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

https://www.snap4City.org

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)

Neuro-Symbolic Artificial Intelligence

17/06/2026

P4: Deep Reinforced Learning and Symbolic at Scale

- multi agent Deep reinforced learning
- RL and simulation

Corso: **Neuro-Symbolic Artificial Intelligence at Scale**
P4: Deep Reinforced Learning and Symbolic at Scale
multi agent Deep reinforced learning
RL and simulation

[Link: Neuro-Symbolic Artificial Intelligence at Scale | Disit]

TESTO CONSIGLIATO

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

Luciano Alessandro Ipsaro Palesi
lucianoalessandro.ipsaropalesi@unifi.it
Room 231 – ex 465
https://www.disit.org
https://www.Snap4City.org

Office hours
Monday 10:30 - 13:00,
Via Santa Marta 3 - 50139 Firenze
Room **231**

- **Deep Learning**
- **Explainable Artificial Intelligence (XAI)**
- **Optimization and Simulation of Complex Systems**
- **Smart City, IoT/WoT e Digital Twin**

Google scholar:
https://scholar.google.com/citations?user=ZVSLAdgAAAAJ
Scopus:
https://www.scopus.com/authid/detail.uri?authorId=57226812823
ORCID:
https://orcid.org/0000-0001-8992-2084

Lecture Objectives

By the end of this course, you will be equipped to design, analyze, and implement reinforcement learning systems.

1

The RL Framework

Understand the agent–environment interaction loop and how reward signals drive learning

2

Value Functions

Study Bellman equations and how they underpin every major RL algorithm

3

Fundamental Algorithms

Analyze Dynamic Programming, Monte Carlo Methods, and Temporal Difference Learning

4

Deep RL

Combine deep neural networks with RL to tackle large-scale, real-world problems

Lecture Structure

PART I

Theoretical Foundations

Building rigorous mathematical intuition from the ground up.

- Markov Decision Processes
- Value functions & Bellman equations
- Dynamic Programming
- Monte Carlo Methods

PART II

Deep Reinforcement Learning

Scaling RL to complex, high-dimensional environments.

- Neural value function approximation
- Temporal Difference Learning
- Deep Q-Networks (DQN)
- Modern applications & case studies

Today

PART III

Application

- Traffic Light Optimization
- Optimization of Traffic Infrastructure

Urban Traffic Light Optimization: From NSGA/GA to Deep Reinforcement Learning

A comparative study of two optimization paradigms — multi-objective genetic algorithms and deep reinforcement learning — applied to real-world urban traffic control. Case study: the Florence urban network.

NEUROSYMBOLIC AI | TRAFFIC SYSTEMS | FLORENCE CASE STUDY

https://www.sciencedirect.com/science/article/pii/S1568494625005800

https://ieeexplore.ieee.org/abstract/document/11029249

Why This Problem Matters

Urban Congestion Impacts

- Increased travel and waiting delays
- Higher vehicle emissions and fuel waste
- Reduced public transport reliability
- Degraded overall urban service quality

Why Control Is Hard

Interdependent intersections

Each signal affects downstream flows across the entire network

Time-varying demand

Traffic patterns shift by hour, day type, and local events

Conflicting objectives

Tram priority vs. private vehicle efficiency cannot both be fully maximized

Real-world case: The Florence urban area — 19 signalized intersections, active tram lines, a major train station, university campus, and hospital — all monitored via city-wide traffic platforms.

Connection with Neurosymbolic AI

These two papers sit at opposite ends of the symbolic–neural spectrum, making them ideal lenses for exploring the core neurosymbolic question: *what should be encoded explicitly, and what should be learned?*

🧠 Symbolic Side — Paper 1 (GA)

- Objectives and constraints defined a priori
- Explicit tram priority rules and penalties
- Synchronization requirements between signals
- Safety conditions as hard structural constraints

⚡ Neural / Adaptive Side — Paper 2 (DRL)

- Policy learning from environmental interaction
- Data-driven state–action–reward optimization
- No explicit encoding of traffic rules
- Emergent coordination across agents

Key Question: In safety-critical domains like urban traffic, which aspects of desired behavior must be imposed as explicit constraints — and which can safely emerge from learning?

Lecture Roadmap

01

Problem Formulation & KPIs

Inputs, outputs, objectives, and constraints that define the traffic control problem

02

Paper 1 — Multi-Objective GA / NSGA

MaMoTLO framework: macroscopic optimization with tram priority and Pareto-front solutions

03

Paper 2 — Deep RL Approaches

SADRL, MADRL, and SMART: single- and multi-agent reinforcement learning for adaptive control

04

Comparative Discussion

Head-to-head evaluation on Florence data — where each paradigm wins and why

05

Neurosymbolic Interpretation & Limits

Methodological reading, open questions, and directions for future research

Problem Formulation

Inputs
- Road network topology
- Signal phase structures
- Observed traffic flows
- Tram schedules & arrivals

Objectives
- Minimize travel time
- Minimize waiting time
- Reduce stop frequency
- Guarantee tram priority

Outputs
- Optimized timing plans *(GA)*
- Adaptive control policies *(RL)*

Constraints
- Safety (conflicting phases)
- Queue length limits
- Cross-intersection coordination
- Operational phase sequencing

This is inherently a **multi-objective, dynamic problem** — not a simple scalar optimization. The interplay between tram priority, network coordination, and time-varying demand makes it a rich testbed for both symbolic and learned control strategies.

$S = \{Roads, Intersections, TrafficLights, TramWay, TrafficInformation, FromDateTime, ToDateTime\}$

where:

- $Roads = \{r_i : i = 1, 2, \ldots, n\}$ are the roads (road graph all details) with their characteristics in terms of length, number of lanes, starting and ending nodes (respectively, denoted by $L, N, node_I, node_F$). Then, $r_i = r_i(L_i, N_i, node_{Ii}, node_{Fi})$
- $Intersections = \{node_j : j = 1, 2, \ldots, k\}$ are the intersections having constraints in terms of incoming roads, outcoming roads and related turning percentage (respectively, denoted by $roadInput, roadOutput, P$). Then, $node_j = node_j(roadInput, roadOutput, P)$.
- $TrafficLights = \{si\,g_j : j = 1, 2, \ldots, k\}$ are the positions and features of traffic lights at junctions. More precisely, $si\,g_j$ may contain 0 or 1 traffic light signal. When $si\,g_j$ has 1 traffic light signal, it may have different phases that are composed by states as described above. The states are related to green/yellow and red time durations for each specific direction.
- $TramWay = \{tr_i : i = 1, 2, \ldots, h\}$ is the description of tramway paths/lines passing in a specific intersection *j* with a given time interval.
- *TrafficInformation*, $\mathrm{TI}_{\bar{\Delta}t} = \{(\mathrm{F}_i(\bar{\Delta}t), v(\bar{\Delta}t)_i) \quad \forall i \in roads\}$ are the values of traffic flow for each entering traffic flow road segment of the scenario in terms of [# vehicles/h], and mean vehicular speed *v* [km/h] on inflow road in the scenario at specific time interval time (1-hour)$(\bar{\Delta}t)$.
- *From DateTime to ToDateTime*, that is, the interval time from the start time to the end time in which the scenario *S* is performed.

Key Performance Indicators

Metrics in Focus

Mean Travel Time (MTT)

Primary KPI in the RL paper — extracted from SUMO tripinfo files. Reflects end-to-end journey efficiency.

Mean Waiting Time (MWT)

Time vehicles spend stationary at signals. Directly linked to driver experience and emissions.

Number of Stops

Counts interruptions per vehicle. Central in the GA paper as one explicit optimization objective.

Tram Priority Performance

Regularity and headway compliance for tram services — a critical constraint in both papers.

An Important Distinction

In the **GA paper**, metrics are built directly into the multi-objective fitness function — the problem is explicitly structured around them.

In the **RL paper**, training uses a local reward signal (vehicles stopped at an intersection), but the final evaluation uses the global MTT metric.

Reward ≠ Evaluation Metric. This mismatch is a critical methodological point: an agent can optimize its reward without fully optimizing the KPI stakeholders actually care about.

Mean Travel Time (MTT)

MTT is the primary performance indicator. It is not derived from a closed-form expression but computed **via SUMO simulation** as the average travel time across all vehicles.

$$MTT = \frac{1}{N}\sum_{v=1}^{N} TT_v$$

TT_v

Travel time of individual vehicle or tram *v*

N

Total number of vehicles (or trams) in the simulation

Includes

Both **time in motion** and **waiting time at signals**

Mean Waiting Delay (MWD)

Formalized in the paper as **Mean Waiting Delay**, MWD quantifies the average red-phase delay experienced by vehicles across all incoming road segments of an intersection.

Per-Road Waiting Delay

$$WD_i = \frac{CT(1-R_i)^2}{2(1-R_i * s)}$$

CT

Cycle time of the signal

$R_i = G_i / CT$

Green ratio for road *i*

$s = d_i / d_{max}$

Degree of saturation

Mean Waiting Delay (Aggregated)

$$WD_i = \frac{CT}{2} \cdot \frac{(1-R_i)^2}{1-R_i \cdot s}$$

Where *I* is the set of all incoming road segments at the intersection.

This is the **official formula used in the optimization objective**.

Interpreting MWD

Green Ratio (Ri)

Higher green ratio reduces waiting delay quadratically — allocating green time is the primary lever for MWD reduction.

Cycle Time (CT)

Longer cycles can reduce stop frequency but increase per-stop delay — a core trade-off in signal optimization.

Saturation (s)

As roads approach saturation, delay increases nonlinearly — capturing congestion effects in the denominator.

Number of Stops (NS & TNS)

Stop count KPIs measure how frequently vehicles are forced to halt — a key indicator of signal efficiency and passenger comfort.

Stops on Road *i*

$$NS_i = F_i \cdot \frac{F_{\max} - F_i}{F_{\max}} \cdot \frac{CT - G_i}{t}$$

F_i	F_{max}
Traffic flow on road *i* (veh/h)	Saturation flow rate
G_i	t
Green time allocated to road *i*	Total simulation time

Total Stops (TNS)

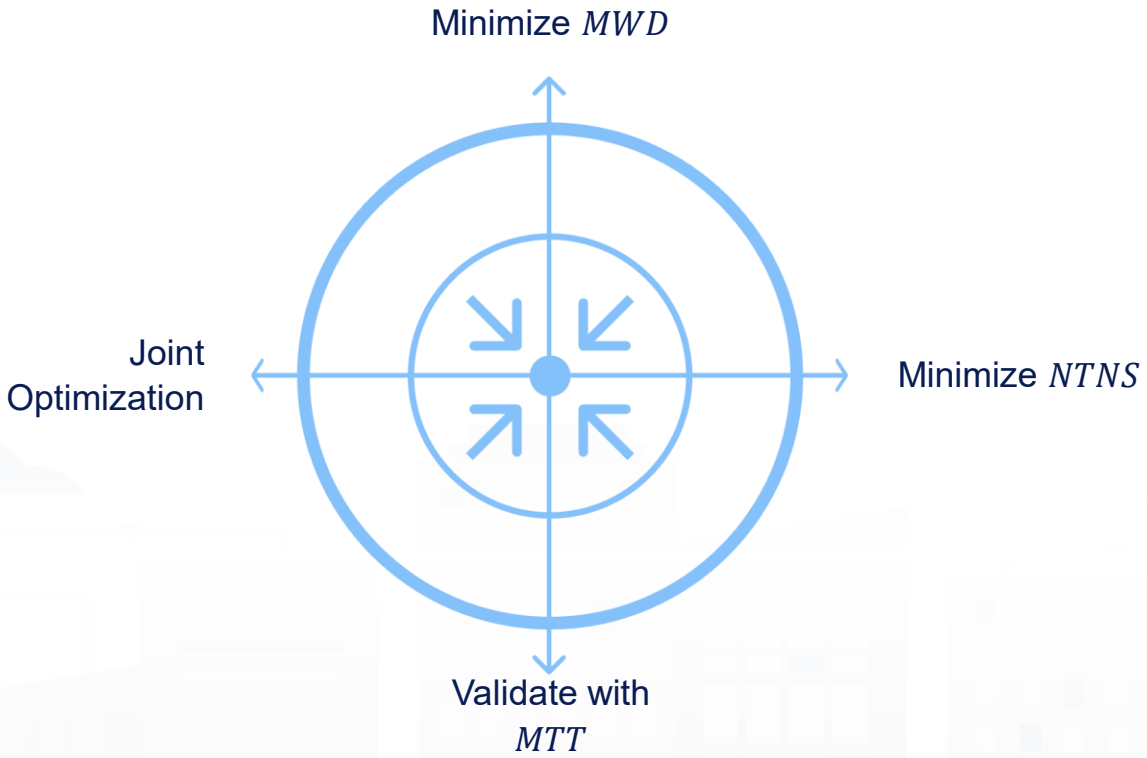
$$TNS = \sum_{i \in I} NS_i$$

Normalized Stops (NTNS) — used in optimization

$$NTNS = \frac{1}{CT} \sum_{i \in I} NS_i$$

$NTNS$ normalizes by cycle time, enabling fair comparison across different signal timing configurations.

Multi-Objective Optimization Framework



Why Separate Analytical and Simulated KPIs?

MWD and NTNS admit closed-form expressions, making them tractable for gradient-based or symbolic optimization. MTT requires simulation to capture full network dynamics.

The optimization jointly minimizes MWD and NTNS analytically, then validates outcomes using SUMO-simulated MTT.

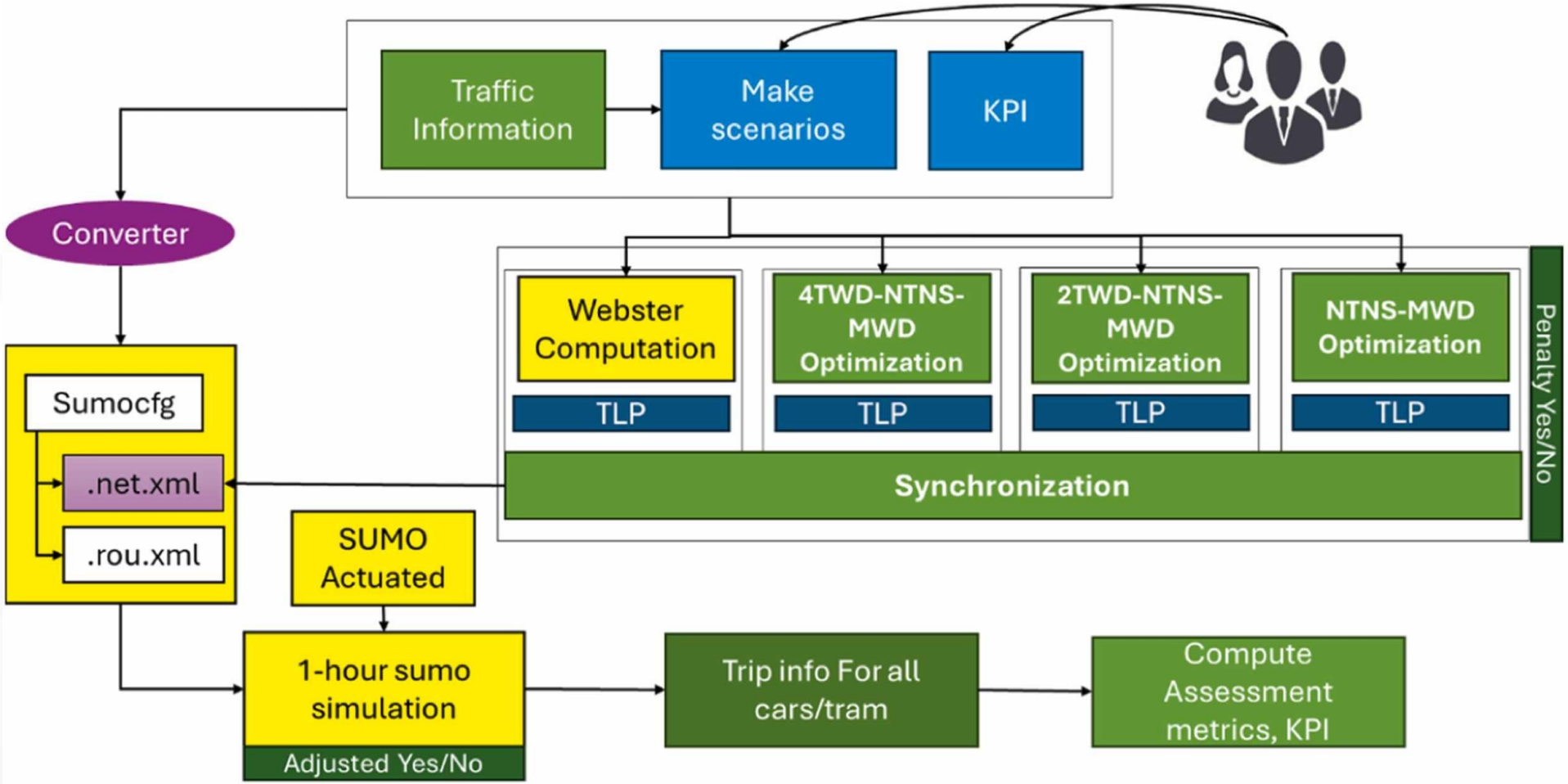
Analytical

$MWD, NTNS$ — used during optimization loop

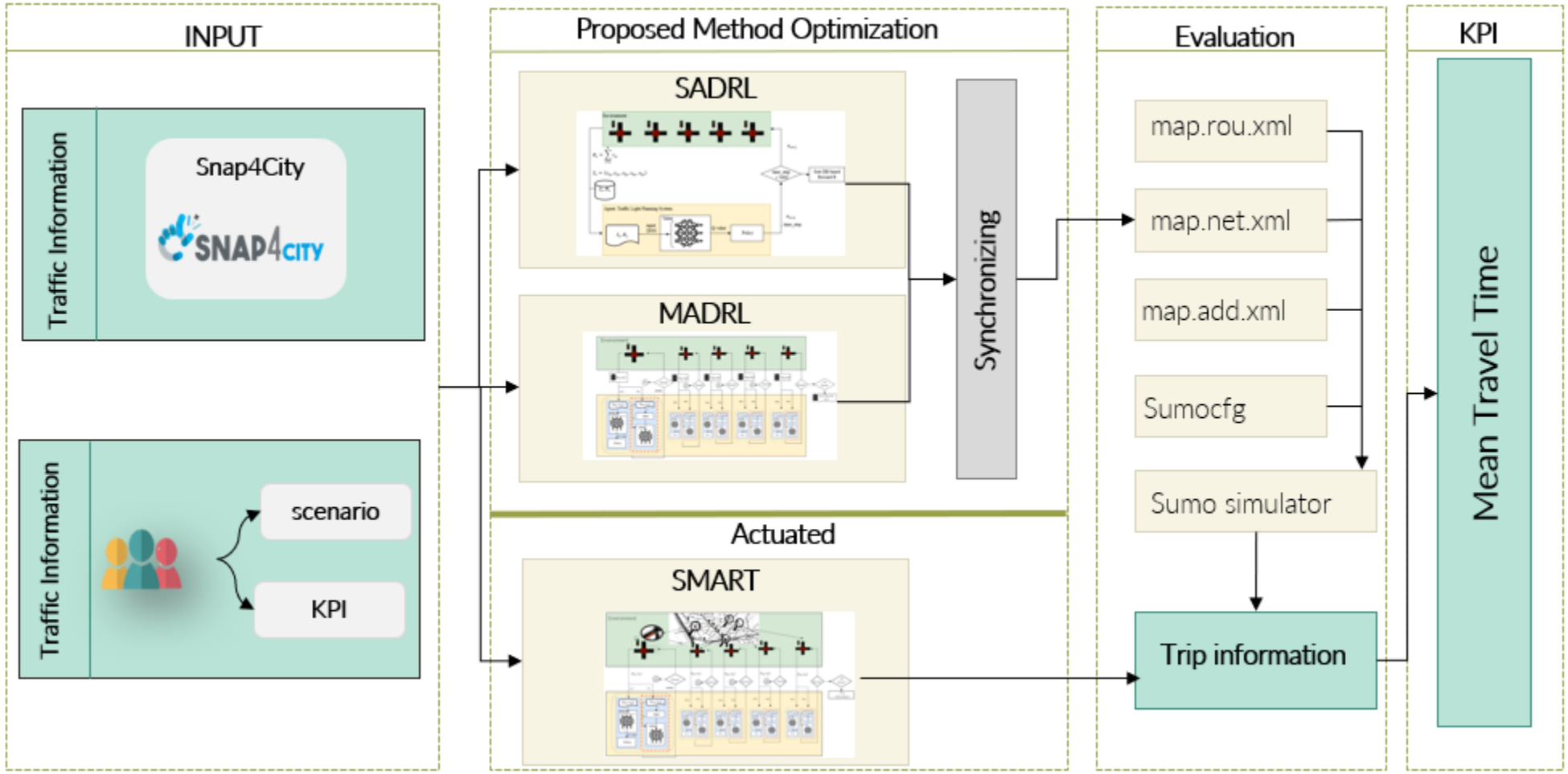
Simulation

MTT via SUMO — used for evaluation and validation

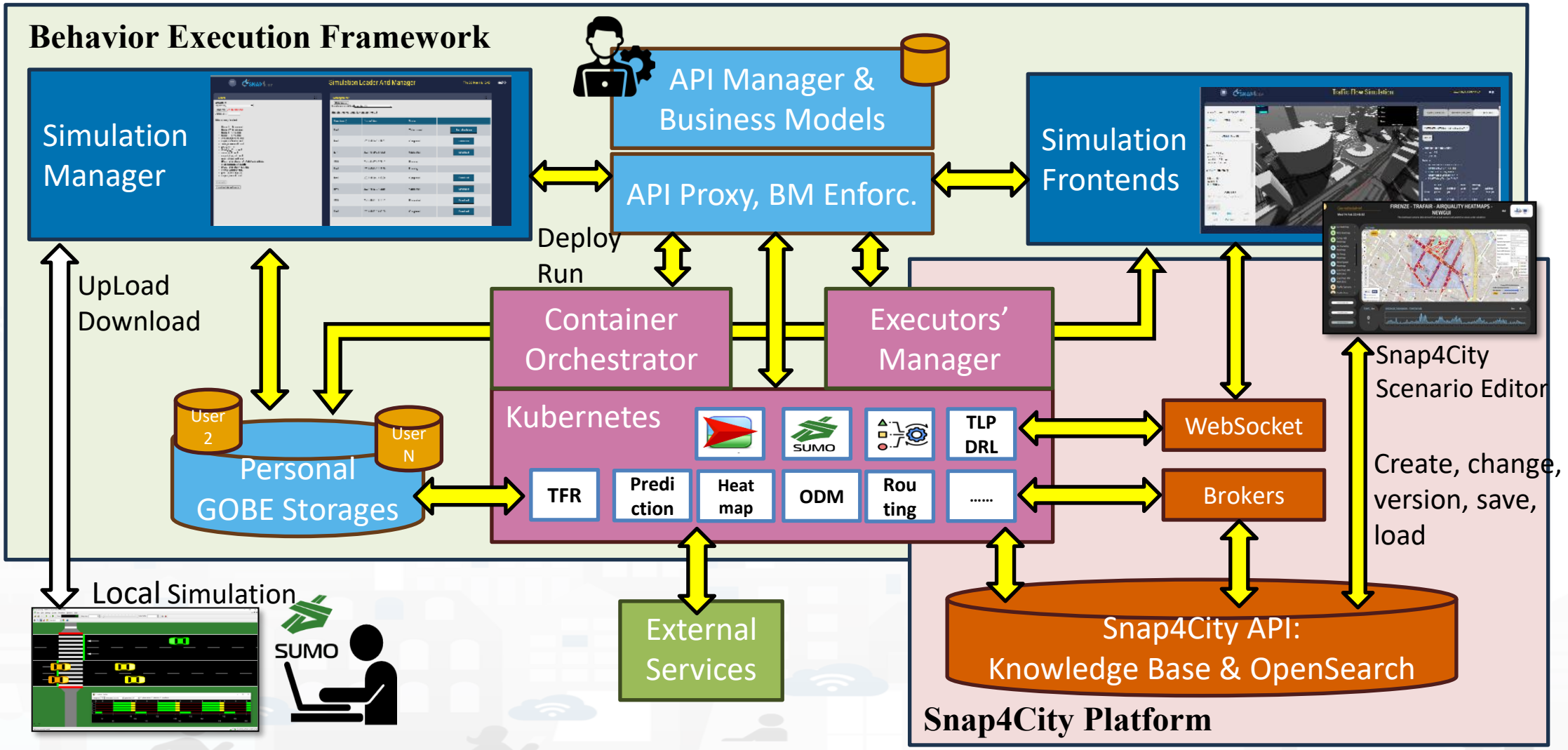
Integrated data flows



Integrated data flows X RL



Behavior Execution Framework for Digital Twins



Paper 1: MaMoTLO — General Idea

MaMoTLO (Macroscopic Multi-Objective Traffic Light Optimization) is a GA-based framework designed for coordinated, network-level signal control in urban environments with mixed traffic including trams.

Multi-Objective Optimization

Simultaneously optimizes travel time, waiting time, stop count, and fairness across directions — producing a Pareto front of trade-off solutions rather than a single answer.

Tram Priority Integration

Explicit penalties and phase synchronization rules ensure tram passages are respected. Priority is not emergent — it is structurally encoded in the fitness function.

Network-Level Coordination

Optimizes across all 19 Florence intersections simultaneously, capturing cross-intersection flow dependencies that local per-signal methods miss.

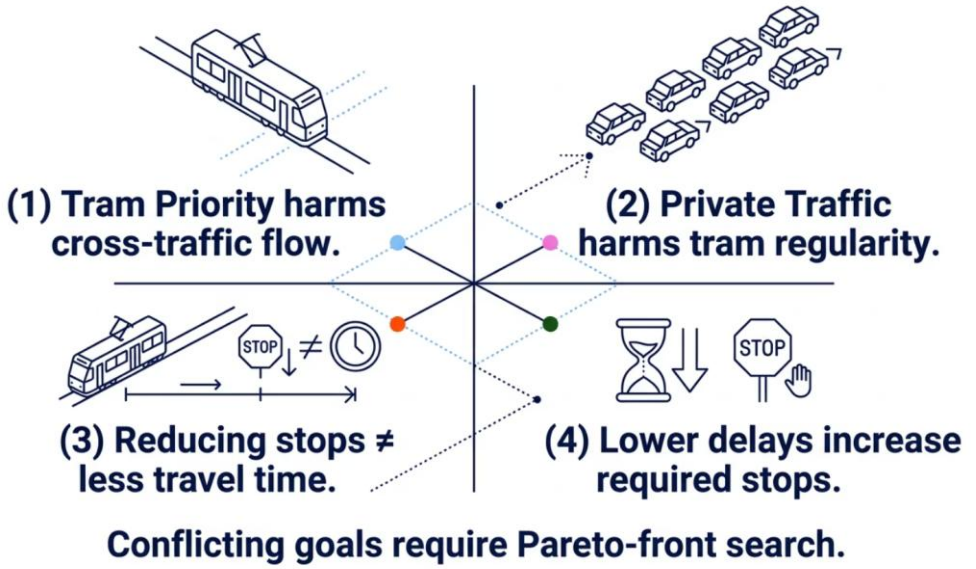
Strong Baseline Comparisons

Evaluated against Webster timing, SUMO actuated control, and prior NSGA variants — demonstrating clear incremental value at each step.

Why a Multi-Objective Genetic Approach?

Traffic light objectives are **fundamentally in conflict**. Aggressively favoring the tram increases delays on cross streets. Minimizing stops does not guarantee minimizing overall travel time. Optimizing private vehicle flow can harm public transport regularity.

Collapsing these into a single scalar metric forces an arbitrary trade-off before optimization even begins. Pareto-optimal solutions instead make the trade-off explicit and navigable by decision-makers.



NSGA-II is the established standard for two- to three-objective problems. NSGA-III extends this to higher-dimensional objective spaces, maintaining diversity along the full Pareto front — particularly important as the number of conflicting goals grows.

MaMoTLO Solution Families

MaMoTLO organizes its solutions into three base families, each extended by behavioral variants — yielding **9 new solutions** compared against 5 state-of-the-art baselines.

Base Families

NTNS-MWD

Optimizes number of stops and waiting delay — no directional priority bias.

2TWD-NTNS-MWD

Extends the problem to two priority traffic directions.

4TWD-NTNS-MWD

Adds four priority directions for richer, more balanced network control.

Behavioral Variants

P — Penalty

Encodes tram priority directly into the optimization objective via a penalty term.

A — Adjust on Demand

Dynamically adapts green/red phases when a tram is detected approaching.

P-A — Penalty + Adjust

Combines structured offline penalty with real-time local adaptation.

Penalty and Adjust-on-Demand

The two variant mechanisms represent distinct but complementary philosophies of tram prioritization — one structural, one reactive.

Penalty (P): Structured Priority

Tram preference is encoded directly into the solution evaluation function. Priority is not emergent or coincidental — it is baked into the optimization itself, ensuring the planner consistently favors tram throughput across candidate solutions.

Adjust on Demand (A): Reactive Flexibility

When a tram approaches, the system shortens competing red phases or extends green phases in the tram's direction. This introduces local, real-time adaptivity without requiring a full re-optimization cycle.

P-A: A Hybrid Paradigm

Combines offline structured planning with operational flexibility. This is conceptually a proto-hybrid architecture — not yet learning-based, but already bridging static optimization and dynamic control.

Key Insight: MaMoTLO is not a rigid static plan — it is a framework that unifies *optimized planning* with *operational flexibility*, setting the stage for the RL comparison in Paper 2.

Objectives and Constraints in MaMoTLO

The modeling richness of MaMoTLO lies in its explicit, multi-layered treatment of both optimization targets and real-world operational constraints across a coordinated intersection network.

Optimization Objectives

Travel Time

Minimize overall vehicle travel time across the network.

Waiting Delay

Reduce cumulative delay at signalized intersections.

Number of Stops

Minimize unnecessary stopping events for all vehicle classes.

Operational Constraints

Queue Management

Upstream and downstream queue dynamics are explicitly modeled — no intersection is treated in isolation.

Traffic Flow Coordination

Green wave progression along arterials must be preserved across coordinated signals.

Tram Green Phase Inclusion

Tram priority phases are embedded as hard constraints, not cosmetic additions, ensuring public transit is systematically favored.

Optimization Method: NSGA-II and NSGA-III

MaMoTLO uses Non-dominated Sorting Genetic Algorithms to search a multi-objective solution space, leveraging population-based evolutionary operators and Pareto dominance ranking.

NSGA-II

Well-suited for 2–3 objective problems. Highly sensitive to initial population size. Uses crowding distance to maintain solution diversity on the Pareto front.

NSGA-III

Designed for higher-dimensional objective spaces (4+ objectives). Uses structured reference points to distribute solutions uniformly across the Pareto front — essential when objectives scale beyond standard bi-criteria problems.

Case Study and Evaluation Setup

The experimental validation is grounded in a real, non-synthetic urban scenario — a portion of the city of Florence featuring heavy private traffic and active tram lines.

01

Real Urban Network — Florence

Intersection topology, tram corridors, and traffic demand are drawn from real-world operational data, not stylized test environments.

02

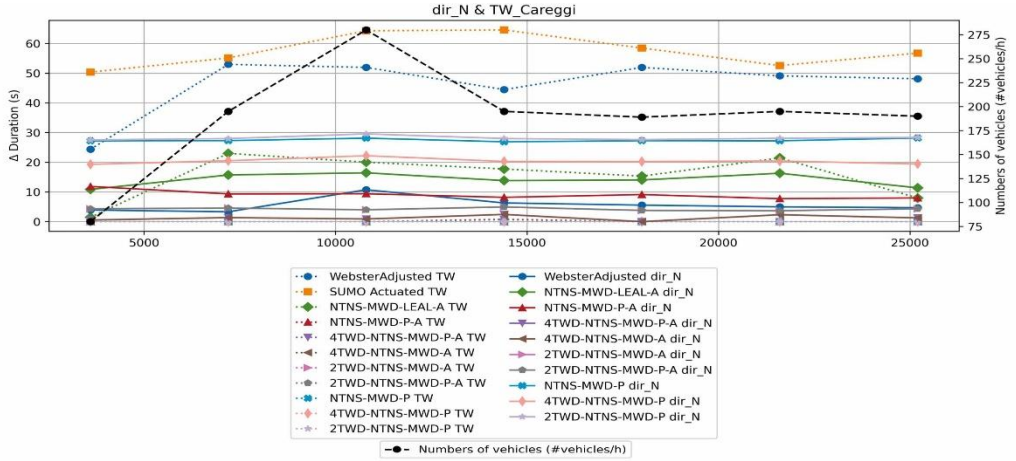
Macroscopic Planning, Microscopic Evaluation

Timing plans are designed at the network level, then evaluated in high-fidelity microsimulation using **SUMO** and the **TraCI** API — enabling global design with detailed testing.

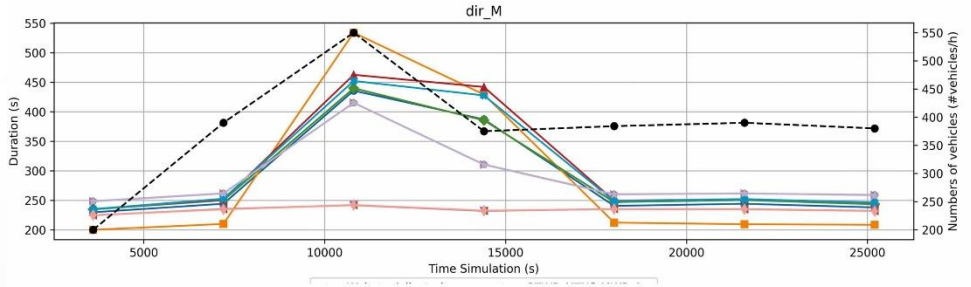
03

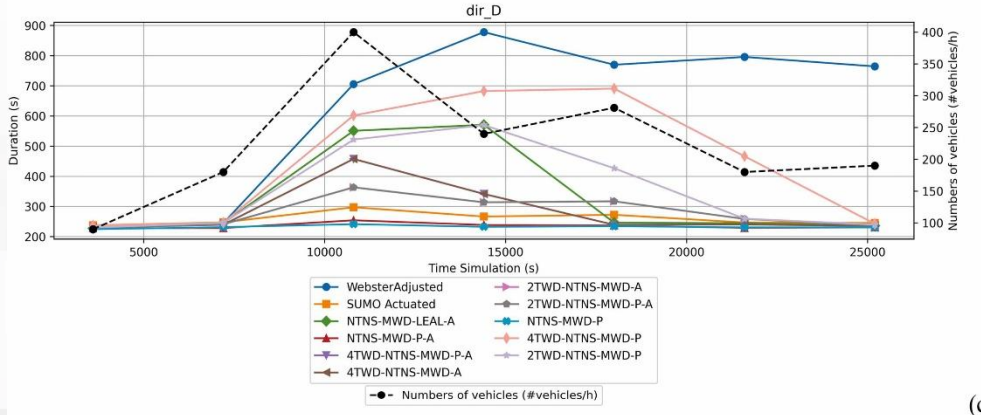
Strong Baseline Comparisons

Benchmarks include Webster timing, SUMO actuated control, and other multi-objective or genetic variants already established in the literature — not trivial strawmen.



(a)





(c)(d)

Mean Travel Time (MTT)

Traffic Load	4TWD-NTNS-MWD-A	SUMO Actuated	Webster
1.0	3,013.85 ✅	2,935.41	5,188.87 ❌
1.5	3,242.71 ✅	3,409.13	6,474.95 ❌
2.0	3,457.86 ✅	4,666.32 ❌	7,636.76 ❌

Values in seconds. Lower is better.

GA Dominates

The GA method achieves the lowest MTT at every load level, maintaining a stable trajectory even as congestion doubles.

Webster Degrades Sharply

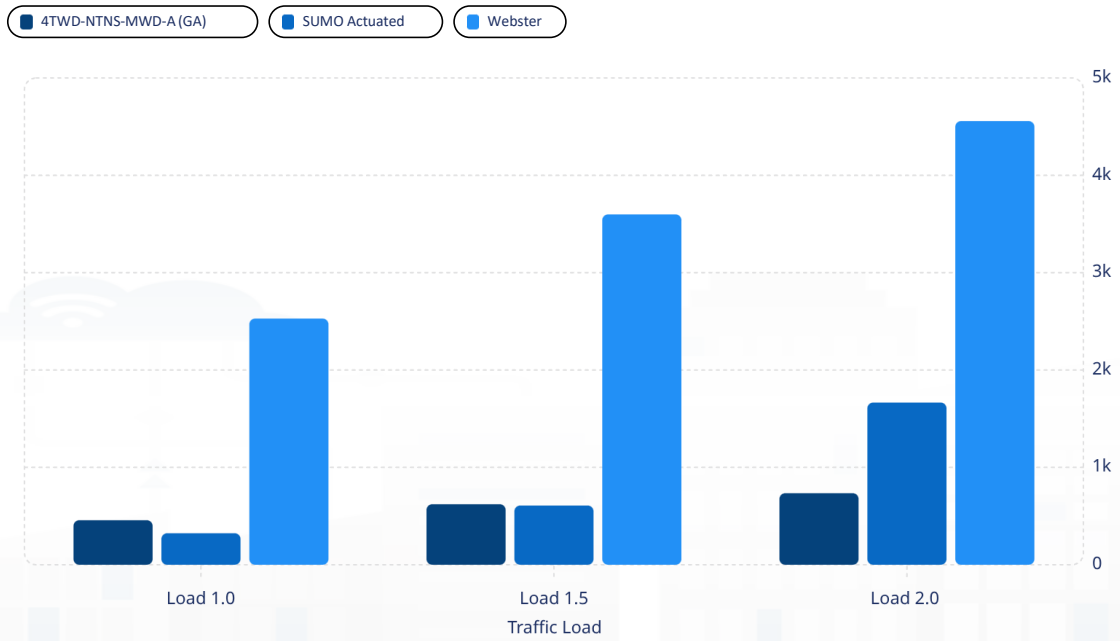
Webster's fixed timing fails under increasing demand — travel time nearly doubles from load 1.0 to 2.0, exposing its core limitation.

Mean Number of Stops (MNS)

Traffic Load	4TWD-NTNS-MWD-A	SUMO Actuated	Webster
1.0	21.99 ✅	38.55 ❌	43.85 ❌
1.5	32.07 ✅	56.08 ❌	72.14 ❌
2.0	41.61 ✅	78.68 ❌	101.10 ❌

The GA method reduces vehicle stops by up to **59% vs. SUMO** and **59% vs. Webster** at high traffic load — a dramatic reduction in both emissions and driver frustration.

Mean Waiting Time (MWT)



Low Load (1.0)

SUMO Actuated edges out GA slightly (323 vs. 457 sec). Both vastly outperform Webster (2,528 sec).

Medium–High Load (1.5–2.0)

GA takes the lead decisively. SUMO degrades rapidly, approaching 1,666 sec at load 2.0.

Webster — Consistent Worst

Waiting times balloon to over 4,500 sec at peak load. Fixed-time control is clearly insufficient under dynamic demand.

Main Results of the GA Paper

The most advanced MaMoTLO variants demonstrate substantial, well-balanced improvements over all baselines — with robustness across varying traffic scenarios being a defining characteristic.

9

New Solutions

Generated by MaMoTLO across 3 families × 3 variants

5

Best results Baselines

Including Webster, SUMO actuated, and other GA/MO methods

10%+

Typical Improvement

Reduction in travel time, delay, and stops over classical methods

Top Performers

- **4TWD-NTNS-MWD-A** — four-direction model with Adjust on Demand
- **4TWD-NTNS-MWD-P-A** — full Penalty + Adjust combination

Why Balance Matters

A solution that is optimal on average but severely suboptimal for one direction or user class is not operationally acceptable in urban traffic management. The top MaMoTLO variants maintain equity across all directions under diverse load conditions.

Strengths and Limits of the GA Approach

✅ Strengths

- **Explicit modeling** — objectives and constraints are transparent and inspectable
- **Interpretable objectives** — behavior of solutions is reasonably explainable
- **Strong tram-priority design** — public transit is structurally embedded, not an afterthought
- **Robust optimization** — top variants generalize well across traffic scenarios

⚠️ Limits

- **Simulation-dependent** — quality of results is bounded by simulation fidelity
- **Costly offline search** — many simulation evaluations required per optimization run
- **Parameter sensitivity** — population size, mutation rate, and crossover all affect outcomes significantly
- **Limited online adaptivity** — even with Adjust on Demand, the system is less naturally reactive than a continuously learned control policy

These limitations motivate the transition to the second paper's paradigm: instead of searching for timing plans offline, can a system *learn* a control policy directly from traffic dynamics?

From Optimized Plans to Learned Policies

The conceptual shift between Paper 1 and Paper 2 is one of the most important transitions in this module — from *searching* for good solutions to *learning* how to make good decisions.

1

GA Approach

Search over a space of timing plans. Evaluate candidate solutions offline. Select the best plan before deployment.

2

Paradigm Shift

Move from offline optimization to sequential decision-making. The agent observes traffic state and selects actions in real time.

3

RL Approach

Learn a control policy mapping traffic states to signal actions. Adapt continuously to evolving dynamics — no pre-computed plan required.

Motivation: Rather than constructing everything a priori, can a system learn *how* to control traffic adaptively — responding to real-time demand, tram arrivals, and unexpected congestion without manual re-optimization?

Paper 2: Deep Reinforcement Learning Approaches

The second paper introduces three DRL-based control paradigms applied to the same Florence network, enabling direct comparison with MaMoTLO's genetic optimization approach.

SADRL

Single-Agent Deep RL. One centralized agent controls the entire signal network. Learns a global policy from aggregate traffic observations.

MADRL

Multi-Agent Deep RL. Each intersection (or cluster) has its own agent. Agents interact and coordinate through shared environment dynamics.

SMART

Reactive real-time actuation. A more responsive variant enabling finer-grained online adaptation to live traffic states and tram arrivals.

Shared Goals

- Reduce network congestion
- Improve directional fairness
- Prioritize tram traffic

Evaluation Baselines

All three DRL approaches are benchmarked against **Webster**, **SUMO actuated**, and crucially, the **MaMoTLO** solutions from Paper 1 — positioning this as a direct family-to-family comparison between evolutionary optimization and adaptive learning.

RL/DRL Refresh Traffic Control

Reinforcement learning defines an agent that observes a **state**, selects an **action**, receives a **reward**, and updates its **policy** based on accumulated experience. In deep RL, value functions or policies are approximated by neural networks, enabling operation over high-dimensional, complex state spaces.

Agent

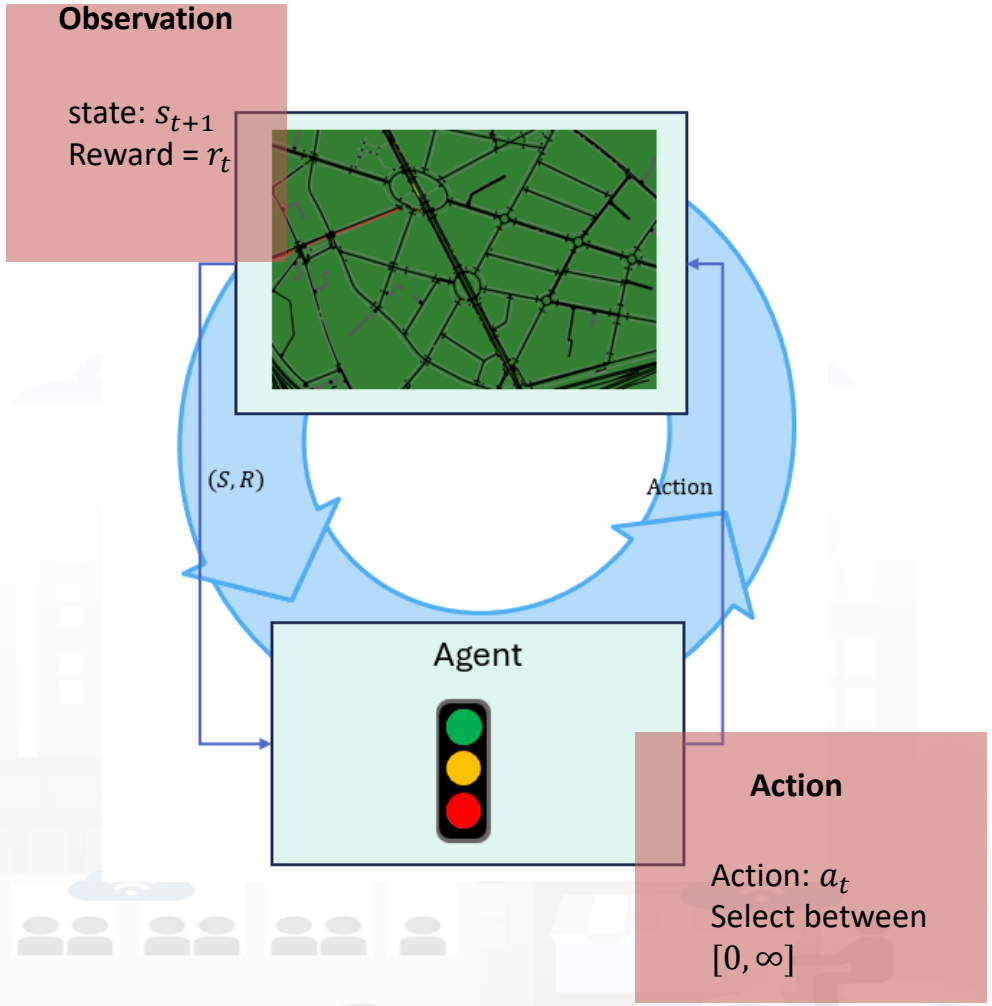
Traffic signal controller — selects phase durations based on observed conditions

Environment

Urban road network simulated in SUMO — responds to agent actions with state transitions

Reward

Penalizes congestion — fewer halted vehicles, smoother flow, and tram-priority adherence

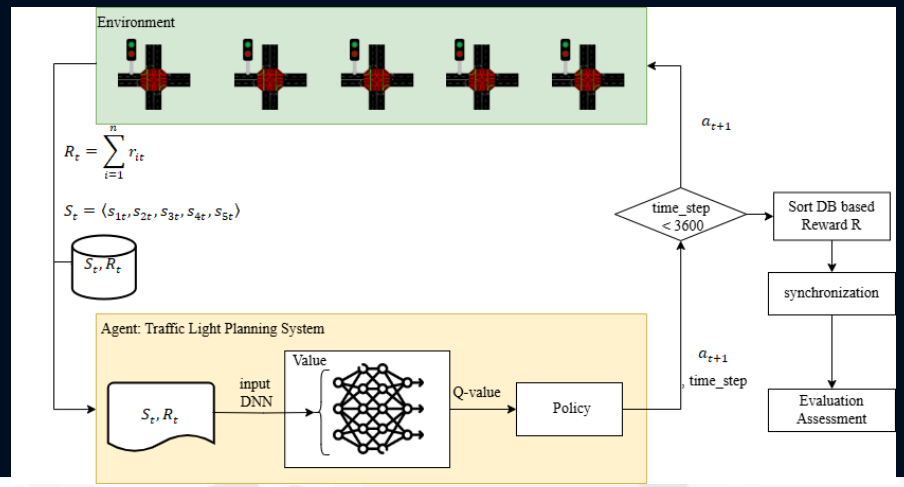


SADRL: Single-Agent Deep Reinforcement Learning

Architecture

A single agent controls the entire traffic network, representing the full joint state and issuing a unified action over all regulated intersections.

In the fixed-cycle variant, the cycle duration (e.g., 60 seconds) is predetermined, and the agent's task is to optimally distribute green time across phases — maintaining compatibility with existing infrastructure.



Trade-offs at a Glance

✅ Centralized Coordination

Global visibility enables theoretically optimal system-wide decisions

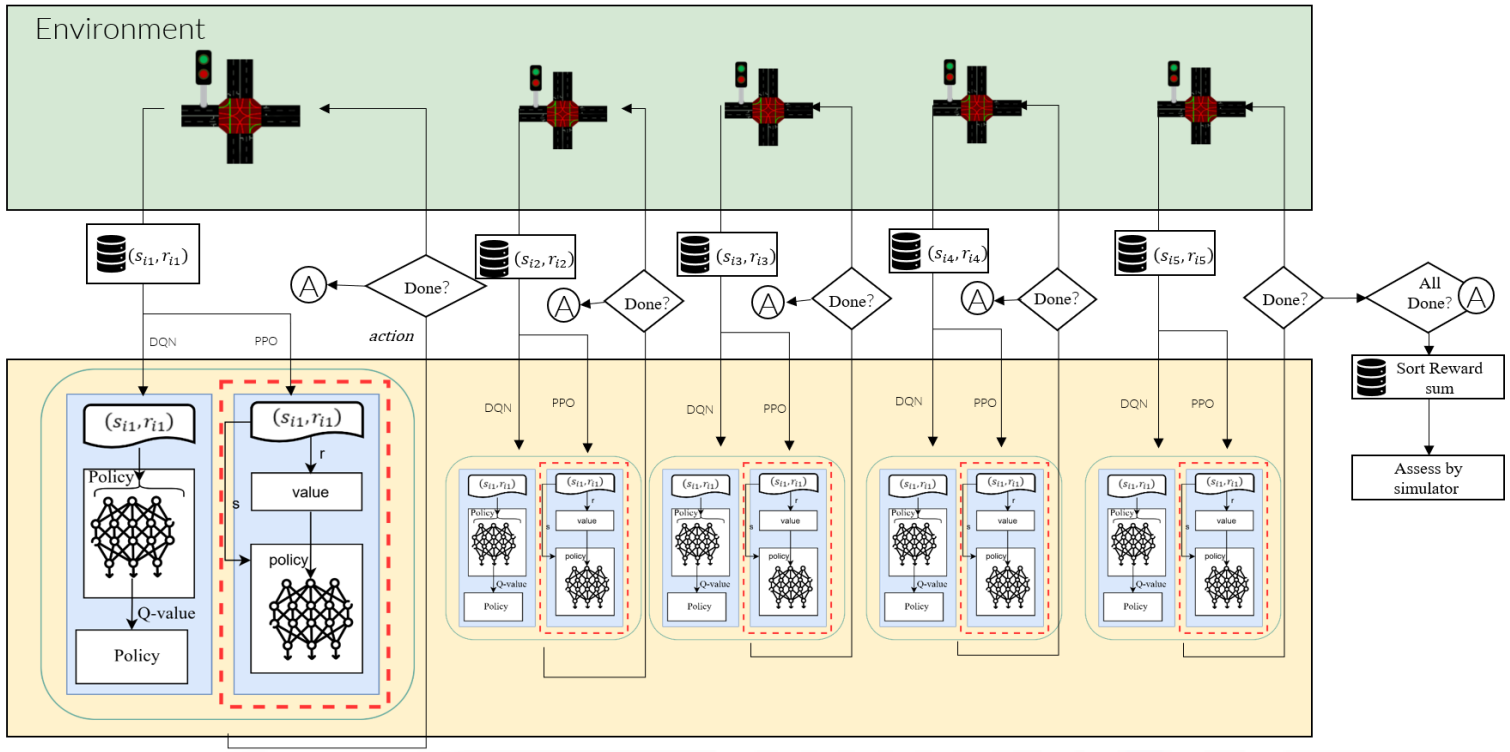
⚠️ Scalability Limits

State and action spaces grow sharply with intersection count

⚠️ Training Complexity

High-dimensional joint spaces make convergence slower and computationally expensive

MADRL: Multi-Agent Deep Reinforcement Learning



Rather than a single centralized controller, MADRL assigns an independent agent to each intersection or local cluster. Each agent observes **local traffic conditions** and reacts rapidly, producing a naturally scalable architecture well-suited to complex urban networks.

Distributed Observation

Each agent perceives only its local environment — queue lengths, speeds, and signal state — enabling fast, reactive decisions without global communication overhead.

Improved Scalability

Adding intersections adds agents rather than expanding a monolithic state space. PPO-based MADRL configurations show strong adaptability, especially under **heavy traffic loads**.

Coordination Challenge

Purely local optimization can produce globally suboptimal behavior. Agents that ignore neighboring conditions risk creating bottlenecks at the network level.

SMART: Real-Time Adaptive Signal Control

What Makes SMART Different

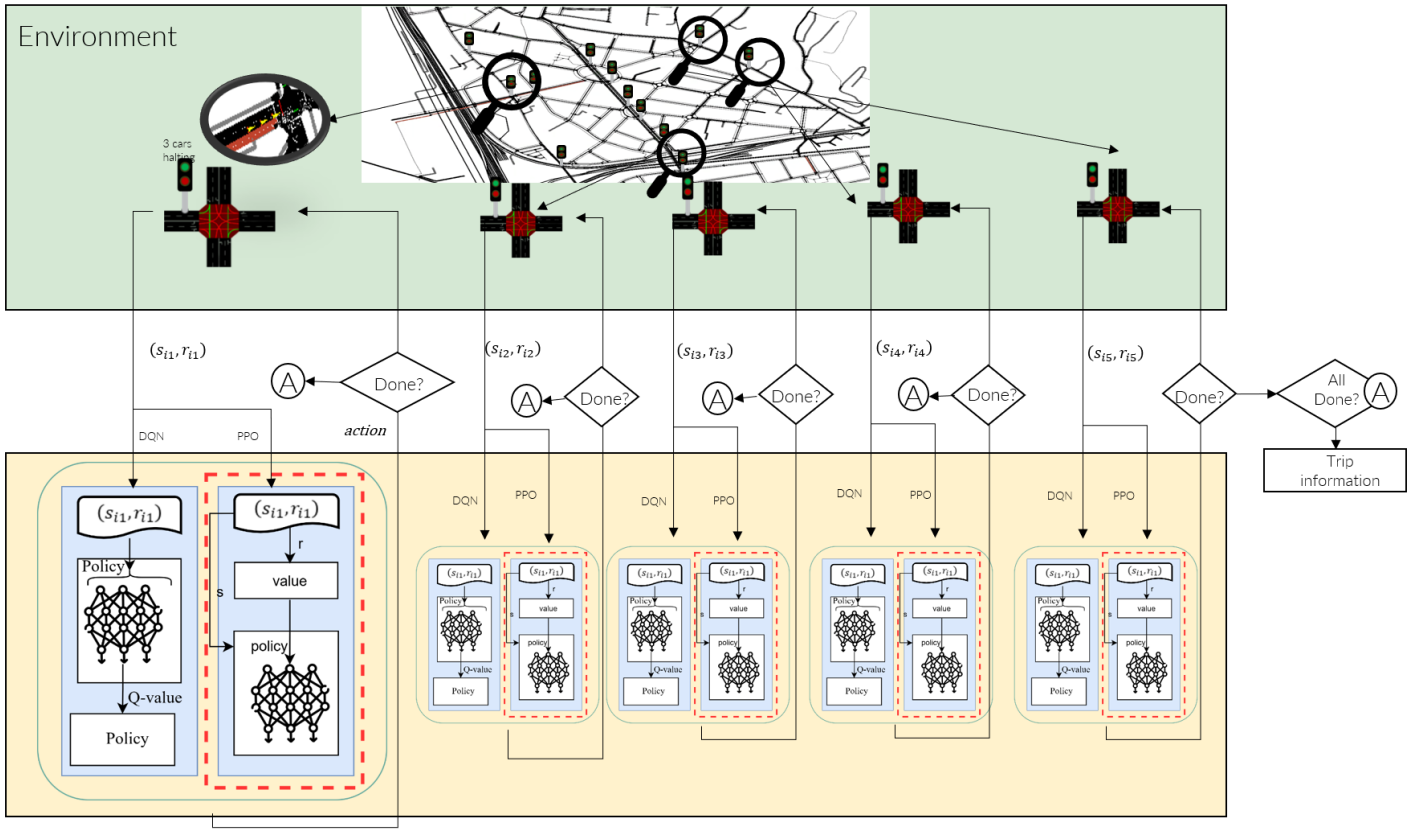
SMART is the most dynamic system in the RL paper benchmark. Rather than distributing green time within a fixed cycle, it applies a **learned, actuated policy** that responds in real time to current traffic conditions — not to hard-coded rules.

This distinguishes it from **SUMO actuated**, which is also dynamic but relies on simulator-defined rule logic rather than trained behavior. SMART's adaptive decisions emerge from the training process itself.

Performance vs. Cost

Strong performance — SMART A consistently achieves top results on Mean Travel Time and tram prioritization across traffic conditions.

High computational cost — A single one-hour simulation can require **over 24 hours** of wall-clock computation, limiting practical deployment at scale.



State, Reward & Evaluation in the RL Paper

A critical methodological distinction separates the **training signal** from the **evaluation metric**. The reward drives learning via a proxy; the evaluation metric tests whether that learning generalizes to a meaningful real-world objective.

1

Observation

Halted vehicles, average speed, and current signal phase status observed at each timestep by the agent

2

Reward Signal

Penalizes the number of stopped cars — a local proxy for congestion minimization that shapes policy during training

3

Evaluation Metrics

Mean Travel Time (MTT) from SUMO tripinfo logs, plus fairness across directions and tram-priority adherence

Training reward and final evaluation metric intentionally differ — a methodologically principled choice that reveals how well proxy-trained policies transfer to application-level goals.

Experimental Setup of the RL Paper

Study Area & Data

A strategic zone of **Florence, Italy** encompassing 19 traffic lights near sensitive nodes — the central station, university campus, major hospital, and the tram network. Real traffic flow data was collected every 10 minutes via the Snap4City platform using traffic flow reconstruction techniques.

Experimental Conditions

Normal Load

Baseline urban traffic; standard signal timing applies

Medium Load

Elevated congestion; tests robustness under moderate demand

Heavy Load

Peak stress scenario; highlights scalability and adaptivity limits

Each condition was also run under an **adjusted tram-priority configuration** to evaluate coexistence with public transit priority logic.

MTT Results: Full Comparison Across Traffic Loads

Mean Travel Time (in seconds) measured at three traffic load levels. Lower is better. Green = best-in-class performance.

Traffic Load	SMART (RL)	MADRL (RL)	SADRL (RL)	MamoTLO (GA)	SUMO	Webster
1.0×	2599.13	2910.76	2760.12	3013.85	2935.41	5188.87
1.5×	2277.58	3229.54	3189.84	3242.71	3409.13	6474.95
2.0↕	3443.96	3301.19	3490.87	3457.86	4666.32	7636.76

Key Takeaway: SMART (RL) achieves the lowest MTT at all load levels. At 2.0× load, MADRL closes the gap significantly — suggesting strong scalability under heavy congestion.

Method Comparison: Performance, Scalability, and Cost

No single method dominates on all three dimensions. The right choice depends on deployment constraints — available compute, network size, and acceptable latency.

Method	Performance	Scalability	Cost
SMART (RL)	Top		
MADRL (RL)			
SADRL (RL)			
MamoTLO (GA)			
SUMO			
Webster			

Research Context

Prioritize SMART (RL) for benchmarking upper bounds

Urban Deployment

MADRL or MamoTLO offer the best real-world balance

Robustness Under Traffic Stress

How does each method hold up as congestion escalates from low to high load? Robustness is a critical criterion for real-world deployment.

Method	Low	Medium	High
SMART (RL)	🟢	🟢	🟢
MADRL (RL)	🟡	🟡	🟢
SADRL (RL)	🟢	🟡	🟡
MamoTLO (GA)	🟢	🟢	🟢
SUMO	🟡	🟡	🔴
Webster	🔴	🔴	🔴

1

Most Robust

SMART and **MamoTLO (GA)** maintain green ratings at all load levels — full-spectrum resilience.

2

Improves Under Pressure

MADRL is uniquely stronger at high load — multi-agent coordination activates under congestion.

3

Most Fragile

Webster fails across all load levels. **SUMO** degrades severely at high congestion.

Interpreting the Results: RL vs. GA

Each method occupies a distinct niche — understanding the trade-offs is essential for practical deployment.

SMART (RL Actuated)

Best global MTT across all load levels. However, its **extremely high computational cost** limits real-world applicability. Ideal as a performance upper bound in research settings.

MADRL (Multi-Agent RL)

Excellent performance under high traffic. Scales gracefully across intersections, making it the **most realistic RL candidate** for deployment in urban networks.

SADRL (Single-Agent RL)

Competitive at low traffic loads, but **degrades under high congestion**. Single-agent architecture limits coordination in complex, multi-intersection scenarios.

MamoTLO (GA)

Consistently stable across all load levels. Offers a **strong performance-to-cost ratio** — highly robust without the training overhead of deep RL approaches.

Direct Comparison: GA vs. DRL

Neither paradigm universally dominates. The choice depends on whether the priority is **interpretable constraint satisfaction** or **state-dependent adaptive control**.

Aspect	GA / NSGA	DRL
Nature	Offline multi-objective optimization	Sequential experience-driven learning
Explicit knowledge	High — objectives, constraints, penalties encoded directly	Medium — embedded in reward design and architecture
Constraints	Easy to encode as hard or soft constraints	Harder — requires reward shaping or safety layers
Online adaptivity	Limited — solution fixed after optimization	High — policy responds to real-time state
Interpretability	Higher — Pareto fronts are inspectable	Lower — neural policy is a black box
Computational cost	High during offline search phase	High during training and runtime (esp. SMART)

A Neurosymbolic Reading of the Two Papers

The GA paper anchors the **symbolic pole**: domain knowledge enters as explicit objectives, hard constraints, penalty terms, synchronization logic, and tram priority rules. The RL paper anchors the **neural/adaptive pole**: policy behavior is not hand-designed but emerges from interaction with a simulated environment.

Constrained RL

Embed symbolic constraints directly into the RL training objective or action space, preventing policy violations without reward hacking

Symbolic Reward Shaping

Use domain rules — priority hierarchies, safety margins — to augment the reward signal with structured, interpretable feedback

Safe Action Filtering

A symbolic safety layer filters agent outputs at runtime, blocking actions that would violate tram priority or hard traffic rules

GA-Initialized RL Policies

Use the Pareto front from genetic optimization to seed or guide RL training — combining offline search quality with online adaptivity

- The most promising direction: not choosing between symbolic and neural paradigms, but engineering their integration at the right architectural seam.

Methodological Limits

Before drawing conclusions, it is essential to read both approaches critically. Every research artifact carries inherent constraints — and understanding them is as important as understanding the results themselves.

Simulation Dependence

Both works rely heavily on **SUMO** for evaluation. This is entirely appropriate for research, but the transition to real-world deployment is far from automatic.

Limited Generalizability

Results are validated on a specific portion of the city of **Florence**. The same configurations or learned policies may not transfer without significant re-tuning to other cities or network topologies.

Computational Cost

High computational requirements are especially pronounced in **SMART**, but are not absent in genetic-method pipelines either — a practical barrier to scalable deployment.

Gaps in Traffic Agent Modeling

Current models focus primarily on vehicles and trams. A more complete urban traffic model would need to explicitly account for a broader range of actors and events.

Pedestrians

Foot traffic interactions with signal timing are largely abstracted away, yet they represent a critical safety and flow dimension in dense urban environments.

Cyclists

Bike lanes and mixed-traffic cycling behavior introduce asymmetric dynamics that current models do not adequately represent.

Rare & Emergency Events

Infrequent but high-impact events — ambulances, accidents, large gatherings — remain underrepresented in training and evaluation scenarios.

Environmental Objectives

Emissions, noise, and energy consumption are rarely primary optimization targets, yet they are increasingly central to urban mobility policy.

The Core Challenge

Urban traffic signal control is not a single optimization problem. It sits at the intersection of four fundamental difficulties that make it one of the hardest problems in applied AI.

Multi-Objective

Throughput, safety, tram priority, and emissions must be balanced simultaneously — with no single "correct" tradeoff.

Dynamic

Traffic conditions change continuously and unpredictably across time of day, weather, and incidents.

Constrained

Hard infrastructure limits, legal requirements, and safety guarantees cannot be violated — even for marginal performance gains.

Multi-Agent

Dozens of intersections must coordinate without centralized bottlenecks — a distributed control problem at scale.

Key Takeaways: What Each Paradigm Does Best

Rather than declaring a winner, the goal is to understand the complementary strengths of genetic algorithms and deep reinforcement learning in traffic control contexts.

1

GA / NSGA-II

Strong for explicit planning. Ideal when domain knowledge — objectives, constraints, priorities, and tradeoffs — can be formally expressed. Produces interpretable, auditable solutions.

2

Deep RL

Strong for adaptive control. Excels when the environment is state-dependent and dynamic. Learns policies that react to real-time traffic conditions without requiring manual rule specification.

3

Neurosymbolic Integration

The most promising frontier. Combining symbolic structure with learned behavior — symbolic components encode constraints and domain priors; learned components handle variability and adaptation.

Questions for Discussion

The following open questions are designed to push beyond recall and toward critical reasoning. There are no clean answers — the goal is rigorous debate.

1

Hard Constraints vs. Reward Shaping

Should tram priority be encoded as a **hard constraint** (never violable) or as **reward shaping** (incentivized but not guaranteed)? What does each choice imply for safety and optimality?

2

Centralized vs. Decentralized Control

A single centralized agent coordinates globally but scales poorly. Decentralized agents scale better but face coordination overhead. Which architecture is appropriate — and when?

3

Safety in RL-Based Control

Average performance is insufficient in real urban traffic. How can we guarantee the **absence of dangerous or unstable behaviors** in a learned policy, especially under distribution shift?

4

Hybrid GA + RL Systems

Can GAs generate robust control structures that RL then adapts online? What would the interface between the two components look like? This is the neurosymbolic design question at the heart of the course.

The Neurosymbolic Traffic Controller: A Vision

What would a *true* neurosymbolic traffic control system look like in practice? The diagram below outlines a principled integration architecture — symbolic planning governs structure, learned policies govern adaptation.

Structural Initialization

GA generates structured candidate signal plans.

Online Adaptation

DRL refines plans using real-time observations.

Symbolic Encoding

Formalize rules, constraints, and priorities.

Constraint Verification

Symbolic verifier enforces hard safety checks.

This architecture reflects the core neurosymbolic principle: **symbolic components provide interpretability and safety guarantees**, while **learned components provide flexibility and real-time responsiveness**. Neither alone is sufficient.