RESEARCH PESENTATION

Investigating the optimisation of traffic flow using reinforcement learning

Name: Lebajoa
Branch: Industrial Engineering
DATE: 12/November/2024

- Rapid urbanization and increasing vehicle ownership are putting pressure on existing transportation systems.
- Inefficient traffic management leads to gridlocks, economic losses, and environmental degradation.
- Traditional traffic control systems (e.g., pre-timed or actuated signals) struggle to adapt to fluctuating traffic conditions, especially during peak hours or disruptions.
- Recent studies have focused on overcoming these limitations by using machine learning techniques, especially reinforcement learning (RL), to optimise traffic signals

Lebajoa Mphaloane
Presenter

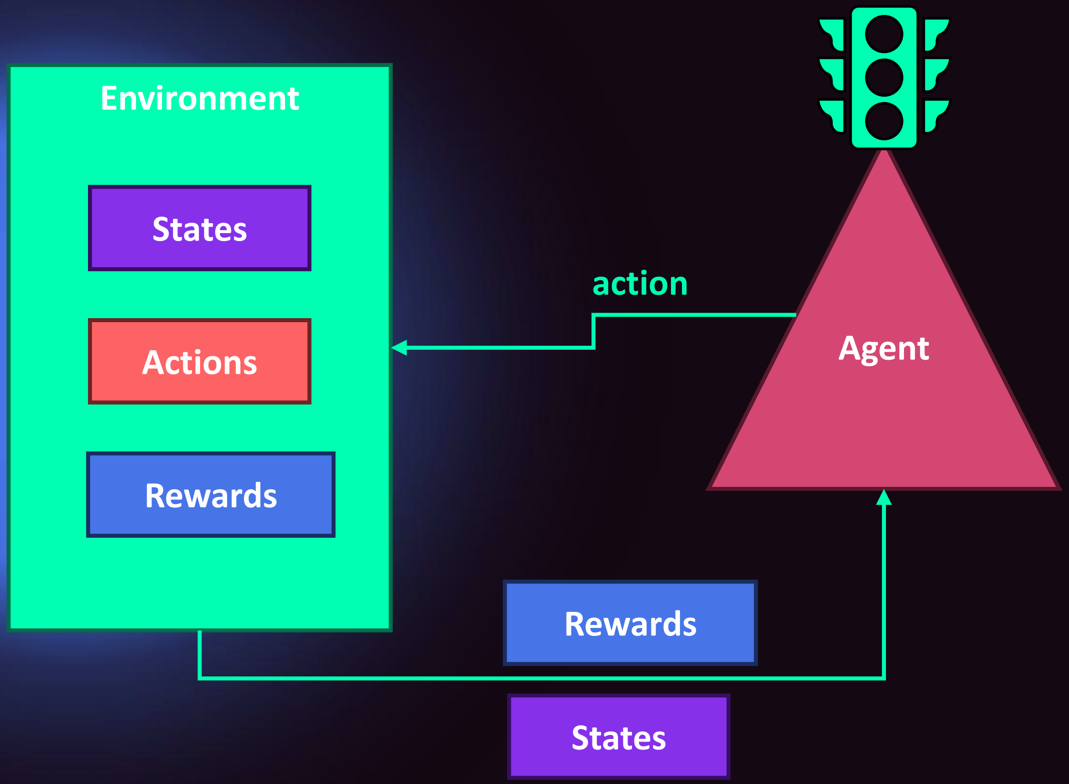
Reinforcement learning

- An AI agent learns by interacting with an environment
- Like a child learning through trial and error
- Core elements:
  - Agent: The learner/decision maker
  - Environment: The world the agent interacts with
  - State: Current situation
  - Action: What the agent can do
  - Reward: Feedback signal (+/-) for each action

•Key principle: Agent learns to maximize long-term rewards

•Real-world examples:

  - Traffic light system learning to control traffic efficiently



Lebajoa Mphaloane
Presenter

Purpose

Problem

Pre-made RL environments often fail to capture real-world traffic complexities, such as fluctuating patterns and disruptions. This research aims to create a custom RL environment to simulate these dynamics, enabling effective agent training for dynamic traffic control.

CRQ

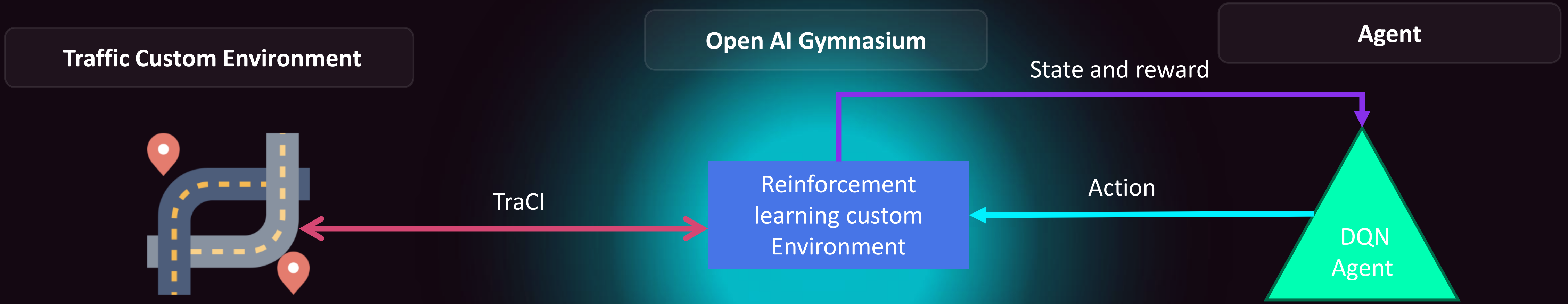
How can a custom reinforcement learning (RL) environment simulate real-world traffic flow, and how does a Deep Q-Network (DQN) agent optimize traffic signal control to reduce congestion and improve efficiency?

Objectives

1. Develop a SUMO-based simulation for realistic urban traffic flow.
2. Design a reward function with key metrics (waiting time, queue length, speed, throughput).
3. Train and evaluate a DQN agent for traffic signal optimization.
4. Test performance on real-world networks (e.g., Braamfontein, Johannesburg CBD).
5. Analyse agent performance and identify DQN limitations.

Lebajoa Mphaloane
Presenter

METHODOLOGY



SUMO
Version: 1.21
• Traffic simulation platform • Road network creation •

Python
Version: 12.0.0 64-bit
• Primary programming language • Environment development • Data processing • Algorithm implementation • Script automation

Stable-Baselines3
Version: 2.3.2
• DQN implementation • Agent training • Policy management • Learning optimization

Visual Studio Code

Version: 1.95

• Code development • Debugging • Version control • Project management

Matplotlib

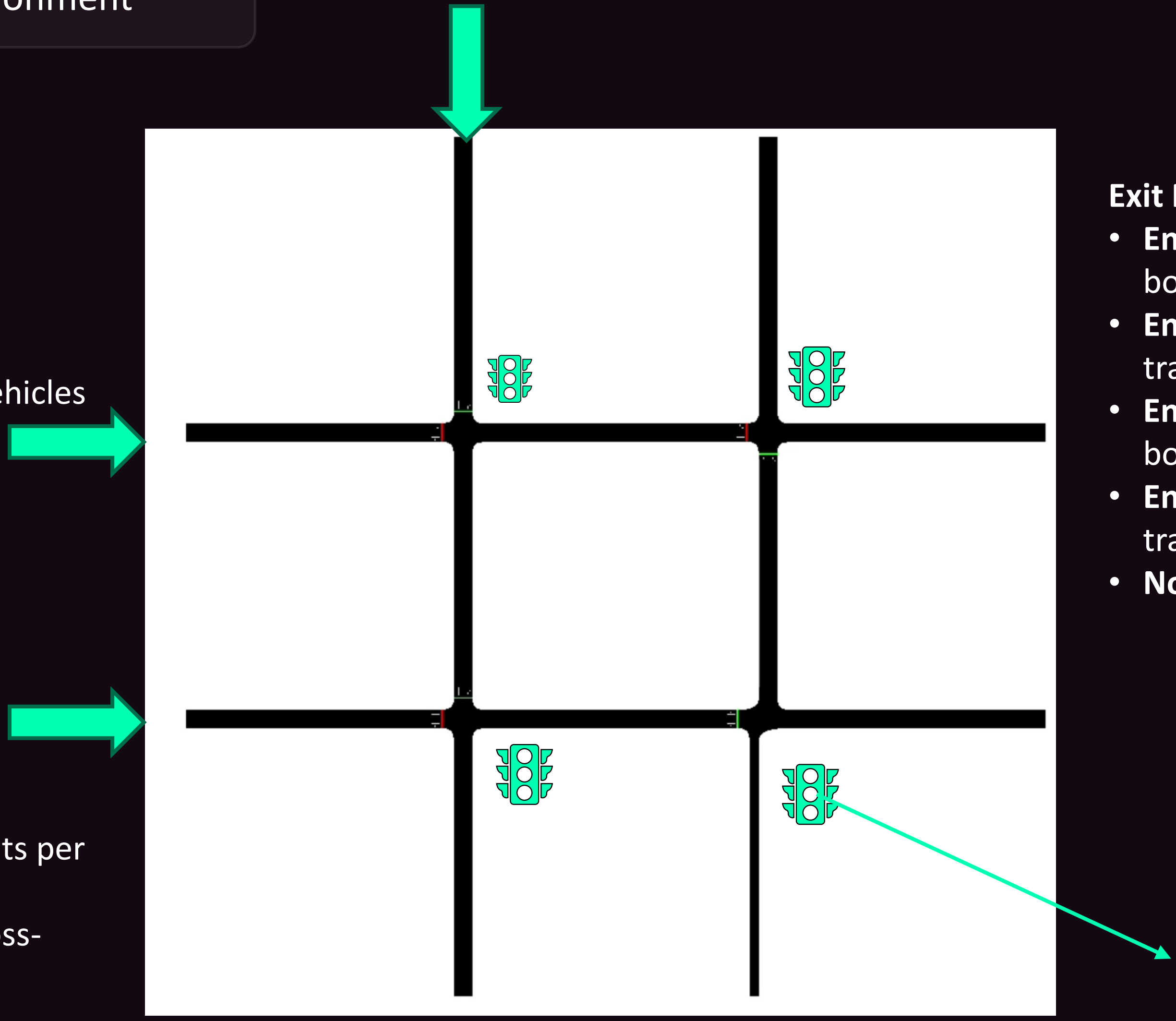
Version: 3.8.3

• Results visualization • Performance plotting • Data analysis • Graph generation

Lebajoa Mphaloane
Presenter

Traffic Custom Environment

Traffic Entry Points:
- North Entry
- West Entry
- Lower-West Entry
- Each entry point has a flow rate of 500 vehicles per hour



Exit Points:
- **End_South1**: Exit for North and West-bound traffic
- **End_South2**: Exit for Lower East-bound traffic
- **End_East1**: Exit for East and South2-bound traffic
- **End_East2**: Exit for Lower West-bound traffic
- **North2**: Additional exit point

Intersection Control:
- 4 controlled intersections with 2 traffic lights per intersection
- Actuated traffic light control to prevent cross-traffic collisions

Traffic Light Phases:
•**Phase 0 (Green):** GGGrrr – 42 seconds
•**Phase 1 (Yellow):** Yyyrrr – 4 seconds
•**Phase 2 (Red):** rrrGGG – 42 seconds
•**Phase 3 (Yellow):** Rrryyy – 4 seconds

Traffic Flow Design:
- Multiple route options from each entry point
- Left and right turn capabilities at each intersection
- Strategically distributed traffic flow to mimic realistic urban patterns

Lebajoa Mphaloane
Presenter

States and Action spaces

State Observation Space:

•**State Space Formula:**

Observation size = **Number of Roads** × **5 metrics**

•**5 Metrics per Edge:**

- Queue Length
- Waiting Time
- Speed
- Throughput

TraCI

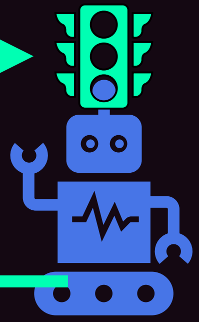
RL custom Environment

Action Space Formula

Formula: Action Space = 2^(Number of Intersections)

- For 4 intersections, the total possible actions = 16
- Binary Control per Intersection:
    - 0: Maintain current phase
    - 1: Switch to next phase

Example: Action 6 (0110) means: - Junction 1: Keep current phase - Junction 2: Switch phase - Junction 3: Switch phase - Junction 4: Keep current phase



DQN AGENT

Lebajoa Mphaloane

Presenter

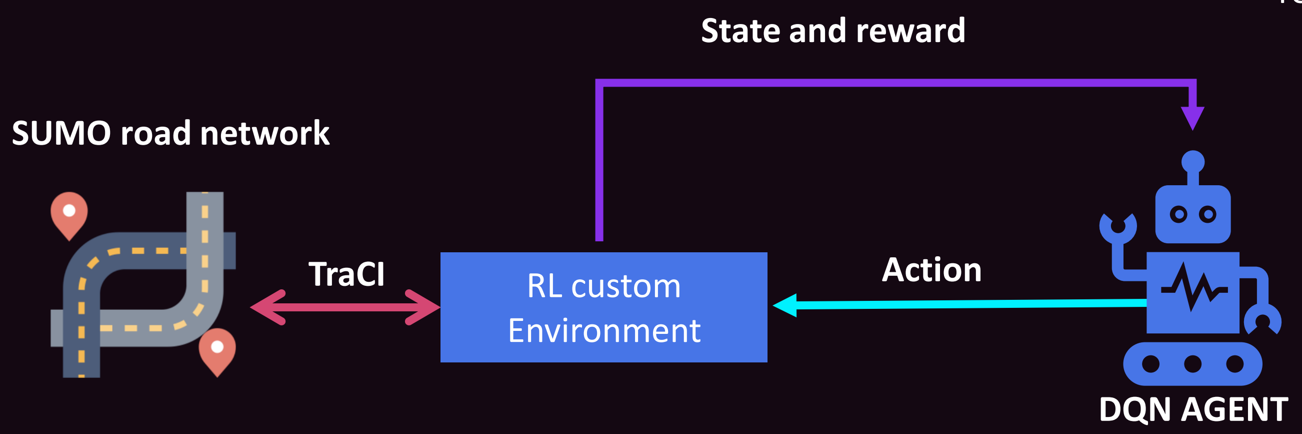
Rewards

Waiting Time Component:
- Measures change in total waiting time before and after action
- Normalized by dividing by 60 (seconds)
- Positive reward for decreased waiting time

Queue Length Component
- Tracks change in number of halting vehicles
- Normalized by dividing by 10 (vehicles)
- Positive reward for decreased queue length

Speed Component
- Measures the average speed in the network
- Higher speeds yield better rewards

Throughput Component
- Measures vehicles successfully exiting the network
- Encourages efficient network clearance
- More cars leaving the network more positive rewards

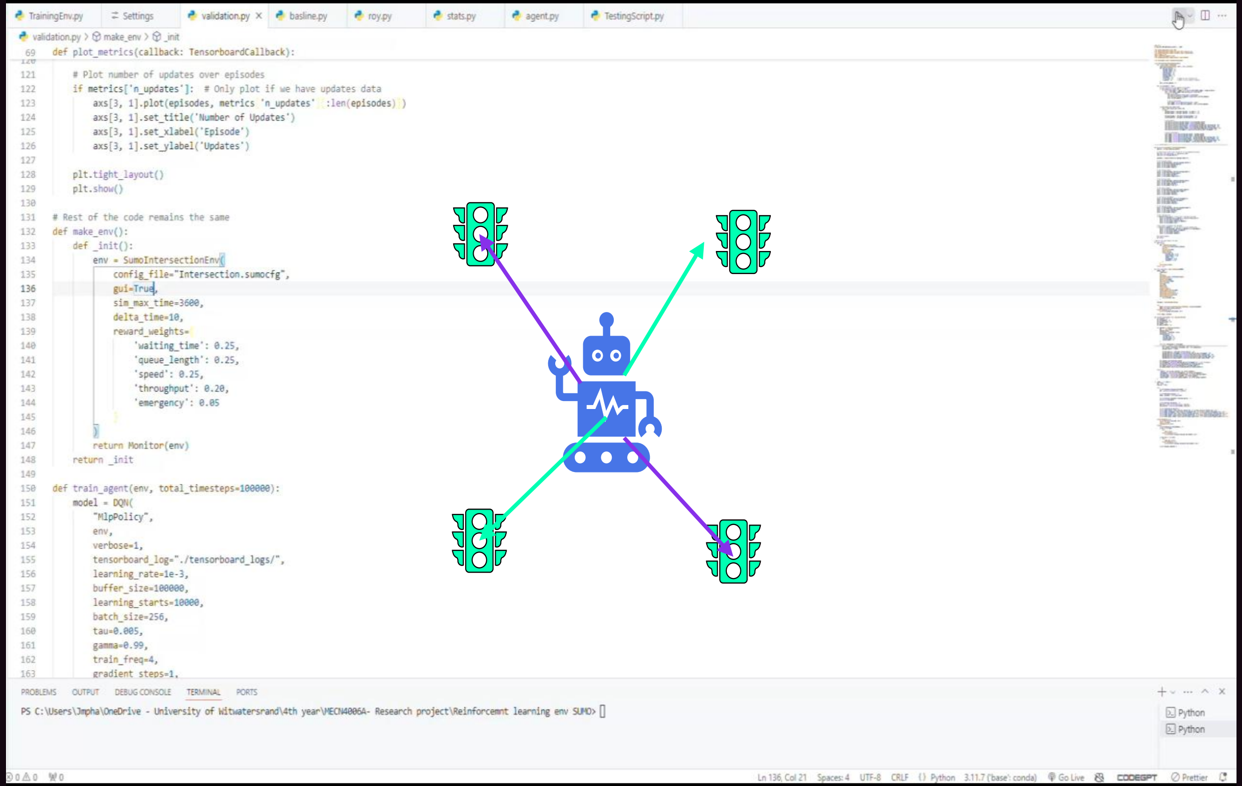


Lebajoa Mphaloane
Presenter

Training

Duration & Phases: The training lasts 100,000 timesteps, divided into four phases:

1. **Initialization (0-10,000 steps):** The agent explores randomly to gather experience without updates.
2. **Early Learning (10,000-40,000 steps):** Exploration is high, and the agent starts updating its policy.
3. **Refinement (40,000-70,000 steps):** Exploration decreases, and the agent fine-tunes its policy.
4. **Final Phase (70,000-100,000 steps):** Minimal exploration (5%), focusing on policy optimization and performance validation.

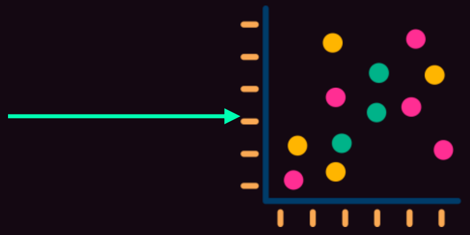


Learning Configuration:

- The agent uses a learning rate of 0.001, large experience buffer (100,000), and batch size of 256.
- Exploration gradually reduces from 100% to 5%, with a discount factor (γ) of 0.99 to prioritize long-term rewards.

Metrics Tracked:

- Episode rewards, waiting times, queue lengths, average speeds, network throughput, and training loss.
- These metrics help monitor performance, congestion reduction, traffic flow, and learning progress.



Lebajoa Mphaloane
Presenter

Verification and Validation

Cross-Environment Validation

•Test Environments:
- Training environment (4 intersections)
- Braamfontein network
- Johannesburg network (23 intersections)

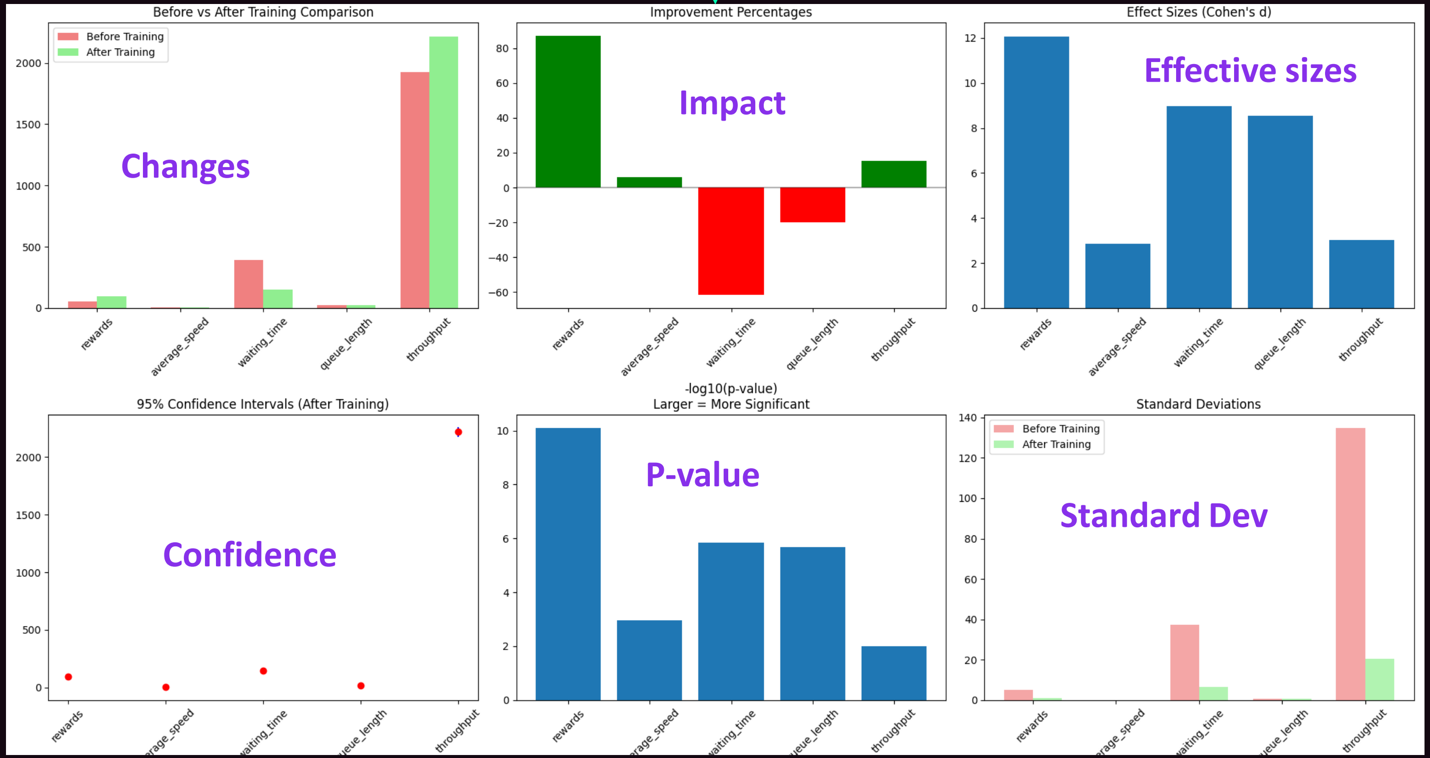
Baseline Performance Comparison

- Pre-Learning Metrics: Total Rewards: 42-65
- Average Speed: 3.4-4.4 m/s
- Waiting Time: 250-500 seconds
- Network Throughput: 1891-2096 cars/hour
- Queue Length: 36-40 vehicles

Statistical Validation

•Performance Metrics:
- Mean values with standard deviations
- Confidence intervals
- Effect sizes (Cohen's d)

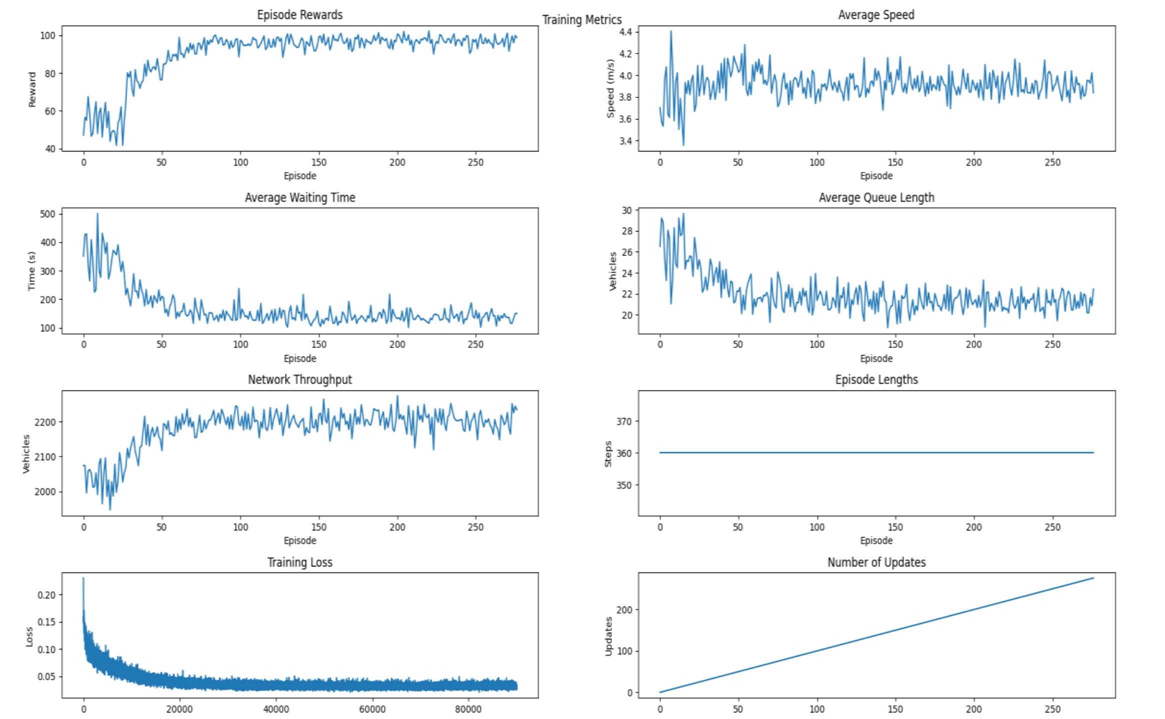
•Stability Measures:
- Consistent performance across episodes
- Reliable response to traffic variations
- Robust handling of edge cases



Lebajoa Mphaloane
Presenter

1. Episode Rewards

- **Initial Phase:** Fluctuated between 40-60 during early exploration (first 10,000 timesteps).
- **Breakthrough:** Around episode 25, rewards rose significantly to ~80.
- **Stabilization:** Final rewards stabilized at 96-100, reflecting optimal traffic management.
- **Insight:** Improved rewards correlate with better traffic efficiency, validating the reward structure



2. Average Speed

- **Initial Volatility:** Ranged between 3.4 and 4.2 units.
- **Stabilization:** Settled at 3.8-4.0 units post-learning, with minor fluctuations.
- **Insight:** The agent balanced traffic flow, avoiding congestion and maintaining safe speeds.

3. Average Waiting Time

- **Initial Peaks:** High waiting times of 400-500 units.
- **Improvement:** Stabilized at 150-200 units (~60% reduction).
- **Insight:** Significant reduction in delays, showing effective signal timing optimization.

4. Average Queue Length

- **Initial Fluctuations:** Queue lengths peaked at 28-30 vehicles.
- **Stabilization:** Decreased and stabilized around 22 vehicles (~25% reduction).
- **Insight:** Improved congestion management, leading to better traffic flow and reduced delays.

6. Training Loss

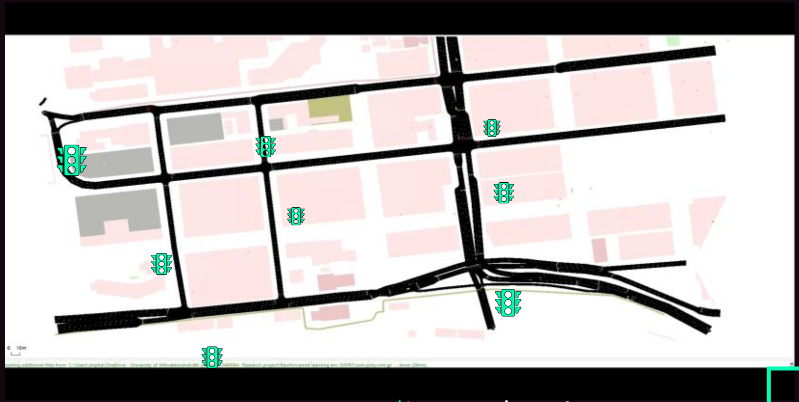
- **Initial Phase:** Started at ~0.20, showing rapid decline early on.
- **Final Phase:** Converged below 0.05, indicating effective learning and model fine-tuning.
- **Insight:** Low final loss values demonstrate a stable and reliable traffic management policy.

5. Network Throughput

- **Initial Throughput:** ~1897 units.
- **Improvement:** Stabilized above 2460 units (~30% increase).
- **Insight:** Enhanced system capacity, with more vehicles efficiently processed through intersections.

Lebajoa Mphaloane
Presenter

Braamfontein Testing



States and rewards

Learned actions

Trained DQN agent

Episode Rewards

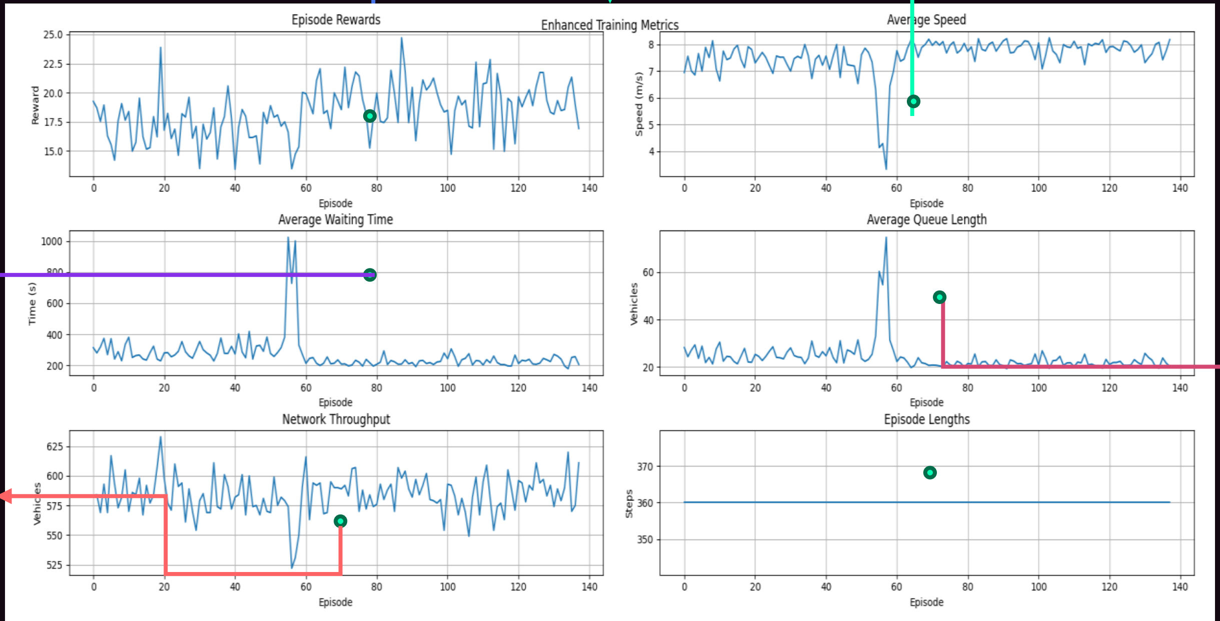
- **Fluctuation:** Varied between 15.0 and 22.5.
- **Observation:** Oscillations with occasional peaks (~22.5) and troughs (~15.0).
- **Insight:** Reflects the agent's adaptive response to dynamic real-world traffic conditions.

Average Waiting Time

- **Normal Range:** 200-400 units.
- **Spike:** Reached ~1000 units around episode 50.
- **Stabilization:** Returned to 200-250 units after the peak.
- **Insight:** Indicates the agent's capability to recover from temporary congestion.

Network Throughput

- **Stable Range:** 525-625 units.
- **Observation:** Maintained consistent throughput despite variations in other metrics.
- **Insight:** Demonstrates effective traffic flow management even under challenging conditions.



Average Speed

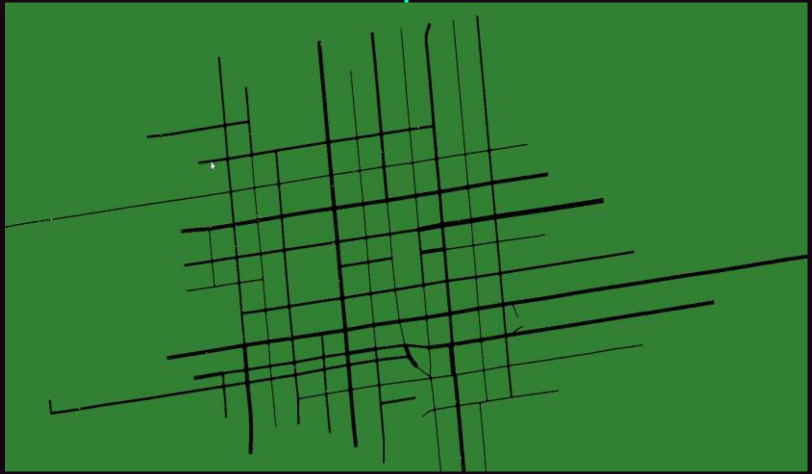
- **Stable Range:** 7-8 units.
- **Disruption:** Temporary drop to ~4 units around episode 60.
- **Recovery:** Quickly returned to normal levels.
- **Insight:** Demonstrates resilience to temporary traffic disturbances.

Average Queue Length

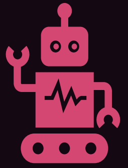
- **Stable Range:** 20-30 vehicles.
- **Spike:** Brief increase to ~60 vehicles around episode 60.
- **Recovery:** Returned to 20-25 vehicles post-disruption.
- **Insight:** Effective congestion management with quick recovery from anomalies

Large Johannesburg Testing

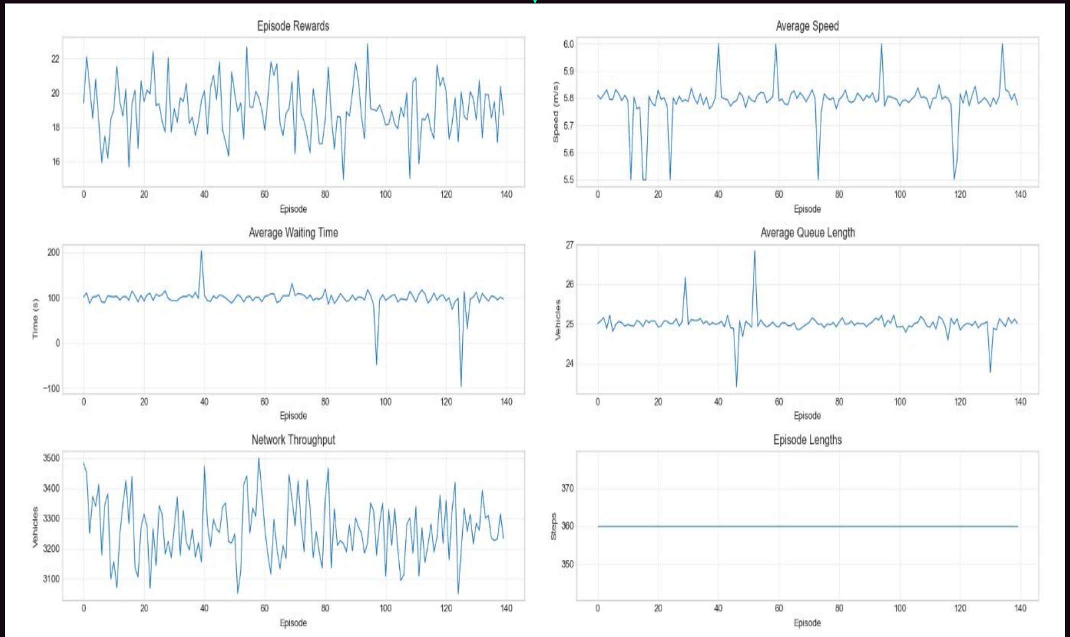
States and rewards



Learned actions



Trained DQN agent



Operational Metrics

- **Average Speed:** Stable at 5.5-6.0 units, with occasional dips, ensuring smooth traffic flow.
- **Average Waiting Times:** Typically, around 100 units, with two anomalies showing negative spikes around episodes 100 and 120.
- **Queue Lengths:** Controlled within 24-26 vehicles, with minor fluctuations and occasional brief spikes.

Episode Rewards and Network Performance

- **Episode Rewards:** Stable range of 15-20 units, with occasional peaks up to 25.
- **Network Throughput:** Consistent performance between 3100-3500 vehicles, indicating robust handling of higher traffic volumes.
- **Observation:** The system maintained effective traffic management across the expanded network despite reduced architecture.

Lebajoa Mphaloane
Presenter

Objective Achievement:

- The research successfully developed a custom SUMO-based simulation environment and trained a DQN agent to optimize traffic signal control effectively.

Traffic Efficiency Improvement:

- The DQN agent significantly reduced waiting times, queue lengths, and overall congestion, enhancing traffic flow in both simulated and real-world urban networks.

Reward Structure Effectiveness:

- A comprehensive and well-designed reward structure was crucial in guiding the DQN agent’s learning and promoting efficient traffic management strategies.

Generalization Capability:

- The trained agent demonstrated strong adaptability across different traffic patterns and congestion levels, performing effectively in the Braamfontein and Johannesburg networks.

Performance Variability:

- Some inconsistencies in performance were observed, especially in larger networks, indicating room for further improvement in model robustness and reliability.

Scalability Limitations:

- The DQN model faced computational and memory challenges when applied to larger, multi-intersection networks, highlighting scalability as a key area for future work.

Potential for Real-World Application:

- Despite limitations, the system shows promise for deployment in urban traffic management, provided further refinements are made.

Need for Broader Testing:

- The current testing scope lacked real-world complexities such as pedestrian crossings, weather variations, and emergency scenarios, limiting the model’s robustness.

Lebajoa Mphaloane
Presenter