



Artificial Intelligence in Transportation

Zheng WANG

DiDiAI Labs

Didi Chuxing



Yan LIU

DiDiAI Labs

Univ. of Southern California



Jieping YE

DiDiAI 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)

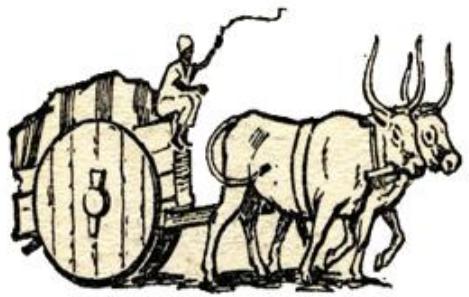
■ Data and tools for transportation AI (15min)

■ Q&A (10min)



Part 1: Challenges and Opportunities in Transportation AI

History of Urban Transportation



First Wheeled Vehicle



First Road Network



First Public Transportation



First Bicycle

3500 B.C.

1st Century

1662

1817



First Affordable Automobile



First Electric Traffic Light



U.S. National Highway System



Smart Transportation System

1908

1910s

1956

Now and Future

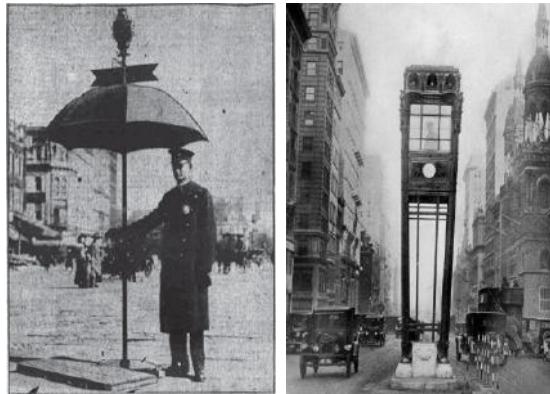
History of Traffic Light

No Traffic Light



Before

Hand-operated Traffic Light



1860s

Electric Traffic Light



1910s

1980s



Camera

1960s



Loop Detector



Weigh in Motion



Radar

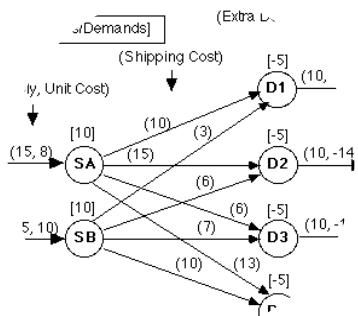
Transportation: A Multi-Disciplinary Industry



Engineering



Planning



Science

Transportation

Policy Making



Design



Management

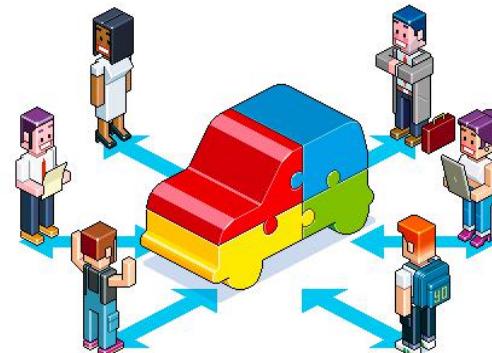


Cutting-Edge Issues

Cooperative Vehicle-Highway Systems



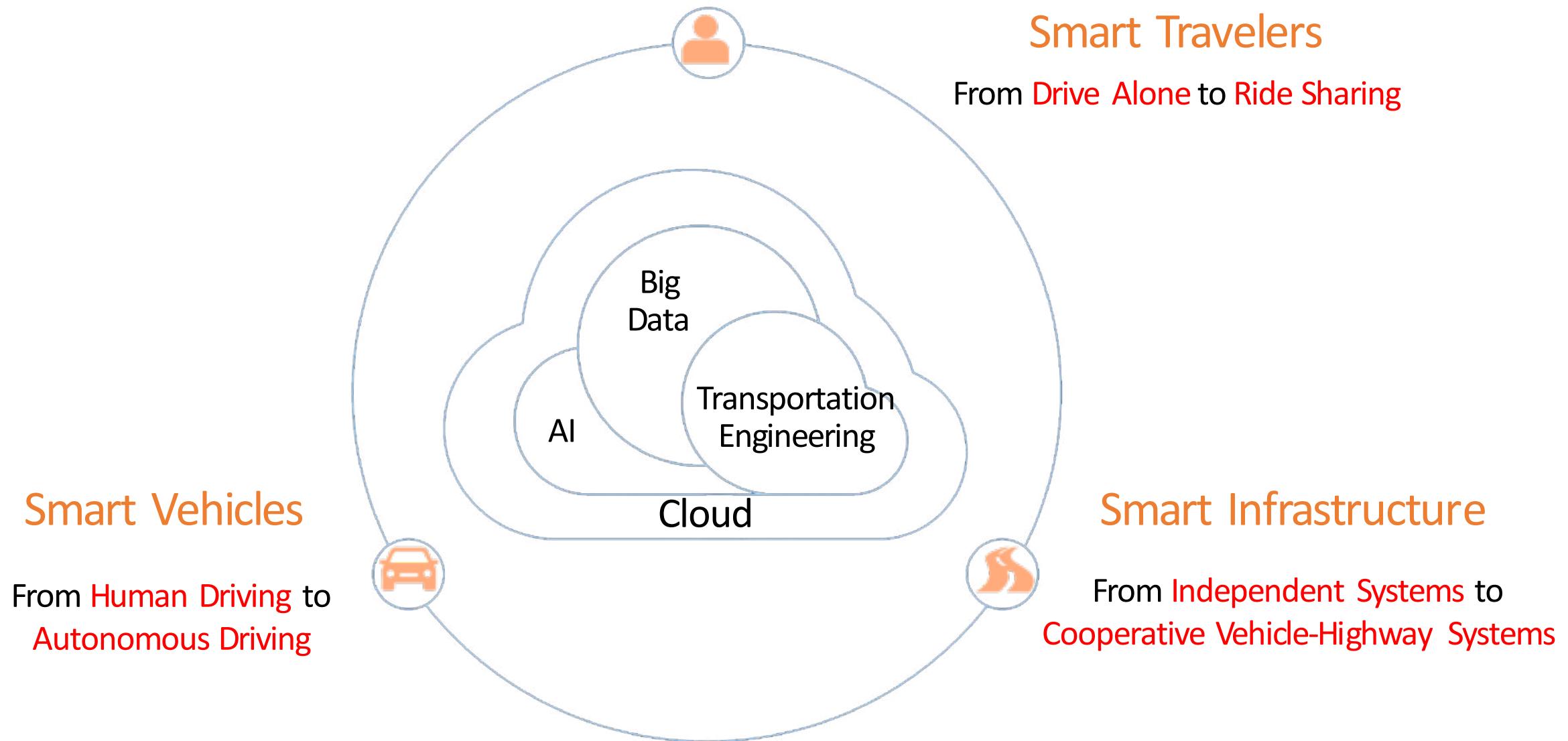
Ride Sharing



Multimodal Transportation

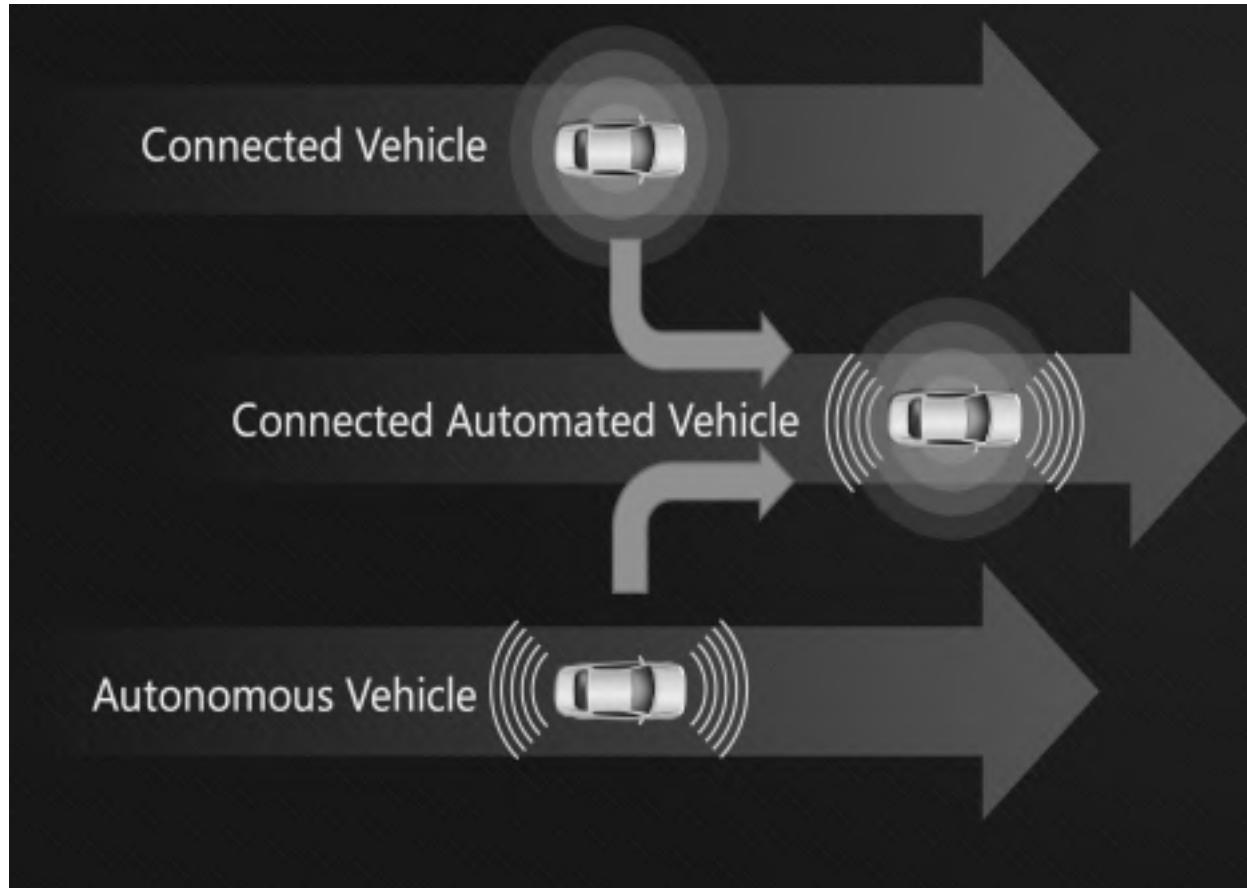


Smart Transportation System

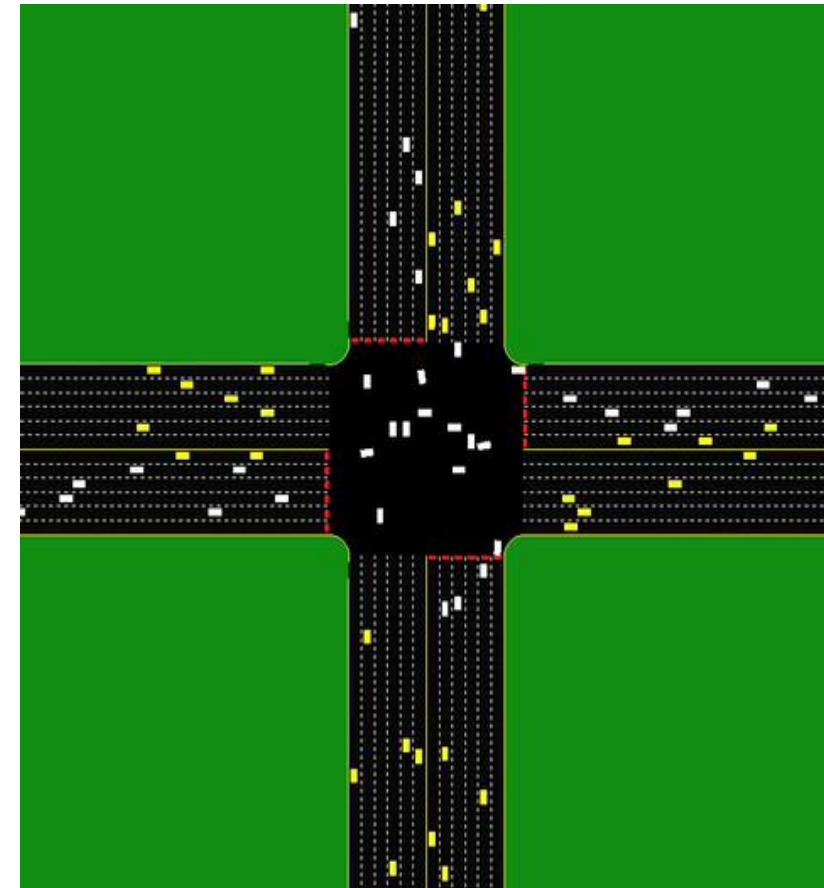


Building the Brains of Smart Transportation

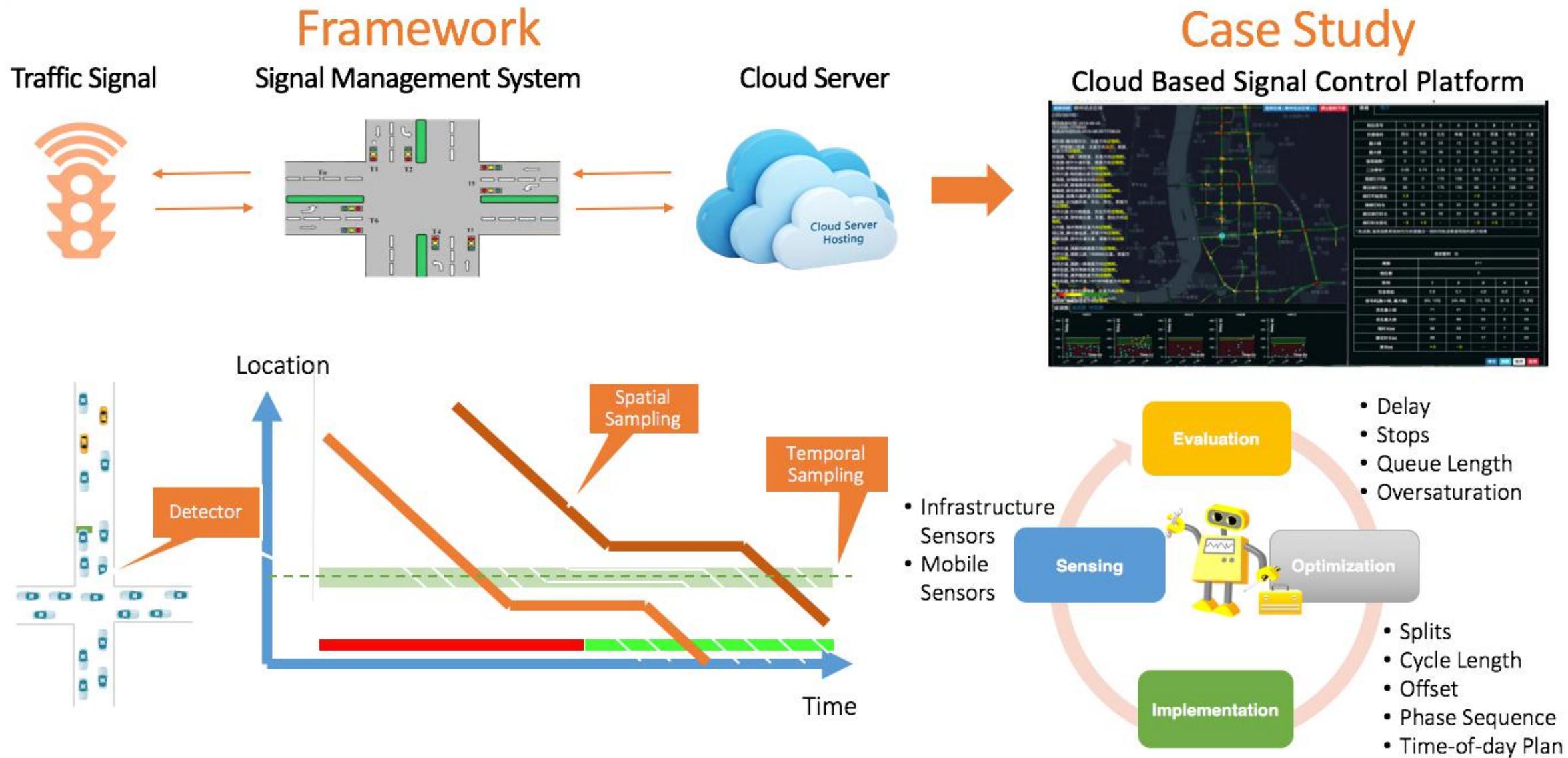
Intelligent and Connected Vehicles



Future Intersection



Adaptive Control as a Service

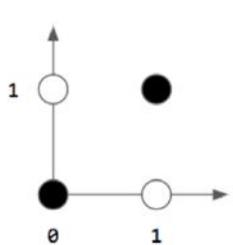


AI and Machine Learning

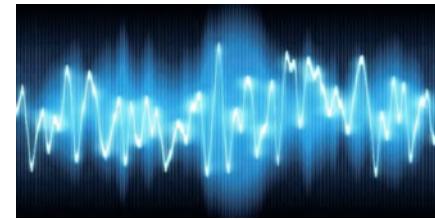
Neural Networks



Deep Learning



Machine Learning:
supervised,
unsupervised



Reinforcement
Learning



AlphaGo

Data Resource

- Location data, Trajectory data



- Transaction data



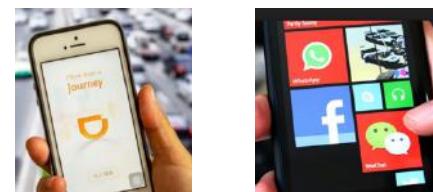
- Profile data



- Sensors: multimedia data

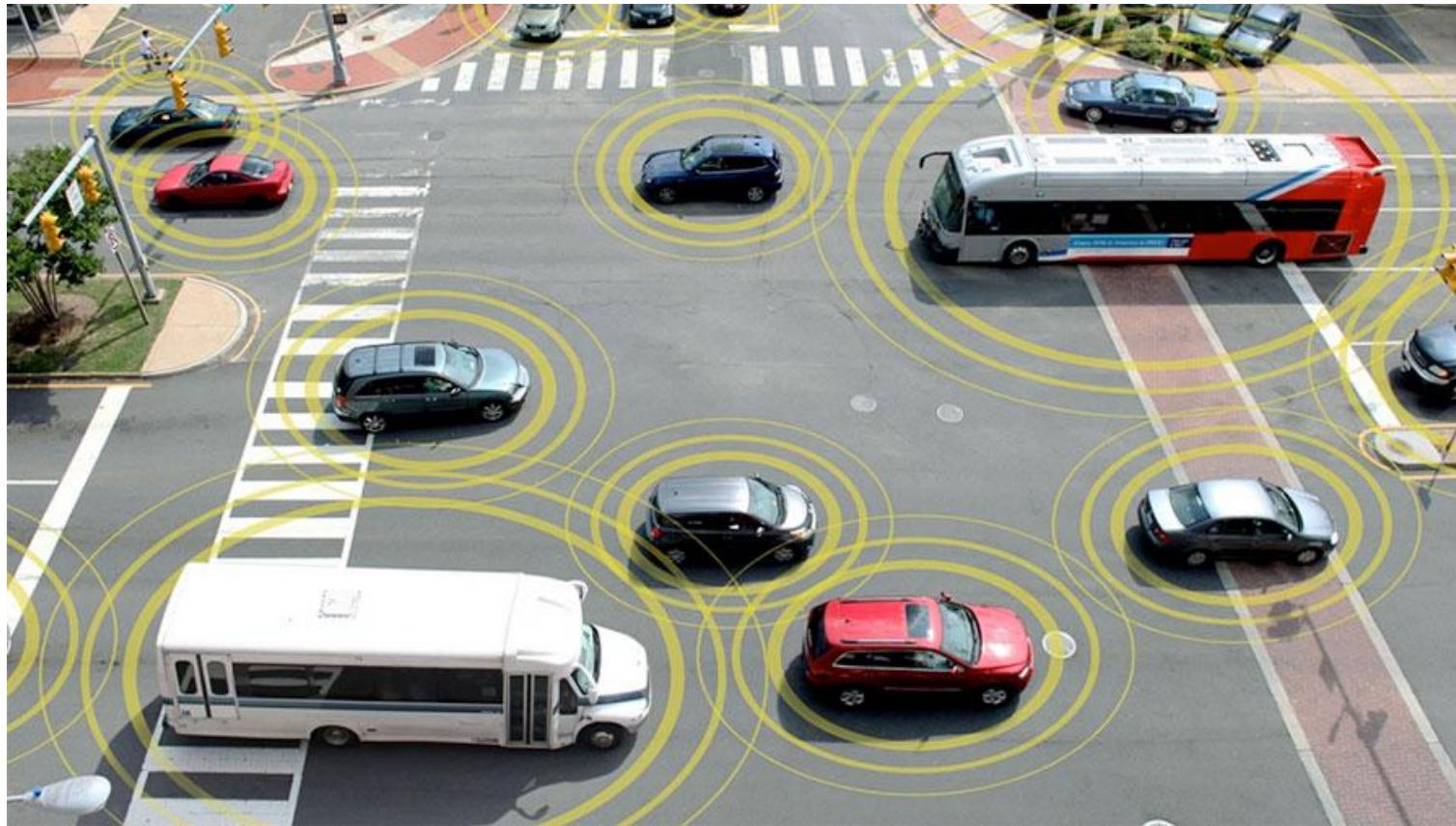


- Cross-platform identification



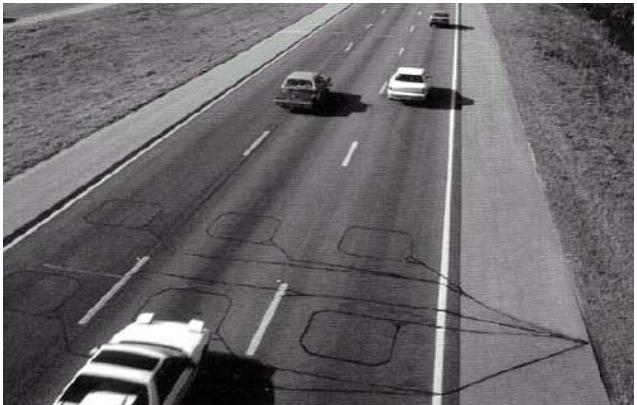
GPS Data

Location Data and Floating-Car Trajectory



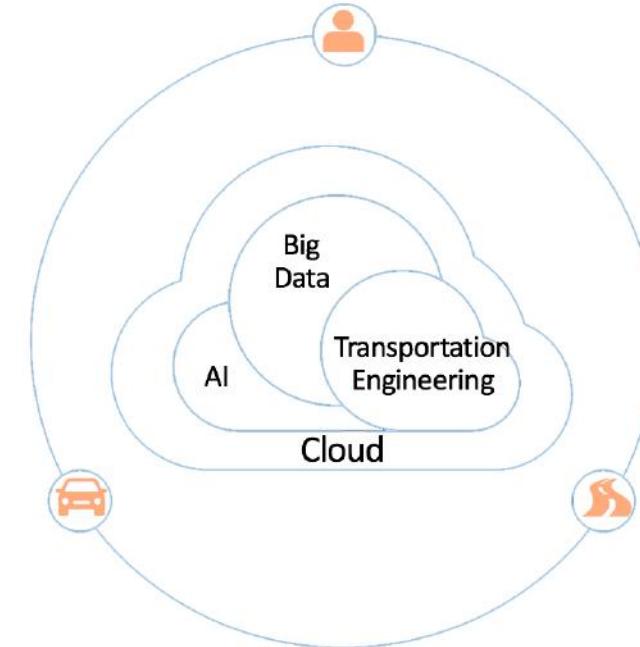
Sensors

Loop detector, camera, microphone, mobile sensors ...

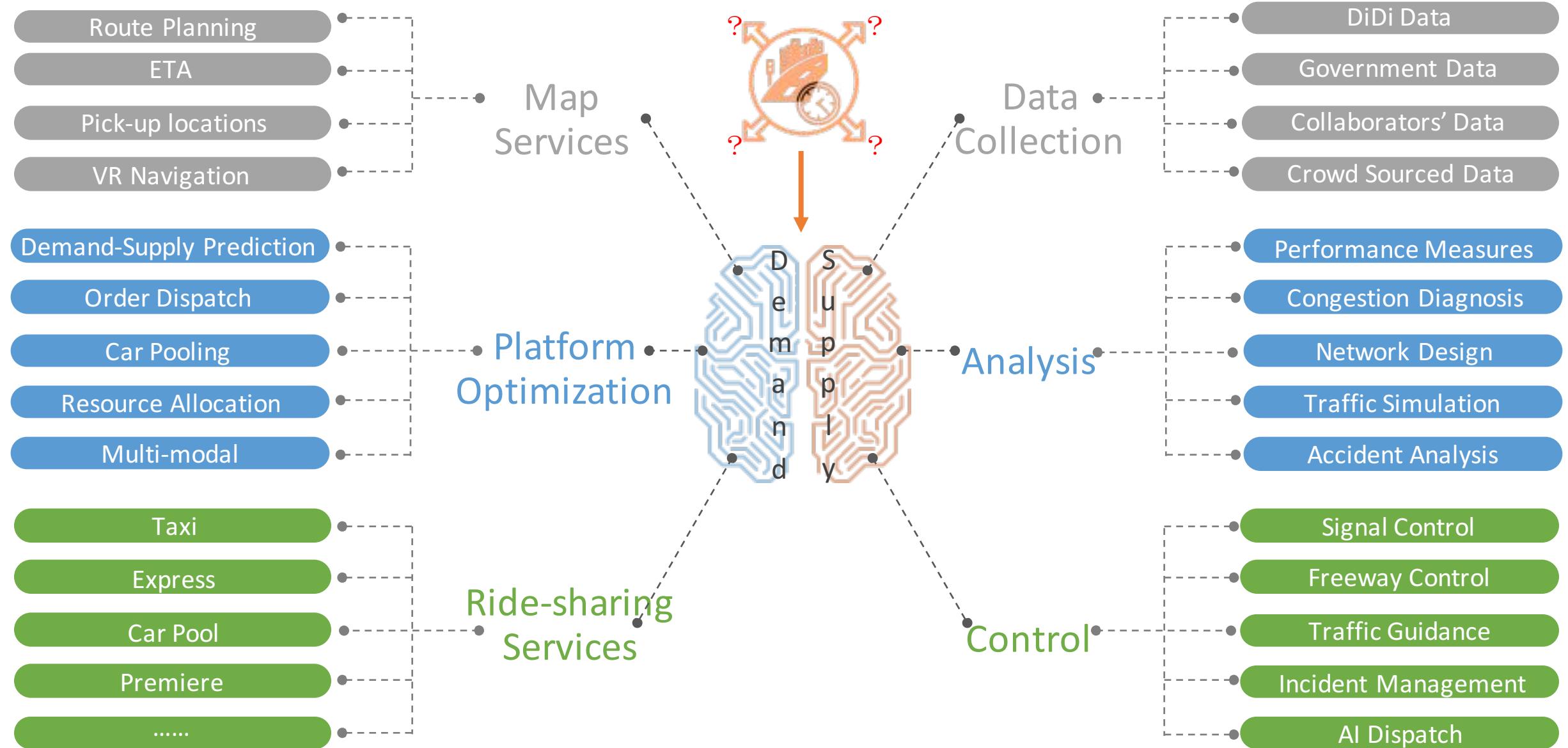


Transportation AI

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)

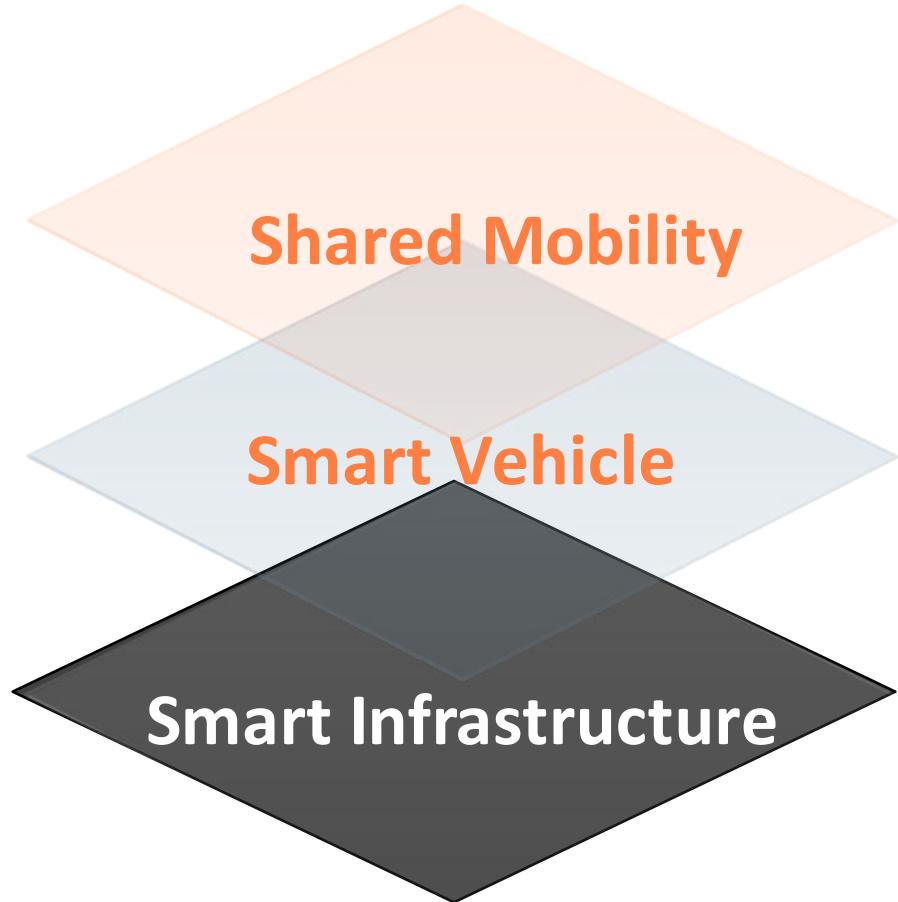
■ Data and tools for transportation AI (15min)

■ Q&A (10min)



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

Navigation



Ride Hailing



Transportation System

Efficiency

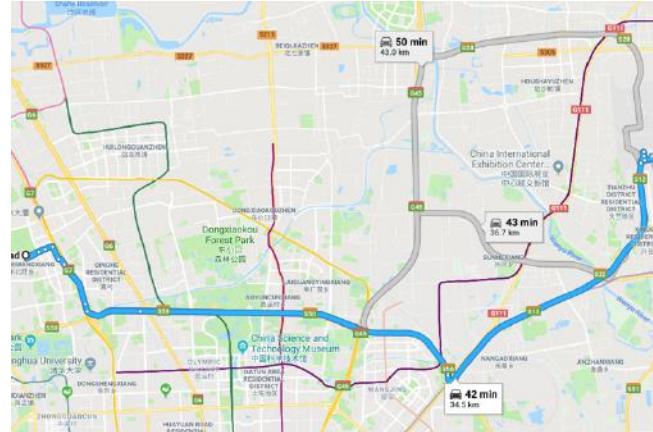


Autonomous Driving



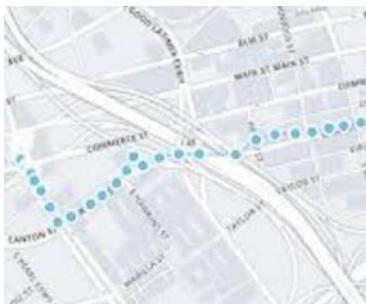
Core Map Service

Route Planning



ETA

Map Matching



Positioning



Traffic



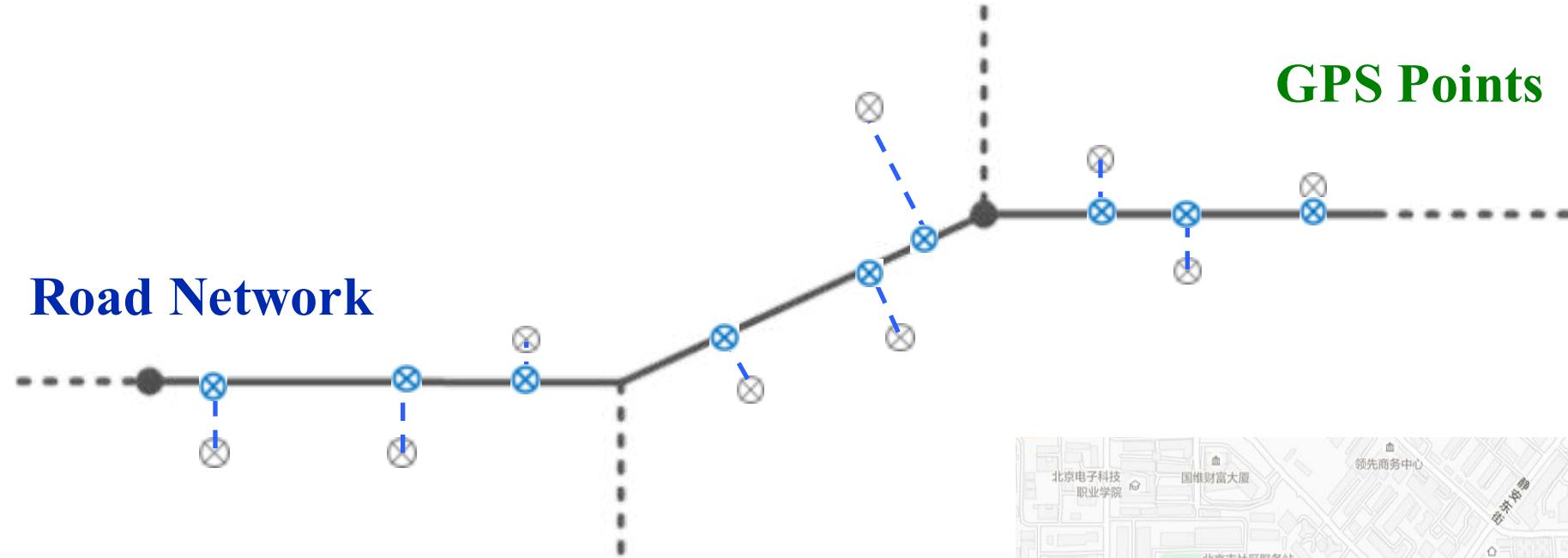


Map Matching

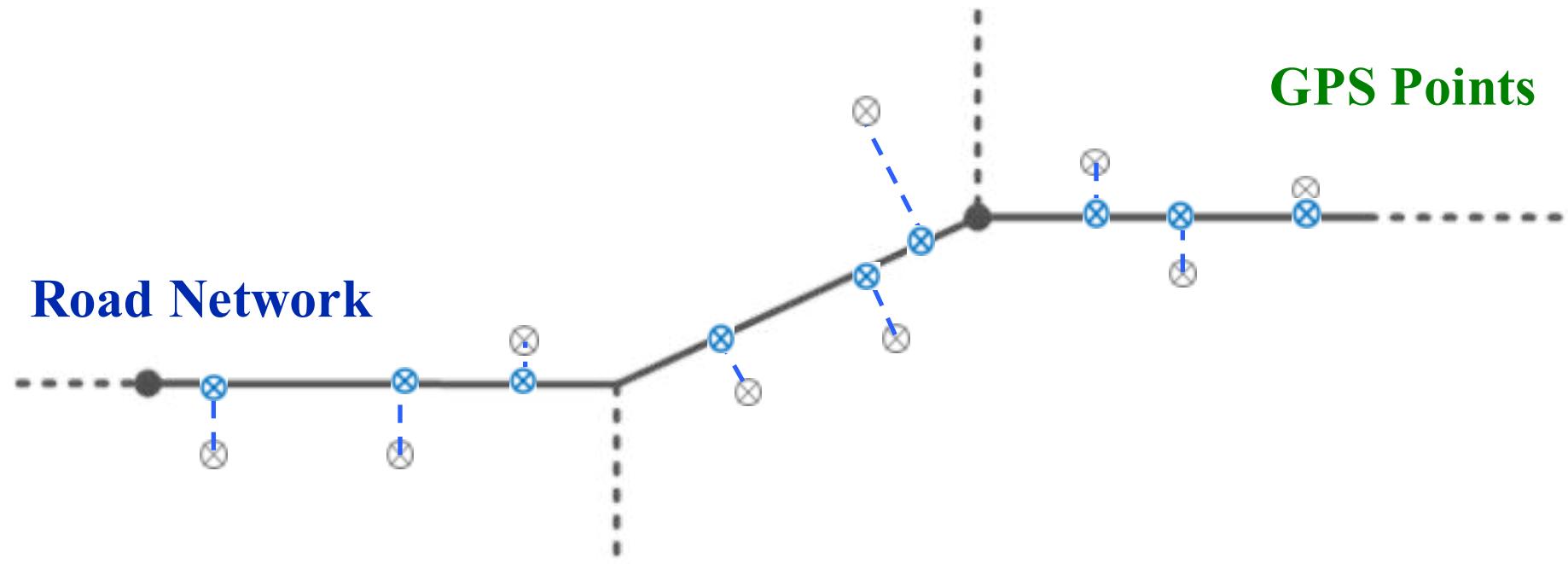
Map Matching

- Problem Definition
- Solutions to Map Matching
- Challenges

Map Matching

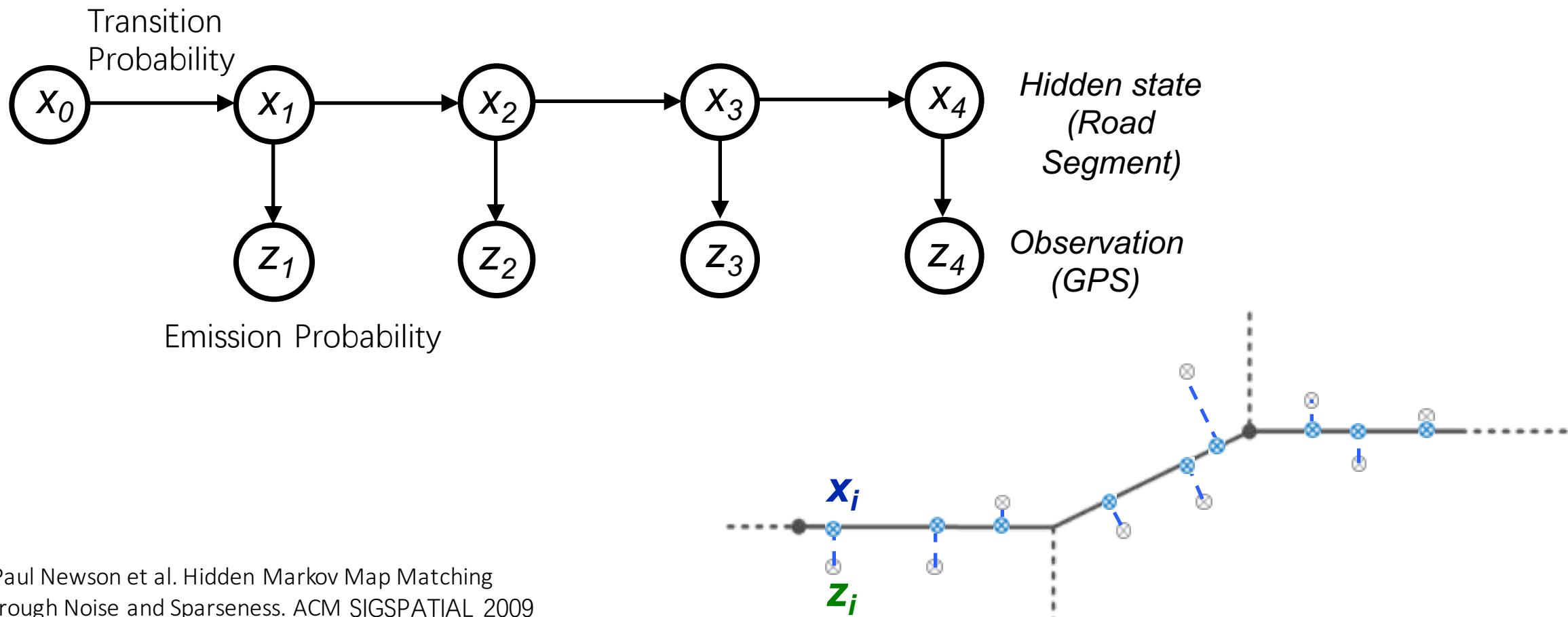


Naive Approach: Nearest Neighbor

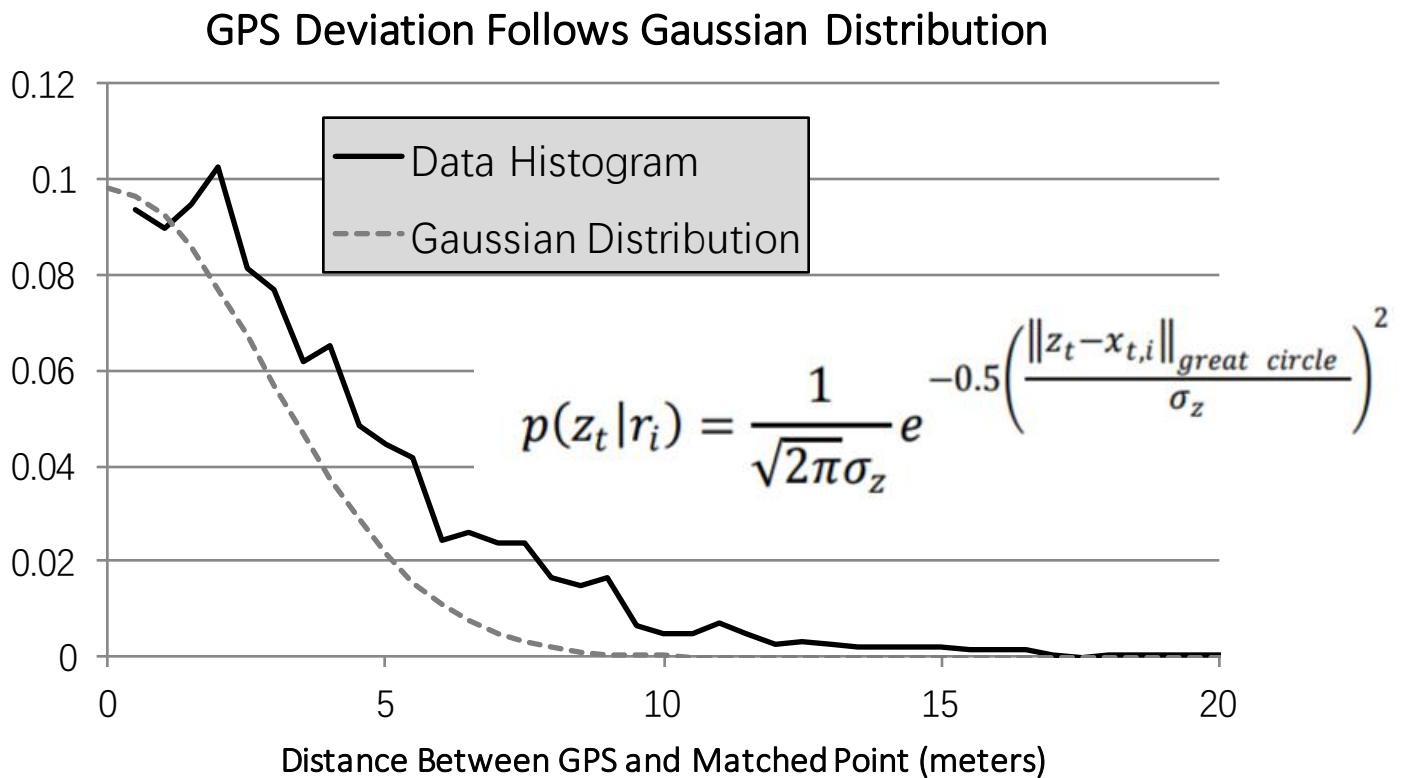
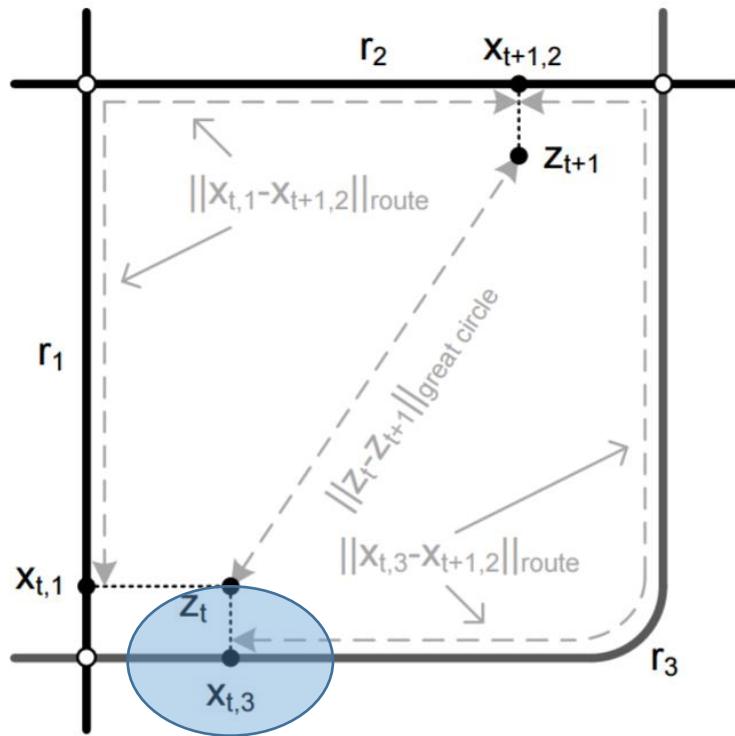


Sequence to Sequence: HMM

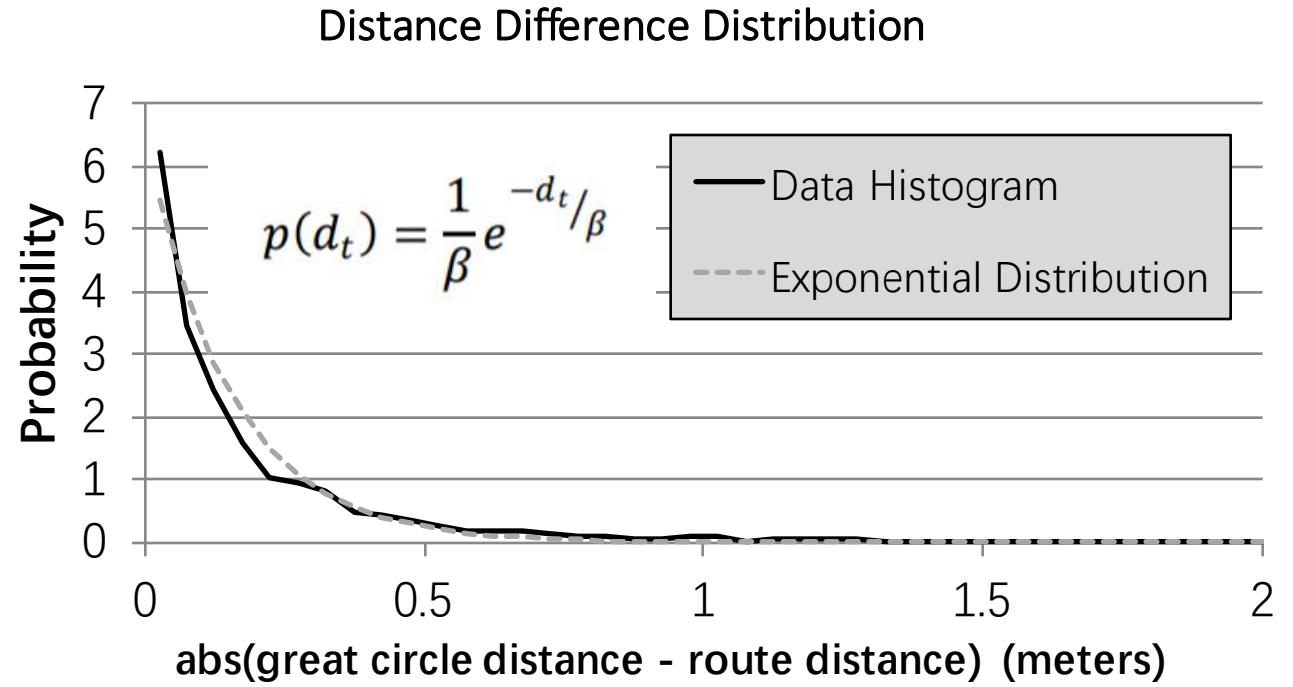
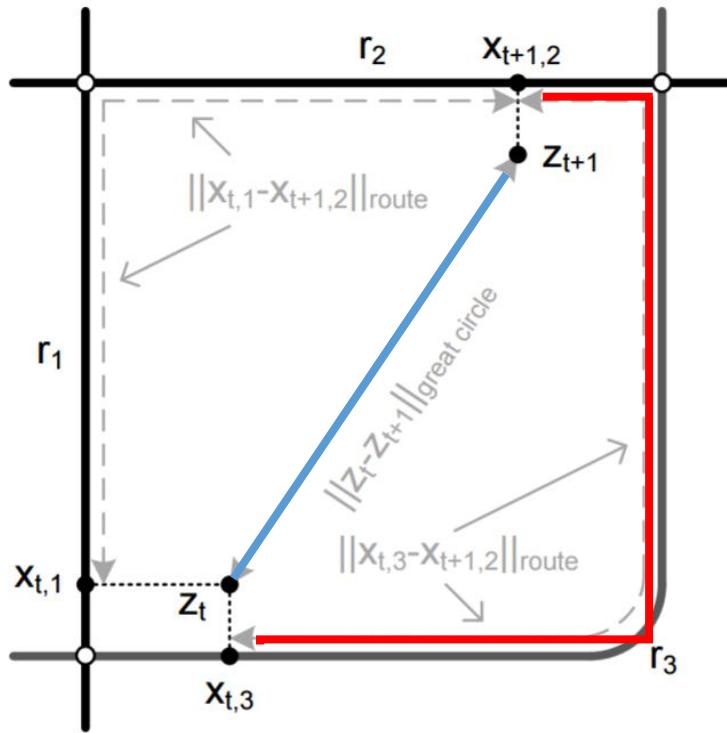
- Model this process using Hidden Markov Model.



Emission Probability



Transition Probability



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

Parameter Estimation

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

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

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

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

- Parameter refinement
- In literature: parameter learning

Parameter Learning with IRL

- Map Matching with Inverse Reinforcement Learning

- Transition probability variant, HMM

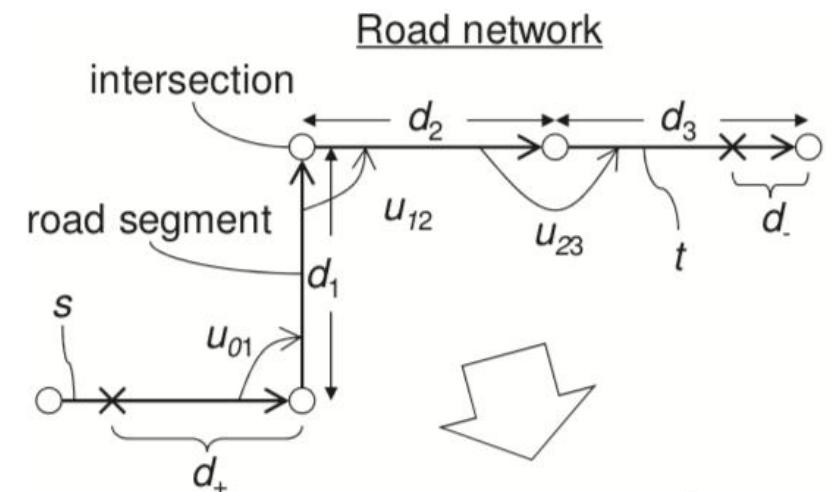
- $P_{trans} = \frac{1}{\beta} \exp(-\frac{|r^* - g_0|}{\beta})$

- Conventionally, $r^* = \|d_1 + d_2 + d_3 + d_+ - d_- \|$

- Here $r^* = \|d_1 + d_2 + d_3 + d_+ - d_- + w_{turn}(u_{01} + u_{12} + u_{23})\|$

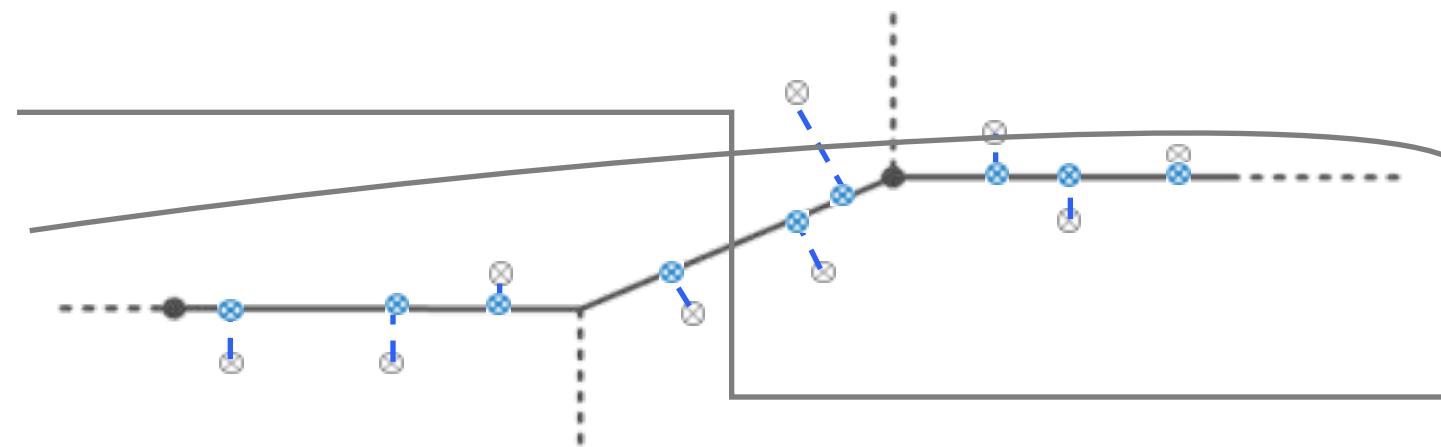
$$u_{vv'} = \begin{cases} 0 & \text{if } |\theta_{vv'}| < \pi/4, \\ 1 & \text{if } \pi/4 \leq |\theta_{vv'}| \leq 3\pi/4, \\ 2 & \text{if } 3\pi/4 < |\theta_{vv'}| < \pi, \\ 10 & \text{if } |\theta_{vv'}| = \pi. \end{cases}$$

- Weight estimation w_{turn} with IRL



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

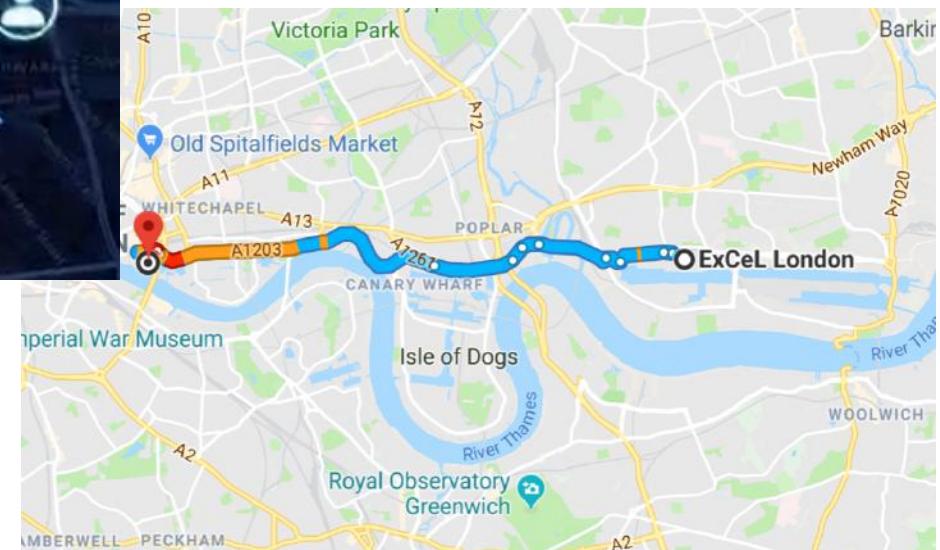
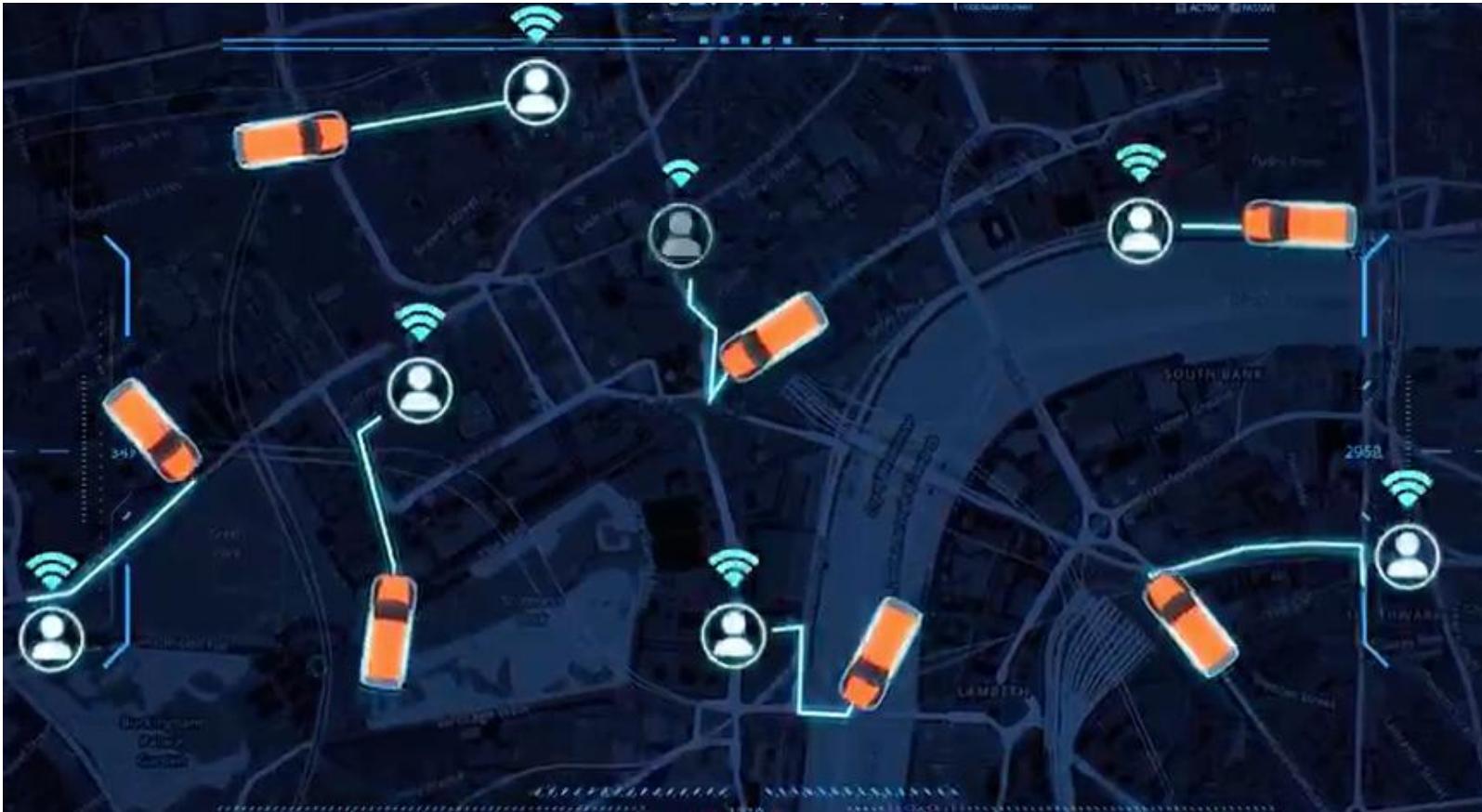
Reference

- 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

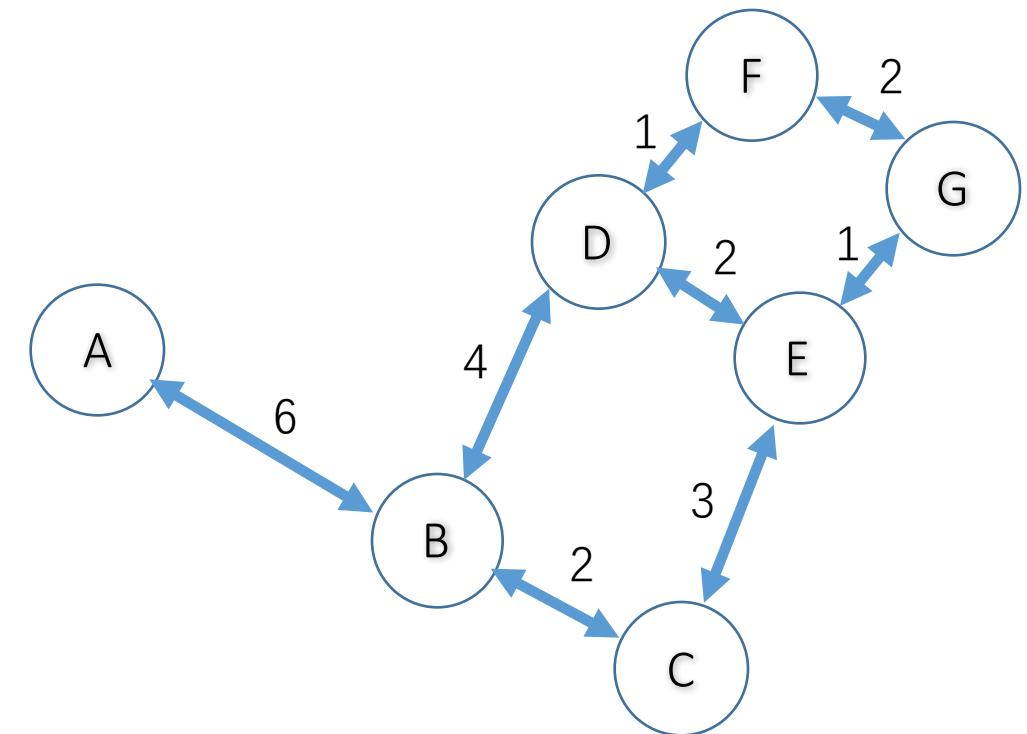


Route Planning

- 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?

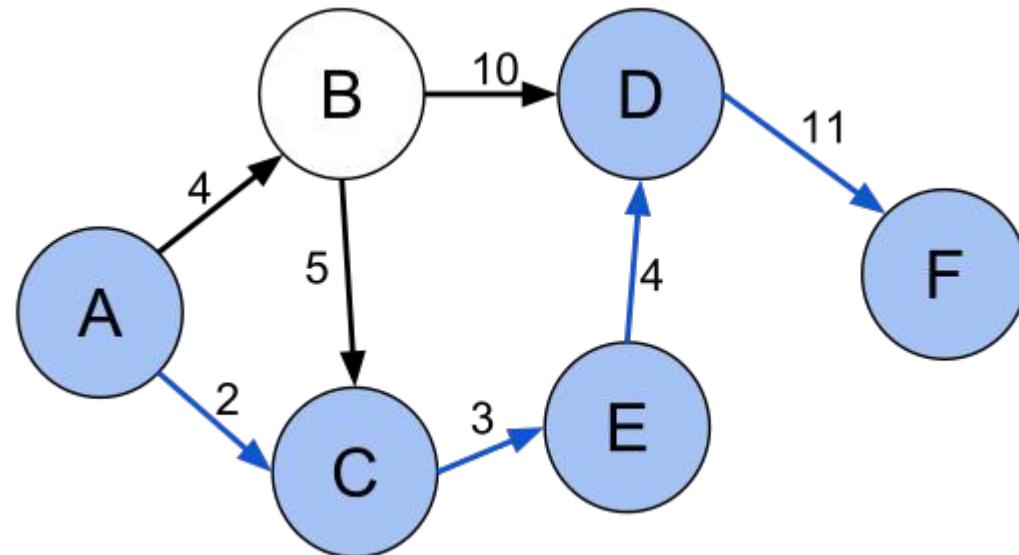
Shortest Path

- 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.



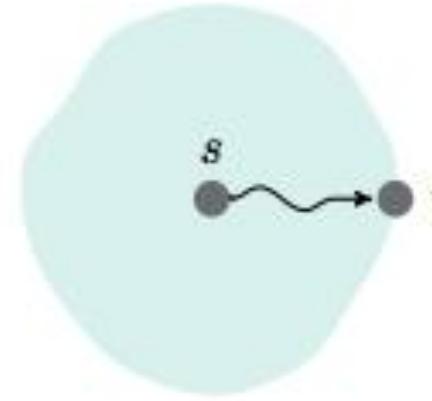
The complexity of Dijkstra's algorithm is $O(|e| + |v|\log |v|)$.

A*-Algorithm: a heuristic approach

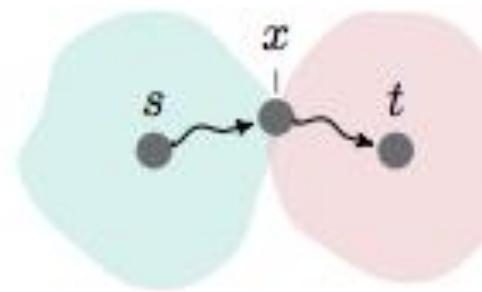
- Search with a heuristic guidance

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

Forward Search



Bidirectional Search



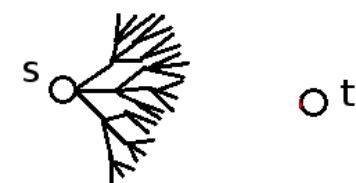
A* Search



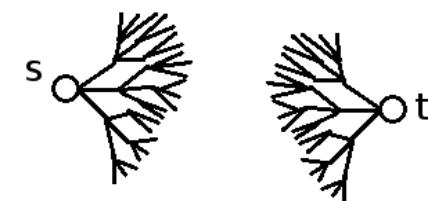
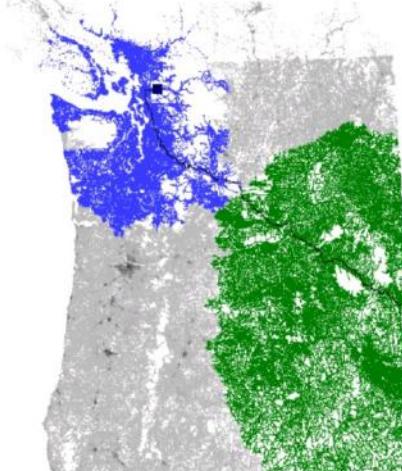
Speedup

- Speedup Dijkstra's Algorithm

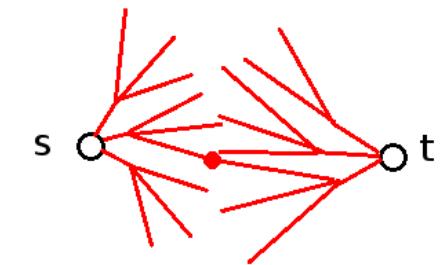
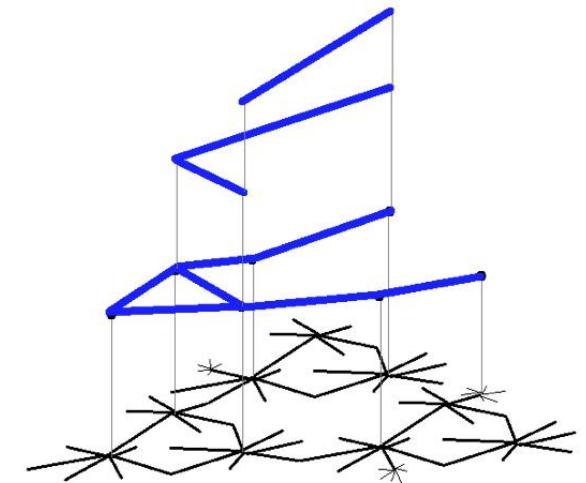
Dijkstra's Algorithm



Bidirectional Dijkstra's Algorithm

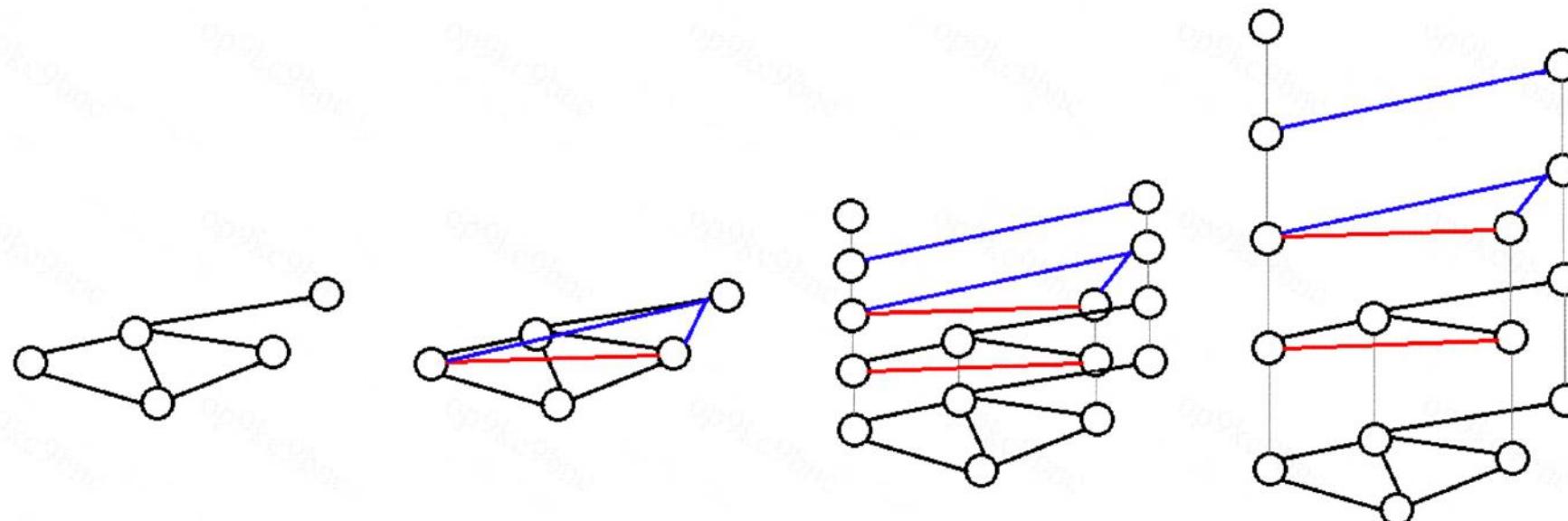


Graph Contraction



Fast Shortest Path

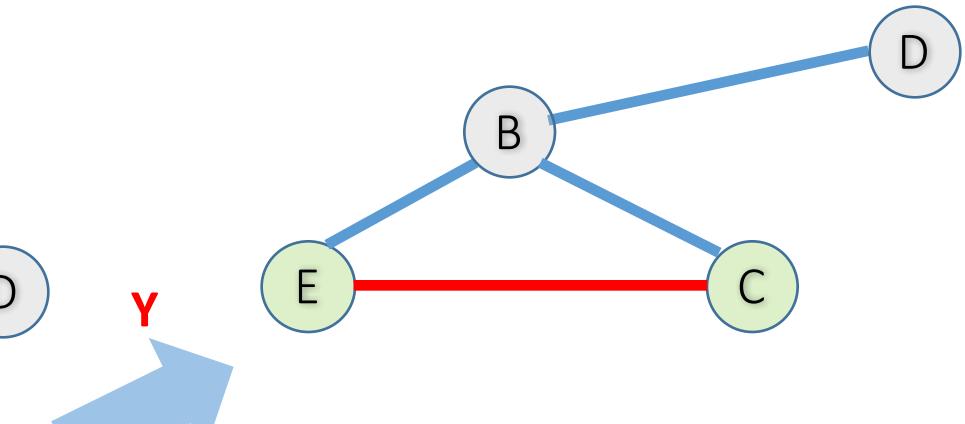
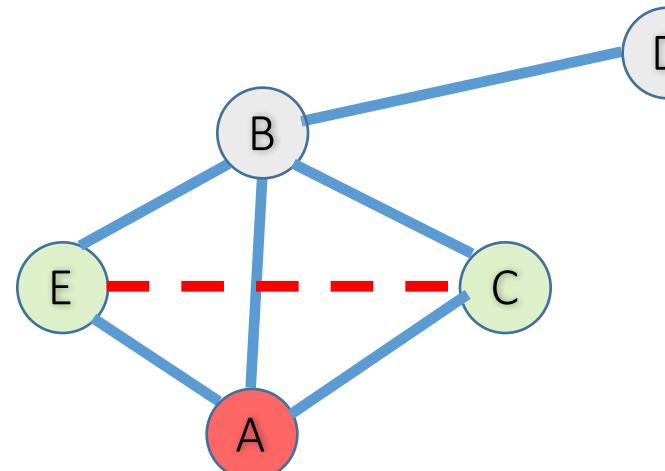
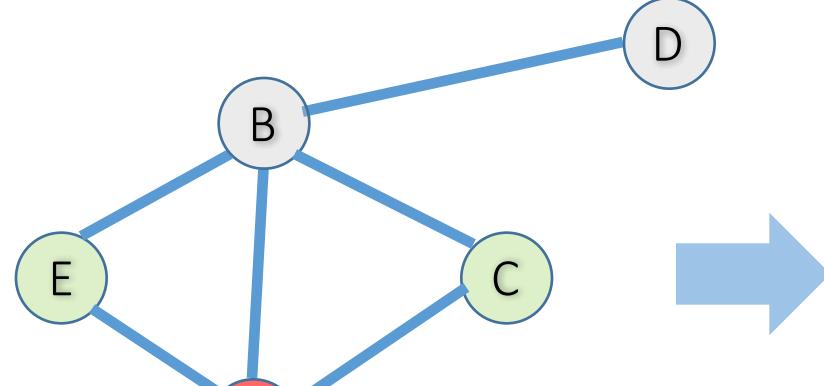
- Contraction Hierarchies (CH)
 - Preprocessing: sorting the nodes, adding shortcut, layer-wise contraction
 - Query: bidirectional Dijkstra's algorithm, unfolding the shortcut



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".

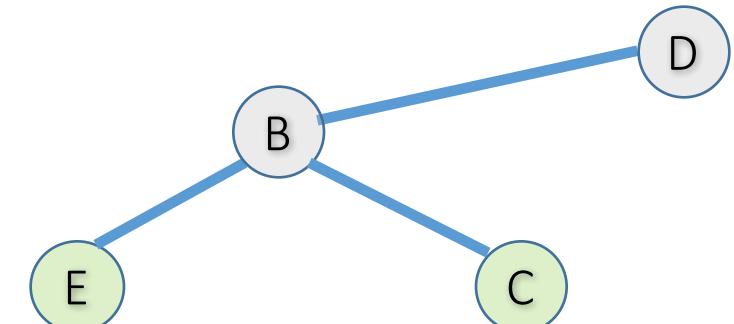
Component Reduction

- Remove nodes and edges w/o shortcut.

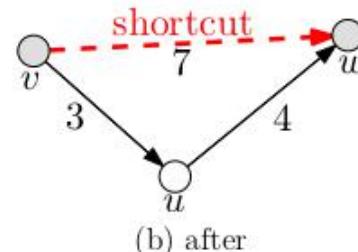
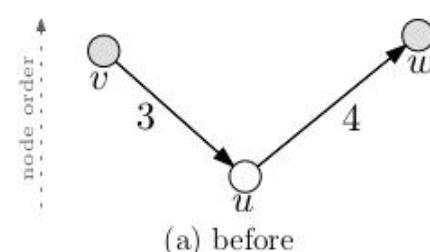


Is A in the shortest path?

N

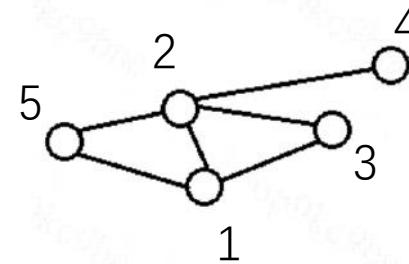


shortcut

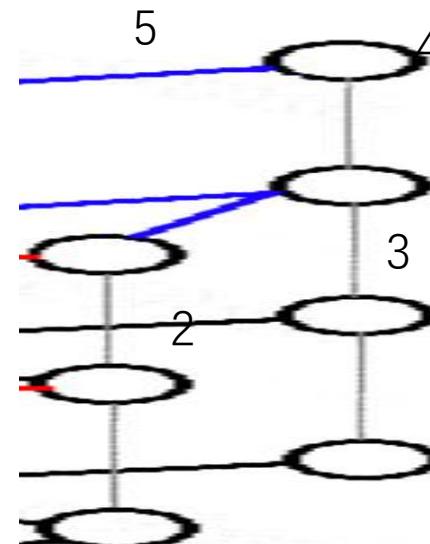


Order for Contraction

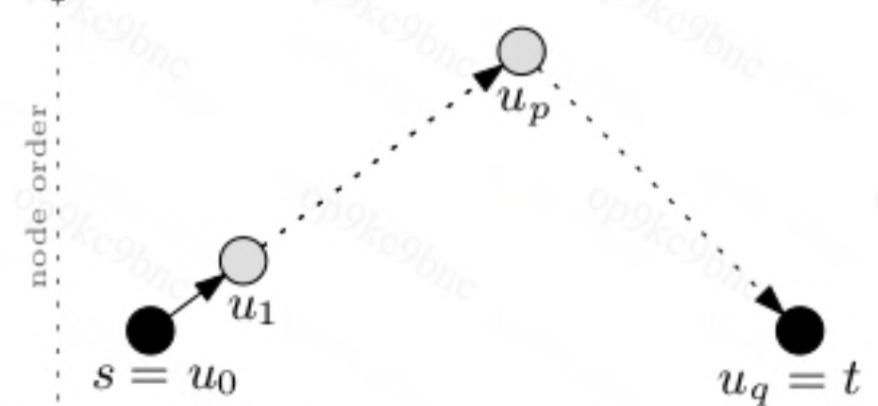
- Setting the node order for contraction: edge difference, cost of contraction, uniformity, cost of queries ...
- Order for contraction and shortest path search.



Node Order

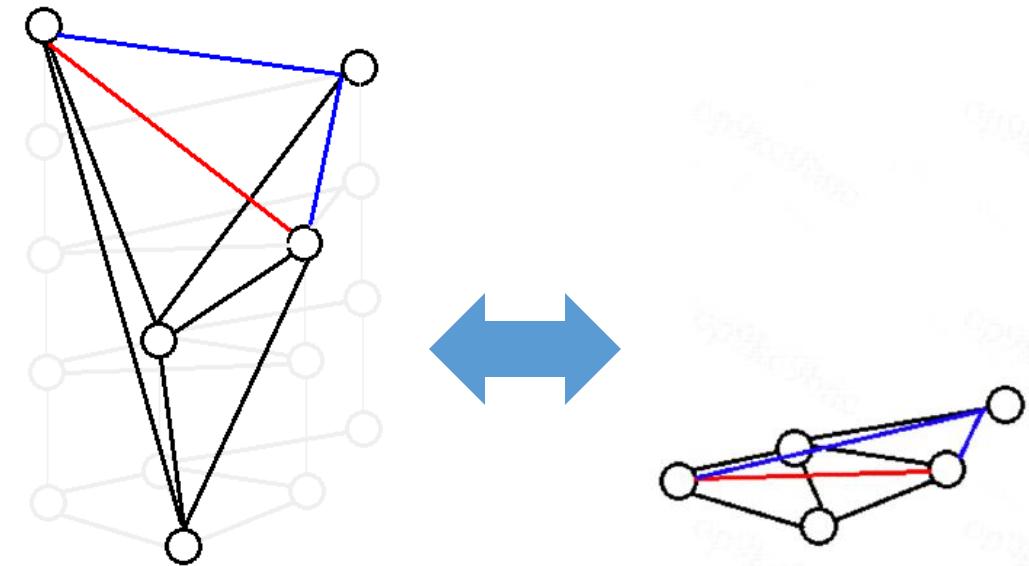
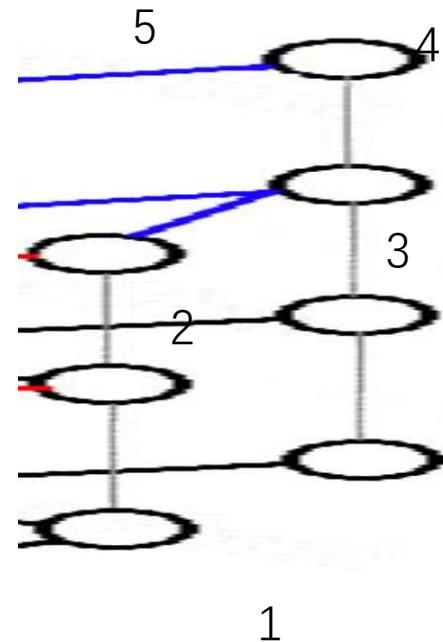
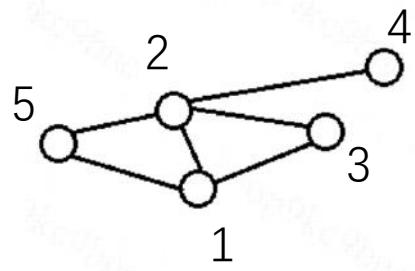


1



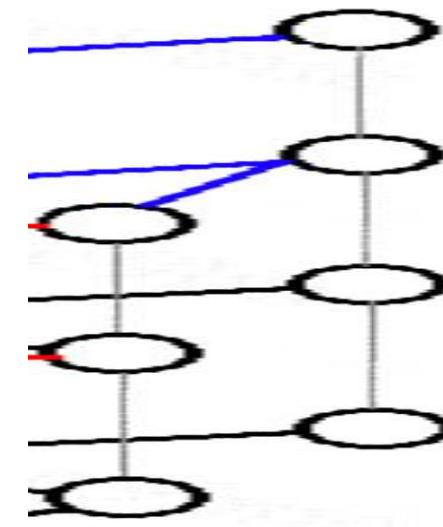
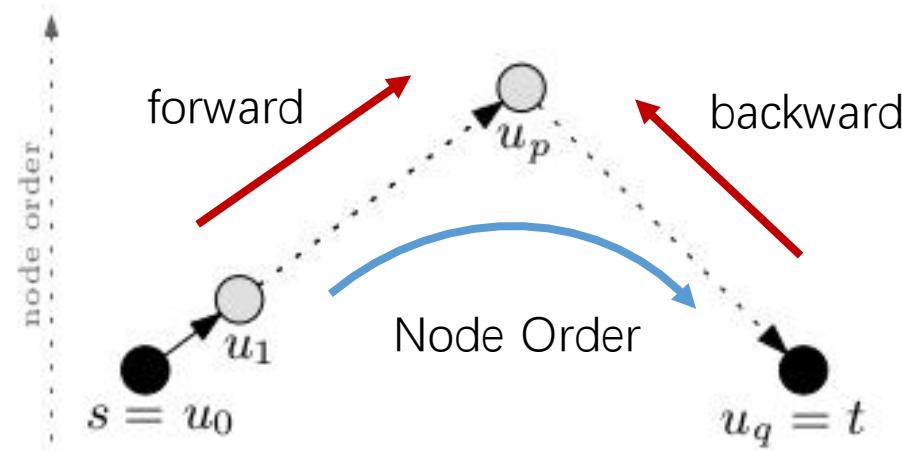
Graph Contraction

- Multi-level contraction



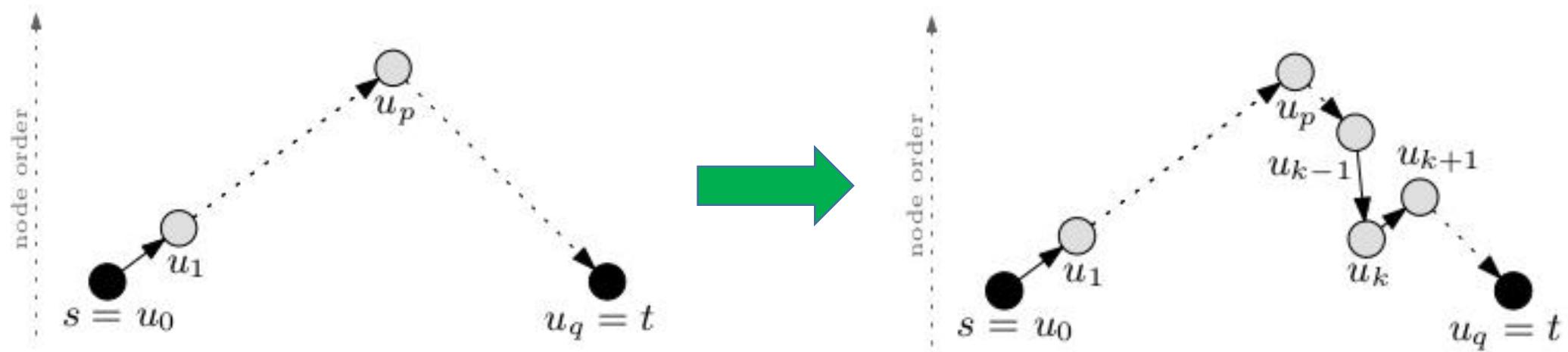
Bidirectional Search

- 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

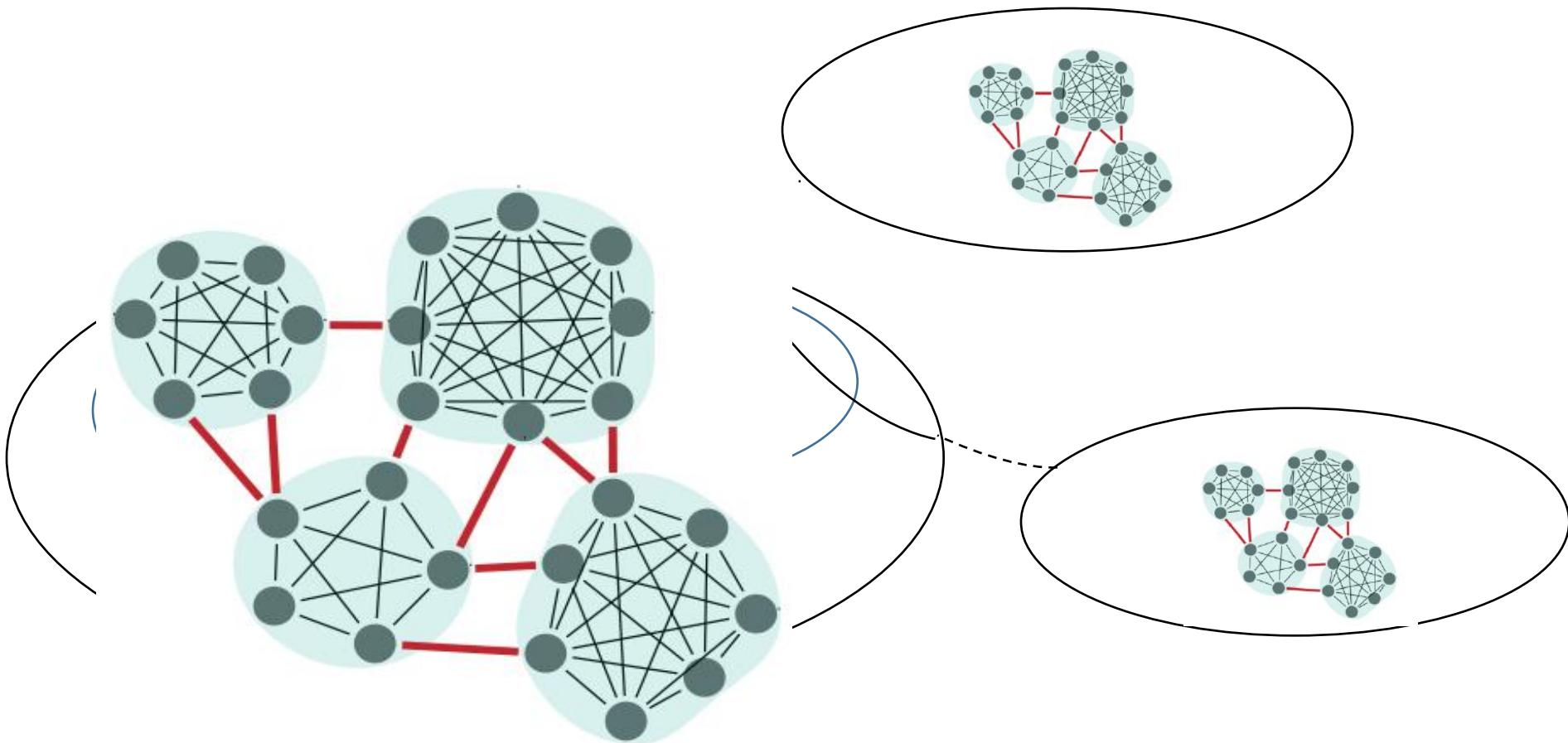


Multiple Metric

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

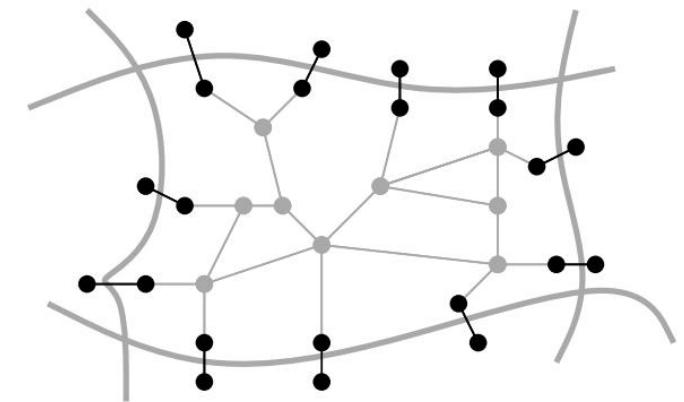
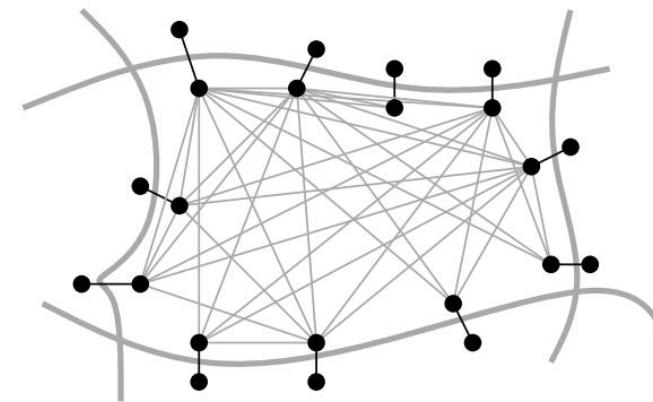
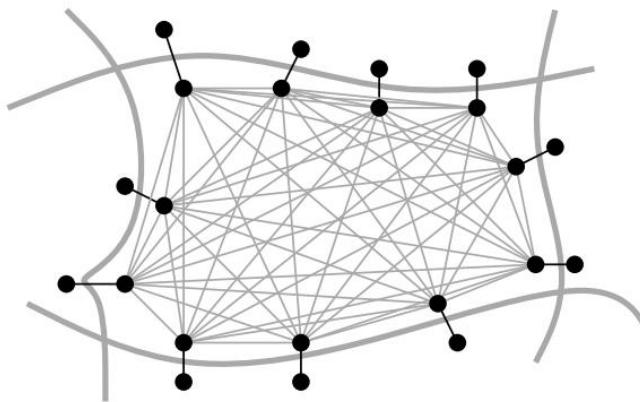
Partition

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



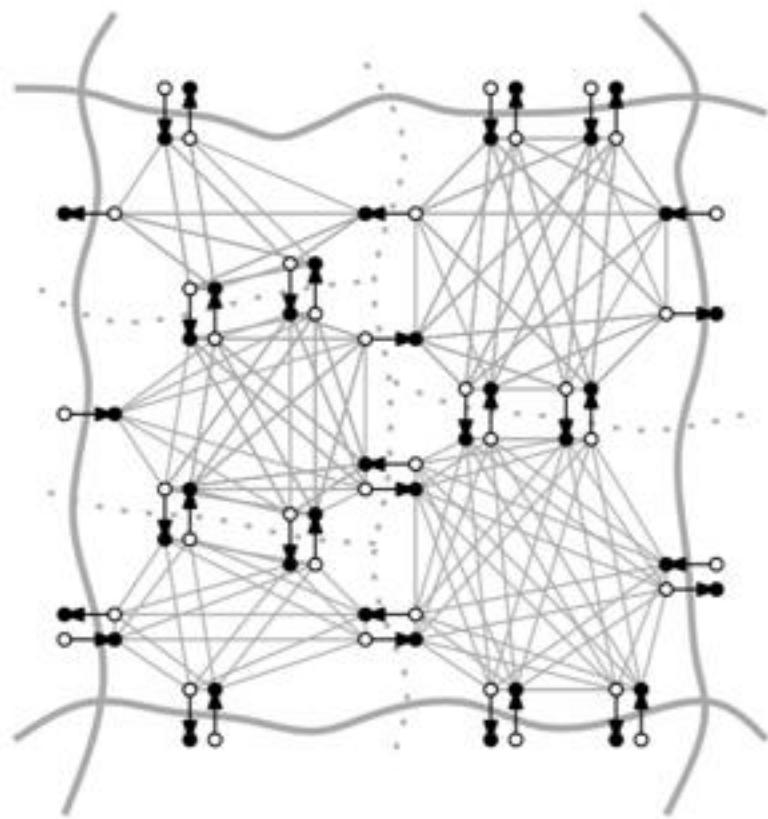
Overlay Graph

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

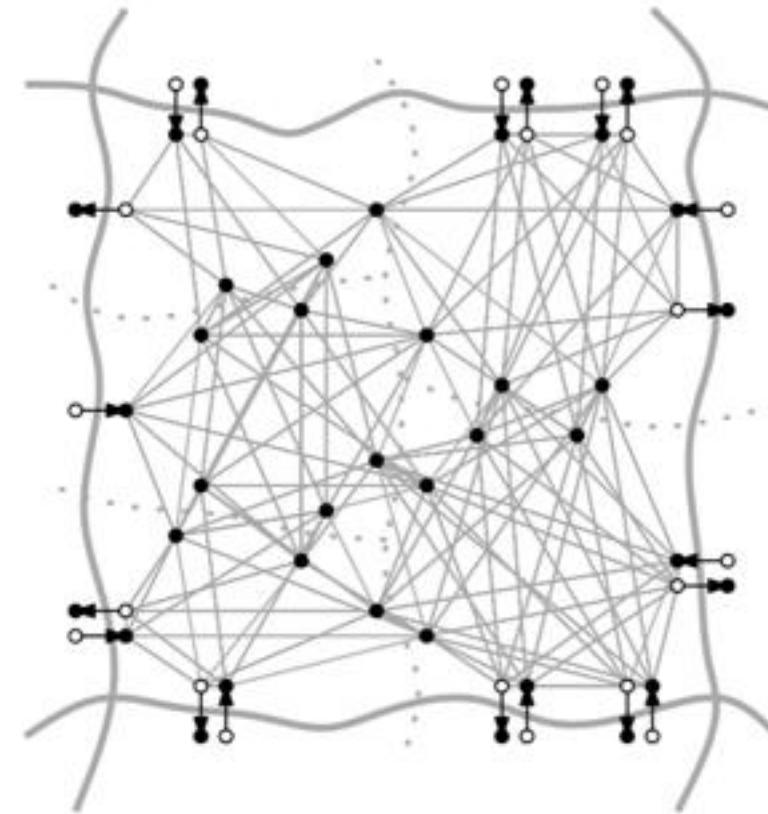


* Picture from Daniel Delling et al. *Customizable Route Planning in Road Networks*. Transportation Science 2017.

Customization



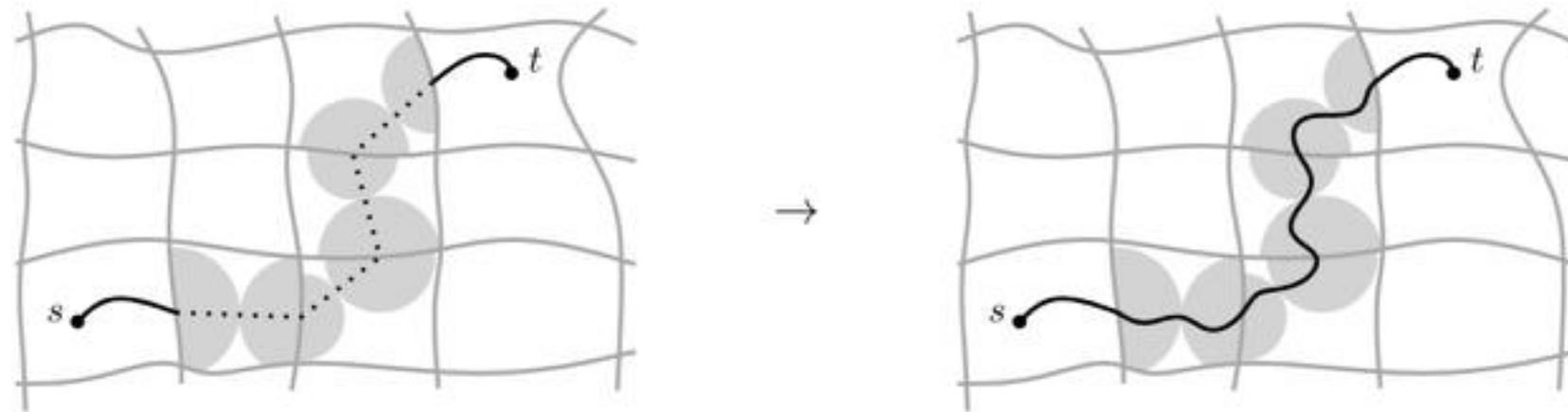
Pruning



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

Query

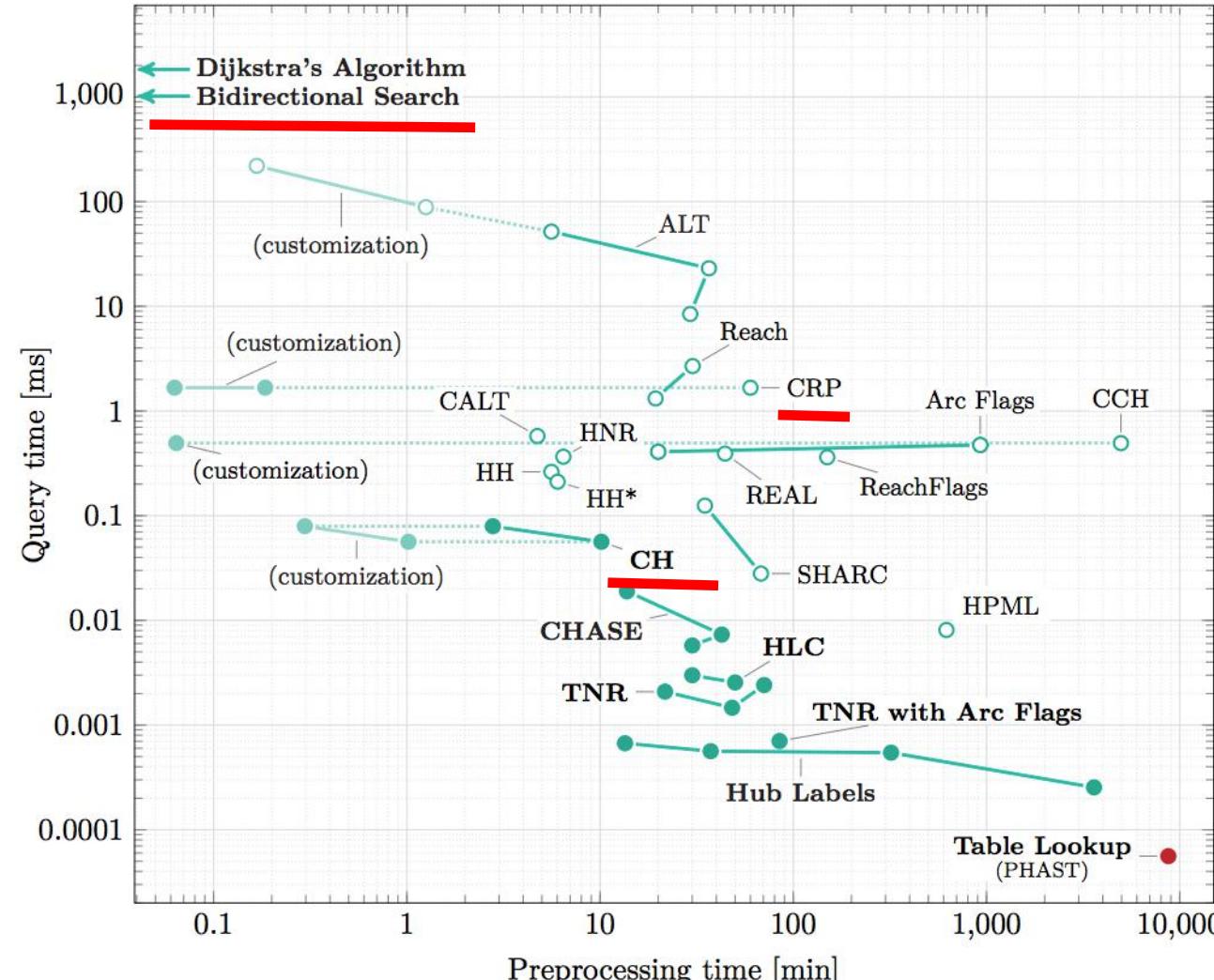
- Shortest path in a subgraph: overlay graph + subgraph including OD nodes.
- Unpack the shortest path: Dijkstra's Algorithm for each clique.



* 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



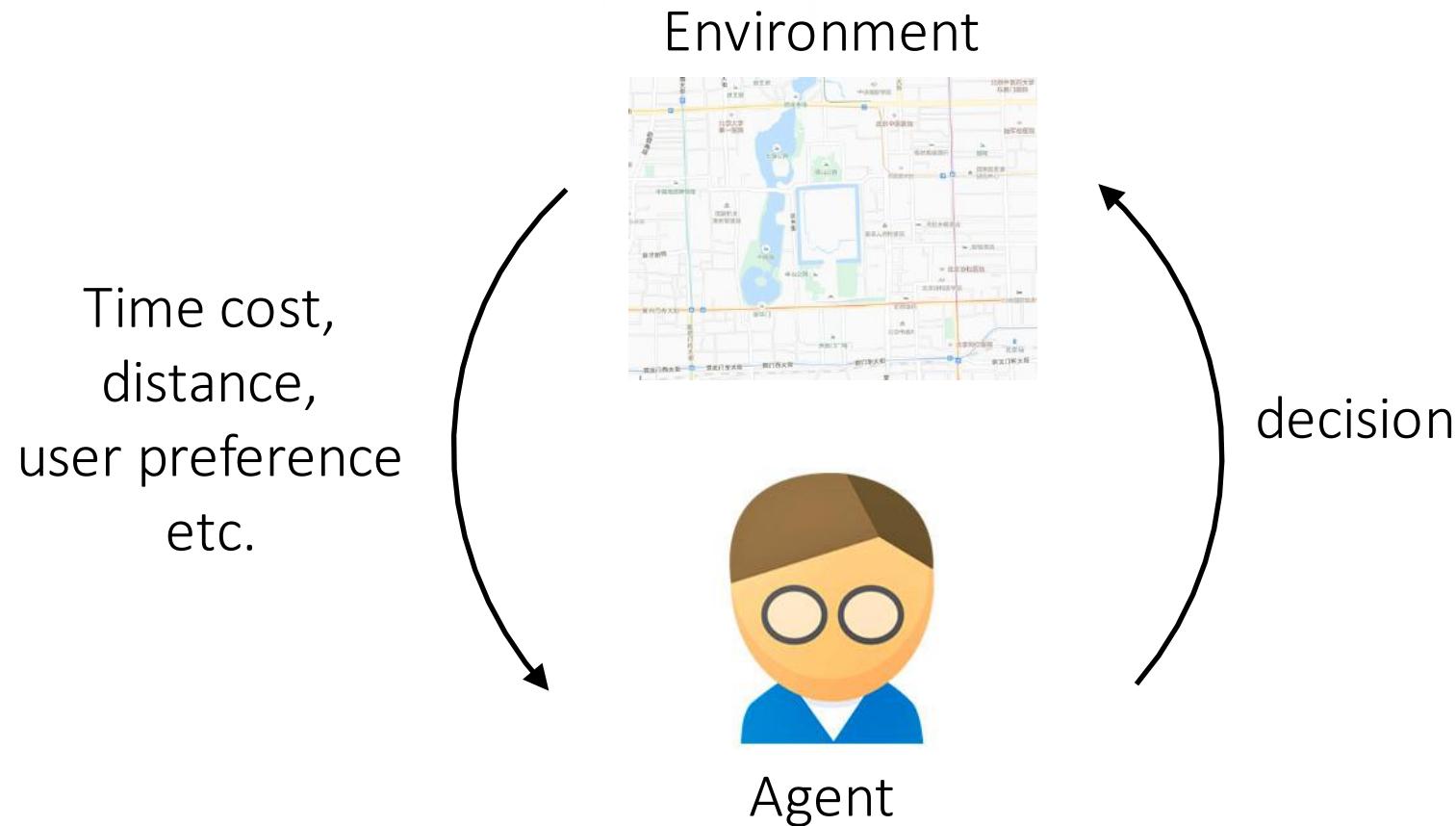
Driver/Passenger
Preference



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

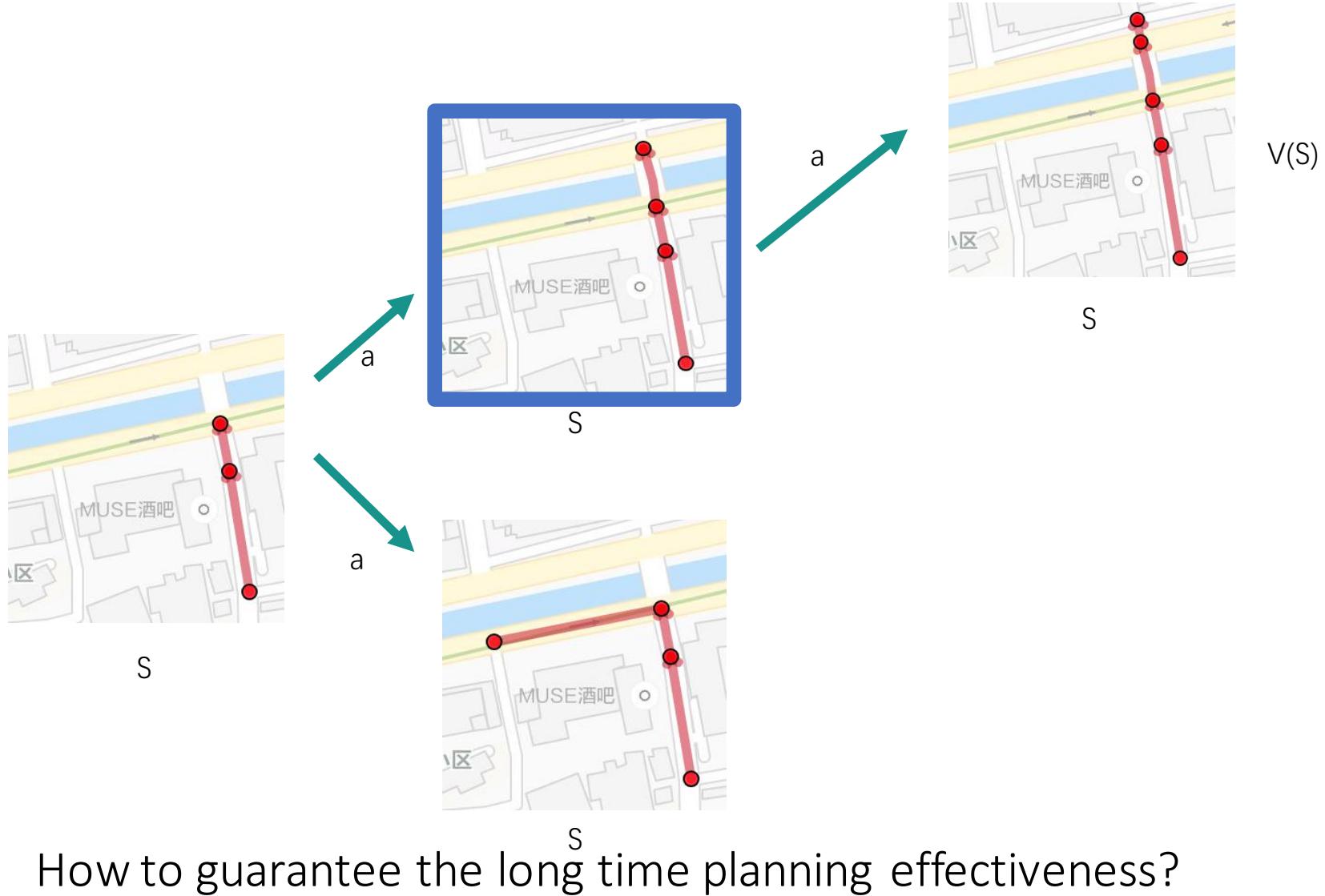
Data Driven



Classification / Reinforcement learning (driving policy) problem

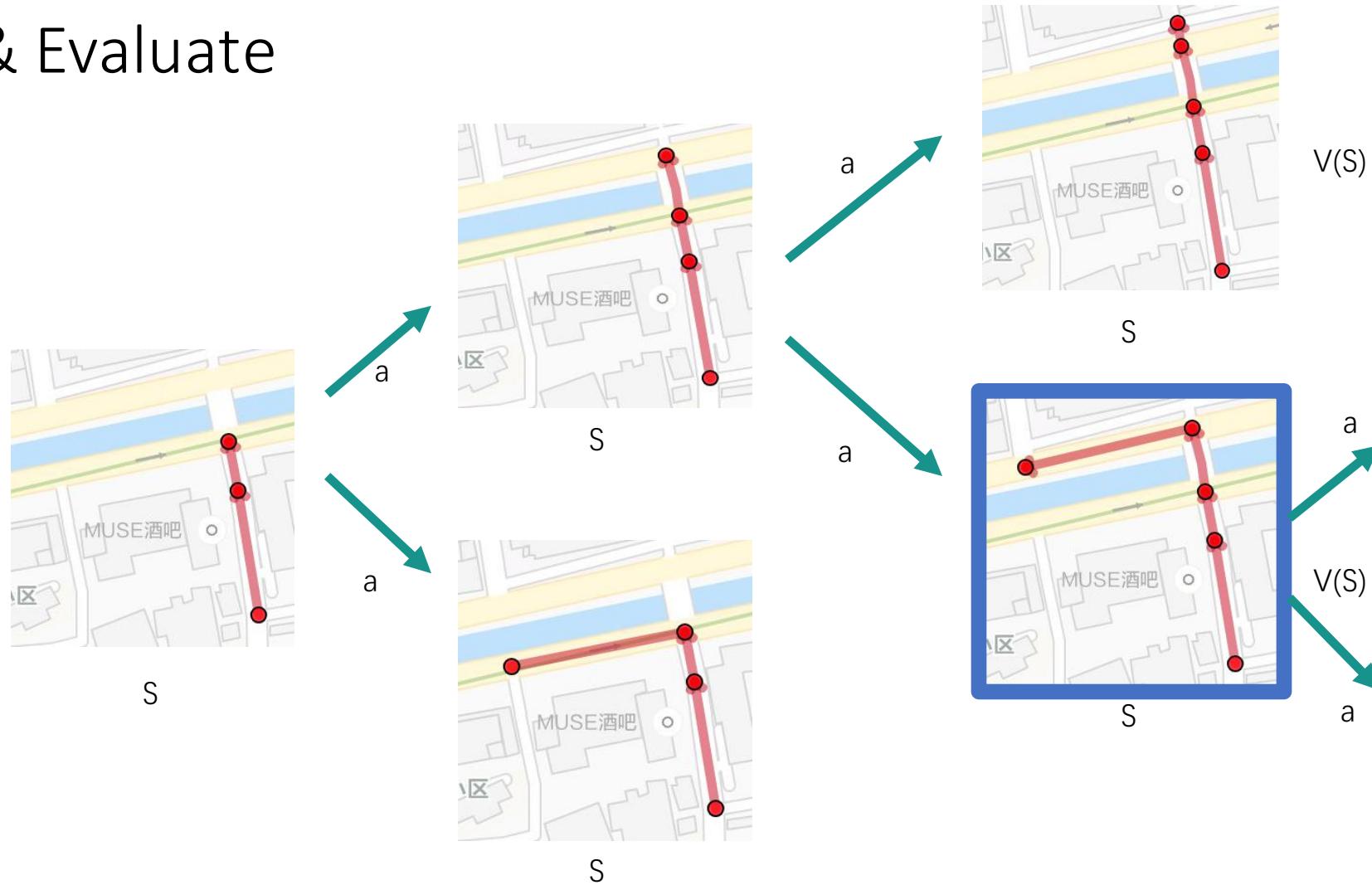
Data Driven

1. Selection



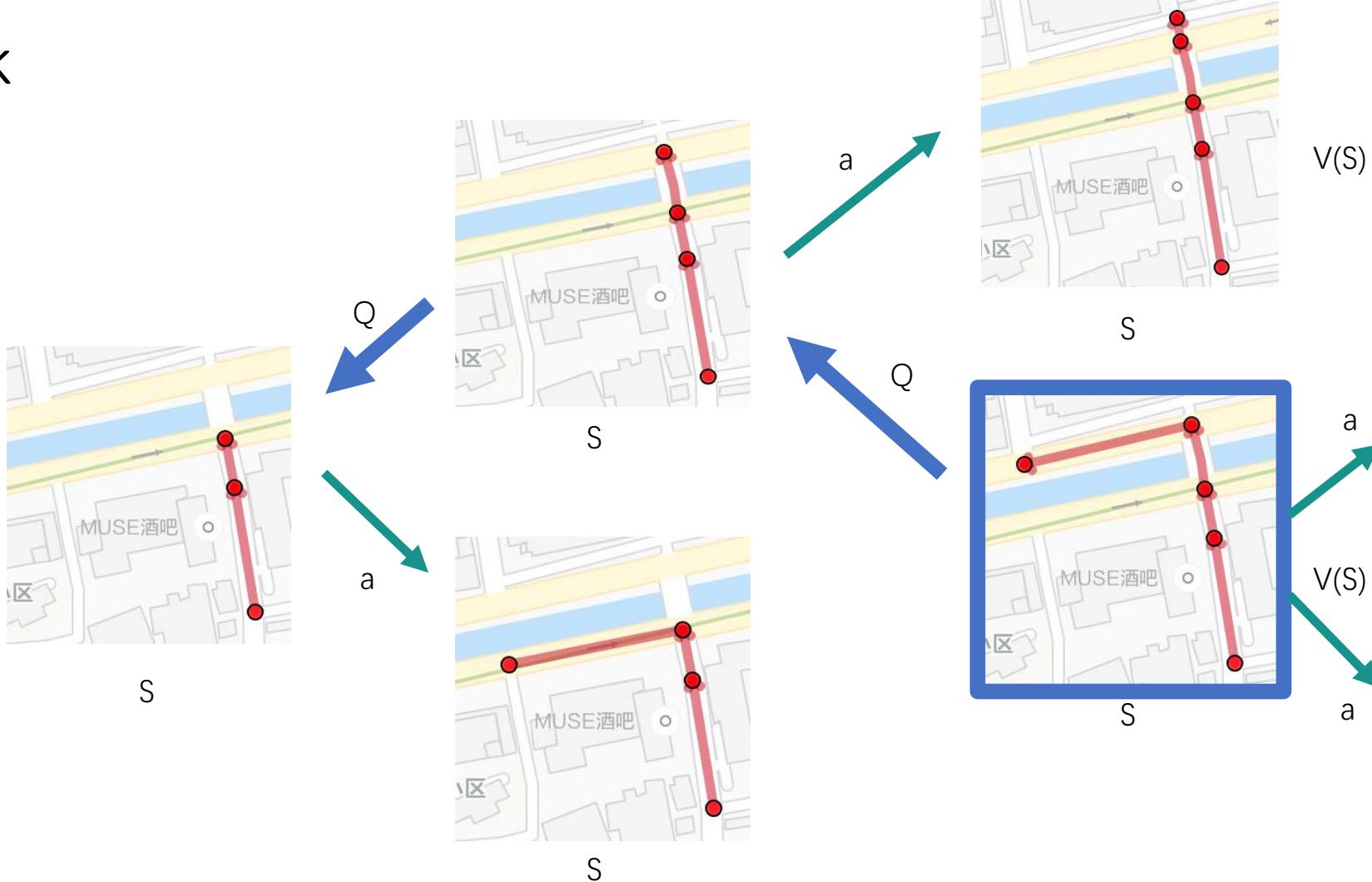
Data Driven

2. Expand & Evaluate



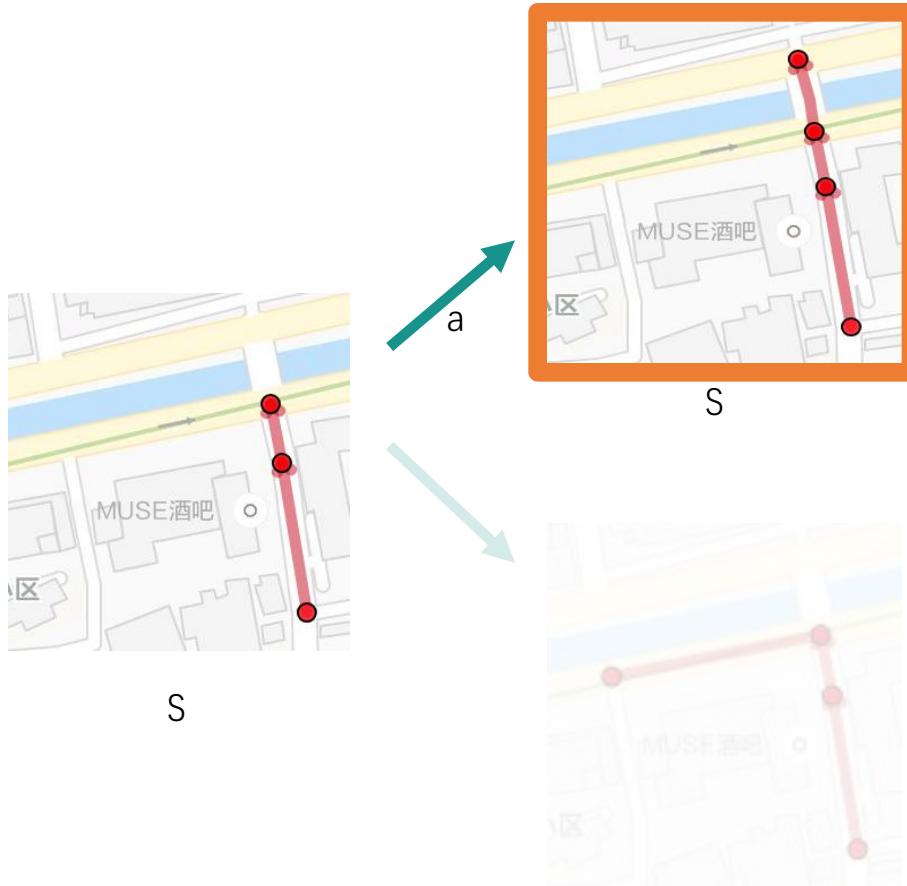
Data Driven

3. Backtrack



Data Driven

4. Execute



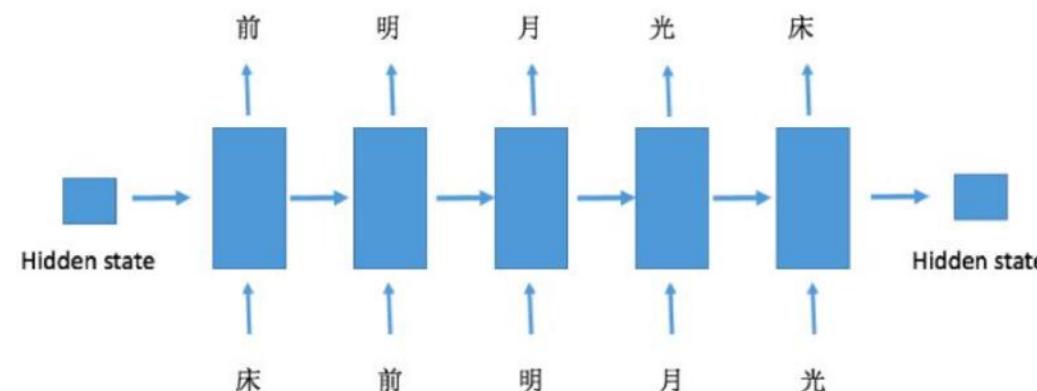
$$L = \sum_t (v(s_t; w) - z_t)^2 - \tilde{\pi}_t \log \pi(a|s_t; \theta) + c\|\theta\|^2 + c\|w\|^2$$

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.



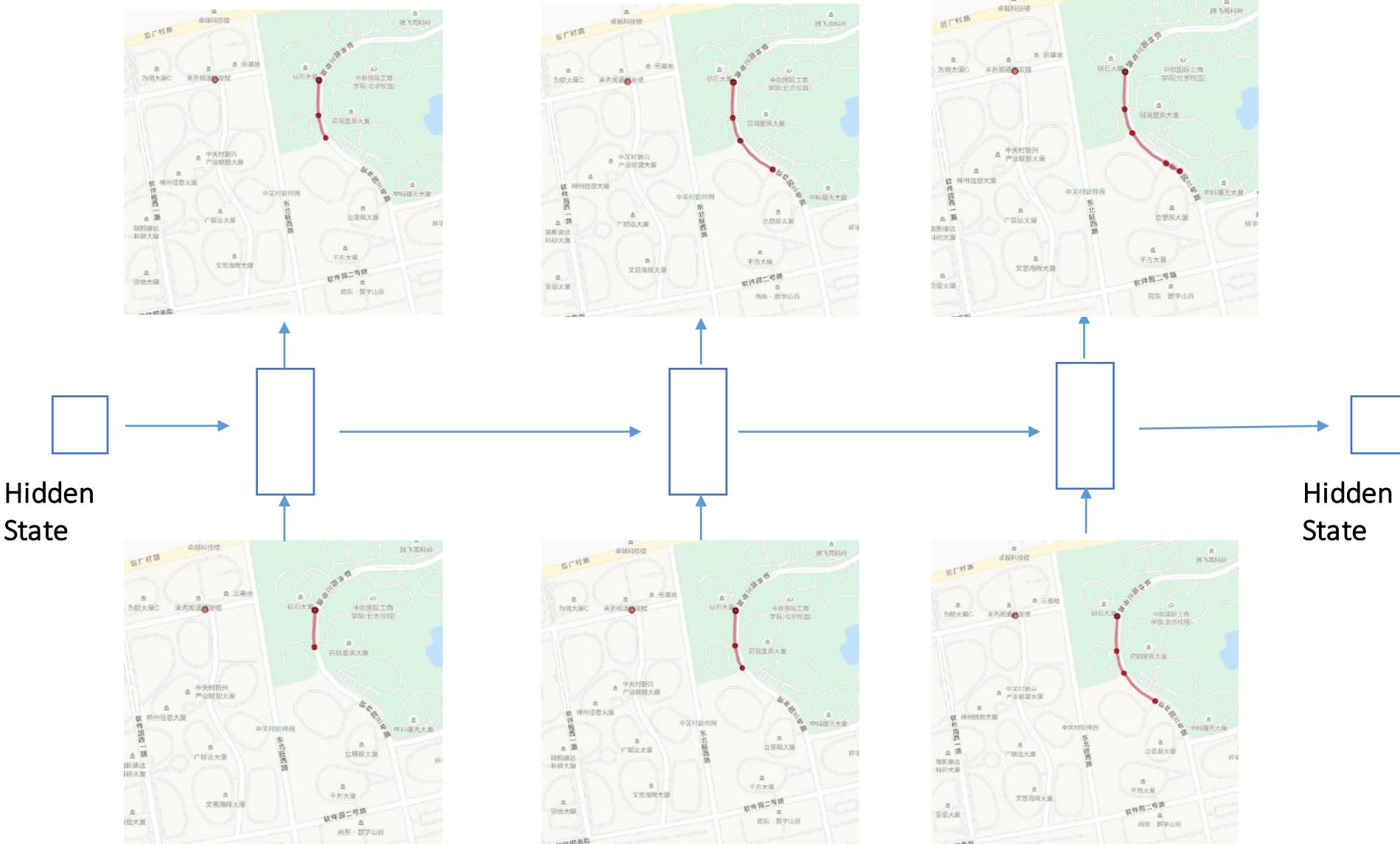
Modeling Trajectories with Recurrent Neural Networks

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

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

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

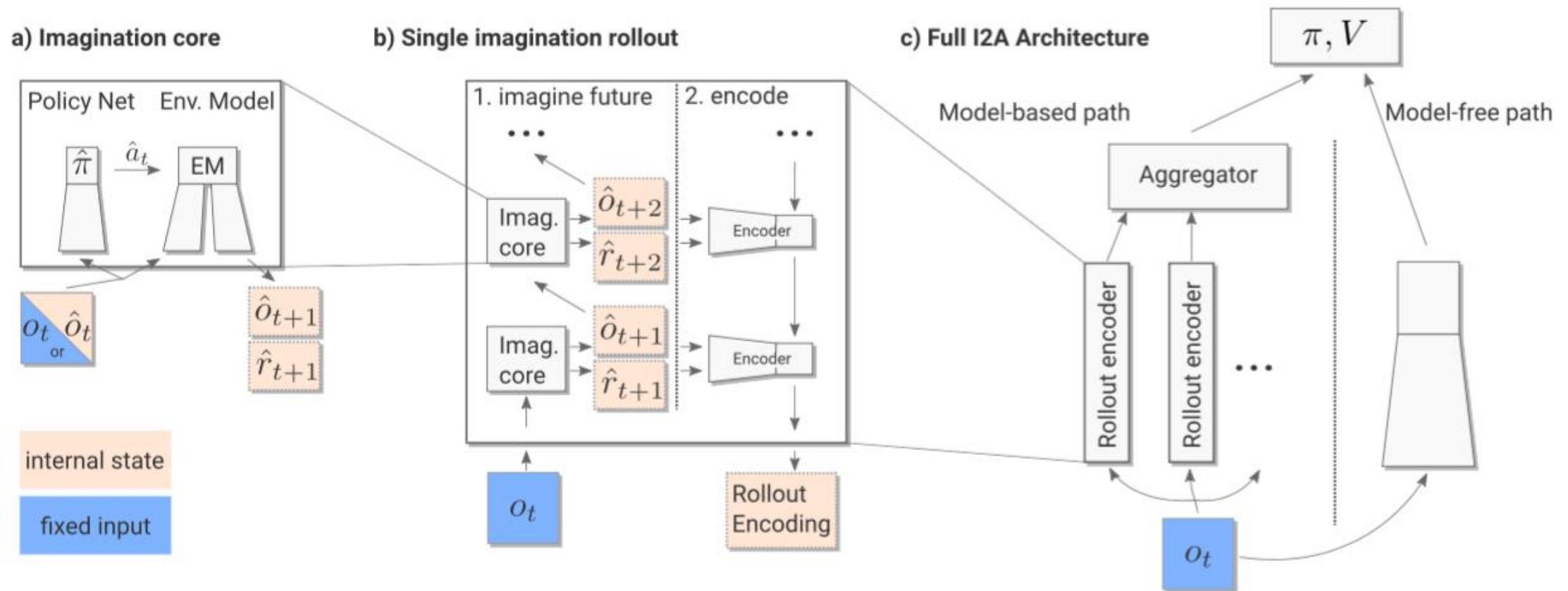
Modeling Trajectories with Recurrent Neural Networks



Modeling Trajectories with Recurrent Neural Networks

Task	With Destination							
	PT _{small}		PT _{large}		SH _{small}		SH _{large}	
Dateset	NLL	ACC	NLL	ACC	NLL	ACC	NLL	ACC
Metric								
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

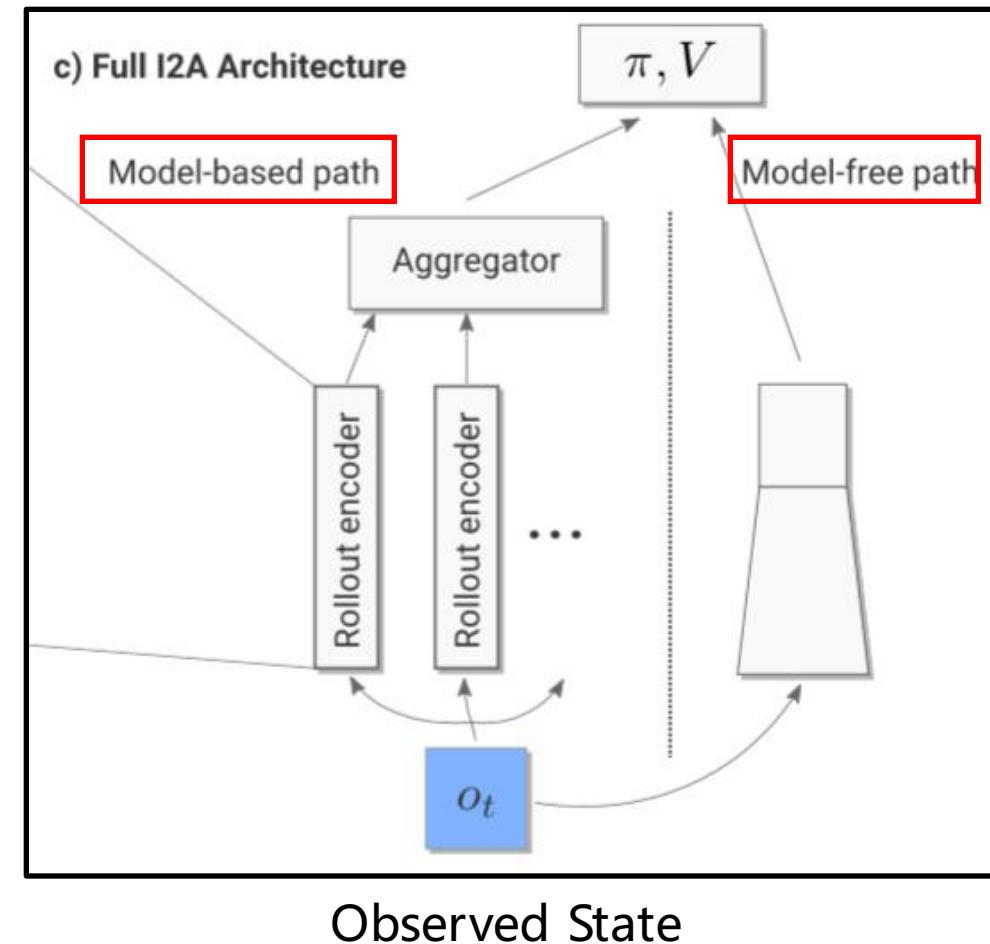
Disadvantages

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

Model-Free Approaches

Disadvantages

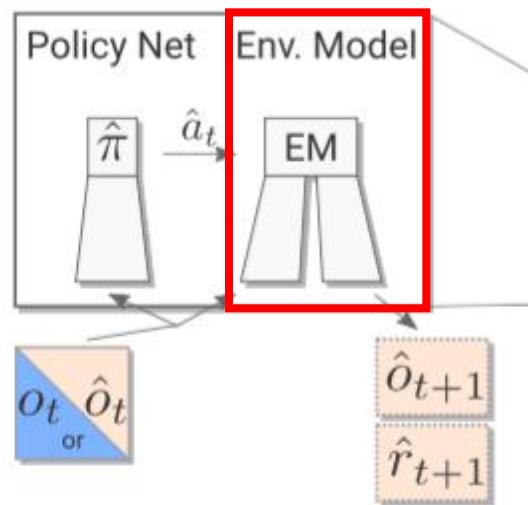
- Requires **large** amounts of training data
- Resulting policies do not readily **generalize** to novel tasks in the same environment



Route Planning with Reinforcement Learning

Key of Model-Based method — Environment Model

a) Imagination core



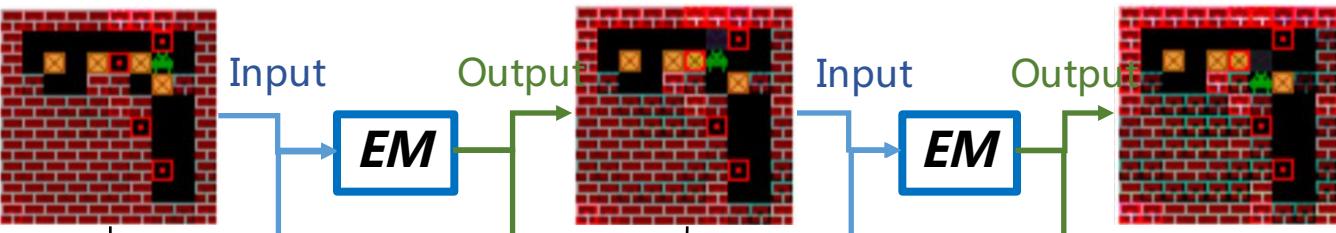
internal state

fixed input

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

e.g.

State



Action

Reward

$10 + (-1)$
Box reach target 10
Step cost -1

Step cost -1

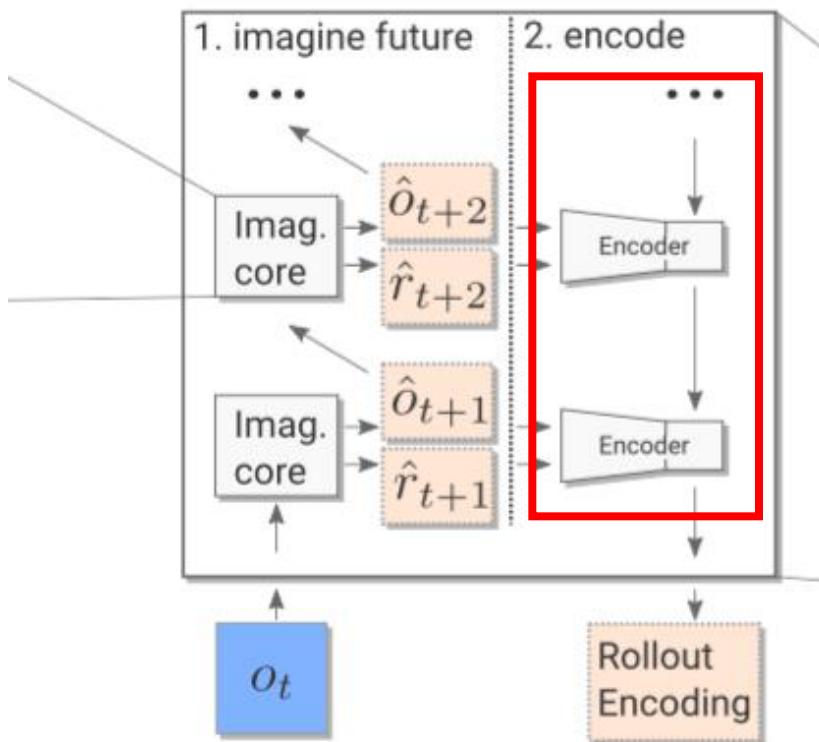
Environment Model Construction Options

- 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

b) Single imagination rollout

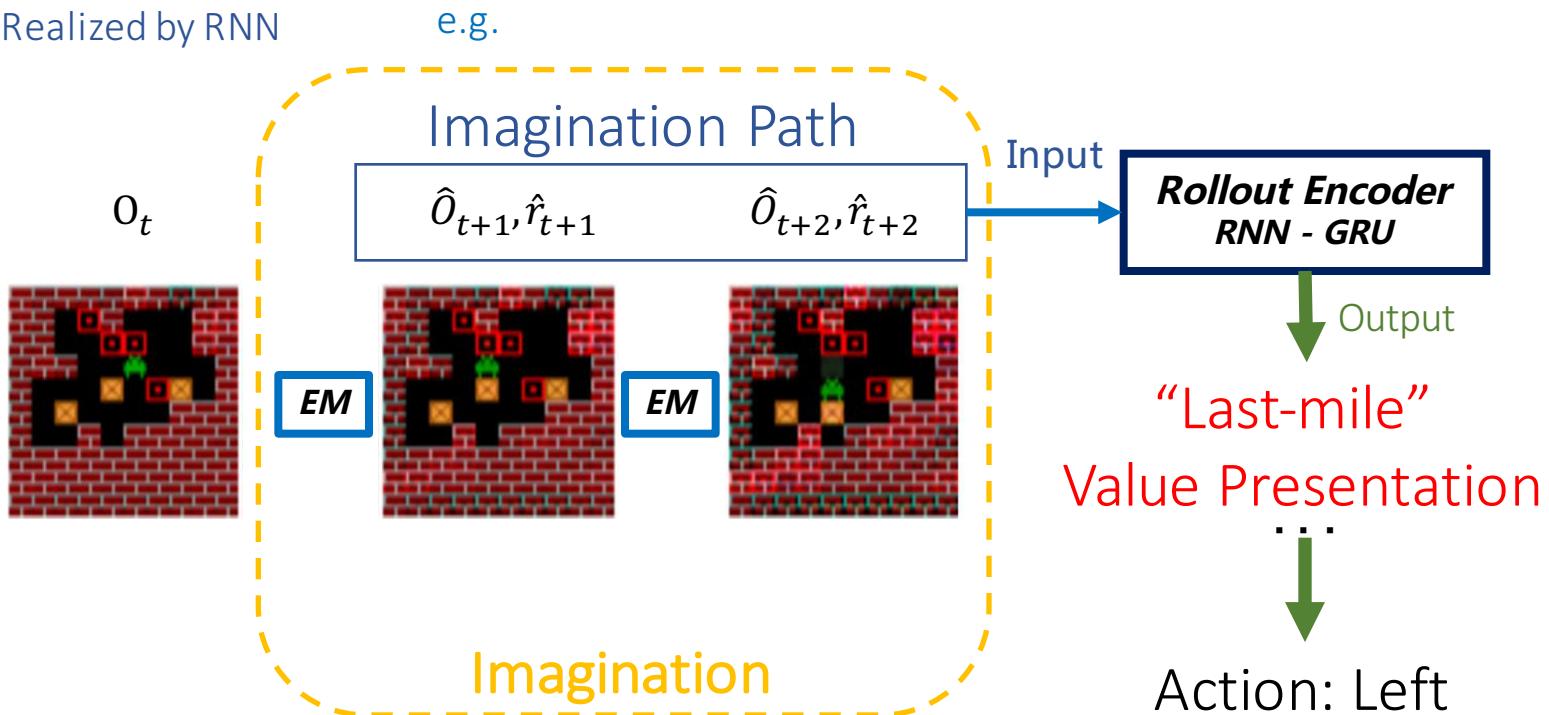


Role :

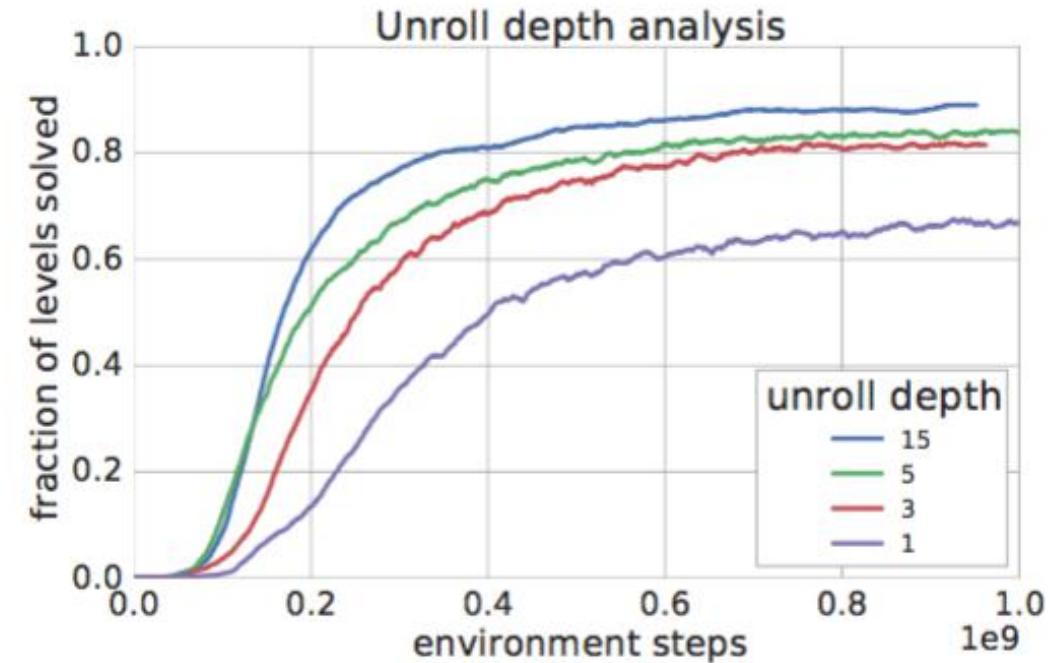
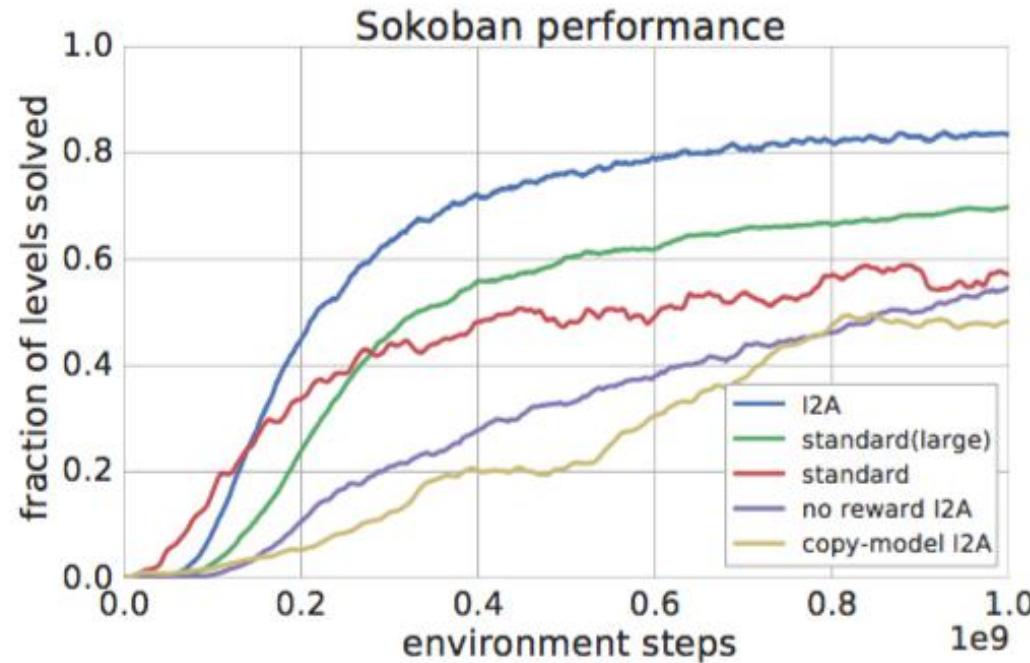
EM generate path after taking a specific action via imagination.

Encoder transform the path into the evaluation of selected action.

Realized by RNN

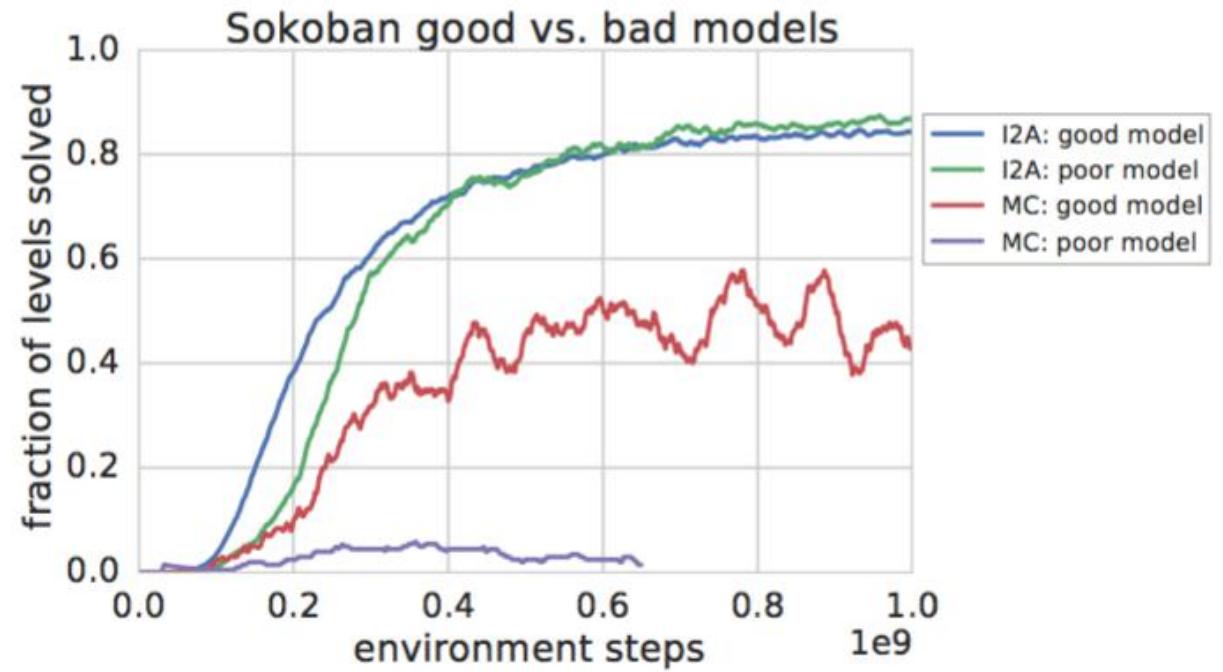
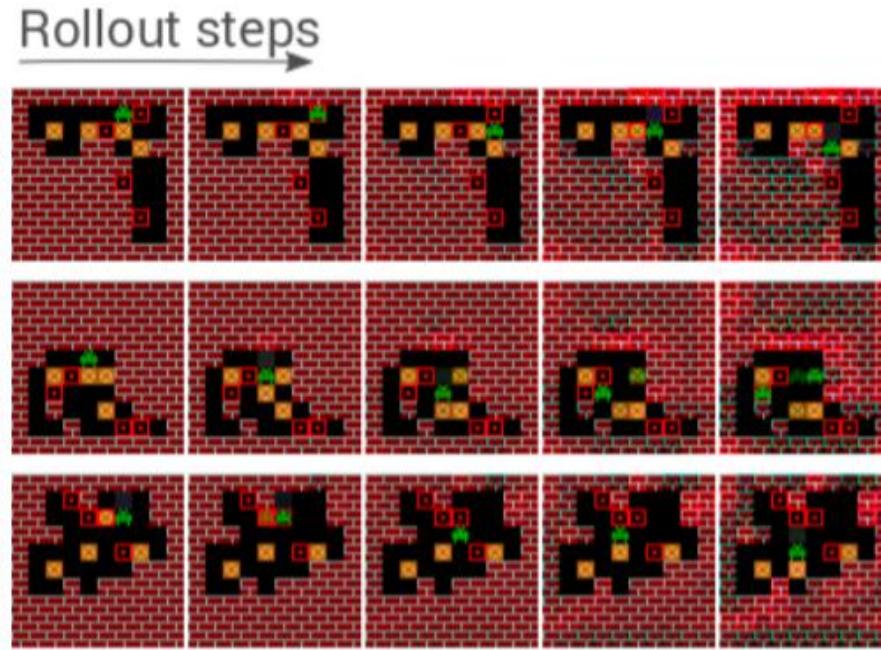


Route Planning with Reinforcement Learning



- I2A performs better than others
- Performance \propto Imagination Steps
- Diminishing returns with more rollout steps

Route Planning with Reinforcement Learning

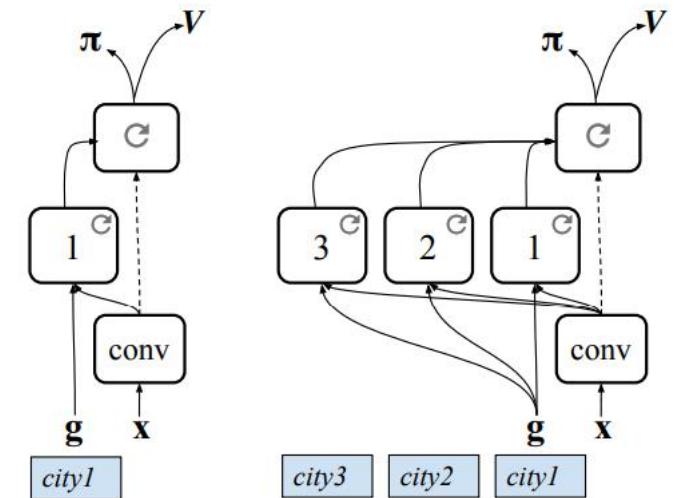
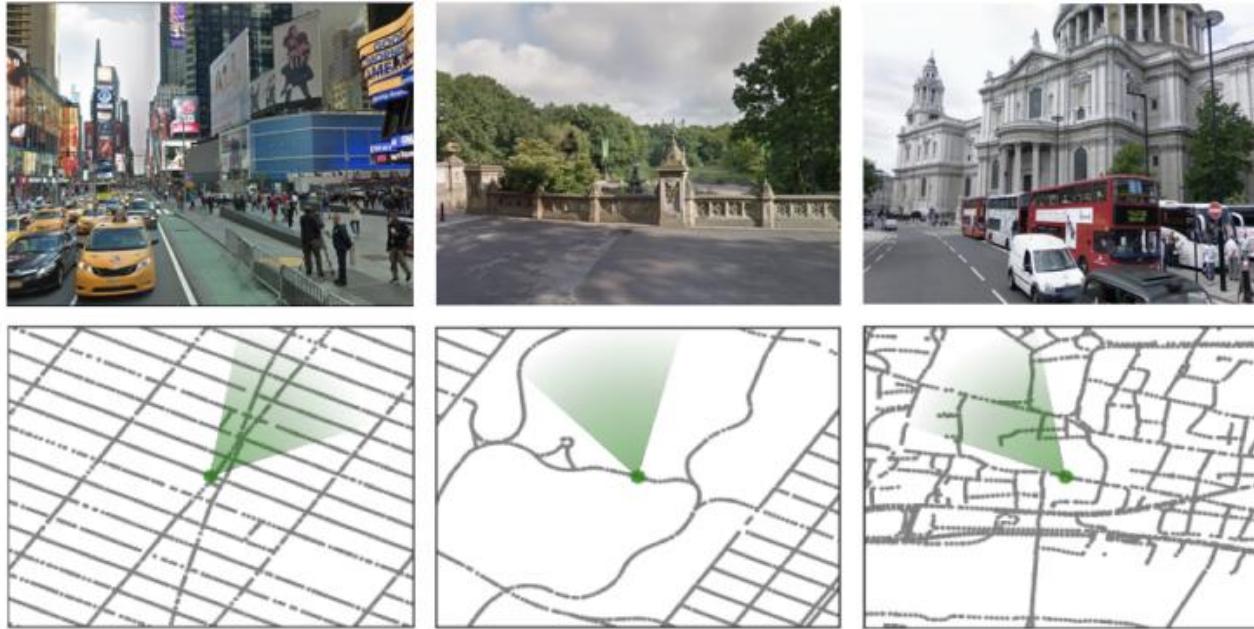


Address the challenge

- 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.

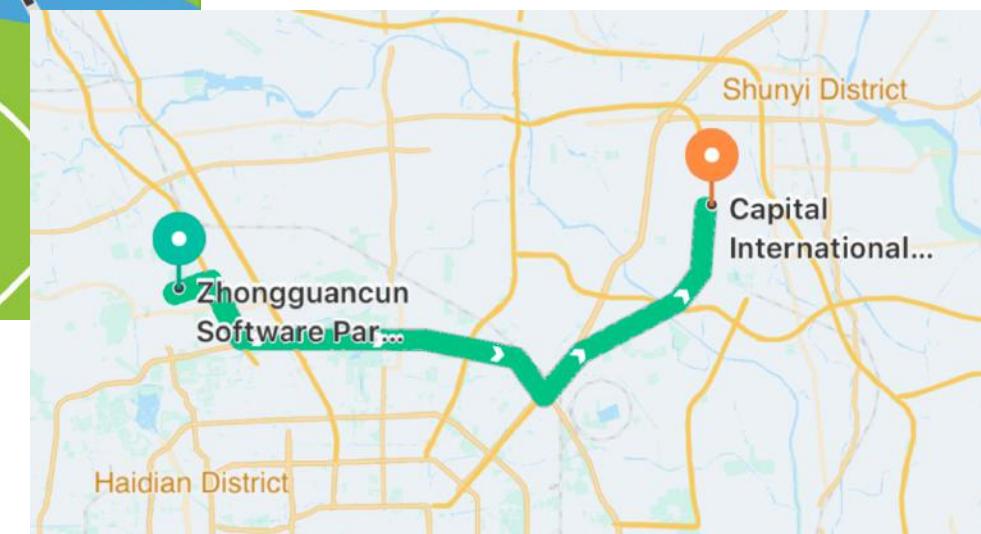
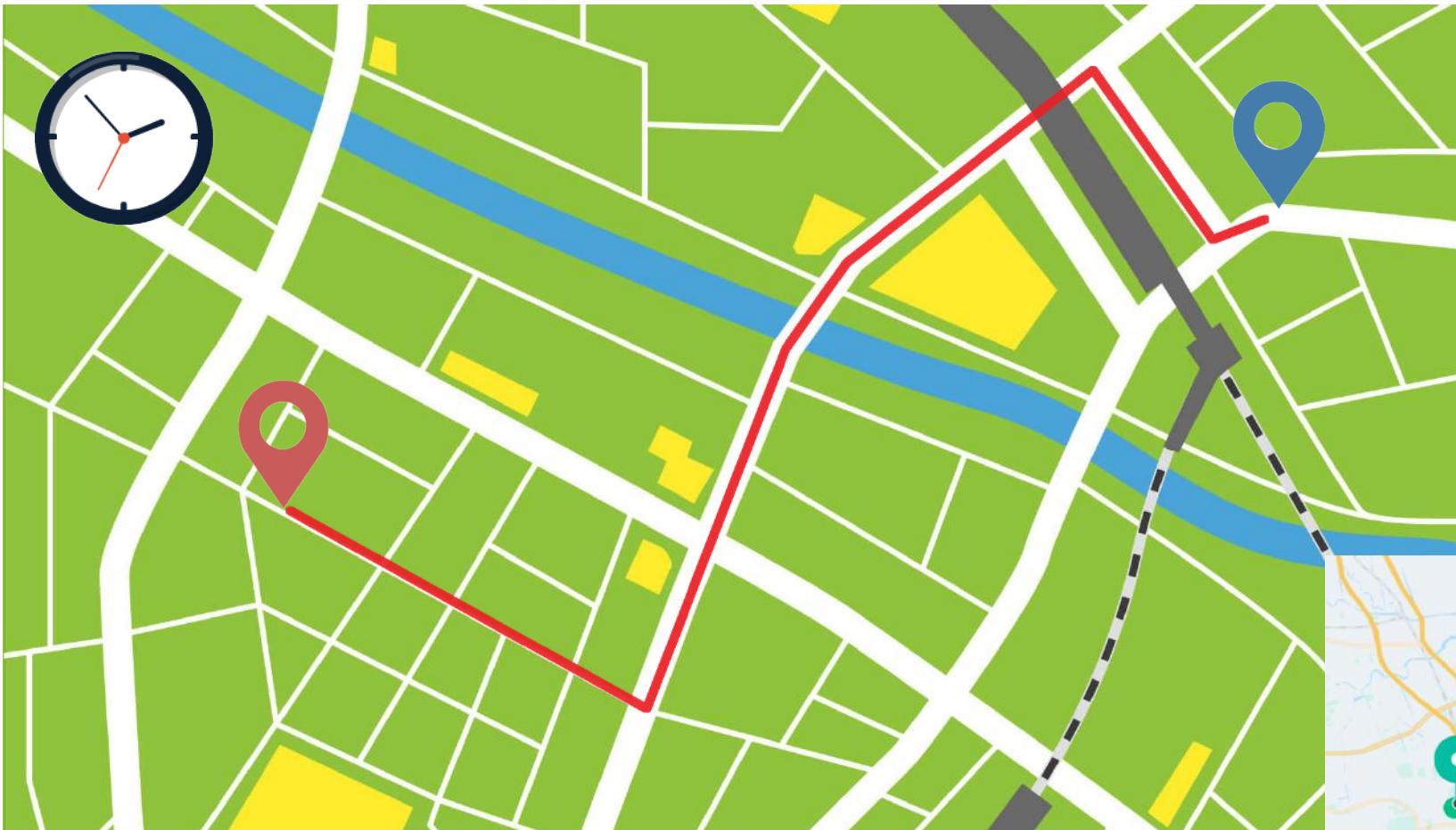
Reference

- 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.



ETA (Estimated Time of Arrival)

ETA (Estimated Time of Arrival)

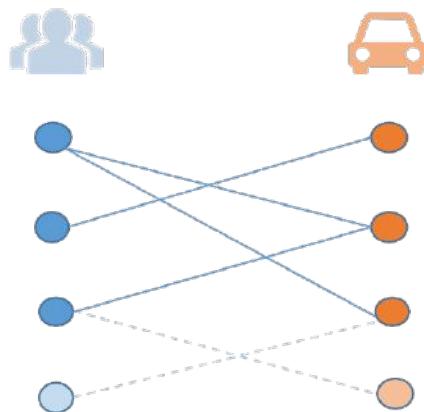


ETA Applications

Route Planning



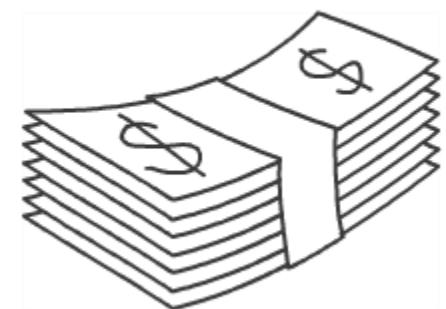
Order Dispatching



Ride Sharing



Pricing

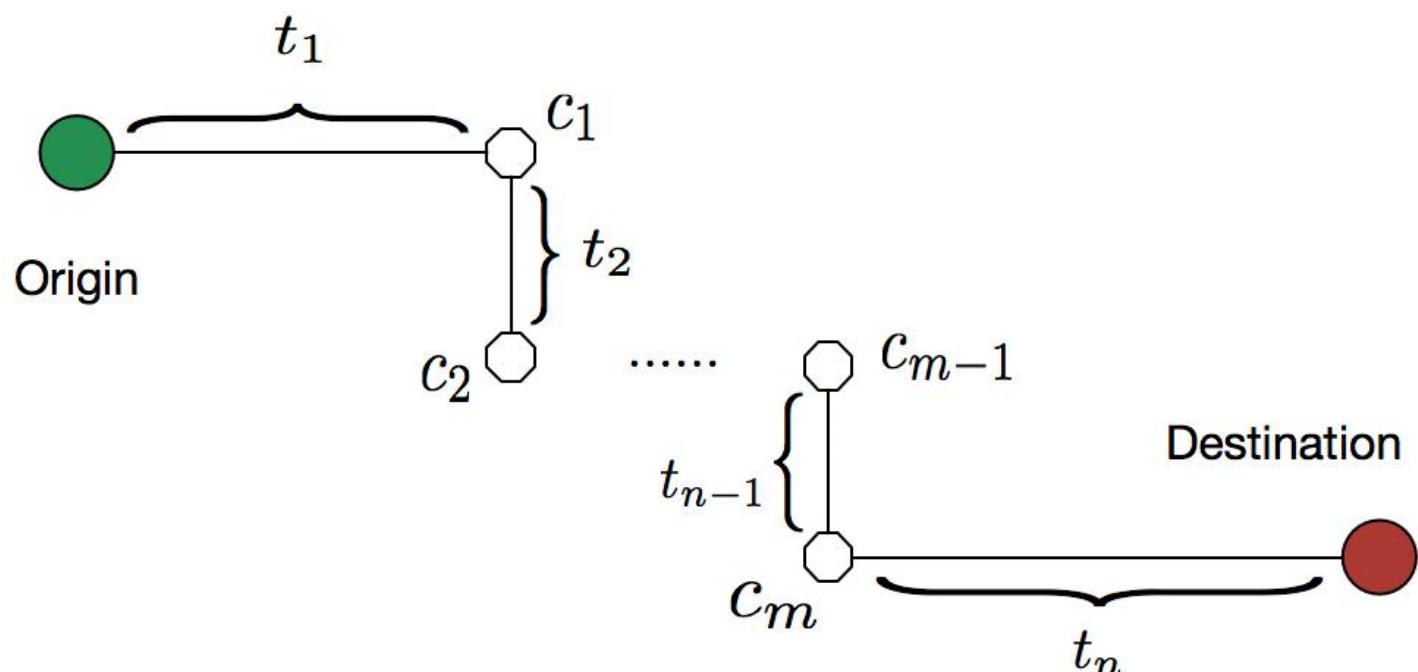


Main Stream ETA Models

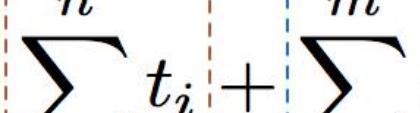
- Additive models:
 - Rule-based additive models: explicitly modeling the segments in a path
 - Aggregating the time of sub-paths
 - Machine learning models for the sub-path problem.
- Global models:
 - Formulating ETA as a regression problem: Learning to Estimate the Travel Times (L2ETT)
 - Simple regression 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



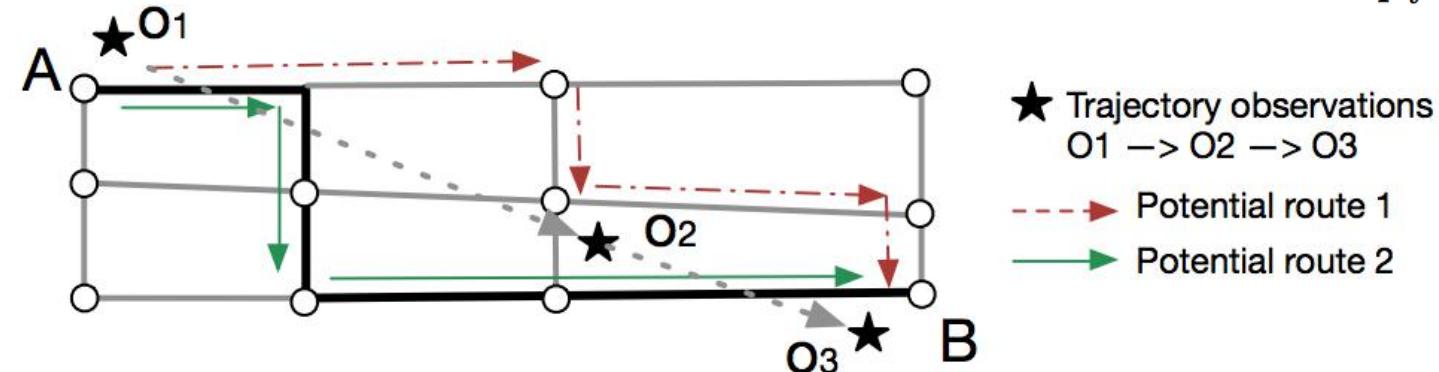
$$ETA = \sum_{i=1}^n t_i + \sum_{j=1}^m c_j$$



Aggregating Sub-Paths

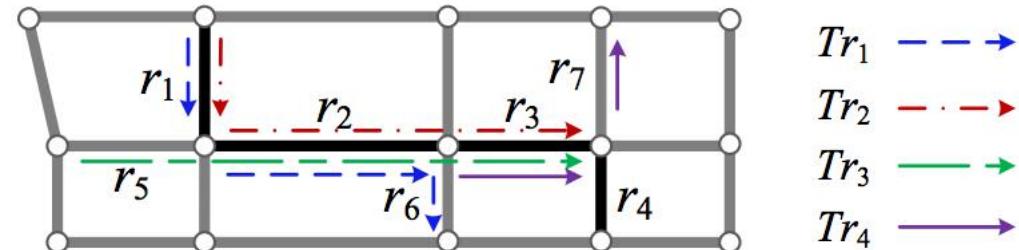
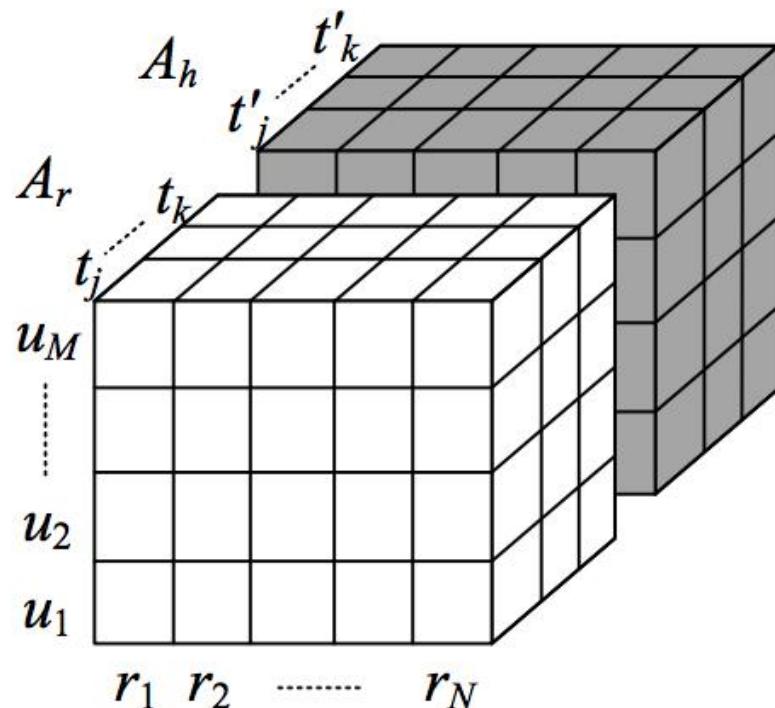
- The path can be decomposed into sub-paths.
- Time of each sub-path is inferreded from the historical trajectories.
- ETA is the summation of the time of sub-paths.

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



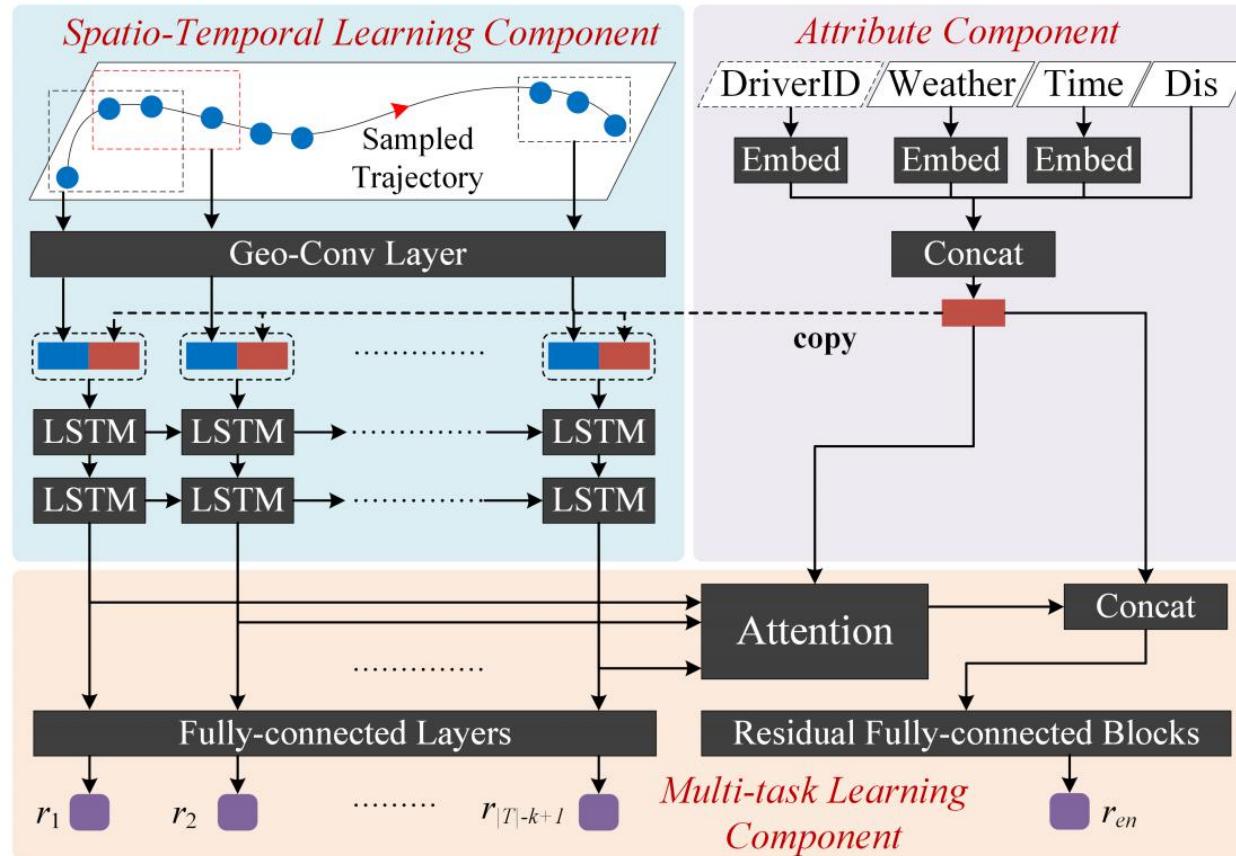
Tensor Decomposition

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



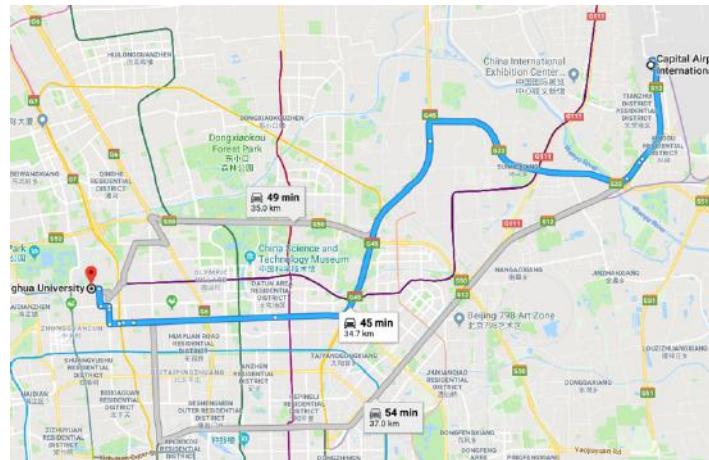
Regression Model on GPS Points

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

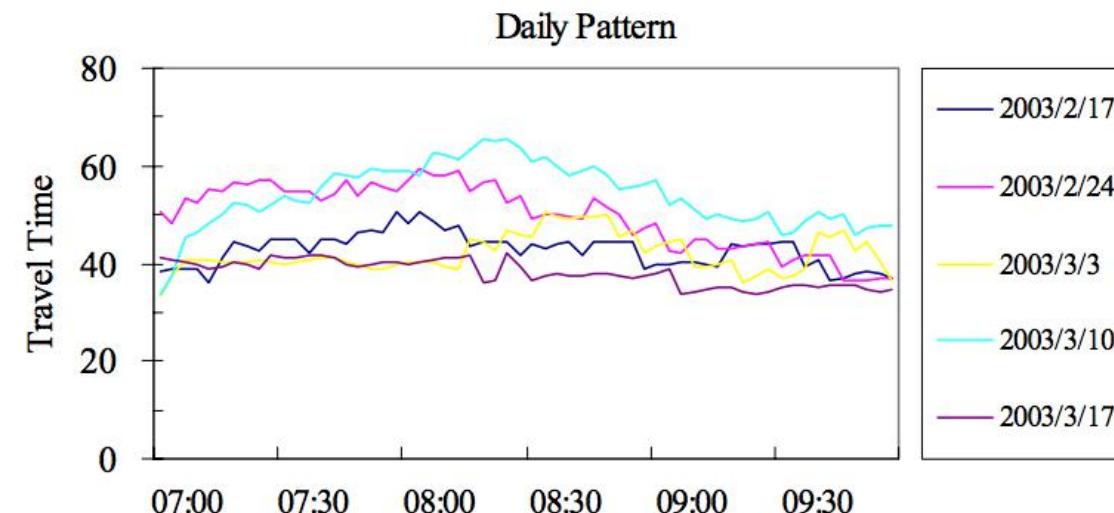


Time Series Prediction

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



$f(t-n), \dots, f(t-1) \rightarrow f(t)$
 $f(t-n), \dots, f(t-m) \rightarrow f(t)$

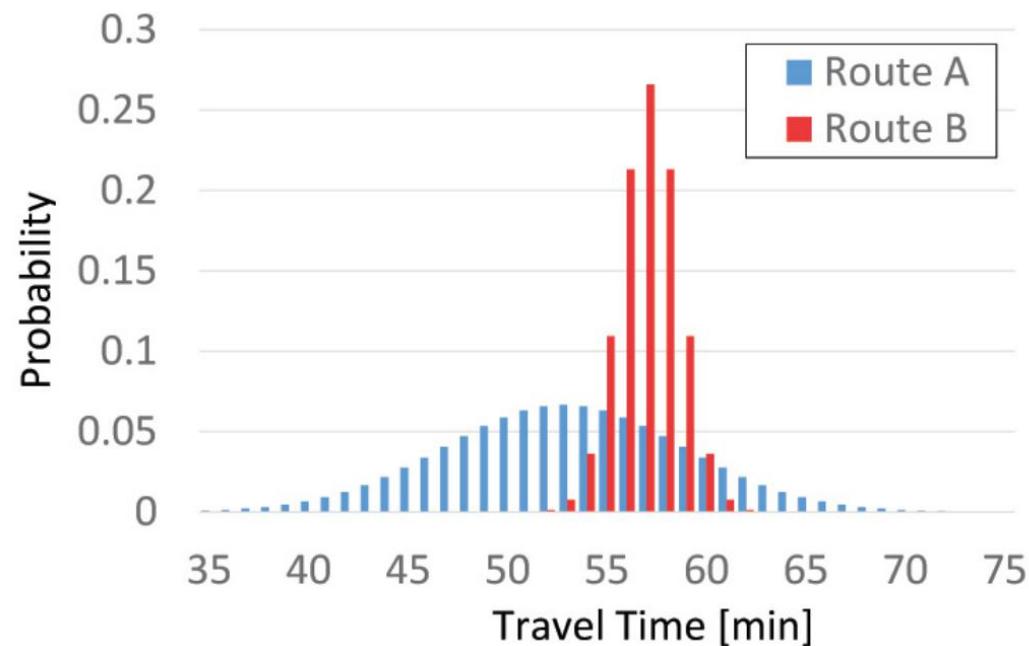


* Chun-Hsin et al. *Travel-time prediction with support vector regression*. IEEE Trans. ITS 2004.

* Yanjie Duan et al. *Travel time prediction with LSTM neural network*. ITSC 2016.

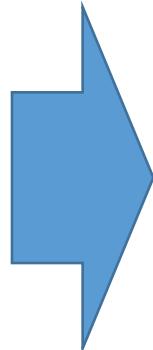
Probabilistic ETA

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



Additive Model vs Global Model

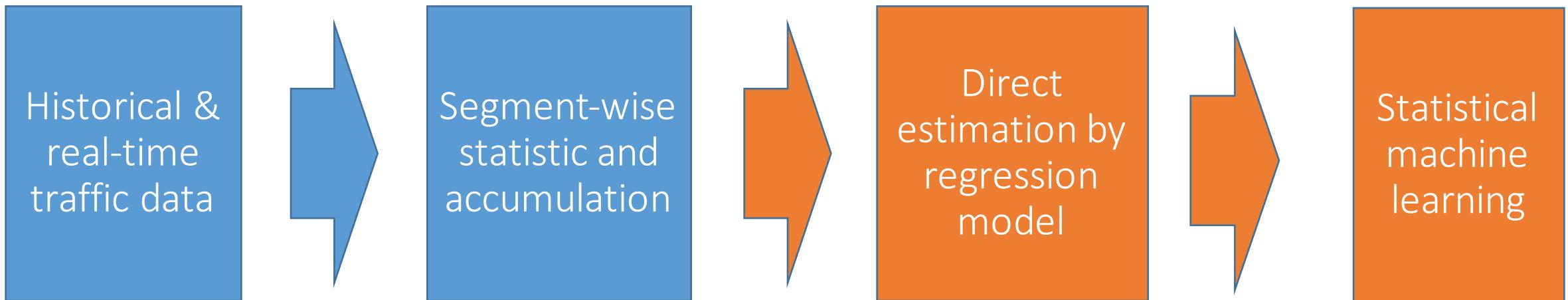
- 1 Heuristic rule
- 2 Indirective objective
- 3 Less robust
- 4 Insufficient use of data



Additive model?

Pure learning?

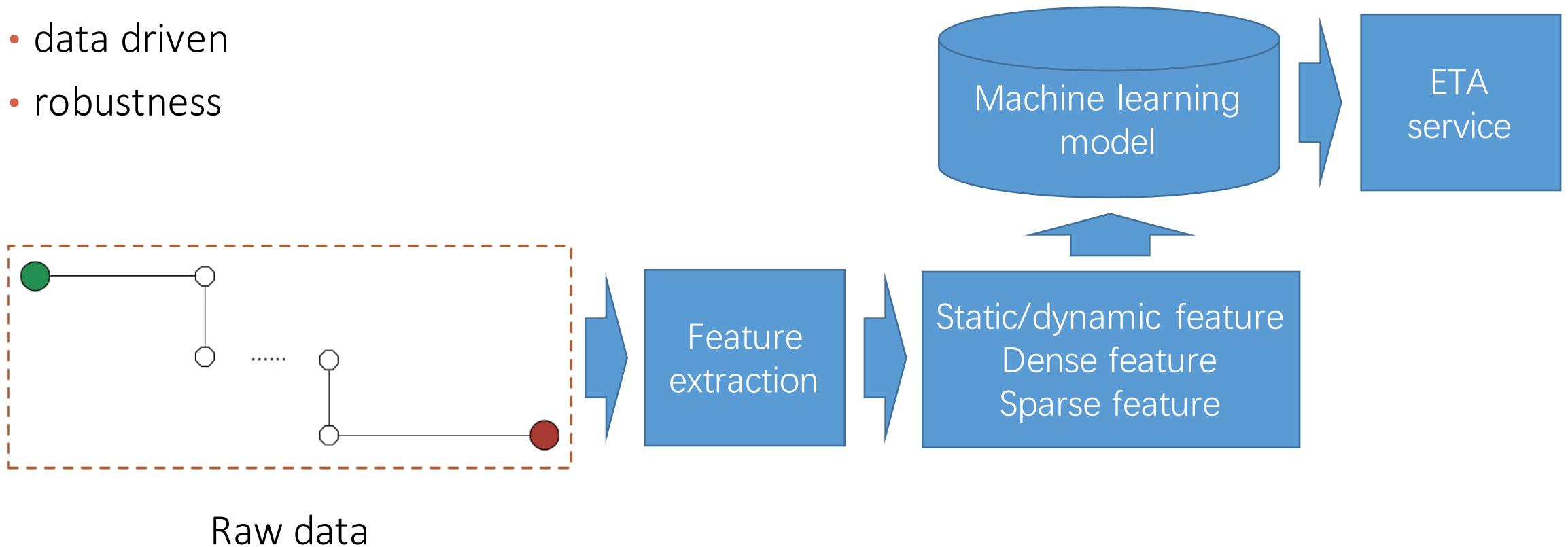
Rethinking ETA



Learning to Estimate the Travel Time

- Big data + machine learning

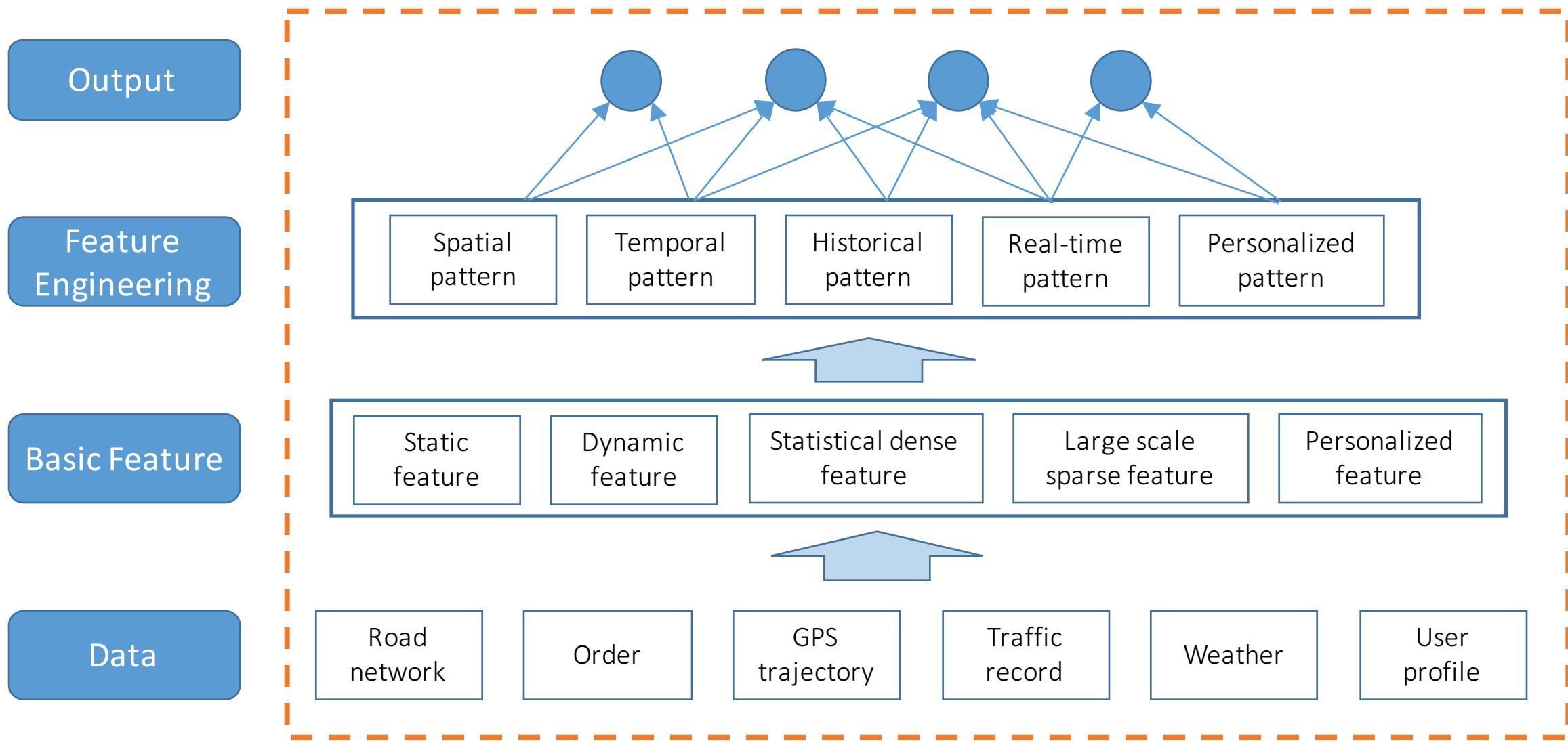
- high accuracy
- data driven
- robustness



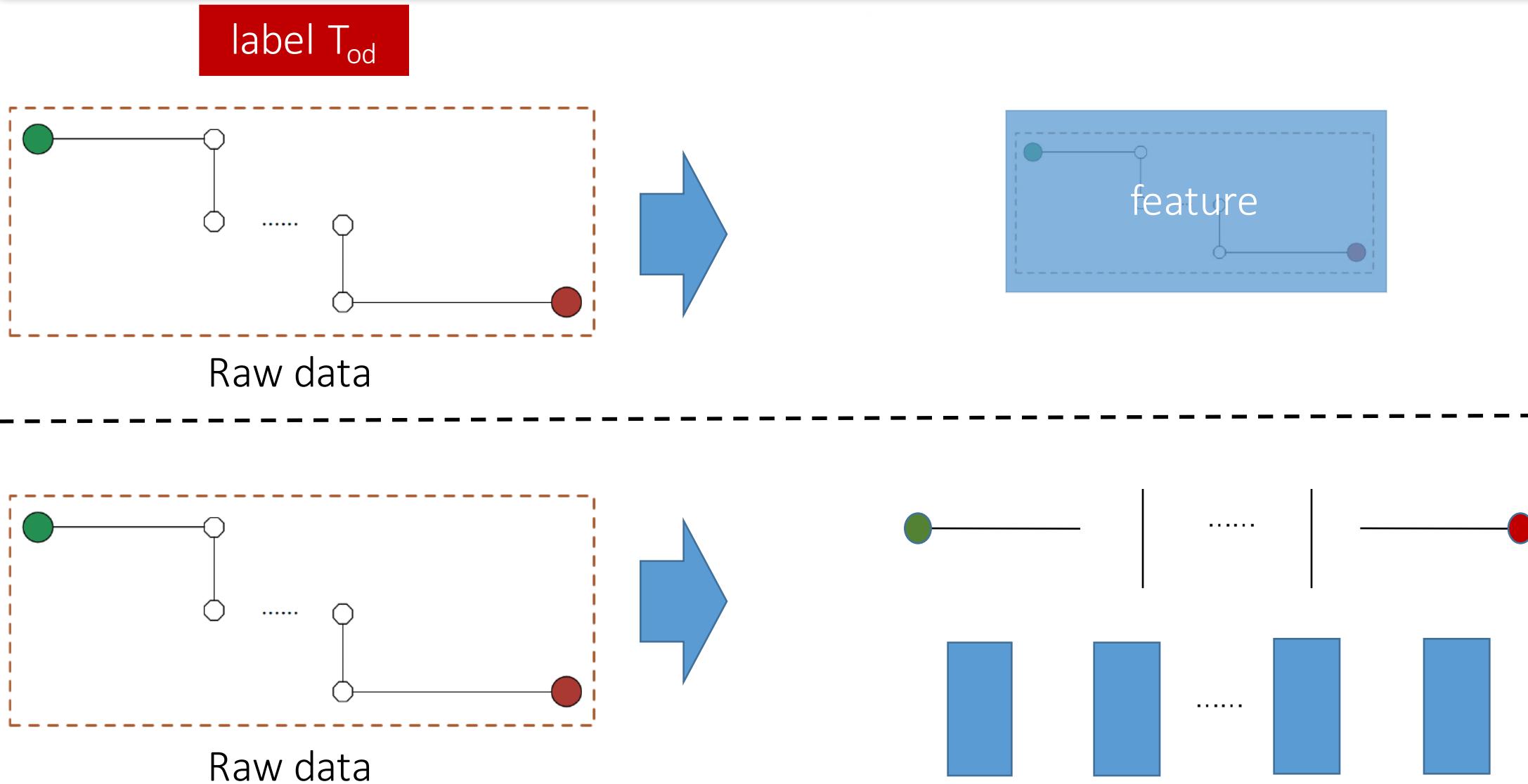
Problem Formulation

- Formulation: ETA as a regression problem on sequential input
- Data: a collection of trips $\mathbf{X} = \{x_i\}_{i=1}^N$
 - 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

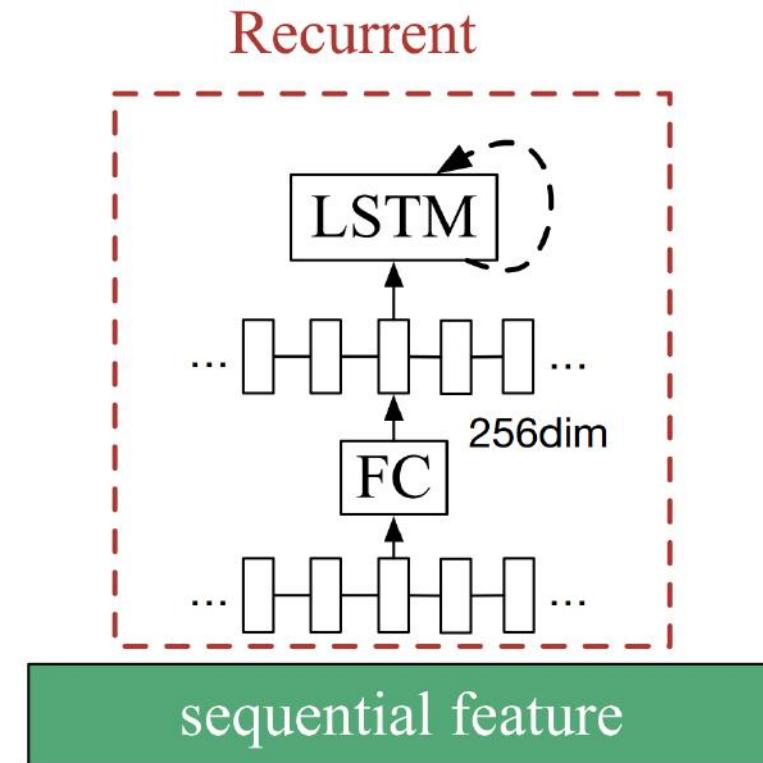
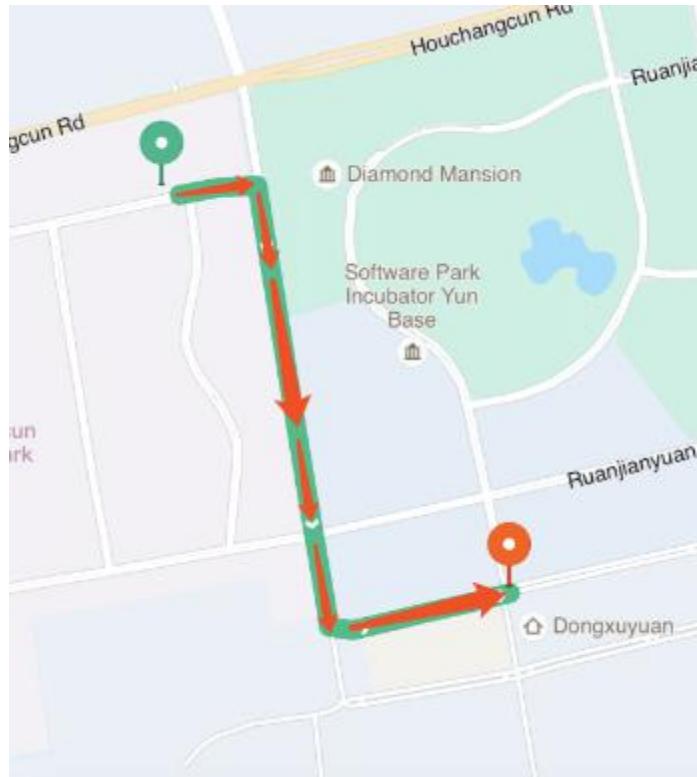


Two faces: global feature vs sequential feature



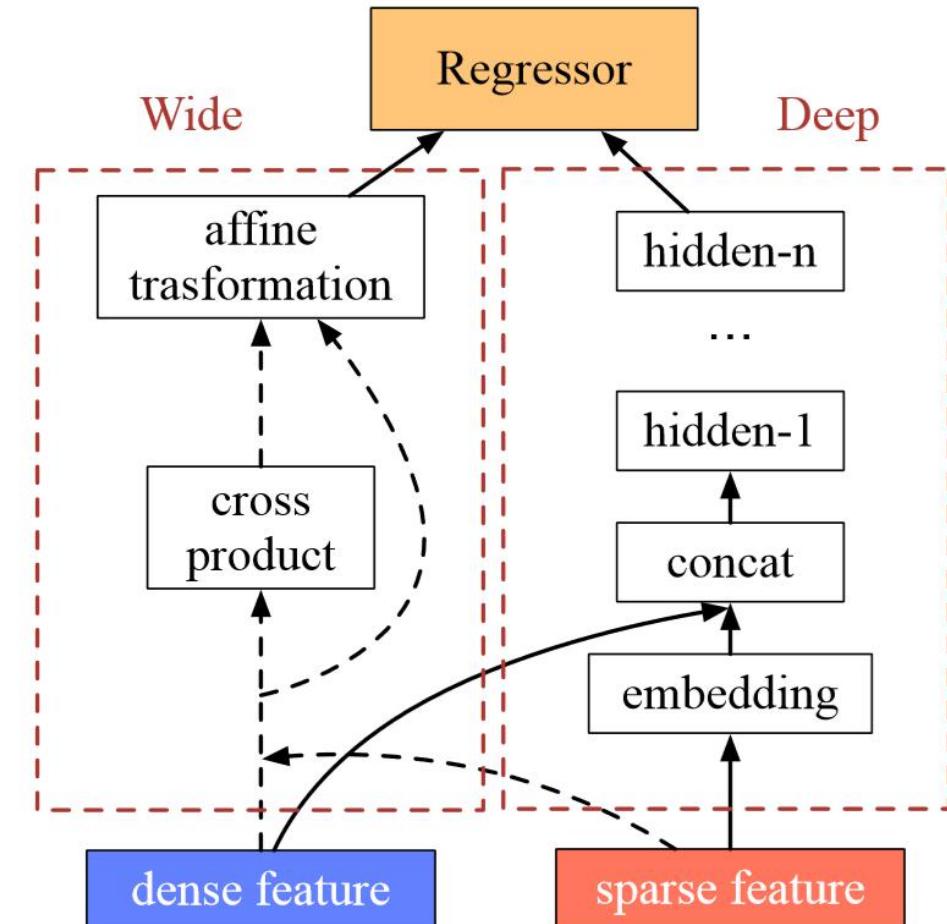
Recurrent Model

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

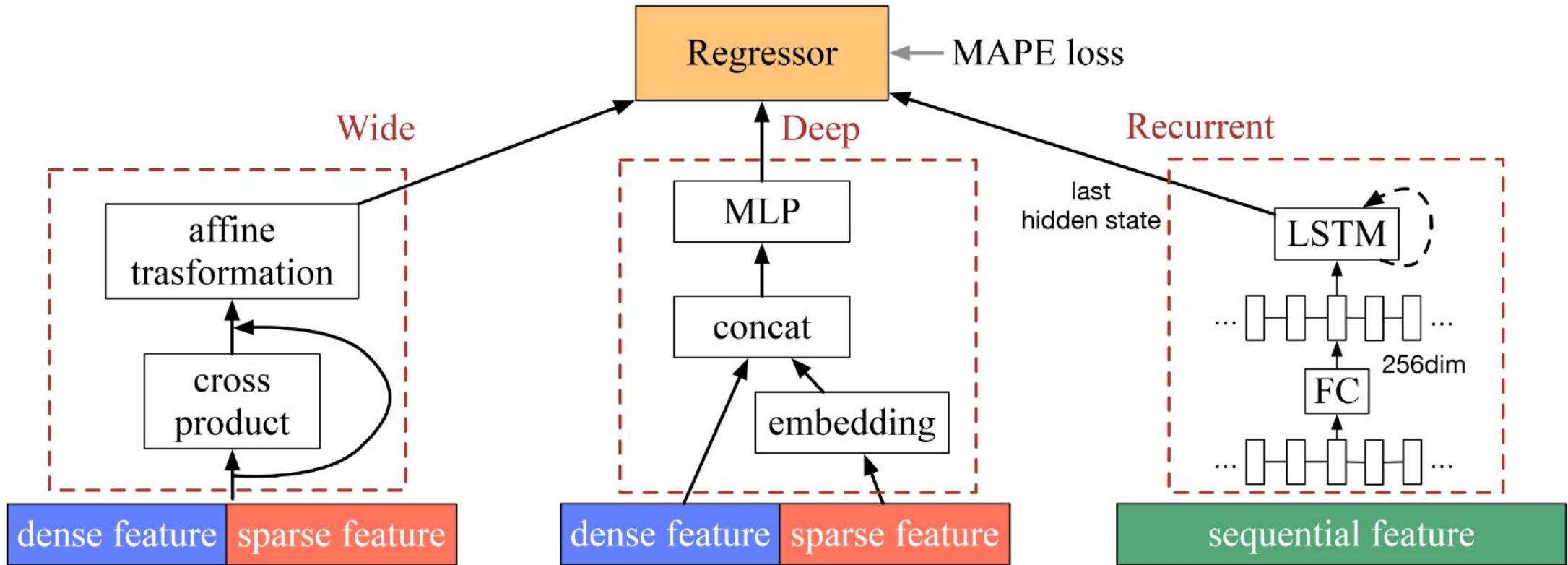


Tree-Based Model -> Wide & Deep Model

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



Wide-Deep-Recurrent Network (WDR)



Offline Evaluation Dataset

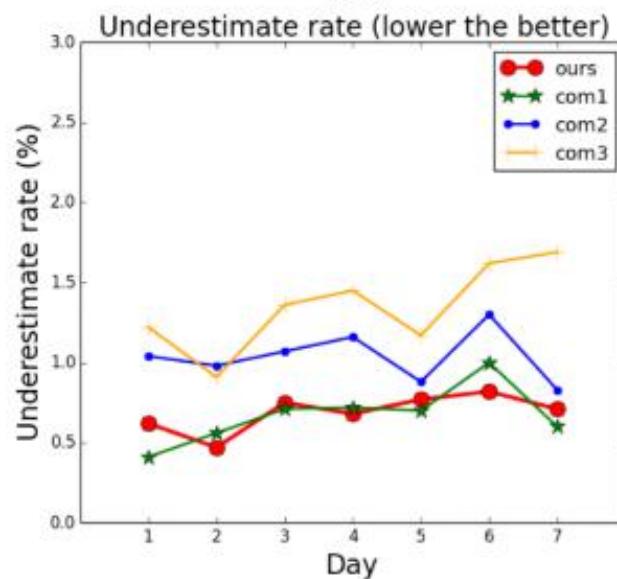
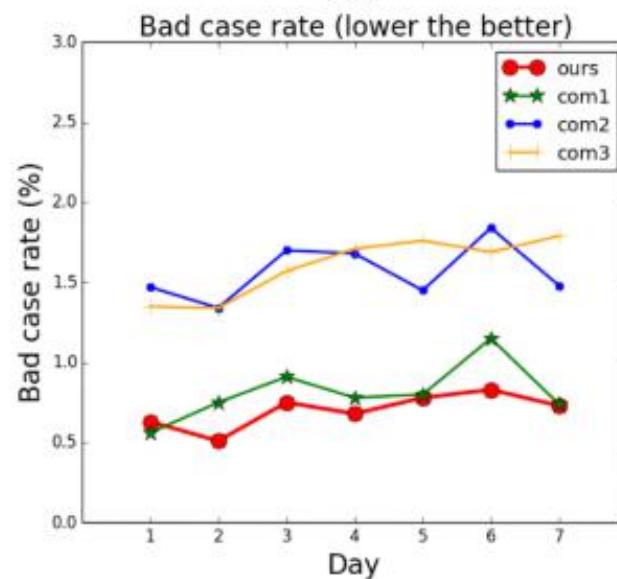
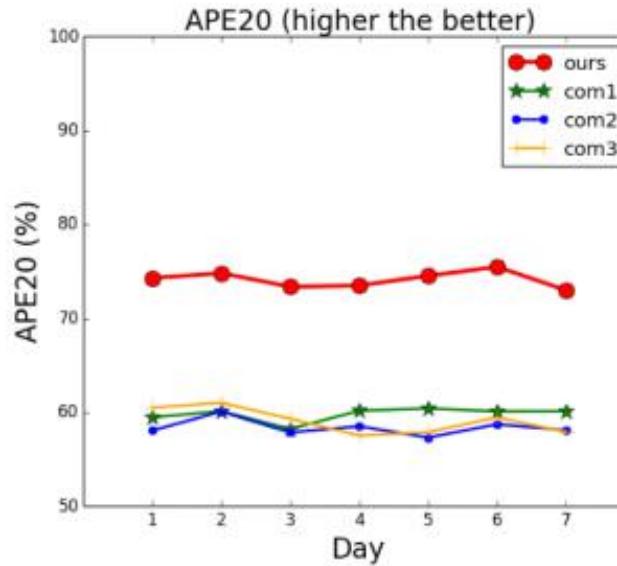
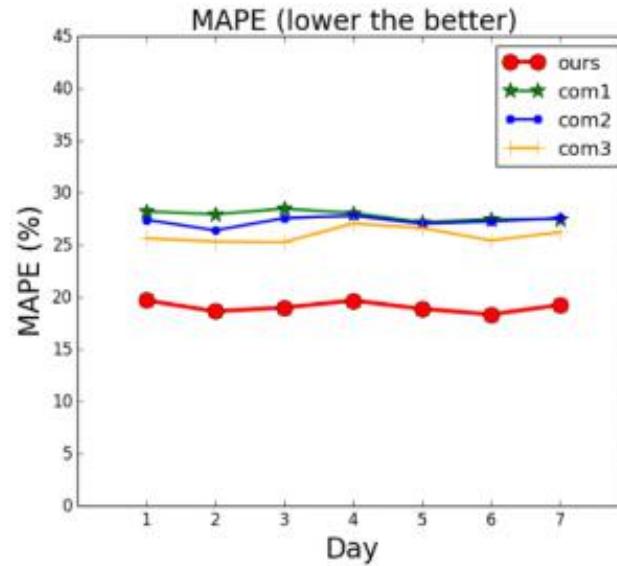
	date	pickup	trip
training set	1.1 - 5.10 (2017)	48M	51M
validation set	5.11 - 5.17 (2017)	3M	4M
test set	5.18 - 5.31 (2017)	6M	7M
unique links	–	0.5M	0.5M
unique drivers	–	0.4M	0.4M

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

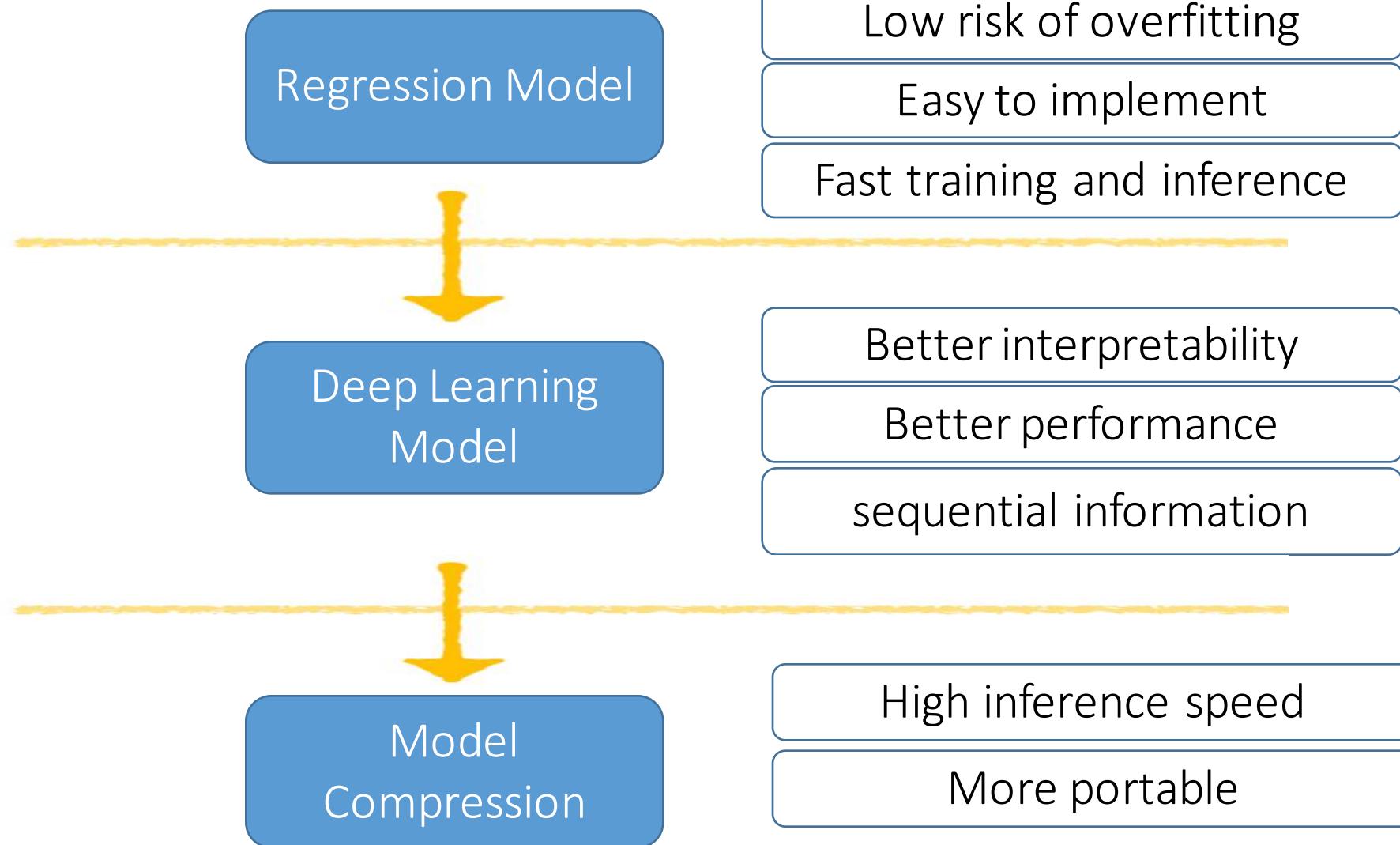
	MAPE	MAE	MSE
route-ETA	15.01%	153.2	66789.8
PTTE [20]	14.78%	150.1	66141.2
GBDT	14.01%	133.2	52006.6
FM	12.87%	122.5	47457,3
WD-MLP	13.43%	127.5	50012.8
WDR	11.66%	112.4	37482.7
*TEMP _{rel} [19]	23.53%	166.8	88767.1
*WDR	12.27%	87.2	20929.3

Online Evaluation Results



WDR achieves the best performance for all metrics.

Progress



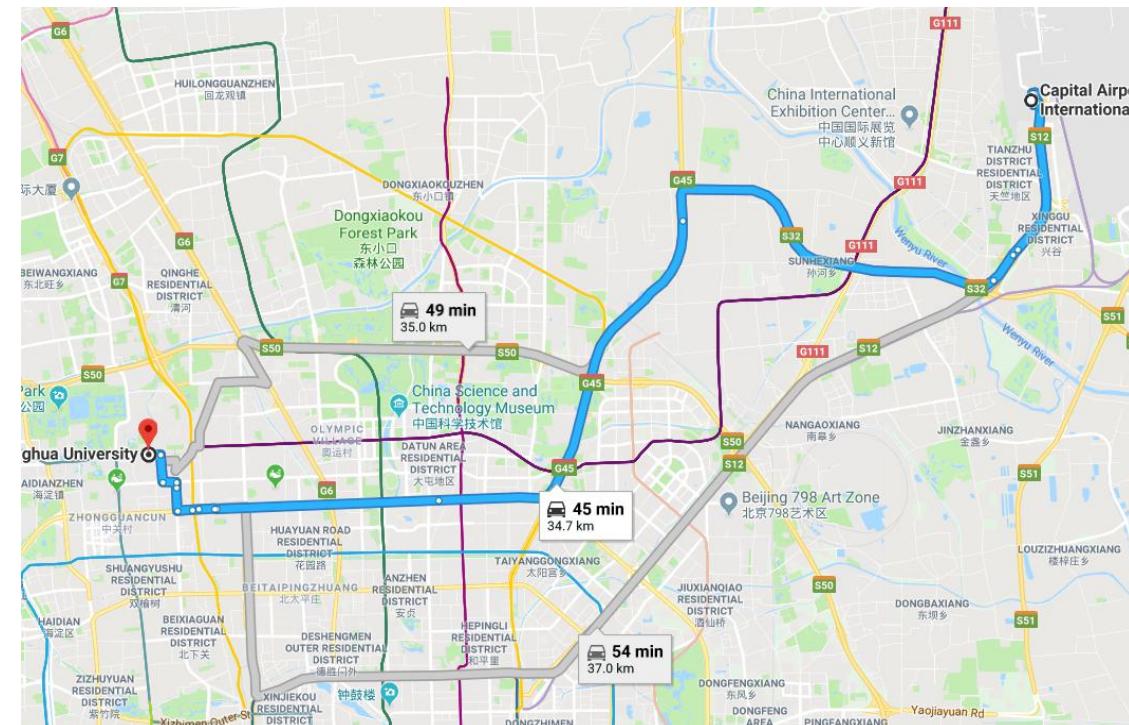
More Practical: Origin Destination ETA

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

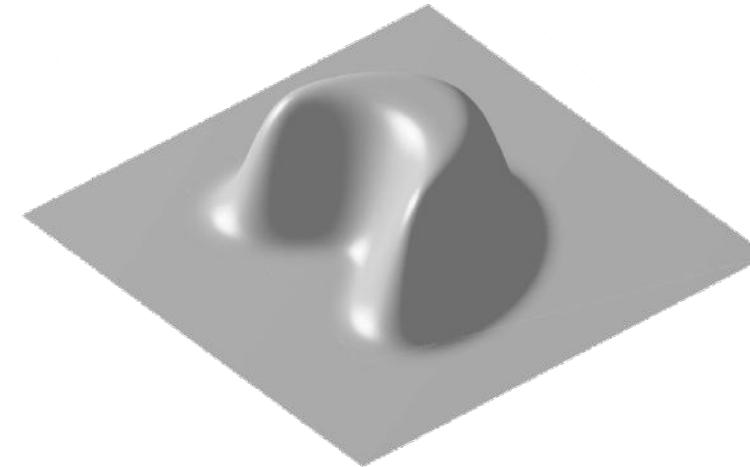


MURAT: Multi-task Representation Learning for ETA

Road Network
Embedding



Spatiotemporal
Smoothness



MURAT: Multi-task Representation Learning for ETA

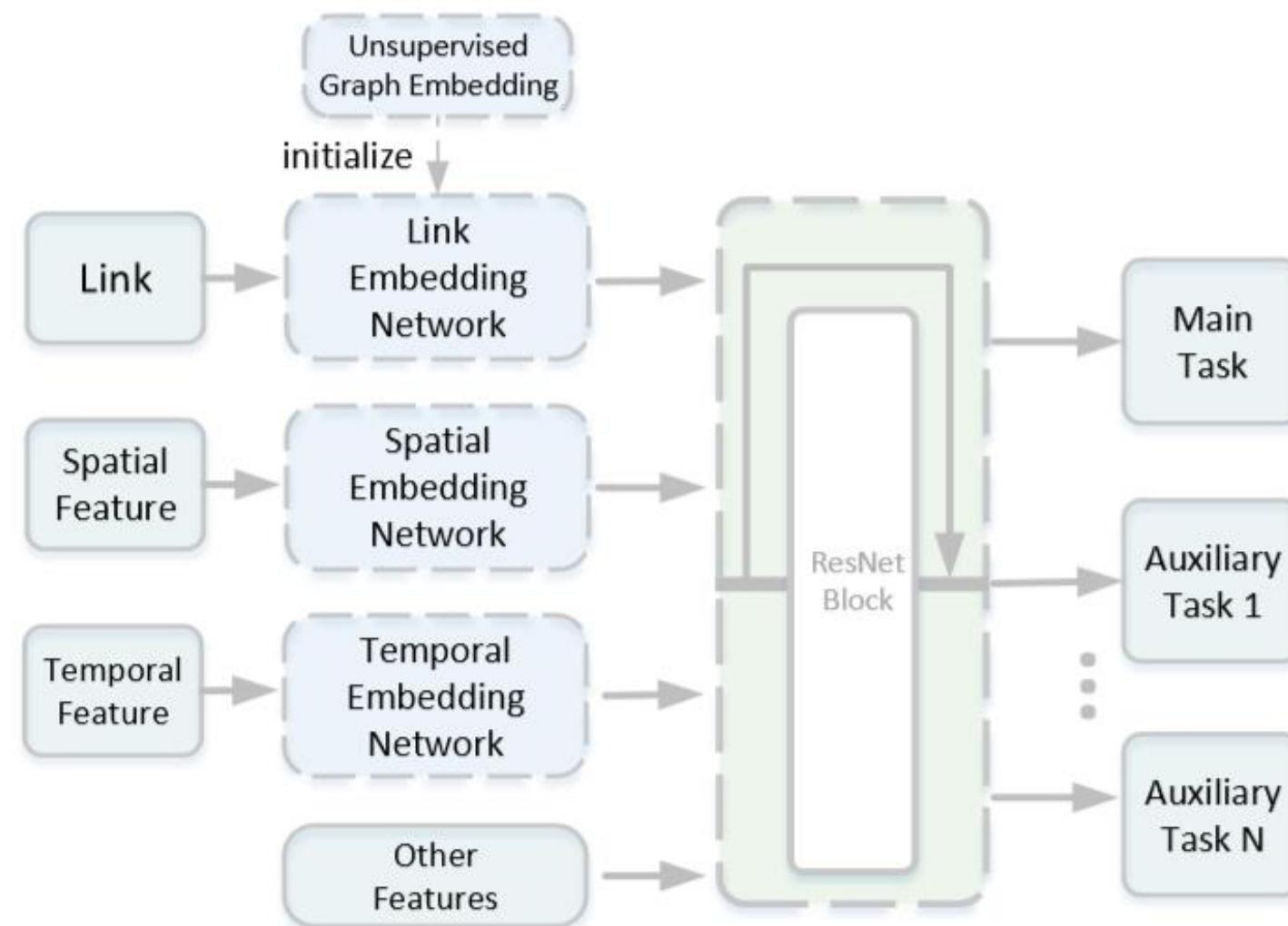


Historical Paths

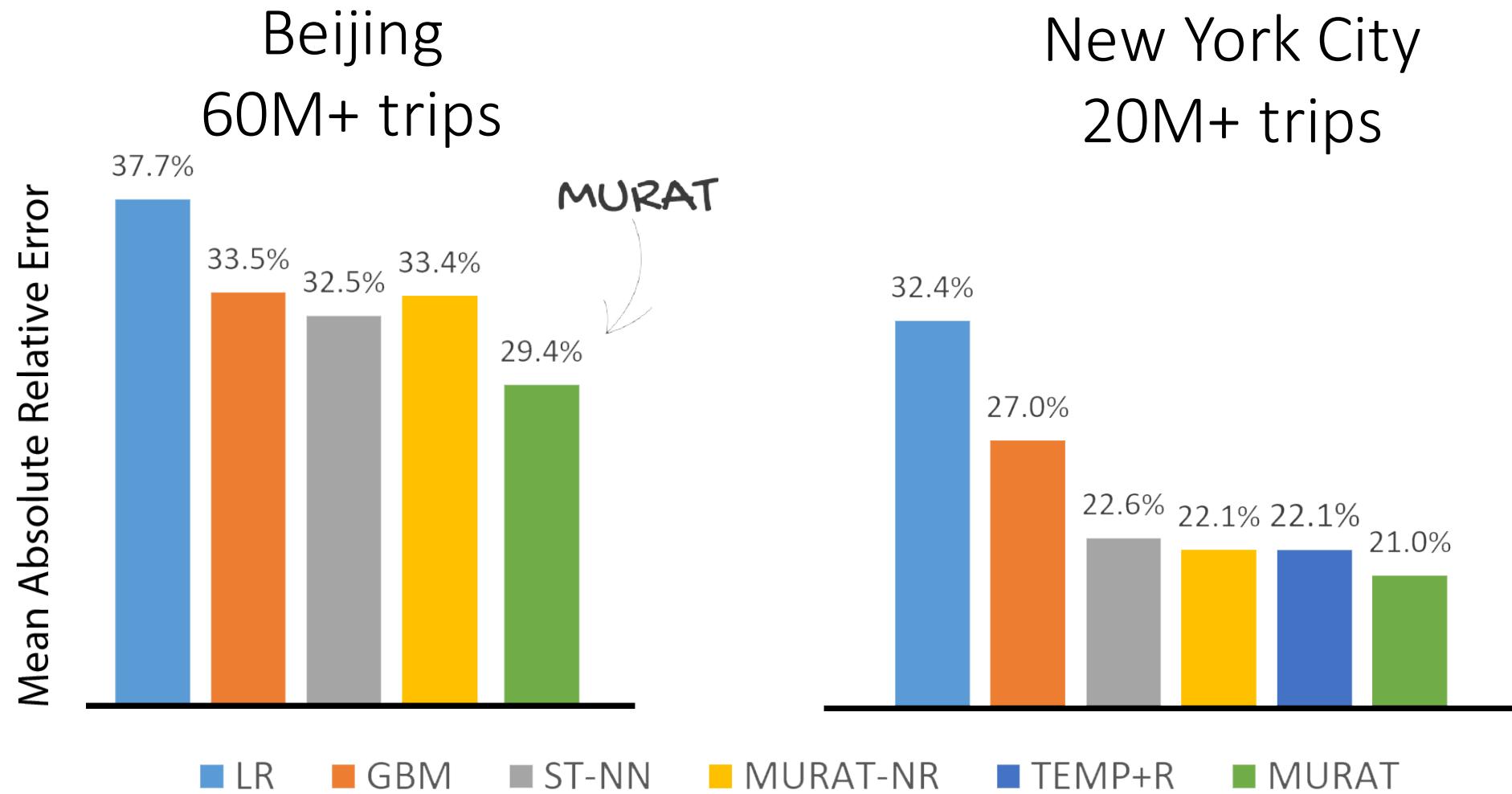
Travel Distance
Road Segments
Traffic lights
Left/right Turns

Auxiliary Tasks

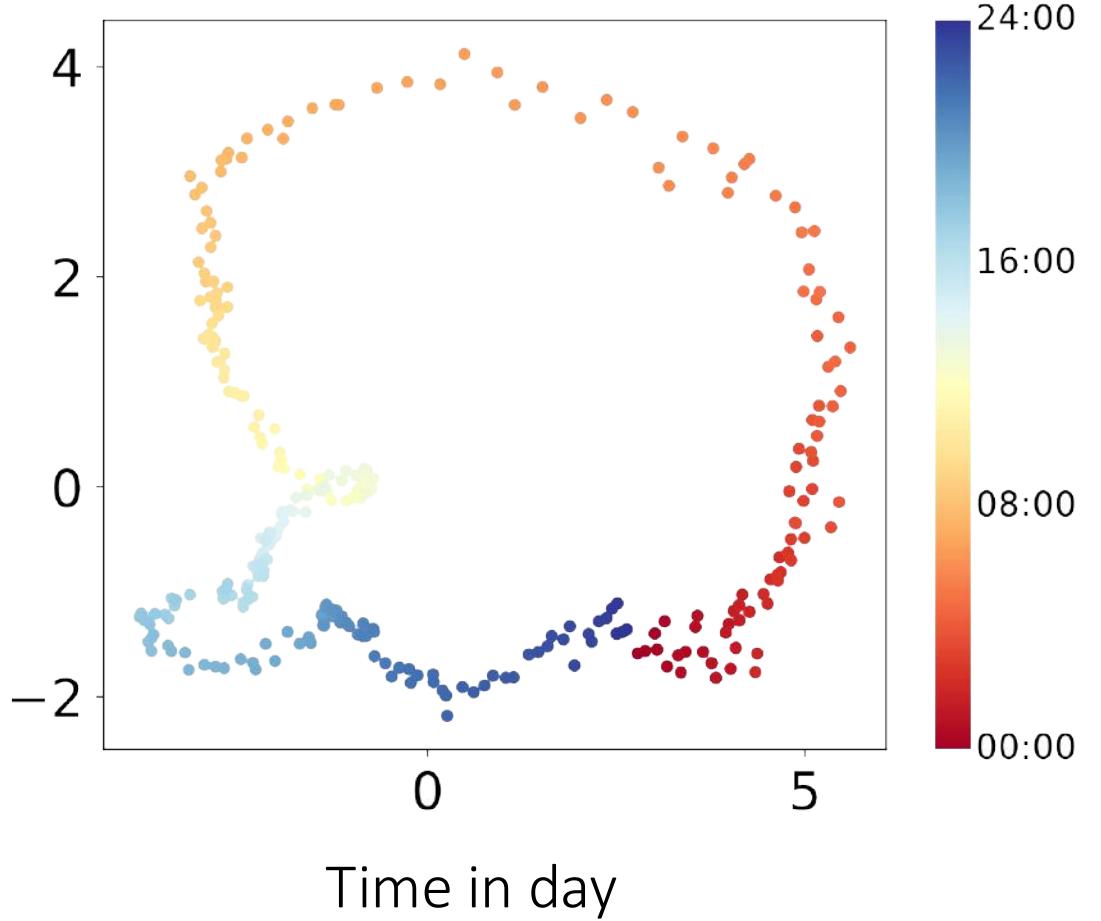
MURAT: Multi-task Representation Learning for ETA



Evaluation



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.



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



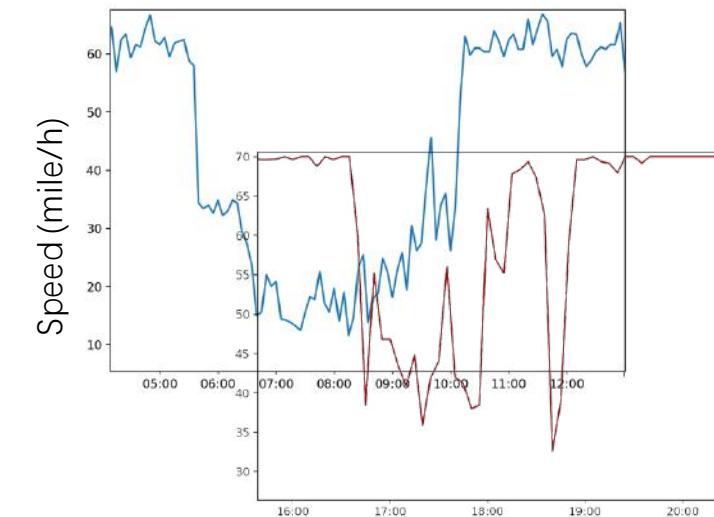
8:10AM, 8:20AM, ..., 9:00 AM

Challenges for Traffic Forecasting

Complex
Spatial Dependency

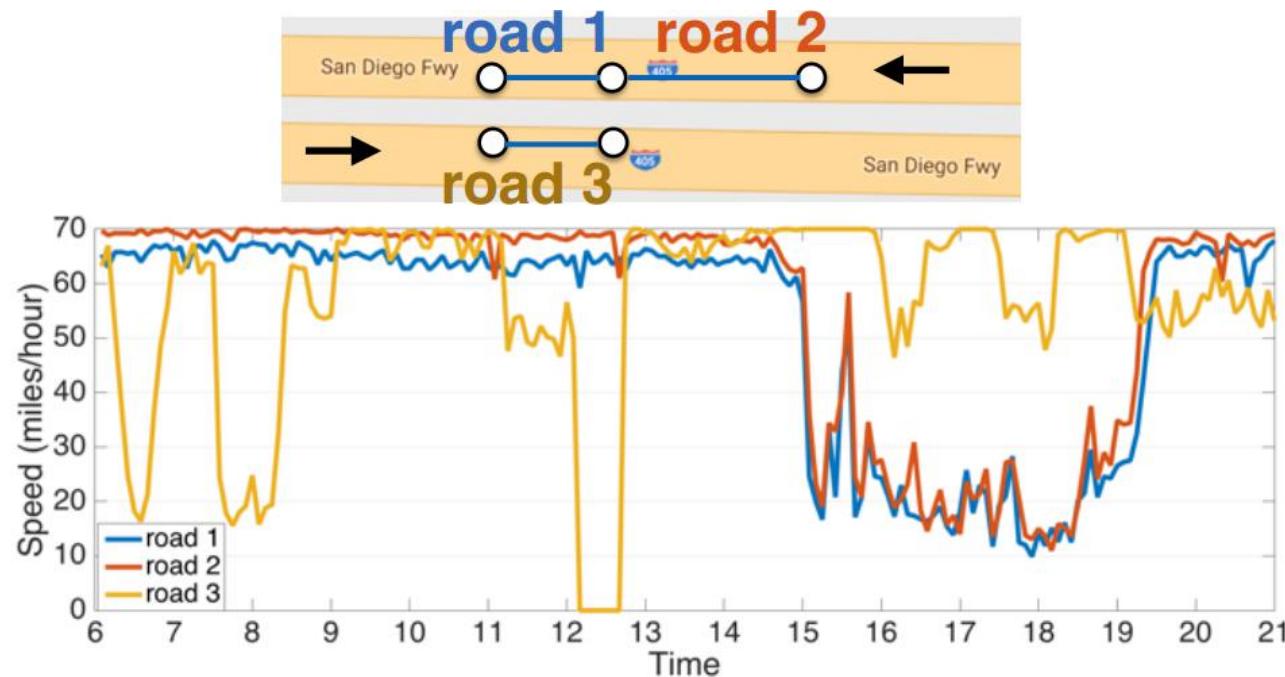


Non-linear, non-stationary
Temporal Dynamic



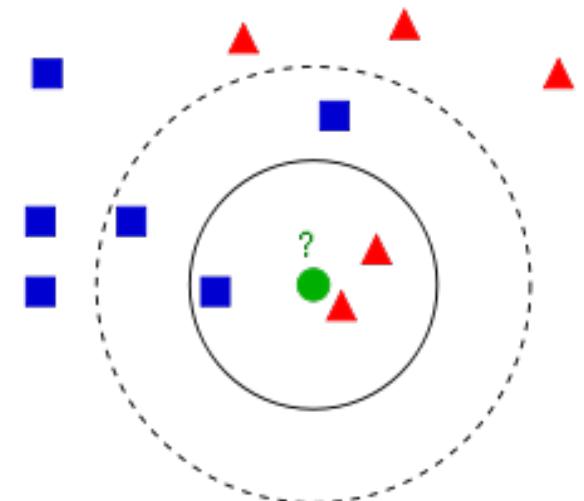
Challenges for Traffic Forecasting

- 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.



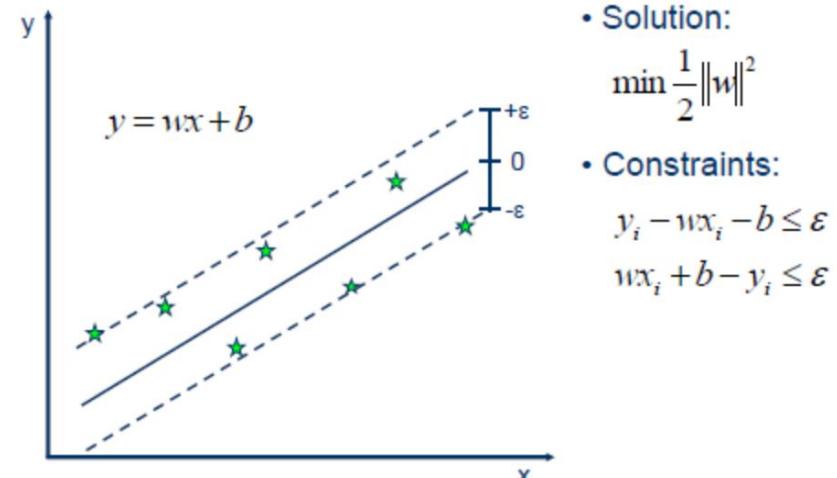
Time Series Methods

- Seasonal Autoregressive Integrated Moving Average (SARIMA)

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

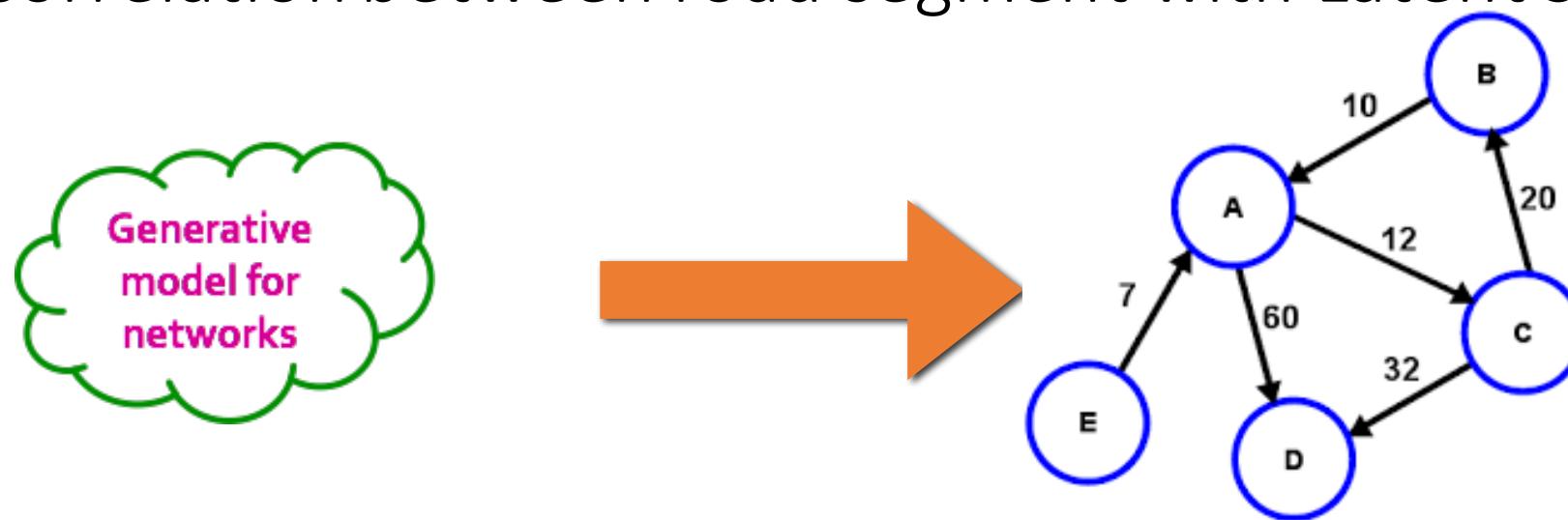
S-ARIMA
Auto-Regressive Integrated Moving Averages
(w/Seasonality)

- Support Vector Regression (SVR)



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

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 $U^{n \times k}$
(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 $B \in R^{k \times k}$

Speed from Latent Attribute

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

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

Diagram illustrating the calculation of traffic speed $d(i, j)$:

Vertices i and j are connected by an arrow labeled "?".

The vector u_i is represented as a row vector [1 1 1 1 1].

The matrix B is a 5x5 matrix representing traffic patterns:

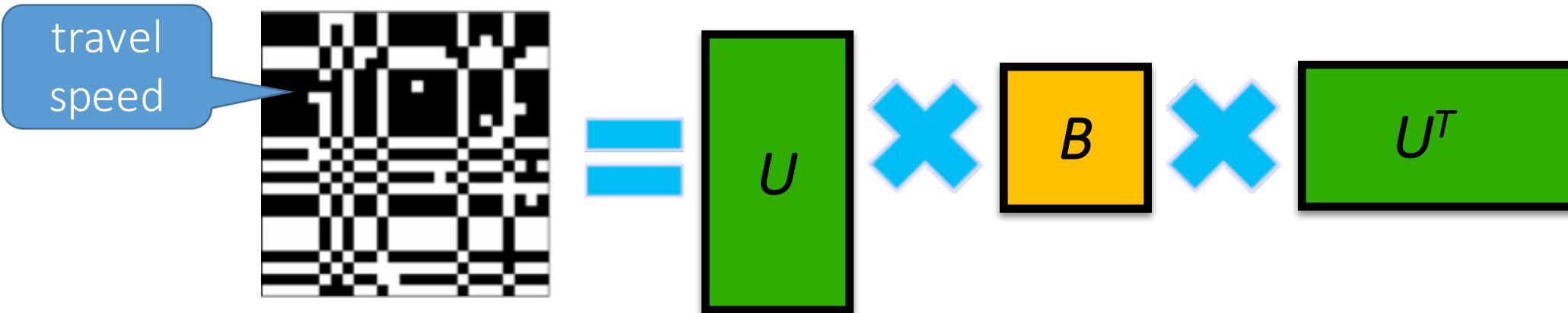
40		0			
0		10			

The vector u_j^T is represented as a column vector [1 1 1 1 1].

The calculation is shown as:

u_i x B x u_j^T = 50

Basic Graph Model

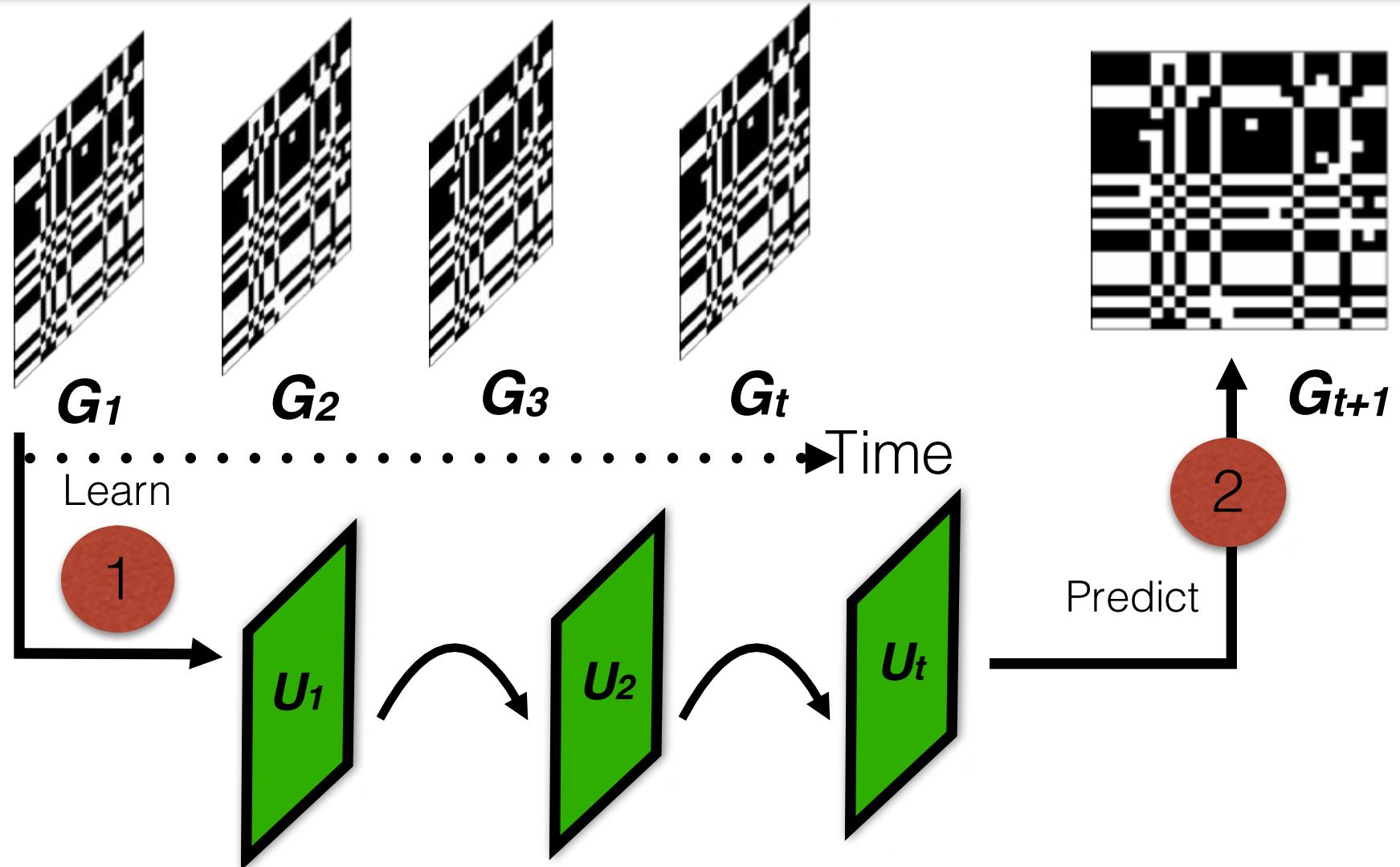


$$\arg \min_{U \geq 0, B \geq 0} J = ||G - UBU^T||_F^2$$

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

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

Temporal Effect

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

Transition Effect

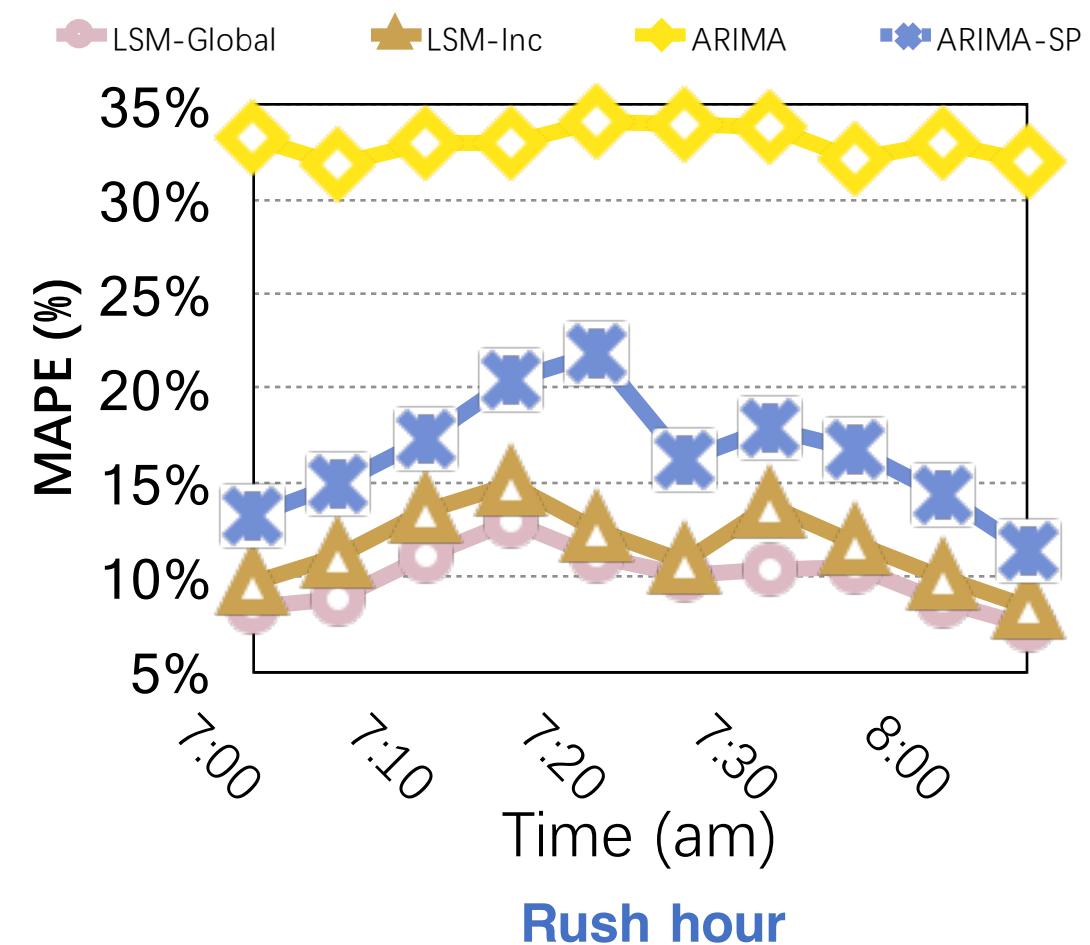
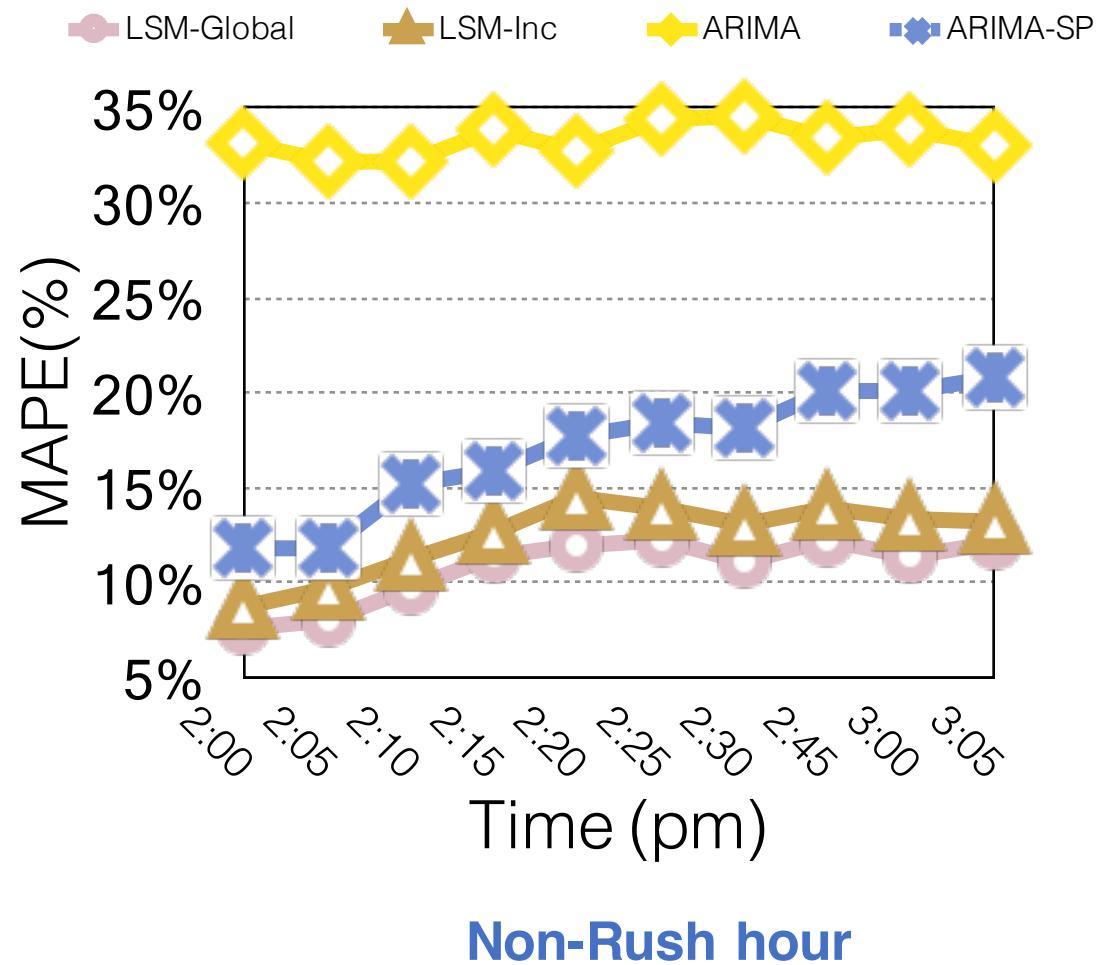
$$\begin{aligned} \arg \min_{U_i, B, A} J = & \sum_{i=1}^t \|Y_i \odot (G_i - U_i B U_i^T)\|_F^2 + \sum_{i=1}^t \lambda \text{Tr}(U_i^T L U_i) + \\ & \sum_{i=2}^t \gamma \|U_i - U_{i-1} A\|_F^2 \end{aligned}$$

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

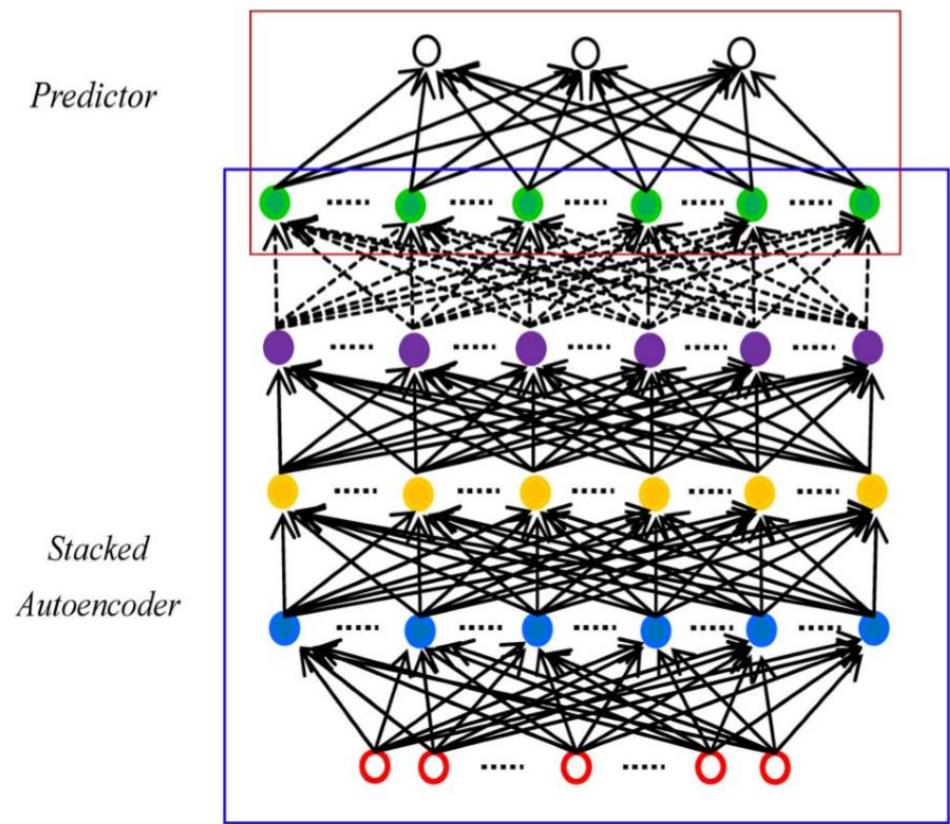
Experimental Results

Latent space models (LSM) outperform other baselines.



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



Deep Learning for Extreme Traffic Prediction

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



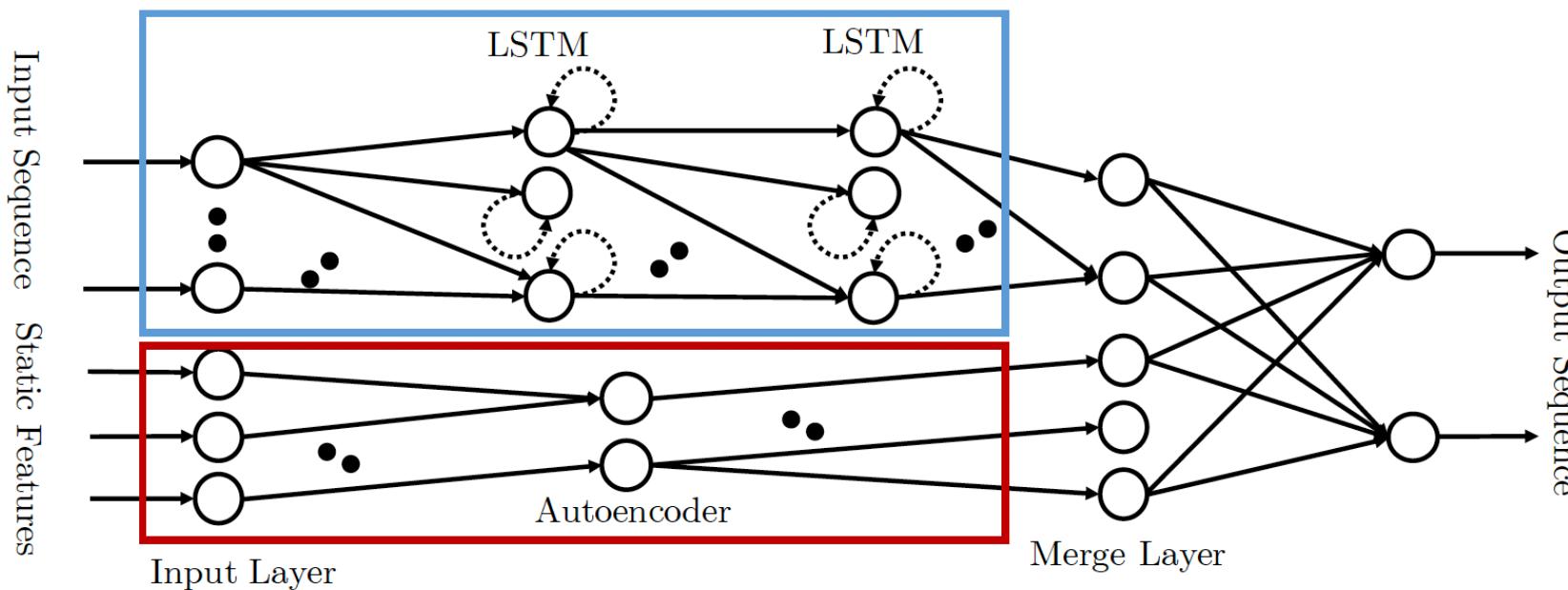
(a) Congestion



(b) Accident

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

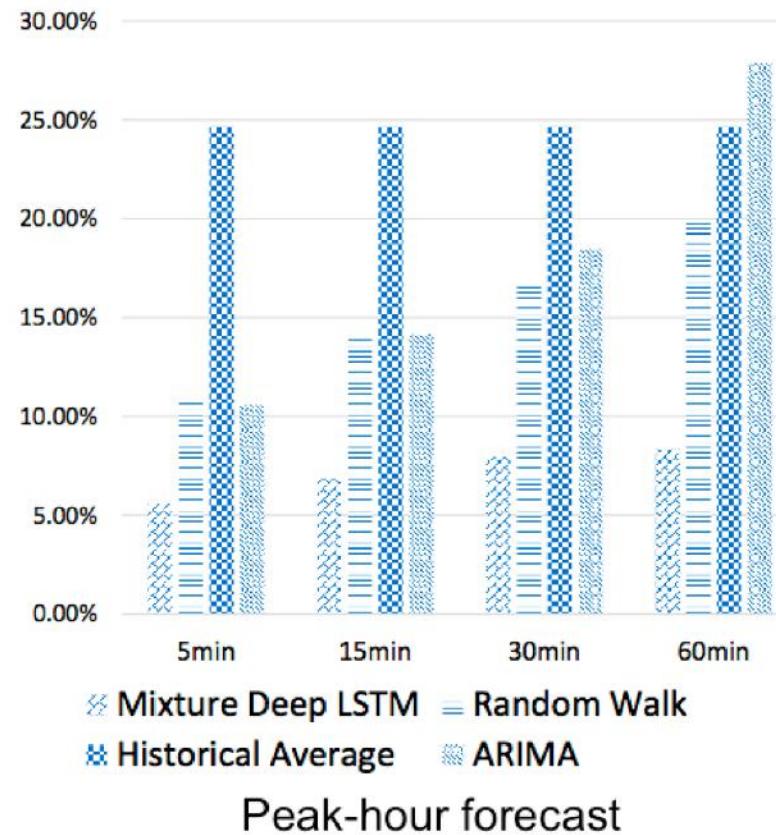
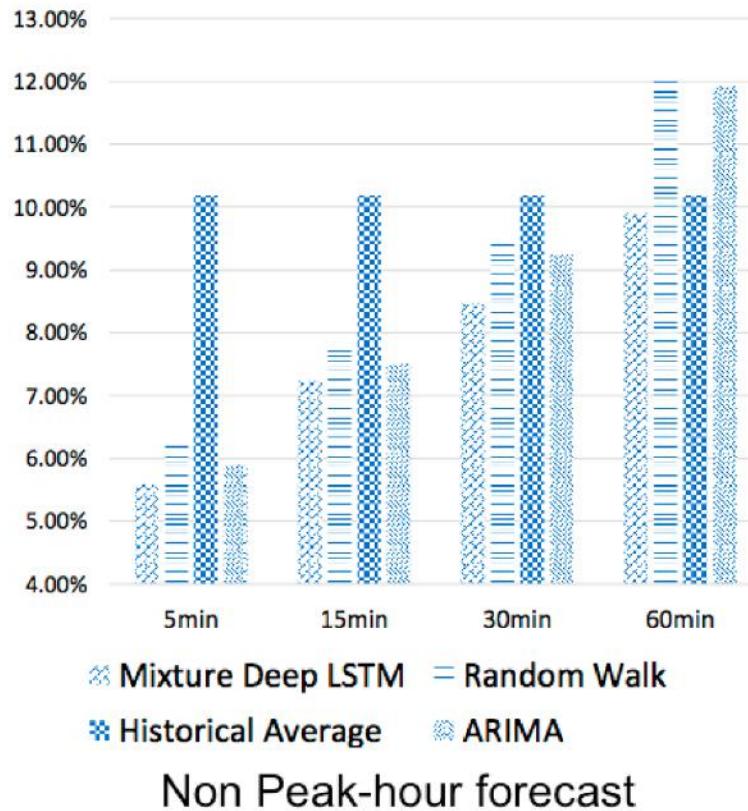


Experiments

- 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

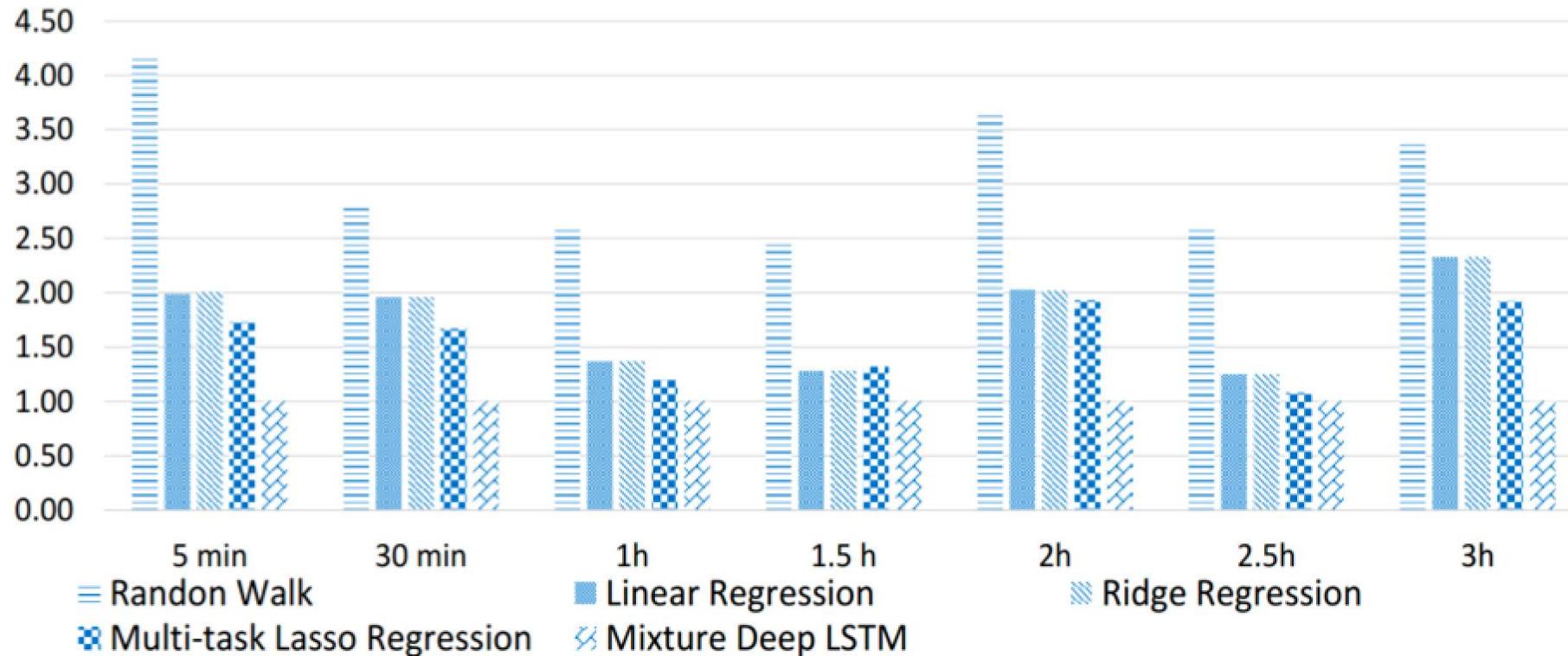
Rush-Hour Forecasting

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



Post-traffic Forecasting

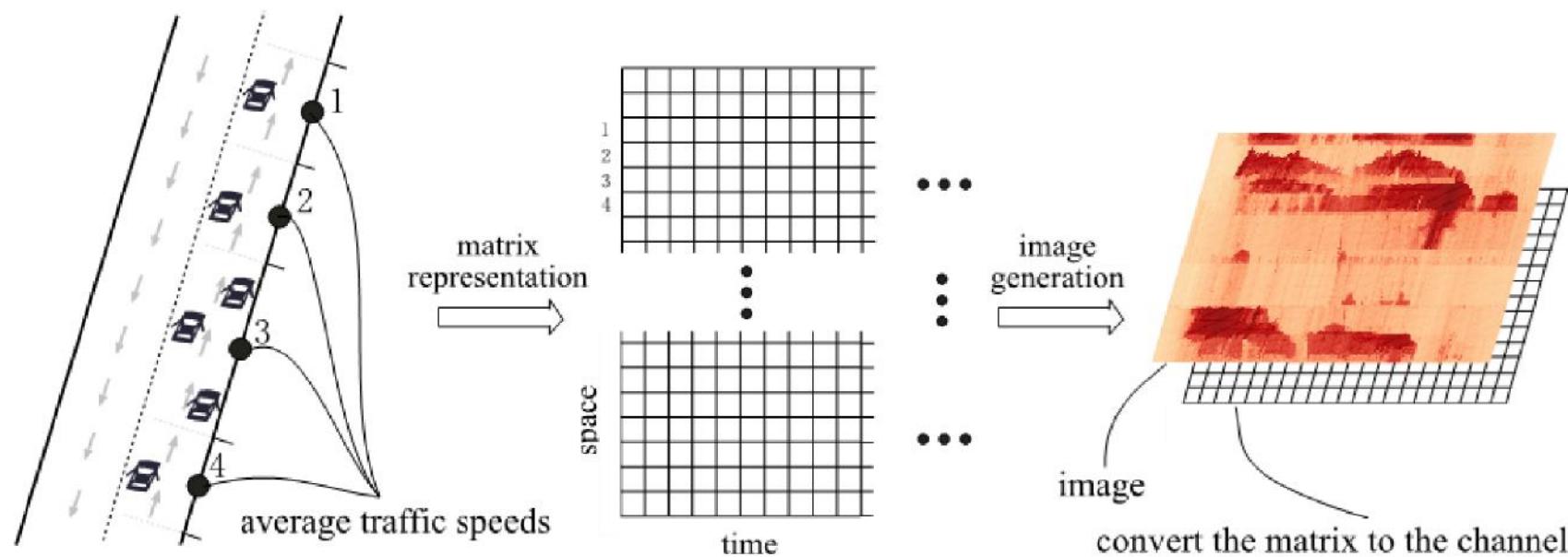
- Deep Mixture LSTM is roughly 30% better than the baseline methods



Post-accident traffic forecasting MAPE of different forecasting horizons

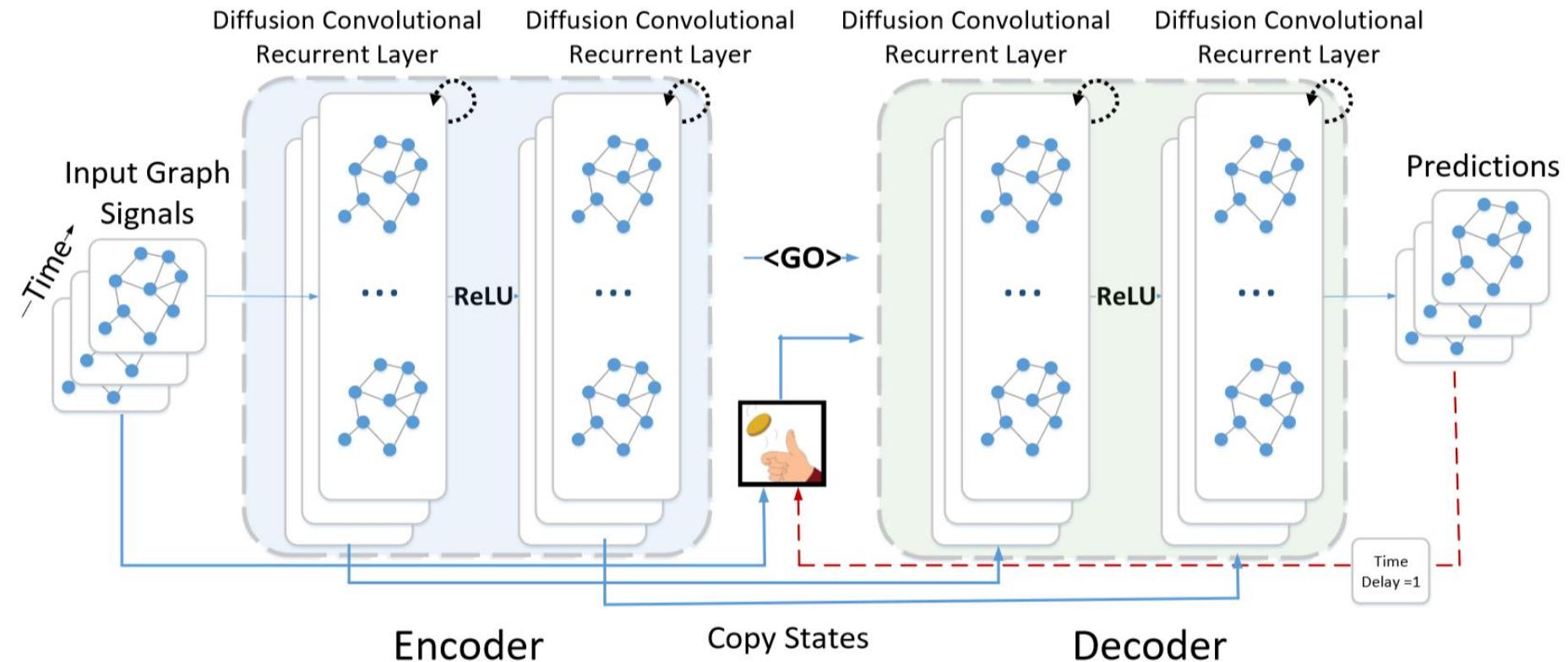
Convolution Neural Network for Traffic Forecasting

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



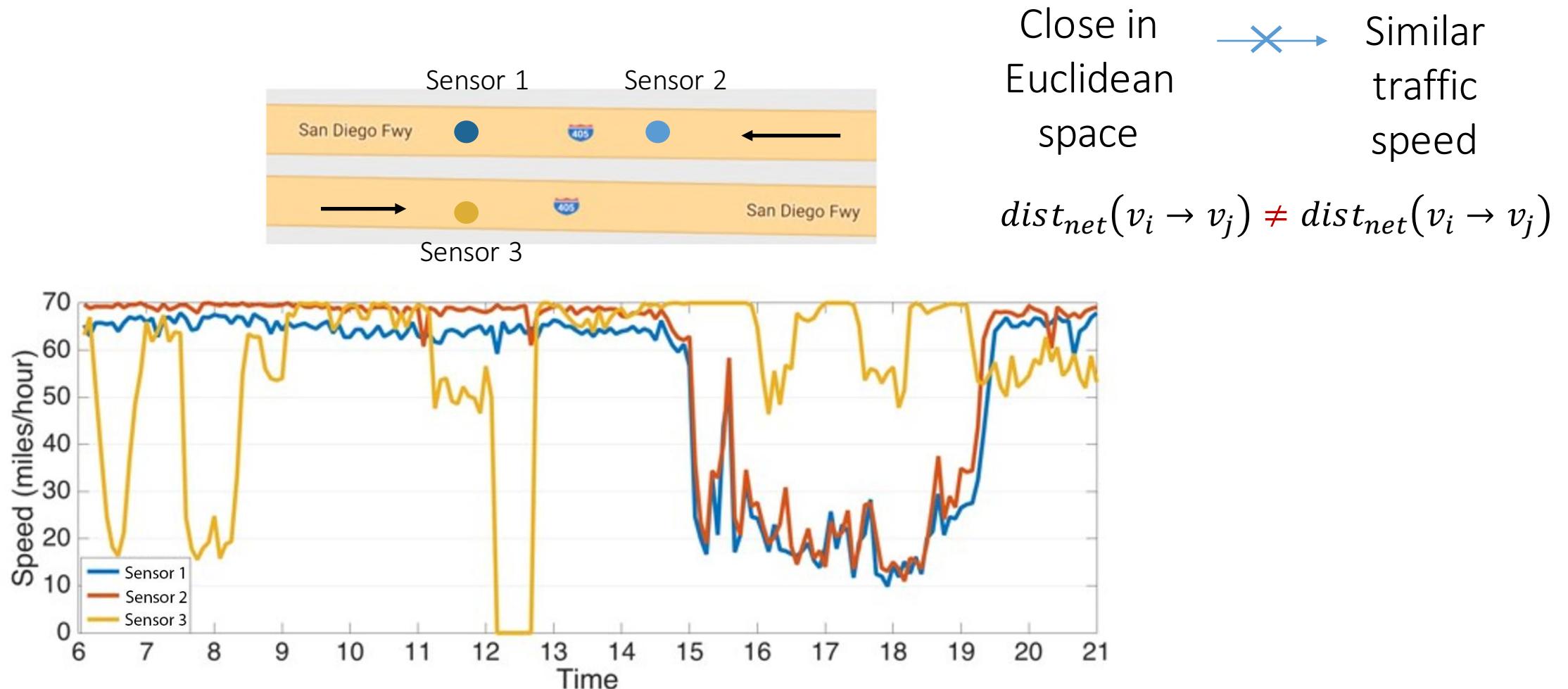
Traffic Forecasting with Convolution on Graph

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



Spatial Dependency in Traffic Prediction

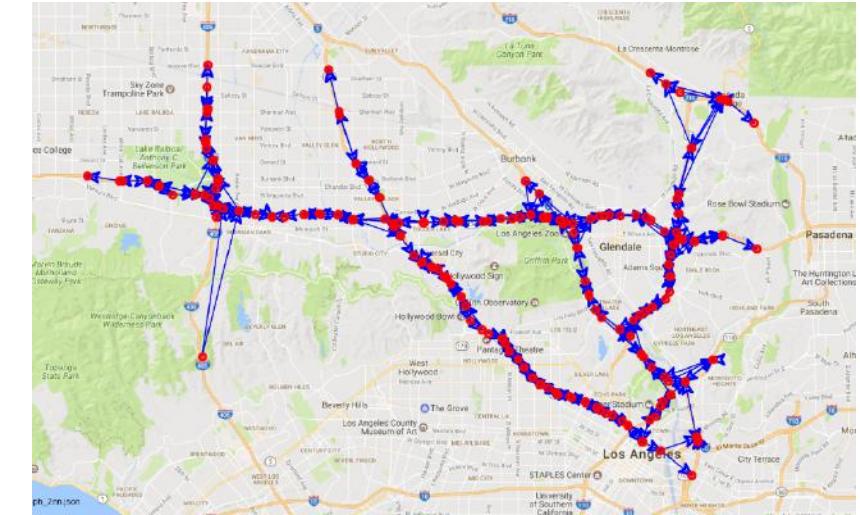
- Spatial dependency among traffic flow is *non-Euclidean* and *directed*



Spatial Dependency Modeling

- Model the network of traffic sensors, i.e., loop detectors, as a *directed graph*

- Graph $\mathcal{G} = (\mathbf{V}, \mathbf{A})$
- Vertices \mathbf{V} :  sensors
- Adjacency matrix \mathbf{A} :  weight between vertices



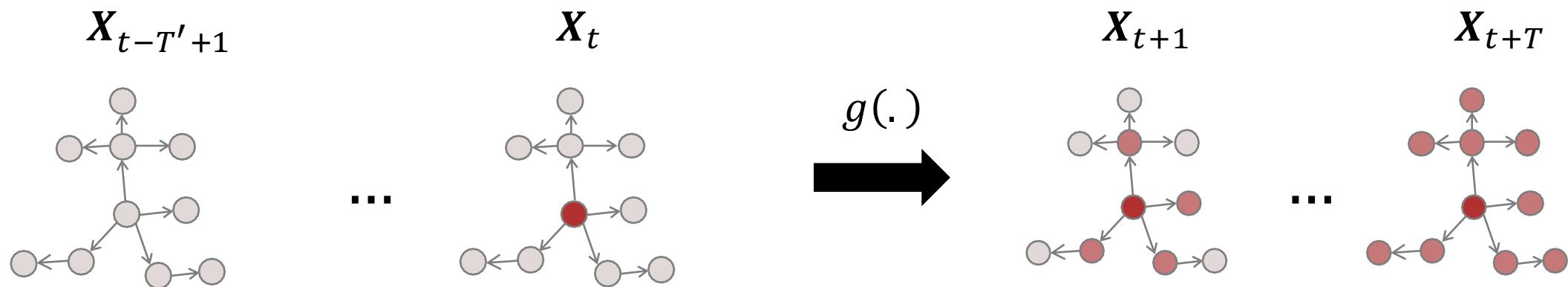
$$A_{ij} = \exp\left(-\frac{\text{dist}_{\text{net}}(v_i, v_j)^2}{\sigma^2}\right) \text{ if } \text{dist}_{\text{net}}(v_i, v_j) \leq \kappa$$

$\text{dist}_{\text{net}}(v_i, v_j)$: road network distance from v_i to v_j ,

κ : threshold to ensure sparsity, σ^2 variance of all pairwise road network distances

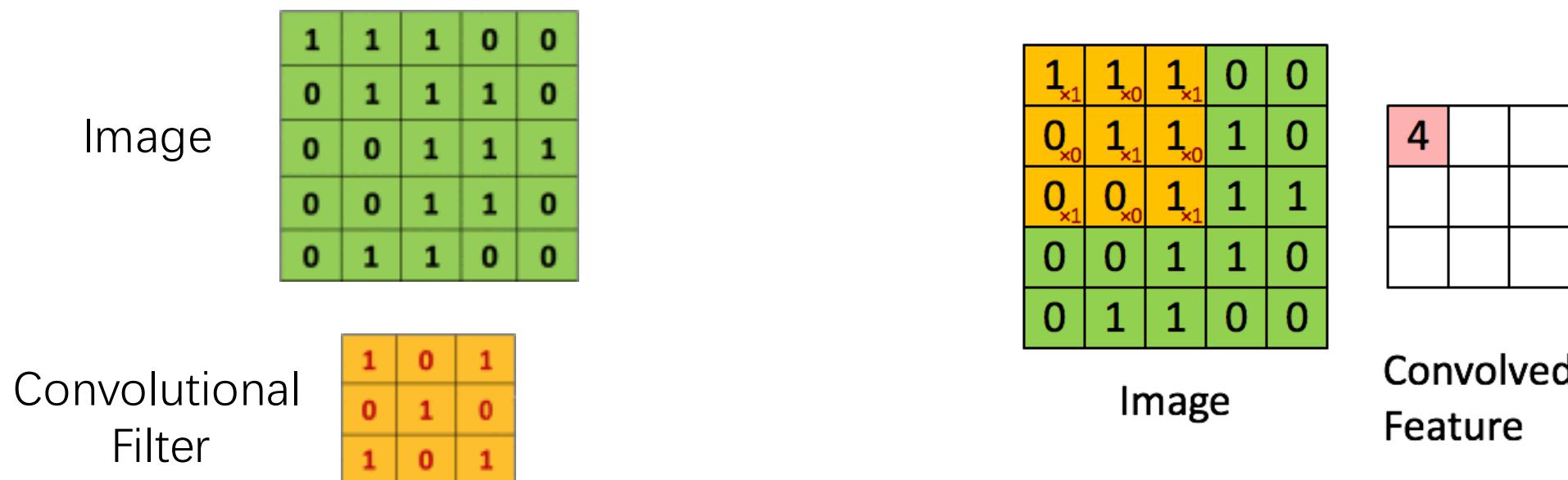
Problem Statement

- Graph signal: $\mathbf{X}_t \in \mathbb{R}^{|V| \times P}$, observation on \mathcal{G} at time t
 - $|V|$: 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

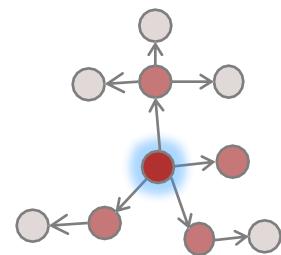
Generalize Convolution to Graph

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

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

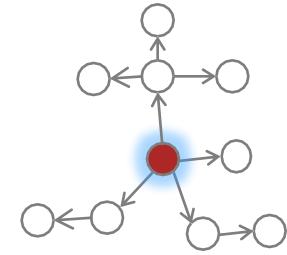
Transition matrices of
the diffusion process

Learning complexity: $O(K)$

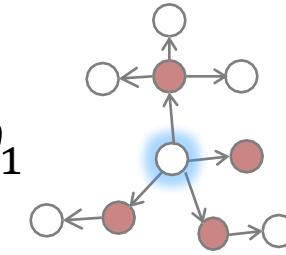


$$= \theta_0$$

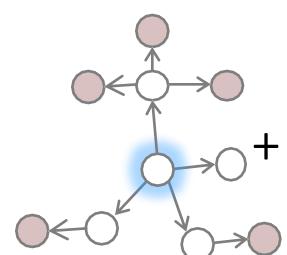
Example diffusion filter
Centered at



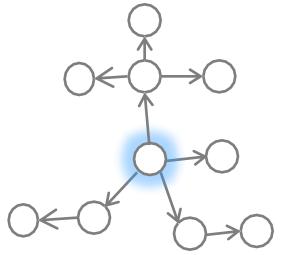
0 Step
Diffusion



1 Step
Diffusion



2 Step
Diffusion



K Step
Diffusion

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

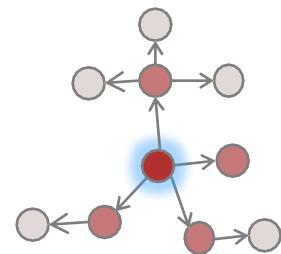


Filter weight

Generalize Convolution to Graph

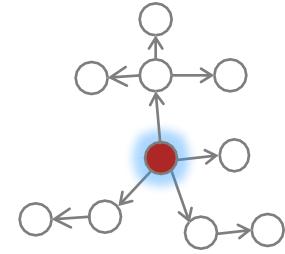
- Diffusion convolution filter: combination of *diffusion processes* with different steps on the graph.
Dual directional diffusion to model upstream and downstream separately

$$\mathbf{X}_{:,p} \star_g f_\theta = \sum_{k=0}^{K-1} \left(\theta_{k,1} (\mathbf{D}_O^{-1} \mathbf{A})^k + \theta_{k,2} (\mathbf{D}_I^{-1} \mathbf{A}^\top)^k \right) \mathbf{X}_{:,p}$$

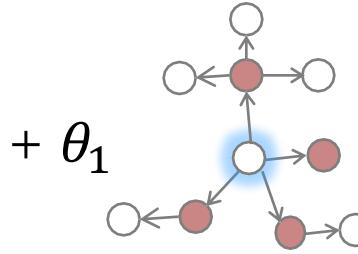


$$= \theta_0$$

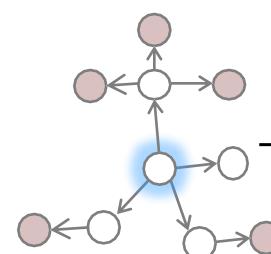
Example diffusion filter
Centered at



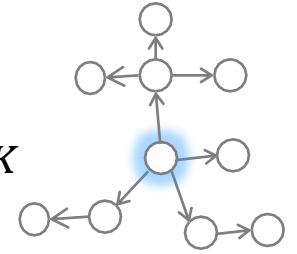
0 Step
Diffusion



1 Step
Diffusion



2 Step
Diffusion



K Step
Diffusion

\star_g : diffusion convolution, D_O : diagonal out-degree matrix, D_I : diagonal in-degree matrix

weight



Min

Max

Advantage of Diffusion Convolution

$$\mathbf{X}_{:,p} \star_{\mathcal{G}} f_{\theta} = \sum_{k=0}^{K-1} \left(\theta_{k,1} (\mathbf{D}_o^{-1} \mathbf{A})^k + \theta_{k,2} (\mathbf{D}_I^{-1} \mathbf{A}^\top)^k \right) \mathbf{X}_{:,p}$$

- 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

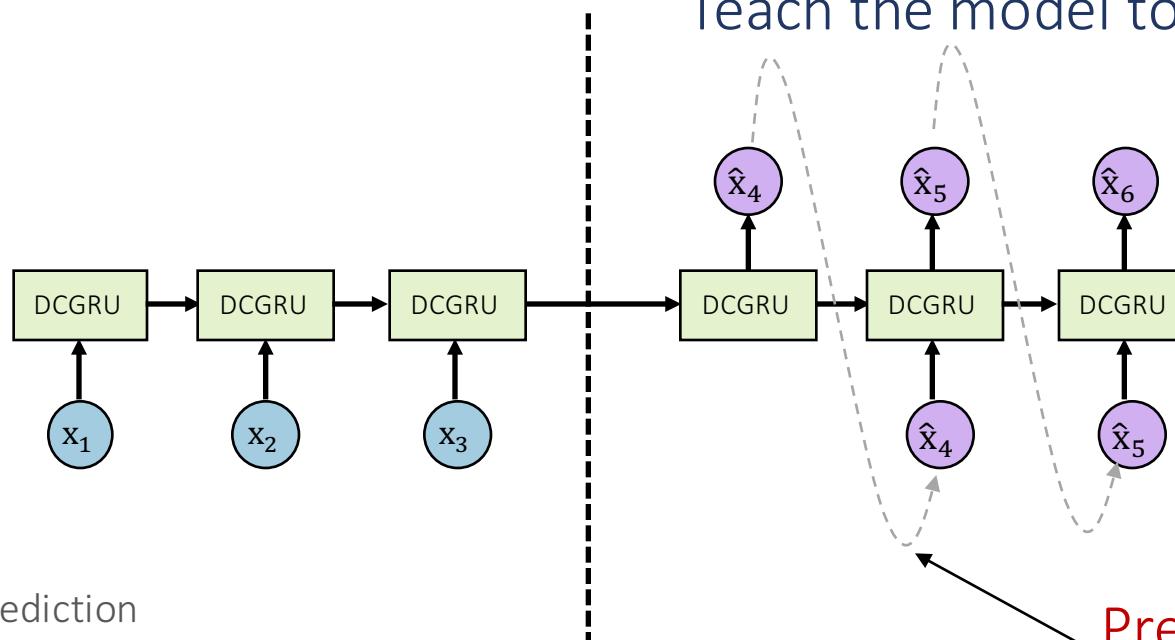
Model Temporal Dynamics using Recurrent Neural Network

Multi-step ahead prediction with RNN

Current Time

Error Propagation

Teach the model to deal with its own error.



Model prediction

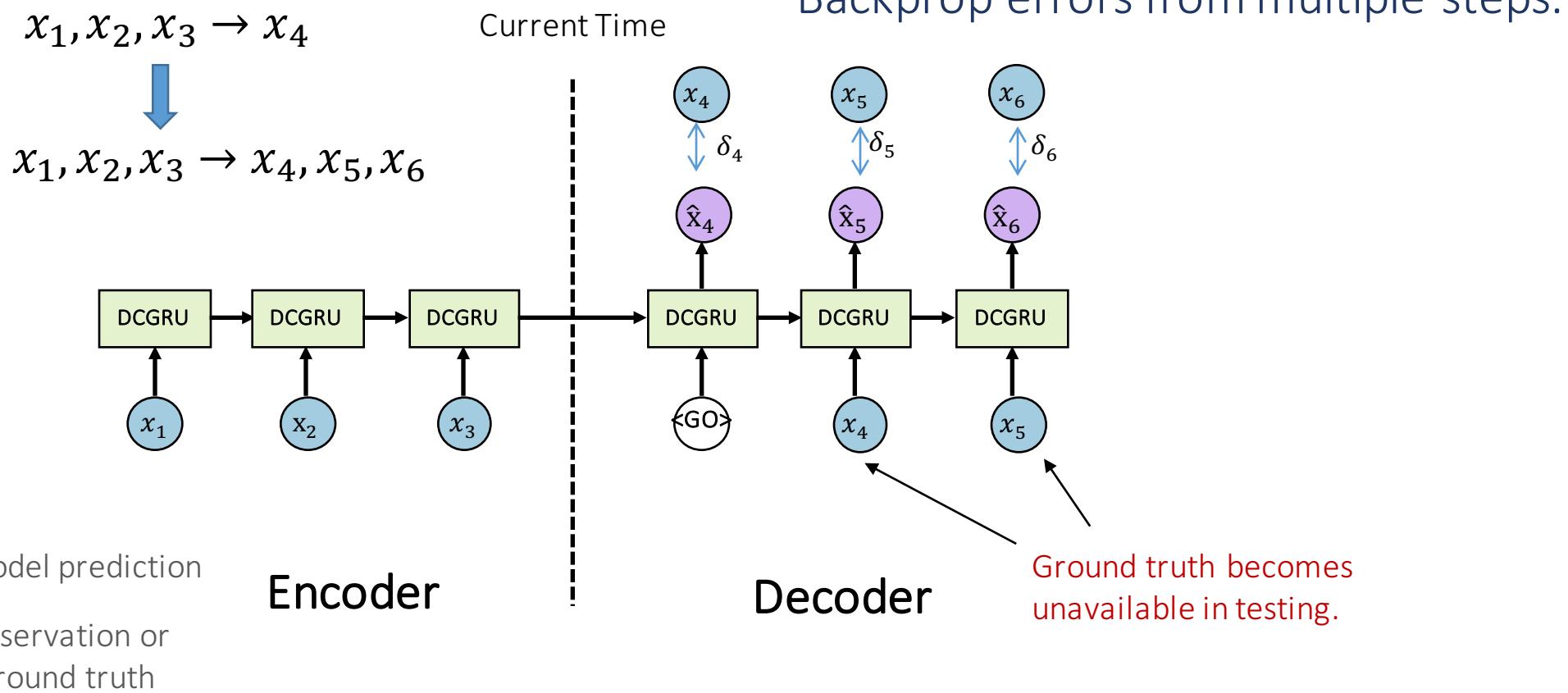


Observation or
ground truth

Previous model
output is fed into
the network

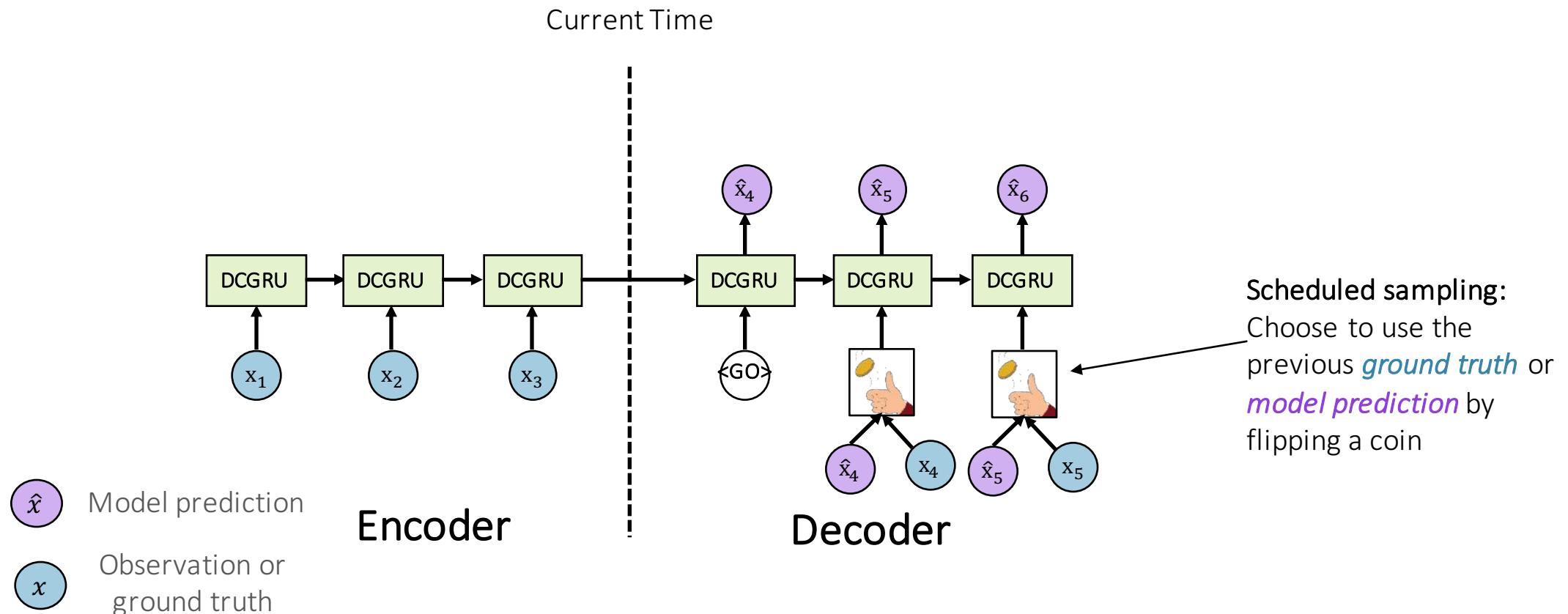
Improve Multi-step ahead Forecasting

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



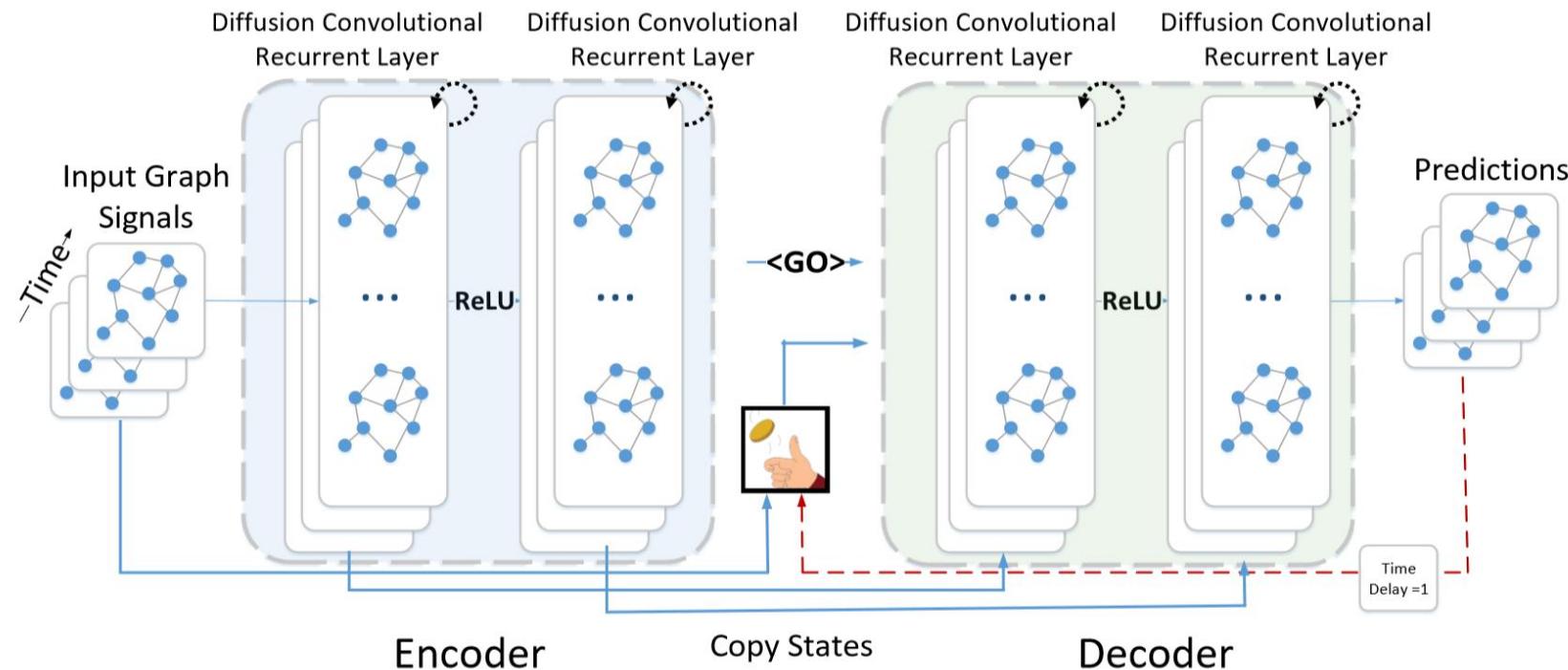
Improve Multi-step ahead Forecasting

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



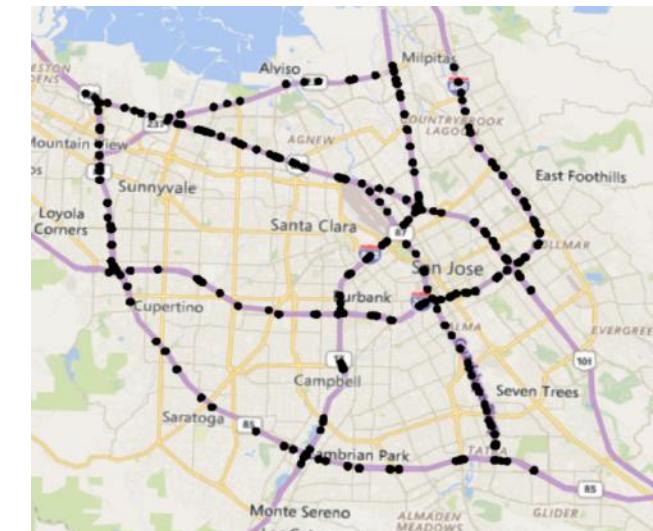
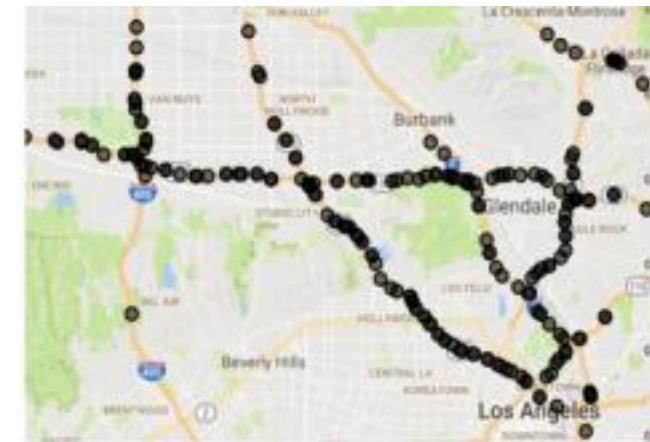
Diffusion Convolutional Recurrent Neural Network

- 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*



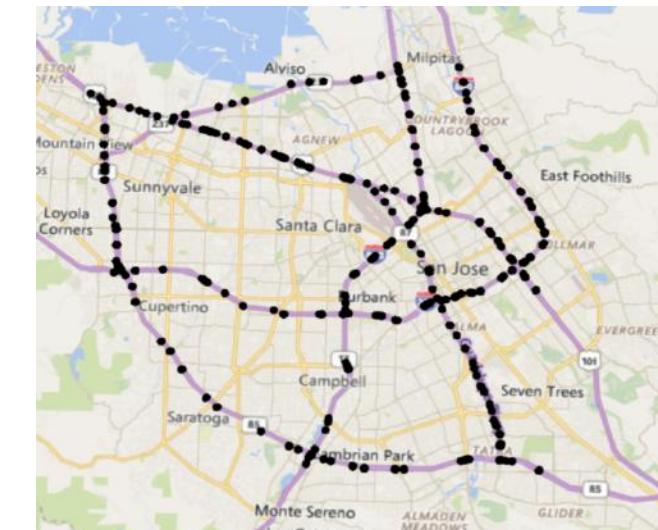
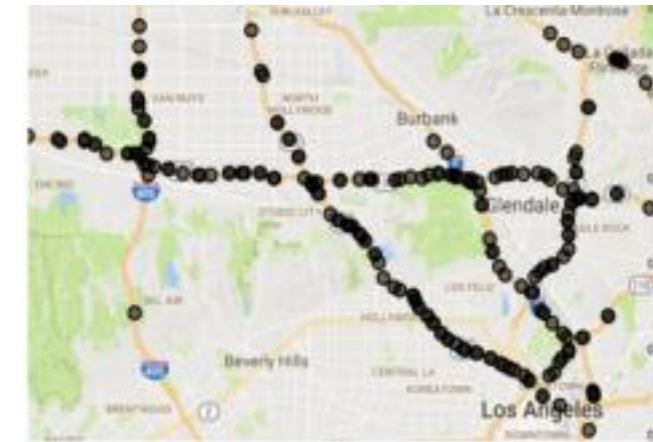
Experiment - Datasets

- 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



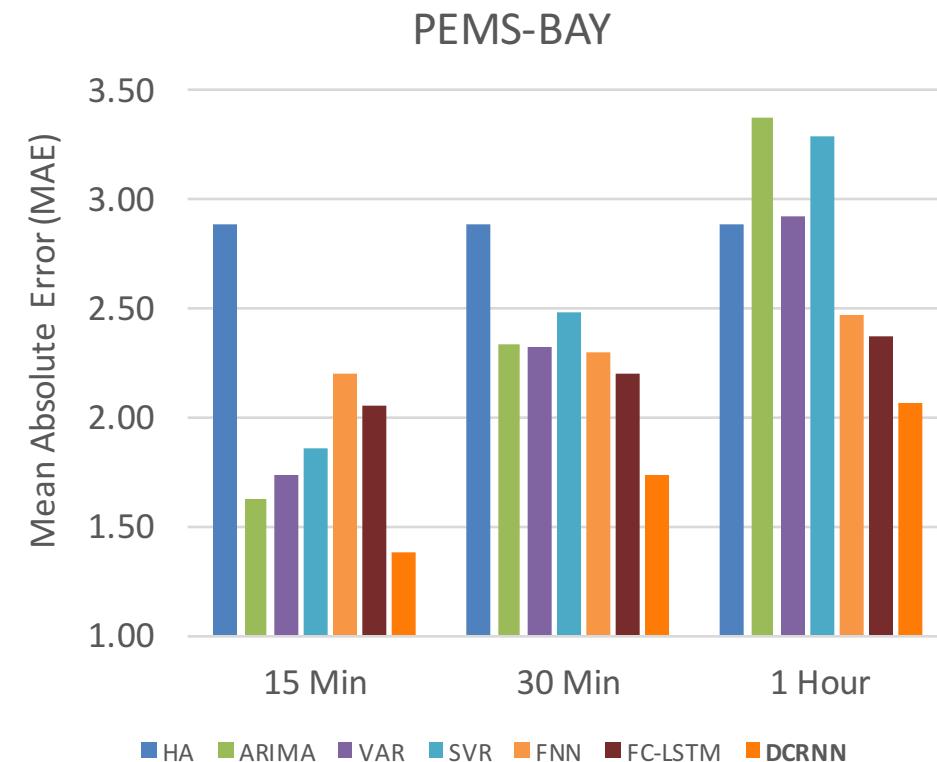
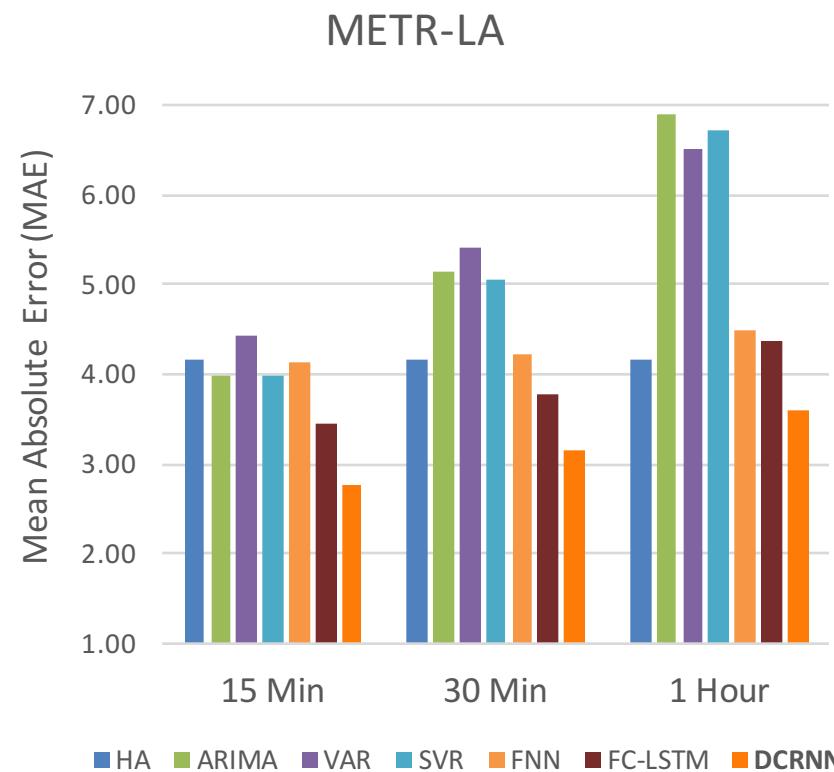
Experiments

- 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

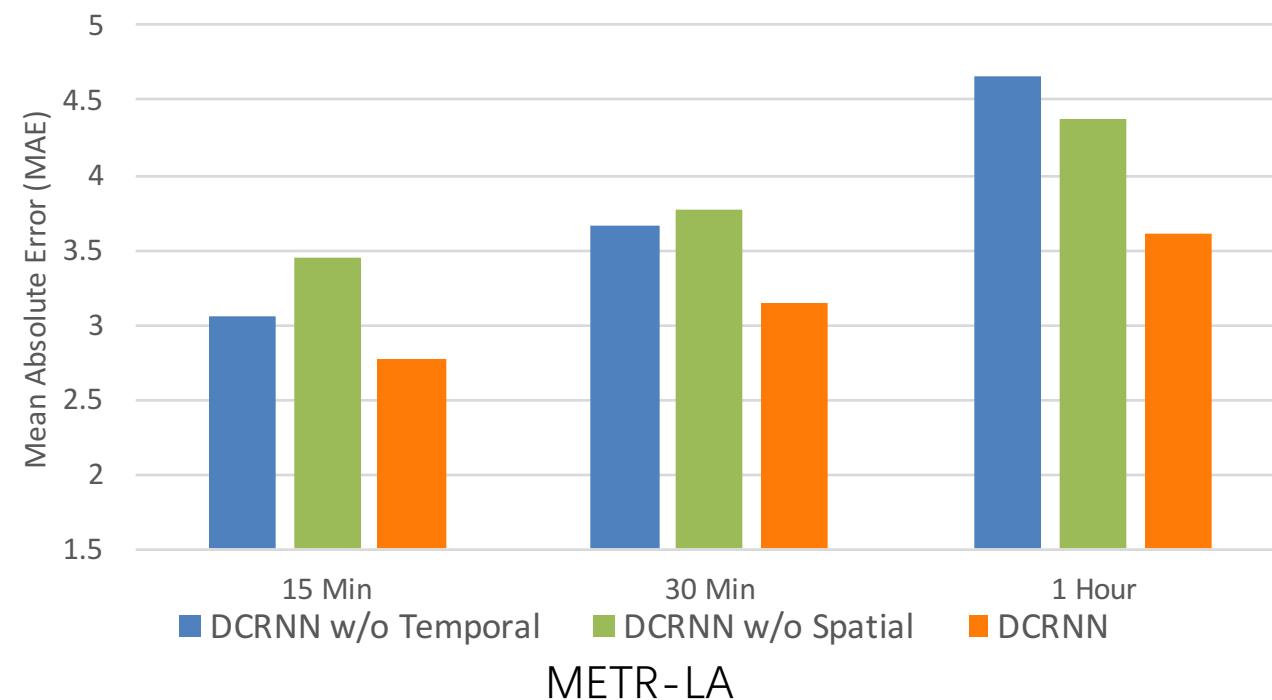
- DCRNN achieves the *best performance* for all forecasting horizons for both datasets



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

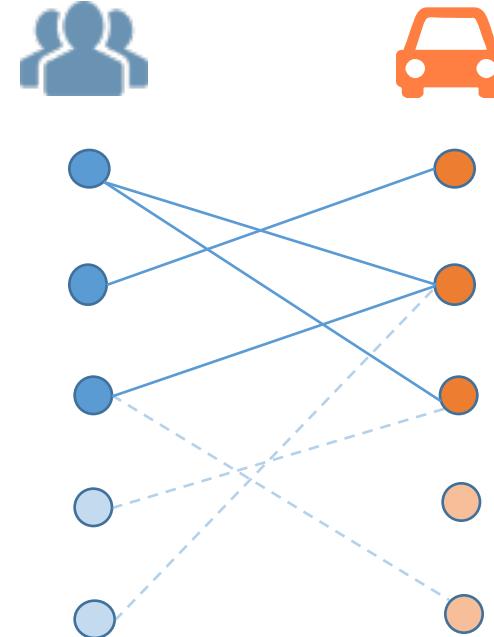
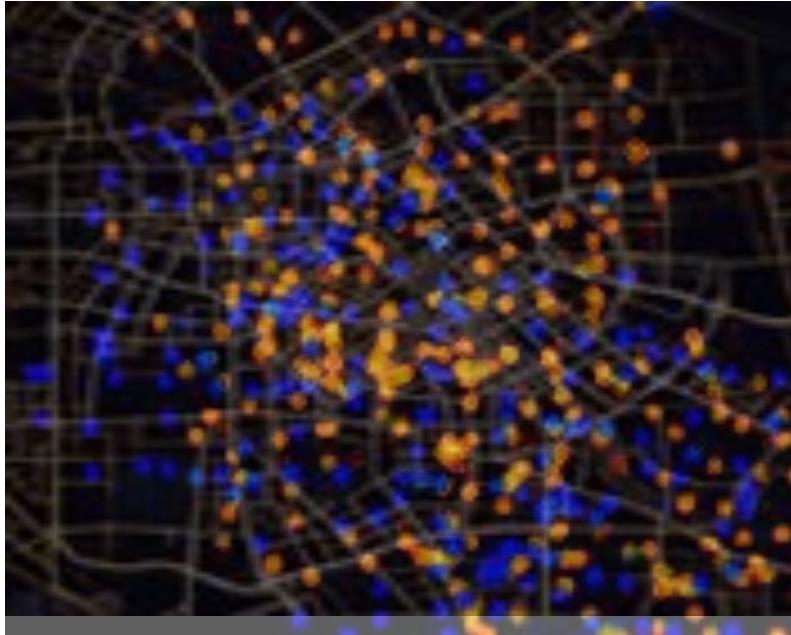
Reference

- [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

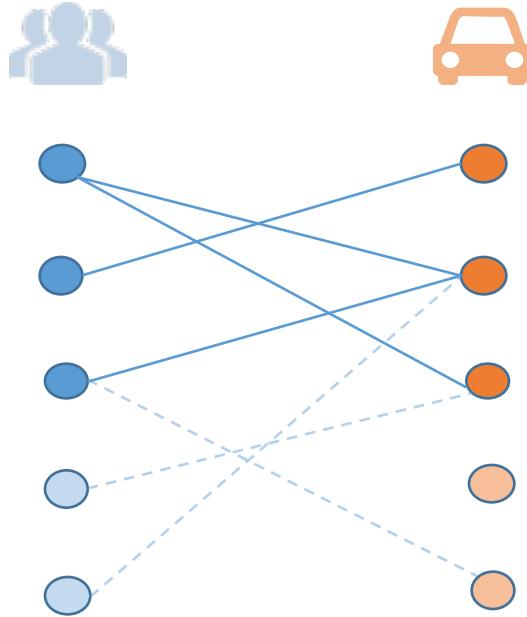
(Estimated Time of Arrival)

- 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)

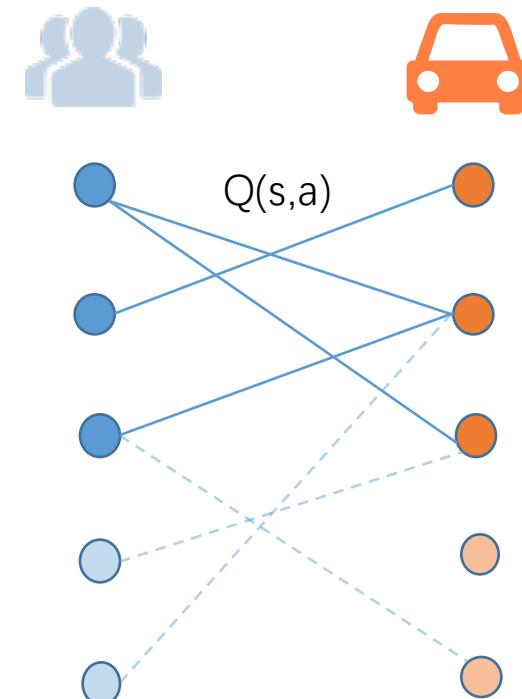
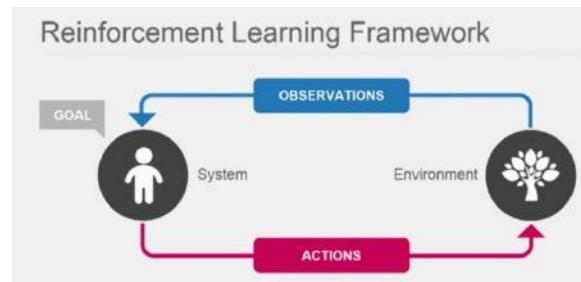


MDP Formulation for Dispatching

Reinforcement Learning

Dispatching

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

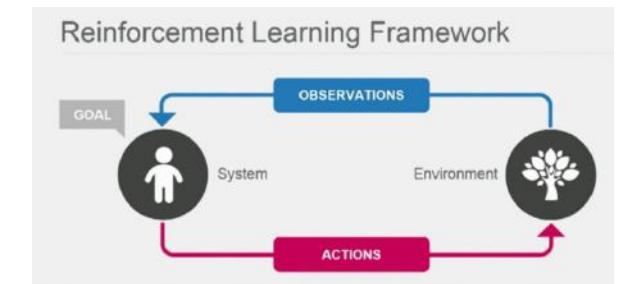
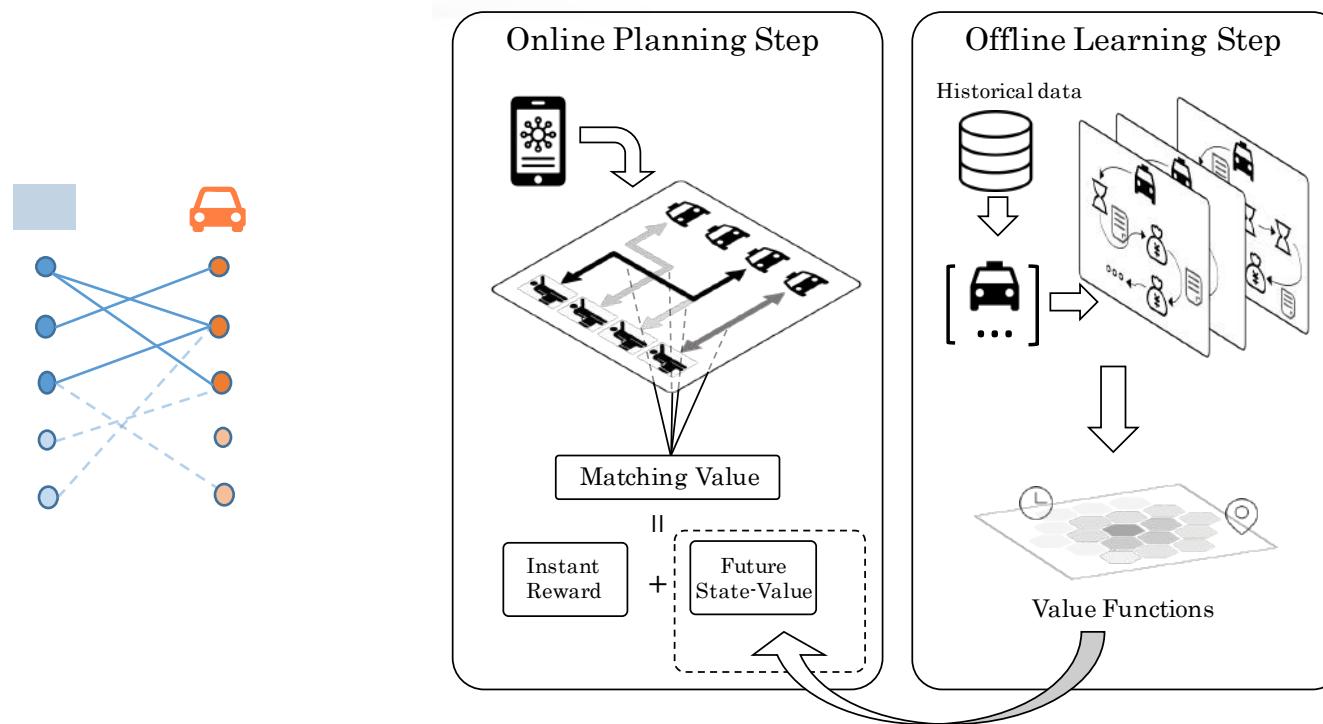


Reinforcement Learning

- 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

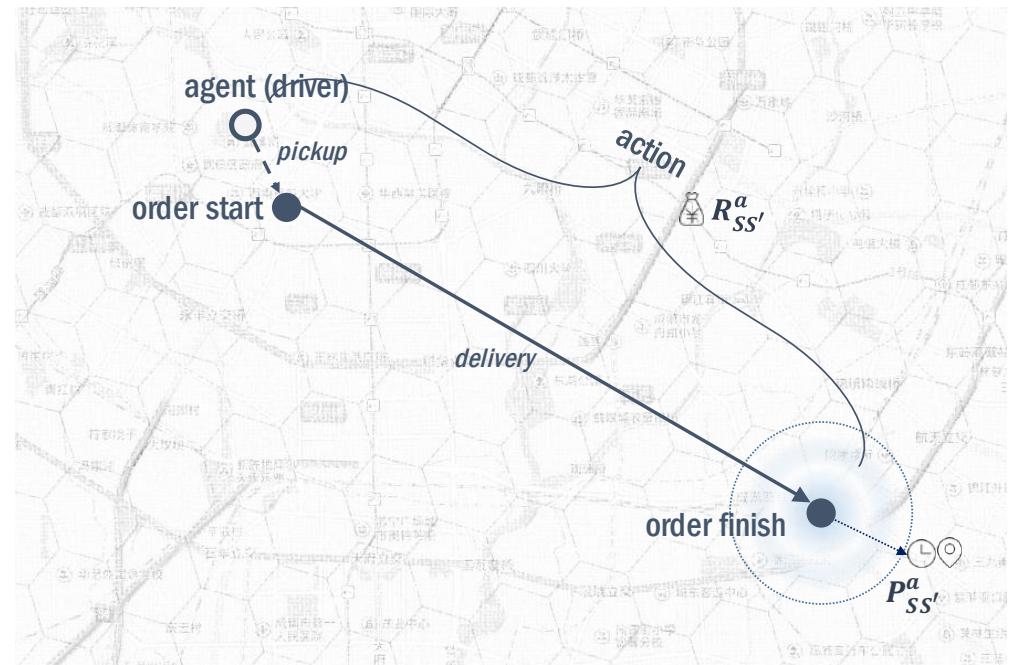


Xu et al., KDD 2018

MDP Definition

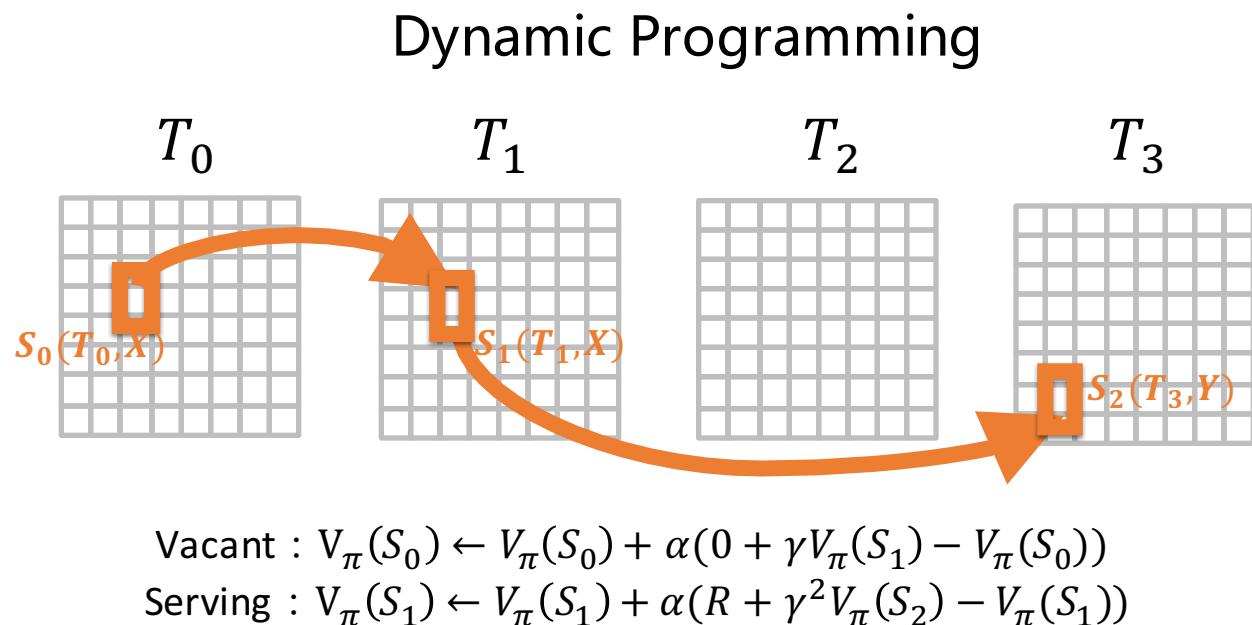
Motivation

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

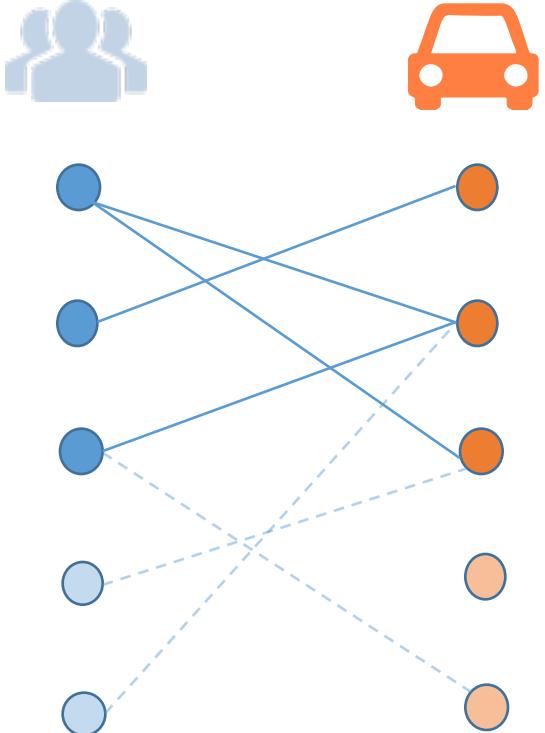


- Agent: A Driver
- State: Time & Space location
- Action: Pickup an order or do nothing
- Reward: Driver's Income (order price)
- Episode: A whole day from 04:00 to 04:00 (+1)
- Value Function: The expected future income of a driver in a state S

Learning – Policy Evaluation



Planning – Advantage Function



Link weight:

$$\begin{aligned} \text{Link weight:} \\ Q(s, a) \\ \rightarrow A(s, a) \\ = R. + V(s') - V(s) \end{aligned}$$

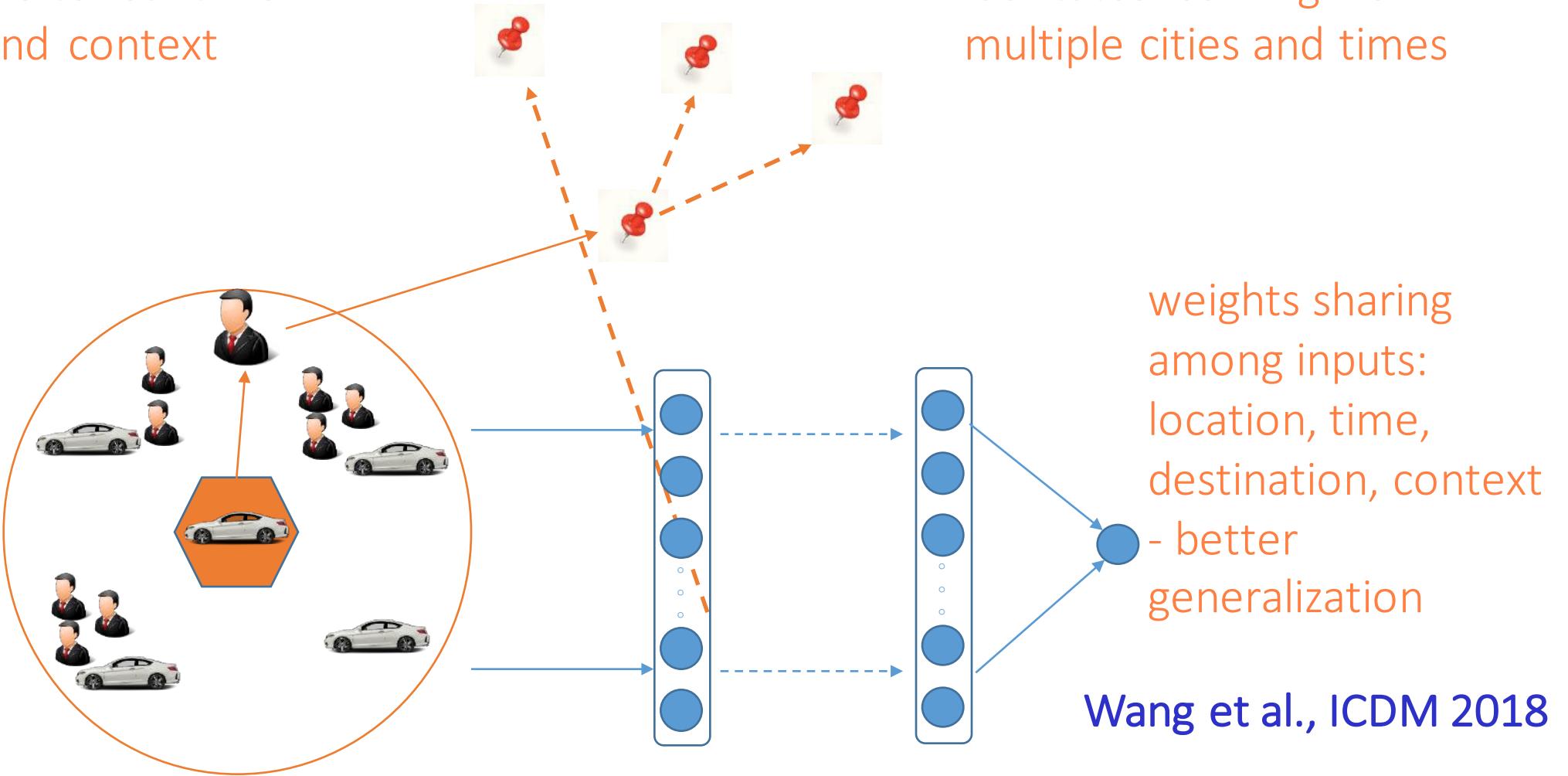
Order's value Value function of driver's expected finishing state Value function of driver's current state



Deep RL for Dispatching

Deep Reinforcement Learning

more adaptive to real-time
supply-demand context
changes



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 (s_0, a, r, s_1) :
 - Current state $s_0 := (l_0, t_0, f_0)$, location, time, and contextual features
 - Action: the assigned trip;
 - Next-state $s_1 := (l_1, t_1, f_1)$
 - Reward: the total fee collected for the trip.
- Implicitly consider pick-up time, trip duration relative to reward

Deep Q-network with action search

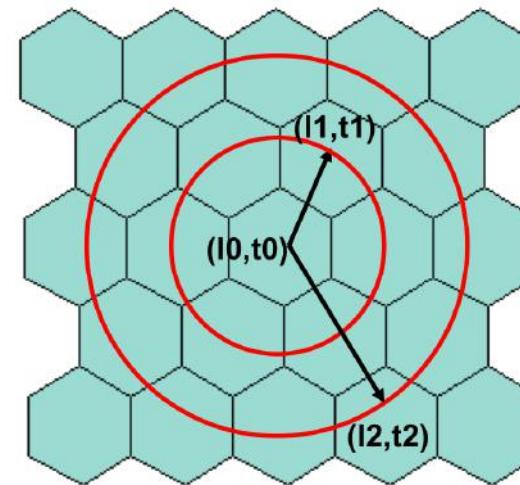
- ❑ Training data
- ❑ Action search
- ❑ Expanded action search

- 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

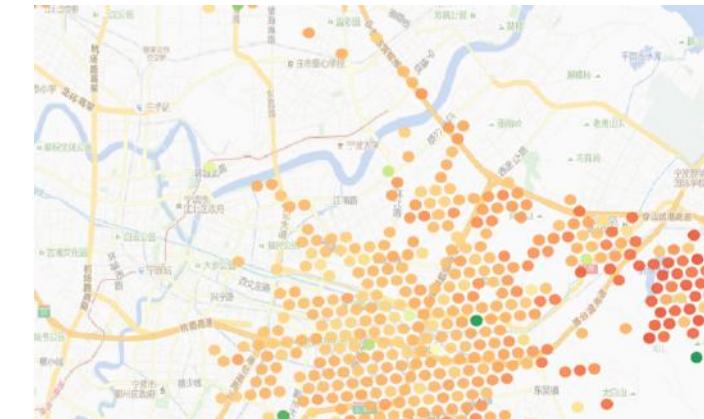
- Due to training data sparsity in certain spatio-temporal regions, we perform an expanded action search in both spatial and temporal spaces.



Training for multiple cities

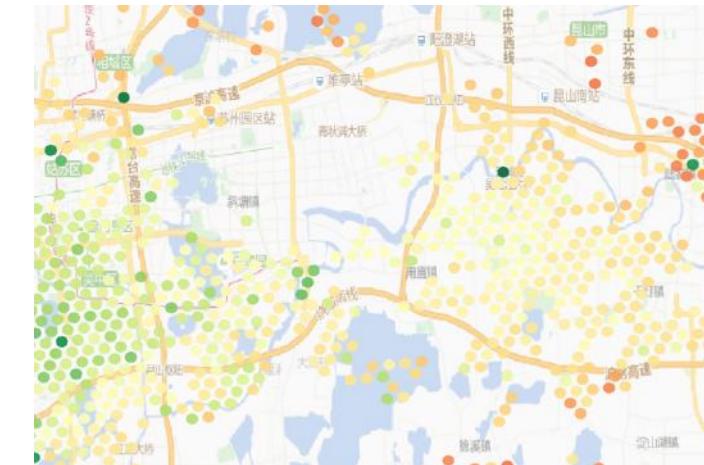
Dispatching system supports a large number of cities

- 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

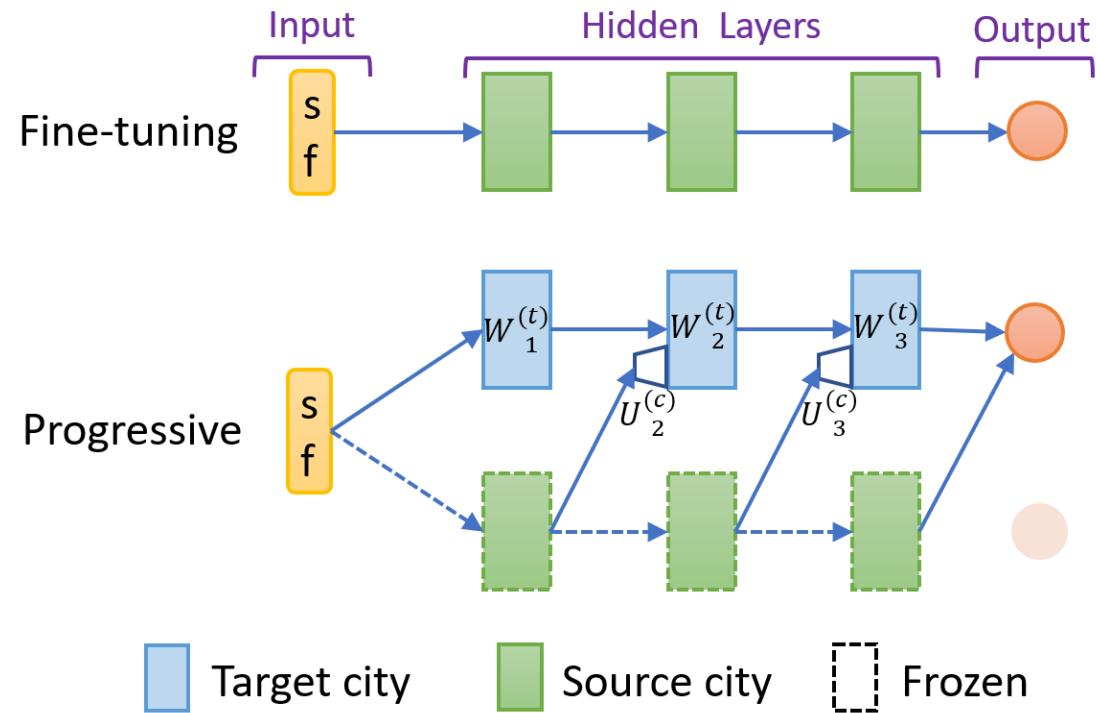
Transfer Learning

Idea

- Re-use weights
- Better initial solution

Existing Methods

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



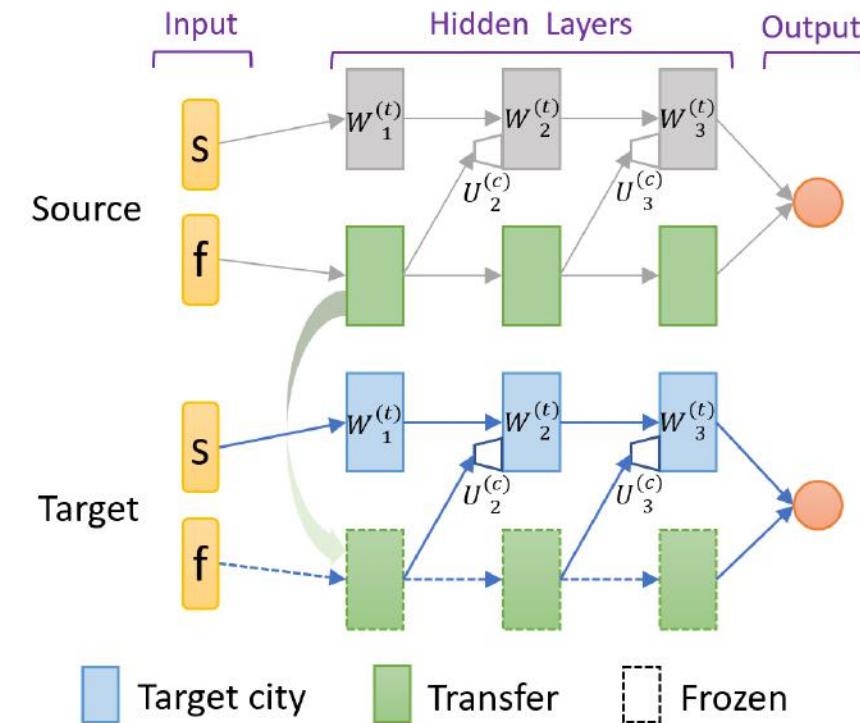
Correlated Feature Progressive Transfer (CFPT)

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

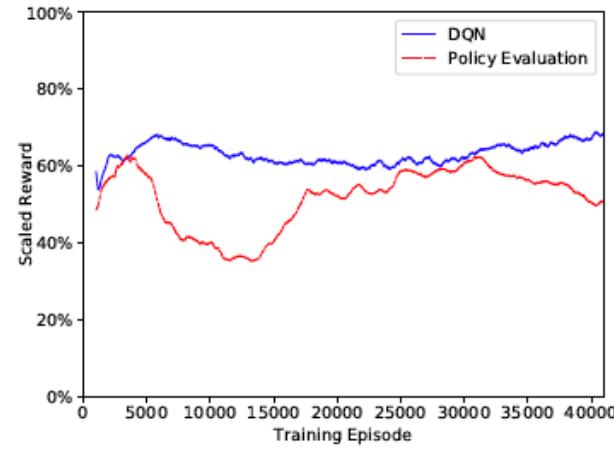
- Same city: one time period to another

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

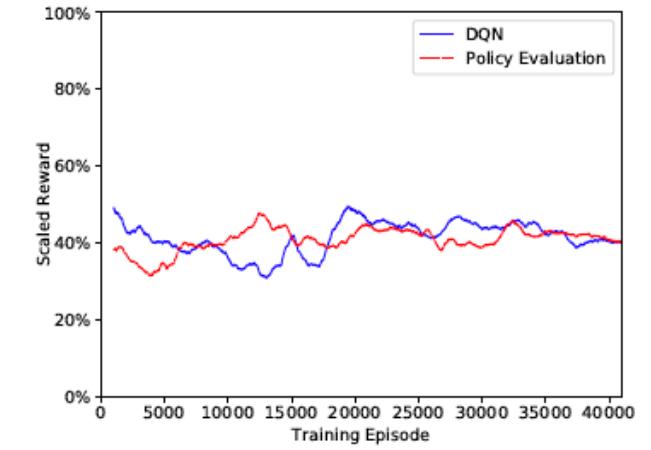
Results

DQN v.s. policy evaluation

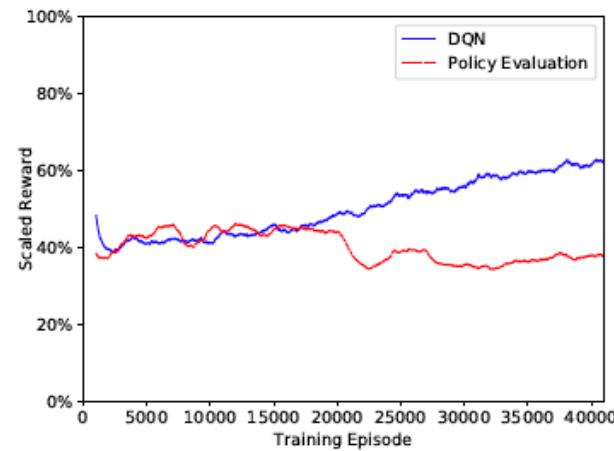
- Optimization helps
- Smaller cities with simpler pattern and fewer transitions have smaller advantages



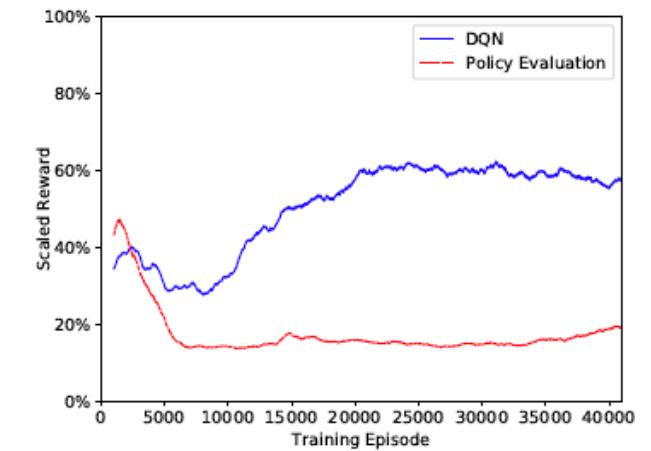
(a) City A



(b) City B



(c) City C



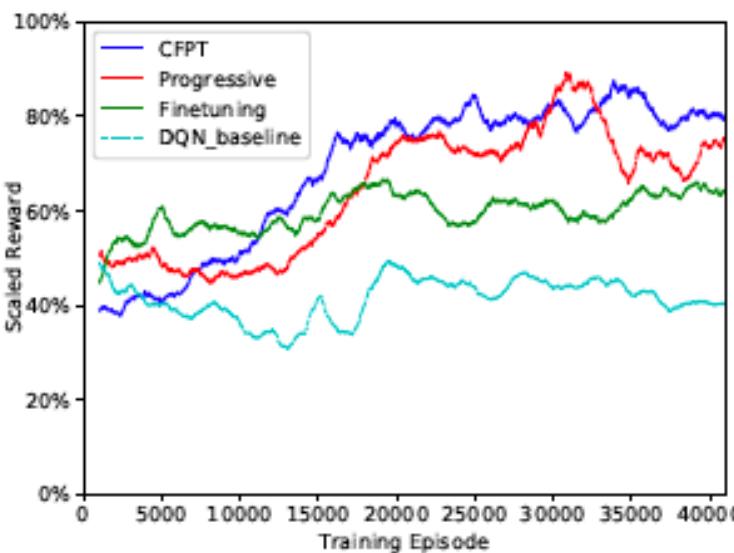
(d) City D

Results

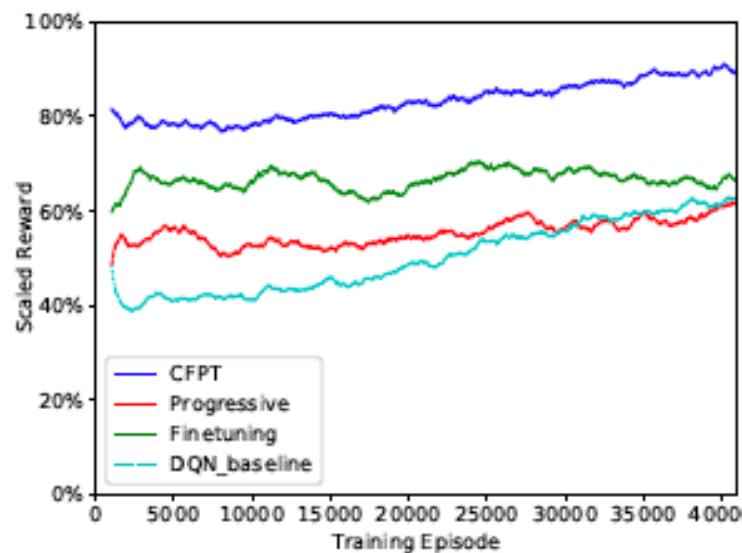
No transfer v.s. transfer

Spatial transfer

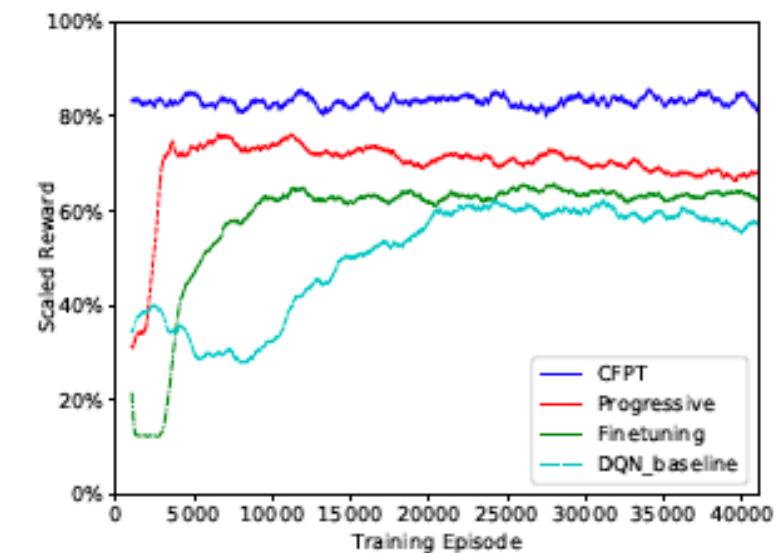
- Robustness
- Better initial solution
- Higher converged cumulative rewards



(a) City B



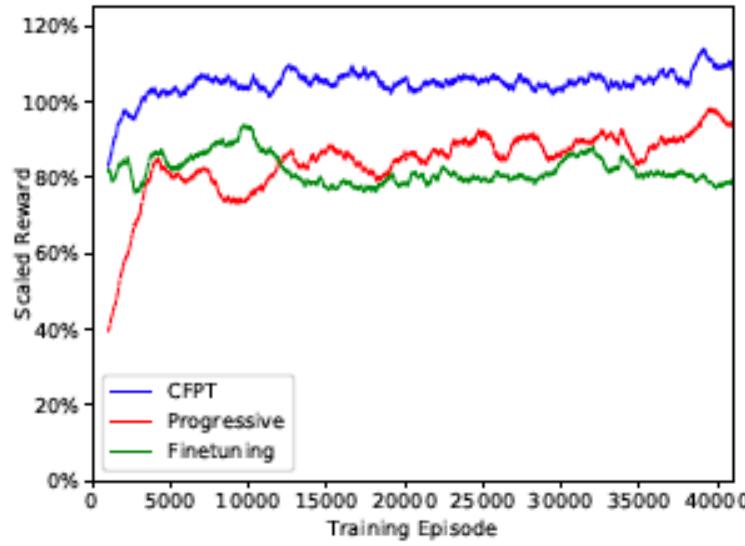
(b) City C



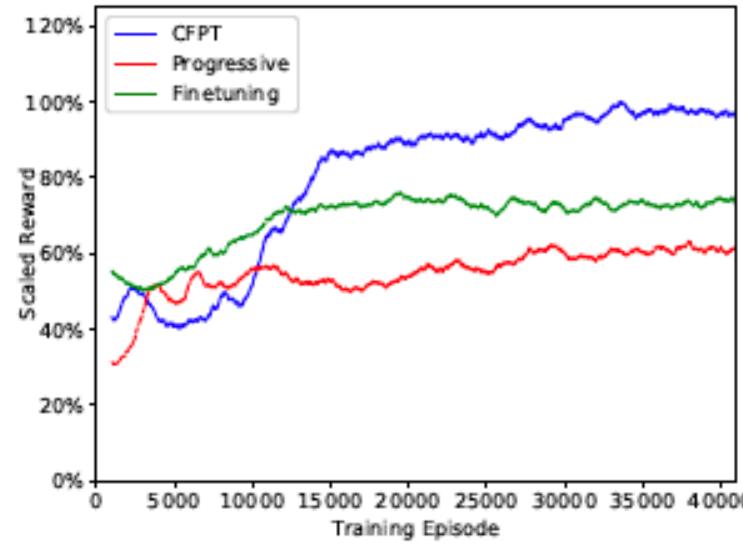
(c) City D

Results

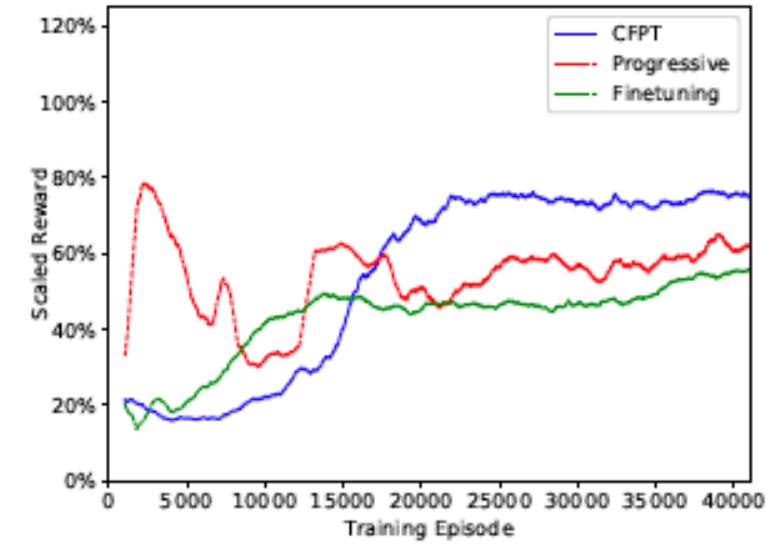
Temporal transfer



(a) City B



(b) City C



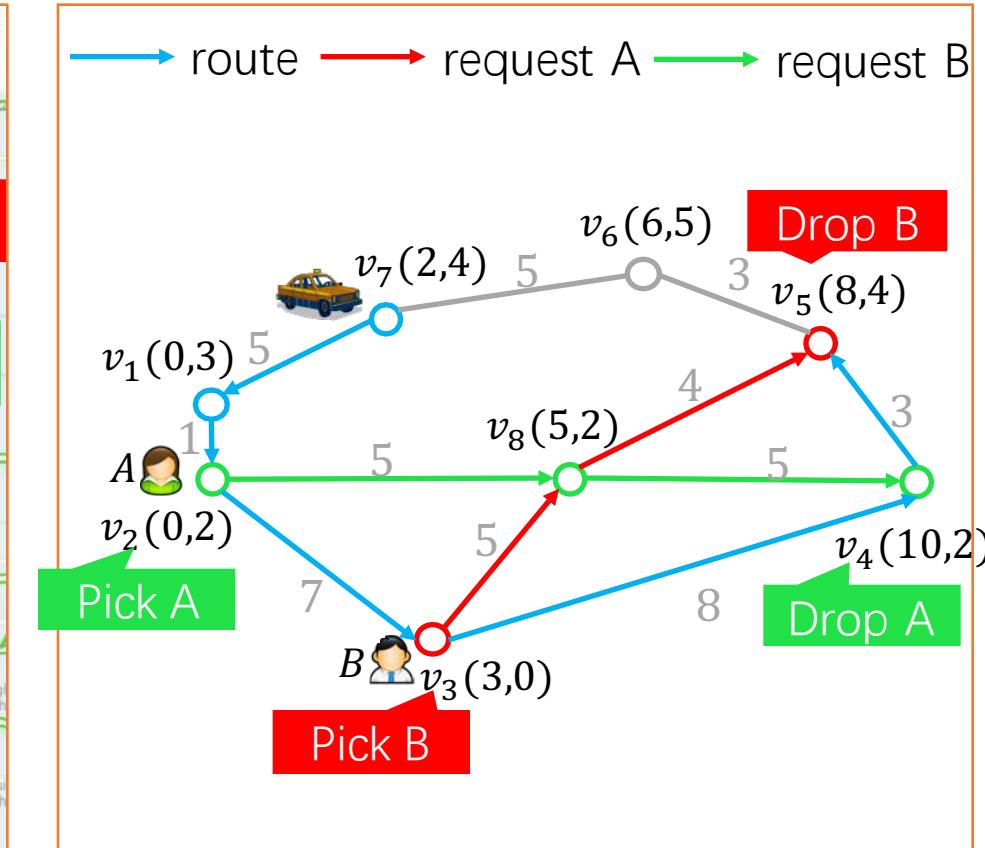
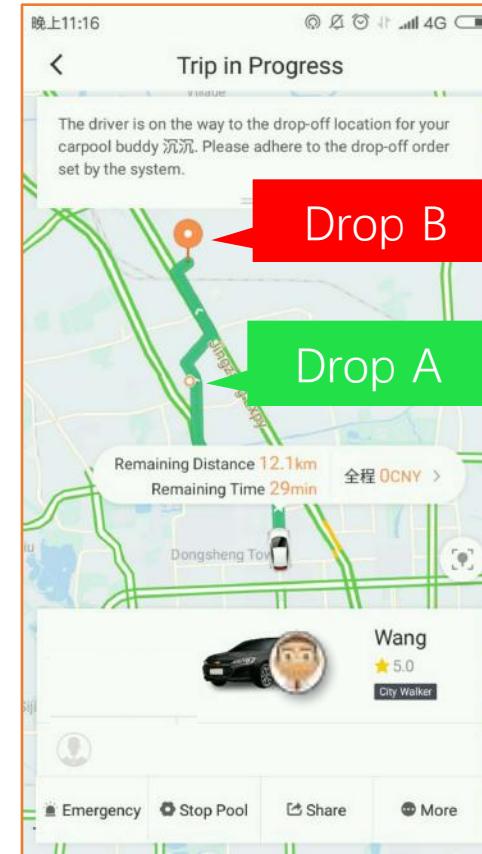
(c) City D



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**



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**.

- $$UC(W, R) = \alpha \cdot \sum_{w \in W} D(S_w) + \sum_{r \in R^-} p_r$$

- α : weight, p_r : penalty for rejecting the request

- $$\begin{cases} \alpha = 1 \\ p_r = \infty \end{cases}$$

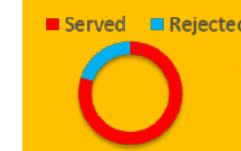
\Rightarrow



Minimize Total Distance

- $$\begin{cases} \alpha = 0 \\ p_r = 1 \end{cases}$$

\Rightarrow



Maximize Served Request

- $$\begin{cases} \alpha = \text{unit payment} \\ p_r = \text{fare of the request} \end{cases}$$

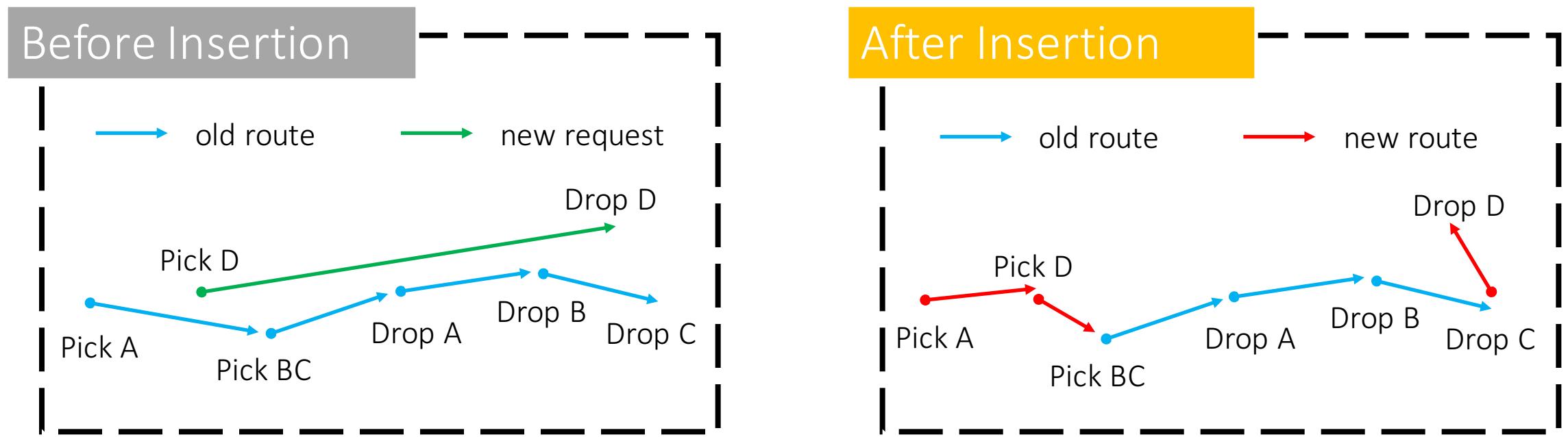
\Rightarrow



Maximize Total Revenue

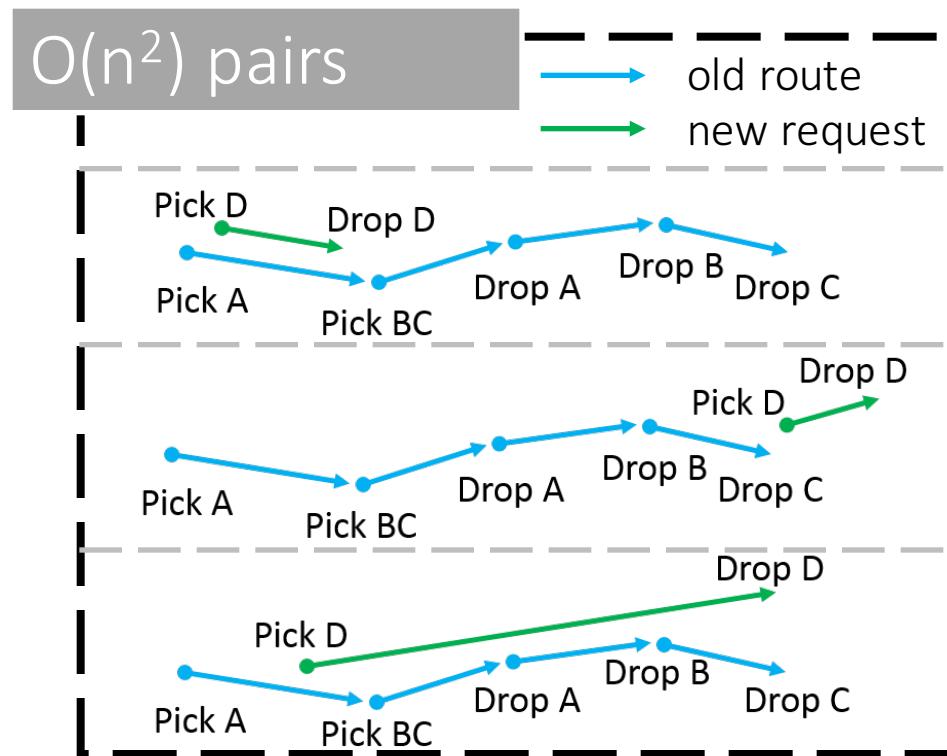
Core Operation in Ridesharing: Insertion

- 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.



Dynamic Programming based Insertion

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



Dynamic Programming

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

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

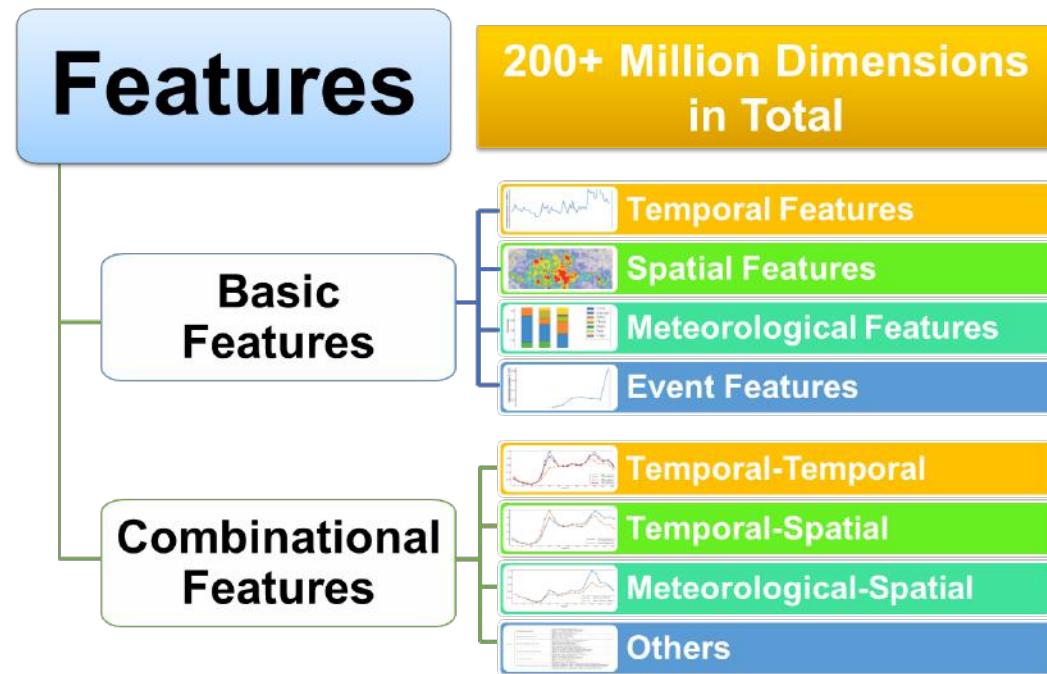
Reference

- [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.

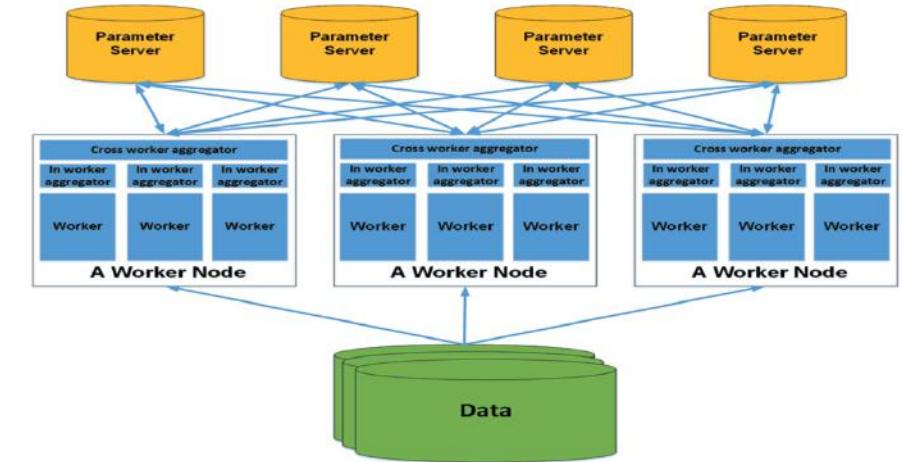


Supply and Demand Forecasting

Supply and Demand Forecasting



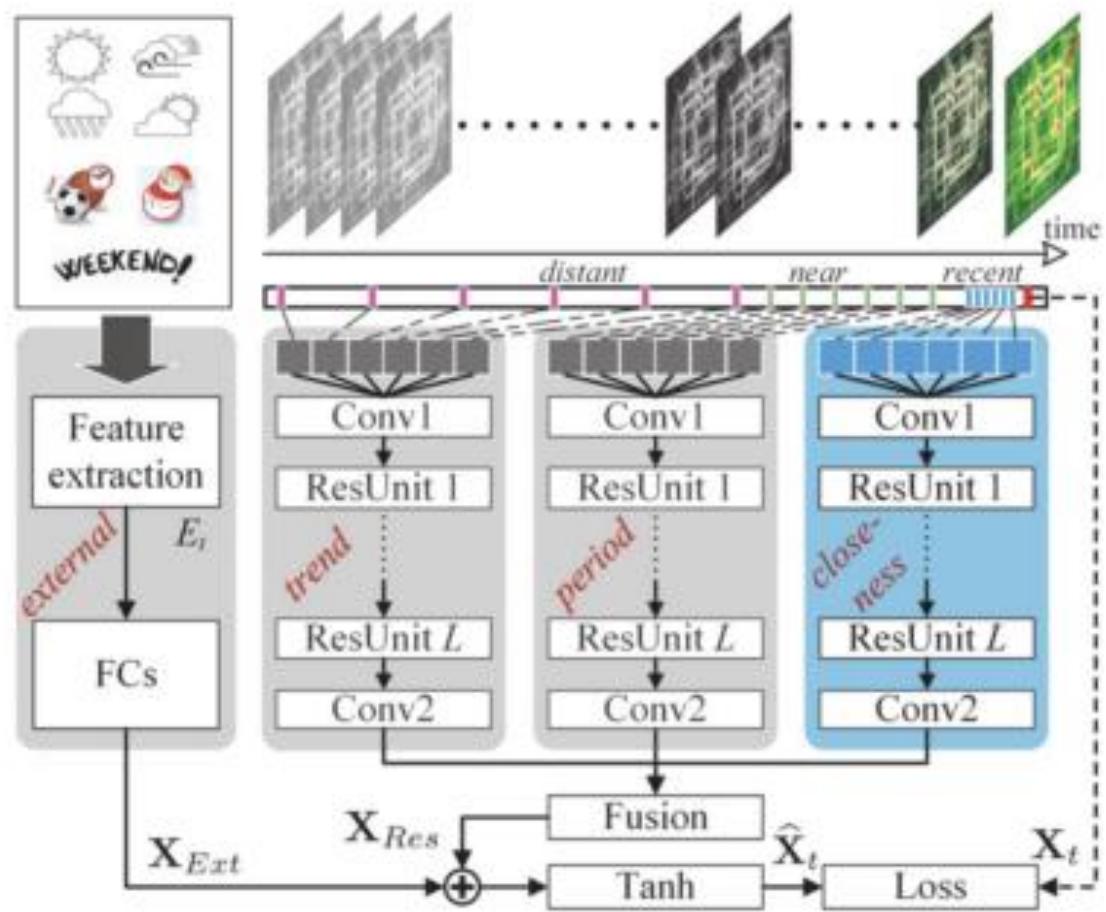
$$\begin{aligned} \text{obj}_{LinUOTD}(\mathbf{w}) = & \sum_{i=1}^N (y_i - p_i)^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2 \\ & + \gamma \sum_{X \subseteq D} \phi(X) \text{var}(\{\mathbf{w}' \mathbf{x} | \mathbf{x} \in X\}) \end{aligned}$$



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)

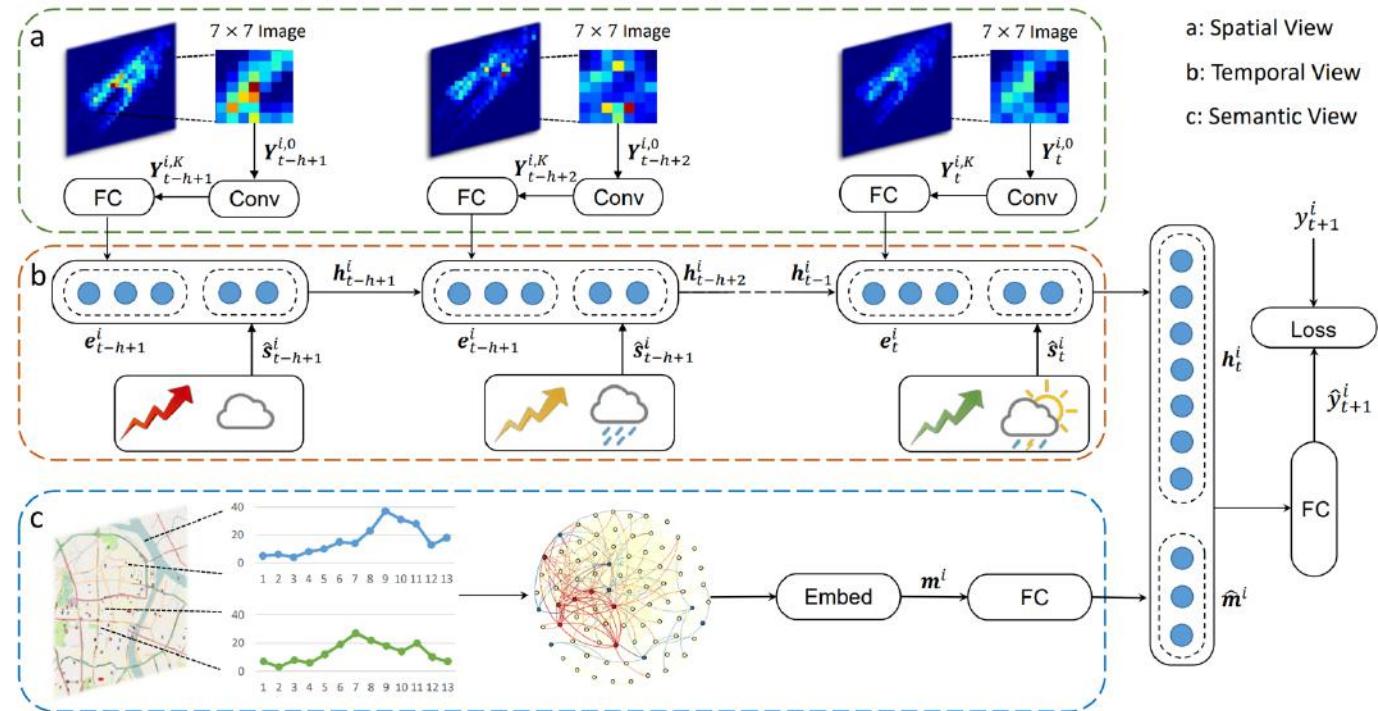


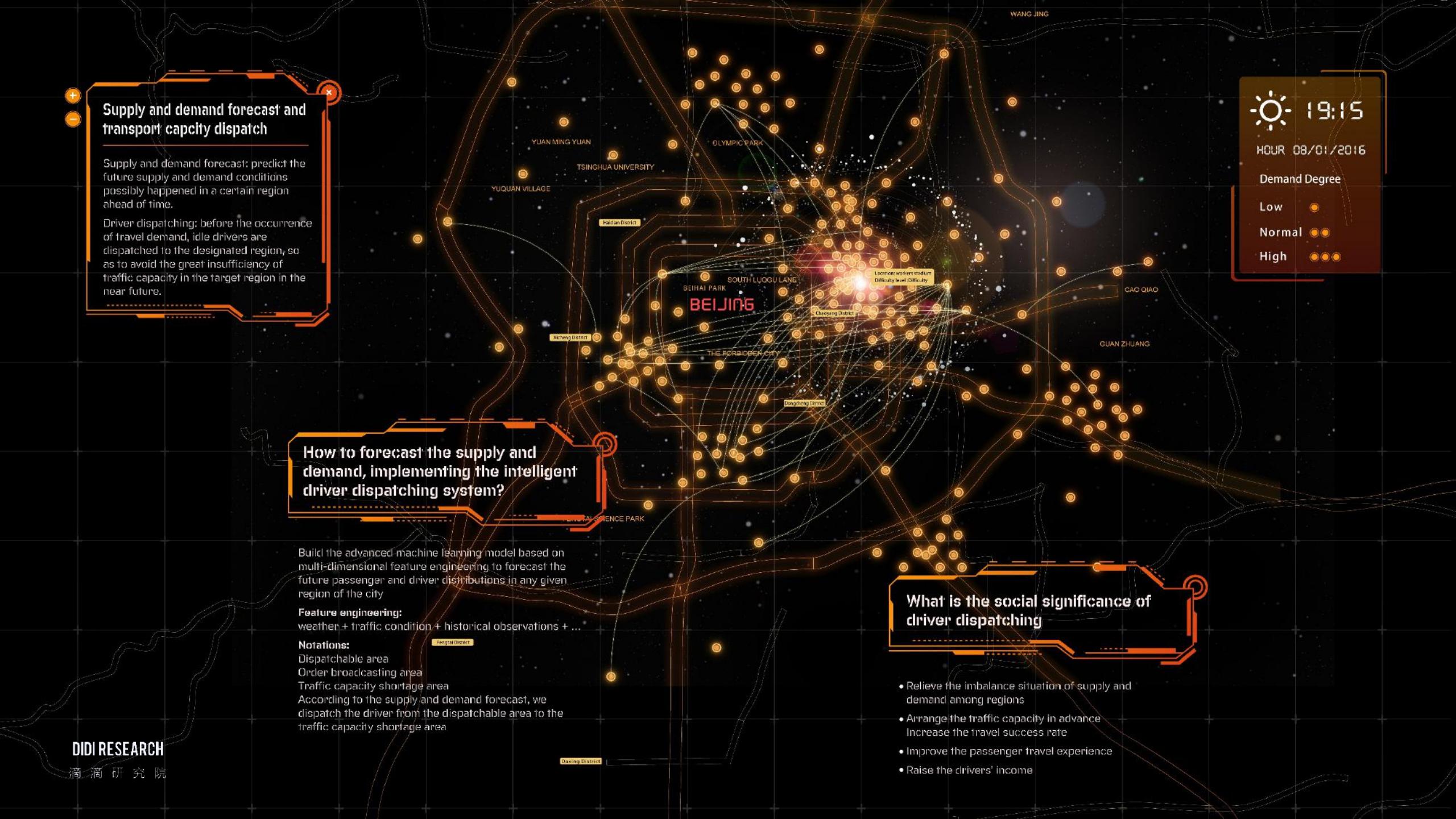
Zhang, et al, AAAI 2017

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

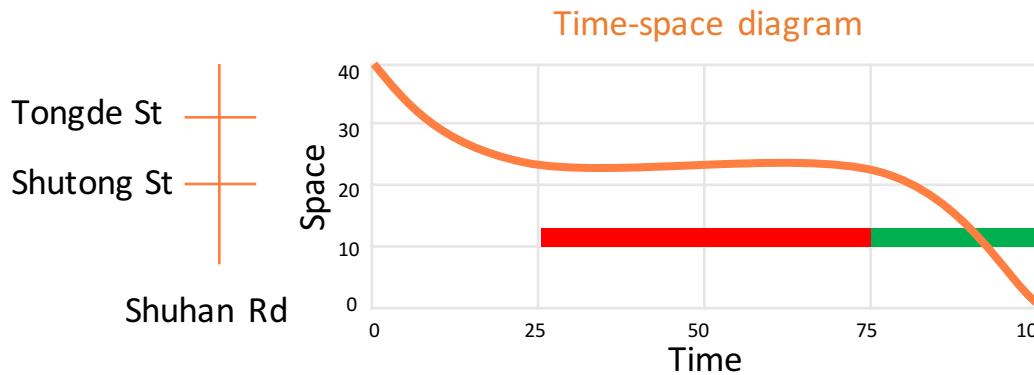
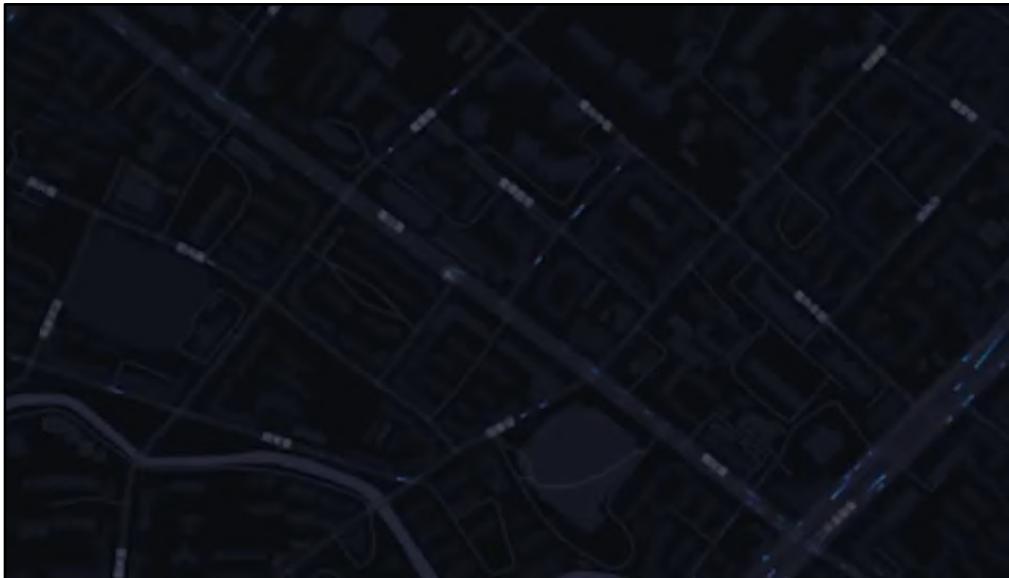






Smart Transportation

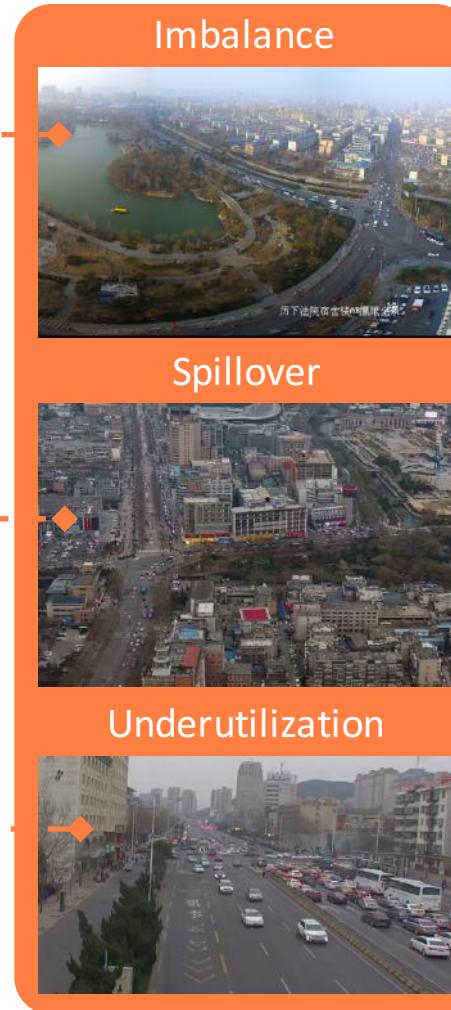
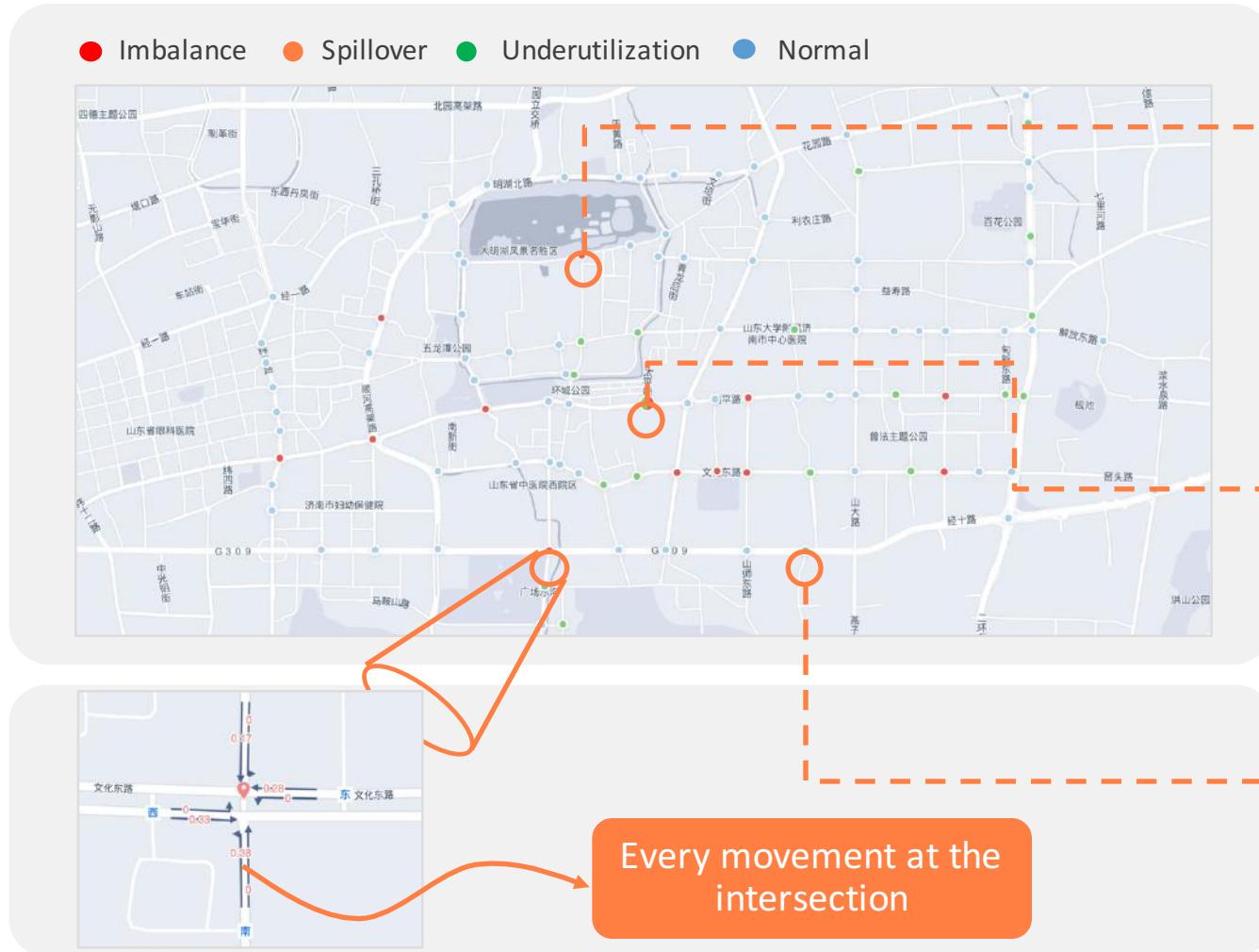
Data Collection, State Estimation, and Prediction



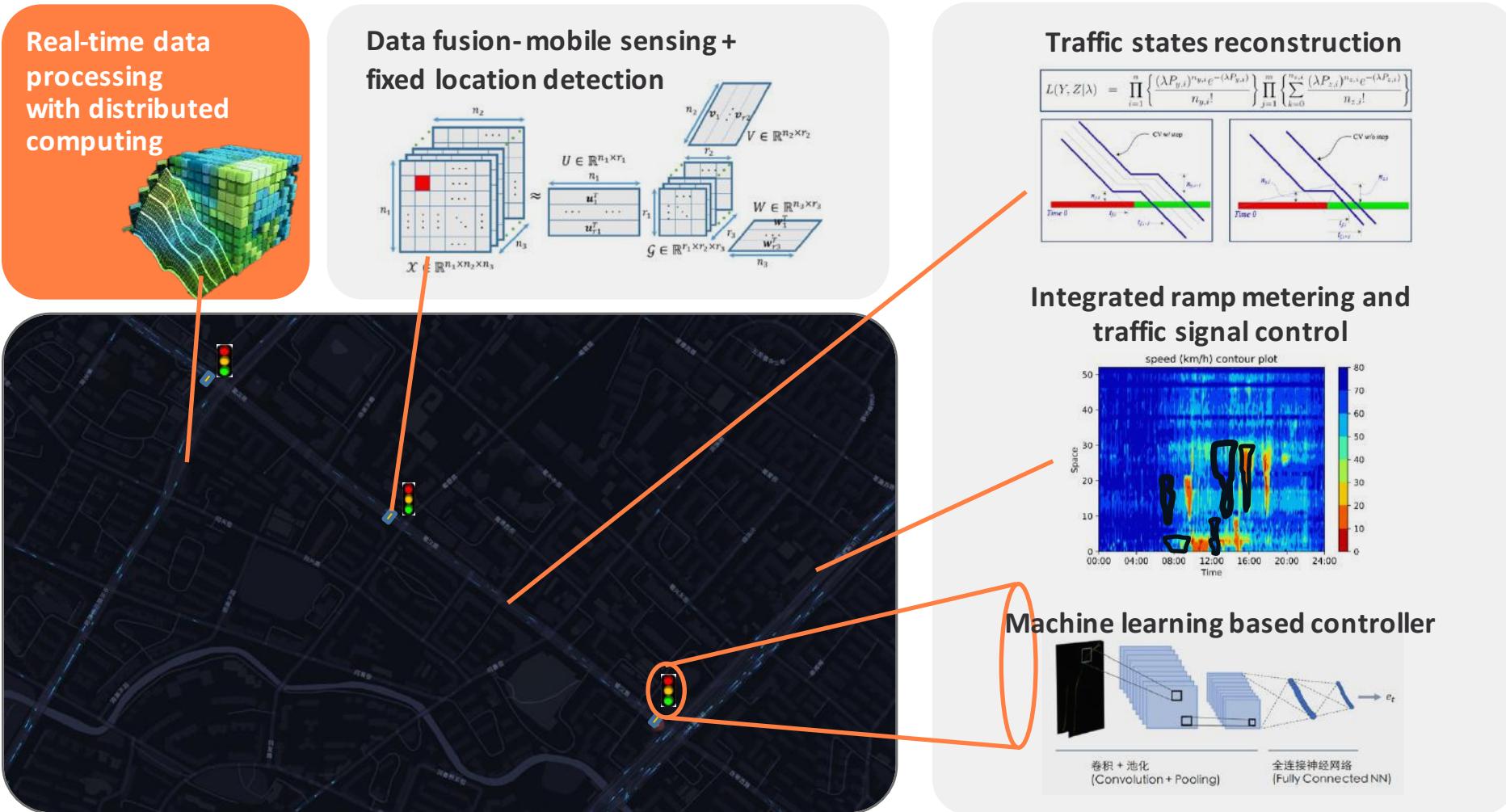
2018-4-18

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
.....		

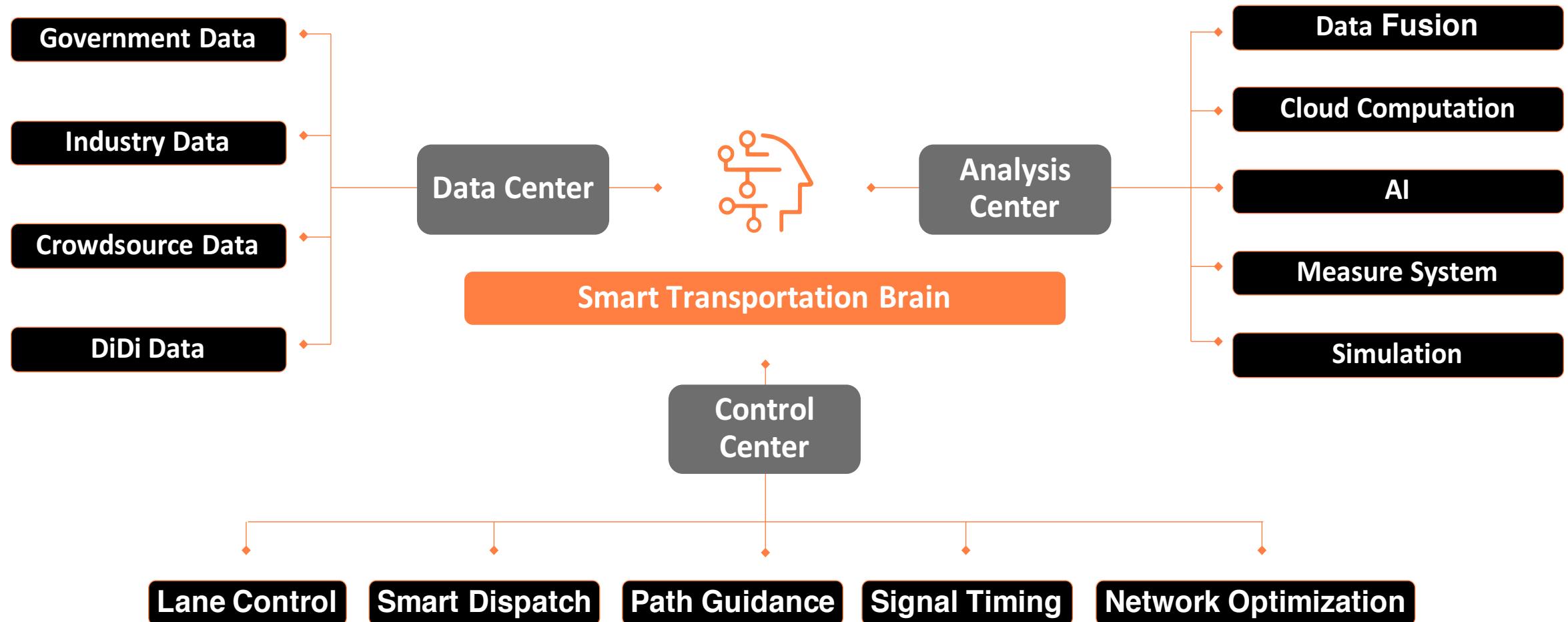
Performance Evaluation and Diagnosis



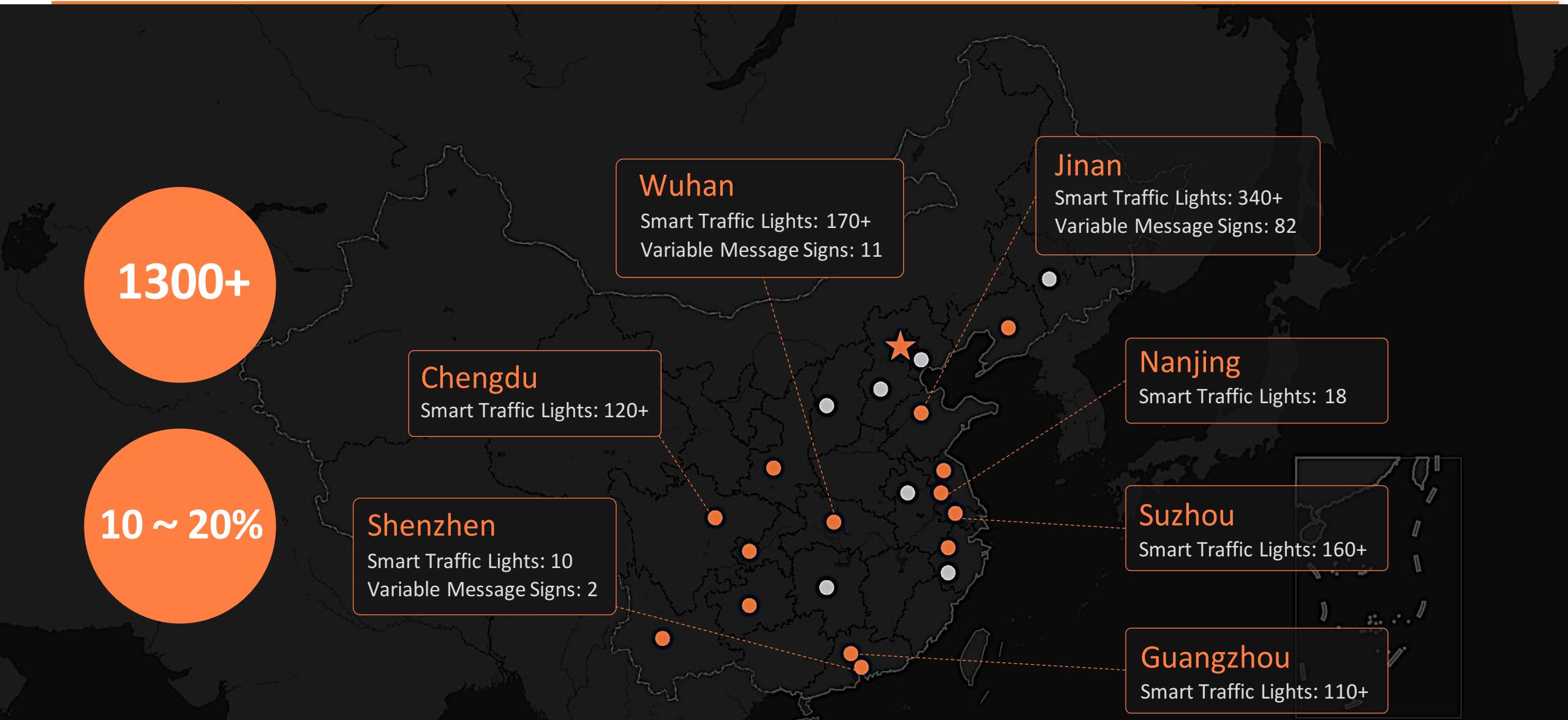
Real Time Traffic Control



Integrated Solution - Smart Transportation Brain



Improving traffic conditions in over 20 cities





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



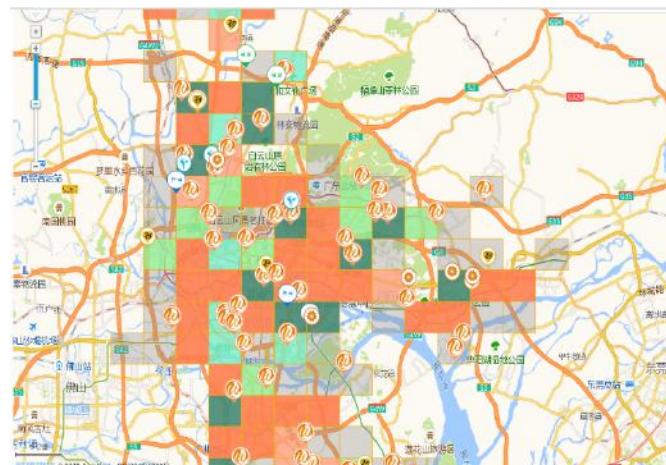
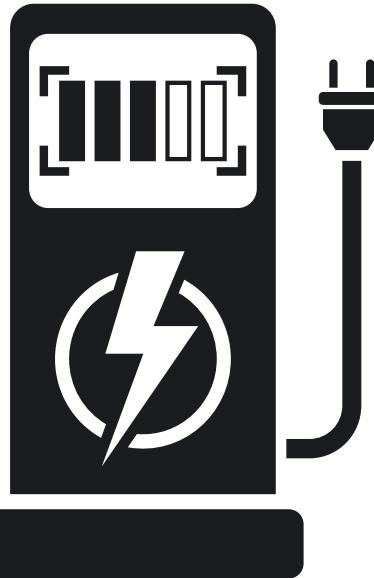
Demand Prediction

Station Capacities

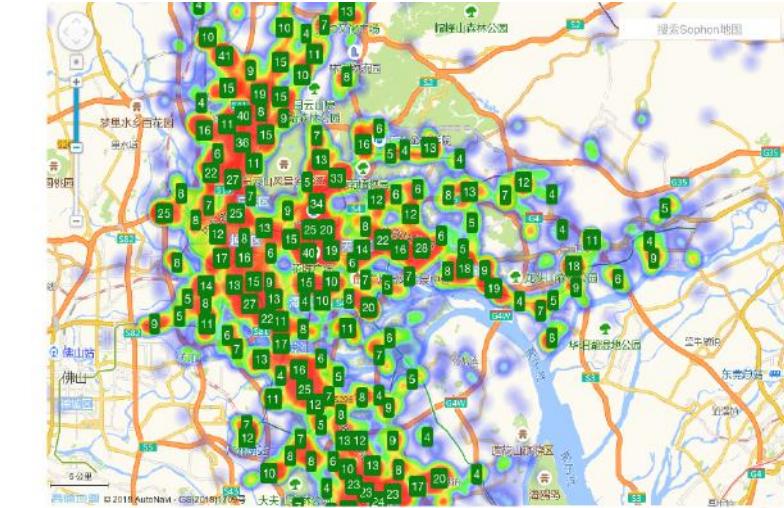


Product

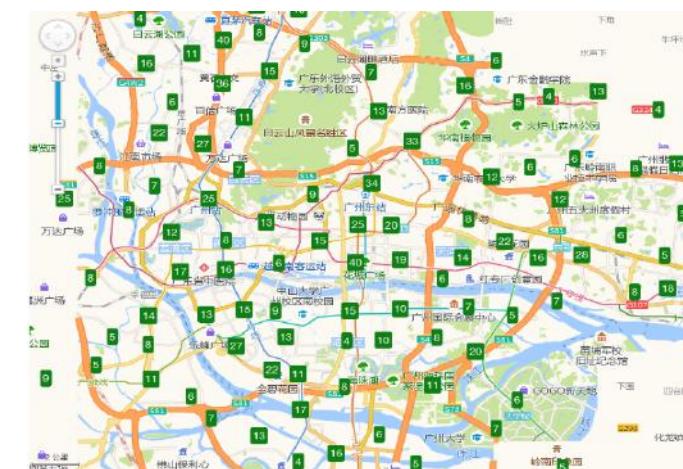
Interactive Map



Supply-demand

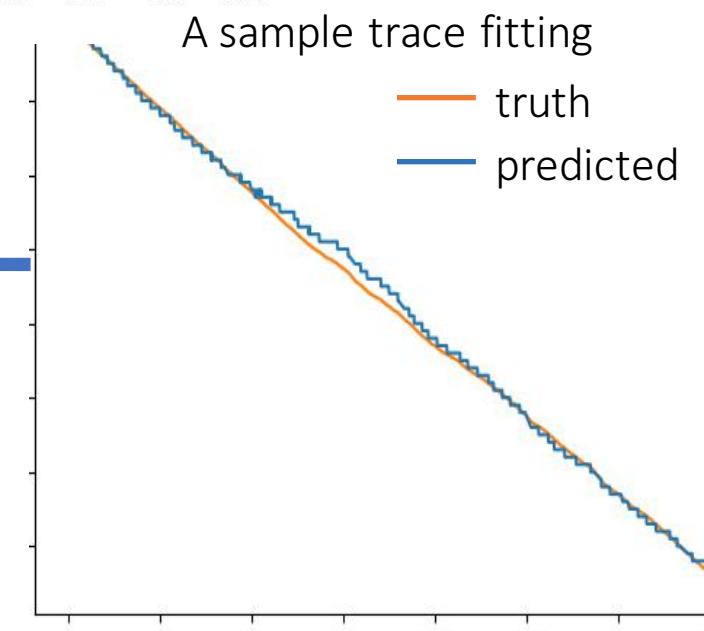
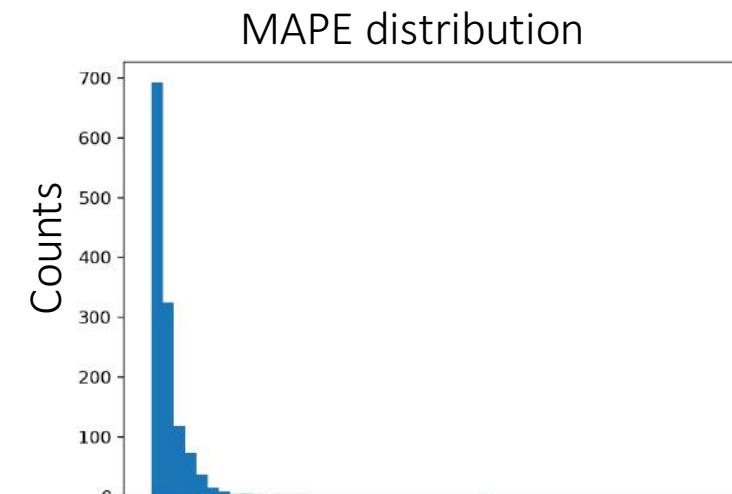
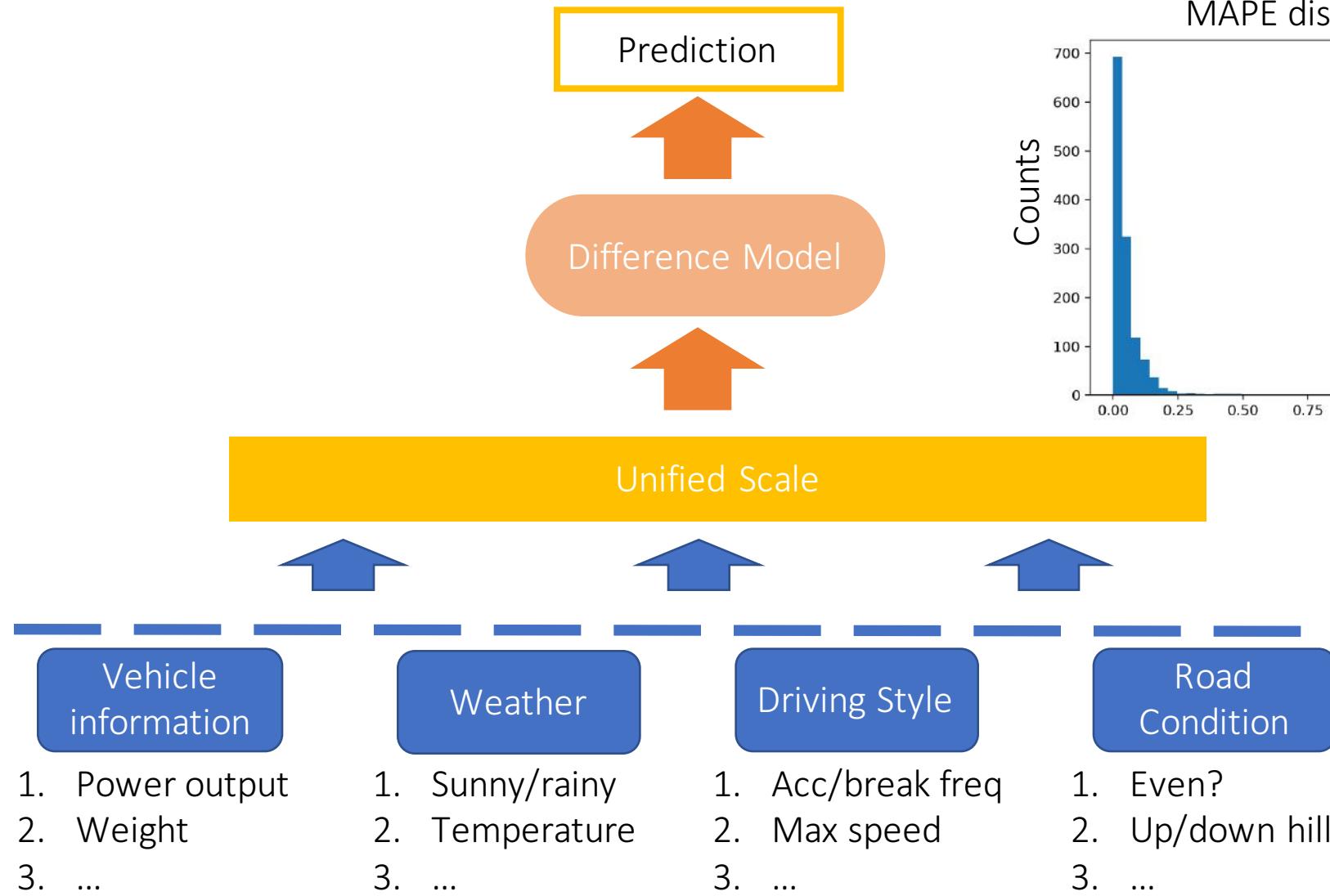


Potential Charging Stations



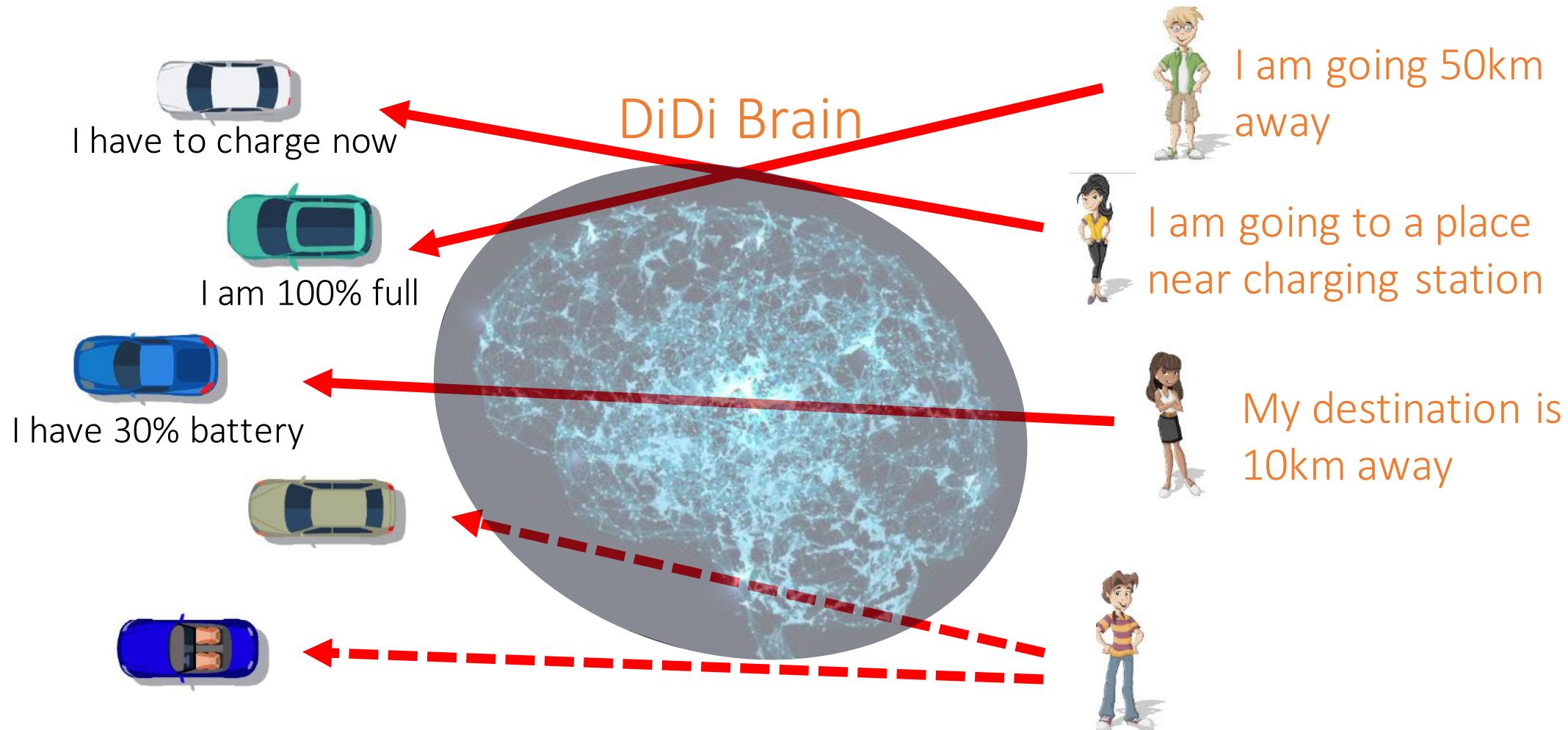
Piles needed

BMS (Consumption Prediction)



Dispatching Strategy

Adapting the unique characteristic of new energy vehicles





AI for Social Good



Deep Learning

Barrier-Free

Intelligent Medical Hardware

AI for Social Good

Disease Warning

Health

Machine Learning

Environmental Sustainability

Reinforcement Learning

Clustering

EV Charging Piles Location

Urban Air Quality Monitoring

Neural Networks

Call for Participant



Part 3: Data and Tools for Transportation AI

Open Dataset



KDD2017

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



Introduction



Open



Collaborative

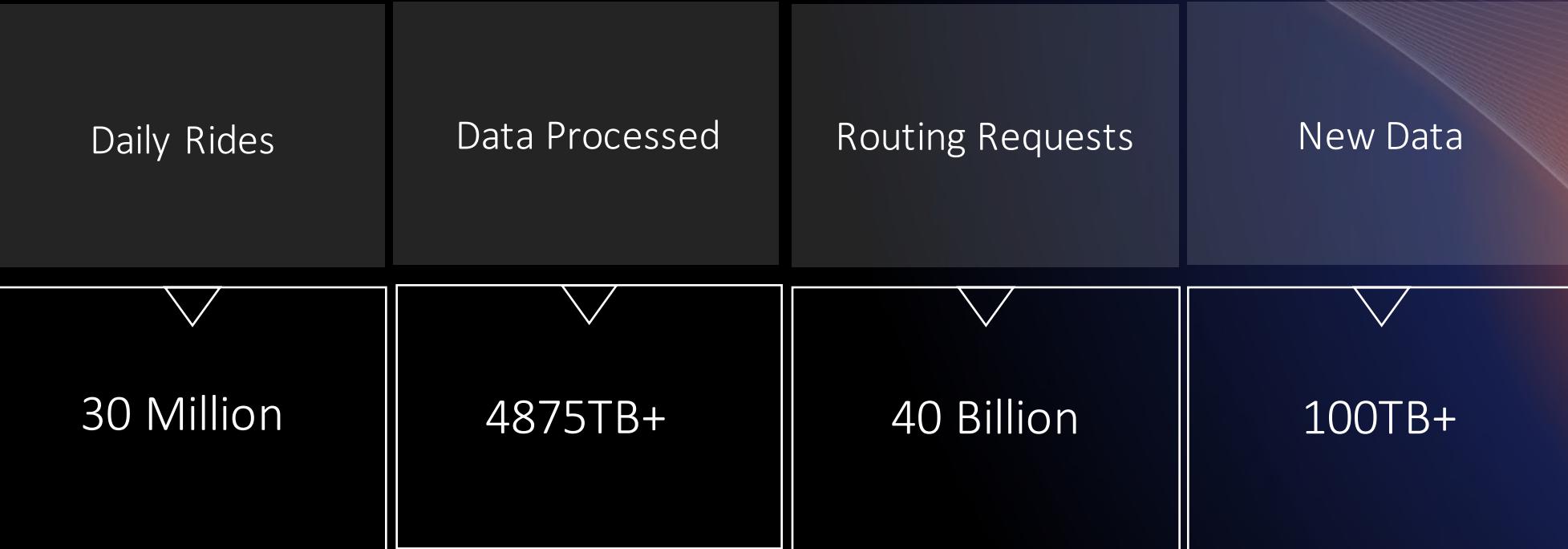


Innovative

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.

Big Data

Every Day



Sharing Anonymized Data

Vehicle Trajectory Data
From: Chengdu and Xi'an, China

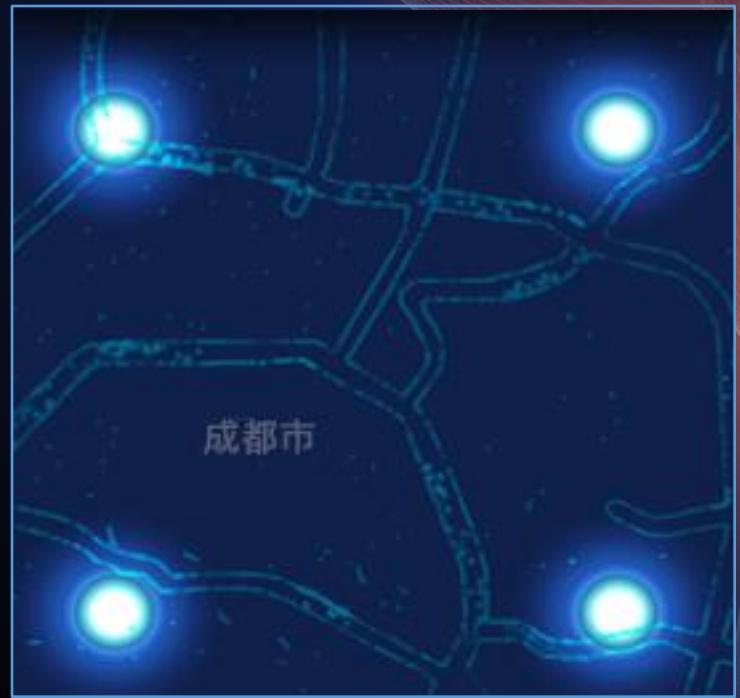
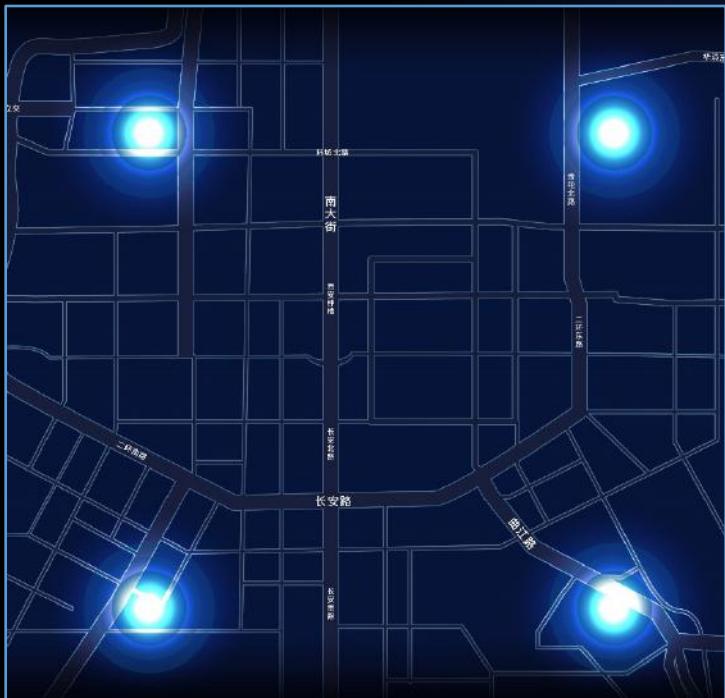
Real



High Accuracy



Full Sample



Process

01

Register

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



02

Verification & Review

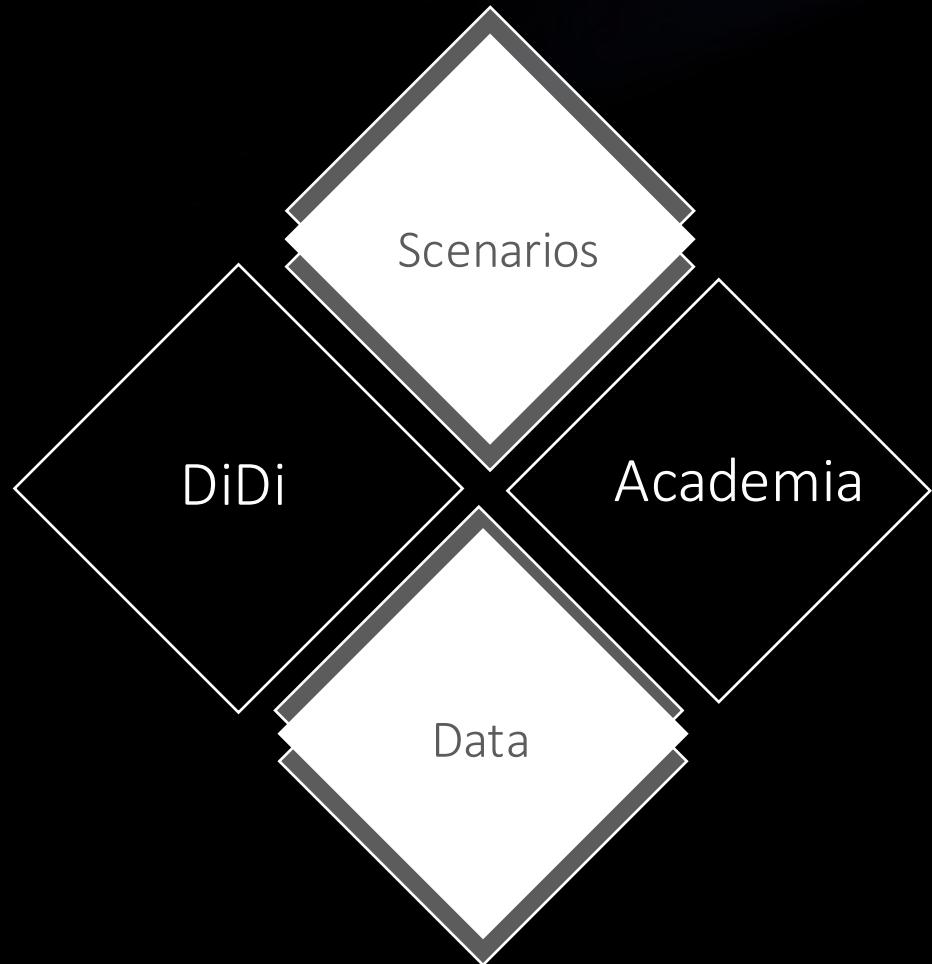
In the Data Center, click “**Data Access Application**”. Read the user agreement carefully and fill out the application form with your info, then submit the form for verification.

03

Access Data

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

Outreach



To Redefine the Future of Mobility



GAIA.didichuxing.com/en





GAIA Open Dataset
THANKS

MORE THAN A JOURNEY

