Learning control policies in smart cities from physical data

Master Thesis Final Presentation

Mykhaylo Marfeychuk
Tiago S. Veiga - Advisor
Pedro U. Lima - Co-Advisor

TÉCNICO LISBOA

Table of contents

1. Motivation
2. Environment
   2.1. Traffic Modelling
   2.2. Pollution Modelling
   2.3. Modular Framework
3. Reinforcement Learning
4. Proposed Solution
   4.1. Single Agent
   4.2. Multi-agent
5. Results
   5.1. Agents description
   5.2. Single agent vs Multi-agent
   5.3. Multi-Agent RL vs Reactive Agent
   5.4. Week Simulation

Motivation

Our goal

Optimize traffic flow to reduce traffic related pollution while maintaining the traffic flow. As it is hard to model the world, we'll use Reinforcement Learning to optimize the traffic.

- Realistic simulated environment
- Simulated emissions
- **Effect of the traffic on the emissions**
- **Traffic control for improving the emissions**

What is AI4EU?

AI4EU is a consortium established to build the first European Artificial Intelligence On-Demand Platform and Ecosystem with the support of the European Commission under the H2020 programme.

What are the goals of AI4EU?

Build an European AI platform and Ecosystem. Several activities to foster this, including the implementation of industry-led pilots, and application of research activities in those pilots.

IST is actively involved in the IoT pilot, which gives the motivation and scenario for this thesis

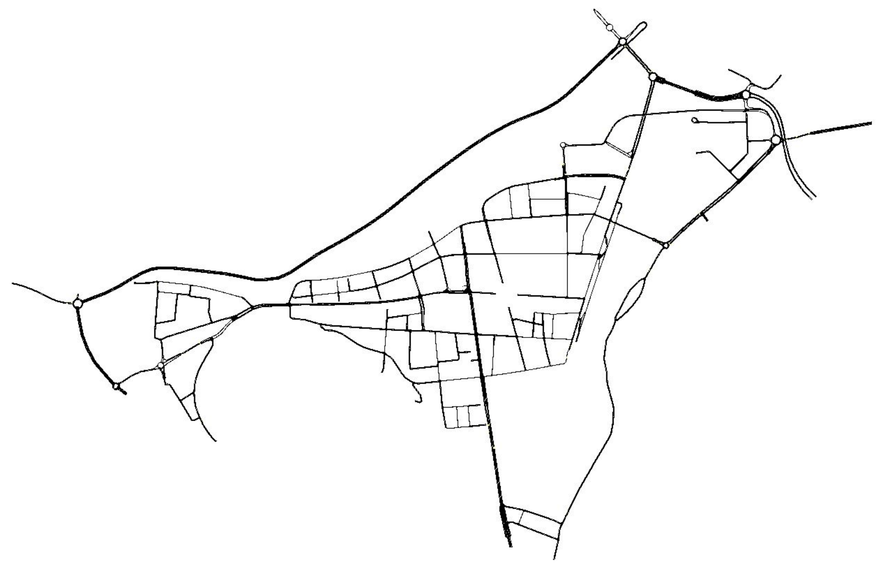
Table of contents

1. Motivation
2. **Environment**
   2.1. **Traffic Modelling**
   2.2. **Pollution Modelling**
   2.3. **Modular Framework**
3. Reinforcement Learning
4. Proposed Solution
   4.1. Single Agent
   4.2. Multi-agent
5. Results
   5.1. Agents description
   5.2. Single agent vs Multi-agent
   5.3. Multi-Agent RL vs Reactive Agent
   5.4. Week Simulation

Map Generation

- SUMO simulator used for traffic simulation
- Map generated from OpenStreetMap + manual cleanup

OSM Web Wizard view



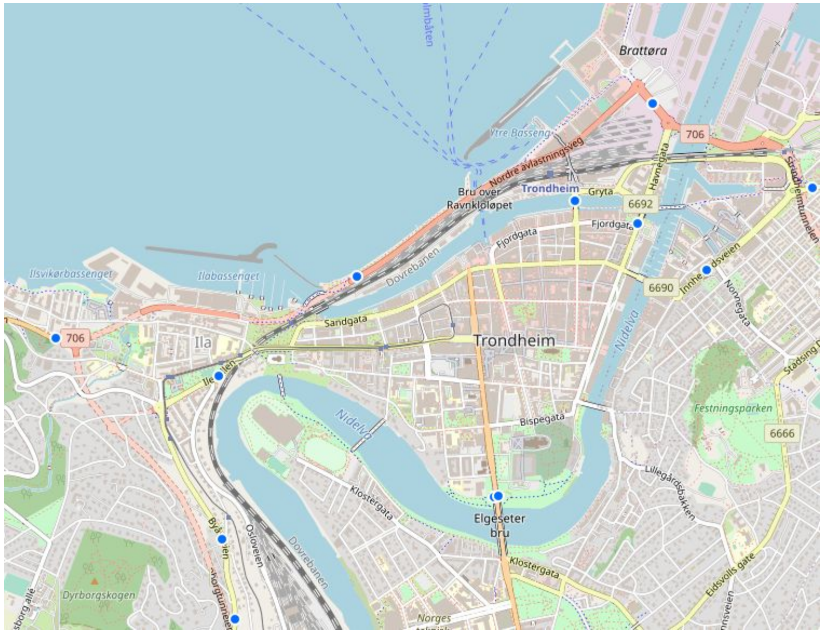
Generated SUMO Simulation Map

Grid System

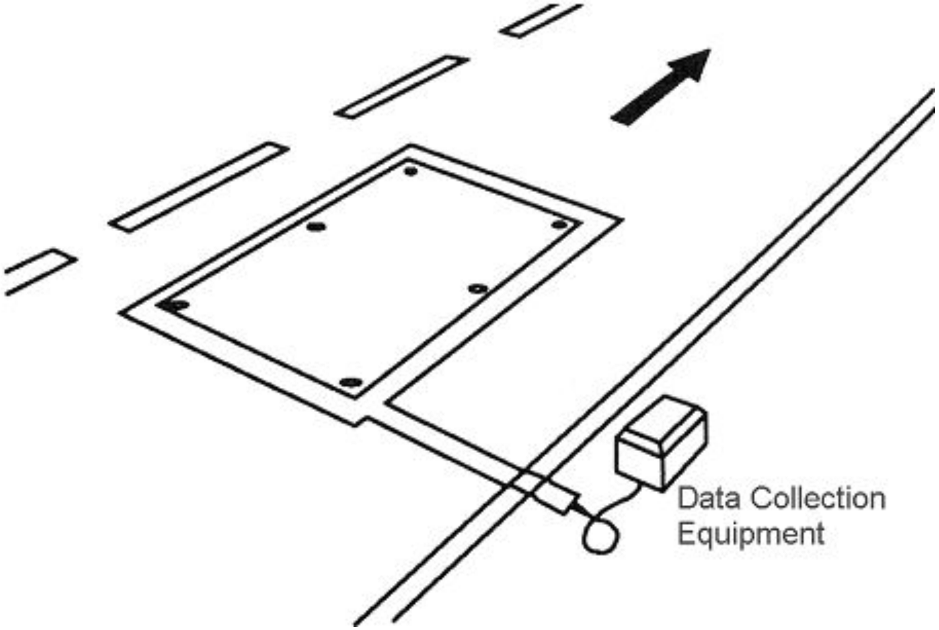
- The map is split into a ***17x16*** grid cell
- A cell is composed of multiple roads
- A road may belong to two different cells
- Additionally each cell tracks the number of vehicles, the cell state (open or closed), emission values, etc.

Traffic Modelling

- Data provided by Norwegian Public Roads Administration
- Data contains induction-loop detections by size
- Real induction-loops matched with the simulated ones
- Traffic generated to match the induction-loop count

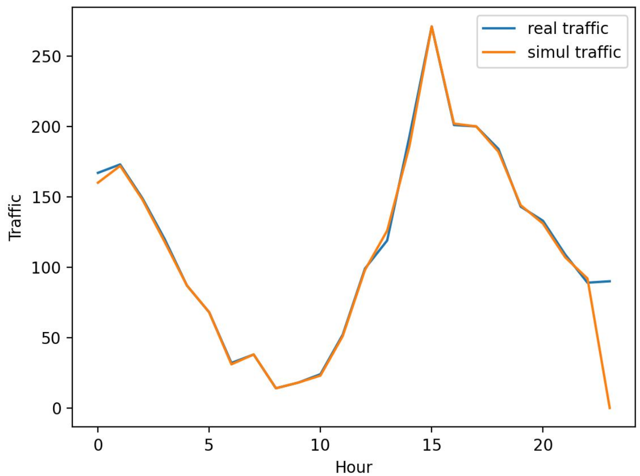


Induction-loop Locations

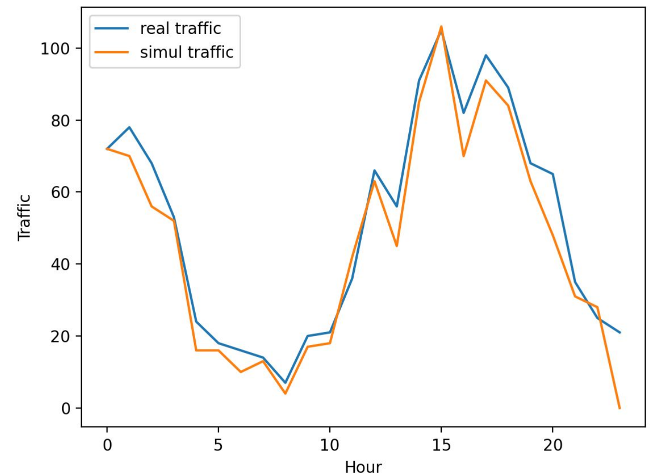


Induction-loop Sensor

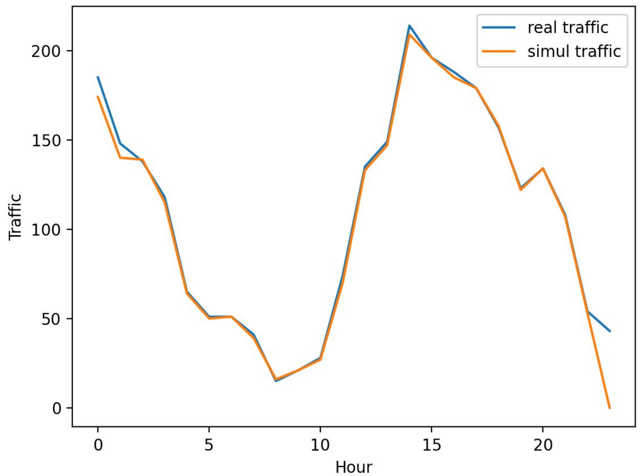
Traffic Modelling



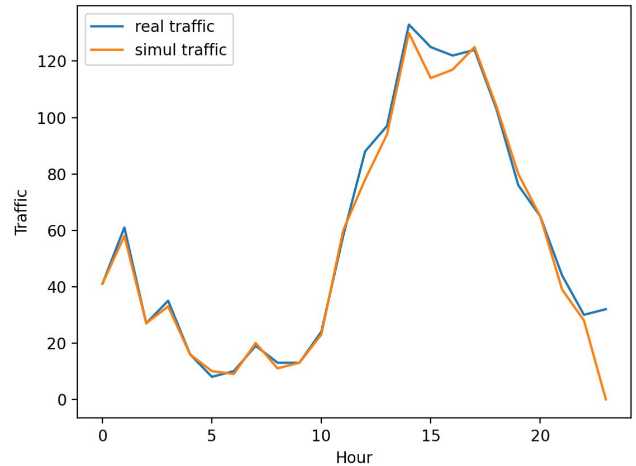
Søndre Ilevollen Sensor, Byåsen Direction



Brattørbrua Sensor, Kjøpmannsgata Direction



Søndre Ilevollen Sensor, Sentrum Direction



Nye Ilsvikøra Sensor, Flakk Direction

Pollution Modelling

- HBEFA v3.1 emissions model was used
- Pollution data provided by Norwegian Institute for Air Research
- Decay and dissipation manually fine-tuned to mimic the data provided by the NILU

Environment State Tracking

- A modular framework was created
- Each experiment lists the modules it needs
- The framework takes care of the lifecycle of the modules
- Each module implements its own functionality

TÉCNICO LISBOA

Framework modules

- **Cells Module:** Represents the map cell matrix and keeps track of all the roads that belong to each cell
- **Emissions Module:** Keeps track of the per cell emission values. Applies emission decay and dissipation
- **Induction Loops Module:** Keeps track of how many vehicles pass through each induction loop
- **Tracking Module:** Tracks statistical information about the simulator
- **Emissions Renderer Module:** Creates a plot visualization of the emission values

Framework representation

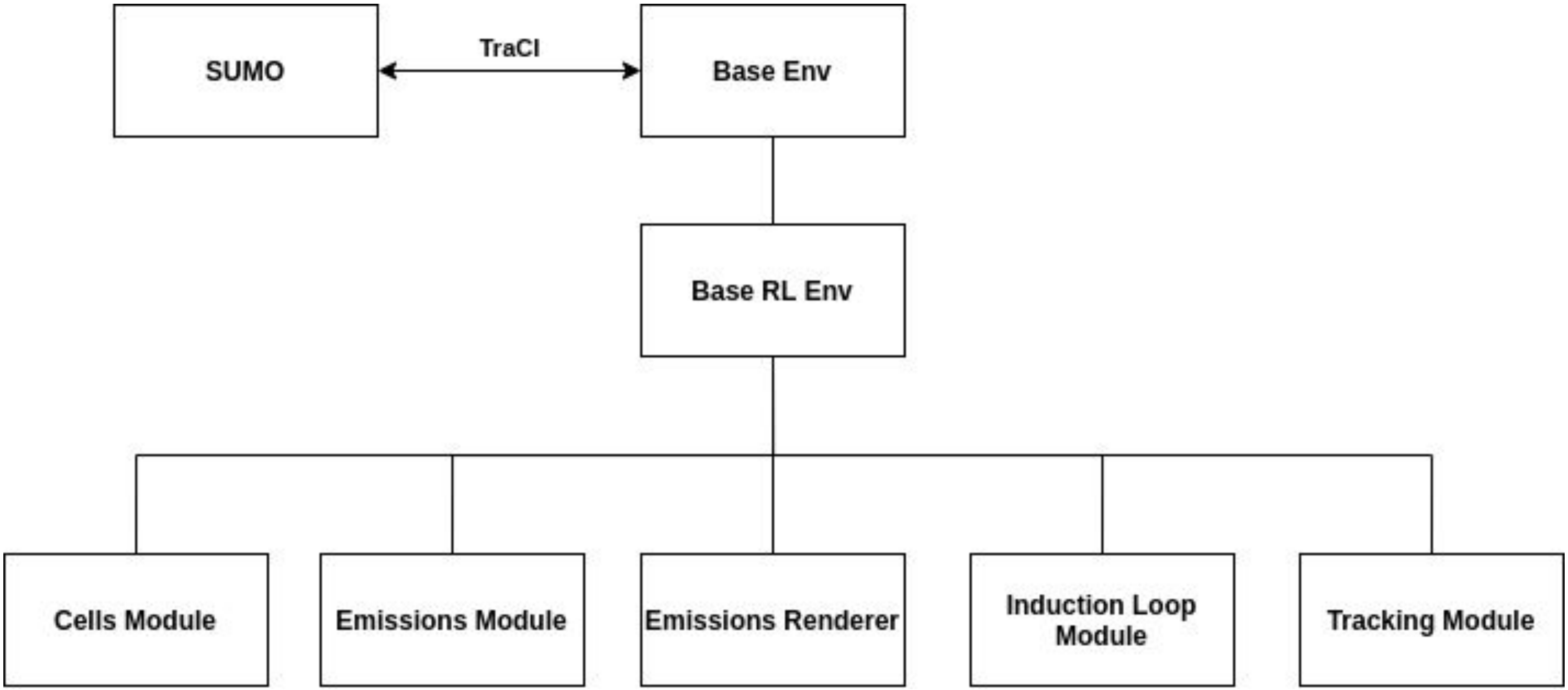
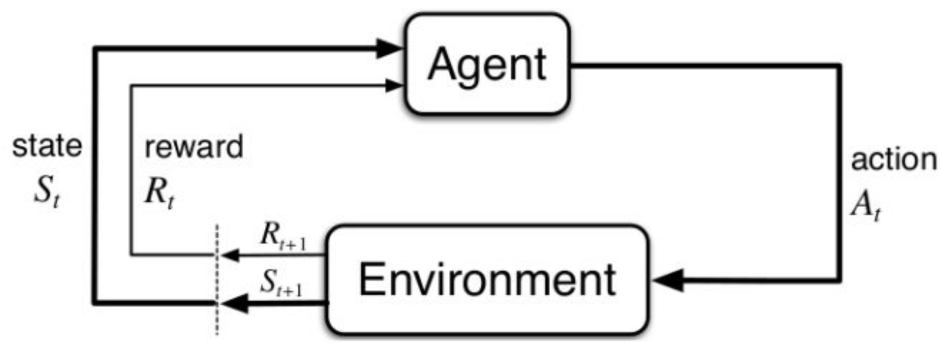


Table of contents

1. Motivation
2. Environment
   2.1. Traffic Modelling
   2.2. Pollution Modelling
   2.3. Modular Framework
3. **Reinforcement Learning**
4. Proposed Solution
   4.1. Single Agent
   4.2. Multi-agent
5. Results
   5.1. Agents description
   5.2. Single agent vs Multi-agent
   5.3. Multi-Agent RL vs Reactive Agent
   5.4. Week Simulation

Reinforcement Learning

- Agent performs actions, experiences rewards
- Goal is to learn an optimal policy, which when given a state, chooses the best action to maximize long-term reward



Reinforcement Learning

Curse of dimensionality:

- In tabular reinforcement learning every state-action pair needs to be visited enough times. In problems with a large number of states, convergence is infeasible

Reinforcement Learning

Curse of dimensionality:

- In tabular reinforcement learning every state-action pair needs to be visited enough times. In problems with a large number of states, convergence is infeasible

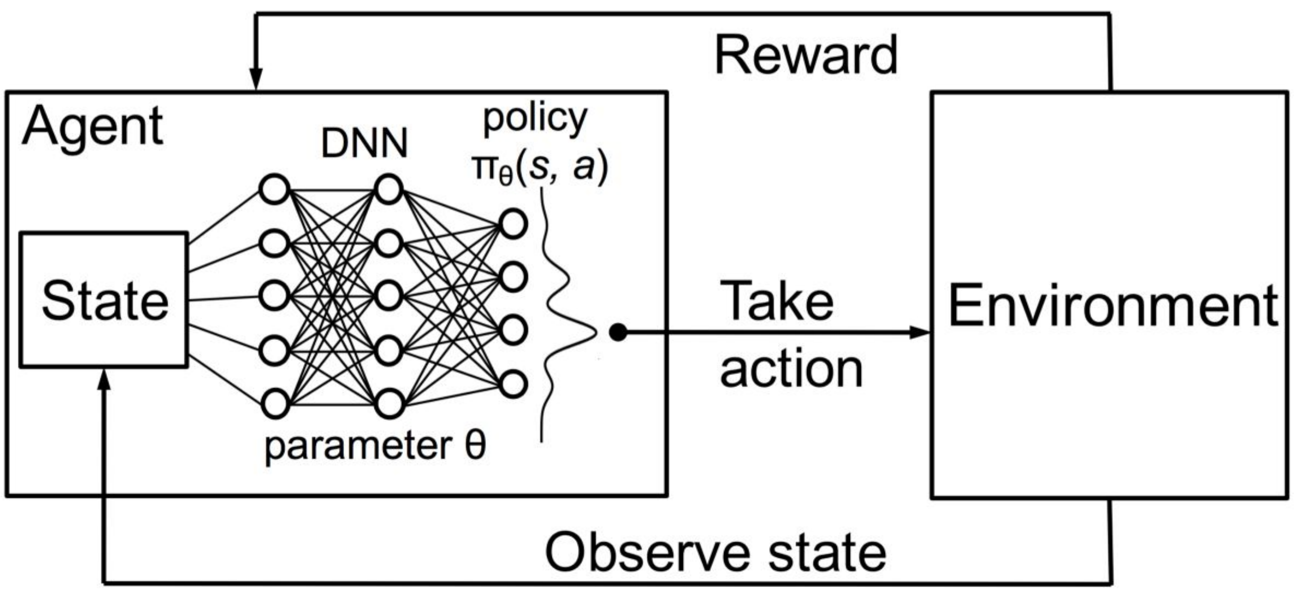
↓

This is where Deep Reinforcement Learning comes in

Deep Reinforcement Learning

Deep Q-Learning:

- States represented as features for a Deep Neural Network
- The Network estimates the Q-value function
- Experience Replay technique is used to reduce experience prioritisation



Advantage Actor-Critic (A2C)

- Uses actor-critic model
- Actor-critic can have parts of their networks in common to guarantee the same feature correlation
- Parallelizable training

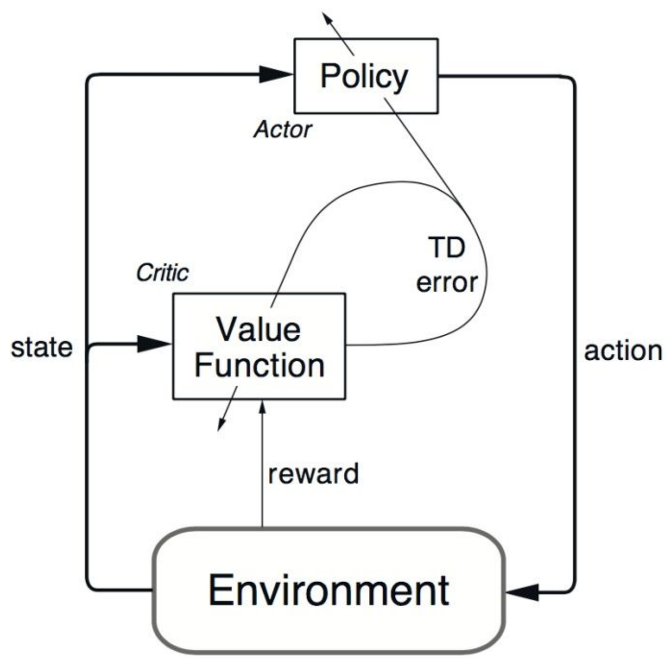
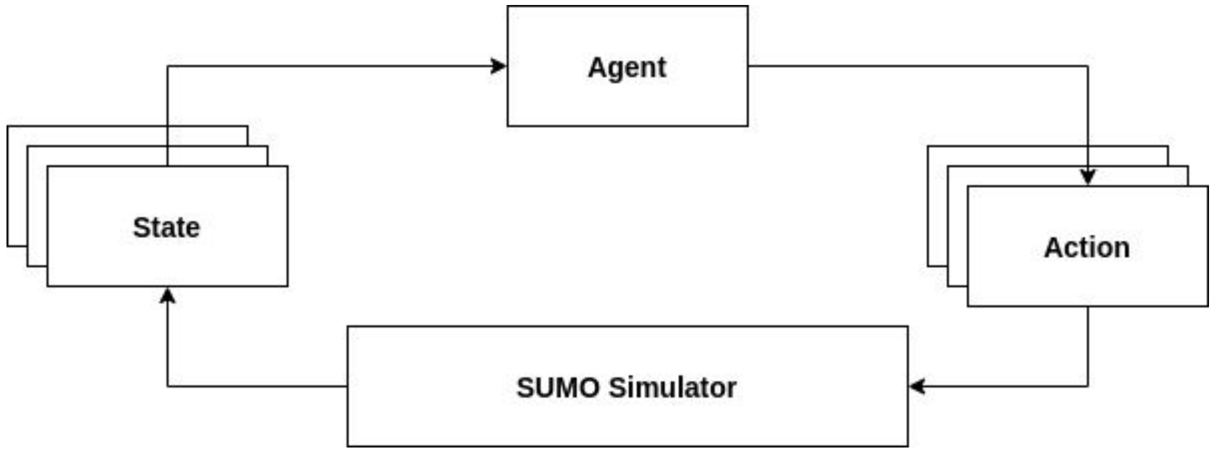


Table of contents

1. Motivation
2. Environment
   2.1. Traffic Modelling
   2.2. Pollution Modelling
   2.3. Modular Framework
3. Reinforcement Learning
4. **Proposed Solution**
   4.1. Single Agent
   4.2. Multi-agent
5. Results
   5.1. Agents description
   5.2. Single agent vs Multi-agent
   5.3. Multi-Agent RL vs Reactive Agent
   5.4. Week Simulation

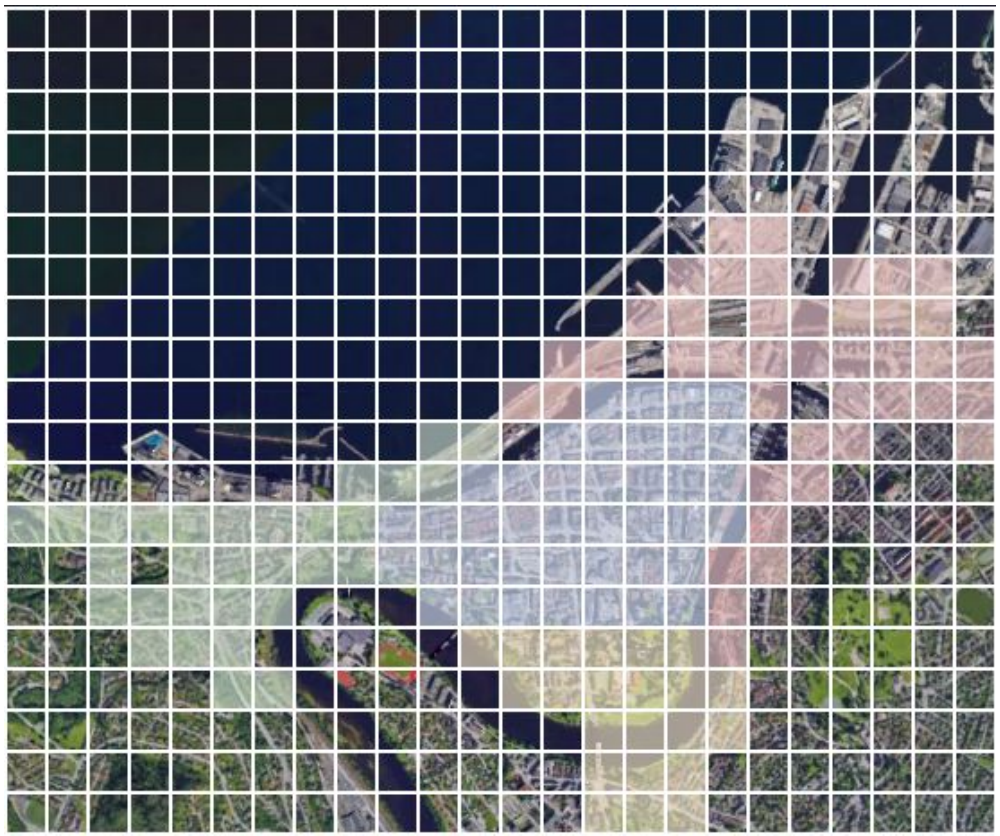
Problem Solution

- The agent iterates over each cell
- For each cell the agent decides if the cells should be open or closed
- **Problem:** The agent doesn’t take the state of the neighbouring cells into account
- **Solution:** Provide a general overview of all the cells



Multi-Agent Framework Proposal

- Split the map into 4 regions
- Each region has its own agent
- The agents have to learn to cooperate



Input and Actions

Matrix Input (17x16x4):

- **Cell Emissions** (same for all cells)
- **Number of Vehicles** (same for all cells)
- **Cell closing State** (same for all cells)
- **Action Cell** (different for each cell decision)

Categorical data (1x96):

- **Action number / Time of day**

Actions:

- **Open** the cell
- **Close** the cell

Agent's Model

- Convolutional Layers are used to find correlation between the cells
- Adding more than three convolutional layers increases the training time but doesn't improve the performance
- The categorical data is added to a dense layer

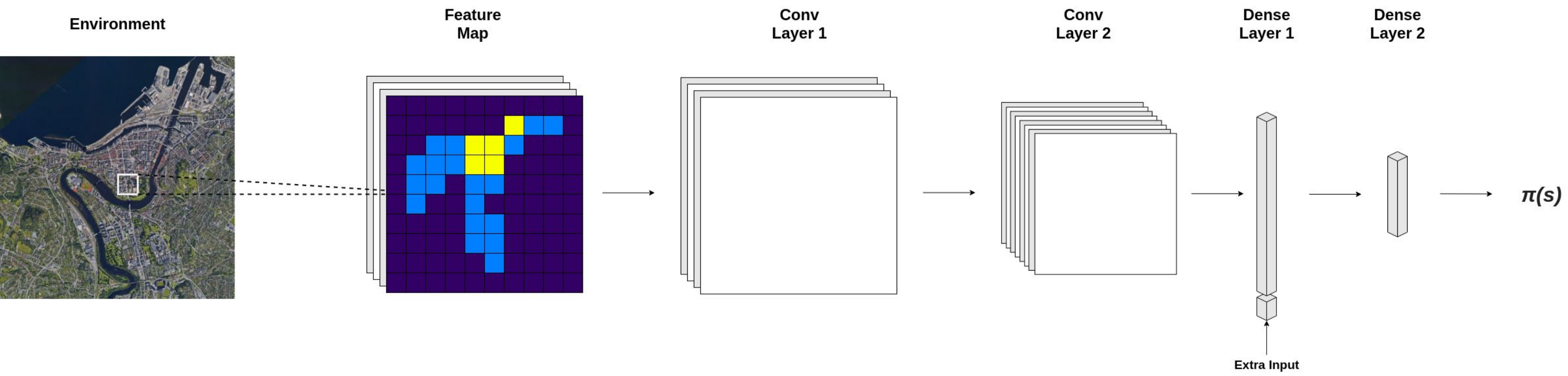


Table of contents

1. Motivation
2. Environment
   2.1. Traffic Modelling
   2.2. Pollution Modelling
   2.3. Modular Framework
3. Reinforcement Learning
4. Proposed Solution
   4.1. Single Agent
   4.2. Multi-agent
5. **Results**
   5.1. **Agents description**
   5.2. **Single agent vs Multi-agent**
   5.3. **Multi-Agent RL vs Reactive Agent**
   5.4. **Week Simulation**

RL Agent

- **Goal:** Minimize emissions + maintain number of arrived vehicles
- **NOx thresholds:** 50 & 100 (μg/m^3)
- **Reward:**

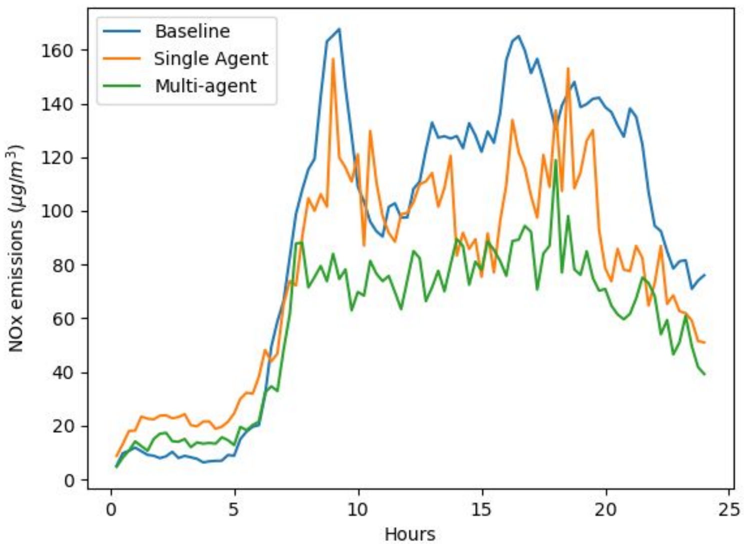
$$reward = \frac{(t - mce)}{200} + \frac{av}{eav}$$

where:

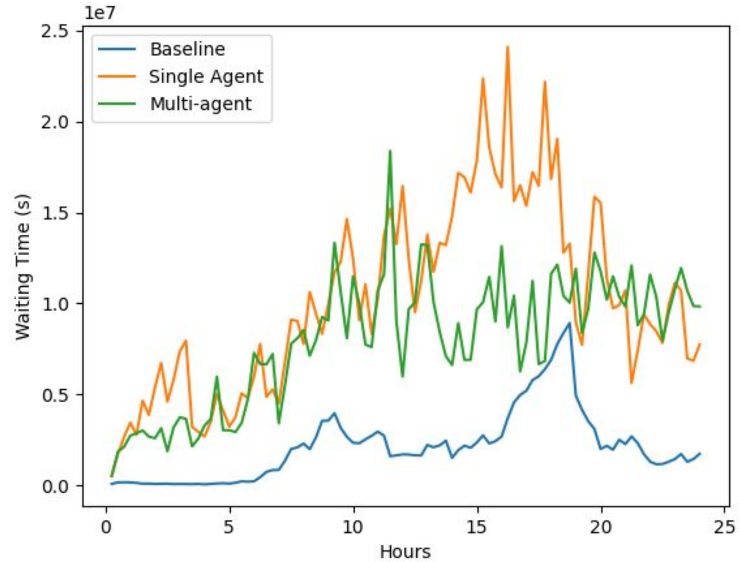
$$\begin{aligned} t &= \text{emissions threshold} \\ mce &= \text{current highest cell emission} \\ eav &= \text{expected step arrived vehicles} \\ av &= \text{step arrived vehicles} \end{aligned}$$

Single agent vs Multi-agent

- **NOx thresholds:** 100 (μg/m^3)



Max Cell Emissions Value



Waiting Time

- Multi-agent the emissions more than the single agent
- Multi-agent causes less waiting time

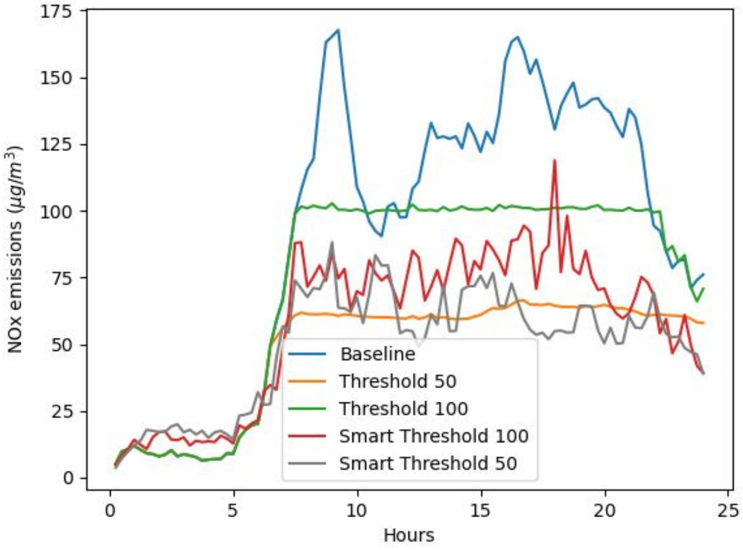
Reactive Agent

- **Goal:** Close cells that exceed the threshold
- **NOx thresholds:** 50 & 100 (μg/m^3)
- **Pseudocode:**

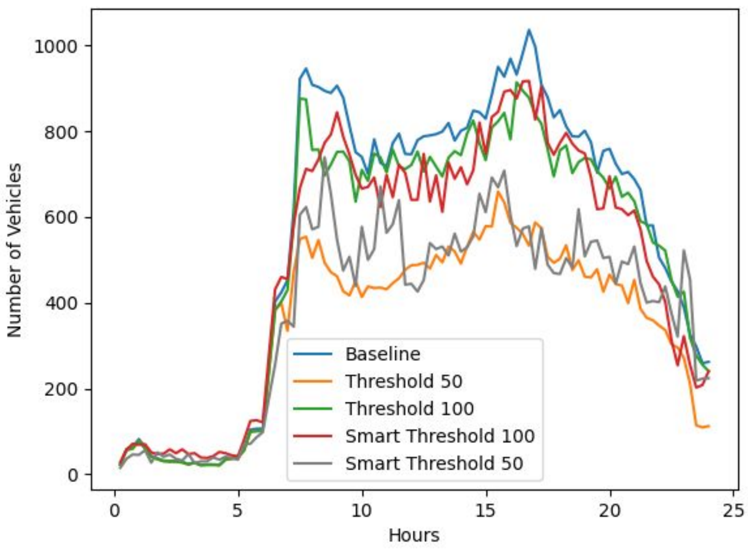
```
procedure REACTIVE_AGENT(threshold, cells)
    for cell in cells do
        get cell emissions
        if cell_emissions >= threshold then
            Close cell
        else
            Open cell
```

Multi-Agent RL vs Reactive Agent

- **NOx thresholds:** 50 & 100 ($\mu g/m^3$)



Max Cell Emissions Value

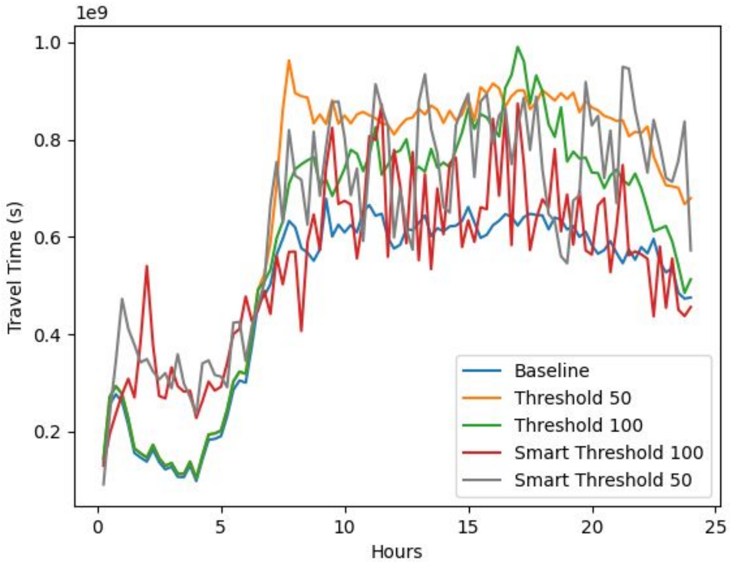


Vehicles Arrived to Destination

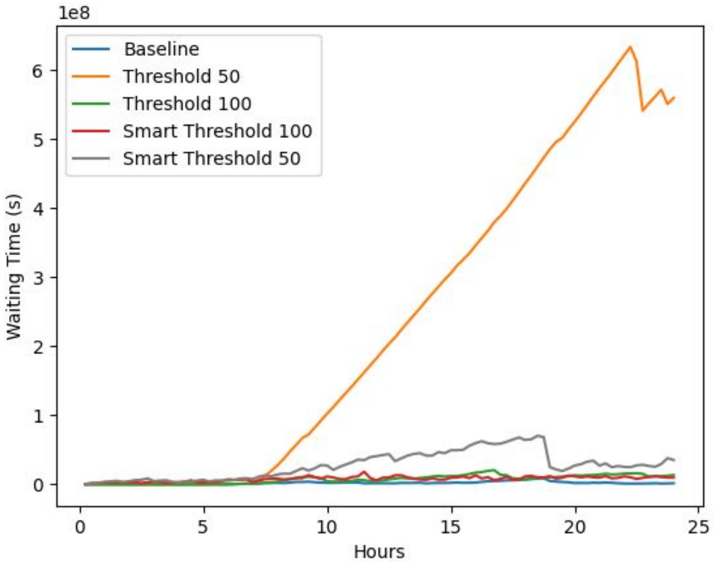
- Multi-agent 100 Threshold lowers the emissions more than the respective Reactive Agent
- For 50 Threshold both agents behave similarly

Multi-Agent RL vs Reactive Agent

- **NOx thresholds:** 50 & 100 ($\mu g/m^3$)



Travel Time

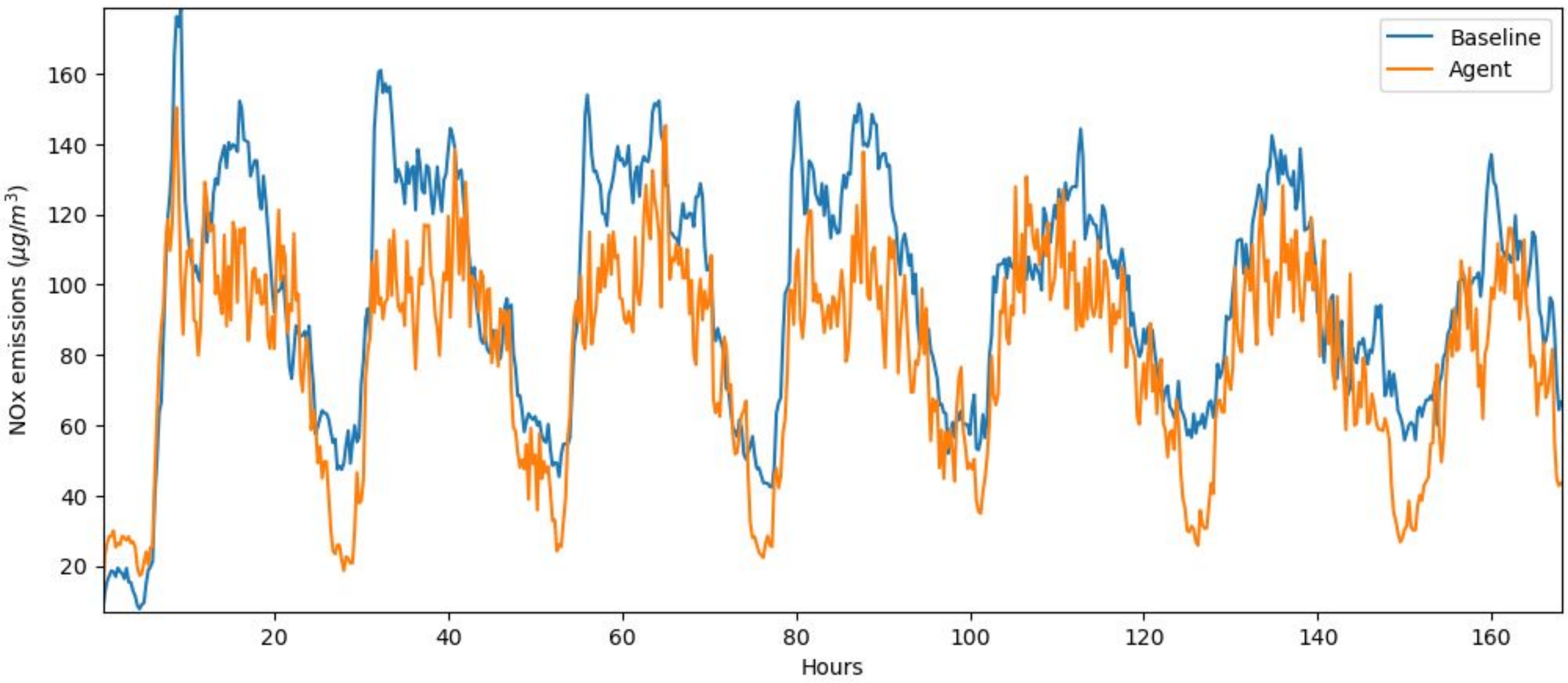


Waiting Time

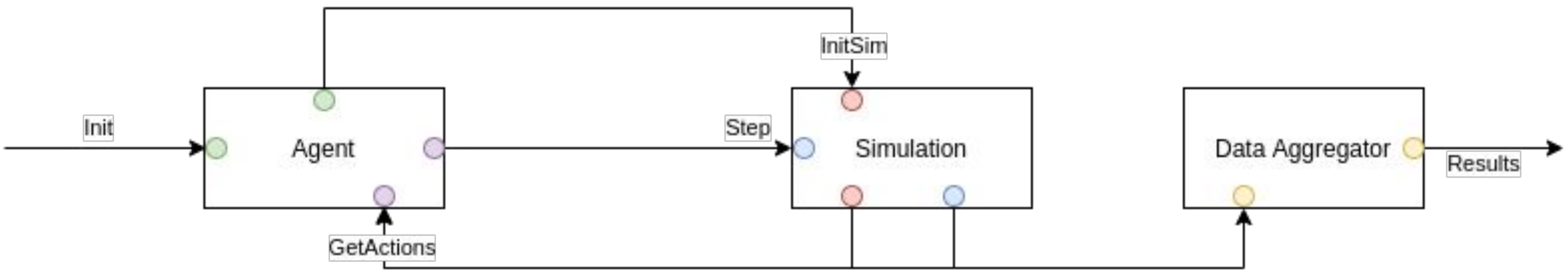
- Multi-agent has lower travel time compared to the Reactive Agent

- Reactive Agent causes traffic jams

Week Simulation Test



Acumos Experiment



Future Improvements

- Rewrite in C++
- Replace the Tensorflow framework by PyTorch
- Improve emissions modelling
- Use Offline RL to improve the training

Thank you!

Questions?