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

SNAP4CITY KM4CITY
Powered by

NeuroSymbolic Artificial Intelligence at Scale

Paolo Nesi, paolo.nesi@unifi.it
Marco Fanfani, marco.fanfani@unifi.it

https://www.disit.org/

Parte: 4 (2025-26)

TOP

Neuro-Symbolic Artificial Intelligence
16/06/2026

P4: Deep Reinforced Learning and Symbolic at Scale
- multi agent Deep reinforced learning
- RL and simulation

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

SNAP4CITY

KM4CITY

Powered by

Corso: **Neuro-Symbolic Artificial Intelligence at Scale**

P4: Deep Reinforced Learning and Symbolic at Scale

multi agent Deep reinforced learning

RL and simulation

Enrico Collini

enrico.collini@unifi.it

Link: Neuro-Symbolic Artificial Intelligence at Scale | Disit

Enrico Collini

University of Florence

Email verificata su unifi.it

Information Engineering

SEGUI

TITOLO	CITATA DA	ANNO
Predicting and understanding landslide events with explainable AI E Collini, LAI Palesi, P Nesi, G Pantaleo, N Nocentini, A Rosi IEEE Access 10, 31175-31189	82	2022
Deep learning for short-term prediction of available bikes on bike-sharing stations E Collini, P Nesi, G Pantaleo IEEE Access 9, 124337-124347	49	2021
Short-term prediction of city traffic flow via convolutional deep learning S Bilotta, E Collini, P Nesi, G Pantaleo IEEE Access 10, 113086-113099	40	2022
Data sources and models for integrated mobility and transport solutions P Bellini, S Bilotta, E Collini, M Fanfani, P Nesi Sensors 24 (2), 441	30	2024
Flexible thermal camera solution for Smart city people detection and counting E Collini, LAI Palesi, P Nesi, G Pantaleo, W Zhao	29	2024

TESTO CONSIGLIATO

Reinforcement Learning: An Introduction di Sutton e Barto (https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf)

DISIT lab, NeSyAIatScale 2025-26

Powered by

Neuro-Symbolic Optimization of Traffic Infrastructure via Reinforced Learning and Deep Graph Neural Network

Enrico Collini, Luciano Alessandro Ipsaro Palesi, Paolo Nesi
University of Florence, DISIT Lab, Snap4City

Https://www.disit.org https://www.Snap4City.org

Paolo Nesi, paolo.nesi@unifi.it
https://www.Km4City.org
https://www.disit.org

The Urban Congestion Problem

- Increasing urban populations (expected to reach 68% globally by 2050) and heavy reliance on private transport have intensified **traffic congestion**
- Congestion leads to significant economic costs, environmental degradation (CO2 emissions), and reduced quality of life due to increased travel times and fuel consumption
- Traditional manual "trial-and-error" analysis is resource-intensive because the search space for infrastructure changes is mathematically vast
- For instance, choosing just 5 lane changes out of 100 roads involves 75 million possible combinations
- <u>Data-driven strategies</u> are required to navigate this huge combinatorial space and find suboptimal or optimal solutions

Limitations of Conventional Approaches

Where State-of-the-Art falls short:

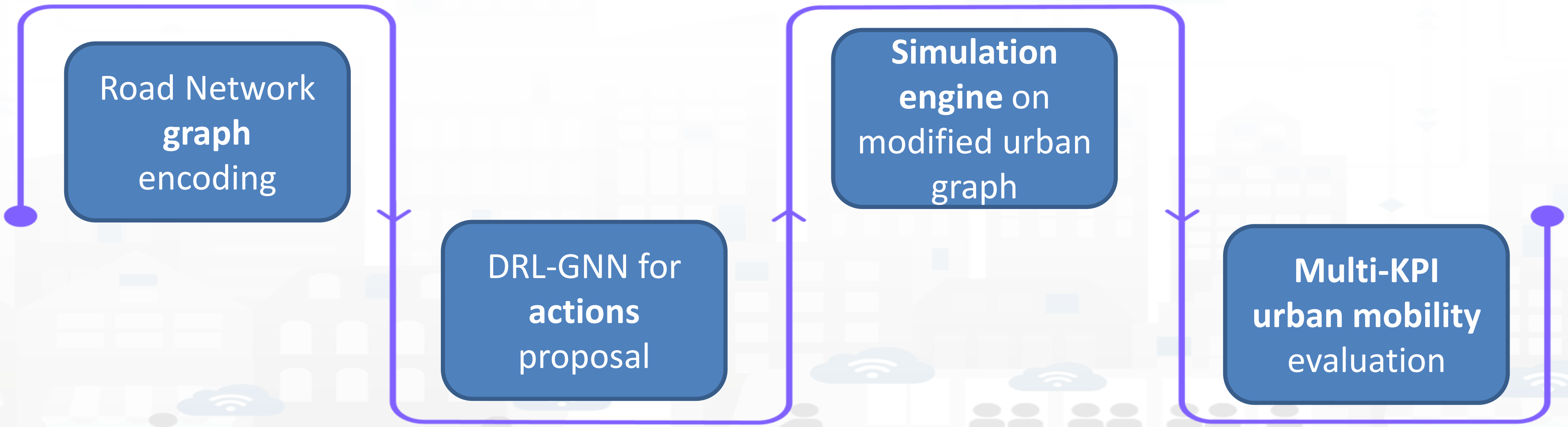
- No physics-grounded traffic simulation models
- Synthetic data inputs reduce real-world applicability
- Missing urban environmental KPIs evaluations via fuel consumption and CO2 emisisons
- Operational constraints and road regulations not enforced in simulation

Constraint-Aware Infrastructure Modification

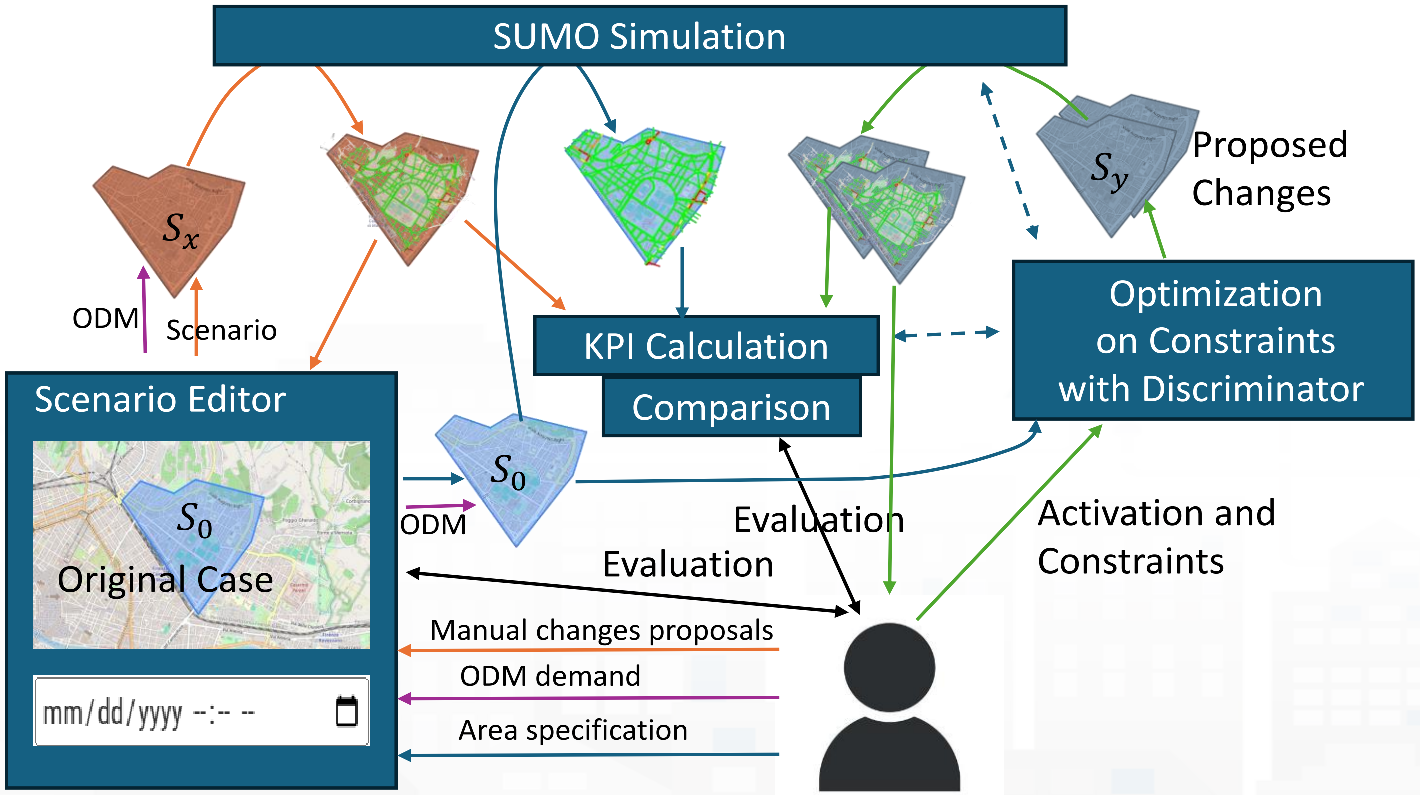
- Actionable Changes via lane reversals
- Operators define a maximum number of allowable changes, keeping solutions economically and operationally realistic for near-term deployment.
- Roundabout configurations, required connectivity, and regulatory constraints are enforced (the network always remains fully traversable)
- Road creation/removal, speed limits, and turn restrictions remain under expert control via the Scenario Editor

A Neuro-Symbolic Deep-RL-GNN Framework

A Deep Reinforcement Learning and Graph Neural Network framework for scalable, constraint-aware, multi-objective optimization of smart city mobility systems.



System Architecture



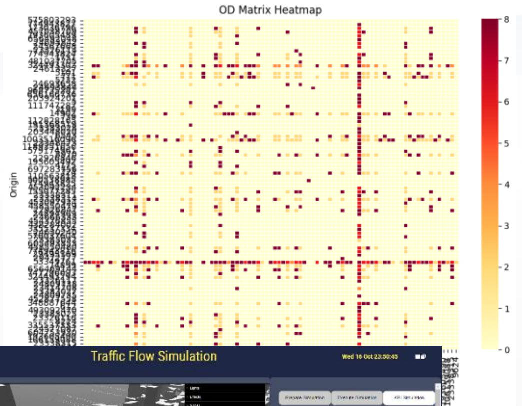
Automated mode delegates exploration to the RL agent, which proposes feasible infrastructure modifications and iteratively improves solutions through simulation-grounded feedback

Manual mode enables operators to define road subgraphs, apply modifications interactively, and compare KPIs against a baseline origin–destination scenario.

Scenario Editor and infrastructure data model

Enables urban scenario definitions.
Allow to represent the Digital Twin of the urban road infrastructure.

The road graph models road elements extending the model of OSM and providing road categories (Primary, Secondary, Tertiary, Residential, pedestrian, cycling, etc.), number of lanes, direction (one-way/two-ways), restriction at the crossroads, start node and end node of the element, etc



Enables what-if-analysis
Integrated with simulation engines and decision support data driven analysis

ODM

SUMO

Urban KPI estimation

Infrastructure data model

Enables urban scenario definitions.
Allow to represent the Digital Twin of the urban road infrastructure.

TABLE III – ROAD INFRASTRUCTURE FEATURES DESCRIPTION AND USAGE

Feature	Description	SYSTEM
Road segment id	OSM ID of the considered road segment.	action selection, GNN graph edges, SUMO simulation, KPI computation
Road id	OSM id of the road of which the segment is part of.	SUMO simulation
Restrictions	Traffic constraint, e.g., allowed turns, go straight only.	SUMO simulation
ODM	Service Uri: ID of Snap4City device that contains ODM information.	SUMO simulation
Road type	Indicators for road types: primary, secondary, tertiary, residential, unclassified.	SUMO simulation, GNN input feature
Road segment length	Length of the road segment.	SUMO, GNN input feature
Lanes	Number of lanes of the road segment.	SUMO, GNN input feature
Modifiable	Binary: 1 if the road can be modified, 0 otherwise.	Action selection
Maximum speed	Maximum speed in m/s.	SUMO, GNN input feature
Vehicle density	Vehicles per kilometer.	SUMO output
Vehicle count	Total number of vehicles.	SUMO output
Average speed	Average vehicle speed in km/h.	SUMO output
Fuel consumption	Amount of fuel consumed.	KPI computation
CO2 emissions	Carbon dioxide emitted.	KPI computation

Use Case

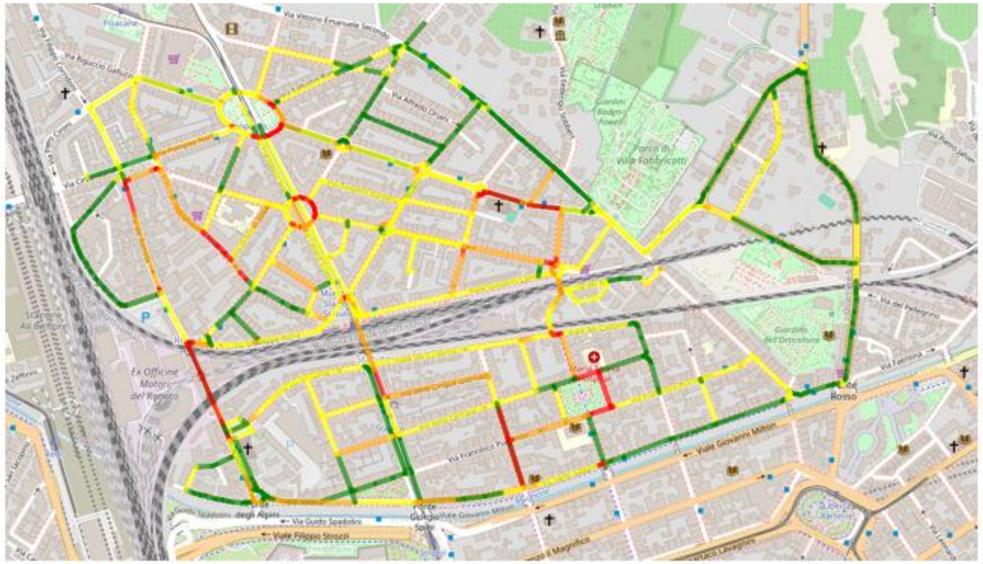


FIGURE 5 – Original road infrastructure as $scenario_1$ with ODM traffic at $t1$ = 2023-09-28T09:00:00.

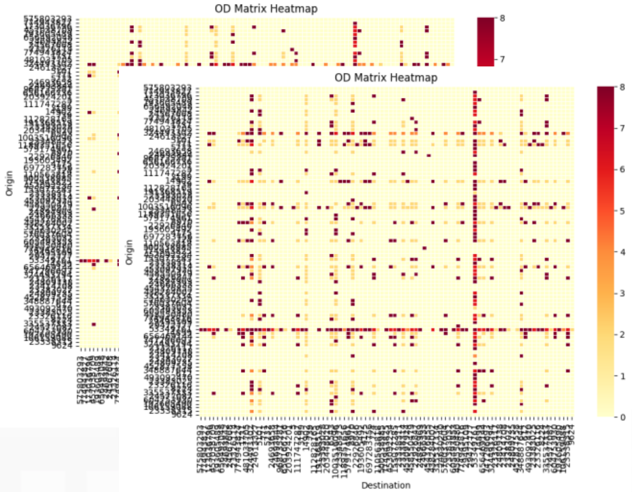
Urban KPI estimation

$scenario_1$ at t1	KPI_{time}	KPI_{fuel}	KPI_{CO2}
Scenario1 KPIs at t1	3056	343.627	0.248

$scenario_1$ at t2	Travel Time [s]	Fuel [l]	CO2 [ton]
Scenario1 KPIs at t2	2927	240.379L	0.214

Frequently congested urban area in Florence

SUMO traffic simulation

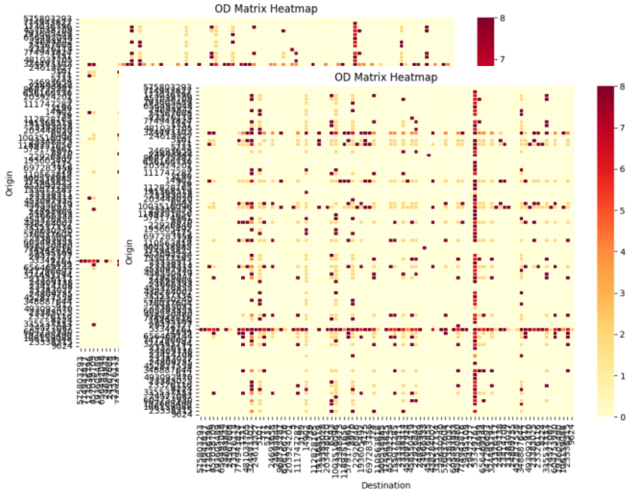


ODM estimated at 09:00 and 19:00 of the 2023-09-28

Use Case

Frequently congested urban area in Florence

SUMO traffic simulation



ODM estimated at 09:00 and 19:00 of the 2023-09-28

The corresponding road graph has 1811 road elements and a total of 399 distinct roads

Via Duarouter traffic flow is generated from the ODM data. The simulation runs for 1 hour

The ODM used for *scenario_1* included a total of 1946 flows across 77 road elements (65 origins and 61 destinations).

Urban KPI

network-wide travel time	To obtain a network-level travel time indicator, the segment travel times are aggregated by weighting them according to the corresponding traffic.	$$KPI_{time} = \sum_e T_e N_e$$ Where N_e refers to the number of vehicles traversing road segment *e* during the observation window, segment travel time T_e which is computed for road segment e with length L_e and mean speed V_e, as: $$T_e = \frac{L_e}{V_e}$$ where: L_e is expressed in meters, and L_e and V_e in meters per second

Urban KPI

fuel consumption	captures the quantity of fuel consumed (in liters, assuming fossil combustion) by the estimated number of vehicles in the scenario if they travel the segment they were on	$fuel100km_e = \begin{cases} 128 + \dfrac{800}{8\,V_e^2} & if\ V_e < 10\ km/h \\ 7 + \dfrac{90}{V_e} & if\ V_e \geq 10\ km/h \end{cases}$ The amount of fuel consumed in the segments is computed by the: $fuel_e = \frac{fuel100km_e}{100} L_e\, N_e$ [liters] On this basis, the total amount of fuel consumed in the whole scenario KPI can be estimated by: $KPI_{fuel} = \sum_e fuel_e$ [liters]

Urban KPI

CO2 emissions	computes the amount of CO2 in ton generated by the vehicles in the scenario in traveling the segments of the scenario over time considering the conditions of fluid flow and heavy traffic/stop-and-go are estimated (emission factors).	$CO2_e = \begin{cases} 272\ (\rho \;\; 1000)\ L_e & if \;\; \rho < \rho_c \\ 496.3\ (\rho \;\; 1000)\ L_e & if \;\; \rho \geq \rho_c \end{cases}$ On this basis, the total amount of CO2 emissions for the scenario KPI can be computed as: $KPI_{CO2} = \sum_e CO2_e \qquad [ton]$

The traffic condition on each road segment is determined by comparing the estimated traffic density ρ with the corresponding critical density ρ_c.

(Free Flow: $\rho < \frac{1}{2}\rho_c$), yellow (Fluid Flow: $\frac{1}{2}\rho_c < \rho < \rho_c$), orange (Heavy Flow: $\rho_c < \rho < \frac{3}{4}\rho_c$), and red (Very Heavy Flow: $\rho > \frac{3}{4}\rho_c$)

Symbolic Discriminator

Formal filter that ensures every infrastructure change proposed by the Deep GNN adheres to legal regulations, physical feasibility, and urban constraints.

Feasible Action Selection The discriminator identifies the set of feasible actions by evaluating potential road direction changes against predefined rules:

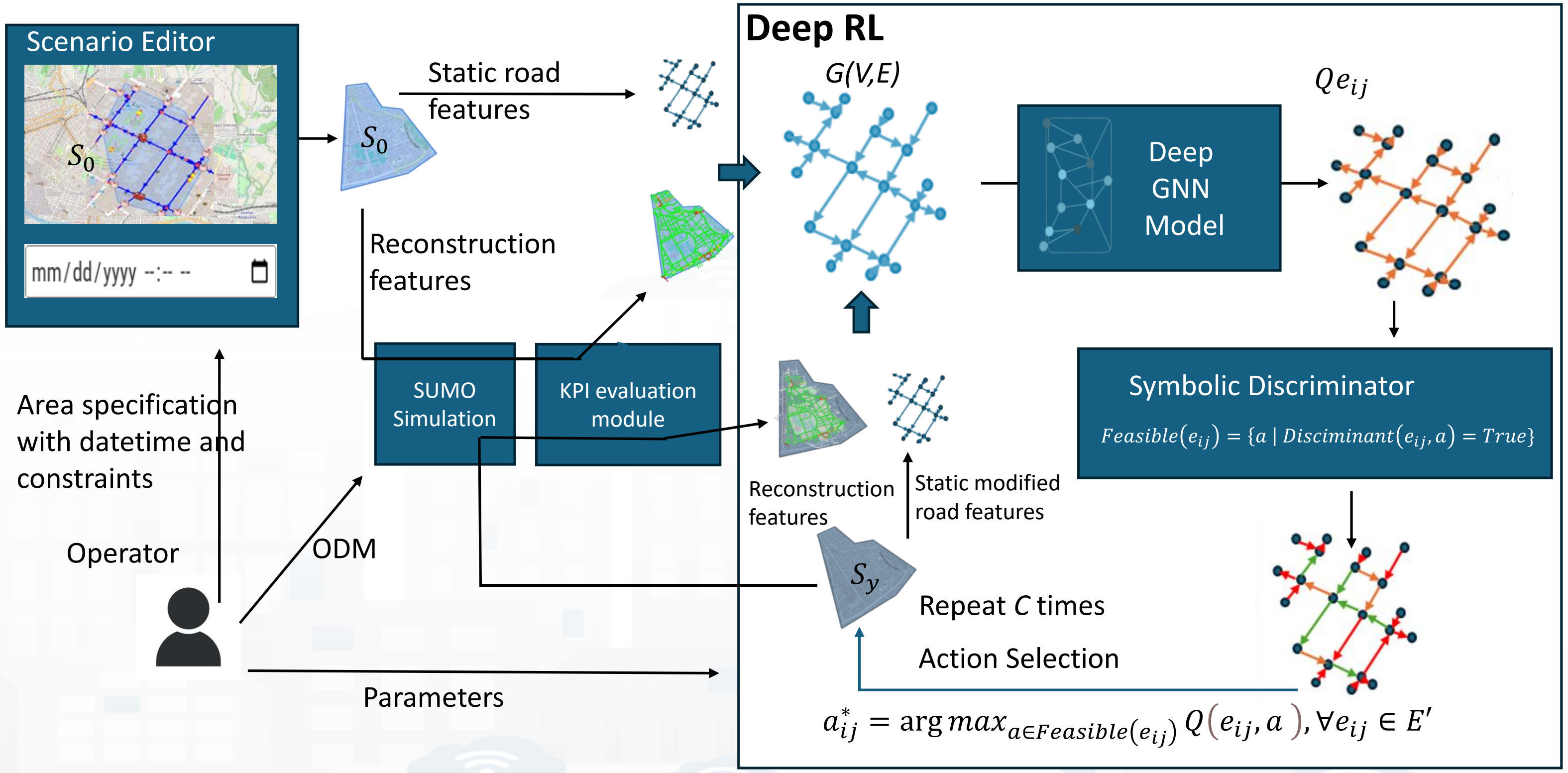
- Non modifyable edges via SE City operators can designate specific road segments as non-modifiable (e.g., critical arterials or roundabouts) due to safety or high intervention costs
- Topological Constraits ensure the road network remains functional, the discriminator prevents "impossible intersections"—conditions where an intersection has only incoming or only outgoing roads

For every non-terminal node v∈V+, after an action a is applied to reach a new graph state G′, the following must hold for the node's in-degree (D −) and out-degree (D +)

$$D^{-}(v) > 0 \text{ } and \text{ } D^{+}(v) > 0$$

NEURO-SYMBOLIC OPTIMIZATION

Deep-RL-GNN Optimization schema in inference/execution. Informed graph is elaborated via the DRL module composed by a GNN that estimates the utilities of changing the direction of the road which the road segment is part of. A symbolic discriminator elaborates the feasibility of actions based on constraints. The policy for action selection is repeated C times to limit changes at C.

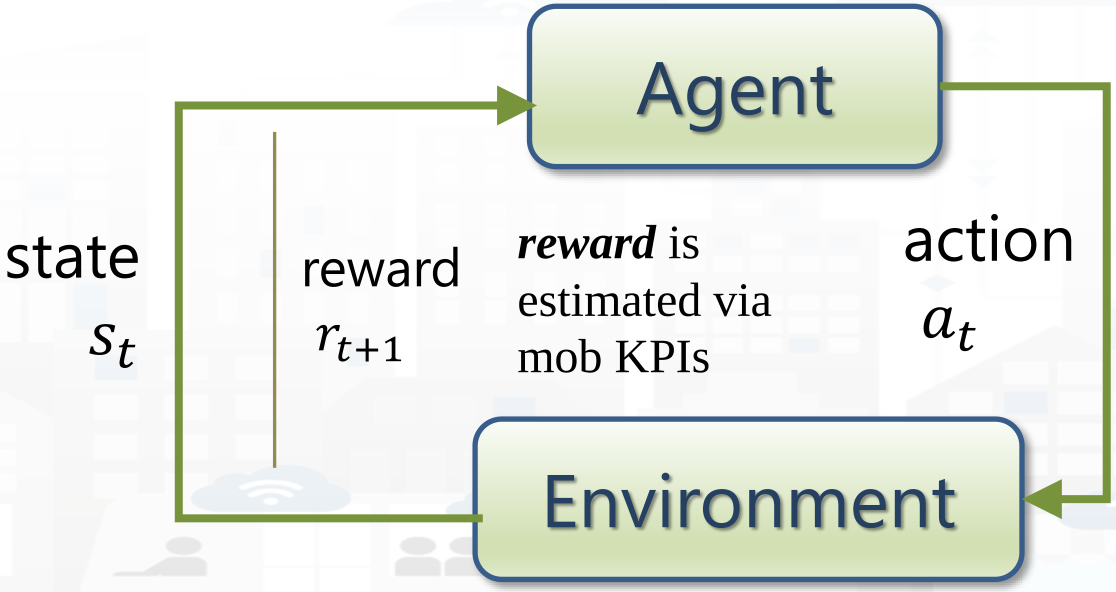


NEURO-SYMBOLIC OPTIMIZATION

The solution is grounded on **Deep-RL** and **GNN** to explore the space of road traffic infrastructure configurations, and conditions in terms of ODM on the basis of selected KPIs to identify the best road graph configuration taking into account the imposed symbolic constraints

RL approaches are based on an agent which interacts with an observed environment to decide actions/changes

state, a configuration (of the road graph)



reward is estimated via mob KPIs

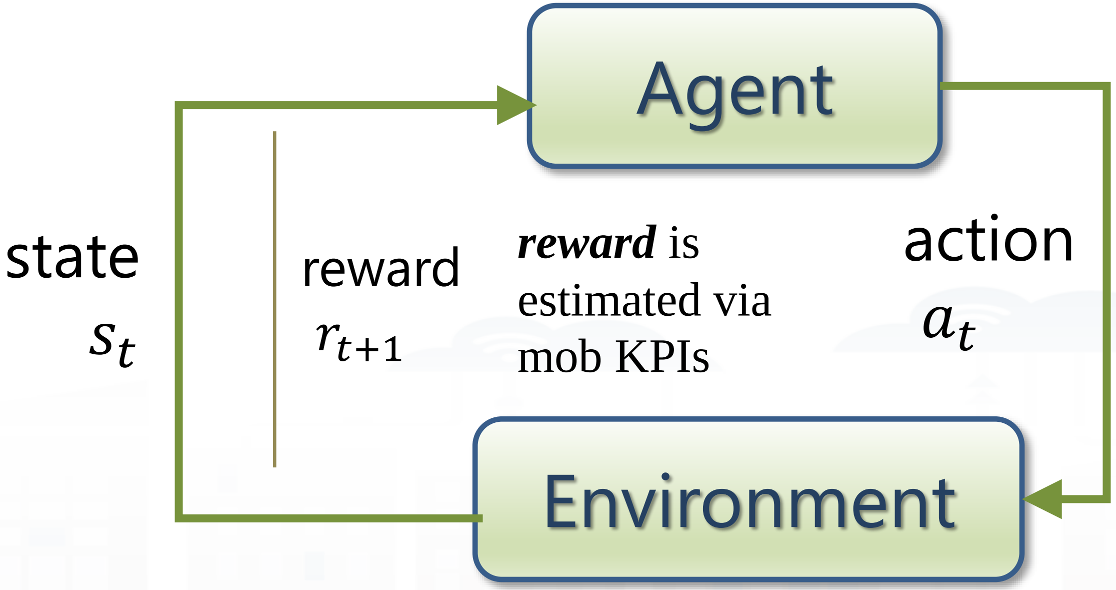
RL algorithm may identify a set of possible ***actions (road dir changes)***

road graph traffic infrastructure + ODM

NEURO-SYMBOLIC OPTIMIZATION

a ***policy*** strategy can be adopted to improve the condition/status. In the process of improving, a ***value*** function includes the prediction of the next status and action on the basis of current status, policy, and action.

state, a configuration (of the road graph)



reward is estimated via mob KPIs

RL algorithm may identify a set of possible ***actions (road dir changes)***

road graph traffic infrastructure + ODM

In order to perform ***predictions*** on the environment, the RL needs to have an environment model which is a Digital Twin, on which the ***exploration*** of changes should be possible and realistic, for example by performing simulation.

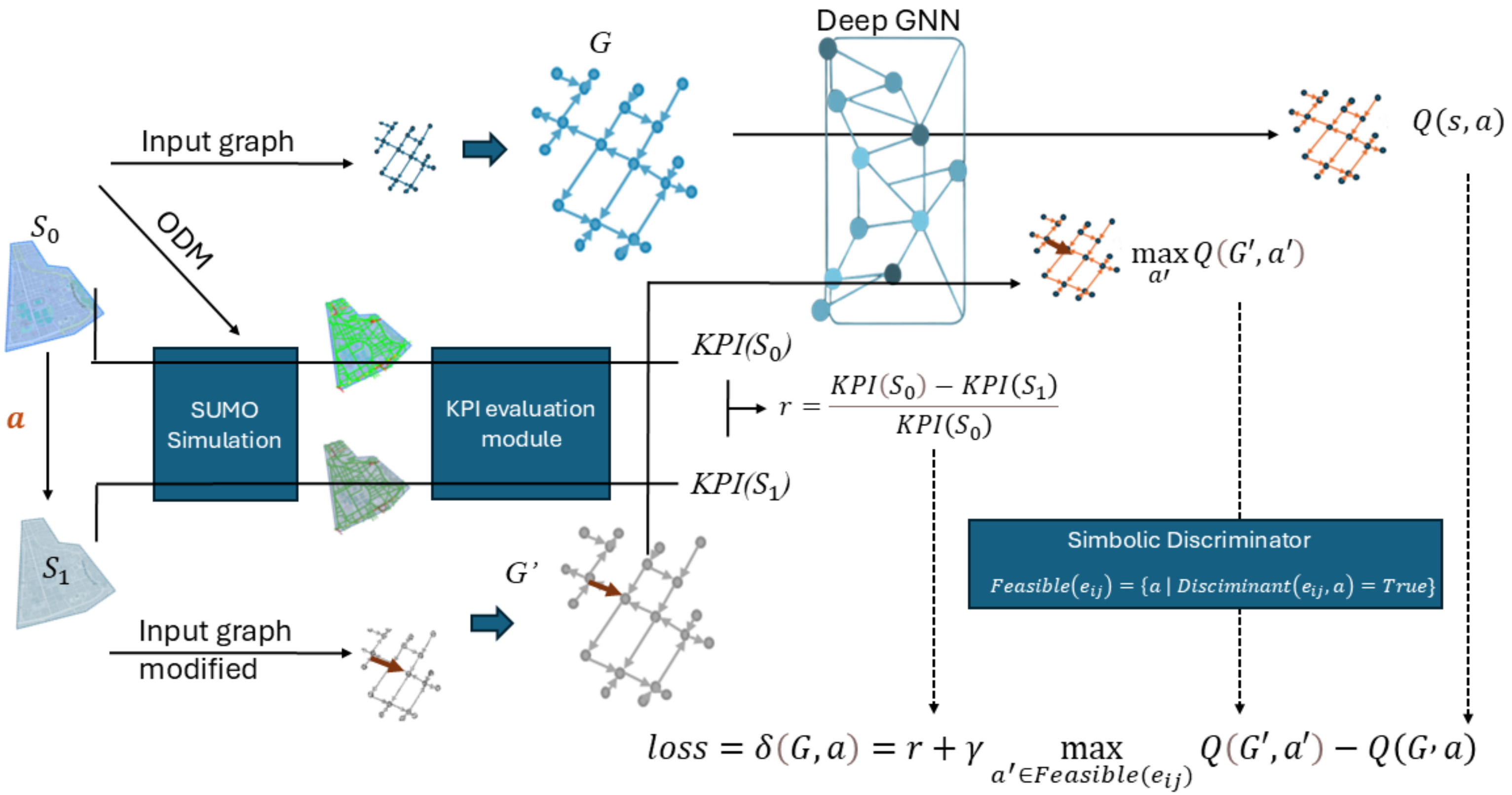
Simulated Dataset

SUMO

The best actions should be ***exploited*** to achieve the highest rewards

NEURO-SYMBOLIC OPTIMIZATION

On the basis of the simulation, it is possible to estimate the ***Temporal Difference, TD,*** between predicted and actual rewards. The learning is introduced within the ***Q-learning***



Dataset

For the training of the Deep GNN model, a specific dataset has been generated starting from the city area on which the optimization activity is performed.

From the initial S_0, 30 different changes have been generated (avoiding performing unacceptable configurations for constraints and rules) by using the Symbolic Discriminator.

The process of generating configurations for the training has been repeated until the max depth has been reached (in our cases limited to 5 since 4 is the max number of accepted changes on *So* by the operators)

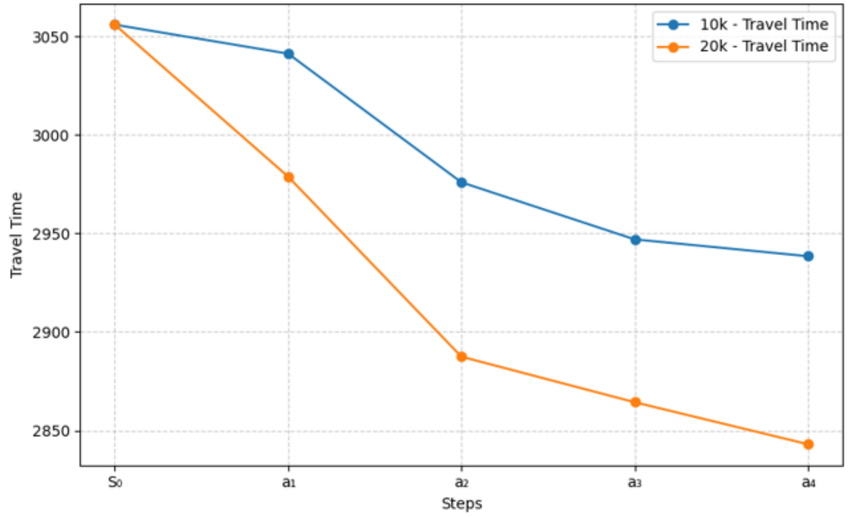


FIGURE 4 - KPI_{time} obtained during inference by progressively applying 4 actions by the Deep-RL-GNN with Deep GNN Models trained on the 10000- and 20000 datasets, respectively.

GNN configuration

Deep GNN in the proposed Deep-RL-GNN system is used to predict the *Q-value* for every future possible action *a'* on *G'*: $Q(G', a')$ based on the TD between estimated and actual rewards.

During the training of the Deep GNN Model both positive and negative feedbacks are important towards the development of a model capable of assessing the expected utility of actions

$$r = \frac{KPI(S_n) - KPI(S_{n+1})}{KPI(S_n)}$$

TABLE IV – DEEP GNN ARCHITECTURE DETAILS

Layer	type	input	output
GNN layer	First graph layer that computes from edge features the first layer edges embeddings	Edge features [num_edges, num_features]	Edge embeddings [num_edges, hidden_dim]
GNN layer	Second graph layer that updates the embeddings	[num_edges, hidden_dim]	[num_edges, hidden_dim]
Additional GNN layers	The next number of additional graph layers that updates the embeddings. Max set to 8.	[num_edges, hidden_dim]	[num_edges, hidden_dim]
MLP	Multi-Layer Perceptron layer that processes the embeddings	[num_edges, hidden_dim]	Output Q-values [num_edges]

TABLE V – HYPERPARAMETERS FOR THE GNN TRAINING OPTIMIZATION

Hyperparameter	Range	Description
Number of GNN Layers	[2 - 8]	Number of GNN layers
Hidden dimension	[32,128]	Size of the latent edge embedding in GCN layers
Learning Rate	[0.01, 0.0001]	Step size for gradient descent.
Discount Factor γ	[0.6-0.999]	Governs the importance of future rewards in the Bellman equation.

GNN configuration

Deep GNN in the proposed Deep-RL-GNN system is used to predict the *Q-value* for every future possible action *a'* on *G'*: $Q(G', a')$ based on the TD between estimated and actual rewards.

TABLE VI – DEEP GNN TRAINING SETUP: HYPERPARAMETER CONFIGURATION AND TRAINING TIME FOR EACH KPI (TRAVEL TIME, FUEL CONSUMPTION, AND CO2 EMISSIONS), AND FOR ALL OF THEM COMBINED

Deep GNN Model	GNN layers	Hidden dim	Learning Rate	Discount Factor γ	Training time (s)
model_t1_time_reduced_dataset of 10000	*4*	*41*	*0.0066124*	*0.920511*	*895*
model_t1_time dataset of 20000	*6*	*78*	*0.000531*	*0.620989*	*767*
model_t1_fuel dataset of 20000	*5*	*118*	*0.006211*	*0.944024*	*974*
model_t1_co2 dataset of 20000	*5*	*91*	*0.003287*	*0.903963*	*1072*
model_t1_multiObj dataset of 20000	*4*	*47*	*0.004457*	*0.906501*	*1879*

During the training of the Deep GNN Model both positive and negative feedbacks are important towards the development of a model capable of assessing the expected utility of actions

$$r = \frac{KPI(S_n) - KPI(S_{n+1})}{KPI(S_n)}$$

Single-KPI

The Deep GNN (DGNN) was trained using a Q-learning–based approach with experience replay. In the case of **multi-objective** optimization, the model produces the Q-values of the respective three KPIs and the loss exploits Multiple-Gradient Descent Algorithm (MGDA) aiming towards reaching a Pareto-optimal solution for the KPIs.

Experimental Results

Assessing Deep-RL-GNN on Scenario 1

1) Deep-RL-GNN on Scenario 1 on travel time,

The Deep GNN in the proposed Deep-RL-GNN solution has been trained on the data regarding $scenario_1$ considering ODM of time window ($t1$) 2023-09-28 at 9:00 for 60 minutes duration.

As a reference metric for the KPI to estimate the rewards in this experiment has been used the travel time, KPI_{time}.

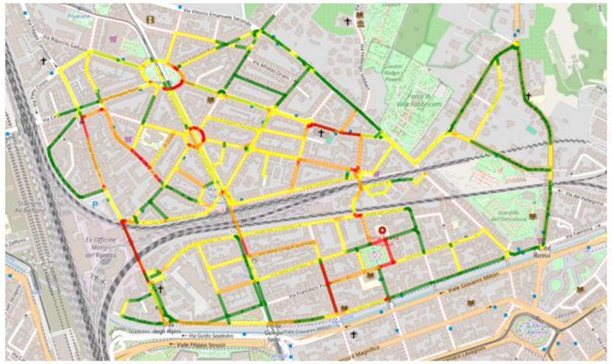


FIGURE 5 – Original road infrastructure as $scenario_1$ with ODM traffic at $t1$ = 2023-09-28T09:00:00.

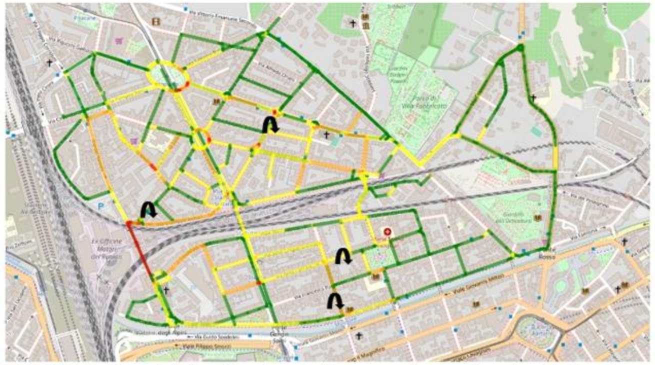


FIGURE 6– Road infrastructure of $scenario_1$ with simulation based on traffic ODM at $t1$ *(morning)* after the changes (curved blue icons) suggested by Deep-RL-GNN by *model_t1_time*.

effects have been also reported in terms of improvements %:

$$\Delta_{\%}^{KPI_i} = \frac{KPI_i - initialKPI_i}{initialKPI_i} * 100$$

where i refers to one of the 3 considered KPIs on time, fuel and CO2. On the basis of defined KPIs, the Deep-RL-GNN proposed effective changes with a mean delta improvement % across the defined KPIs of -8.22% computed by:

$$\Delta_{\%}^{mean} = \frac{\sum \Delta_{\%}^{KPI_i}}{3}$$

TABLE VII –DEEP-RL-GNN WITH *model_t1_time* ON $scenario_1$ AT $t1$.

$scenario_1$ at t1		Optimized on	Other KPI values	
		KPI_{time}	KPI_{fuel}	KPI_{CO2}
Scenario1 KPIs at t1		3056	343.627	0.248
model_t1_time	Fist change	2978	340.198	0.238
model_t1_time	With 1st and 2nd road change	2887	314.253	0.232
model_t1_time	+ 3rd change	2864	314.78	0.226
model_t1_time	All 4 changes	2842	320.362	0.221
$\Delta_{\%}^{KPI_i}$		-7.00%	-6.77%	-10.89%

Experimental Results

Assessing Deep-RL-GNN on Scenario 1

4) Multi Objective Optimization Deep-RL-GNN on Scenario

In the multi-objective setting, where the model outputs a vector of Q-values (one for each KPI), action selection is performed by identifying the set of Pareto non-dominated actions. If a single action dominates all others, it is directly selected. Otherwise, among the Pareto-optimal actions, the final choice is made by selecting the one closest to the ideal point in the objective space.

TABLE XII– DEEP-RL-GNN VS MODELS COMPARISON ON $scenario_1$ AT *t1*

$scenario_1$ at t1	KPI_{time}	KPI_{fuel}	KPI_{CO2}	$\Delta_{\%}^{mean}$
Scenario1 KPIs at t1	3056	343.627	0.248	
model_t1_time	**2842**	320.362	0.221	-8.22
model_t1_fuel	2845	**310.558**	0.227	**-8.33**
model_t1_co2	3051	399.771	**0.202**	-0.79
model_t1_multiObj	2903	342.201	0.238	-3.15

Transferability Deep-RL-GNN actions on different traffic ODM conditions

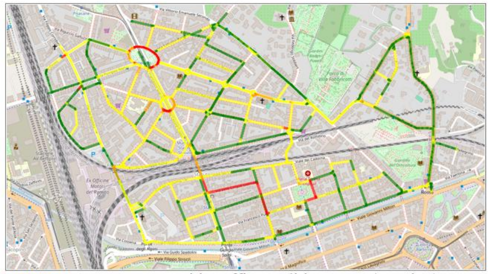


FIGURE 8 – $scenario_1$ with traffic conditions at *t2* (evening) 2023-09-28T19:00:00.

FIGURE 9 Road infrastructure of $scenario_1$ with simulation based on traffic ODM at *t2 (evening)* after the changes (curved blue icons) suggested by Deep-RL-GNN by *model_t1_time.*

TABLE XIII – DEEP-RL-GNN MODELS ACTIONS COMPARISON ON $scenario_1$ AT *t2* (EVENING)

$scenario_1$ at t2	Travel Time [s]	Fuel [l]	CO2 [ton]	$\Delta_{\%}^{mean}$
Scenario1 KPIs at t2	2927	240.379L	0.214	
model_t1_time	**2793**	233.195	0.206	-3.77
model_t1_fuel	2812	**227.314**	0.219	**-2.34**
model_t1_co2	2944	246.046	**0.201**	-1.05
model_t1_multiObj	2807	239.008	0.210	-2.18

CONCLUSIONS

The proposed neuro-symbolic optimization framework (Deep-RL-GNN) supports transportation operators in improving urban mobility by automatically proposing feasible road infrastructure changes (lane reversals/road direction changes), while enforcing operational constraints and traffic regulations through a Symbolic Discriminator.

The approach is integrated with the Snap4City Scenario Editor and a SUMO-based simulation loop to compute and compare network-wide KPIs

Experimental validation has been conducted Florence urban scenarios to identify small-scale intervention towards producing measurable improvements across the selected KPIs.