UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Corso di Big Data Architecture
Prof. Nesi

Using Deep Reinforcement Learning for Traffic Light Optimization

Presenter: Zahra Fereidooni
University of Florence, DINFO dept, DISIT Lab, https://www.disit.org

December 2024

Agenda

1. Introduction
2. Reinforcement Learning Approach
3. Proposed Solution
4. Analysis Results
5. Conclusion

Introduction

Introduction

Multiple approaches to optimizing traffic light cycles (Eom & Kim, 2020).

- Analytical (e.g., Linear Programming, Bayesian Optimization).
- Rule-based (e.g., Fuzzy Logic Systems).
- Heuristic (e.g., Genetic Algorithms).
- Simulation-based (e.g., Traffic Simulation Tools).
- Machine Learning (e.g., Reinforcement Learning, Supervised Learning).

Introduction

Gaps in Existing Methods

- **Public Transport**: Limited focus on tramways and urban mobility.
- **Fairness**: Lack of equitable systems for diverse travel needs.
- **Scalability**: Inefficiency across multiple intersections.
- **Cost-effectiveness**: High costs of hardware upgrades.

Using **Machine Learning** to tackle these issues

Machine Learning

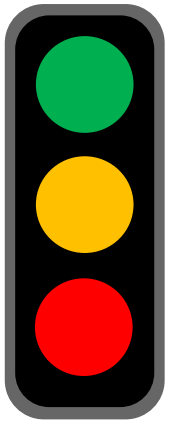
Three paradigms of AI learning.

Supervised learning

Data:(x, y)

x is data and y is label

Goal: learn from the function to map



Based feature and historical labeled data

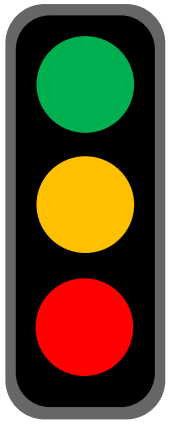
Green-Time = 30

Unsupervised learning

Data:(x)

x is data and no lable

Goal: predict the cluster by using the featuers



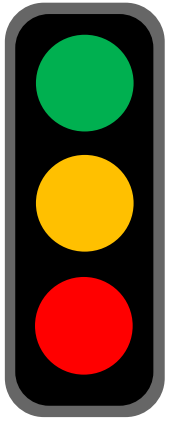
Based features it can be in this range

Green-Time = (28,32)

Reinforcement Learning

Data: state-action pairs

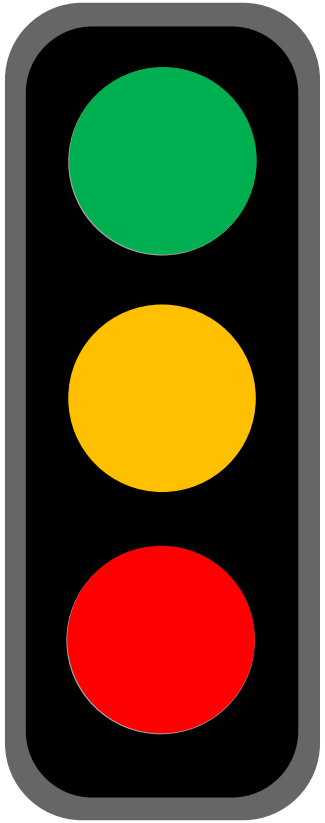
Maximize future rewards after many steps



For this state observed from the environment, it is better to set

Green-Time = 31

Reinforcement Learning Approach



Reinforcement Learning Approach

What is it?

Reinforcement Learning Approach

How can an Intelligent Agent **learn** to make a **good sequence of decisions?**

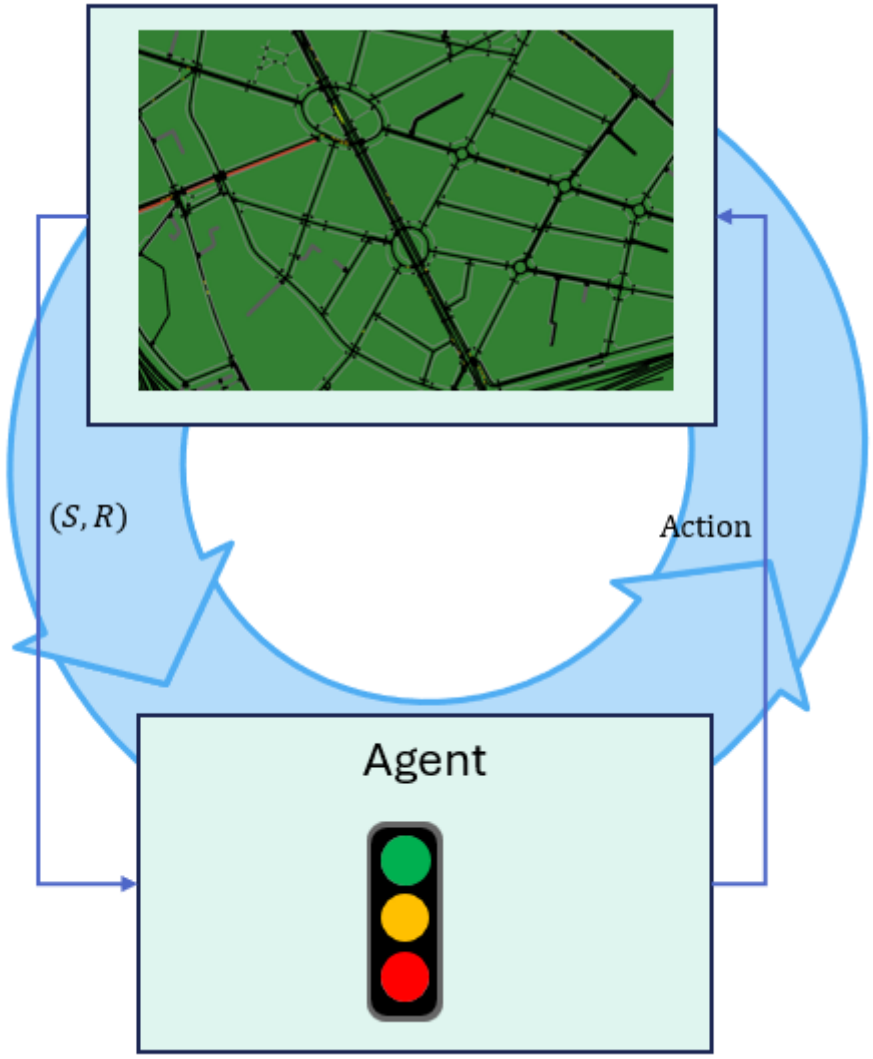
Key aspects of RL

1. Optimization
2. Delayed consequences
3. Exploration
4. Generalization

Reinforcement Learning (RL)

Observation

state: s_{t+1}
Reward = r_t

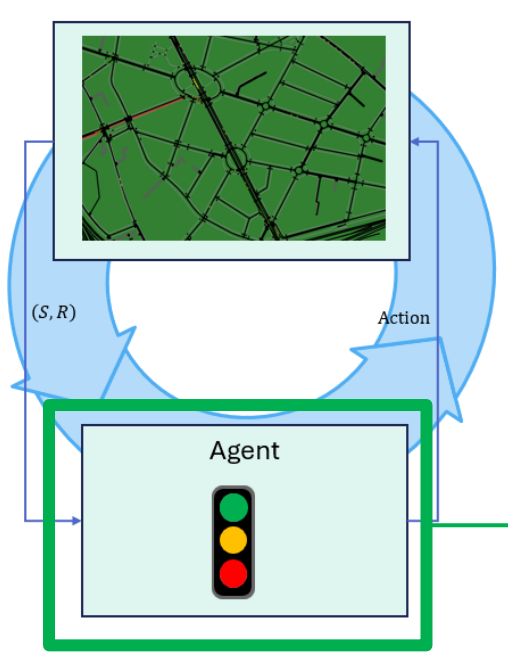


Action

Action: a_t
Select between $[0, \infty]$

Reward: feedback that measure the success of failure of the agent's action

Agent



Model-Free learning by trial and error

Model-based use Model to predict future

Agent componets

Models: Representation of how the world changes in response to the agent's action

$$P(s_{t+1} = s' | s_t = s_1 a_t = a)$$

Policy: function mapping agent state to the action (agent behavior)

Value function: future rewards from being in the state and/or action when following the particular policy

Reward

Reward is feedback that measure the success of failure of the agent's action

$$Total\ Reward = \sum_{i=t}^{\infty} r_i$$

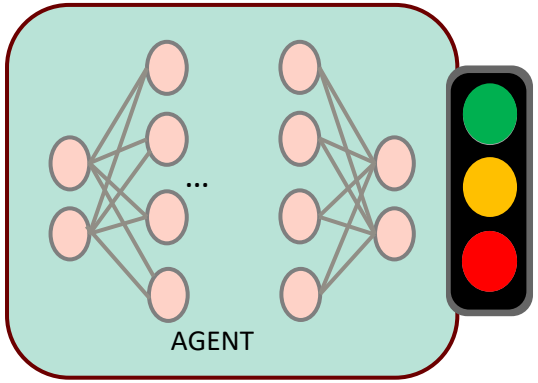
$$Discount\ and\ Total\ Reward = \sum_{i=t}^{\infty} \gamma^i r_i = \gamma^1 r_1 + \gamma^2 r_2 + \cdots + \gamma^\infty r_\infty$$

$\gamma\ is\ the\ discont\ multiple\ to\ emphesize\ now\ rather\ than\ future\ 0 < \gamma < 1$

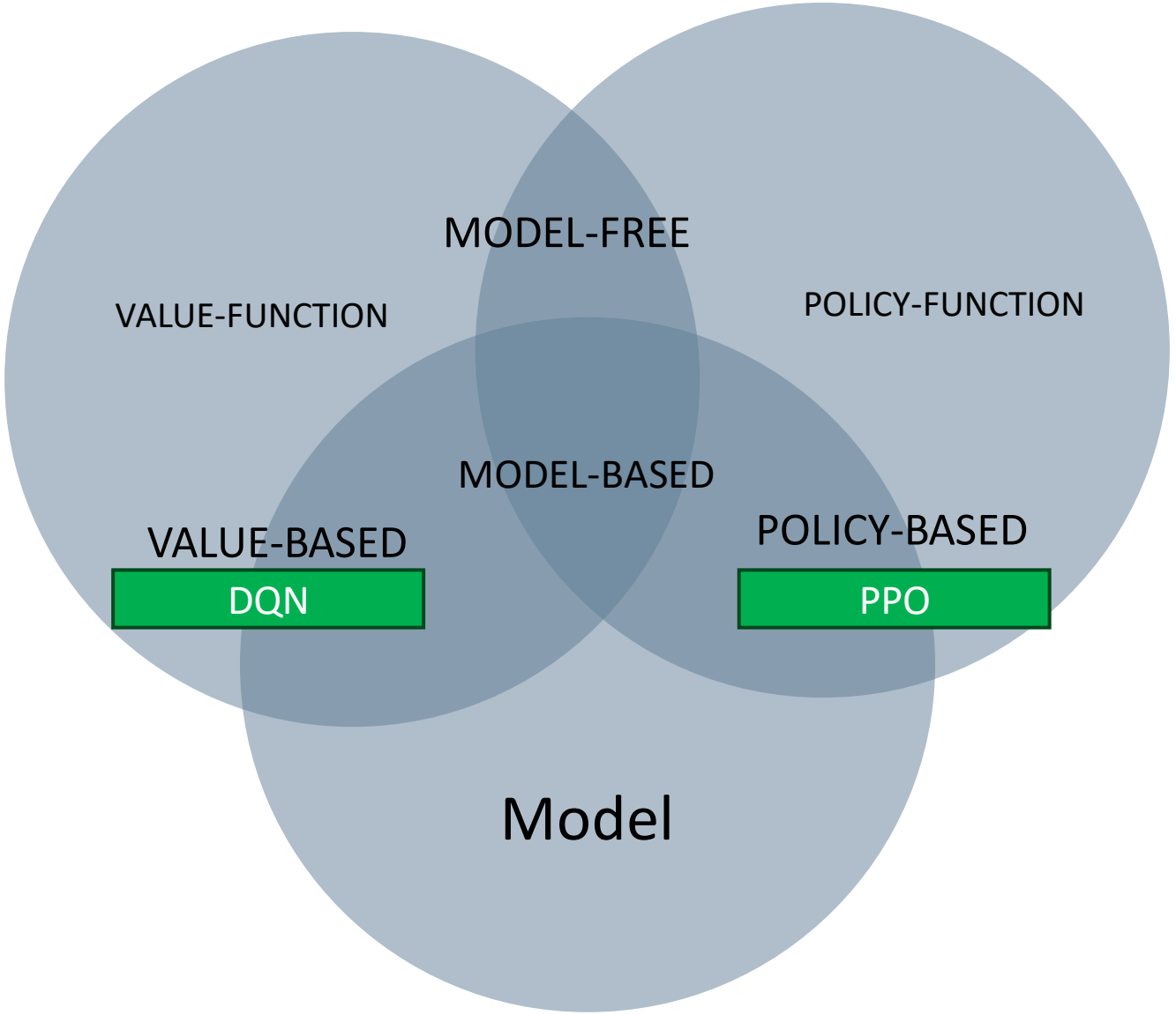
γ = 0.5

Deep Reinforcement Learning?

A Reinforcement Learning (RL) model is considered **deep** when it utilizes **deep neural networks (DNNs)** as function approximators within its learning process.



Agents


MODEL-FREE
VALUE-FUNCTION
POLICY-FUNCTION
MODEL-BASED
VALUE-BASED
DQN
POLICY-BASED
PPO
Model

Q-Function

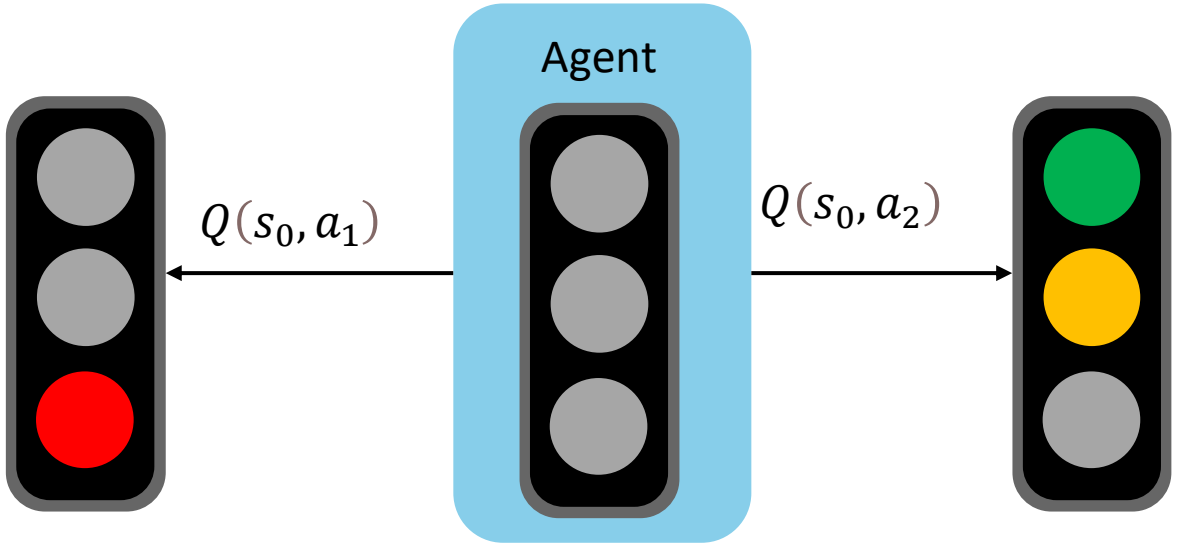
$$Q(s_t, a_t) = E[R_t | s_t, a_t]$$

$(state, actin)$

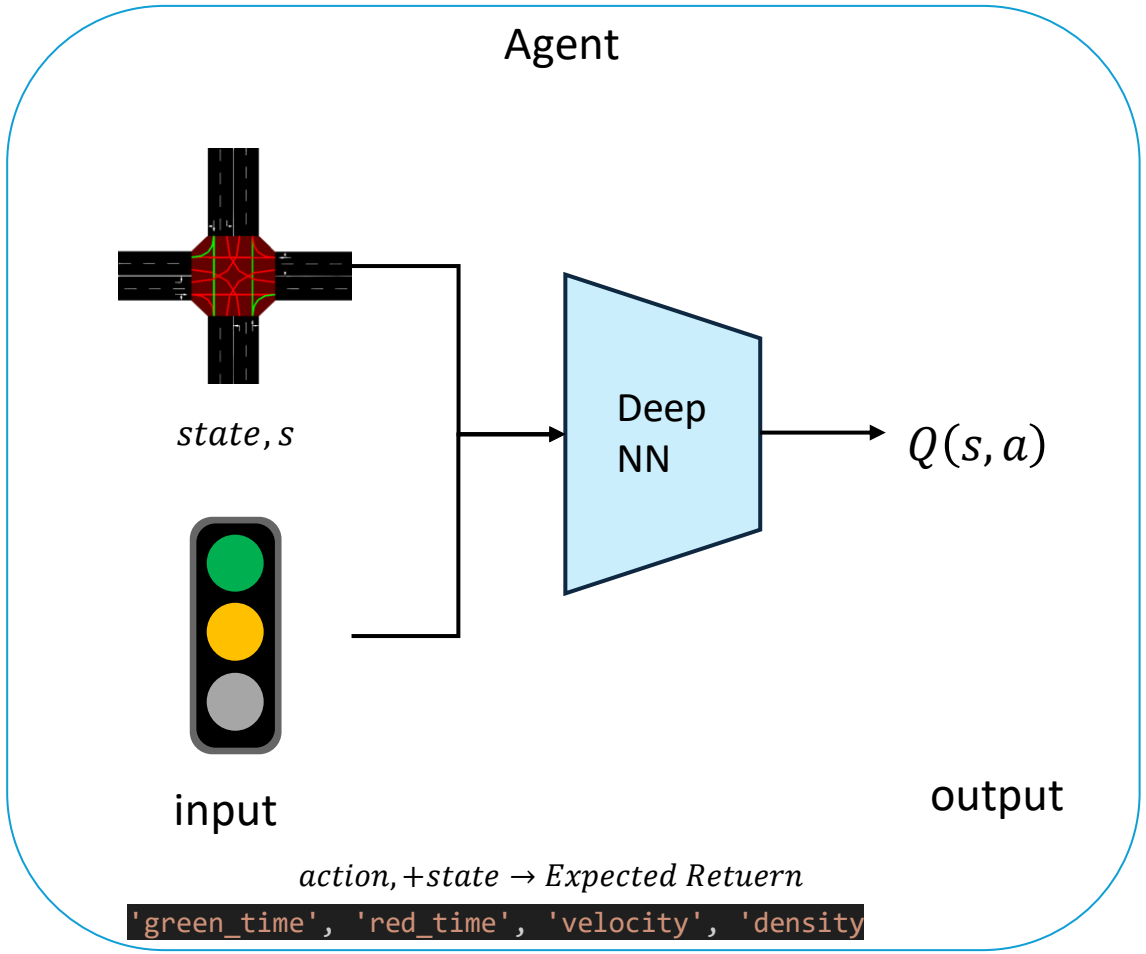
Using this expected function to find out what is the best action that we can take

The inputs are the current state and the action so for each action find out the reward and find the best results for the future selection

Strategy: The policy should choose an action that maximize future reward

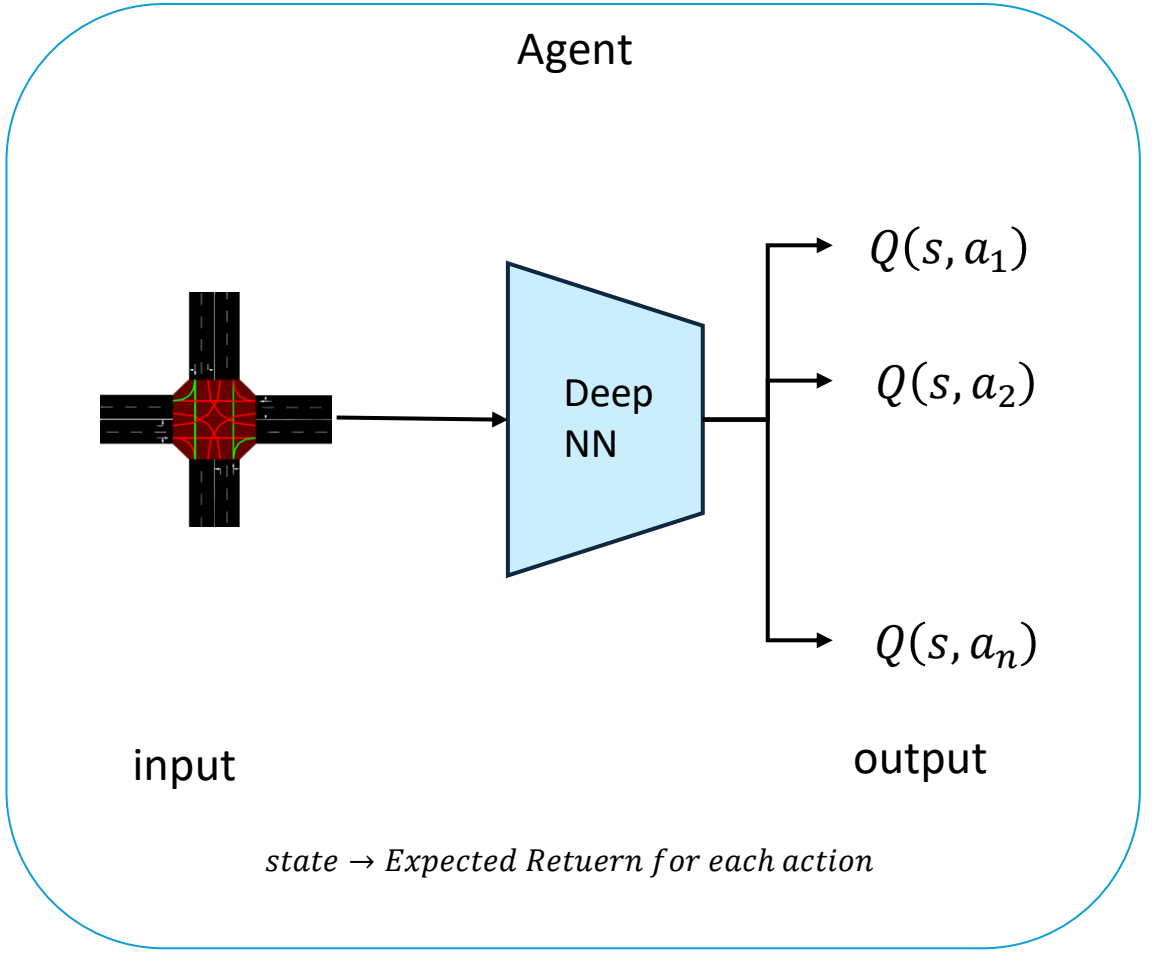


Deep Q Network



Little action space
after every action, NN run

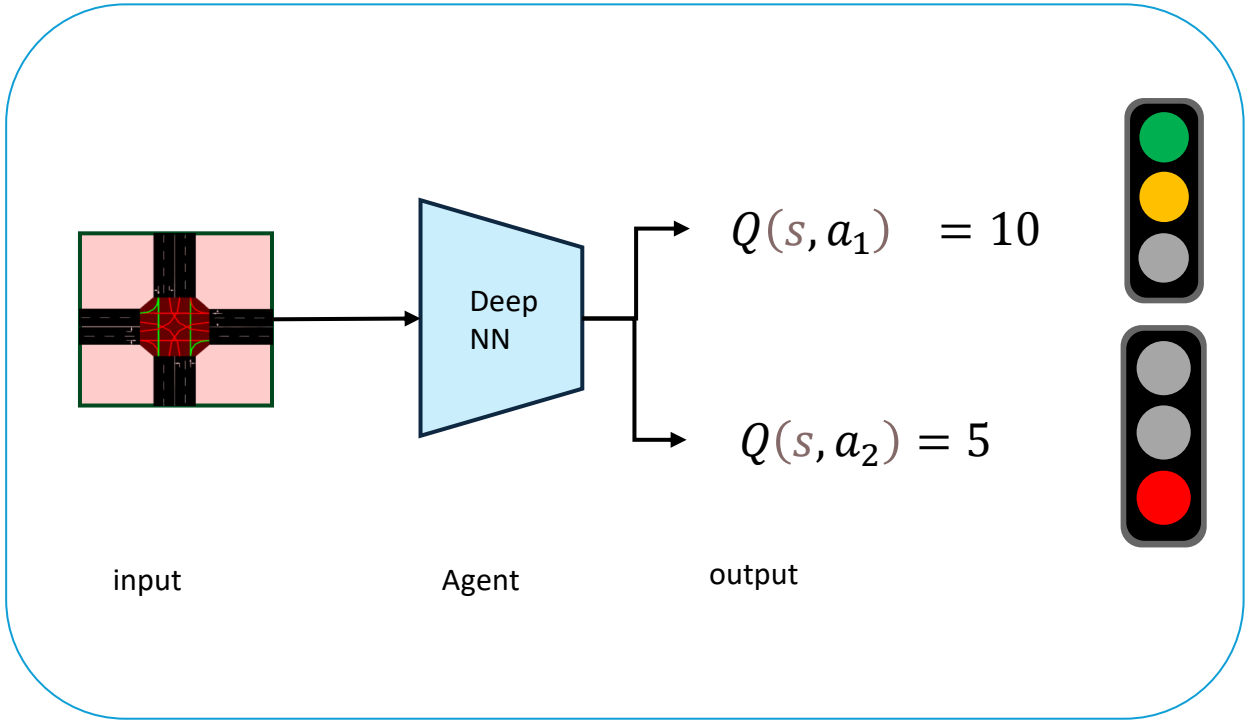
$action, a = green\ time\ or\ red\ time$



Large action space
NN works once for all Q

Deep Q Network

More efficient



$state \rightarrow Expected\ Retuern\ for\ each\ action$

$$Loss\ function = r + \gamma \max\big(Q(s,a)\big) - Q(s,a)$$

We should find a good loss-function for the NN to find out what is the best-case scenario then

Maximize the target $r + \gamma \max(Q(s,a))$ then the predicted $Q(s,a)$ calculated so the best-case scenario comes from this formula

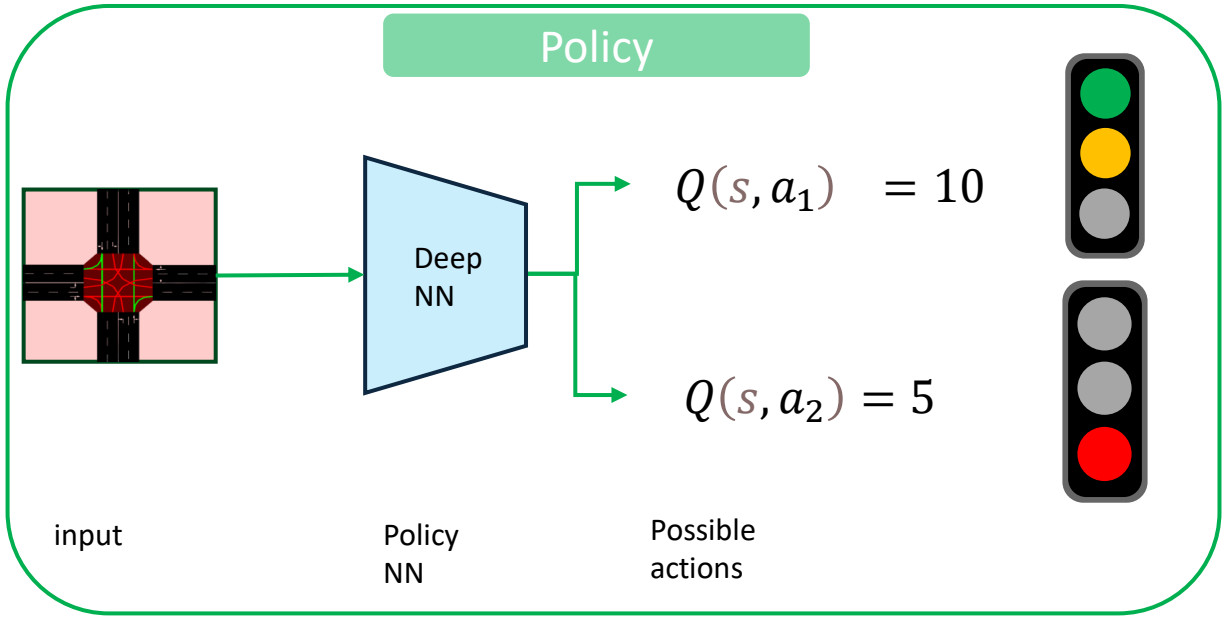
PPO

Proximal Policy Optimization (PPO)-2017

family of policy gradient methods for reinforcement learning

Scalable to large methods and parallel implementations, data efficient and robust.

successful on variety of problem without hypermeter tuning



Data Store

$(s, a_1, reward, possiblity)$

$(s, a_2, reward, possiblity)$

$(s, a_3, reward, possiblity)$

$(s, a_4, reward, possiblity)$

$(s, a_5, reward, possiblity)$

$(s, a_6, reward, possiblity)$

...

$(s, a_{40}, reward, possiblity)$

Compute Loss

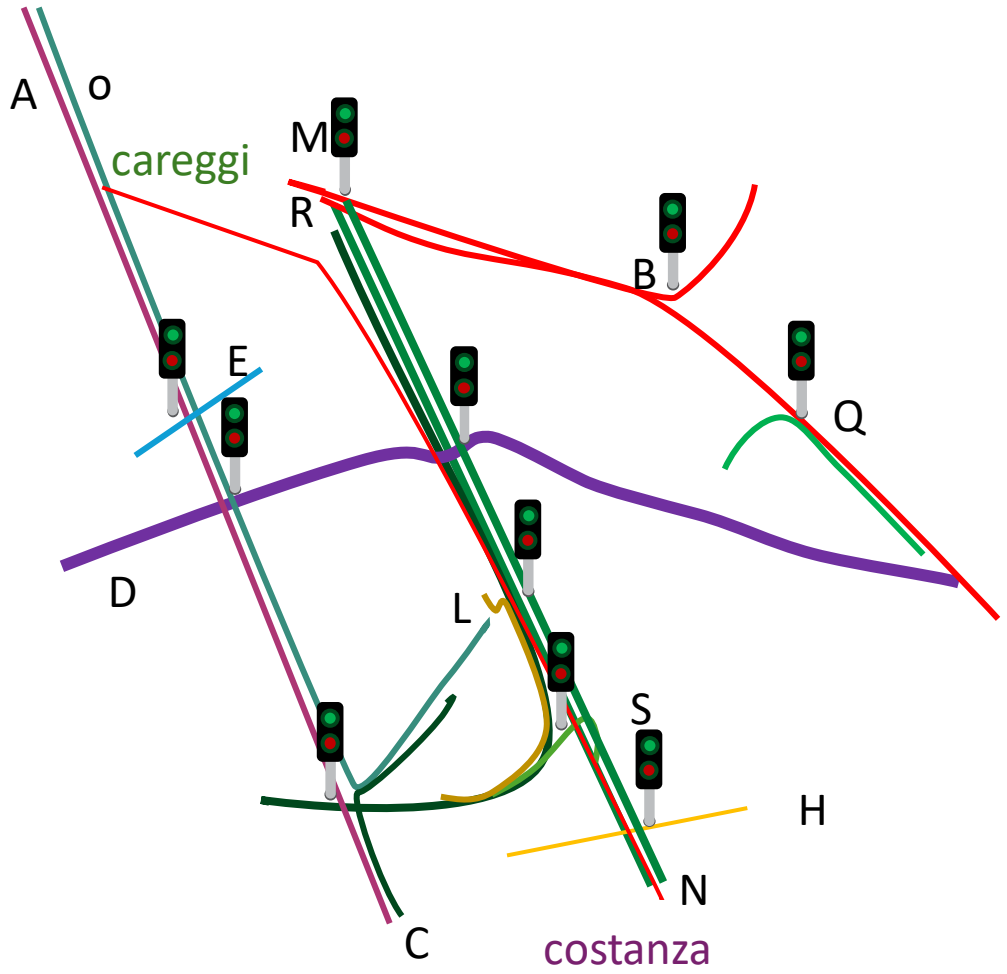
$$r_t(\theta) \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$$

$Advantages\ calculte$

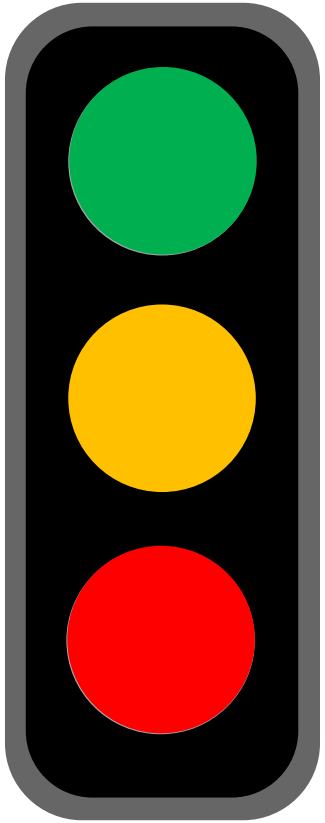
$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right].$$

After Loss calculated the back Propagation happened

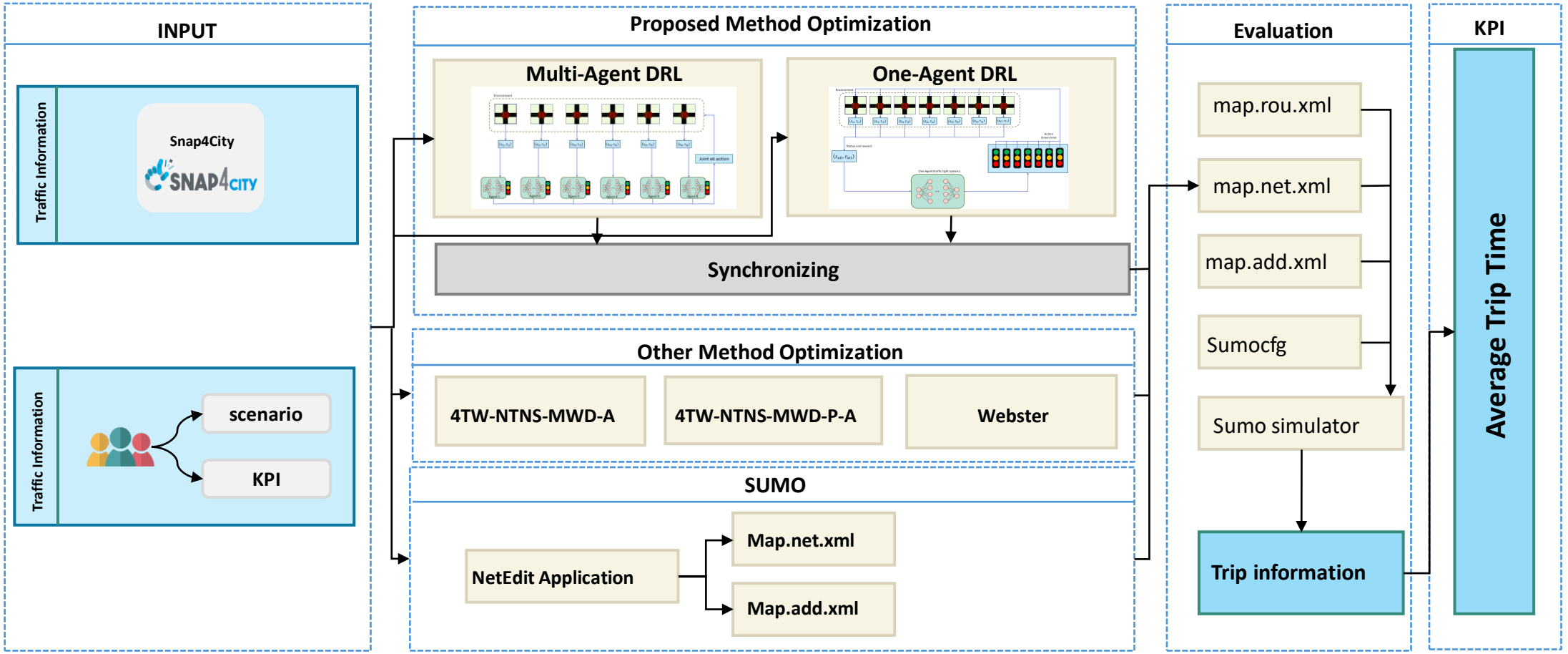
Fixed vs Actuated Traffic light



Proposed Solution…

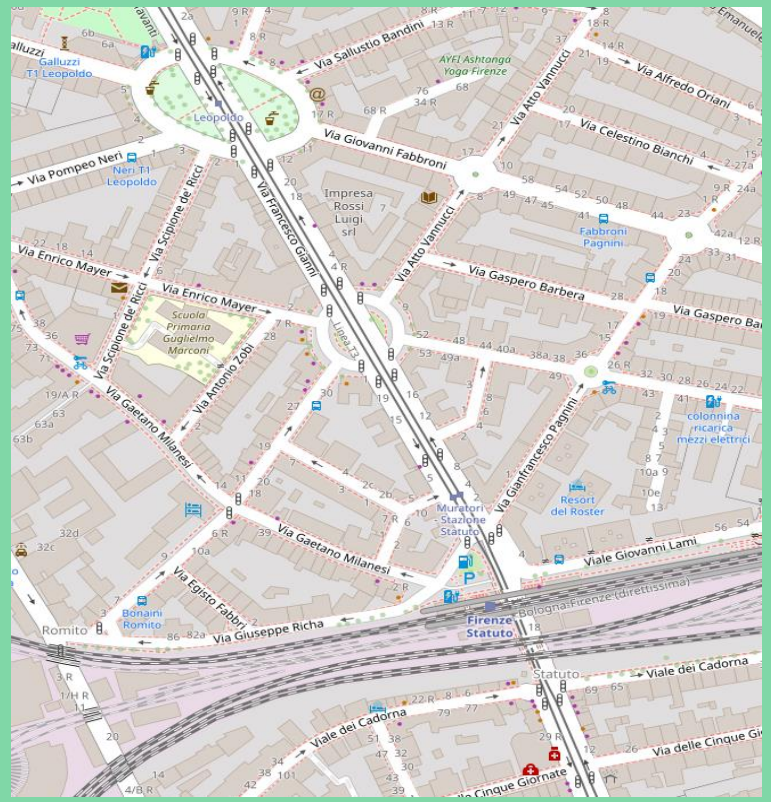


Proposed sloution



Input Data

Get Data from Snap4city



Define the Environment based this data

Our Reward

- *Calculate the proportion of green time in the cycle time*:

$$px = \frac{G}{C}$$

- *Calculate the saturation flow rate*:

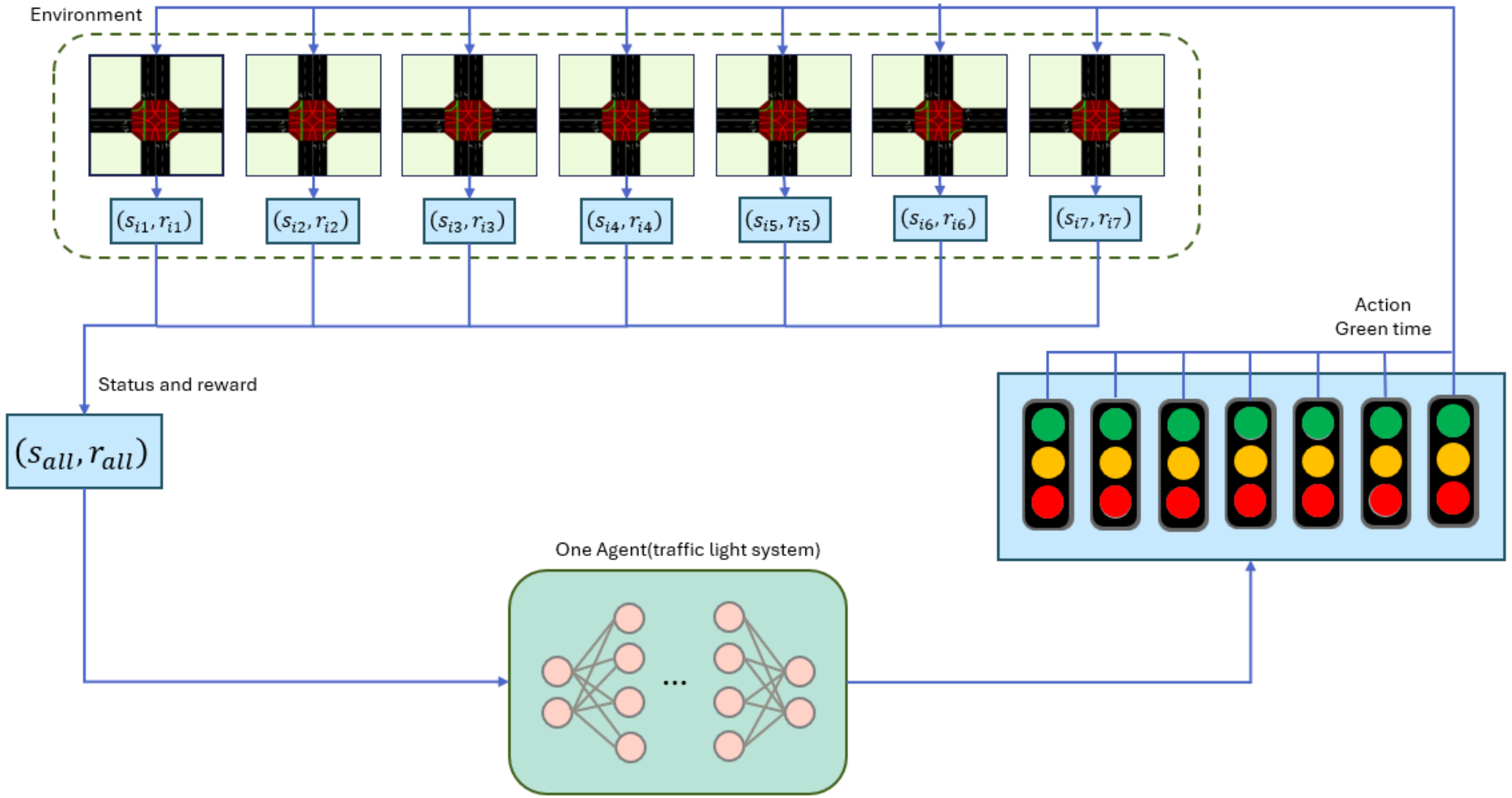
$$s = \frac{f}{d \cdot v}$$

- *Calculate the average delay*:

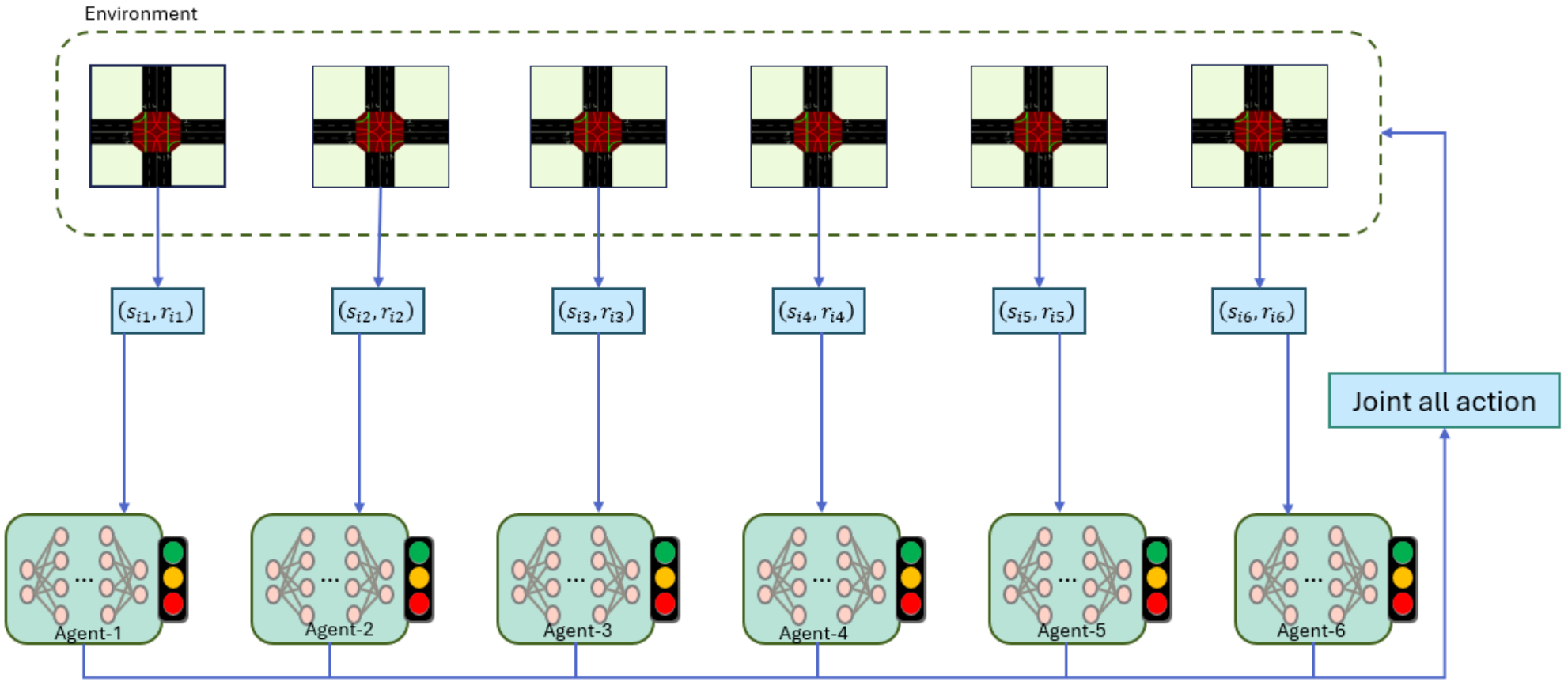
$$average\_delay = \frac{C \cdot (1 - px)^2}{2.(1 - px.s)}$$

Calculation of Average Delay at Traffic Light
Define the parameters:
C : *Cycle time*
G : *Green time*
f : *Flow of vehicles*
d : *Density of vehicles*
v : *Velocity of vehicles*

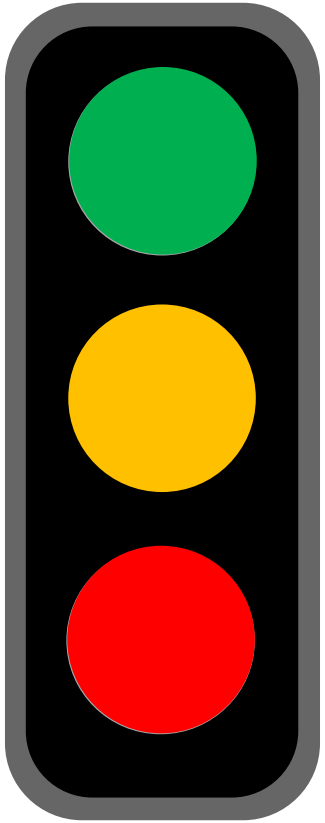
Single-Agent Deep Reinforcement Learning



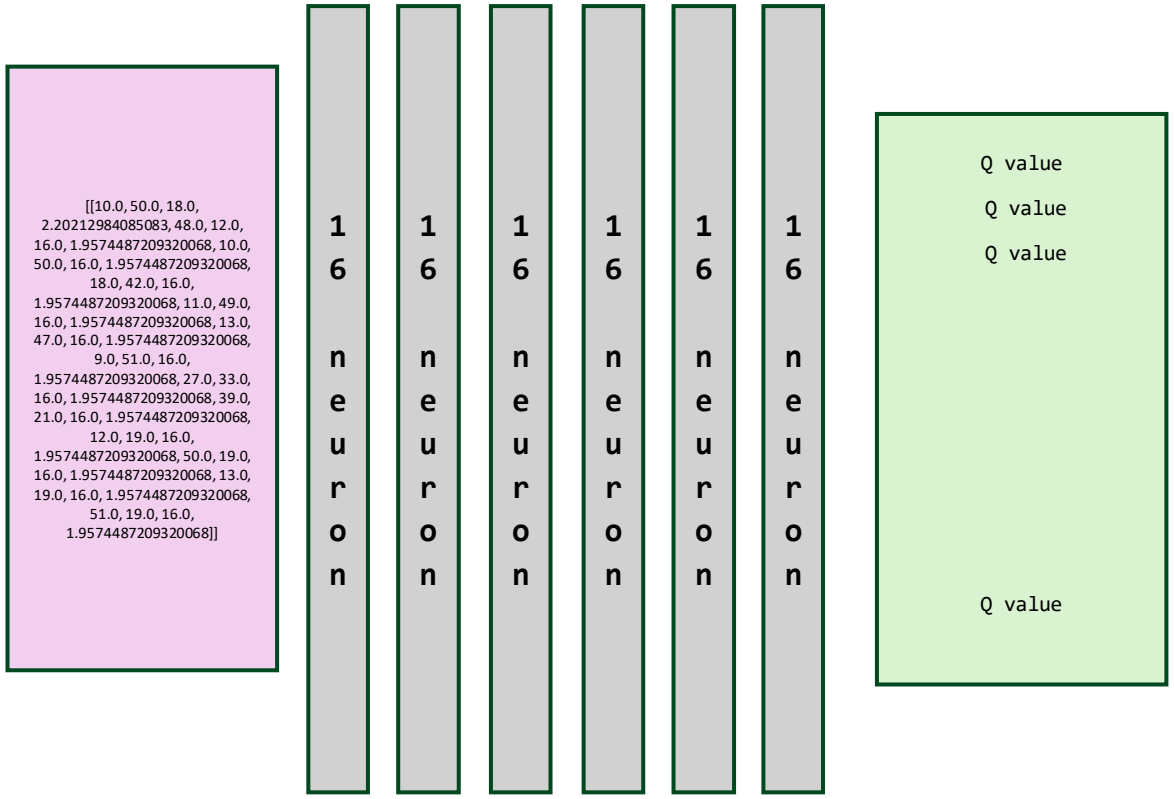
Multi-Agent Deep Reinforcement Learning



Results…



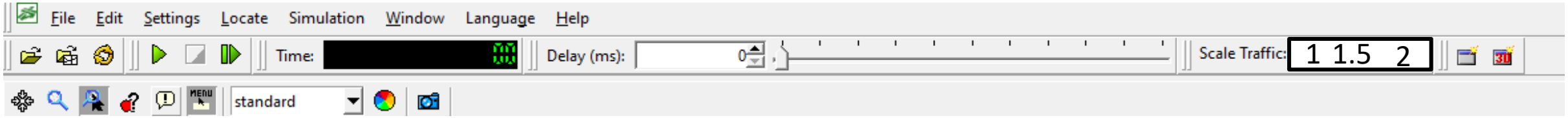
Sample of Results for Single Agent



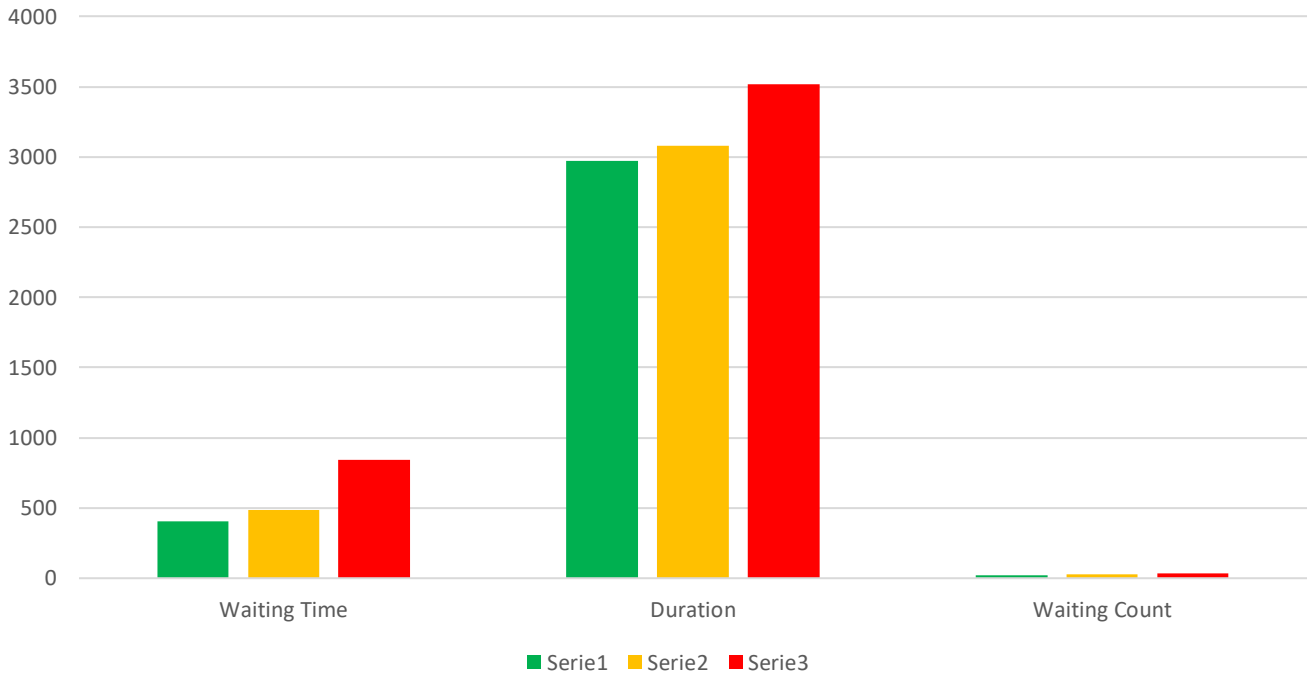

```
Using  'best result'
activation='relu'
```

	Waiting Time	Duration	Waiting Count	Waiting Time
lineaAuto_OtherN	17.05	33.96	1.67	17.05
	111.86	135.12	3.45	111.86
	125.7	149.2	3.49	125.7
	116.78	139.88	3.46	116.78
	22.95	44.94	3.35	22.95
lineaAuto_H	7.75	114.97	0.77	7.75
	12.95	119.53	0.74	12.95
	7.29	113.76	0.76	7.29
	13.68	119.93	0.74	13.68
	8.82	116.85	1.17	8.82
lineaAuto_R	14.74	139.55	1.27	14.74
	51.8	179.95	2.4	51.8
	50.33	179.57	2.57	50.33
	47.6	176.37	2.46	47.6
	26	154.08	1.97	26
lineaAuto_N	33.38	204.34	2.45	33.38
	22.83	197.22	2.76	22.83
	31.99	204.58	2.53	31.99
	29.36	205.26	3.27	29.36
	59.48	249.6	5.1	59.48
lineaAuto_D	29.12	253.66	1.81	29.12
	21.3	245.91	2.13	21.3
	18.27	243.6	2.19	18.27
	17.93	242.63	1.87	17.93
	34.61	270.851	5.14	34.61
lineaTram_3_Costanza	36.66	479.83	2.16	36.66
	0	427	0	0
	0	427	0	0
	0	427	0	0
	34.9	478.363	4.45	34.9

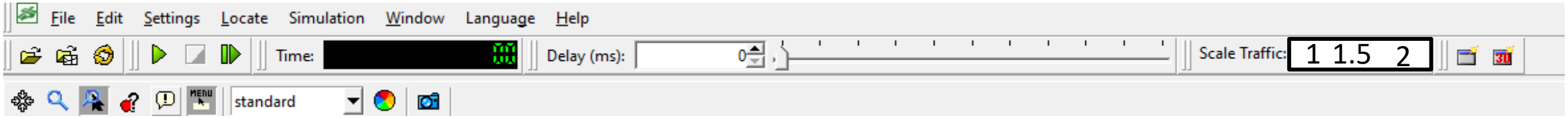
Workload



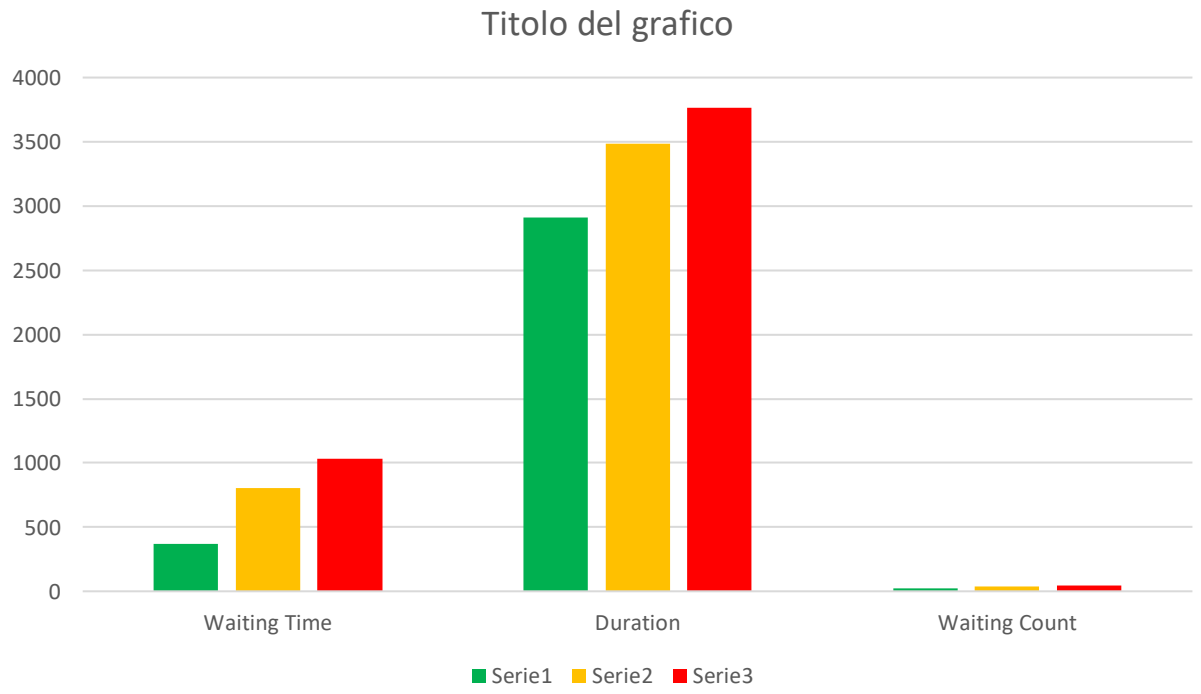
DQN Multi Agent



Workload



PPO Multi Agent

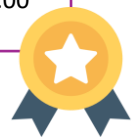


Trip info Sumo

It is consuming to fill in the trip info

```python
def _calculate_reward(self, tl_id):
        # Get the edge IDs controlled by the traffic light
        controlled_edges = self._get_control_lane(tl_id)

        # Initialize reward based on halting cars
        total_halting_cars = 0

        for edge in controlled_edges:

            halting_cars = traci.edge.getLastStepHaltingNumber(edge)
            total_halting_cars += halting_cars

        # Reward is negative for congestion
        return -total_halting_cars
```

sum	dir_N	dir_M	dir_A	dir_D	Careggi	Costanza
2545.54	168.84	194.61	185.83	227.88	450.00	436.00



```xml
<tripinfos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/tripinfo_file.xsd">
    <tripinfo id="other_N.0" depart="0.00" departLane="614268487_0"
departPos="3.68" departSpeed="7.00" departDelay="0.00" arrival="10.00"
arrivalLane="29495109#0_1" arrivalPos="0.20" arrivalSpeed="2.95" duration="10.00"
routeLength="54.34" waitingTime="0.00" waitingCount="0" stopTime="0.00"
timeLoss="1.77" rerouteNo="0" devices="tripinfo_other_N.0" vType="car"
speedFactor="0.94" vaporized=""/>
    <tripinfo id="other_O.0" depart="0.00" departLane="-1195117976_0"
departPos="5.10" departSpeed="7.00" departDelay="0.00" arrival="18.00"
arrivalLane="22926847#1_0" arrivalPos="1.96" arrivalSpeed="6.51" duration="18.00"
routeLength="100.48" waitingTime="0.00" waitingCount="0" stopTime="0.00"
timeLoss="2.74" rerouteNo="0" devices="tripinfo_other_O.0" vType="car"
speedFactor="1.03" vaporized=""/>
    <tripinfo id="other_O.1" depart="12.00" departLane="-1195117976_0"
departPos="5.10" departSpeed="7.00" departDelay="0.61" arrival="33.00"
arrivalLane="22926847#1_0" arrivalPos="1.96" arrivalSpeed="6.38" duration="21.00"
routeLength="100.48" waitingTime="3.00" waitingCount="1" stopTime="0.00"
timeLoss="5.93" rerouteNo="0" devices="tripinfo_other_O.1" vType="car"
speedFactor="0.94" vaporized=""/>
    <tripinfo id="other_N.1" depart="16.00" departLane="614268487_0"
departPos="3.68" departSpeed="7.00" departDelay="0.75" arrival="37.00"
arrivalLane="29495109#0_1" arrivalPos="0.20" arrivalSpeed="3.85" duration="21.00"
routeLength="54.34" waitingTime="5.00" waitingCount="2" stopTime="0.00"
timeLoss="12.72" rerouteNo="0" devices="tripinfo_other_N.1" vType="car"
speedFactor="1.02" vaporized=""/>
<tripinfo id="other_O.2" depart="23.00" departLane="-1195117976_0"
departPos="5.10" departSpeed="7.00" departDelay="0.22" arrival="40.00"
arrivalLane="22926847#1_0" arrivalPos="1.96" arrivalSpeed="6.58" duration="17.00"
routeLength="100.48" waitingTime="0.00" waitingCount="0" stopTime="0.00"
timeLoss="1.81" rerouteNo="0" devices="tripinfo_other_O.2" vType="car"
speedFactor="0.95" vaporized=""/>
```

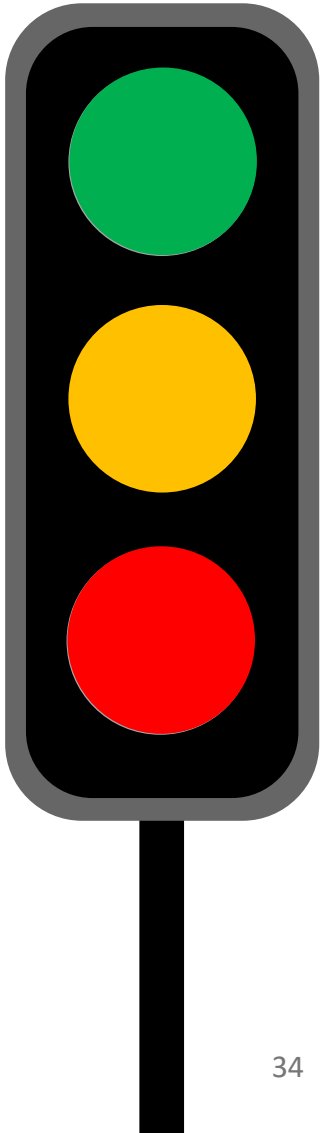
Conclusion

DQN for normal traffic conditions.
PPO for heavy traffic scenarios.

•RL methods **outperform fixed-cycle controls** in high-traffic conditions. Models adapt to **dynamic traffic patterns**:**Reduced delays.**
•**Improved fairness** across traffic directions.

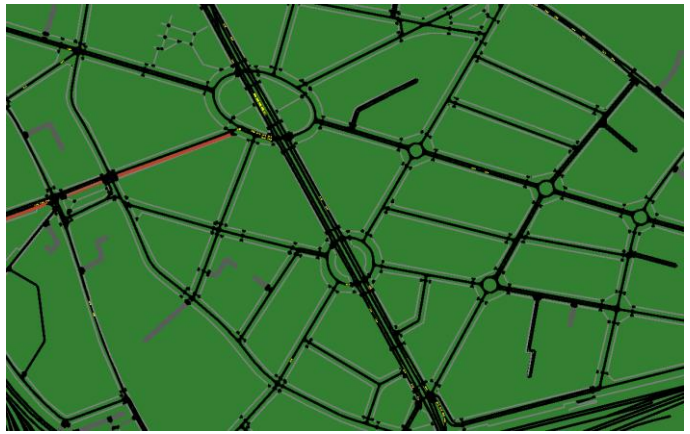
Future Scope
•Integrate **public transportation** and **pedestrian traffic**.
•Expand coverage to **larger city areas.**

Thank you



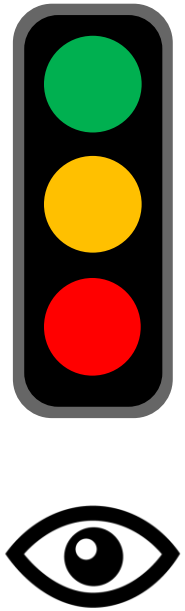
Introduction

SUMO



Actuated vs. Fixed traffic light

Actuated



Fixed