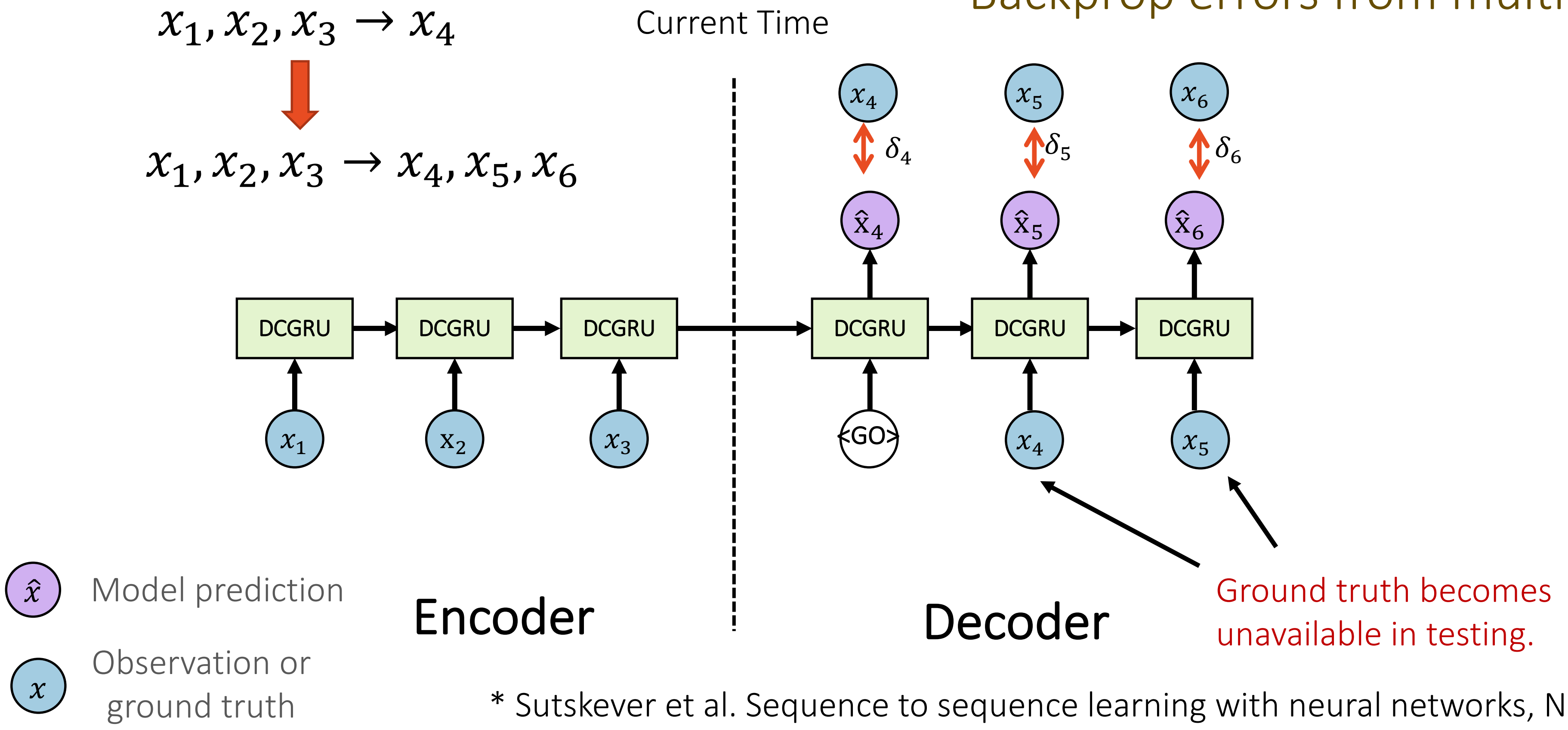
Improve Multi-step ahead Forecasting

- Traffic prediction as a ***sequence to sequence*** learning problem
  - Encoder-decoder framework

$x_1, x_2, x_3 \rightarrow x_4$

$x_1, x_2, x_3 \rightarrow x_4, x_5, x_6$

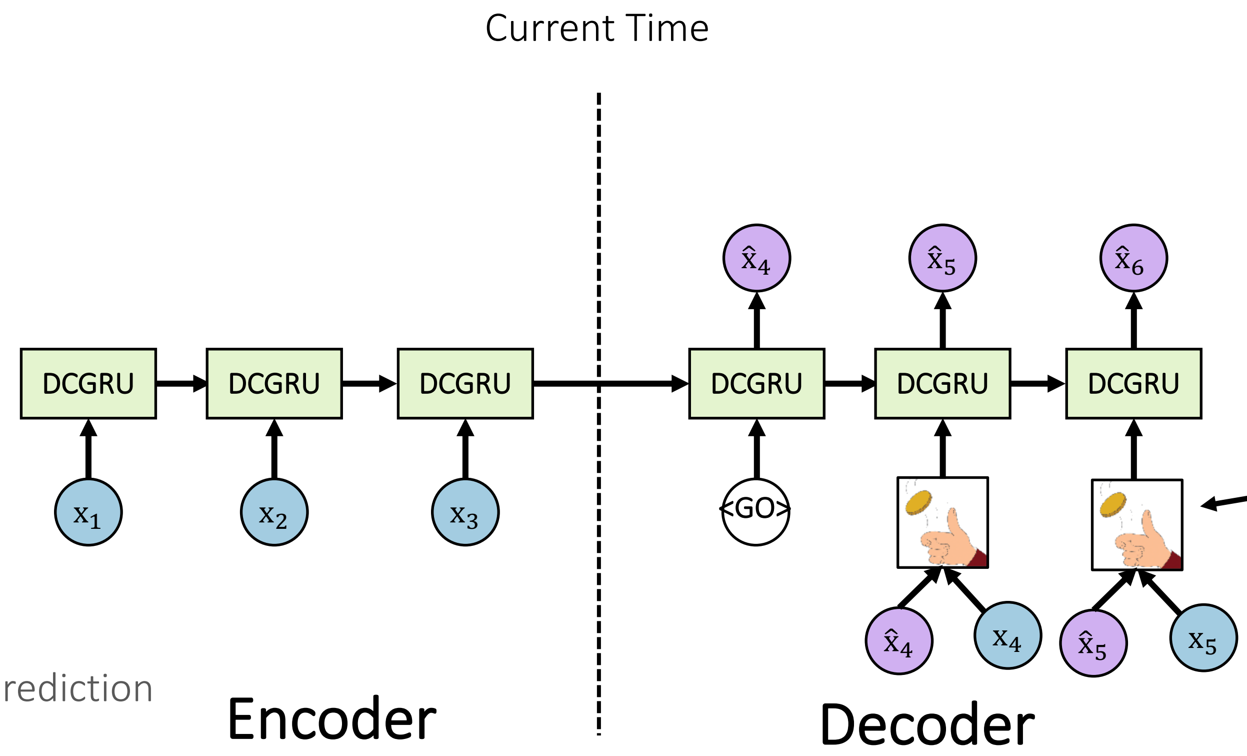
Backprop errors from multiple steps.



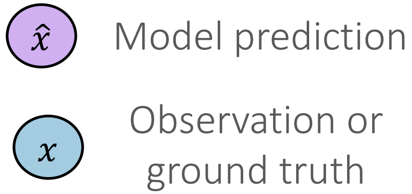
* Sutskever et al. Sequence to sequence learning with neural networks, NIPS 2014

Improve Multi-step ahead Forecasting

- Improve multi-step ahead forecasting with ***scheduled sampling***



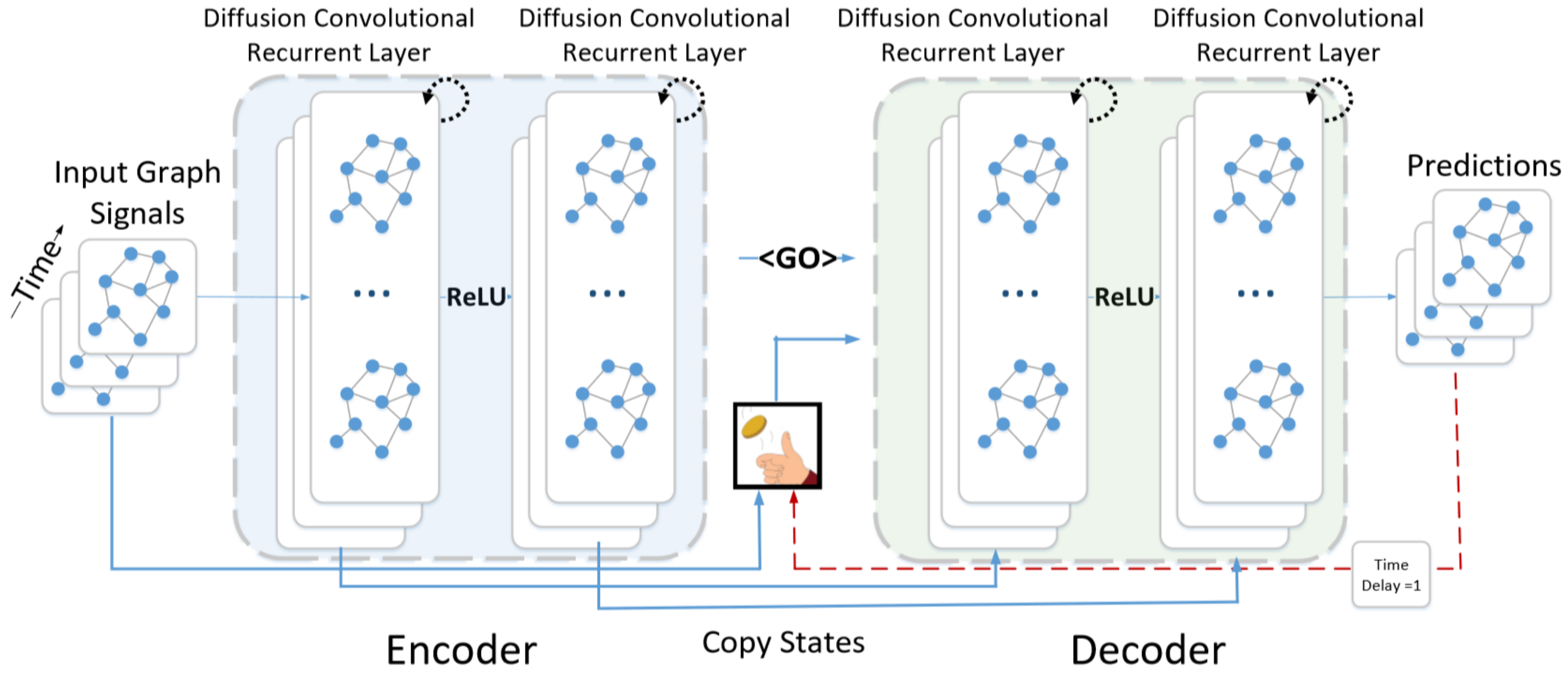
Scheduled sampling:
Choose to use the previous *ground truth* or *model prediction* by flipping a coin



* Bengio,Samy et al. Scheduled sampling for sequence prediction with recurrent neural networks. NIPS 2015

Diffusion Convolutional Recurrent Neural Network

- Diffusion Convolutional Recurrent Neural Network (DCRNN)
  - Model spatial dependency with ***diffusion convolution***
  - Sequence to sequence learning with ***encoder-decoder*** framework
  - Improve multi-step ahead forecasting with ***scheduled sampling***

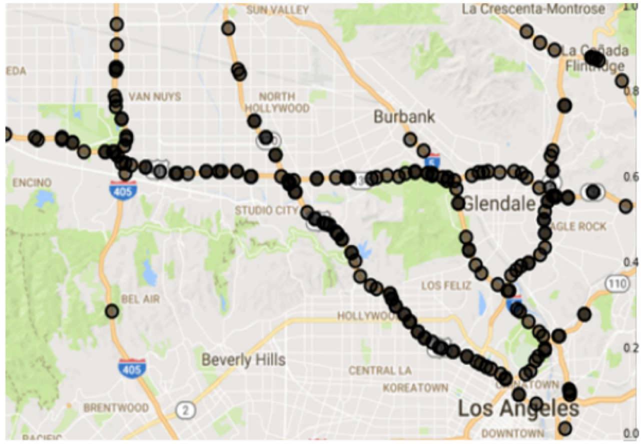


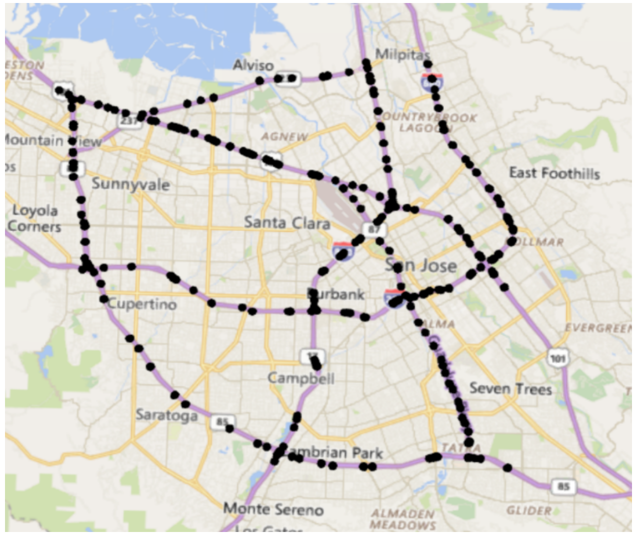
* Yaguang Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018

Experiment - Datasets

- METR-LA:
  - 207 traffic sensors in Los Angeles
  - 4 months in 2012
  - 6.5M observations

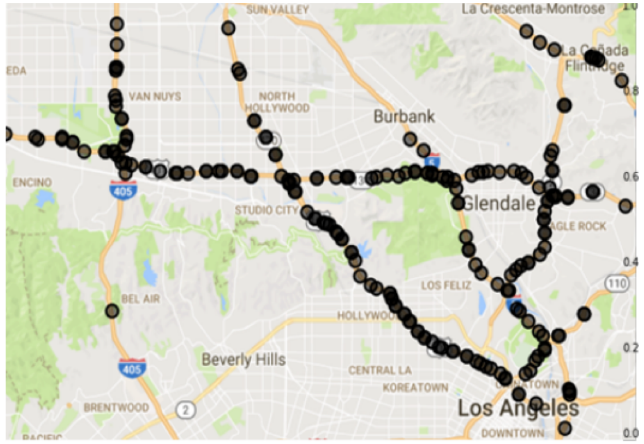
- PEMS-BAY:
  - 345 traffic sensors in Bay Area
  - 6 months in 2017
  - 17M observations

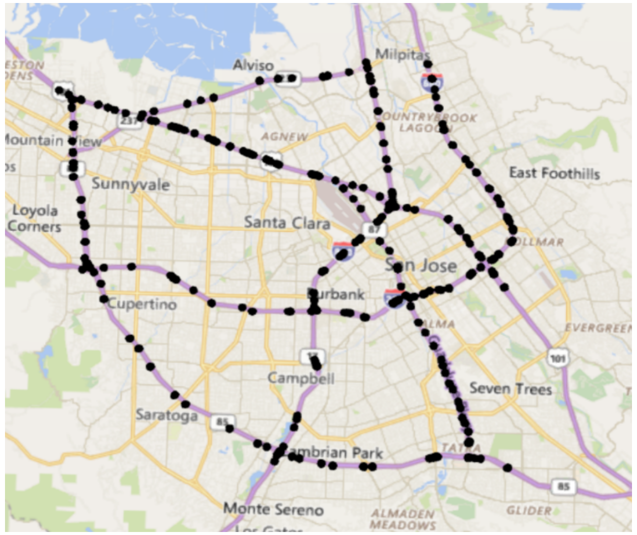




Experiments

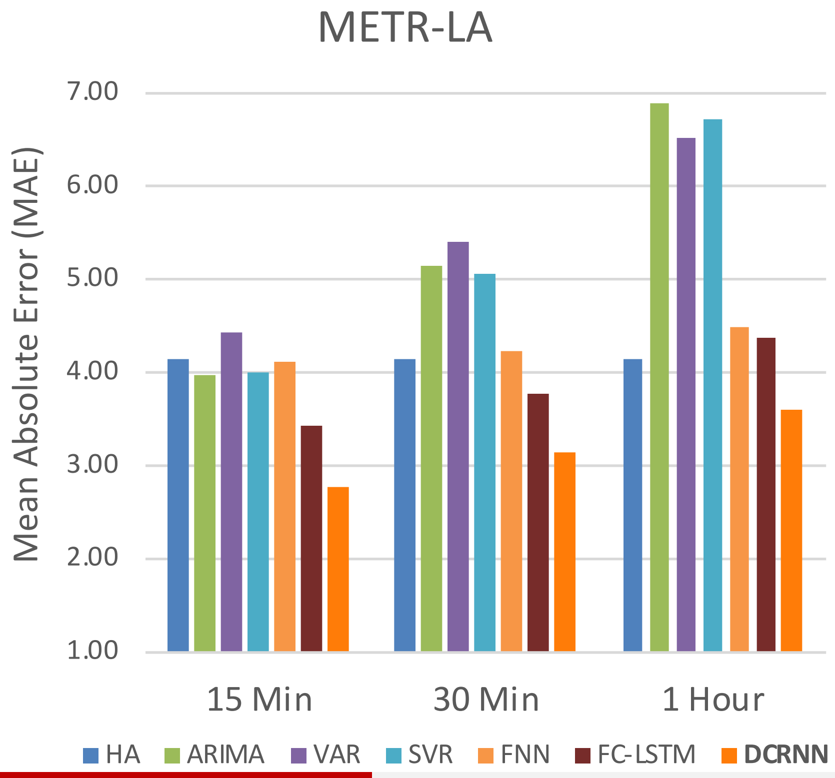
- Baselines
  - Historical Average (HA)
  - Autoregressive Integrated Moving Average (ARIMA)
  - Support Vector Regression (SVR)
  - Vector Auto-Regression (VAR)
  - Feed forward Neural network (FNN)
  - Fully connected LSTM with Sequence to Sequence framework (FC-LSTM)
- Task
  - Multi-step ahead traffic speed forecasting

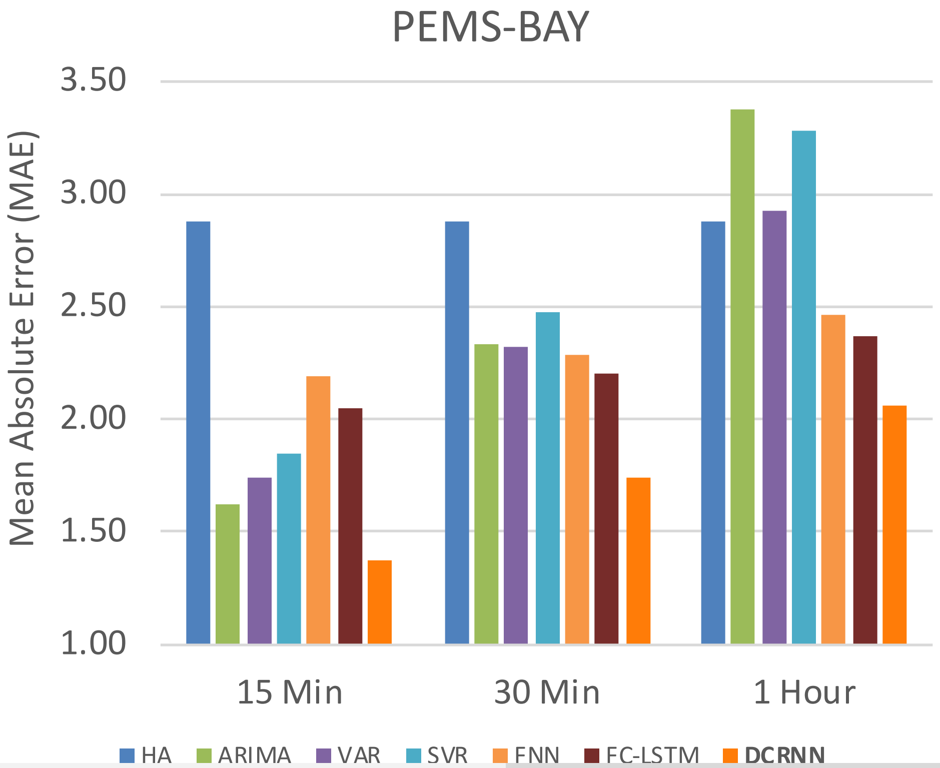




Experimental Results

- DCRNN achieves the ***best performance*** for all forecasting horizons for both datasets

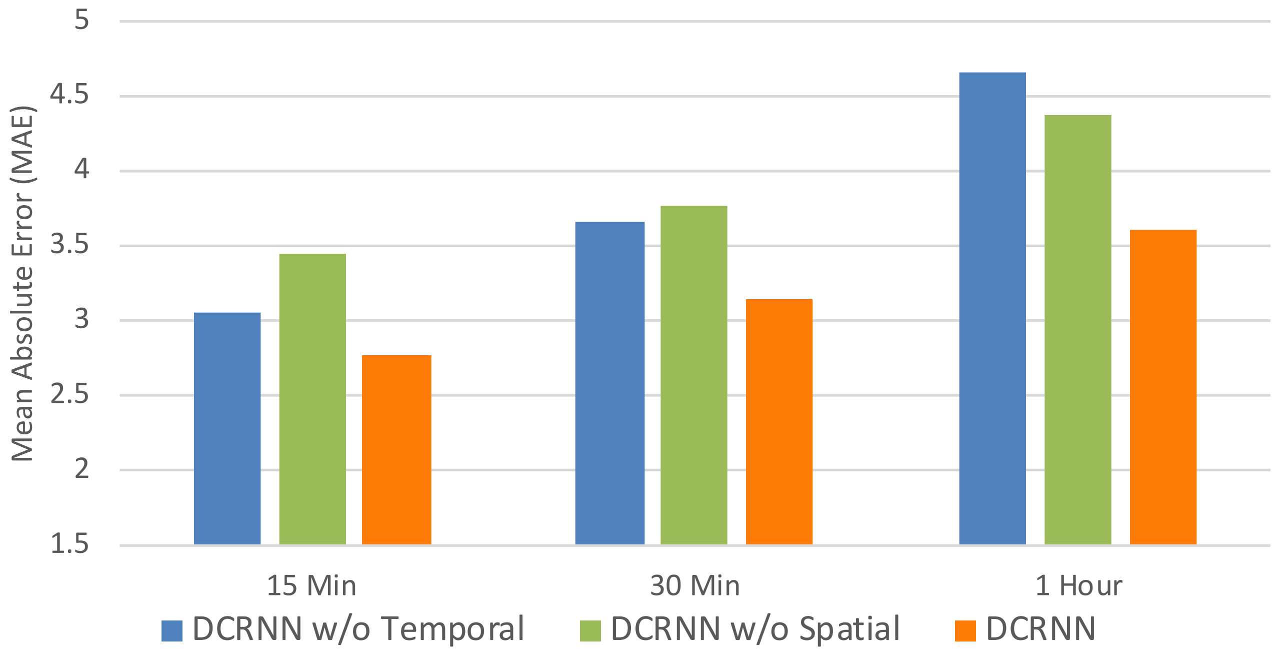




Effects of Spatiotemporal Dependency Modeling

- **w/o temporal**: removing sequence to sequence learning.
- **w/o spatial**: remove the diffusion convolution.

Removing either spatial or temporal modeling results in ***significantly worse*** results.



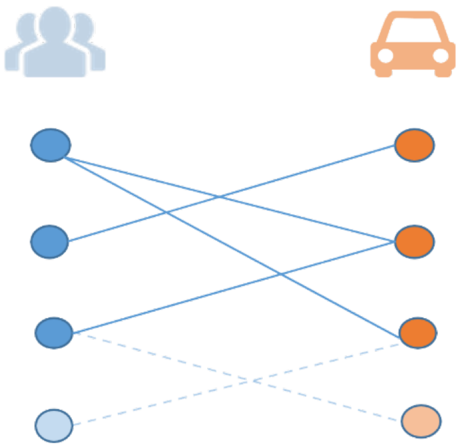
Outline

- Traffic estimation and forecasting
  - Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018
- Demand forecasting
  - Li et al, Spatiotemporal Multi-Graph Convolution for Ride-hailing Demand Forecasting, AAAI 2019
- Multi-rate multi-resolution forecasting/interpolation
  - Che et al, Hierarchical Deep Generative Models for Multi-Rate Multivariate Time Series, ICML 2018

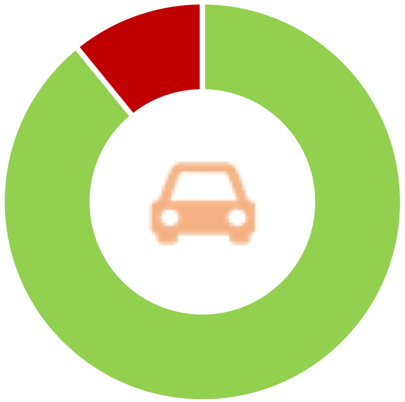
Introduction

- More than 18 billion ride-hailing trips worldwide in 2018*
  - Twice as much as the world population.
- Benefit of better ride-hailing demand forecasting

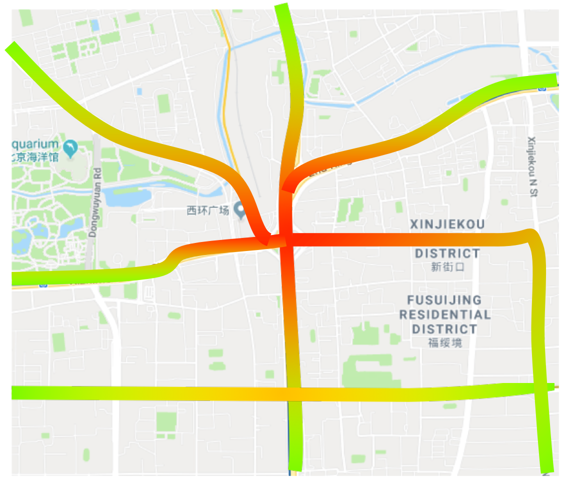
Better Vehicle Dispatching



Higher vehicle utilization



Early congestion warning

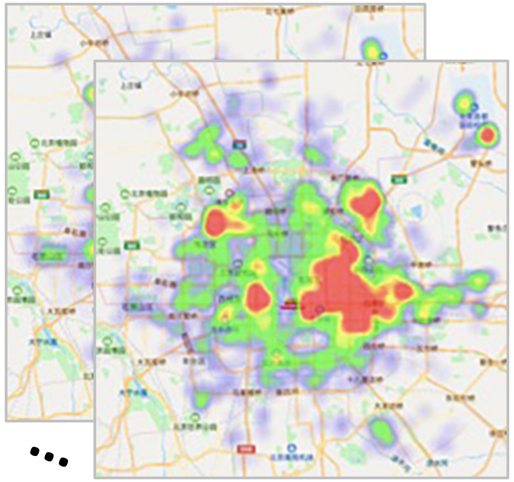


* http://www.businessofapps.com/data/uber-statistics/, Nov 2018.

Region-level Ride-hailing Demand Forecasting

- Input: past T observations of demands of all $|V|$ regions
- Output: demands of all $|V|$ regions in the next time stamp

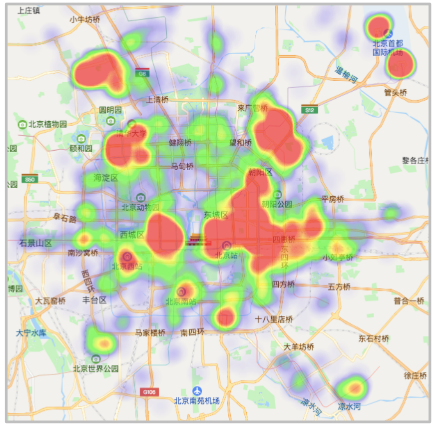
Input



$\mathbb{R}^{T\times|V|}$

$$f:\ \mathbb{R}^{T\times|V|} \rightarrow \mathbb{R}^{|V|}$$

Output



$\mathbb{R}^{|V|}$

Complicated spatial and temporal correlations

Related Work

- Spatiotemporal forecasting on grid
  - Classical settings for demand forecasting problem
  - CNN-based approaches: region-wise relationship is Euclidean
    - DeepST/STResNet: Crowd flow forecasting (Zhang et al., 2017)
    - DMVST: Demand forecasting (Yao et al., 2018)

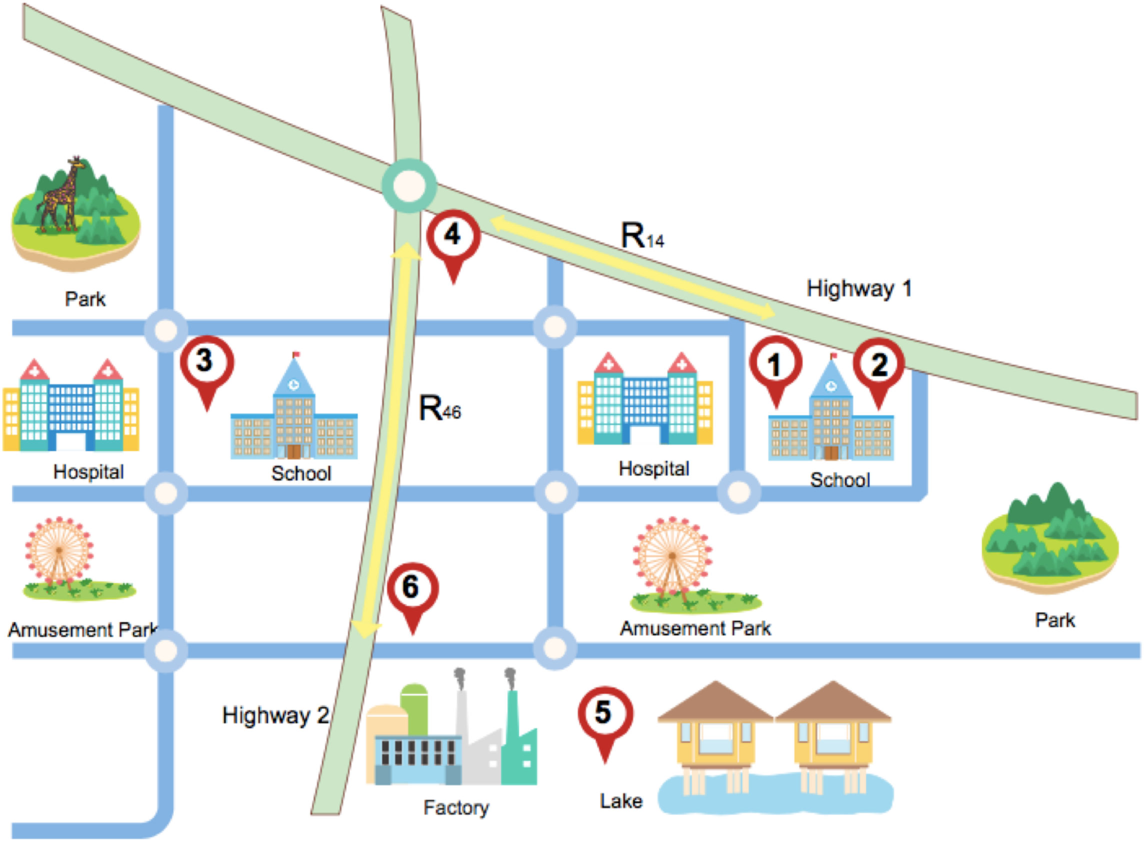
Hard to capture the **non-Euclidean** correlations

- Spatiotemporal forecasting on graph
  - LinUOTD: handcrafted feature + LR for demand forecasting (Tong et al., 2017)
  - DCRNN/ST-GCN: Graph convolution based traffic forecasting (Li et al., 2018a, Yu et al., 2018, Li et al., 2018b, Yan et al., 2018)

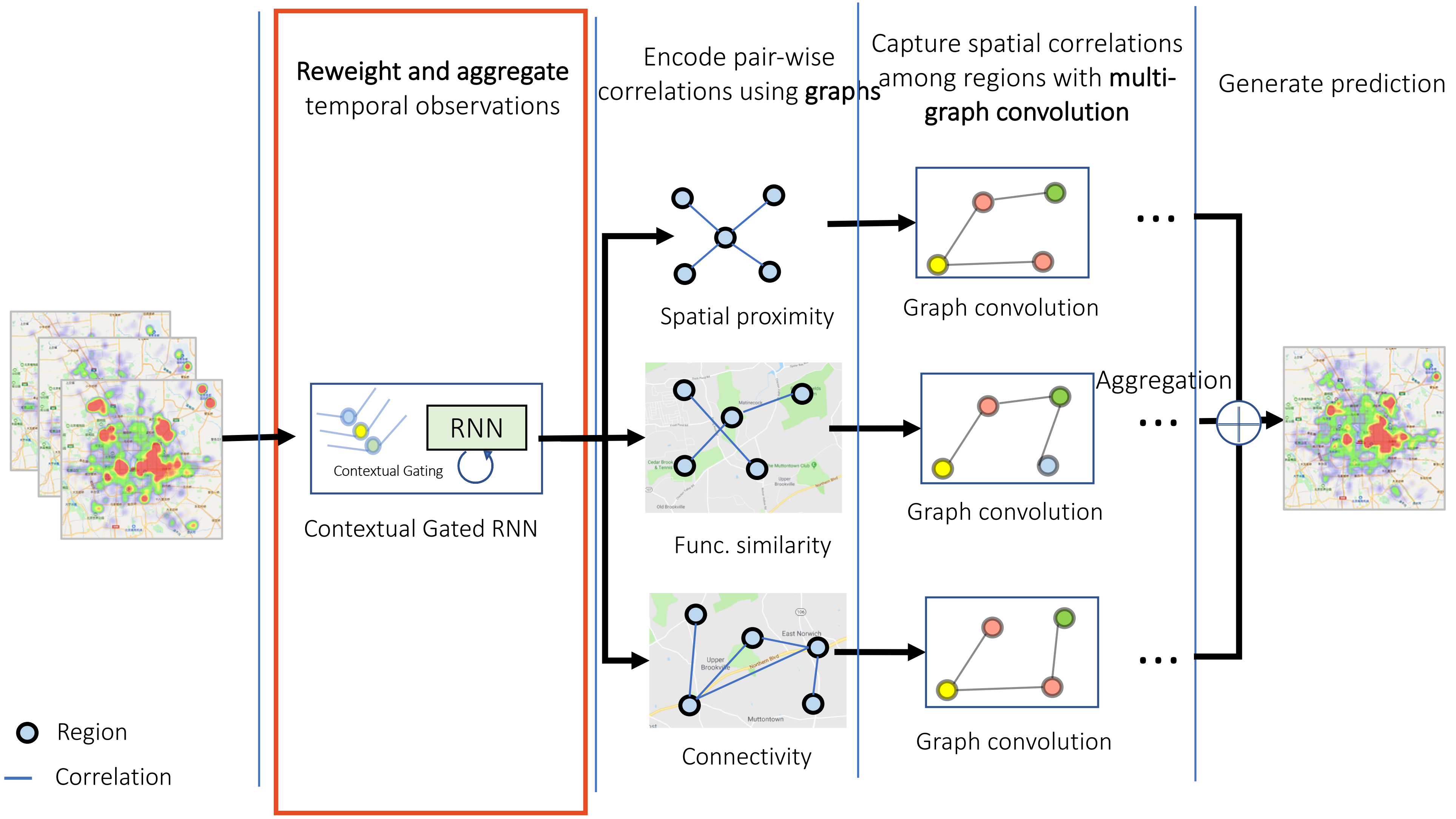
Hard to capture the **multimodal** correlations

Multimodal Correlations among Regions

- Spatial proximity
  - Region 1 and 2
- Functional similarity
  - Regions with similar context show similar demand patterns
  - Region 1 and 3
- Road connectivity
  - High-speed transportation facilitate correlation
  - Region 1 and 4

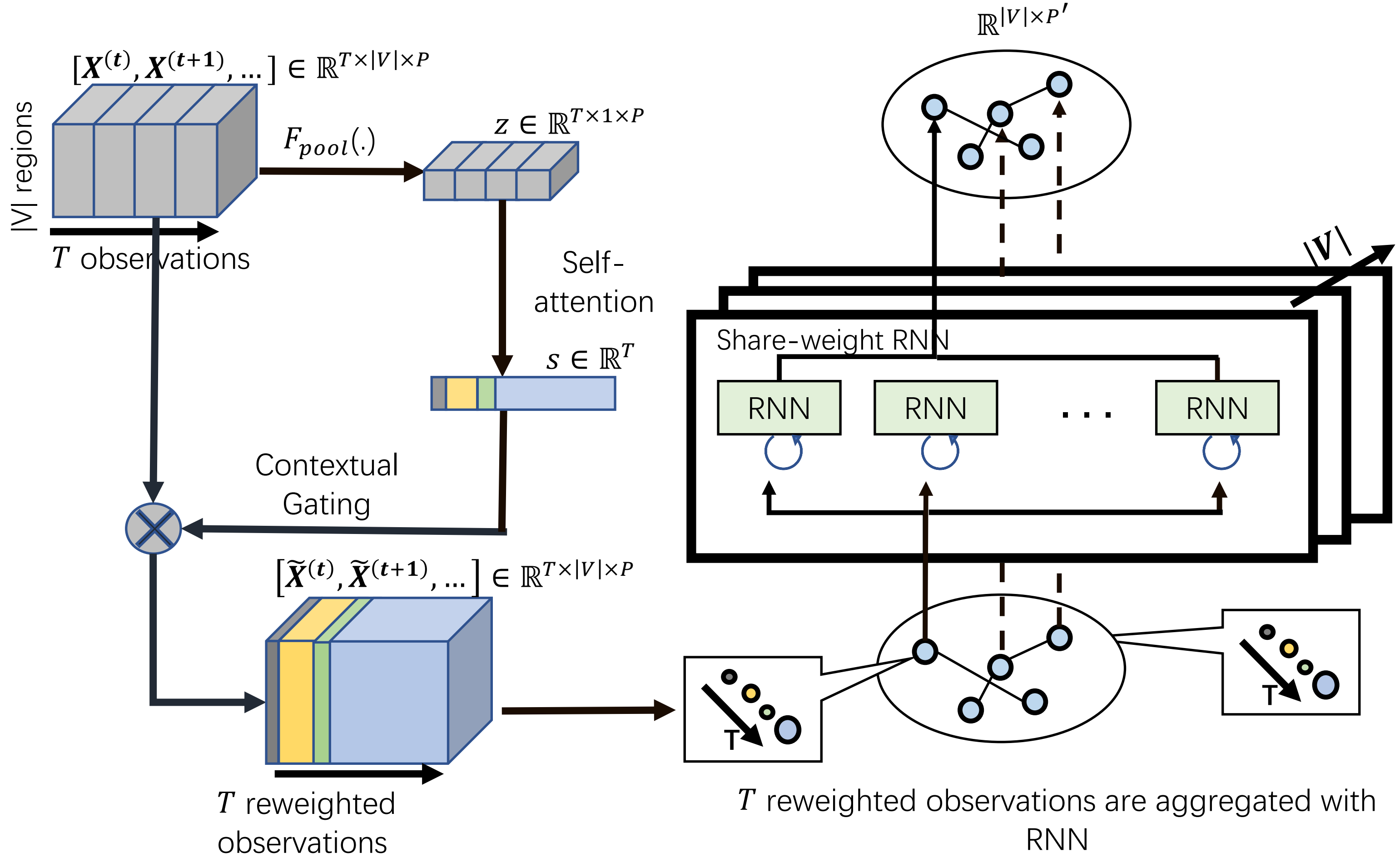


Spatiotemporal Multi-Graph Convolution Network

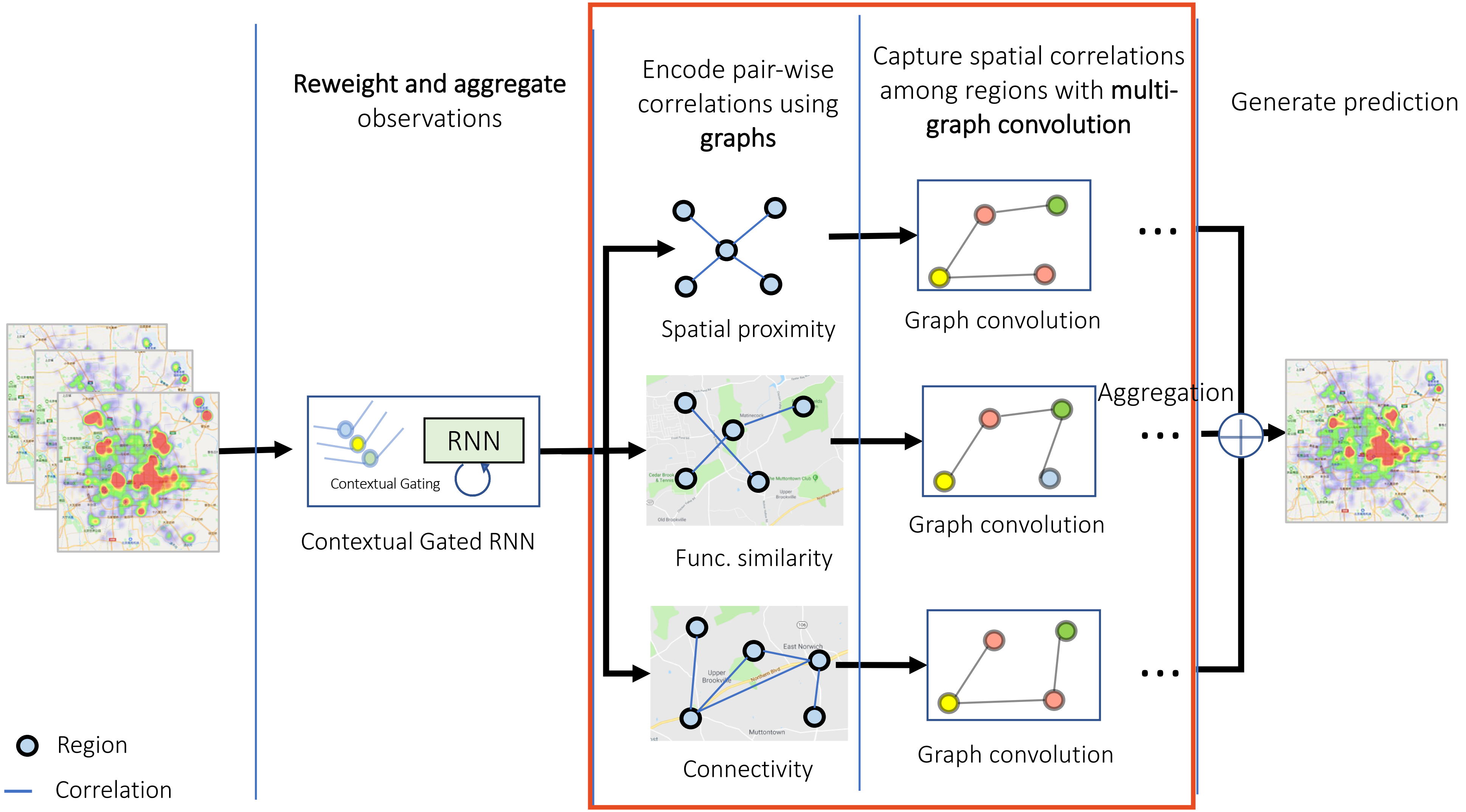


CGRNN: Context-aware Temporal Aggregation

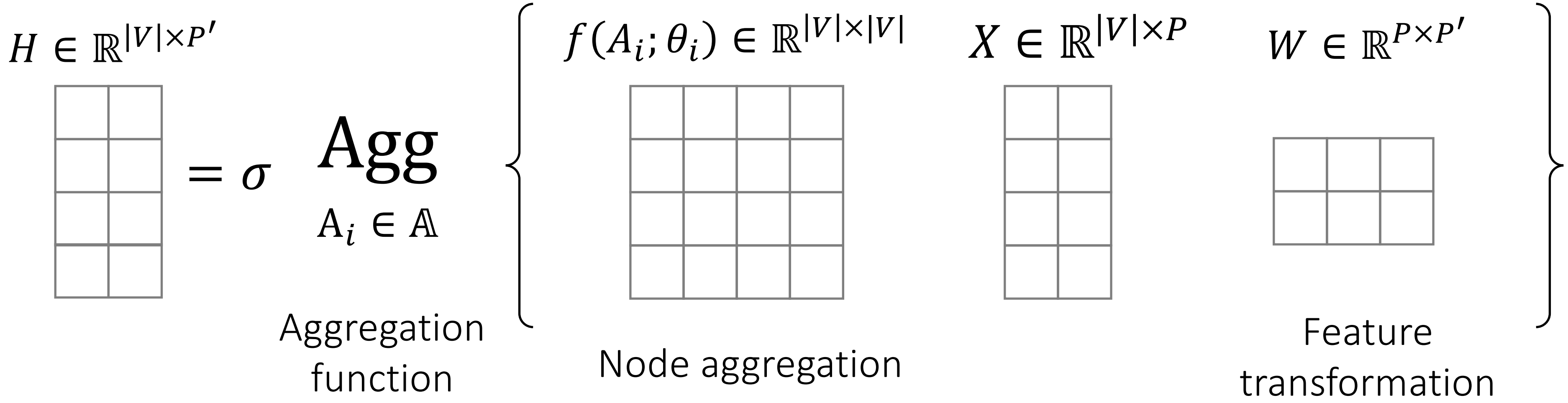
- Summarize contextual information
- Calculate gates based on interdependencies between observations with self-attention
- Reweight observations with gates
- Aggregate reweighted observations with share-weight RNN



Spatiotemporal Multi-Graph Convolution Network



$$H = \mathrm{MGC}(X) = \sigma\left(\underset{\mathrm{A}_i \in \mathbb{A}}{\mathrm{Agg}}\, f(A_i;\theta_i) X W\right)$$

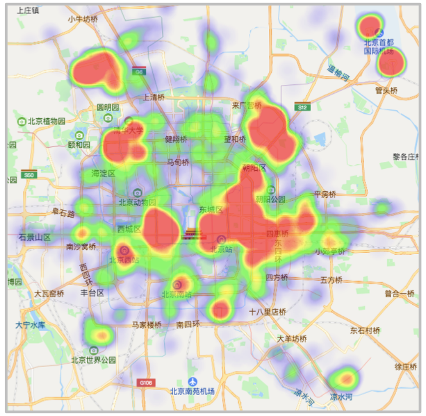


- $f(A_i;\theta_i)$: function of adjacency matrix A_i with parameter θ_i
  - Polynomial of graph Laplacian, graph attention etc.
- Agg: Aggregation function
  - Sum, average, attention-based aggregation

Datasets

- Beijing:
  - 1296 regions, 19M samples
  - 10 months in 2017
- Shanghai
  - 896 regions, 13M samples
  - 10 months in 2017
- POI/Road network
  - OpenStreetMap

Beijing



Shanghai

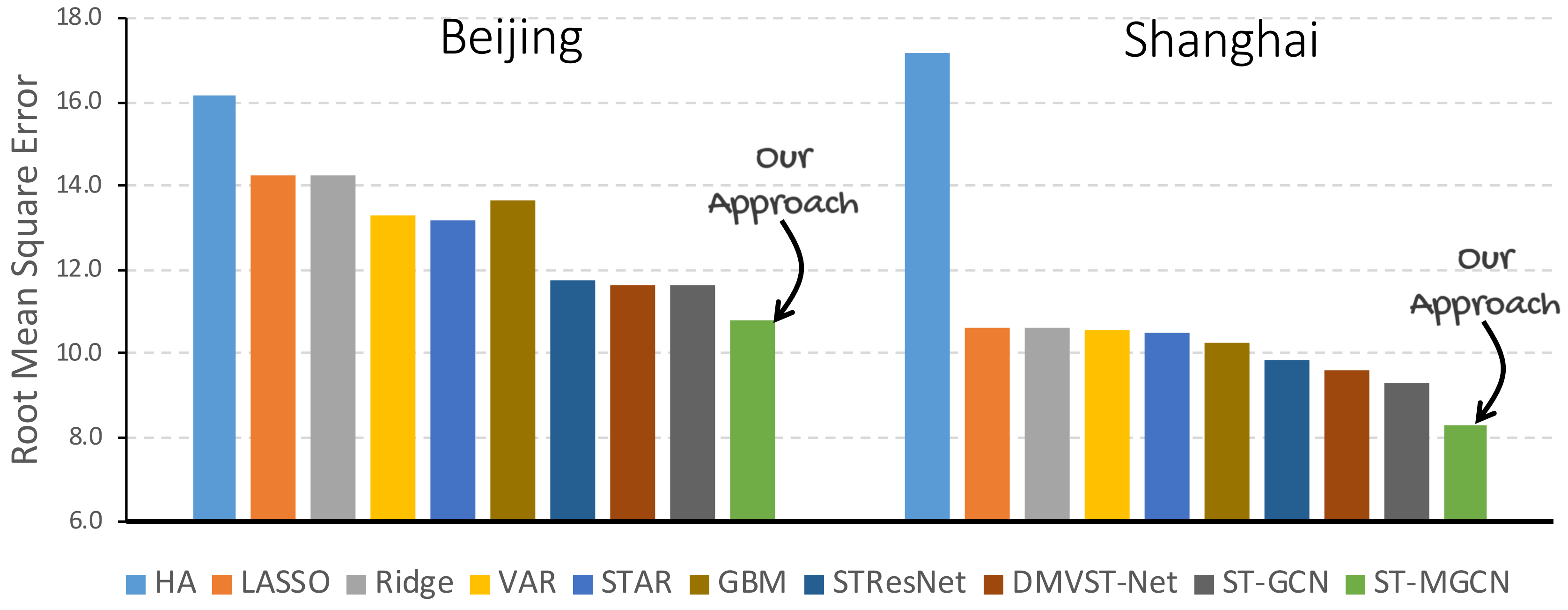
Experiments

- Baselines
  - Historical Average (HA)
  - Linear Regression (LASSO, Ridge)
  - Vector Auto-Regression (VAR)
  - Spatiotemporal Auto-Regressive Model (STAR)
  - Gradient Boosted Machine (GBM)

  - Spatiotemporal Residual Network (ST-ResNet), with Euclidean grid
  - Spatiotemporal graph convolutional network (ST-GCN), with road network graph
  - Deep Multi-view Spatiotemporal Network (DMVST-Net), with Euclidean grid, SOTA for ride-hailing demand forecasting
- Task
  - One step ahead ride-hailing demand forecasting

Experimental Results

- ST-MGCN achieves the **best performance** on both datasets
  - 10+% improvement*.

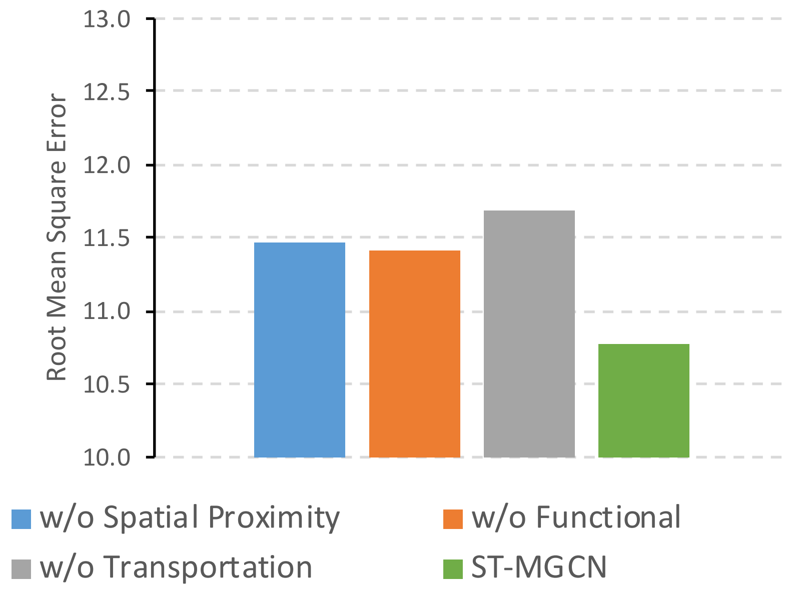


10+% improvement on **real-world large-scale datasets**

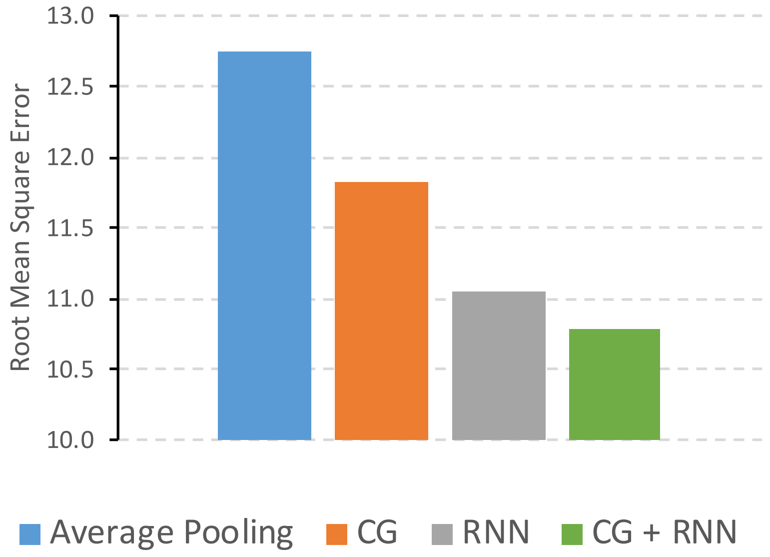
* In terms of relative error reduction of RMSE.

Experimental Results

- Both spatial and temporal correlations modeling are necessary
  - Removing either graph component leads to **significantly worse** performance.
  - With **CGRNN**, ST-MGCN achieves the best performance.



Effect of spatial correlation modeling



Effect of temporal correlation modeling

Outline

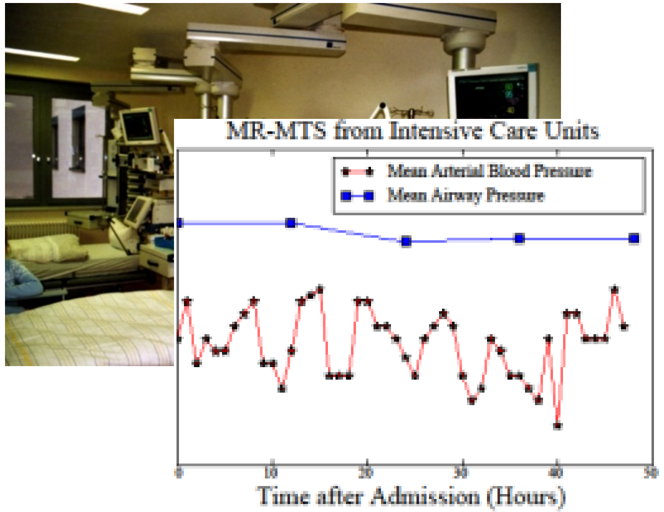
- Traffic estimation and forecasting
  - Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018
- Demand forecasting
  - Li et al, Spatiotemporal Multi-Graph Convolution for Ride-hailing Demand Forecasting, AAAI 2019
- Multi-rate multi-resolution forecasting/interpolation
  - Che et al, Hierarchical Deep Generative Models for Multi-Rate Multivariate Time Series, ICML 2018

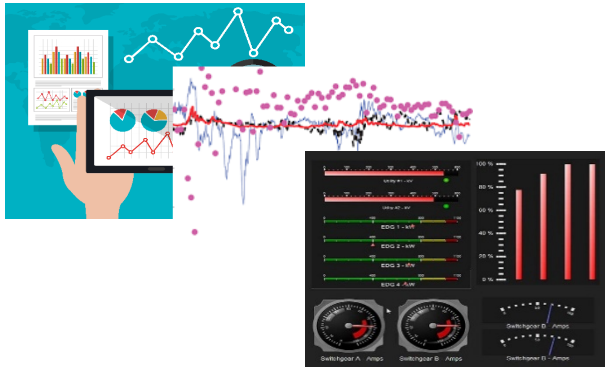
Outline

- Traffic estimation and forecasting
  - Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018
- Demand forecasting
  - Li et al, Spatiotemporal Multi-Graph Convolution for Ride-hailing Demand Forecasting, AAAI 2019
- Multi-rate multi-resolution forecasting/interpolation
  - Che et al, Hierarchical Deep Generative Models for Multi-Rate Multivariate Time Series, ICML 2018

Introduction

- Multivariate Time Series (MTS) -- many real-world applications
  - Healthcare, climate, traffic, financial forecasting, engineering…





- One of the key challenges -- Multi-Rate Multivariate Time Series (MR-MTS)
  - Different sampling rates
  - Multiple data sources / sensors

Motivation

- Major challenges of modeling MR-MTS
  - Need to handle *different* sampling rates
  - *Multi-scale* temporal dependencies
  - Complex underlying *generation* mechanism

- Existing solutions to MR-MTS forecasting/interpolation problems
  - *Single-rate* model? *(Kalman filter, VAR, deep Markov models, ...)*
    - Ignoring dependencies across different rates
  - Simple *imputations*? *(mean-imputation, Spline, MICE, MissForest, ...)*
    - May introduce unrelated/hide necessary dependencies
  - Multi-rate *discriminative* models? *(PLSTM, HM-RNN, ...)*
    - Not able to learn how the data is generated

Motivation

- Major challenges of modeling MR-MTS
  - Need to handle *different* sampling rates
  - *Multi-scale* temporal dependencies
  - Complex underlying *generation* mechanism

- Key point
  - To learn the **latent hierarchical structures** of the **data generation mechanism**

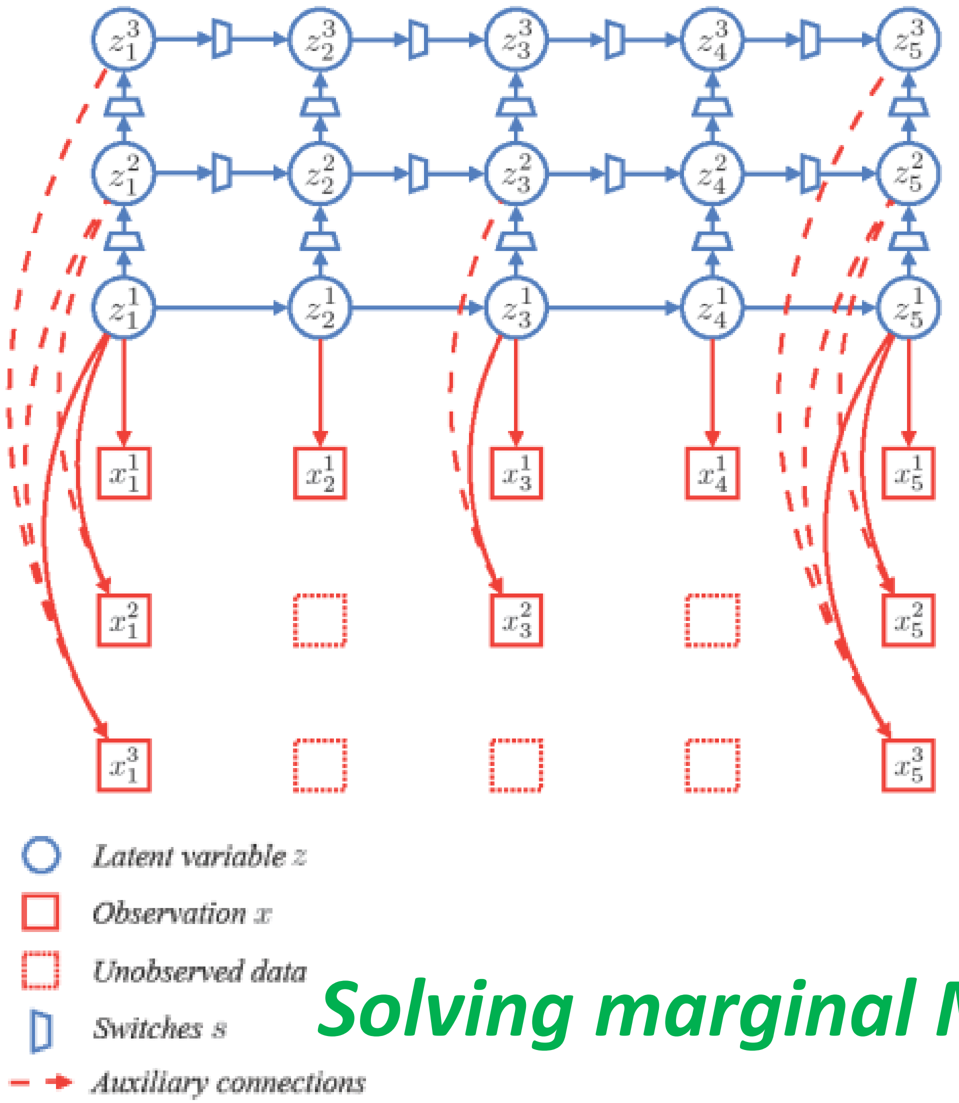
- Our proposed solution
  - **MR-HDMM**: **M**ulti-**R**ate **H**ierarchical **D**eep **M**arkov **M**odel

Overview

- Problem definitions
  - Input -- MR-MTS of L different sampling rates and T time steps ($\boldsymbol{x}_{1:T}^{1:L}$)
  - Case 1 -- Forecasting problem
    - Output -- Given $x_{1:T}^{1:L}$, predict $x_{T:T'}^{1:L}$
  - Case 2 -- Interpolation problem
    - Output -- Fill-in missing values of lower sampling rates in $x_{1:T}^{1:L}$

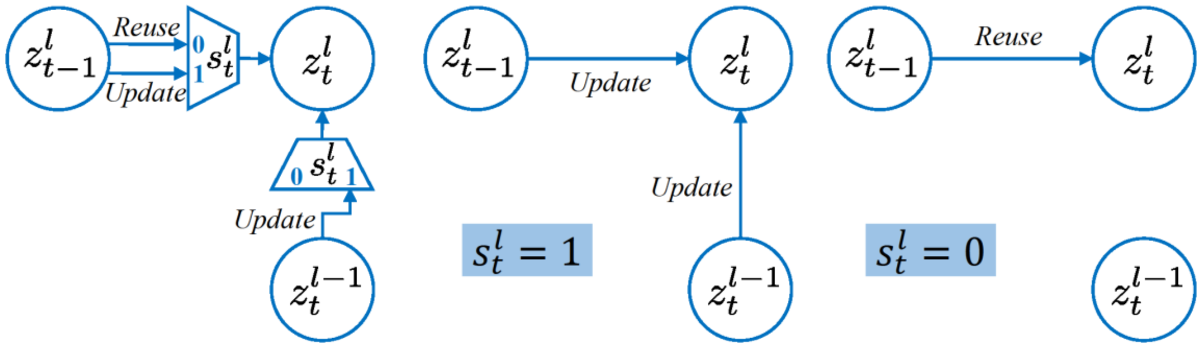
- **MR-HDMM**: Multi-Rate Hierarchical Deep Markov Model
  - Component -- a **generation model** and an **inference model**
  - Motivation -- capturing hierarchical structures in underlying data generation process
    - **Learnable switches**
    - **Auxiliary connections**

Generation Model



Solving marginal MLE?

- **Transition**
  - Learning latent states z
  - To capture **hierarchical structure**
    - Learnable switches
      - Update-and-reuse

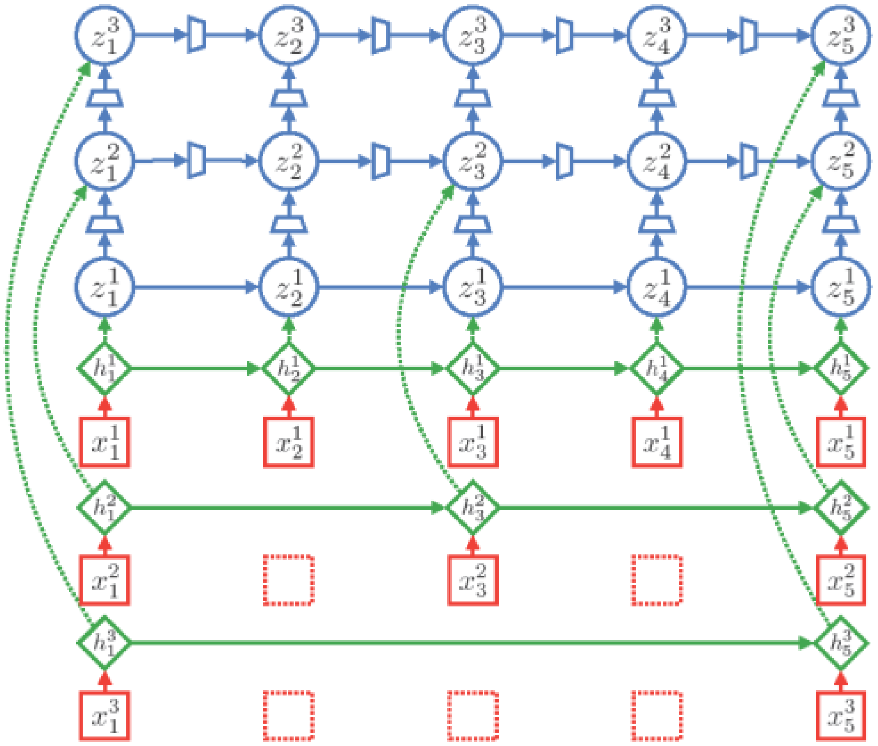


- **Emission**
  - Generating MR-MTS x
  - To capture **multi-scale dependencies**
    - Auxiliary connections

Inference and Learning

- Keep similar structure as the generative model
  - Keeping the Markov properties of z
  - Inheriting the same switches s
  - Capturing MR-MTS observation by multiple RNNs
- Maximize the variational evidence lower bound (ELBO)
  - Conditional likelihood $\sum_{t=1}^{T}\sum_{l=1}^{L} \mathbb{E}_{Q^*(\boldsymbol{z}_t^{1:l})} \, log\, p_{\theta_x}\left(\boldsymbol{x}_t^l \,|\boldsymbol{z}_t^{1:l}\right)$
  - KL at each time step and for each layer

$$\sum_{t=1}^{T} \mathbb{E}_{Q^*(\boldsymbol{z}_{t-1}^1)} D_{KL}\left(q_\phi(\boldsymbol{z}_t^1|\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_{t-1}^1)||p_\theta(\boldsymbol{z}_t^1|\boldsymbol{z}_{t-1}^1)\right) + \sum_{t=1}^{T}\sum_{l=2}^{L} \mathbb{E}_{Q^*(\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1})} D_{KL}\left(q_\phi(\boldsymbol{z}_t^l|\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1})||p_\theta(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1})\right)$$



Jointly learning all parameters

by stochastic backpropagation and ancestral sampling

Experimental settings

- Datasets

Domain	Dataset	# of Samples	Sampling Rates	# of Variables	Time Series Length
Healthcare	**MIMIC-III**	10709 (admissions)	1 / 4 / 12 Hours	7 / 12 / 44	72 Hours
Climate	**USHCN**	100 (years)	1 / 5 / 10 Days	70 / 69 / 69	365 Days

  - MIMIC-III: 5 runs × 5-fold CV (*train/valid/test split*)
  - USHCN: 5 runs of train/valid/test split with 1-month stride

- Forecasting baselines
  - **Single-rate**: Kalman Filter, VAR, Deep Markov Model, HM-RNN, LSTM, and PLSTM
  - **Multi-rate**: Multiple KF, Multi-Rate KF, and two simplified models of MR-HDMM
- Interpolation baselines
  - **Imputation**: Mean, CubicSpline, MICE, MissForest, SoftImpute
  - **Deep learning**: Deep Markov Model and the two simplified models of MR-HDMM

Quantitative results

- Forecasting

Method \ Dataset		MIMIC-III All	MIMIC-III HSR	MIMIC-III MSR	MIMIC-III LSR	USHCN All	USHCN HSR	USHCN MSR	USHCN LSR
Single-Rate Baselines	**Kalman Filter** (KF)	1.91×10^{18}	3.34×10^{18}	8.38×10^{9}	1.22×10^{6}	1.236	1.254	1.190	1.148
	Vector Autoregression (VAR)	1.233	1.735	0.779	0.802	2.415	2.579	1.921	1.748
	Deep Markov Model (DMM)	1.530	1.875	1.064	1.070	0.795	0.608	0.903	0.877
	HM-RNN	1.388	1.846	0.904	**0.713**	0.692	0.594	1.151	**0.775**
	LSTM	1.512	1.876	1.006	1.036	0.849	0.688	0.934	0.928
	PLSTM	1.244	1.392	1.030	1.056	0.813	0.710	0.870	0.915
Multi-Rate Baselines	**Multiple KF**	2.05×10^{18}	3.58×10^{18}	3.63×10^{4}	9.54×10^{2}	1.212	1.082	1.727	1.518
	Multi-Rate KF	1.691	2.289	0.944	0.860	0.628	0.542	0.986	0.799
	Multi-Rate DMM (MR-DMM)	1.061	1.192	0.723	1.065	0.667	0.611	0.847	0.875
	Hierarchical DMM (HDMM)	1.047	1.168	0.702	1.076	0.626	0.568	0.815	0.836
MR-HDMM		**0.996**	**1.148**	**0.678**	0.911	**0.591**	**0.541**	**0.742**	0.795

- Interpolation

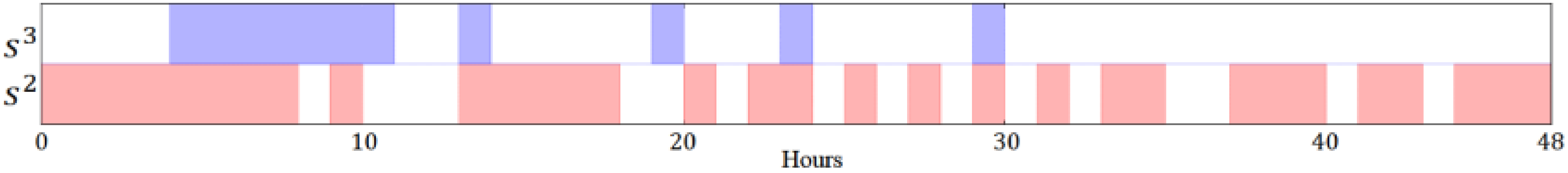
Method \ Dataset		MIMIC-III In-Sample	MIMIC-III Out-Sample	USHCN In-Sample
Imputation Baselines	**Simple-Mean**	3.812	3.123	0.987
	CubicSpline	3.713	3.212×10^{4}	0.947
	MICE	3.747	7.580×10^{2}	0.670
	MissForest	3.863	3.027	0.941
	SoftImpute	3.715	3.086	0.759
Deep Learning Baselines	**DMM**	3.714	3.027	0.782
	MR-DMM	3.710	3.021	0.696
	HDMM	3.790	3.100	0.750
MR-HDMM		**3.582**	**2.921**	**0.626**

HSR/MSR/LSR:
High/Mid/Low sampling rate

In/Out-Sample:
Interpolating training/testing dataset

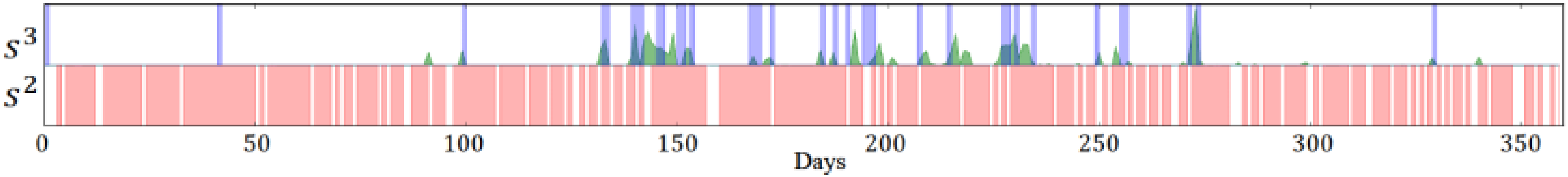
Visualizations of the learned latent hierarchical structures

- First 48 hours of an admission from MIMIC-III dataset



  - Blue: update of higher-layer states (s^3)
  - Red: update of lower-layer states (s^2)
  - Higher layer ⇒ fewer updates ⇒ longer-term dependencies

- A 1-year climate observation from USHCN dataset



  - Green: precipitation records
  - Changes in precipitations ⇒ significant differences ⇒ captured by the higher layer

Outline

- Traffic estimation and forecasting
  - Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018
- Demand forecasting
  - Li et al, Spatiotemporal Multi-Graph Convolution for Ride-hailing Demand Forecasting, AAAI 2019
- Multi-rate multi-resolution forecasting/interpolation
  - Che et al, Hierarchical Deep Generative Models for Multi-Rate Multivariate Time Series, ICML 2018

Open Dataset

KDD Cup 2017

Highway Tollgates Traffic Flow Prediction

Uber Movement

Public Data

Federal Highway Administration

Next Generation Simulation (NGSIM) Program

GAIA Open Dataset

Trajectory and OD data

Acknowledgement

- PhD students and Postdoc: Yaguang Li (Google), Zhengping Che (DiDi), Rose Yu (Northeastern), Sanjay Purushotham (U of Maryland, Baltimore)
- Collaborators

HKUST

DiDi

- Funding agency

NSF