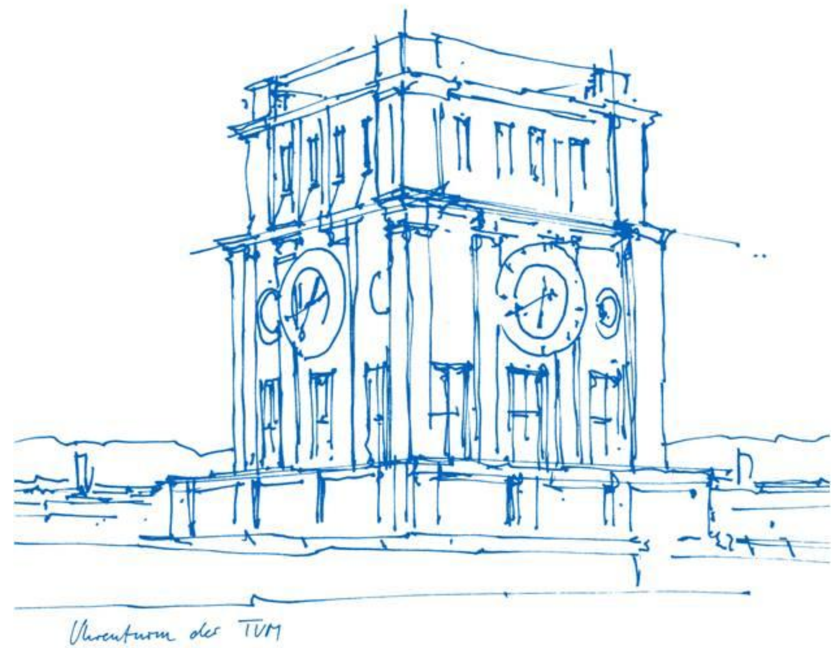
Project 5 - Mining Public Transport Data

Final presentation

Students: Christopher Lang, Chengyu Sheu, Georg Aures, Jens Petit

Large Scale Machine Learning Lab
Department of Informatics
Technical University of Munich
Munich, 04. July 2018

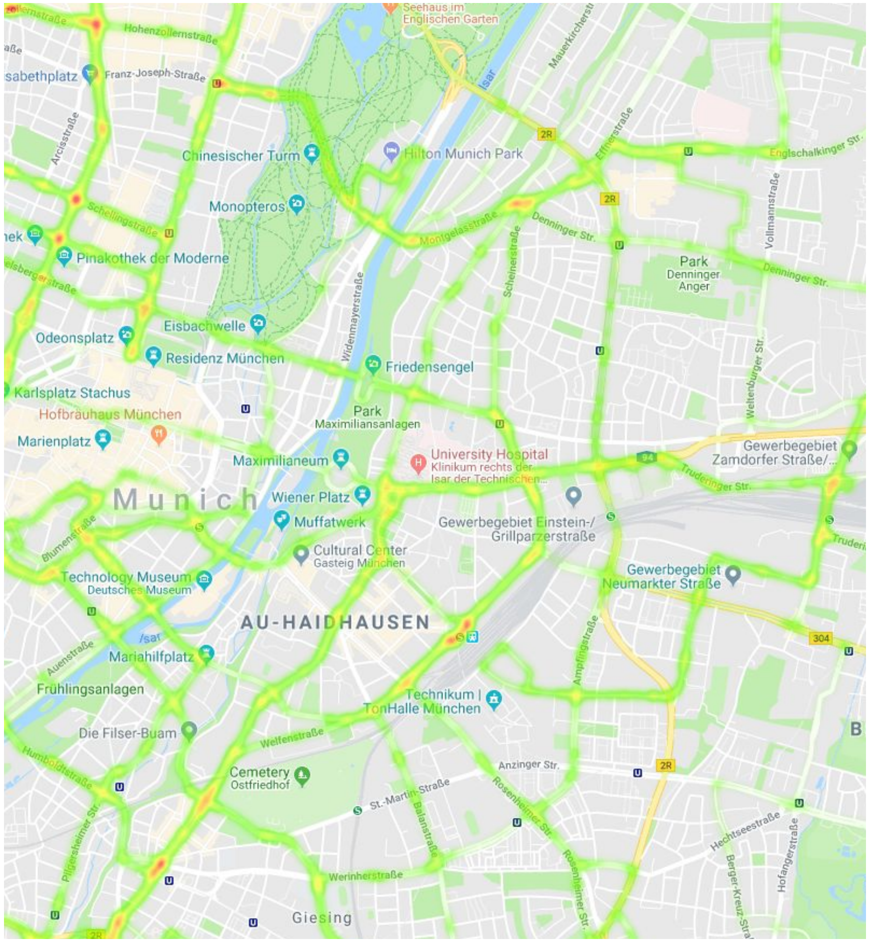


Agenda

1. Problem Statement
2. Understanding the Data
3. Creating the Graph
4. Using the Graph
5. Discussion

Problem Statement

- Track vehicles in a public transport network
- Describe actual behavior from log-data
- Capture deviation from pattern



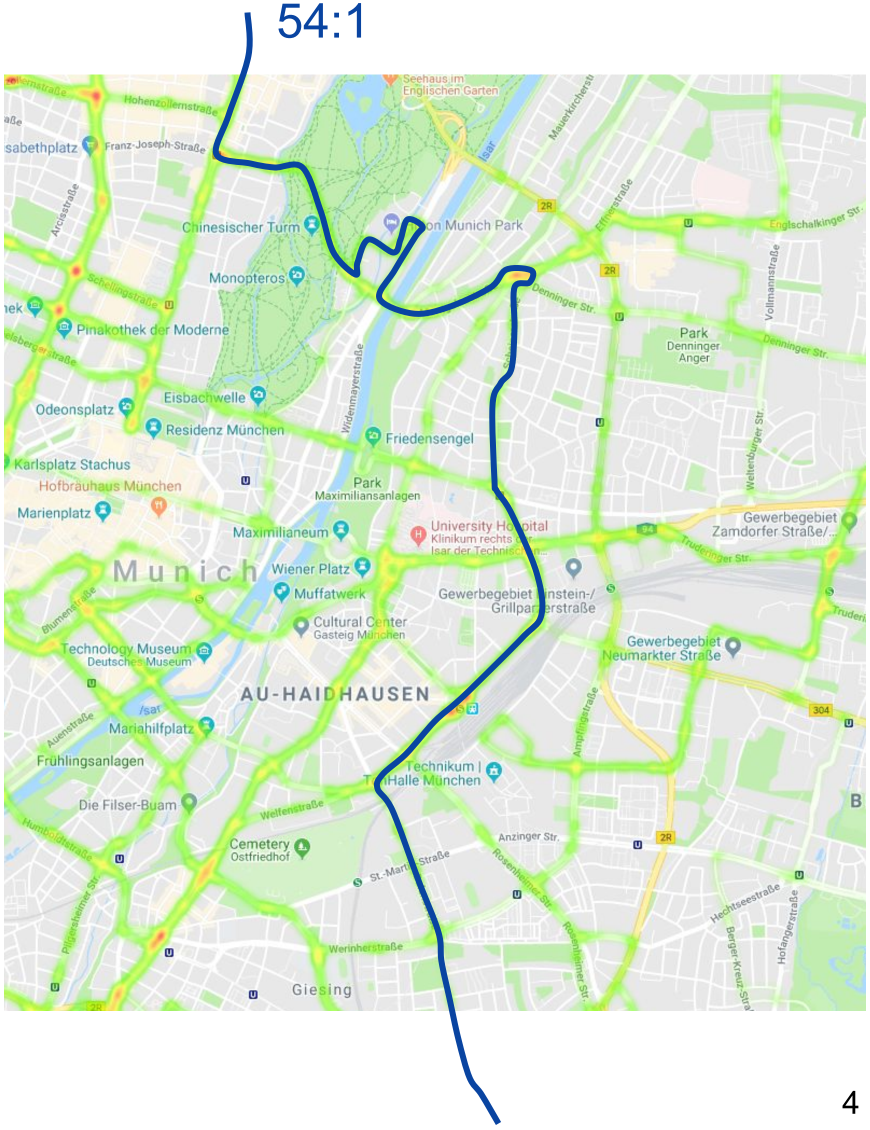
Problem Statement

Track vehicles in a public transport network

Describe actual behavior from log-data

Capture deviation from pattern

54:1

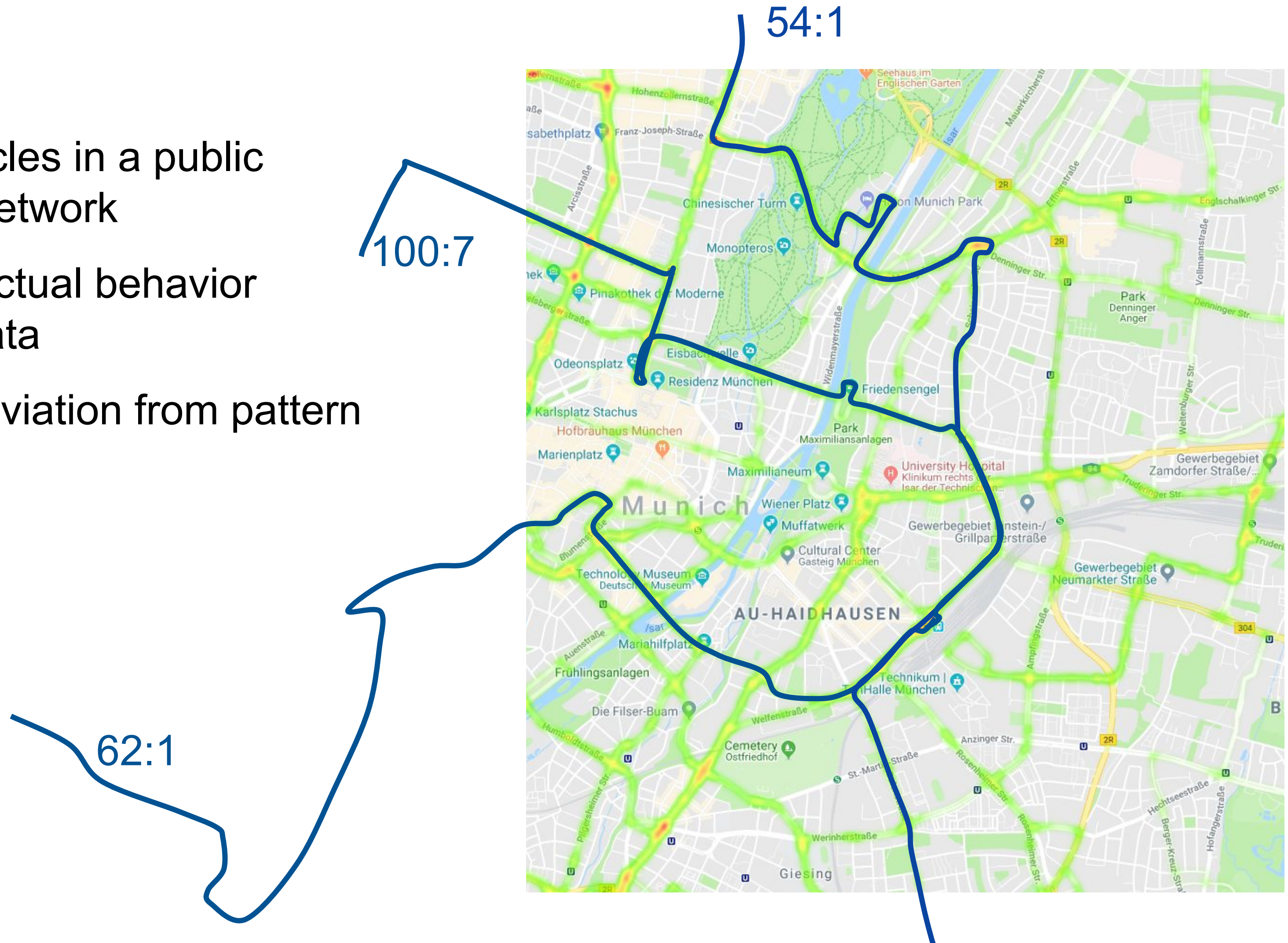


Problem Statement

Track vehicles in a public transport network

Describe actual behavior from log-data

Capture deviation from pattern



Log Data

Labeled,
sparse
time series

$$\ldots, x_{i-1}, x_i, x_{i+1}, \ldots$$

$$\text{with } x_i = \begin{pmatrix} timestamp_i \\ GPS_i \\ odometry_i \\ door\_event_i \end{pmatrix}$$

Log Data

labeled,
sparse
time series

$$\ldots, x_{i-1}, x_i, x_{i+1}, \ldots$$

$$\text{with } x_i = \begin{pmatrix} timestamp_i \\ GPS_i \\ odometry_i \\ door\_event_i \end{pmatrix}$$

Log Data

labeled,
sparse
time series

$$\ldots, x_{i-1}, x_i, x_{i+1}, \ldots$$

$$\text{with } x_i = \begin{pmatrix} timestamp_i \\ GPS_i \\ odometry_i \\ door\_event_i \end{pmatrix}$$

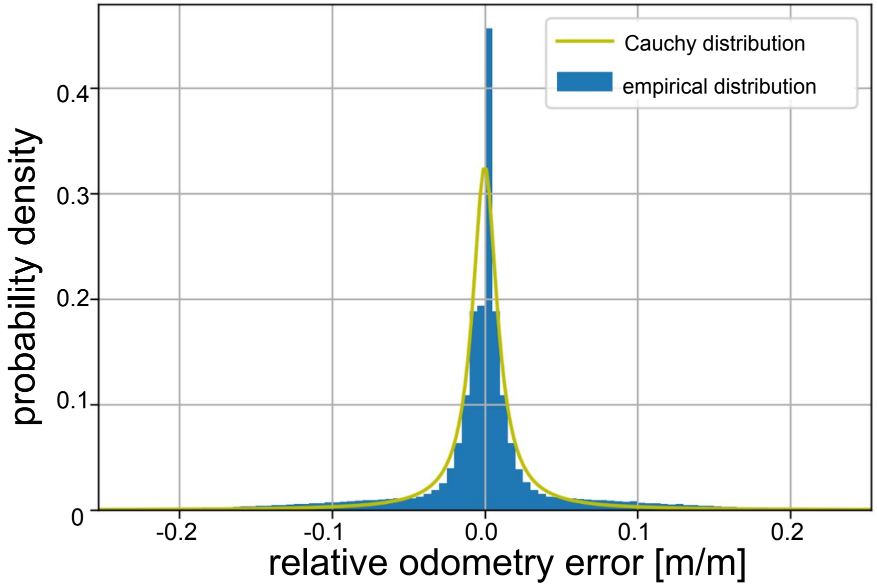
Difficulties in the Data

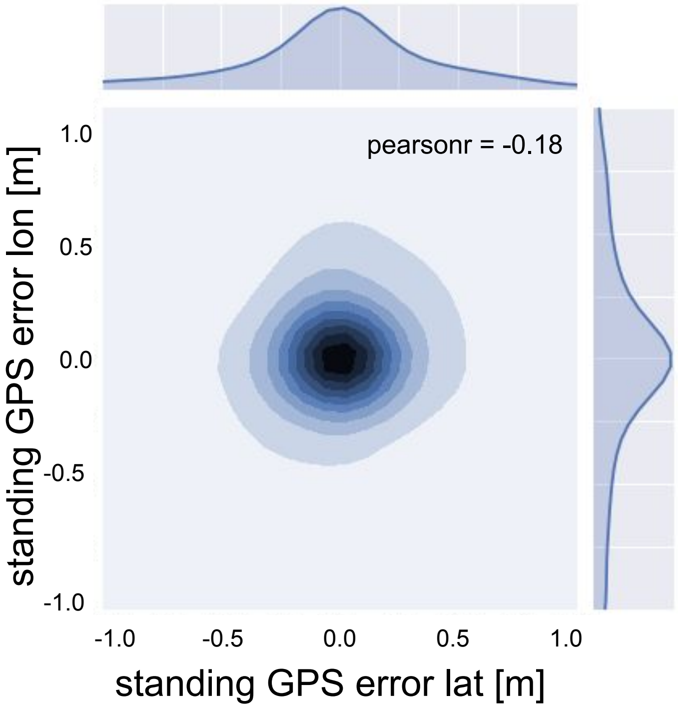
Noise in odometry, GPS, door-events

No heading information

Low sampling rate

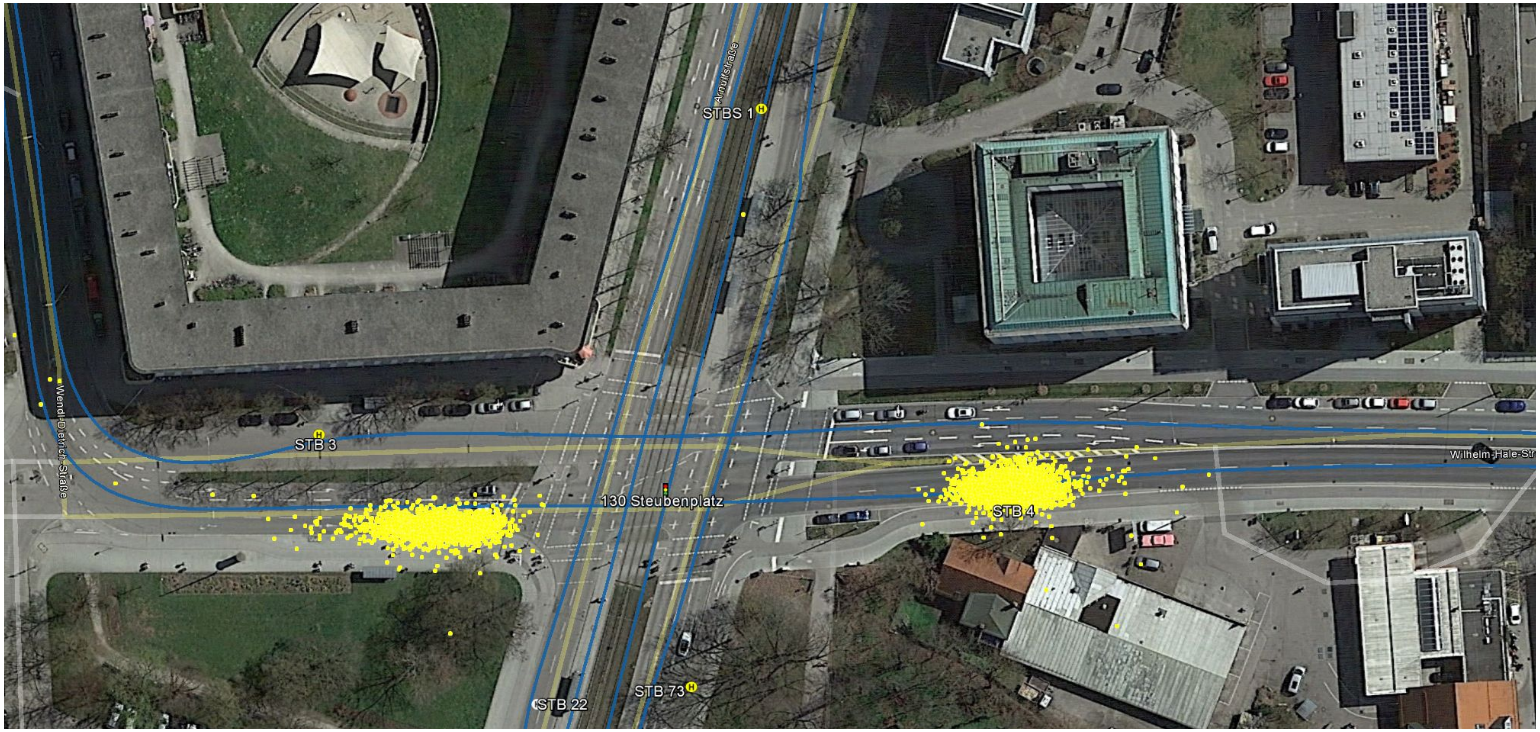
Labeling noise

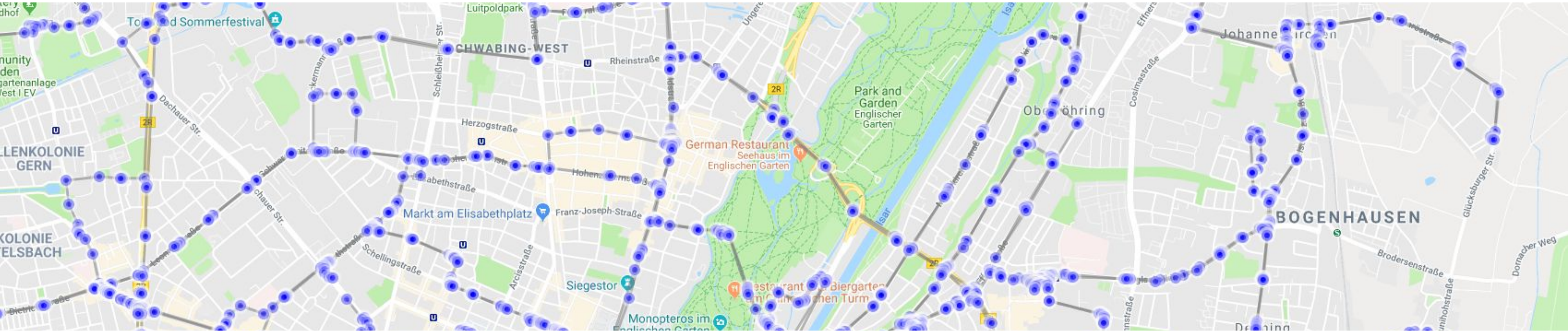




Goal

Graph to generalize route and bay structure
Data driven to detect change





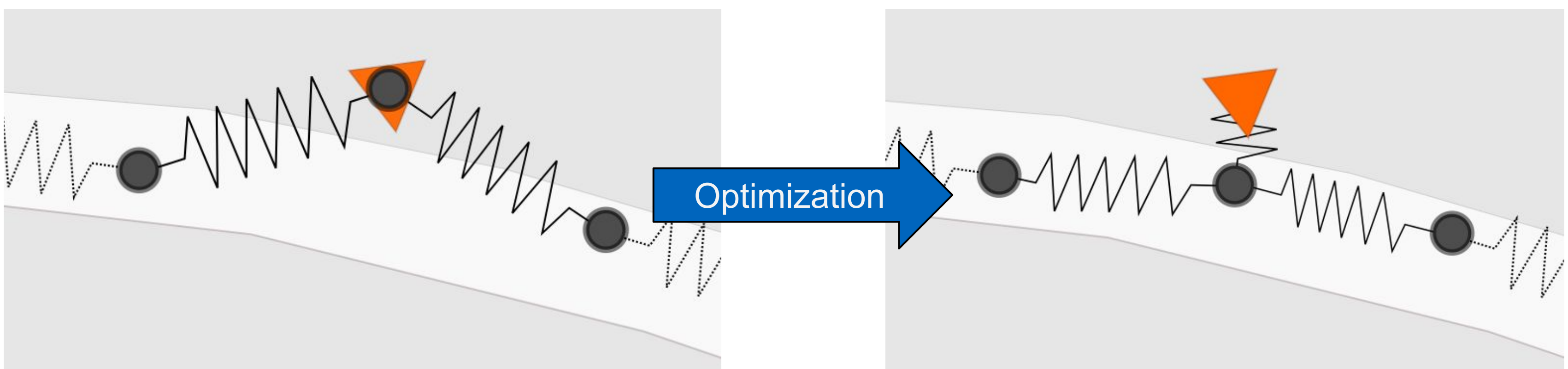
Creating the Graph

Single Trajectory Denoising

Approach: make odometry and GPS consistent

Minimize using gradient descent:

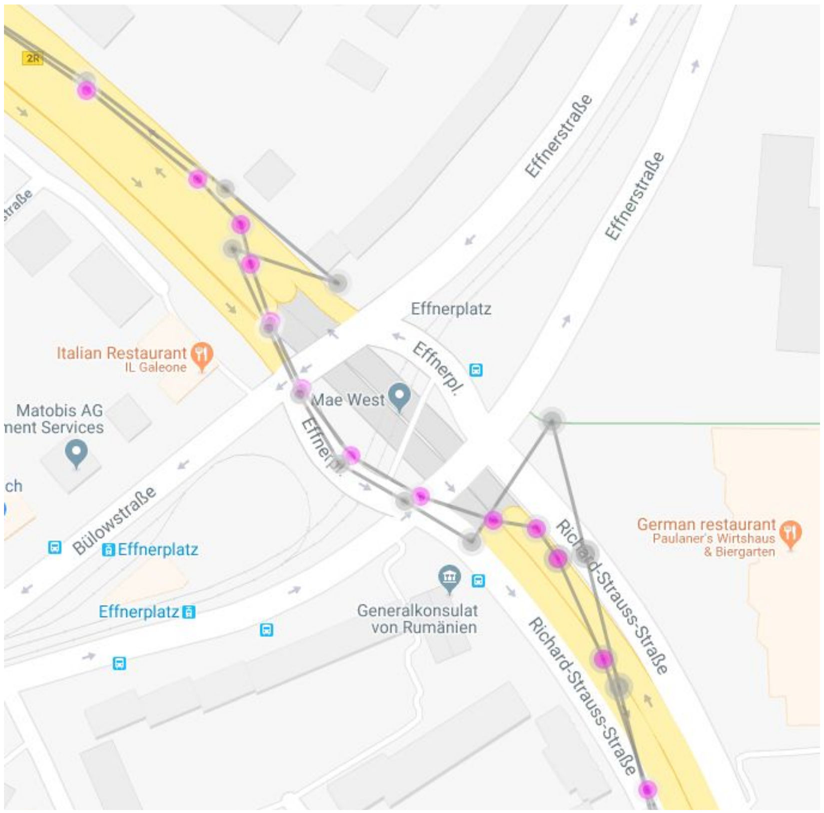
$$J(\mathbf{x}) = \sum_{i}^{M} \frac{1}{c(z_i)^2\sigma^2} \underbrace{\|\mathbf{x_i} - \mathbf{z_i}\|_2^2}_{\text{GPS error}} + \frac{\lambda}{1+u_{ij}} \underbrace{(u_{ij}^2 - \|\mathbf{x}_j - \mathbf{x}_i\|_2^2)^2}_{\text{odometry error}}, \quad (i,j) \in E$$

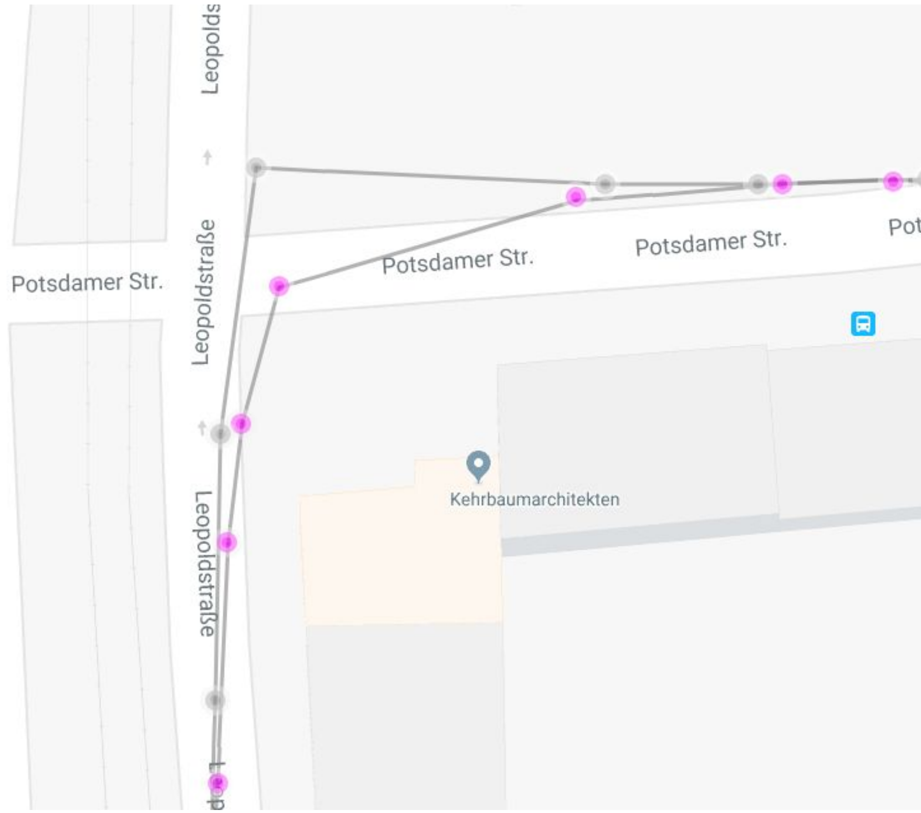


Optimization Results

Before denoising

After denoising





Applying a Window Averager

Move node to mean of neighbors in radius 15 meters
Heading information as filtering criteria

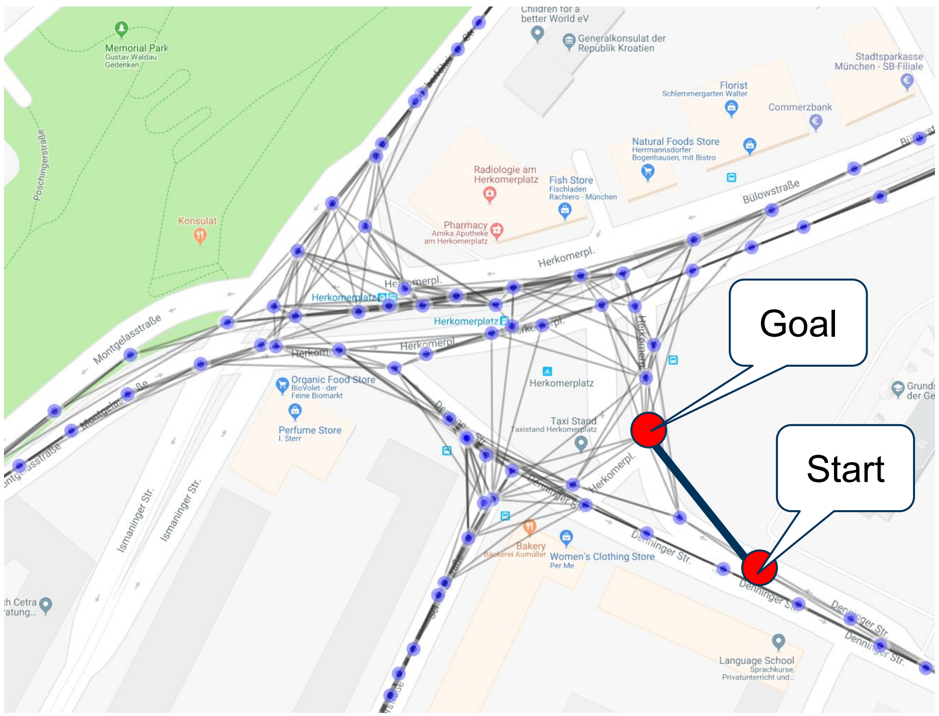
[1] D. Guo, S. Liu, H. Jin, *“A graph-based approach to vehicle trajectory analysis”*, Journal of Location Based Services

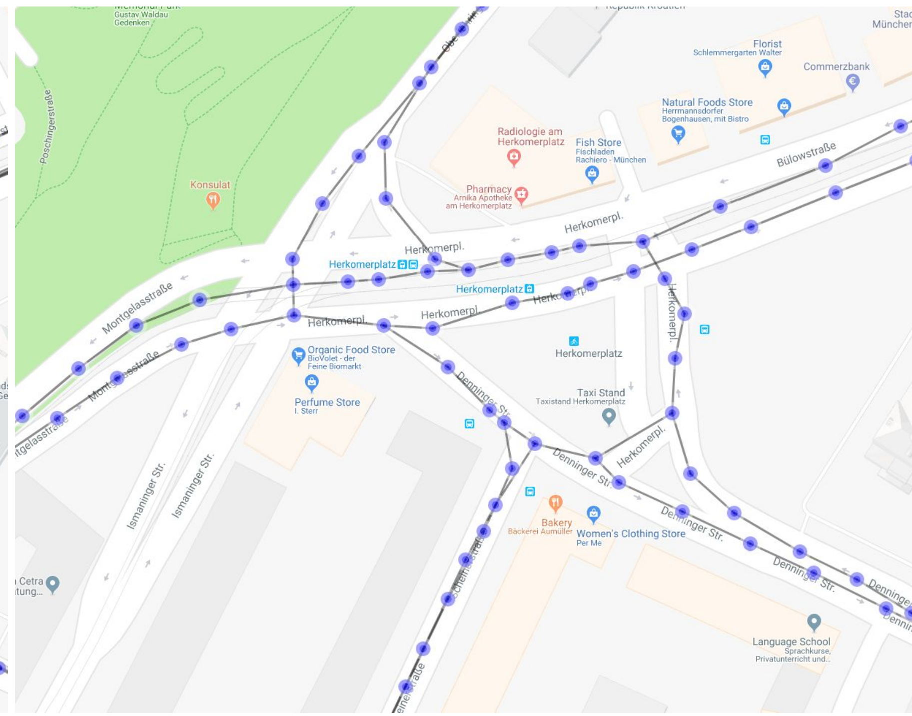
Finding Representative Centroids

Choose node and assign neighbors to centroid
Continue with nearest unrepresented node

Connecting the Centroids

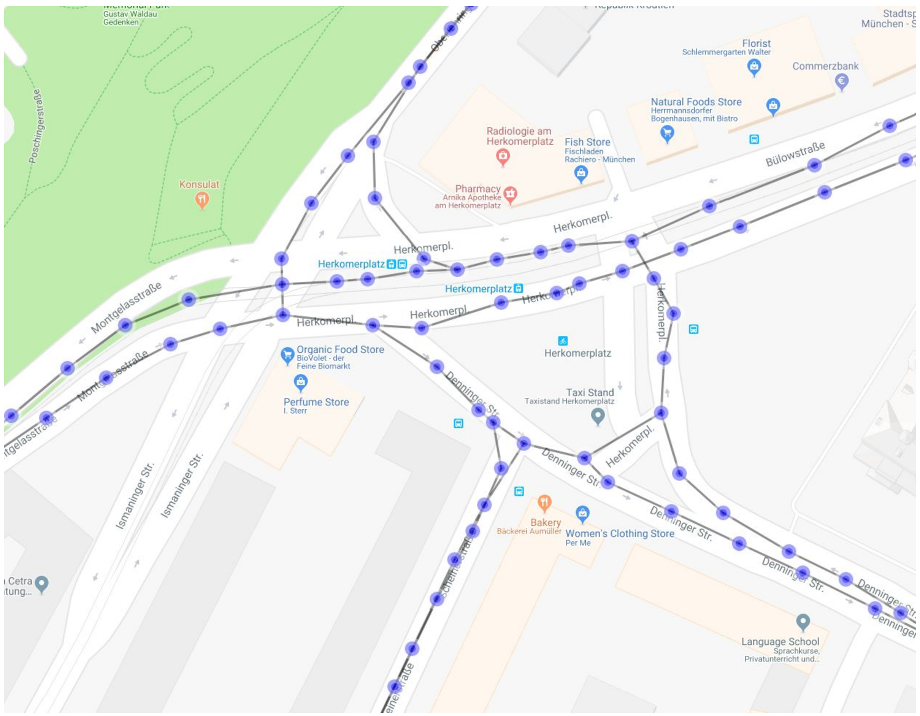
- Connect centroids according to original vehicle trajectories
- Apply shortest distance search on intermediate edges
  - Use $w_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^{\boldsymbol{\alpha}}, \quad (i,j) \in E$ as edge distance





Visual Graph Evaluation

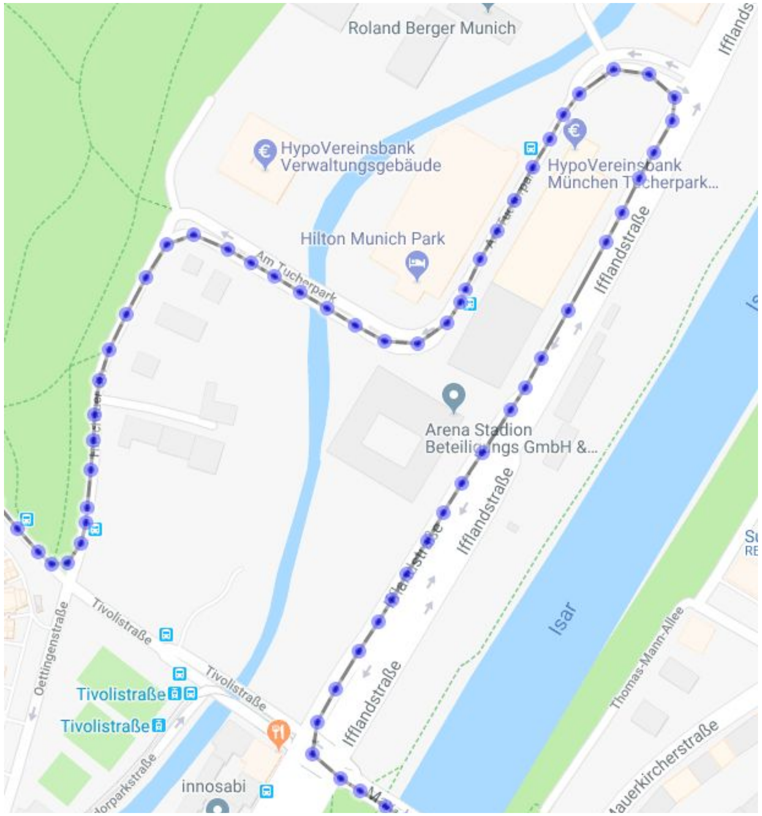
Data-driven graph

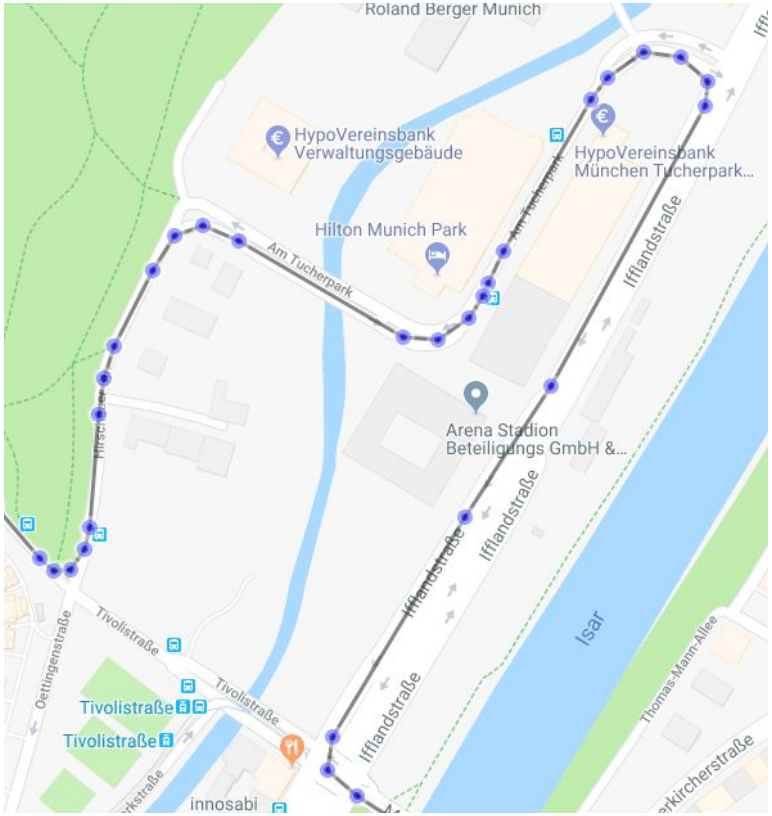


Reference SWM graph

Graph Sparsification

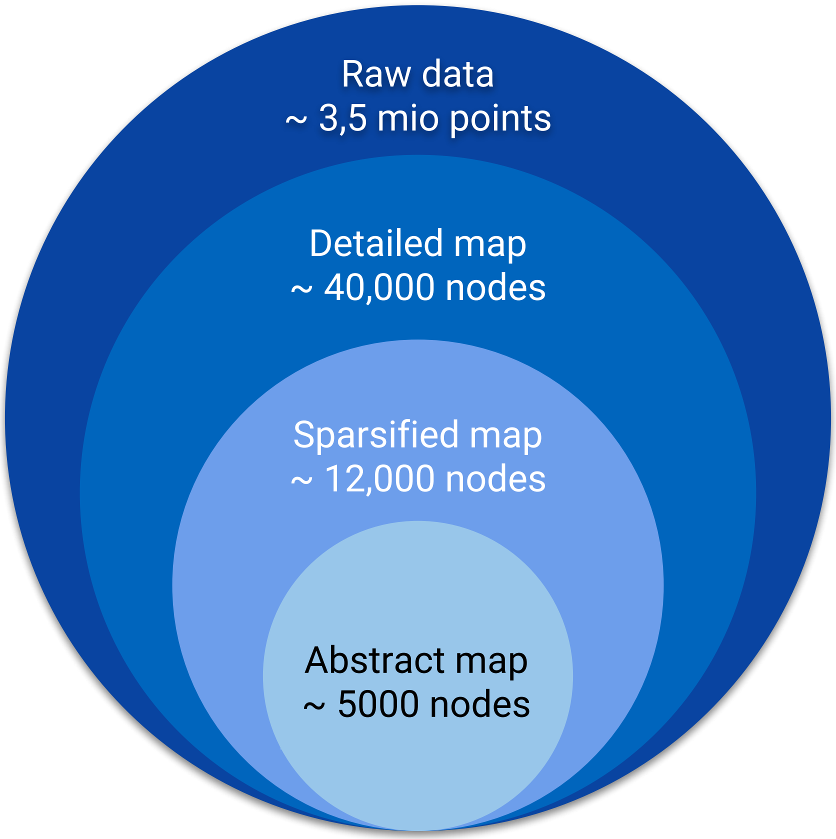
Drop nodes on near straight line sequences





Graph Pipeline

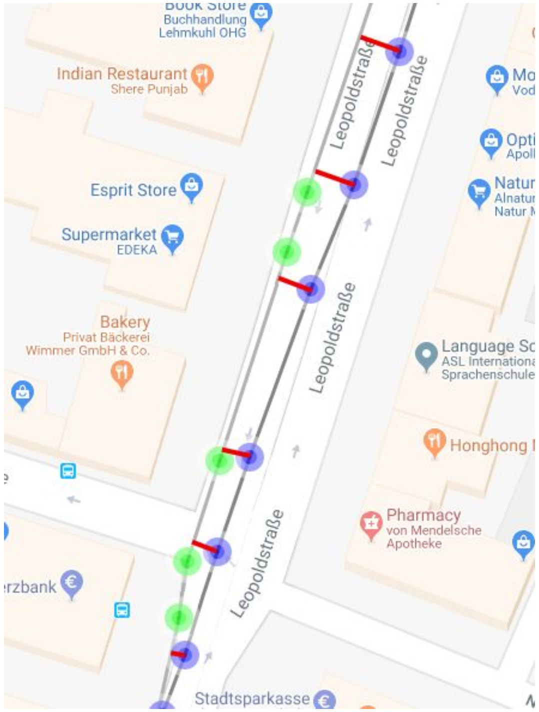
One day data

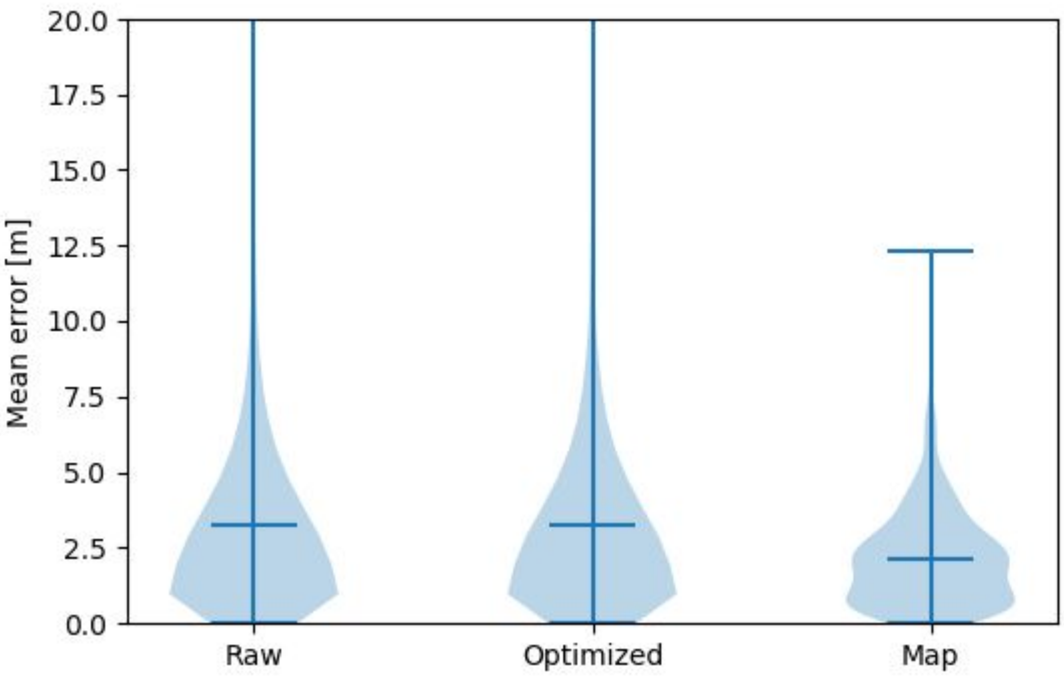


Statistical Graph Evaluation

Evaluation with SWM hand-crafted graph

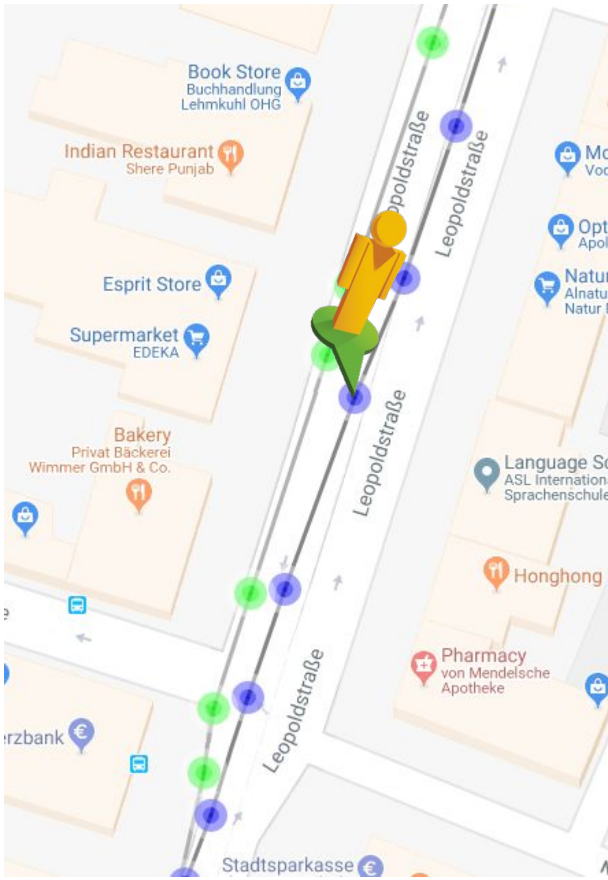
Mean error distance of projection onto evaluation graph

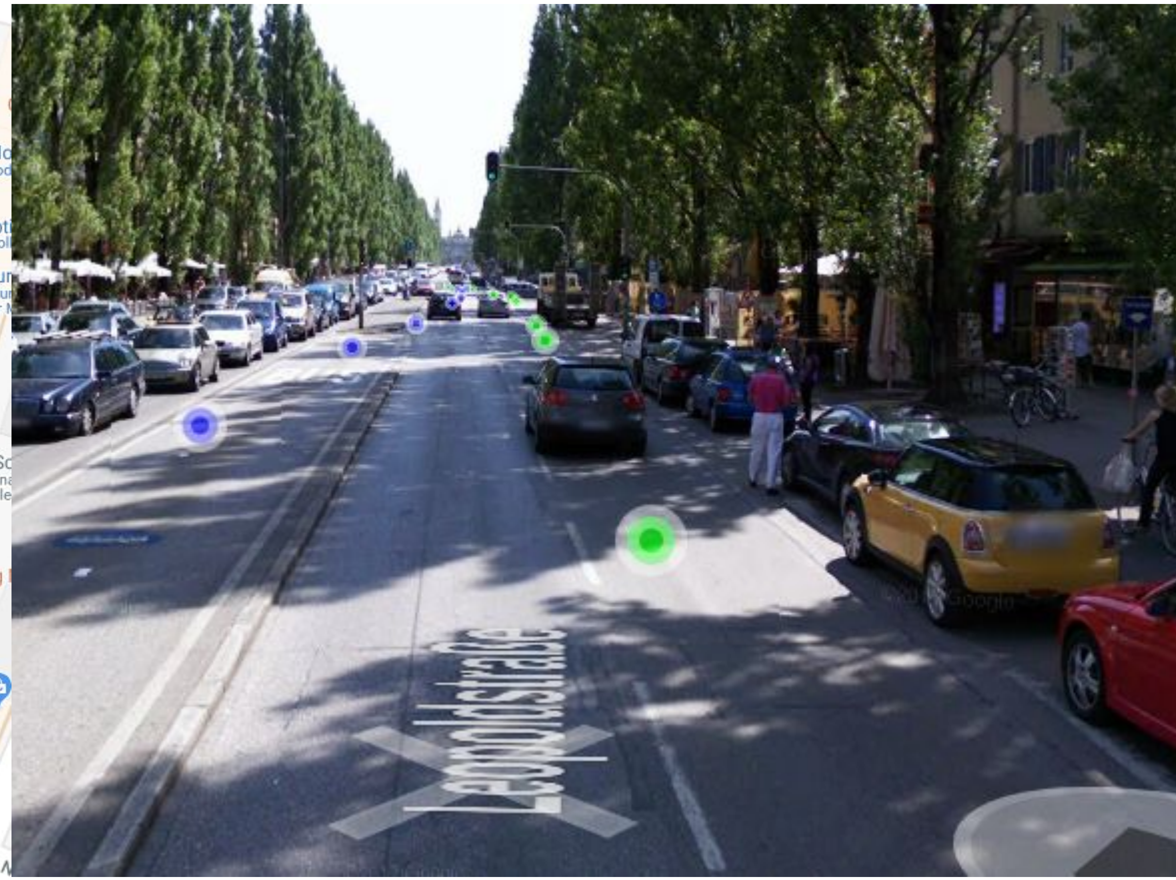




Visual Map Evaluation

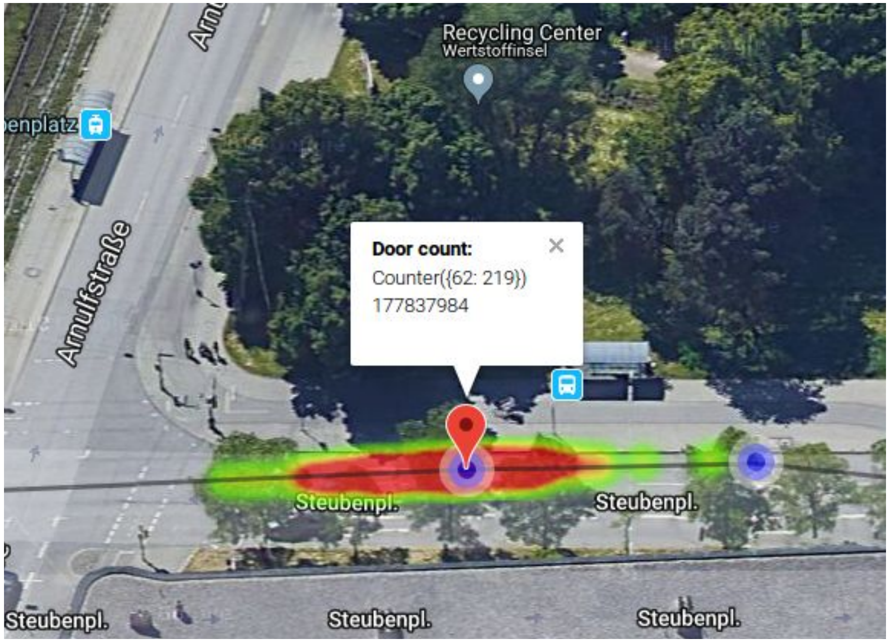
Graph captures bus lanes

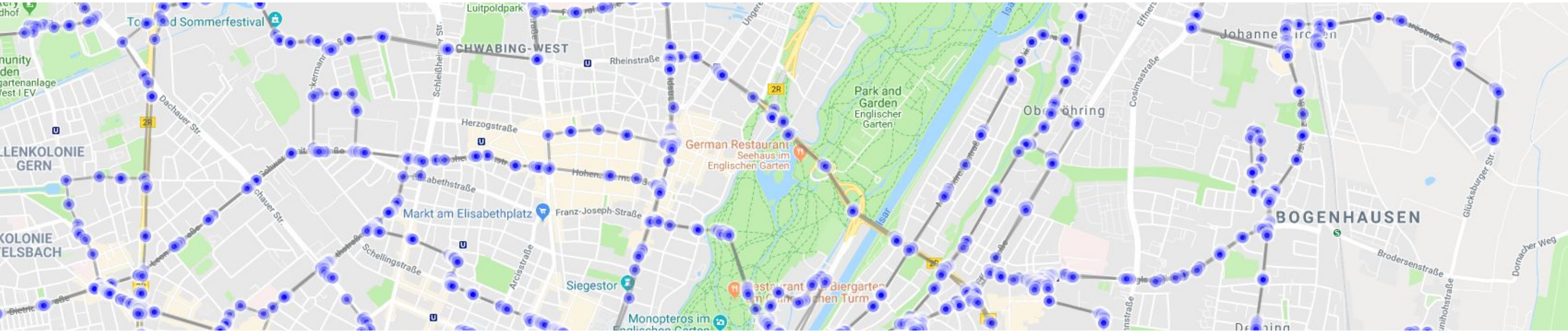




Additionally Encoded Information

Time and odometry distance on edges
Door events with counter and distribution

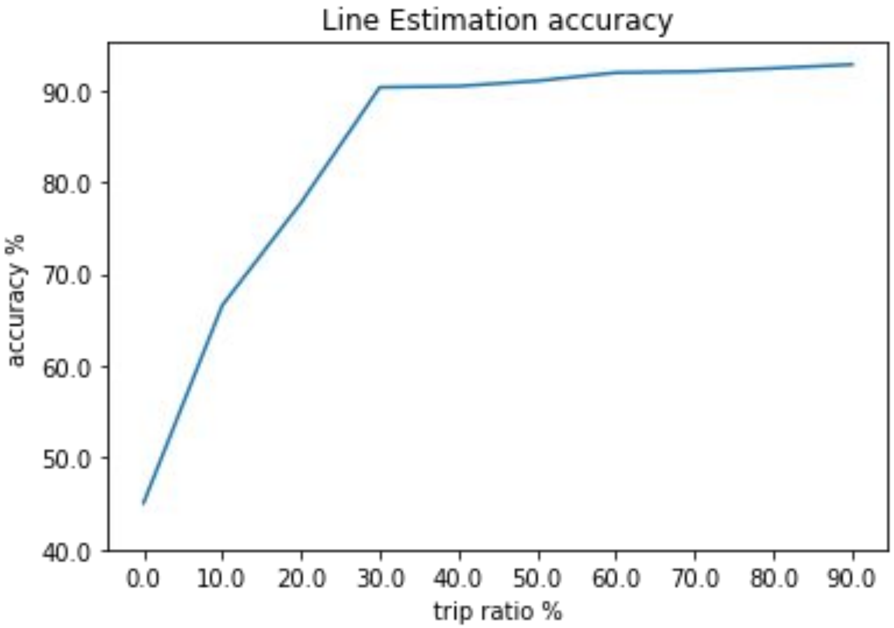


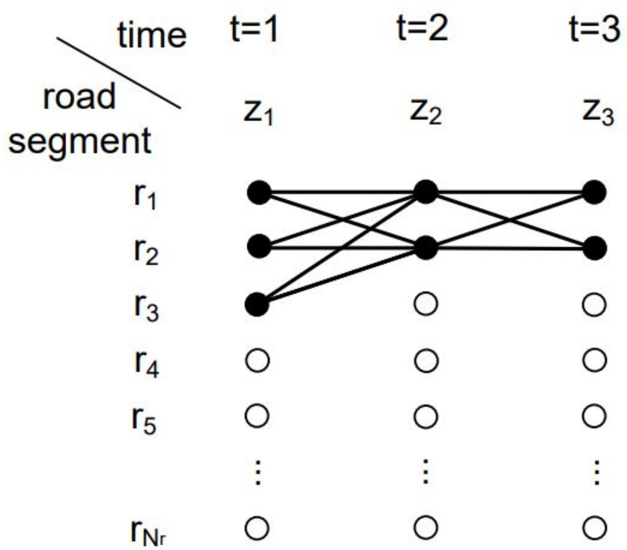


Using the graph

Hidden Markov Model with Line Estimation

Use map matching skill*[2]*
Estimate the path to next bay
Incorporate line estimation





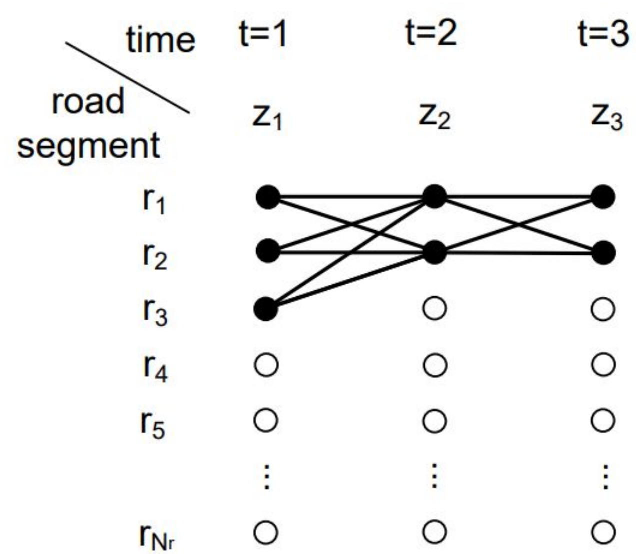
[2] P. Newson, J. Krumm, *"Hidden Markov map matching through noise and sparseness",* ACM SIGSPATIAL GIS 2009

Hidden Markov Model with Line Estimation

Transition probability depends on observations

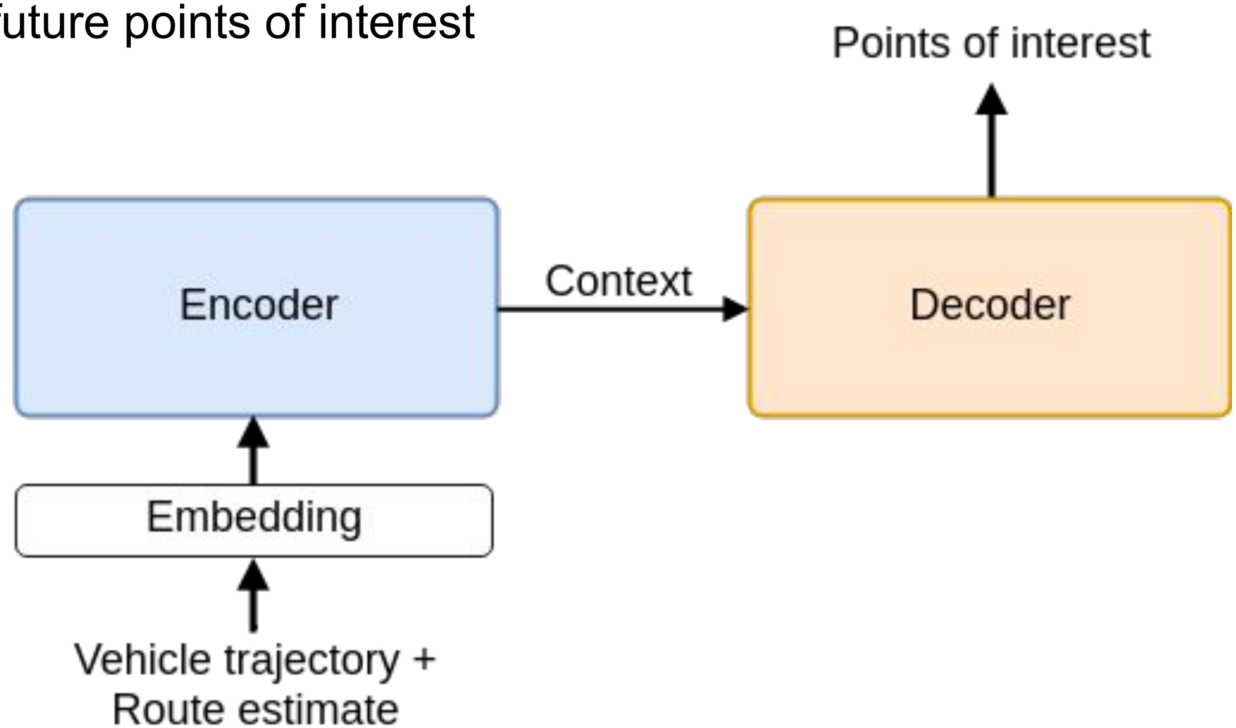
➔ Creating a new model every time

Forecasting without considering all historical behavior



Trajectory Prediction

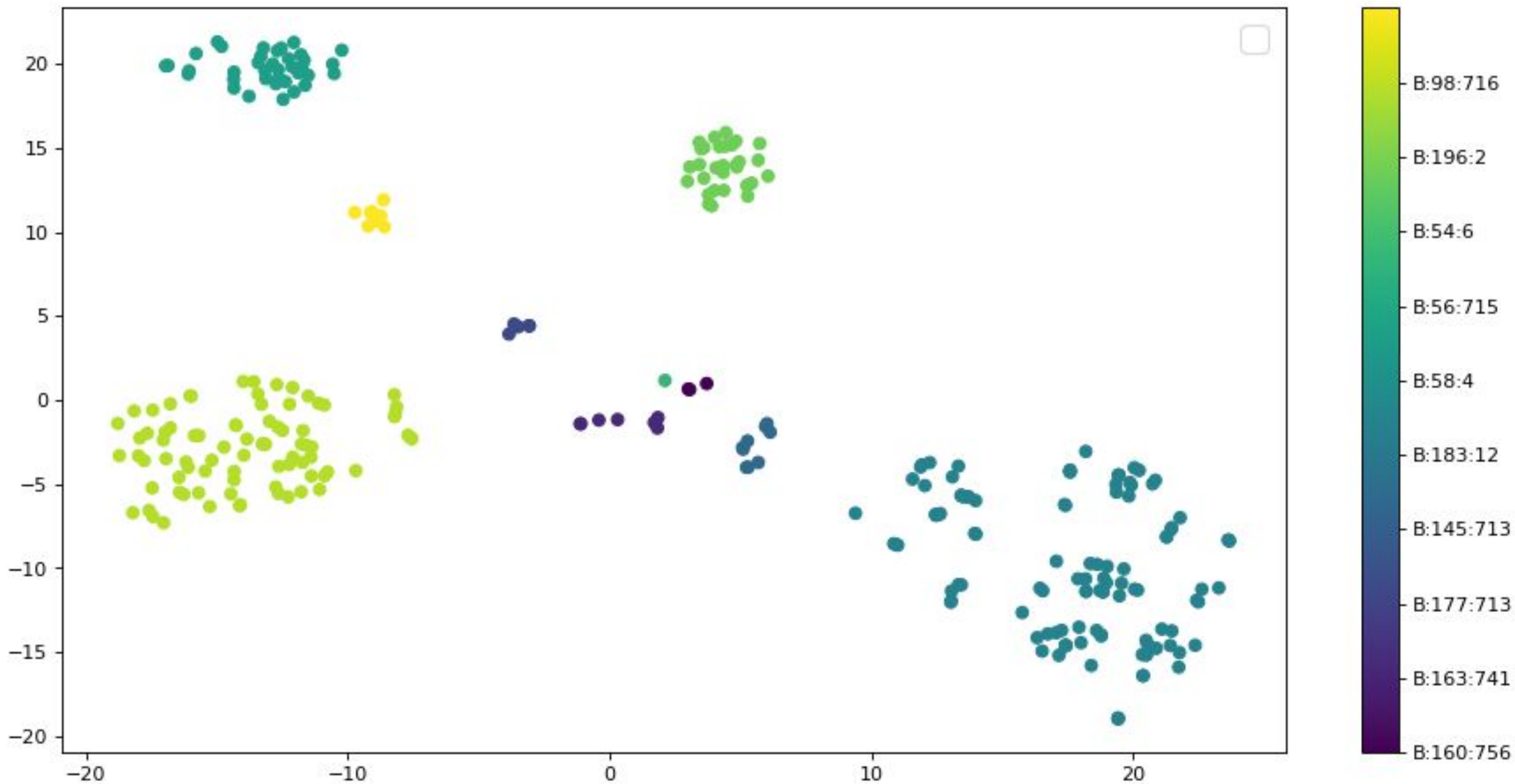
Sequence to sequence model*[3]*
Encode past vehicle trajectory in LSTM layers
Predict future points of interest



[3] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks.

Trajectory Prediction - Latent Space

tSNE projection of encoder output for 10 bus patterns

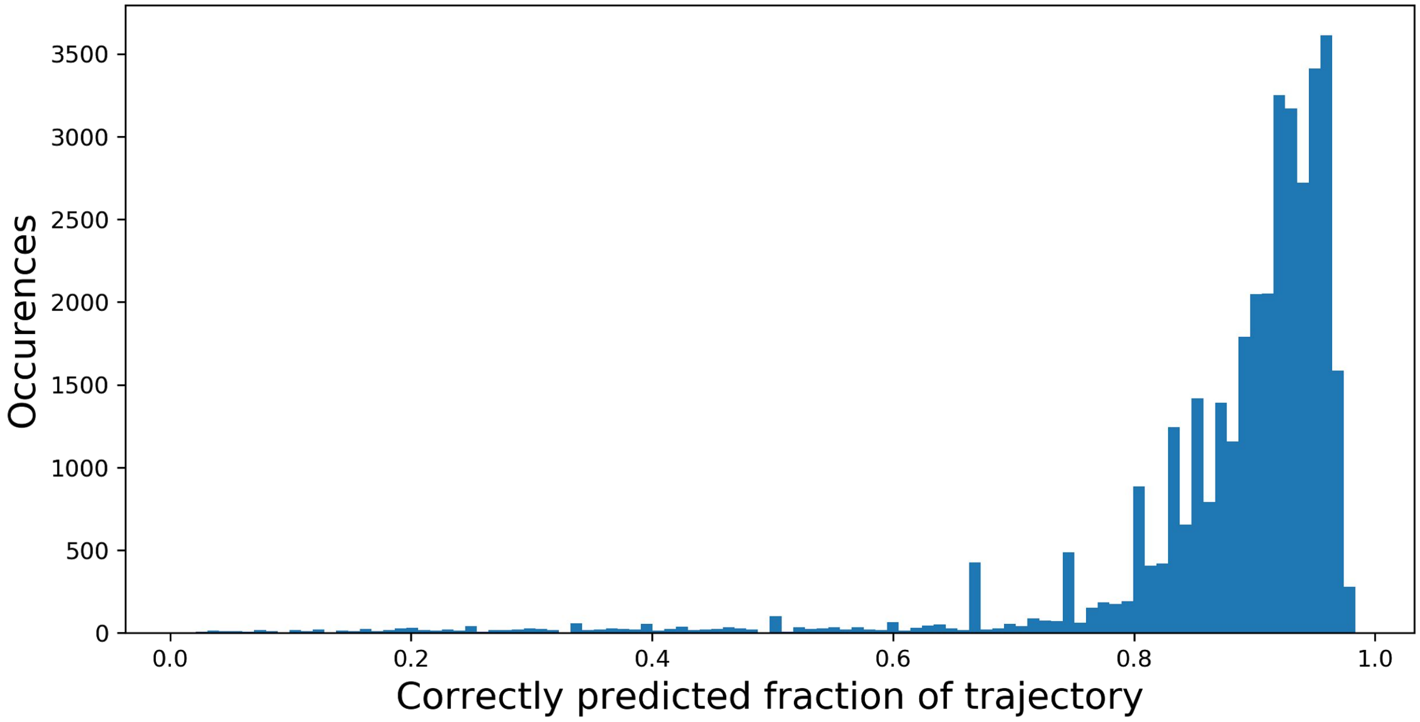


Trajectory Prediction - Test Result

Prediction accuracies for >36,000 trips in March 2018

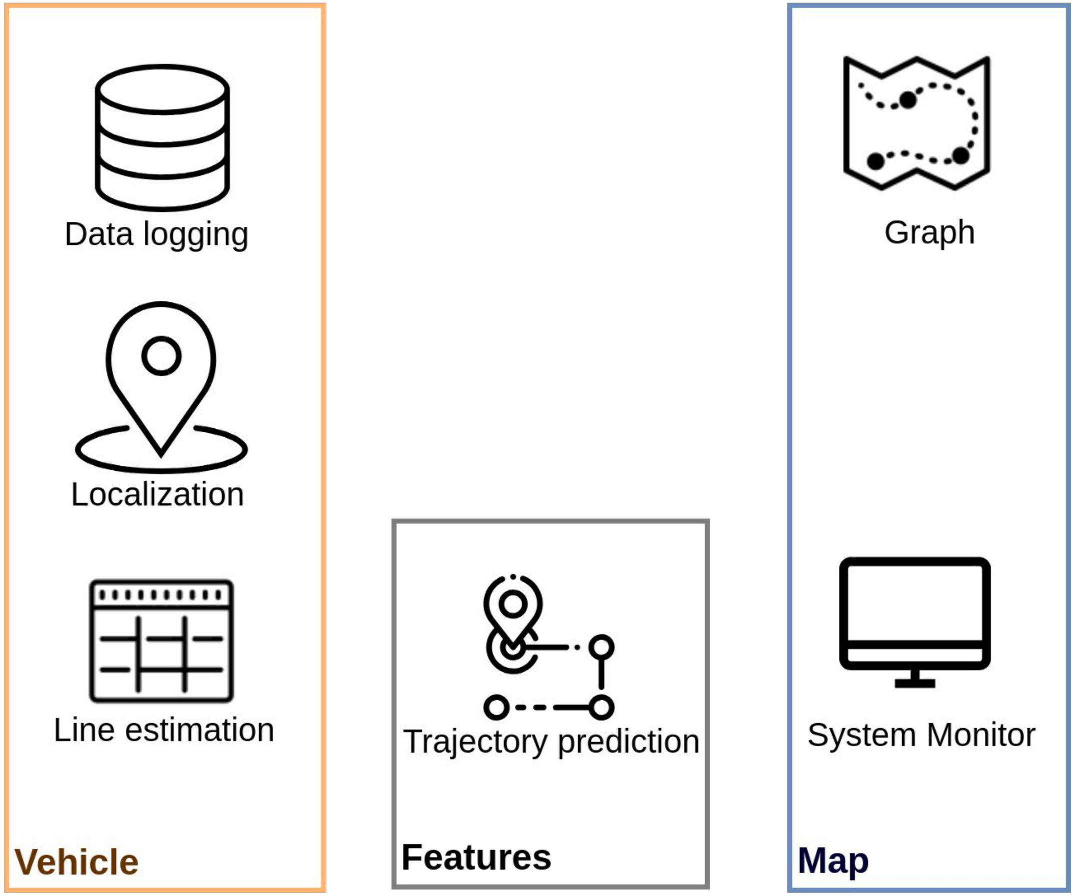
Trajectories fed up to 10% of the actual trip

Test set accuracy: 87.98%



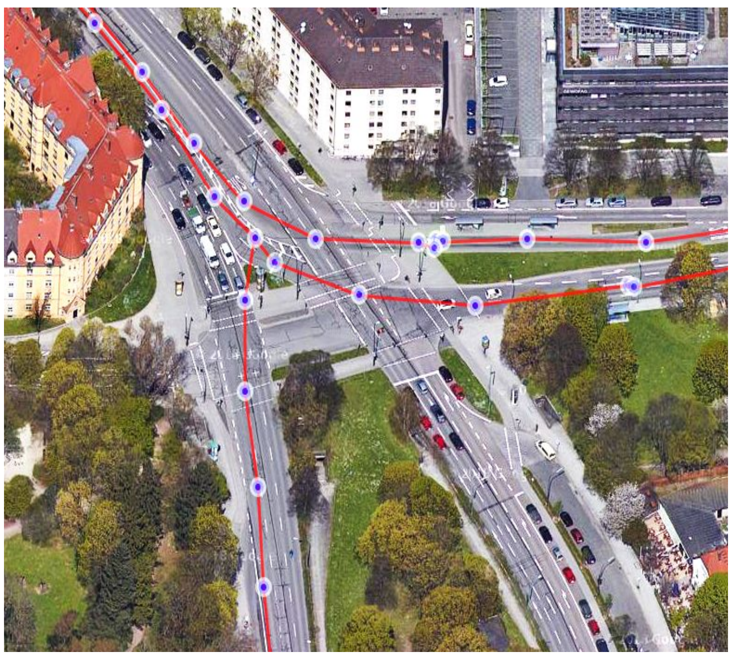
Discussion

System overview

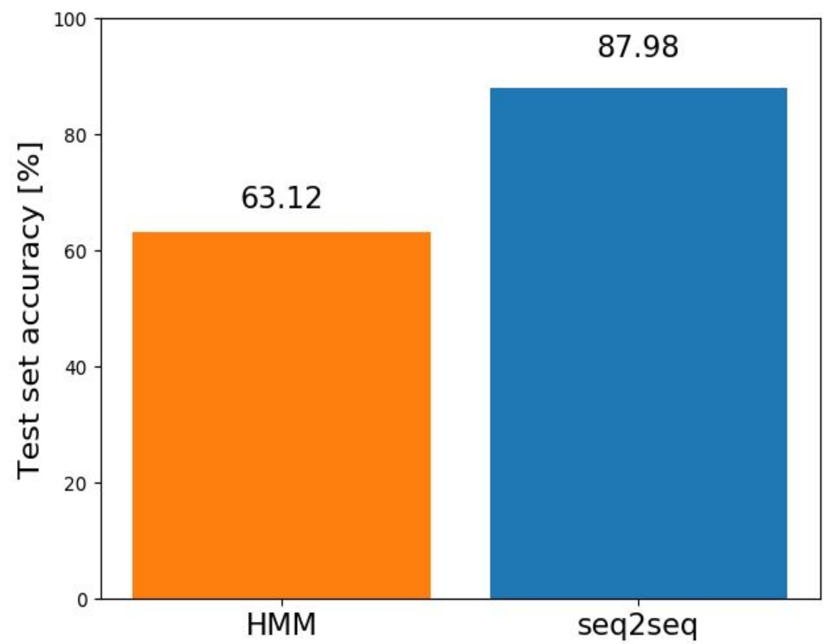


Results

Data driven map generation
Maps entire network in < 8 hours

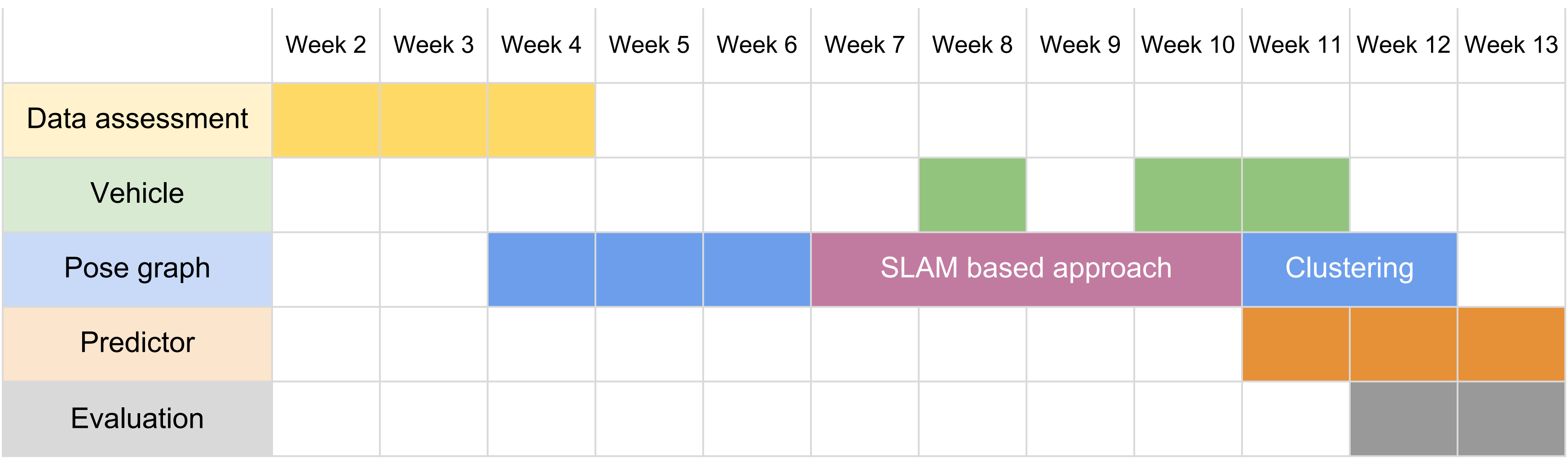


Data driven trajectory prediction
Test accuracy > 80%



Lessons Learned

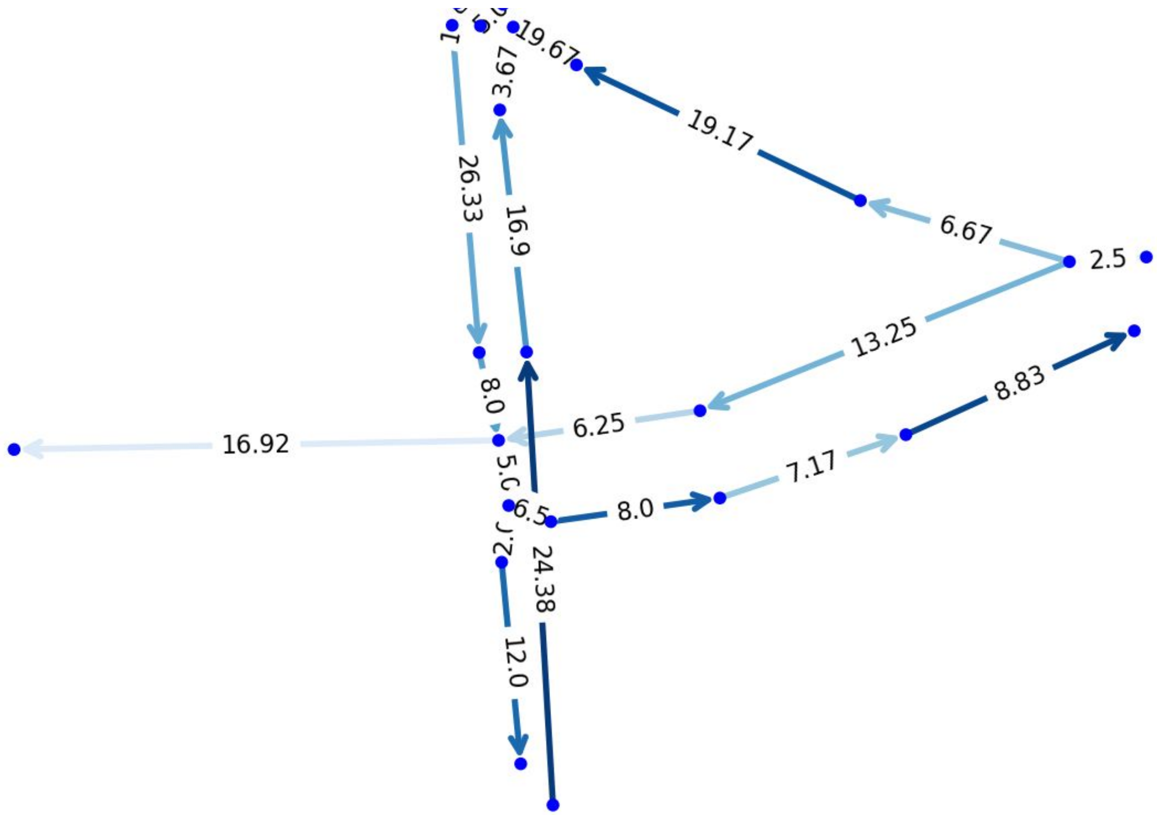
High complexity of public transport data
Spatial databases
Pareto principle

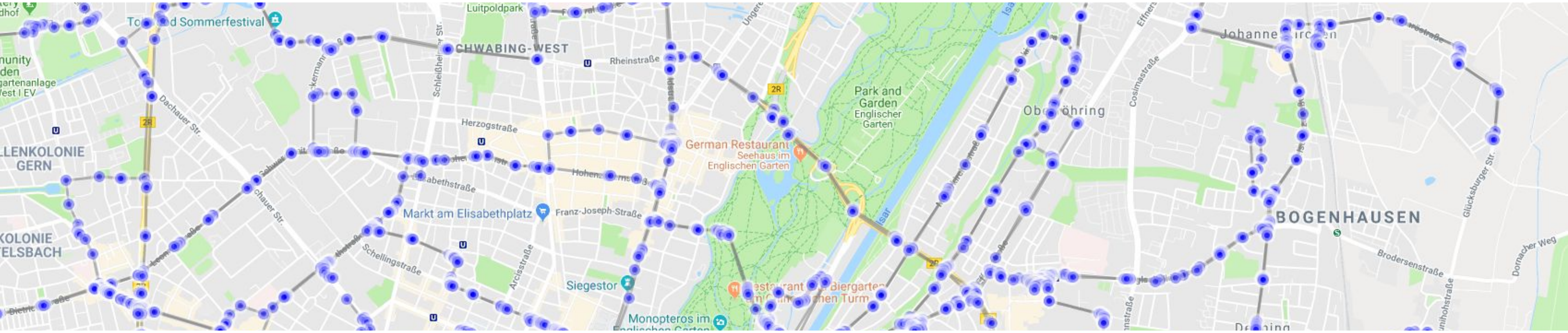


Future work

Travel time embedding in network

Lifelong map learning





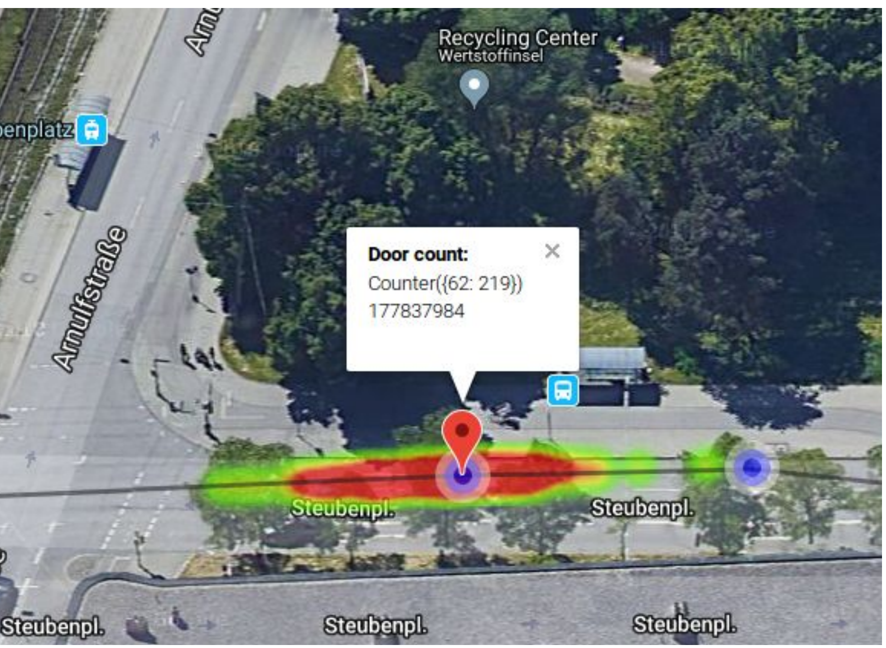
Thank you for your attention.

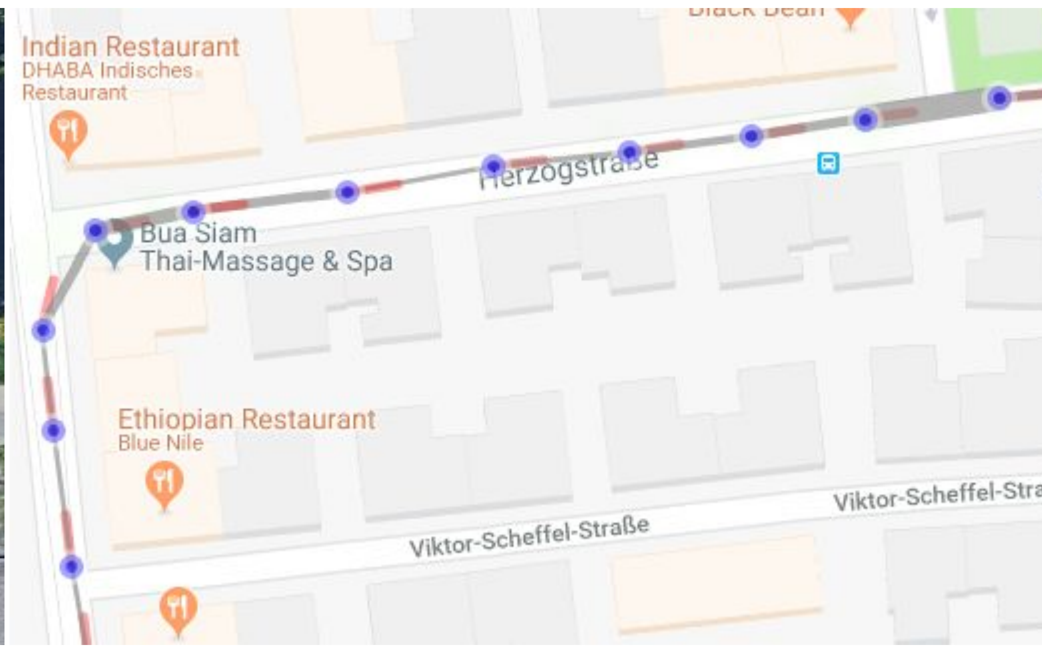
Do you have any questions?

Backup

Additionally Encoded Information

Time and odometry distance on edges
Clustered median heading
Door events with counter and distribution

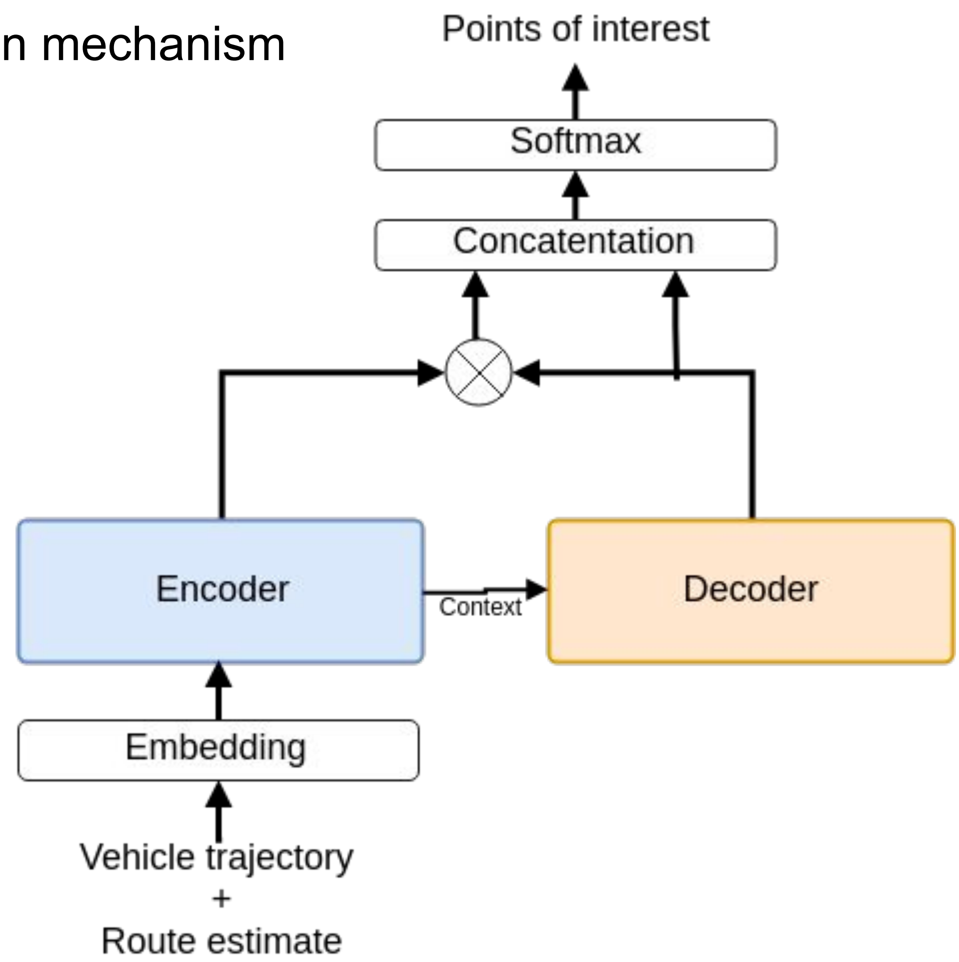




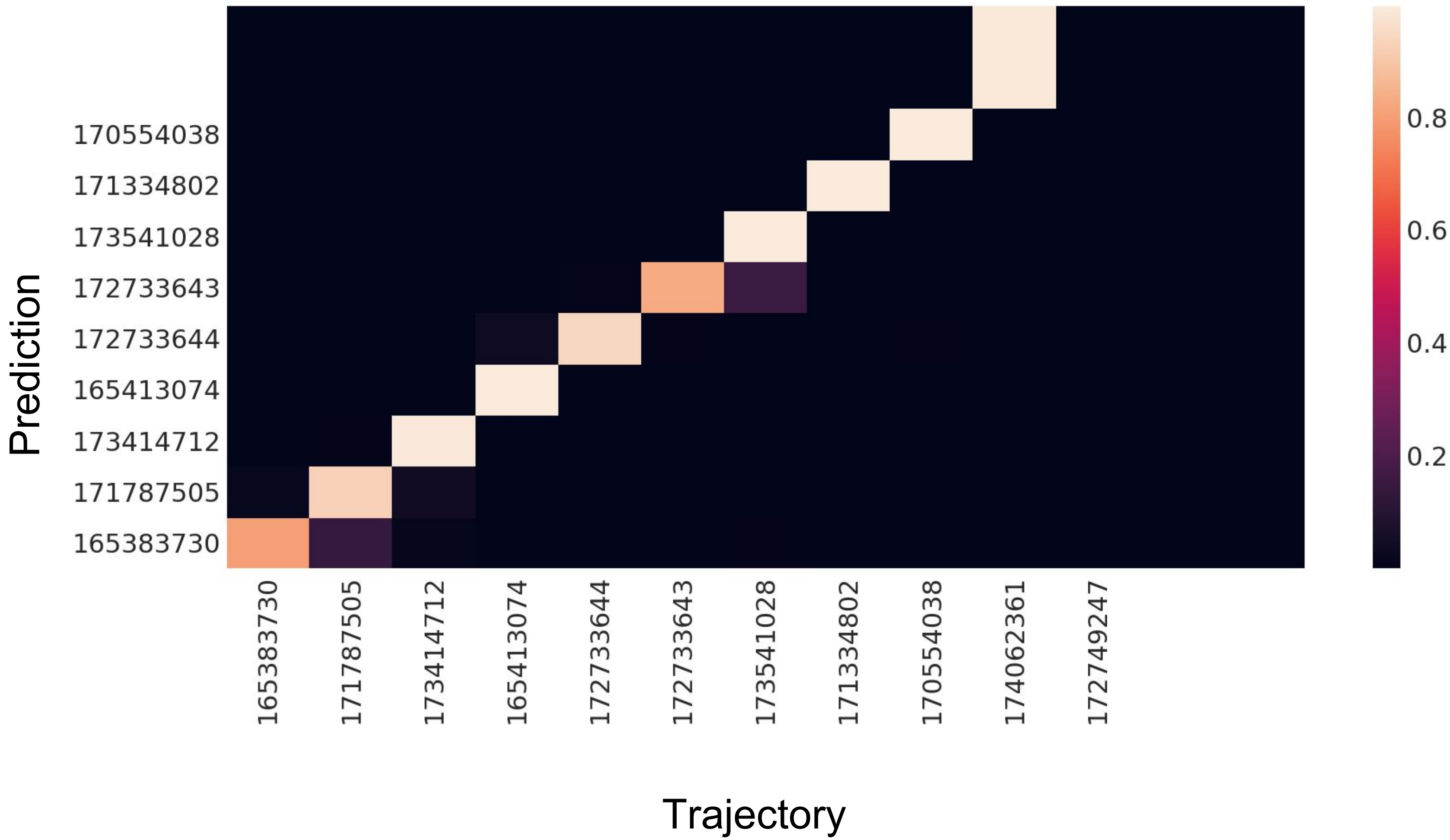
Trajectory Prediction - Model

seq2seq model with global attention mechanism

- Encoder / Decoder
  - stacked layers of RNN with LSTM
- Concatenation
  - tanh activation

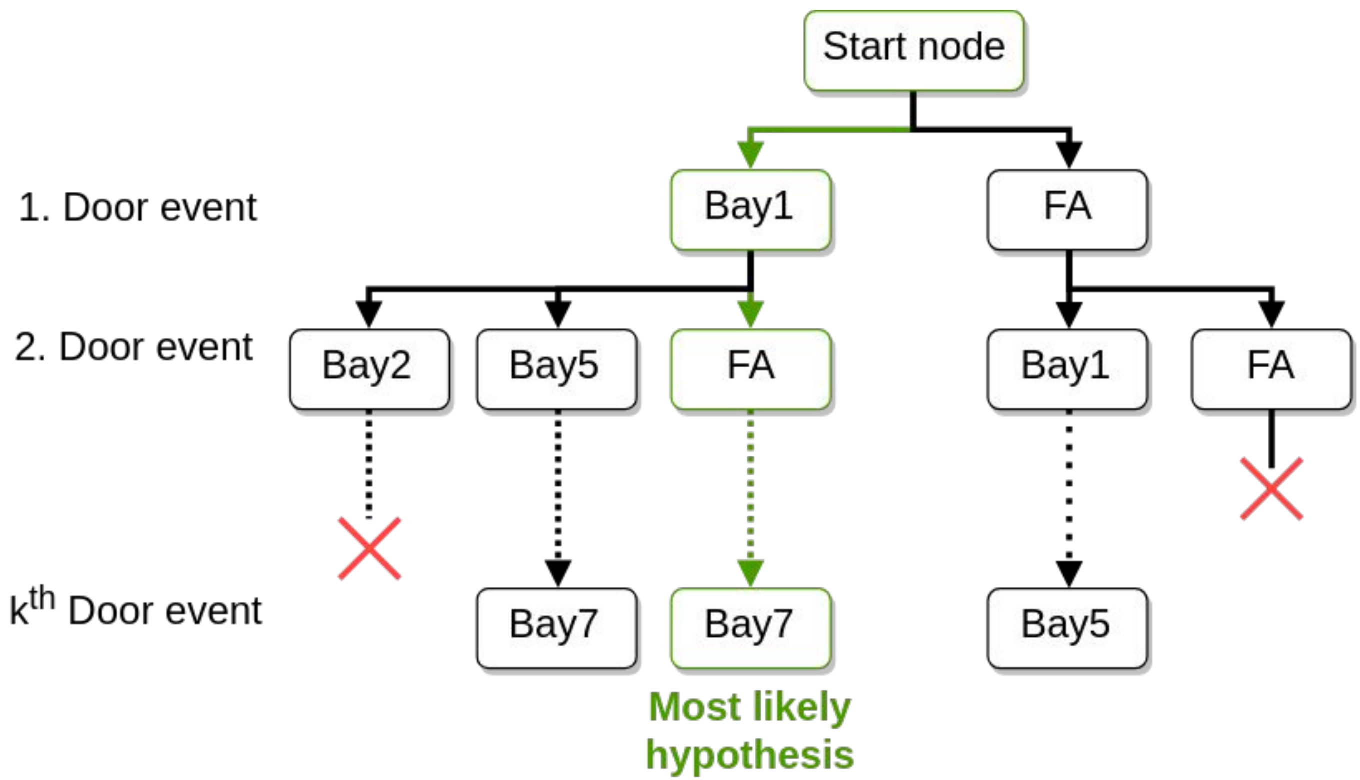


Trajectory Prediction - Attention Map



Door event association

- Vehicle position estimate by Extended Kalman Filter
- Door event as position measurement
- Hypothesis tracking of bay associations



Data Preprocessing

Simple outlier heuristics.

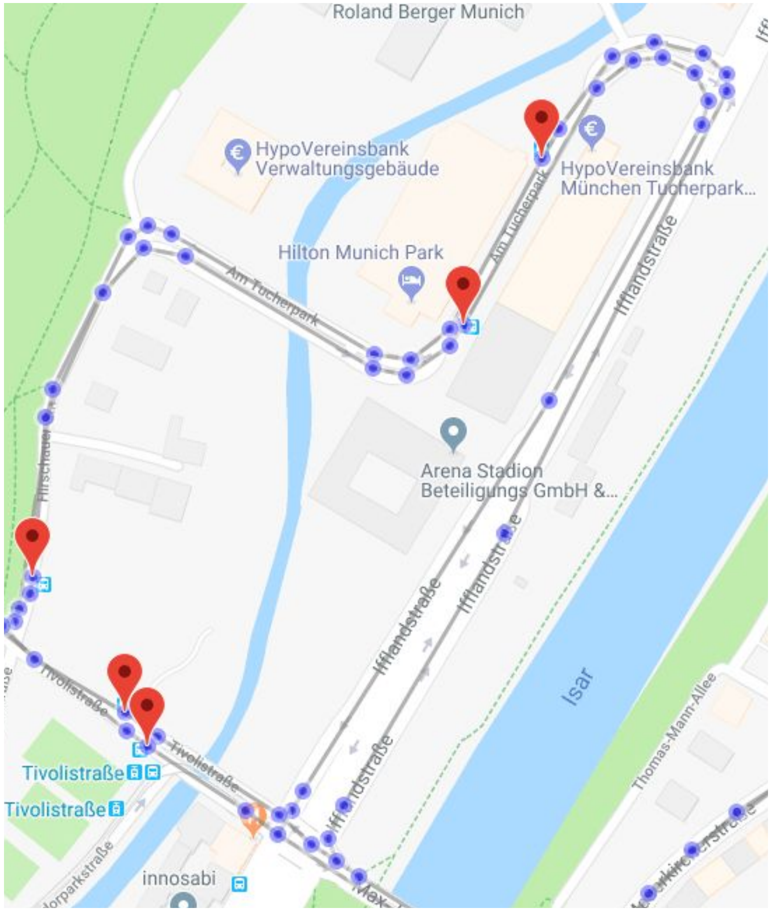
Averaging when the vehicle stands.

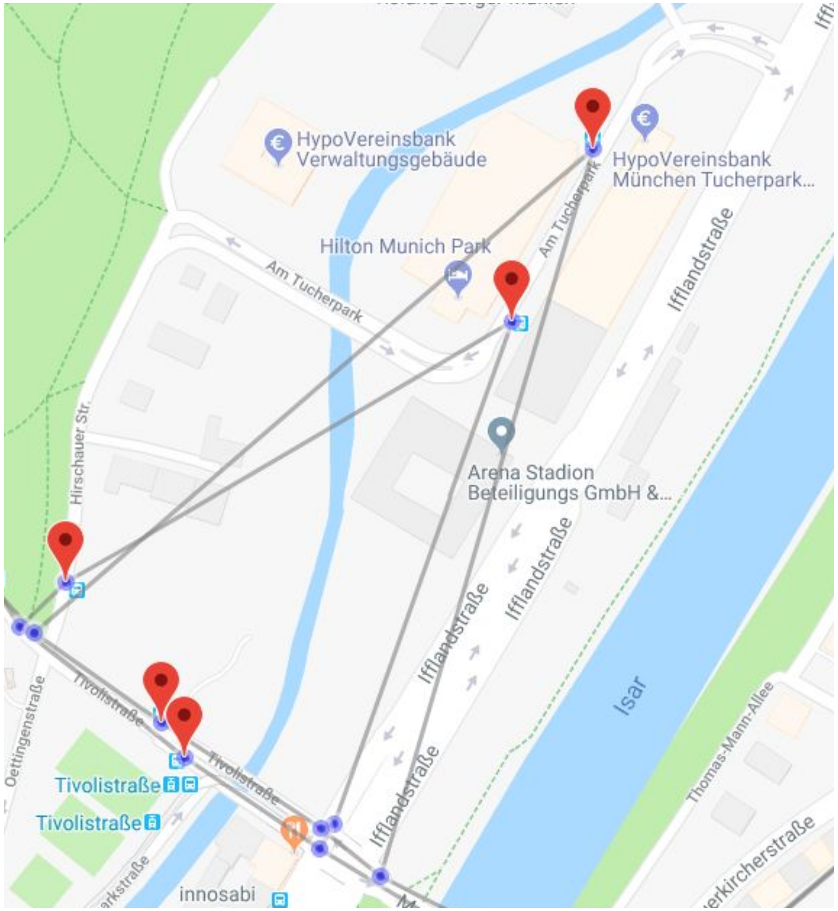
Enforce minimum distance between two measurements.

Single trajectory optimization → SLAM approach.

Abstract Graph

Reduce full graph to only intersections and bays





Steubenplatz more detail

Understanding the Data