Deep Dive into Space and Time

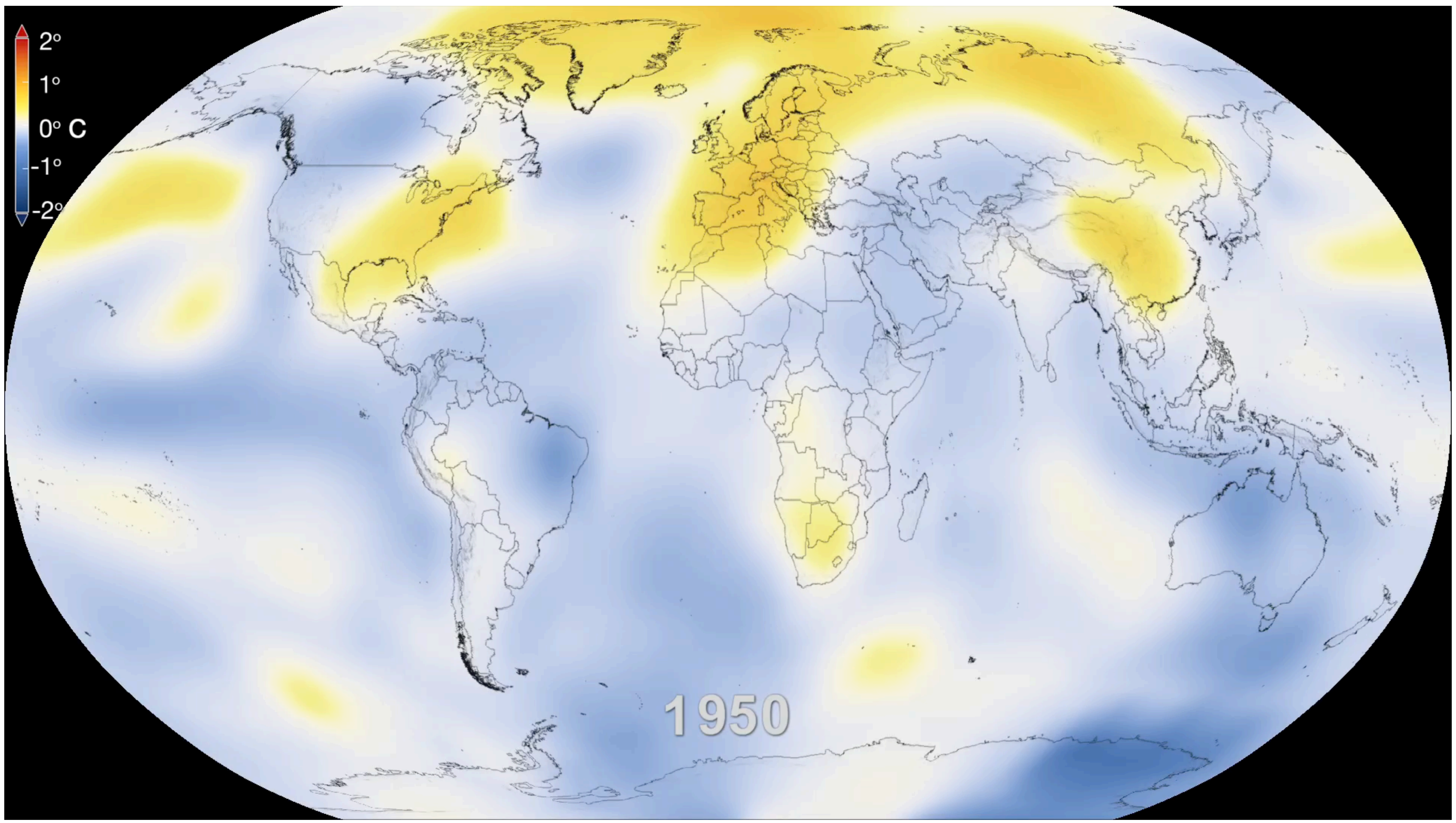
Deep Learning for Spatiotemporal Data

Rose Yu
Assistant Professor
Northeastern University

NVIDIA GTC S9806

Predicting Global Climate

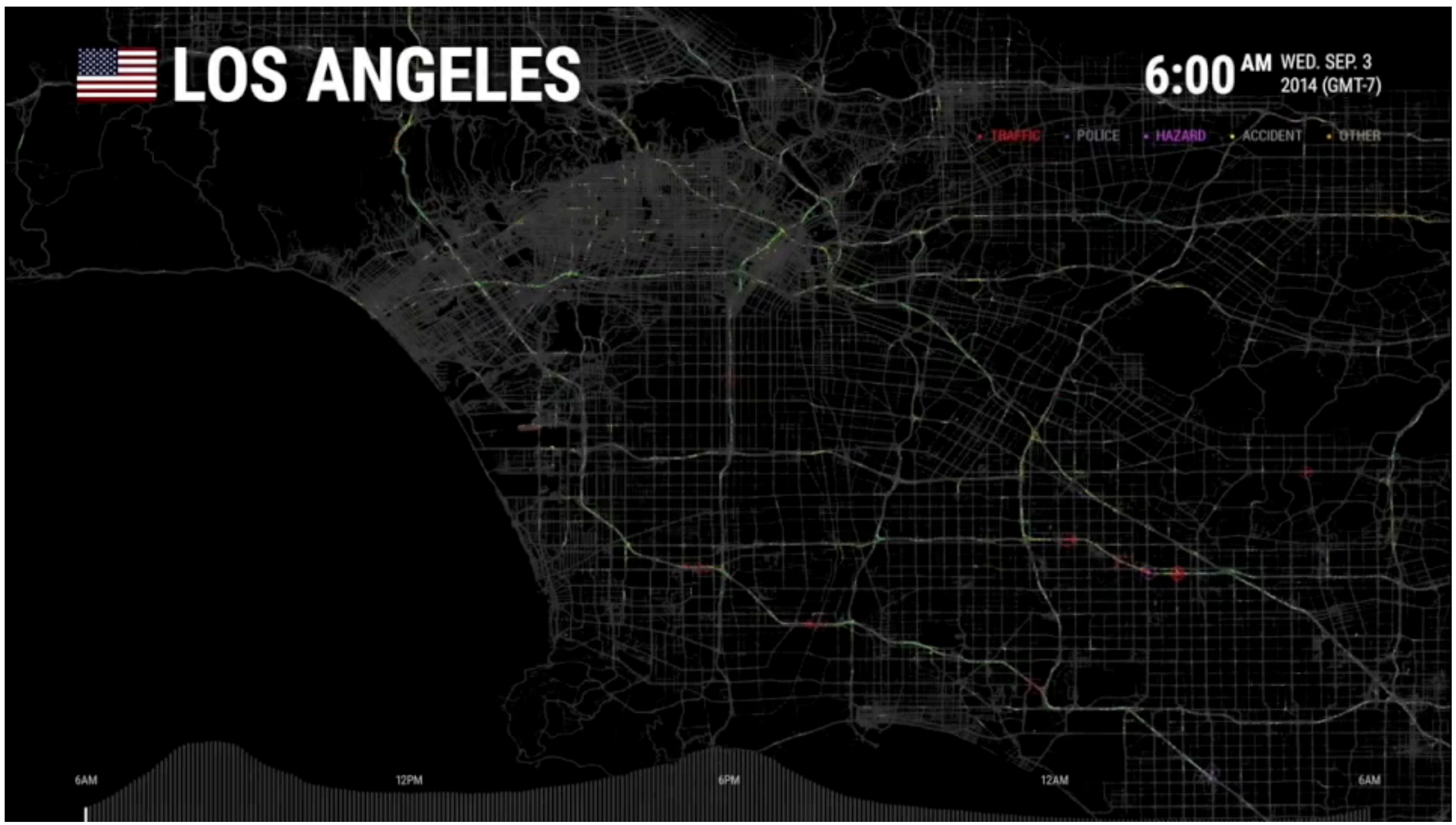
100,000 stations, 180 countries



credit: NASA

Forecasting Daily Traffic

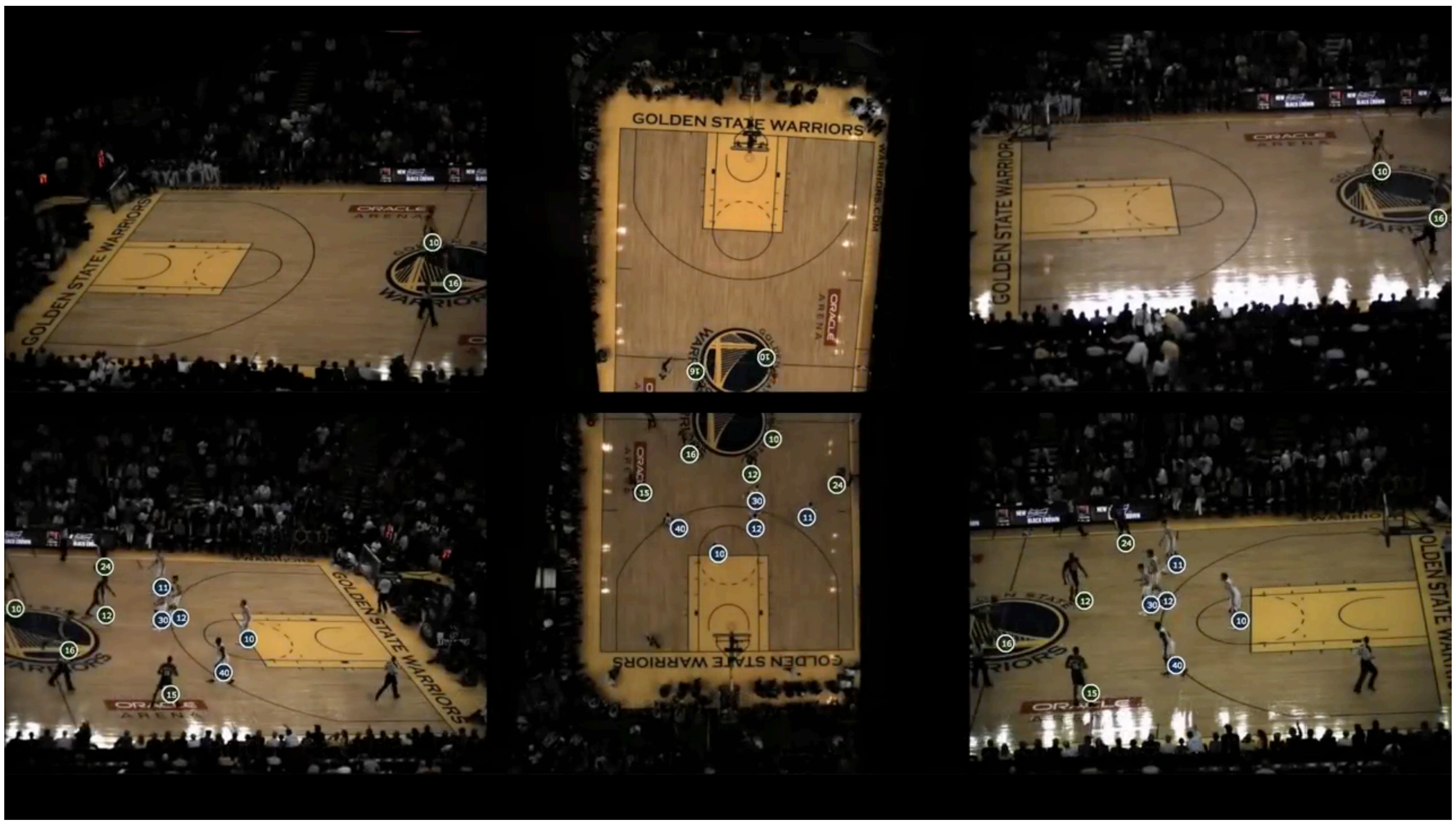
35,000 detectors, every 30 seconds



credit: Waze

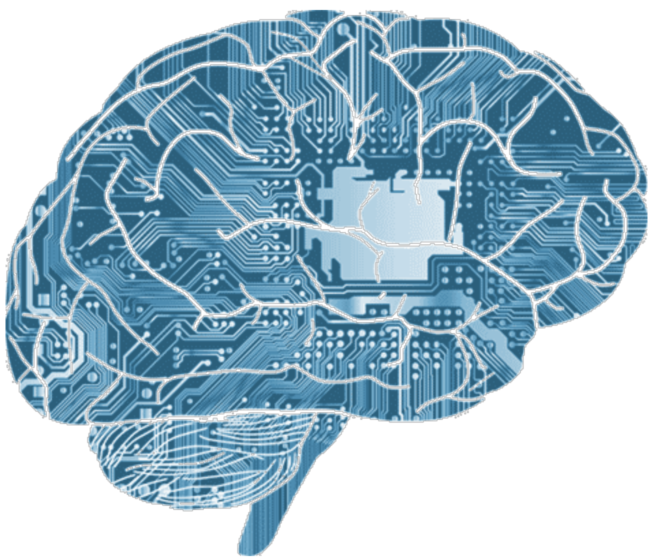
Modeling Basketball Play

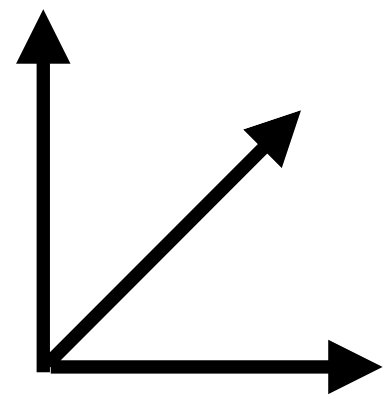
2,000 events, 1.5 million data points

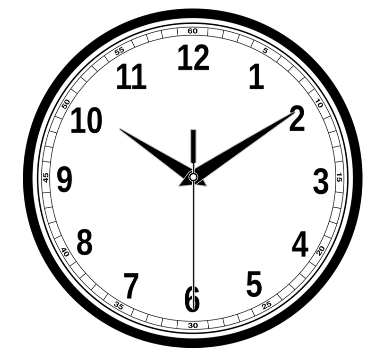


credit: STATS

Spatiotemporal Learning

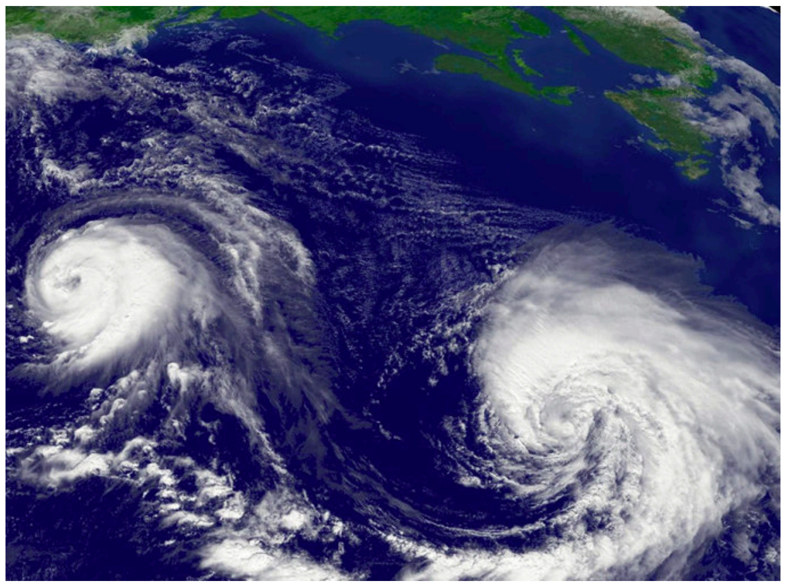






- Make sense of large amount of data collected over **space** and **time**
- Enable **AI** systems to understand and reason in space and time
- Critical to **real-time decision making** in science and engineering

Technical Challenges



nonlinear dynamics

- **sensitive** to initial conditions
- hard to simulate

non-Euclidean geometry

- sensors deployed on an **irregular grid**
- hard to represent

higher-order correlations

- players/teams/game are correlated
- hard to learn

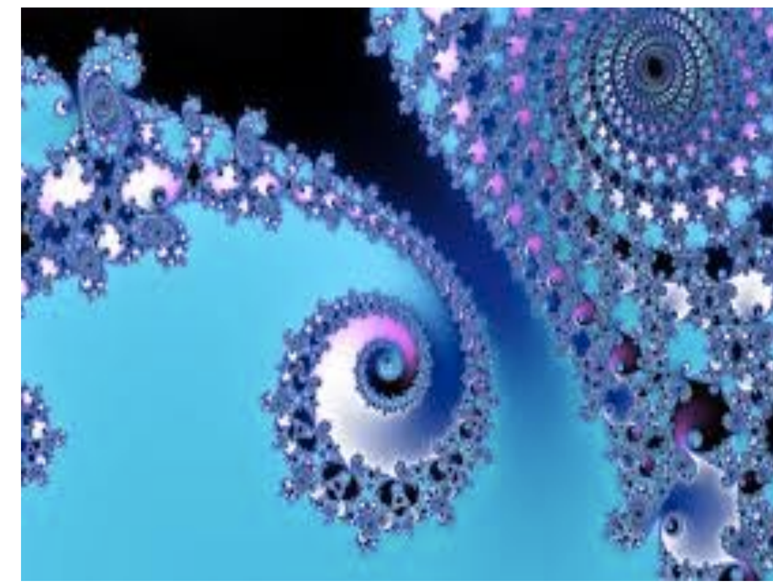
Technical Challenges

error cascading

- predictions are **sequential**
- error propagate

missing values

- sensors failure, transmission problem
- dirty data



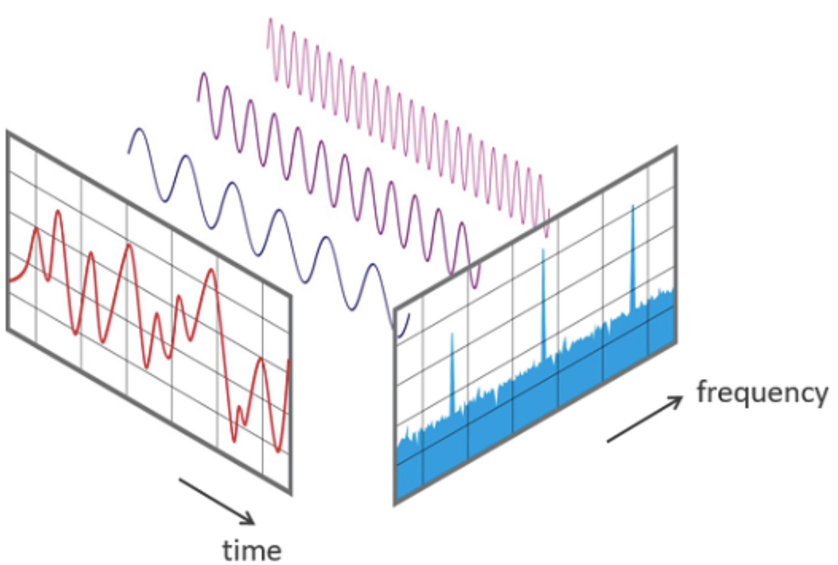
multi-resolution

- space and time at different scales
- complex hierarchy

Traditional Methods

Feature engineering

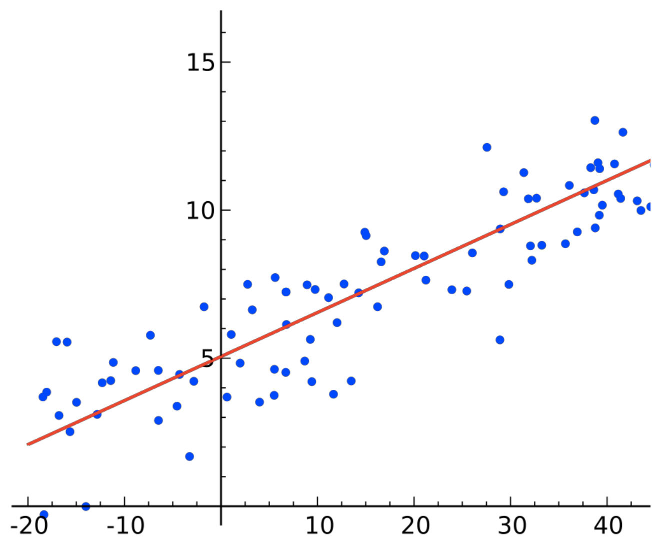
expensive, requires domain knowledge



Discrete Fourier Transformation

Simple models

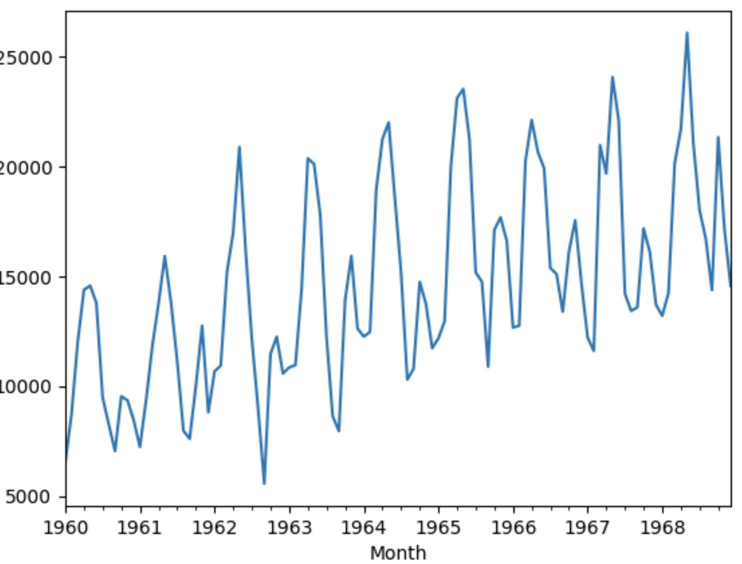
autoregressive (AR) model, ARIMA, etc



Linear Models

Small data

univariate, MB size data

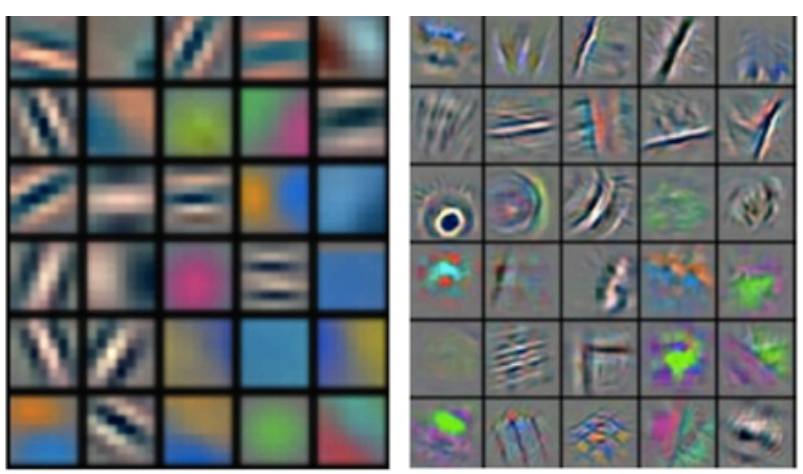


Univariate Time Series

Promise of Deep Learning

Feature learning

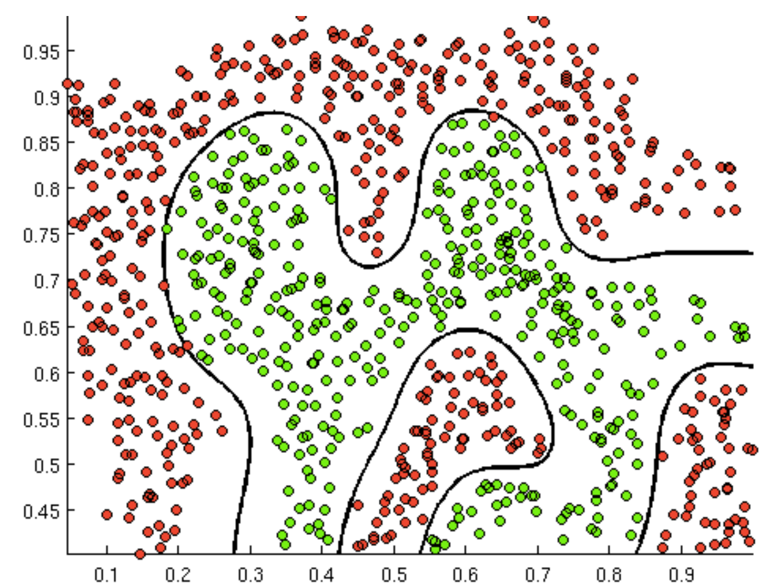
cheap, no domain knowledge



Automatic Feature Learning

Complex models

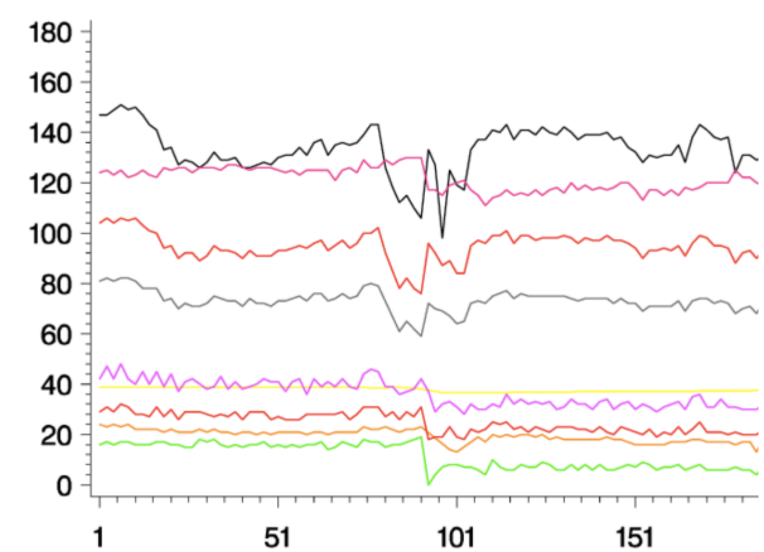
nonlinear, deep neural networks



Nonlinear Models

Big data

multivariate, TB-PB size data



Multivariate Time Series

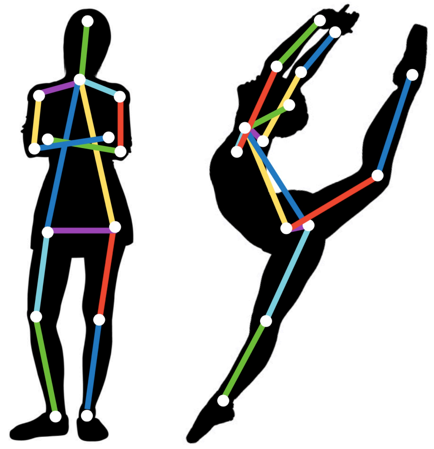
Spatiotemporal Forecasting

Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting
Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu.
International Conference on Learning Representations (ICLR), 2018

Introduction

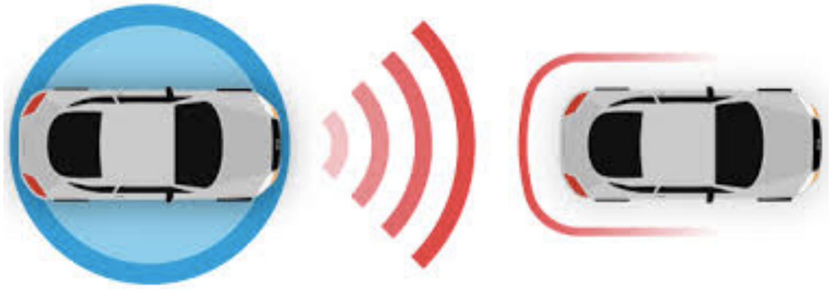
- Spatiotemporal forecasting
  - Input: history from P locations $x_0, \ldots, x_t$
  - Output: future values over H time steps $x_{t+1}, \ldots, x_{t+H}$

- Various applications



activity recognition

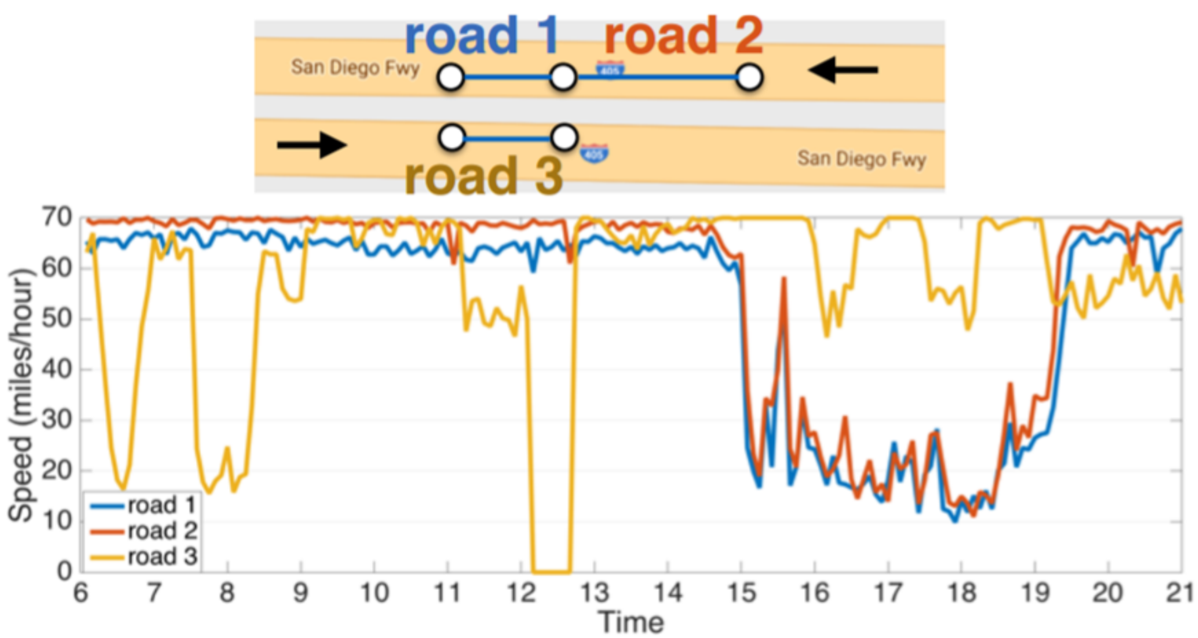
Internet of Things

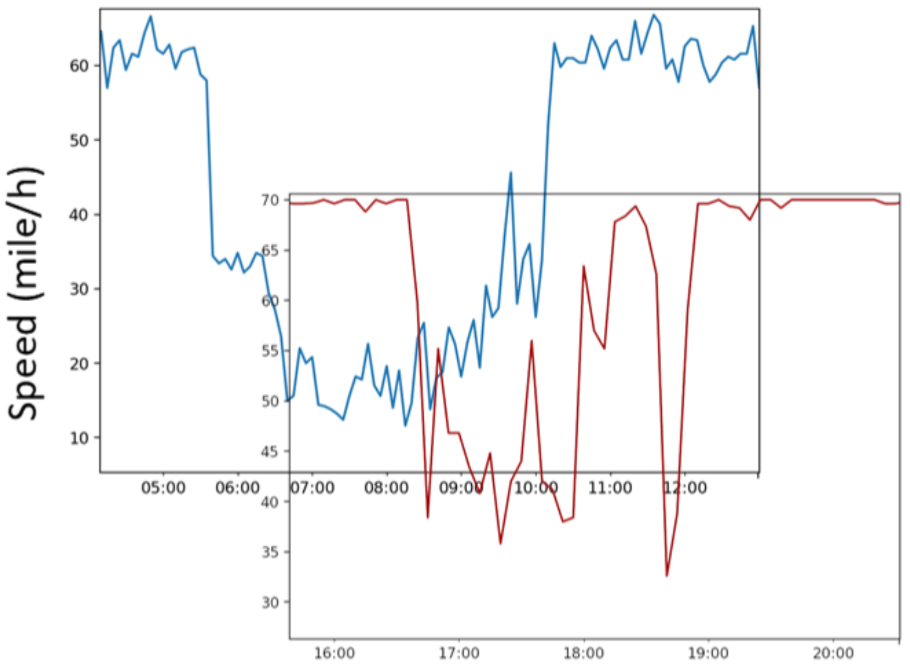


autonomous fleets

Traffic Forecasting

- Spatiotemporal forecasting problem requires modeling complex **spatial** and **temporal** dynamics





Non-Euclidean Spatial Dependency

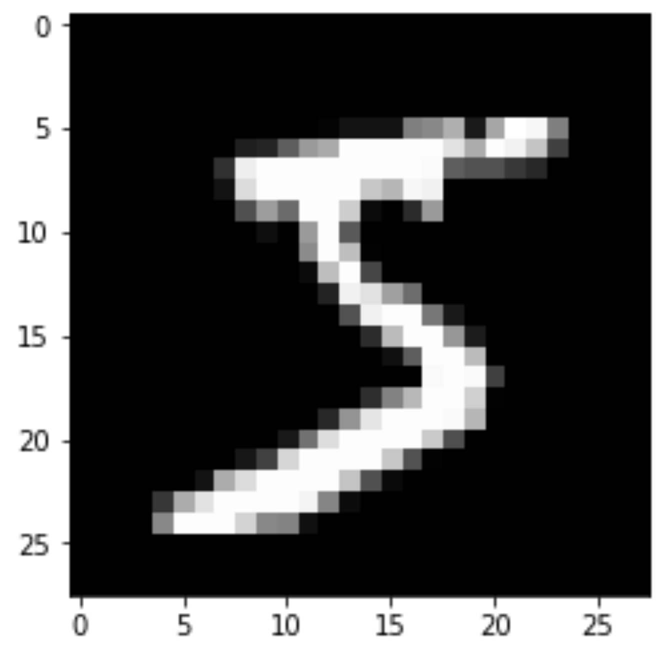
Diffusion Convolution

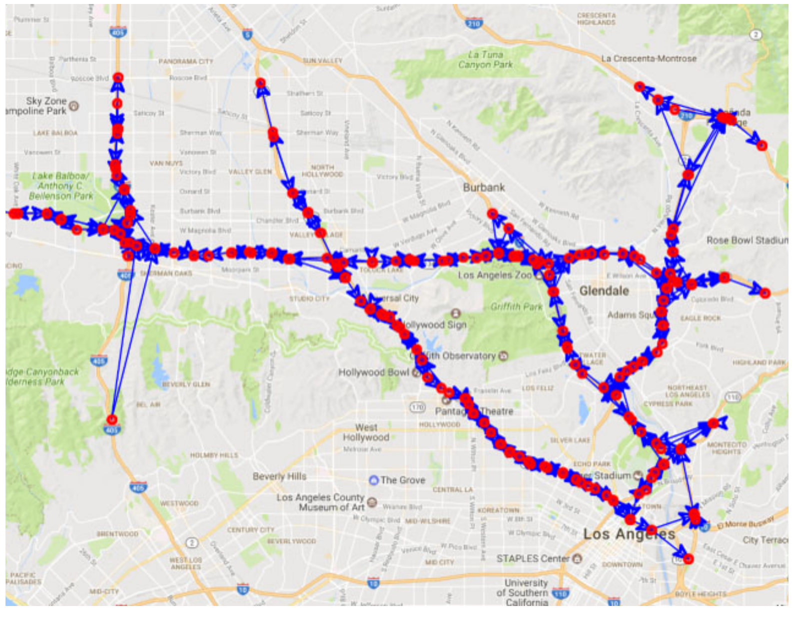
Non-stationary Temporal Dynamics

Sequence to Sequence

+

Spatial Dependence

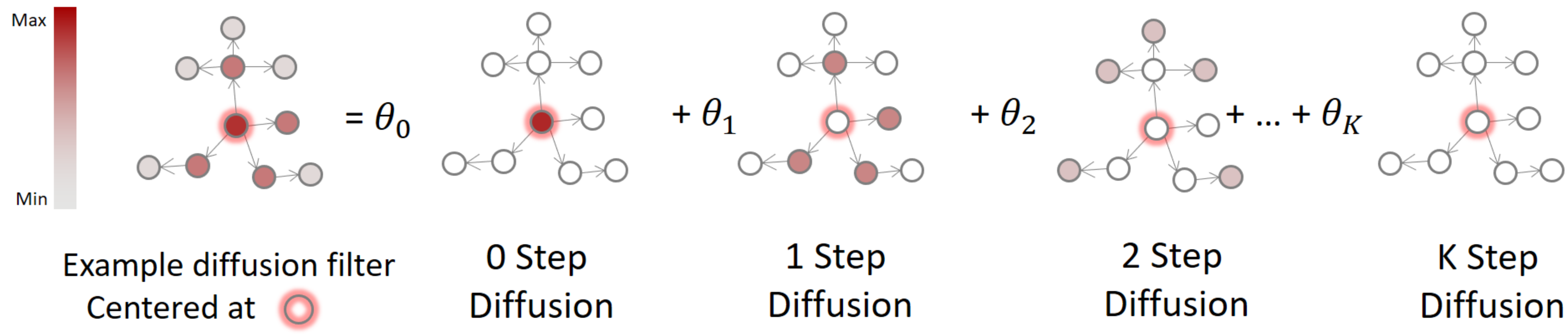




- Image data = a regular grid
- Model with convolutional neural networks

- Model the traffic flow as a **diffusion process** on an irregular grid
- Generalize convolutional operation to **directed graphs**

Diffusion Convolution

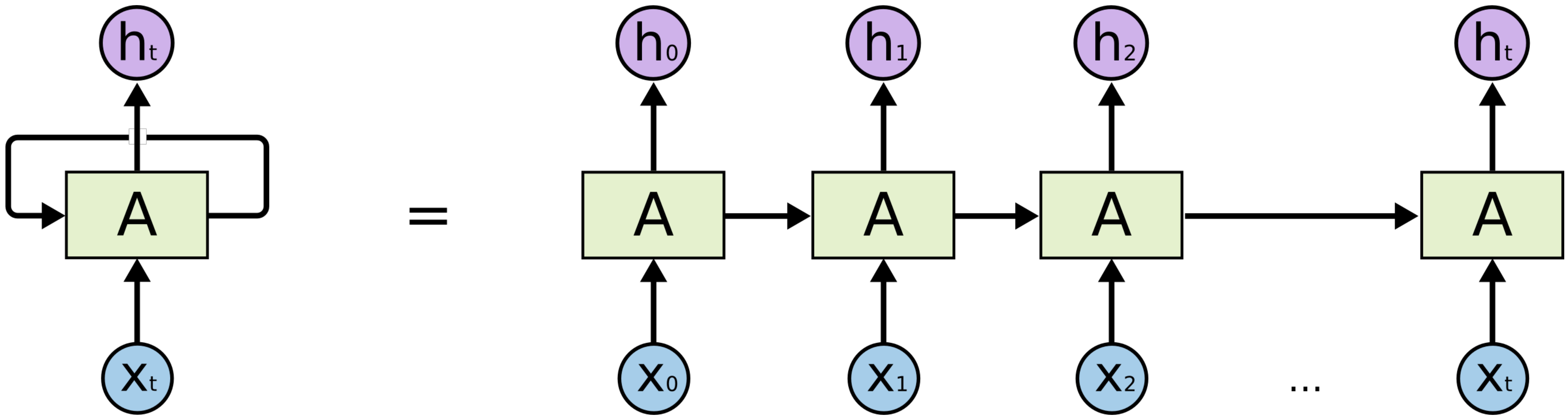


- Adjacency matrix $\mathbf{A}_{ij} = \exp\left(-\frac{\text{dist}_{net}(v_i, v_j)^2}{\sigma^2}\right)$
- In/Out degree Matrix $\mathbf{D}_{Ii} = \sum_j \mathbf{A}_{ij} \quad \mathbf{D}_{Oj} = \sum_i \mathbf{A}_{ij}$

$$f_\theta \star_g \mathbf{x}_t = \sum_{k=0}^{K} (\underbrace{\theta_{I,k}}_{\text{weights}} (\mathbf{D}_I^{-1}\mathbf{A}^\top)^k + \underbrace{\theta_{O,k}}_{\text{weights}} (\mathbf{D}_O^{-1}\mathbf{A})^k) \underbrace{\mathbf{x}_t}_{\text{inputs}}$$

DCRNN

- Input state $\mathbf{x}_t$, hidden state $\mathbf{h}_t$, output $\mathbf{y}_t$

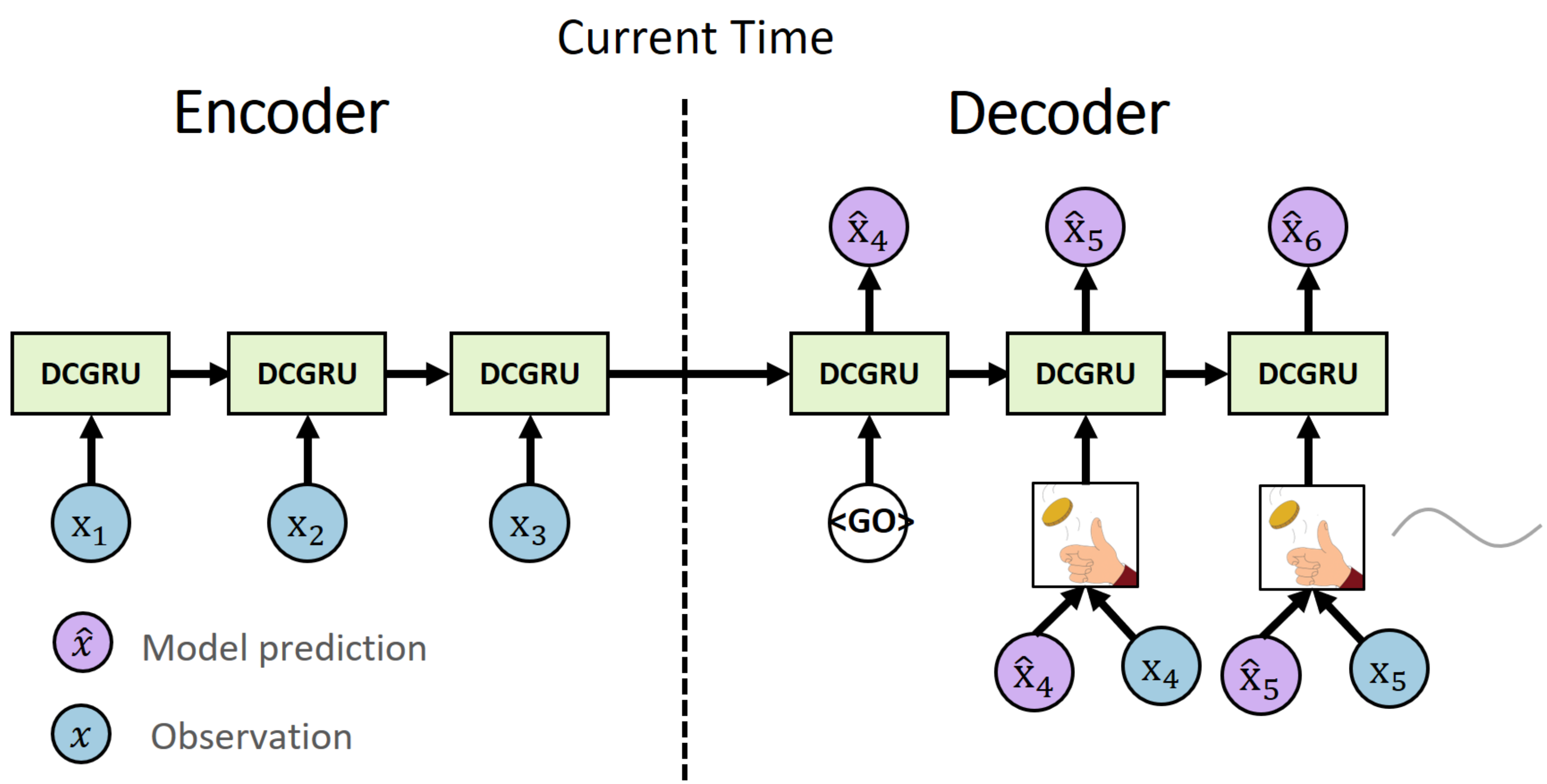


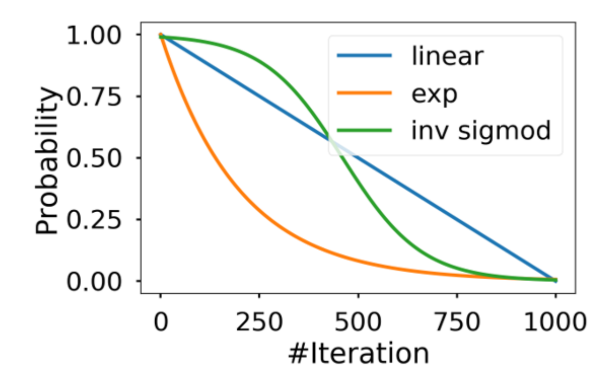
$$h_t = f_h(W^x x_t + W^h h_{t-1} + b) \qquad y_t = f_o(W h_t + b)$$

$$h_t = f_h(W^x *_g x_t + W^h *_g h_{t-1} + b)$$

Temporal Dependence

- Encoder-decoder architecture in **sequence to sequence**
- Mitigate error propagation with **scheduled sampling** †

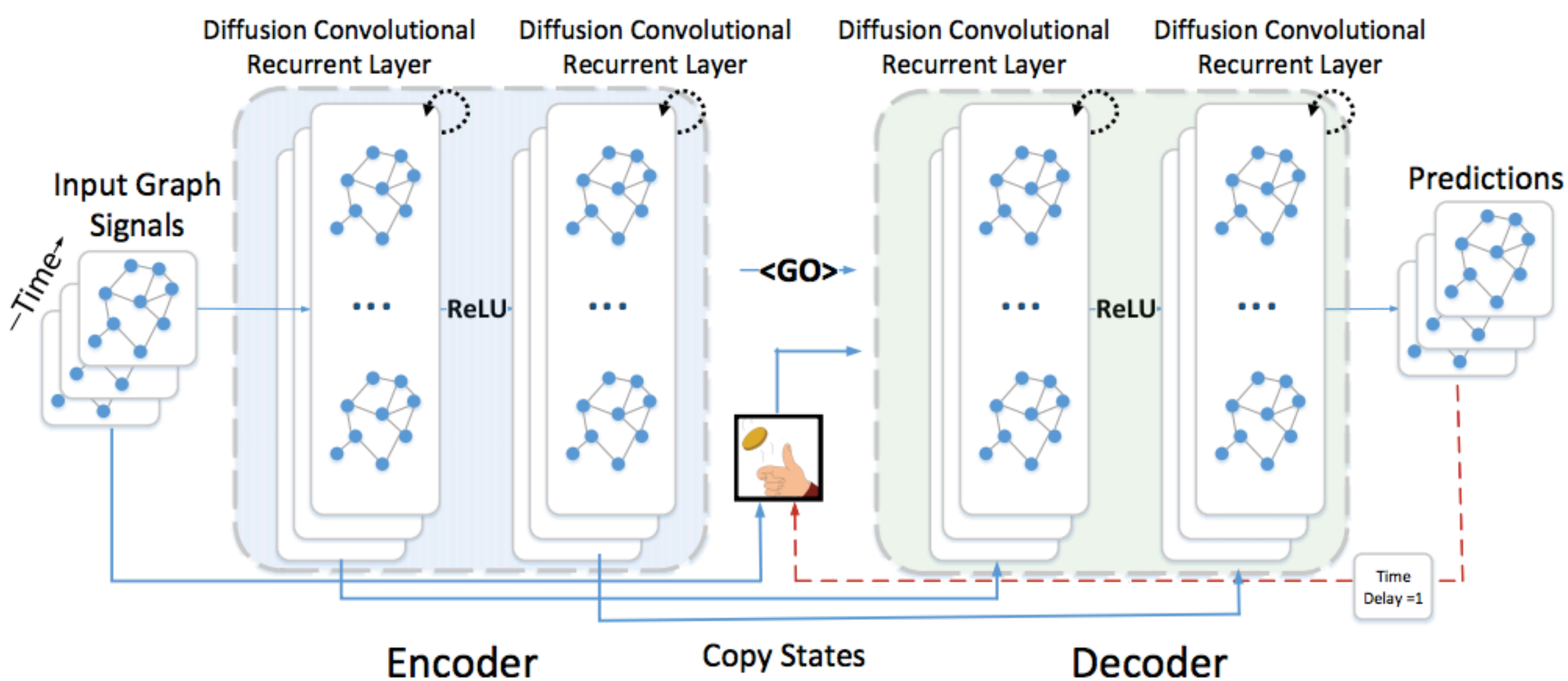




† **Scheduled sampling for sequence prediction with recurrent neural networks**
Bengio, Samy, et al.

Model Architecture

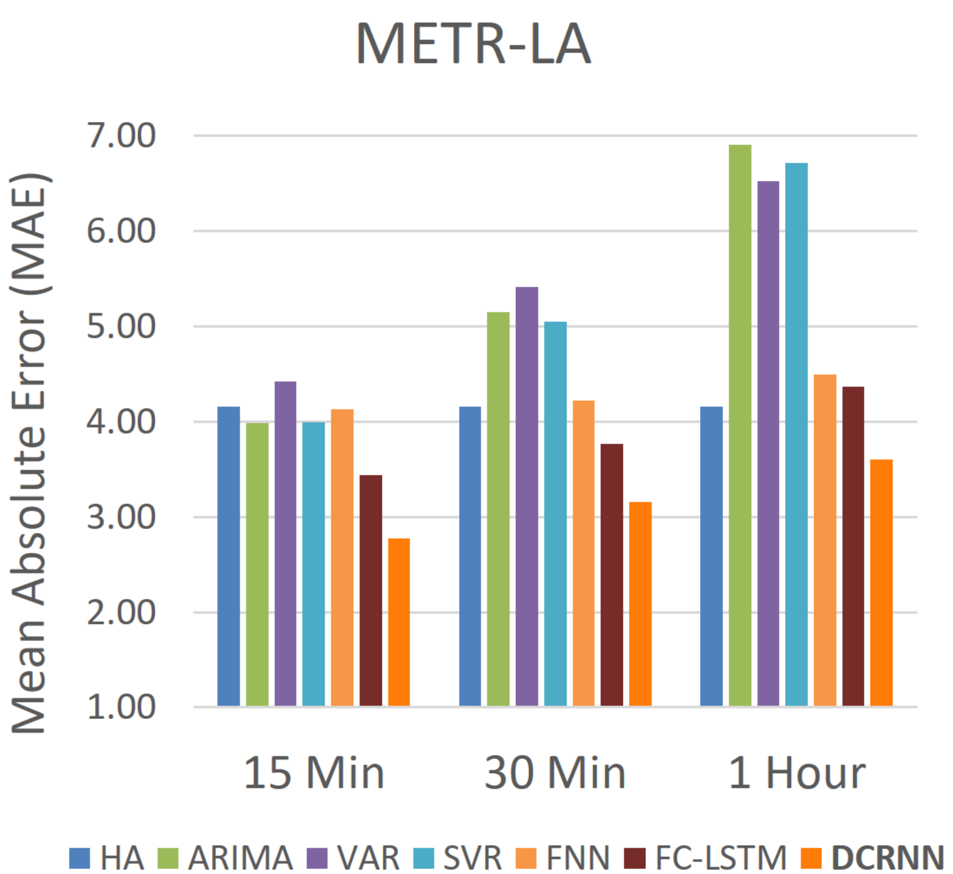
- **Input**: a sequence of history graphs
- **Output**: a sequence of future graphs
- Diffusion Convolutional Gated Recurrent Unit (**DCGRU**) in Seq2Seq framework

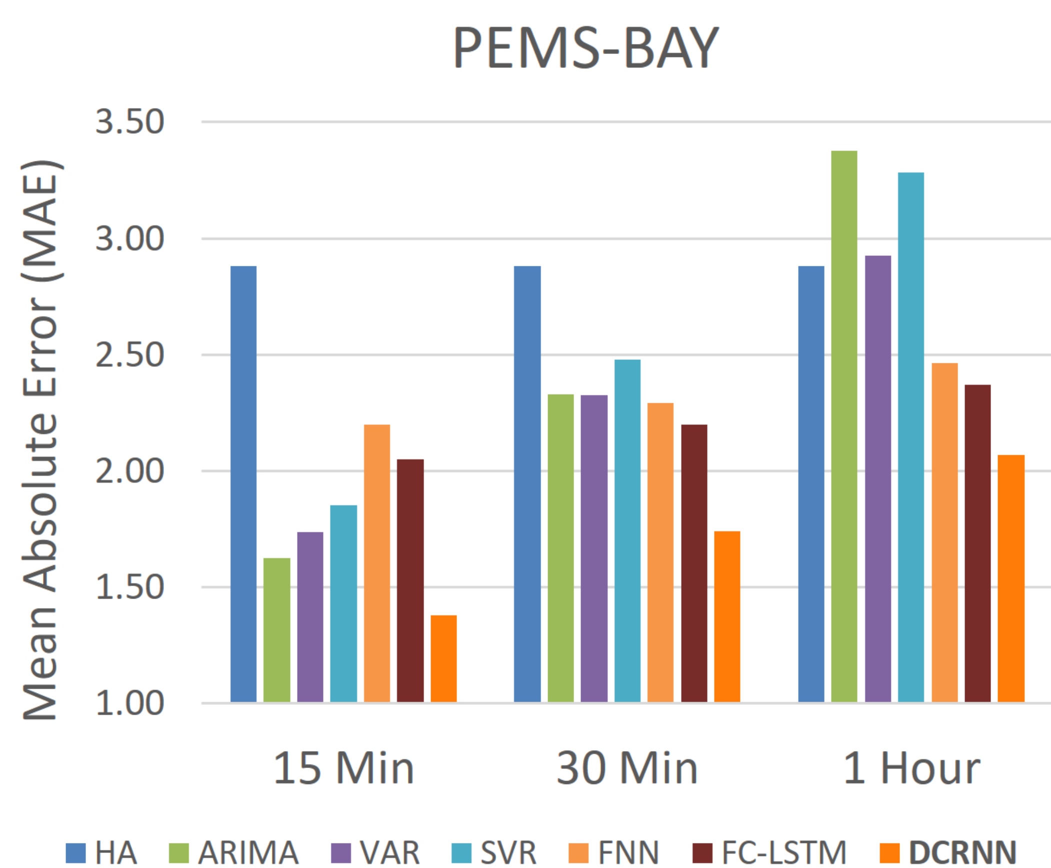


Forecasting Accuracy

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

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

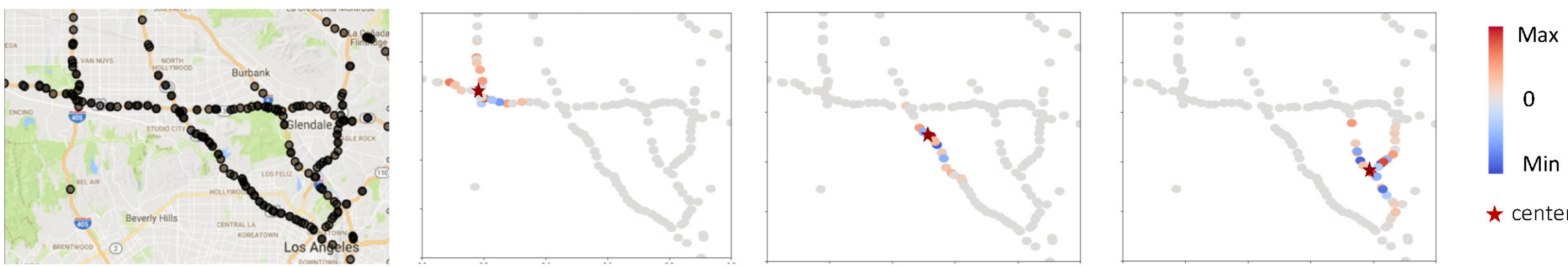


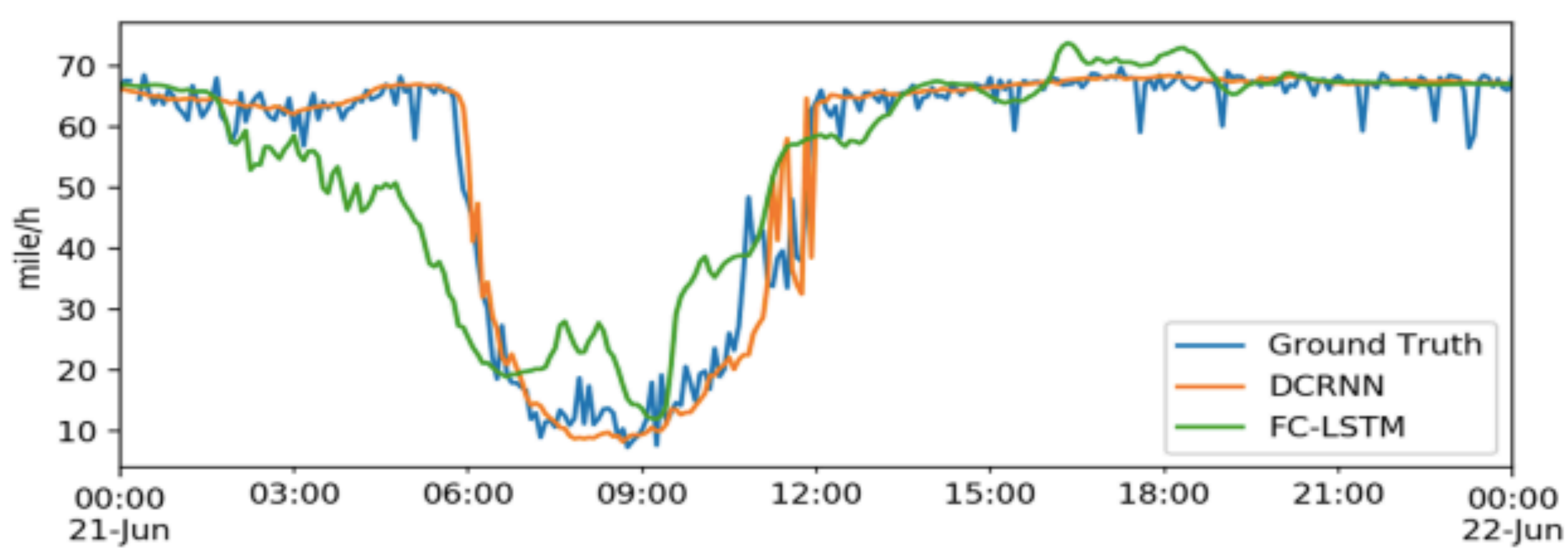


Prediction Visualizations

- Learned convolutional filter with weights localized around the center, and diffuse alongside the road network.

- More likely to accurately predict abrupt changes in the traffic speed



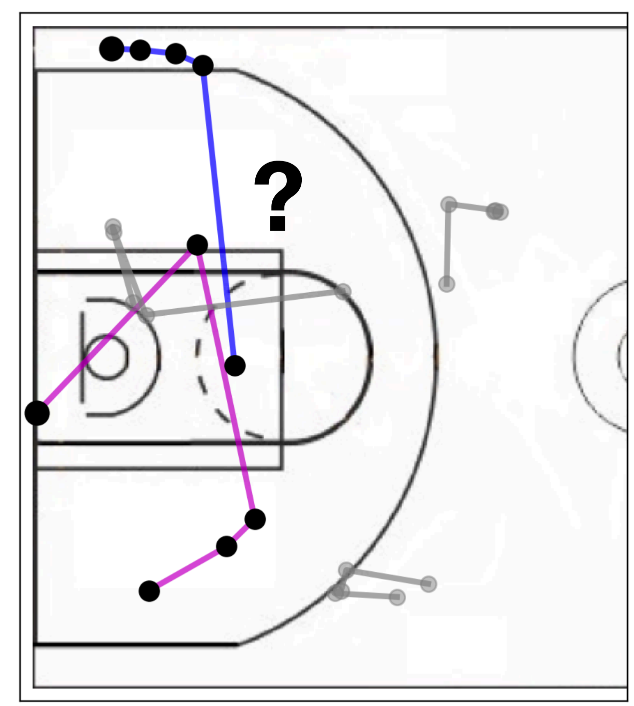


Spatiotemporal Imputation

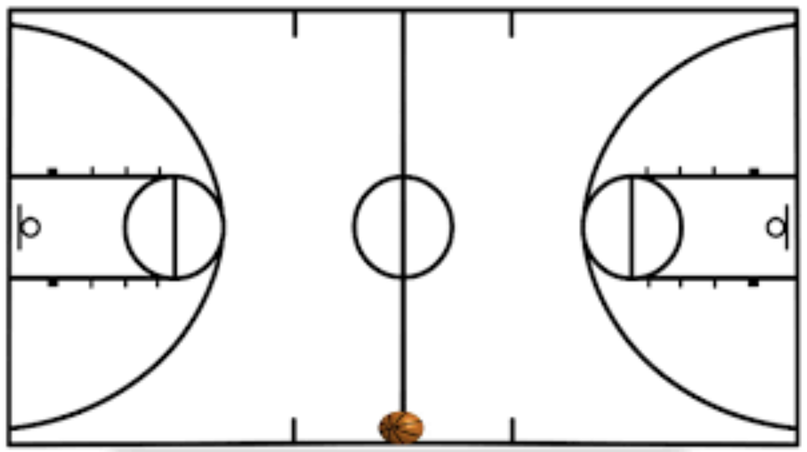
NAOMI: Non-Autoregressive Multiresolution Sequence Imputation
Yukai Liu, Stephan Zheng, Rose Yu, Yisong Yue
Arxiv Preprint *https://arxiv.org/abs/1901.10946*

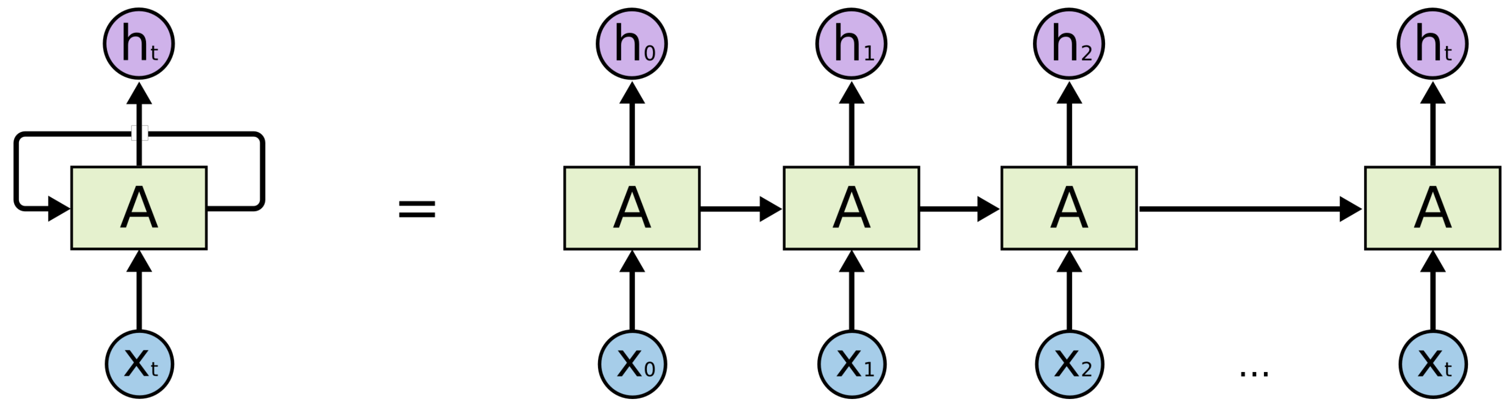
Introduction

- Real-world applications often have missing data
- Hard to impute due to infinite number of possibilities

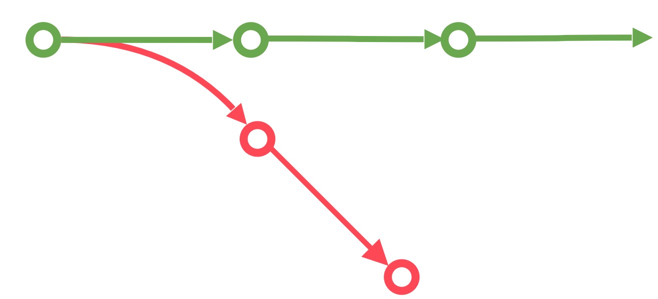


Spatiotemporal Sequences

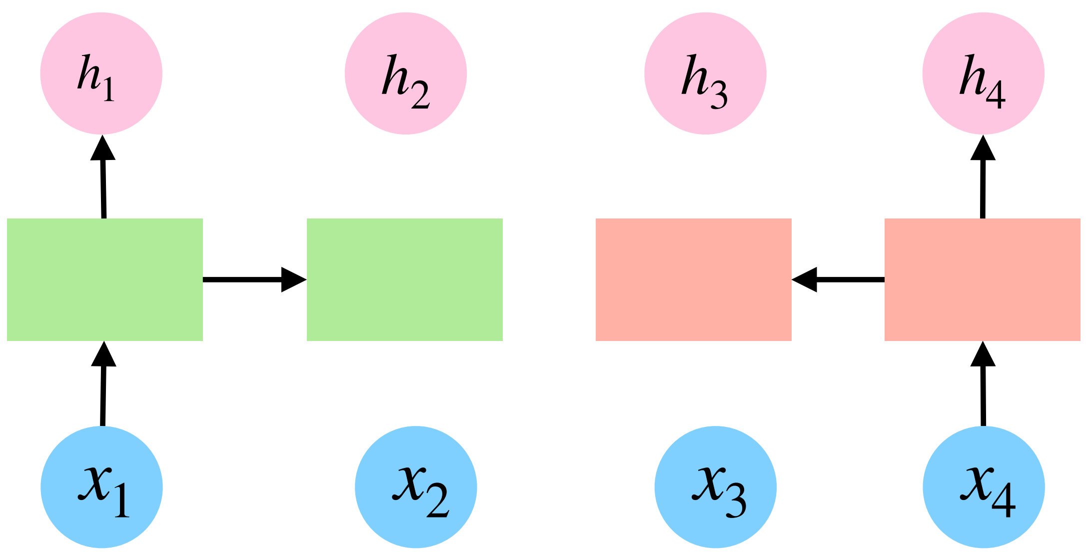




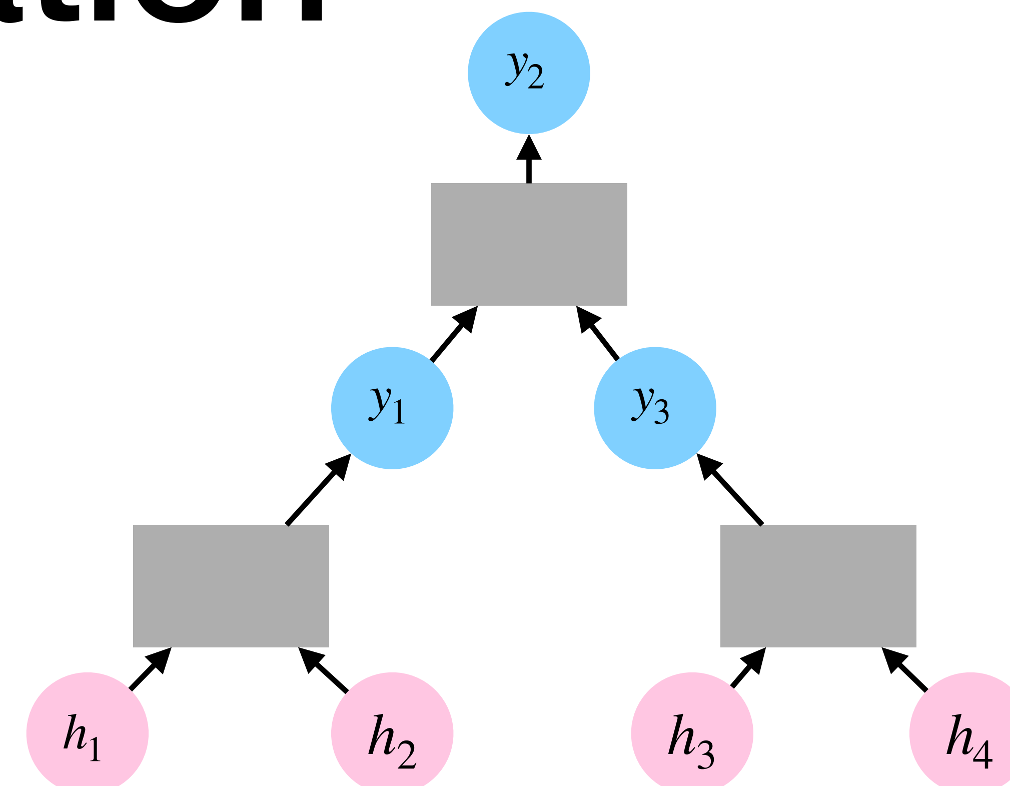
- Correlated at multiple spatial and temporal resolutions
- RNNs only operate on a single temporal resolution
- Autoregressive models are susceptible to error propagation



Non-Autoregressive Generation



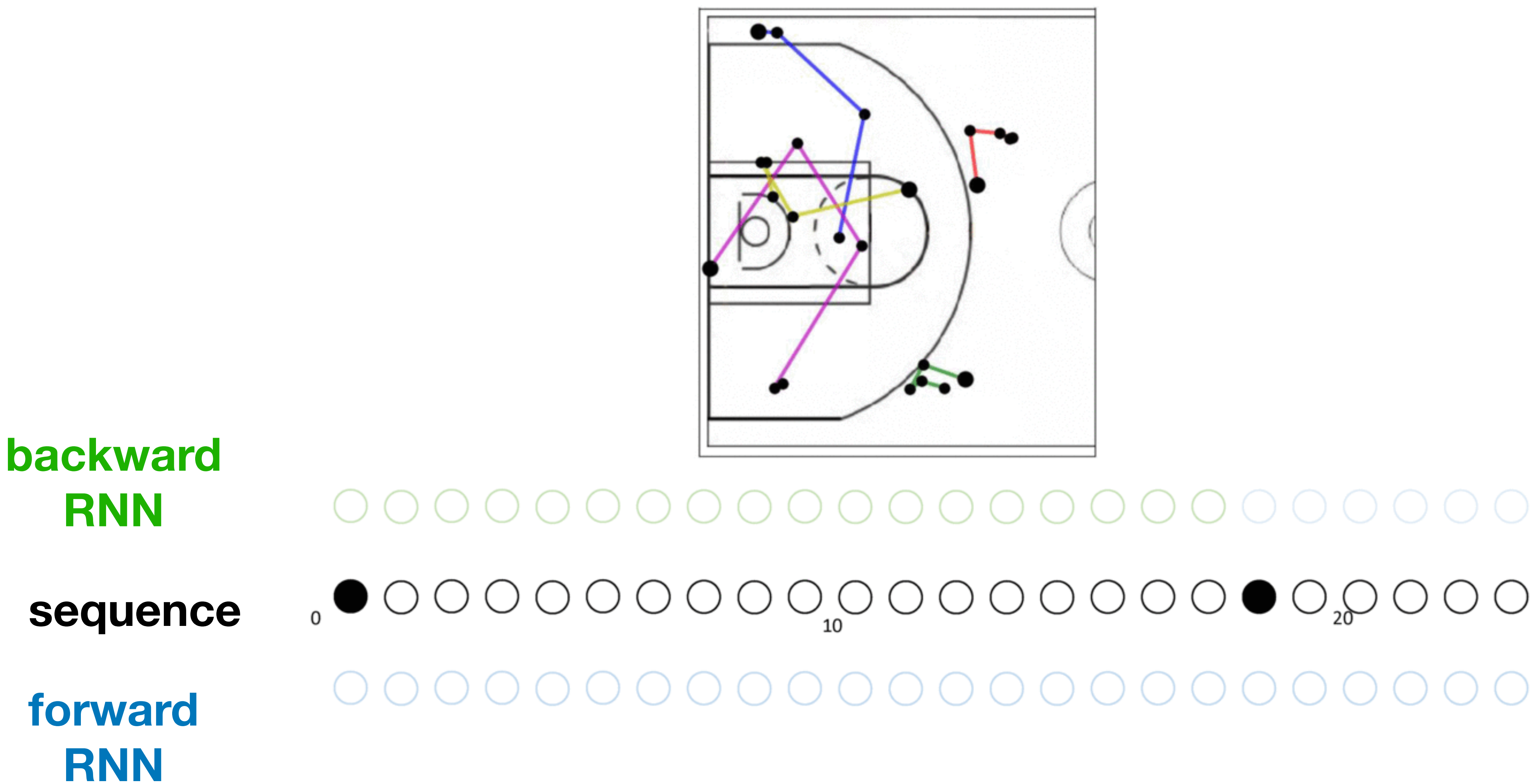
Encoding



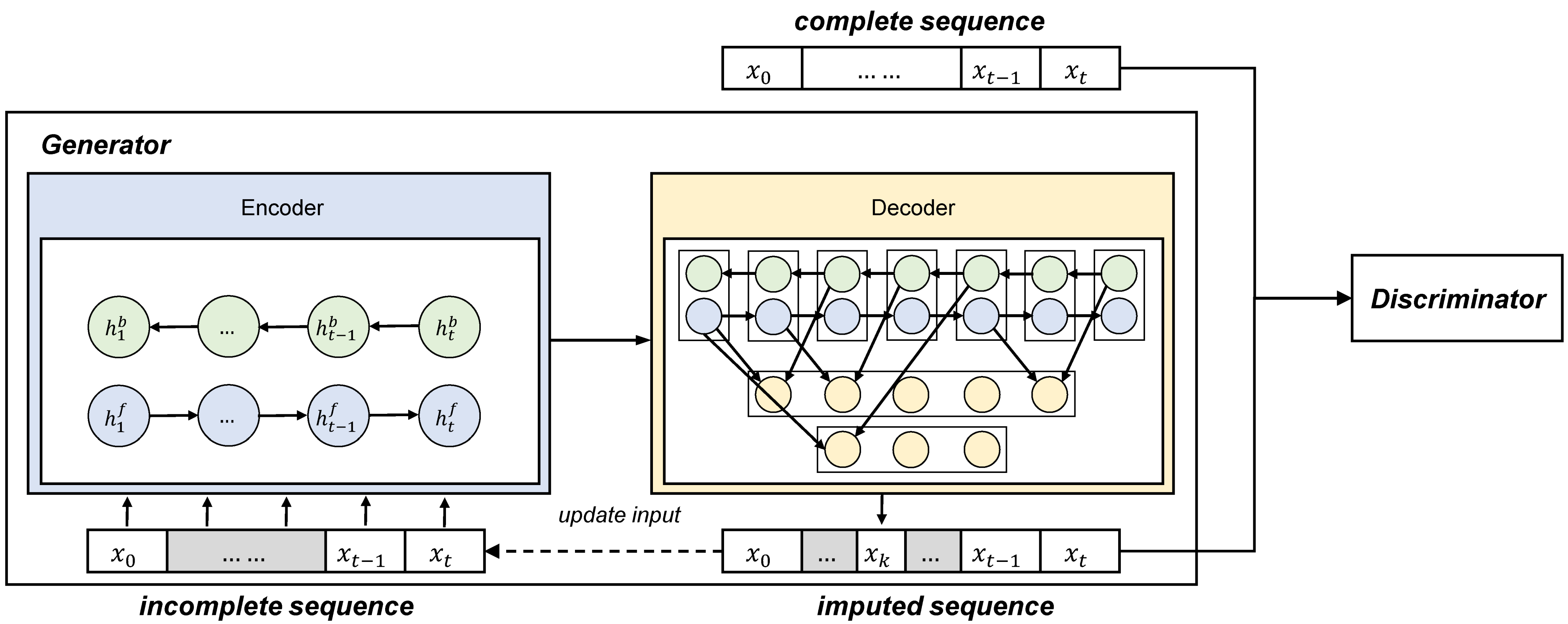
Decoding

- Encoding the sequence with a forward and backward RNN
- Decoding using a divide-and-conquer strategy at multiple resolutions

Non-Autoregressive Generator

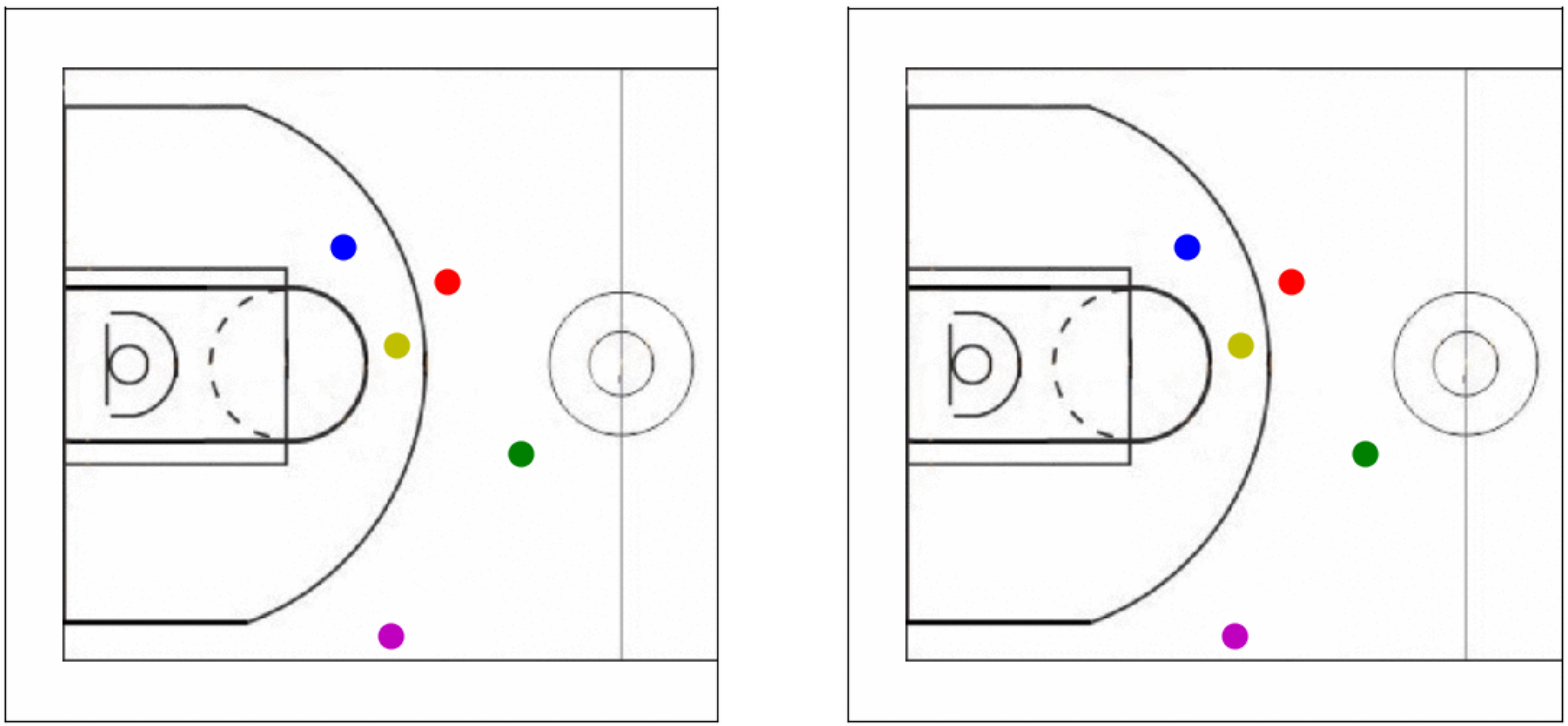


NAOMI Model Architecture

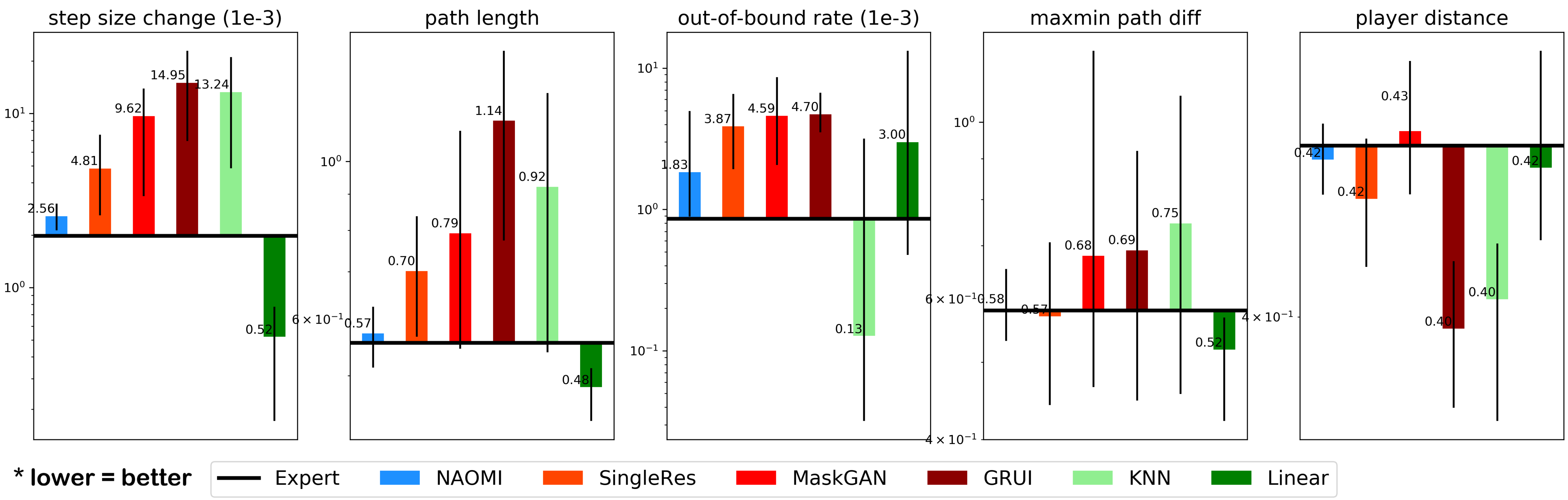


- Adversarial training: distinguish between real/fake trajectories
- Multi-resolution non-autoregressive generator
- Recursive imputation: divide and conquer strategy

Trajectory Turing Test

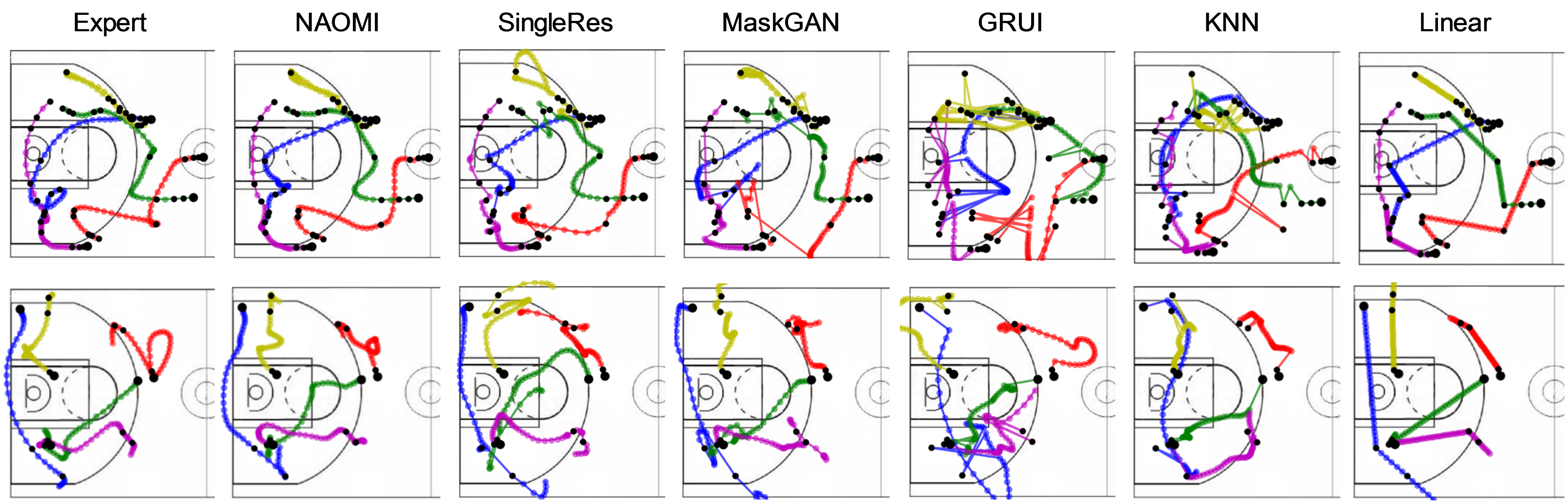


Quantitative Performance



- 5 metrics: movement consistency, length, regularity, variance, team coordination
- Statistics closer to the expert data indicate better performance

Interpolation Comparison



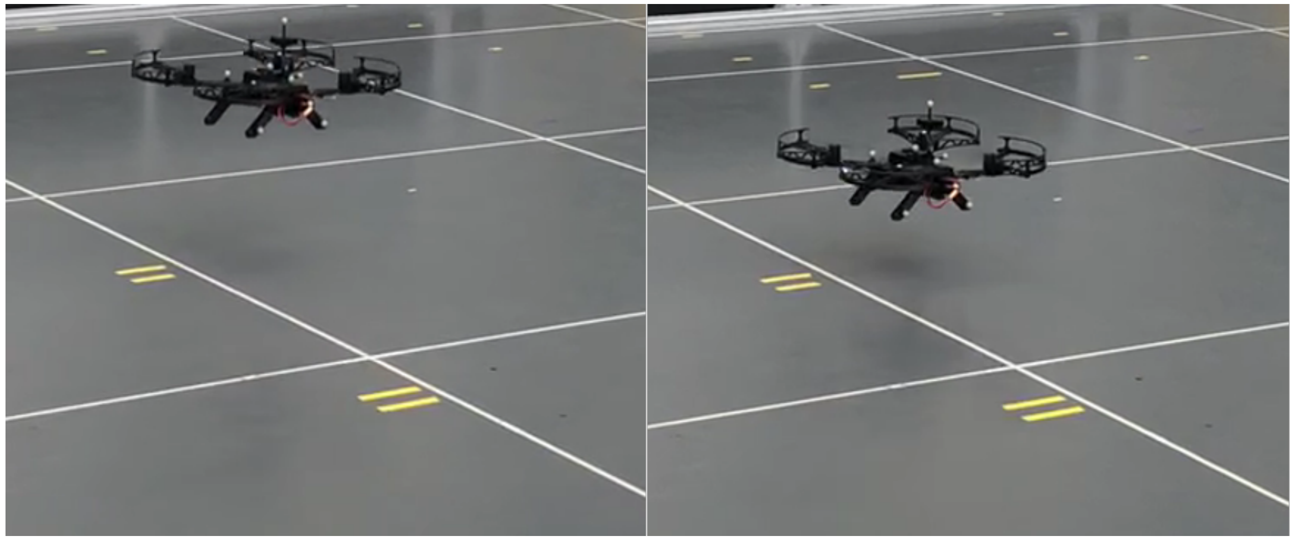
NAOMI produces trajectories mostly consistent with the expert data, with realistic player movements

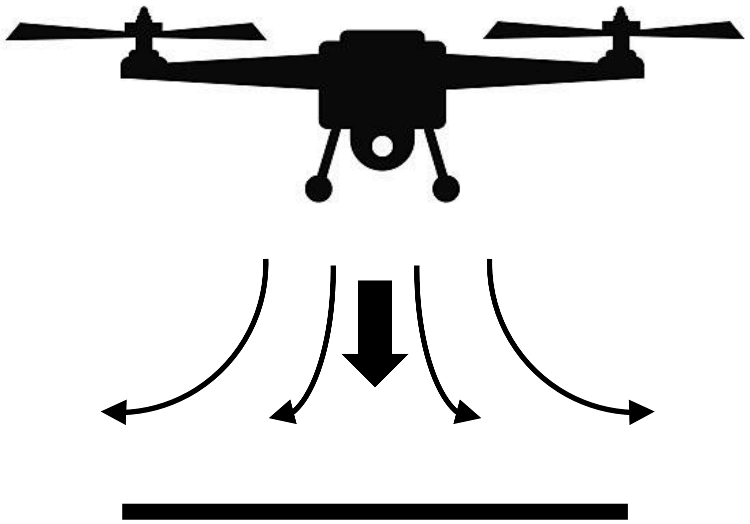
Spatiotemporal Dynamics

Neural Lander: Stable Drone Landing Control using Learned Dynamics
Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung
International Conference on Robotics and Automation (ICRA), 2019

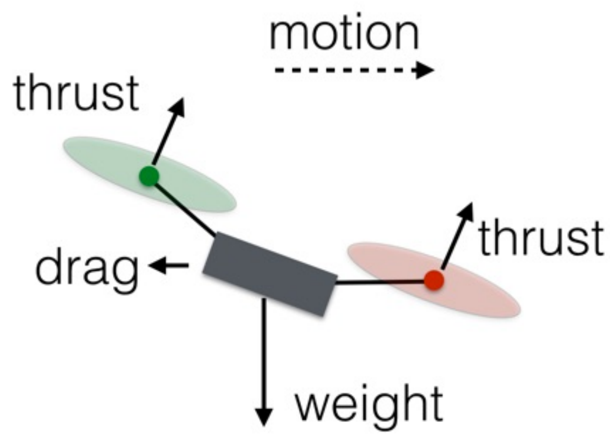
Introduction

- Ground effects: disturbance during landing/taking off
- Complex nonlinear aerodynamics
- Impossible to simulate accurately or find analytical solutions





Hybrid Learning Framework



Pressure: p Velocity: v Angular Velocity: ω

Total Thrust, Torque: $\mathbf{f}_u, \tau_u$

Unknown Disturbance Force, Torque: $\mathbf{f}_a, \tau_a$

States: $\mathbf{s}_{t-1}$

+

Controls: $\mathbf{u}_{t-1}$

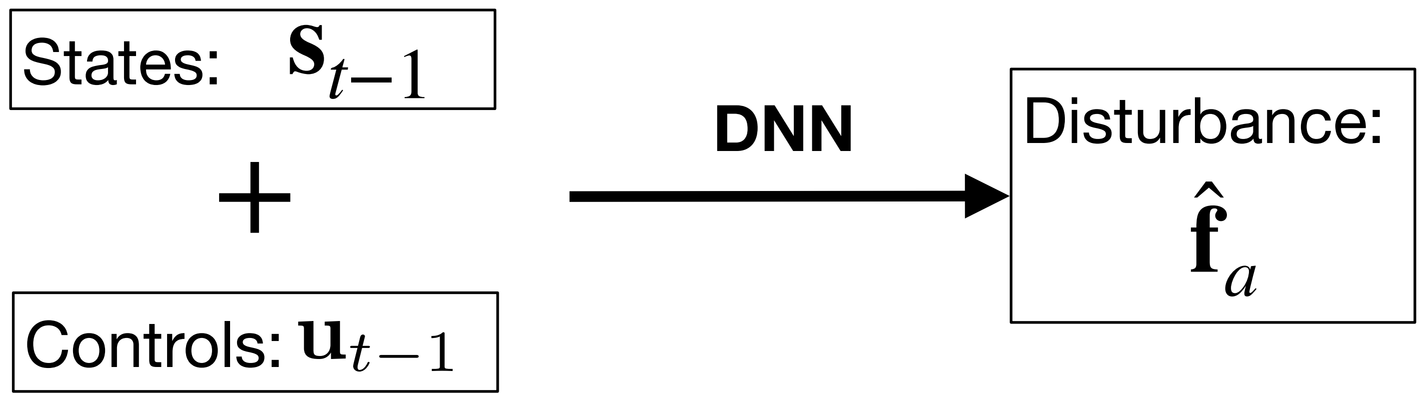
$$\mathbf{p} = \dot{\mathbf{v}}$$
$$m\dot{\mathbf{v}} = m\mathbf{g} + R\mathbf{f}_u \boxed{+\mathbf{f}_a}$$
$$\dot{R} = RS(\omega)$$
$$J\dot{\omega} = J\omega \times \omega + \tau_u + \tau_a$$

DNN

Learn this part

New States: $\mathbf{s}_t$

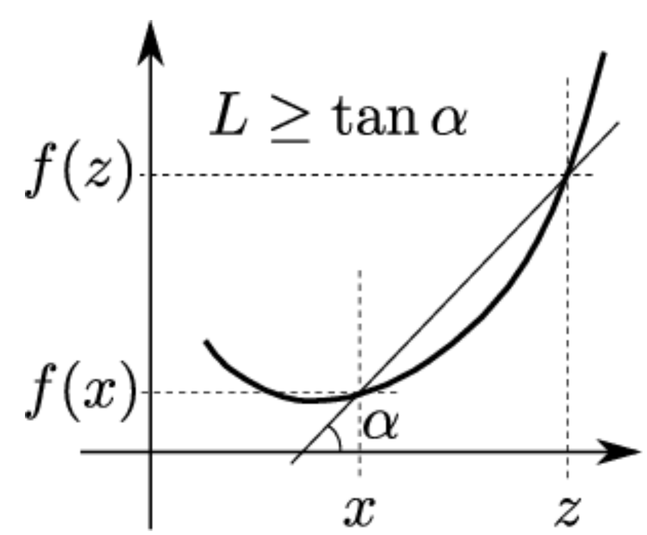
Learning Stable Control



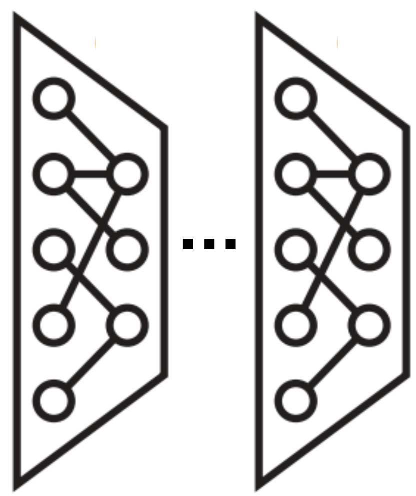
- Dynamics approximated by a DNN can lead to **unstable control**
- Constrain the Lipschitz constant of the DNN estimator with **spectral normalization**

Lipschitz constant: upper limit of the slope

$$L = \sup \frac{|f(z) - f(x)|}{|z - x|}$$



Spectral Normalization



function composition

$$f(\mathbf{x}) = g^{L} \circ g^{L-1} \cdots g^{1}(\mathbf{x})$$

$$g^{L}(\mathbf{x}) = a(W^{L}\mathbf{x})$$

Constrain the Lipschitz constant

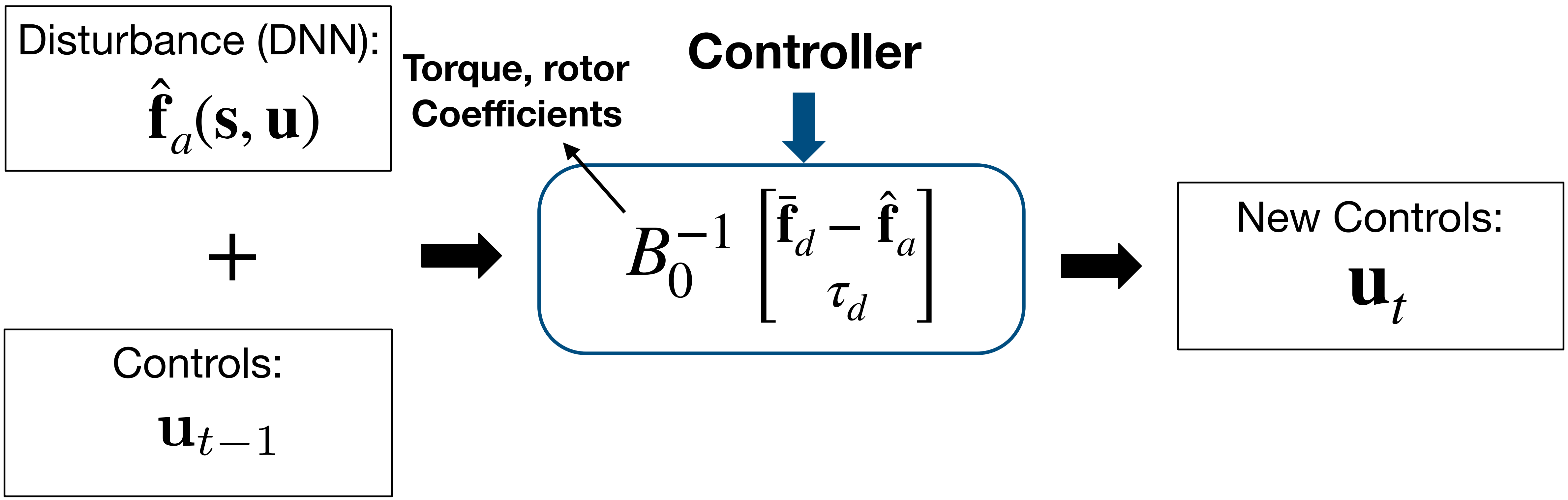
$$\|f\|_{Lip} \le \|g^{L}\|_{Lip} \cdot \|\phi\|_{Lip} \cdots \|g^{1}\|_{Lip}(\mathbf{x}) = \prod_{l=1}^{L} \sigma(W^{l})$$

Normalize the weights of a DNN by their singular values

$$\boxed{\bar{W} = W/\sigma(W)}$$

Largest Singular Value

Neural Lander



- **First** DNN-based nonlinear feedback linearization controller
- Solved with a fixed point iteration method
- **Guaranteed** stability in feedback control loop

Combat Ground Effect

Neural Lander

Stable Drone Landing Control using Learned Dynamics

Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli,
Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung

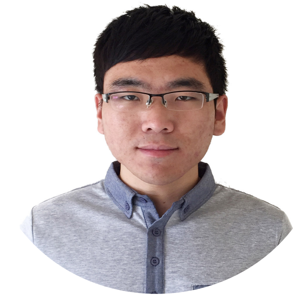
“Time and space are not conditions of existence,
time and space is a model of thinking.”

–Albert Einstein

Anima Anandkumar
Caltech/NVIDIA

Kamyar Azizzadenesheli
UCI

Soon-Jo Chung
Caltech



Yaguang Li
USC

Yan Liu
USC/DiDi

Yukai Liu
Caltech

Cyrus Shahabi
USC

Guanya Shi
Caltech

Yisong Yue
Caltech

Stephan Zheng
Salesforce

NSF Caltrans Raytheon STATS NVIDIA amazon

roseyu@northeastern.edu

@yuqirose