2024/2025

REINFORCEMENT LEARNING BASED **ADAPTIVE TRAFFIC SIGNAL CONTROL USING SUMO**

TEAM MEMBERS :

- AMIEUR ZINEB ICHRAKE
- HAMDI AYA
- MEFLAH YOUSRA

SUPERVISOR :

MR. MALKI ABED EL HAMID

1. INTRODUCTION

- Urban traffic congestion is a persistent challenge in modern cities,
- Traditional fixed-time signal control systems lack adaptability to real-time traffic dynamics and often result in inefficient flow.
- Our goal is to build an intelligent, adaptive traffic signal control system using Deep Reinforcement Learning (DRL), capable of making real-time decisions based on traffic state

2. PROBLEM STATEMENT

Modern cities struggle with traffic congestion due to static, pre-defined traffic signal schedules that cannot respond to real-time changes in traffic demand. This rigidity leads to:

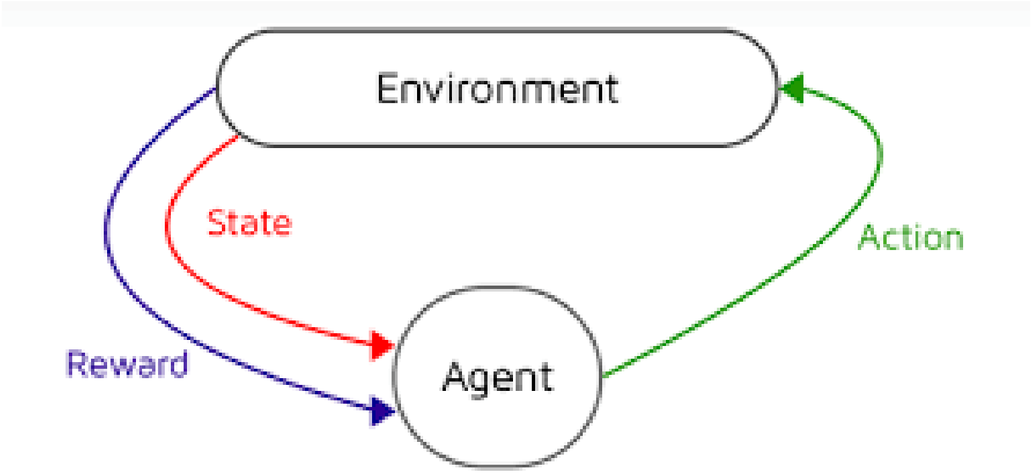
- Long queues at intersections
- Increased waiting times for drivers
- Higher fuel consumption and emissions
- Poor traffic distribution across lanes

The core challenge lies in developing a **control system capable of adapting dynamically to varying traffic patterns** while ensuring safety and fairness across all directions.

We frame this problem as a **Markov Decision Process,** where an RL agent interacts with a simulated intersection and learns to make intelligent, phase-based decisions. The goal is to **maximize vehicle throughput**, **minimize queue lengths**, and improve intersection efficiency through data-driven learning — without relying on hard-coded traffic rules.

3. MODELING TRAFFIC CONTROL AS A MARKOV DECISION PROCESS (MDP)

To effectively train a reinforcement learning agent for traffic light control, we model the intersection as a Markov Decision Process (MDP), where the agent learns optimal decisions through interaction with a simulated environment.



State (S):

The agent observes the current traffic conditions, represented by a state vector.

Action (A):

The action space includes selecting the current active green phase.
In advanced versions, the agent also controls phase duration or throughput, enabling more adaptive strategies.

Reward (R):

The reward function balances traffic efficiency and safety. It penalizes queue length and waiting time while rewarding higher throughput, encouraging the agent to minimize congestion.

Transition Dynamics (T):

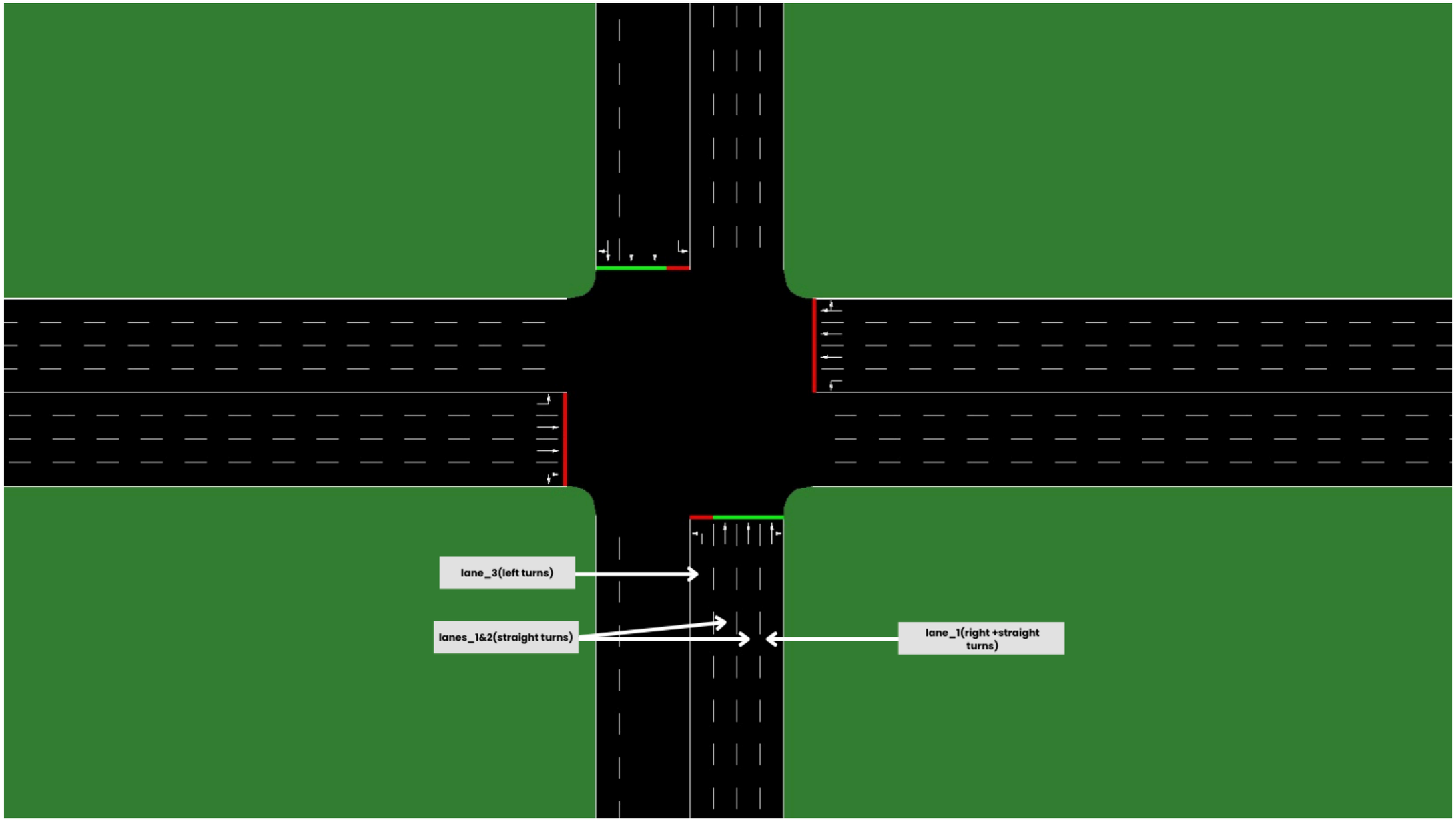
State transitions are governed by the SUMO traffic simulator, which provides realistic feedback based on the current signal phase and vehicle behavior.

4.1. THE INTERSECTION SETUP

Our simulation environment models a realistic four-way urban intersection, with two main roads crossing in the north-south and east-west directions. Each road includes four lanes per approach, resulting in a total of 16 incoming lanes and a rich variety of maneuvers.

Each approach (North, South, East, West) has lanes labeled from 0 to 3, each dedicated to specific movement types:

4.1.THE INTERSECTION SETUP

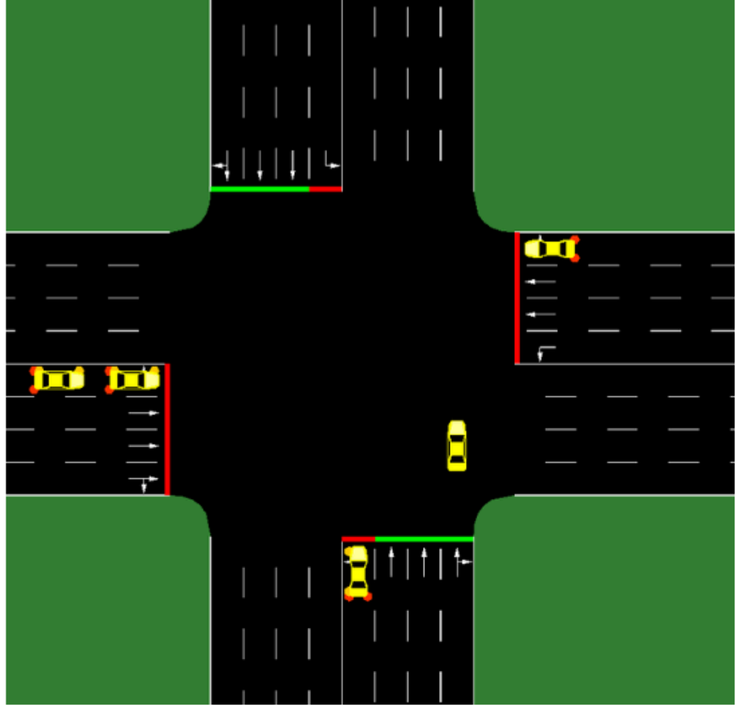


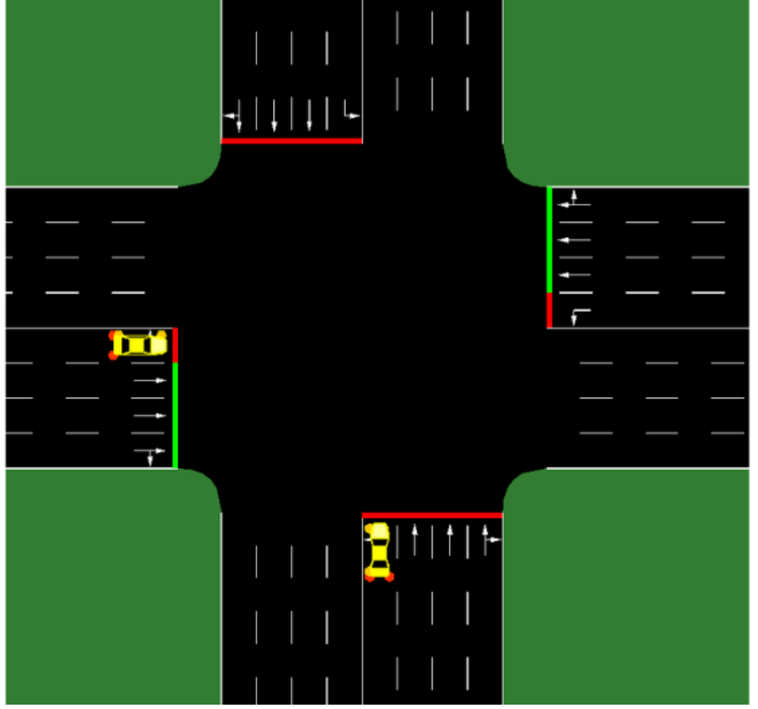
4.2.TRAFFIC LIGHT LOGIC DESCRIPTION

The traffic light system at the simulated intersection follows a well-defined eight-phase control logic, alternating between green and yellow phases. This setup ensures both traffic safety and efficient vehicle movement, especially addressing the complex dynamics of left-turn maneuvers.

GREEN PHASES – DETAILED OVERVIEW

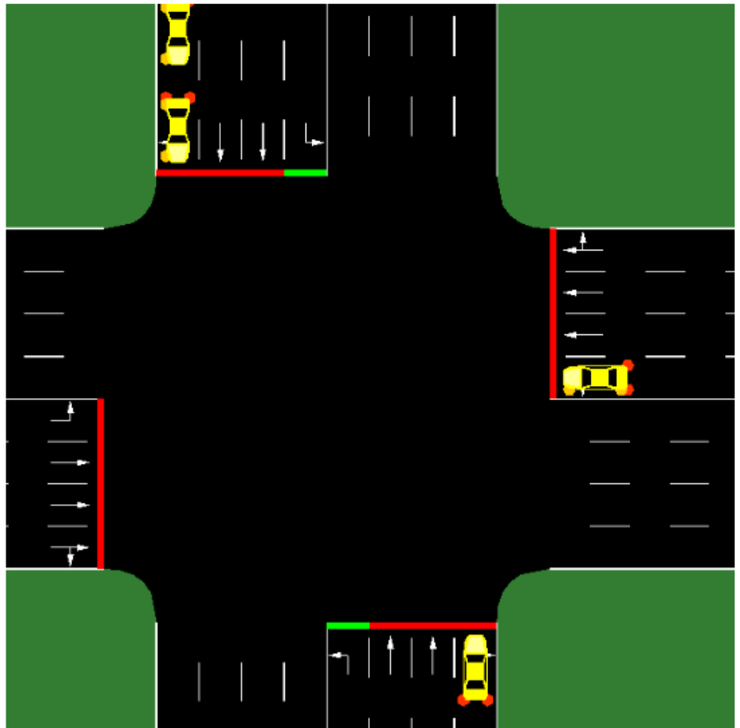
North-South
straight+right

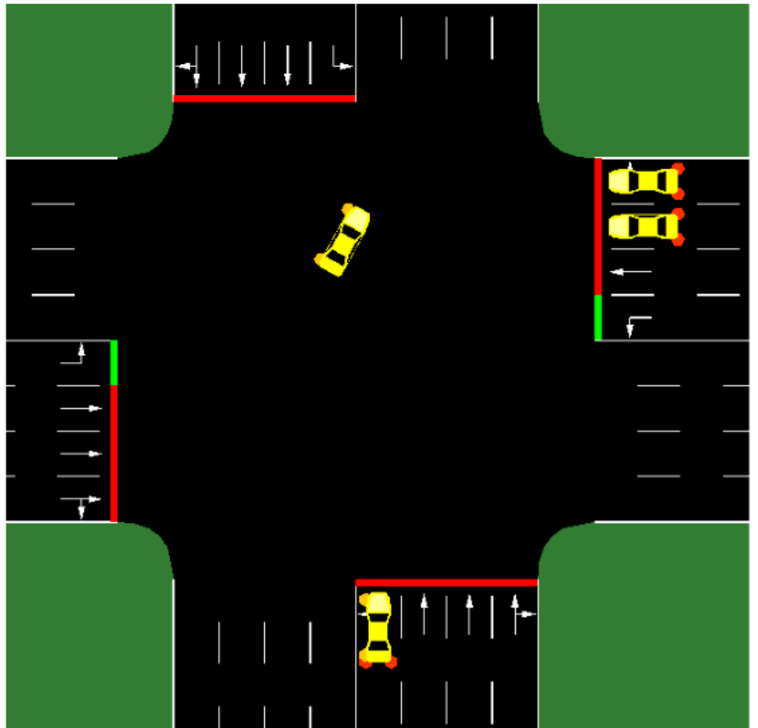




East-West
straight+right

North-South
left only





East-West
left only

4.2.TRAFFIC LIGHT LOGIC DESCRIPTION

YELLOW PHASES – TRANSITION SAFETY

Each green phase is followed by a yellow phase of fixed duration (5 seconds), corresponding to the previous movement phase. For example:

Phase 1 follows Phase 0 (North–South Straight/Right),
Phase 3 follows Phase 2 (North–South Left Turns),and so on

Yellow phases are crucial for:

- Giving vehicles time to exit the intersection safely.
- Preventing sudden stops that could lead to rear-end collisions.
- Maintaining a smooth flow between signal changes.

WHY USE SEPARATE LEFT-TURN PHASES?

Left turns are inherently complex due to their cross-path movement with oncoming vehicles. Allowing them during general green phases introduces multiple conflict points. To mitigate this:

- Dedicated green phases are assigned for left-turns (Phases 2 and 6).
- During these phases, straight and right-turn traffic from the same direction is held.
- This eliminates cross-traffic conflicts, improving both safety and predictability.

4.3.TRAFFIC GENERATION MECHANISM

To emulate real-world traffic flow and progressively challenge the RL agent, the environment employs a dynamic, curriculum-based traffic generation system. This mechanism ensures gradual complexity during training, supporting robust and adaptable learning.

- **Traffic Profiles: Base vs. Target**

Traffic generation is governed by three traffic profiles :
Low, Medium, and High,each defined by the following parameters:

- Probability (prob): Likelihood that a specific traffic profile is selected.
- Spawn Rate (spawn): Probability of a vehicle being spawned when the profile is active.
- Maximum Lanes (max_lanes): Number of lanes used for spawning vehicles on each edge.

4.3.TRAFFIC GENERATION MECHANISM

Each profile is defined in two versions:

- Base Traffic: Simulates light, easy traffic scenarios (used at the start of training).
- Target Traffic: Represents complex, high-density conditions (used toward the end).

Profile	Base Prob.	Target Prob.	Base Spawn	Target Spawn	Base Max Lanes	Target Max Lar
Low	0.8	0.1	0.1	0.3	1	1
Medium	0.2	0.3	0.3	0.6	2	3
High	0.0	0.6	0.5	0.9	3	4

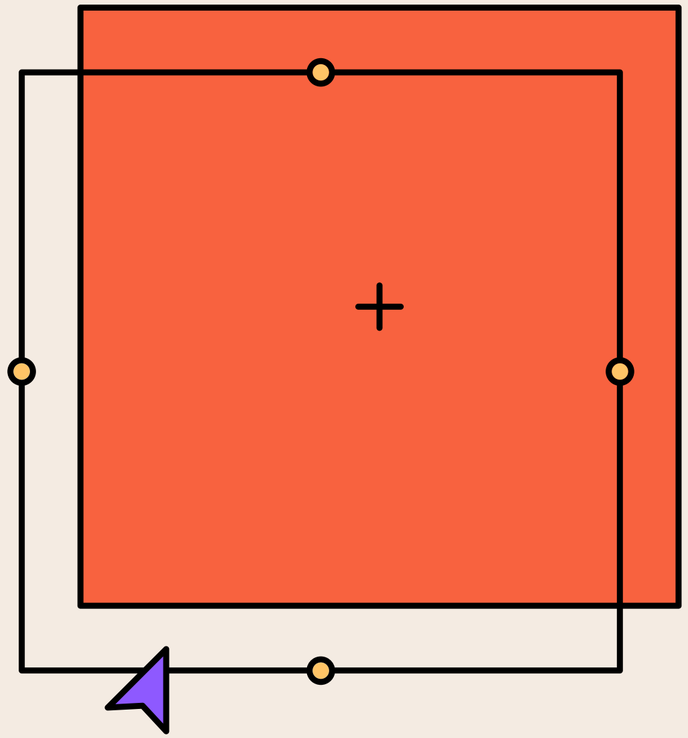
4.3.TRAFFIC GENERATION MECHANISM

- **Curriculum Learning: Dynamic Adjustment**

  To ensure a smooth and progressive increase in difficulty, the environment uses linear interpolation between the base and target traffic settings:

$$\text{current\_value} = \text{base\_value} + (\text{target\_value} - \text{base\_value}) \times \text{difficulty}$$

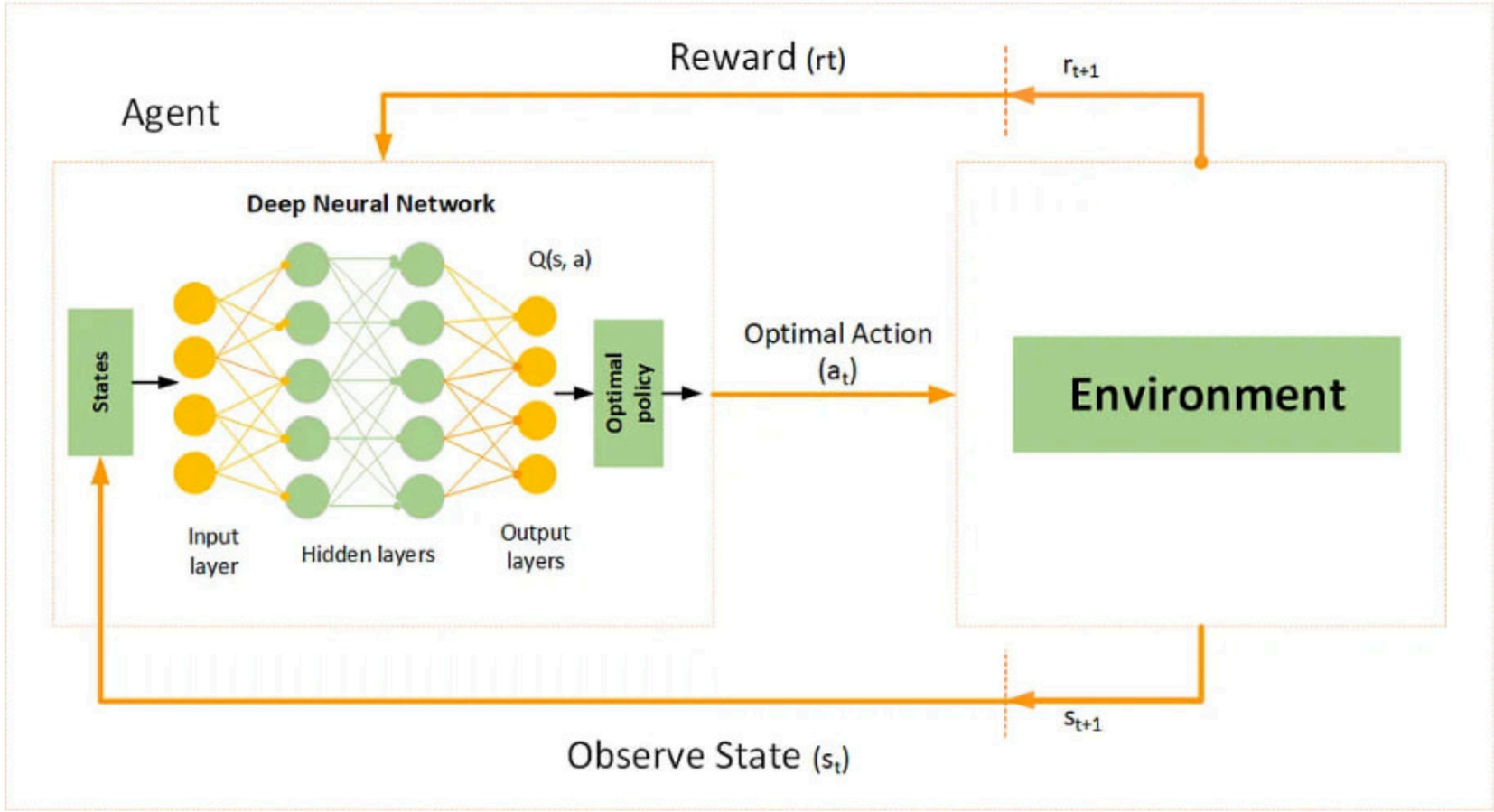
  - The difficulty parameter gradually increases during training.
  - This allows traffic conditions to evolve from easy to challenging in a controlled and learnable manner.



METHODOLOGY

1. VALUE BASED ALGORITHMS

DEEP Q-LEARNING



Structure of DQN

APPROACH 1

FIXED PHASE DURATIONS + 16-DIMENSIONAL STATE

APPROACH 2

DYNAMIC PHASE DURATIONS + CONTROLLED LANES AS STATE

APPROACH 3

THROUGHPUT-BASED CONTROL+ CONTROLLED LANES AS STATE

APPROACH 1 : ENVIRONMENT & MDP DESIGN

State Representation (16 Dimensions)

-The state is a vector capturing traffic + signal info from incoming edges (E2TL, N2TL, S2TL, W2TL).

-Components:

- Traffic Light Phase (4 dims): One-hot encoded (e.g., [1, 0, 0, 0] = NS green).
- Vehicle Count (4 dims): Normalized by 200. Shows congestion.
- Waiting Time (4 dims): Normalized by 3600s. Reflects delays.
- Halting Vehicle Count (4 dims): Normalized by 200. Represents immediate queue size.

-Normalization improves learning stability and consistency across episodes.

APPROACH 1 : ENVIRONMENT & MDP DESIGN

Action Space

- 4 discrete actions → 4 traffic signal phases (e.g., NS-straight, EW-left, etc.).
- Yellow Phase (5s) is inserted when switching between green phases to simulate safety.
- After yellow, the selected green phase becomes active.

APPROACH 1 : ENVIRONMENT & MDP DESIGN

Reward Function Design

The reward is calculated using changes (deltas) in traffic metrics:

✅ Δ Waiting Time (Weight: 1.5): Reducing wait time improves user experience.

✅ Δ Queue Length (Weight: 1.0): Helps manage congestion.

✅ Throughput (Weight: 0.8): Encourages smooth vehicle flow.

Why use deltas?

- Promotes improvement: Rewards the agent only when it reduces delay or congestion, not for maintaining the status.
- Normalizes feedback across traffic conditions.
- Encourages action: Forces the agent to do better than before, not just survive.

VERSION 1 : DQN

Q-Network Architecture:

Input: 16 units (state: phase + traffic metrics).
Hidden: 2 layers (64 neurons each, ReLU activation).
Output: 4 units (Q-values for 4 green phases).

$$y = \begin{cases} r, & \text{if } s' \text{ is terminal} \\ r + \gamma \max_{a'} Q_{\text{target}}(s', a'; \theta^-), & \text{otherwise} \end{cases}$$

The loss function is defined as:

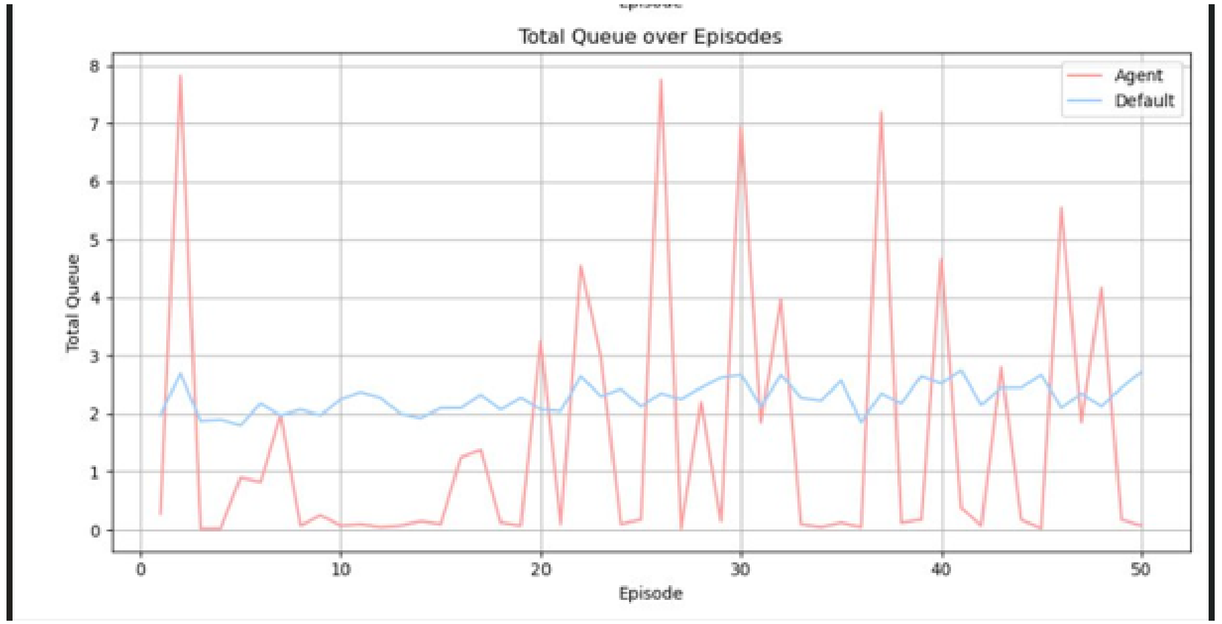
$$\mathcal{L}(\theta) = E_{(s,a,r,s') \sim D} \left[(y - Q_{\text{online}}(s, a; \theta))^2 \right]$$

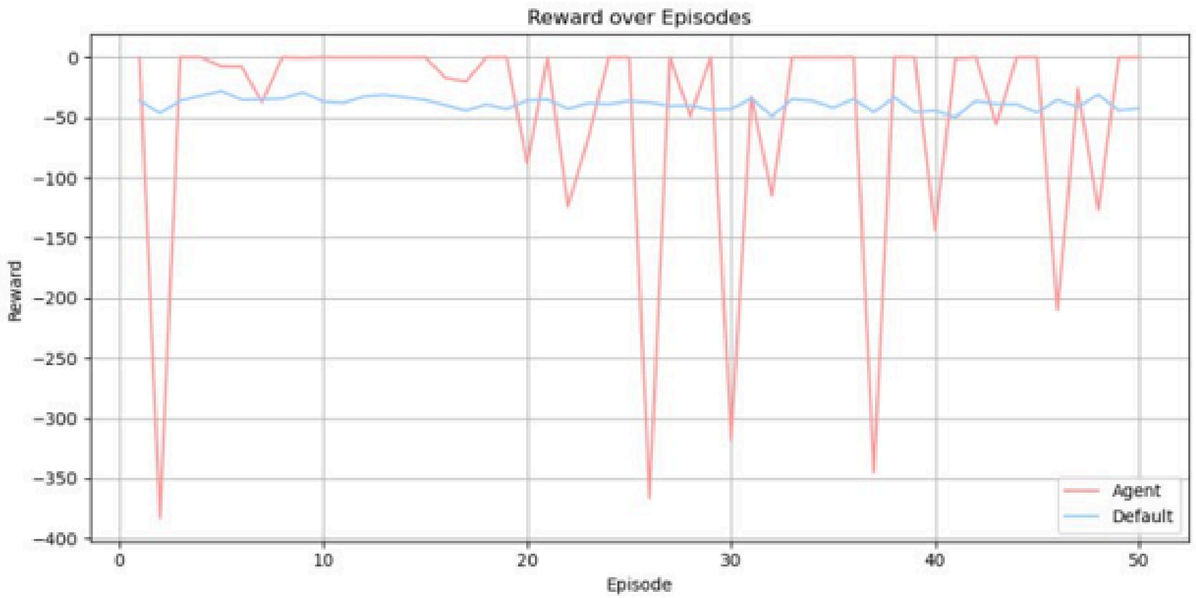
Soft Update:

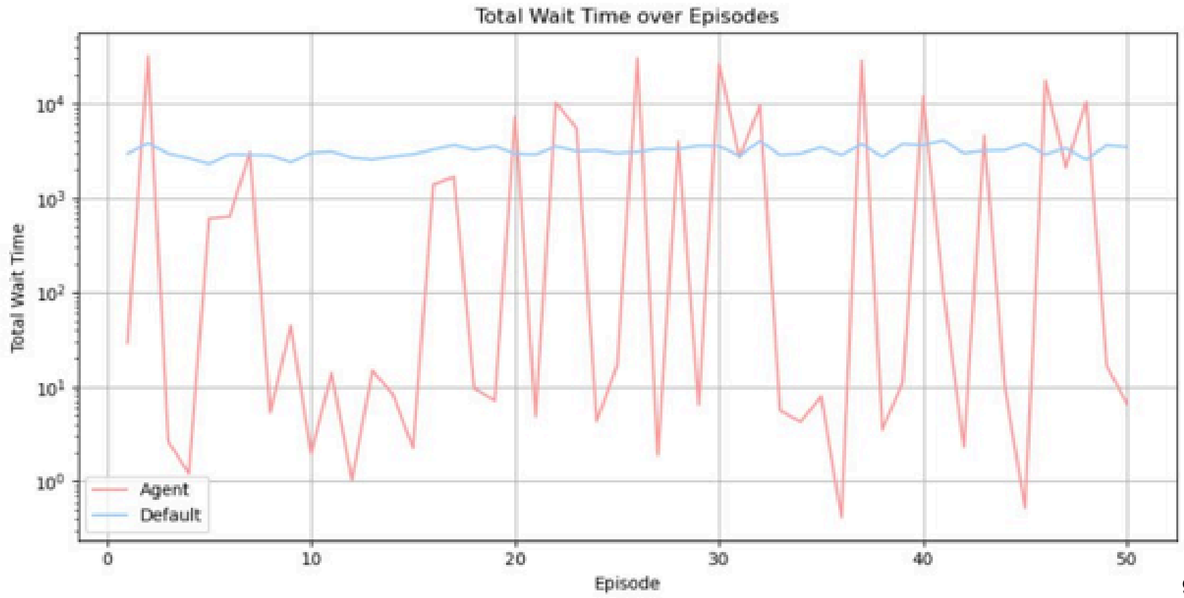
$$\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^-$$

VERSION 1 : DQN

RESULTS:







VERSION 2 : DUELING + DOUBLE DQN

Q-Network Architecture:

Q-Network: This version adopts a dueling architecture, consisting of:

- Feature Layer: Maps 16 input features to 64 neurons with ReLU activation.
- Value Stream: A fully connected layer from 64 neurons to 1 value output, estimating the state value.
- Advantage Stream: A fully connected layer from 64 neurons to 4 advantage outputs, one for each possible action.

Double DQN :

$$a^* = \arg\max_{a'} Q_{\text{online}}(s', a'; \theta)$$

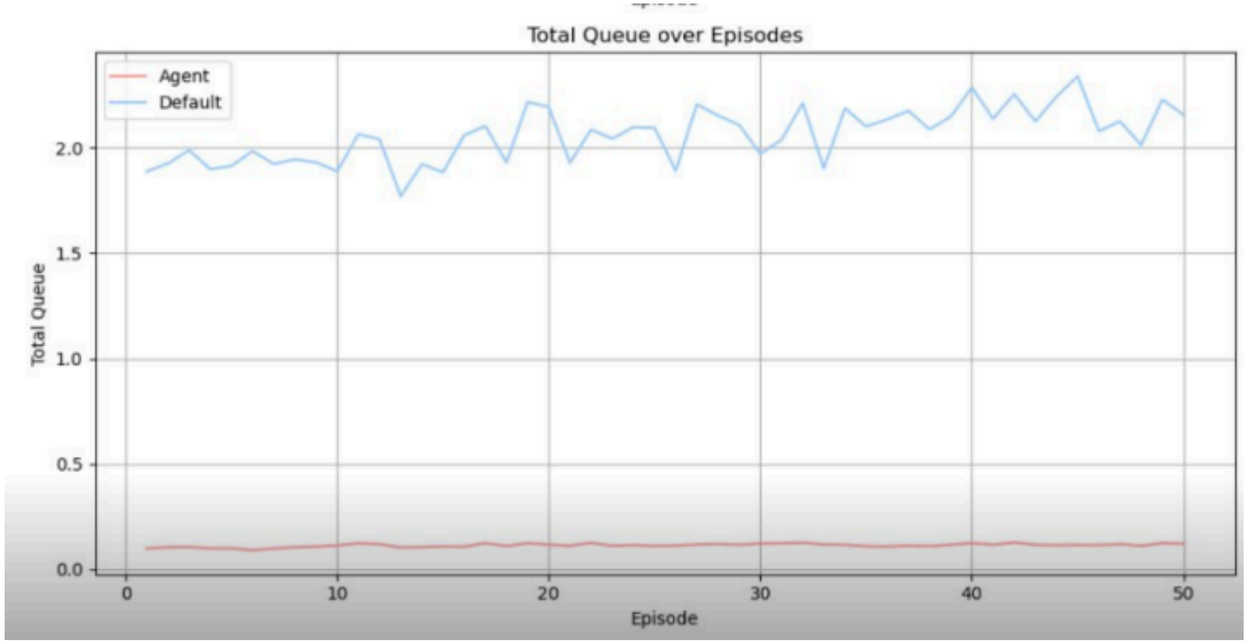
$$y = r + \gamma Q_{\text{target}}(s', a^*; \theta^-)$$

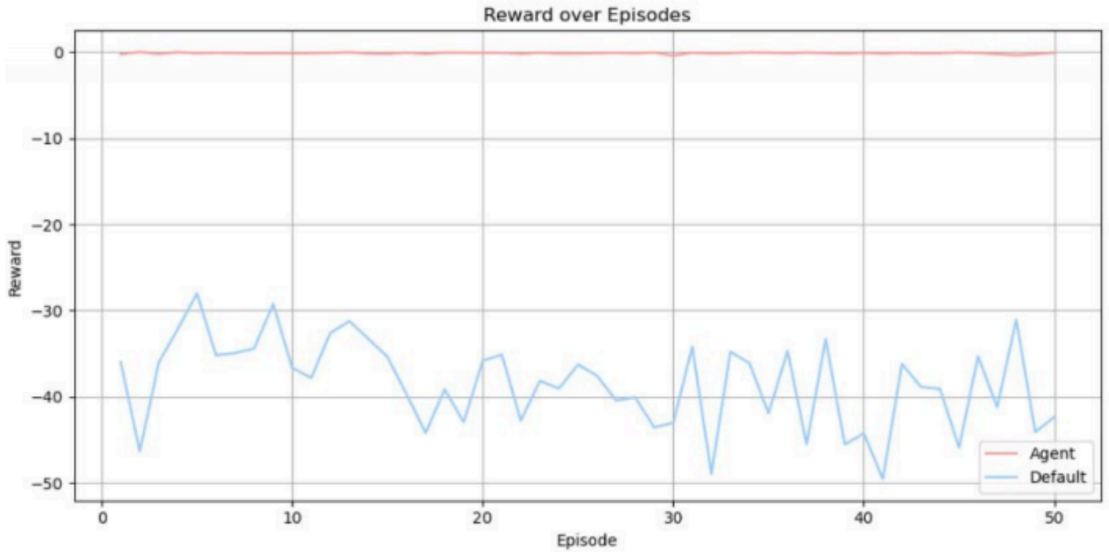
Dueling DQN :

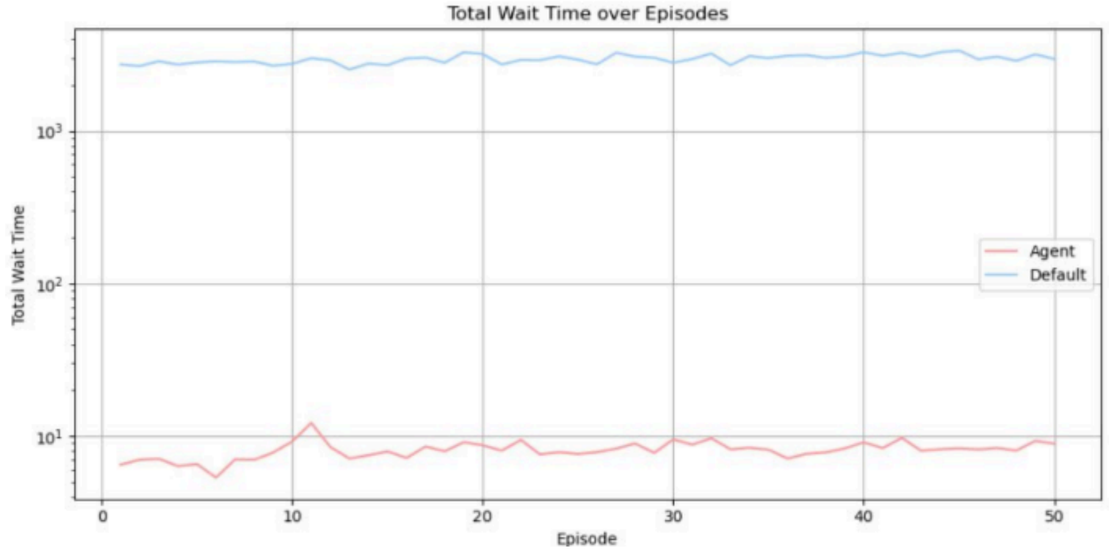
$$Q(s,a) = V(s) + A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a')$$

VERSION 2 : DUELING + DOUBLE DQN

RESULTS:







VERSION 3 : DOUBLE DQN WITH DUELING AND PRIORITIZED ARCHITECTURE

Q-Network Architecture:

Dueling DQN Architecture:

Feature Layer: 16 (input) → 128 (ReLU, layer normalization, 20% dropout) → 128 (ReLU, layer normalization).

Value Stream: 128 → 64 (ReLU) → 1 .

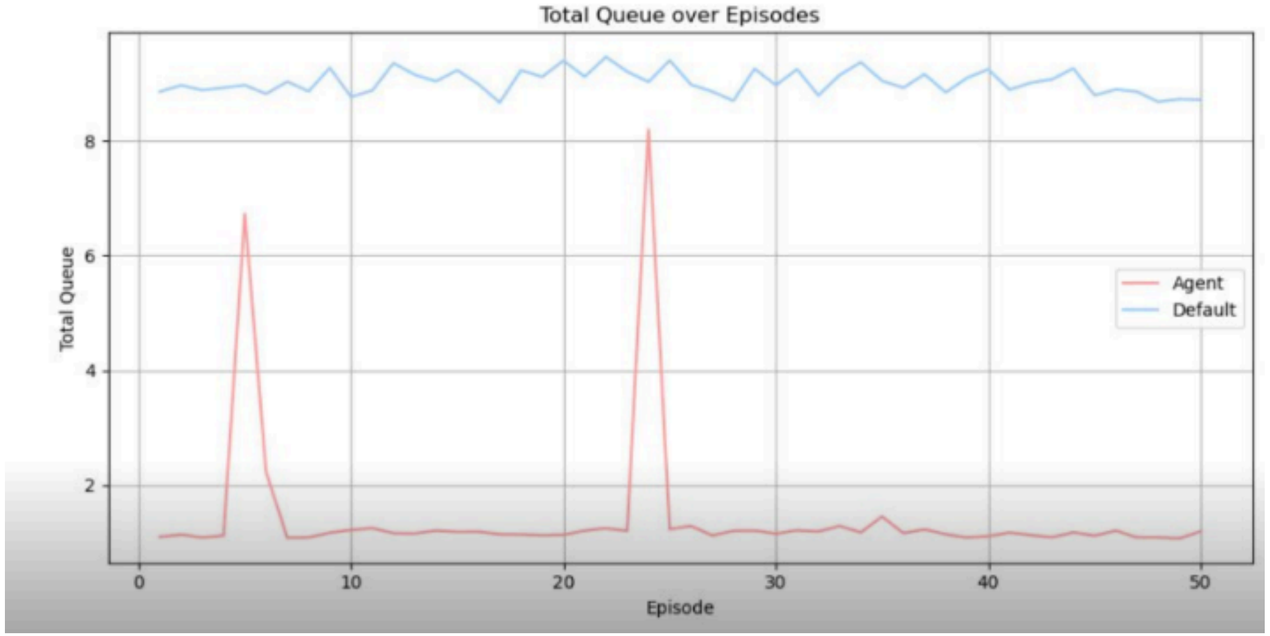
Advantage Stream: 128 → 64 (ReLU) → 4

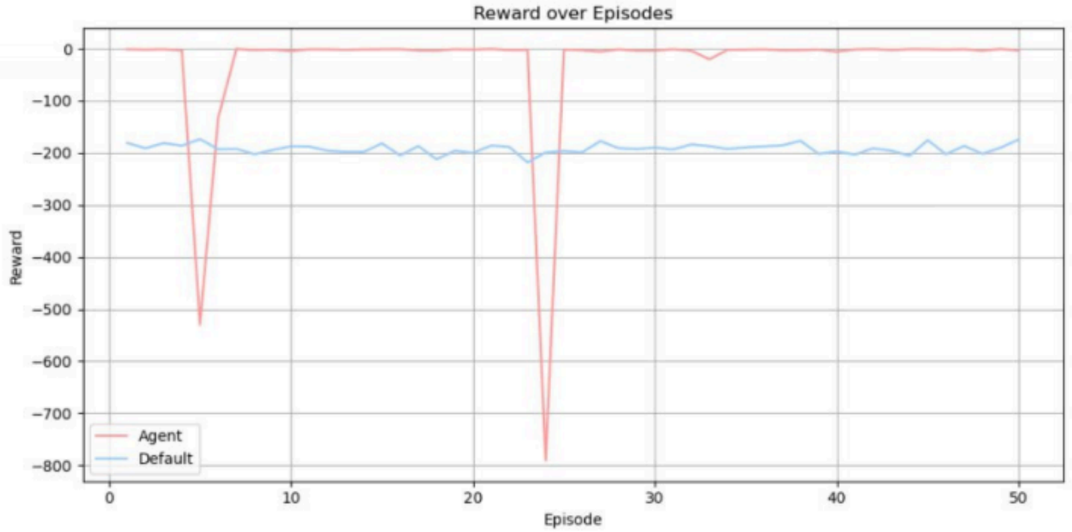
Prioritized Experience Replay (PER)

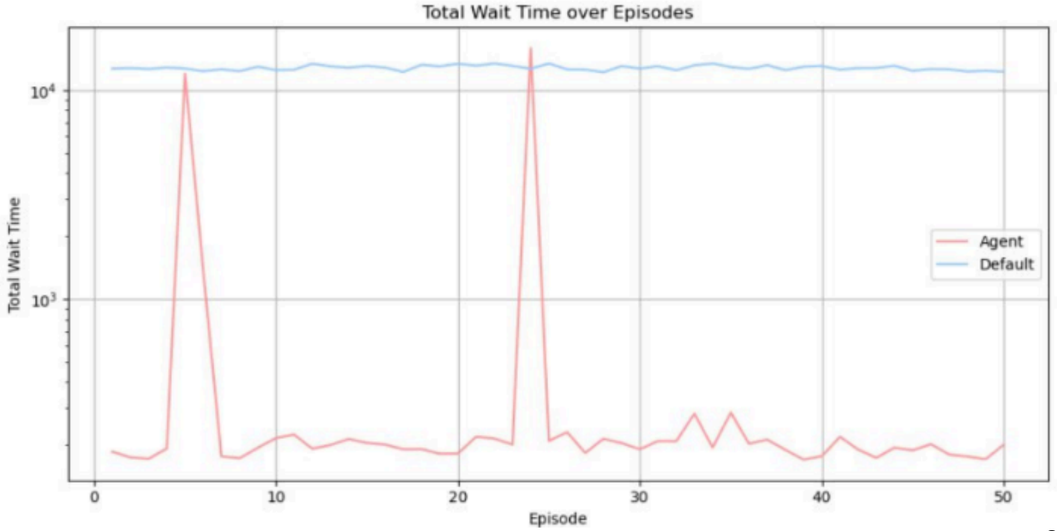
PER improves standard experience replay by sampling transitions based on their TD error.

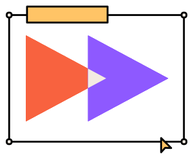
VERSION 3 : DOUBLE DQN WITH DUELING AND PRIORITIZED ARCHITECTURE

RESULTS:









VERSION 4 : DYNAMIC DURATION

Overview:

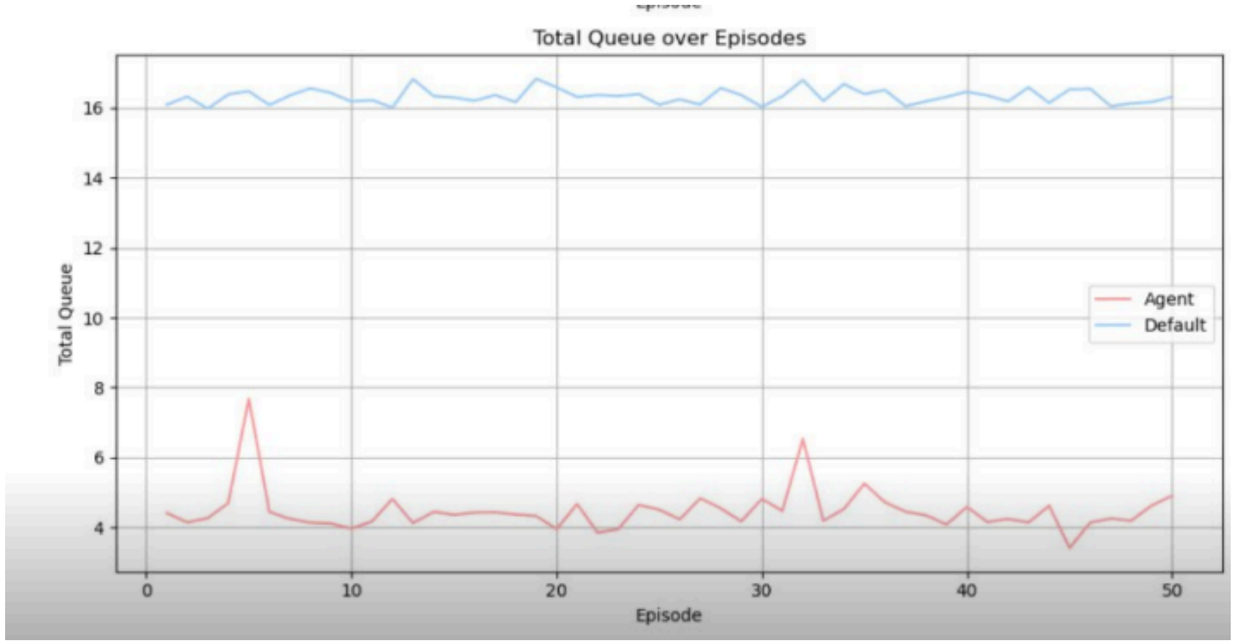
Introduces dynamic action space: selects green phase and duration (20, 30, or 50s).

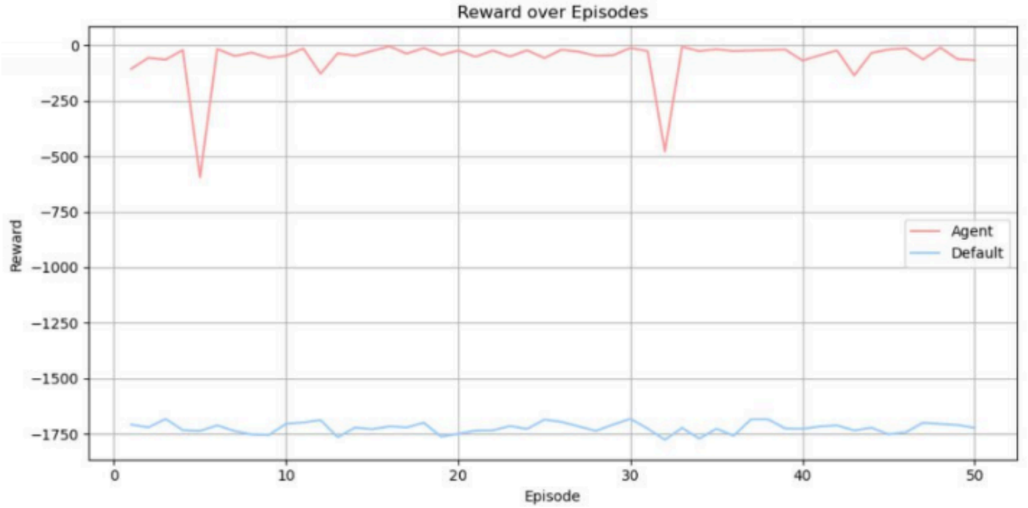
Replaces fixed 30s durations from Versions 1-3.

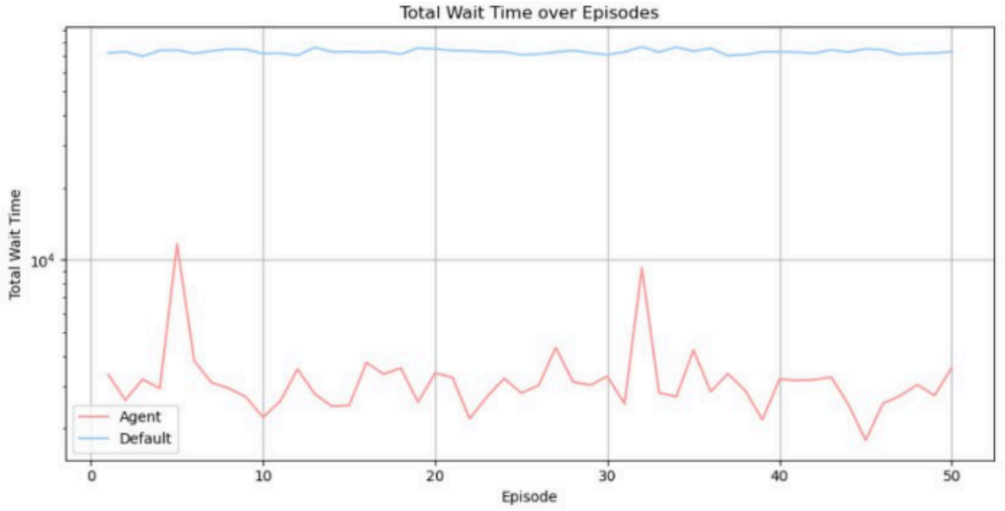
Action size increased from 4 to 12 (4 phases × 3 durations).

VERSION 4 : DYNAMIC DURATION

RESULTS:







APPROACH 2 : ENVIRONMENT & MDP DESIGN

State Representation: Controlled Lanes

State is now based on lanes affected by the current green phase, not full incoming edges.

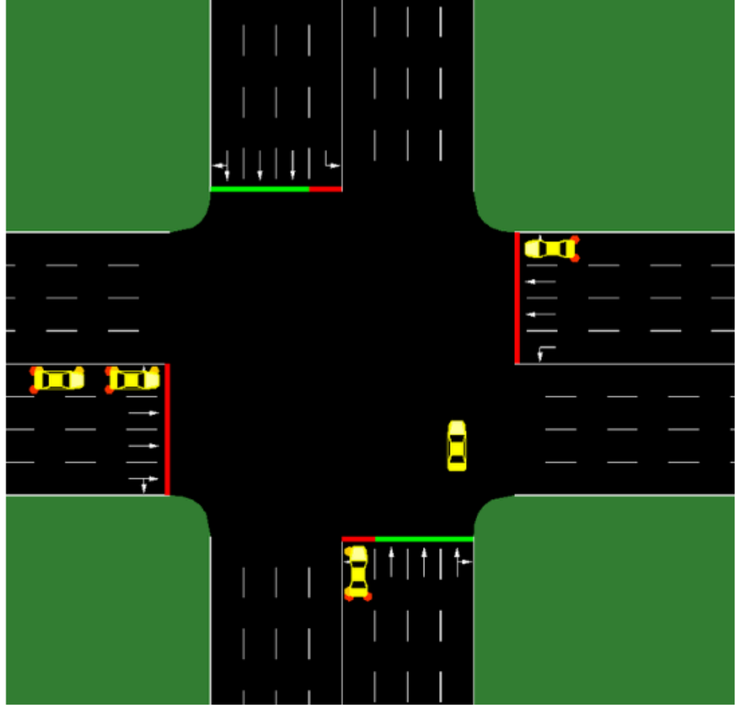
Extracted features per controlled lane:

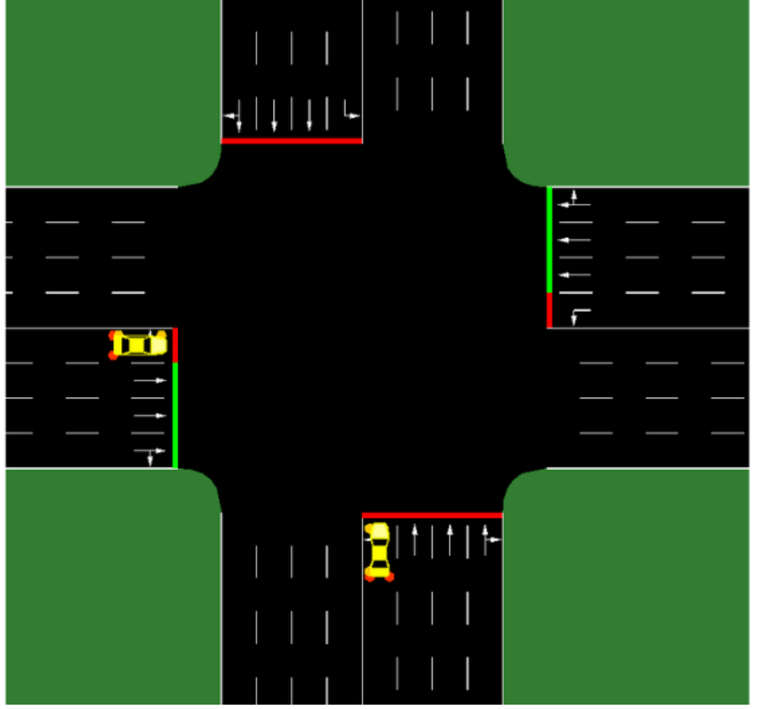
- Average Waiting Time
- Vehicle Count
- queue lenght

✅ Why this change?

- Focuses only on relevant lanes → less noise, better learning.
- Anticipates which phases need activation.
- Prevents phase starvation and improves fairness.

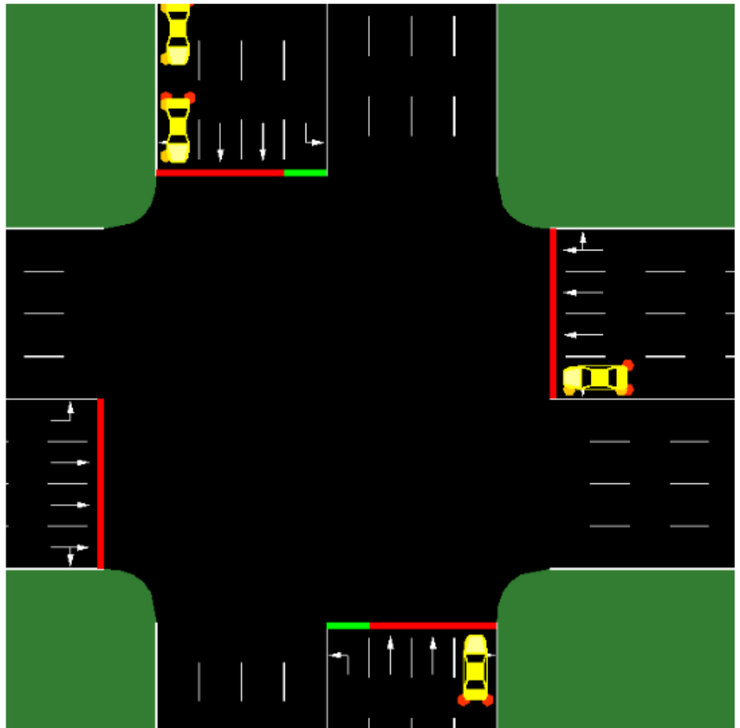
North-South
straight+right

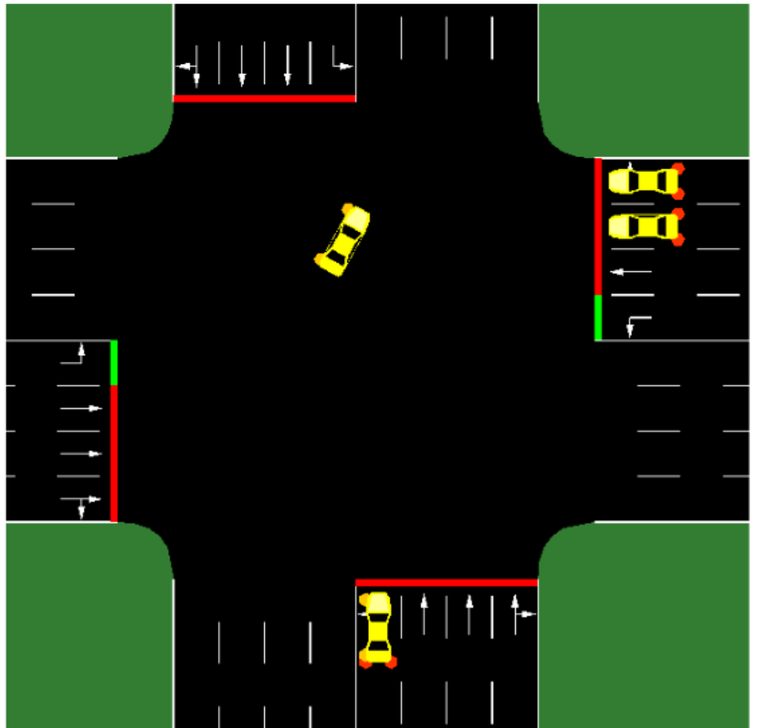




East-West
straight+right

North-South
left only





East-West
left only

APPROACH 2 : ENVIRONMENT & MDP DESIGN

Action Space: Phase + Duration :

Agent selects:

- Which phase to activate (e.g., NS straight, EW left, etc.)
- How long to keep it green (e.g., 10–30s)

✅ Why it matters:

- Increases green time when traffic is high.
- Cuts delay when traffic is low.
- Leads to adaptive, traffic-aware control.

APPROACH 2 : ENVIRONMENT & MDP DESIGN

Reward Function: Balanced by Lane Activity

$$R = w_a \cdot f(\text{active\_lanes}) + w_i \cdot f(\text{inactive\_lanes})$$

This new reward function approach :

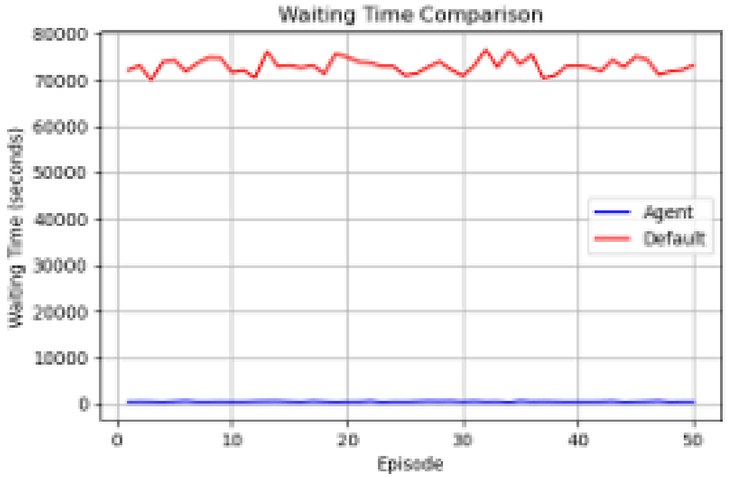
- Prioritizes current traffic flow.
- Still monitors inactive lanes to prevent future congestion.

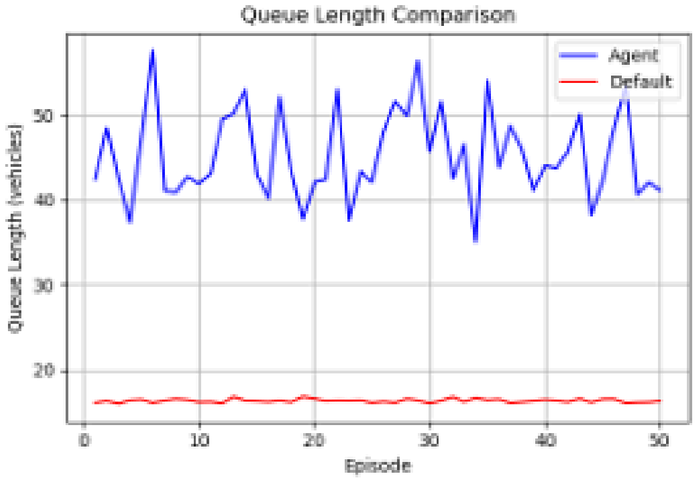
CONTROLLED LANES AS STATE WITH DYNAMIC PHASE DURATIONS

Approach 2 builds on Version 4 by retaining the same Dueling DQN architecture and techniques—Double DQN, Dueling DQN, Prioritized Experience Replay , and soft updates—while introducing a new state representation based on controlled lanes affected by the current traffic light phase.

RESULTS:


Agent vs Default Reward
Total Reward
Episode
Agent
Default
6000
4000
2000
0
−2000
0
10
20
30
40
50


Waiting Time Comparison
Waiting Time (seconds)
Episode
Agent
Default
80000
70000
60000
50000
40000
30000
20000
10000
0
0
10
20
30
40
50


Queue Length Comparison
Queue Length (vehicles)
Episode
Agent
Default
50
40
30
20
0
10
20
30
40
50

APPROACH 3 : ENVIRONMENT & MDP DESIGN

Key Objective :

- Agent no longer chooses a fixed duration.
- Instead, it:

1. Selects a phase.
2. Keeps it green until enough vehicles pass through or a timeout occurs.

APPROACH 3 : ENVIRONMENT & MDP DESIGN

Key Objective :

- Agent no longer chooses a fixed duration.
- Instead, it:

1. Selects a phase.
2. Keeps it green until enough vehicles pass through or a timeout occurs.

APPROACH 3 : ENVIRONMENT & MDP DESIGN

Step Function & Phase Control Logic

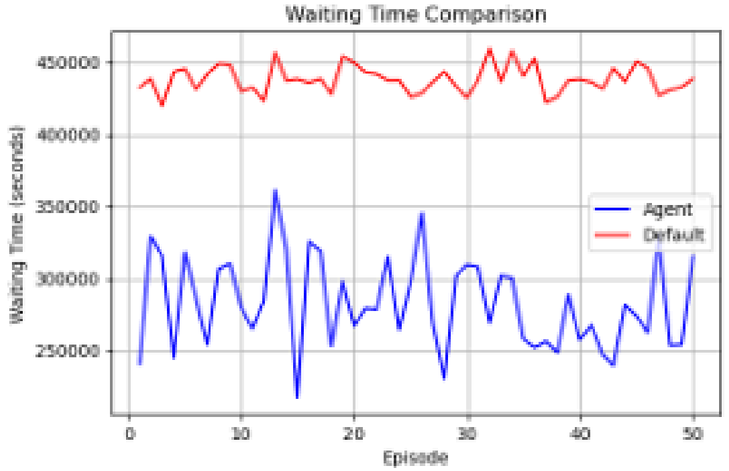
- At the start of green phase:
  - Reset throughput counter.
- During simulation:
  - Track how many vehicles exit the junction.
- Agent switches phase when:
  - Throughput threshold is met (e.g., 10 vehicles), OR
  - Timeout is reached (e.g., 50s) to prevent starvation.

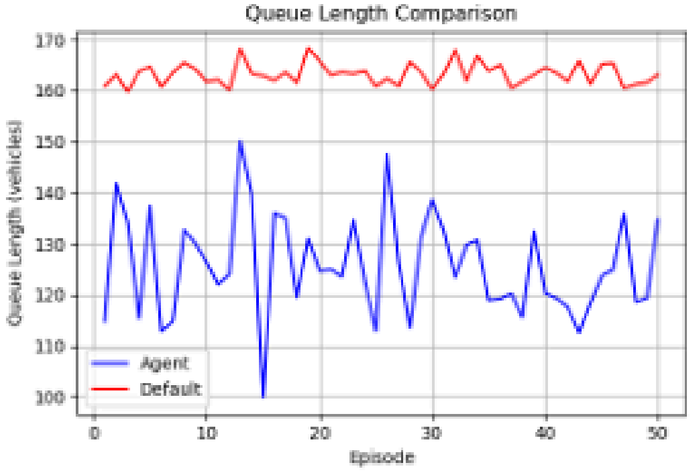
THROUGHPUT-BASED PHASE CONTROL

Approach 3 introduces a throughput-based phase control strategy, maintaining the same Dueling DQN architecture and techniques as Approach 2—Double DQN, Dueling DQN, Prioritized Experience Replay , and soft updates , while adopting a new state representation that includes queue length, waiting time, current phase index, and time elapsed in the current green phase per controlled lane. It replaces fixed durations with a percentage-based action space (12 actions: 4 phases × 3 throughput thresholds of 20%, 50%, 80%), allowing the agent to dynamically adjust green phases in the SUMO simulator based on real-time vehicle flow until the target throughput or a 90-second cap is reached, enhancing intersection efficiency and fairness.

RESULTS:


Agent vs Default Reward
Total Reward
Episode
Agent
Default
6000
4000
2000
0
−2000
0
10
20
30
40
50


Waiting Time Comparison
Waiting Time (seconds)
Episode
Agent
Default
450000
400000
350000
300000
250000
0
10
20
30
40
50

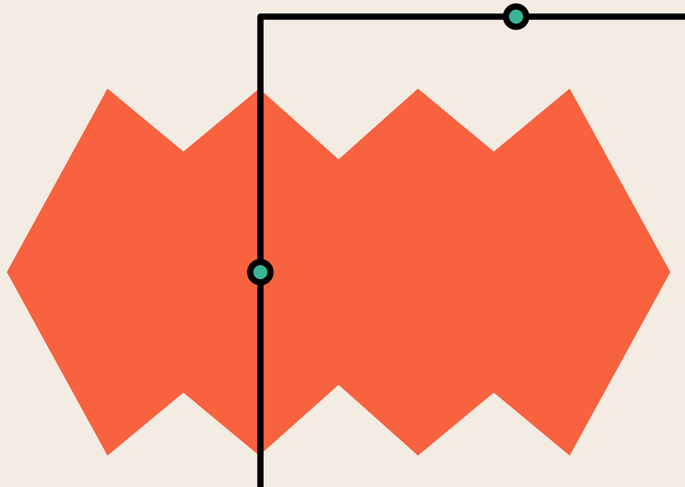

Queue Length Comparison
Queue Length (vehicles)
Episode
Agent
Default
170
160
150
140
130
120
110
100
0
10
20
30
40
50

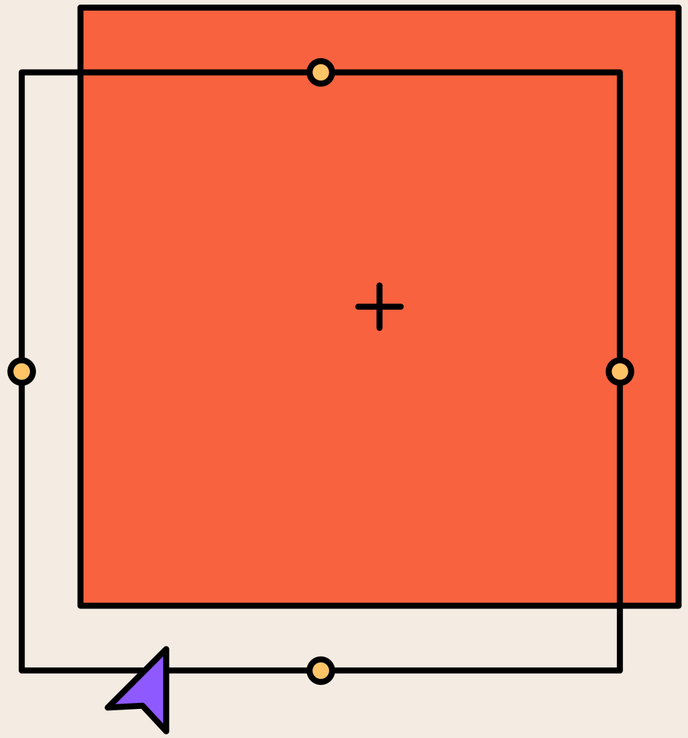
CONCLUSION

In conclusion, Version 4, with its dynamic duration action space (4 phases × 3 durations: 20, 30, 50s), emerged as the most stable and effective approach, consistently outperforming the default controller in the SUMO simulator by significantly reducing wait times and queue lengths . While the throughput-based approach (Approach 3) demonstrated strong potential with high rewards , its performance was less stable and required more training episodes to achieve optimal results, making Version 4 the preferred choice for reliable and adaptive traffic signal contro

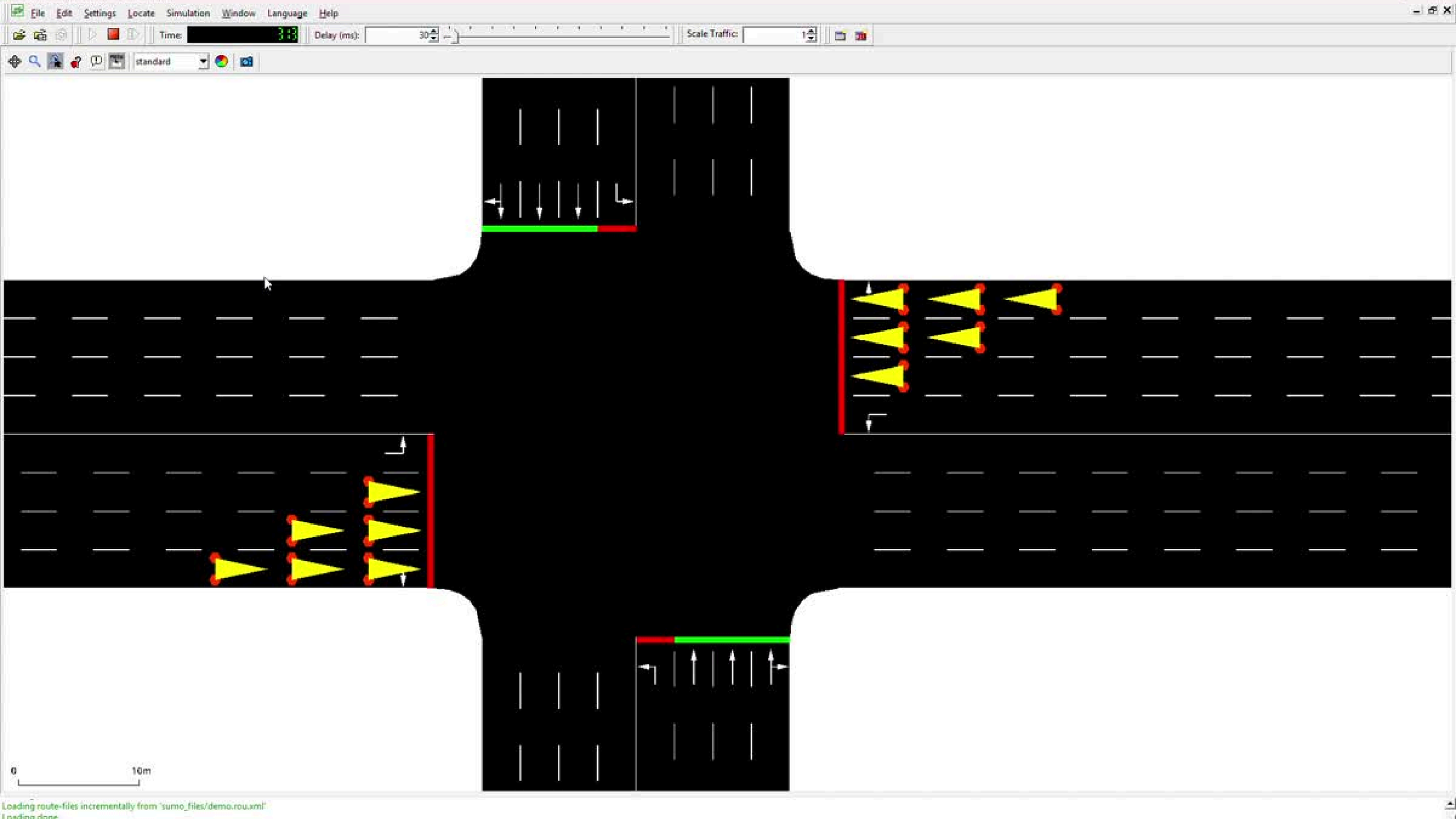
Challenges

- Complexity of the Action Space
- Reward Function Design
- Computational Resource Constraints
- Integration with SUMO Simulator

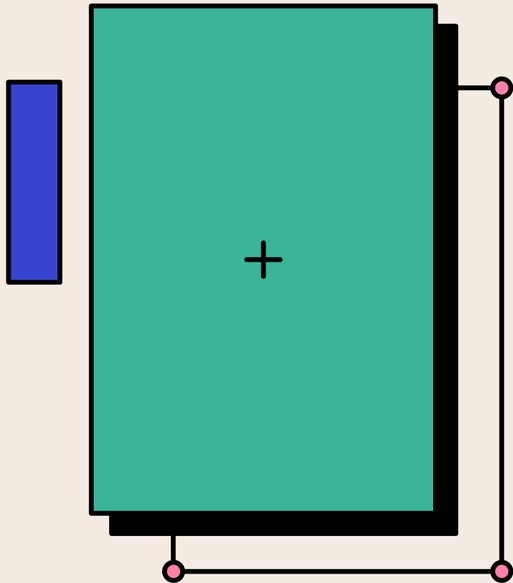




DEMOSTRATION


File Edit Settings Locate Simulation Window Language Help
Time:
Delay (ms):
30
Scale Traffic:
1
standard
0
10m
Loading route-files incrementally from 'sumo_files/demo.rou.xml'
Loading done

THANK YOU



APPROACH 1 : ENVIRONMENT & MDP DESIGN

Step Function and Transition Logic

Each reinforcement learning step involves:

- Phase Transition: Yellow phase (5s) applied if switching green phases.
- Green Phase Execution: New green phase is activated.
- Simulation: SUMO runs for 30 steps (5 seconds per step).
- Vehicle Spawning: Based on traffic generator and difficulty.
- Observation + Reward: New state vector is extracted, and reward is computed using deltas.

APPROACH 1 : ENVIRONMENT & MDP DESIGN

Reset Function and Episode Control

At the beginning of each episode:

- SUMO simulation is closed and restarted.
- Episode counter is incremented.
- All internal variables (counters, queues, timers) are reset.
- Difficulty is updated using curriculum learning strategy.