Joachim C K B Hansen

Real-time calibration of the ALICE TPC and ML traffic predictions

ESR10

SMARTHEP Yearly Meeting 2024

Lund University

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Overview

- Past year
  - Ximantis
  - Parental leave
  - Physics
- Summary

Currently

- Teaching C++/ROOT
- Started a new physics analysis in Run3 with O2Physics
  - Mostly software related currently
- Courses

Parental leave

Input data

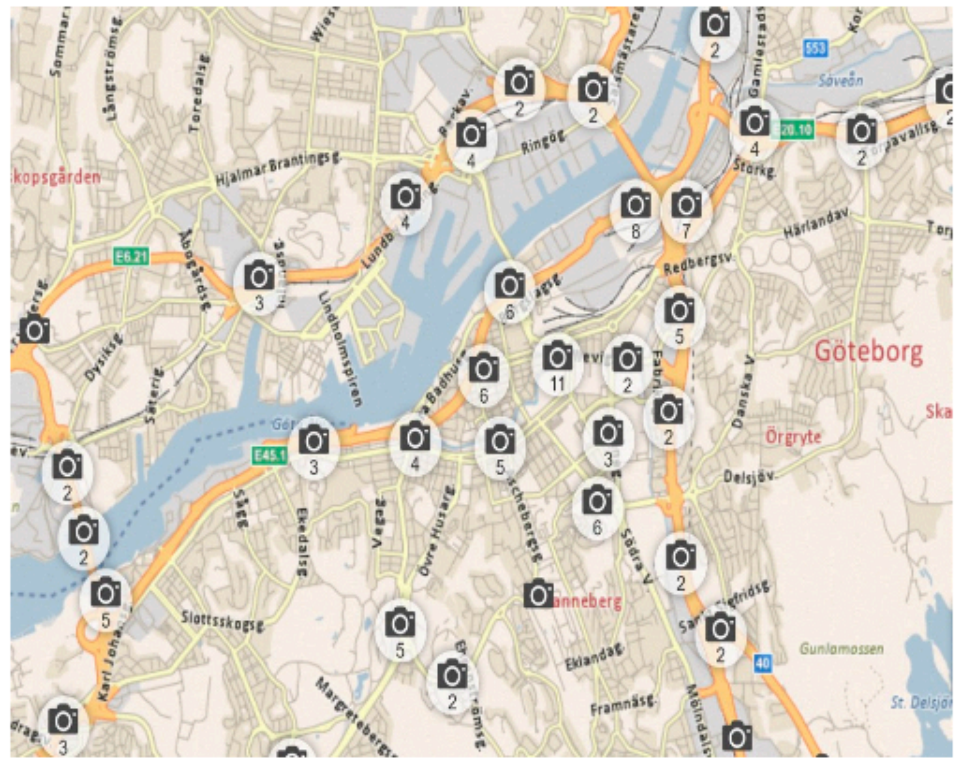


Fig. 1. A section of the Goteborg traffic network with multiple cameras indicated along each road each collecting images every minute. Data provided by the Swedish traffic authoridy: Trafikverket.

Sopasakis, A. (2019)

One cam example →

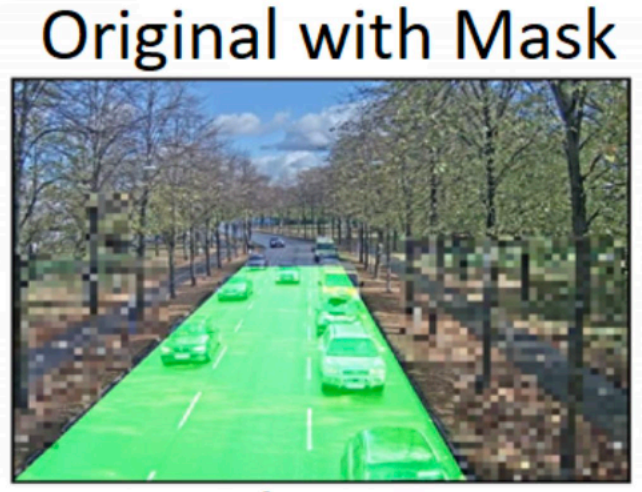
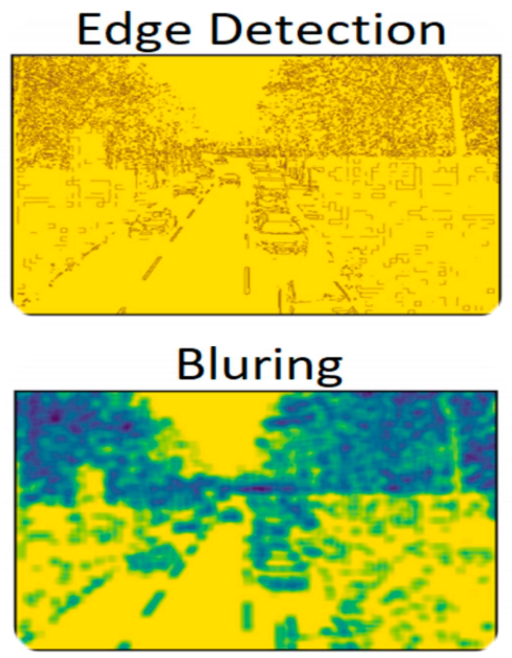


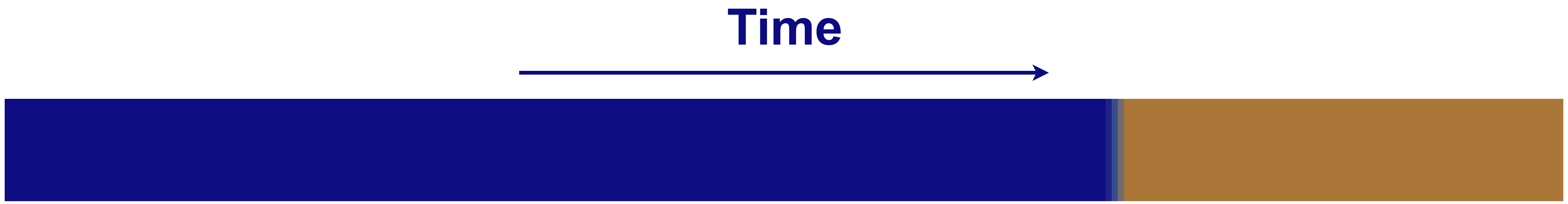
Fig. 2. Original image with superimposed mask on the region of interest.

→ **Image Analysis** →



→ **"Flow"** →

- Input data of Network
- Done for every available camera



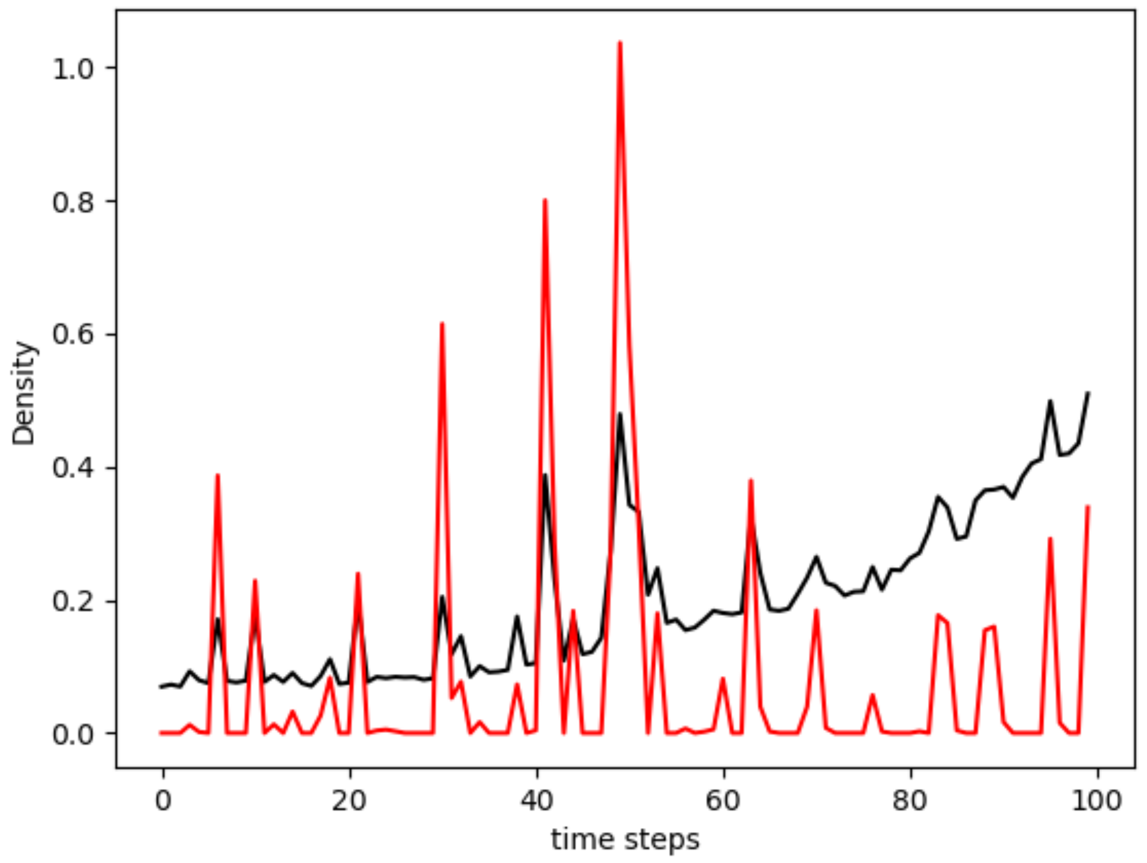
Time

Training

Validation

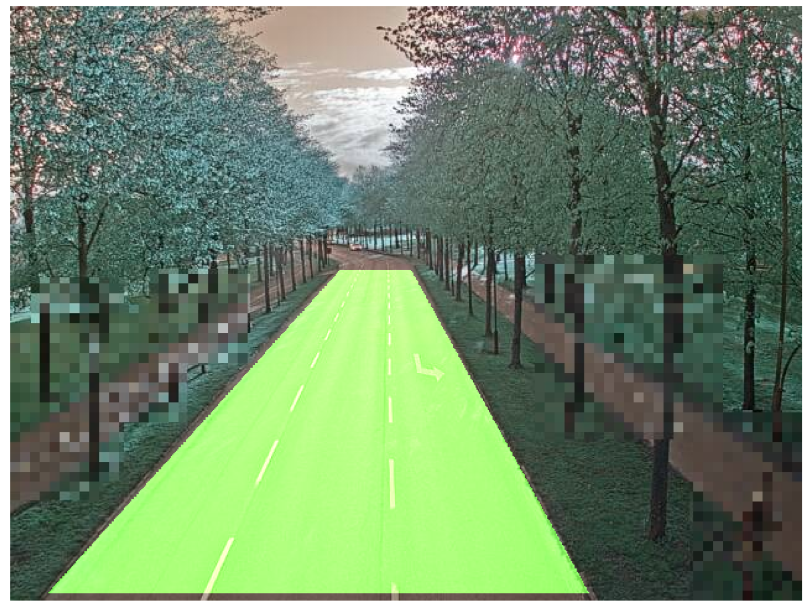
“Background” density

- Systematic shift during the day (black)
- Remade density estimation



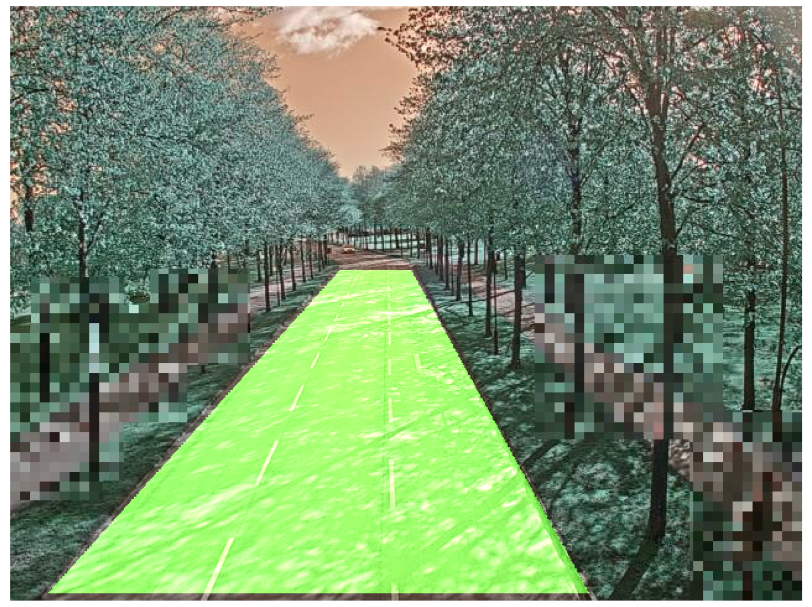
Morning

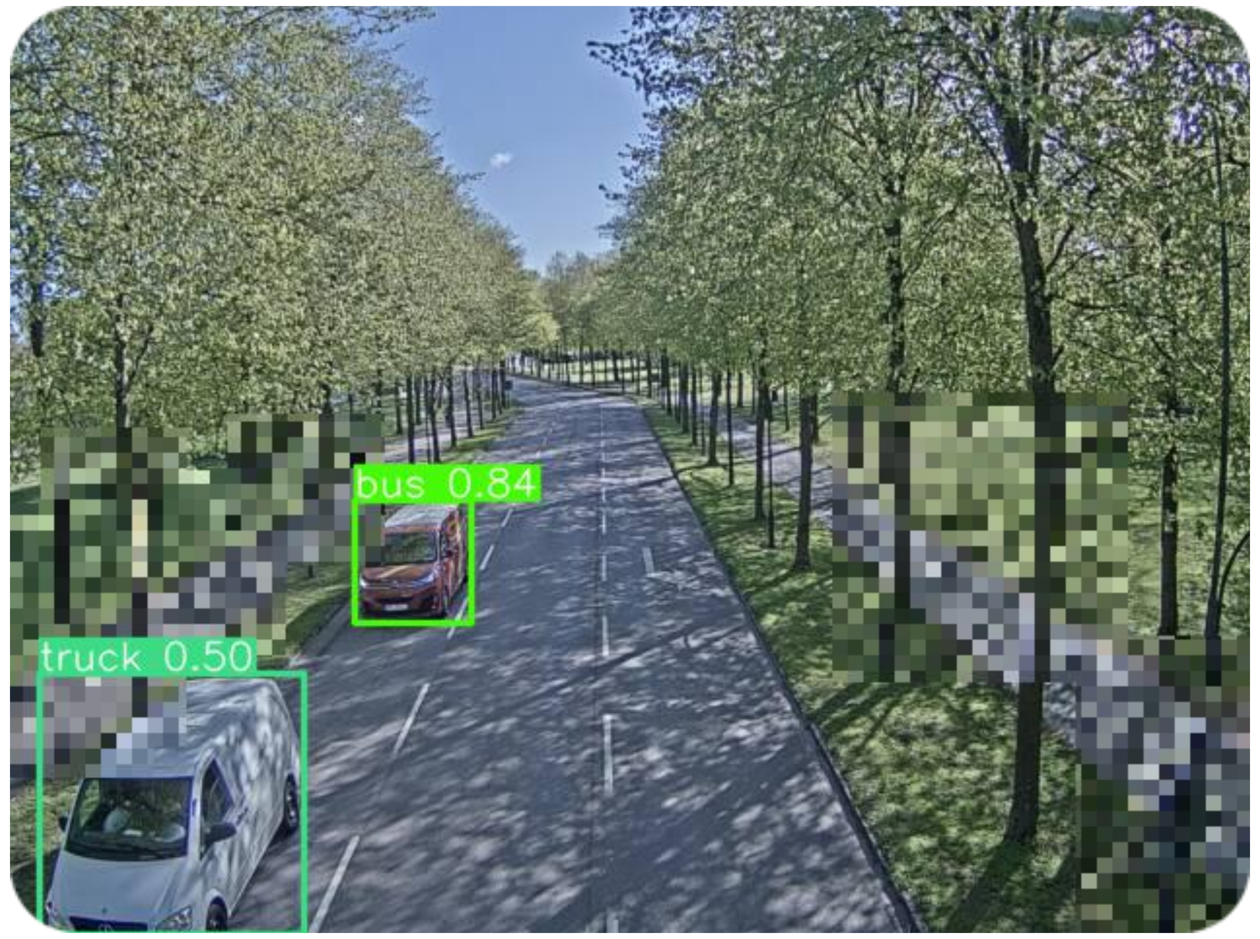
Density: 6.9596842838350215%



Afternoon

Density: 40.47357424746576%

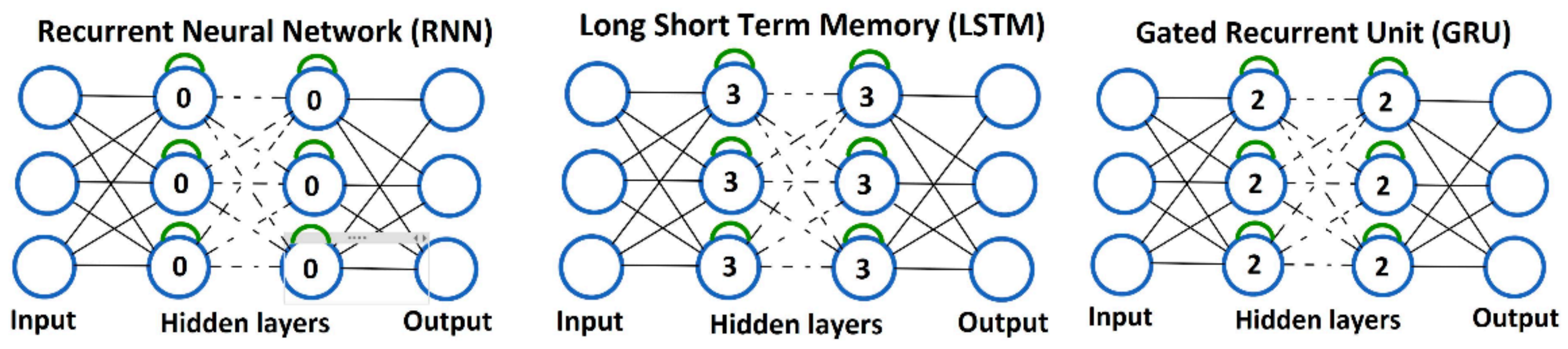


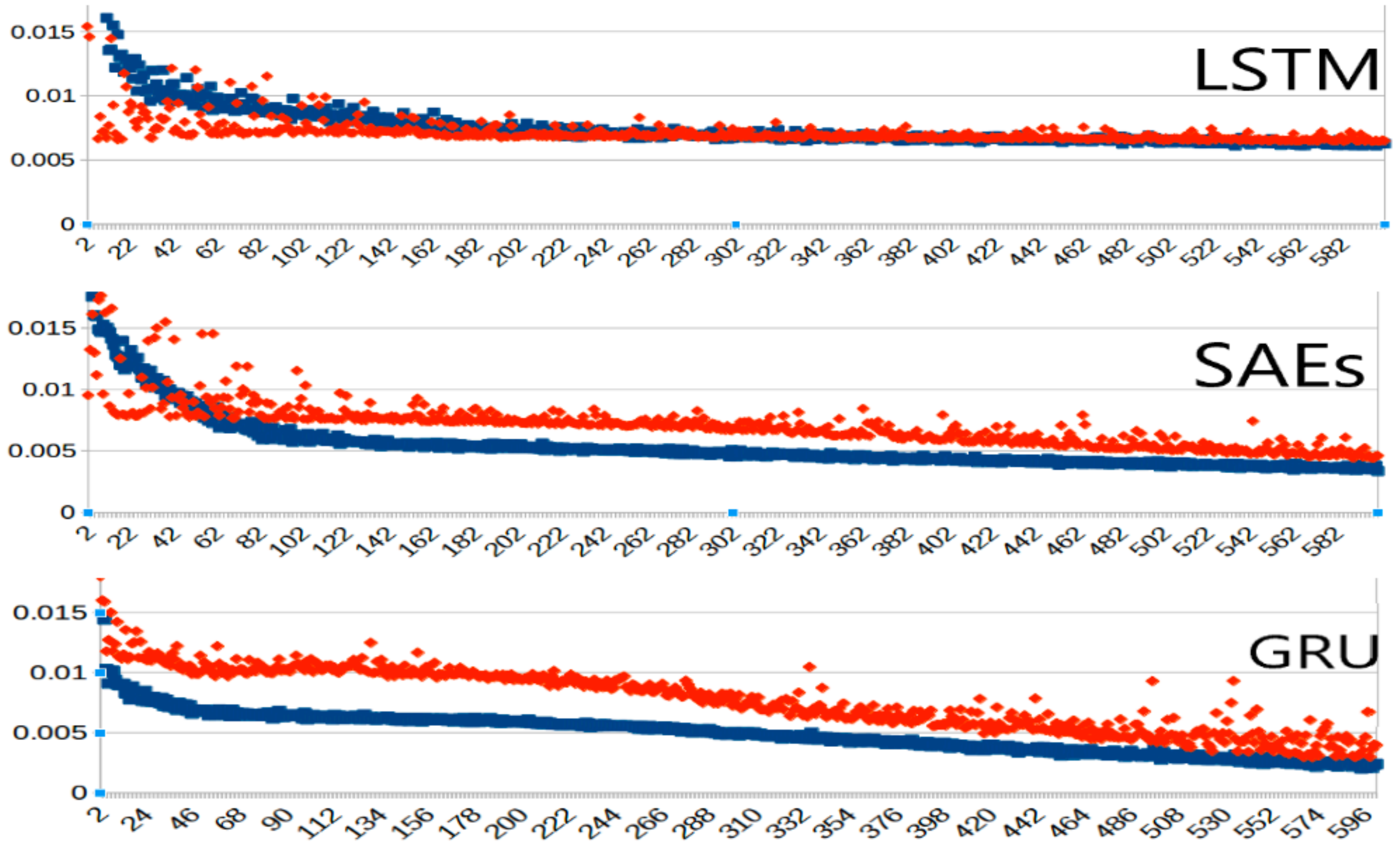


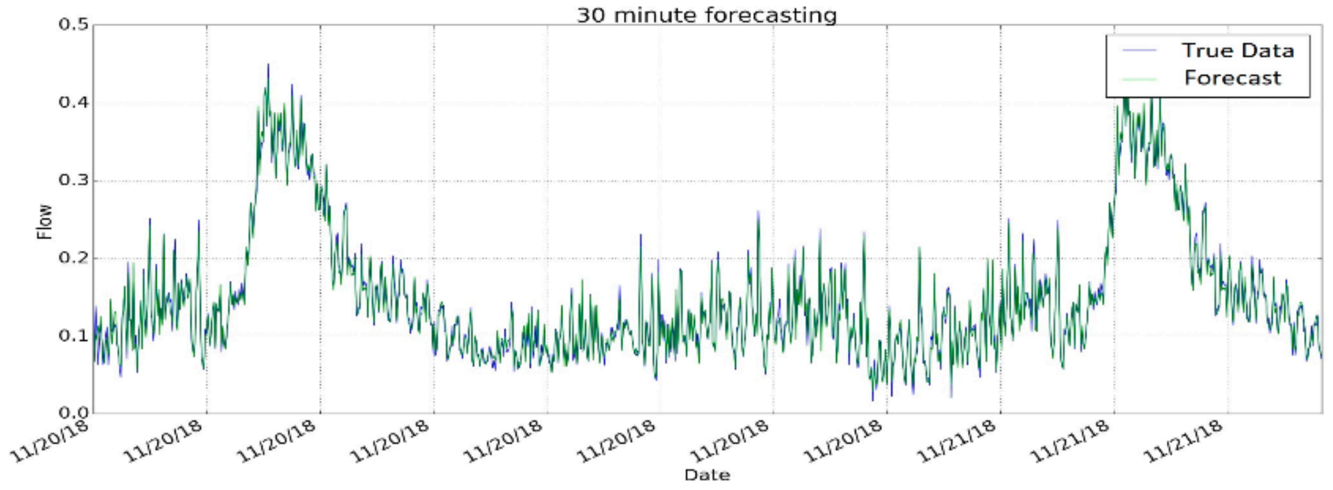
- Used YOLO pre-trained on coco
- Specifically made for real-time detection

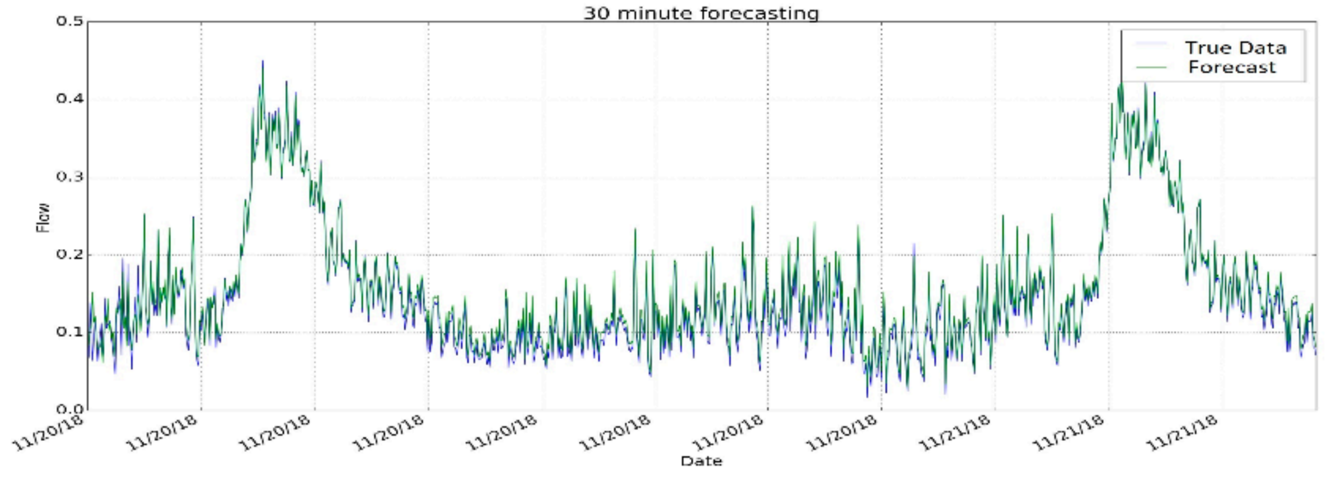
Previous work

- Traffic forecasting using different approaches…
- Works well, but can be improved
- Only temporal information
- Hard to benchmark (density algorithm)


Recurrent Neural Network (RNN)
Input
Hidden layers
Output
0
Long Short Term Memory (LSTM)
Input
Hidden layers
Output
3
Gated Recurrent Unit (GRU)
Input
Hidden layers
Output
2

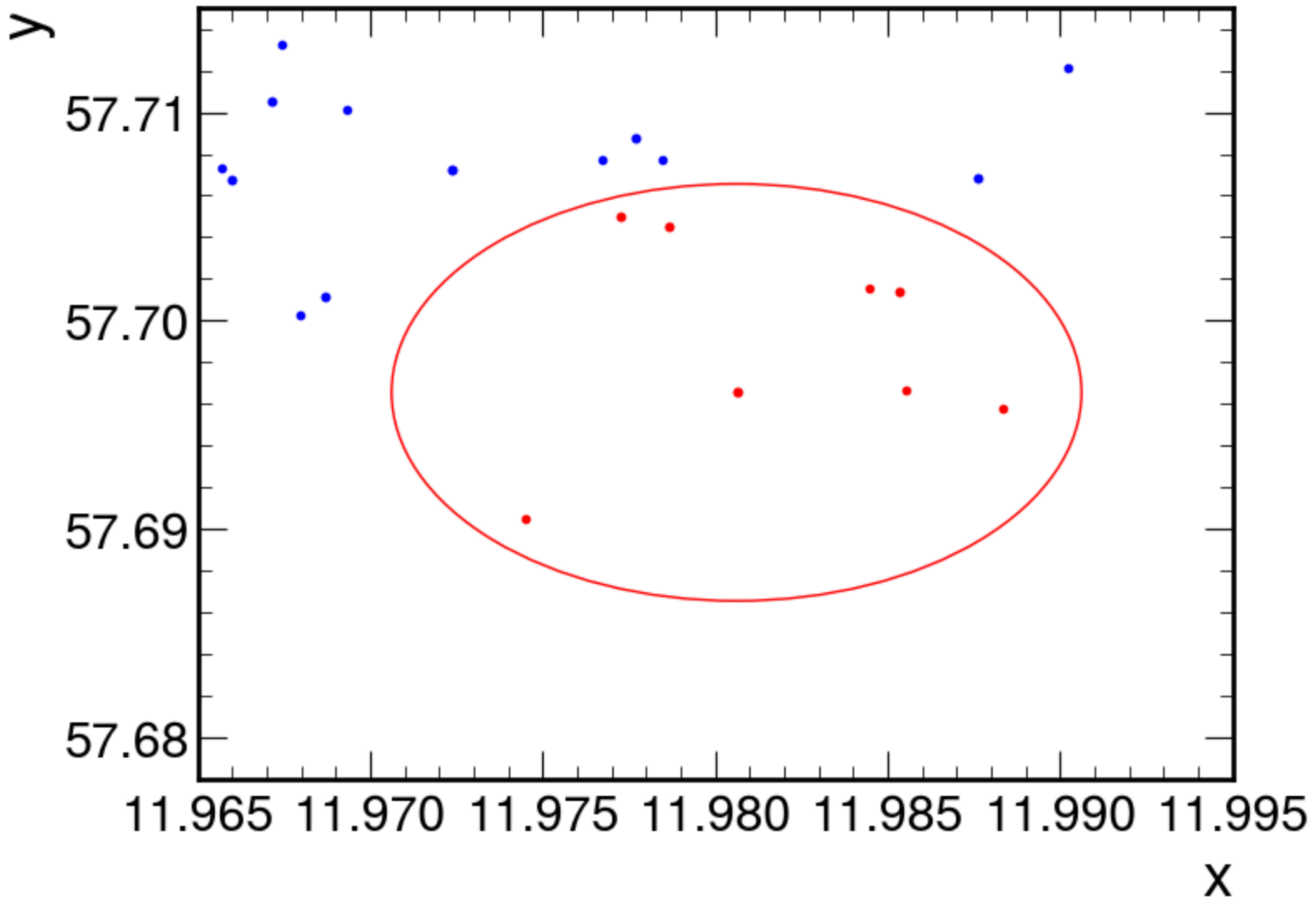


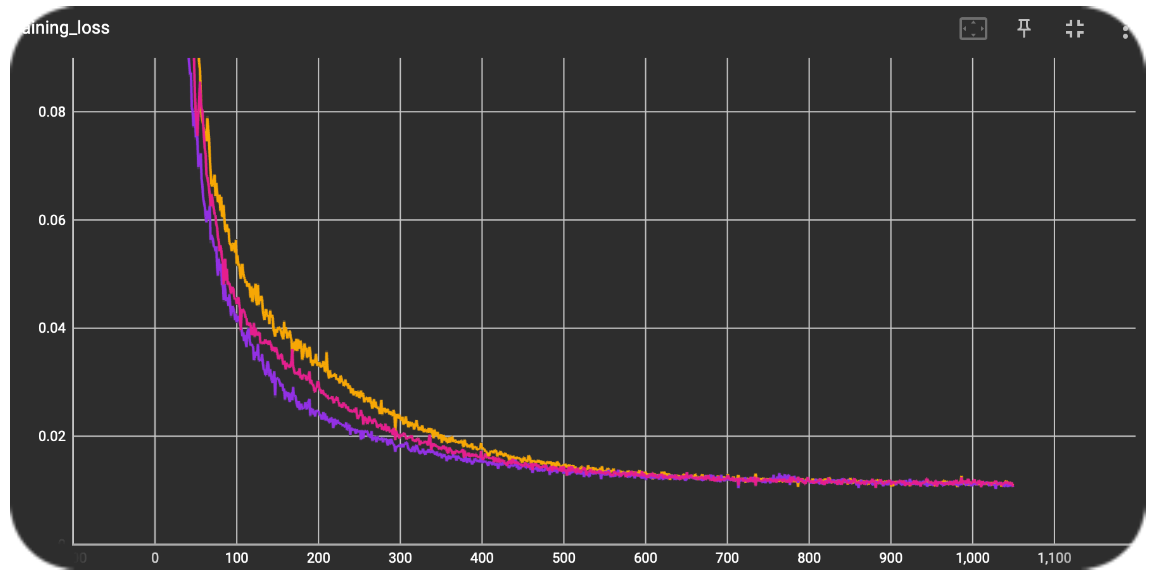




Reminding you..

- We want to use both the temporal and the spatial information
- Initially naïve assumptions:
    - Create the (symmetric) adjacency matrix based on some D_{min}
    - Cameras will use nearby nodes as spatial information

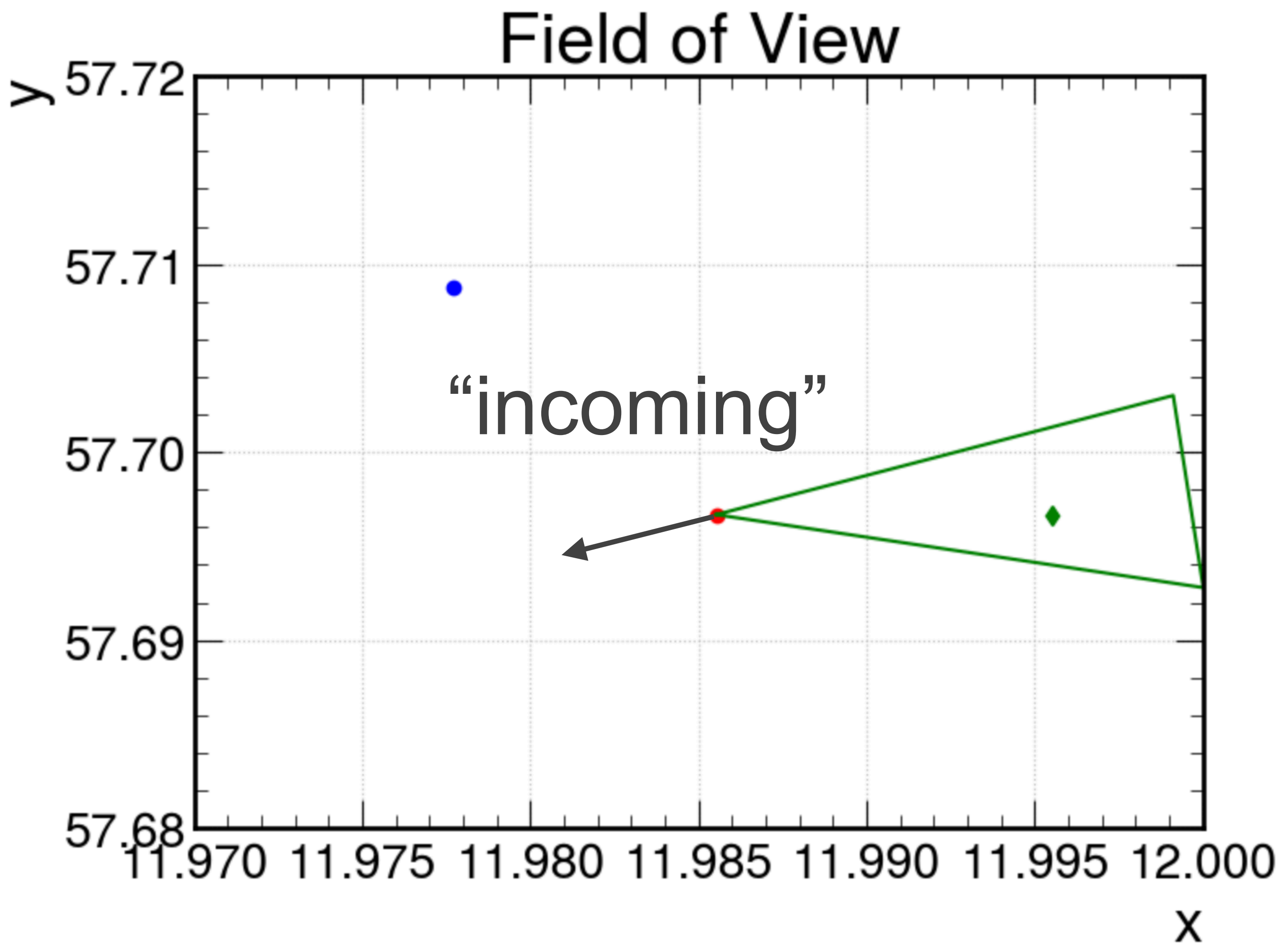


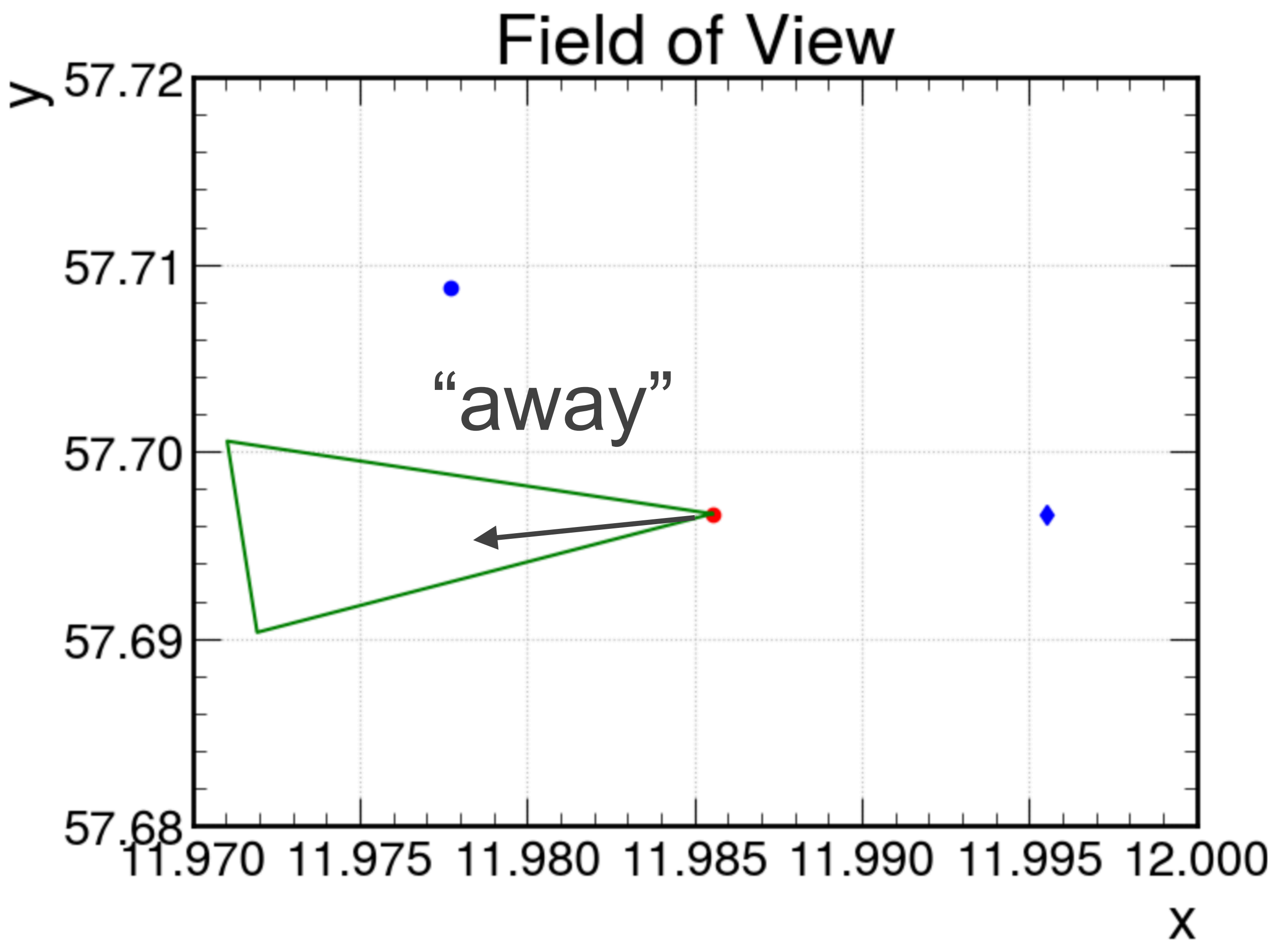


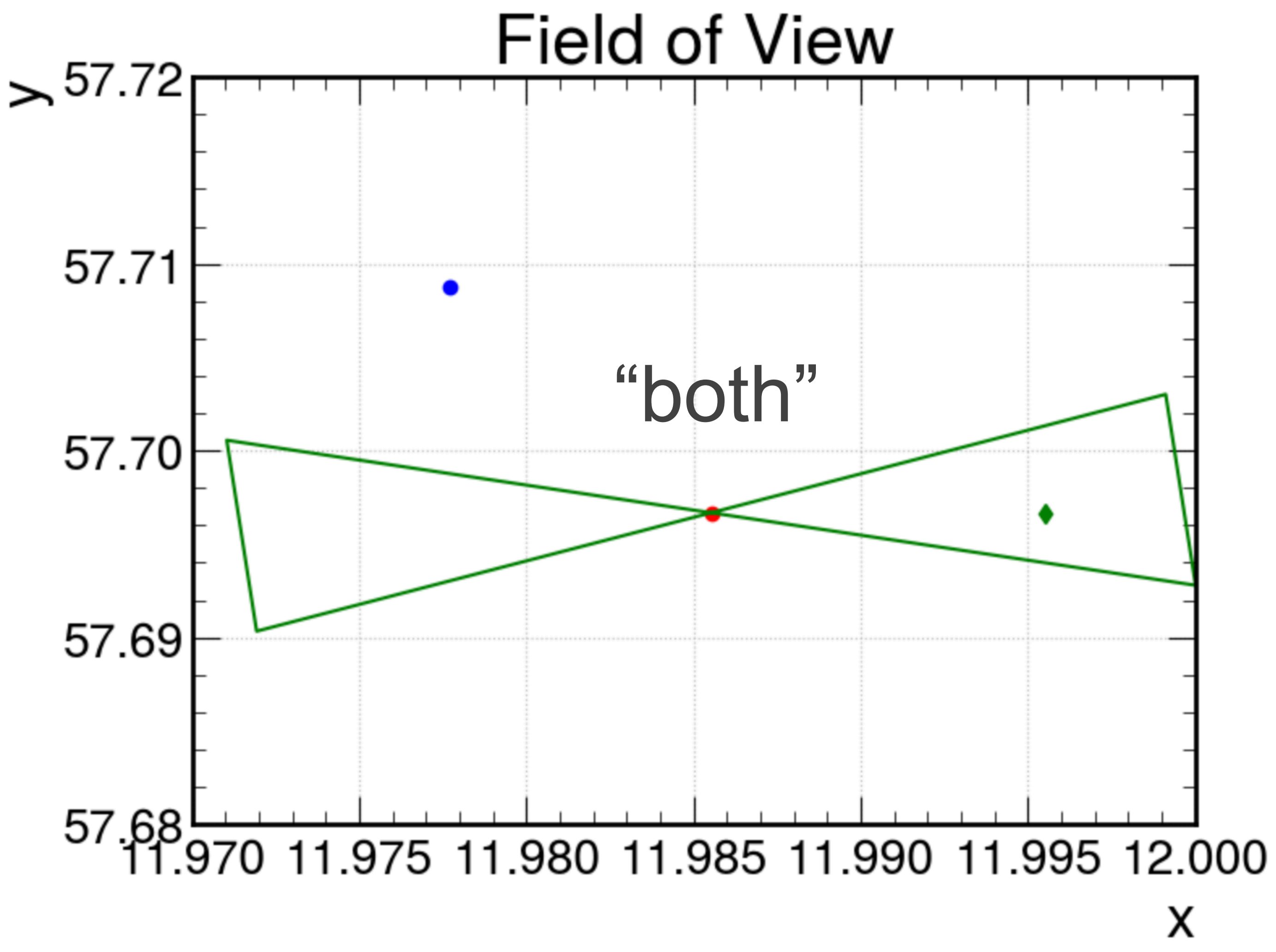
Göteborg
Örgryte

Adjacency modified

- Categorize directional traffic based on the cameras as either:
  - Incoming
  - Away
  - Both
- In short
  - Cameras gets a constrained FoV with a view cone
  - Increases spatial accuracy

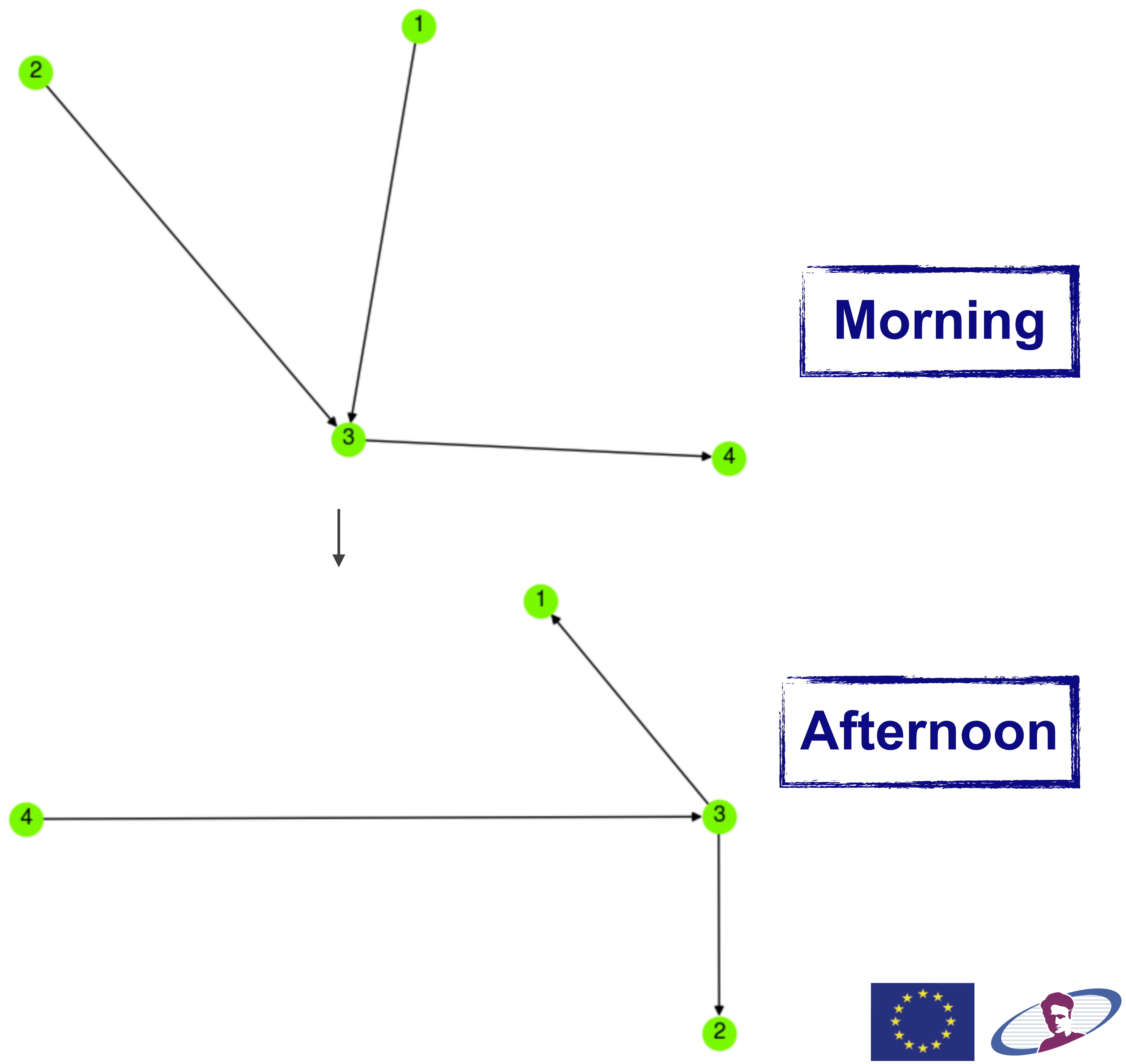






Time-dependent Laplacian

- In principle the traffic varies a lot by hour and day
- We wanted to create the adjacency matrix as a function of time of day
    - Dynamically using a neural network
    - Using input “flow” as a means to estimate it



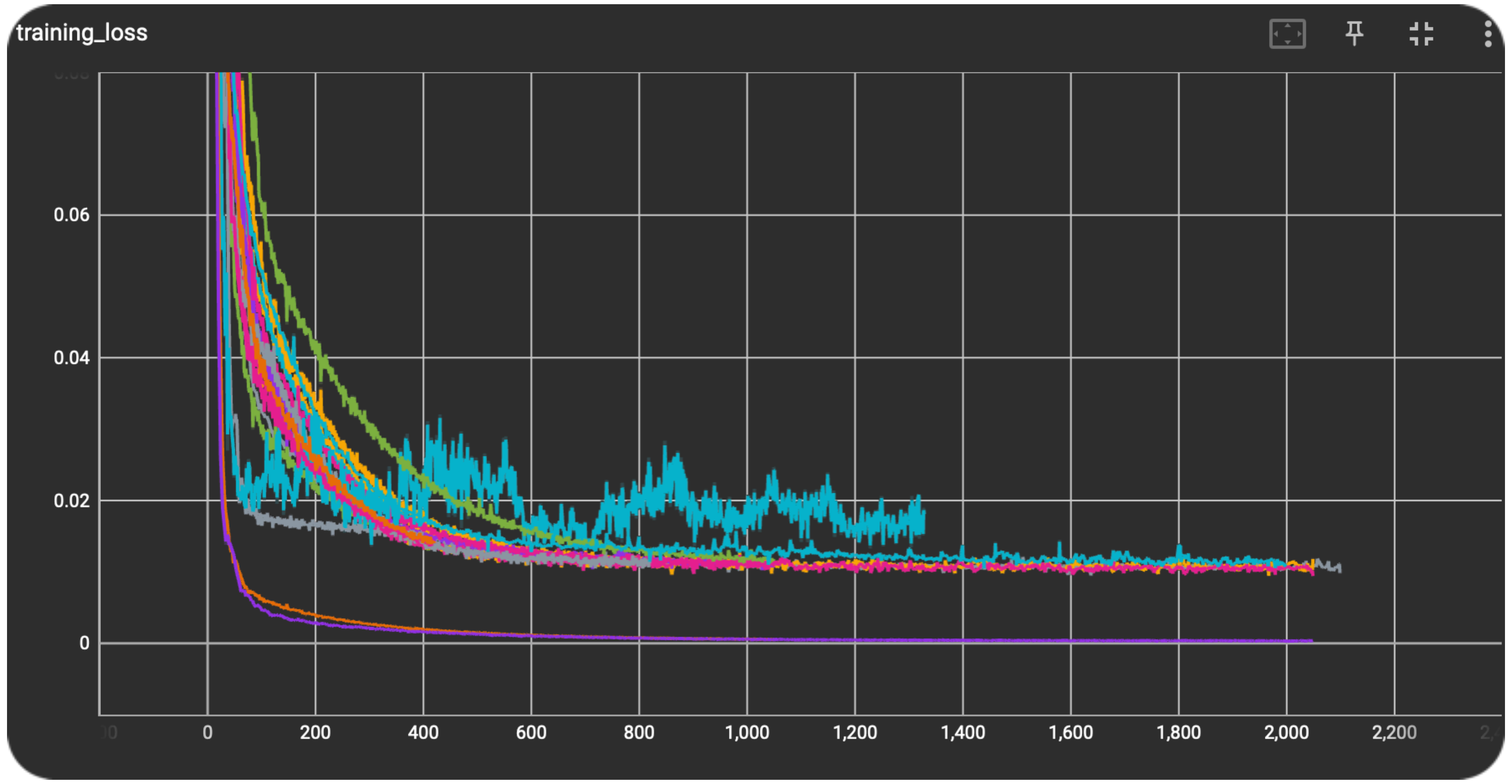
Data

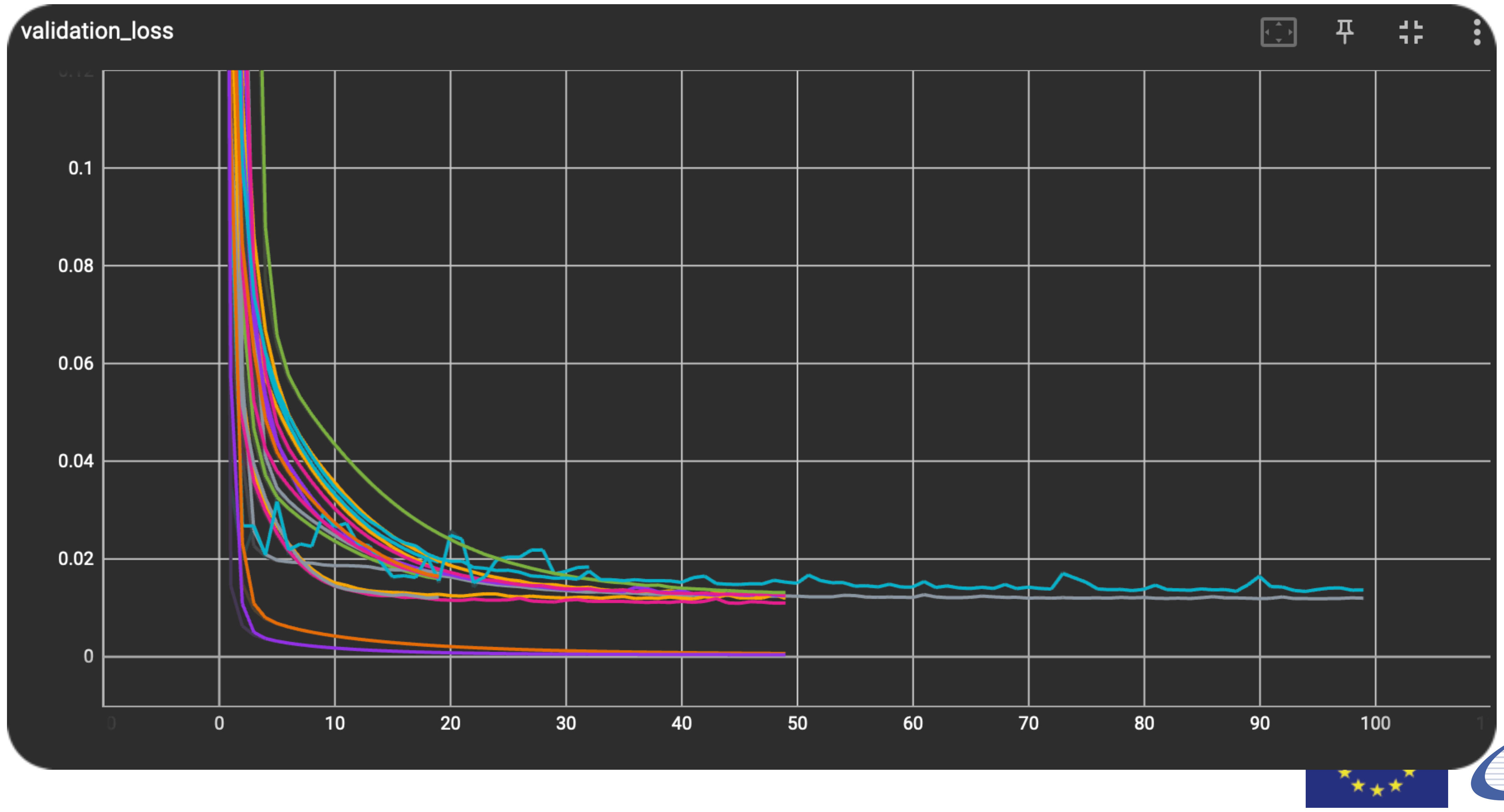
- We end up wth data from each camera and a relation between cameras
- Laplacian
  - (Dynamical one was still a WiP)
- Input data
  - [B,N,F,T] : Batch, Node, Feature, Time-step
  - Feature = 2 (3)
    - Density, time embedding

- Use 1 hour input to predict next hour
  - Several approaches

Learning anything?

- Short answer: yes
- Are the results good?





Results

- **Select few** results
- Randomly selected network
- Small scale prediction is not excellent, but does follow trend on larger scales
- Improved time-embedding and increased data size could improve results
- Similar to what was privously obtained

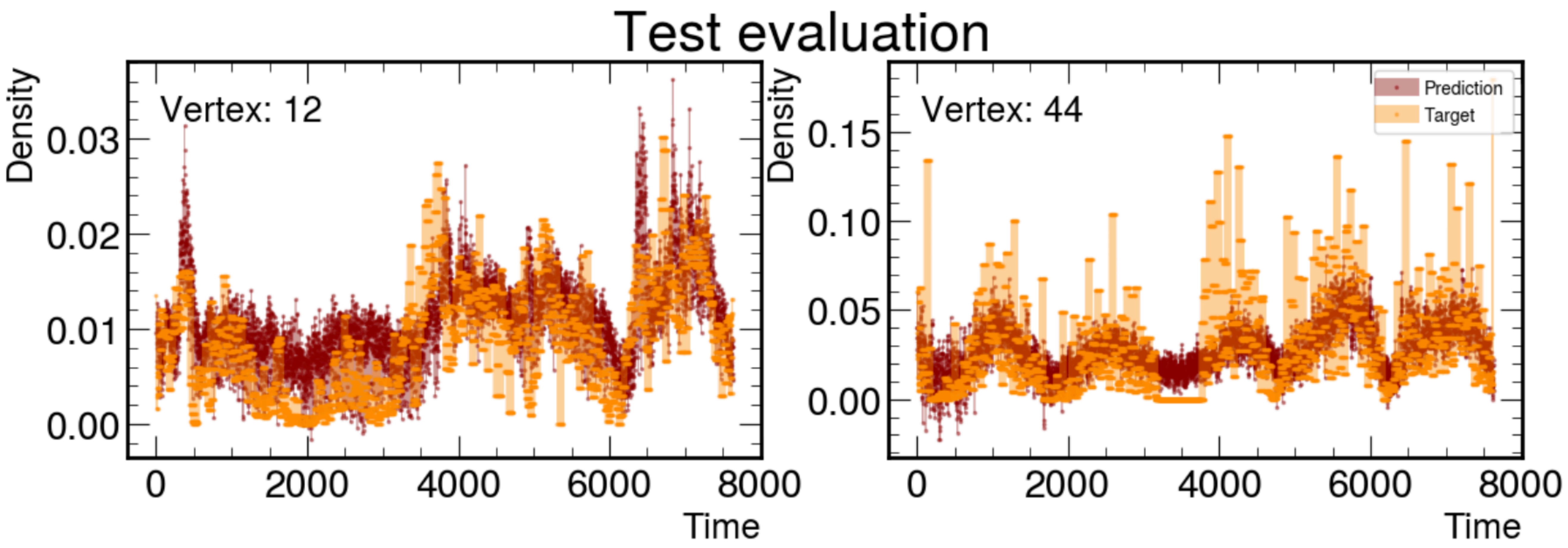
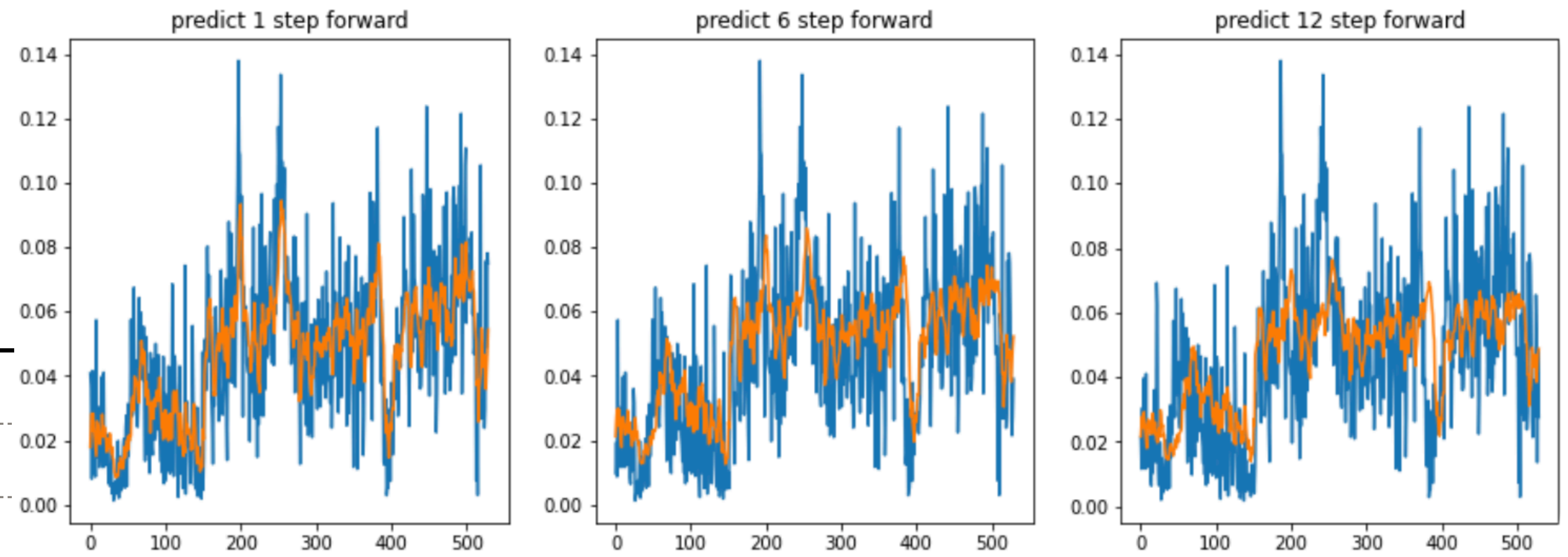

Test evaluation

Table 1

Data length (days)	Test MAPE
3	1.5206
7	1.1028
14	0.9926



Summary and outlook

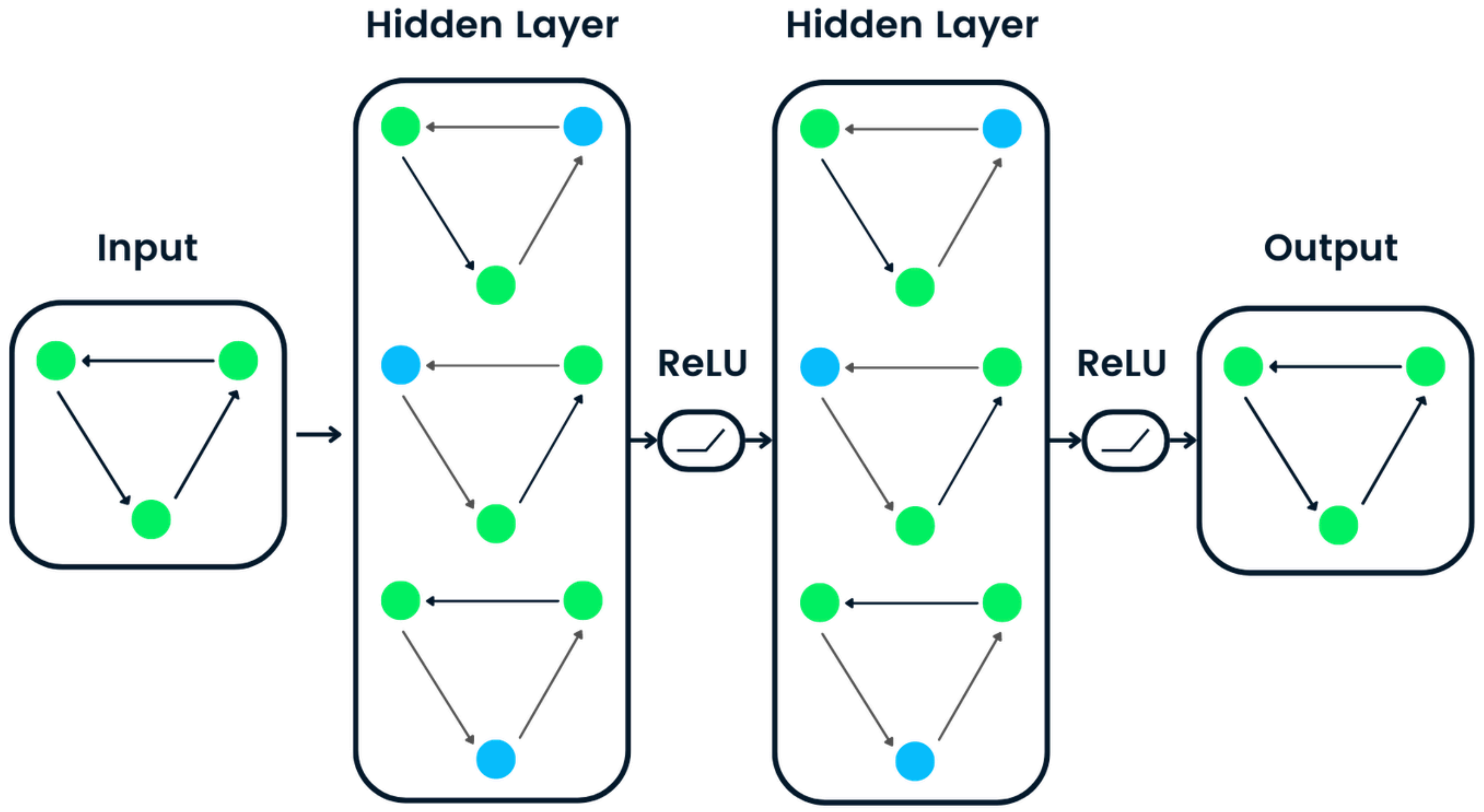
- Generalized density algorithm
- Improved Laplacian
- Used several GNNs to do traffic predictions
    - Comparing to baseline
    - Data not following daily pattern
        - Confusing

- Next?
    - Continue my jet analysis
    - Courses

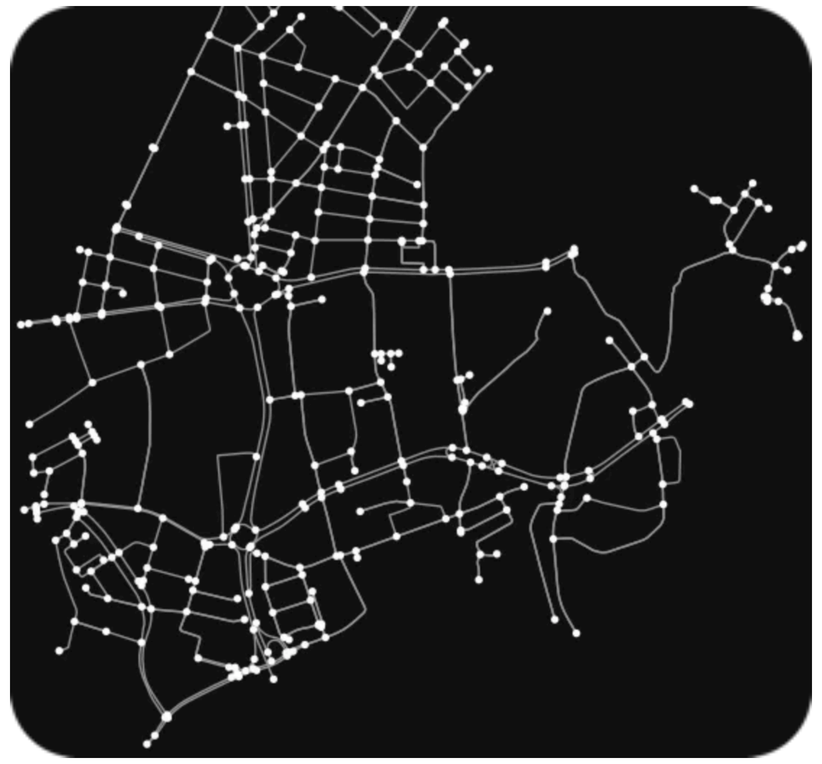
Backup

.. short on GNN

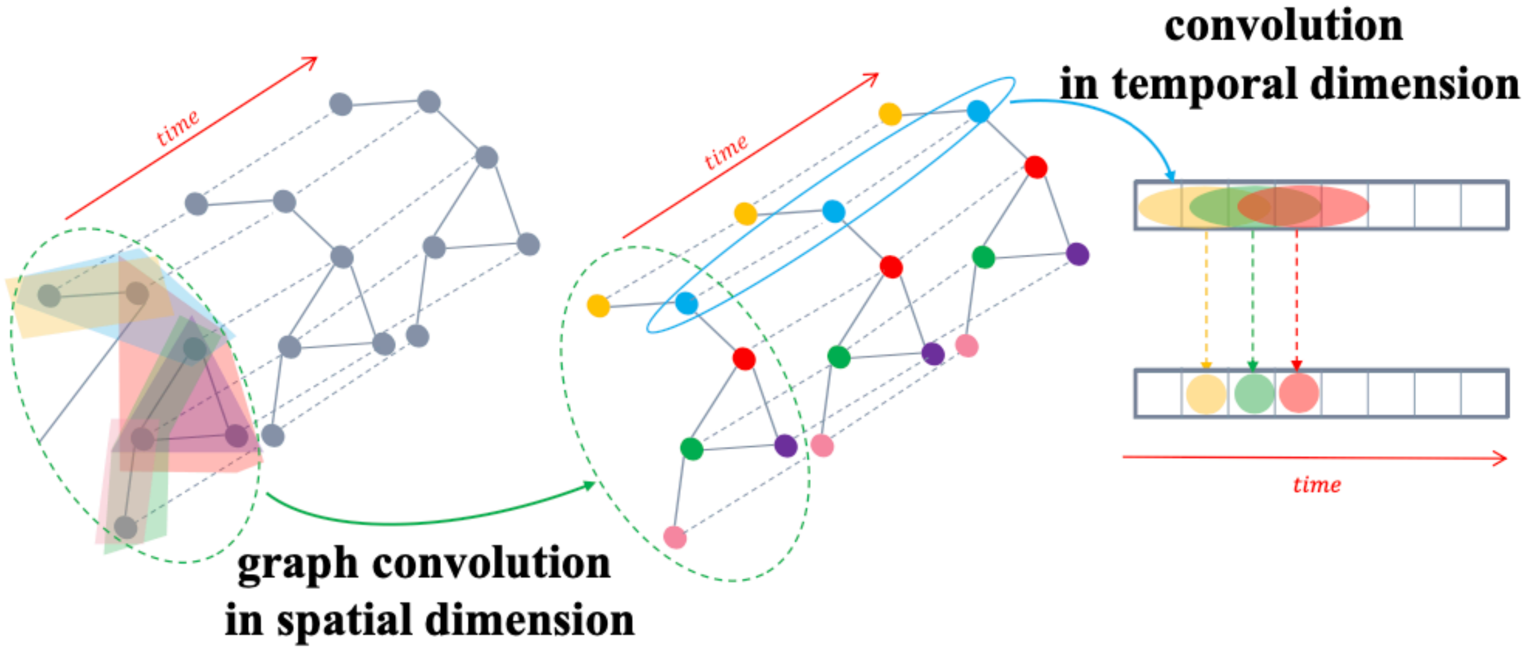
- Real world data does not (always) live on a **grid**
- $G = (V, E)$
  - V: Vertices (nodes)
  - E: Edges (links)
    - Directional (non-directional)
- We use 5 different architectures
- More focus on the spatial information of the cameras



source



source


convolution
in temporal dimension
time
time
time
graph convolution
in spatial dimension