KDD2018

Artificial Intelligence in Transportation

Zheng WANG

DiDi AI Labs

Didi Chuxing

Yan LIU

DiDi AI Labs

Univ. of Southern California

Jieping YE

DiDi AI Labs

Univ. of Michigan, Ann Arbor

Outline

- Challenges and opportunities in transportation AI (20min)
  - Overview of urban transportation
  - The emerging challenges in transportation AI
- AI applications in transportation (165min+Break)
  - Map services I: map matching, route planning, estimated time of arrival (ETA) (60min)
  - Break (30min)
  - Map services II: traffic estimation, traffic forecast (45min)
  - Decision making services: dispatching (30min)
  - AI applications and AI for social good (30min)
- Data and tools for transportation AI (15min)
- Q&A (10min)

Part 1: Challenges and Opportunities in Transportation AI

History of Urban Transportation

First Wheeled Vehicle

3500 B.C.

First Road Network

1st Century

First Public Transportation

1662

First Bicycle

1817

First Affordable Automobile

1908

First Electric Traffic Light

1910s

U.S. National Highway System

1956

Smart Transportation System

Now and Future

History of Traffic Light

No Traffic Light

Before

Hand-operated Traffic Light

1860s

Electric Traffic Light

1910s

1920s

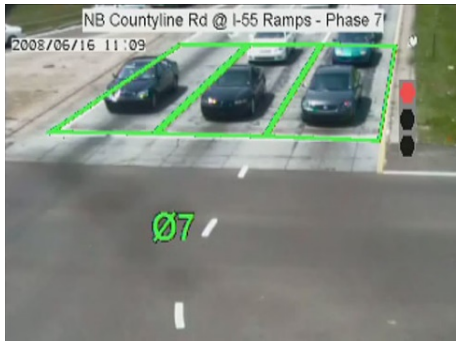
Weigh in Motion

Radar

1960s

Loop Detector

1980s



Camera

Transportation: A Multi-Disciplinary Industry

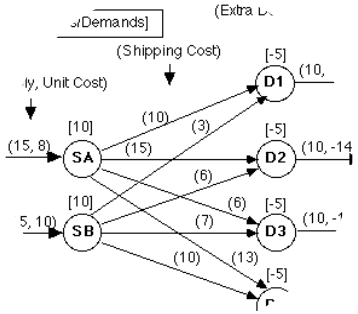
Engineering

Policy Making

Planning

Transportation

Design



Science

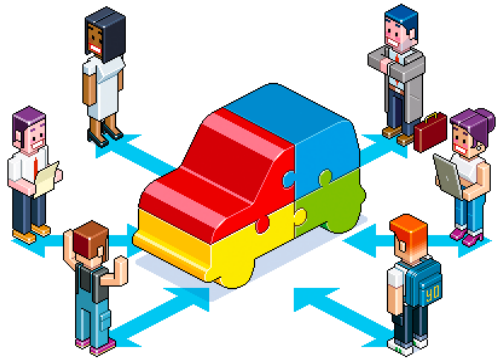
Management

.....

Cutting-Edge Issues

Cooperative Vehicle-Highway Systems

Ride Sharing

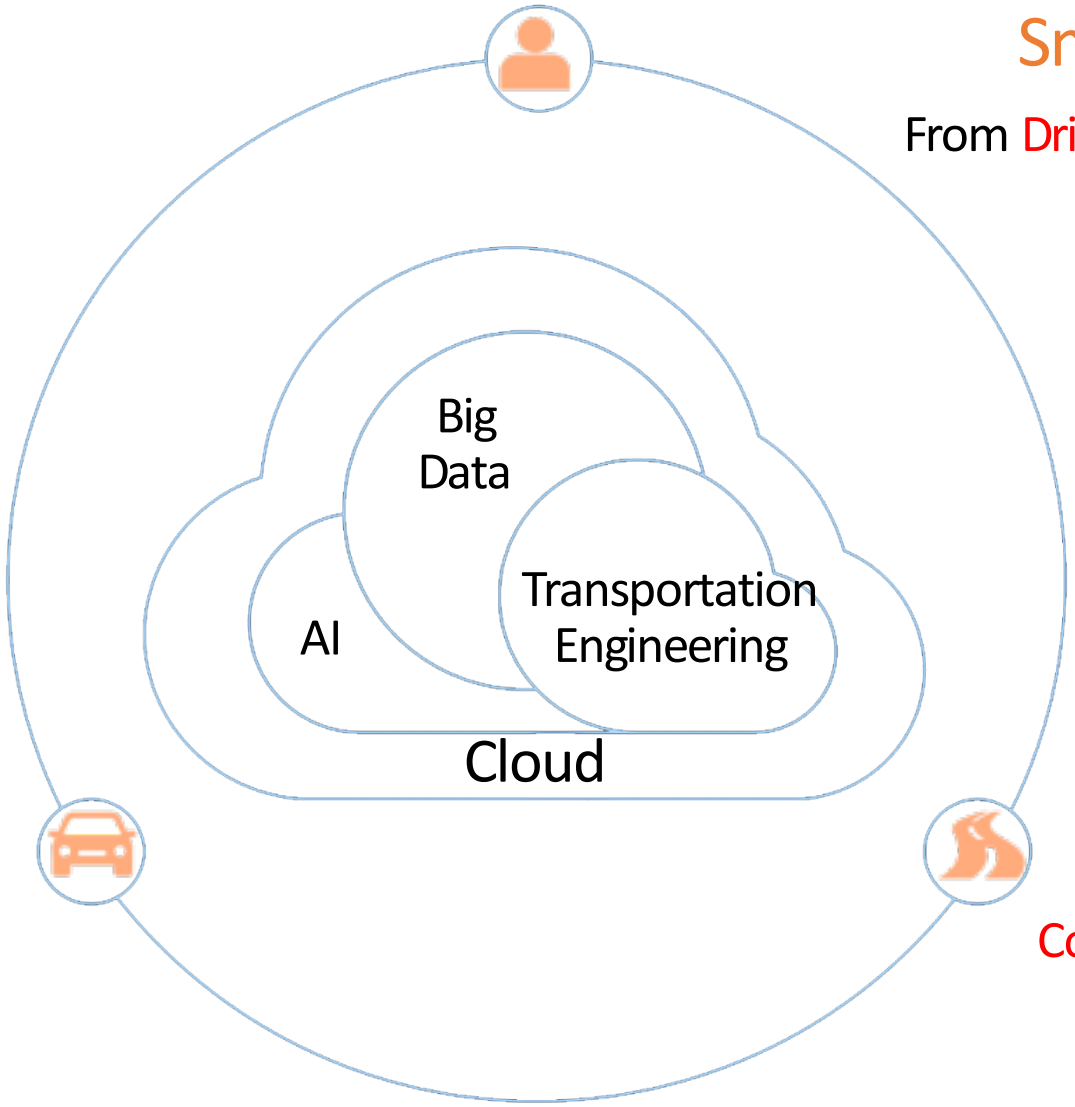


Multimodal Transportation

Smart Transportation System

Smart Travelers

From Drive Alone to Ride Sharing



Smart Vehicles

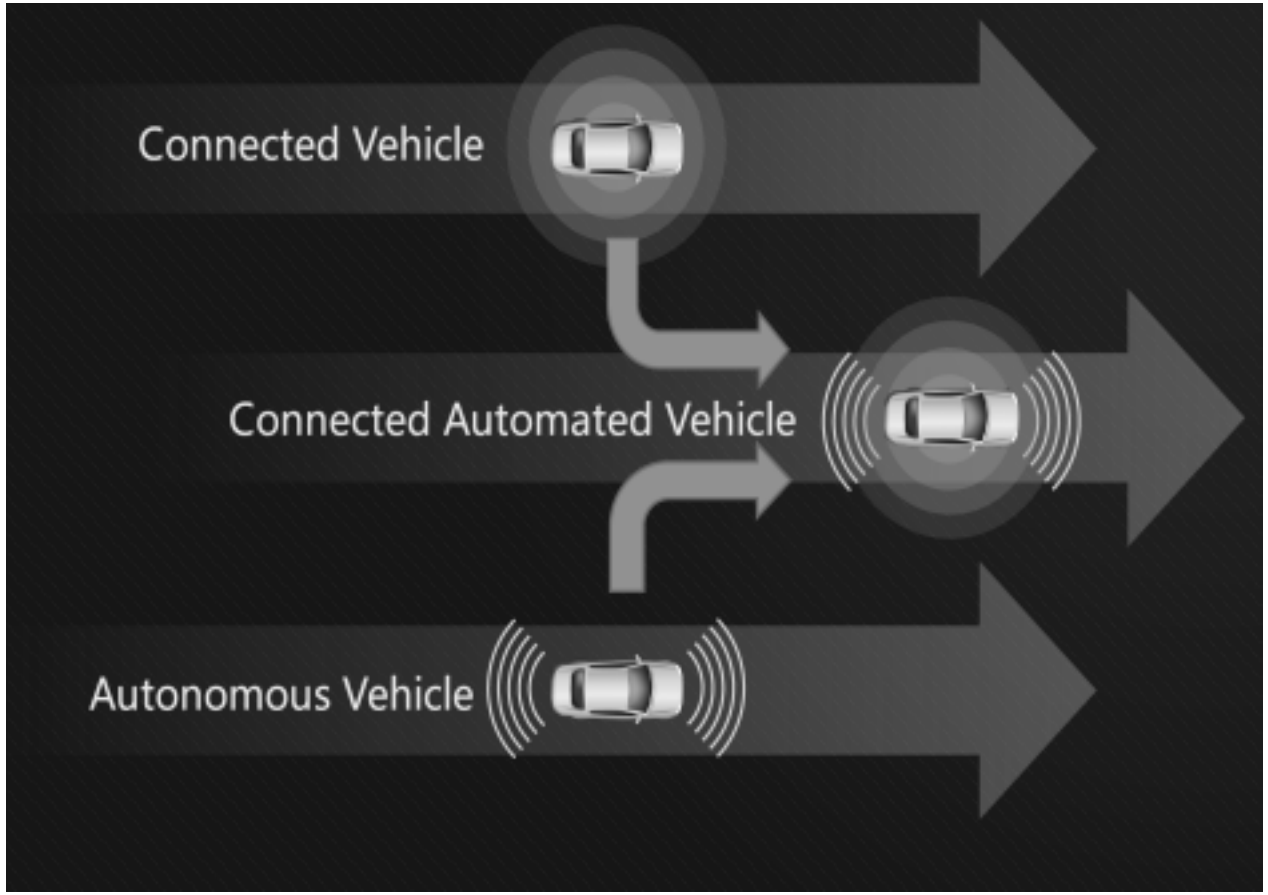
From Human Driving to
Autonomous Driving

Smart Infrastructure

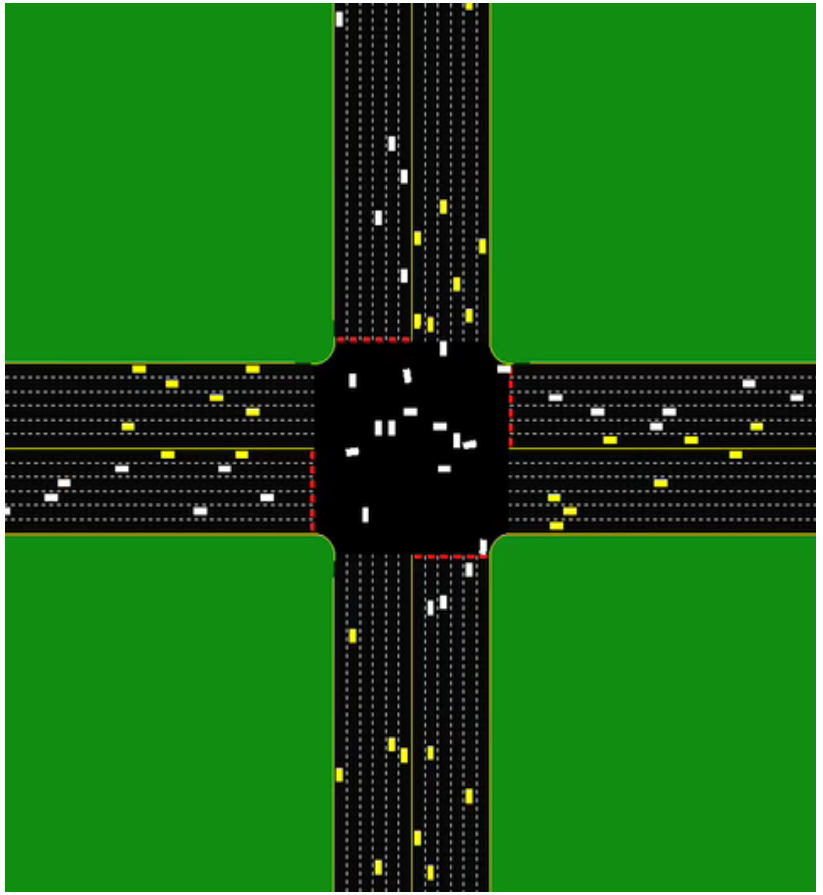
From Independent Systems to
Cooperative Vehicle-Highway Systems

Building the Brains of Smart Transportation

Intelligent and Connected Vehicles



Future Intersection



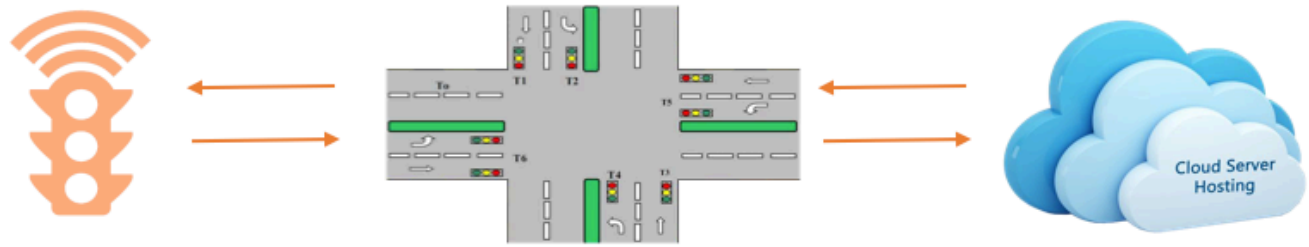
Adaptive Control as a Service

Framework

Traffic Signal

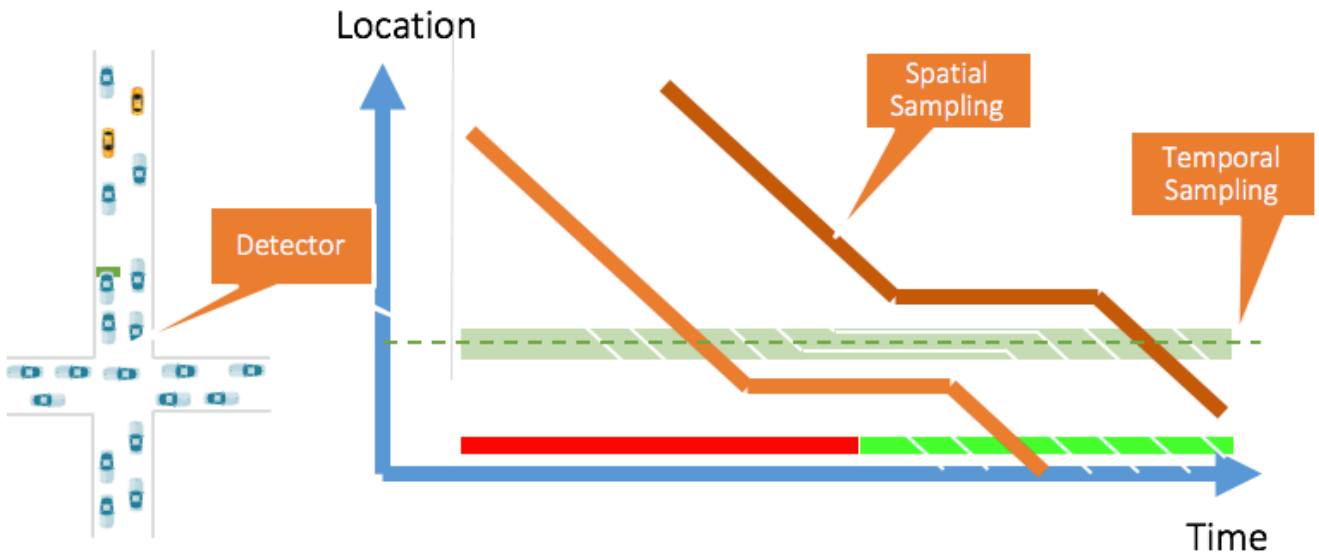
Signal Management System

Cloud Server

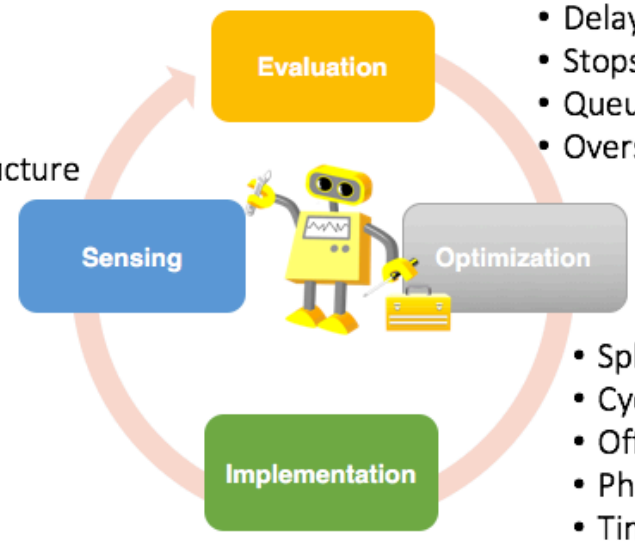


Case Study

Cloud Based Signal Control Platform



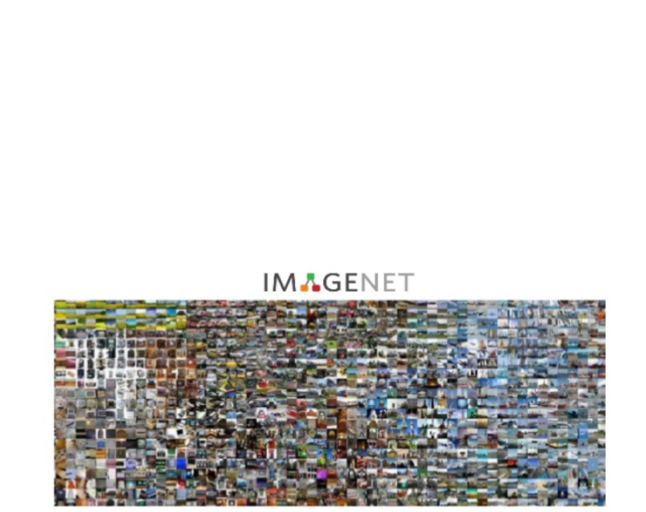
- Infrastructure Sensors
- Mobile Sensors



- Delay
- Stops
- Queue Length
- Oversaturation

- Splits
- Cycle Length
- Offset
- Phase Sequence
- Time-of-day Plan

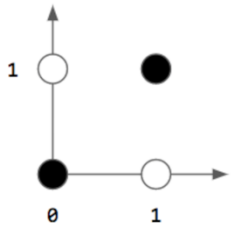
AI and Machine Learning



?

Neural Networks

Deep Learning



Machine Learning:
supervised,
unsupervised

Reinforcement
Learning

AlphaGo

Data Resource

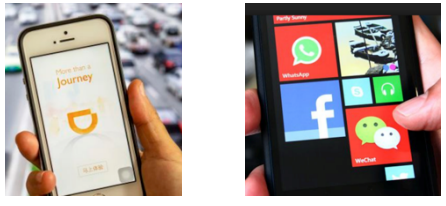
- Location data, Trajectory data

- Transaction data

- Profile data

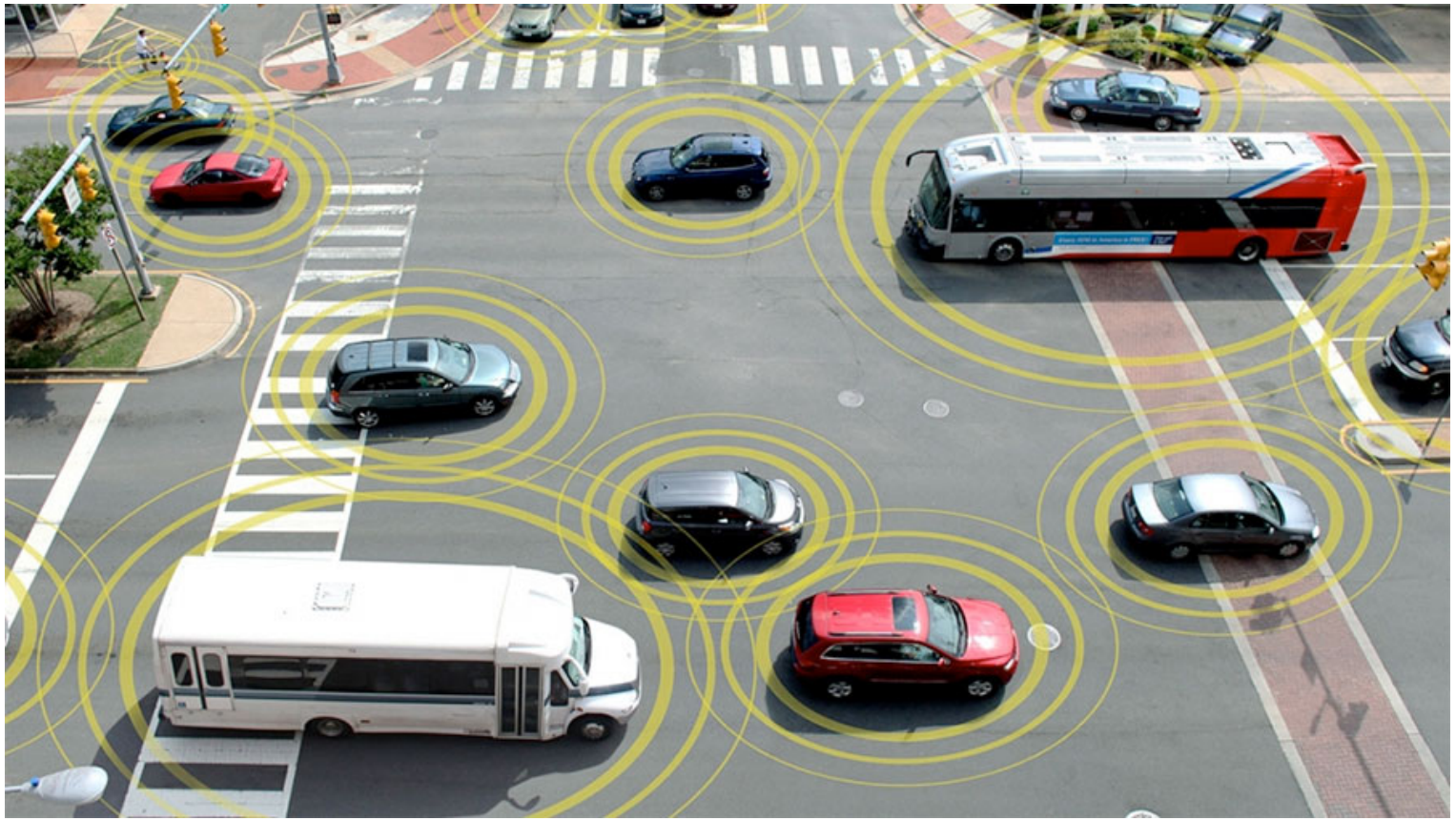
- Sensors: multimedia data

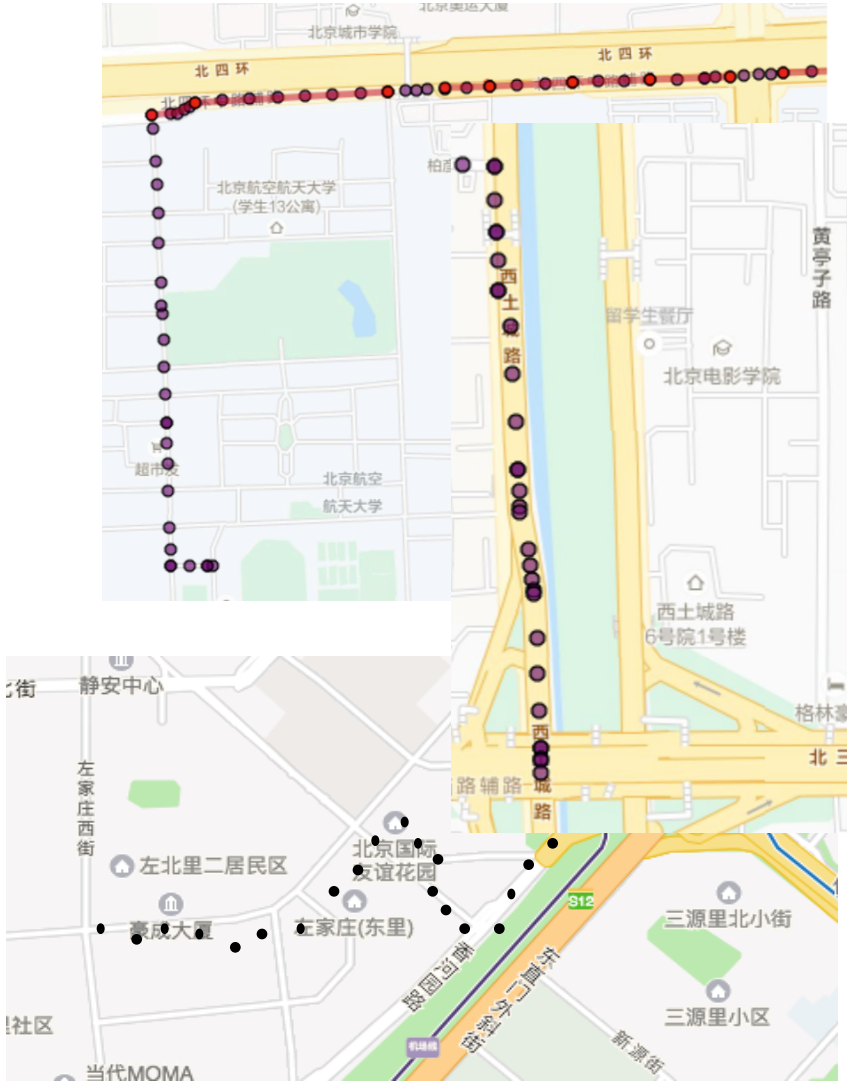
- Cross-platform identification



GPS Data

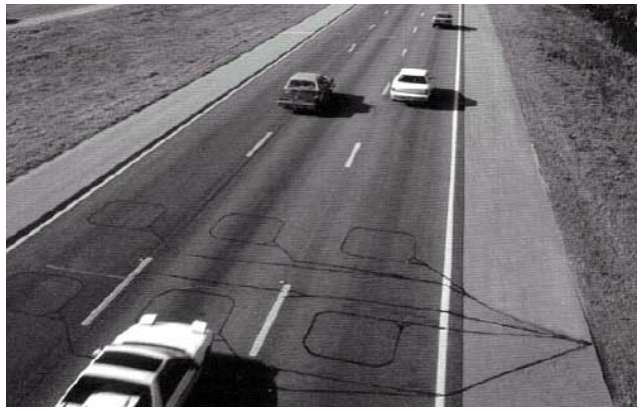
Location Data and Floating-Car Trajectory

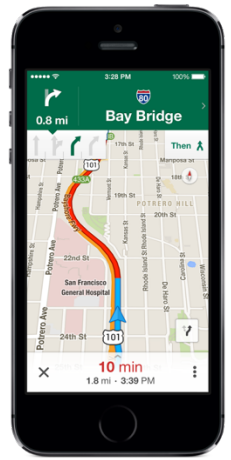




Sensors

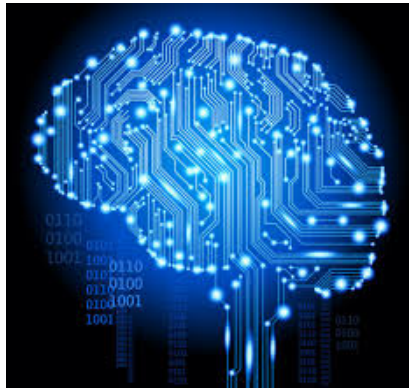
Loop detector, camera, microphone, mobile sensors …

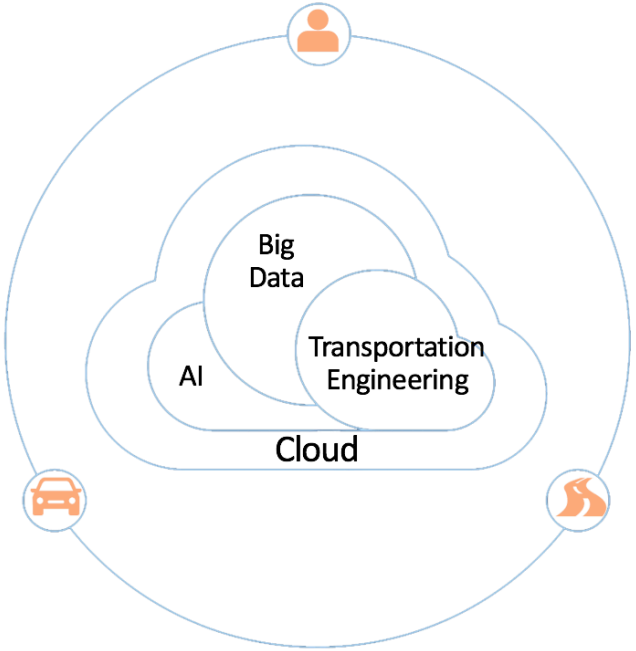




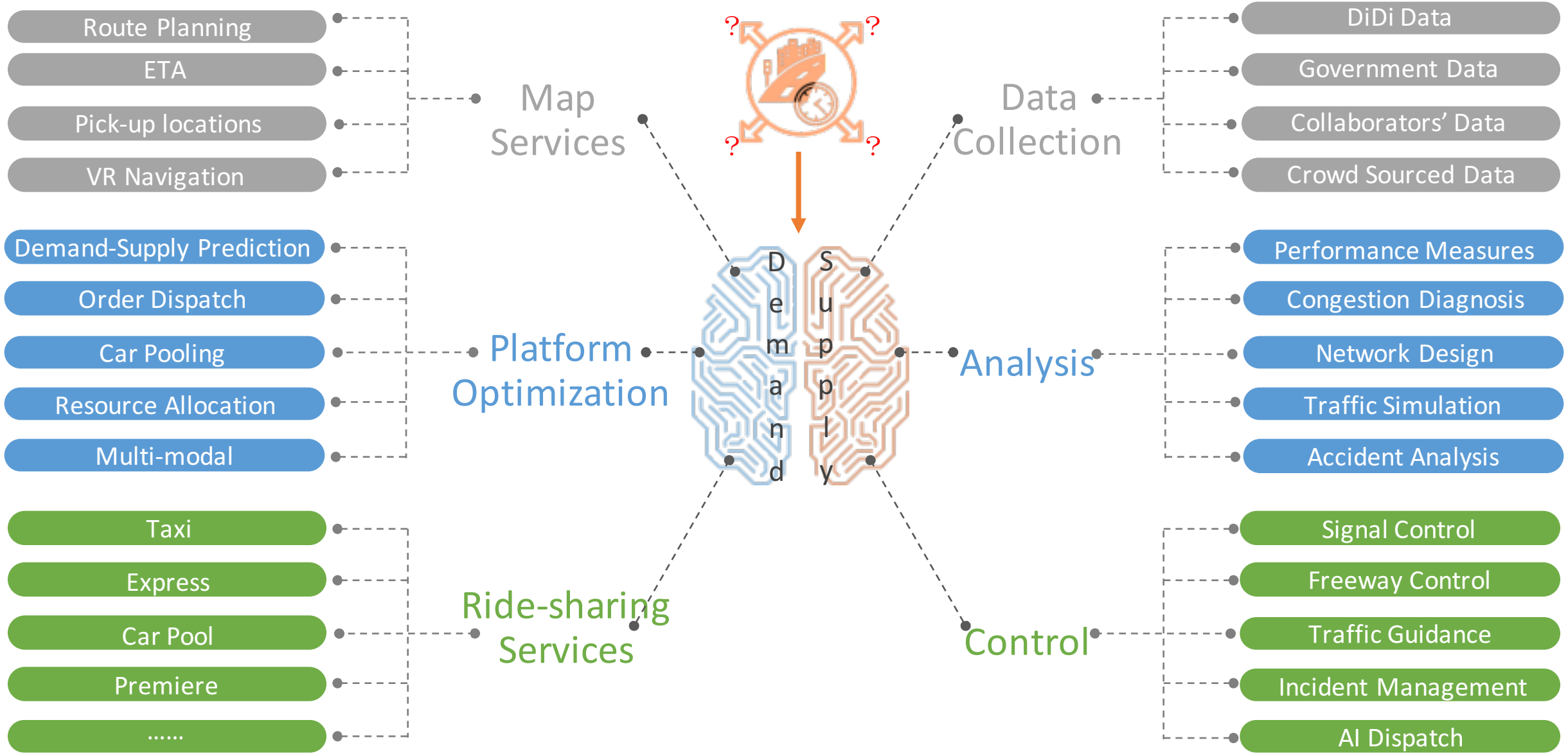
Transportation AI

Big data makes AI possible for transportation.





Smart Transportation Brain



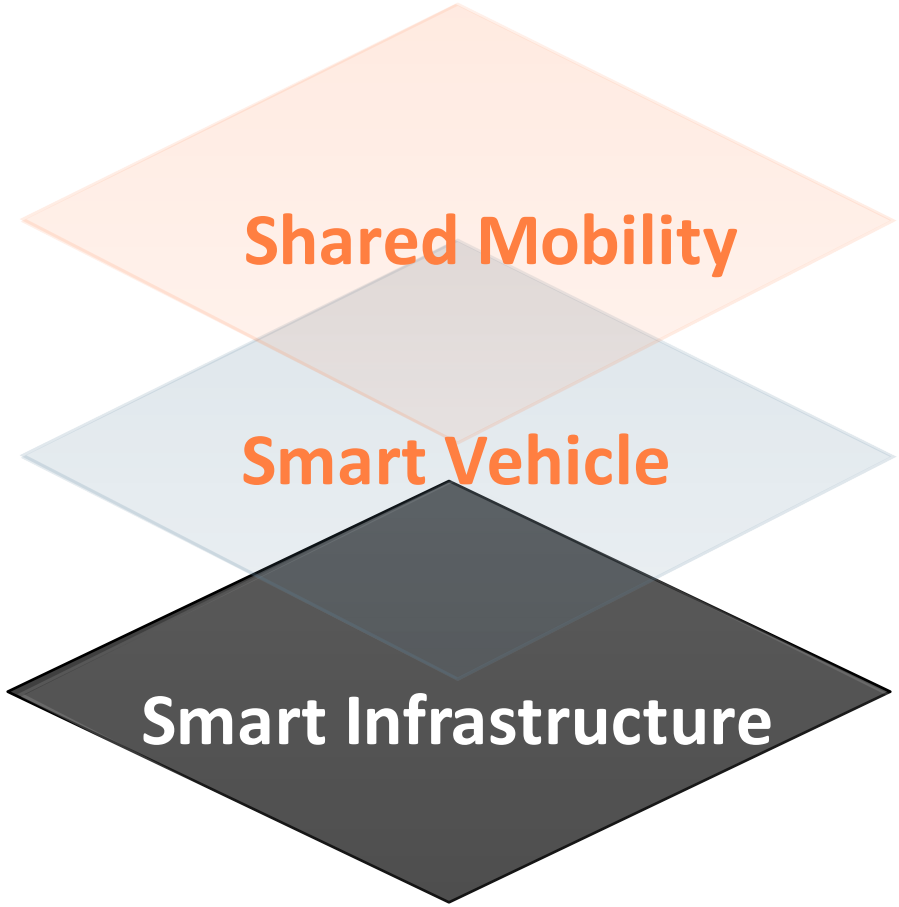
Outline

- Challenges and opportunities in transportation AI (20min)
  - Overview of urban transportation
  - The emerging challenges in transportation AI
- AI applications in transportation (165min+Break)
  - Map services I: map matching, route planning, estimated time of arrival (ETA) (60min)
  - Break (30min)
  - Map services II: traffic estimation, traffic forecast (45min)
  - Decision making services: dispatching (30min)
  - AI applications and AI for social good (30min)
- Data and tools for transportation AI (15min)
- Q&A (10min)

KDD2018

Part 2: AI Applications in Transportation

Future Transportation



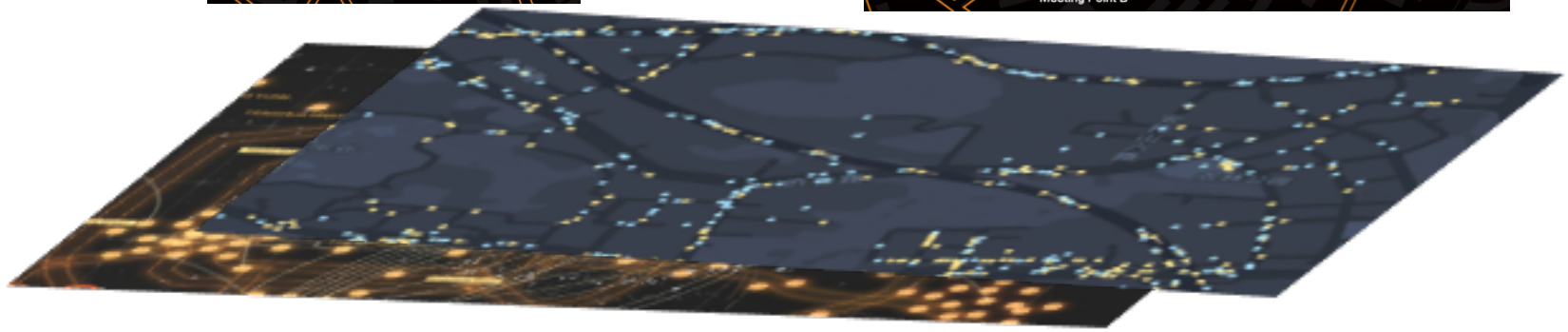
Map Service, Decision Service, …

Electrical Vehicle, Autonomous Vehicle, …

Highway, Road, Smart Traffic Light, …

Key Components

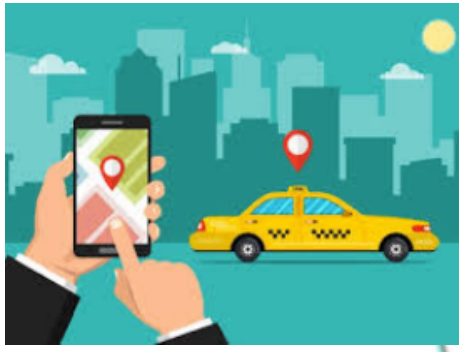
- Basic Layer : map service and LBS
- Upper Layer: decision service and marketplace



Map Service I

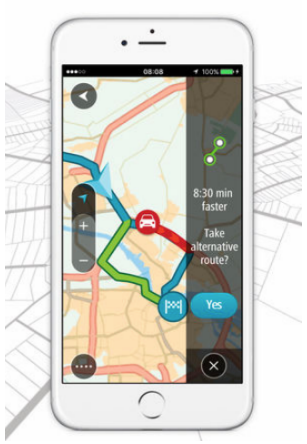
Smart Map for Modern Transportation

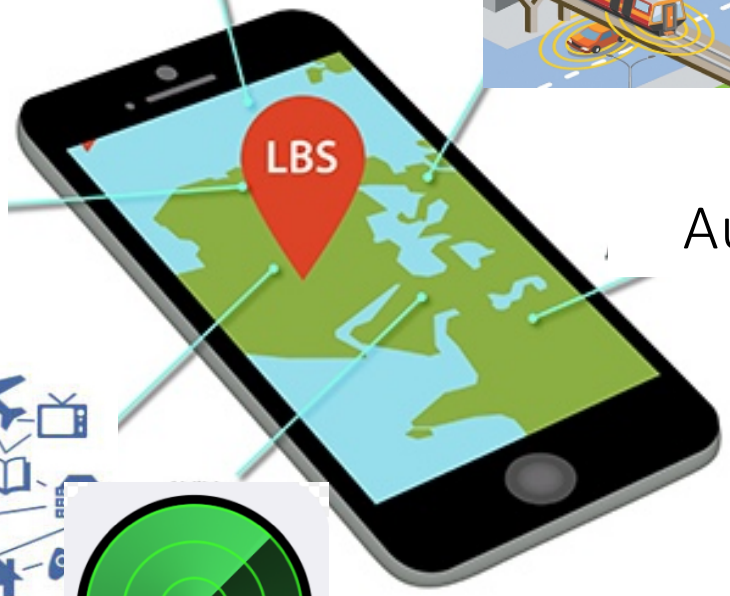
Ride Hailing



Transportation System Efficiency

Navigation



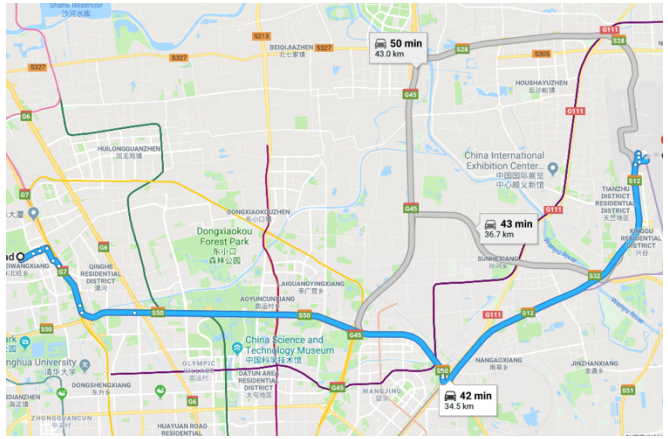


Autonomous Driving

Core Map Service

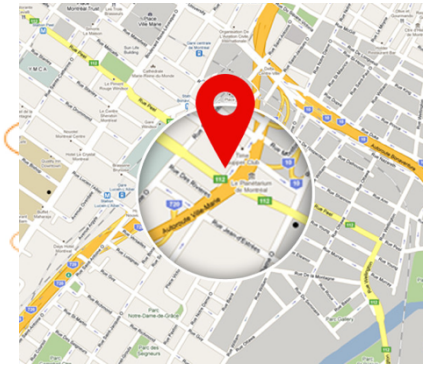
Route Planning

ETA

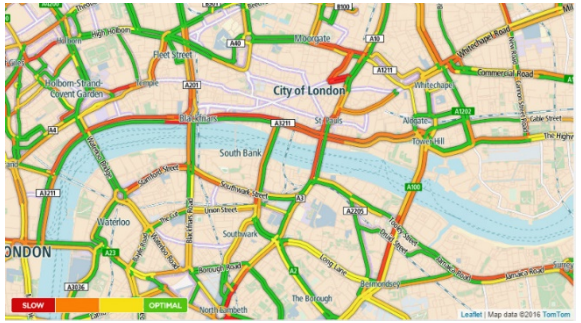


Map Matching

Positioning



Traffic

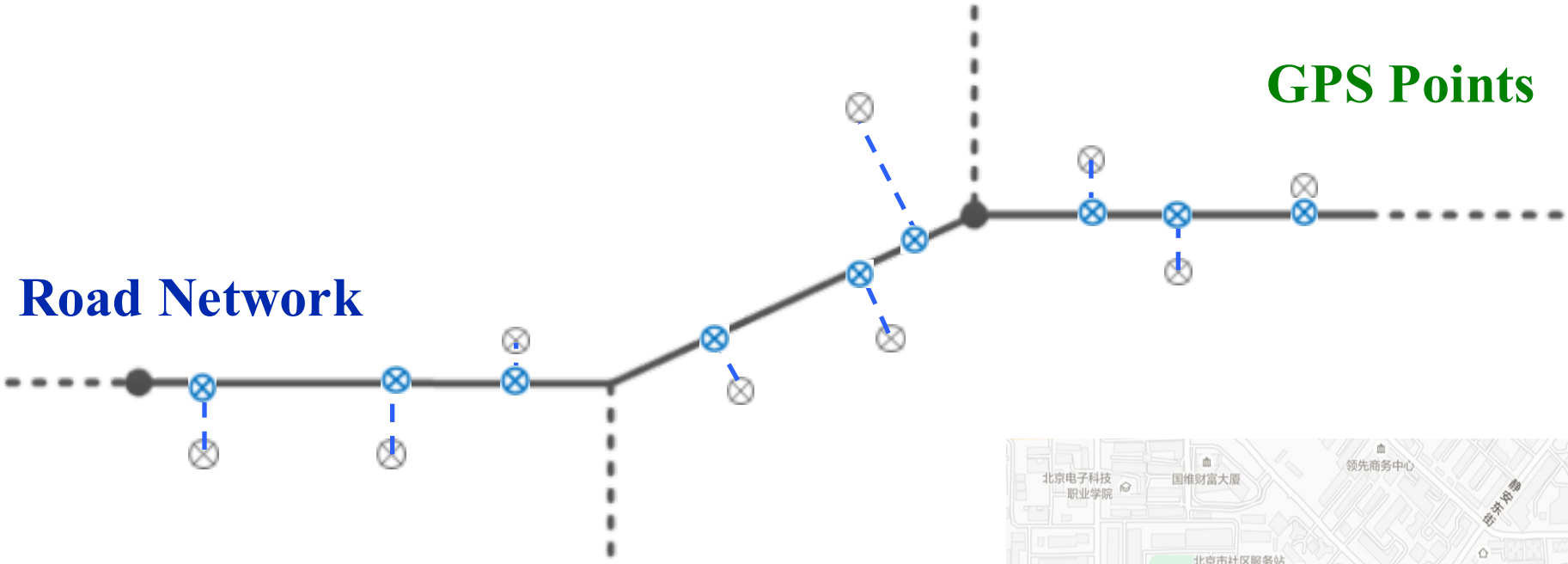


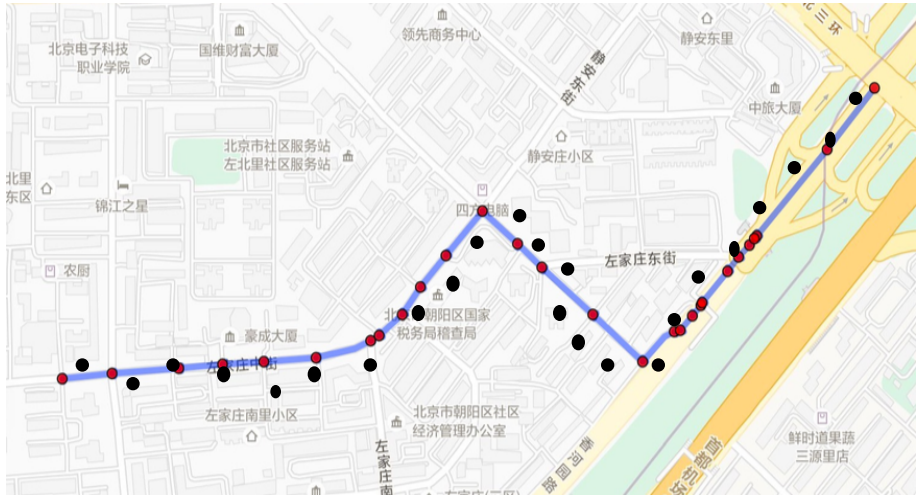
Map Matching

Map Matching

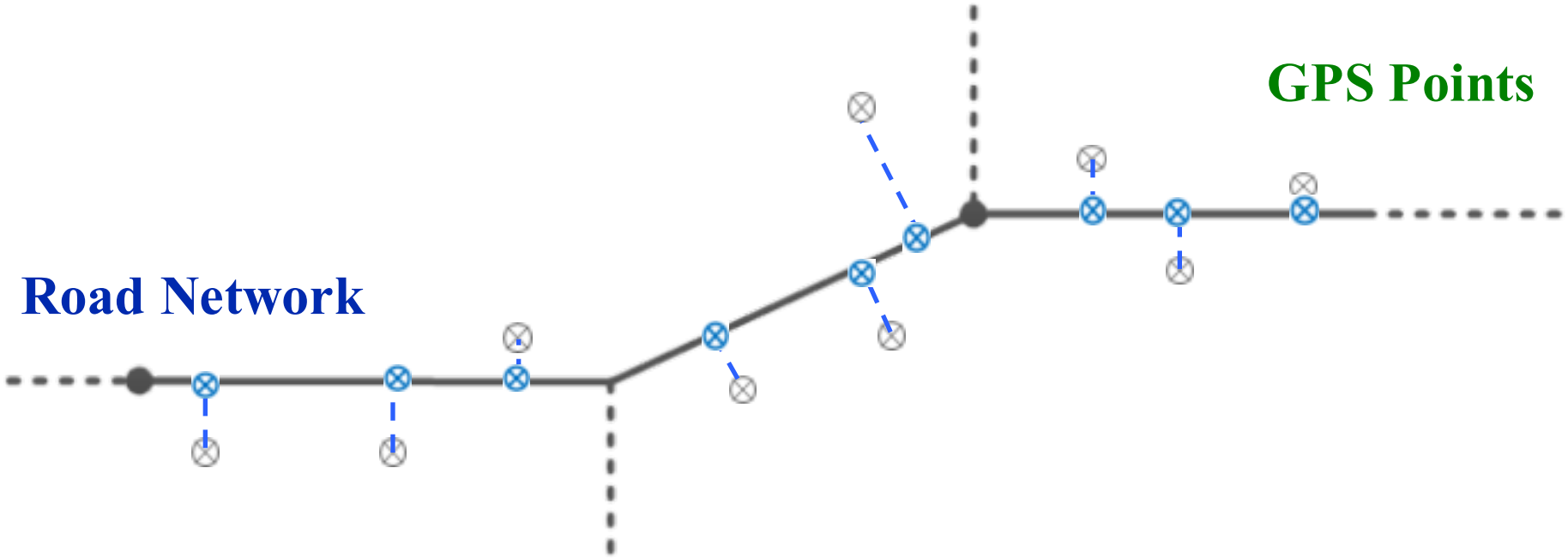
- Problem Definition
- Solutions to Map Matching
- Challenges

Map Matching



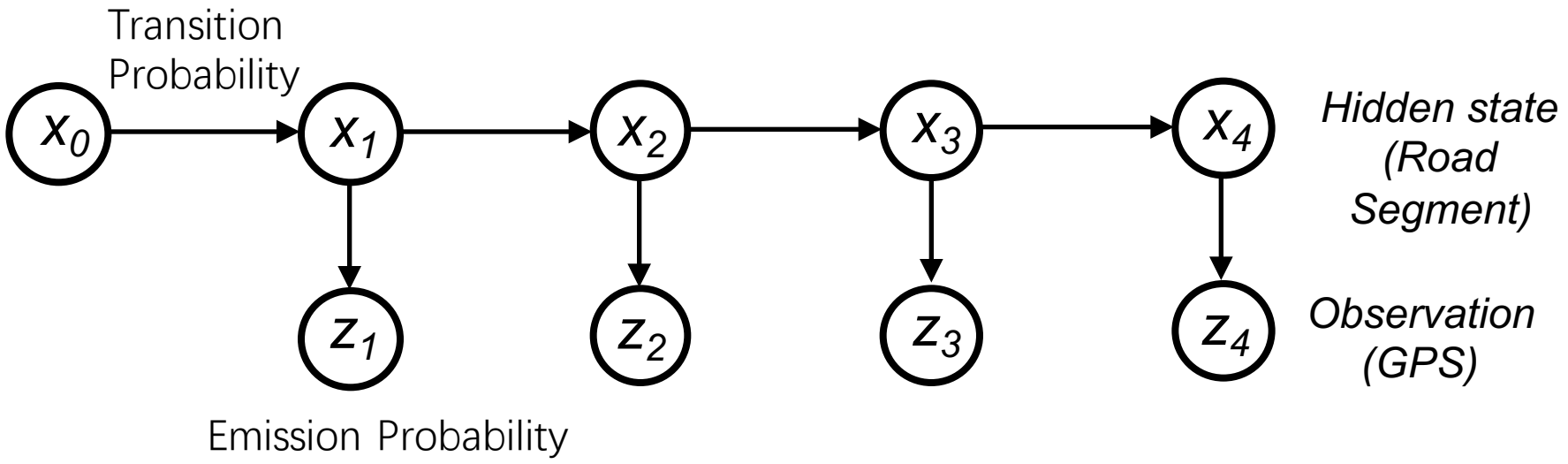


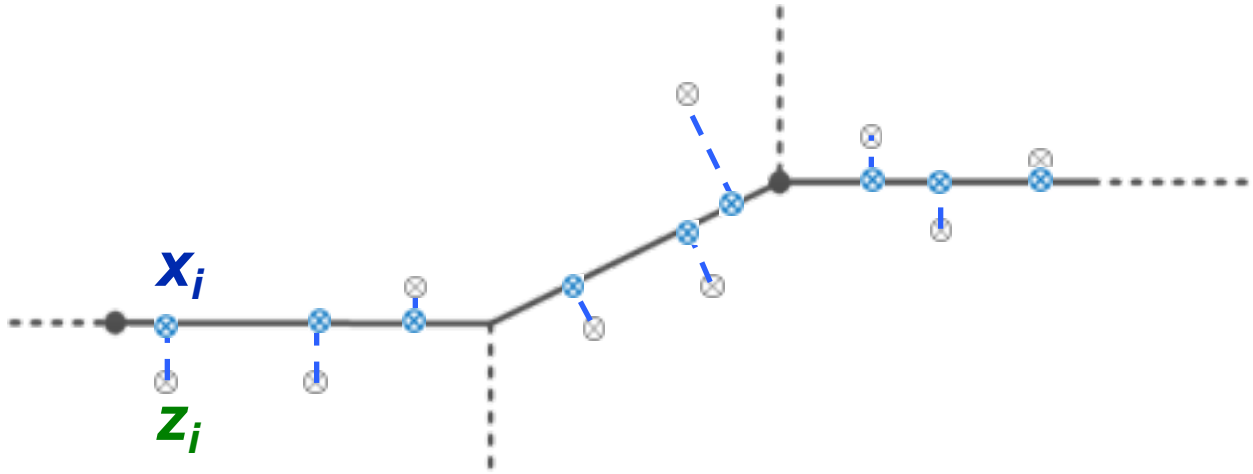
Naive Approach: Nearest Neighbor



Sequence to Sequence: HMM

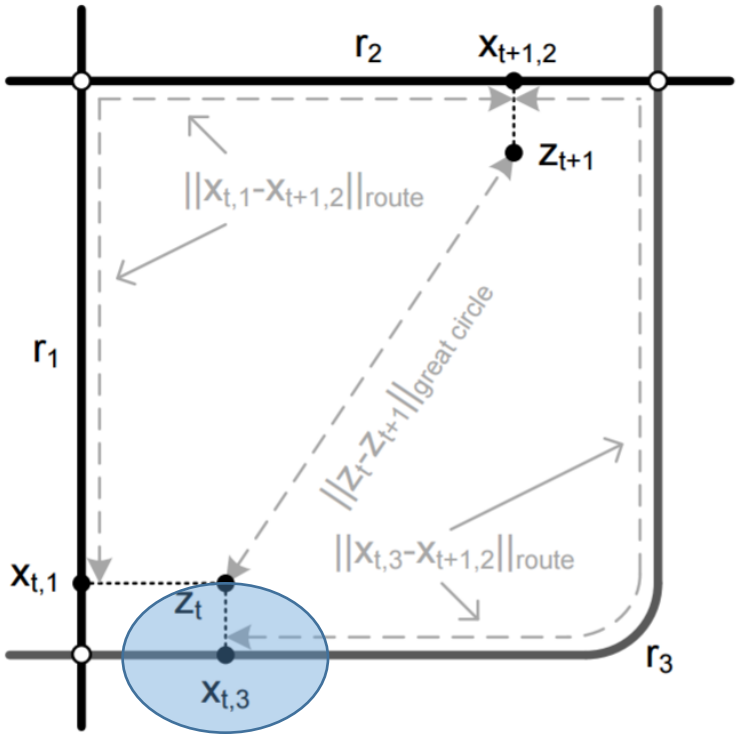
- Model this process using Hidden Markov Model.

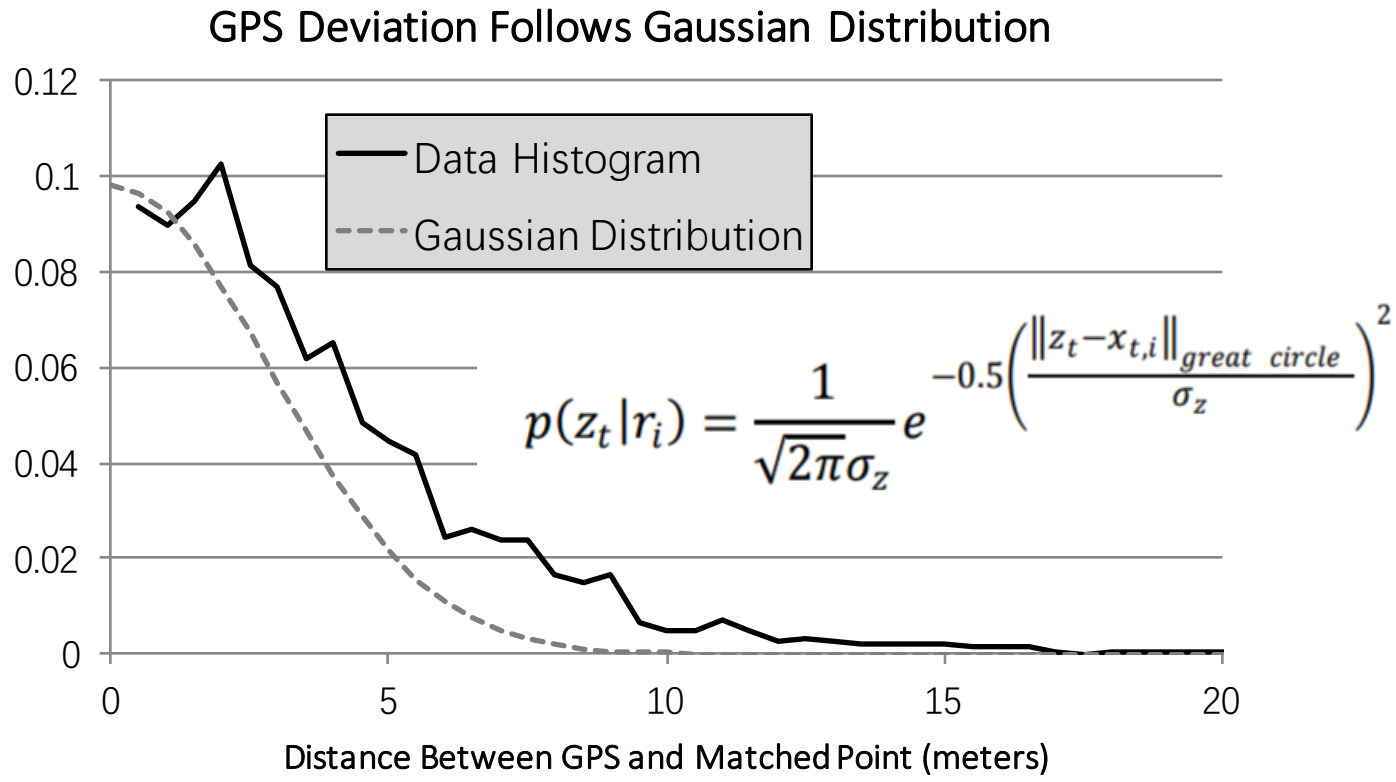




* Paul Newson et al. Hidden Markov Map Matching Through Noise and Sparseness. ACM SIGSPATIAL 2009

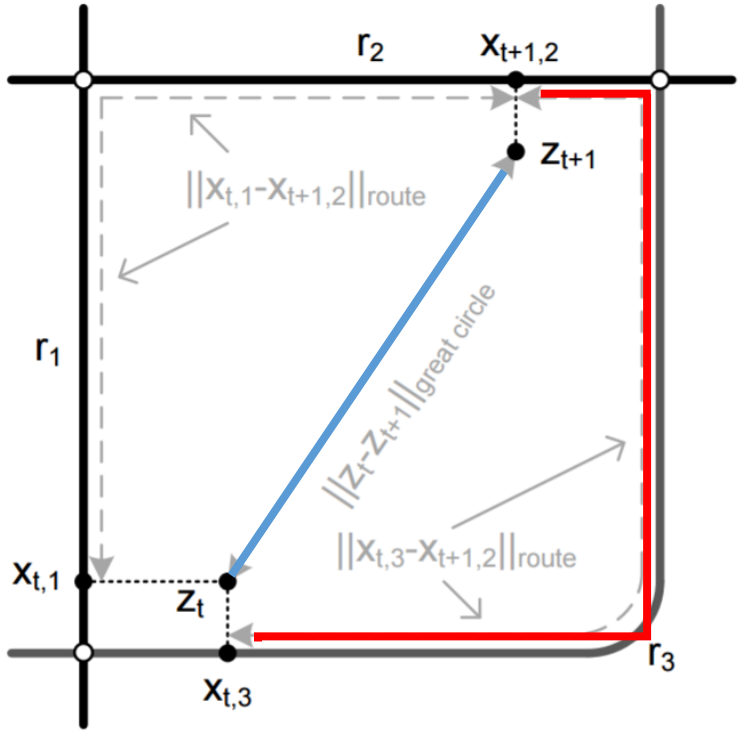
Emission Probability

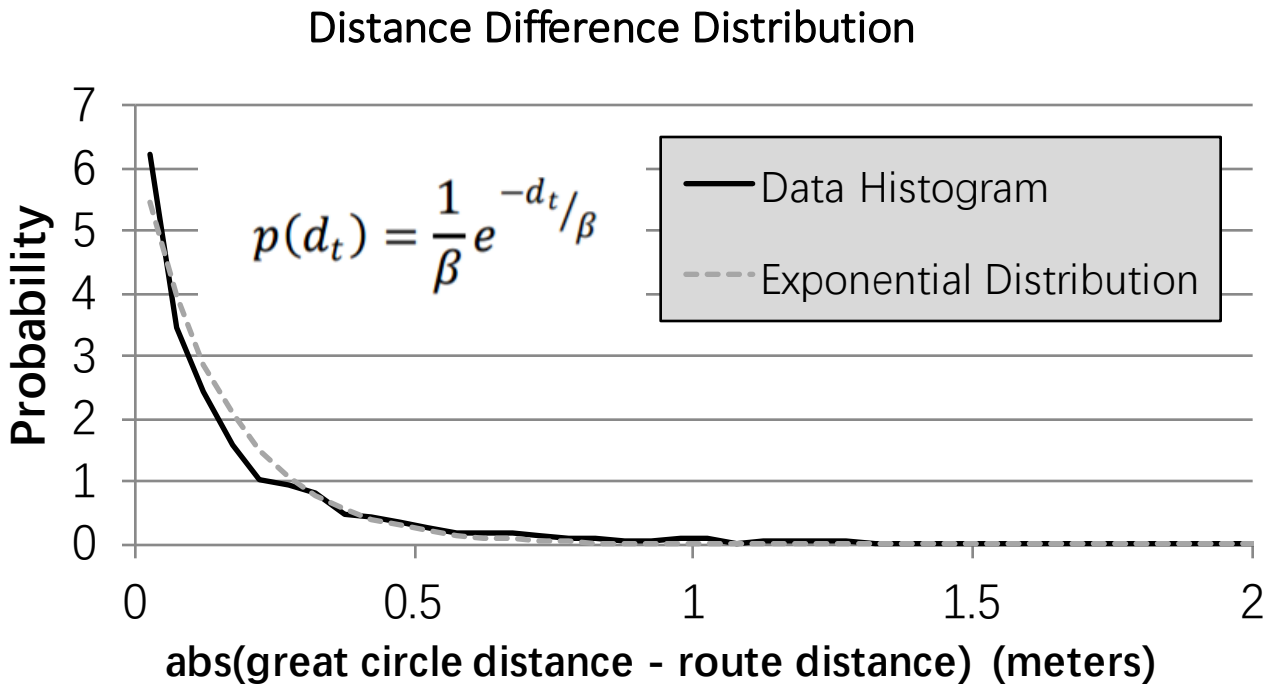




* Part of this slide is borrowed from: Paul Newson et al. Hidden Markov Map Matching Through Noise and Sparseness. ACM SIGSPATIAL 2009

Transition Probability





$$d_t = \left| \|z_t - z_{t+1}\|_{great\ circle} - \|x_{t,i^*} - x_{t+1,j^*}\|_{route} \right|$$

* Part of this slide is borrowed from: Paul Newson et al. Hidden Markov Map Matching Through Noise and Sparseness. ACM SIGSPATIAL 2009

Parameter Estimation

- In reality: heuristic estimation
  - Emission probability parameter (noise in location measurements):

$$\sigma_z = 1.4826\ \mathrm{median}_t \left(\left\| z_t - x_{t,i^*} \right\|_{great\ \ circle} \right)$$

  - Transition probability parameter (tolerance of non-direct routes):

$$\beta = \frac{1}{\ln(2)} \mathrm{median}_t \left(\left| \left\| z_t - z_{t+1} \right\|_{great\ \ circle} - \left\| x_{t,i^*} - x_{t+1,j^*} \right\|_{route} \right| \right)$$

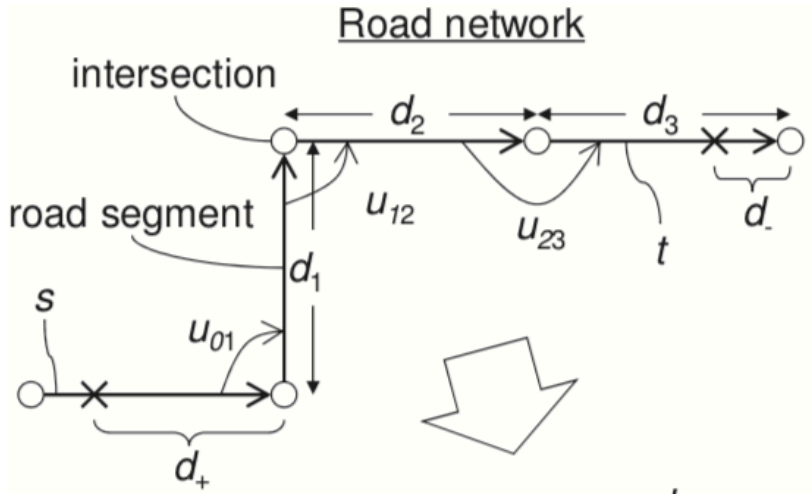
  - Parameter refinement
- In literature: parameter learning

Parameter Learning with IRL

- Map Matching with Inverse Reinforcement Learning
  - Transition probability variant, HMM
  - $P_{trans} = \frac{1}{\beta}\exp(-\frac{|r^* - g_0|}{\beta})$
  - Conventionally, $r^* = \|d_1 + d_2 + d_3 + d_+ - d_-\|$
  - Here $r^* = \|d_1 + d_2 + d_3 + d_+ - d_- + w_{turn}(u_{01} + u_{12} + u_{23})\|$

$$u_{v\,v'} = \begin{cases} 0 & \text{if } |\theta_{v\,v'}| < \pi/4, \\ 1 & \text{if } \pi/4 \le |\theta_{v\,v'}| \le 3\pi/4, \\ 2 & \text{if } 3\pi/4 < |\theta_{v\,v'}| < \pi, \\ 10 & \text{if } |\theta_{v\,v'}| = \pi. \end{cases}$$

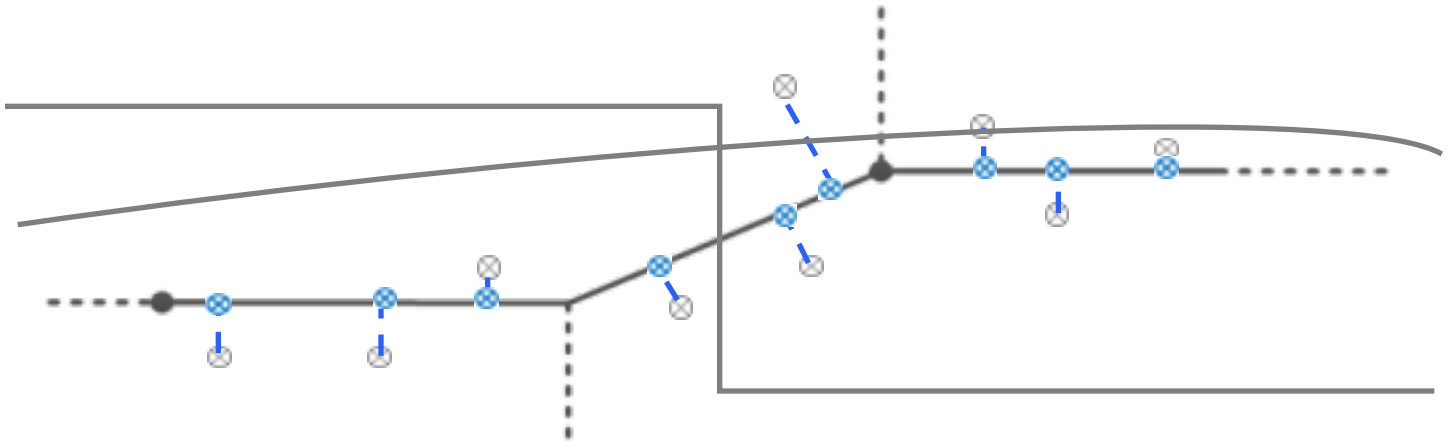
  - Weight estimation w_{turn} with IRL



* T. Osogami et al. *Map Matching with Inverse Reinforcement Learning*, IJCAI 2013.

State Estimation using Viterbi Algorithm

- Map-matching as state estimation
  - Input: a sequence of GPS points, HMM
  - Output: the most likely state sequence, i.e., a sequence of edges in the road network.
- Viterbi Algorithm
  - Dynamic Programming based method to identify the state sequence with the highest probability.



Challenges

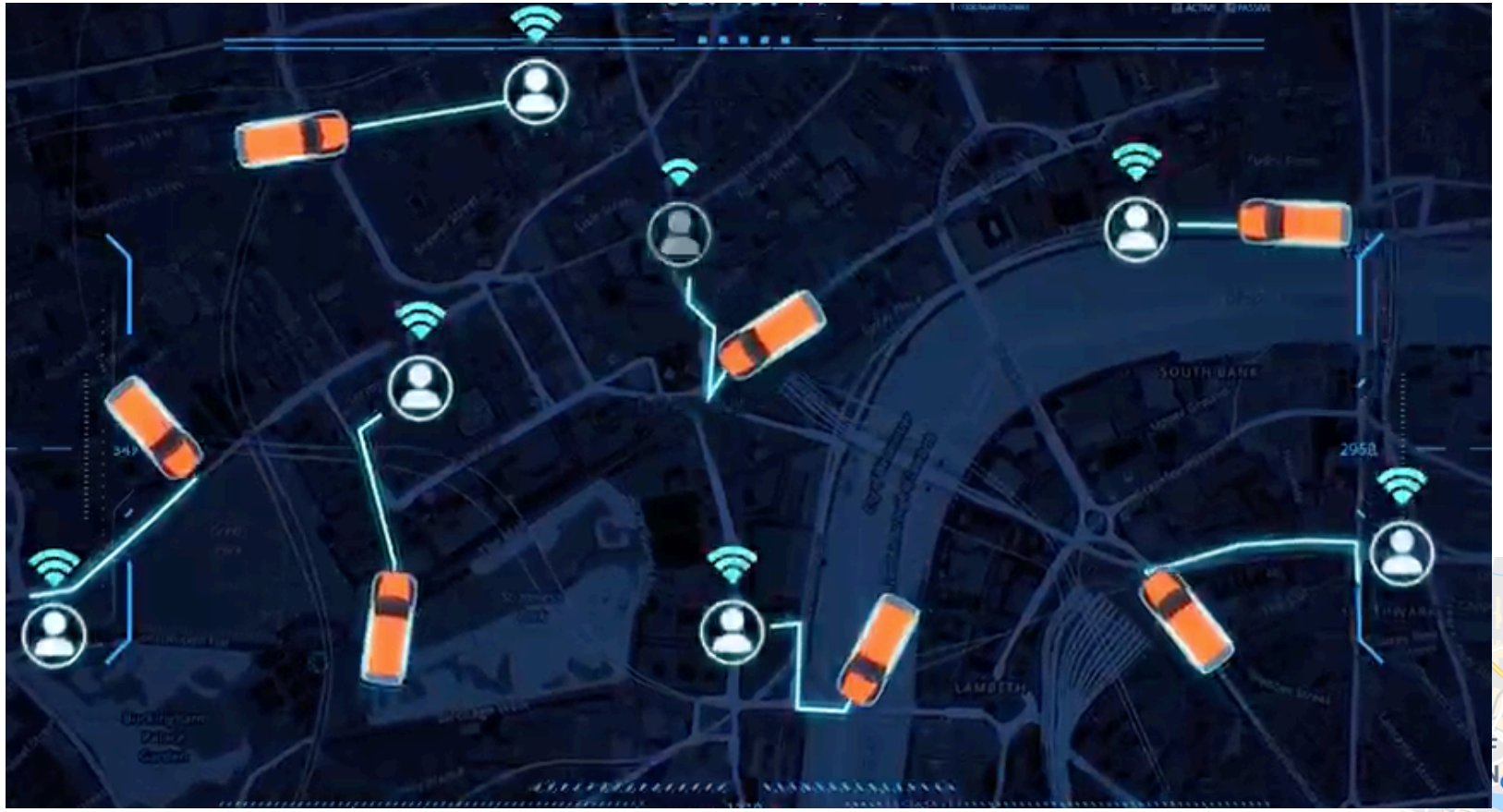
- Large-scale GPS data
- Low-quality GPS data for mobile device
- Limited amount of labeled data
  - Unsupervised learning: EM for HMM
  - Semi-supervised learning

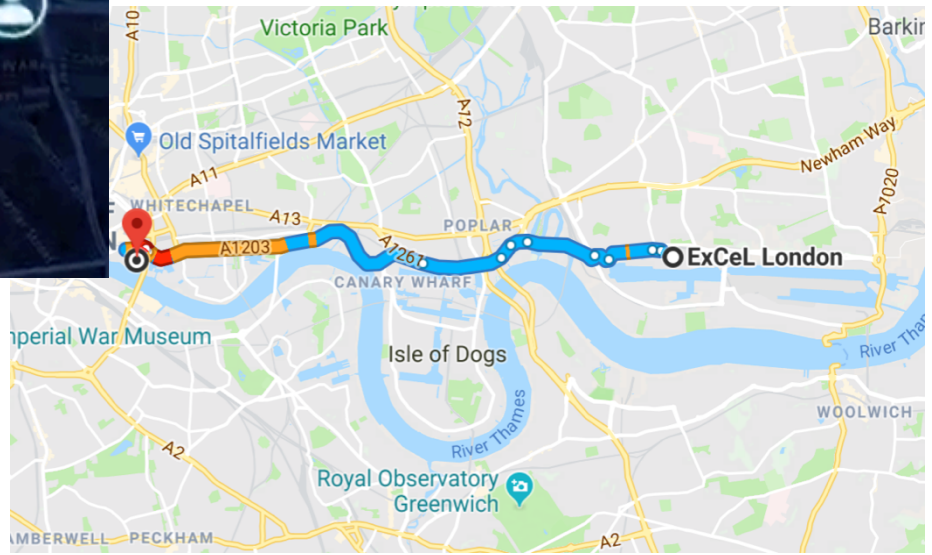
Reference

- Sotiris Brakatsoula et al. On map-matching vehicle tracking data, VLDB 2005
- Paul Newson et al. Hidden Markov map matching through noise and sparseness, ACM SIGSPATIAL 2009
- Yin Lou et al. Map-matching for low-sampling-rate GPS trajectories, ACM SIGSPATIAL 2009
- T. Osogami et al. Map Matching with Inverse Reinforcement Learning, IJCAI 2013

Route Planning

Route Planning



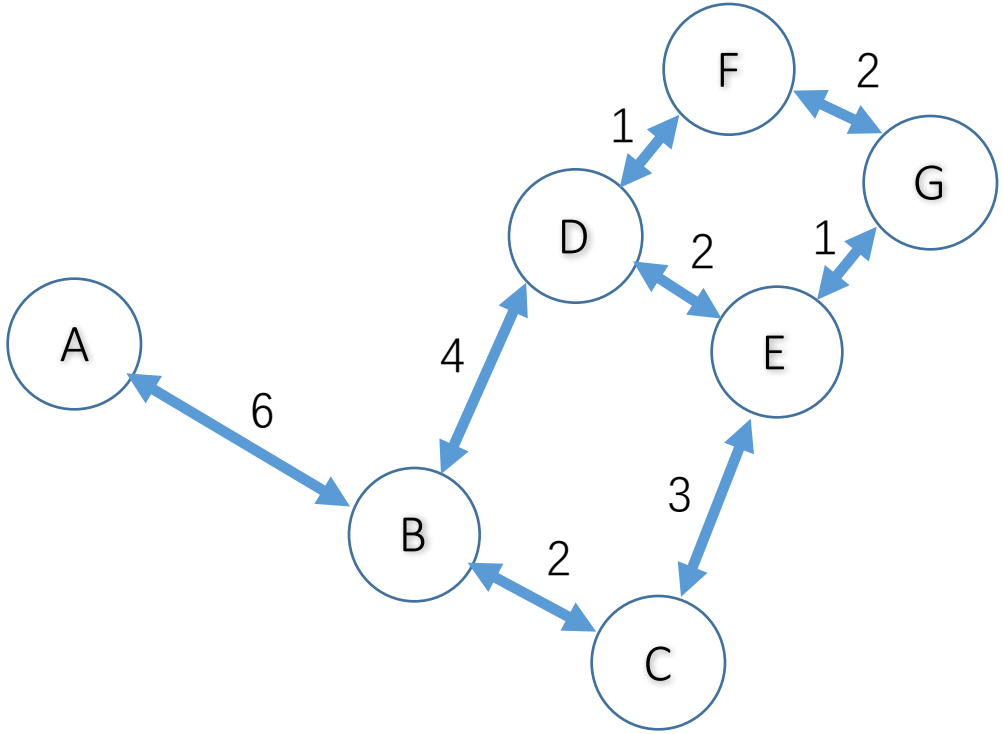


Route Planning

- Challenges:
  - Large-scale transportation network
  - High query speed
  - Accurate result
- Classical problem: shortest path algorithm
  - Dijkstra's algorithm and its extensions: (high) query speed, less robust
  - A*-algorithm: robust, low query speed
  - Customizable routing: robust, relative high query speed
- Data-driven approaches
  - What is the proper edge weight for the graph?
  - How can we take advantage of the big data?

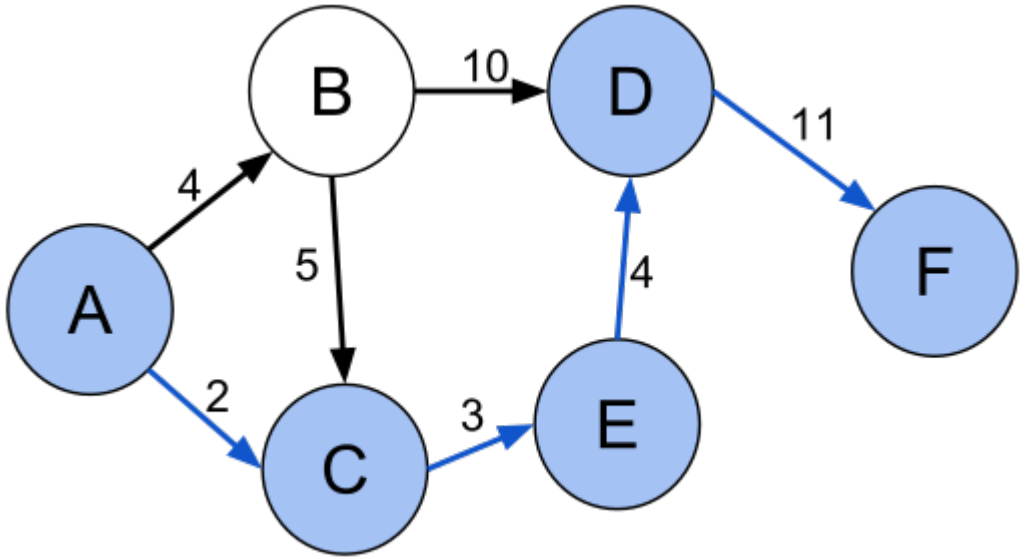
Shortest Path

- Build a graph based on map and traffic information.
- Graph edge weight for travel cost
- Find a route with the minimum travel cost.



Dijkstra's algorithm: a greedy approach

- Given a weighted graph with nonnegative edge weights, Dijkstra's algorithm finds the shortest path between nodes in the graph.



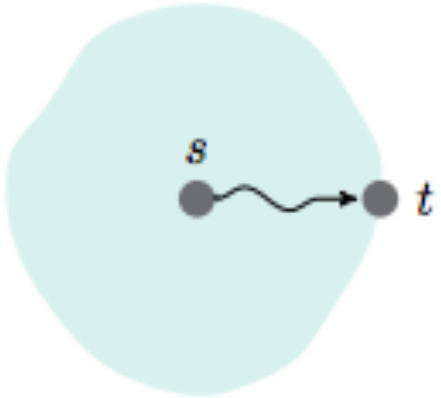
The **complexity** of Dijkstra's algorithm is $O(|e| + |v|\log|v|)$.

A*-Algorithm: a heuristic approach

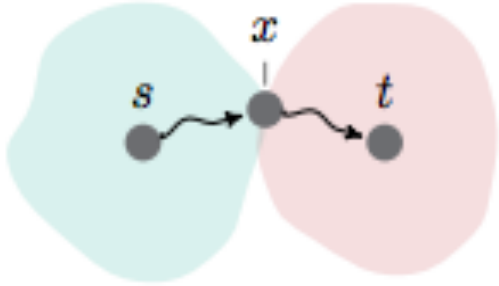
- Search with a heuristic guidance

$$f(n) = g(n) + h(n)$$

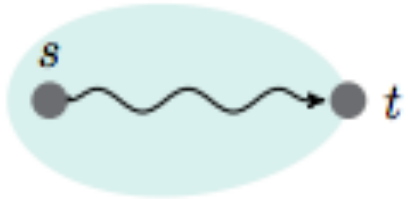
Forward Search



Bidirectional Search



A* Search

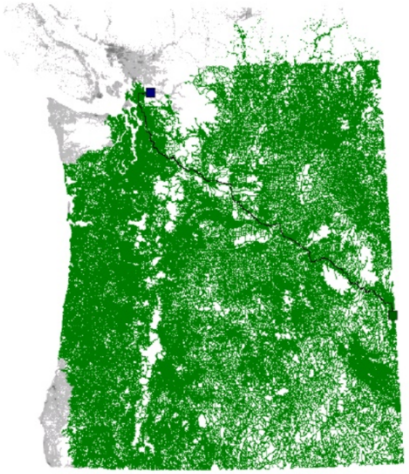


* Picture from Hannah Bast et al. *Route Planning in Transportation Networks.* Arxiv 2015.

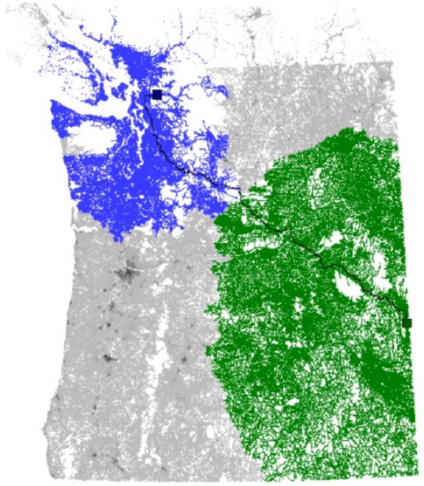
Speedup

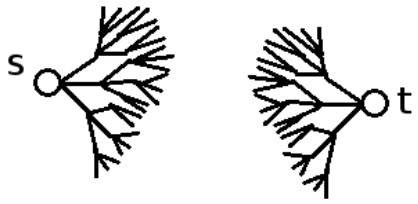
- Speedup Dijkstra's Algorithm

Dijkstra's Algorithm

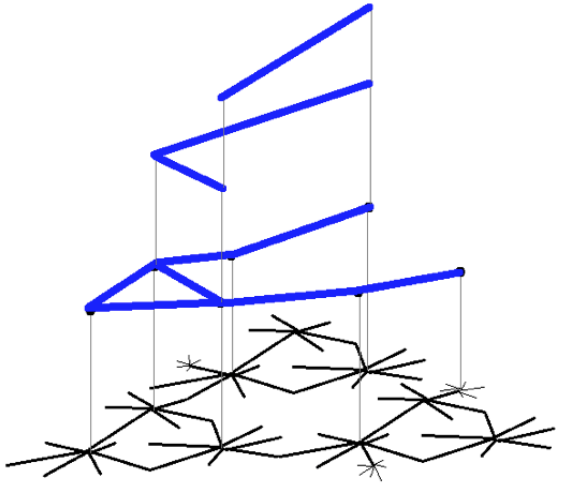


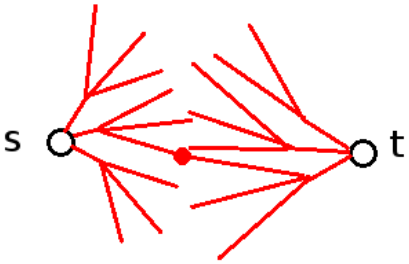
Bidirectional Dijkstra's Algorithm





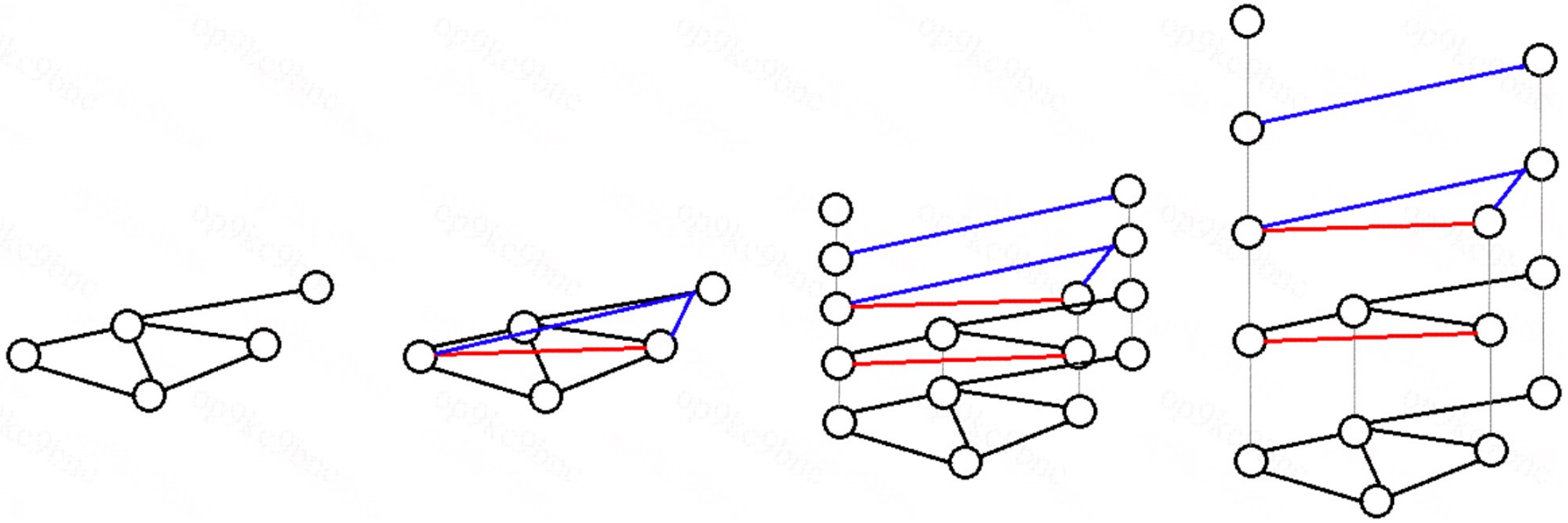
Graph Contraction





Fast Shortest Path

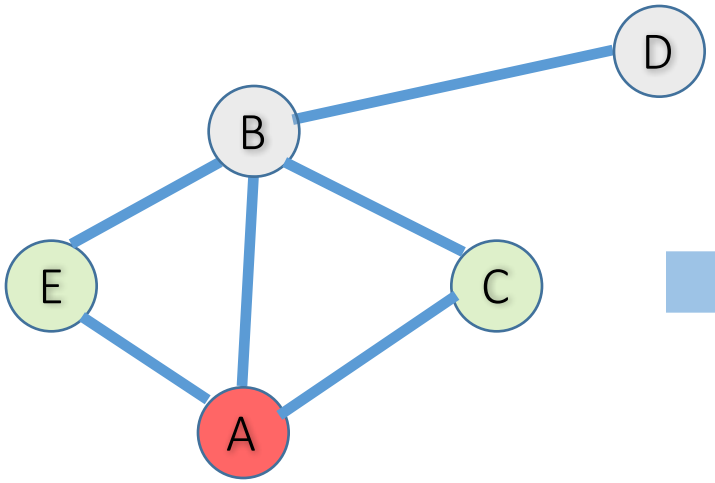
- Contraction Hierarchies (CH)
  - Preprocessing: sorting the nodes, adding shortcut, layer-wise contraction
  - Query: bidirectional Dijkstra's algorithm, unfolding the shortcut

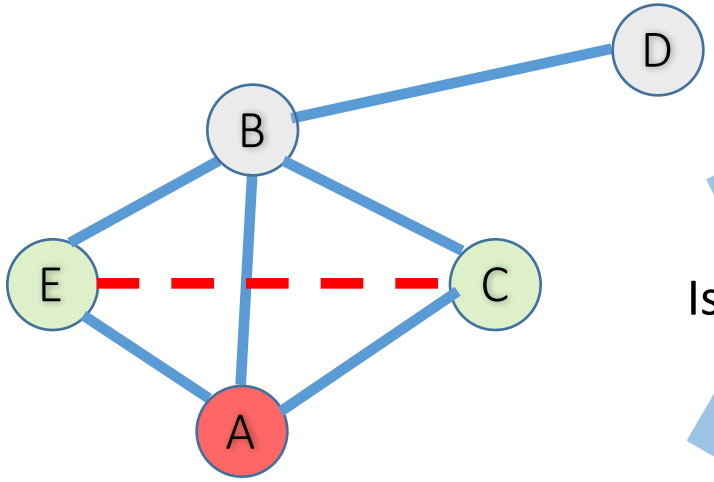


Contraction Hierarchies method is a technique to speed up shortest path routing by first creating precomputed "contracted" versions of the connection graph. It can be regarded as a special case of "highway-node routing".

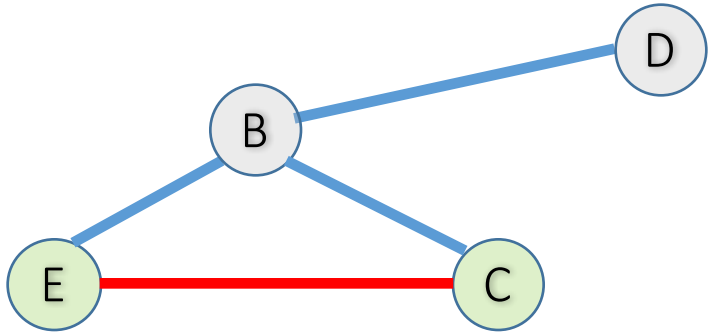
Component Reduction

- Remove nodes and edges w/o shortcut.



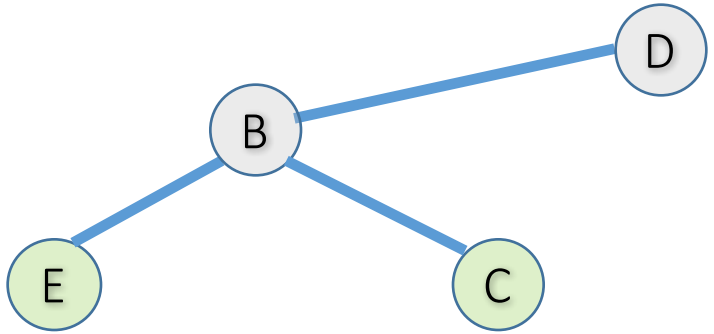


Y

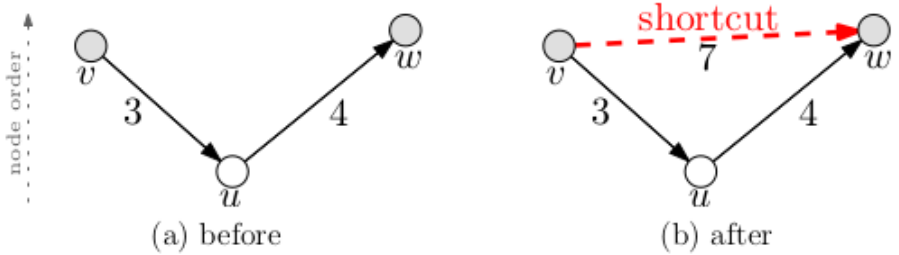


Is A in the shortest path?

N

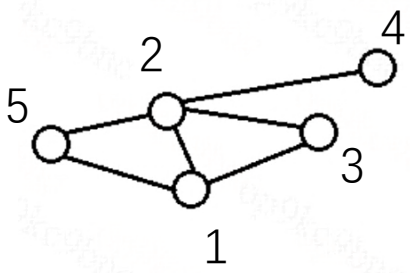


shortcut

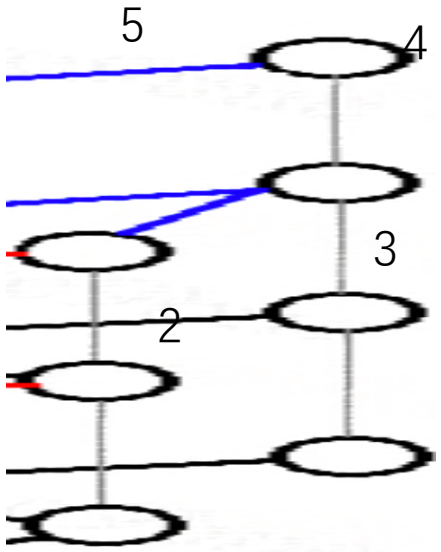


Order for Contraction

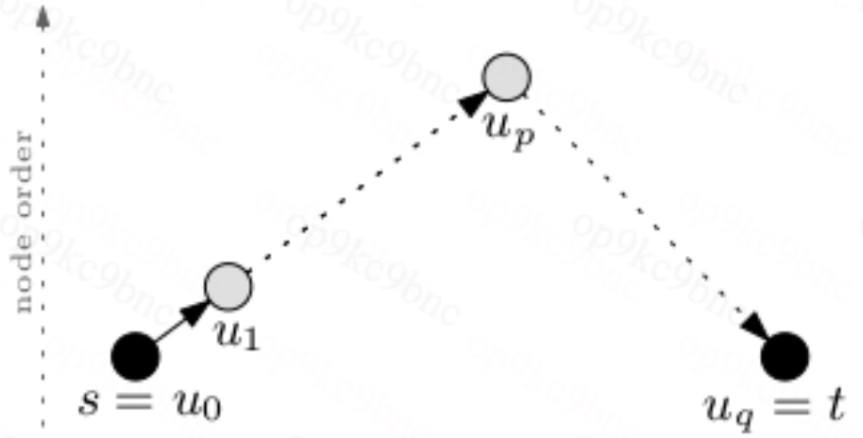
- Setting the node order for contraction: edge difference, cost of contraction, uniformity, cost of queries …
- Order for contration and shortest path search.



Node Order

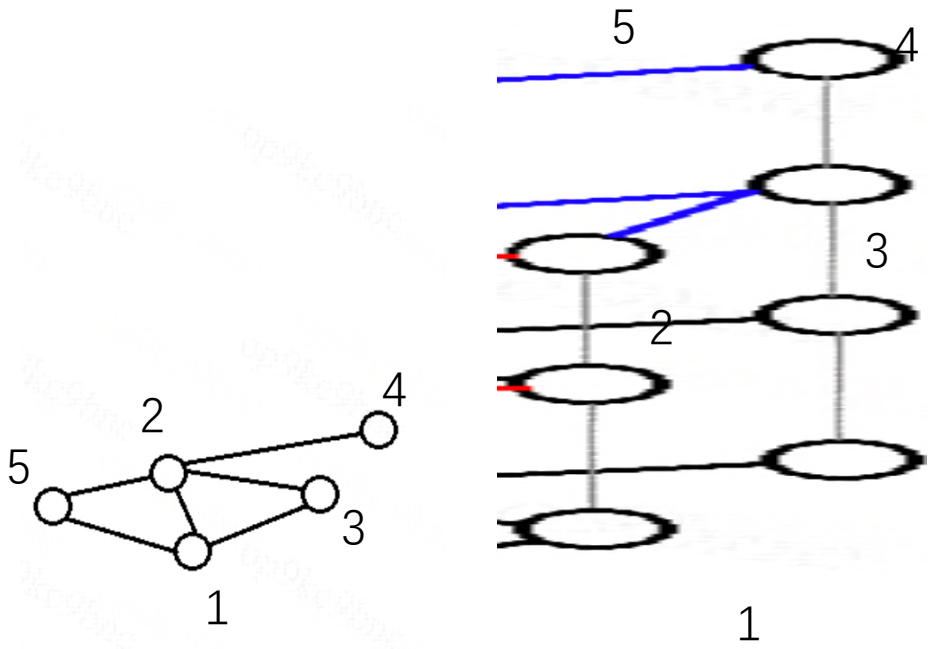


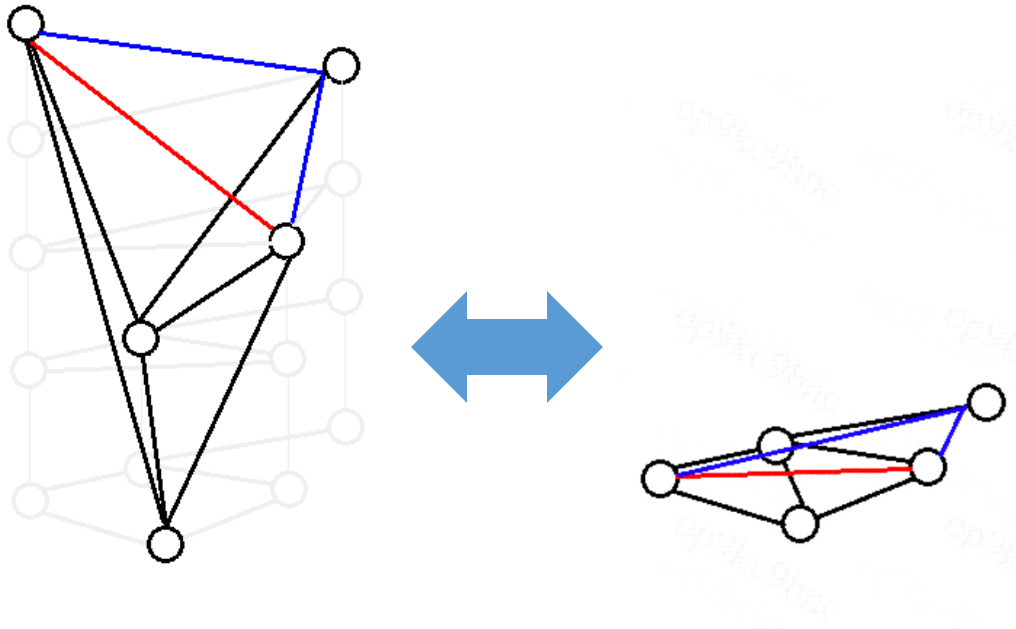
1



Graph Contraction

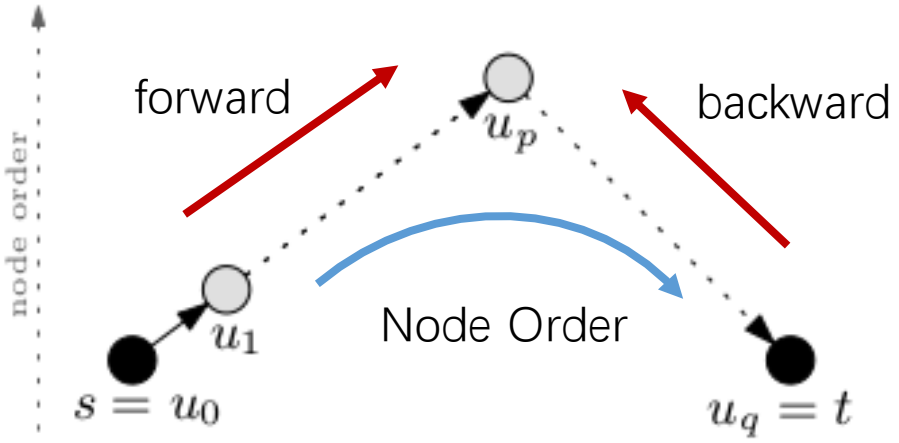
- Multi-level contraction

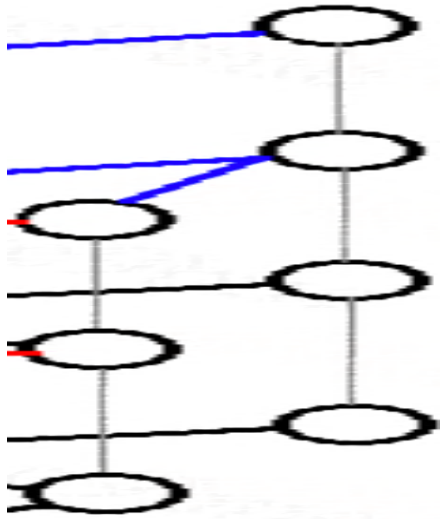




Bidirectional Search

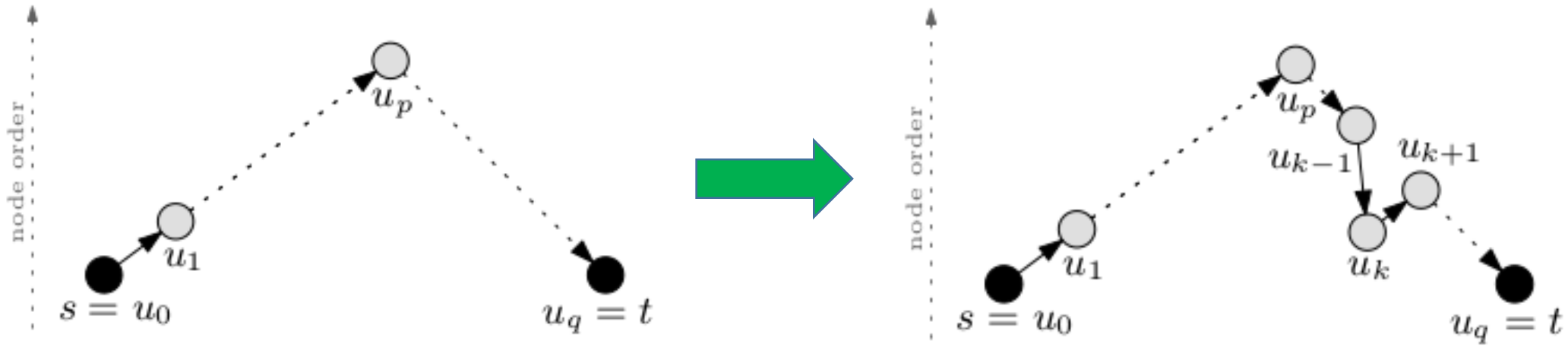
- Bidirectional Dijkstra's algorithm: forward search goes from low order node to high order node, and vice versa.
- Unpack the shortest path.





Guarantee

- Proved by contradiction

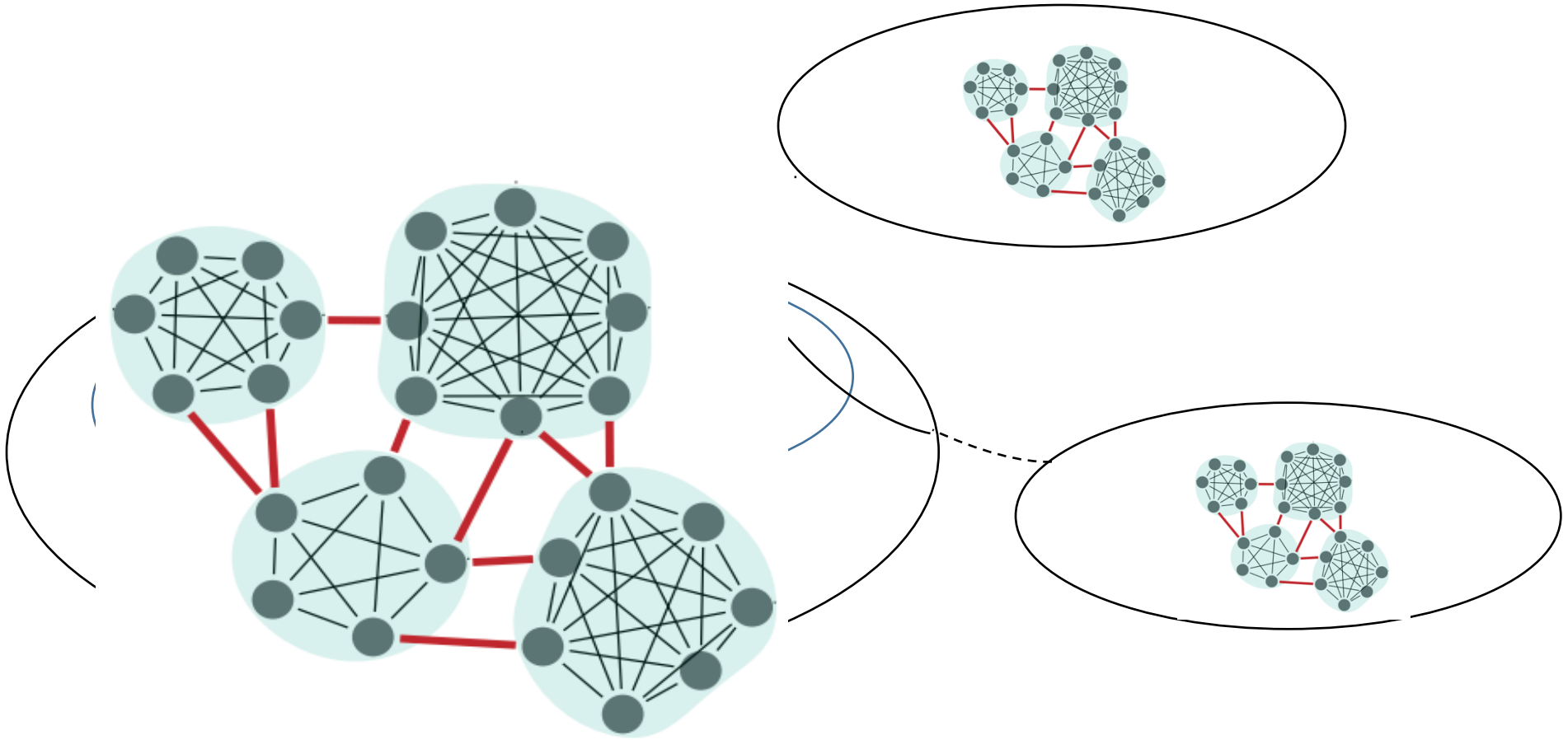


Multiple Metric

- CRP: customizable route planning
  - Metric independent processing (partition)
  - Metric customization
  - Query

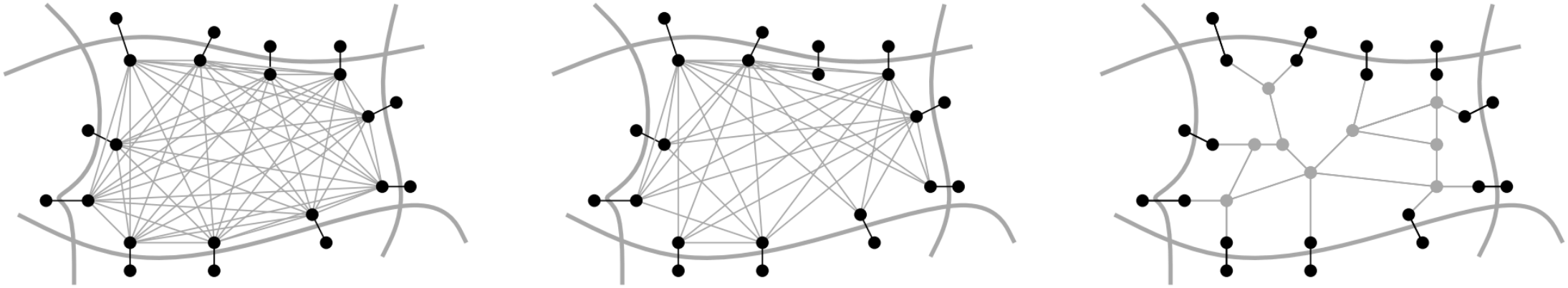
Partition

- Multi-layer partition on unweighted graph
- Overlay graph: distance preserving subgraph



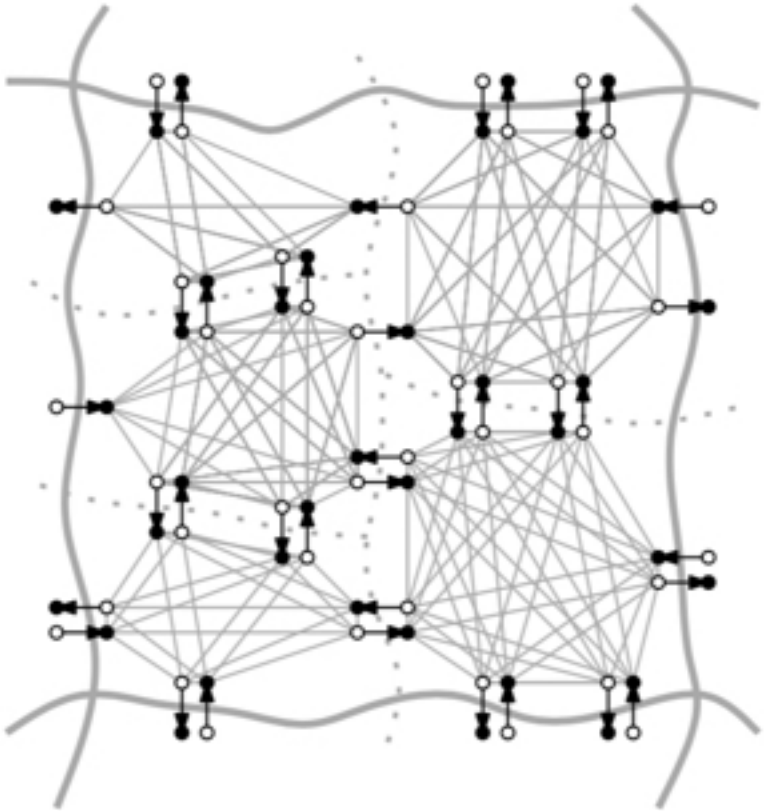
Overlay Graph

- Three possible ways of preserving distances within the overlay graph: full clique, arc reduction and skeleton.

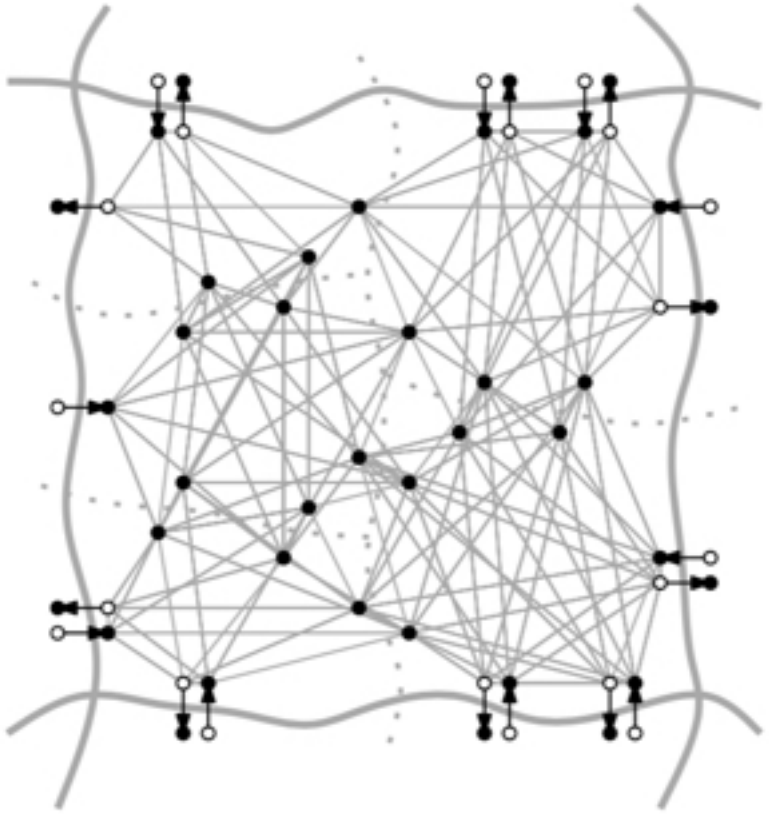


* Picture from Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science 2017.

Customization



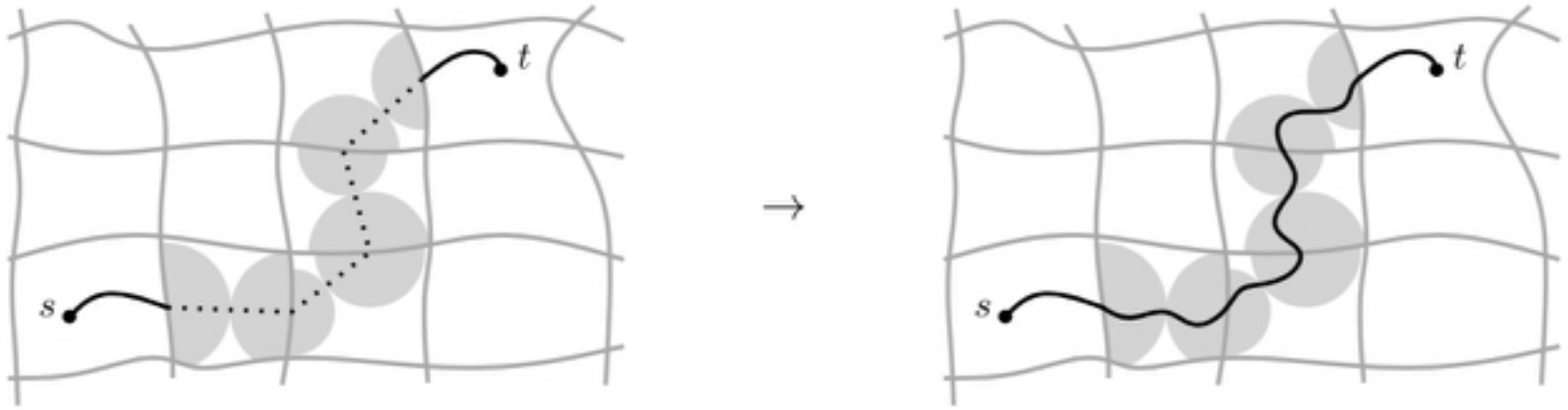
Pruning



* Picture from Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science 2017.

Query

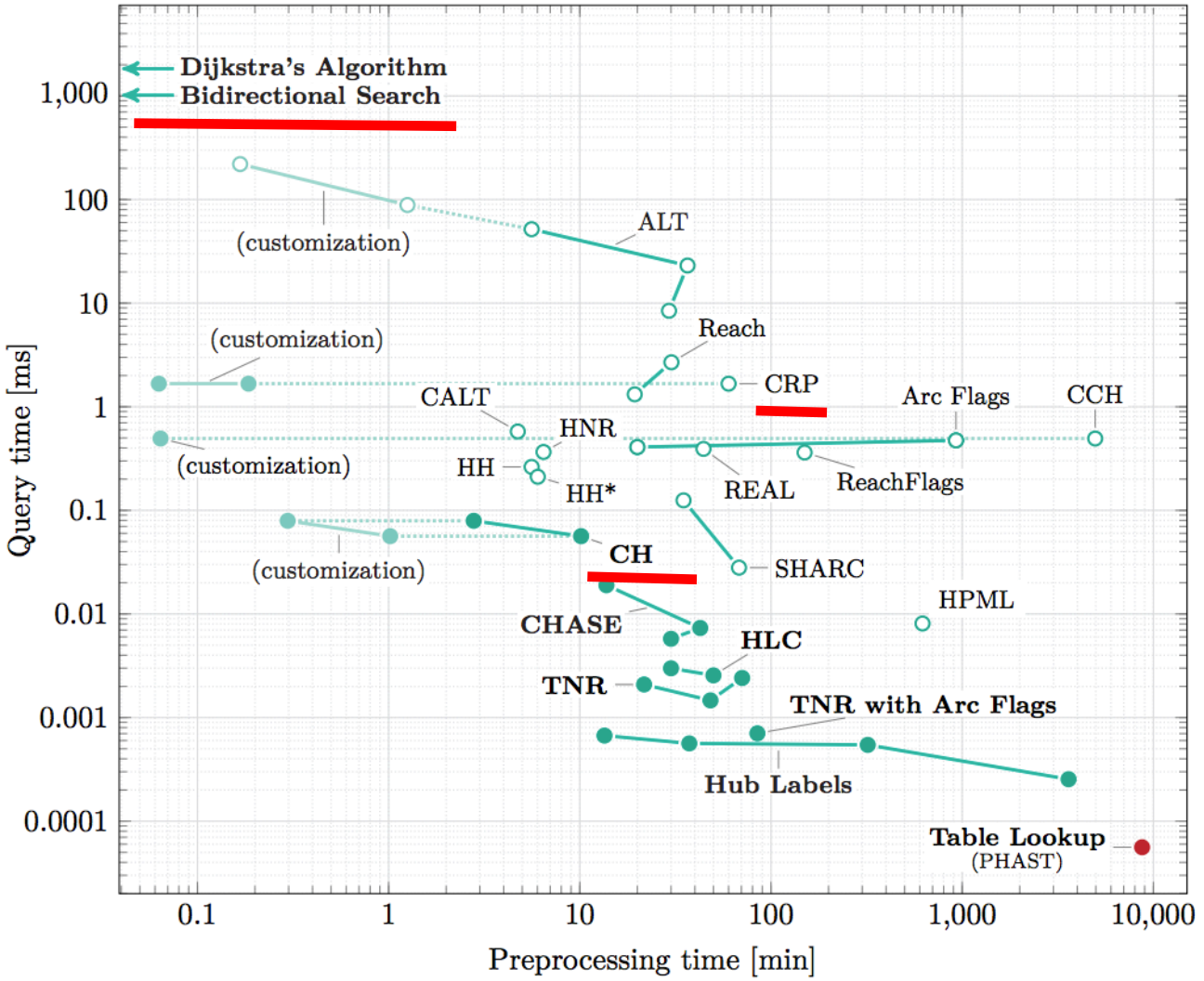
- Shortest path in a subgraph: overlay graph + subgraph including OD nodes.
- Unpack the shortest path: Dijkstra's Algorithm for each clique.



* Picture from Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science 2017.

Summary of Shortest Path Search

Results on road network of West Europe, using travel times as edge weights. (18M nodes and 42.5M edges)



* Hannah Bast et al. *Route Planning in Transportation Networks.* Arxiv 2015.

Challenges

Travel Distance

Travel Speed

Traffic condition

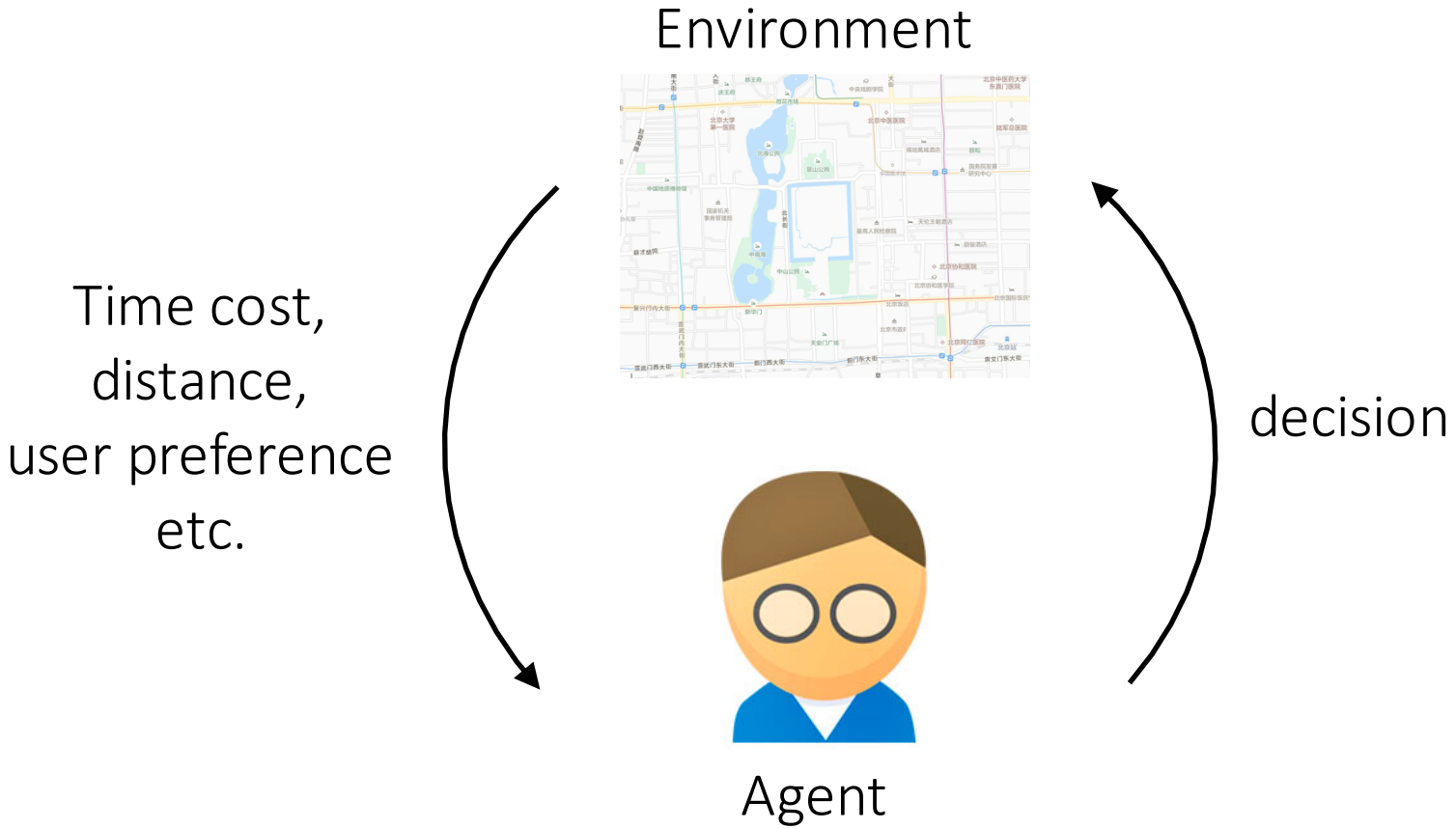
Driver/Passenger Preference

Data Driven vs Heuristic

It is very difficult to preset proper penalty weight to represent all those factors

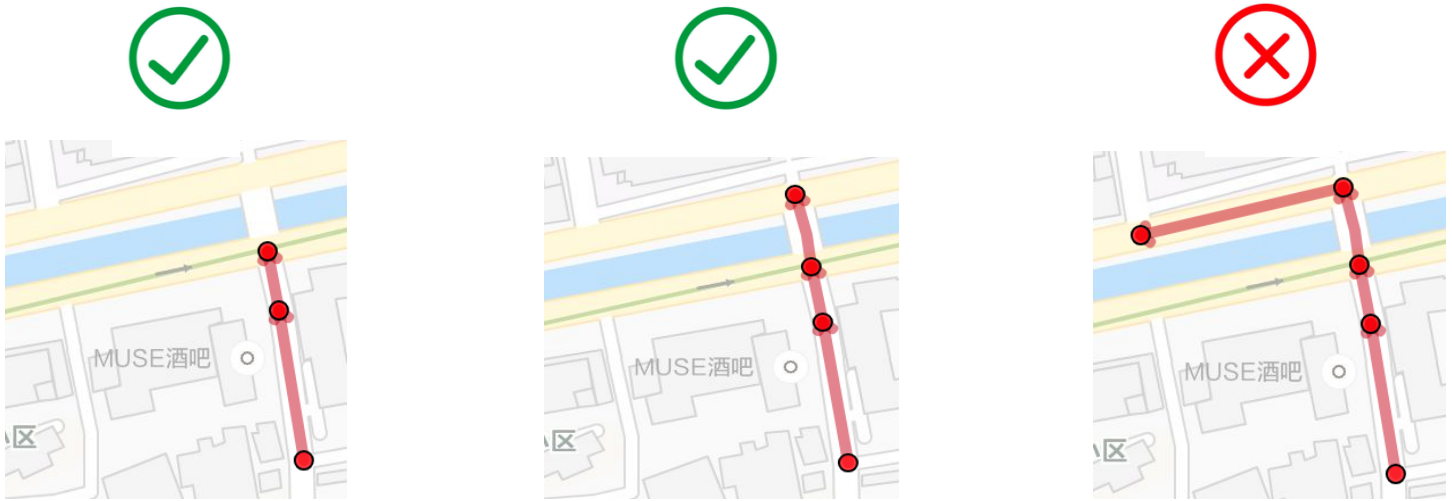
Can we learn human driving patterns?

Data Driven: decision process



Approximately, Route Planning problem can be seen as a deterministic MDP problem. Our goal is to maximize/minimize customer's reward/cost

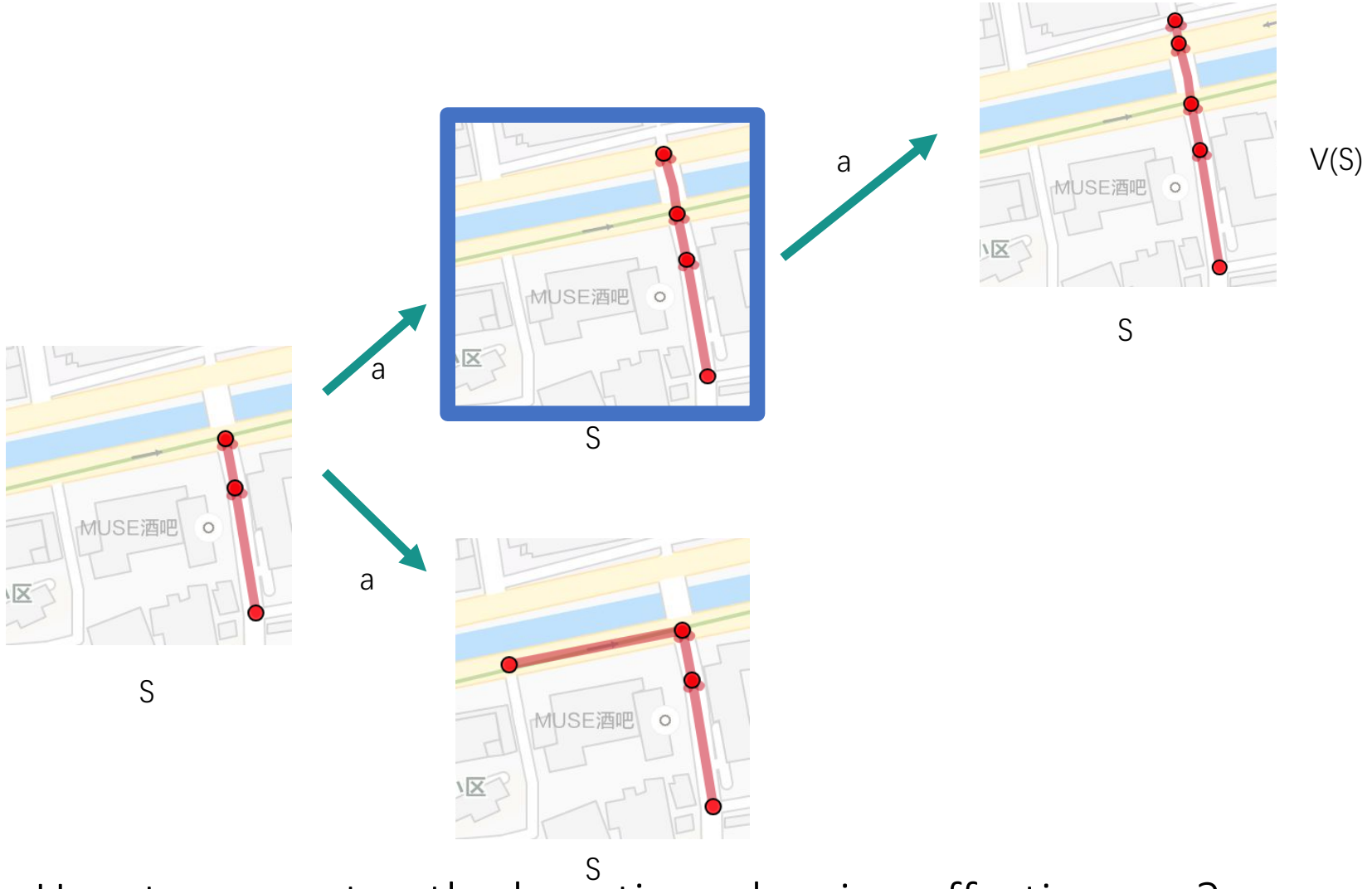
Data Driven



Classification / Reinforcement learning (driving policy) problem

Data Driven

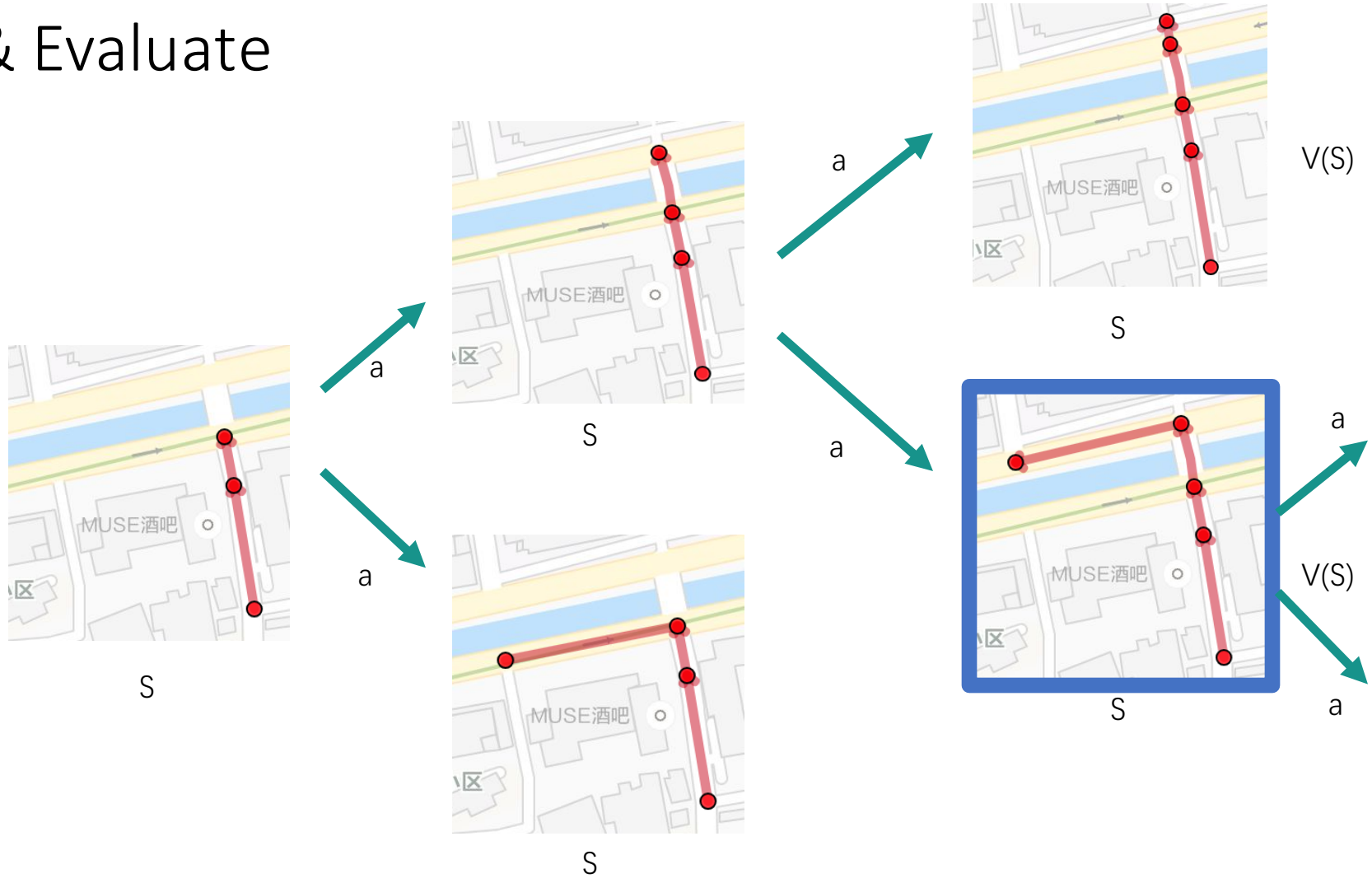
1. Selection



How to guarantee the long time planning effectiveness?

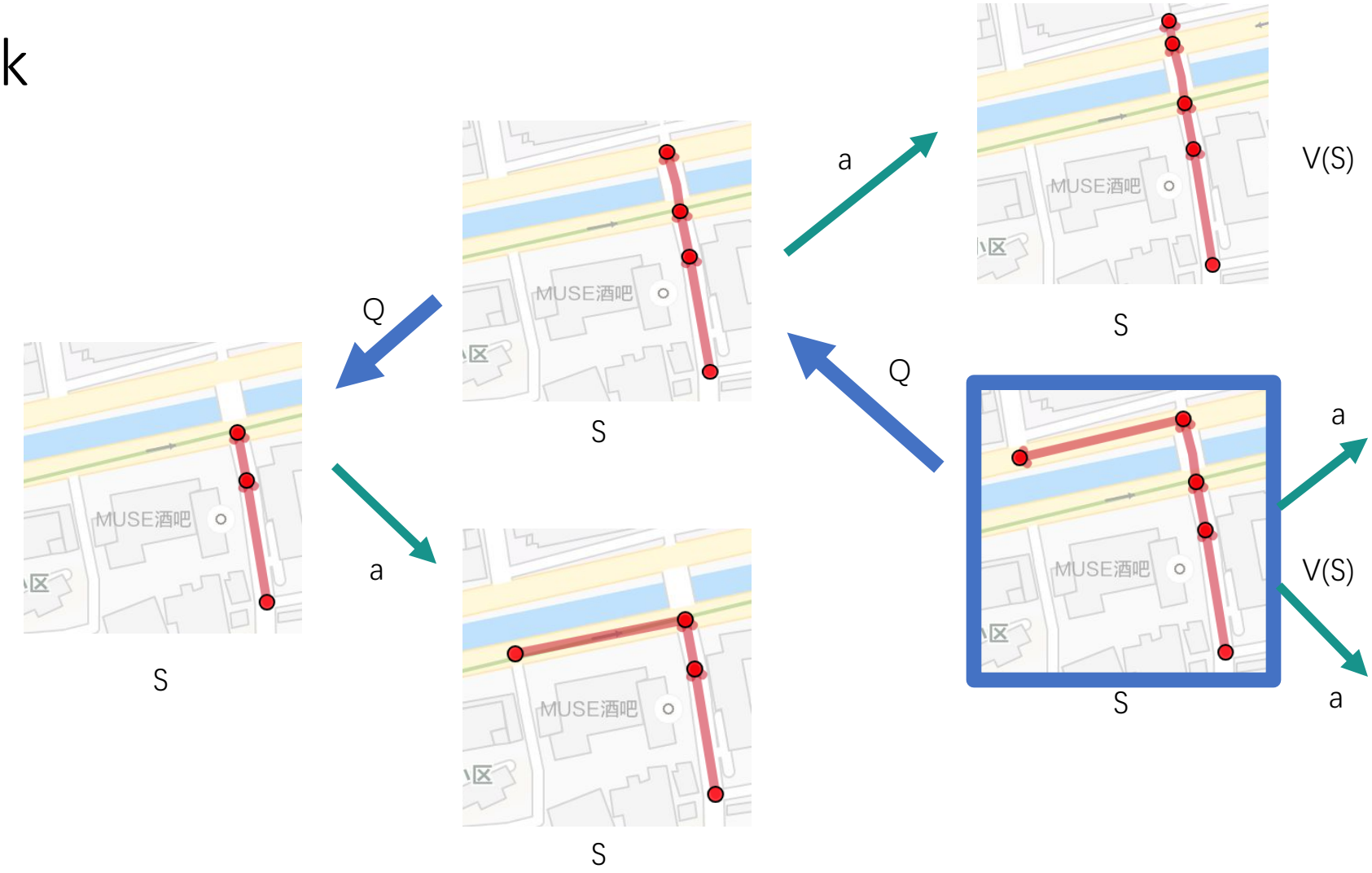
Data Driven

2. Expand & Evaluate



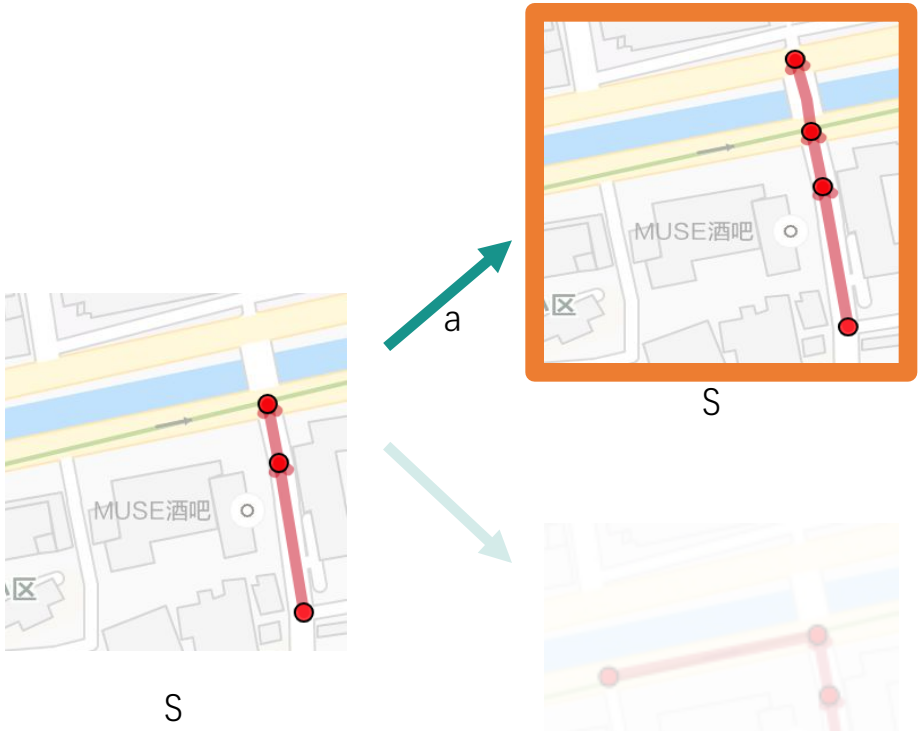
Data Driven

3. Backtrack



Data Driven

4. Execute



$$L = \sum_t (v(s_t; w) - z_t)^2 - \tilde{\pi}_t log\pi(a|s_t; \theta)$$

$$+c||\theta||^2 + c||w||^2$$

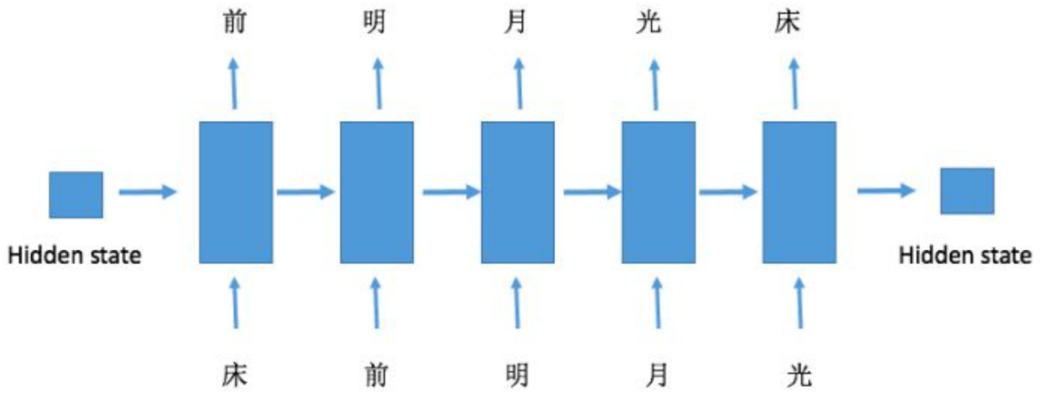
Learning to Plan the Route

- Modeling Trajectories with Recurrent Neural Networks
  - H. Wu, Z. Chen, W. Sun, et al. Modeling Trajectories with Recurrent Neural Networks. IJCAI 2017.

- Route Planning with Reinforcement Learning
  - T. Weber, S. Racanière, D. P. Reichert, et al. Imagination-Augmented Agents for Deep Reinforcement Learning. NIPS 2017.

Modeling Trajectories with Recurrent Neural Networks

- RNN: capture variable length sequence
- Similarity & difference in language/trajectory modeling using RNN
  - Similar: generate words/edges step by step, depending on the present and past words/edges.
  - Different: the transition from one word to any other word is free, while only the transition from one edge to its adjacent edges is possible.



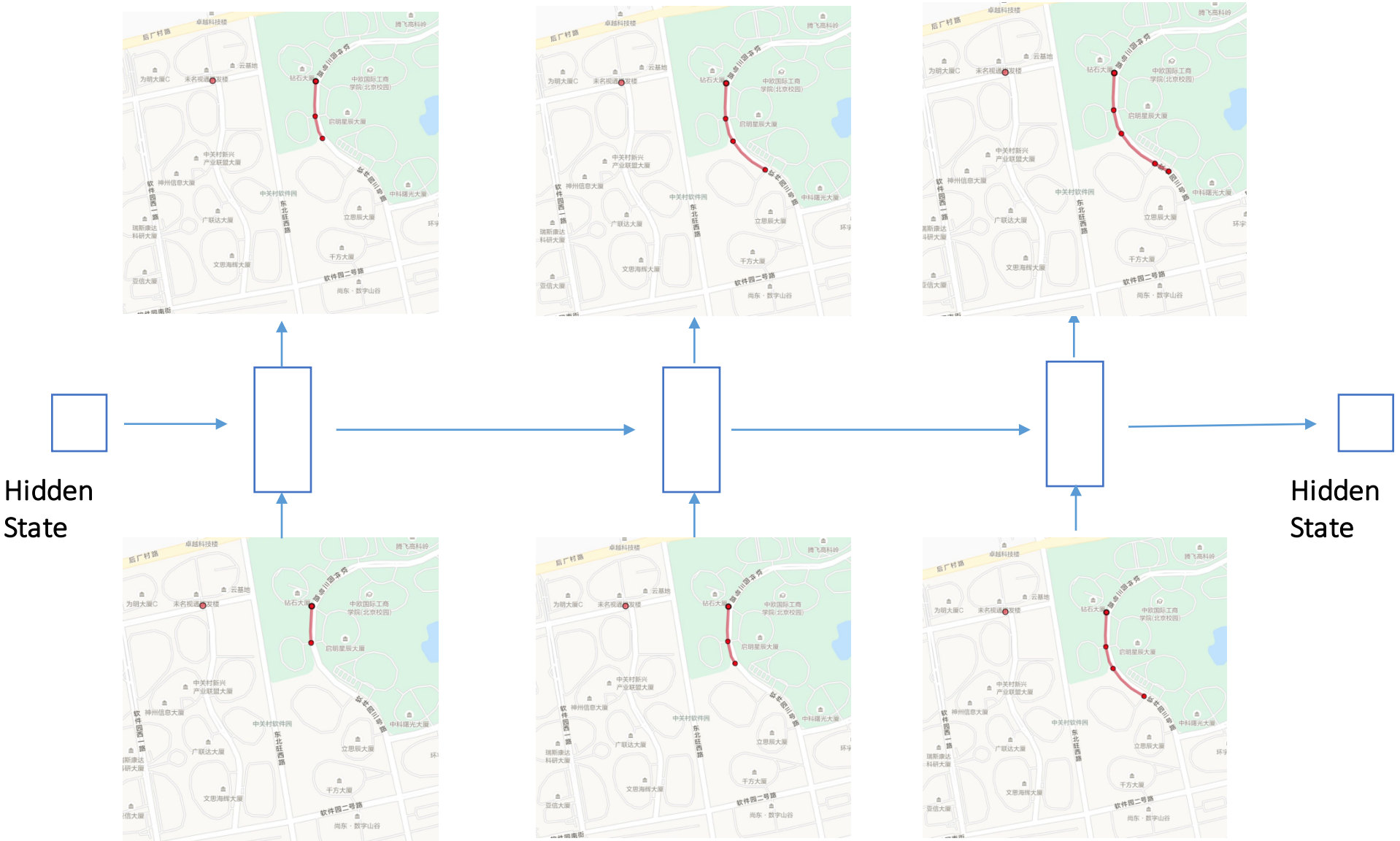
Modeling Trajectories with Recurrent Neural Networks

- Goal: use RNN to learn the topological constraints.
- Solution: modification of state-constrained softmax function.

$$p(\tilde{r}_{t+1}|r_{1:t}) = \mathcal{C}(Wh_t + b, r_t) = \frac{\exp(Wh_t + b) \odot \mathcal{M}_i}{||\exp(Wh_t + b) \odot \mathcal{M}_i||_1}$$

$$\mathcal{M}_{ij} = \begin{cases} 1 & \text{if } r_i \text{ can reach } r_j \\ 0 & \text{otherwise} \end{cases}$$

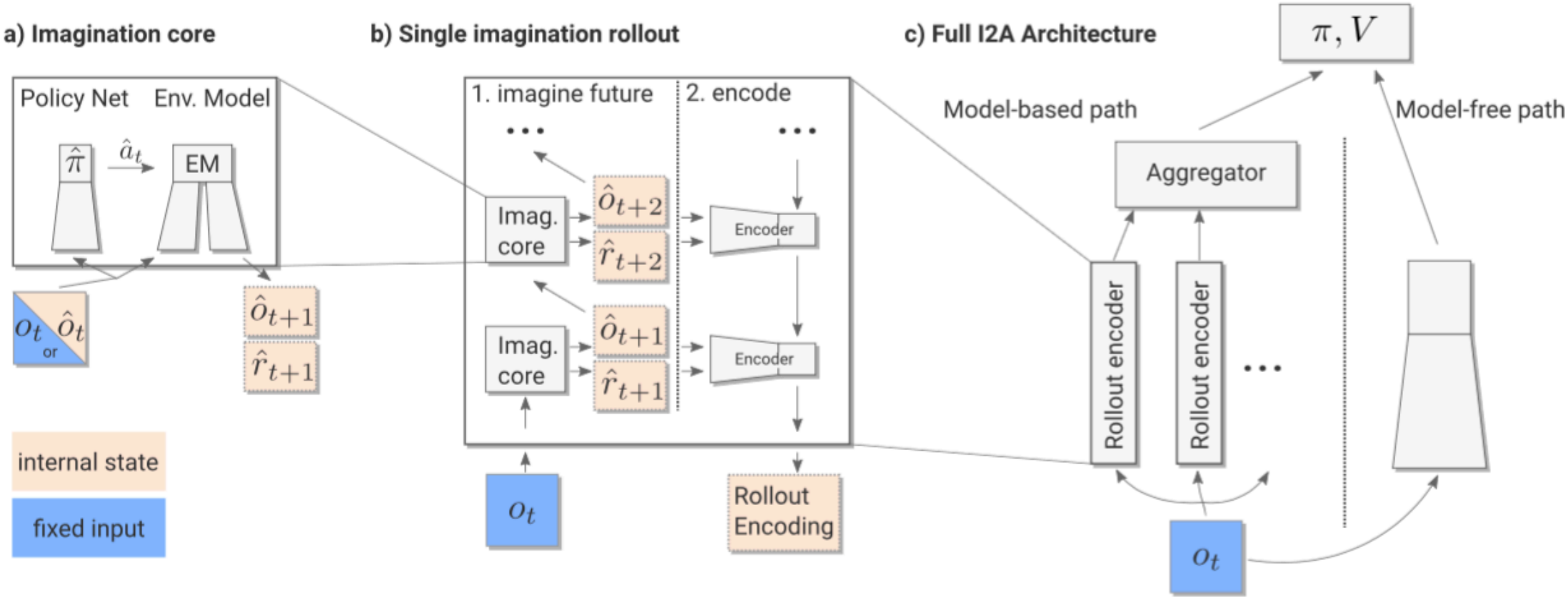
Modeling Trajectories with Recurrent Neural Networks



Modeling Trajectories with Recurrent Neural Networks

Task	With Destination							
Dateset	PT_{small}		PT_{large}		SH_{small}		SH_{large}	
Metric	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC
Bi-gram	6.20	94.15%	8.91	92.30%	7.40	88.54%	7.38	89.11%
Tri-gram	6.20	93.88%	8.92	91.99%	7.34	87.81%	7.29	88.60%
4-gram	6.21	93.57%	8.93	91.66%	7.31	87.02%	7.24	88.04%
BIRL	5.84	95.53%	–	–	6.67	91.42%	–	–
MERIL	7.84	93.70%	8.87	93.23%	7.28	91.19%	6.59	92.00%
RNN	3.74	97.13%	5.63	96.65%	5.27	93.58%	5.67	94.42%
CSSRNN	3.21	97.16%	**3.96**	96.89%	**4.21**	94.10%	3.97	**94.9%**
LPIRNN	**3.12**	**97.21%**	3.98	**96.97%**	4.22	**94.15%**	**3.96**	94.88%

Route Planning with Reinforcement Learning



Imagination-Augmented Agent (I2A), which incorporating model-free RL and model-based RL, improves data efficiency, performance, and robustness.

Route Planning with Reinforcement Learning

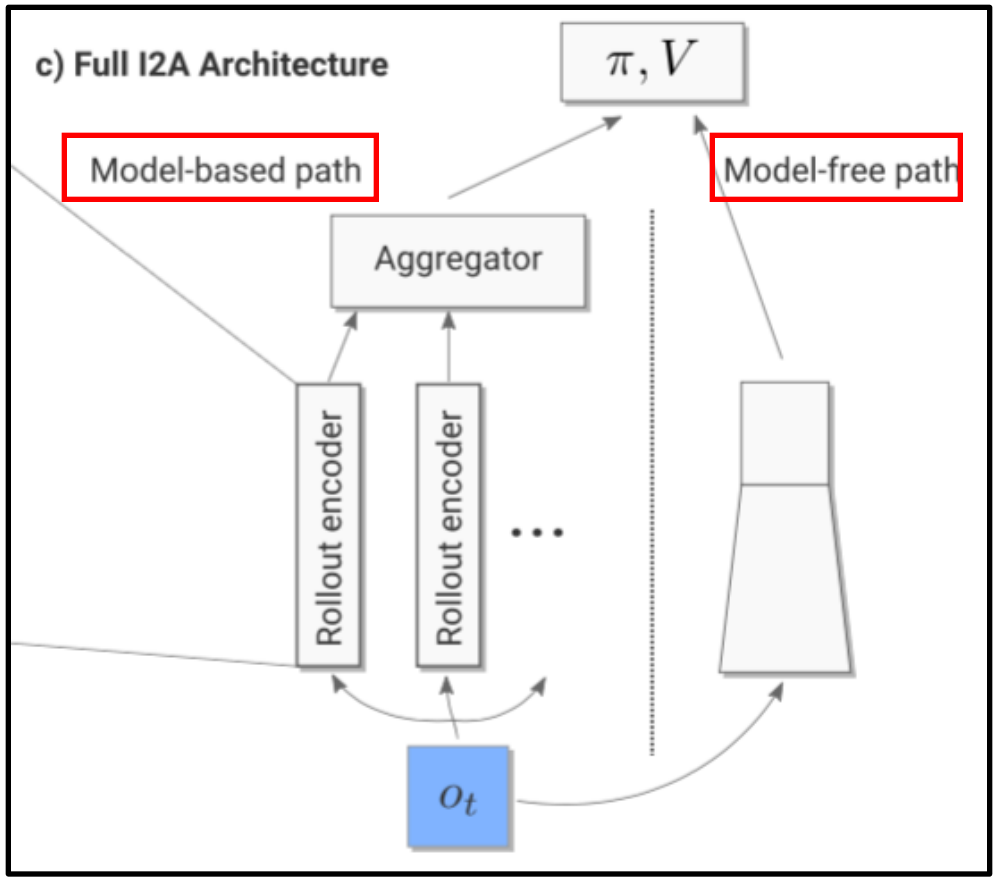
Model-Based Approaches

Advantages

- Endowing agents with a model of the world
- Support generalization to states not previously experienced

Disadvantages

- Complex domains hard to build environment models.
- Performance suffers from model errors resulting.



Observed State

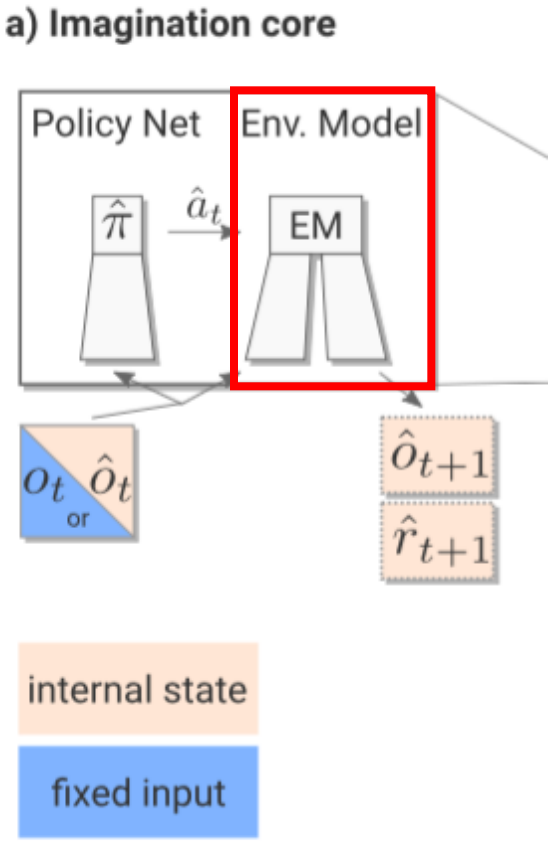
Model-Free Approaches

Disadvantages

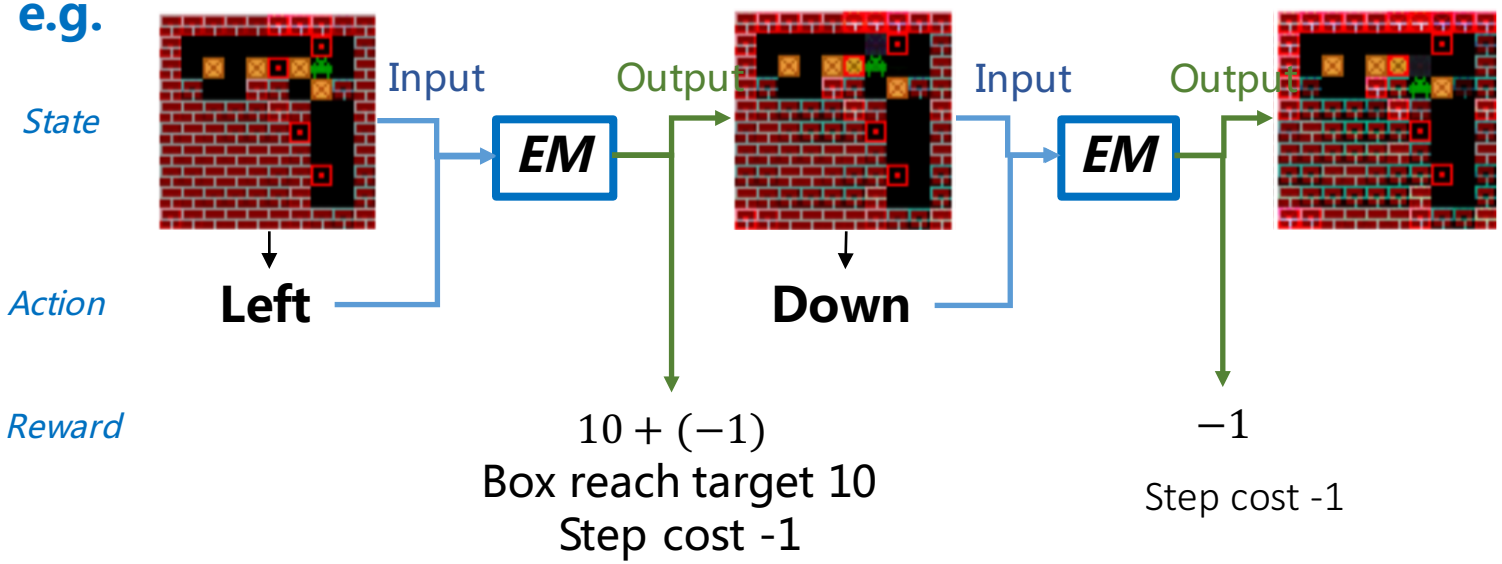
- Requires large amounts of training data
- Resulting policies do not readily generalize to novel tasks in the same environment

Route Planning with Reinforcement Learning

Key of Model-Based method —— Environment Model



Input：Observed State，Chosen Action → Output：Next State，Reward

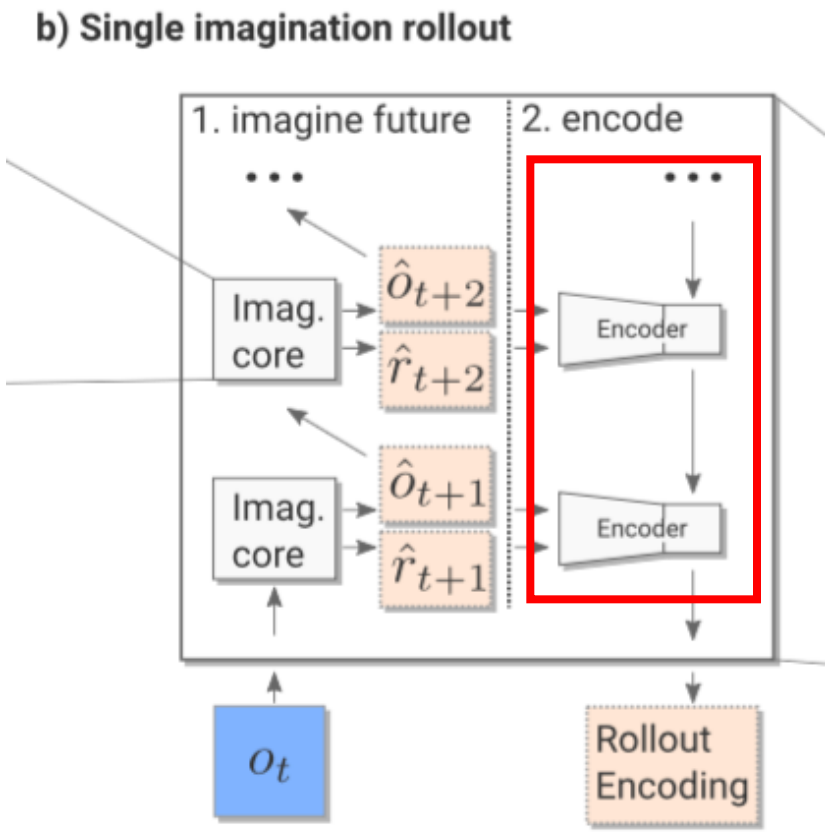


Environment Model Construction Options

- Make assumptions about the structure of the environment model with domain knowledge
- Trained directly on low-level observations with little domain knowledge, similarly to recent model-free successes. (e.g., I2A)
- Perfect World Model

Route Planning with Reinforcement Learning

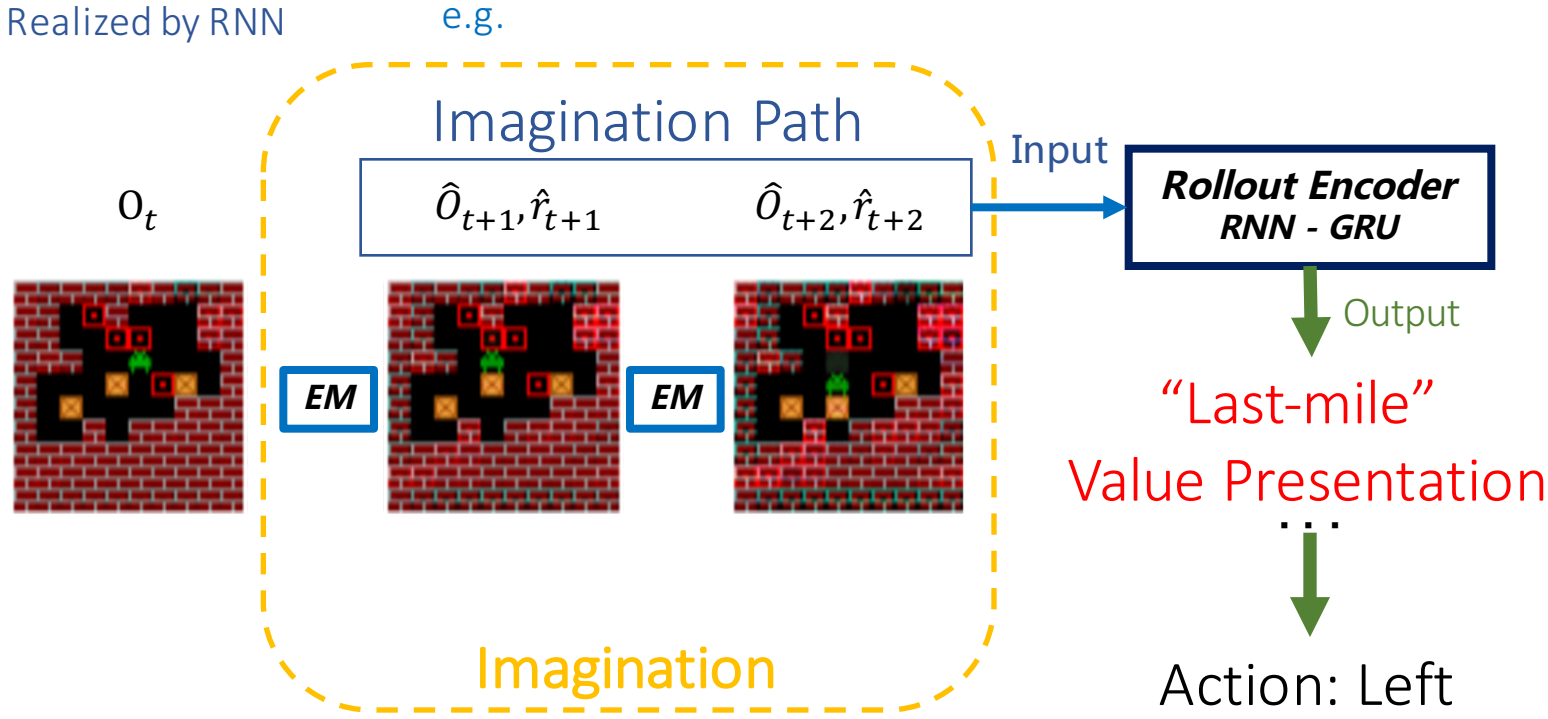
Output Layer of Model-based: Path Encoder



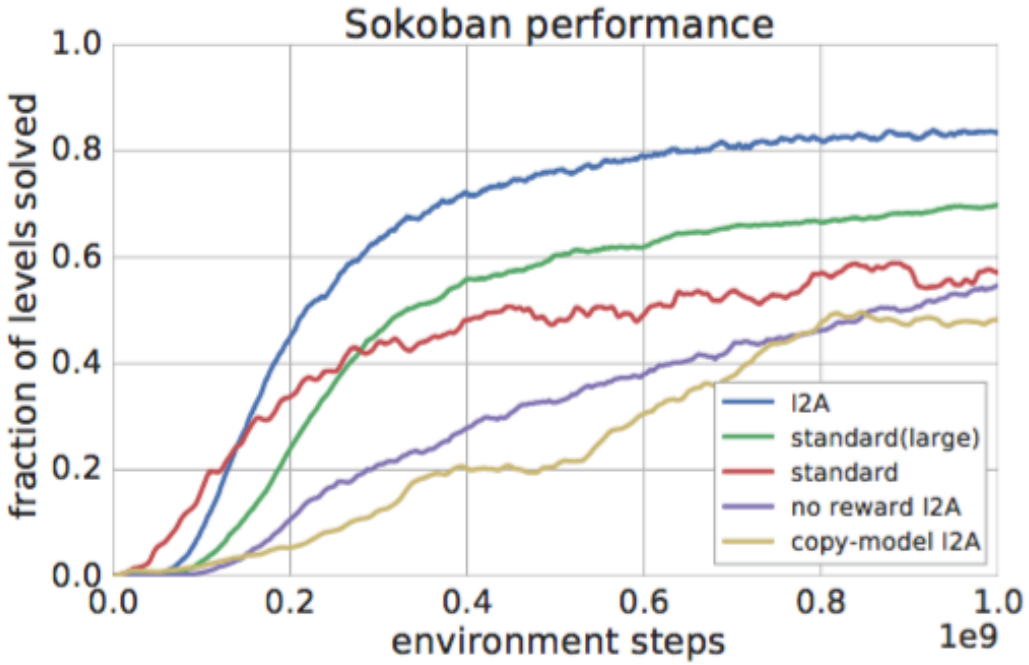
Role：

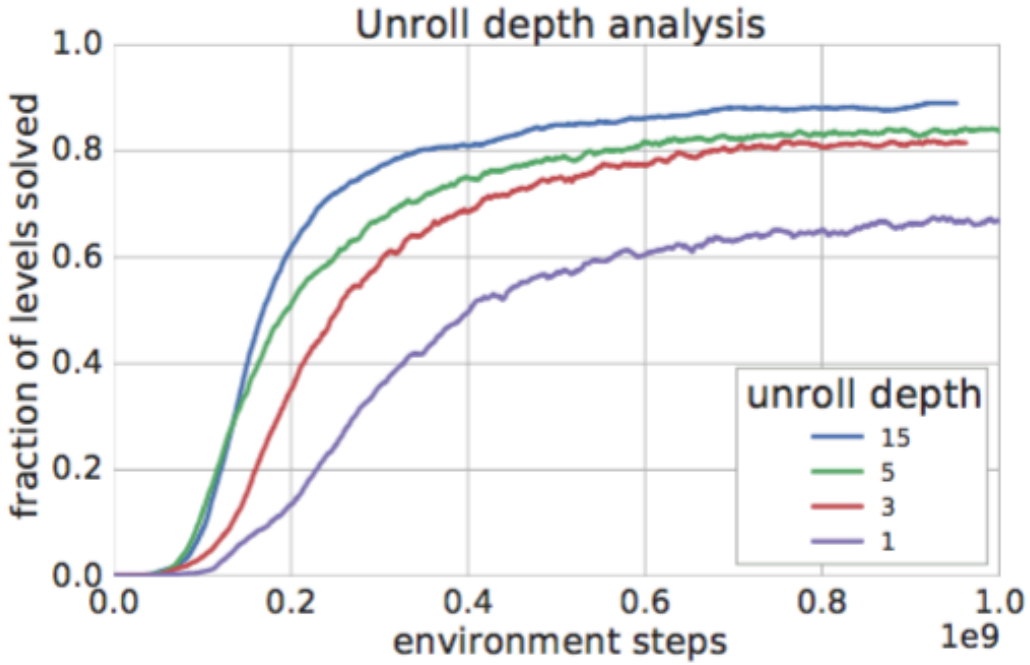
EM generate path after taking a specific action via imagination.

Encoder transform the path into the evaluation of selected action.



Route Planning with Reinforcement Learning

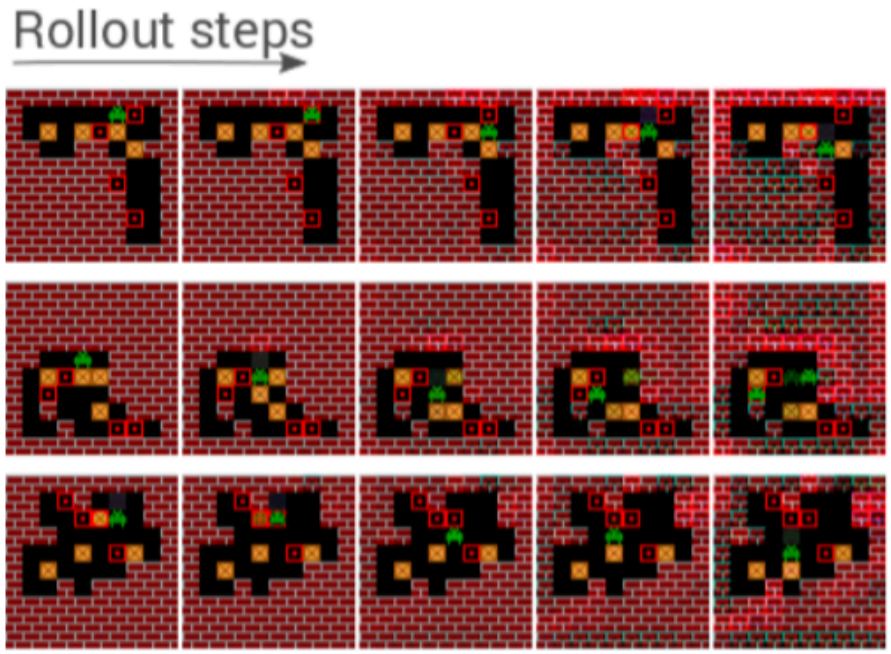


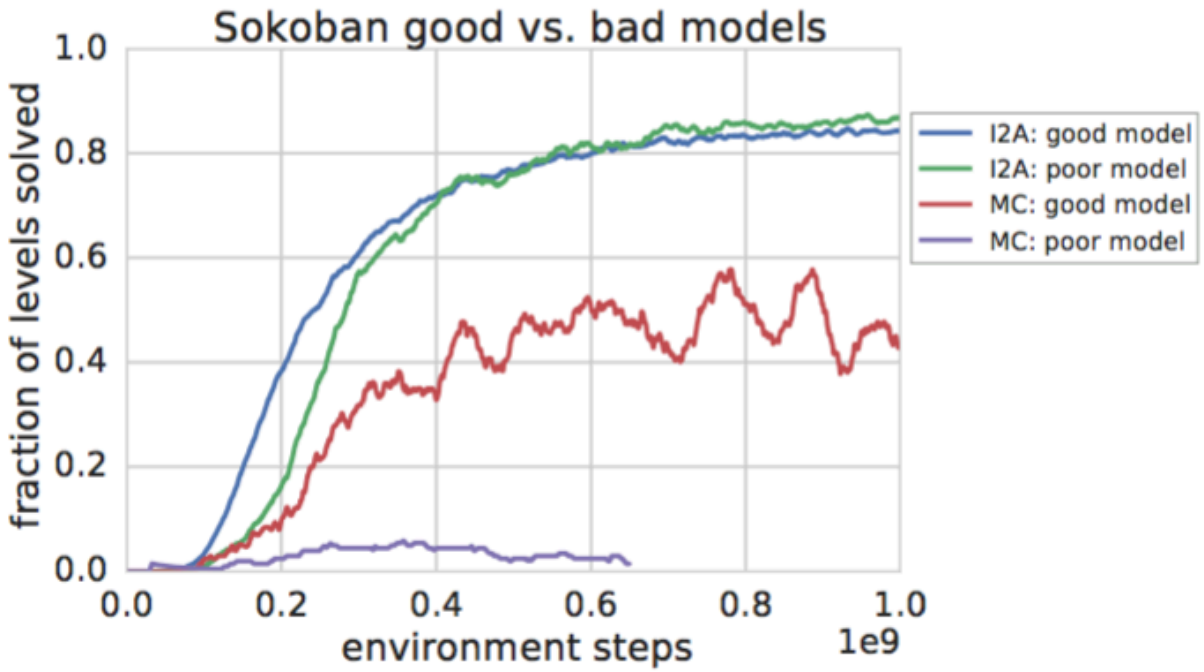


- I2A performs better than others

- Performance ∝ Imagination Steps
- Diminishing returns with more rollout steps

Route Planning with Reinforcement Learning



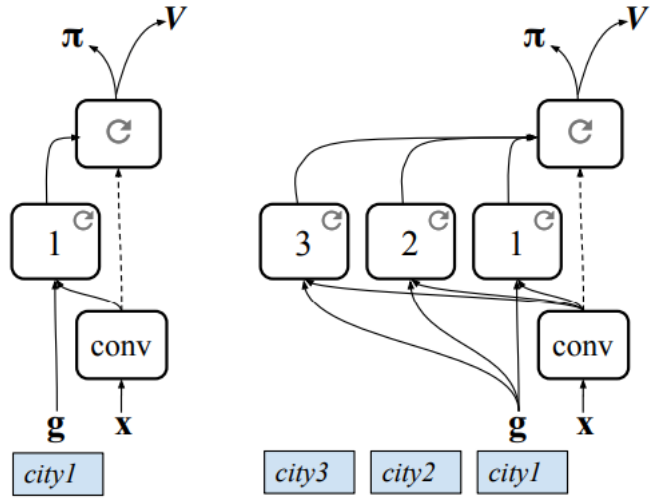


Address the challenge

- RNN Encoder captures the sequential information.
- Work well even with imperfect EM (learn to ignore the latter as errors accumulate)

Alternative Solution

- Learning to navigate in cities without a map, from deep mind.



* Piotr Mirowski, et al. *Learning to navigate in cities without a map*. Arxiv 2018.

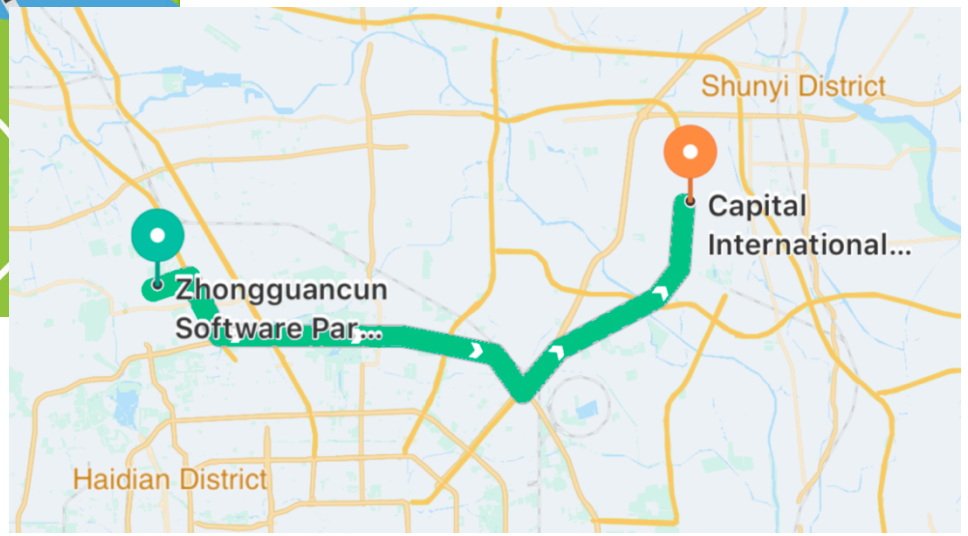
Reference

- R. Geisberger et al. *Contraction hierarchies: Faster and simpler hierarchical routing in road networks*. In 7th Workshop on Experimental Algorithms. LNCS Series, vol. 5038. Springer, pp 319-333, 2008
- Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science. vol. 51(2), pp 395-789, 2017.
- Hannah Bast et al. *Route Planning in Transportation Networks.* Arxiv 2015.
- H Wu et al. *Modeling Trajectories with Recurrent Neural Networks*. IJCAI 2017.
- T. Weber et al. *Imagination-Augmented Agents for Deep Reinforcement Learning*. NIPS 2017.
- Piotr Mirowski et al. *Learning to navigate in cities without a map*. Arxiv 2018.

KDD2018

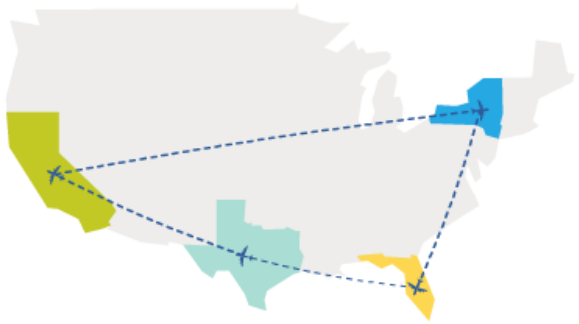
ETA (Estimated Time of Arrival)

ETA (Estimated Time of Arrival)

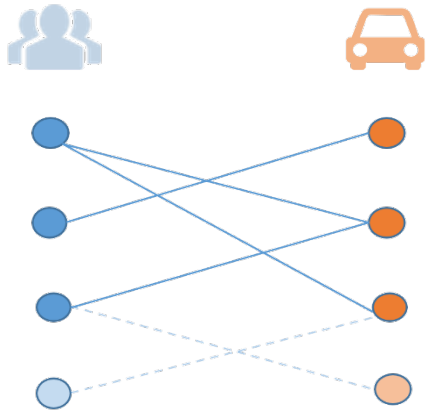


ETA Applications

Route Planning



Order Dispatching



Ride Sharing

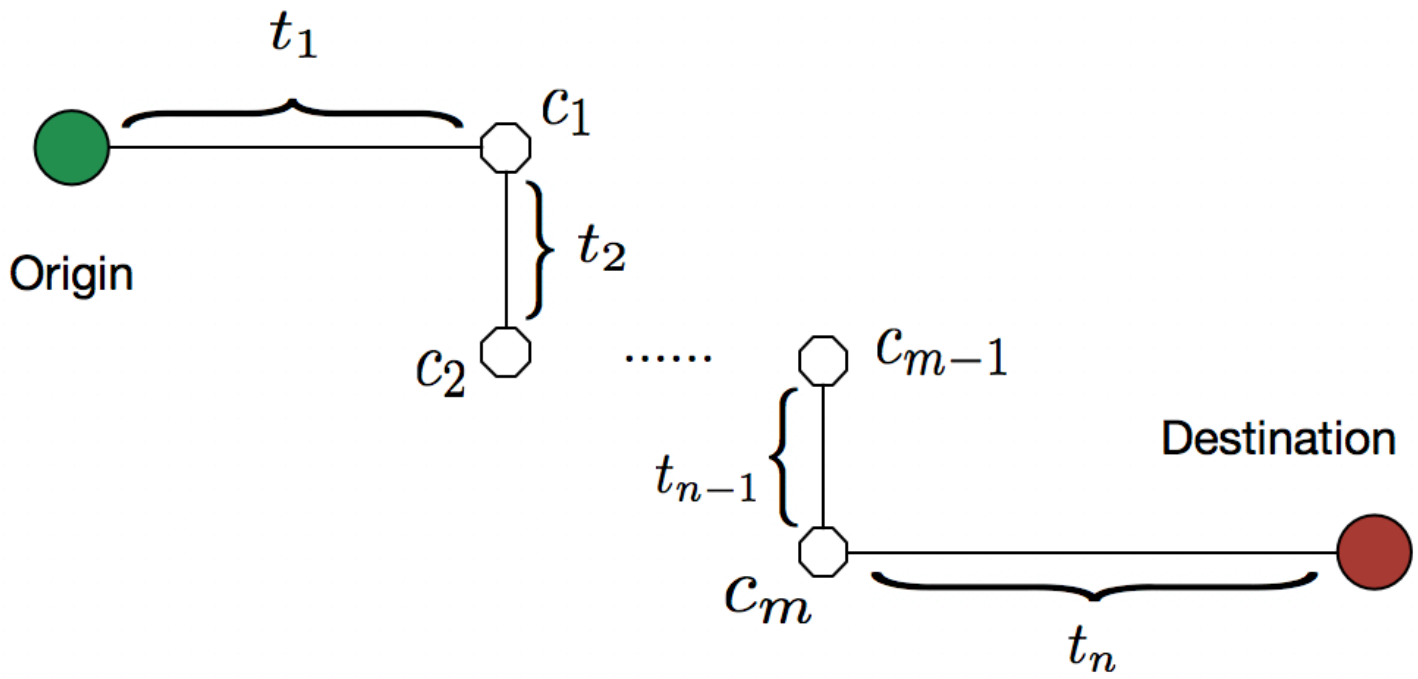
Pricing

Main Stream ETA Models

- Additive models:
  - Rule-based additive models: explicitly modeling the segments in a path
  - Aggregating the time of sub-paths
  - Machine learning models for the sub-path problem.
- Global models:
  - Formulating ETA as a regression problem: Learning to Estimate the Travel Times (L2ETT)
  - Simple regression model and deep learning model
- Path-free models:
  - Path is not available

Simple Additive Model

- Simple rules based on physical structure of road network
- Popular solution in digital map industry
- Challenges: precise speed estimation and error accumulation



$$ETA = \sum_{i=1}^{n} t_i + \sum_{j=1}^{m} c_j$$

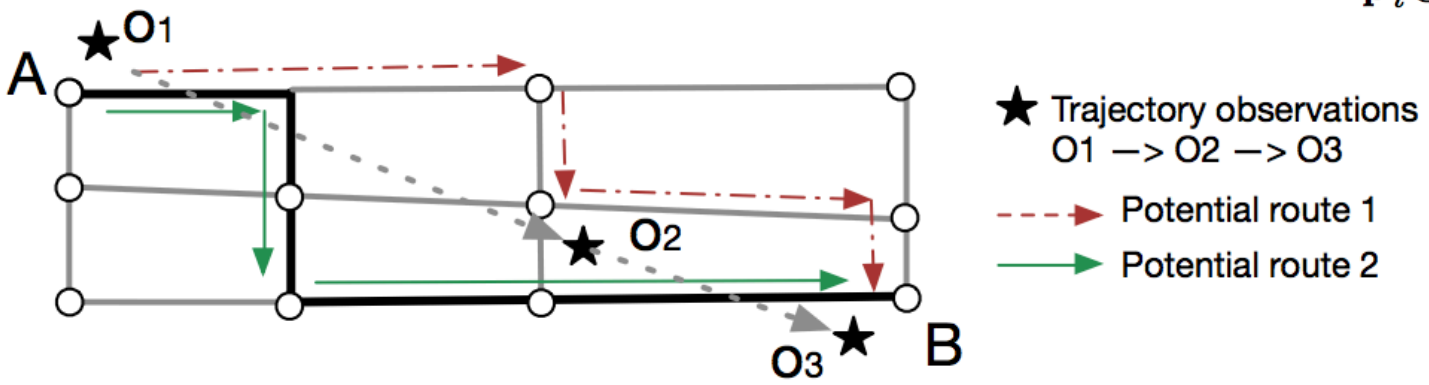
road segments

crossings

Aggregating Sub-Paths

- The path can be decomposed into sub-paths.
- Time of each sub-path is inferenced from the historical trajectories.
- ETA is the summation of the time of sub-paths.

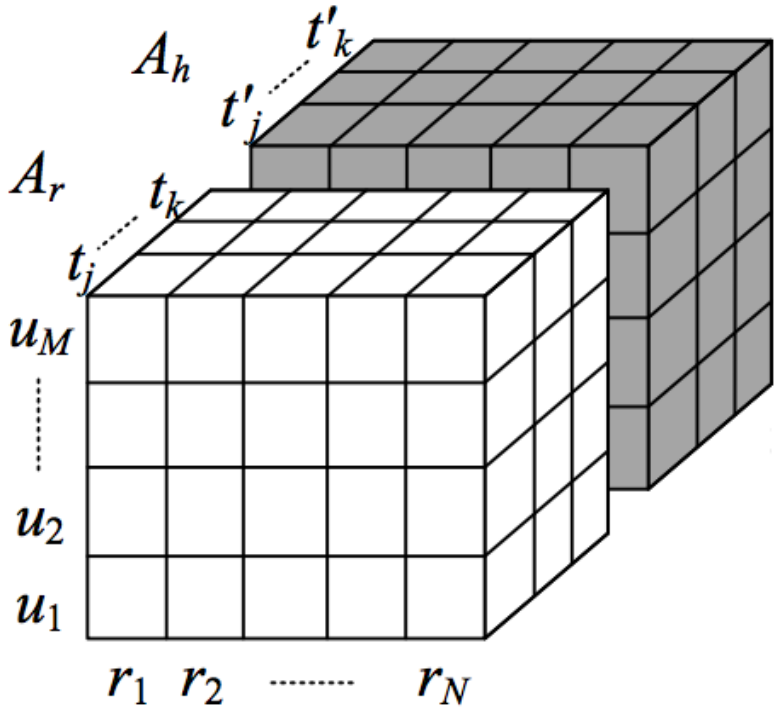
$$\mathcal{N}(\mathbf{q}) = \{\mathbf{p}_i \in \mathcal{D} | dist(o_i, o_q) \leq \tau \text{ and } dist(d_i, d_q) \leq \tau\}, \qquad \widehat{t}_q = \frac{1}{|\mathcal{N}(\mathbf{q})|} \sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{q})} w_i t_i.$$

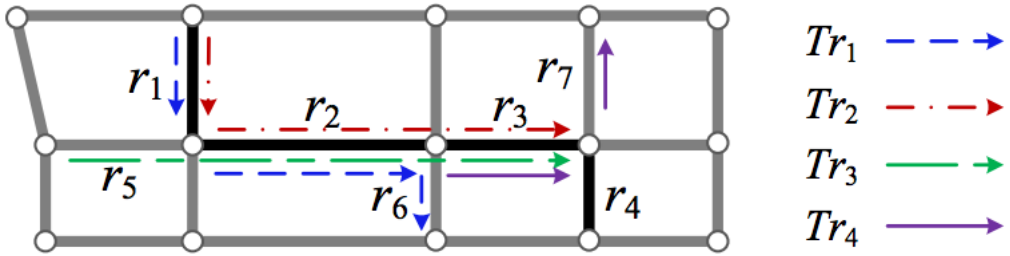


* Hongjian Wang et al. *A simple baseline for travel time estimation using large-scale trip data*. SIGSPATIAL 2016.

Tensor Decomposition

- Build a 3D tensor (driver, road segment, time slot) and use tensor decomposition to estimate the travel time of each segment.
- Use dynamic programing to find the optimal concatenations.

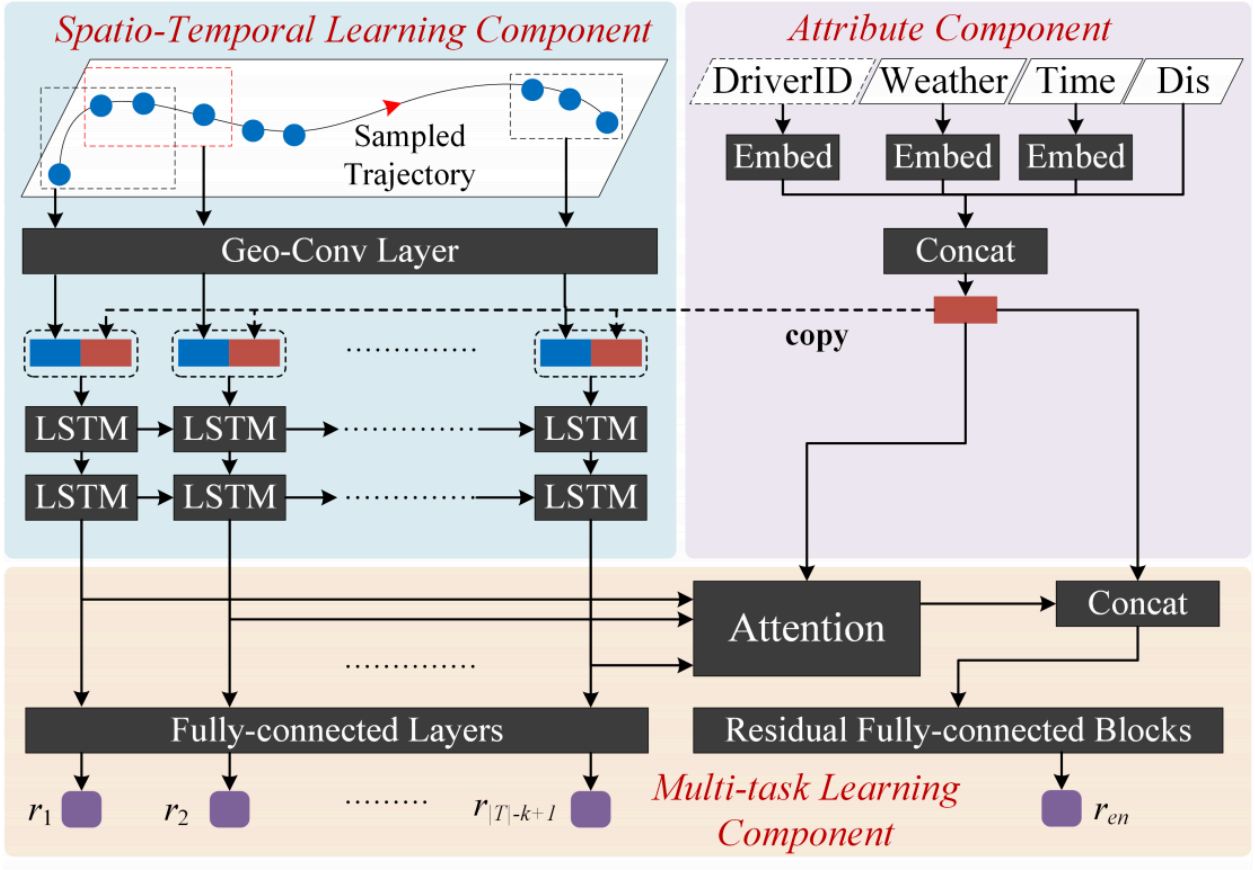




* Yilun Wang et al. Travel time estimation of a path using sparse trajectories. KDD 2014.

Regression Model on GPS Points

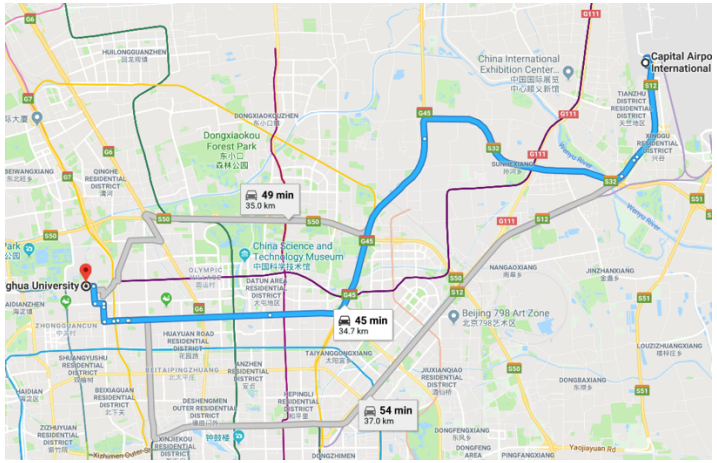
- Deep regression model on GPS data
  - GPS points will not be available before trip in real-world case.



* Dong Wang, et al. *When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks*. AAAI 2018.

Time Series Prediction

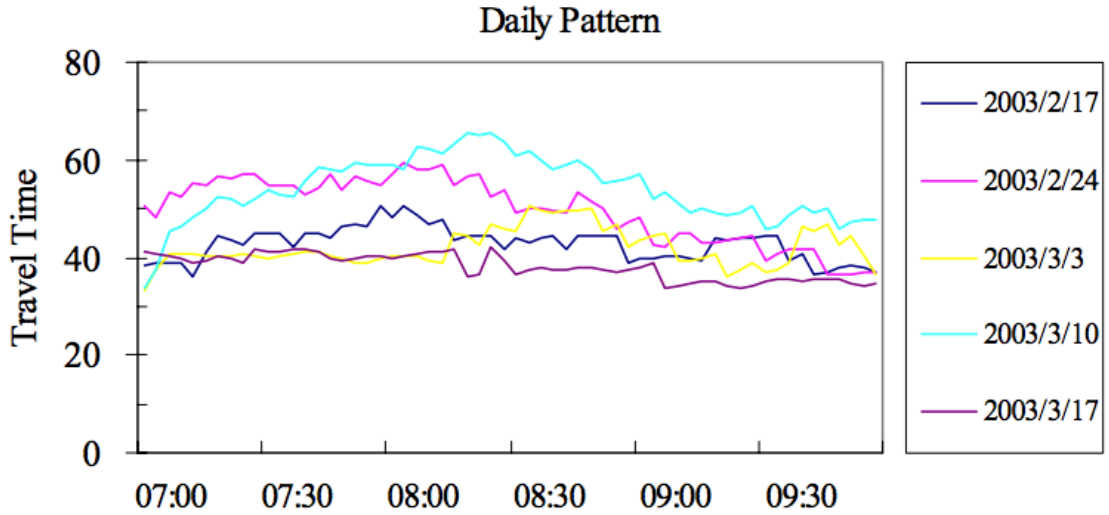
- Conventional regression model: support vector regression
- Deep learning model: recurrent neural network



selected routes

f(t-n),..,f(t-1) → f(t)

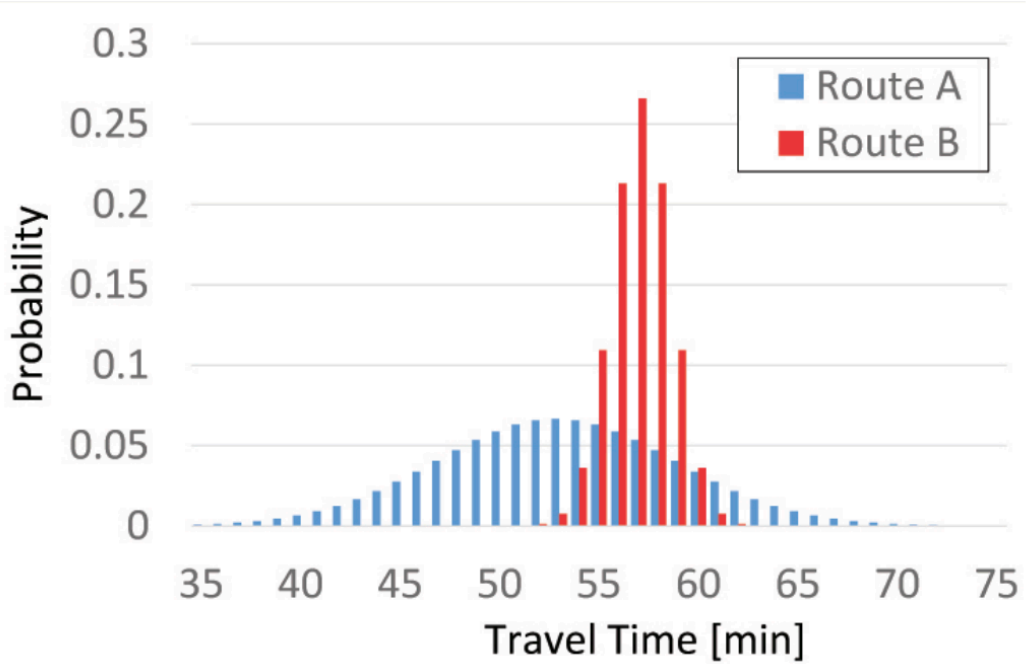
f(t-n),..,f(t-m) → f(t)



* Chun-Hsin et al. *Travel-time prediction with support vector regression*. IEEE Trans. ITS 2004.
* Yanjie Duan et al. *Travel time prediction with LSTM neural network*. ITSC 2016.

Probabilistic ETA

- Output a distribution of ETA
- Consider the variance of travel time
- Assume Gaussian distribution for link travel time



* Mohammad Asghari et al. *Probabilistic estimation of link travel times in dynamic road networks*. SIGSPATIAL 2015.

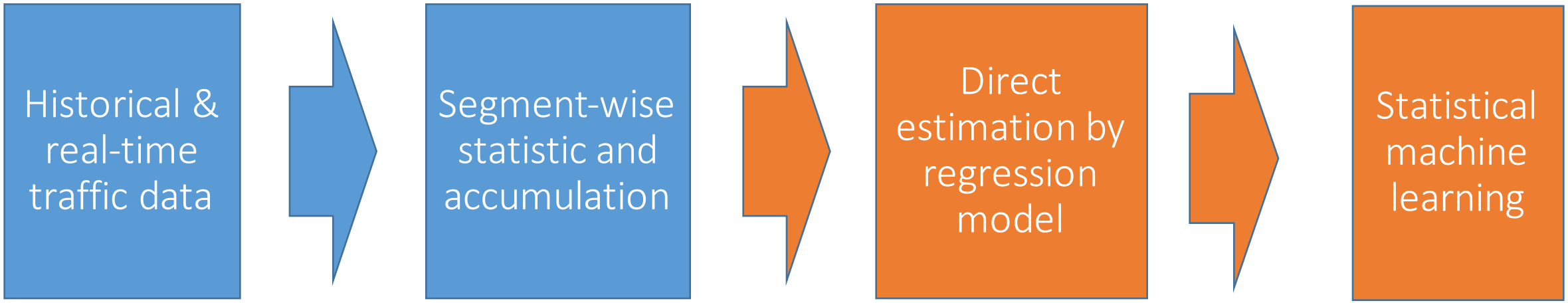
Additive Model vs Global Model

- 1Heuristic rule
- 2 Indirective objective
- 3 Less robust
- 4 Insufficient use of data

Additive model?

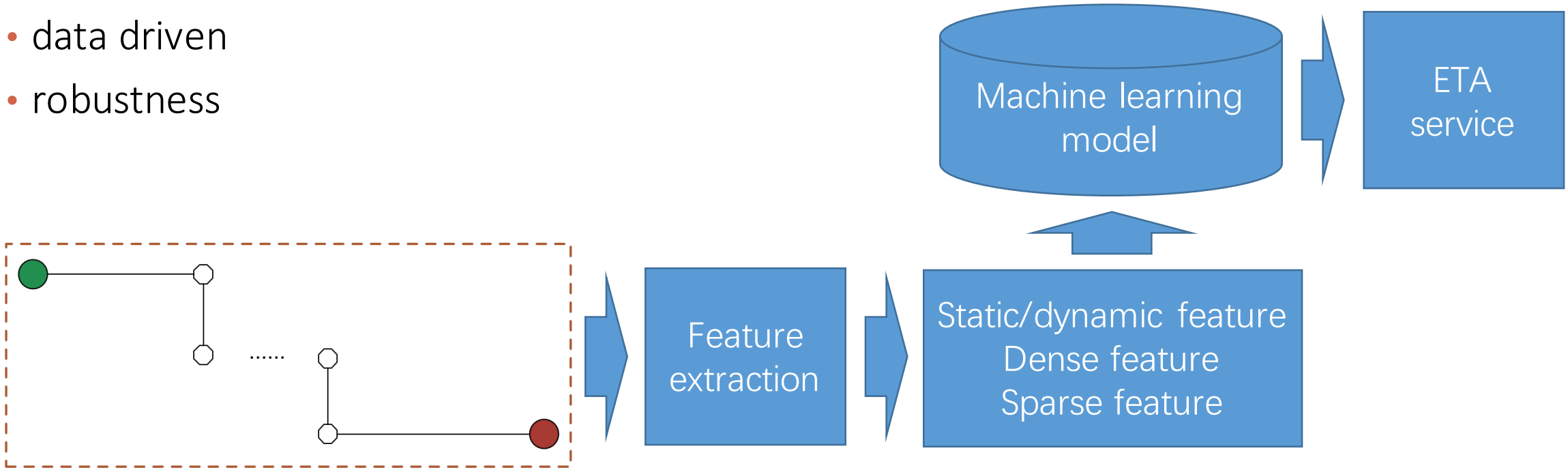
Pure learning?

Rethinking ETA



Learning to Estimate the Travel Time

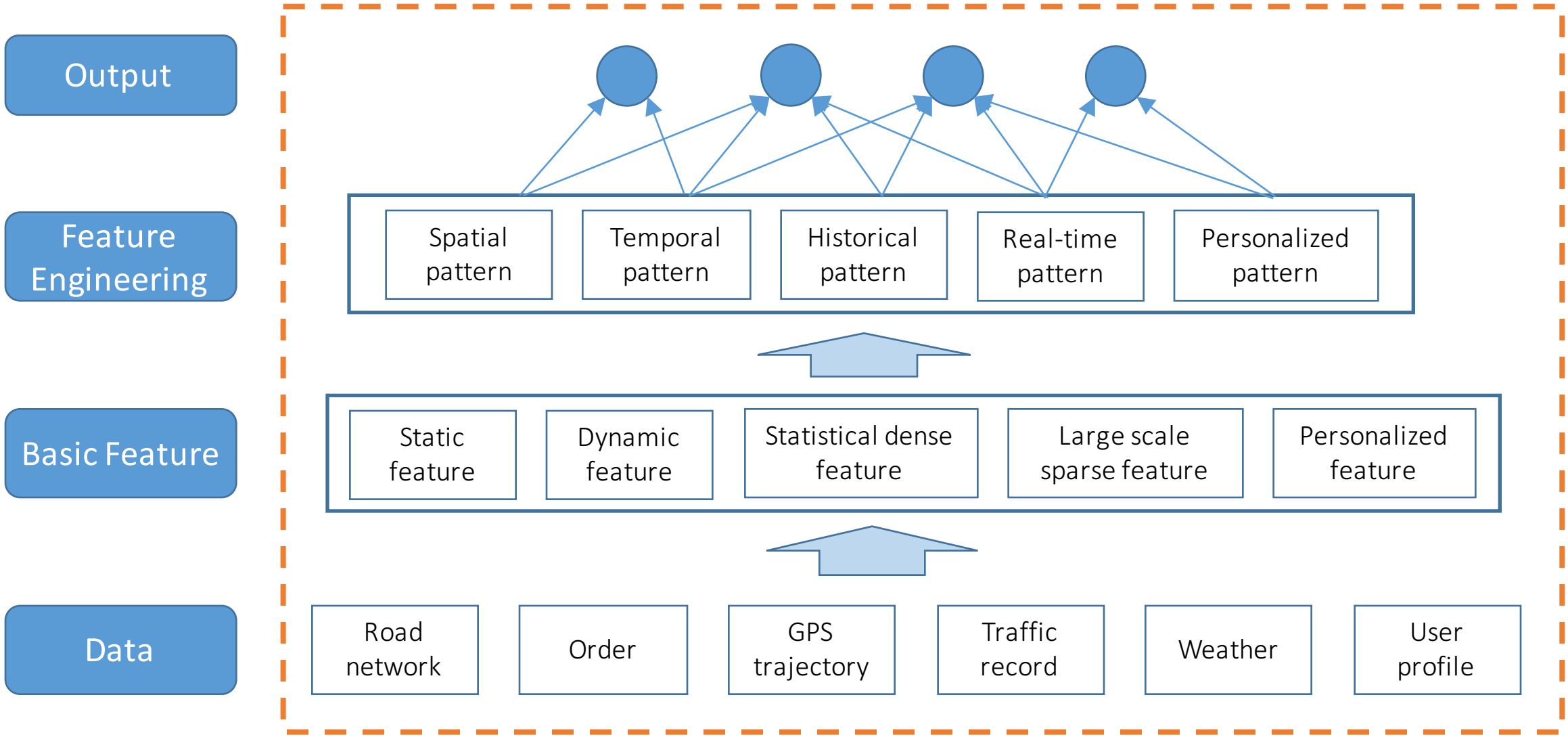
- Big data + machine learning
  - high accuracy
  - data driven
  - robustness



Problem Formulation

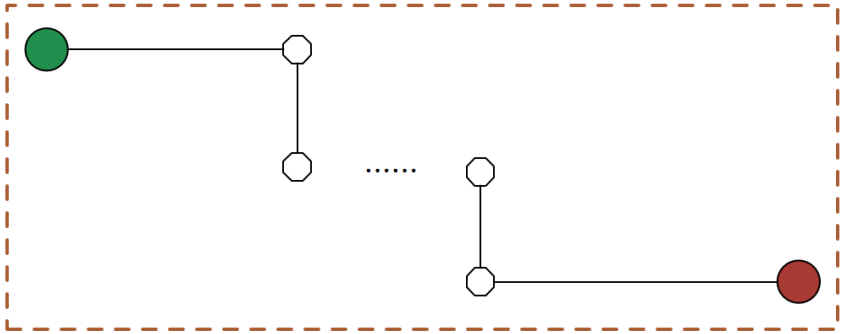
- Formulation: ETA as a regression problem on sequential input
- Data: a collection of trips $\boldsymbol{X} = \{x_i\}_{i=1}^{N}$
  - x_i denotes feature vector on the trajectory along the route path p_i
  - more than 20 million trips a day
- Objective: for MAPE loss, $\min_{f} \sum_{i=1}^{N} \frac{|y_i - f(x_i)|}{y_i}$
- Robust: different training data for different scenarios
  - picking the passenger
  - delivering the passenger
  - order dispatch
  - car pooling

Rich Feature System

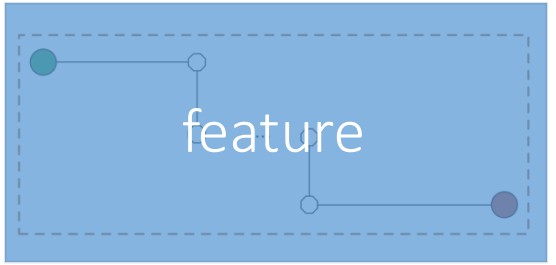


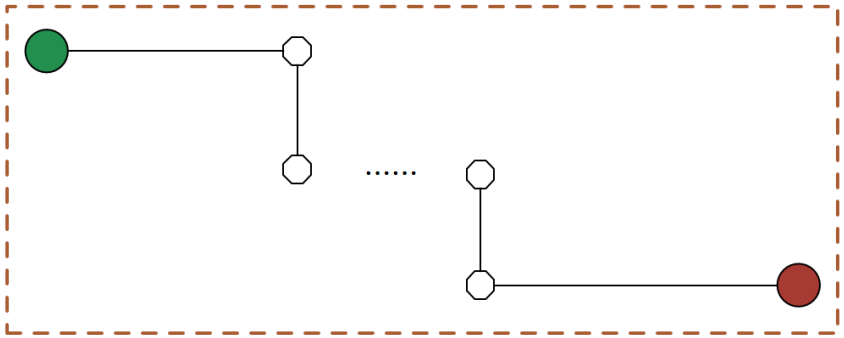
Two faces: global feature vs sequential feature

label T_{od}

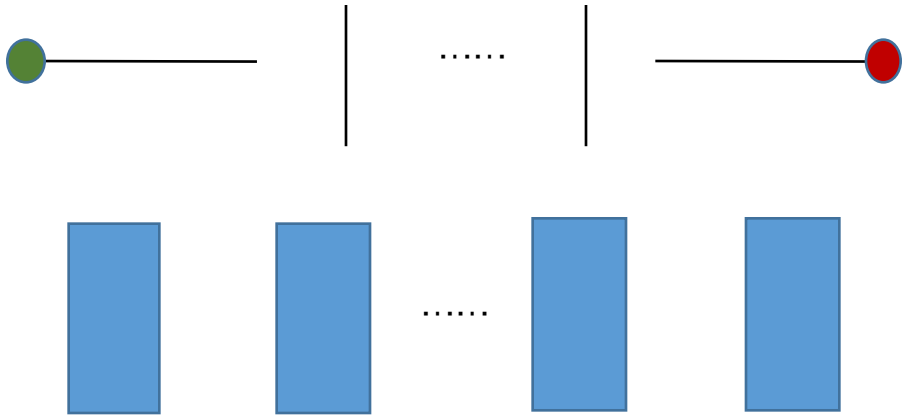


Raw data



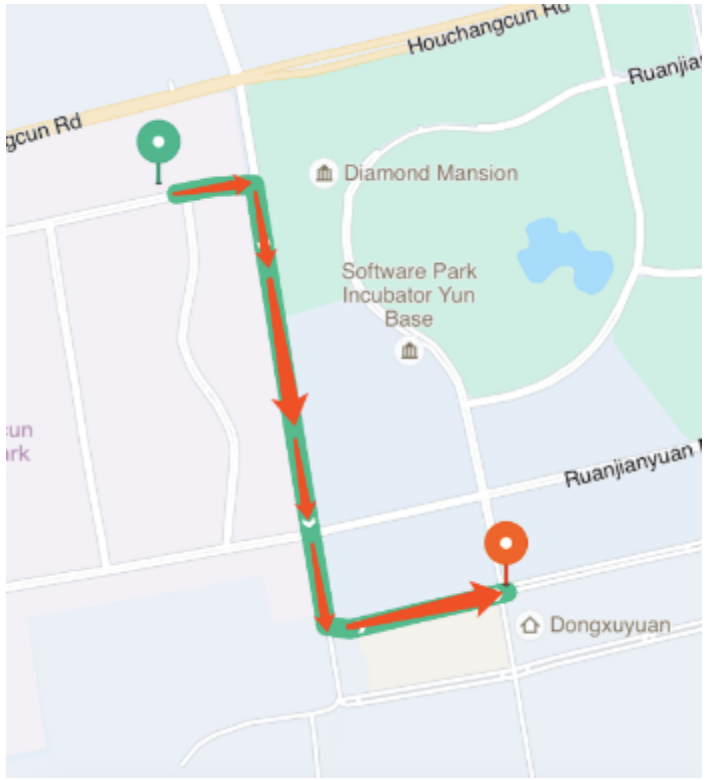


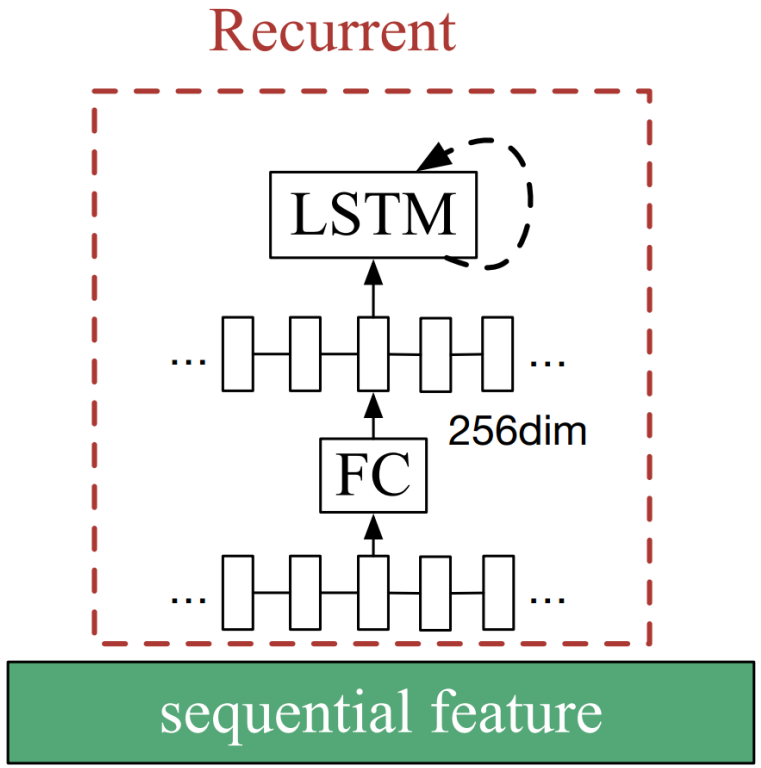
Raw data



Recurrent Model

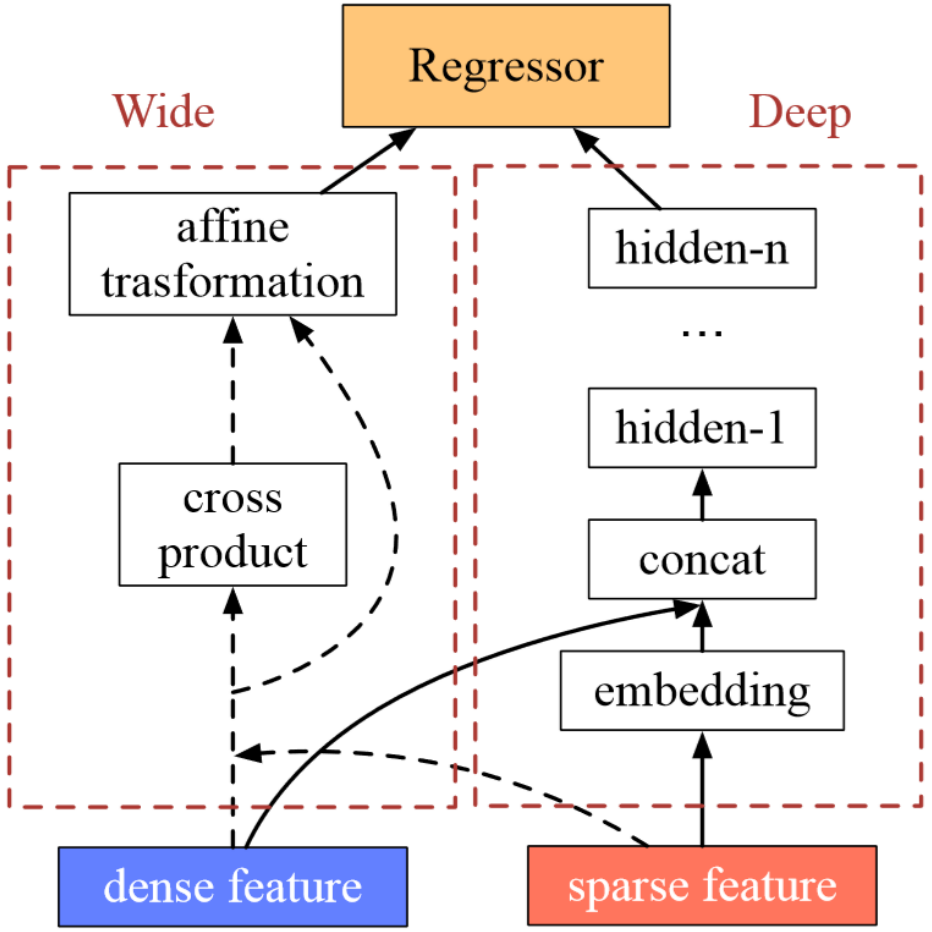
- Model sequential features with Recurrent Neural Network (RNN).



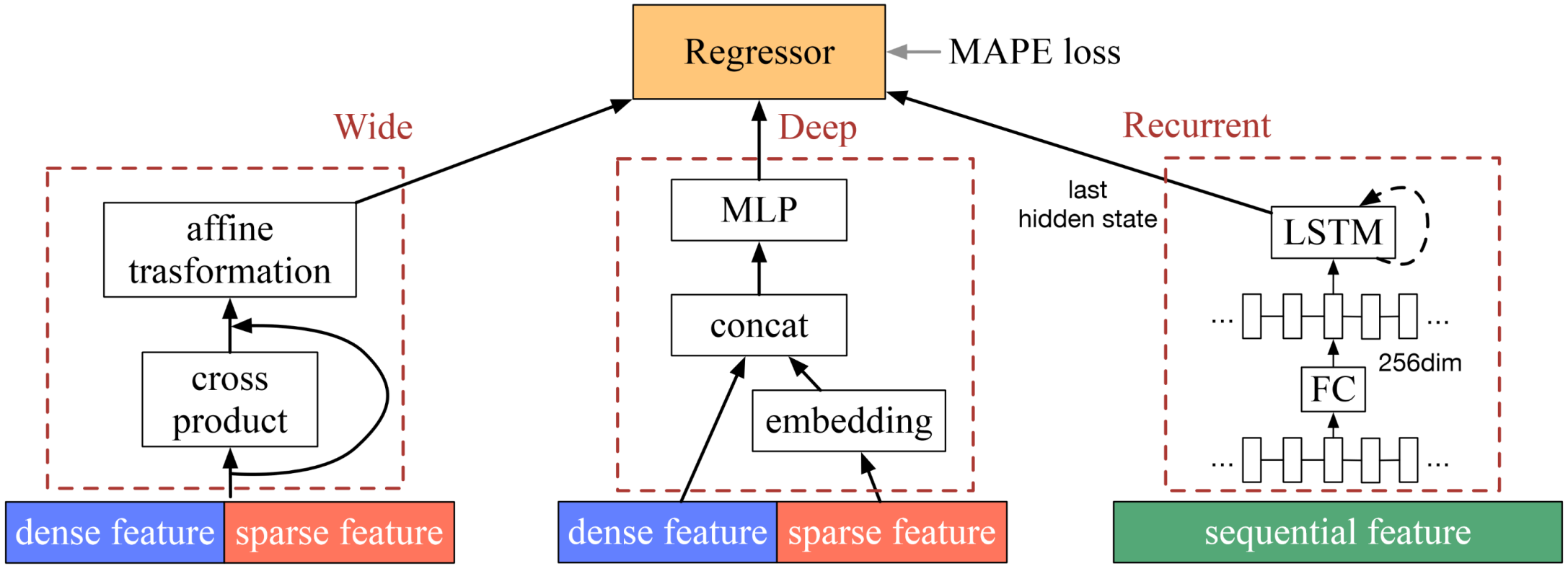


Tree-Based Model -> Wide & Deep Model

- Features
  - Spatial information
  - Temporal information
  - Traffic information
  - Personalized information
  - Augmented information
- W&D model processes the global information



Wide-Deep-Recurrent Network (WDR)



* Zheng Wang et al. *Learning to Estimate the Travel Time*, KDD 2018.

Offline Evaluation Dataset

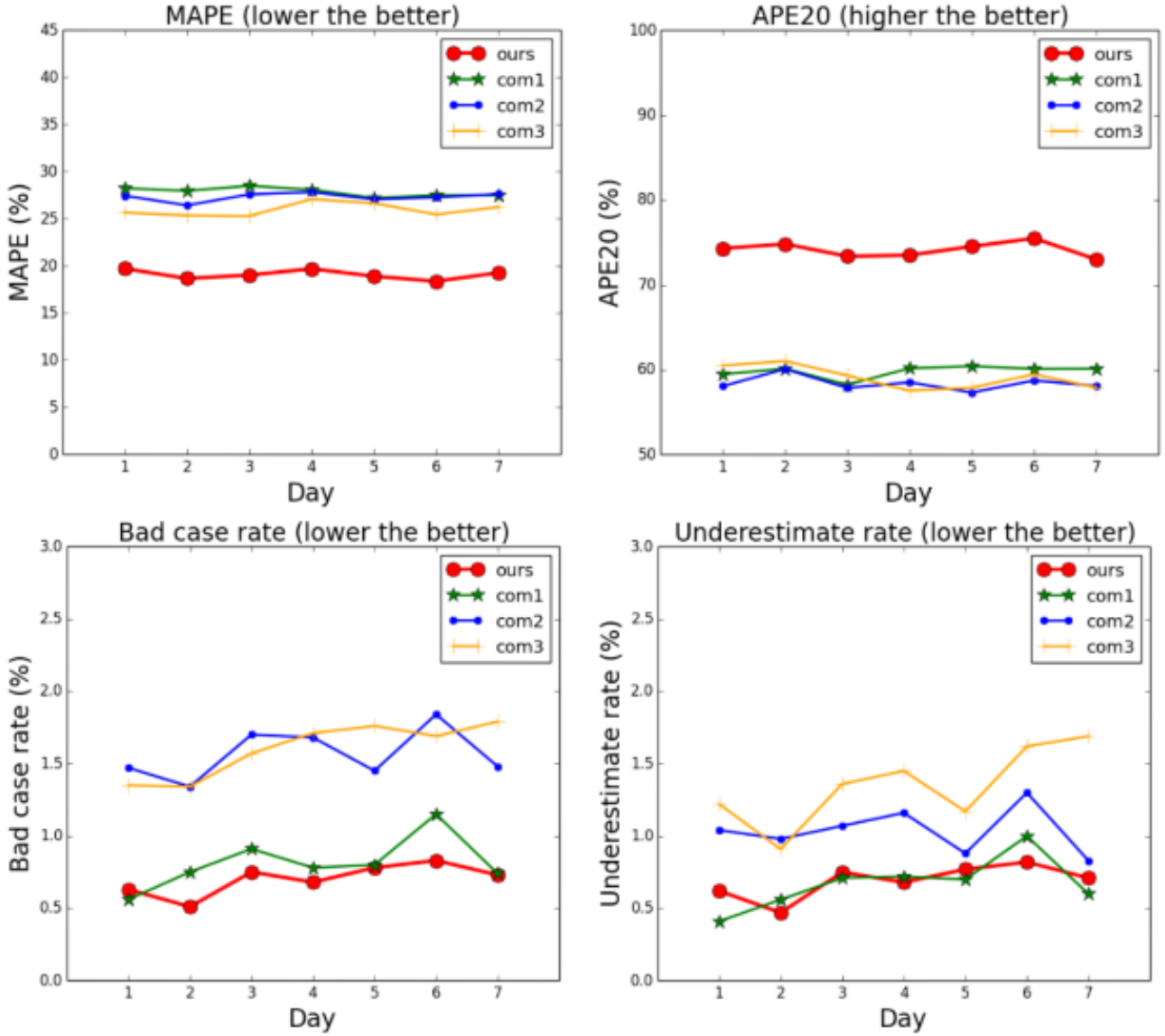
	date	pickup	trip
training set	1.1 - 5.10 (2017)	48M	51M
validation set	5.11 - 5.17 (2017)	3M	4M
test set	5.18 - 5.31 (2017)	6M	7M
unique links	–	0.5M	0.5M
unique drivers	–	0.4M	0.4M

Offline Evaluation Results

	MAPE	MAE	MSE
route-ETA	31.27%	88.0	16555.8
PTTE [20]	29.35%	83.3	13153.5
GBDT	23.64%	68.3	11674.2
FM	21.41%	63.6	9664.2
WD-MLP	21.58%	64.1	9816.3
WDR	**20.83%**	**59.9**	**9078.2**
*$TEMP_{rel}$ [19]	35.30%	79.7	15480.7
*WDR	**21.68%**	**55.6**	**7962.0**

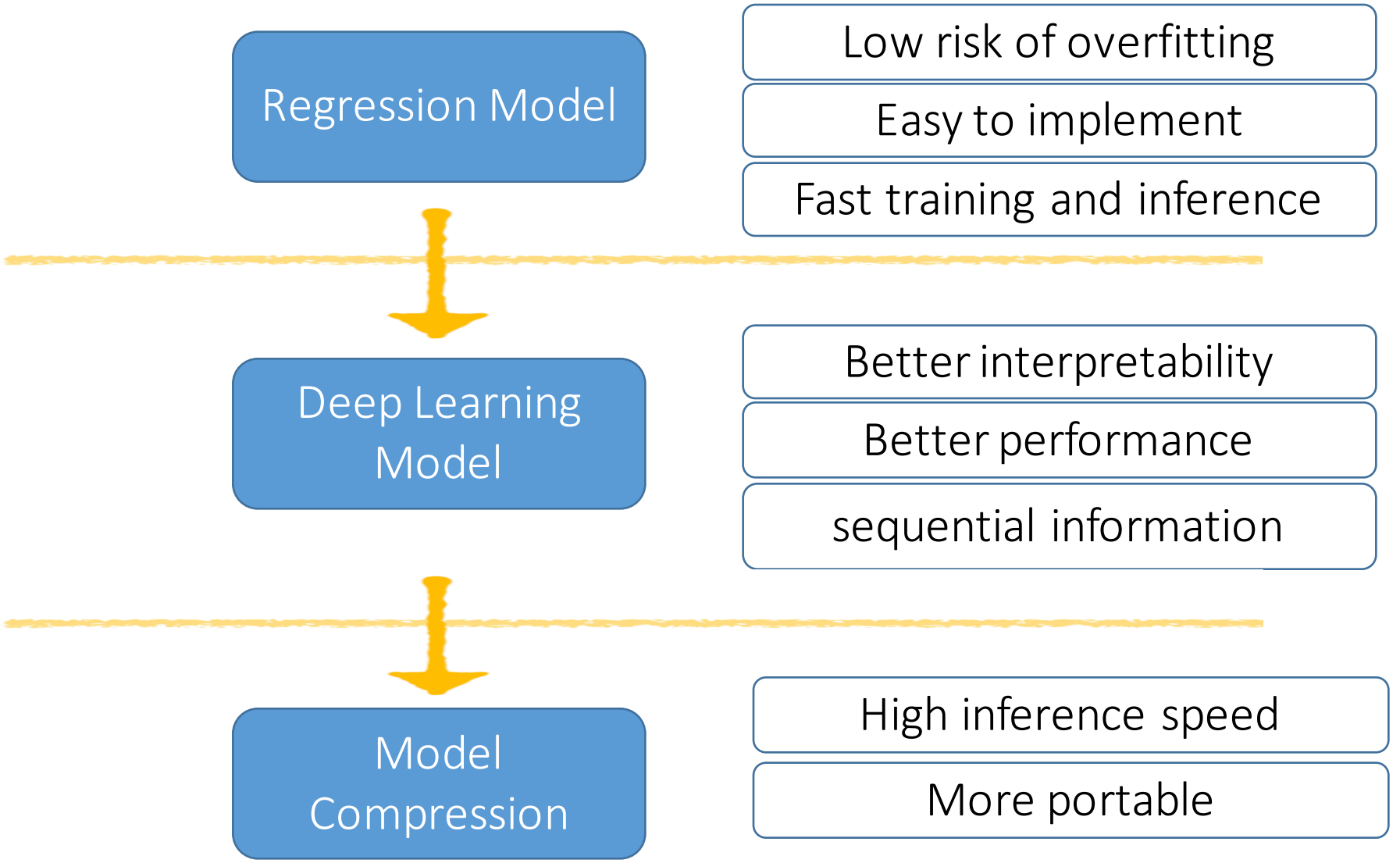
	MAPE	MAE	MSE
route-ETA	15.01%	153.2	66789.8
PTTE [20]	14.78%	150.1	66141.2
GBDT	14.01%	133.2	52006.6
FM	12.87%	122.5	47457,3
WD-MLP	13.43%	127.5	50012.8
WDR	**11.66%**	**112.4**	**37482.7**
*$TEMP_{rel}$ [19]	23.53%	166.8	88767.1
*WDR	**12.27%**	**87.2**	**20929.3**

Online Evaluation Results



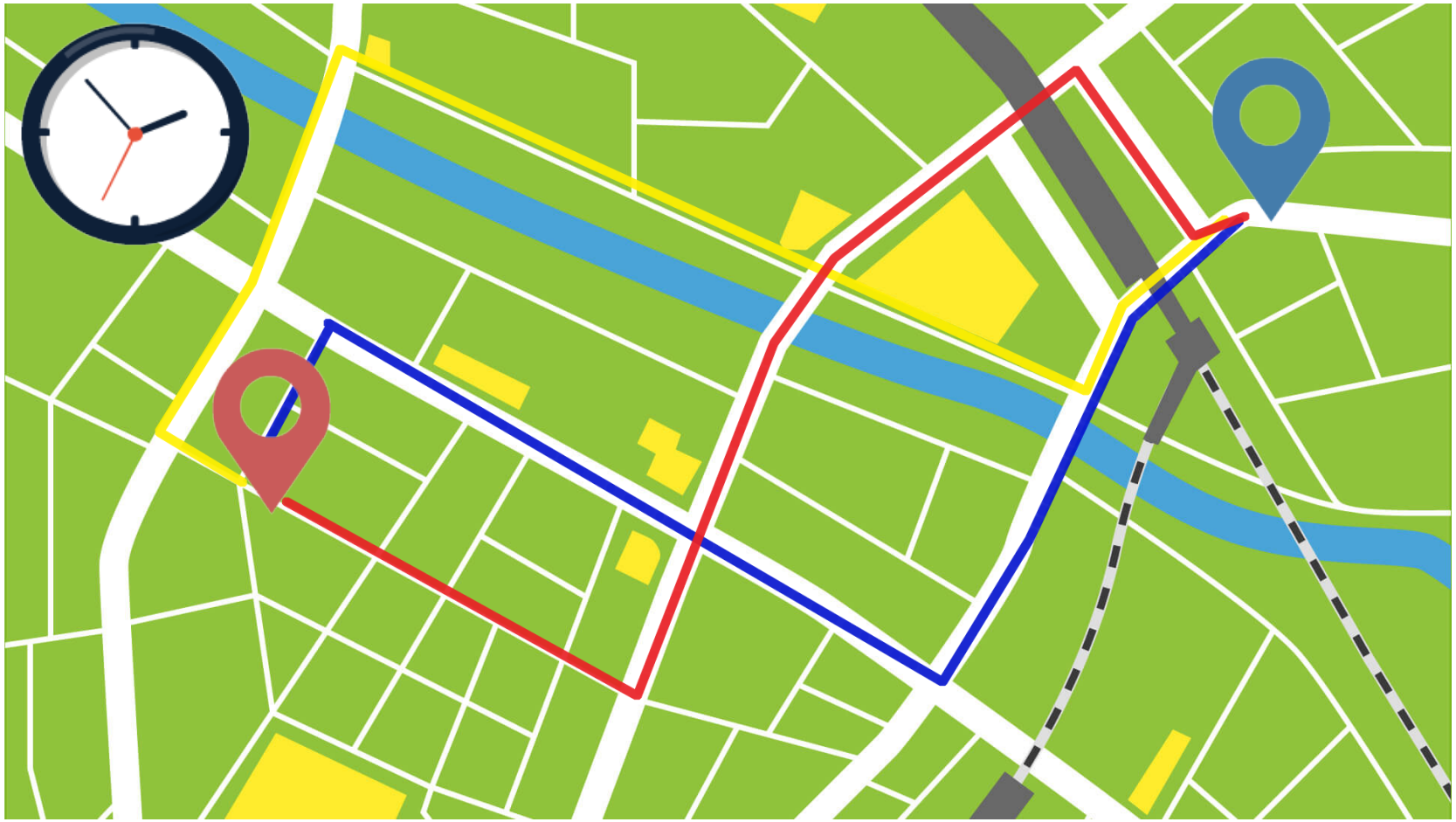
WDR achieves the best performance for all metrics.

Progress


Regression Model
Low risk of overfitting
Easy to implement
Fast training and inference
Deep Learning Model
Better interpretability
Better performance
sequential information
Model Compression
High inference speed
More portable

More Practical: Origin Destination ETA

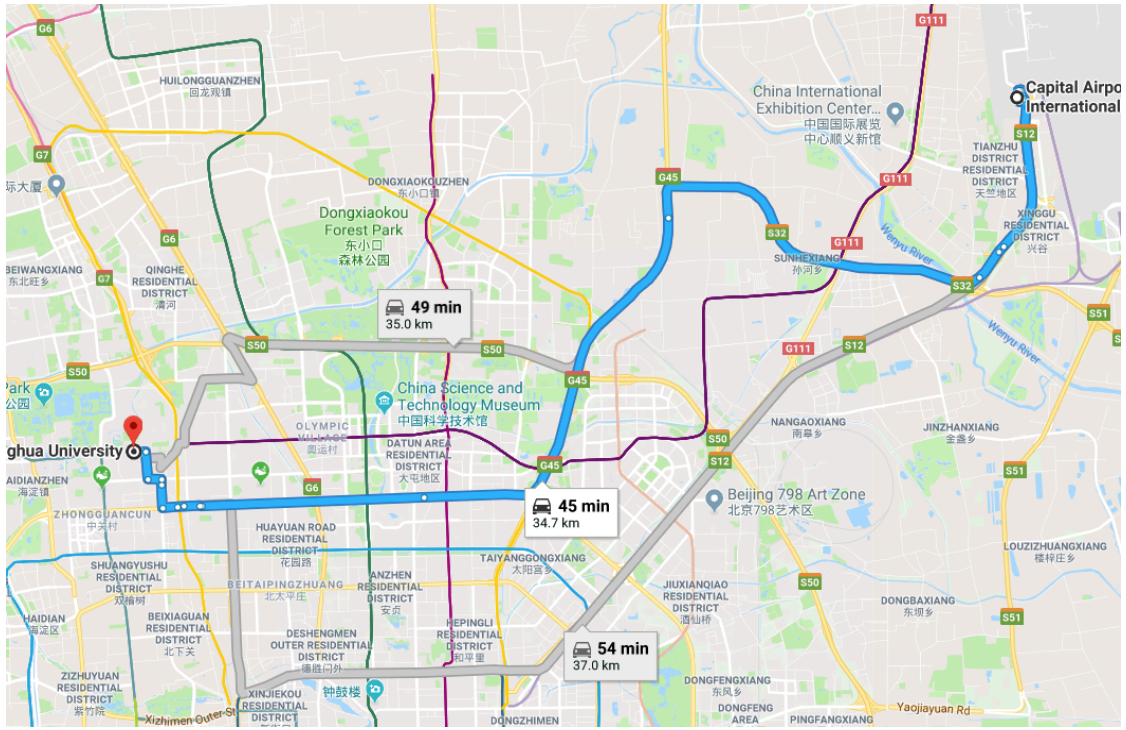
Route is not available until the end of the trip.



Origin Destination ETA

- Origin-destination ETA (path free)

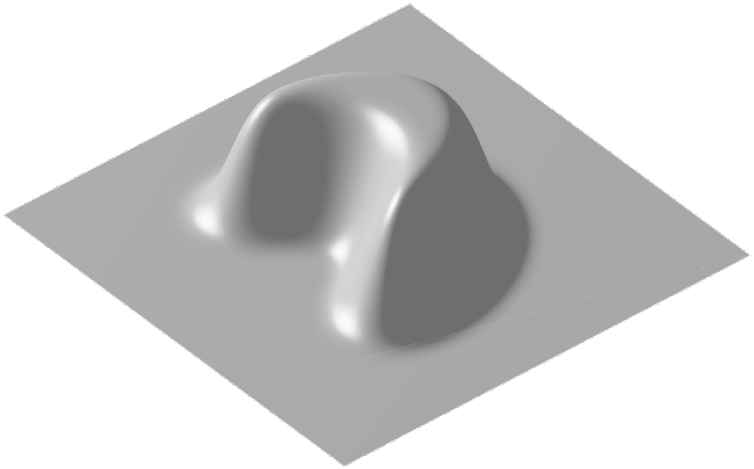
- Challenges
  - Limited information: no actual path
  - Complicated spatiotemporal dependencies



MURAT: Multi-task Representation Learning for ETA

Road Network Embedding

Spatiotemporal Smoothness



MURAT: Multi-task Representation Learning for ETA

Historical Paths

Travel Distance

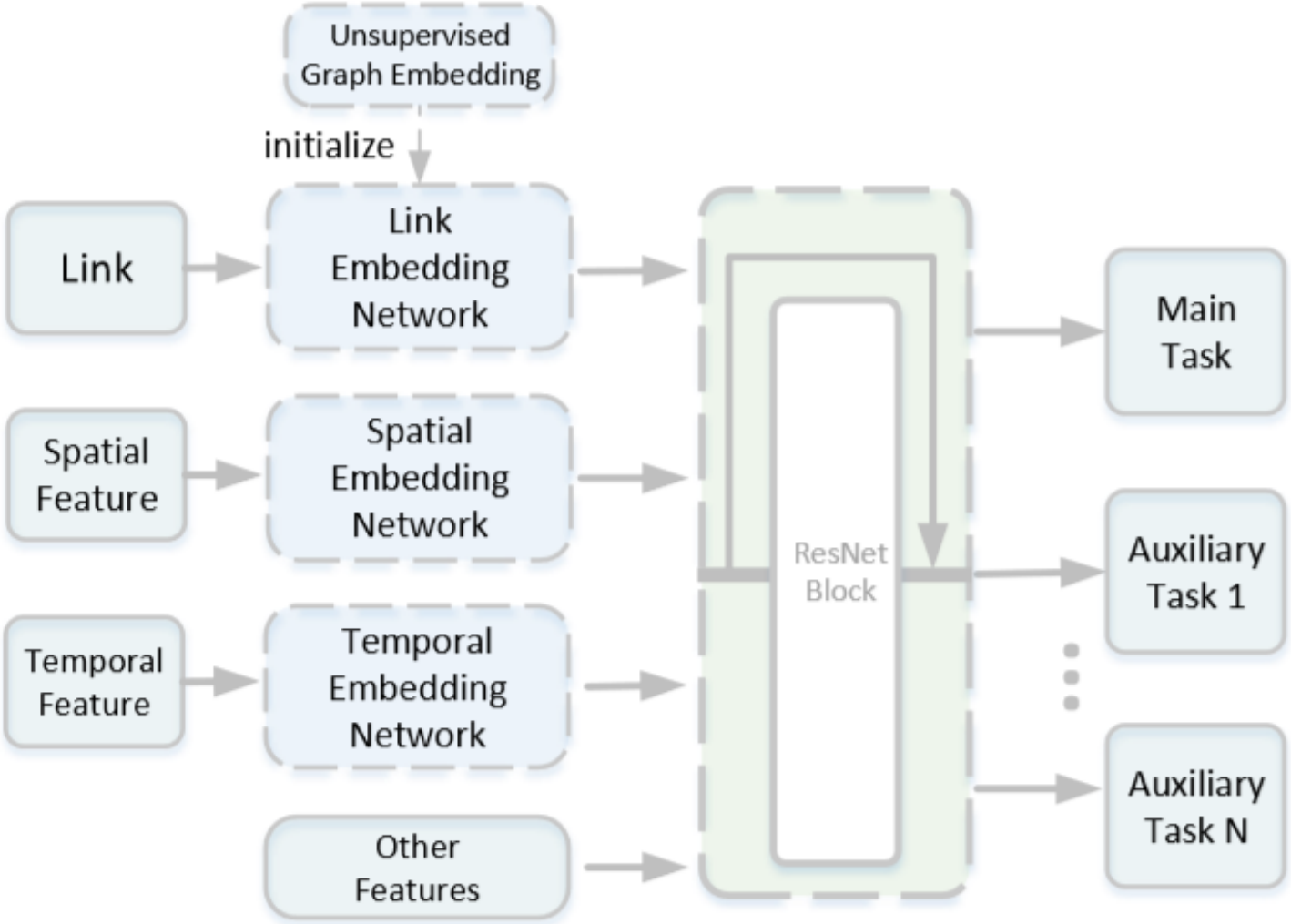
Road Segments

Traffic lights

Left/right Turns

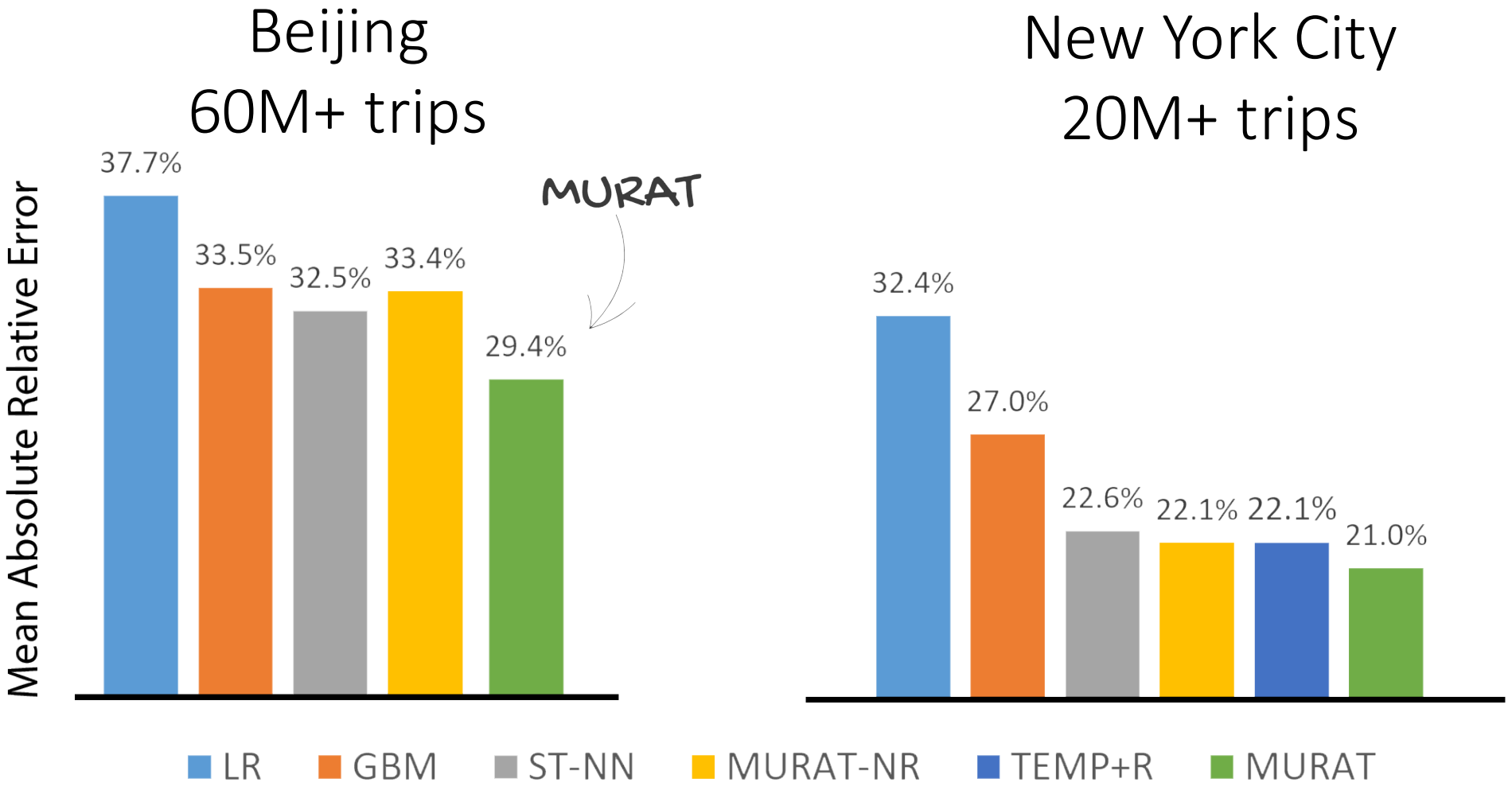
Auxiliary Tasks

MURAT: Multi-task Representation Learning for ETA

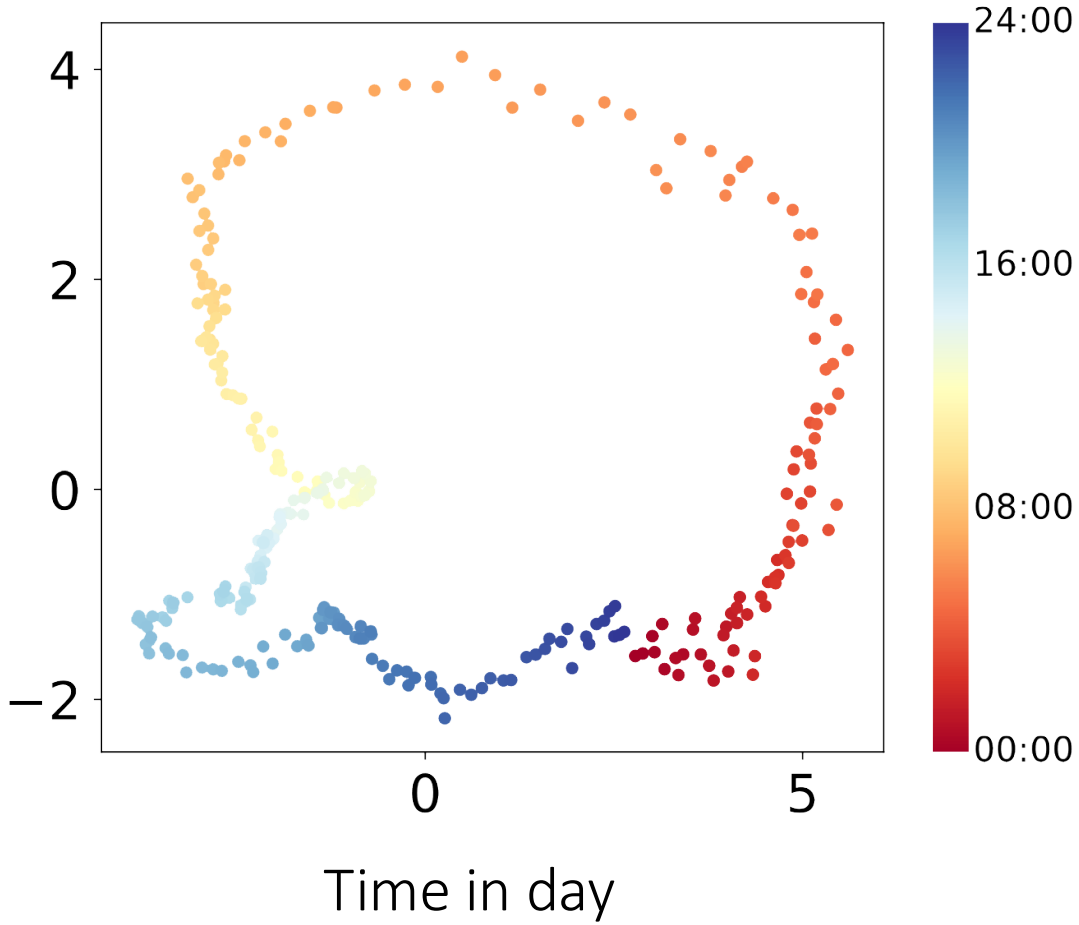


* Yaguang Li et al. Multi-task Representation Learning for Travel Time Estimation, KDD, 2018

Evaluation


Beijing
60M+ trips
New York City
20M+ trips
Mean Absolute Relative Error
37.7%
33.5%
32.5%
33.4%
29.4%
MURAT
32.4%
27.0%
22.6%
22.1%
22.1%
21.0%
LR
GBM
ST-NN
MURAT-NR
TEMP+R
MURAT

Learned Representation



Circular Shape

Smooth transition

Reference

- Corrado de Fabritiis et al. Traffic Estimation And Prediction Based On Real Time Floating Car Data, IEEE TITS, 2008.
- Erik Jenelius et al. Travel time estimation for urban road networks using low frequency probe vehicle data, Transportation Research B, 2013
- Yilun Wang et al. Travel time estimation of a path using sparse trajectories, KDD 2014
- Zheng Wang et al. Learning to estimate the travel time, KDD 2018.
- Yaguang Li et al. Multi-task Representation Learning for Travel Time Estimation, KDD 2018.

KDD2018

Map Service II

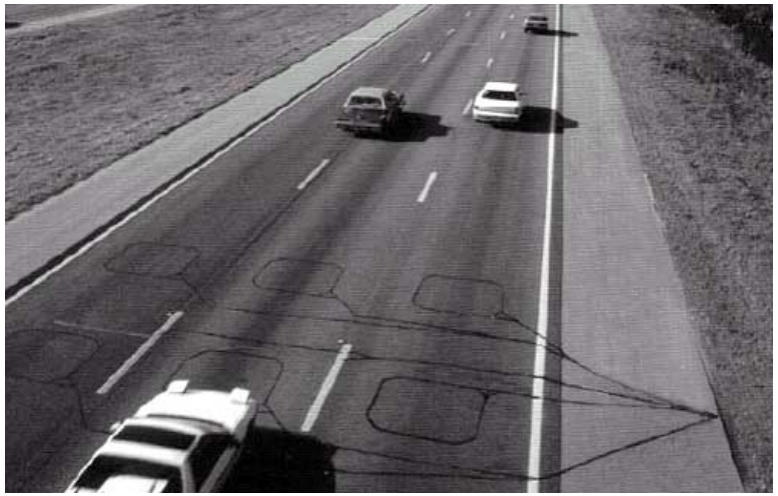
Traffic Estimation and Forecasting

Introduction

- Traffic congesting is wasteful of time, money and energy
  - Traffic congestion costs Americans $124 billion+ direct/indirect loss in 2013.
- Accurate traffic forecasting could substantially improve route planning and mitigate traffic congestion.

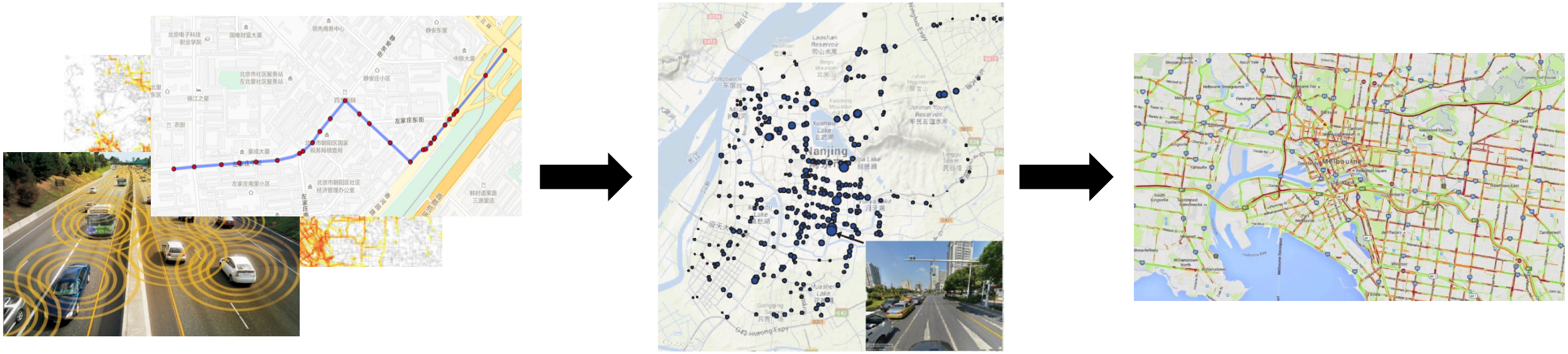
Data Source

- Traffic estimation and forecasting
  - Data source: loop detector, GPS trajectory and camera
  - Target: traffic speed in real time and for the future



Traffic Estimation

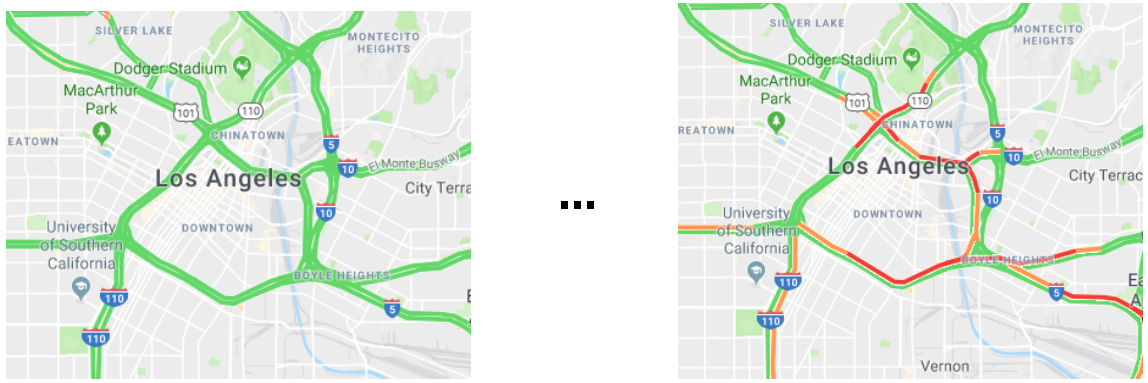
- Calculate the ground-truth traffic data: multiple source data fusion, spatial-temporal spline.



Traffic Prediction

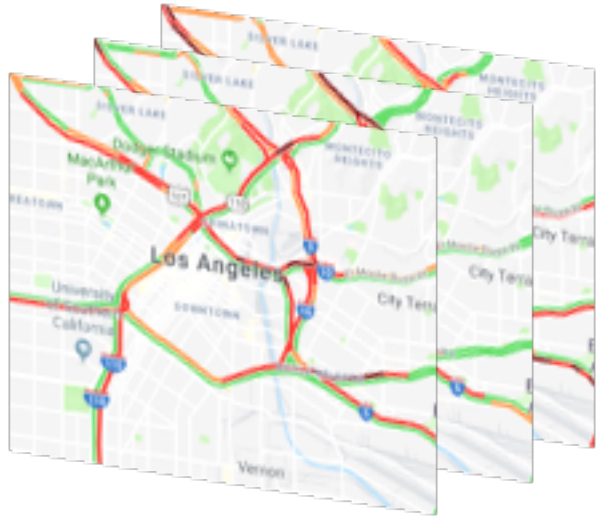
- Input: road network and past T' traffic speed observed at sensors
- Output: traffic speed for the next T steps

Input: Observations



7:00 AM ... 8:00 AM

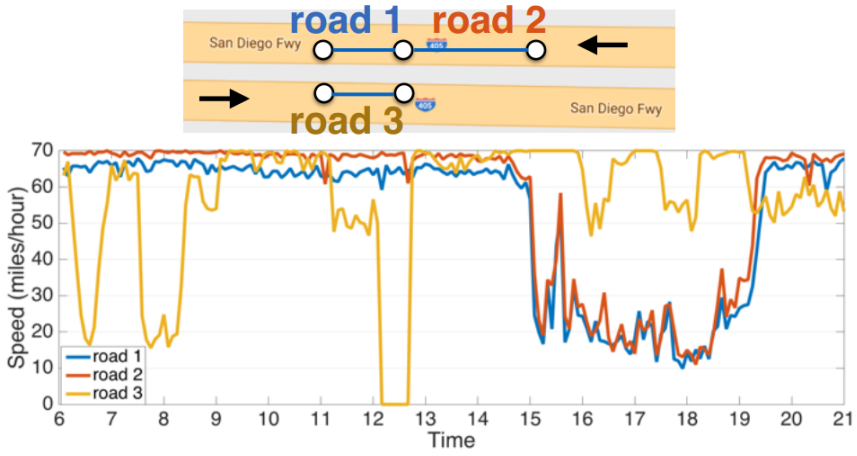
Output: Predictions



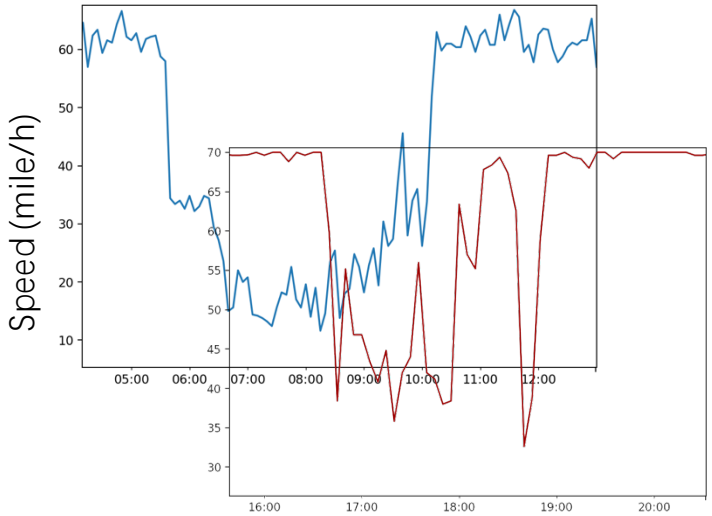
8:10AM, 8:20AM, ..., 9:00 AM

Challenges for Traffic Forecasting

Complex
Spatial Dependency

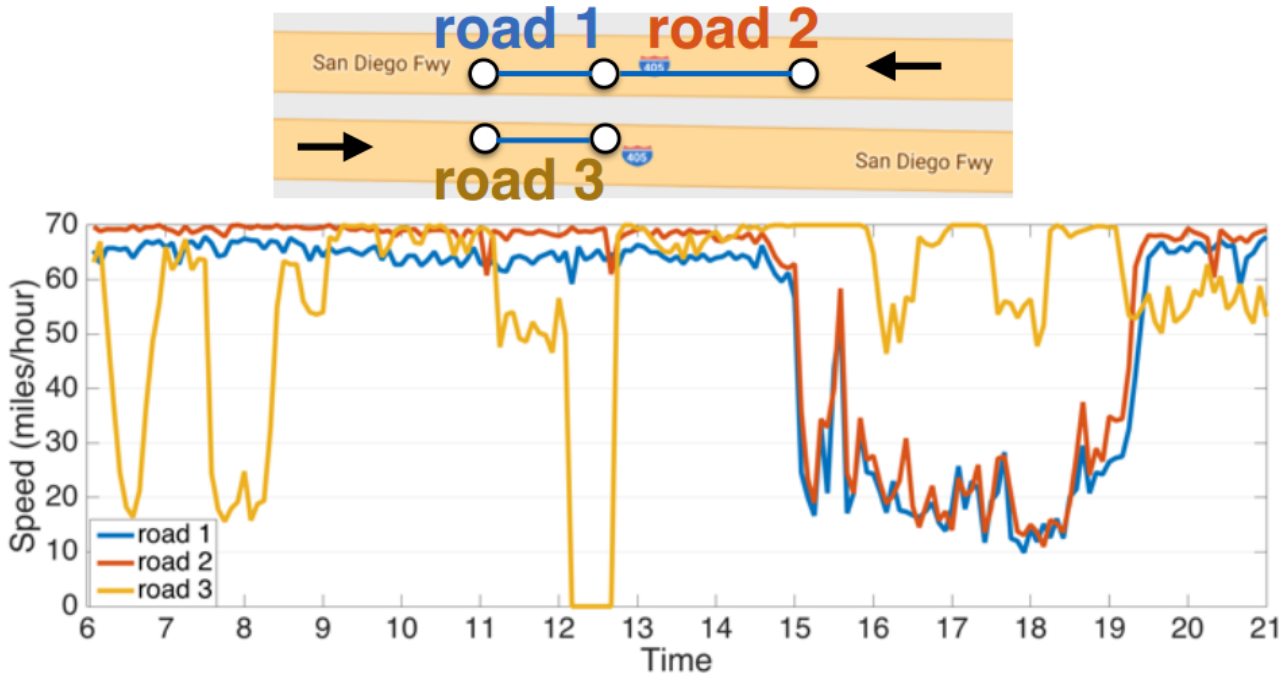


Non-linear, non-stationary
Temporal Dynamic



Challenges for Traffic Forecasting

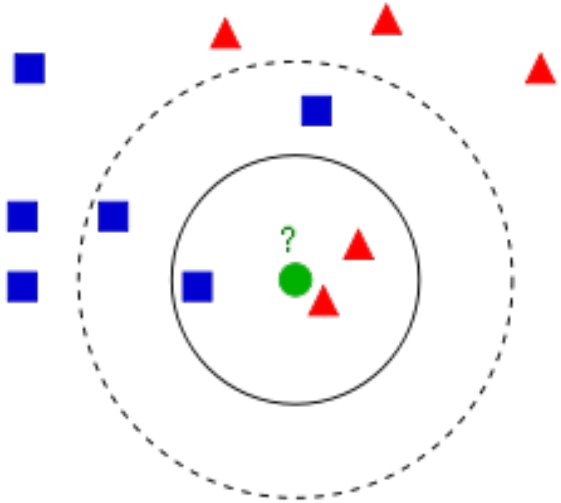
- Spatial relationship among traffic flow is ***non-Euclidean*** and ***directed***



KNN-based Approaches

- Find similar historical traffic time series
  - Extract various features from traffic time series
  - Define similarity metrics between traffic time series

- Calculate the prediction by aggregating next traffic readings of nearest neighbors.



* Picture from Wikipedia. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

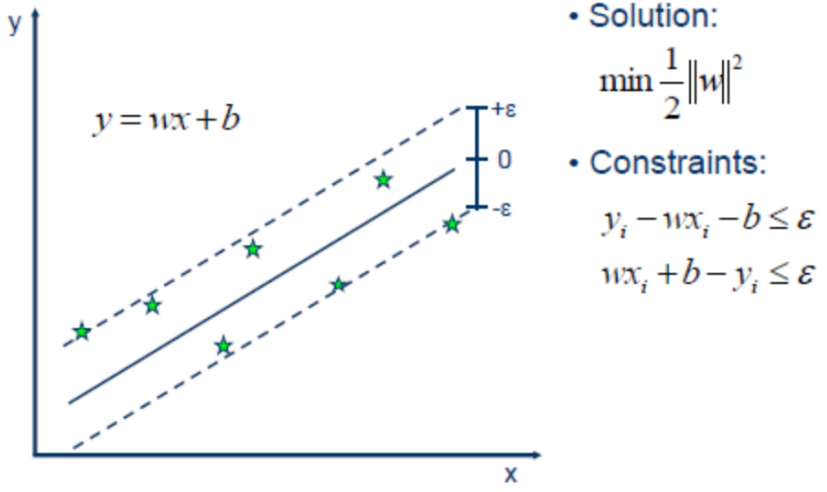
Time Series Methods

- Seasonal Autoregressive Integrated Moving Average (SARIMA)

$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)(1-L)^d X_t = \delta + \left(1+\sum_{i=1}^{q}\theta_i L^i\right)\varepsilon_t.$$

S-ARIMA
Auto-Regressive Integrated Moving Averages
(w/Seasonality)

- Support Vector Regression (SVR)



- Solution:

$$\min \frac{1}{2}\|w\|^2$$

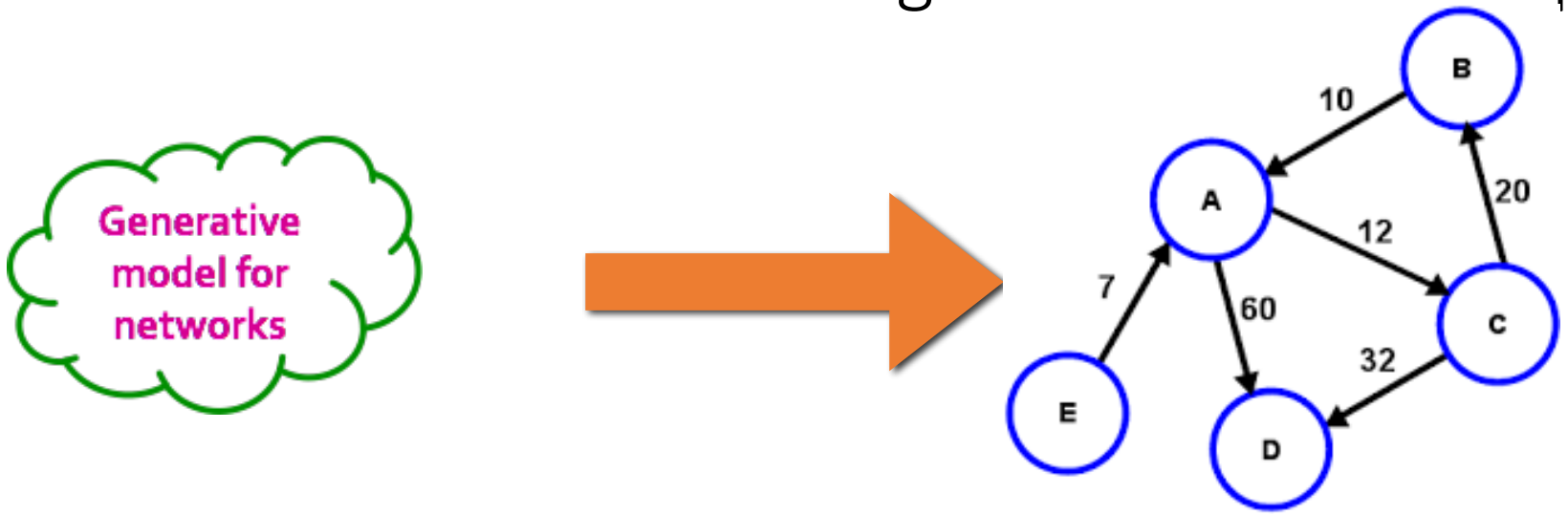
- Constraints:

$$y_i - wx_i - b \le \varepsilon$$
$$wx_i + b - y_i \le \varepsilon$$

* Picture from http://www.saedsayad.com/support_vector_machine_reg.htm

Traffic Prediction with Latent Space Model (LSM)

- Model the road network as a graph
- Model correlation between road segment with Latent Space model



Define a model that can generate the traffic condition of road network

* Dingxiong Deng et al, Latent Space Model for Road Networks to Predict Time-Varying Traffic. KDD, 2016

Latent Attributes

- Each vertex has **latent** attributes
  - Vertex ***i*** has latent attribute vector u_i $\quad u_i \in R_+^{1\times k}$

	highway	arterial	business	resident	intersection	non-inter
i	0.9	0.1	0.8	0.2	1	0
j	0.8	0.1	0.7	0.3	0	1
…						

Node attribute matrix U^{nxk}
(n is the number of vertices)

Attribute Interaction

- Interaction matrix ***B*** between different attributes

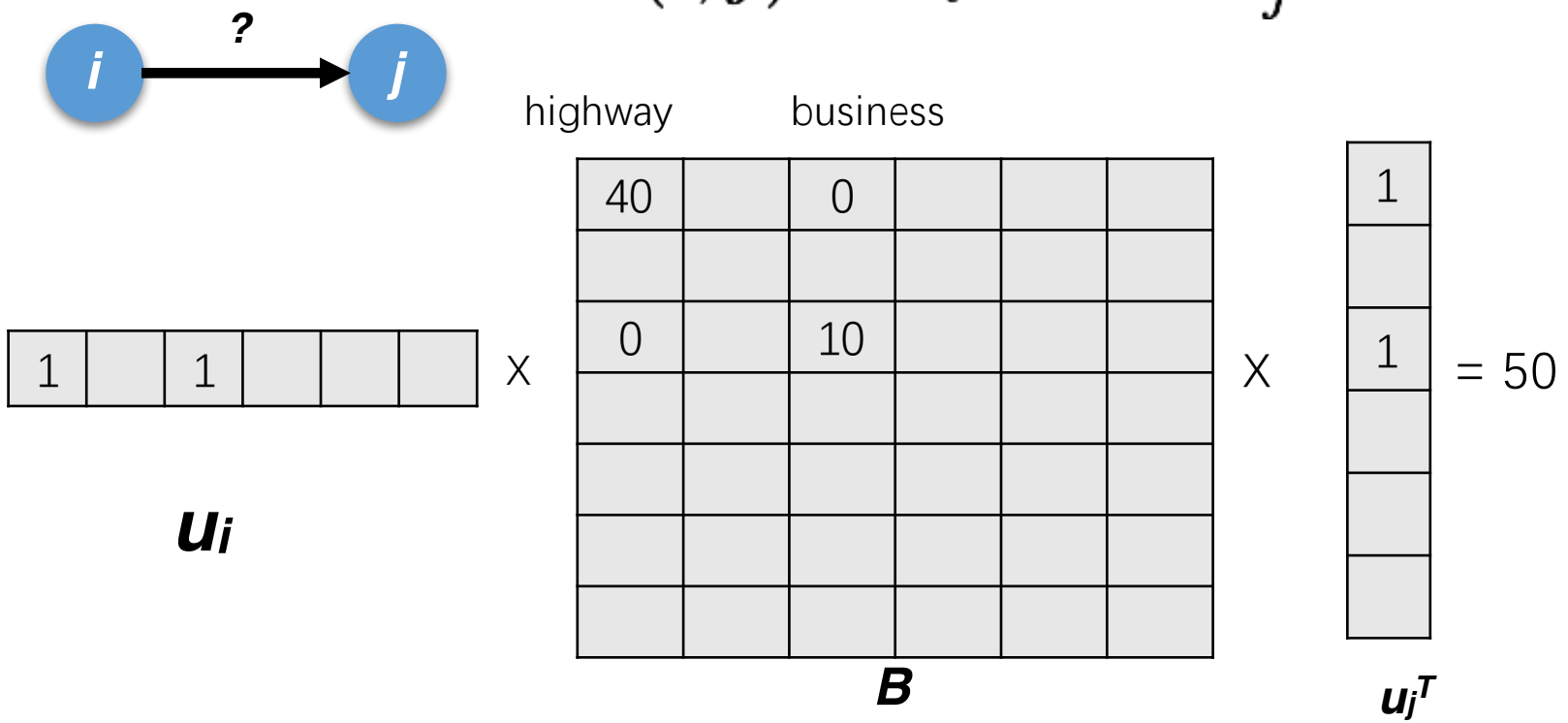
	highway	arterial	business	resident	intersection	non-inter
highway	40	20	20	40	10	20
arterial		…				
business			…			
residential				…		
intersection					…	
non-inter						…

Attribute interaction matrix $\boldsymbol{B} \in \boldsymbol{R}^{k \times k}$

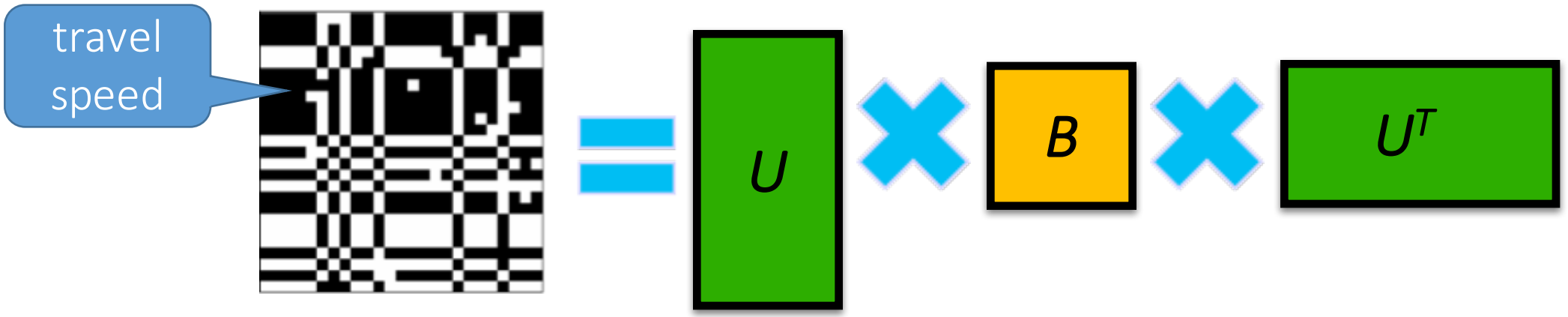
Speed from Latent Attribute

- Traffic speed between vertices i and j ($i \rightarrow j$) is a linear combination of the corresponding traffic patterns

$$d(i,j) = u_i \times B \times u_j^T$$



Basic Graph Model



Graph matrix: $\boldsymbol{G} \in \boldsymbol{R}^{N \times N}$

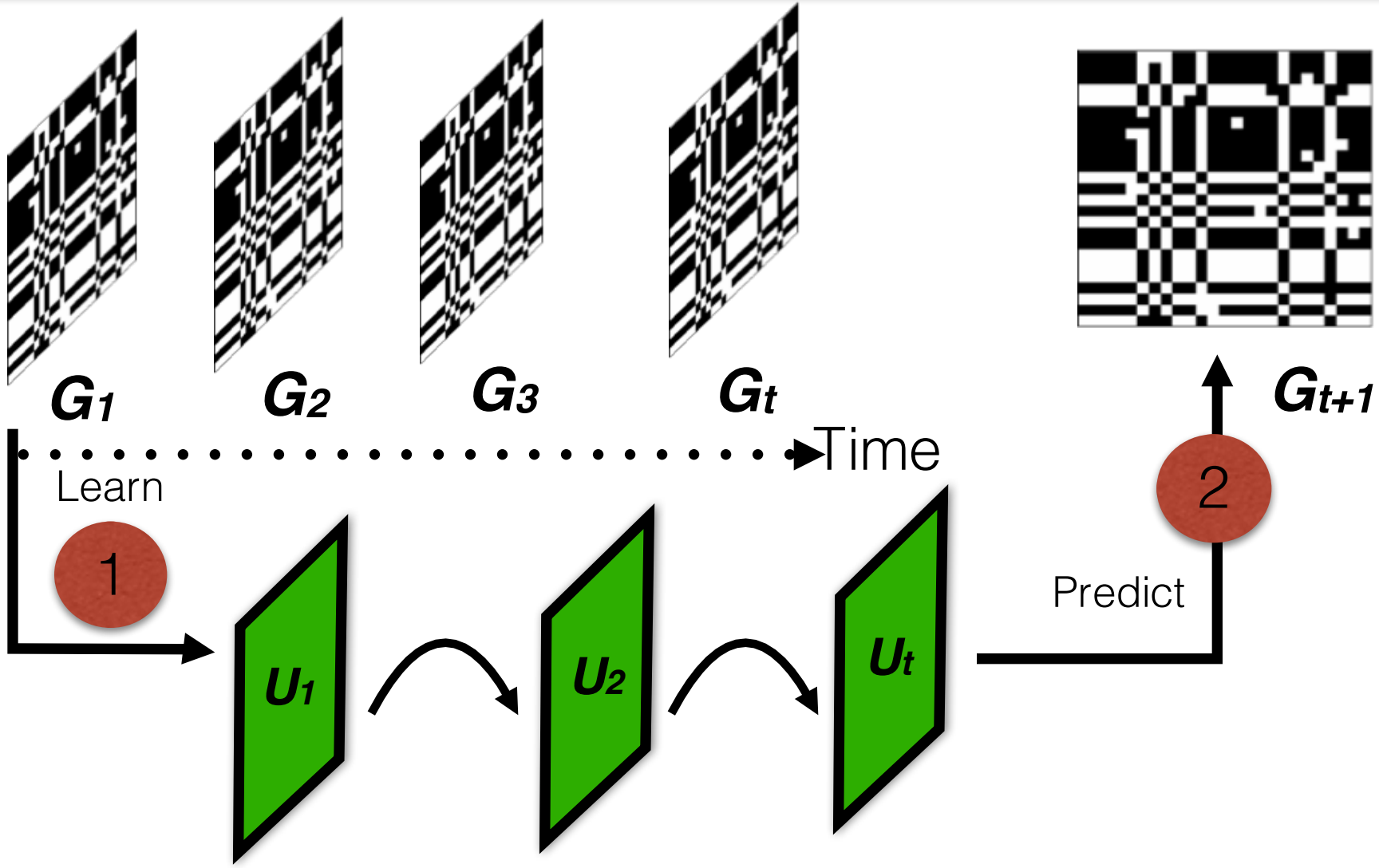
Latent properties: $\boldsymbol{U} \in \boldsymbol{R}^{n \times k}$ and $\boldsymbol{B} \in \boldsymbol{R}^{k \times k}$

$$\arg \min_{U \geq 0, B \geq 0} J = ||G - UBU^T||_F^2$$

Non-negative Graph Matrix Factorization (NMF) [1]

[1] Wang, F., Li, T., Wang, X., Zhu, S. and Ding, C., 2011. Community discovery using nonnegative matrix factorization. Data Mining and Knowledge Discovery, 22(3), pp.493-521.

Overview of LSM for Road Network



Latent attributes evolve with different timestamps

LSM for Road Network

Basic Graph Model

$$\arg \min_{U \geq 0, B \geq 0} J = ||G - UBU^T||_F^2$$

Overcome the Sparsity

$$\arg \min_{U,B} J = ||Y \odot (G - UBU^T)||_F^2 + \lambda Tr(U^T L U)$$

Temporal Effect

$$\arg \min_{U_i,B,A} J = \sum_{i=1}^{t} ||Y_i \odot (G_i - U_i B U_i^T)||_F^2 + \sum_{i=1}^{t} \lambda Tr(U_i^T L U_i)$$

Transition Effect

$$\arg \min_{U_i,B,A} J = \sum_{i=1}^{t} ||Y_i \odot (G_i - U_i B U_i^T)||_F^2 + \sum_{i=1}^{t} \lambda Tr(U_i^T L U_i) + \sum_{i=2}^{t} \gamma ||U_i - U_{i-1} A||_F^2$$

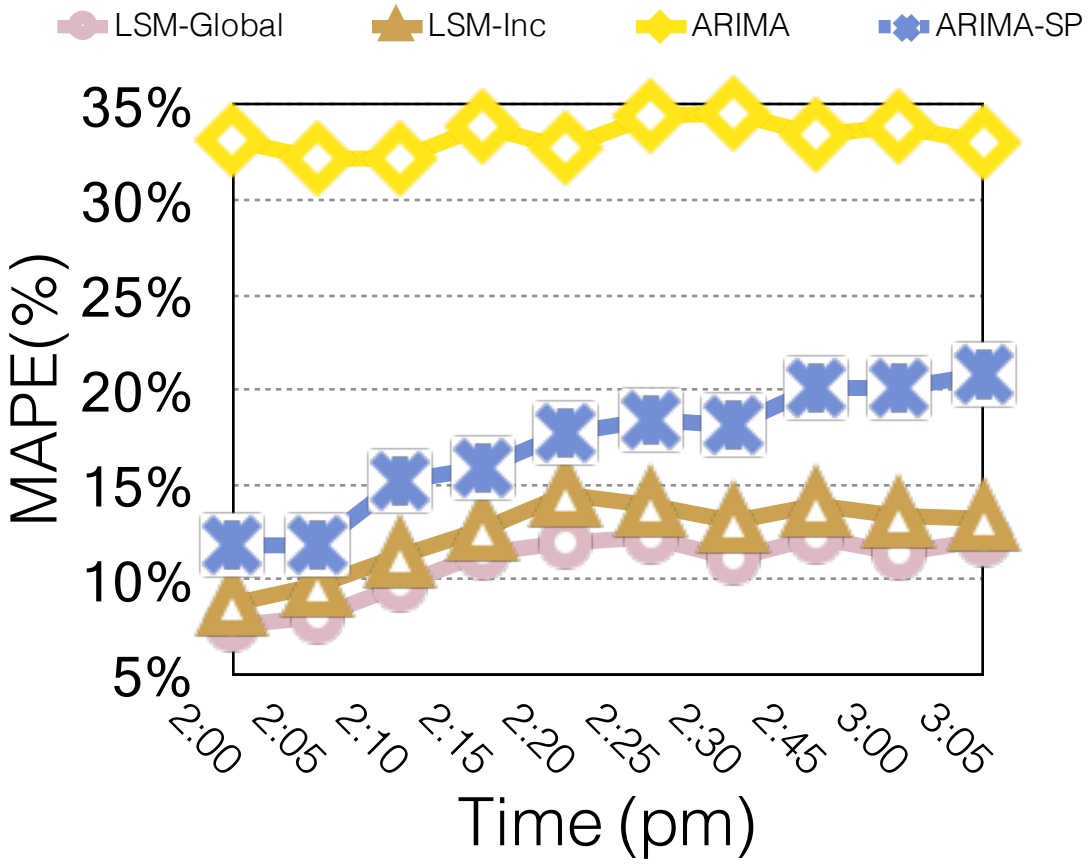
* Dingxiong Deng et al, Latent Space Model for Road Networks to Predict Time-Varying Traffic. KDD, 2016

Experimental Settings

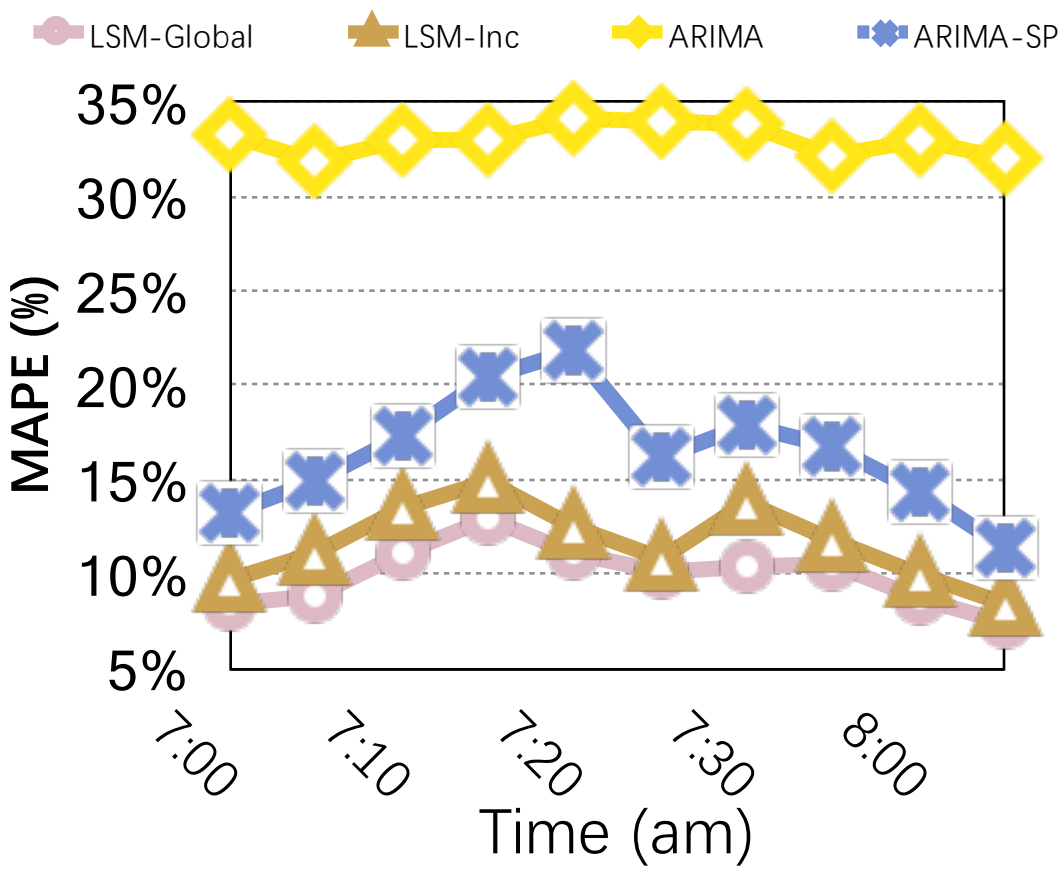
- Dataset
  - March and April, 2014 sensor data with more than 60 million records
  - Two subgraphs of Los Angeles road network
- Baselines:
  - LSM-Naive [Zhang et al. KDD'12]
  - ARIMA [Williams et al. TRB'98], ARIMA-SP
  - SVR[Wu et al. ITS'04], SVR-SP
- Measurement: MAPE, RMSE

Experimental Results

Latent space models (LSM) outperform other baselines.



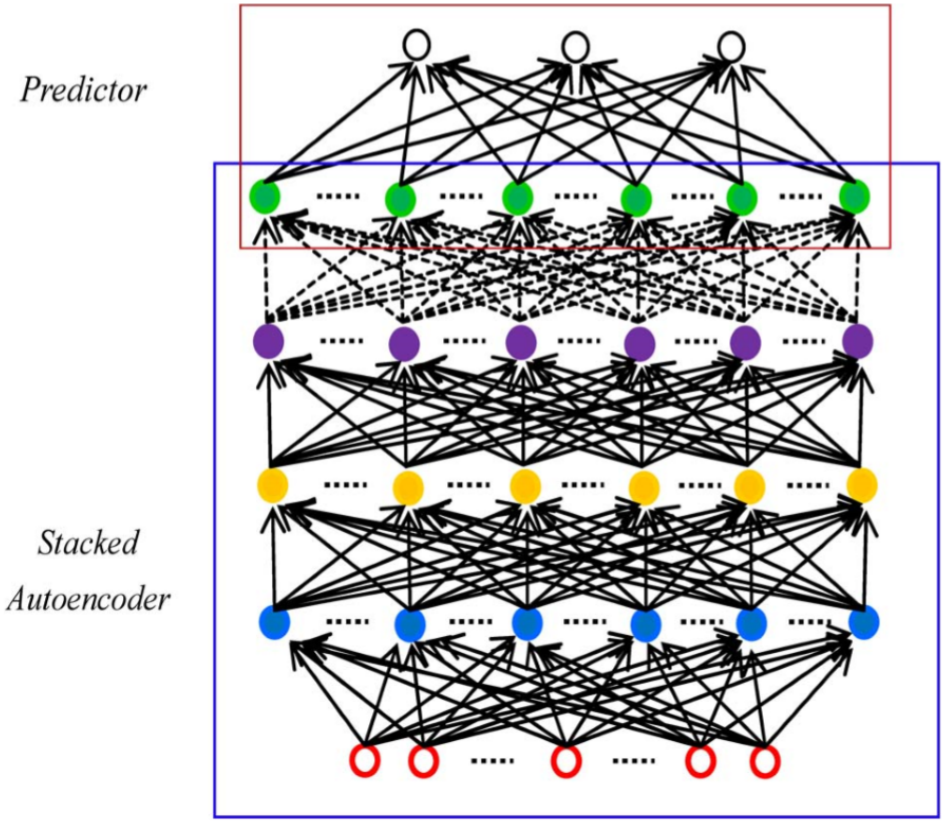
Non-Rush hour



Rush hour

Deep Learning for Traffic Forecasting

- Traffic Prediction using Stacked Autoencoder (SAE) together with logistic regression on top of the network for supervised traffic flow prediction



* Yisheng Lv et al. Traffic flow prediction with big data: A deep learning approach, IEEE TITS, 2015

Deep Learning for Extreme Traffic Prediction

- Main Goal: forecasting traffic for extreme including rush-hour and post-accident

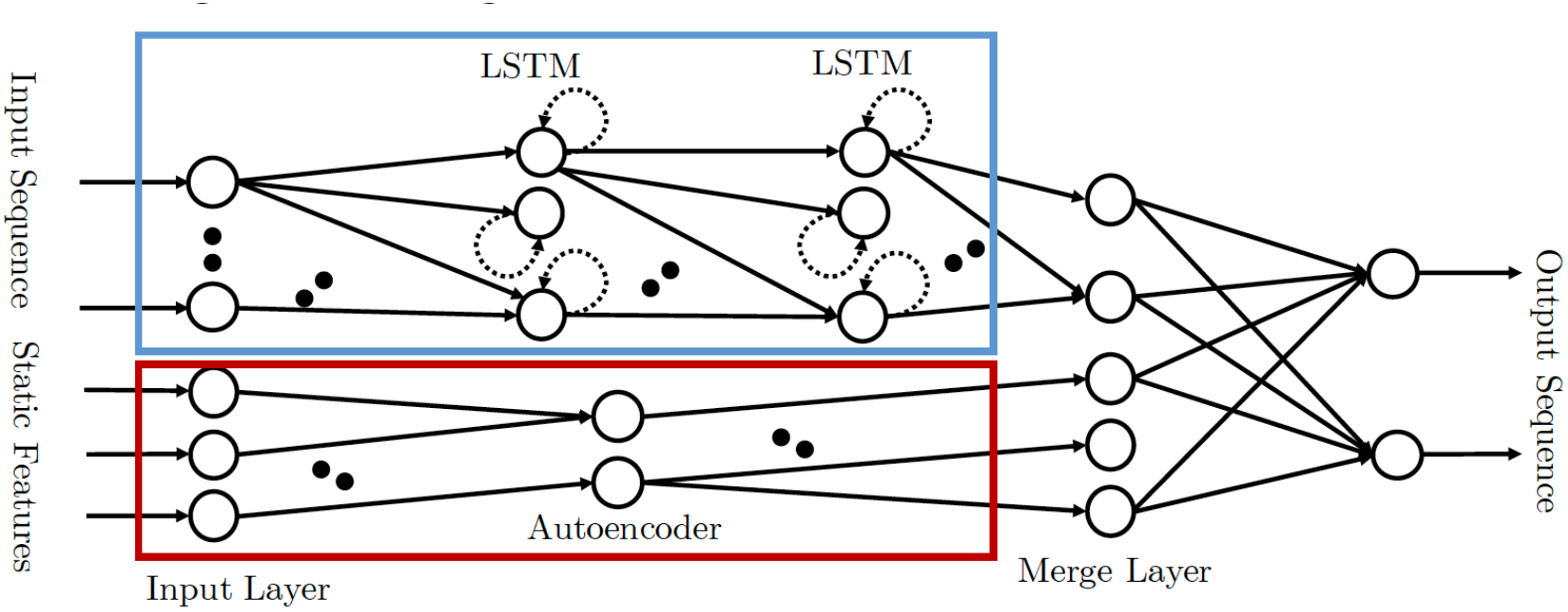
(a) Congestion

(b) Accident

Y. Qi et al, Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. SDM 2016

Deep Mixture LSTM

- Deep Mixture LSTM is a mixture of LSTM and Autoencoder
  - LSTM: normal condition traffic
  - Autoencoder: accident specific features (time, severity)
  - Merge: linear regression

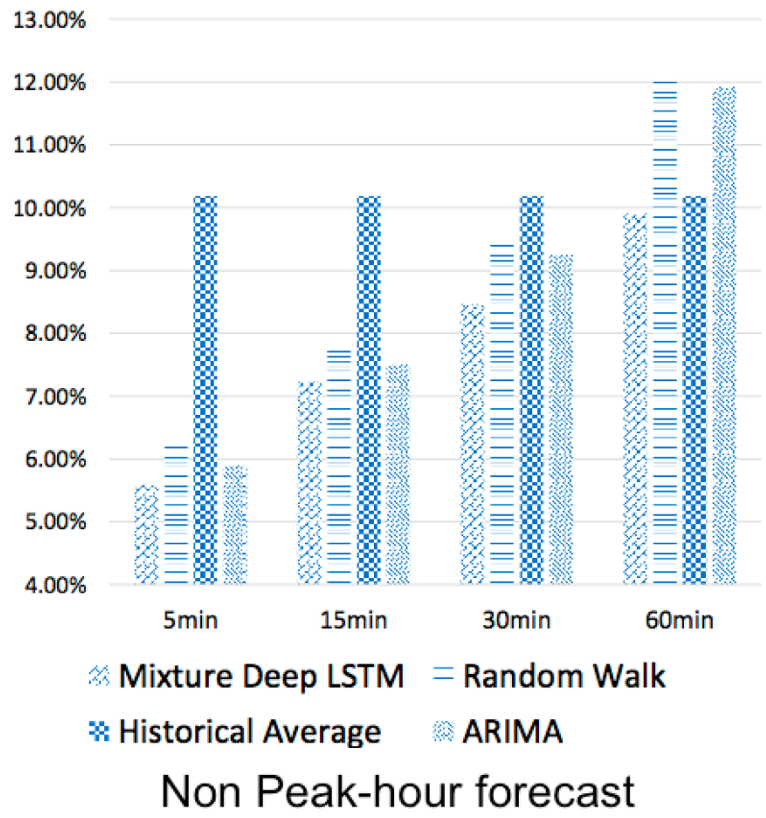


Experiments

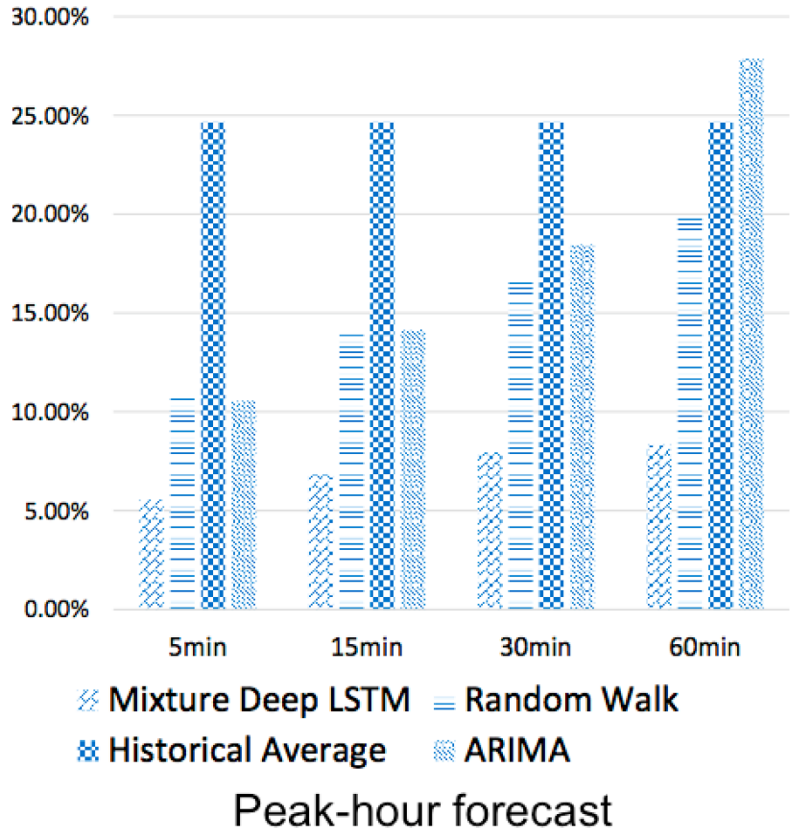
- Baselines:
  - ARIMA: Auto Regressive Integrated Moving Average
  - Random Walk: constant time series with random noise.
  - Historical Average: weighted average of previous seasons
  - Support Vector Regression(SVR): regression using Support Vector Machine
- Data:
  - Traffic: 2,018 sensors from May 19, 2012 to June 30, 2012
  - Accident: 6,811 incidents spread across 1,650 sensors

Rush-Hour Forecasting

- We observe almost 50% improvement for the peak hour traffic forecast



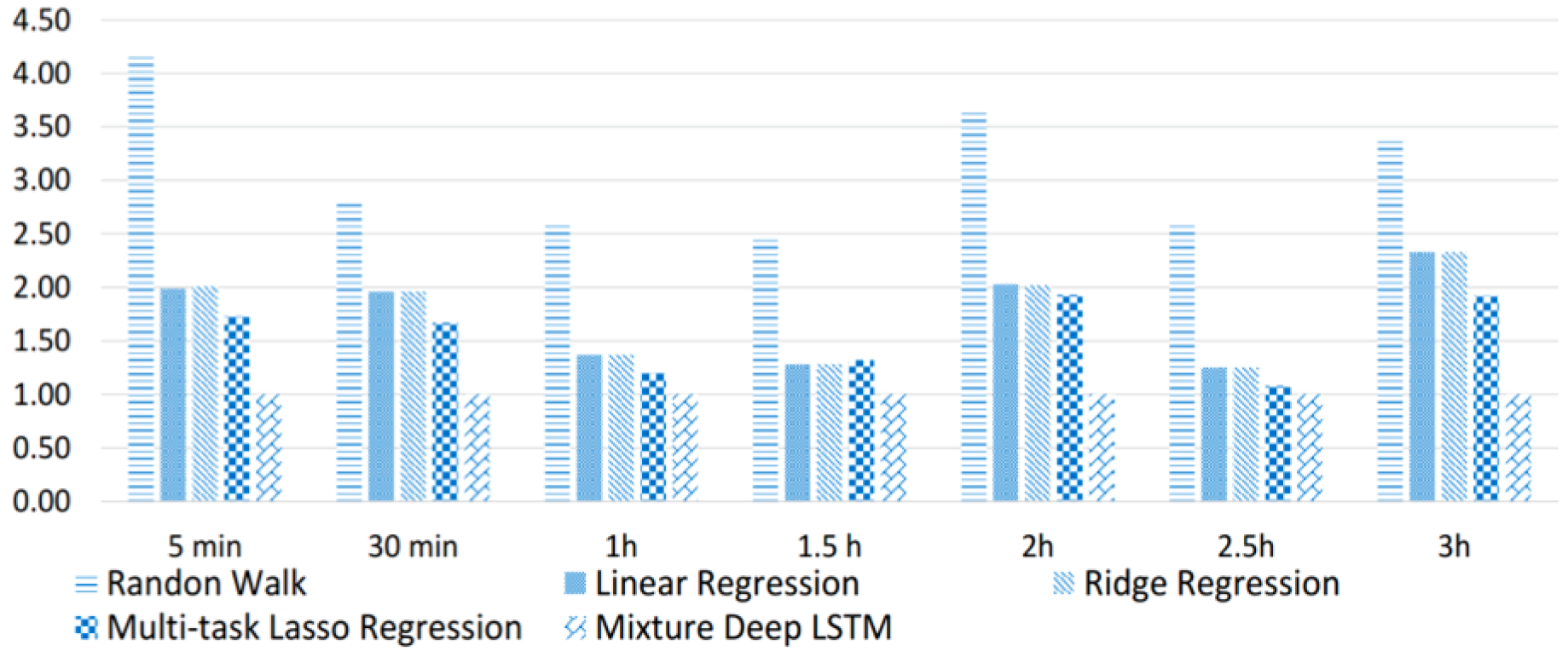
Non Peak-hour forecast



Peak-hour forecast

Post-traffic Forecasting

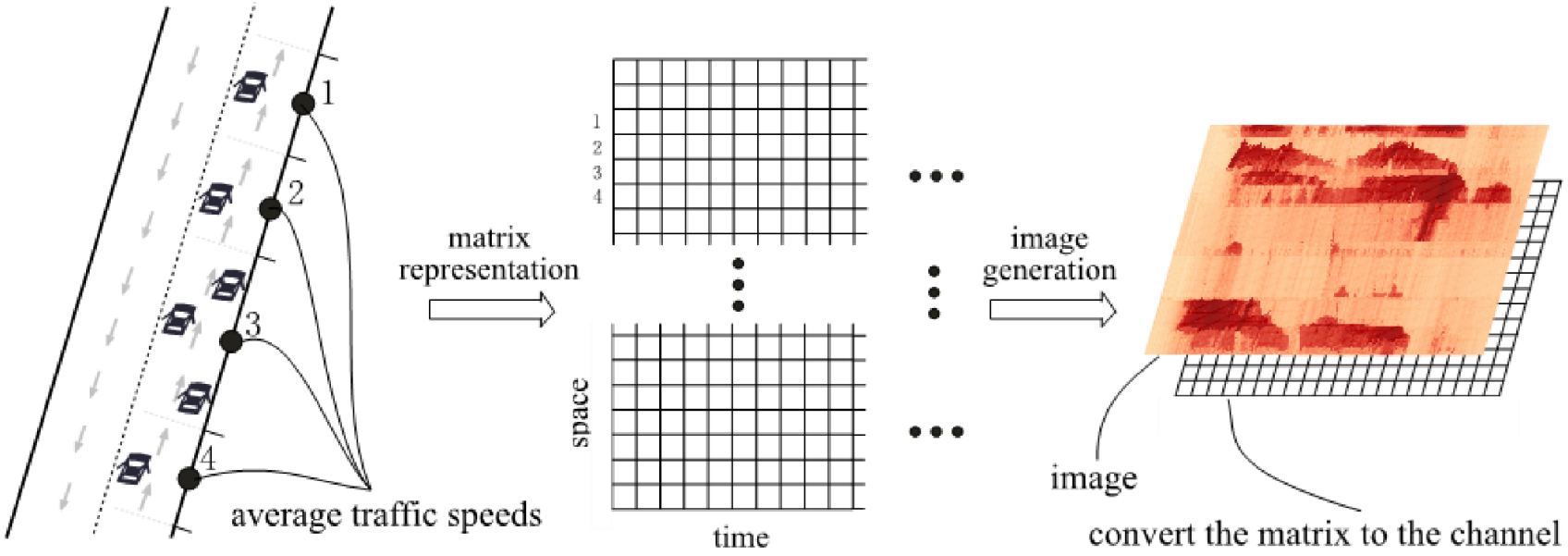
- Deep Mixture LSTM is roughly 30% better than the baseline methods



Post-accident traffic forecasting MAPE of different forecasting horizons

Convolution Neural Network for Traffic Forecasting

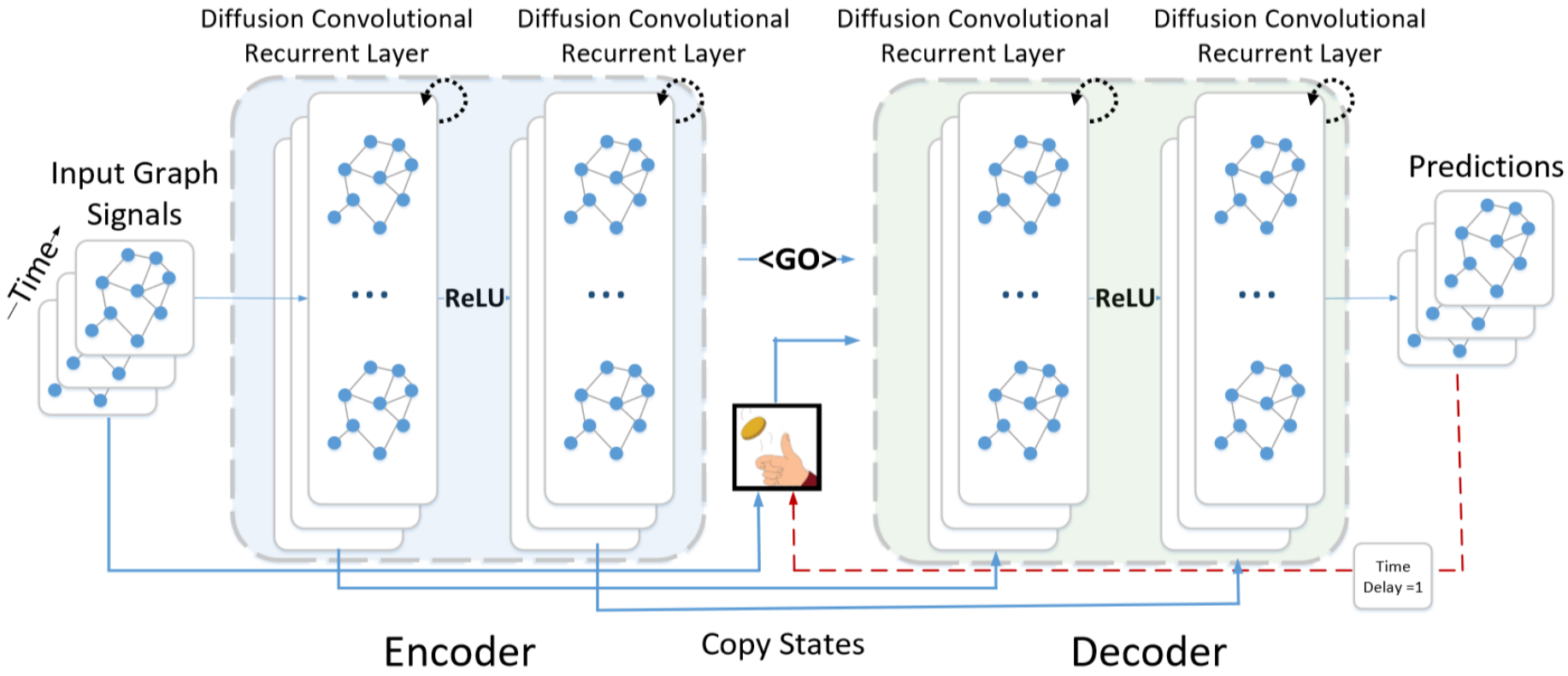
- Model traffic speeds in different locations as a matrix (image).
- Apply convolution to model the spatiotemporal dependency.



* Xiaolei Ma et al. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors, 2017

Traffic Forecasting with Convolution on Graph

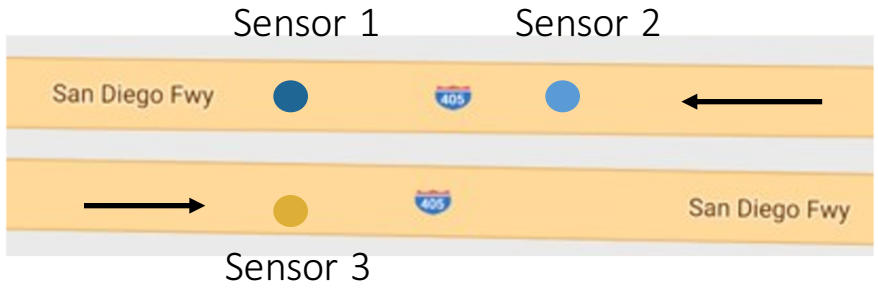
- Model spatial dependency with proposed ***diffusion convolution on graph***
- Model temporal dependency with ***augmented recurrent neural network***



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

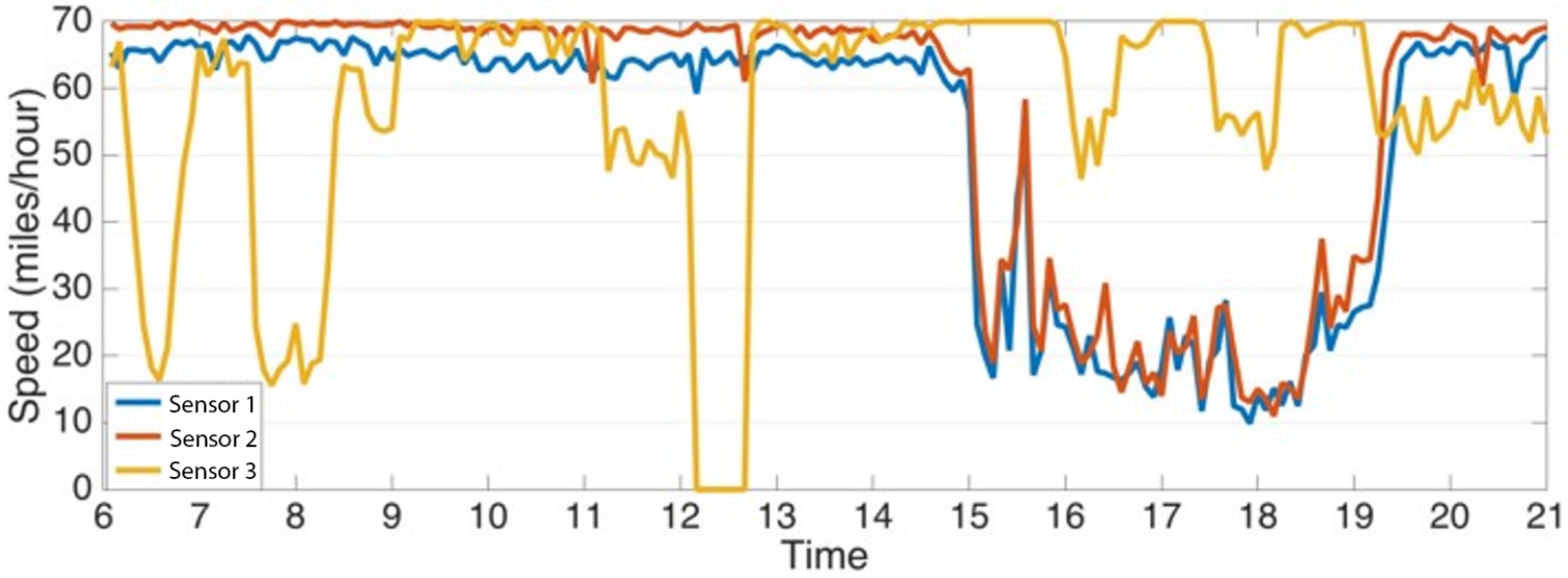
Spatial Dependency in Traffic Prediction

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



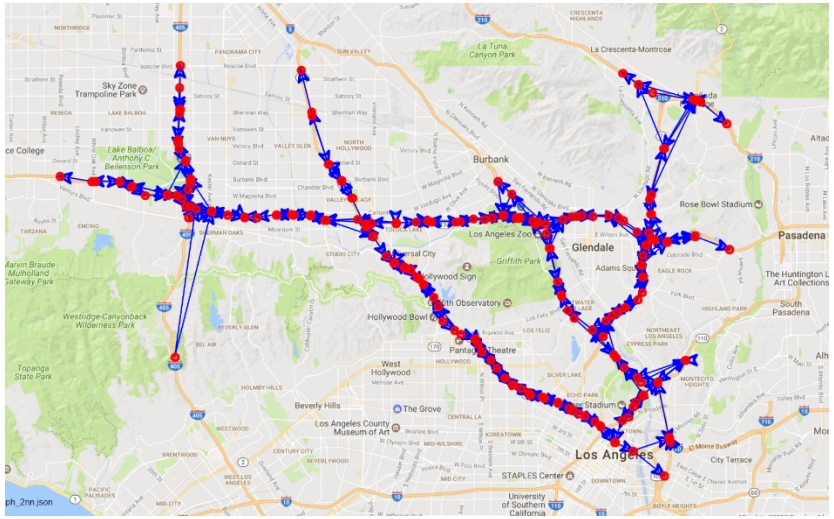
Close in Euclidean space ⇸ Similar traffic speed

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



Spatial Dependency Modeling

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

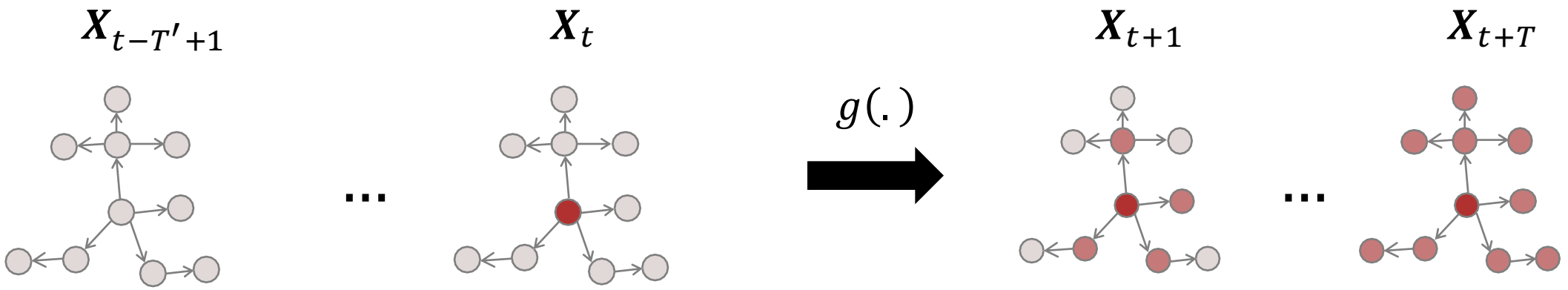


$$A_{ij} = \exp\left(-\frac{\mathrm{dist}_{\mathrm{net}}(v_i, v_j)^2}{\sigma^2}\right) \quad \text{if } \mathrm{dist}_{\mathrm{net}}(v_i, v_j) \leq \kappa$$

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

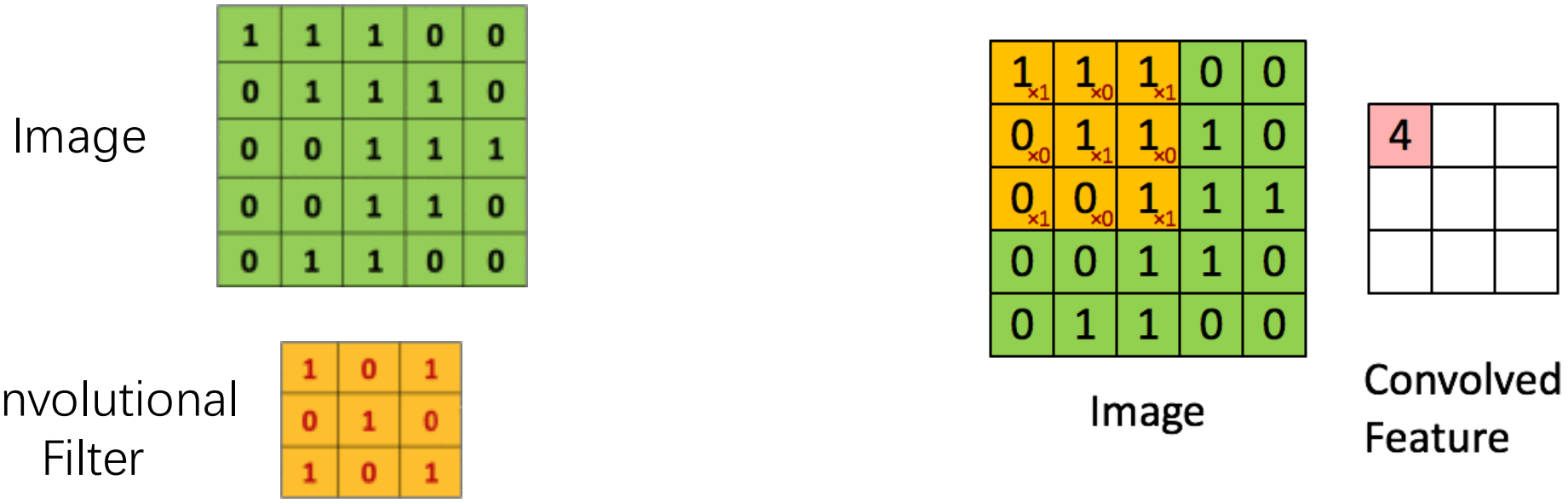
Problem Statement

- Graph signal: $\boldsymbol{X_t} \in \mathbb{R}^{|V| \times P}$, 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



Spatial Dependency Modeling

- Convolution Neural Networks* (CNN) learn meaningful ***spatial patterns***
  - State-of-the-art results on image related tasks



* Y LeCun et al. Gradient-based learning applied to document recognition. Proc. IEEE 1998

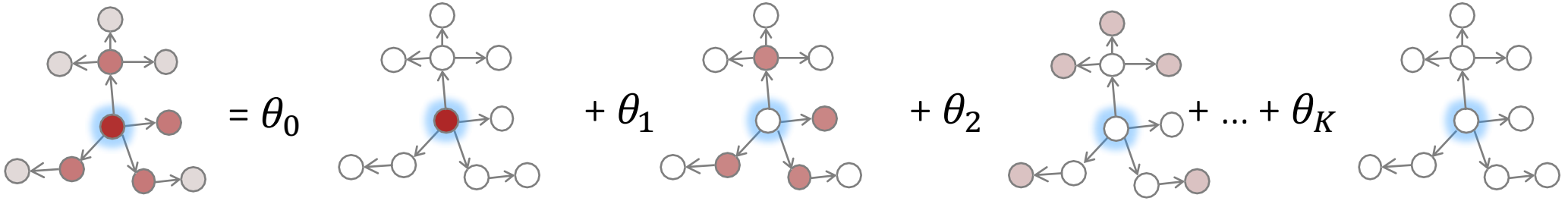
Generalize Convolution to Graph

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

Transition matrices of the diffusion process

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

Learning complexity: $O(K)$



Example diffusion filter Centered at ○

0 Step Diffusion

1 Step Diffusion

2 Step Diffusion

K Step Diffusion

Min Max

Filter weight

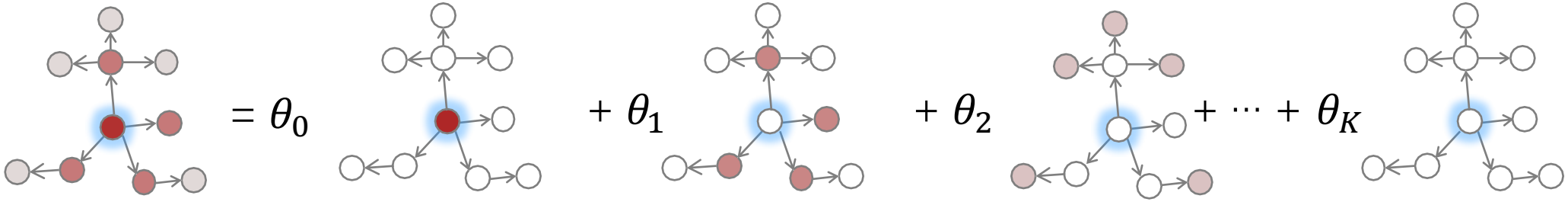
$\star_{\mathcal{G}}$: diffusion convolution, D_o: diagonal out-degree matrix.

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

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



Example diffusion filter Centered at

0 Step Diffusion

1 Step Diffusion

2 Step Diffusion

K Step Diffusion

Min Max

weight

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

Advantage of Diffusion Convolution

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

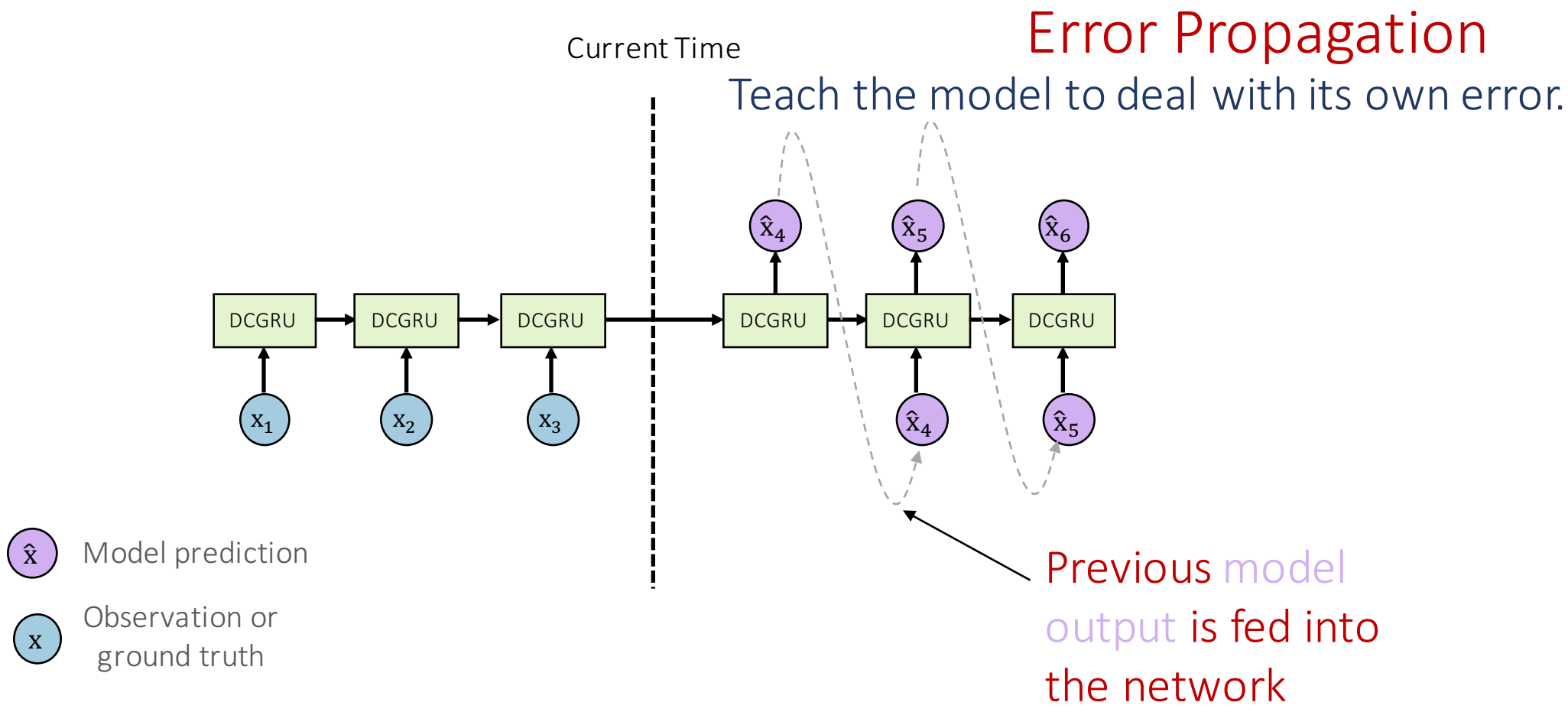
- Efficient
  - Learning complexity: $O(K)$
  - Time complexity: $O(K|E|)$, $|E|$ number of edges
- Expressive
  - Many popular convolution operations, including the ChebNet [Defferrard et al., NIPS '16], can be seen as special cases of the diffusion convolution [Li et al. ICLR '18].

$\star_{\mathcal{G}}$: 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

Model Temporal Dynamics using Recurrent Neural Network

Multi-step ahead prediction with RNN



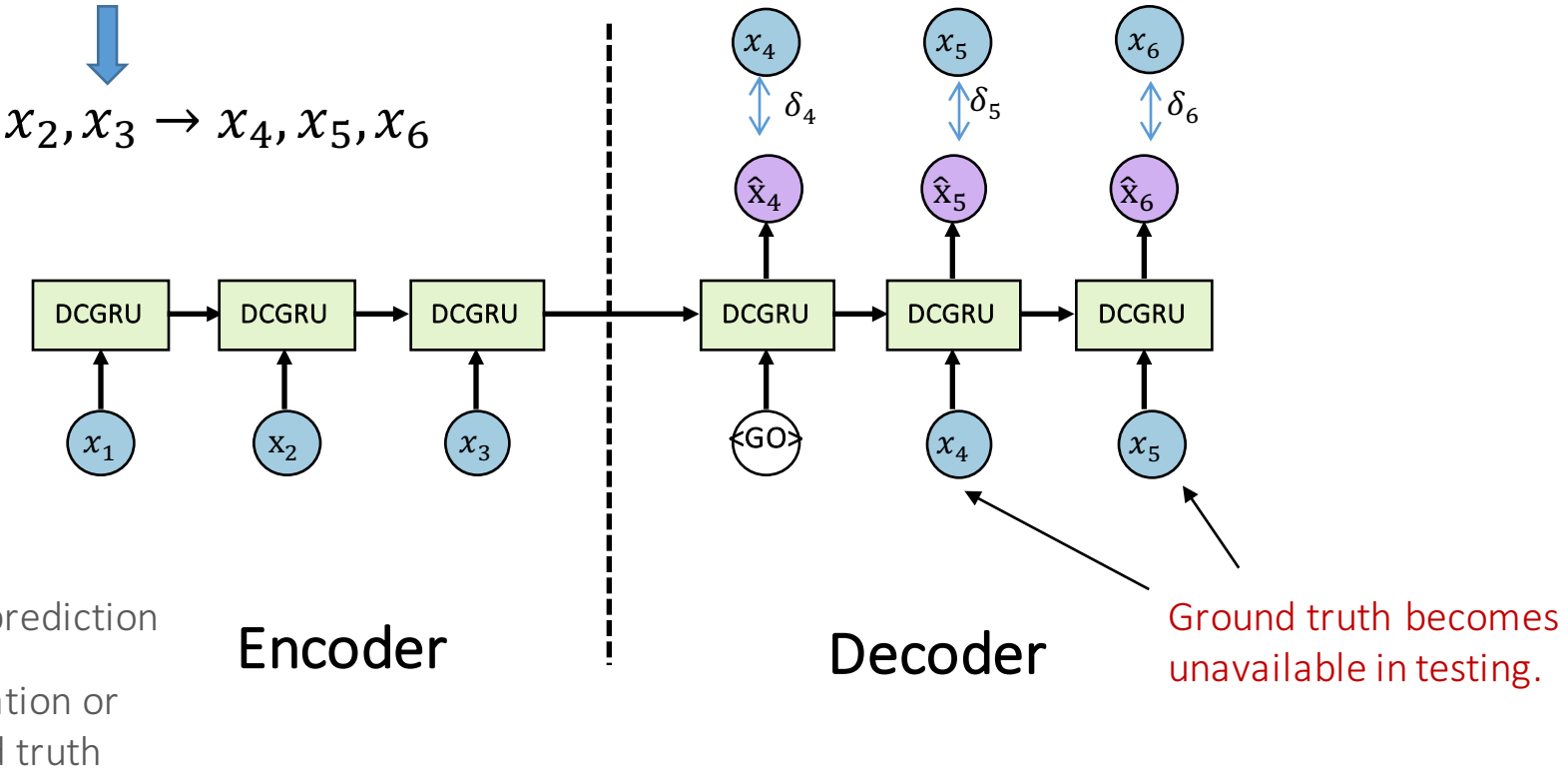
Improve Multi-step ahead Forecasting

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

$x_1, x_2, x_3 \rightarrow x_4$

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

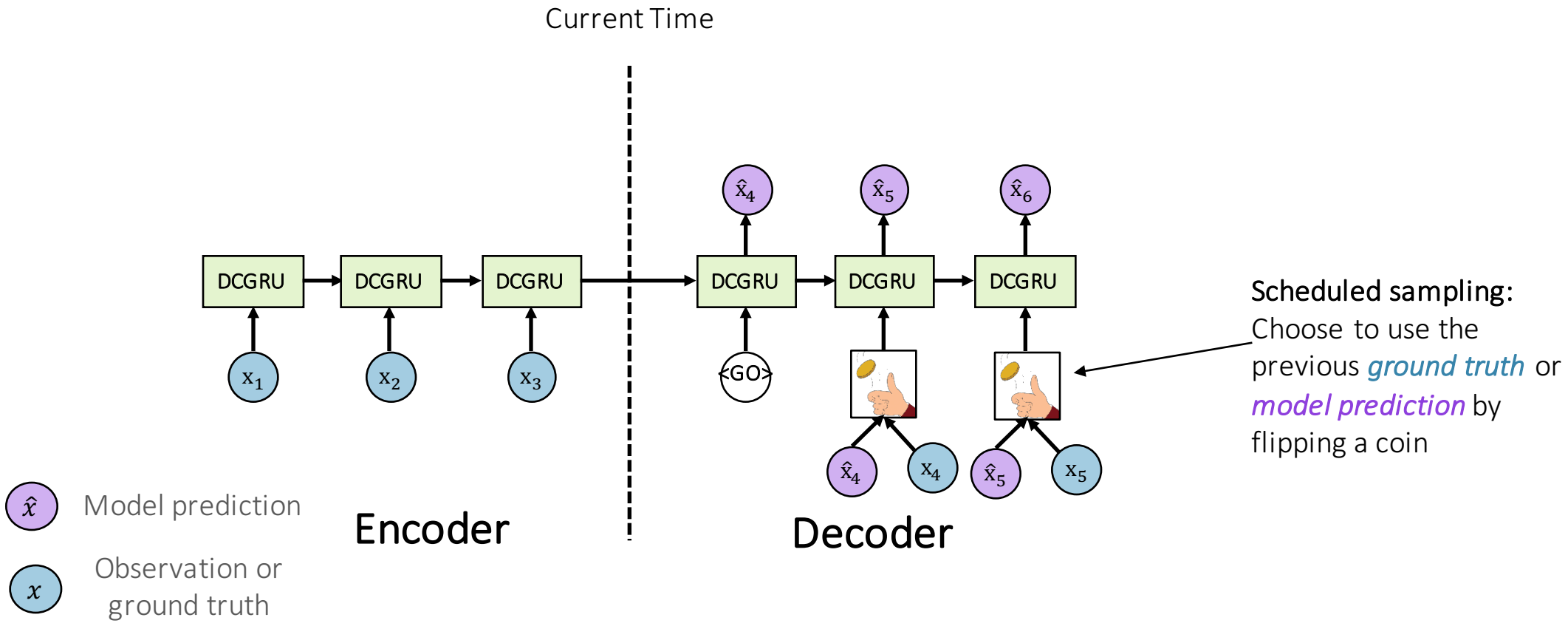
Backprop errors from multiple steps.



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

Improve Multi-step ahead Forecasting

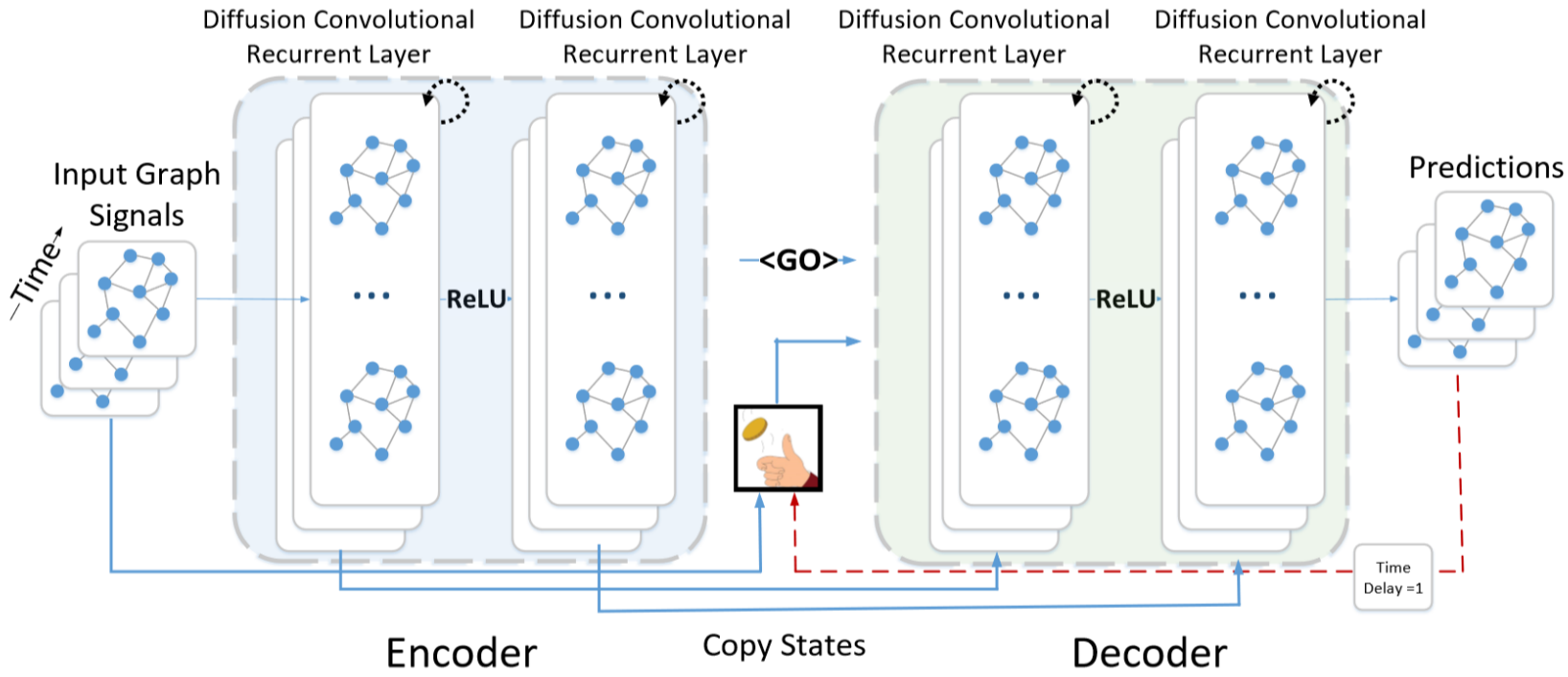
- Improve multi-step ahead forecasting with ***scheduled sampling***



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

Diffusion Convolutional Recurrent Neural Network

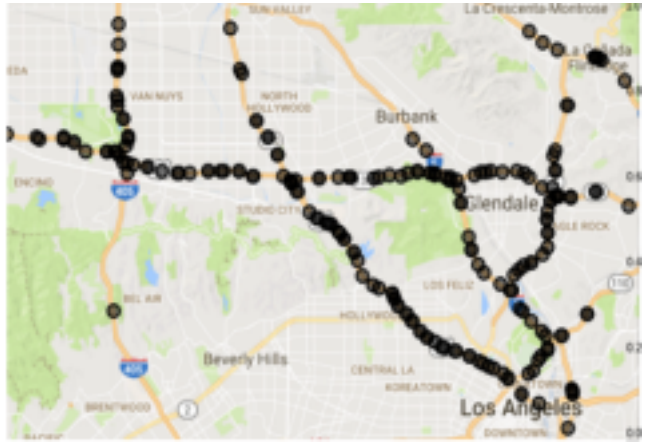
- Diffusion Convolutional Recurrent Neural Network (DCRNN)
  - Model spatial dependency with ***diffusion convolution***
  - Sequence to sequence learning with ***encoder-decoder*** framework
  - Improve multi-step ahead forecasting with ***scheduled sampling***



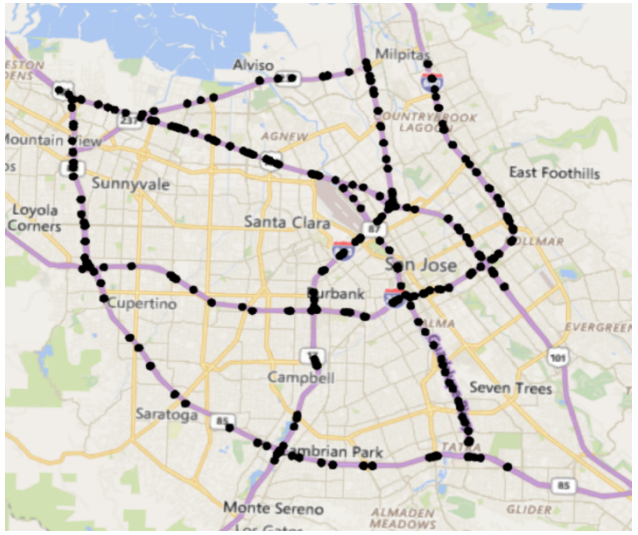
* Yaguang Li et al. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting, ICLR 2018

Experiment - Datasets

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

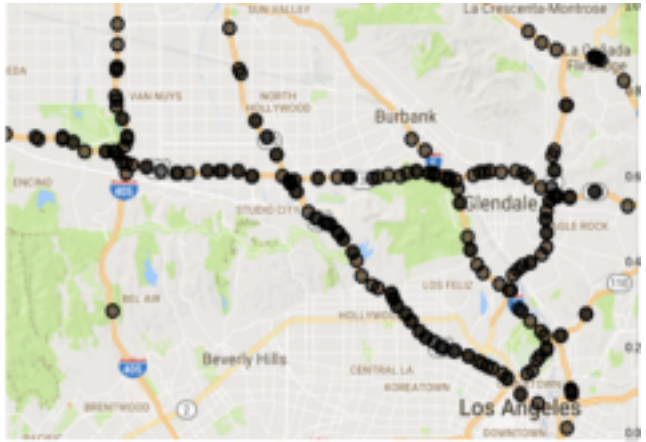


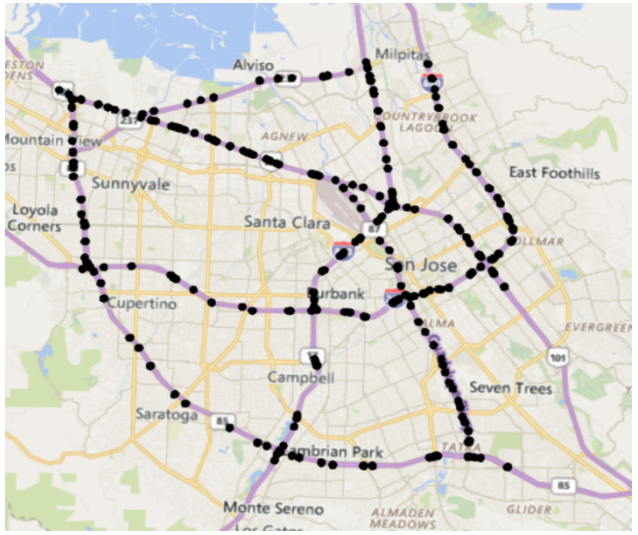
- PEMS-BAY:
  - 345 traffic sensors in Bay Area
  - 6 months in 2017
  - 17M observations



Experiments

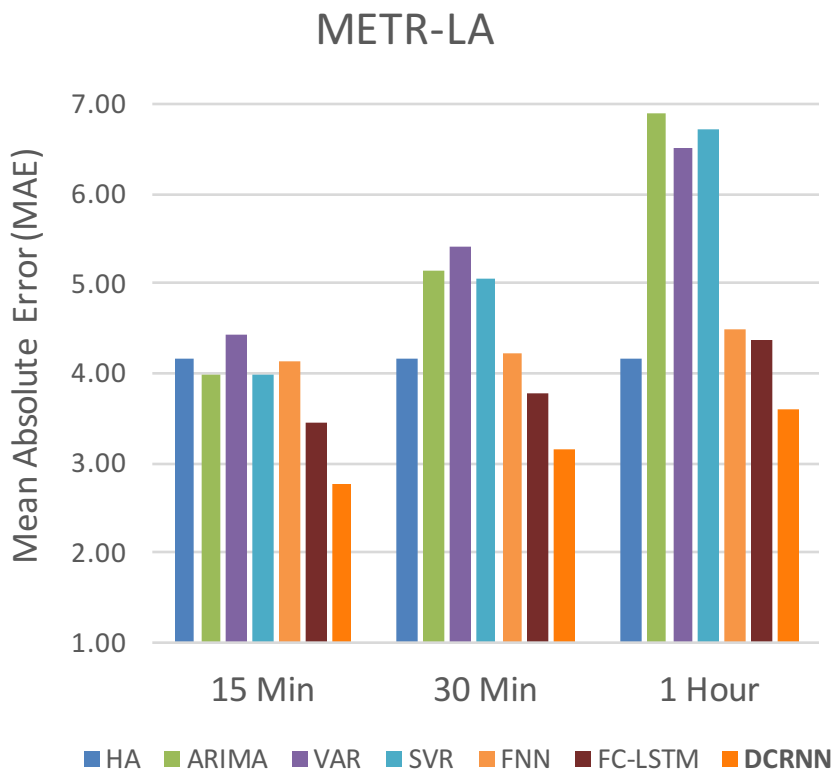
- Baselines
  - Historical Average (HA)
  - Autoregressive Integrated Moving Average (ARIMA)
  - Support Vector Regression (SVR)
  - Vector Auto-Regression (VAR)
  - Feed forward Neural network (FNN)
  - Fully connected LSTM with Sequence to Sequence framework (FC-LSTM)
- Task
  - Multi-step ahead traffic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

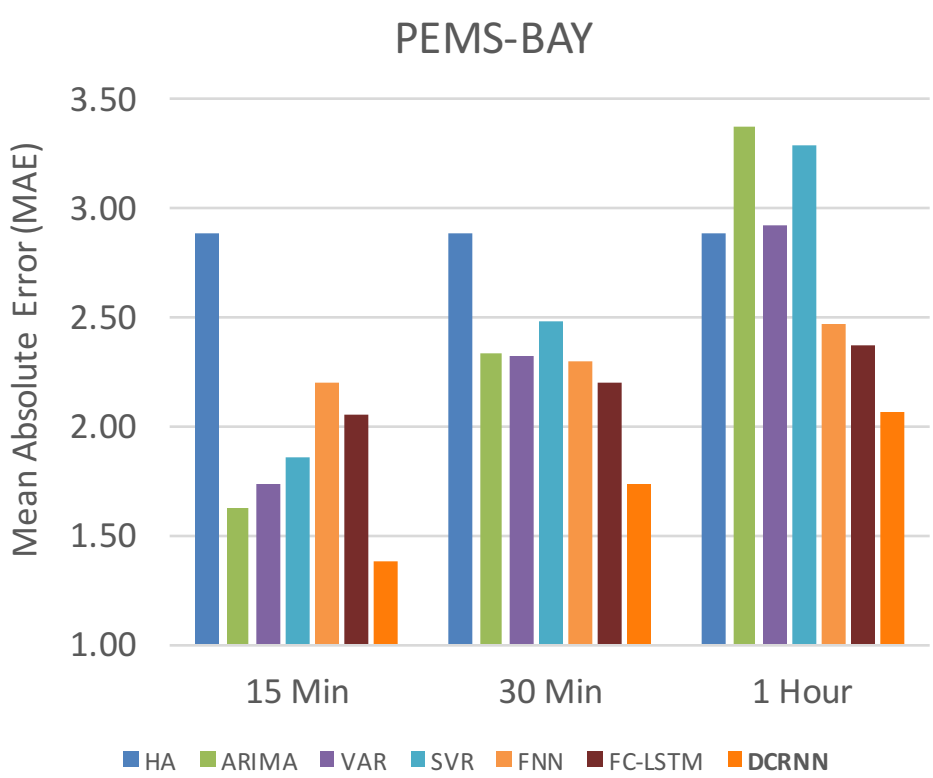




Experimental Results

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

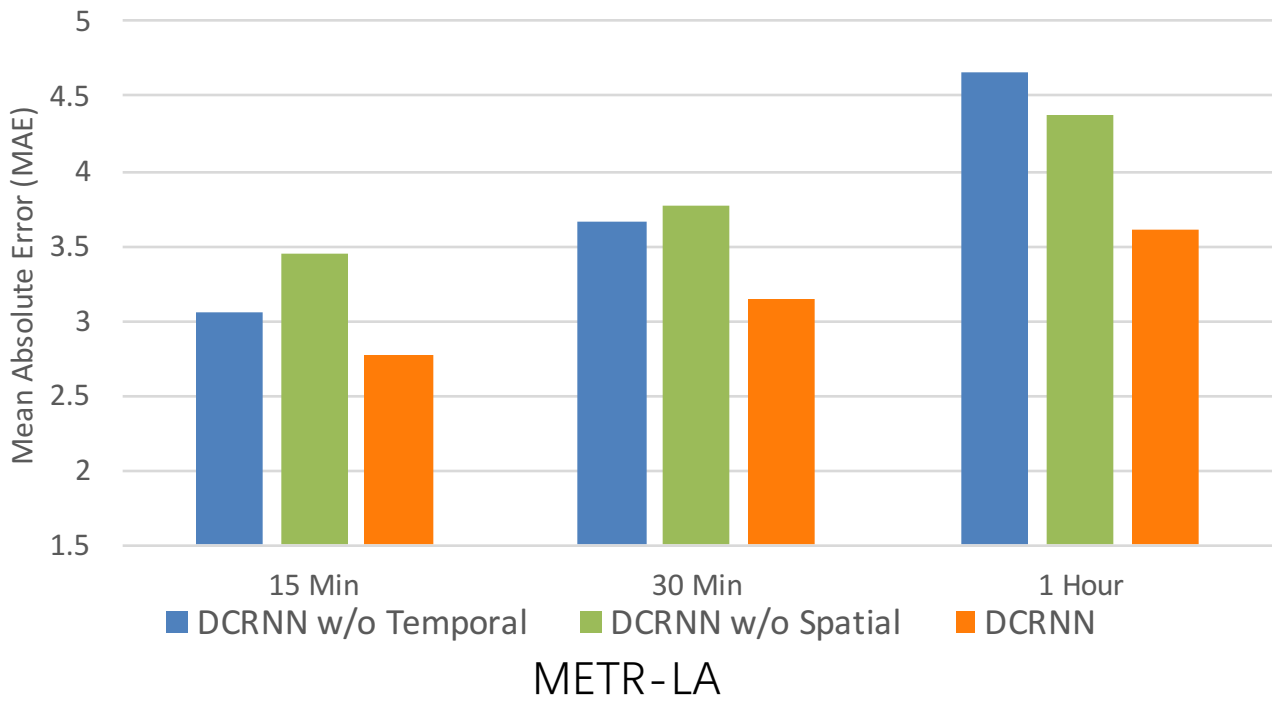




Effects of Spatiotemporal Dependency Modeling

- 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.



Related Work

- Traffic Prediction **without** spatial dependency modeling
  - Simulation and queuing theory [Drew 1968]
  - Kalman Filter: [Okutani et al. TRB'83] [Wang et al. TRB'05]
  - ARIMA: [**Williams et al. TRB'98**] [Pan et al. ICDM'12]
  - Support Vector Regression (SVR): [Muller et al, ICANN' 97] [Wu et al. ITS '04]
  - Gaussian process [Xie et al. TRB'10] [Zhou et al. SIGMOD'15]
  - Recurrent neural networks and deep learning: [Lv et al ITS '15] [Ma et al. TRC'15] [**Yu and Li et al SDM'17**]

Related Work

- Traffic Prediction **with** spatial dependency modeling
  - Vector ARIMA [Williams and Hoel JTE'03], [Chandra et al. ITS'09]
  - Spatiotemporal ARIMA [Kamarianakis et al., TRB'03] [Min and Wynter, TRC'11]
  - k-Nearest Neighbor [**Li et al. ITS'12**] [Rice et al. ITS'13]
  - Latent Space Model [**Deng et al. KDD' 16**]
  - Convolutional Recurrent Neural Network [Ma et al. Sensors'17]
  - Diffusion Convolutional Recurrent Neural Network [**Li et al. ICLR'18**]

Reference

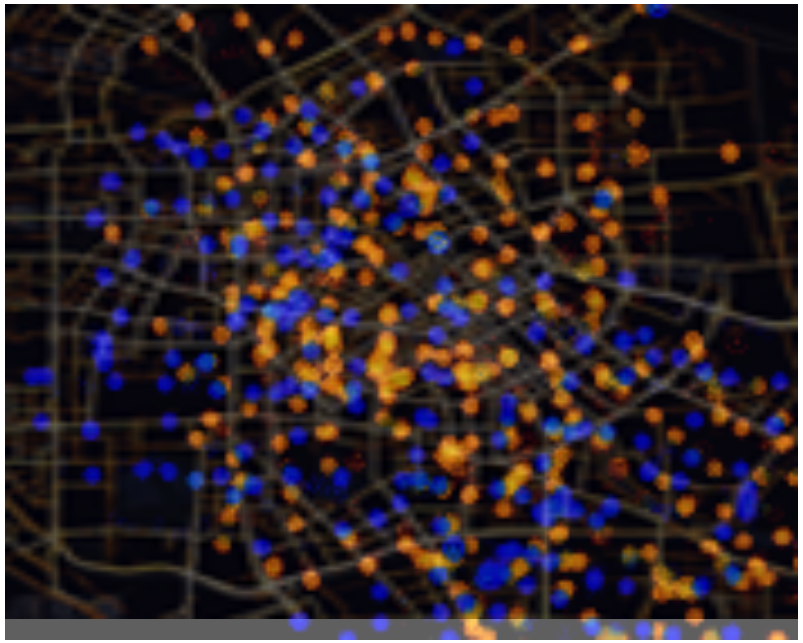
- [Chandra et al. ITS'09] Chandra, S.R. and Al-Deek, H., Predictions of freeway traffic speeds and volumes using vector autoregressive models. ITS, 2009
- [Deng et al. KDD' 16] Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R. and Liu, Y., Latent Space Model for Road Networks to Predict Time-Varying Traffic. KDD, 2016.
- [Drew 1968] Donald R Drew. Traffic flow theory and control. Technical report, 1968.
- [Li et al. ITS'12] Li, S., Shen, Z. and Xiong, G., A k-nearest neighbor locally weighted regression method for short-term traffic flow forecasting, ITS, 2012
- [Yu and Li et al. SDM'17] Li, Y., Yu, R., Shahabi C., Demiryurek, U., Liu, Y., Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting, SDM, 2017
- [Li et al. ICLR'18] Li, Y., Yu, R., Shahabi C., Liu, Y., Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting. ICLR, 2018
- [Lv et al ITS '15] Lv, Y., Duan Y., Kang W., Li, Z., Wang, F., Traffic Flow Prediction With Big Data: A Deep Learning Approach, ITS, 2015.
- [Ma et al. TRC'15] Ma, X., Tao Z., Wang, Y., Yu, Y., Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, TRC 2015
- [Ma et al. Sensors'17] Ma, X., Dai, Z., Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction, Sensors 2017
- [Muller et al, ICANN' 97] Müller, K.R., Smola, A.J., Rätsch, G., Schölkopf, B., Kohlmorgen, J. and Vapnik, V., Predicting time series with support vector machines. ICANN, 1991

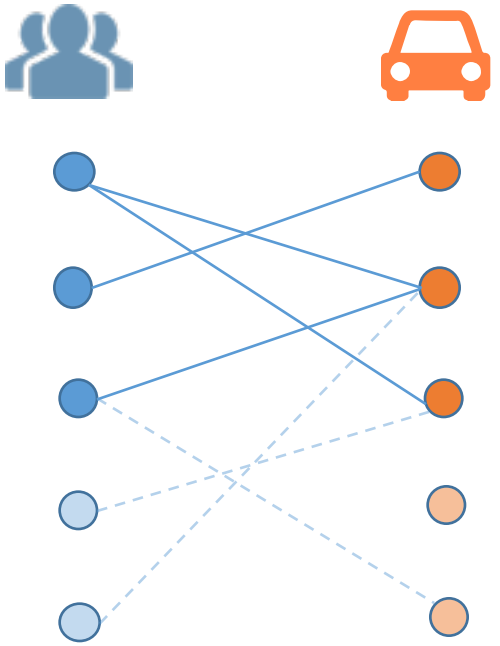
Reference

- [Min and Wynter, TRC'11] Min, W., Wynter, L., Real-time road traffic prediction with spatio-temporal correlations, RTC, 2011
- [Okutani et al. TRB'83] Okutani, I, and Y. J. Stephanides. Dynamic Prediction of Traffic Volume Through Kalman Filtering Theory. Transportation Research B, 1984
- [Pan et al. ICDM'12] Pan, B., Demiryurek, U. and Shahabi, C., Utilizing real-world transportation data for accurate traffic prediction. ICDM, 2012
- [Wang et al. TRB'05] Wang, Y., Papageorgiou, M., Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. Transportation Research Part B: Methodological, 2015
- [Williams and Hoel JTE'03] Williams, B.M., Hoel, L.A., Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of transportation engineering, 2003
- [Wu et al. ITS '04] Wu, C.H., Ho, J.M. and Lee, D.T., Travel-time prediction with support vector regression. IEEE transactions on intelligent transportation systems, 2004
- [Xie et al. TRB'10] Xie, Y., Zhao, K., Sun, Y. and Chen, D., Gaussian processes for short-term traffic volume forecasting. Transportation Research Record, 2010
- [Zhou et al. SIGMOD'15] Zhou, J. and Tung, A.K., 2015, May. Smiler: A semi-lazy time series prediction system for sensors. In Proceedings of the, ACM SIGMOD, 2015

Decision Service

Intelligent Order-Dispatching System





Order and Driver Forecasting

A Large-Scale Distributed Computing

Platform Efficiency and Customer Experience Optimization

Map Service

Route Planning

The Core of Dispatching

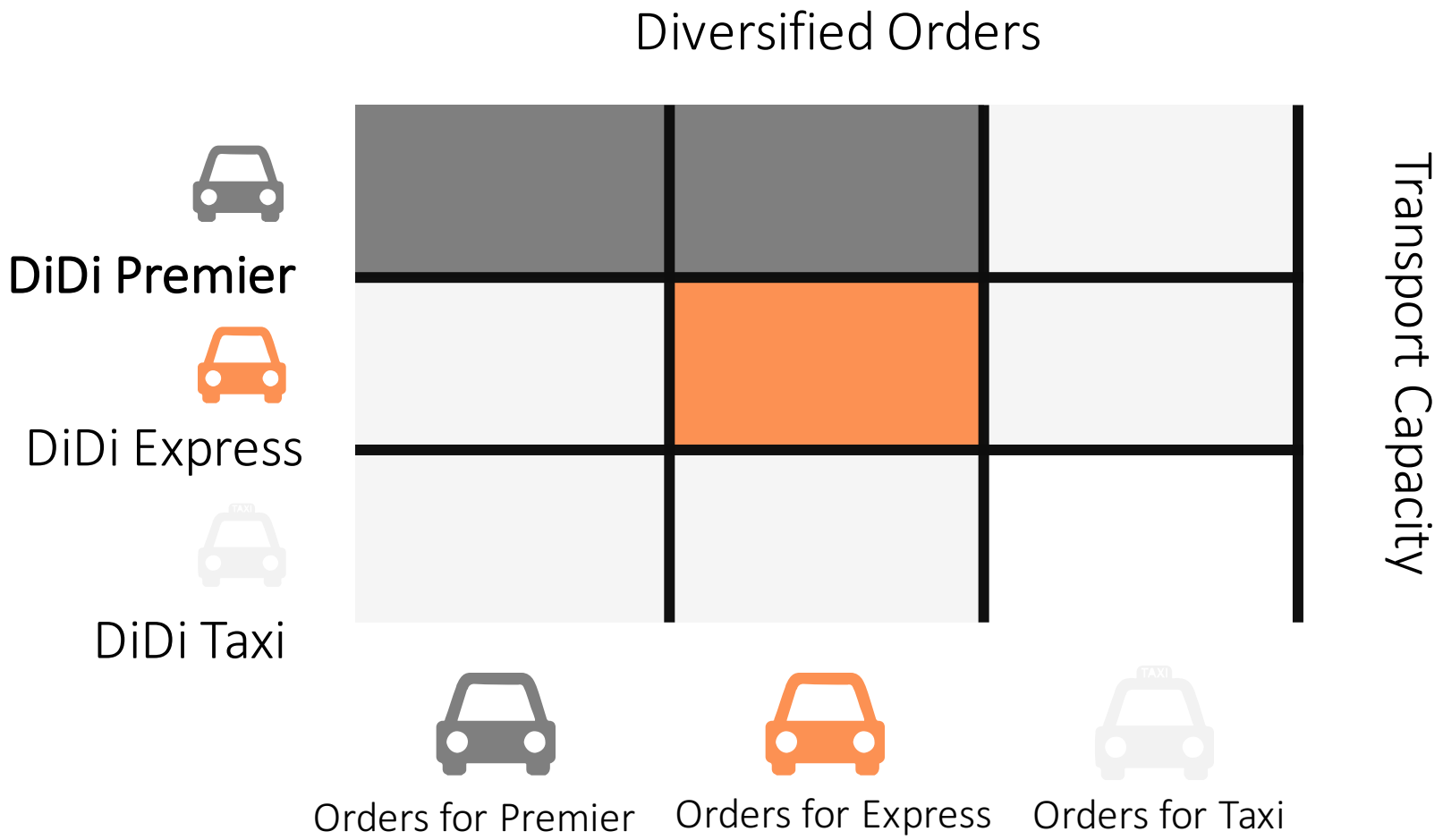
- To minimize cost
- To maximize driver efficiency
- To optimize transportation efficiency

ETA

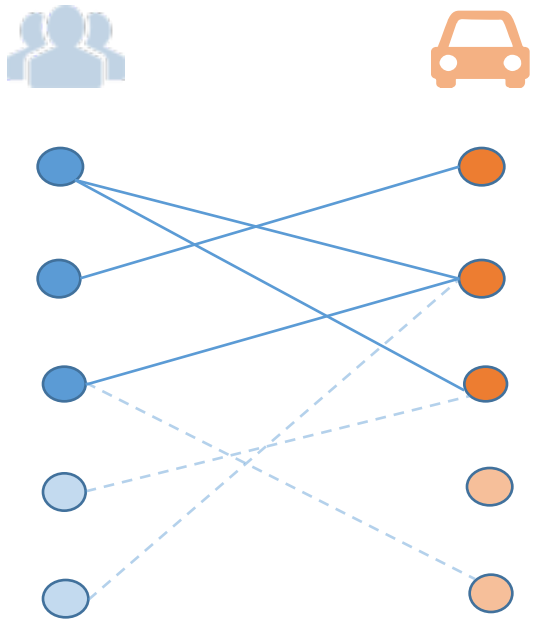
(Estimated Time of Arrival)

- To estimate the traveling time and the waiting time of each ride

Order-Dispatching Matrix



Bipartite Graph Matching



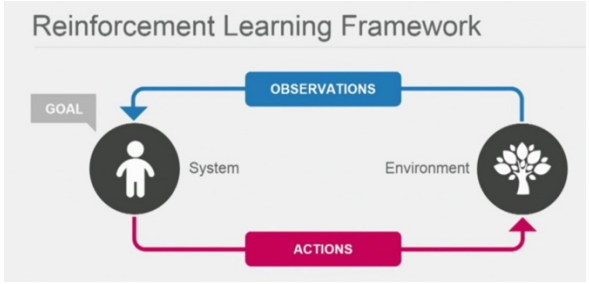
Hungarian matching algorithm (also called the Kuhn-Munkres algorithm)

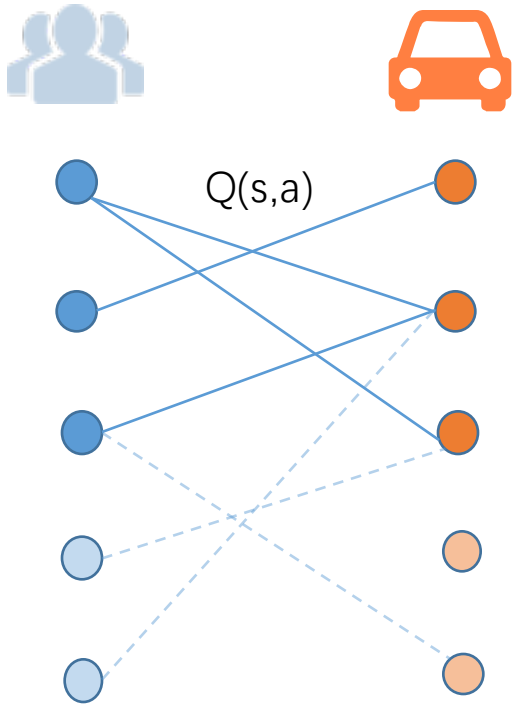
MDP Formulation for Dispatching

Reinforcement Learning

Dispatching

- Every dispatching decision affects future supply (drivers) distribution
- Maximize drivers' collective income through optimized dispatching, while ensuring good customer experience



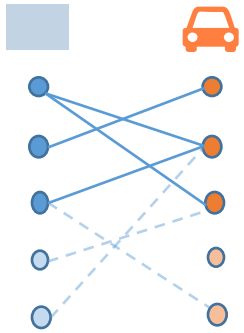


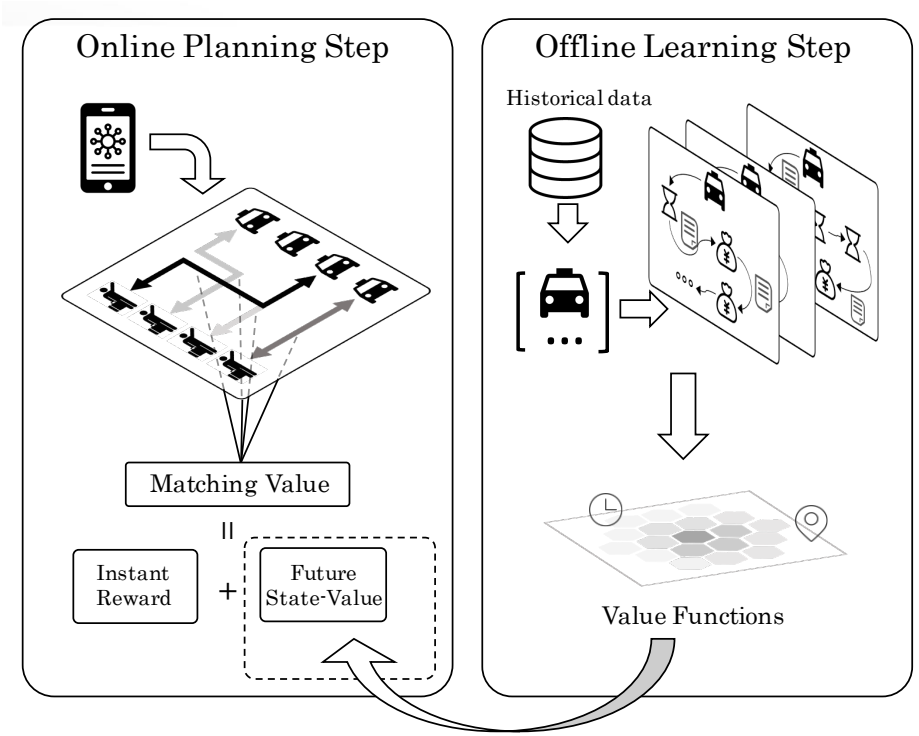
Reinforcement Learning

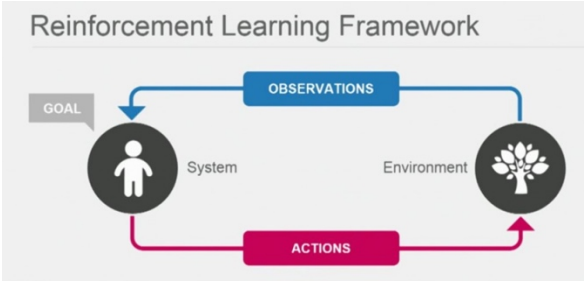
- Focus on long-term reward (e.g. one day)
- Consider future impact of current decision
- Value function is a key quantity to learn

Method

Combining Reinforcement Learning and Combinatorial Optimization







Xu et al., KDD 2018

MDP Definition

Motivation

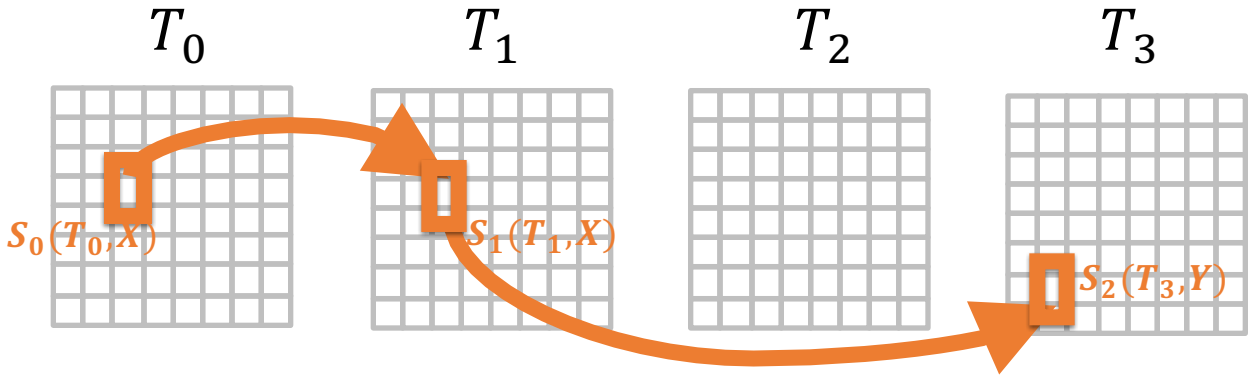
- Each order dispatch accords to a decision made by the platform
- Formulate order dispatch into a Markov Decision Process (MDP)
- Optimal policy generates the optimal revenue of the platform – by satisfying more requests and maximizing driver's income

- $\max V_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$

- Agent: A Driver
- State: Time & Space location
- Action: Pickup an order or do nothing
- Reward: Driver's Income (order price)
- Episode: A whole day from 04:00 to 04:00 (+1)
- Value Function: The expected future income of a driver in a state S

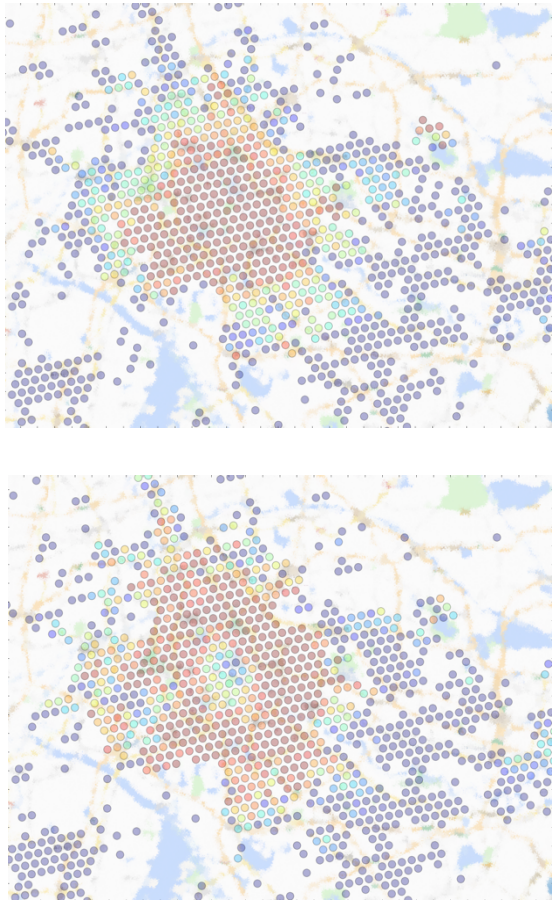
Learning – Policy Evaluation

Dynamic Programming

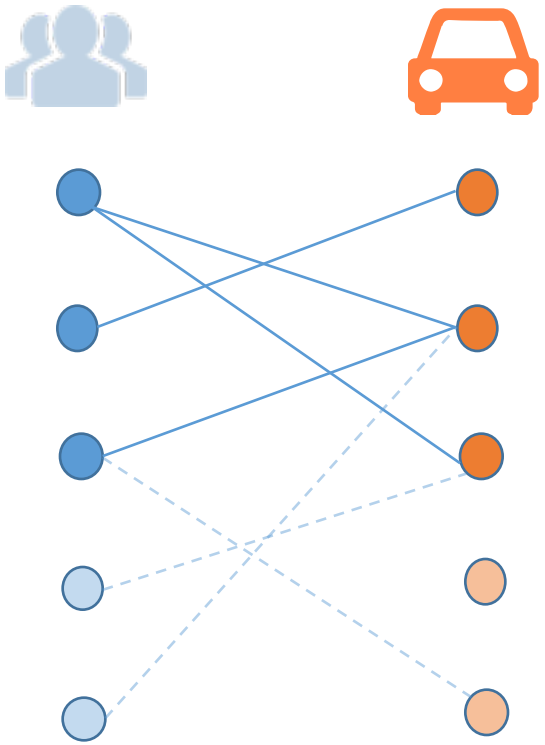


Vacant : $V_\pi(S_0) \leftarrow V_\pi(S_0) + \alpha(0 + \gamma V_\pi(S_1) - V_\pi(S_0))$

Serving : $V_\pi(S_1) \leftarrow V_\pi(S_1) + \alpha(R + \gamma^2 V_\pi(S_2) - V_\pi(S_1))$



Planning – Advantage Function



Link weight:

$$Q(s,a) \rightarrow A(s,a)$$

$$= R. + V(s') - V(s)$$

Order's value

Value function of driver's expected finishing state

Value function of driver's current state

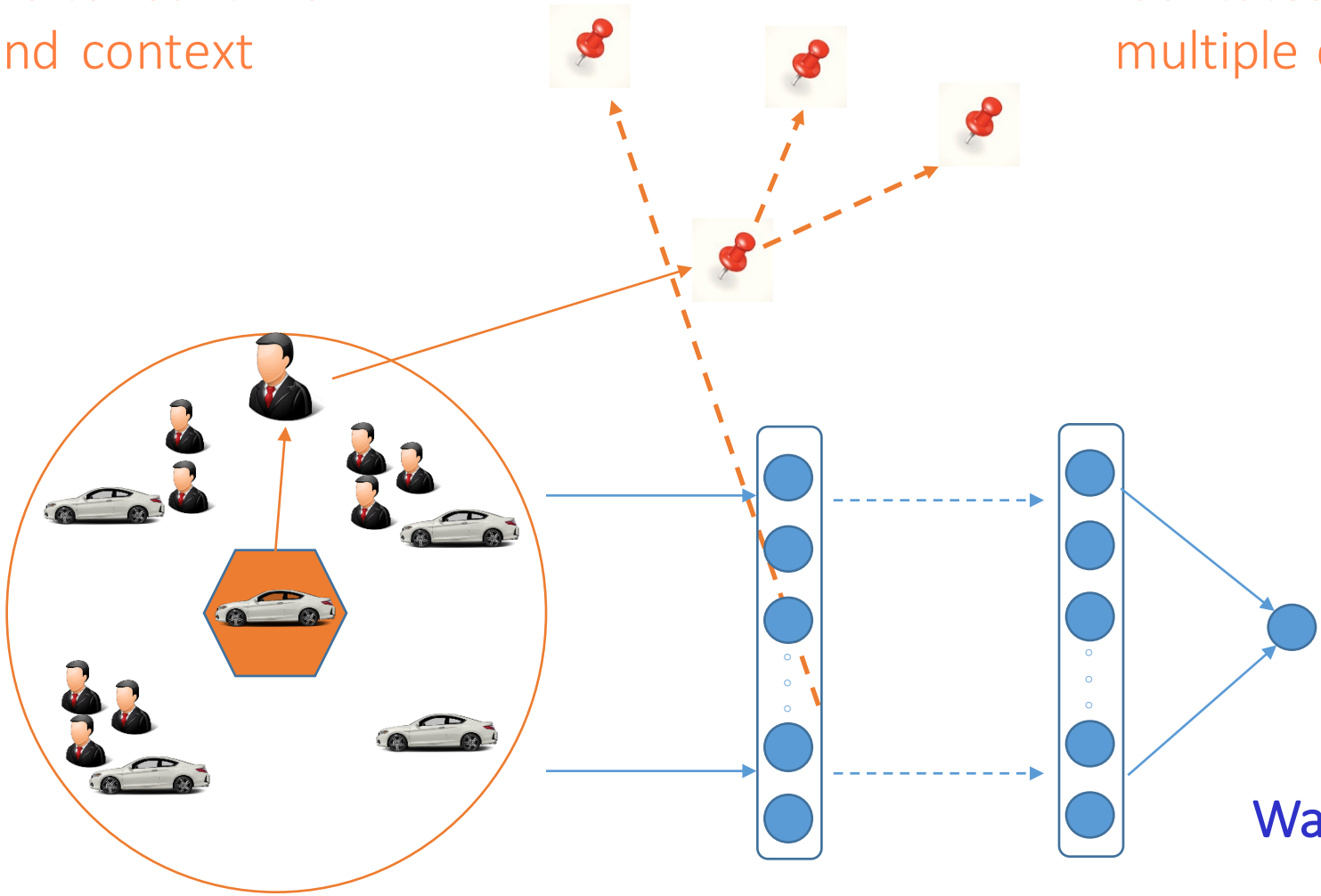
KDD2018

Deep RL for Dispatching

Deep Reinforcement Learning

more adaptive to real-time supply-demand context changes

facilitates learning from multiple cities and times



weights sharing among inputs: location, time, destination, context - better generalization

Wang et al., ICDM 2018

Deep Q-network with action search

- ❑ Training data
- ❑ Action search
- ❑ Expanded action search

- Use historical trips data as training transitions.
- Each trip x defines a transition of the agent's state

  (s0 , a, r, s1):
  - Current state s0 := (l0 , t0, f0), location, time, and contextual features
  - Action: the assigned trip;
  - Next-state s1 := (l1 , t1, f1)
  - Reward: the total fee collected for the trip.
- Implicitly consider pick-up time, trip duration relative to reward

Deep Q-network with action search

- ❑ Training data
- ❑ Action search
- ❑ Expanded action search

- Construct an approximate feasible space for the actions for computing $\max_{a' \in A} Q(s_1, a')$ for the targets
- Instead of searching through all valid actions, we search within the historical trips originating from the vicinity of s:

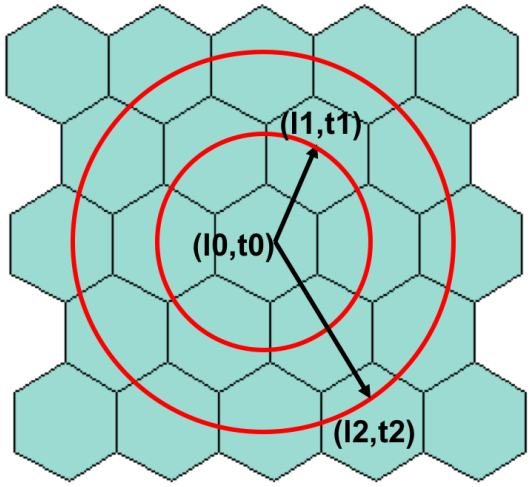
$$\tilde{\mathcal{A}}(s) := \{x_{s_1} | x \in \mathcal{X}, B(x_{s_0}) = B(s)\}$$

- The same search procedure is used for evaluation, where we simulate the driver's trajectory during the day using historical trips data.

Deep Q-network with action search

- ❑ Training data
- ❑ Action search
- ❑ Expanded action search

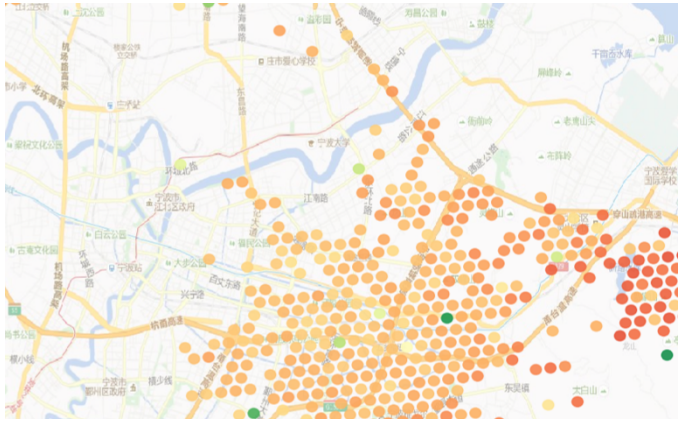
- Due to training data sparsity in certain spatio-temporal regions, we perform an expanded action search in both spatial and temporal spaces.



Training for multiple cities

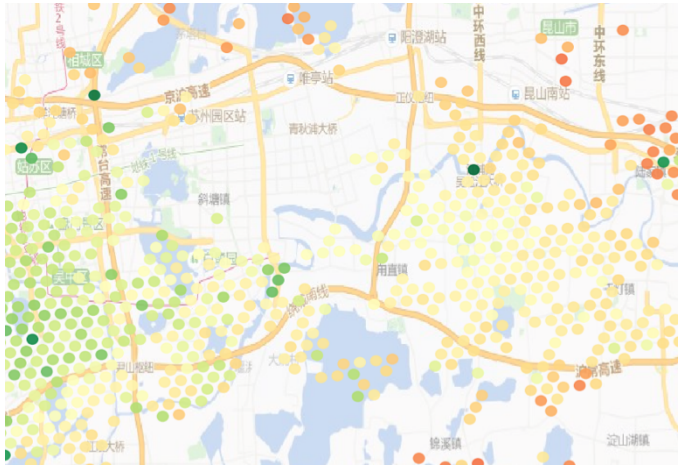
Dispatching system supports a large number of cities

- Computationally intensive
- Diverse supply-demand settings



Knowledge transfer

- Leverage common properties: e.g. rush-hour traffic pattern
- Improve learning efficiency



Wang et al., ICDM 2018

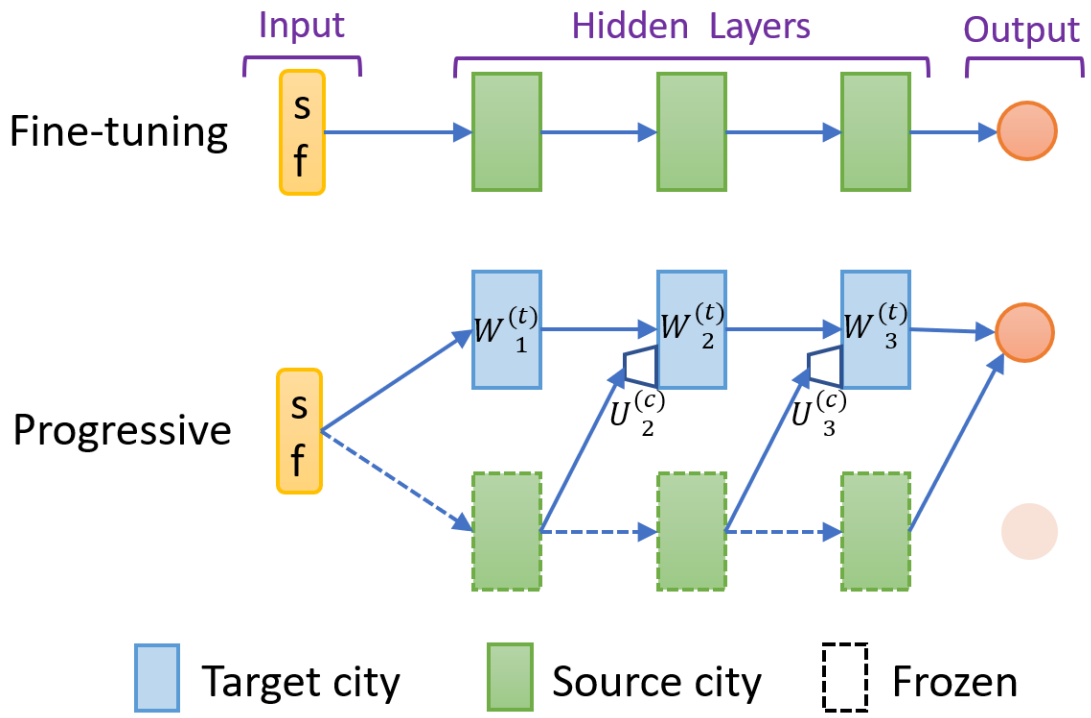
Transfer Learning

Idea

- Re-use weights
- Better initial solution

Existing Methods

- Fine-tuning
- Progressive: lateral connections between source city nodes and target city nodes



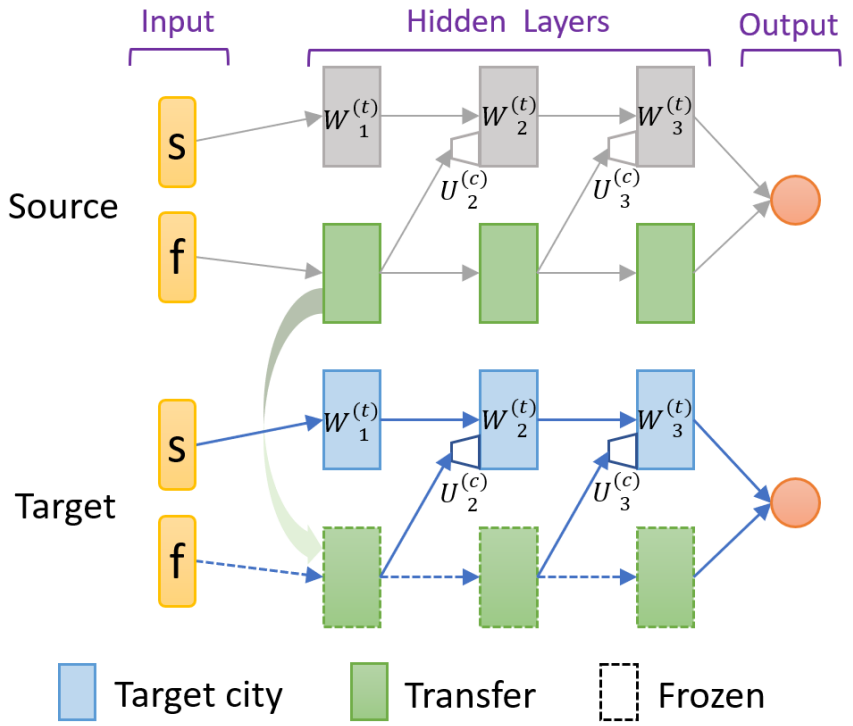
Correlated Feature Progressive Transfer (CFPT)

Spatio-temporal displacement

- Involve relative transition pattern like distance and time cost

General online features

- Number of idle drivers
- Number of orders created
- Average pick-up time for passengers
- …



Experiments

Training data: one month of Express Car trip data

Single-driver test environment

DQN v.s. policy evaluation

- Policy evaluation: max-Q -> mean-Q in mini-batch updates

Transfer learning v.s. no-prior DQN

Spatial transfer

- Source city to target city

Temporal transfer

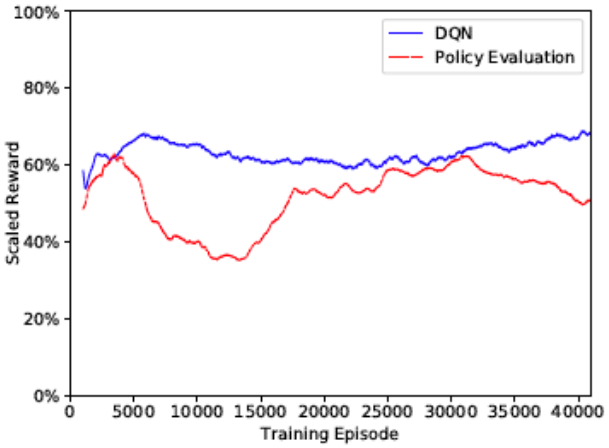
- Same city: one time period to another

City	Size	Region
A	Large	Northern
B	Small	Southern
C	Medium	Southern
D	Large	Western

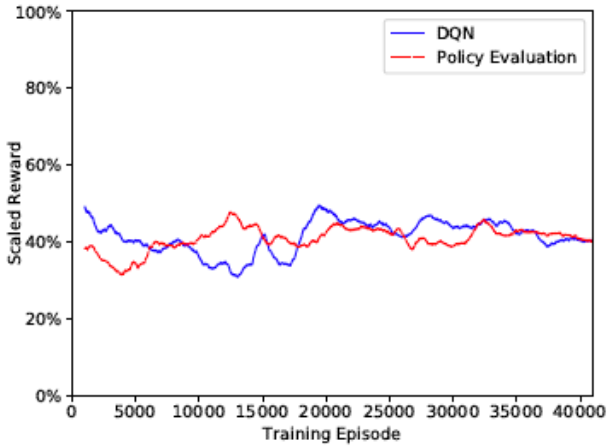
Results

DQN v.s. policy evaluation

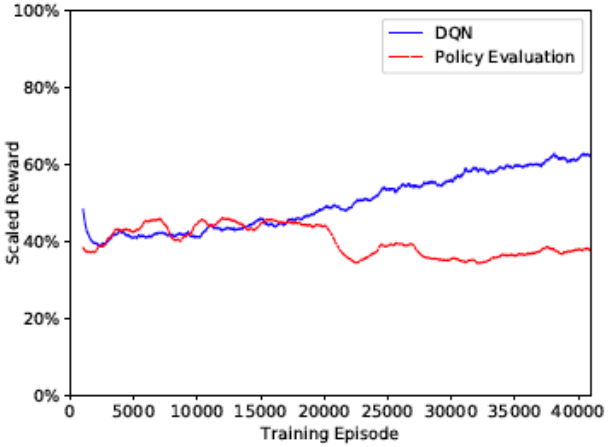
- Optimization helps
- Smaller cities with simpler pattern and fewer transitions have smaller advantages



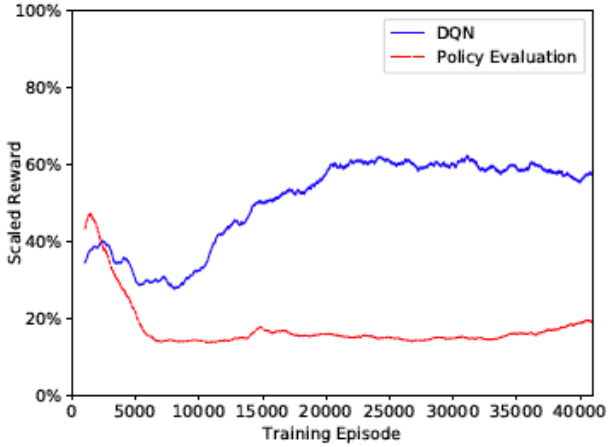
(a) City A



(b) City B



(c) City C



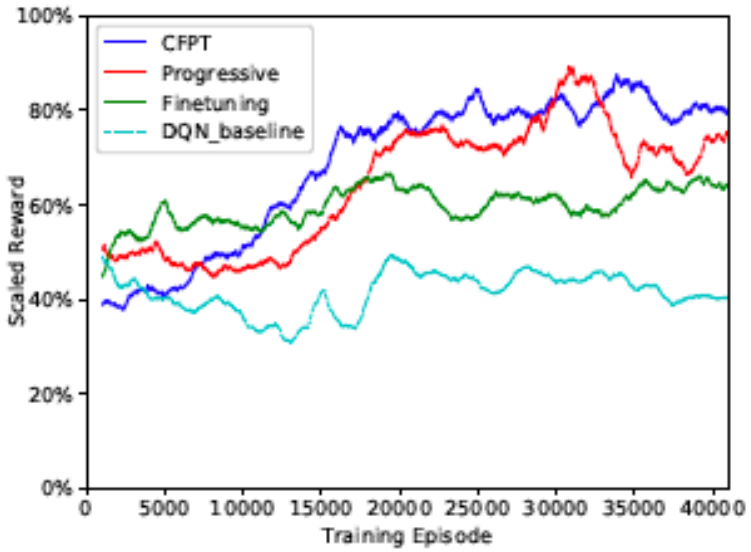
(d) City D

Results

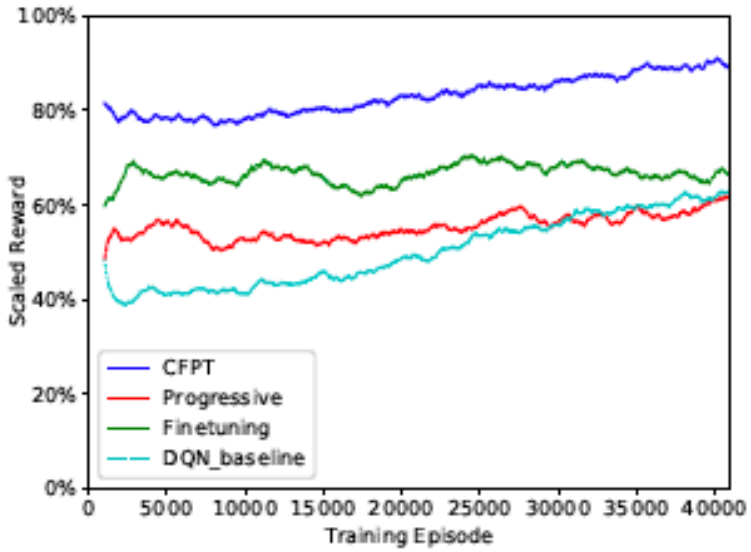
No transfer v.s. transfer

Spatial transfer

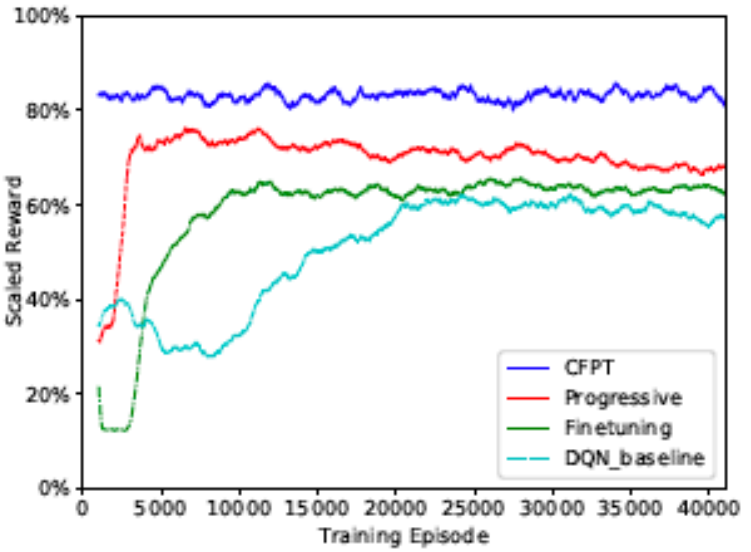
- Robustness
- Better initial solution
- Higher converged cumulative rewards



(a) City B



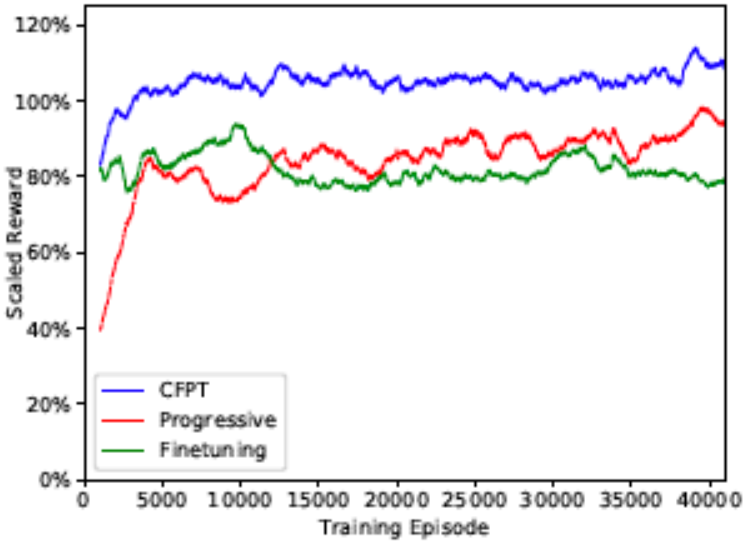
(b) City C



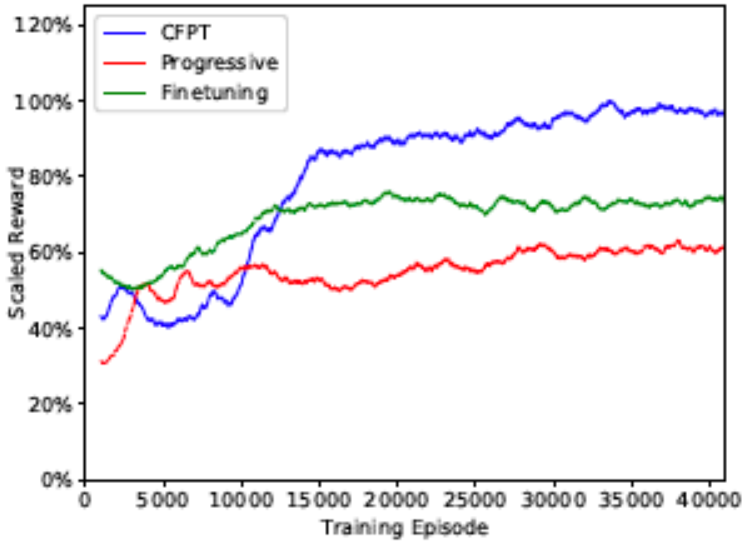
(c) City D

Results

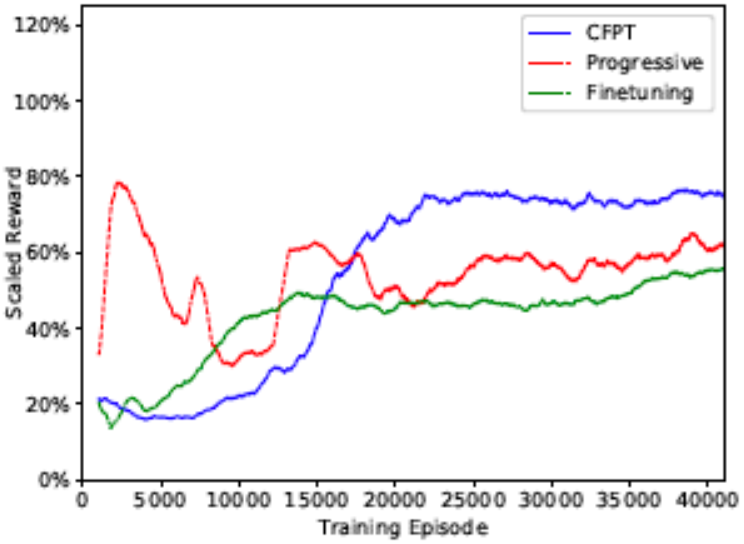
Temporal transfer



(a) City B



(b) City C



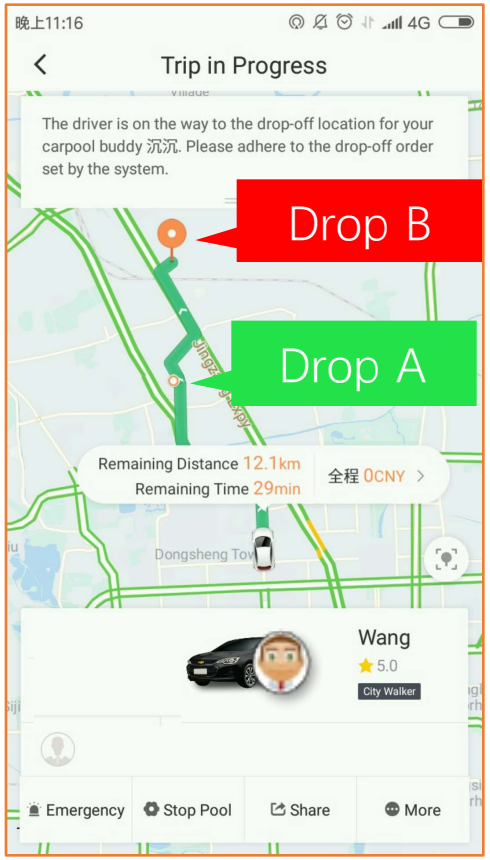
(c) City D

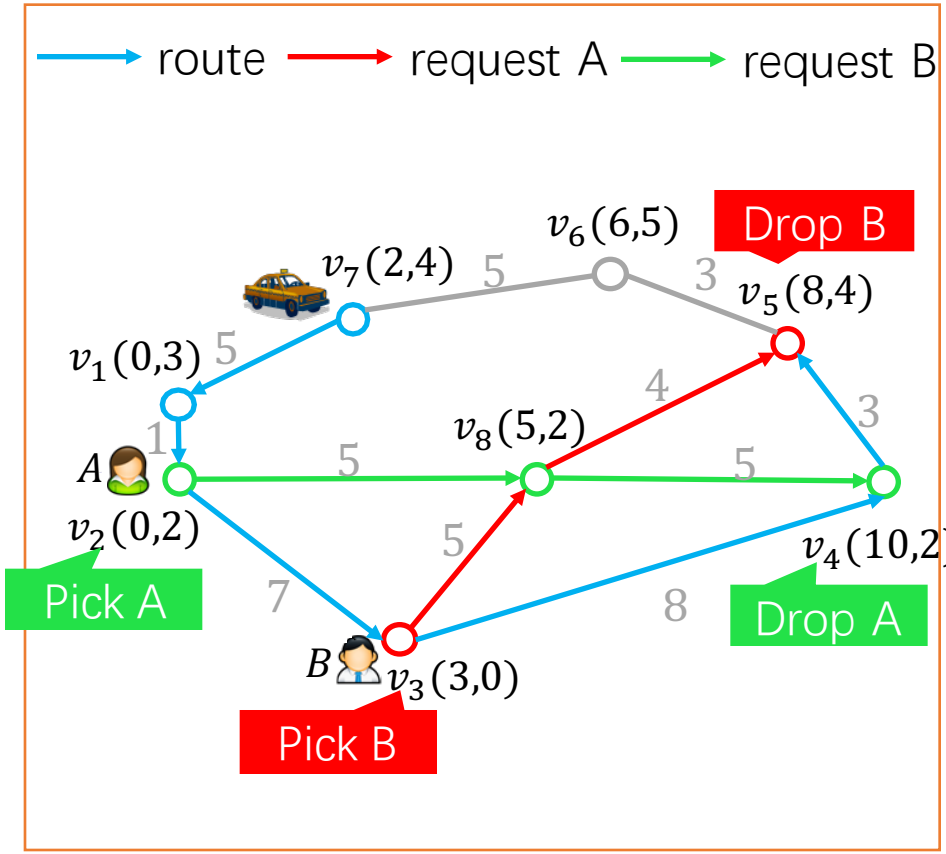
KDD2018

Dynamic Ride Sharing

Dynamic Ride Sharing

- Limitations
  - target on single objective
  - rely on inefficient insertion, $O(n^2)$ or $O(n^3)$
- Contributions
  - a unified cost function, generalize three main objectives
  - dynamic programming based insertion, linear time

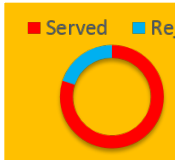




* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.

Unified Problem Definition

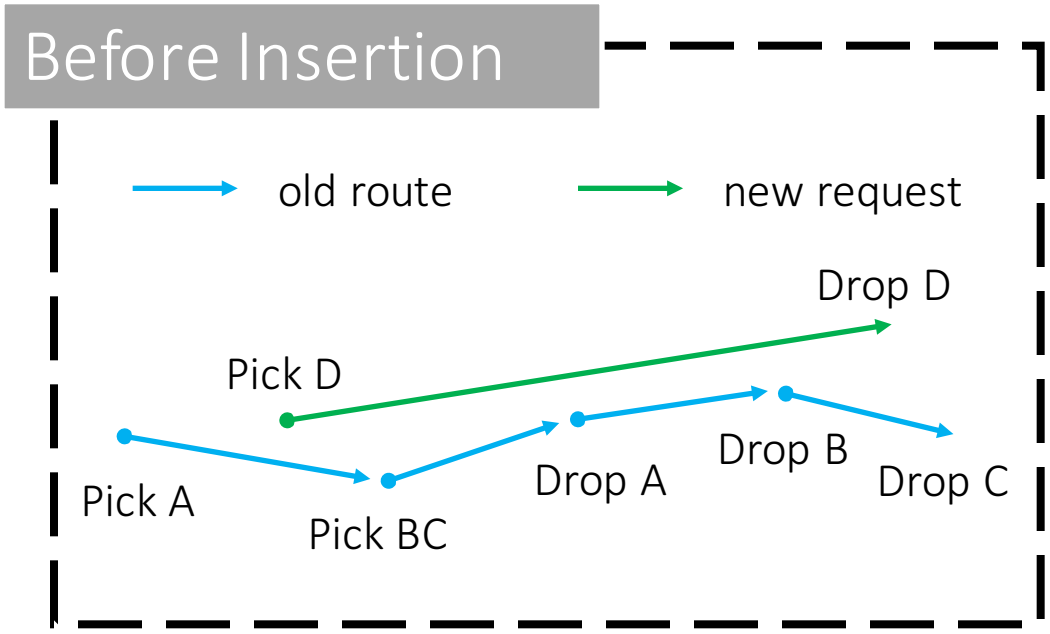
- Given a set of drivers, a set of requests dynamically arrived, the platform aims to plan route of each driver for serving the requests to minimize the unified cost.
  - $\mathrm{UC}(\mathrm{W}, \mathrm{R}) = \alpha \cdot \sum_{w \in W} \mathrm{D}(\mathrm{S_w}) + \sum_{r \in R^-} \mathrm{p_r}$
    - α: weight, $\mathrm{p_r}$: penalty for rejecting the request
    - $\begin{cases} \alpha = 1 \\ \mathrm{p_r} = \infty \end{cases} \Rightarrow$ Minimize Total Distance
    - $\begin{cases} \alpha = 0 \\ \mathrm{p_r} = 1 \end{cases} \Rightarrow$ Maximize Served Request
    - $\begin{cases} \alpha = \text{unit payment} \\ \mathrm{p_r} = \text{fare of the request} \end{cases} \Rightarrow$ Maximize Total Revenue

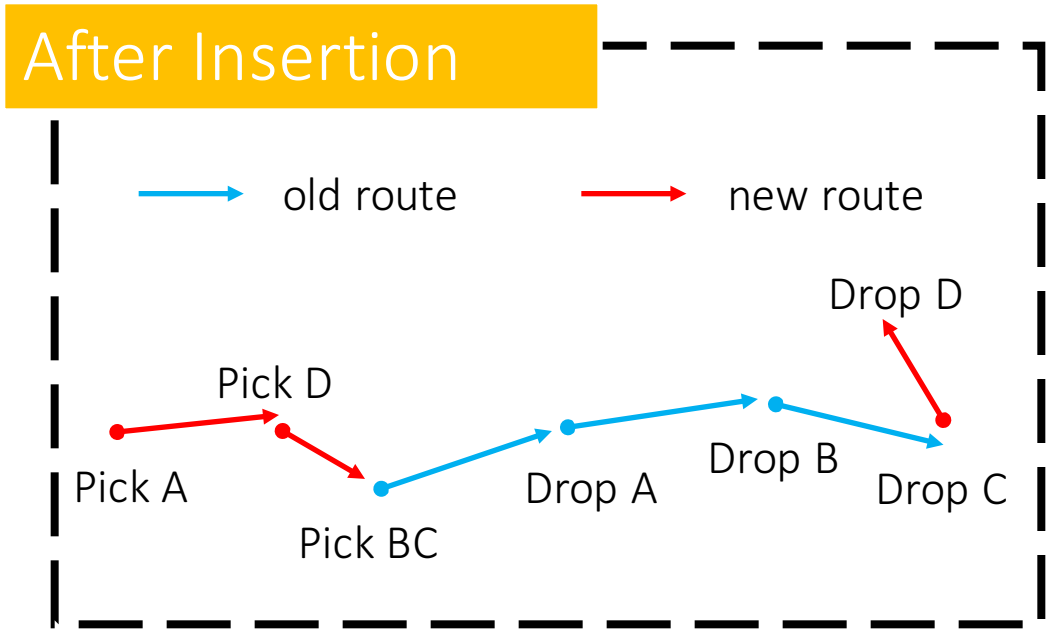


* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.

Core Operation in Ridesharing: Insertion

- Given a worker with current route, a new request is inserted into current route with minimal increased distance, while orders of old requests remain the same.

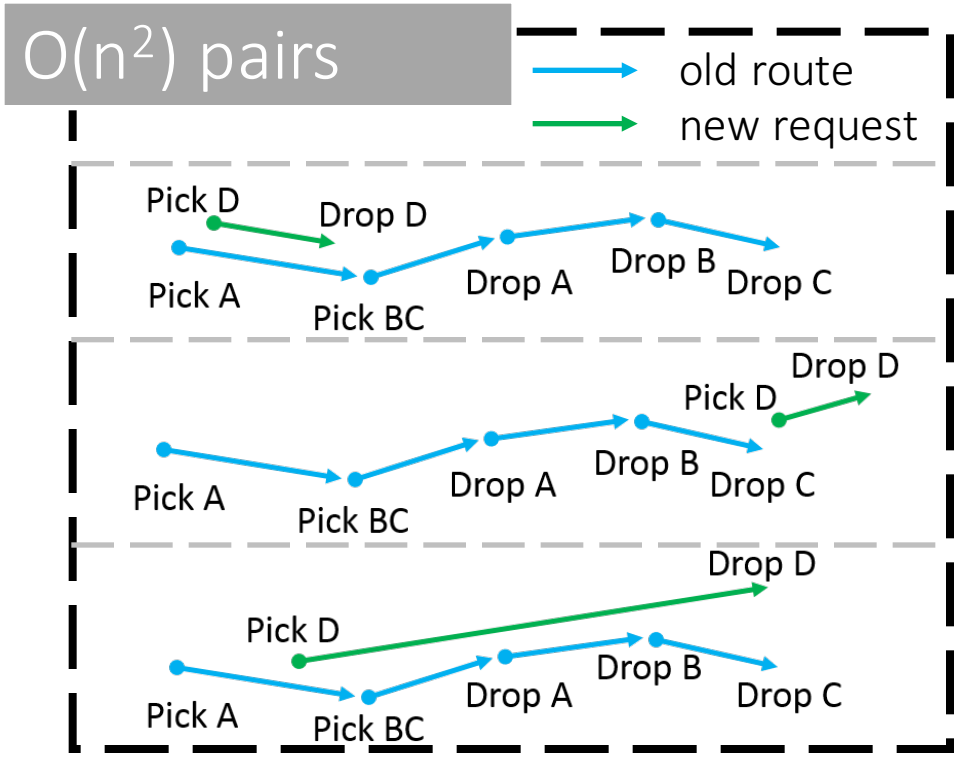




* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.

Dynamic Programming based Insertion

- Even though there are $O(n^2)$ pairs, insertion can be implemented by dynamic programming in linear time.



Dynamic Programming

$$\mathrm{Dio}[j] = \begin{cases} \infty, & \text{if } \mathrm{picked}[j-1] > K_w - K_r \\ \mathrm{Dio}[j-1], & \text{if } \det(l_{j-1}, o_r, l_j) > \mathrm{slack}[j-1] \\ \min\{\mathrm{Dio}[j-1], \det(l_{j-1}, o_r, l_j)\}, & \text{otherwise} \end{cases}$$

$$\mathrm{Plc}[j] = \begin{cases} \mathrm{NIL}, & \text{if } \mathrm{picked}[j-1] > K_w - K_r \\ \mathrm{Plc}[j-1], & \text{if } \det(l_{j-1}, o_r, l_j) > \mathrm{slack}[j-1] \\ \mathrm{Plc}[j-1], & \text{if } \mathrm{Dio}[j-1] < \det(l_{j-1}, o_r, l_j) \\ j-1, & \text{if } \mathrm{Dio}[j-1] \geq \det(l_{j-1}, o_r, l_j) \end{cases}$$

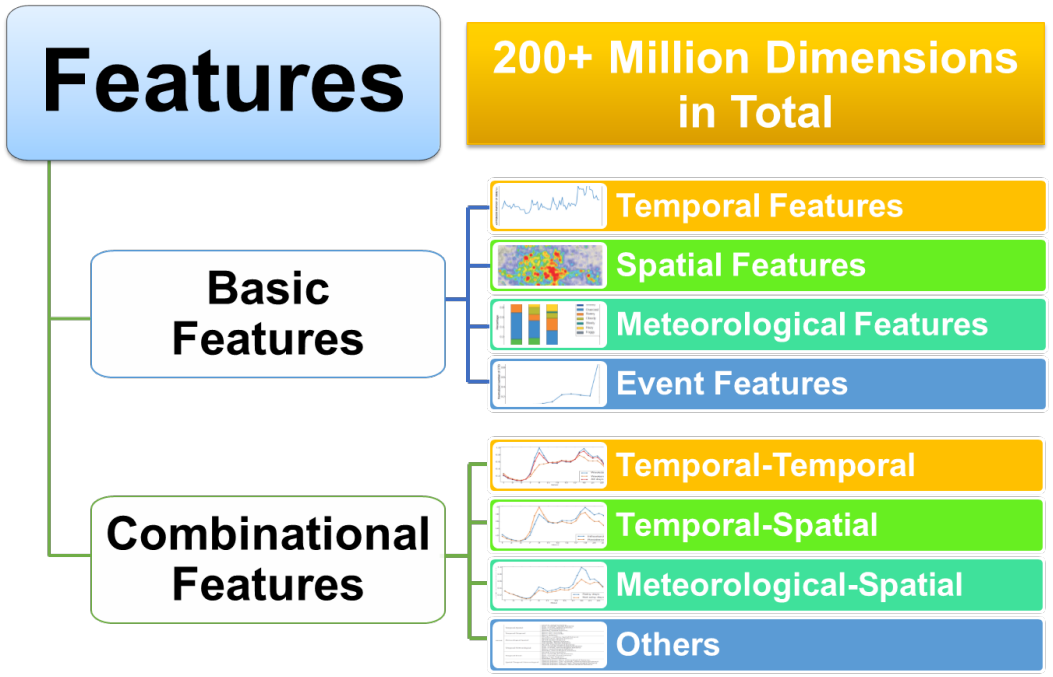
* Yongxin Tong et al. A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.

Reference

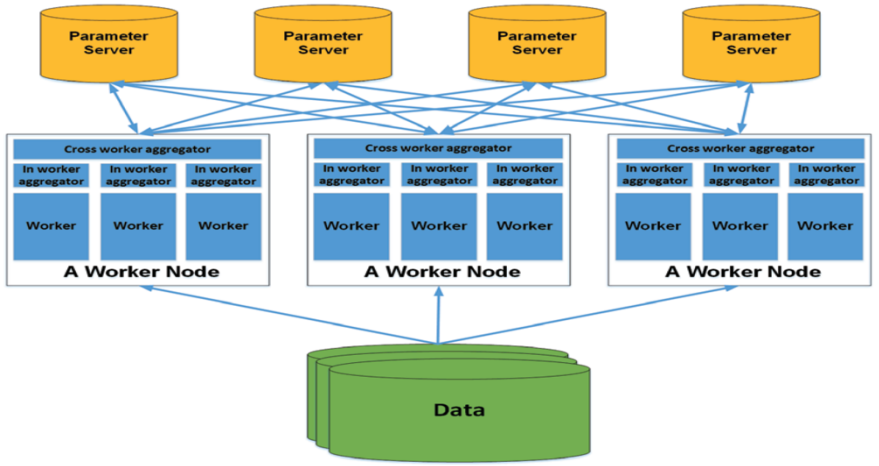
- [Tong et al. VLDB' 18] Tong, Y., Zeng, Y., Zhou, Z., Chen, L., Ye, J., Xu, K., A Unified Approach to Route Planning for Shared Mobility, VLDB, 2018.
- [Tong et al. SIGMOD' 18] Tong, Y., Wang, L., Zhou, Z., Chen, L., Du, B., Ye, J., Dynamic Pricing in Spatial Crowdsourcing: A Matching-Based Approach, SIGMOD, 2018.
- [Tong et al. VLDB' 17] Tong, Y., Wang L., Zhou Z.,Ding. B., Chen L., Ye J., Xu K., Flexible Dynamic Task Assignment in Real-Time Spatial Data, VLDB, 2017.

Supply and Demand Forecasting

Supply and Demand Forecasting



$$\mathrm{obj}_{LinUOTD}(\boldsymbol{w}) = \sum_{i=1}^{N}(y_i - p_i)^2 + \lambda_1 \|\boldsymbol{w}\|_1 + \lambda_2 \|\boldsymbol{w}\|_2 + \gamma \sum_{X \subseteq D} \phi(X) var(\{\boldsymbol{w}'\boldsymbol{x} | \boldsymbol{x} \in X\})$$

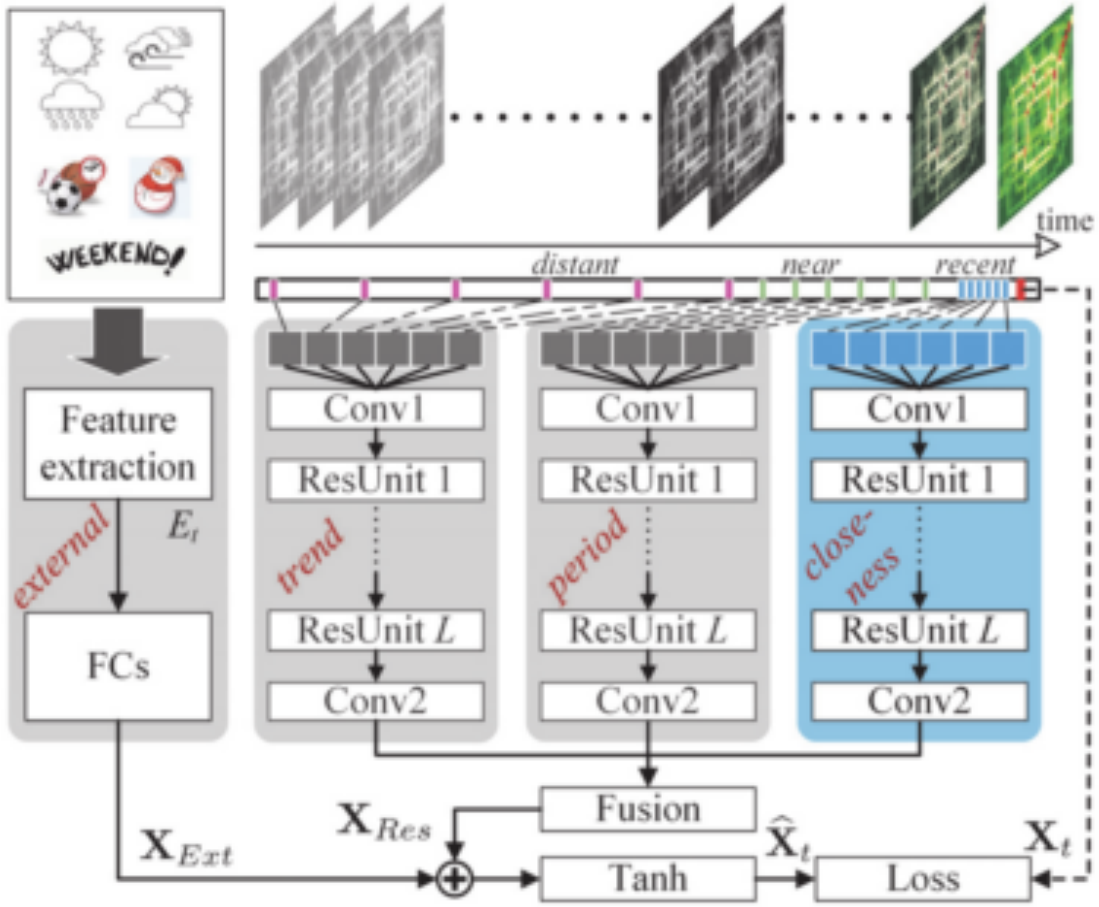


Tong et al., KDD 2017

Supply and Demand Forecasting

DeepST or ST-ResNet

- Spatial proximity: extract features from spatially proximate neighborhoods
- Temporal feature: temporal proximity and periodicity (n-weeks ago, n-days ago, n-hours ago)

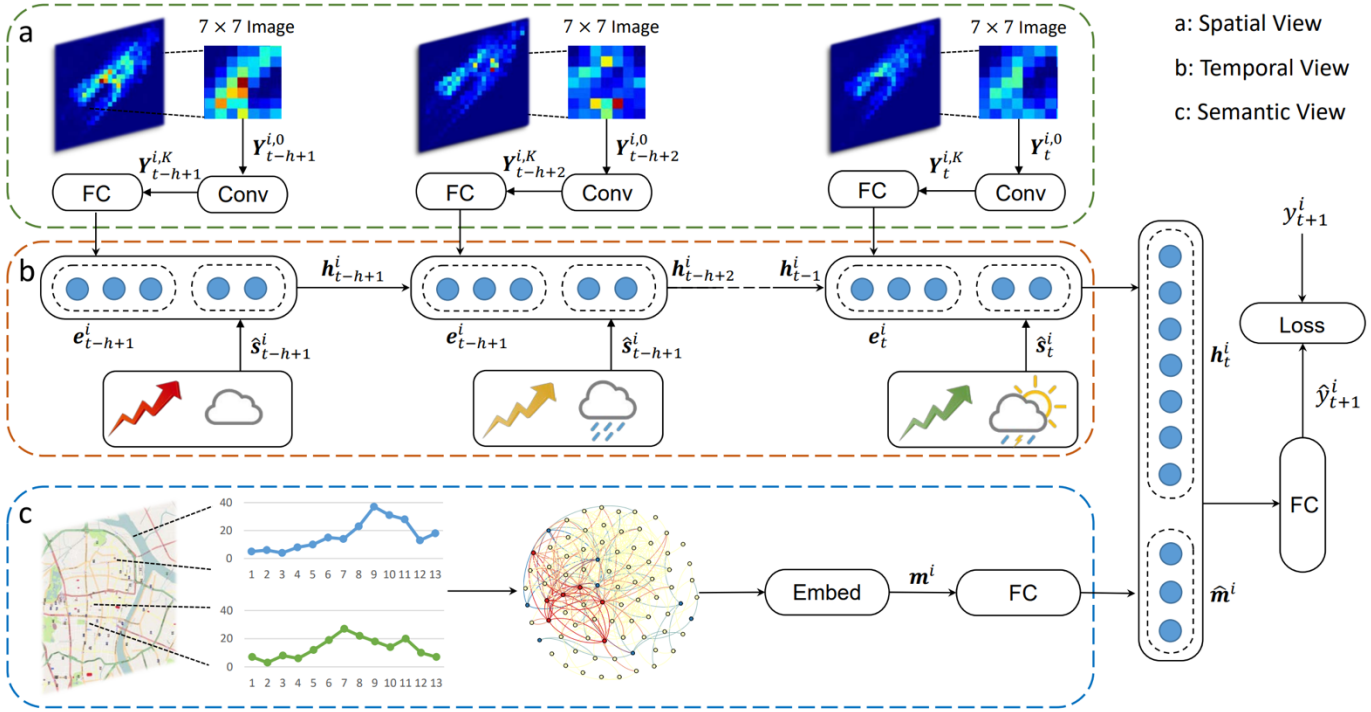


Zhang, et al, AAAI 2017

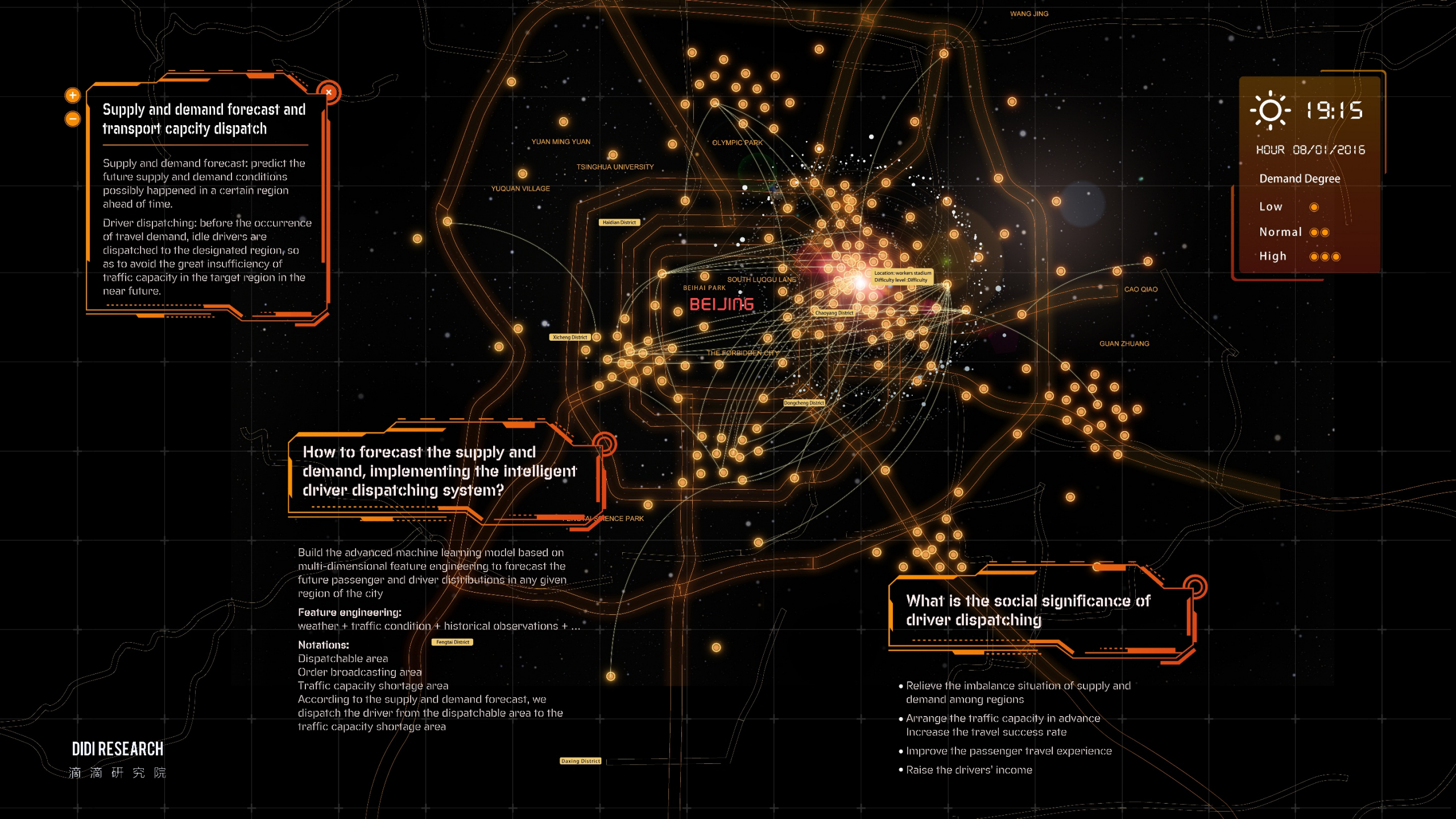
Supply and Demand Forecasting

Deep Multi-View ST Network

- Spatial proximity: extract features from spatially proximate neighborhoods
- Temporal feature: Recurrent Neural network
- Semantic similarity: build semantic graph followed by embedding



Yao, et al, AAAI 2018



Supply and demand forecast and transport capcity dispatch

Supply and demand forecast: predict the future supply and demand conditions possibly happened in a certain region ahead of time.

Driver dispatching: before the occurrence of travel demand, idle drivers are dispatched to the designated region, so as to avoid the great insufficiency of traffic capacity in the target region in the near future.

How to forecast the supply and demand, implementing the intelligent driver dispatching system?

Build the advanced machine learning model based on multi-dimensional feature engineering to forecast the future passenger and driver distributions in any given region of the city

Feature engineering:
weather + traffic condition + historical observations + ...

Notations:
Dispatchable area
Order broadcasting area
Traffic capacity shortage area
According to the supply and demand forecast, we dispatch the driver from the dispatchable area to the traffic capacity shortage area

What is the social significance of driver dispatching

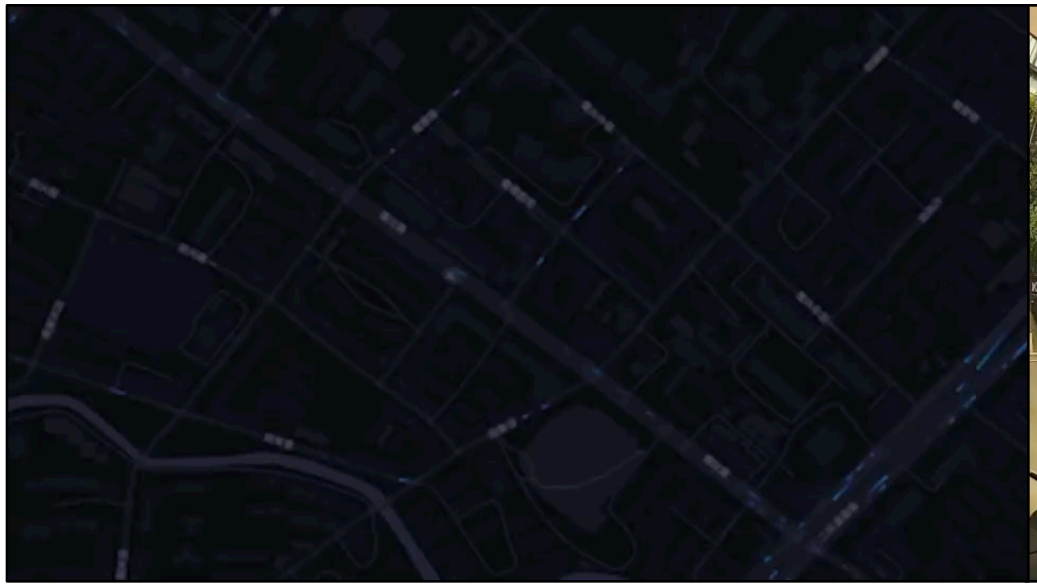
- Relieve the imbalance situation of supply and demand among regions
- Arrange the traffic capacity in advance Increase the travel success rate
- Improve the passenger travel experience
- Raise the drivers' income

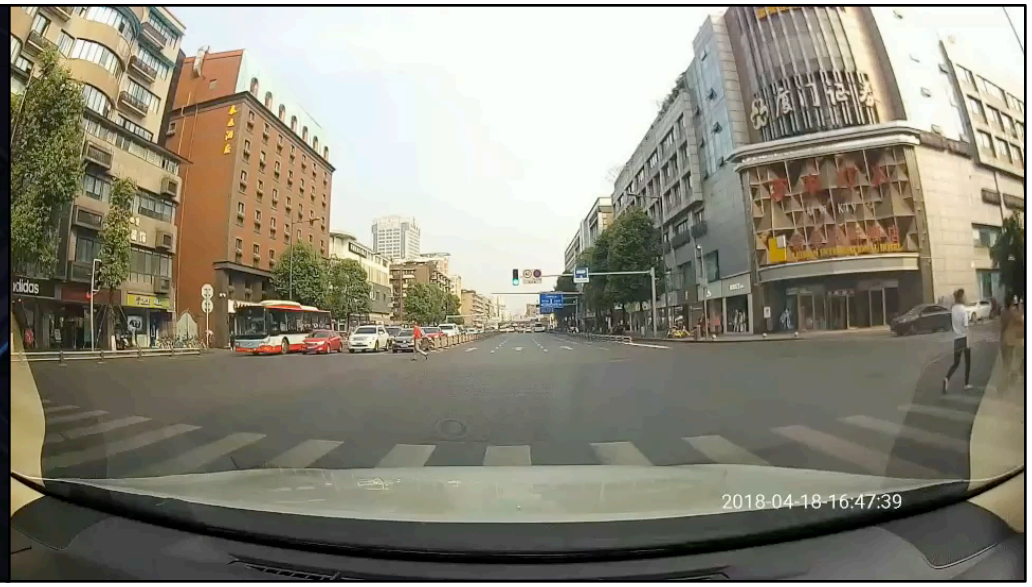
DIDI RESEARCH
滴滴研究院

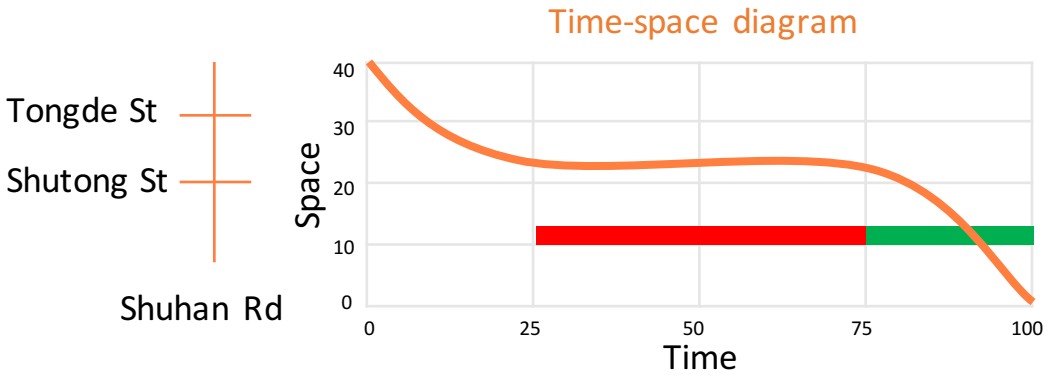
KDD2018

Smart Transportation

Data Collection, State Estimation, and Prediction





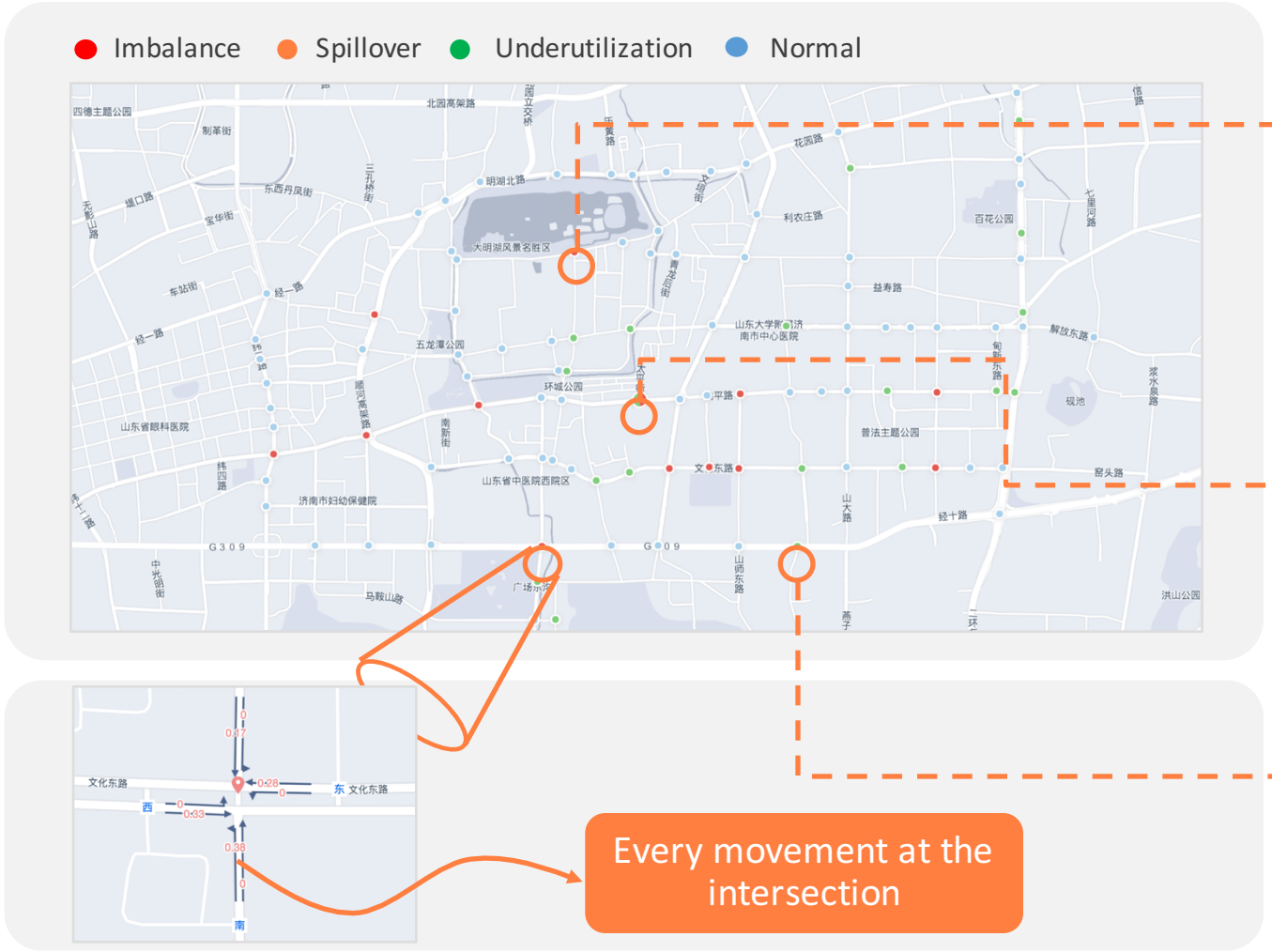


2018-4-18

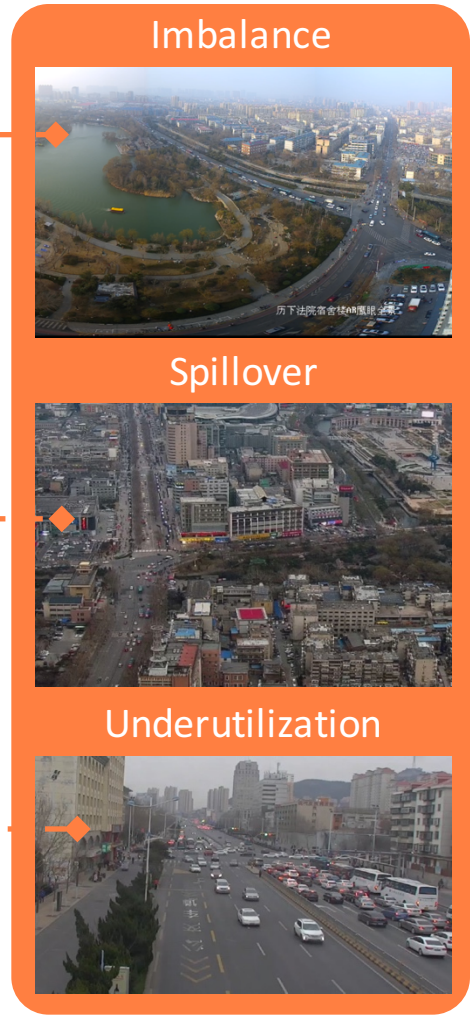
Time	MsgType	Msg
16:47:39	Traffic Sign	No Parking
16:47:39	Traffic Sign	Speed Limit 60km/h
16:47:39	Traffic Sign	Bus Lane
16:47:51	Traffic Sign	No Left Turn
16:48:02	Congestion	Stop in Queue

……

Performance Evaluation and Diagnosis

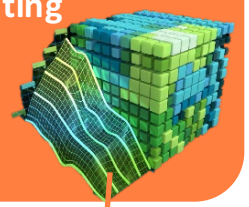


Every movement at the intersection

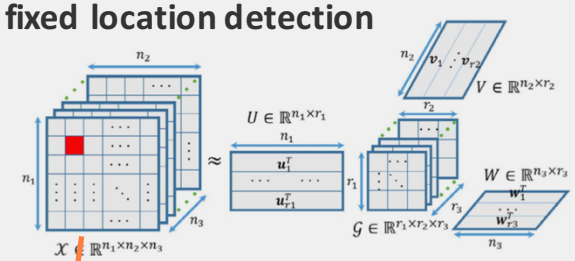


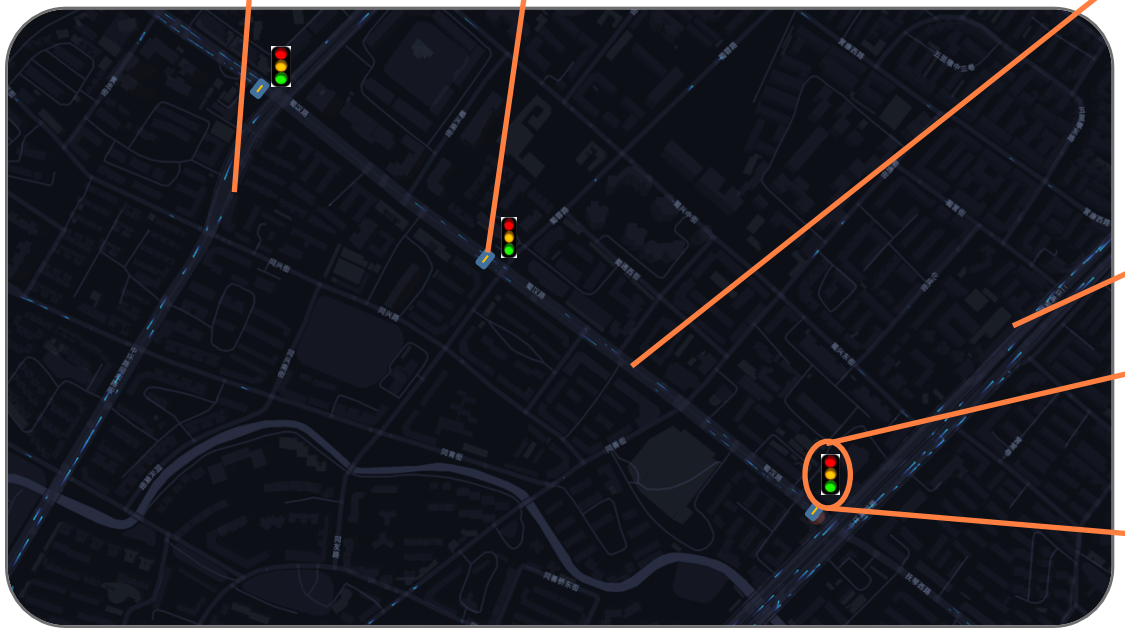
Real Time Traffic Control

Real-time data processing with distributed computing



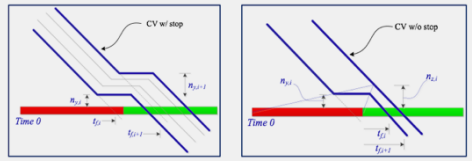
Data fusion- mobile sensing + fixed location detection



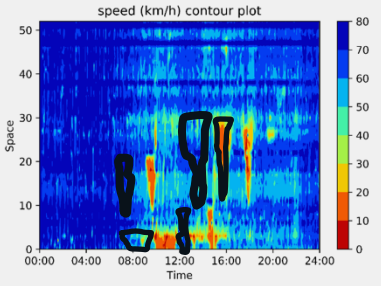


Traffic states reconstruction

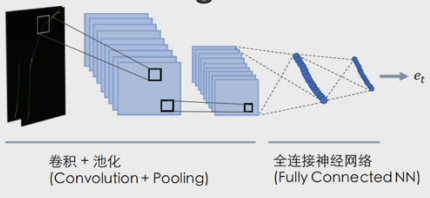
$$L(Y,Z|\lambda) = \prod_{i=1}^{n}\left\{\frac{(\lambda P_{y,i})^{n_{y,i}}e^{-(\lambda P_{y,i})}}{n_{y,i}!}\right\}\prod_{j=1}^{m}\left\{\sum_{k=0}^{n_{z,i}}\frac{(\lambda P_{z,i})^{n_{z,i}}e^{-(\lambda P_{z,i})}}{n_{z,i}!}\right\}$$



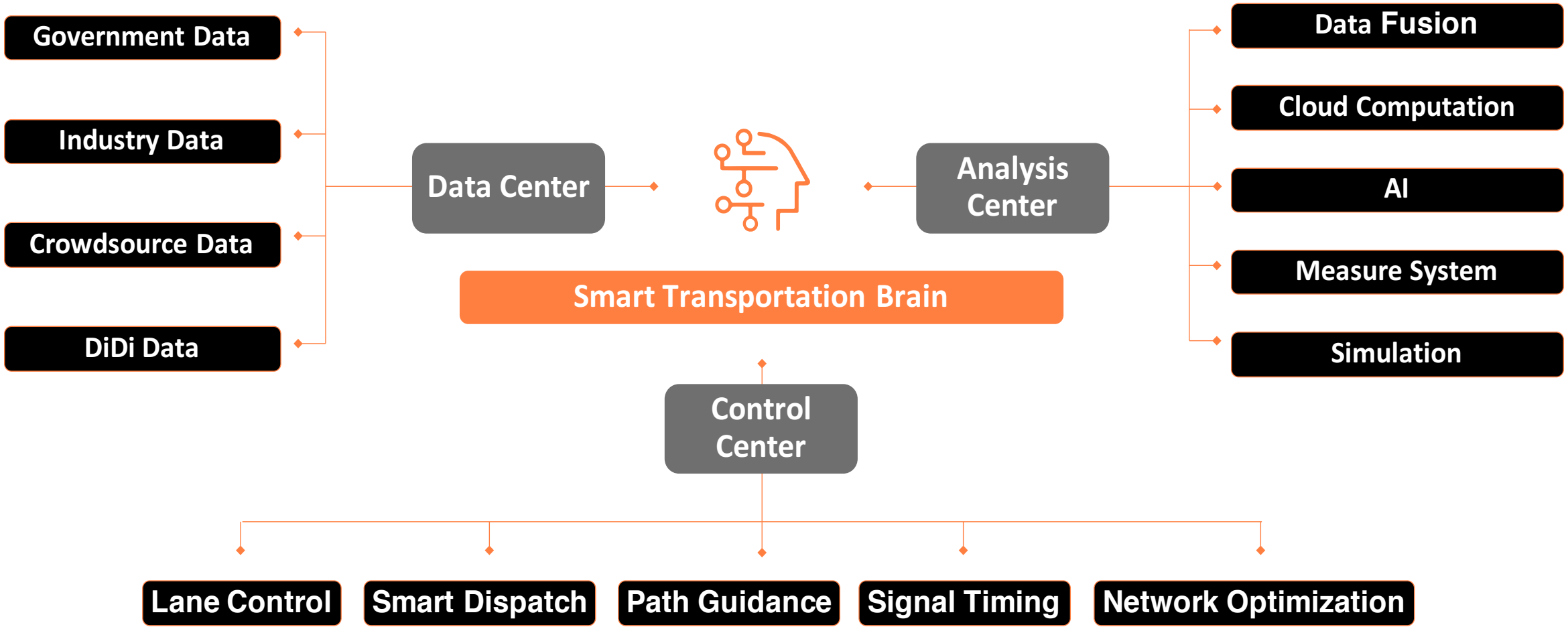
Integrated ramp metering and traffic signal control



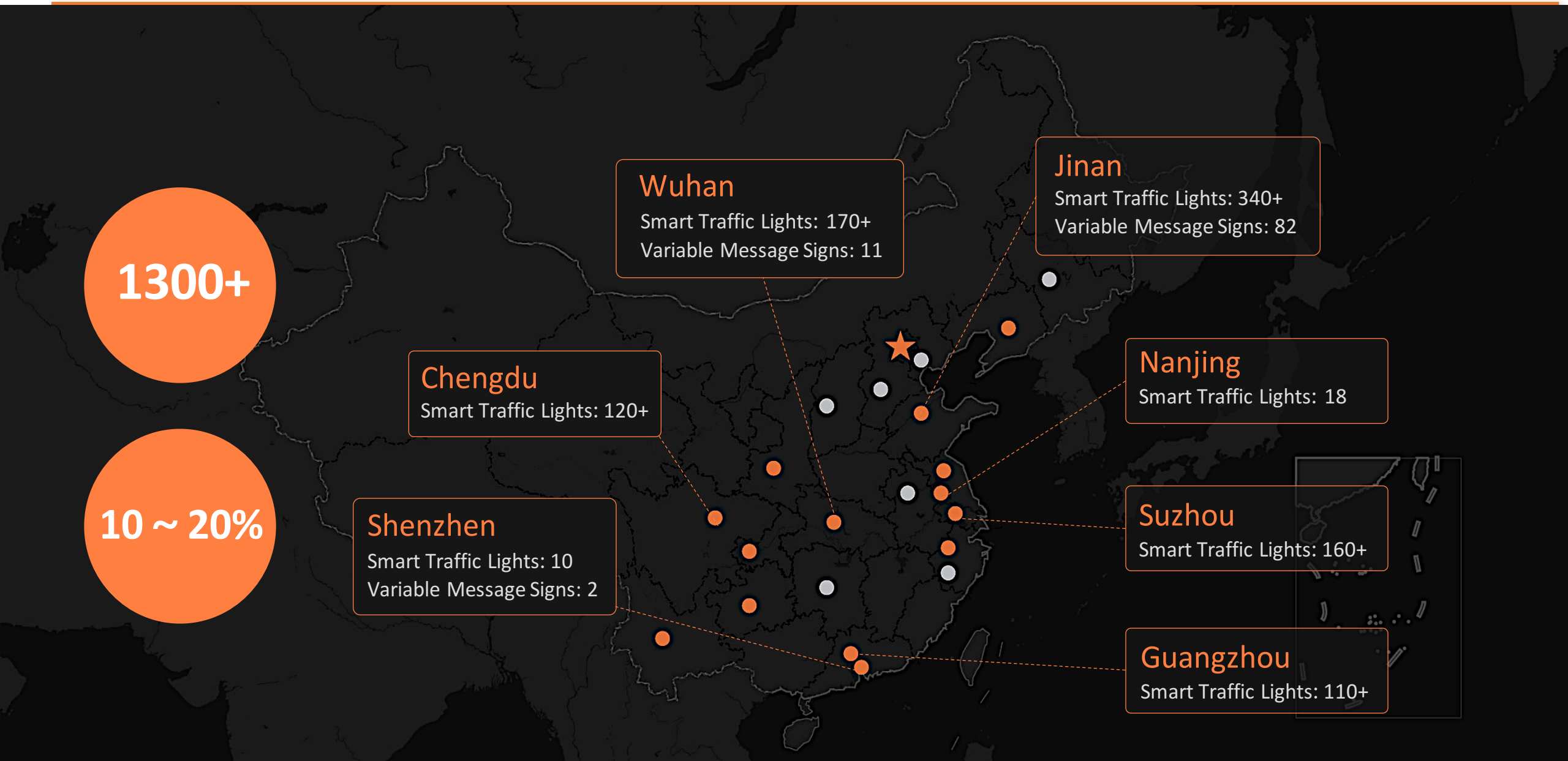
Machine learning based controller



Integrated Solution - Smart Transportation Brain



Improving traffic conditions in over 20 cities



KDD2018

Electric Vehicle

Motivations

Less Emission Brighter Future!

>1.7 million new energy vehicles in China (the world's largest)

- 400,000 electric cars operating on DiDi's platform

Supporting the new energy vehicles

- Charging station
- Battery managing system (BMS)
- Dispatch strategy

Charging Station

Rich Trace Data

Spatial Clustering

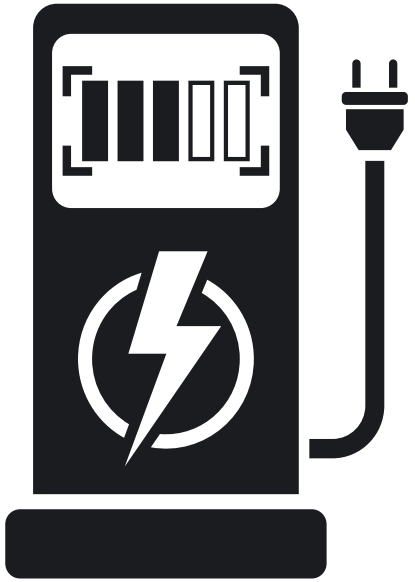
Charging Station Locations

Demand Prediction

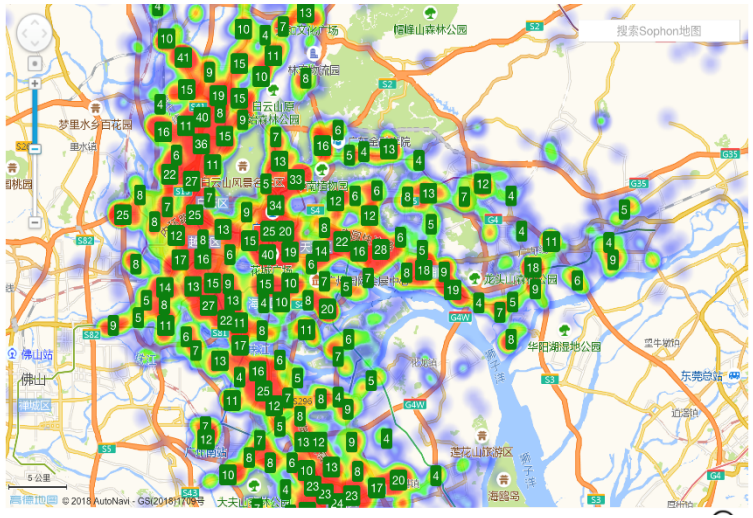
Station Capacities

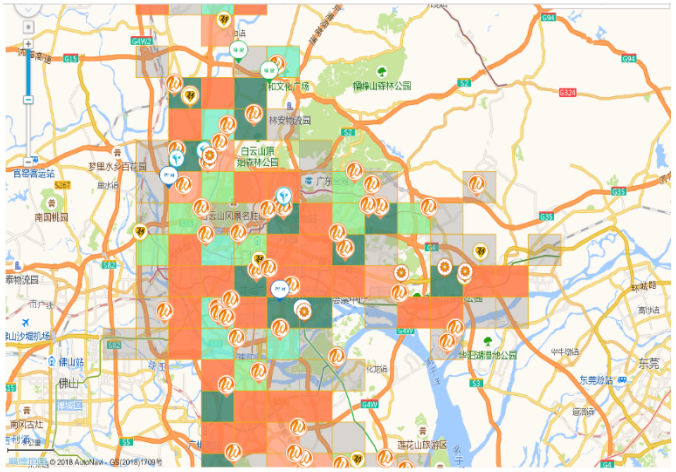
Product

Interactive Map

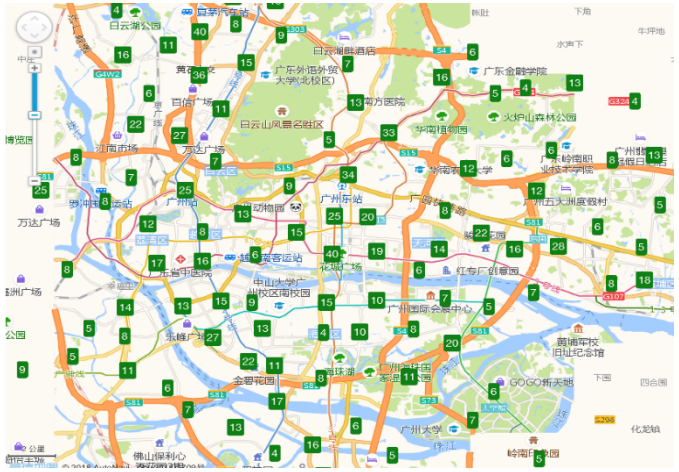


Potential Charging Stations



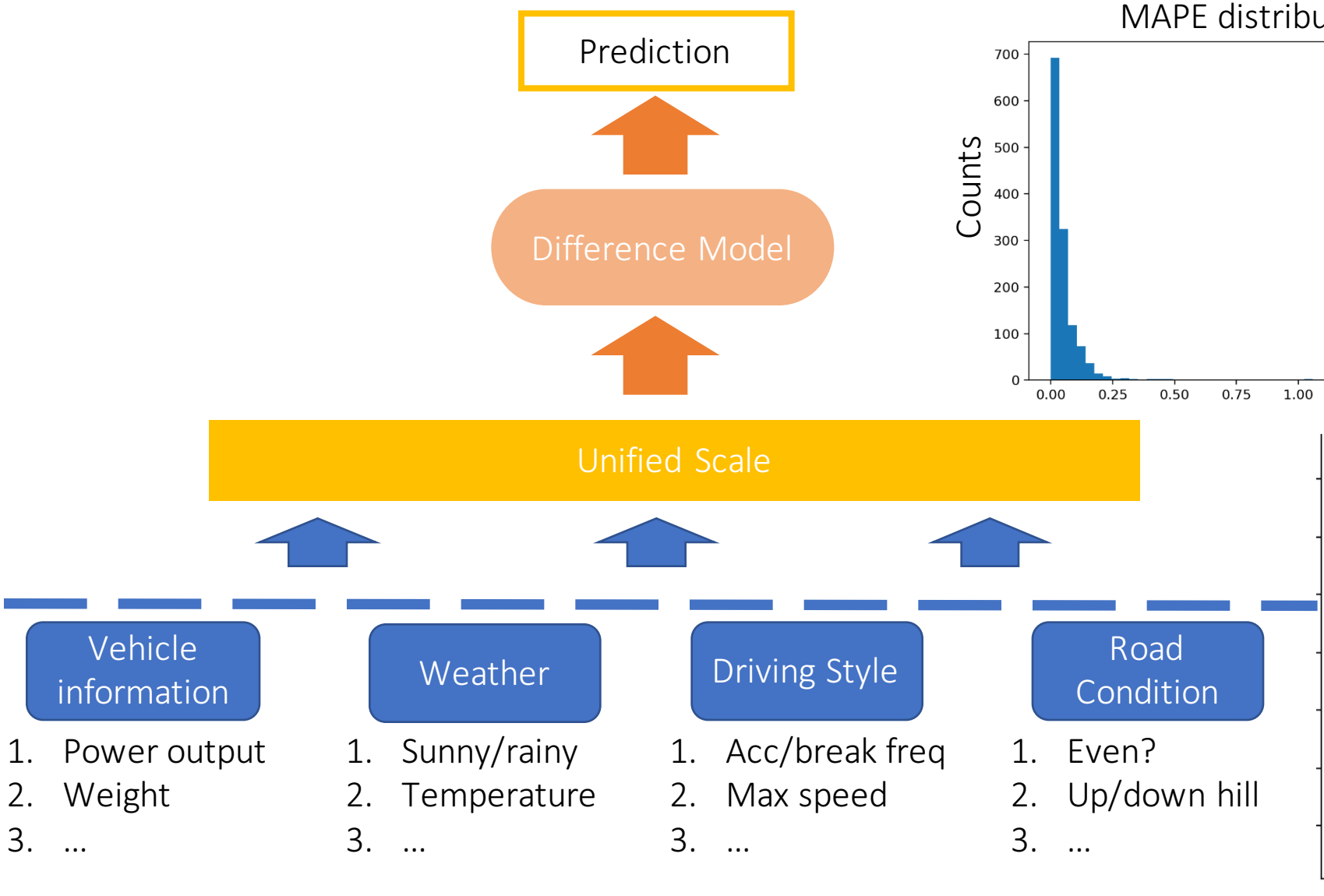


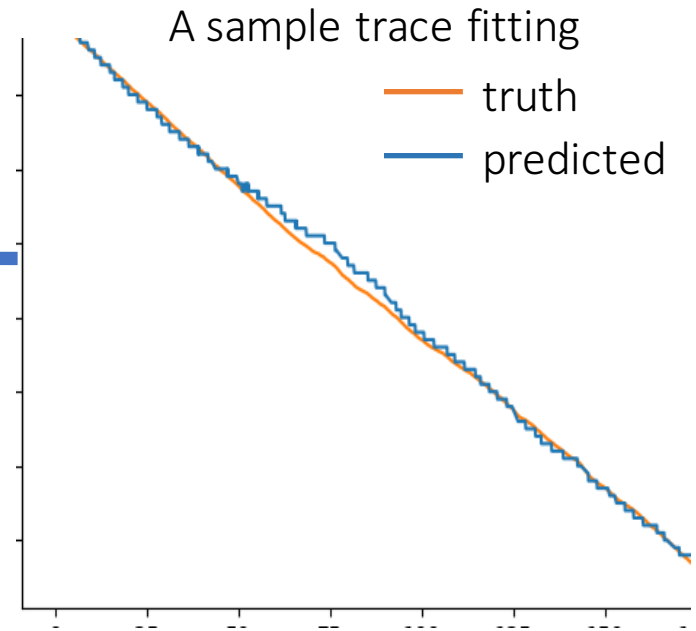
Supply-demand



Piles needed

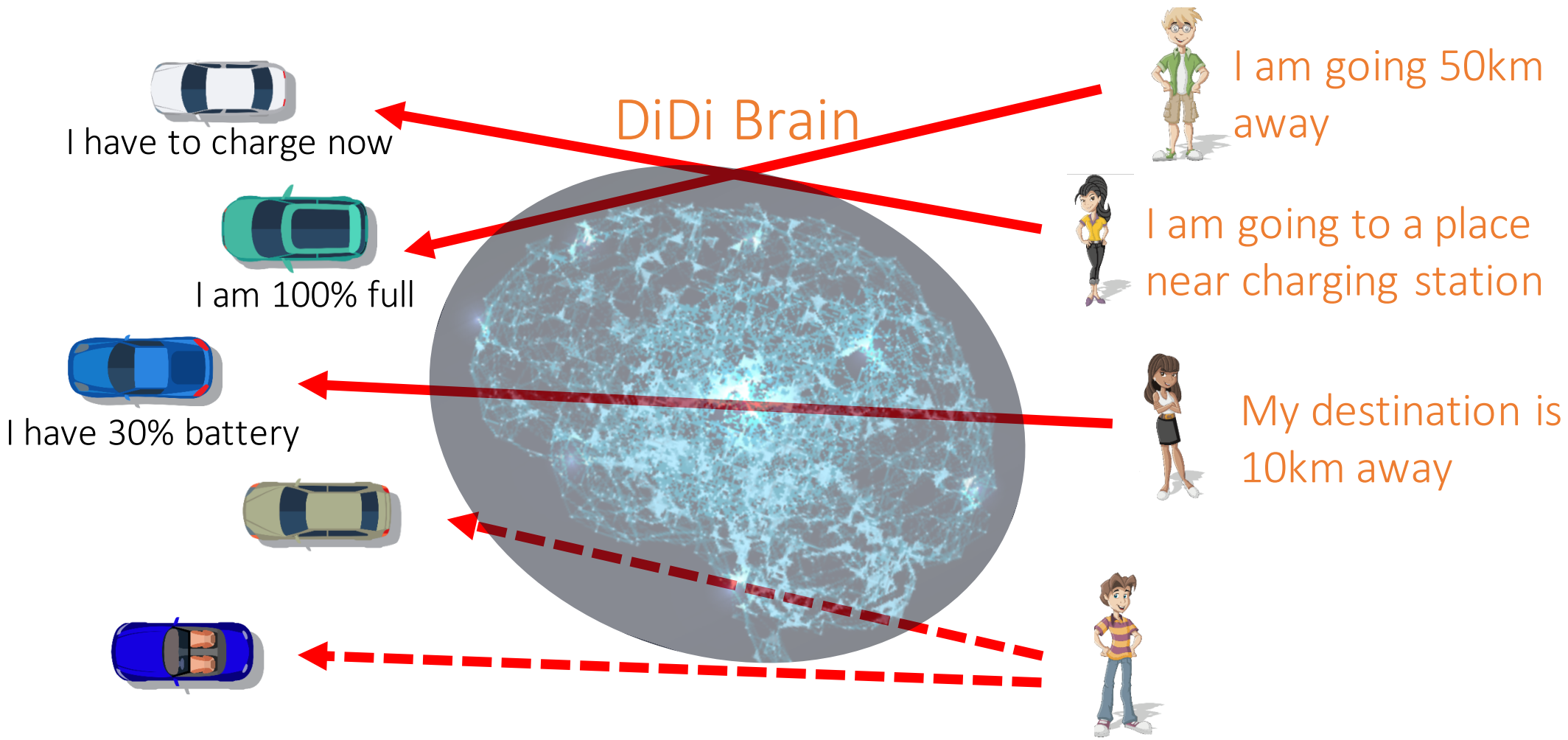
BMS (Consumption Prediction)





Dispatching Strategy

Adapting the unique characteristic of new energy vehicles



KDD2018

AI for Social Good


Deep Learning
Environmental Sustainability
Barrier-Free
EV Charging Piles Location
Reinforcement Learning
Intelligent Medical Hardware
Urban Air Quality Monitoring
AI for Social Good
Clustering
Disease Warning
Neural Networks
Health
Call for Participant
Machine Learning

KDD2018

Part 3: Data and Tools for Transportation AI

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 Data

GAIA Open Dataset

MORE THAN A JOURNEY

DiDi

Introduction

Open

Collaborative

Innovative

The GAIA Initiative aims to establish strong links between DiDi and academia, and facilitate data-driven research in transportation. It provides a platform for the academia to access anonymized data from real-life urban scenarios.

DiDi

Big Data

Every Day

Daily Rides	Data Processed	Routing Requests	New Data
30 Million	4875TB+	40 Billion	100TB+

DiDi

Sharing Anonymized Data

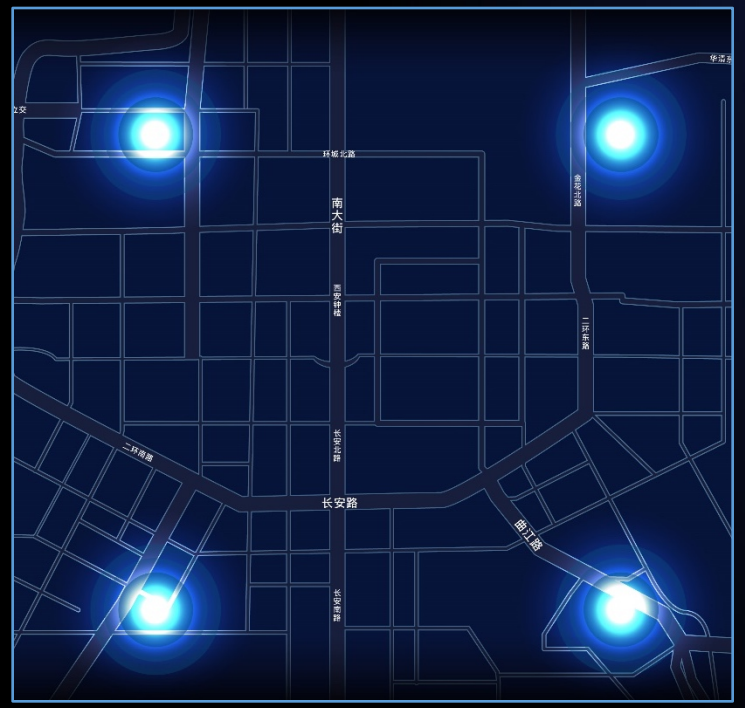
Vehicle Trajectory Data

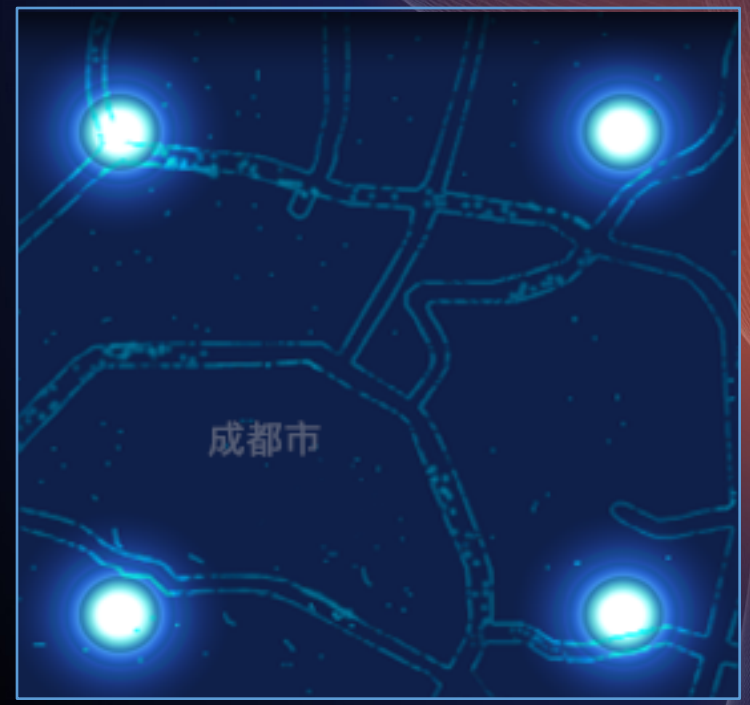
From: Chengdu and Xi'an, China

Real

High Accuracy

Full Sample





MORE THAN A JOURNEY

DiDi

Process

01

Register

Visit the GAIA Initiative website: GAIA.didichuxing.com/en and click "Apply Now" to register.

GAIA Open Dataset

02

Verification & Review

In the Data Center, click "Data Access Application". Read the user agreement carefully and fill out the application form with your info, then submit the form for verification.

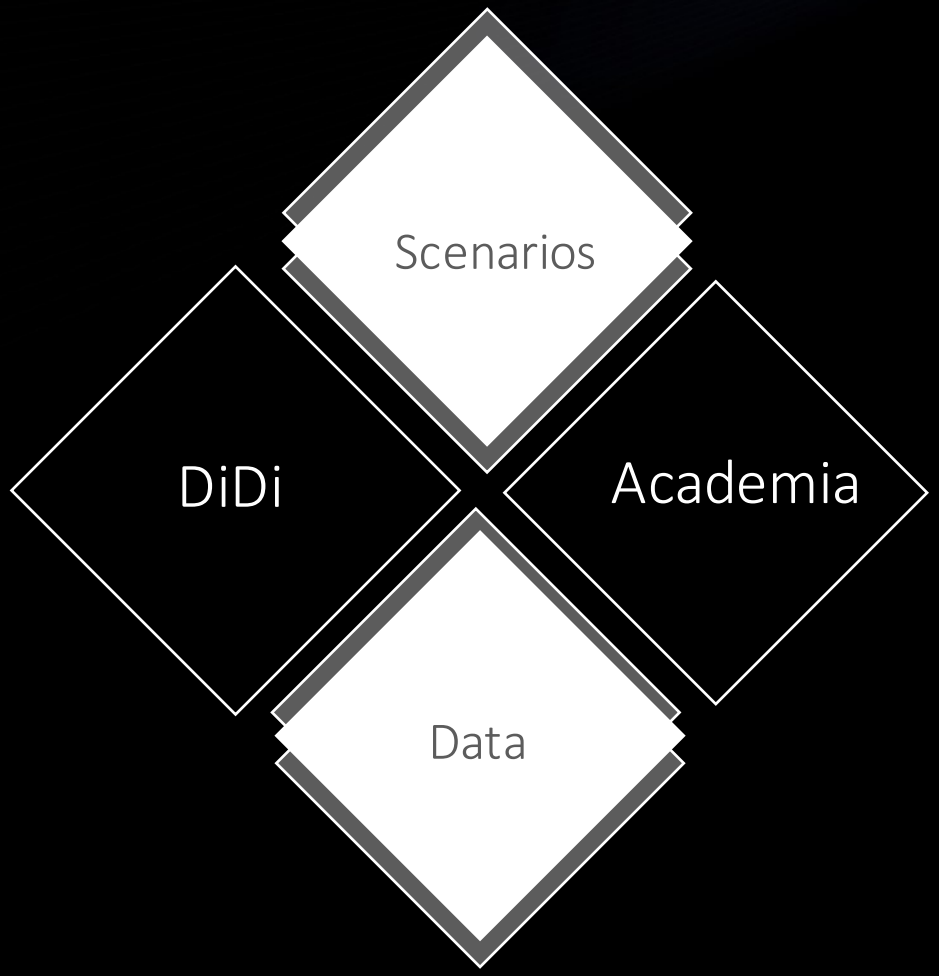
03

Access Data

Once your application is approved, DiDi will send you an email with instructions for data access. Please check your email.

DiDi

Outreach



To Redefine the Future of Mobility

GAIA Open Dataset

GAIA.didichuxing.com/en

DiDi

GAIA Open Dataset

THANKS

MORE THAN A JOURNEY

DiDi