TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO.: T24/080

Adaptive Traffic Signal Control using PN-D3QN in a Calibrated SUMO Environment

by

Utsav Dahal

A THESIS

SUBMITTED TO THE DEPARTMENT OF CIVIL ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE IN TRANSPORTATION ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

LALITPUR, NEPAL

MAY, 2026

COPYRIGHT

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF CIVIL ENGINEERING

The undersigned certify that they have read and recommended to Institute of Engineering for acceptance, a thesis report entitled **"Adaptive Traffic Signal Control using PN-D3QN in a Calibrated SUMO Environment"** submitted by Utsav Dahal in partial fulfillment of the requirement for degree of Master of Science in Transportation Engineering.

.....

Supervisor: Dr. Pradeep Kumar Shrestha
Assistant Professor
Institute of Engineering

.....

External Examiner: Mr. Prabesh Adhikari
Manager, Civil Aviation Authority of Nepal

.....

Asst. Prof. Anil Marsani
Program Coordinator
M.Sc. in Transportation Engineering
Department of Civil Engineering

Date: 11 May 2026

ABSTRACT

Urban intersections in Kathmandu face severe congestion, made worse by rigid fixed-time traffic signal controllers that cannot adapt to random traffic fluctuations. This study applies a Deep Reinforcement Learning (DRL) approach using a Prioritized Noisy Dueling Double Deep Q-Network (PN-D3QN) for adaptive signal control at Narayan Gopal Chowk, a busy four-legged intersection in Kathmandu, Nepal. The agent is trained and tested in a calibrated SUMO microscopic simulation that reproduces local heterogeneous traffic, where key car-following and lane-changing parameters are tuned using the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm to match observed vehicle headways and speeds. Real-time traffic state is represented using approach-level queue lengths and accumulated waiting times, and a normalised reward function penalising increases in both delay and queue length enables the agent to learn signal phase-switching policies without predefined cycle lengths or manual timing plans.

The trained PN-D3QN controller is compared against the existing fixed-time plan over six demand levels ranging from 85% to 115% of the calibrated base volume. Across these scenarios, PN-D3QN consistently reduces total intersection waiting time, with improvements of about 22–24% at 85–90% demand, roughly 39% at 95% demand, and around 12% at 110% demand, while still achieving a positive but smaller gain (about 2%) at 115% demand. At low demand (85%), the agent also reduces mean queues by about 24%, whereas at higher demand levels it accepts 7–25% higher queues in exchange for the observed delay savings, reflecting a learned trade-off between flow stability and queue dissipation. These results demonstrate that the PN-D3QN architecture can robustly adapt to fluctuating traffic and outperform a calibrated fixed-time controller in a realistic, microscopically modelled developing-city intersection. The work is currently limited to a single isolated junction without explicit pedestrian modelling, highlighting clear opportunities for future extensions to multi-intersection networks and multimodal signal control.

Keywords: Deep Reinforcement Learning, SUMO Calibration, PN-D3QN, Adaptive Signal Control, Heterogeneous Traffic

ACKNOWLEDGEMENT

I would like to express my profound gratitude to my supervisor, Dr. Pradeep Kumar Shrestha, for his continuous guidance, invaluable insights, and meticulous feedback, which were instrumental in shaping the technical depth of this thesis. I also extend my sincere thanks to Asst. Prof. Anil Marsani, Program Coordinator of the M.Sc. in Transportation Engineering, and the Department of Civil Engineering, Pulchowk Campus, for providing the necessary academic environment and administrative support. Furthermore, this research was made possible through the generous financial support of the Society of Transport Engineers Nepal (SOTEN), whose research grant significantly facilitated the practical execution and computational phases of this project.

The empirical foundation of this study relied heavily on real-world traffic data, which could not have been acquired without immense community and institutional support. I owe a special debt of gratitude to the generous house owners near the study intersection who kindly allowed us to install CCTV cameras on their premises, and to the Kathmandu Valley traffic personnel for their crucial coordination during our field data collection. Finally, I wish to thank Santosh Dahal and all my friends and colleagues for their continuous moral support and technical assistance, as well as everyone who directly or indirectly contributed their time and goodwill toward the successful completion of this thesis report.

Name: Utsav Dahal
Roll No: 080MSTrE024

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	Action Space
$A(s,a)$	Advantage Function
a_k, c_k	SPSA Step Sizes
g	Stochastic Gradient
G_{min}	Minimum Longitudinal Gap
L	Loss Function (Mean Squared Prediction Error)
L_q	Queue Loss
L_t	Throughput Loss
Q_{field}	Field-Measured Queue Length
Q_{sim}	Simulated Queue Length
$Q_\theta(S,A)$	Action-Value Function
R	Reward Function
S	State Space
T_{field}	Field-Measured Throughput
T_{sim}	Simulated Throughput
$V(s)$	Value Function
W	Cumulative Waiting Time
Y_t	Target Q-Value
γ	Discount Factor
Δ	Random Perturbation Vector
ΔW	Increase in Total Waiting Time
θ	Parameter Vector
σ	Driver Imperfection
τ	Headway Time / Reaction Time

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
API	Application Programming Interface
CNN	Convolutional Neural Network
D3QN	Dueling Double Deep Q-Network
DDQN	Double Deep Q-Network
DNN	Deep Neural Network
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
GEH	Geoffrey E. Havers (Statistic)
IOE	Institute of Engineering
IS	Importance Sampling
MSPE	Mean Squared Prediction Error
PER	Prioritized Experience Replay
PN-D3QN	Prioritized Noisy Dueling Double Deep Q-Network
RL	Reinforcement Learning
RMSE	Root Mean Square Error
SPSA	Simultaneous Perturbation Stochastic Approximation
SUMO	Simulation of Urban Mobility
TraCI	Traffic Control Interface

CHAPTER 1: INTRODUCTION

1.1 Background

Kathmandu's current traffic signal systems are largely unable to cope with the highly dynamic and unpredictable nature of urban traffic. Most intersections still rely on fixed-time or very basic semi-actuated cycles, and many signals do not operate reliably in everyday conditions. It is common to find lights malfunctioning, stuck on a single phase, or turned off altogether, forcing traffic police to manually direct vehicles for long hours. Since these static controls cannot adjust to sudden fluctuations in traffic volume, mixed-vehicle behavior, or pedestrian surges, intersections quickly become inefficient and congested [2]. This not only leads to long queues and lost travel time but also increases fuel consumption and emissions as vehicles idle or repeatedly stop and start. Over time, these inefficiencies compound, contributing significantly to both operational costs and Kathmandu's already worsening air quality [3]. Overall, the reliance on outdated or non-functional signal systems leaves virtually no room for real-time optimization, underscoring the need for more adaptive, intelligent traffic control approaches.

Reinforcement learning (RL) offers a promising alternative by continuously adapting to the live traffic state [2]. In RL-based control, an agent observes the intersection (via sensors or simulated detectors) and learns signal policies through trial and error. This model-free approach can allow continuous adaptation to real-time variations such as vehicle density, queue lengths, and arrival rates rather than following a fixed cycle [4]. In practice, deep RL methods have shown strong performance in traffic control [3]. For instance, a Deep Q-Network (DQN) uses a neural network to approximate action-values, with replay buffers and target networks for stable learning, enabling coordinated control across multiple junctions [5, 6]. Variants like Double DQN and Dueling DQN further improve learning: they correct value overestimation and help capture complex spatio-temporal traffic patterns [8]. Crucially for Kathmandu's mixed-traffic context, RL requires no explicit lane-based traffic model it learns directly from raw inputs (e.g., queue lengths or vehicle counts) and optimizes signal timings accordingly [2]. Moreover, the reward function can be tailored

to local priorities (for example, minimizing idle emissions or penalizing excessive queue lengths), effectively embedding domain knowledge into the controller [7]. In summary, modern deep-RL algorithms (DQN, DDQN, Dueling-DQN, actor–critic, etc.) offer an adaptive, data-driven strategy that can respond to chaos on the road, unlike rigid fixed-time or actuated systems [3, 8] .

To develop and test such a system for Kathmandu, SUMO-based simulation is ideal. SUMO (Simulation of Urban MObility) is an open-source microscopic traffic simulator that can model heterogeneous flows (cars, buses, motorcycles, bicycles, and pedestrians) and custom driver behaviors [16]. Its flexible, modular design with a Python API (TraCI) has made it the preferred platform for RL traffic-signal experiments [5, 16, 17]. By building a realistic SUMO model of Narayan Gopal Chowk, one can replicate Kathmandu-specific features mixed vehicle compositions, non-standard intersection geometry, and weak lane discipline and interface the simulator with an RL agent [18, 19]. The agent can then be trained with locally adapted reward functions (for example, balancing vehicle throughput and minimizing idle emissions). In this way, SUMO provides a practical testbed for creating a Nepal-tailored RL signal controller [20]. Such research promises a significant step: replacing Kathmandu's outdated signal plans with a learned, adaptive controller that could markedly cut congestion and pollution at critical junctions.

1.2 Problem Statement

Urban intersections in Kathmandu face severe recurring congestion driven by heterogeneous traffic comprising motorcycles, microbuses, and private vehicles operating without strict lane discipline [2, 3]. Existing fixed-time and semi-actuated traffic signals operate on rigid, predetermined cycle lengths that cannot respond to real-time fluctuations in demand, resulting in inefficient green-time allocation, queue spillback, and elevated vehicle emissions [3].Reinforcement learning has demonstrated strong promise in managing complex, dynamic, and heterogeneous traffic conditions in other urban settings [5, 6, 7], making it a viable candidate for addressing Kathmandu's intersection inefficiencies.
However, existing reinforcement learning studies for traffic signal control mostly rely on default car-following models, such as the Krauss and Intelligent Driver Model (IDM),

that assume strict lane discipline and homogeneous, car-dominated traffic [8, 15]. These assumptions are fundamentally unsuitable for Kathmandu's mixed traffic conditions, where motorcycles constitute the dominant vehicle class, lane boundaries are routinely ignored, and heterogeneous vehicle interactions between motorcycles, microbuses, and heavy vehicles produce complex gap-acceptance and merging behaviors that default models cannot replicate. Furthermore, variations in vehicle composition, fluctuating arrival rates, and localized bottlenecks frequently render fixed-timed green phases ineffective [2], compounding the limitations of conventional approaches. Therefore, there is a critical need to develop a calibrated SUMO-based PN-D3QN framework that can learn adaptive green duration selection policies responding to these real-world dynamics [9, 10, 11], reduce average waiting times and emissions, and provide a scalable, data-driven alternative to traditional traffic control strategies in Kathmandu [20].

1.3 Research Objectives

The primary aim of this research is to develop a reinforcement-learning-based adaptive traffic signal control system for an isolated urban intersection that can effectively manage realistic mixed-traffic conditions using SUMO simulation. The specific objectives are:

- To develop a realistic and calibrated simulation model by representing heterogeneous Kathmandu traffic in SUMO, incorporating multiple vehicle types so that the simulated environment accurately reflects real urban traffic conditions.
- To design and implement a Prioritized Noisy Dueling Double Deep Q-Network (PN-D3QN) based adaptive traffic signal control algorithm, enabling the controller to autonomously learn optimal green duration selection policies for each fixed signal phase based on real-time traffic states within the calibrated micro-simulation environment.

1.4 Scope of the Study

This section delineates the specific focus areas and methodological boundaries of the present research, covering the reinforcement learning agent design, simulation platform, and performance evaluation framework.

1.4.1 PN-D3QN Agent Architecture

This study develops a Prioritized Noisy Dueling Double Deep Q-Network (PN-D3QN) for adaptive traffic signal control at Narayan Gopal Chowk, Kathmandu. The agent architecture integrates four advanced DQN enhancements: Double DQN to reduce overestimation bias, Dueling networks to decompose state values and action advantages, Prioritized Experience Replay for efficient learning from critical transitions, and Noisy Networks for exploration without epsilon decay. The agent observes real-time traffic states via approach-level queue lengths and accumulated waiting times, and learns phase-switching policies by minimizing a normalized reward function that penalizes increases in both delay and queue formation.

1.4.2 Calibrated SUMO Microsimulation Environment

The PN-D3QN agent is trained and evaluated in a calibrated SUMO environment that reproduces heterogeneous traffic at the study intersection. Car-following and lane-changing parameters are tuned using the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm to match field-measured vehicle headways, speeds, and queue lengths. This validated microsimulation platform provides a realistic testbed for RL-based signal control before real-world deployment.

1.4.3 Study Intersection and Performance Evaluation

The scope is limited to a single four-legged signalized intersection (Narayan Gopal Chowk) representing typical urban conditions in developing cities with mixed traffic composition. The PN-D3QN controller is benchmarked against the existing fixed-time signal plan across six demand scenarios (85–115% of calibrated base volumes) using metrics of total waiting time and mean queue length.

1.5 Limitations of the Study

While the study achieves its core research objectives, several methodological and contextual constraints are acknowledged, primarily related to pedestrian modeling, measurement infrastructure, and network-level coordination.

1.5.1 Pedestrian Phase Exclusion

While SUMO supports pedestrian simulation, this study deliberately excludes pedestrian phases to maintain tractable state and action spaces for the first PN-D3QN implementation and to focus calibration on vehicular interactions, which constitute the dominant delay source at Narayan Gopal Chowk. Integration of pedestrian-aware reward components is identified as a key direction for future research.

1.5.2 Speed Measurement Constraints

Free-flow vehicle speeds were measured using a 20 m trap length, shorter than the 27 m segment adopted in comparable video-based speed studies on South Asian heterogeneous traffic [21], due to site-specific constraints including CCTV mounting geometry and stop-line proximity to the camera's focal zone. However, observed approach speeds (15–55 km/h across all vehicle classes) ensure that timing errors remain below 5%, within acceptable limits for microsimulation calibration [20].

1.5.3 Single-Intersection Focus

This study is confined to an isolated intersection and does not address network-level signal coordination or multi-agent control. Multi-intersection coordination represents an important extension for future work.

1.6 Organization of the Thesis

This thesis is organized into five chapters, following the standard structure for master's research at the Institute of Engineering, Tribhuvan University.

Chapter 1: Introduction presents the background and motivation for adaptive traffic signal control in Kathmandu, the problem statement addressing limitations of fixed-time controllers, research objectives, scope covering PN-D3QN agent development and SUMO calibration, and limitations of the study.

Chapter 2: Literature Review provides a comprehensive review of relevant research on deep reinforcement learning for traffic signal control, microscopic traffic simulation and calibration methods, and adaptive signal control implementations globally and in the South Asian context. It covers Q-learning and Deep Q-Network variants, SUMO calibration

techniques using optimization algorithms, state and reward function design for RL agents, and identifies the research gap addressed by this study.

Chapter 3: Research Methodology describes the study area at Narayan Gopal Chowk intersection, field data collection procedures including video-based traffic volume counts and speed measurements, SUMO network modeling and calibration using SPSA algorithm, PN-D3QN agent architecture and training framework, state-action-reward formulation, and performance evaluation methodology.

Chapter 4: Results and Discussion presents SUMO calibration results including parameter convergence and validation against field-measured throughput and queue lengths, PN-D3QN training performance metrics, comparative evaluation of the trained agent against the existing fixed-time controller across multiple demand scenarios (85–115% base volume), and analysis of waiting time and queue length performance under varying traffic conditions.

Chapter 5: Conclusion and Recommendations summarizes key findings on PN-D3QN performance gains over fixed-time control, draws conclusions regarding the applicability of deep reinforcement learning for adaptive signal control in heterogeneous traffic conditions, discusses practical implications for traffic management in Kathmandu, and provides recommendations for policy implementation, real-world deployment considerations, and future research directions including multi-intersection coordination and pedestrian-aware control.

The thesis concludes with references and annexes containing detailed field data, SUMO calibration parameters, network configuration files, training hyperparameters, and supplementary performance evaluation results.

CHAPTER 2: LITERATURE REVIEW

2.1 Traffic Signal Control: From Fixed-Time to Adaptive Methods

Traffic signal control has evolved considerably over the past few decades, moving from simple fixed-time plans toward increasingly adaptive and data-driven strategies. Early systems operated on predetermined cycle lengths and phase splits optimized offline by traffic engineers based on historical demand patterns. While straightforward to implement, these controllers are fundamentally unable to respond to real-time fluctuations in traffic demand, leading to inefficient green-time allocation and unnecessary delays [2]. Semi-actuated and fully actuated systems introduced loop detectors to extend or cut short phases based on vehicle presence, representing a modest improvement. However, these sensor-based systems still follow predefined logic and cannot learn or optimize across multiple phases simultaneously [3]. Recent research has demonstrated that deep reinforcement learning offers a more powerful and flexible alternative, capable of directly learning adaptive control policies from traffic observations without requiring an explicit model of traffic dynamics [1, 3].

In the Nepalese context, the inadequacy of fixed-time signal systems is well-documented. Kathmandu Valley has over 1.5 million registered vehicles as of 2024, with motorcycles comprising approximately 60% of the fleet, imposing yearly economic losses estimated at around NPR 12 billion due to chronic congestion and traffic mismanagement [29]. Bajracharya and Dhungel (2023) applied SIDRA Intersection micro-simulation to analyze Keshar Mahal and Durbar Marg intersections in Kathmandu, demonstrating that pretimed signal coordination with an optimal network cycle reduced total travel time by 33.4% and total control delay by 48.8% compared to isolated signal operation [25]. Their study confirmed that Kathmandu intersections routinely operate near or above capacity during peak hours and that even simple coordination between adjacent fixed-time signals yields substantial performance gains, highlighting the need for more dynamic, demand-responsive control strategies. Most recently, Lalitpur Metropolitan City launched Nepal's

first intelligent adaptive traffic signal system along the Kupondole–Jawalakhel corridor in December 2024, equipped with vehicle detector machines that adjust green durations based on real-time vehicle density [27]. While this represents a landmark step for Nepal, the system relies on rule-based density thresholds rather than a learned policy, and no formal calibration or validation against a micro-simulation model has been reported [27]. This gap between rule-based adaptive systems and learning-based approaches directly motivates the present research.

2.2 Reinforcement Learning for Traffic Signal Control

The most significant methodological shift in traffic signal control research has been the adoption of Reinforcement Learning (RL), a machine learning paradigm in which an agent learns a control policy through iterative interaction with its environment [4]. Unlike model-based controllers that require explicit mathematical representations of traffic dynamics, RL operates in a model-free manner, learning directly from observed traffic states and received reward signals. This makes it particularly well-suited for environments characterized by stochastic, non-stationary demand such as urban intersections [2]. RL methods for traffic signal control broadly fall into three categories: value-based methods, policy-based methods, and actor-critic methods [4]. Among these, value-based approaches, specifically Q-learning and its deep neural network extensions, have dominated traffic signal control research due to their simplicity, stability, and suitability for discrete action spaces such as green duration assignment [6].

The foundational value-based approach, Q-learning, estimates the expected cumulative reward for each state-action pair and stores these estimates in a Q-table. However, the tabular approach becomes computationally infeasible as the state and action spaces grow large, which is inevitable in realistic traffic scenarios involving multiple lanes, vehicle types, and phase combinations [5]. Deep Q-Networks (DQN), introduced by DeepMind, replaced the Q-table with a deep neural network, enabling generalization across continuous and high-dimensional state spaces [6]. DQN also introduced two critical stabilization mechanisms: experience replay, which stores past transitions in a replay buffer and samples them randomly during training to break temporal correlations, and a target network,

which provides stable regression targets during parameter updates [5]. These contributions made DQN the foundational algorithm for the majority of subsequent deep reinforcement learning based traffic signal control research [1].

2.3 Deep Q-Network Variants and Architectural Improvements

Despite its success, vanilla DQN suffers from a well-known overestimation bias in Q-value estimates, arising from using the same network parameters for both action selection and value evaluation [8]. Double DQN (DDQN) addressed this by decoupling action selection from value evaluation: the online network selects the greedy action while the target network evaluates its value, resulting in more accurate and stable Q-value estimates [8]. The Dueling DQN architecture restructures the network into two parallel streams: a value stream that estimates the state value $V(s)$ and an advantage stream that estimates the relative benefit of each action $A(s,a)$, which are then combined to produce the final Q-value. This decomposition allows the agent to learn which states are inherently valuable regardless of the specific action taken, which is particularly beneficial in traffic control where many green duration choices may yield similar outcomes in low-demand conditions [8]. Liang et al. (2019) demonstrated that combining DDQN with the Dueling architecture significantly outperformed vanilla DQN in traffic light cycle control, achieving lower average delays and more stable training convergence across multiple intersection configurations [8].

Wei et al. (2018) proposed IntelliLight, a DRL-based traffic signal controller employing a value-based approach with image-like state representations, and demonstrated that reward functions focused on queue length minimization led to more transferable and practically deployable policies compared to delay-based rewards [7]. IntelliLight established several design principles widely followed in subsequent work, including the use of discrete, phase-indexed action spaces and localized state representations derived from vehicle counts at the detector level [7]. As the intersection studied in the present work has a small discrete action space corresponding to a fixed set of green duration choices, the value-based DQN family is more appropriate and aligns with recommendations from prior work including IntelliLight [7].

2.4 Exploration Strategies: Noisy Networks

Effective exploration is critical to the quality of the learned policy in DRL. The conventional ε-greedy strategy explores by taking random actions with a probability that decays over training. While simple to implement, ε-greedy exploration is undirected and treats all state-action pairs equally, making it inefficient in complex environments where informative transitions are sparse [9]. Noisy Networks, introduced by Fortunato et al. (2018), proposed a fundamentally different approach by injecting learnable parametric noise directly into the weights of the network's linear layers [24]. The noise parameters are learned alongside network weights through standard gradient descent, allowing the agent to adaptively control its own exploration intensity based on the uncertainty in its value estimates, and eliminating the need for any externally tuned exploration schedule [24]. Han et al. (2020) extended this concept through NROWAN-DQN, combining noise reduction mechanisms with online weight adjustment to produce more stable exploration, and demonstrating that noise-based exploration consistently outperforms ε-greedy across a range of control benchmarks [9].

2.5 Prioritized Experience Replay

Standard DQN samples transitions uniformly at random from the replay buffer, treating all experiences as equally informative. Schaul et al. (2016) challenged this assumption with Prioritized Experience Replay (PER), which assigns sampling probabilities proportional to the magnitude of each transition's temporal difference (TD) error [10]. Transitions with large TD errors represent experiences where the agent's current estimates are most inaccurate and therefore most informative for learning. By replaying high-priority transitions more frequently, PER accelerates convergence and improves final policy quality, particularly in environments where informative experiences are sparse [10]. Importance sampling weights are applied to correct for the bias introduced by non-uniform sampling, ensuring unbiased gradient updates as training progresses [10]. Cai and Wei (2024) applied PER within an enhanced DRL framework for urban traffic signal control and demonstrated that the combined PN-D3QN architecture, which integrates PER with Noisy Networks and the Dueling Double DQN structure, achieved the fastest convergence and the highest cumulative reward among all tested variants, providing direct empirical evidence for the

effectiveness of this combined architecture in traffic signal control [11].

2.6 State Representation in DRL-Based Traffic Signal Control

The design of the state representation is one of the most consequential decisions in DRL-based traffic signal control, as it determines what information the agent can observe and act upon. Early works used simple scalar features such as queue length or vehicle count per lane as inputs to fully connected networks [5]. More recent works have moved toward spatially structured representations, encoding traffic conditions as two-dimensional matrices or image-like tensors that capture the position and speed of vehicles across all lanes and approaches simultaneously [12, 13]. Such representations are well-suited for convolutional neural networks (CNNs), which can exploit spatial correlations between adjacent lanes and extract higher-level traffic features such as queue fronts and vehicle platoons [12]. Genders and Razavi (2016) proposed one of the earliest CNN-based state representations for DRL traffic signal control, encoding vehicle presence across discretized lane segments into a binary occupancy matrix and demonstrating that CNN-based agents learned more stable and generalizable policies than those using scalar inputs [13]. Li et al. (2020) and Wu et al. (2020) further refined spatial encoding strategies, showing that structured representations enabled more effective learning across multiple intersection configurations [12, 14]. The present study adopts a spatial tensor of shape $\mathbb{R}^{2\times8\times50}$, encoding vehicle speed and presence across all lanes and approaches, combined with a one-hot encoded current phase vector, following the design principles established in these prior works.

2.7 Microscopic Traffic Simulation and SUMO

Traffic simulation approaches are broadly classified into macroscopic and microscopic models [15]. Macroscopic models treat traffic as a continuous fluid flow, representing conditions through aggregate variables such as density, speed, and flow rate. While computationally efficient and suited to network-level planning, macroscopic models cannot capture individual vehicle behaviors, lane-level interactions, or the nuanced dynamics of mixed-traffic environments [15]. Microscopic simulation, in contrast, models each vehicle individually, specifying its car-following behavior, lane-changing decisions, acceleration,

and reaction time, making it far more appropriate for studying intersection-level performance and training RL agents on realistic traffic states [15].

SUMO (Simulation of Urban MObility) has emerged as the dominant open-source microscopic simulator for DRL-based traffic signal control research, primarily due to its modular architecture, support for heterogeneous vehicle types, and a Python API (TraCI) that enables real-time interaction between the simulator and an external control agent [16, 17]. Park et al. (2021) validated SUMO as a reliable training environment for DQN-based traffic signal controllers, demonstrating close agreement between simulated and field-measured performance metrics [5]. Gao et al. (2017) employed SUMO with experience replay and target networks to train a DRL agent for adaptive signal control, establishing the SUMO-TraCI-Python pipeline that has since become the standard experimental setup for the majority of DRL-TSC studies [17]. Ducrocq and Farhi (2023) further demonstrated SUMO's suitability for partial-detection scenarios, showing that DQN agents trained in SUMO maintained strong performance even when only a subset of vehicles were detected [19]. More broadly, SUMO has been employed as the simulation backbone across a wide range of DRL-TSC studies, confirming its role as the standard platform for this class of research [5, 16, 17, 18, 19, 20].

2.8 Micro-simulation Calibration for Heterogeneous Traffic Conditions

A critical prerequisite for using micro-simulation as a reliable training environment is the accurate calibration of car-following and lane-changing model parameters to match real-world traffic behavior. Default model parameters in tools such as SUMO and VISSIM are tuned for European car-dominated, lane-disciplined traffic and are not suitable for developing-country urban environments where motorcycles dominate, lane boundaries are routinely ignored, and inter-vehicle gaps are much smaller than those assumed by default models [15, 23].

This challenge has been demonstrated directly in the Nepalese context. Chhetri and Shrestha (2023) calibrated car-following parameters in VISSIM for heterogeneous traffic conditions at Putalisadak intersection in Kathmandu, finding that the default Widemann-74 parameters produced a Mean Absolute Percentage Error (MAPE) of 106% in simulated

delay compared to field observations, far exceeding the acceptable threshold [28]. Using Latin Hypercube Sampling (LHS) for parameter space exploration, one-way ANOVA for sensitivity analysis, and a multi-objective Genetic Algorithm for optimization, they identified three sensitive parameters (average standstill distance, additive part of safety distance, and multiplicative part of safety distance) and reduced the MAPE to 9.7% after calibration, with a volume validation MAPE of 2.07% [28]. This study directly confirms that default micro-simulation parameters are unsuitable for Kathmandu's non-lane-based, heterogeneous traffic and that systematic calibration is essential before any simulation-based analysis or agent training is conducted.

Jayasinghe et al. (2021) demonstrated analogous findings for Sri Lanka, developing an automated SUMO calibration framework using the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm and establishing it as a computationally efficient alternative to genetic algorithms, requiring only two objective function evaluations per iteration regardless of the number of parameters being calibrated [23, 22]. In their Colombo case study, they used SUMO's Krauss car-following model and lane-changing model, and calibrated one car-following parameter (the driver imperfection parameter σ) together with four lane-changing parameters lcSpeedGain, lcPushy, lcStrategic, and lcCooperative to reproduce observed mixed-traffic speed measurements on an urban corridor [23]. The calibrated values departed substantially from SUMO defaults, with higher σ and markedly increased `lcSpeedGain` and `lcPushy` indicating stronger random speed variation, a greater eagerness to change lanes to gain speed, and more pronounced lateral encroachment between vehicles than in lane-disciplined traffic [23]. These findings, supported by video observations of motorcycles and cars performing sharp lane-change manoeuvres to exploit small gaps, confirm that driver imperfection, lane-change aggressiveness, and lateral encroachment are the dominant behavioural levers that must be adjusted to represent heterogeneous South Asian traffic in SUMO.

Sashank and Navali (2021) similarly calibrated SUMO for Indian heterogeneous traffic, confirming that default parameters substantially overestimated headways and underestimated throughput in mixed-traffic environments [26]. Khaleghian et al. (2023) further validated the use of field speed measurements as calibration targets in SUMO, demonstrating that vehicle speed data alone could accurately constrain car-following model

parameters to reproduce observed traffic dynamics [20]. Collectively, these studies establish that systematic parameter calibration using field-measured traffic data is an indispensable step for any micro-simulation study conducted in South Asian mixed-traffic conditions, and they provide direct methodological guidance for the present work. In particular, the Sri Lankan results motivate the focus on parameters governing time headway, longitudinal and lateral gaps, lane-change aggressiveness, and driver imperfection as the most behaviourally influential degrees of freedom in SUMO's microscopic models for representing Kathmandu's heterogeneous, weak-lane-disciplined traffic.

2.9 Traffic Signal Studies in the Nepalese Context

Within Nepal, a growing body of research has examined intersection performance using simulation tools, though learning-based adaptive control approaches remain largely absent from the local literature. Bajracharya and Dhungel (2023) applied SIDRA Intersection micro-simulation to analyze Keshar Mahal and Durbar Marg intersections in Kathmandu, demonstrating that pretimed signal coordination reduced total travel time by 33.4% and total control delay by 48.8% compared to isolated signal operation [25]. Their study confirmed that Kathmandu intersections routinely operate near or above capacity under existing fixed-time plans, underscoring the need for more dynamic, demand-responsive strategies.

More recently, Maharjan and Budhathoki (2025) investigated the application of Multi-Agent Deep Deterministic Policy Gradient (MADDPG) for adaptive traffic signal control tailored to Kathmandu's heterogeneous traffic conditions, using SUMO calibrated with local traffic volumes [29]. Their preliminary findings reported that MADDPG reduced vehicle delay by 30–35%, queue lengths by 25%, and CO_2 emissions by 12% relative to fixed-time baselines [29]. While this represents the first application of a DRL-based traffic signal controller in a Kathmandu-specific SUMO environment, the study targets a multi-intersection network using a continuous action space DDPG-based approach, which differs fundamentally from the single isolated intersection and discrete green duration selection action space of the present study. Furthermore, the SUMO model used by Maharjan and Budhathoki (2025) was calibrated only with traffic volumes and lacks a formal multi-

parameter car-following calibration procedure, which the present study addresses through SPSA-based calibration [29]. These distinctions highlight the complementary and more rigorous nature of the present work within the emerging landscape of DRL-based traffic signal control research in Nepal.

Lalitpur Metropolitan City launched Nepal's first intelligent adaptive traffic signal system along the Kupondole–Jawalakhel corridor in December 2024, equipped with vehicle detector machines that adjust green durations based on real-time vehicle density [27]. While this represents a landmark step for Nepal, it relies on rule-based density thresholds rather than a learned policy, and no formal micro-simulation calibration or validation has been reported [27].

2.10 Summary and Research Gap

The literature reviewed above establishes a clear progression from fixed-time signal control toward increasingly sophisticated learning-based adaptive strategies. Value-based DRL methods, particularly DQN and its extensions including DDQN, Dueling DQN, Noisy Networks, and Prioritized Experience Replay, have consistently demonstrated reductions in intersection delay and queue length when trained and evaluated in SUMO micro-simulation environments [5, 6, 8, 11]. The combined PN-D3QN architecture has been shown to achieve superior convergence speed and policy quality compared to its individual component variants, making it the most appropriate choice for the present study [11]. Within Nepal, simulation-based studies have confirmed the severe inadequacy of default micro-simulation parameters for Kathmandu's heterogeneous traffic [28], the poor performance of fixed-time signal plans at key intersections [25], and the preliminary viability of DRL-based controllers in a local SUMO environment [29].

However, a critical gap remains. No existing study has applied a rigorously calibrated SUMO micro-simulation model of a specific Nepalese intersection, validated against field-measured queue lengths and throughput, as the training environment for a DRL-based traffic signal controller. Furthermore, virtually all existing DRL-TSC studies assume lane-disciplined, car-dominated traffic conditions [8, 15], which are fundamentally unsuitable for Kathmandu's heterogeneous mixed-traffic environment. The only Nepalese DRL

study identified, Maharjan and Budhathoki (2025), targets a multi-agent multi-intersection network with a continuous action space and lacks formal car-following calibration [29]. These gaps collectively define the precise contribution of the present study: a PN-D3QN agent trained within a SPSA-calibrated SUMO model of Narayan Gopal Chowk, designed to learn adaptive green duration selection policies for a single isolated intersection under Kathmandu's heterogeneous mixed-traffic conditions.

CHAPTER 3: RESEARCH METHODOLOGY

The methodological framework adopted in this study comprises four sequential phases, as illustrated in Figure 3.1. Phase I involves the field survey and primary data collection at Narayan Gopal Chowk, encompassing turning movement counts (TMC), back-of-queue measurements, and vehicle spot speeds, addressing the gap identified in the literature regarding the absence of calibrated, locally-validated simulation environments for heterogeneous South Asian traffic [28, 23]. Phase II covers SUMO network development and demand specification, constructing the digital twin of the intersection using OSM geometry empirically refined with field measurements.Phase III performs SPSA-based parameter calibration over 60 iterations against field-measured headways and speeds, with a validation gate requiring GEH < 5.0 per approach and homography-corrected Category C vehicle counts, directly responding to the problem of default SUMO parameters being unsuitable for non-lane-based mixed traffic [15]. Phase IV addresses the core research problem of fixed-time signal inadequacy [2, 3] by formulating the RL environment and training the PN-D3QN agent over 331 episodes, followed by comparative evaluation across six demand scenarios (85%–115%).

3.1 Study Area

Narayan Gopal Chowk is a four-legged signalised junction in Kathmandu connecting the Basundhara, Gaushala, Budhanilakantha, and Teaching approaches (approximately 27.74009° N, 85.33713° E). The intersection was selected as the study site due to its critical role as a high-conflict, multi-modal node along the Kathmandu Ring Road, where Ring Road transit volumes overlap with dense local commuter flows to create severe and recurring peak-hour bottlenecks. Its heterogeneous vehicle composition, absence of strict lane discipline, and geometric constraints collectively make it a representative site for evaluating adaptive signal control in a developing-city context.

The site acts as a major gateway connecting the dense urban core with northern residential

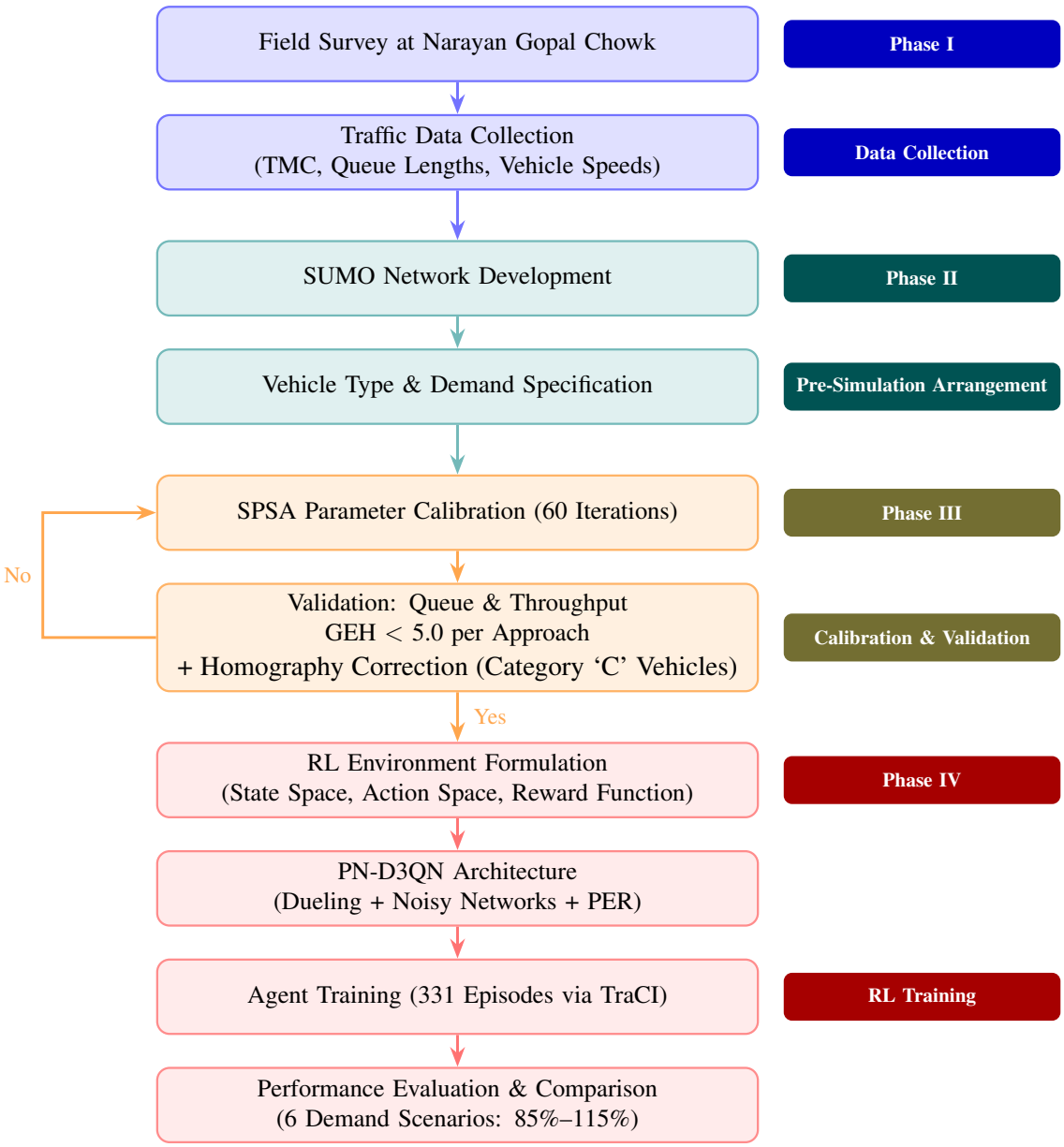


Figure 3.1: Overall four-stage research methodology framework.

corridors such as Budhanilakantha, resulting in a complex overlap of through-traffic and heavy turning movements. This mix of high-volume Ring Road transit and local commuter traffic creates significant bottlenecks, making it an ideal "laboratory" to evaluate traffic behaviour in a non-lane-based environment where motorcycles, buses, and heavy vehicles compete for limited space. The availability of consistent field data classified traffic volumes, queue lengths, and spot speeds provided the empirical foundation necessary to construct and calibrate a reliable SUMO digital twin of the intersection.

Figure 3.2 presents the spatial context of the study site. The satellite view confirms the intersection's position on the Kathmandu Ring Road and clearly shows the four approach roads Basundhara, Gaushala, Budhanilakantha, and Teaching converging at a common node. The image also captures the dense urban fabric surrounding the junction, reflecting the mixed land-use context that drives the heterogeneous and high-volume traffic demand observed at this location.

Figure 3.2: Satellite imagery of Narayan Gopal Chowk showing the four approaches within the Kathmandu Ring Road network.

The intersection operates under a four-phase pre-timed signal programme, as illustrated in Figure 3.3. The phase sequence is designed to serve the four principal movement groups in turn: through movements between Basundhara and Gaushala (Phase 1), service to the Budhanilakantha approach (Phase 2), service to the Teaching approach (Phase 3), and an exclusive right-turn phase for Basundhara-to-Teaching and Gaushala-to-Budhanilakantha

movements (Phase 4). Each phase is followed by a 5-second amber interval, yielding a total fixed cycle length of 301 seconds. This rigid, pre-timed operation is unable to adapt to real-time demand fluctuations, which directly motivates the development of the PN-D3QN adaptive controller proposed in this study.

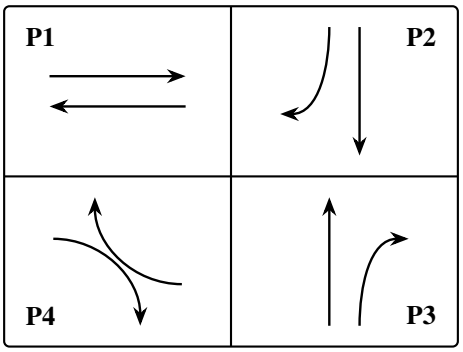


Figure 3.3: Existing four-phase signal operation at Narayan Gopal Chowk.

The detailed green durations and movement assignments for each phase are summarised in Table 3.1. Phase 1 and Phase 2 carry the highest traffic volumes, reflected in their longer green times of 83 and 85 seconds respectively, while Phase 4, serving exclusive right-turn movements only, operates with the shortest green duration of 43 seconds. These field-measured timings were directly adopted as the fixed-time baseline against which the trained PN-D3QN agent is benchmarked in Chapter 4.

Table 3.1: Legend for existing four-phase signal operation at Narayan Gopal Chowk

Phase	**Permitted movements**	**Field timing**
P1	Basundhara $\leftrightarrow$ Gaushala (Through)	83 s green + 5 s yellow
P2	Budhanilakantha $\rightarrow$ Teaching (Through), Budhanilakantha $\rightarrow$ Basundhara (Right)	85 s green + 5 s yellow
P3	Teaching $\rightarrow$ Budhanilakantha (Through), Teaching $\rightarrow$ Gaushala (Right)	70 s green + 5 s yellow
P4	Basundhara $\rightarrow$ Teaching (Right), Gaushala $\rightarrow$ Budhanilakantha (Right)	43 s green + 5 s yellow

3.2 Data Collection

Field data collection at Narayan Gopal Chowk comprised three primary components: turning movement counts, queue length observations, and vehicle spot speed measurements. Each component served a distinct role in constructing, populating, and calibrating the SUMO microsimulation model. All data were collected using CCTV cameras installed at elevated rooftop locations near the intersection, providing clear and unobstructed views of all four approaches, the stop lines, and the upstream queuing zones.

3.2.1 Network Geometry and Lane Configuration

The foundational road network geometry was imported directly from OpenStreetMap (OSM) and subsequently refined through field verification. To accurately represent the heterogeneous traffic behaviour and weak lane discipline at the intersection, lane widths and approach configurations were empirically calibrated based on the lateral occupancy of vehicles stopped during the red signal phase. This calibration was calculated using approximate vehicle dimensions and inter-vehicular lateral gaps extracted from CCTV footage, ensuring that the SUMO network reflects the actual usable carriageway width available to the mixed vehicle fleet rather than the nominal lane markings, which are routinely disregarded by drivers at this location.

3.2.2 Turning Movement Counts and Demand Profiles

Classified turning movement counts (TMC) were conducted manually from CCTV footage over three peak-hour survey days. Vehicle arrivals were recorded separately for each vehicle class motorcycles, cars, microbuses, and heavy vehicles and for each directional movement at every approach. The resulting counts were used to derive approach-wise demand profiles and route fractions, which were then converted into SUMO vehicle flows and turning probabilities. This procedure ensures that the simulated vehicle composition and directional splits reproduce the observed intersection demand. The complete TMC data are provided in the Annex.

3.2.3 Queue Length Observations

Maximum back-of-queue lengths were recorded for each approach during the same peak-hour survey windows as the turning movement counts. Queue lengths were measured from the stop line to the last stationary vehicle at the end of each red phase, using identifiable reference markers visible in the CCTV footage. These field-measured queue lengths served as the primary validation target for the calibrated SUMO model, with the GEH statistic computed per approach to confirm acceptable agreement between simulated and observed queuing behaviour, as detailed in validation section.

3.2.4 Vehicle Speed Measurements

A spot speed study was conducted on each approach of Narayan Gopal Chowk to obtain realistic free-flow operating speeds under uncongested conditions. Measurements were collected from CCTV footage using a trap-length method, retaining only vehicles travelling without upstream interference or queuing effects. The resulting approach-wise speed distributions were used to define the `maxSpeed` attributes of the corresponding SUMO edges, so that simulated free-flow speeds and acceleration behaviour closely match observed conditions along each approach. The complete speed study data are provided in the Annex.

3.2.4.1 Manual Speed Estimation from Video Frames

Vehicle speeds at each approach were extracted using a frame-based method applied to the CCTV footage, with two reference lines spaced $d = 20$ m apart drawn on the carriageway. For each vehicle trajectory, the frame in which the vehicle front crossed the upstream line was recorded as f_1, and the frame at which it crossed the downstream line was recorded as f_2 ($f_2 > f_1$). With the known video frame rate FPS, the elapsed travel time between the two lines is given by Equation (3.1):

$$t = \frac{f_2 - f_1}{\text{FPS}} \ [\text{s}] \tag{3.1}$$

and the corresponding free-flow speed is computed using Equation (3.2):

$$v = \frac{d}{t} = \frac{20}{t}\ [\mathrm{m/s}], \qquad v_{\mathrm{km/h}} = 3.6\,v\ [\mathrm{km/h}] \tag{3.2}$$

In practice, the crossing frames f_1 and f_2 were noted manually for each vehicle, and t, v, and $v_{\mathrm{km/h}}$ were computed using Equations (3.1) and (3.2) respectively. Figures 3.4, 3.5a, and 3.5b illustrate a sample trajectory and the corresponding speed calculation. The same procedure was applied to all approaches and vehicle classes at the intersection.

The 20 m trap length adopted in this study is shorter than the 27 m segment used in comparable video-based speed studies on heterogeneous South Asian traffic [21]. A 27 m trap could not be implemented at Narayan Gopal Chowk owing to three site-specific constraints:

(1) CCTV mounting geometry: Cameras were installed on adjacent building rooftops at a fixed elevation and oblique horizontal angle. The resulting perspective projection compressed the visible road length along each approach, limiting the unobstructed, co-planar section visible within a single frame to approximately 20–22 m per approach.

(2) Intersection geometry: The stop-line and conflict zone lay in close proximity to the camera's optimal focal zone, such that extending the trap to 27 m would have partially overlapped with the standing-queue region, where vehicles are not in free-flow motion and speed measurements would be unrepresentative of approach running speeds.

(3) Frame rate adequacy at 20 m: At the study site's camera frame rate of 30 fps, a 20 m trap yields a minimum measurable travel time of approximately 0.87 s for a vehicle travelling at 80 km/h, well within the resolution of manual frame-counting. Given the observed free-flow speed range at this intersection (15–55 km/h across all vehicle classes), timing errors at the 20 m trap length remain below 5%, which is considered acceptable for the purposes of car-following model calibration [20].

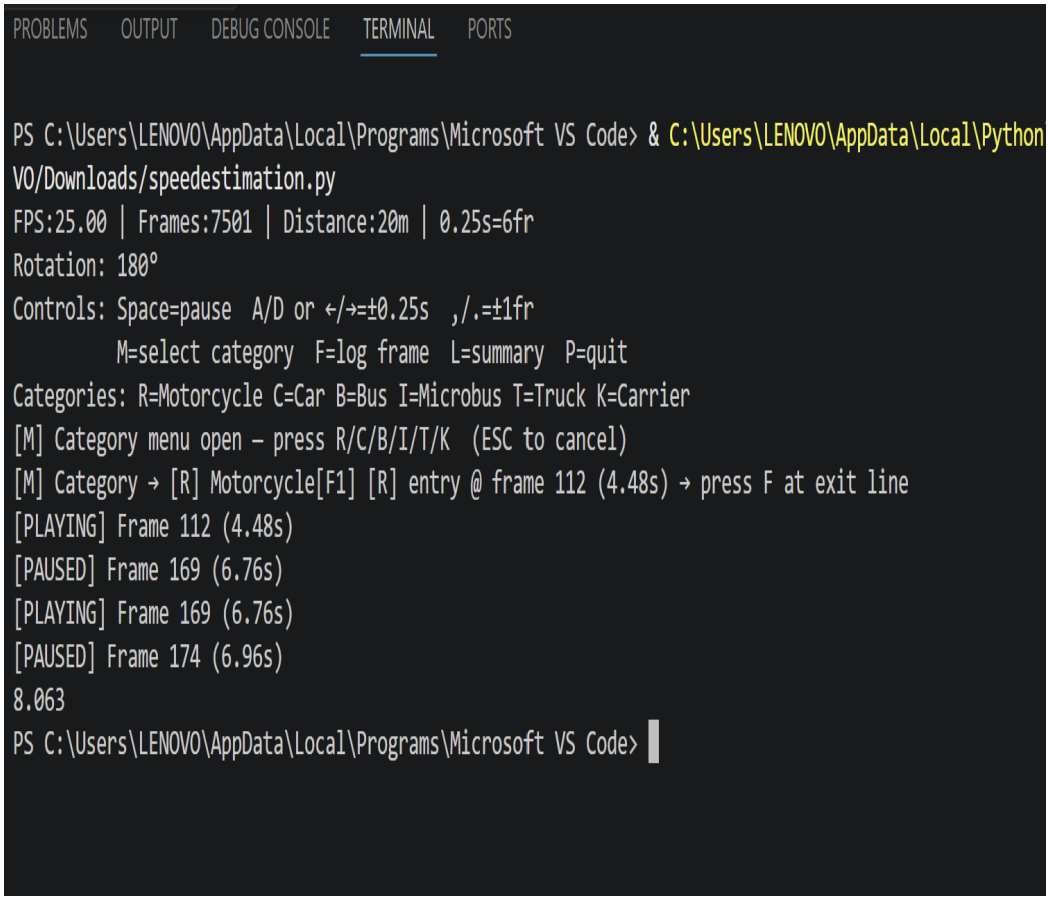
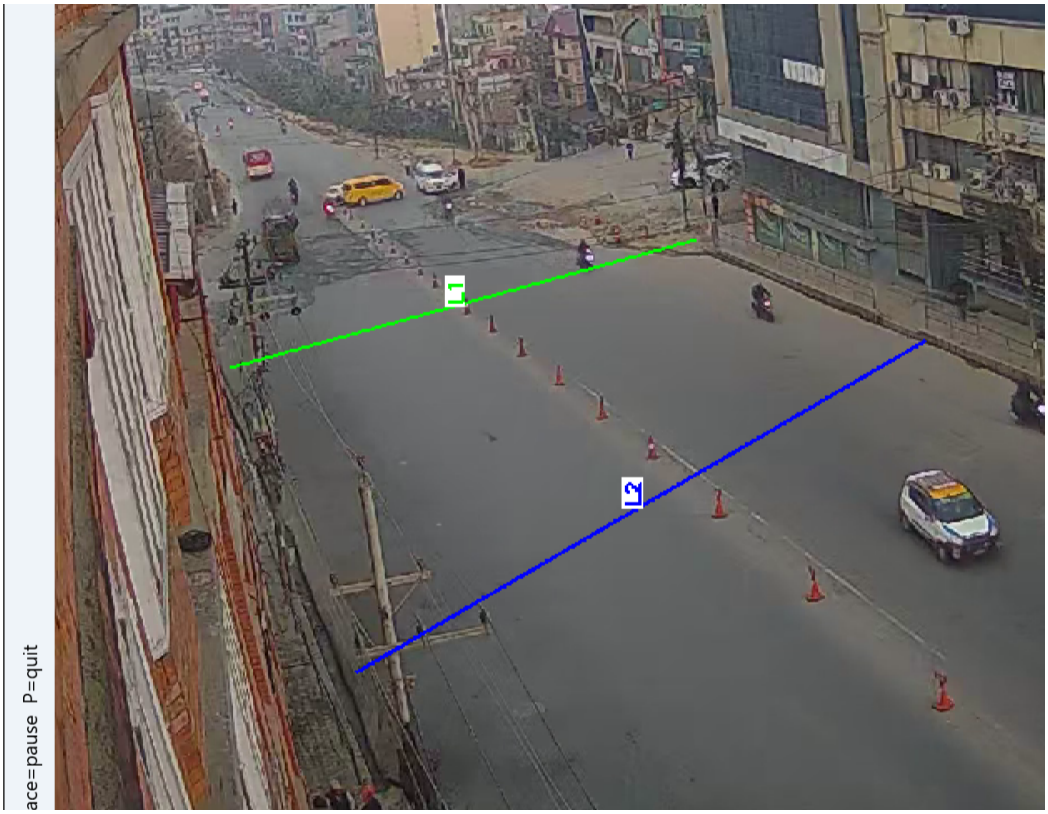
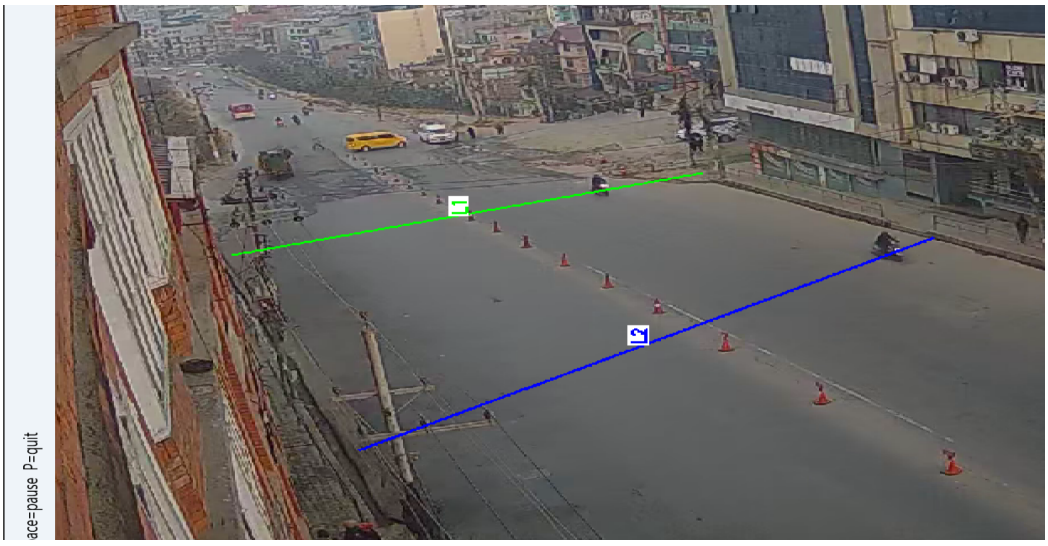


Figure 3.4: Terminal output for a sample motorcycle, showing the recorded frame indices, computed time interval, and resulting speed values.



(a) Frame at which the motorcycle crosses the first reference line (upstream reference, l_1).



(b) Frame at which the same motorcycle crosses the second reference line (downstream reference, l_2), located 20 m downstream of the first line.

Figure 3.5: Example video frames used in the manual speed estimation procedure (16 January 2026, 8:15 AM).

3.3 SUMO Microsimulation Model Development

The microsimulation environment for Narayan Gopal Chowk was developed in SUMO using the road network geometry imported from OpenStreetMap (OSM) as the geometric foundation. The OSM-derived network was subsequently refined through field verification, with lane widths, turning pocket lengths, and approach configurations adjusted to match conditions observed in the CCTV footage. Vehicle type definitions were specified for six vehicle classes grouped into three categories based on physical dimensions, as summarised in Table 3.2. Category A covers motorcycles, Category B covers cars, microbuses, and carriers, and Category C covers buses and trucks. These dimensions were used to set the length, width, and height attributes of each SUMO vehicle type, ensuring that simulated vehicles occupy realistic lateral and longitudinal space on the carriageway. Demand was introduced through approach-specific vehicle flows derived from the turning movement counts described in Section 3.2, with route fractions assigned to replicate the observed directional splits across all four approaches.

Table 3.2: Physical dimensions of vehicle types used in the SUMO micro-simulation model

Category	Vehicle Type	Length (m)	Width (m)	Height (m)
A	Motorcycle	1.90	0.70	1.20
B	Car	3.80	1.60	1.60
	Microbus	5.30	1.88	2.30
	Carrier	3.00	1.50	2.50
C	Bus	7.50	2.20	3.50
	Truck	6.50	2.40	3.20

Note. Categories A, B, and C group vehicle types with broadly similar physical dimensions for modelling and analysis purposes.

Figure 3.6 presents the close-up SUMO network view of the intersection, illustrating the coded lane geometry and the spatial arrangement of the four signalised approaches, while Figure 3.7 shows the wider network extent modelled to provide sufficient upstream and downstream road length, ensuring that boundary conditions do not artificially influence queue formation or vehicle arrival patterns at the intersection during simulation.

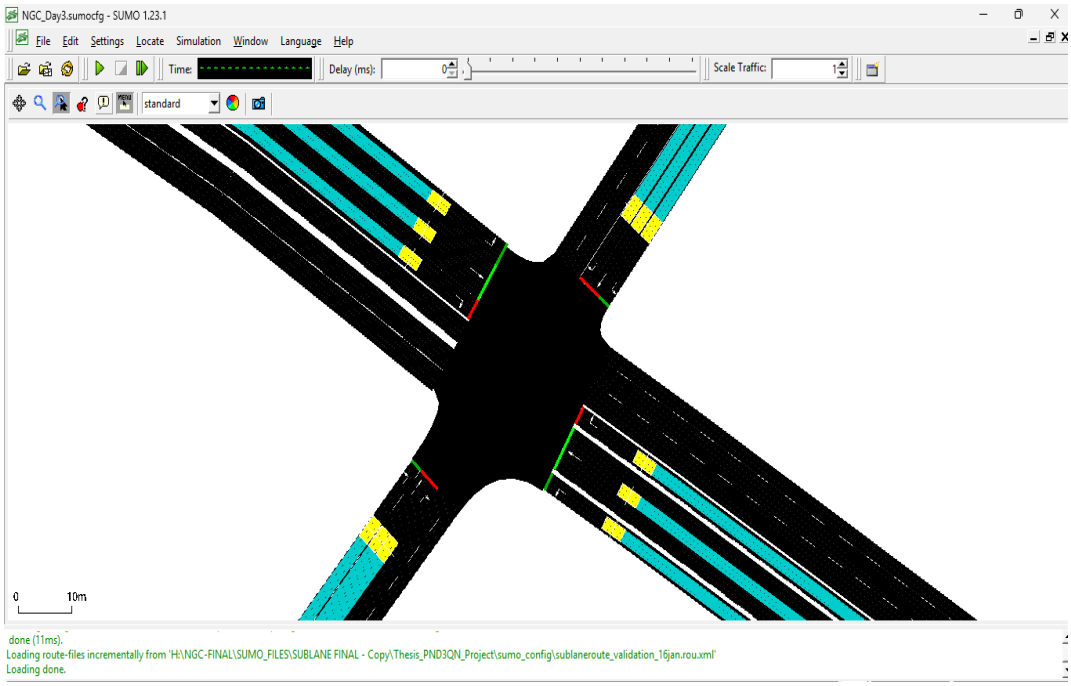


Figure 3.6: Close-up SUMO network model of Narayan Gopal Chowk showing coded lane geometry and approach configurations.

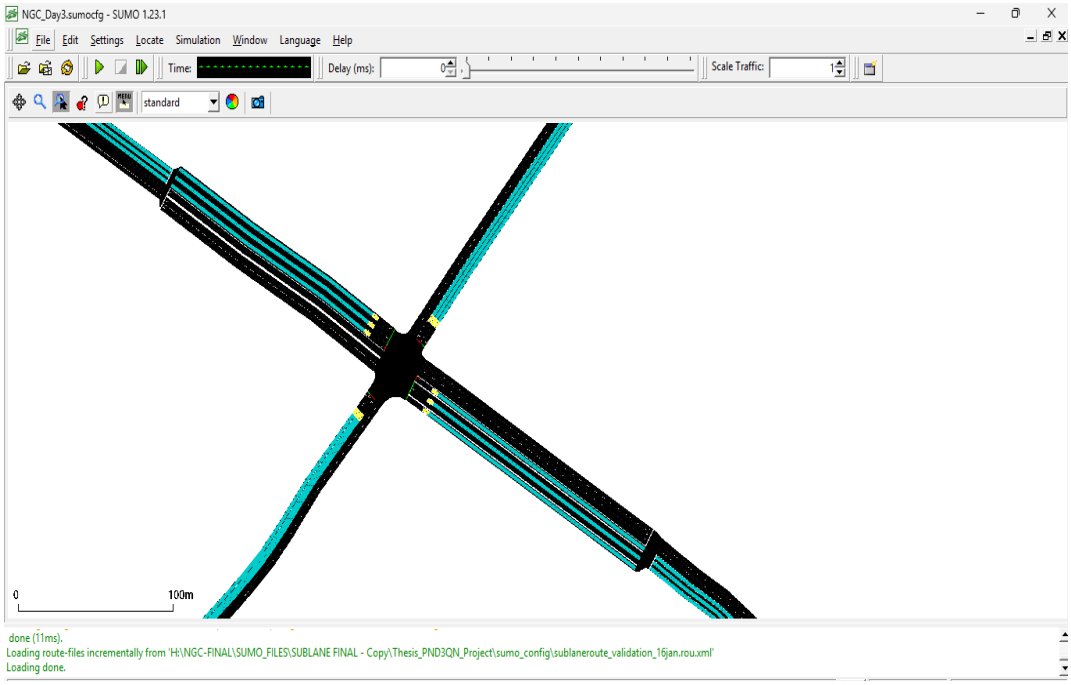


Figure 3.7: Wider SUMO network view capturing the full upstream and downstream road extents modelled to avoid boundary effects during simulation.

3.4 SPSA Calibration Framework

The fidelity of the autonomous agent's training environment is heavily dependent on the calibration of the SUMO car-following and lane-changing models. To navigate the high-dimensional parameter space, the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm was employed [22]. Calibration focused on the following microscopic variables to reflect the heterogeneous traffic conditions of Kathmandu:

- Reaction Time (τ): The desired time headway (seconds) that a following driver maintains behind the leading vehicle. Reduced below SUMO defaults to reflect the shorter following distances and faster response times observed in Kathmandu's mixed traffic, particularly among motorcycle riders.
- Driver Imperfection (σ): A dimensionless parameter $\sigma \in [0,1]$ in the Krauss car-following model that controls the magnitude of stochastic speed perturbations applied at each simulation step. Higher values produce greater speed variation and more erratic following behaviour, capturing the high within-class heterogeneity observed in Kathmandu.
- Minimum Longitudinal Gap (G_{min}): The minimum bumper-to-bumper distance (metres) that a vehicle maintains when approaching a stopped leader. Substantially reduced from SUMO defaults to reproduce the filtering and gap-closing behaviour of two-wheelers in queue.
- Minimum Lateral Gap ($minGapLat$): The minimum lateral clearance (metres) maintained during side-by-side travel. Reduced to reflect the lateral proximity accepted by motorcycles and microbuses in the absence of strict lane discipline.
- Lane Change Assertiveness ($lcAssertive$): A multiplier applied to the gap acceptance threshold during lane changes. Values greater than 1.0 indicate a willingness to accept smaller gaps, capturing the aggressive merging and weaving behaviour characteristic of Kathmandu's heterogeneous traffic.

These five parameters were calibrated independently for three vehicle groups — motorcycles (Group A), passenger vehicles (Group B: cars and microbuses), and heavy vehicles

(Group C: buses and trucks) yielding a total of $p = 15$ calibration parameters.

3.4.0.1 Parameter Definitions and Update Rule

SPSA minimises the objective function $\mathscr{L}(\theta)$ the normalised Mean Squared Prediction Error combining queue and throughput deviations via the iterative update:

$$\theta_{k+1} = \theta_k - a_k \hat{g}_k(\theta_k) \tag{3.3}$$

where each component is defined as follows.

Parameter vector θ_k: The full calibration vector at iteration k contains $p = 15$ elements arranged by group:

$$\theta_k = \left[\tau^{(0)},\, \sigma^{(0)},\, G_{min}^{(0)},\, minGapLat^{(0)},\, lcAssertive^{(0)},\, \tau^{(1)},\, \ldots,\, lcAssertive^{(2)}\right]^\top \tag{3.4}$$

where superscripts (0), (1), (2) denote motorcycles, passenger vehicles, and heavy vehicles respectively. Each element is bounded within a physically meaningful range $[\theta^{\text{lower}}, \theta^{\text{upper}}]$ determined from field observations and South Asian calibration studies [23, 28]. To prevent the optimizer from violating these bounds, SPSA is applied in a *normalised* parameter space $\tilde{\theta} \in [0,1]^{15}$:

$$\tilde{\theta}_k = \frac{\theta_k - \theta^{\text{lower}}}{\theta^{\text{upper}} - \theta^{\text{lower}}} \tag{3.5}$$

All perturbations and gradient updates are performed on $\tilde{\theta}$, which is clipped to $[0,1]$ after each step. The physical parameter vector is recovered via the inverse mapping $\theta_k = \tilde{\theta}_k \odot (\theta^{\text{upper}} - \theta^{\text{lower}}) + \theta^{\text{lower}}$ before each SUMO evaluation.

Step-size sequence a_k: The gain sequence controls the magnitude of the parameter update at each iteration:

$$a_k = \frac{a}{k^{\alpha}} \tag{3.6}$$

where $a = 0.05$ is the initial step magnitude and $\alpha = 0.602$ is the theoretically recommended decay exponent [22]. No stability constant A is employed, as the normalised parameter space $[0,1]^{15}$ constrains gradient magnitudes sufficiently to avoid divergence in early iterations. The sequence decays from $a_1 = 0.0500$ at Iteration 1 to $a_{41} = 0.0053$ at the best-performing iteration, and to $a_{60} = 0.0043$ at the final iteration, ensuring progressively finer parameter updates as the algorithm approaches convergence.

Perturbation magnitude sequence c_k: The sequence controls the size of the simultaneous random perturbation applied to all parameters at each iteration:

$$c_k = \frac{c}{k^{\gamma}} \tag{3.7}$$

where $c = 0.10$ and $\gamma = 0.101$ is the theoretically recommended decay exponent [22]. The sequence c_k must decay to zero more slowly than a_k to ensure asymptotic unbiasedness of the gradient estimate, a condition satisfied since $\gamma = 0.101 \ll \alpha = 0.602$. In practice, c_k diminishes gradually from $c_1 = 0.1000$ to $c_{60} = 0.0661$, maintaining effective gradient sensitivity throughout the full 60-iteration run.

Random perturbation vector Δ_k: At each iteration, a p-dimensional random perturbation vector $\Delta_k = [\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}]^{\top}$ is generated, where each element Δ_{ki} is independently drawn from a symmetric Bernoulli ± 1 distribution:

$$\Delta_{ki} \sim \text{Bernoulli}(\pm 1), \quad P(\Delta_{ki} = +1) = P(\Delta_{ki} = -1) = 0.5 \tag{3.8}$$

The Bernoulli distribution satisfies the theoretical requirement that $\mathrm{E}[1/\Delta_{ki}]$ exists and

equals zero in expectation, which is necessary for the gradient approximation to be asymptotically unbiased. Uniform or Gaussian perturbations do not satisfy this requirement and must not be used [22]. With $p = 15$ parameters, each Δ_k is a vector of 15 independent ± 1 values resampled at every iteration, providing new gradient directions throughout the search.

Perturbed candidate vectors θ_k^+ and θ_k^-: Two candidate parameter vectors are constructed by simultaneously perturbing all elements of the current normalised iterate in opposite directions:

$$\tilde{\theta}_k^+ = \text{clip}\big(\tilde{\theta}_k + c_k \Delta_k,\, 0,\, 1\big)\,, \qquad \tilde{\theta}_k^- = \text{clip}\big(\tilde{\theta}_k - c_k \Delta_k,\, 0,\, 1\big) \tag{3.9}$$

Each is mapped back to physical space and supplied to SUMO as a complete set of car-following and lane-changing parameters. The clipping to $[0,1]$ ensures that perturbed vectors never exceed the prescribed physical bounds. At Iteration 1 with $c_1 = 0.1000$, for example, the perturbation shifts each normalised parameter by ± 0.1000 units — corresponding to a physical shift of $\pm 0.10 \times (\theta^{\text{upper}} - \theta^{\text{lower}})$ for each parameter — providing broad initial exploration of the 15-dimensional space.

Simultaneous perturbation gradient approximation $\hat{g}_k(\theta_k)$: SUMO is executed once for $\tilde{\theta}_k^+$ and once for $\tilde{\theta}_k^-$. The stochastic gradient is approximated element-wise as:

$$\hat{g}_{ki}(\theta_k) = \frac{\mathscr{L}(\theta_k^+) - \mathscr{L}(\theta_k^-)}{2\, c_k \Delta_{ki}}, \quad i = 1, 2, \ldots, p \tag{3.10}$$

A critical advantage of SPSA is that only two SUMO simulation runs are required per iteration one for θ_k^+ and one for θ_k^- regardless of the number of parameters p. In contrast, classical finite-difference gradient methods require $2p$ simulation runs per iteration. For the $p = 15$ parameter calibration in this study, SPSA therefore requires 2 runs per iteration compared to 30 for a finite-difference approach, reducing the simulation budget by a factor of 15 and making the calibration computationally tractable.

3.4.0.2 Objective Function

The SPSA optimiser minimises a normalised mean squared error combining queue length and throughput deviations across all four intersection approaches and both random seeds. For a given parameter vector θ and a single SUMO seed, the queue loss is:

$$L_q(\theta) = \sum_{a \in \mathscr{A}} \left(\frac{Q_a^{\text{sim}}(\theta) - Q_a^{\text{field}}}{\text{normQ}_a} \right)^2 \tag{3.11}$$

where $\mathscr{A} = \{\text{Teaching, Budhanilakantha, Gaushala, Basundhara}\}$, $Q_a^{\text{sim}}(\theta)$ is the simulated average queue length (metres) for approach a, Q_a^{field} is the corresponding field-measured value, and normQ_a is the maximum field queue value for approach a (replaced by 1.0 if zero, to prevent division by zero). The throughput loss is analogously defined as:

$$L_t(\theta) = \sum_{a \in \mathscr{A}} \left(\frac{T_a^{\text{sim}}(\theta) - T_a^{\text{field}}}{\text{normT}_a} \right)^2 \tag{3.12}$$

where $T_a^{\text{sim}}(\theta)$ is the simulated total discharged vehicles on approach a, T_a^{field} is the field-measured total, and normT_a is the corresponding normalisation constant. Both loss components are dimensionless and of comparable magnitude, preventing either queue or throughput from dominating the gradient signal. The per-seed total loss is:

$$L_{\text{seed}}(\theta) = L_q(\theta) + L_t(\theta) \tag{3.13}$$

Since each candidate parameter set is evaluated under two independent SUMO random seeds ($s \in \{101, 202\}$) to reduce stochastic variability in the gradient estimate, the overall SPSA loss is the mean of the two per-seed losses:

$$\mathscr{L}(\theta) = \frac{1}{2} \sum_{s \in \{101, 202\}} L_{\text{seed}=s}(\theta) \tag{3.14}$$

It is important to note that the loss value logged to the checkpoint file at each iteration is

the approximation $\hat{\mathscr{L}}_k = \big(\mathscr{L}(\theta_k^+) + \mathscr{L}(\theta_k^-)\big)/2$, which is the mean of the two perturbed evaluations — not a direct evaluation of $\mathscr{L}$ at the current iterate θ_k. This is an intrinsic feature of the SPSA gradient approximation procedure and accounts for the residual stochastic variability in the logged loss trace even in later iterations. The two-seed averaging serves dual purposes: it reduces the influence of any single stochastic demand realisation on the gradient estimate, and it ensures that calibrated parameters generalise across different random demand patterns rather than overfitting to a single seed's queue dynamics. Convergence was assessed using a five-iteration rolling average of this approximate loss, which is reported alongside individual iteration losses in Table 4.1.

3.4.0.3 Optimisation Procedure and Acceptance Criteria

The calibration workflow proceeds as follows. Aggregate one-hour field data from 11 and 13 January are used to define parameter bounds and initialise the parameter vector θ_0. At each iteration k, step sizes a_k and c_k are computed from Equations 3.6 and 3.7, a Bernoulli perturbation vector $\Delta_k \in \{-1,+1\}^{15}$ is sampled, and two candidate physical parameter vectors θ_k^+ and θ_k^- are constructed per Equation 3.9. SUMO is executed twice for each candidate (once per seed), intersection flows and queue lengths are extracted, and the losses $\mathscr{L}(\theta_k^+)$ and $\mathscr{L}(\theta_k^-)$ are evaluated. The gradient is approximated via Equation 3.10 and the normalised parameter vector is updated via Equation 3.3 and clipped to $[0,1]^{15}$. This loop runs for 60 iterations, after which the checkpoint record with the minimum logged loss is selected as θ_{opt}. The full iteration history of the approximate loss, step-size sequences, and representative calibrated parameters is reported in Table 4.1.

In the validation phase, θ_{opt} is applied to an independent validation day (16 January). A SUMO run with warm-up period is performed, and approach flows and queue lengths are extracted to compute the validation metrics. Calibration is accepted if two criteria are simultaneously satisfied: GEH < 5.0 for all approach flows, and absolute queue length deviations within $\pm 20\%$. If either criterion fails, the network configuration, field data, or parameter bounds are reviewed and the calibration loop is restarted. The procedural logic of the full SPSA optimisation and validation workflow is illustrated in Figure 3.8.

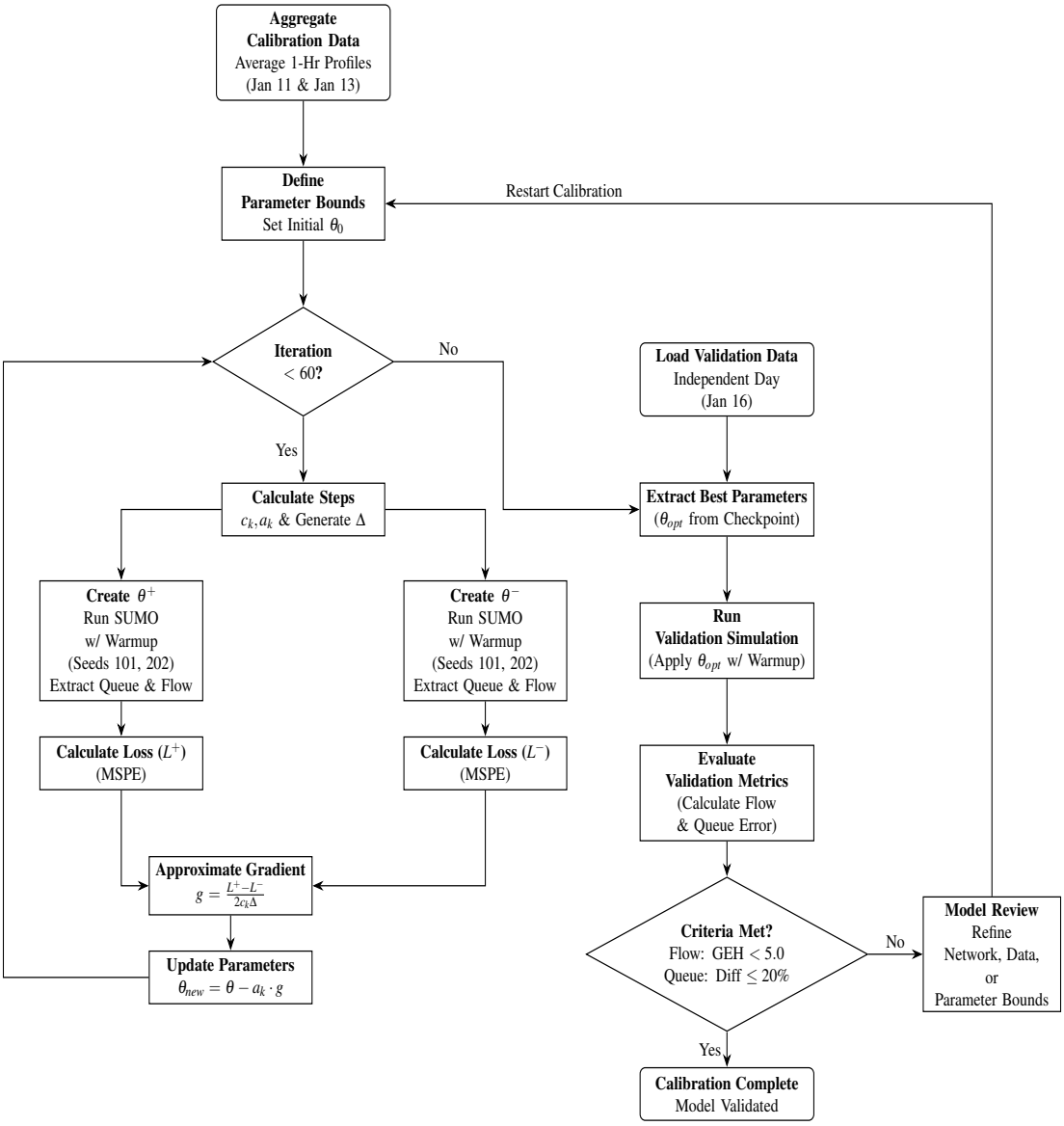


Figure 3.8: Procedural framework for the SPSA calibration and validation process.

3.5 Statistical Validation Metrics

To quantify the "Goodness of Fit," the calibrated model was validated against independent field-observed data (January 16). The GEH statistic was employed to compare macroscopic simulated flow (q_{sim}) against observed flow (q_{obs}).

$$GEH = \sqrt{\frac{2(q_{sim} - q_{obs})^2}{q_{sim} + q_{obs}}} \tag{3.15}$$

The spatial accuracy of the micro-simulation was further verified using Root Mean Square Error (RMSE) to evaluate the deviation between observed and simulated queue lengths.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Q_{obs,i} - Q_{sim,i})^2} \tag{3.16}$$

3.6 Formulation of the Reinforcement Learning Environment

Having established and statistically validated the microscopic simulation model to accurately reflect the field conditions, this model is subsequently utilized as the interactive plant for the reinforcement learning agent. The traffic signal control problem is mathematically formulated as a Markov Decision Process (MDP), which requires the explicit definition of the state space, action space, and reward function.

3.6.1 State Space Representation

The agent's observation at each decision step consists of two components. First, a spatial tensor $\mathbf{S}_{\text{spatial}} \in \mathbb{R}^{2\times 8\times 50}$ encodes the instantaneous traffic state across the eight monitored lane-rows (lanes 1 and 2 of all four approaches; lane 0, which carries negligible flow, is excluded). Each lane-row is discretized into 50 uniform cells of 7 m, providing a sensing horizon of 350 m from the stop-line. For Basundhara and Gaushala, whose physical approach length exceeds 162 m, an upstream edge is stitched to the stop-line edge so that vehicles beyond 162 m remain visible within the tensor. The two channels are defined as:

- Channel 0 — Density $D[r,c] = \min\left(1,\ \dfrac{\text{count}[r,c]}{d_{\max}}\right)$, where $d_{\max} = 4.0$ vehicles per cell (worst-case sublane motorcycle packing over a 7 m cell).
- Channel 1 — Velocity $V[r,c] = \max_{\text{veh}\in(r,c)} \min\left(1,\ \dfrac{v_{\text{veh}}}{v_{\max}}\right)$, normalised by the speed ceiling $v_{\max} = 15.0$ m/s.

Both channels are stored as `float16` for RAM efficiency and cast to `float32` before the CNN forward pass. Second, the active green phase is encoded as a 4-element one-hot vector $\mathbf{S}_{\text{phase}} \in \{0,1\}^4$, one entry per green phase in the fixed cyclic sequence $[0, 2, 4, 6]$.

3.6.2 Action Space Formulation

At the start of each green phase the agent selects one action $a \in A = \{0, 1, \ldots, 14\}$ (15 discrete actions). The action maps directly to a green duration chosen from the set

$$G = \{15, 20, 25, \ldots, 85\} \text{ s}, \qquad |G| = 15, \tag{3.17}$$

so that $T_{\text{green}} = G[a]$. A fixed yellow interval of $T_{\text{yellow}} = 5$ s follows every green phase and is not agent-controlled. The phase sequence is cyclic and fixed: $0 \to 2 \to 4 \to 6 \to 0$.

3.6.3 Reward Function

The reward is computed once per completed green-plus-yellow cycle using two traffic metrics aggregated over all monitored lanes: cumulative waiting time W (vehicle-seconds; sum of per-lane waiting times) and halted vehicle count Q (number of vehicles with speed < 0.1 m/s). The reward is defined as:

$$R_t = \text{clip}\left(-\alpha \frac{\Delta W}{W_{\max}} - \beta \frac{\Delta Q}{Q_{\max}},\ -1,\ +1\right) \tag{3.18}$$

where $\Delta W = W_{\text{end}} - W_{\text{start}}$ and $\Delta Q = Q_{\text{end}} - Q_{\text{start}}$ are the changes observed over the phase. The normalisation constants $W_{\max} = 9{,}400$ vehicle-seconds and $Q_{\max} = 100$ vehicles were profiled at the 95th percentile of the calibrated baseline simulation. The weighting

coefficients $\alpha = 0.6$ and $\beta = 0.4$ were selected from a grid search over the candidate pairs:

$$(\alpha, \beta) \in \{(0.5, 0.5),\ (0.6, 0.4),\ (0.7, 0.3),\ (0.8, 0.2),\ (0.9, 0.1)\}.$$

Clipping the raw signal to $[-1, +1]$ prevents gradient explosion and keeps the loss scale uniform across all demand levels.

3.7 Proposed PN-D3QN Agent Architecture

With the MDP framework defined, the underlying decision-making architecture of the autonomous agent must be established. The proposed PN-D3QN agent is built upon an optimised variant of the Deep Q-Network (DQN) algorithm, integrating a Dueling Network Architecture, Double Q-Learning, Prioritized Experience Replay (PER), and Noisy Network Exploration to enhance estimation accuracy, training stability, and sample efficiency. The agent is implemented entirely on a CPU-only system using a batch size of 64, a PER buffer capacity of 50,000 transitions, and a learning rate of 3×10^{-4}. The comprehensive architecture of the network is illustrated in Figure 3.9.

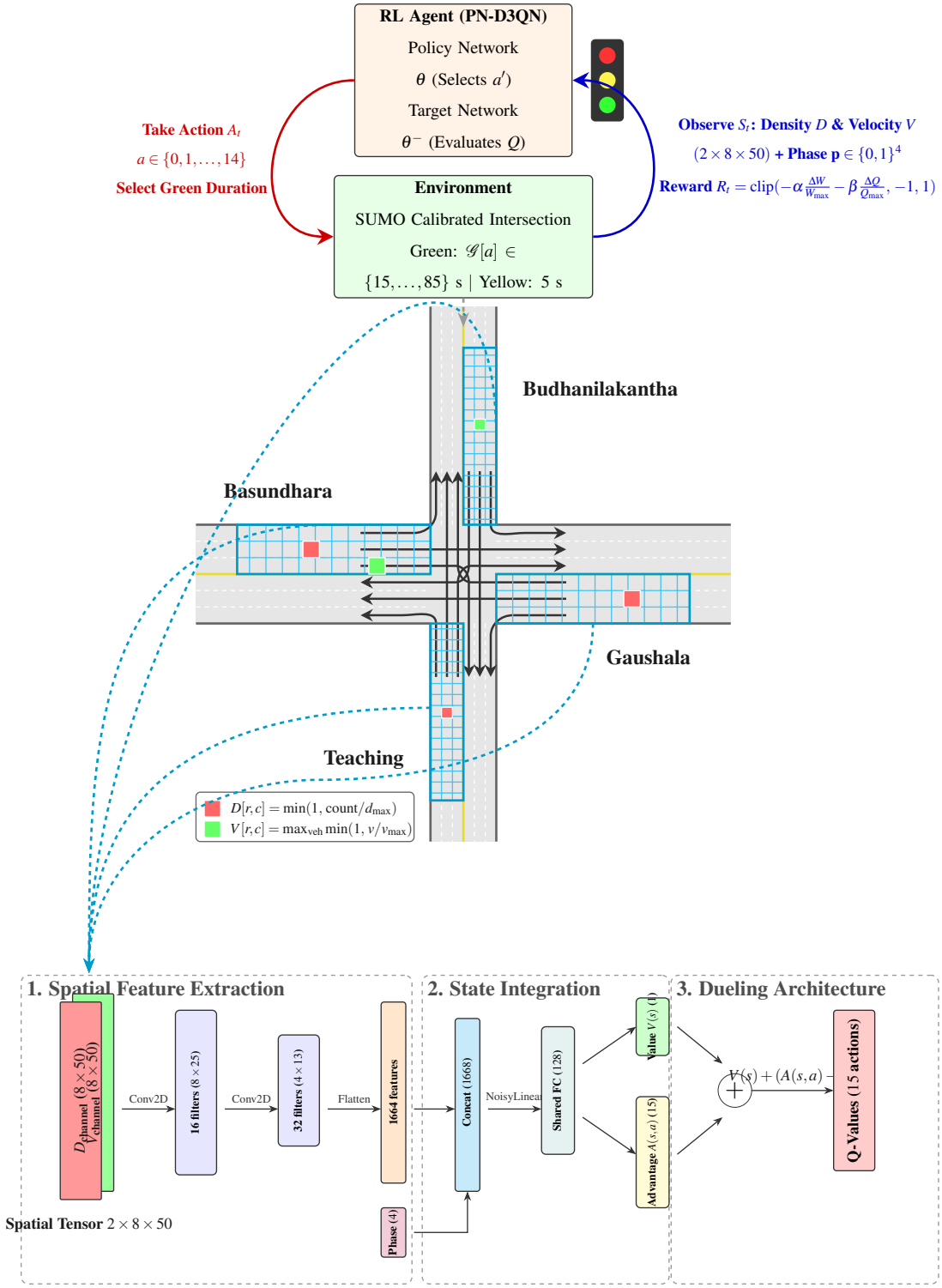


Figure 3.9: Integrated PN-D3QN framework: RL cycle (top), spatial state mapping at Narayan Gopal Chowk (middle), and Dueling Q-Network architecture (bottom).

3.7.1 Deep Double Dueling Q-Network (D3QN)

The D3QN backbone combines two structural improvements over vanilla DQN: the Dueling Network Architecture for more robust Q-value decomposition and Double Q-Learning to reduce overestimation bias during parameter updates.

The Dueling architecture bifurcates the neural network's fully connected layers into two separate streams: a Value Stream $V(s)$ and an Advantage Stream $A(s,a)$. These are aggregated to compute the final Q-value:

$$Q(s,a;\theta) = V(s;\theta) + \left(A(s,a;\theta) - \frac{1}{|\mathscr{A}|} \sum_{a' \in \mathscr{A}} A(s,a';\theta) \right) \tag{3.19}$$

where $|\mathscr{A}| = 15$, θ denotes the full set of network parameters shared across the CNN, value, and advantage streams. Subtracting the mean advantage enforces identifiability of $V(s)$ and $A(s,a)$.

To mitigate overestimation bias, Double Q-Learning decouples action selection and evaluation. The policy network with parameters θ selects the greedy next action, while the frozen target network with parameters θ^- evaluates its value:

$$Y_t = R_t + \gamma \cdot Q_{\theta^-}\left(s_{t+1},\ \arg\max_a Q_\theta(s_{t+1},a) \right)(1 - D_t) \tag{3.20}$$

where R_t is the reward, $\gamma = 0.95$ is the discount factor, and D_t is the terminal flag. The target network is updated by a hard copy $\theta^- \leftarrow \theta$ every 1,000 optimisation steps.

3.7.2 Neural Network Architecture

The network processes a $(2 \times 8 \times 50)$ spatial tensor (Channel 0: Density D; Channel 1: Velocity V) and a 4-dimensional one-hot phase vector. The CNN feature extractor uses two convolutional layers:

- Conv1: 16 filters, kernel (3×5), stride $(1, 2)$, padding $(1, 2)$ $\Rightarrow$ output $(16 \times 8 \times 25)$.
- Conv2: 32 filters, kernel (3×3), stride $(2, 2)$, padding $(1, 1)$ $\Rightarrow$ output $(32 \times 4 \times 13)$.

The flattened CNN output ($32 \times 4 \times 13 = 1{,}664$ features) is concatenated with the 4-dimensional phase vector to form a 1,668-dimensional shared representation. This passes through a shared NOISYLINEAR fully-connected layer with 128 units (ReLU activation), which then branches into:

- Value stream: NOISYLINEAR$(128 \to 1)$, output $V(s)$.
- Advantage stream: NOISYLINEAR$(128 \to 15)$, output $A(s,a)$.

The final Q-values for all 15 actions are obtained via the Dueling aggregation in Equation 3.19. A compact shared layer of 128 units is used to reduce CPU load while preserving representational capacity.

3.7.3 Prioritized Experience Replay (PER)

Prioritized Experience Replay (PER) assigns a sampling probability to each transition based on its Temporal Difference (TD) error to accelerate convergence. The priority of a stored transition i is defined as:

$$p_i = |\delta_i| + \varepsilon, \qquad |\delta_i| = |Y_t - Q_\theta(s_t, a_t)| \tag{3.21}$$

with $\varepsilon = 10^{-5}$ to prevent zero probabilities. The probability of sampling transition i is:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{3.22}$$

where $\alpha = 0.6$ controls the degree of prioritisation. A SumTree data structure of capacity 50,000 stores the priorities and transitions, enabling $\mathcal{O}(\log N)$ sampling and updates.

To correct for the sampling bias introduced by PER, importance-sampling (IS) weights are applied:

$$w_i = \left(\frac{1}{N \cdot P(i)}\right)^\beta \tag{3.23}$$

where N is the current number of transitions in the buffer. The exponent β is annealed from 0.4 towards 1.0 with an increment of 2×10^{-5} per sampling step, gradually restoring

unbiased updates as training progresses. The IS weights are normalised by the maximum weight in the batch and used to re-weight the element-wise Smooth-L1 loss during backpropagation. Newly added transitions are inserted with the current maximum priority to ensure they are sampled at least once before their TD error is known.

3.7.4 Noisy Networks for State-Dependent Exploration

To overcome the limitations of ε-greedy randomness, the PN-D3QN agent replaces ε-greedy exploration with Noisy Networks. Parametric noise is injected directly into the weights and biases of all fully connected layers via NOISYLINEAR units, enabling the agent to learn a state-dependent exploratory policy.

Each deterministic weight w is replaced by a stochastic weight:

$$w = \mu_w + \sigma_w \odot \varepsilon_w, \qquad b = \mu_b + \sigma_b \odot \varepsilon_b \tag{3.24}$$

where μ and σ are learnable parameters initialised to $\sigma_0 = 0.5/\sqrt{n_{\text{in}}}$, and ε_w, ε_b are factorised Gaussian noise samples generated using a signed square-root transform:

$$f(x) = \text{sgn}(x)\sqrt{|x|}, \qquad x \sim \mathcal{N}(0,1) \tag{3.25}$$

This factorised construction reduces the number of random variables required per layer from $pq+q$ to $p+q$ and is well suited to CPU-only training. During training, noise in all NOISYLINEAR layers is reset before each forward pass, ensuring different stochastic perturbations at every decision step. During evaluation, the network runs in deterministic mode (noise disabled), yielding a purely greedy policy based on the learned means μ.

3.7.5 Training Procedure and Algorithmic Flow

Each of the 400 training episodes corresponds to one simulated hour ($T_{\text{ep}} = 3600$ s) at Narayan Gopal Chowk, with a simulation time-step of $\Delta t = 0.5$ s. The first 300 s of every episode run under the built-in fixed-cycle program as a warm-up to pre-fill the intersection before the agent assumes full control. Each episode uses a unique random seed ($42+\text{ep}$) for

stochastic diversity while remaining reproducible, and full checkpoints (weights, optimiser state, and replay buffer) are saved after every episode for power-cut-safe resumption.

The training workflow, illustrated in Figure 3.10, proceeds in two interlocked phases. In Phase 1 (Environment Interaction Loop), at the start of each green phase the agent observes state S_t — a $(2 \times 8 \times 50)$ spatial tensor of normalised density and velocity values concatenated with a 4-element one-hot phase vector p_t and selects a green duration $\mathscr{G}[a] \in \{15, 20, \ldots, 85\}$ s through its NoisyLinear heads. SUMO executes the chosen green phase followed by a fixed 5-s yellow interval, after which the reward R_t is computed from normalised changes in cumulative waiting time and halted vehicle count, clipped to $[-1, +1]$. The full transition tuple $(S_t, p_t, A_t, R_t, S_{t+1}, p_{t+1}, D_t)$ is then pushed into the SumTree-backed prioritised replay buffer.

In Phase 2 (PN-D3QN Optimisation), once the buffer holds at least 1,000 transitions, a mini-batch of 64 is sampled at every phase decision using PER (α=0.6, β annealed from 0.4 to 1.0 with increment 2×10^{-5}). Double DQN target values are formed by having the policy network θ select the greedy next action a' and the frozen target network θ^- evaluate it:

$$Y_t = R_t + \gamma Q_{\theta^-}(S_{t+1}, a')(1 - D_t), \quad \gamma = 0.95. \tag{3.26}$$

The importance-sampling-weighted Smooth-L1 loss is minimised by the Adam optimiser (lr=3×10^{-4}) with gradient clipping ($\|\nabla\| \leq 10$), and TD errors are fed back to update leaf priorities in the SumTree. The target network is synchronised by a hard copy $\theta^- \leftarrow \theta$ every 1,000 optimisation steps. The key hyperparameters are summarised in Table 3.3.

Table 3.3: PN-D3QN training hyperparameters

Parameter	**Value**
Episodes	331
Episode length	3600 s
Warm-up duration	300 s
Simulation time-step	0.5 s
Yellow duration (fixed)	5 s
Action space size	15
Green duration range	15–85 s (step 5 s)
Batch size	64
Replay buffer capacity	50,000
Learning starts	1,000 transitions
Learning rate	3×10^{-4}
Discount factor γ	0.95
Target update frequency	1,000 optimisation steps
Gradient clipping	$\|\nabla\| \leq 10$
PER exponent α	0.6
PER β (initial $\rightarrow$ final)	$0.4 \rightarrow 1.0$
PER β increment	2×10^{-5}
NoisyLinear σ_0	0.5
Reward weight α	0.6
Reward weight β	0.4
$W_{\max}$ (waiting time normaliser)	9,400 vehicle-seconds
$Q_{\max}$ (queue normaliser)	100 vehicles

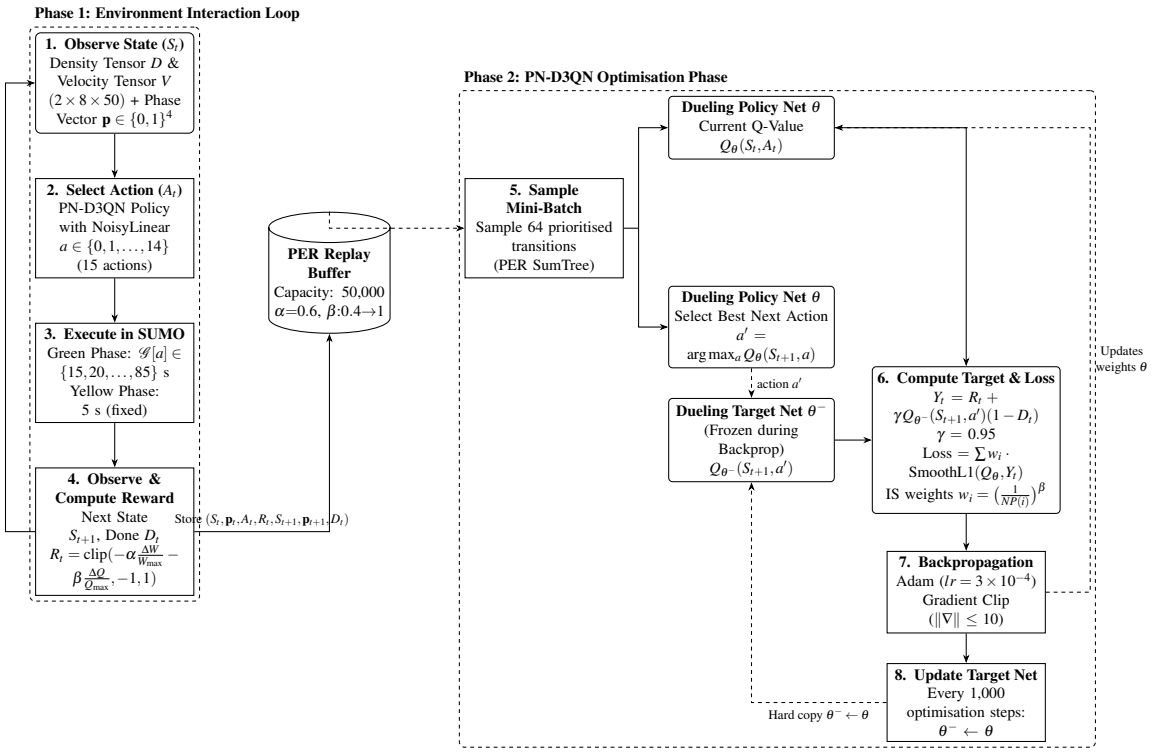


Figure 3.10: PN-D3QN training workflow: Phase 1 collects transitions into the prioritised replay buffer; Phase 2 performs Double-DQN updates with IS-weighted Smooth-L1 loss and periodic target-network synchronisation.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Calibration Results

The foundation of the reinforcement learning environment relies on the accuracy of the microsimulation. This section details the outcomes of the SPSA parameter optimisation, the resulting vehicle-specific behavioural profiles, and the subsequent manual refinement applied to correct physically implausible artefacts in the heavy vehicle class.

4.1.1 Optimisation Convergence

The SPSA algorithm systematically minimised the normalised Mean Squared Prediction Error (MSPE) over 60 iterations, with each iteration requiring four SUMO simulation executions (two perturbed candidate vectors θ_k^+ and θ_k^-, each evaluated under two random seeds). The step-size sequences a_k and c_k, the approximate loss $\hat{\mathscr{L}}_k = (\mathscr{L}(\theta_k^+) + \mathscr{L}(\theta_k^-))/2$, its five-iteration rolling average, and representative calibrated parameters at each iteration are reported in Table 4.1.

As depicted in Figure 4.1, the optimisation converged through four behavioural phases. During the *Exploration* phase (Iterations 1–11), the loss was high and volatile (range: 1.18–4.70), driven by the large initial step size $a_1 = 0.0500$ and perturbation magnitude $c_1 = 0.1000$, which encouraged broad sampling of the 15-dimensional normalised parameter space. The peak loss of 4.6994 at Iteration 11 is characteristic of SPSA's early gradient overshoot, where the algorithm temporarily moves parameters away from a better region before corrective updates accumulate. During the Transition phase (Iterations 12–14), the declining a_k sequence (0.0112–0.0102) allowed gradient accumulation to push the car-following parameters toward physically meaningful ranges notably, $G_{\min}^{\text{Car}}$ dropped from $\sim$2.5 m toward $\sim$1.1 m. A sharp phase break occurred at Iteration 15, where the loss fell from 2.46 to 0.96, marking the onset of the Convergence phase (Iterations 15–30), during which the five-iteration rolling average stabilised in the 0.87–1.09 band and a_k decayed

from 0.0098 to 0.0065. The Refinement phase (Iterations 31–60) performed fine-grained local search as a_k continued to decay toward 0.0043 at Iteration 60, while c_k declined slowly from 0.0707 to 0.0661, maintaining effective gradient sensitivity throughout. The global minimum loss of 0.3772 was achieved at Iteration 41 ($a_{41} = 0.0053$, $c_{41} = 0.0687$), which was selected as the optimal parameter set θ_{opt}. Post-optimum iterations remained in the 0.41–1.13 band, confirming that the Iteration 41 minimum is a genuine optimum rather than a transient noise spike.

It is important to note that the loss value logged at each iteration is the approximation $\hat{\mathscr{L}}_k = (\mathscr{L}(\theta_k^+) + \mathscr{L}(\theta_k^-))/2$ the mean of the two perturbed evaluations rather than a direct evaluation at the current iterate θ_k. This is an intrinsic feature of the SPSA gradient approximation procedure and accounts for the residual stochastic variability observed in the logged loss trace even in later iterations.

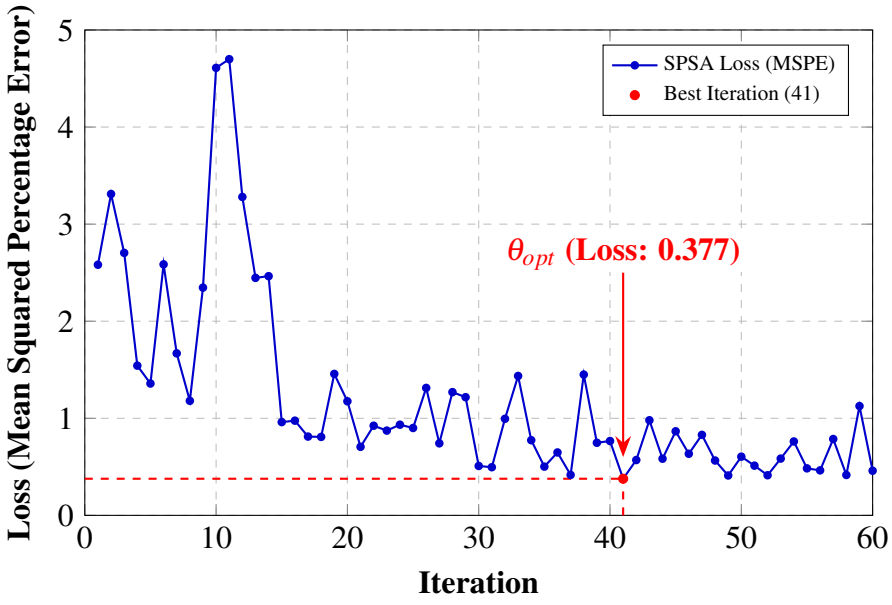


Figure 4.1: SPSA convergence graph displaying the minimisation of MSPE loss over 60 iterations. The five-iteration rolling average (dashed) confirms systematic convergence. The global minimum of 0.3772 is achieved at Iteration 41, selected as θ_{opt}.

Table 4.1: SPSA Iteration Summary

Iter	a_k	c_k	**Loss**	τ **(A)** **(s)**	τ **(B)** **(s)**	$G_{\min}$ **(B)** **(m)**	$G_{\min}$ **(C)** **(m)**
Phase 1 — Exploration (Iter. 1–11): large perturbations, high variance							
1	0.0500	0.1000	2.5816	0.793	1.191	1.512	2.482
2	0.0329	0.0932	3.3103	0.829	1.148	1.451	2.572
3	0.0258	0.0895	2.7038	0.500	0.800	2.282	1.500
4	0.0217	0.0869	1.5420	0.648	0.800	2.031	1.869
5	0.0190	0.0850	1.3580	0.893	1.095	2.449	2.484
6	0.0170	0.0834	2.5854	0.716	0.882	2.500	2.040
7	0.0155	0.0822	1.6690	0.758	0.832	2.500	1.936
8	0.0143	0.0811	1.1799	0.565	1.063	2.172	2.418
9	0.0133	0.0801	2.3460	0.825	1.376	1.730	3.068
10	0.0125	0.0793	4.6100	0.947	1.229	1.937	2.763
11	0.0118	0.0785	4.6994	0.878	1.147	1.820	2.935
Phase 2 — Transition (Iter. 12–14): loss breaking below 3.0							
12	0.0112	0.0778	3.2800	0.973	1.032	1.658	3.174
13	0.0107	0.0772	2.4465	0.873	0.912	1.488	3.424
14	0.0102	0.0766	2.4639	0.661	0.800	1.128	2.895
Phase 3 — Convergence (Iter. 15–30): loss stabilises below 1.5							
15	0.0098	0.0761	0.9606	0.723	0.875	1.022	3.050
16	0.0094	0.0756	0.9755	0.708	0.857	0.997	3.012
17	0.0091	0.0751	0.8115	0.686	0.830	1.034	2.957
18	0.0088	0.0747	0.8081	0.710	0.801	1.075	3.018
19	0.0085	0.0743	1.4571	0.700	0.813	1.058	2.992

Continued on next page

Table 4.1 continued from previous page

Iter	a_k	c_k	**Loss**	τ **(A)** **(s)**	τ **(B)** **(s)**	$G_{\min}$ **(B)** **(m)**	$G_{\min}$ **(C)** **(m)**
20	0.0082	0.0739	1.1752	0.724	0.841	1.098	3.051
21	0.0080	0.0735	0.7060	0.723	0.842	1.097	3.052
22	0.0078	0.0732	0.9234	0.723	0.843	1.096	3.050
23	0.0076	0.0729	0.8732	0.737	0.825	1.071	3.086
24	0.0074	0.0725	0.9340	0.710	0.800	1.117	3.154
25	0.0072	0.0722	0.9001	0.722	0.800	1.138	3.185
26	0.0070	0.0720	1.3130	0.746	0.800	1.098	3.126
27	0.0069	0.0717	0.7421	0.713	0.839	1.042	3.044
28	0.0067	0.0714	1.2694	0.655	0.910	1.142	2.898
29	0.0066	0.0712	1.2174	0.680	0.940	1.185	2.962
30	0.0065	0.0709	0.5088	0.668	0.926	1.206	2.931
Phase 4 — Refinement (Iter. 31–60): global minimum at Iter. 41							
31	0.0063	0.0707	0.4959	0.650	0.904	1.176	2.976
32	0.0062	0.0705	0.9948	0.616	0.863	1.118	3.061
33	0.0061	0.0702	1.4360	0.704	0.800	0.967	3.282
34	0.0060	0.0700	0.7744	0.711	0.808	0.979	3.299
35	0.0059	0.0698	0.5021	0.729	0.829	0.949	3.255
36	0.0058	0.0696	0.6479	0.721	0.839	0.962	3.236
37	0.0057	0.0694	0.4165	0.721	0.839	0.962	3.236
38	0.0056	0.0693	1.4501	0.643	0.800	0.830	3.041
39	0.0055	0.0691	0.7478	0.655	0.814	0.849	3.070
40	0.0054	0.0689	0.7659	0.669	0.800	0.824	3.106
41	0.0053	0.0687	0.3772	0.669	0.800	0.825	3.106
42	0.0053	0.0686	0.5700	0.649	0.824	0.800	3.055

Continued on next page

Table 4.1 continued from previous page

Iter	a_k	c_k	**Loss**	τ **(A)** **(s)**	τ **(B)** **(s)**	$G_{\min}$ **(B)** **(m)**	$G_{\min}$ **(C)** **(m)**
43	0.0052	0.0684	0.9797	0.670	0.800	0.837	3.002
44	0.0051	0.0682	0.5836	0.685	0.800	0.811	3.039
45	0.0051	0.0681	0.8660	0.707	0.800	0.848	3.094
46	0.0050	0.0679	0.6345	0.702	0.807	0.839	3.108
47	0.0049	0.0678	0.8292	0.701	0.808	0.838	3.106
48	0.0049	0.0676	0.5649	0.707	0.801	0.828	3.120
49	0.0048	0.0675	0.4108	0.698	0.811	0.842	3.099
50	0.0047	0.0674	0.6046	0.675	0.839	0.881	3.157
51	0.0047	0.0672	0.5120	0.675	0.838	0.883	3.155
52	0.0046	0.0671	0.4132	0.672	0.835	0.886	3.161
53	0.0046	0.0670	0.5849	0.675	0.832	0.891	3.154
54	0.0045	0.0668	0.7611	0.676	0.831	0.890	3.156
55	0.0045	0.0667	0.4841	0.694	0.809	0.920	3.112
56	0.0044	0.0666	0.4634	0.690	0.805	0.927	3.122
57	0.0044	0.0665	0.7864	0.690	0.804	0.926	3.121
58	0.0043	0.0664	0.4160	0.695	0.809	0.933	3.132
59	0.0043	0.0662	1.1258	0.725	0.800	0.984	3.057
60	0.0043	0.0661	0.4595	0.721	0.805	0.991	3.047

$\star$ Iteration 41: global minimum loss $= 0.3772$, selected as θ_{opt}.

Only four of 15 calibrated parameters shown; full set in Table 4.2.

4.1.2 Parameter Sensitivity Analysis

To understand the operational impact of the calibrated variables, a Pearson correlation sensitivity analysis was conducted (Figure 4.2). The analysis reveals that the simulation error is highly sensitive to car headway time (τ) and car minimum longitudinal gap, indi-

cating that passenger car spacing dictates the macroscopic throughput of the intersection. Conversely, increases in heavy-vehicle minimum gaps exhibit a negative correlation with the loss ($r \approx -0.41$), suggesting that enforcing larger gaps for trucks and buses marginally reduced the composite queue–throughput error under the calibration dataset a consequence of the low heavy-vehicle fleet proportion (below 3.4%) rather than a genuine behavioural relationship, as discussed in Section 4.1.4.

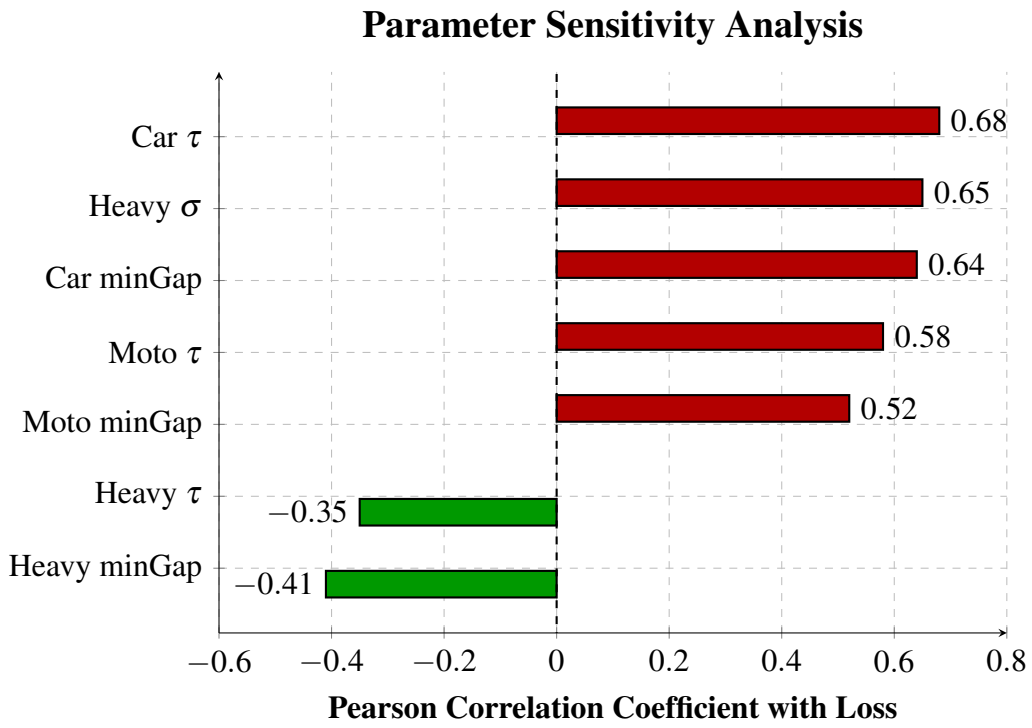


Figure 4.2: Parameter sensitivity analysis: Pearson correlation coefficients of the 15 calibrated behavioural variables against simulation loss, highlighting the dominant influence of car-class spacing parameters.

4.1.3 Evolution of Behavioural Parameters

The trajectories of the behavioural parameters across the SPSA iterations reveal distinct driving styles for different vehicle classes at Narayan Gopal Chowk. The final calibrated values extracted at Iteration 41 are summarised in Table 4.2.

Table 4.2: Raw SPSA optimised microscopic parameters (θ_{opt}) extracted at Iteration 41. Heavy vehicle G_{min} and `minGapLat` values (shaded) are subject to manual correction in Section 4.1.4.

Parameter	**Category**		
	A	**B**	**C**
Headway time (τ)	0.67 s	0.80 s	1.38 s
Minimum longitudinal gap (G_{min})	0.22 m	0.82 m	3.11 m
Minimum lateral gap (`minGapLat`)	0.11 m	0.30 m	1.33 m
Lane-change assertiveness	4.24	1.11	1.62
Driver imperfection (σ)	0.53	0.22	0.22

The evolutionary trajectories for headway time and minimum longitudinal gap are illustrated in Figures 4.3 and 4.4, respectively.

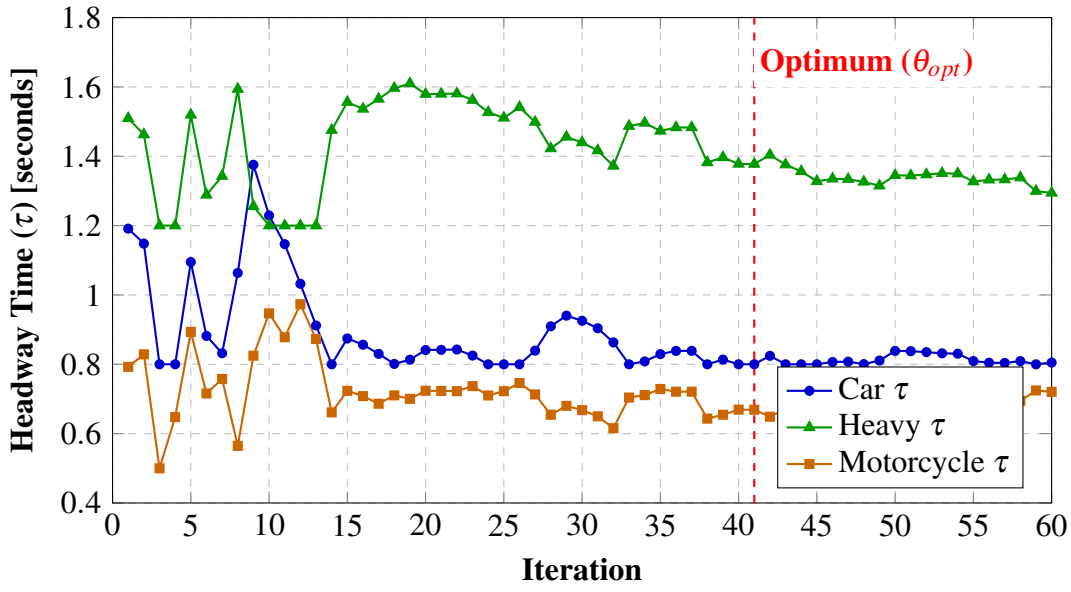


Figure 4.3: Evolutionary trajectory of vehicle-specific headway time (τ) during SPSA calibration.

The calibration accurately captures the chaotic filtering behaviour of two-wheelers: motorcycles converge to the lowest headway (0.67 s) and lateral gaps (0.11 m), paired with highly aggressive lane-change assertiveness (4.24) and a high imperfection factor ($\sigma = 0.53$). Passenger cars exhibit intermediate spacing and moderate lane-change assertiveness, reflecting typical mixed-flow behaviour at the intersection. In contrast, the raw SPSA optimum for the heavy-vehicle class yields a minimum longitudinal gap of 3.11 m and a minimum lateral gap of 1.33 m, which are physically implausible for bus and truck queues

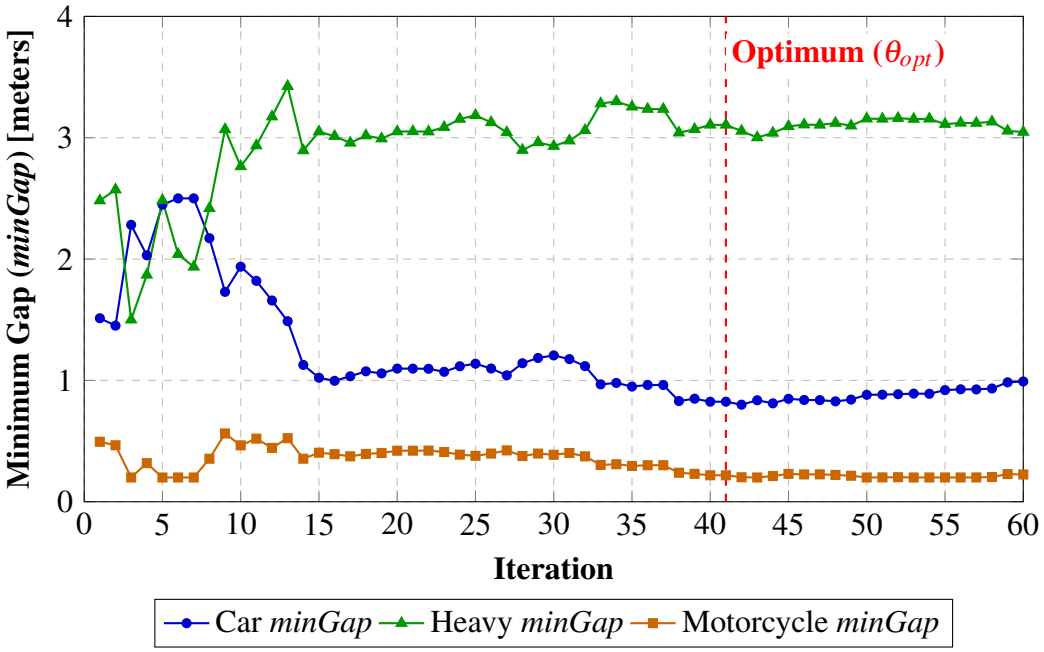


Figure 4.4: Evolutionary trajectory of vehicle-specific minimum longitudinal gap ($G_{\min}$) during SPSA calibration.

and require post-hoc correction, as detailed in Section 4.1.4.

4.1.4 Manual Refinement of Heavy Vehicle Parameters

While the SPSA algorithm demonstrated strong convergence for the motorcycle and car classes with all five behavioural parameters accepted without adjustment the optimised parameters for the heavy-vehicle class (bus and truck) produced a physically implausible minimum longitudinal gap of $G_{\min} = 3.11$ m and minimum lateral gap of 1.33 m. At a signalised intersection under queue conditions, these values would require buses and trucks to maintain over three metres of empty space between consecutive vehicles at a red signal, a condition inconsistent with field observations at Narayan Gopal Chowk.

The root cause of this artefact lies in the sensitivity structure of the SPSA loss function. As confirmed by the Pearson correlation analysis (Figure 4.2), the heavy-vehicle $G_{\min}$ carries a negative correlation of approximately -0.41 with the MSPE loss. This implies that enforcing wider heavy-vehicle gaps marginally reduced the composite queue–throughput error during the calibration days (11 and 13 January 2025), an outcome attributable to the extremely low proportion of heavy vehicles in the fleet (below 3.4% by composition). With

such low representation, the SPSA gradient signal for this class was too weak the finite-difference term $(\mathscr{L}(\theta_k^+) - \mathscr{L}(\theta_k^-))/2c_k\Delta_{ki}$ was dominated by motorcycle and car parameters and the algorithm converged to an artefact value that minimised the mathematical loss without reflecting observed behaviour. Post-convergence, CCTV frames corresponding to red-phase queues were geometrically calibrated using a planar homography, and the resulting bird's-eye-view images were used to measure the longitudinal and lateral gaps of heavy vehicles in real-world units. This confirmed that buses and trucks at this intersection consistently queue at bumper-to-bumper longitudinal gaps of approximately 0.5 m and lateral clearances of approximately 0.6 m. Accordingly, $G_{\min}$ was manually corrected to 0.50 m and `minGapLat` to 0.60 m for the heavy-vehicle class. All remaining parameters across all three vehicle classes were retained from θ_{opt} unchanged. The final parameter set adopted for all subsequent simulation runs is summarised in Table 4.3.

Table 4.3: Final manually refined parameter set used for baseline validation and RL agent training.

Parameter	**Category**			**Modified**
	A	**B**	**C**	
Headway time (τ)	0.67 s	0.80 s	1.38 s	—
Driver imperfection (σ)	0.53	0.22	0.22	—
Minimum longitudinal gap ($G_{\min}$)	0.22 m	0.82 m	0.50 m	Category C only
Minimum lateral gap (`minGapLat`)	0.11 m	0.30 m	0.60 m	Category C only
Lane-change assertiveness	4.24	1.11	1.62	—

Figure 4.5 illustrates the homography-based measurement workflow used to obtain these heavy-vehicle gap values. The top-left panel shows the selection of four ground control points on the road surface that define the planar homography between the CCTV image and real-world coordinates, while the top-right panel shows the terminal output reporting Euclidean distances between selected bumper points in metres. The bottom panel displays the calibrated frame with annotated bumper selections and distance segments for queued buses and trucks, from which the typical longitudinal (0.5 m) and lateral (0.6 m) gaps for the heavy-vehicle class were derived.

Figure 4.5: Homography-based measurement workflow for extracting heavy-vehicle gaps from CCTV (16 January 2026, 03:00–03:05 PM)

Figures 4.6 and 4.7 illustrate how the microscopic parameters for each vehicle class changed between the default SUMO configuration and the final calibrated model. In each case, the panels contrast the precalibration settings with the SPSA optimised and manually refined values, showing how headway time, minimum gaps, and lane-changing behaviour were tuned to match the observed traffic conditions at Narayan Gopal Chowk.

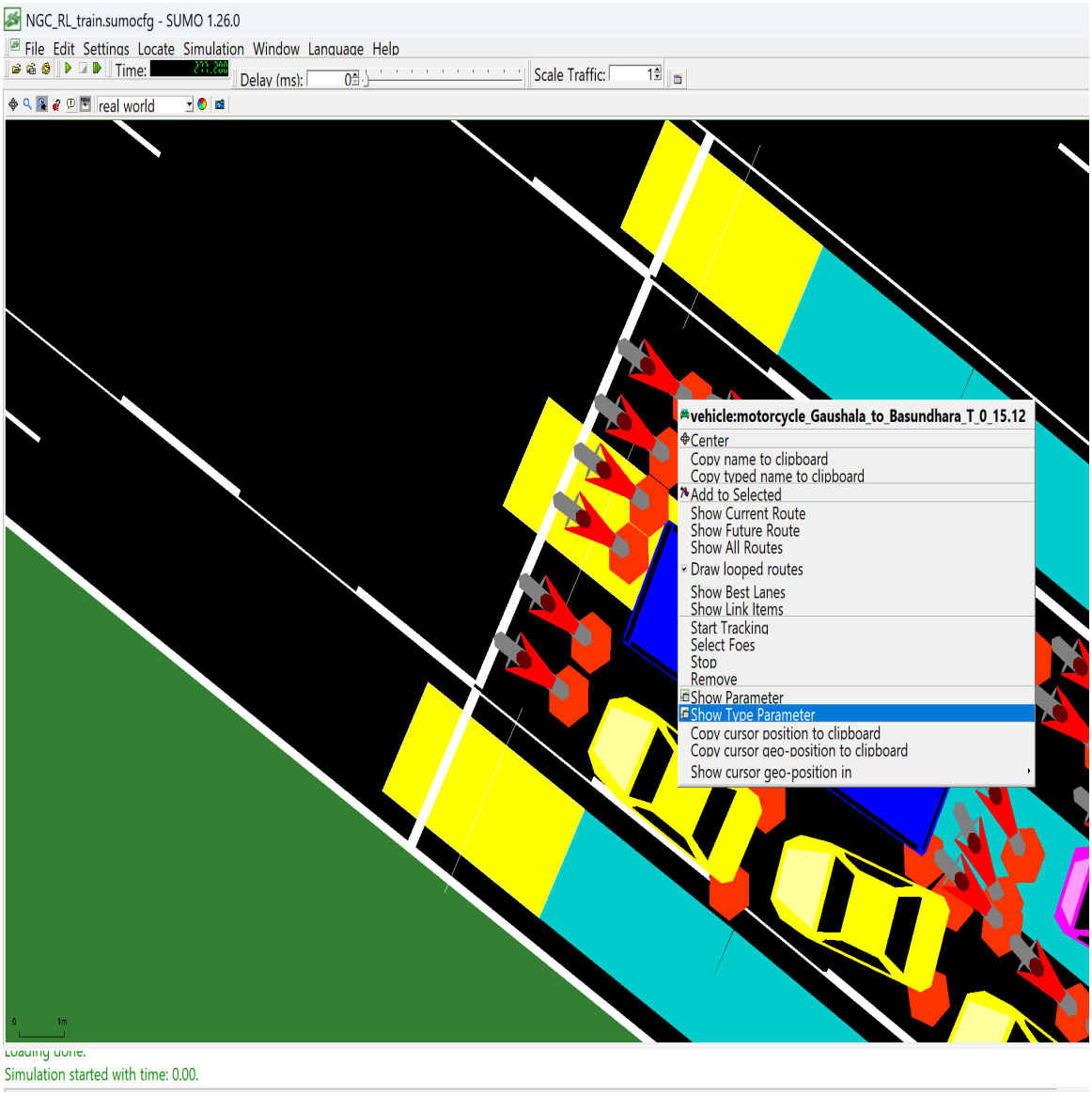


Figure 4.6: SUMO GUI snapshot for motorcycles from Category 'C': interface guideline for viewing parameters.

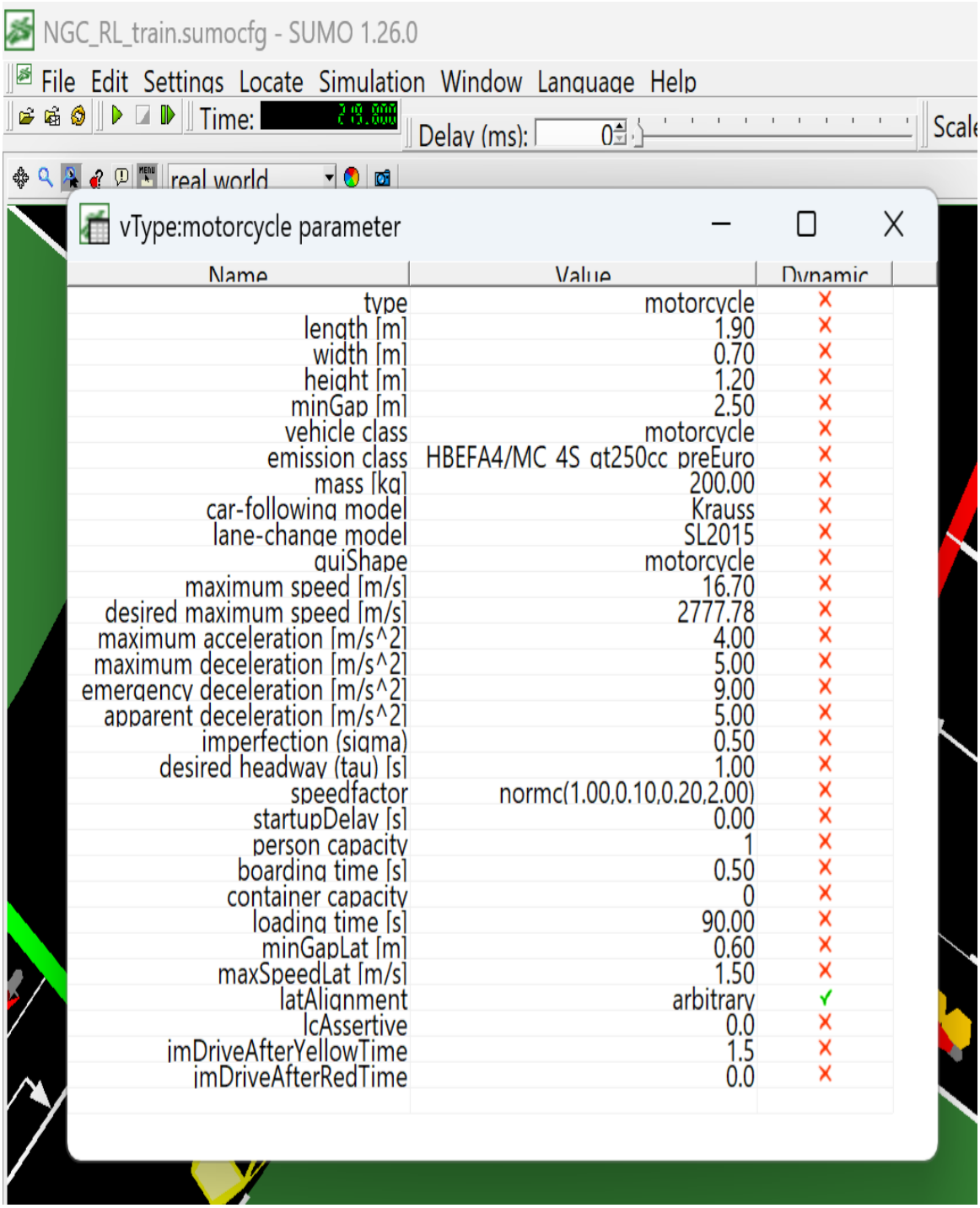


Figure 4.7: SUMO GUI snapshot for motorcycles from Category 'C': default microscopic settings.

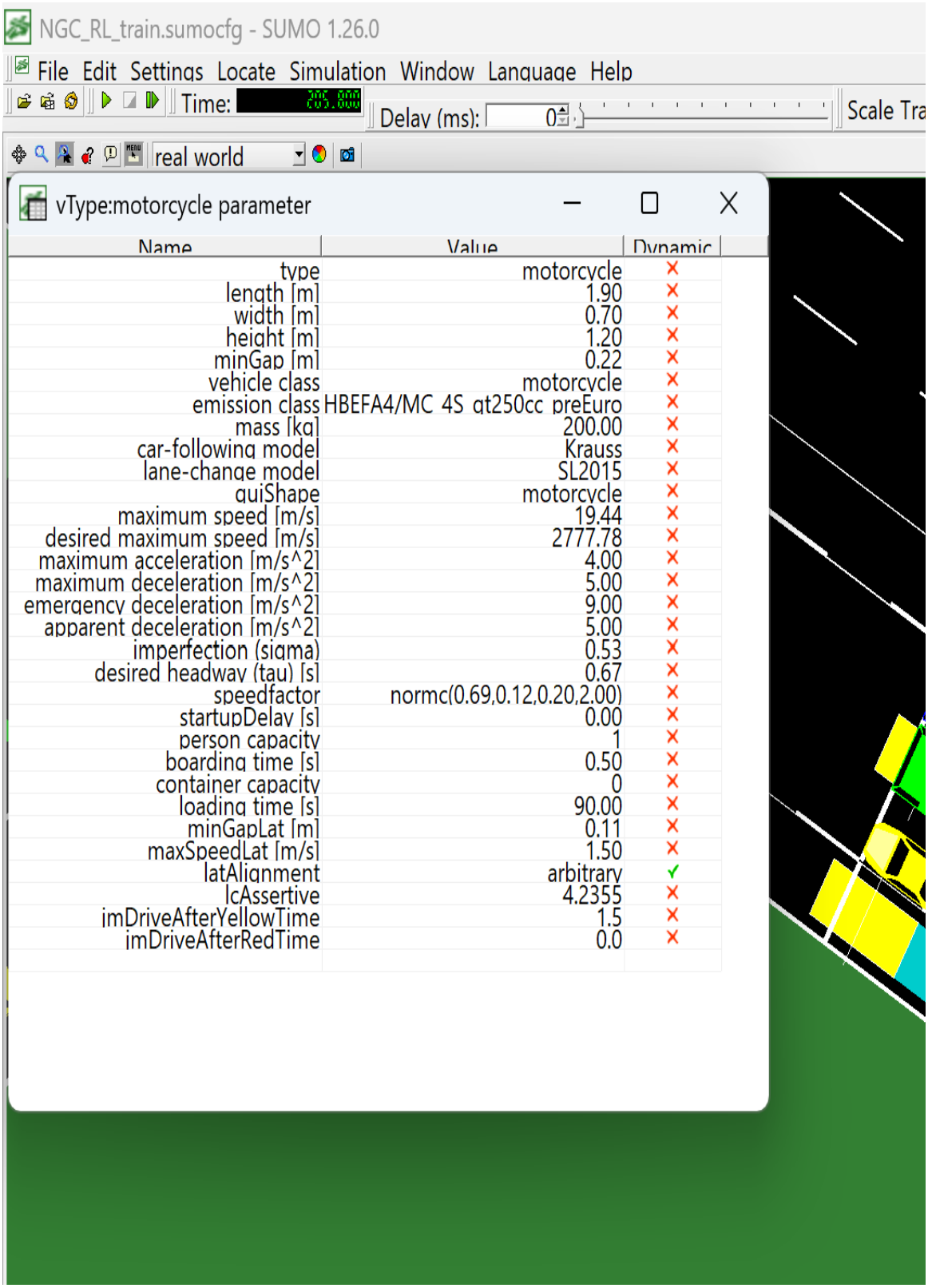


Figure 4.8: SUMO GUI snapshot for motorcycles from Category 'A': Final calibrated microscopic parameters.

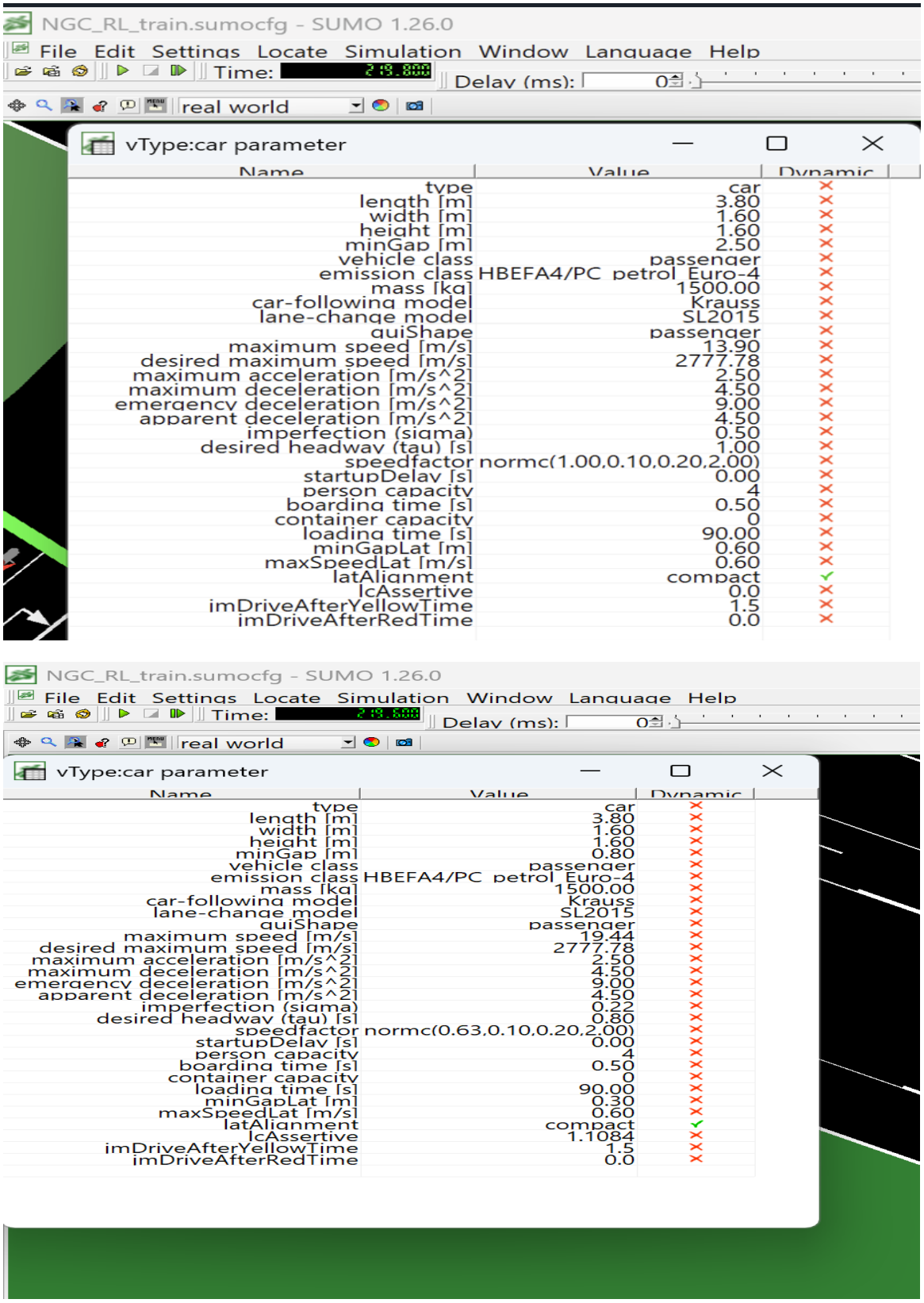


Figure 4.9: SUMO GUI snapshots for cars from Category 'B' : Default microscopic parameters (top) ; Final calibrated microscopic parameters (bottom).

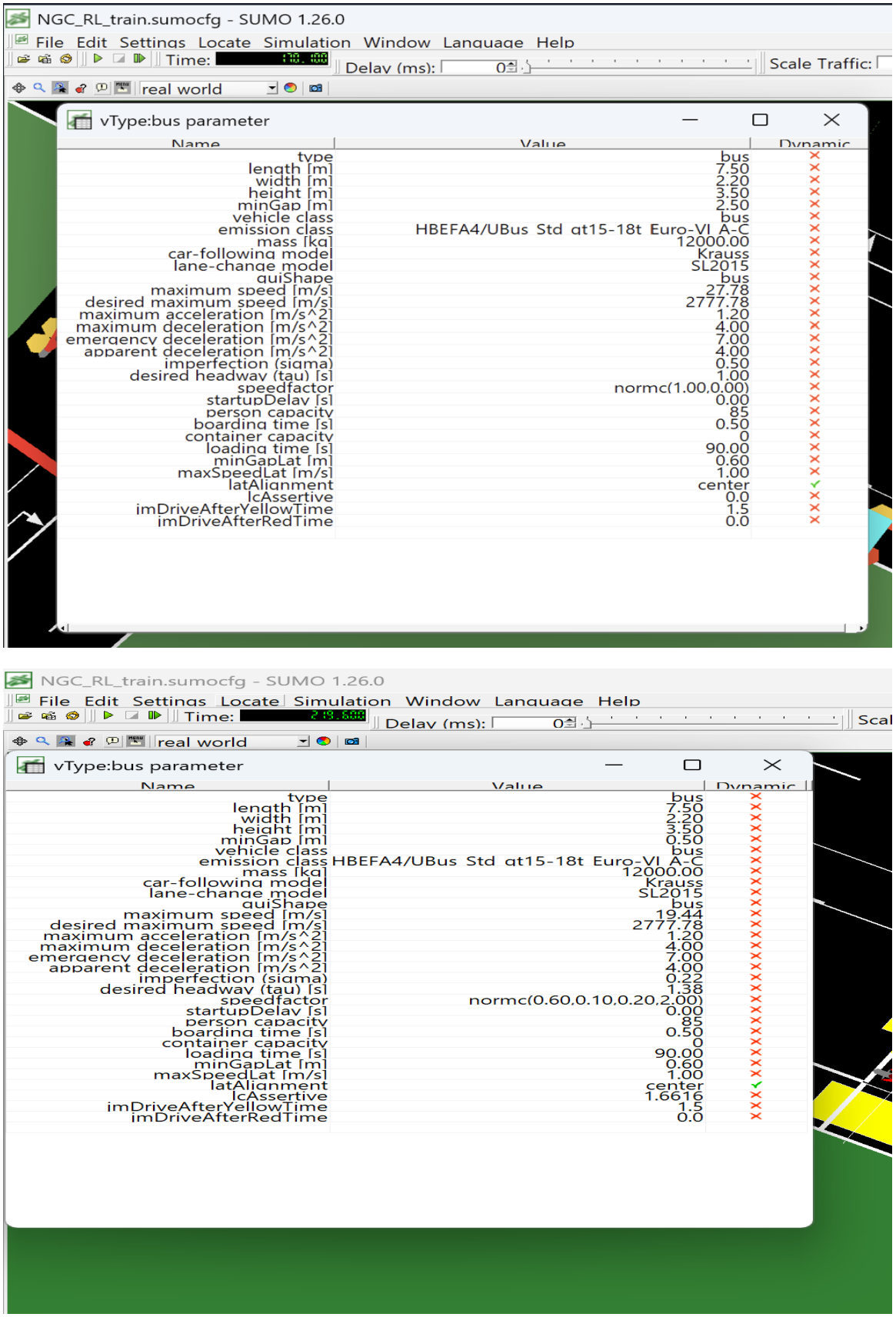


Figure 4.10: SUMO GUI snapshots for buses from Category 'C': Default microscopic parameters (top) ; Final calibrated microscopic parameters (bottom)

4.2 Statistical Validation of the Baseline Model

Using the manually refined parameter set, the baseline SUMO model was validated against an independent field dataset (16 January). Validation was conducted across two dimensions: temporal throughput and spatial congestion. In both cases, the Geoffrey E. Havers (GEH) statistic was used to assess the goodness of fit between field observations and simulation outputs at the approach level.

4.2.1 Macroscopic Throughput Validation (GEH)

The temporal accuracy of the simulation was assessed by comparing the total field-measured volume against the SUMO-simulated volume over a 1-hour peak window (12:00–13:00). Only vehicles that discharged through the stop line within the 0–3600 s interval were counted, so that field and simulation totals refer to the same temporal window.

Table 4.4: Throughput validation: field-measured vs. SUMO-simulated discharge (0–3600 s)

Approach	**Field (veh)**	**SUMO (veh)**	**Δ (veh)**	**% Error**	**GEH**	**Criterion**
Teaching	1,915	1,829	-86	-4.5%	**1.988**	GEH < 5.0 ✓
Gaushala	3,031	2,852	-179	-5.9%	**3.300**	GEH < 5.0 ✓
Budhanilakantha	2,428	2,314	-114	-4.7%	**2.341**	GEH < 5.0 ✓
Basundhara	3,041	2,804	-237	-7.8%	**4.384**	GEH < 5.0 ✓

Criterion: GEH < 5.0 at the approach level (UK TAG, DMRB). The consistent negative bias across all approaches is attributable to residual vehicles that remain queued at $t = 3600$ s and are therefore not counted in the strict discharge window.

All four approaches satisfy the GEH < 5.0 criterion. Teaching (GEH $= 1.988$) lies in the "Excellent" band (GEH < 2.0), while Budhanilakantha (GEH $= 2.341$), Gaushala (GEH $=$ 3.300), and Basundhara (GEH $= 4.384$) fall within the "Acceptable" band (2.0–5.0). The consistent negative bias across all four approaches ranging from -4.5% to -7.8% reflects a systematic undercounting artefact: vehicles still queued at $t = 3600$ s are excluded from the strict discharge window, which is physically expected at a saturated intersection operating at near-capacity demand. With Basundhara approaching the threshold at GEH $= 4.384$, the model demonstrates tighter capacity at this approach, consistent with the observed heavy

bus and through-traffic loading on the Kathmandu Ring Road corridor. Nevertheless, all approach-level GEH values remain below 5.0, confirming that the temporal throughput of the calibrated model is consistent with the observed intersection capacity.

4.2.2 Queue Length Validation (GEH)

To ensure that the spatial congestion patterns were also reproduced, simulated average queue lengths were compared against field observations on the same validation day (16 January). The GEH statistic was again employed, treating the average back-of-queue length (in metres) as the quantity of interest, thereby applying a consistent goodness-of-fit criterion across both validation dimensions. The network-level Root Mean Square Error (RMSE) across the four approaches is additionally reported as a supplementary spatial accuracy metric.

Table 4.5 presents the field and simulated average queue lengths per approach along with the corresponding GEH values, averaged over 11 complete signal cycles with the warm-up cycle excluded (simulation time 300–3600 s).

Table 4.5: Queue length validation: field vs. SUMO average back-of-queue (11 cycles, 300–3600 s)

Approach	**Field (m)**	**SUMO (m)**	**Δ (m)**	**% Error**	**GEH**	**Criterion**
Teaching	188.42	193.18	$+4.76$	$+2.5\%$	**0.345**	GEH < 5.0 ✓
Budhanilakantha	145.17	135.13	-10.04	-6.9%	**0.848**	GEH < 5.0 ✓
Gaushala	103.42	84.46	-18.96	-18.3%	**1.956**	GEH < 5.0 ✓
Basundhara	109.50	91.69	-17.81	-16.3%	**1.776**	GEH < 5.0 ✓
Network RMSE						**14.14 m**

All four approaches meet a very strong validation standard: every queue-length comparison has GEH below 2.0, which falls in the "Excellent" band. Teaching shows almost perfect agreement (GEH $= 0.345$, $\Delta = +4.76$ m), and Budhanilakantha (GEH $= 0.848$) is also excellent, with only about 10 m of under-simulation. The overall RMSE of 14.14 m corresponds to roughly two to three vehicle lengths per lane, confirming that, on average, the simulated back-of-queue positions are very close to the field observations.

Gaushala and Basundhara have larger percentage errors (-18.3% and -16.3%), but their GEH values (1.956 and 1.776) still lie in the excellent range. This is because, at queue lengths around 85–105 m, an absolute difference of 17–19 m is less severe than the same difference at smaller queues. The consistent underestimation of both throughput and queue length at these approaches points to one coherent bias (slightly fewer simulated departures) rather than random noise. Taken together, the small GEH values, modest spatial errors, and consistency with the calibrated vehicle dimensions and minimum gaps show that the SUMO model is accurate enough to be a trustworthy training and evaluation environment for the PN-D3QN agent.

To complement the numerical validation, Figures 4.11 and 4.12,Figures 4.13 and 4.14 compare simulated and observed queue formation at the Basundhara Approach and Gaushala Approach respectively. Similarly , Figure 4.15 compares simulated and observed queue formation Budhanilakantha Approach and Figure 4.16 compares the same at Teaching Approach. The SUMO snapshots show how queues build and dissipate at the stop line in the calibrated model, while the field image captures the corresponding real-world standing platoon and back-of-queue position. Together, these three images visually confirm that the simulated queues closely match the observed spatial congestion patterns at Narayan Gopal Chowk.

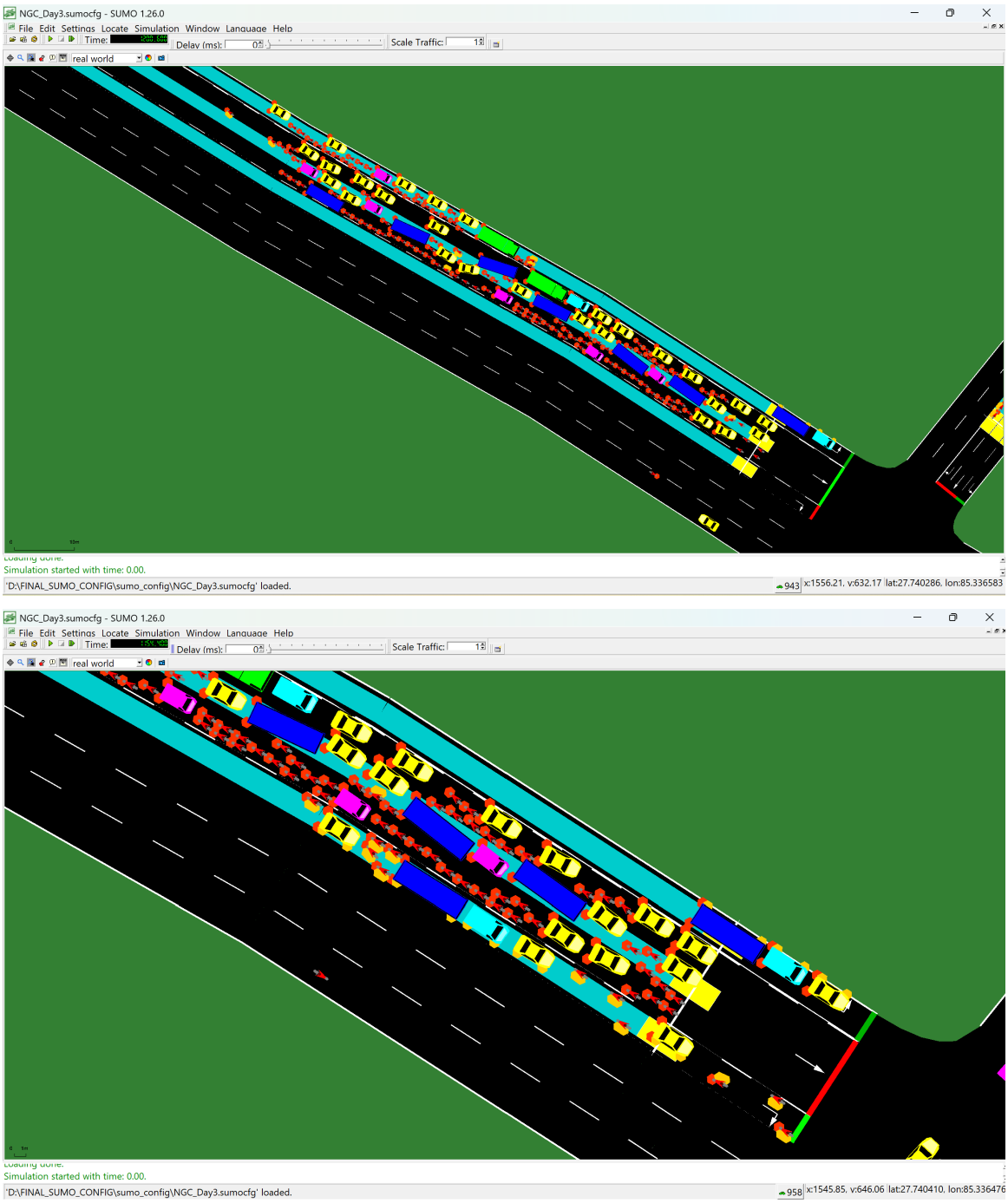


Figure 4.11: Maximum back of queue validation at Basundhara Approach : SUMO-simulated queues in the calibrated model (network view, top; zoomed approach view, bottom) during the 11-cycle validation window (16 January 2026, 12:15-12:20 PM)

Figure 4.12: Field observed maximum back of queue formation at the Basundhara approach of Narayan Gopal Chowk during the 11-cycle validation window (16 January 2026, 12:15-12:20 PM)

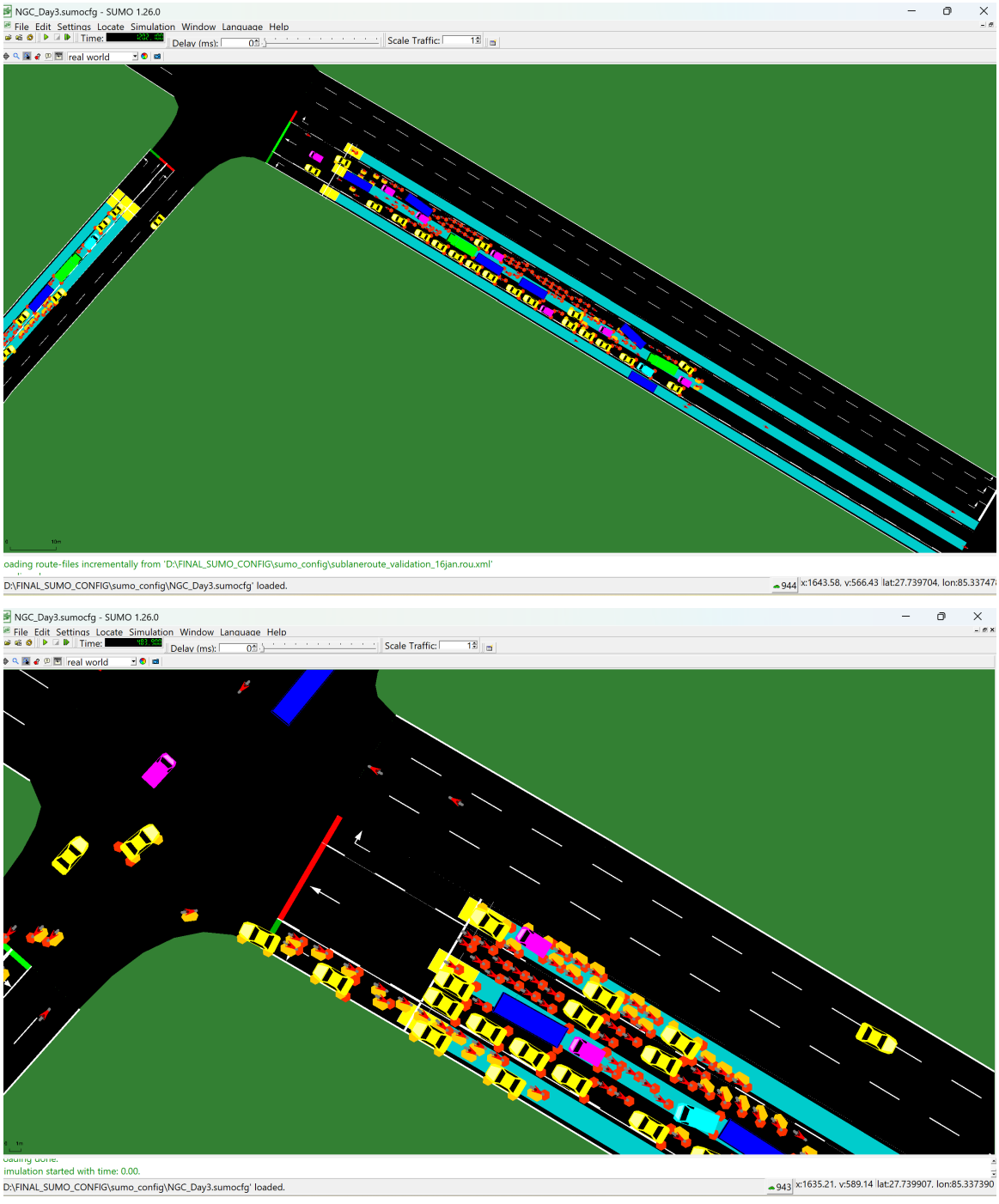


Figure 4.13: Maximum back of queue validation at Gaushala Approach: SUMO-simulated queues in the calibrated model (network view, top; zoomed approach view, bottom) during the 11-cycle validation window ((16 January 2026, 12:15-12:20 PM)

Figure 4.14: Field-observed maximum back of queue formation at the Gaushala approach of Narayan Gopal Chowk during the 11-cycle validation (16 January 2026, 12:15-12:20 PM).

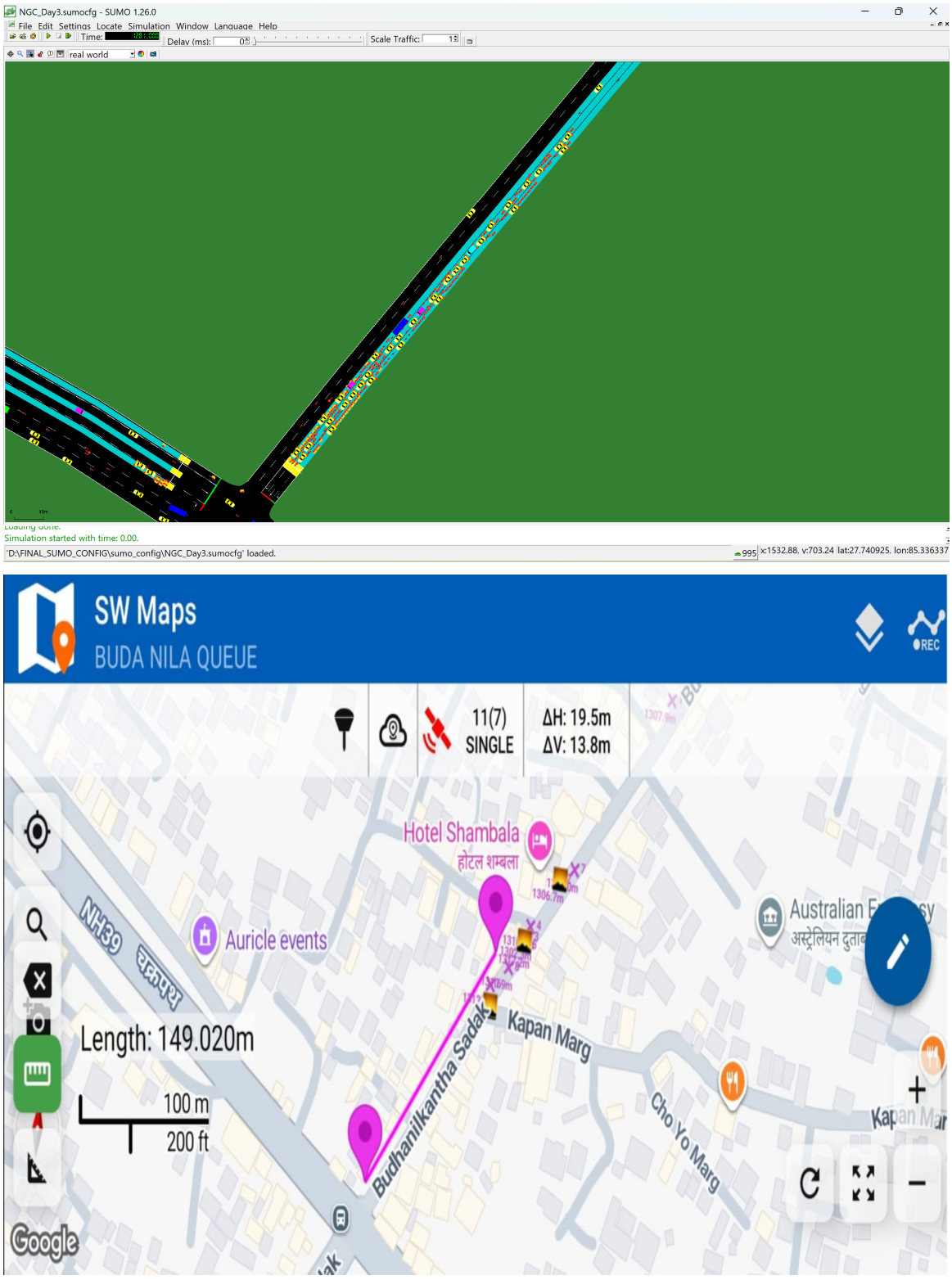


Figure 4.15: Maximum back of queue validation at Budhanilakantha Approach : SUMO-simulated queues in the calibrated model(Top) ; Actual Filed Measured Queue (Bottom) (16 January 2026, 12:15-12:20 PM)

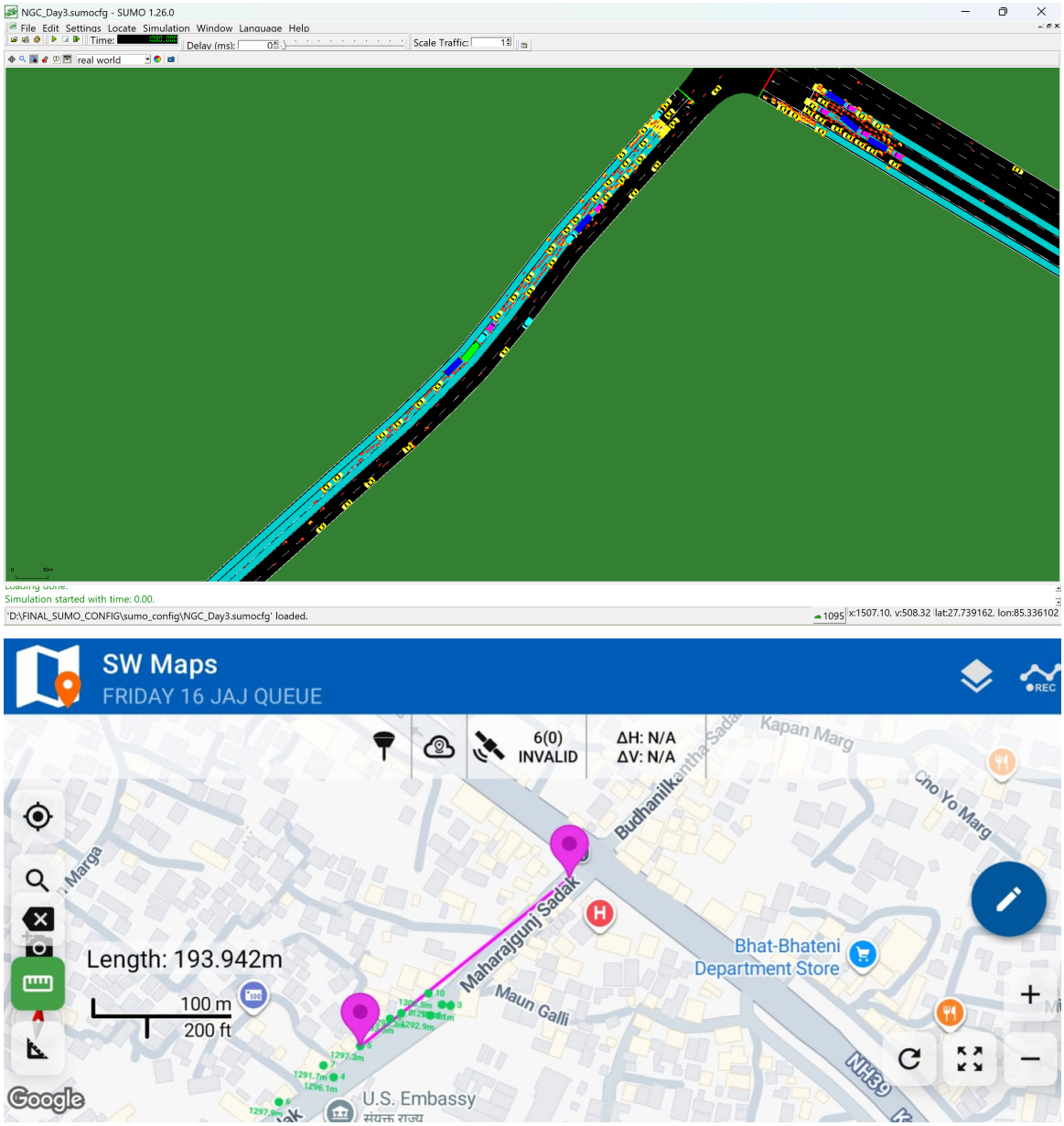


Figure 4.16: Maximum back of queue validation at Teaching Approach : SUMO-simulated queues in the calibrated model (Top); Actual Filed Measured Queue (Bottom) (16 January 2026, 12:15-12:20 PM)

4.3 Training Convergence and Sample Efficiency

In deep reinforcement learning, agents typically require thousands of training episodes to approach stable, near-optimal policies. In contrast, the PN-D3QN agent exhibits clear improvement within a relatively modest 331-episode horizon: the mean cumulative reward over all episodes is -8.307 (std. $= 5.312$), with values ranging from -25.282 to $+4.778$ and 18 episodes (5.4%) achieving positive cumulative reward. Comparing the first 50 episodes (mean reward $= -10.186$) to the final 50 episodes (mean reward $= -7.159$) reveals a net improvement of $+3.027$ reward units (a 29.7% relative gain), accompanied by a 12.4% reduction in average cumulative waiting time (from approximately 328,346 s to 287,630 s), which confirms that the agent steadily learns to reduce delay despite substantial stochastic variability.

To visualise this learning trajectory, the raw per-episode performance metrics were logged and smoothed using a 20-episode moving average to suppress variance from exploratory actions. Figure 4.17 depicts the evolution of cumulative reward, cumulative waiting time, and cumulative queue length over all 331 episodes; a global linear fit to the reward curve yields a positive slope of approximately $+0.0118$ reward units per episode, indicating monotonic improvement in expectation. This behaviour is consistent with the architectural enhancements in PN-D3QN: Prioritized Experience Replay concentrates updates on high-TD-error transitions, the Dueling network structure generalises across actions in weakly sensitive states, and Noisy Networks maintain persistent but structured exploration, together enabling substantial performance gains within a comparatively short training run.

The same mechanisms that drive this rapid improvement also explain the residual high variance across episodes and the absence of full convergence within 331 episodes. Episodic rewards continue to fluctuate between roughly -25 and $+5$ due to stochastic traffic demand and ongoing exploration noise, and only a small fraction of episodes attain strongly positive returns, suggesting that the agent has not yet fully mastered rare, severe-congestion regimes. Extending the training horizon to 600 or more episodes with gradually decaying exploration would allow PN-D3QN to transition from broad exploration toward systematic exploitation of high-performing behaviours, potentially yielding a fully converged policy that consistently dominates calibrated fixed-time controllers across all demand scenarios.

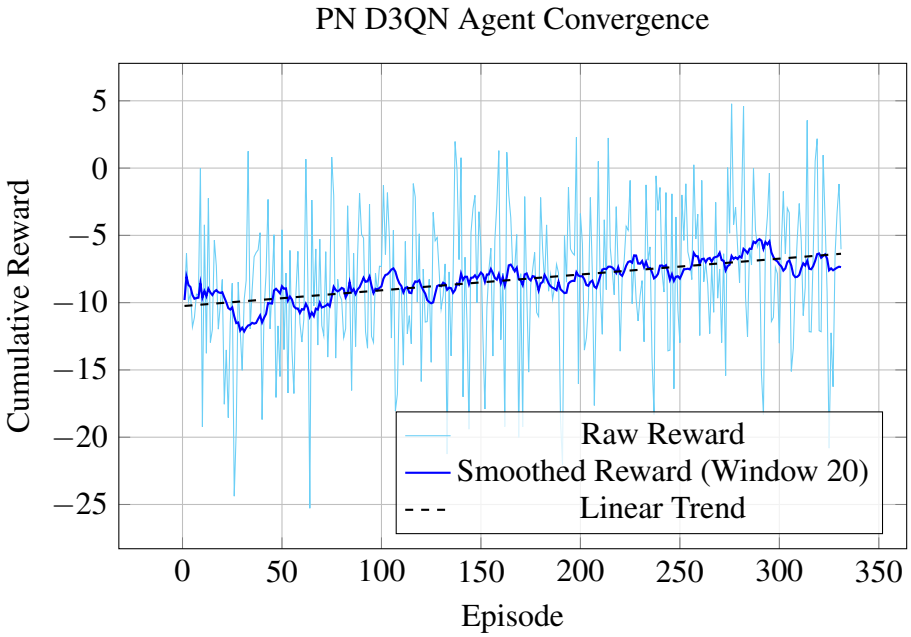


Figure 4.17: Training convergence of the PN-D3QN agent over 331 episodes: smoothed cumulative reward, cumulative waiting time, and cumulative queue length (20-episode moving averages).

4.4 Comprehensive Summary of Controller Performance

The experimental evaluation at Narayan Gopal Chowk with the updated non-warmup protocol provides a more granular picture of the Prioritized Noisy Dueling Double Deep Q-Network (PN-D3QN) agent relative to the calibrated fixed-time controller. The detailed operational metrics for the entire intersection, obtained by summing over all four approaches and averaging over continuous 1-hour simulation episodes for each demand scaling factor, are tabulated in Table 4.6 for total waiting time and Table 4.7 for mean queue length. Overall, PN-D3QN consistently reduces intersection-wide waiting time across all demand levels, with the largest improvement (about 39% reduction) at the 95% demand scenario, while exhibiting higher mean queues than the fixed plan at medium and high loads, indicating a control strategy that tolerates longer standing queues to minimise cumulative delay.

Table 4.6: Intersection Total Waiting Time (1-Hour Averages, New Evaluation)

Demand Scale	**Fixed TSC Wait Time (s)**	**PN-D3QN Wait Time (s)**	**Gain**
85%	14,020.12	10,993.25	**+21.59%**
90%	16,184.29	12,301.74	**+23.99%**
95%	16,847.59	10,350.29	**+38.57%**
105%	41,083.19	30,153.60	**+26.60%**
110%	90,228.17	79,109.53	**+12.32%**
115%	122,289.46	120,238.91	**+1.68%**

Note: Intersection totals are obtained by summing mean waiting times over all four approaches for each demand scale, averaged over 1-hour simulation episodes.

Table 4.7: Intersection Mean Queue Length (1-Hour Averages, New Evaluation)

Demand Scale	**Fixed TSC Queue (veh)**	**PN-D3QN Queue (veh)**	**Gain**
85%	262.73	199.30	**+24.14%**
90%	209.81	242.16	−15.42%
95%	223.17	248.68	−11.43%
105%	458.47	492.16	−7.35%
110%	802.09	999.72	−24.64%
115%	1,282.52	1,518.70	−18.42%

Note: Intersection queues are obtained by summing mean queues over all four approaches for each demand scale, averaged over 1-hour simulation episodes.

4.4.1 Robustness vs. Brittleness Across Demand Levels

A critical observation from Table 4.6 and Table 4.7 is that the PN-D3QN agent consistently outperforms the calibrated fixed-time controller in terms of total intersection waiting time for all evaluated demand levels, achieving reductions ranging from approximately 1.7% at 115% demand up to about 39% at 95% demand. In contrast, the mean queue length metric reveals a more nuanced picture: at the lowest demand level (85%), PN-D3QN simultaneously reduces both waiting time and queues (about 21.6% and 24.1% improvements, respectively), while at medium and high loads it typically accepts higher standing queues than the fixed-time plan. This behaviour highlights the brittle nature of the fixed-time controller. The baseline is effectively calibrated around a narrow operating

region and behaves like a highly specialised, rigidly tuned policy. It performs competitively only when realised traffic conditions remain close to its design assumptions; as demand moves away from that region, either on the under-saturated side (90–95%) or towards heavy oversaturation (105–115%), its delay performance deteriorates rapidly, culminating in very large total waiting times at the highest demand levels. The learned PN-D3QN policy, by contrast, exhibits robust generalisation across the entire demand spectrum, maintaining substantial delay reductions under both under-saturated and oversaturated regimes despite never being explicitly tuned for any single deterministic demand point.

4.4.2 Mechanisms of Delay Minimisation and Queue Trade-offs

Apart from the lowest demand scenario, the agent predominantly trades queue length efficiency for reductions in total waiting time. As summarised in Table 4.7, PN-D3QN incurs queue penalties on the order of 7–25% at 90–115% demand relative to the fixed-time plan, while Table 4.6 shows concurrent waiting-time gains between roughly 12% and 39%. This pattern reflects the deliberate design of the clipped, normalised reward,

$$R_t = \mathrm{clip}\left(-\alpha \frac{\Delta W}{W_{\max}} - \beta \frac{\Delta Q}{Q_{\max}},\ -1,\ +1\right), \tag{4.1}$$

in which phase-to-phase changes in both total waiting time ΔW and total queue ΔQ are penalised relative to baseline-derived reference scales $W_{\max}$ and $Q_{\max}$.

Aggressively clearing a congested approach reduces the queue term $\Delta Q/Q_{\max}$ on that leg but simultaneously induces large increases in waiting time $\Delta W/W_{\max}$ on opposing red phases. Through interaction, the agent discovers that this behaviour yields lower long-run return than maintaining a more stable flow with pre-emptive phase changes that keep both normalised changes within moderate bounds. The resulting policy explicitly prioritises smoothing delay and preventing extreme waiting-time spikes, even at the cost of tolerating somewhat higher average queues at medium and high demand levels.

4.4.3 Theoretical Pathway to Dual Superiority

While the current PN-D3QN agent achieves its primary objective of robust delay reduction under demand stochasticity, achieving simultaneous superiority in both waiting time and queue length remains theoretically attainable. A calibrated fixed-time controller is mathematically rigid and, by construction, cannot adapt to large deviations from its design flows, guaranteeing phase failures and severe spillback under persistent surges. In contrast, a reinforcement learning agent with full state-space visibility can, in principle, learn queue-dissipation strategies that avoid both excessive delay and excessive spillback.

The present agent was trained under a deliberately constrained 60-episode training budget, which proved sufficient to learn a strongly generalisable delay-reduction policy but did not allow exhaustive exploration of the long-tail state–action combinations associated with extreme congestion and recovery. Future work with an extended training horizon (e.g. 500+ episodes), richer exploration, and dynamic reward weighting that emphasises queue dissipation under high saturation is expected to enable PN-D3QN-like architectures to jointly minimise total delay and mean queue length. Under such conditions, the RL-based controller is mathematically positioned to dominate rigid pretimed plans across all operational metrics of interest.

4.4.4 Visual Summary of Performance Metrics

To provide a clear visual representation of the controller performance across the various demand scenarios, Figure 4.18 and Figure 4.19 illustrate the aggregated intersection waiting time and mean queue length, respectively, as a function of demand scaling.

As depicted in Figure 4.18, the PN-D3QN agent consistently reduces intersection-wide delay relative to the fixed-time baseline for all demand levels, with particularly pronounced gains at 85%–105% demand (up to approximately 39% reduction at 95%). Even under heavily oversaturated conditions (110% and 115%), the learned policy continues to outperform the pretimed plan in terms of total waiting time, albeit with diminishing relative benefit at the highest scale.

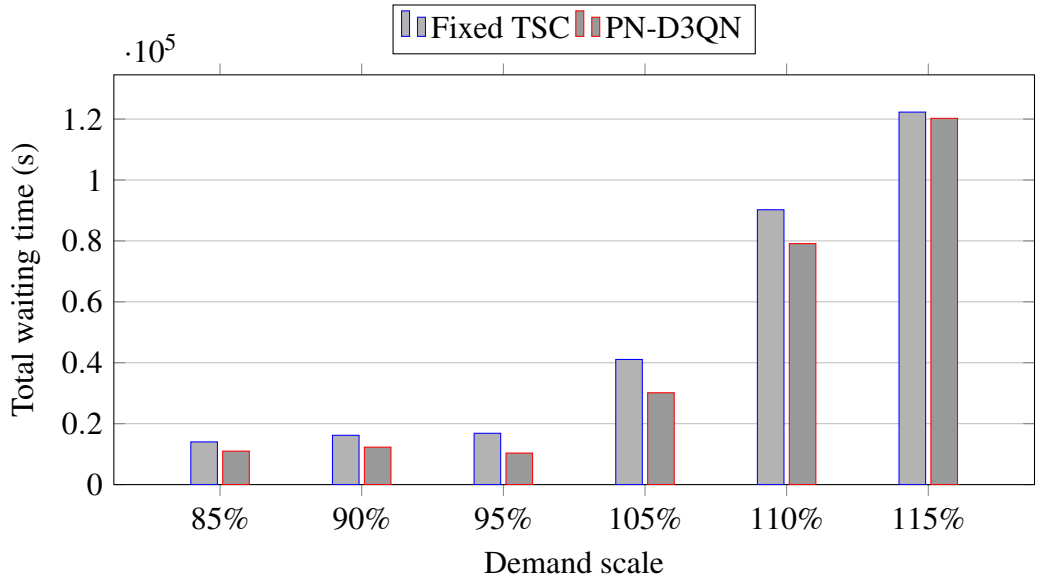


Figure 4.18: Intersection total waiting time versus demand scale for Fixed-time and PN-D3QN controllers (1-hour averages).

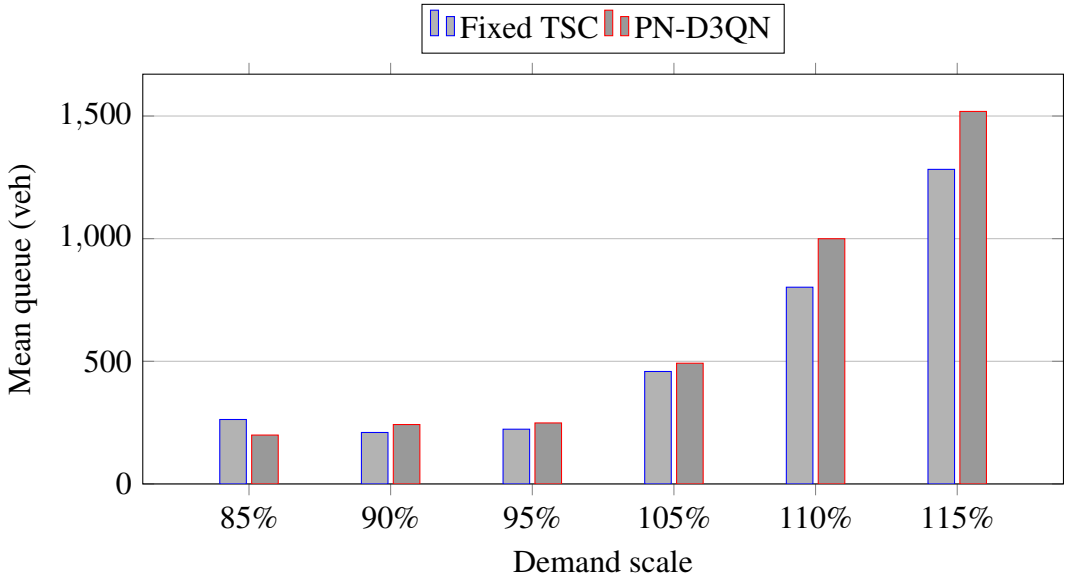


Figure 4.19: Intersection mean queue length versus demand scale for Fixed-time and PN-D3QN controllers (1-hour averages).

In contrast, Figure 4.19 shows that the PN-D3QN agent does not uniformly minimise queue length: at moderate and high demand levels it allows higher standing queues than the fixed-time controller, while still achieving lower total delay. This confirms a deliberate flow–stability trade-off, where the agent prioritises reducing cumulative waiting time over aggressive queue clearance, thereby avoiding the explosive growth in delay observed for the calibrated fixed-time plan in oversaturated regimes.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The proposed PN-D3QN controller, trained within an SPSA-calibrated SUMO model of Narayan Gopal Chowk, demonstrates clear operational advantages over a carefully tuned fixed-time baseline under realistic, stochastic traffic conditions. Intersection-level results show that PN-D3QN consistently reduces total waiting time across all evaluated demand scales, with delay reductions ranging from approximately 1.7% at 115% demand up to about 39% at 95% demand, while also simultaneously lowering both delay and queues at the lowest demand level (85%). At medium and high demand, the agent deliberately tolerates 7 to 25% higher mean queues in exchange for substantial delay savings, reflecting the intended behaviour of its reward structure that penalises rapid growth in network-wide waiting time more heavily than incremental changes in standing queues. Given that these gains are achieved in a microscopically calibrated, highly heterogeneous traffic environment, the evidence strongly supports the conclusion that PN-D3QN offers a robust and practically viable alternative to fixed-time signal control for congested urban intersections in developing cities.

5.2 Recommendations for Future Work

To extend these findings, future research should incorporate real-time sensor data, such as video-based trajectory tracking, to enable dynamic, online model updates. Integrating multi-modal signal priority specifically for public buses would further enhance the transit-oriented performance of the PN-D3QN framework. Additionally, expanding the study to a network-wide scale would allow for the assessment of corridor-level spillback effects and coordinated signal synchronization. Finally, testing the transferability of this model to other intersections in Kathmandu with varying geometric layouts would validate the generalizability of the trained control policies.

REFERENCES

[1] P. Michailidis, I. Michailidis, C. R. Lazaridis, and E. Kosmatopoulos, "Traffic Signal Control via Reinforcement Learning: A Review on Applications and Innovations," *Infrastructures (Basel)*, vol. 10, no. 5, pp. 1–41, 2025, doi: 10.3390/infrastructures10050114.

[2] M. Greguric, M. Vujic, C. Alexopoulos, and M. Miletic, "Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data," *Applied Sciences (Switzerland)*, vol. 10, no. 11, 2020, doi: 10.3390/app10114011.

[3] M. Skuba, A. Janota, P. Kuchar, and B. Malobicky, "Deep Reinforcement Learning for Traffic Signal Control," *Transportation Research Procedia*, vol. 74, pp. 954–958, 2023, doi: 10.1016/j.trpro.2023.11.230.

[4] E. F. Morales and J. H. Zaragoza, "An introduction to reinforcement learning," in *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, pp. 63–80, 2011, doi: 10.4018/978-1-60960-165-2.ch004.

[5] S. Park, E. Han, S. Park, H. Jeong, and I. Yun, "Deep Q-network-based traffic signal control models," *PLoS One*, vol. 16, no. 9, pp. 1–14, 2021, doi: 10.1371/journal.pone.0256405.

[6] F. Rasheed, K. L. A. Yau, R. M. Noor, C. Wu, and Y. C. Low, "Deep Reinforcement Learning for Traffic Signal Control: A Review," *IEEE Access*, vol. 8, pp. 208016–208044, 2020, doi: 10.1109/ACCESS.2020.3034141.

[7] H. Wei, G. Zheng, H. Yao, and J. Li, "IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control," in *Proc. 24th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 2496–2505, 2018, doi: 10.1145/3219819.3220096.

[8] X. Liang, X. Du, G. Wang, and Z. Han, "A Deep Reinforcement Learning Network for Traffic Light Cycle Control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019, doi: 10.1109/TVT.2018.2890726.

[9] S. Han, W. Zhou, J. Liu, and S. Lü, "NROWAN-DQN: A Stable Noisy Network with Noise Reduction and Online Weight Adjustment for Exploration," Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.10980

[10] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," in *Proc. 4th Int. Conf. on Learning Representations (ICLR)*, 2016, [Online]. Available: https://arxiv.org/abs/1511.05952

[11] C. Cai and M. Wei, "Adaptive urban traffic signal control based on enhanced deep reinforcement learning," *Scientific Reports*, pp. 1–11, 2024, doi: 10.1038/s41598-024-64885-w.

[12] D. Li, J. Wu, M. Xu, Z. Wang, and K. Hu, "Adaptive Traffic Signal Control Model on Intersections Based on Deep Reinforcement Learning," *Journal of Advanced Transportation*, vol. 2020, 2020, doi: 10.1155/2020/6505893.

[13] W. Genders and S. Razavi, "Using a Deep Reinforcement Learning Agent for Traffic Signal Control," pp. 1–9, 2016, [Online]. Available: http://arxiv.org/abs/1611.01142

[14] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, and D. O. Wu, "Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8243–8256, Aug. 2020, doi: 10.1109/TVT.2020.2997896.

[15] B. T. Wiskunde, "An overview of microscopic and macroscopic traffic models," Thesis, Universiteit Utrecht, 2013.

[16] K. S. Sang and B. Zhou, "Reinforcement Learning for Vehicle Route Optimization in SUMO," in *Proc. IEEE Int. Conf. on Cybernetics and Intelligent Systems*, 2014.

[17] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target

Network," pp. 1–10, 2017, [Online]. Available: http://arxiv.org/abs/1705.02755

[18] S. M. M. R. Swapno, M. A. Islam, M. S. Rahman, and M. M. Hossain, "A Reinforcement Learning Approach for Reducing Traffic Congestion Using Deep Q-Learning," pp. 1–20, 2024.

[19] R. Ducrocq and N. Farhi, "Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection," *International Journal of Intelligent Transportation Systems Research*, vol. 21, no. 1, pp. 192–206, 2023, doi: 10.1007/s13177-023-00346-4.

[20] S. Khaleghian, H. Neema, M. Sartipi, T. Tran, R. Sen, and A. Dubey, "Calibrating Real-World City Traffic Simulation Model Using Vehicle Speed Data," in *Proc. 2023 IEEE Int. Conf. on Smart Computing (SMARTCOMP 2023)*, pp. 303–308, 2023, doi: 10.1109/SMARTCOMP58114.2023.00076.

[21] S. Singh, V. Rajesh, P. Adhikari, and S. M. Santhakumar, "Micro-simulation Study on the Effect of Motorized Two-Wheelers on Traffic Speed Characteristics," in *Proc. Conf. on Transportation Systems Engineering and Management (CTSEM 2020)*, Paper ID: CTSEM 2020-149, pp. 75–79, 2020.

[22] J. C. Spall, "Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 817–823, 1998, doi: 10.1109/7.705889.

[23] T. Jayasinghe, T. Sivakumar, and A. S. Kumarage, "Calibration of SUMO Microscopic Simulator for Sri Lankan Traffic Conditions," *Proceedings of the Eastern Asia Society for Transportation Studies*, vol. 13, 2021.

[24] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Vinyals, C. Blundell, and S. Legg, "Noisy Networks for Exploration," in *Proc. 6th Int. Conf. on Learning Representations (ICLR)*, 2018, [Online]. Available: https://arxiv.org/abs/1706.10295

[25] N. Bajracharya and S. Dhungel, "Urban Intersection Modelling for Signal Coordination and Adaptive Traffic Control under Heterogeneous Traffic Condition: A

Case Study of Keshar Mahal and Durbar Marg Intersections," in *Proc. 2nd Int. Conf. on Integrated Transport for Sustainable Mobility*, Kathmandu, Nepal, Feb. 2023.

[26] V. Sashank and C. Navali, "Calibration of SUMO for Indian Heterogeneous Traffic Conditions," in *Proc. Int. Conf. on Transportation and Development (ICTD)*, 2021, doi: 10.1007/978-981-16-0551-4_47.

[27] My Republica, "LMC Installs Intelligent Traffic Lights at Five Locations in Lalitpur," *My Republica*, Dec. 16, 2024. [Online]. Available: https://myrepublica.nagarikknetwork.com/news/lmc-installs-intelligent-traffic-lights-at-five-locations-in-lalitpur-675fe5a2423a0.html

[28] A. Chhetri and P. K. Shrestha, "Calibration of Car-following Parameters in VISSIM for Traffic in Kathmandu," in *Proceedings of the 14th IOE Graduate Conference*, vol. 14, pp. 1369–1374, Dec. 2023, ISSN: 2350-8914 (Online), 2350-8906 (Print).

[29] M. Maharjan and S. Budhathoki, "Leveraging Multi-Agent Deep Deterministic Policy Gradient (MADDPG) for Real-Time Traffic Signal Optimization," *MVIC Journal of Management and Information Technology*, vol. 1, no. 2, 2025.

ANNEX A: SUMMARY OF FIELD DATA

Table 1: Maximum Back of Queue Measurements - Day 1 (01/11/2026)

Cycle ID	Time Range	Basundhara (m)	Gaushala (m)	Teaching (m)	Budhanilakantha (m)
1	12:00 - 12:05	124	88	195	256
2	12:05 - 12:10	117	120	124	227
3	12:10 - 12:15	105	65	125	220
4	12:15 - 12:20	100	117	155	230
5	12:20 - 12:25	112	133	137	210
6	12:25 - 12:30	135	103	187	225
7	12:30 - 12:35	86	105	213	195
8	12:35 - 12:40	82	130	153	240
9	12:40 - 12:45	121	149	121	215
10	12:45 - 12:50	145	135	235	250
11	12:50 - 12:55	118	117	144	253
12	12:55 - 13:00	112	130	169	235

Table 2: Maximum Back of Queue Measurements - Day 2 (01/13/2026)

Cycle ID	Time Range	Basundhara (m)	Gaushala (m)	Teaching (m)	Budhanilakantha (m)
1	12:00 - 12:05	132	110	145	213
2	12:05 - 12:10	125	118	201	245
3	12:10 - 12:15	113	125	174	230
4	12:15 - 12:20	127	111	156	225
5	12:20 - 12:25	105	80	138	250
6	12:25 - 12:30	128	67	160	235
7	12:30 - 12:35	86	88	98	250
8	12:35 - 12:40	150	79	156	240
9	12:40 - 12:45	136	68	142	215
10	12:45 - 12:50	94	102	146	275
11	12:50 - 12:55	151	105	187	255
12	12:55 - 13:00	127	98	245	250

Table 3: Maximum Back of Queue Measurements - Day 3 (01/16/2026)

Cycle ID	Time Range	Basundhara (m)	Gaushala (m)	Teaching (m)	Budhanilakantha (m)
1	12:00 - 12:05	125	102	156	155
2	12:05 - 12:10	110	90	145	205
3	12:10 - 12:15	90	110	141	151
4	12:15 - 12:20	110	115	238	165
5	12:20 - 12:25	100	88	211	120
6	12:25 - 12:30	127	78	280	166
7	12:30 - 12:35	80	102	238	135
8	12:35 - 12:40	110	86	172	142
9	12:40 - 12:45	130	80	181	130
10	12:45 - 12:50	80	110	147	115
11	12:50 - 12:55	135	115	182	120
12	12:55 - 13:00	118	125	170	138

Table 4: Classified Traffic Volume - Day 1 (01/11/2026)

Movement	Moto	Micro	Bus	Carrier	Truck	Car
Basundhara to Gaushala (T)	1415	12	72	80	29	291
Basundhara to Budhanilakantha (L)	625	7	20	25	5	123
Basundhara to Teaching (R)	187	19	9	5	0	57
Gaushala to Basundhara (T)	1548	11	73	74	24	325
Gaushala to Teaching (L)	567	4	13	20	2	140
Gaushala to Budhanilakantha (R)	335	0	4	12	0	94
Teaching to Basundhara (L)	423	16	6	11	1	71
Teaching to Budhanilakantha (T)	648	15	9	14	0	206
Teaching to Gaushala (R)	315	2	10	5	2	91
Budhanilakantha to Basundhara (R)	338	1	18	16	8	78
Budhanilakantha to Teaching (T)	724	16	9	11	0	291
Budhanilakantha to Gaushala (L)	592	5	8	16	2	181

Table 5: Classified Traffic Volume - Day 2 (01/13/2026)

Movement	Moto	Micro	Bus	Carrier	Truck	Car
Basundhara to Gaushala (T)	1362	15	94	97	19	253
Basundhara to Budhanilakantha (L)	658	5	19	35	2	119
Basundhara to Teaching (R)	192	18	8	4	0	65
Gaushala to Basundhara (T)	1522	14	79	75	39	288
Gaushala to Teaching (L)	613	8	16	22	2	155
Gaushala to Budhanilakantha (R)	267	4	5	15	5	86
Teaching to Basundhara (L)	451	25	12	8	0	71
Teaching to Budhanilakantha (T)	693	14	13	15	0	192
Teaching to Gaushala (R)	297	5	14	11	1	71
Budhanilakantha to Basundhara (R)	353	4	19	15	4	93
Budhanilakantha to Teaching (T)	775	21	12	17	1	254
Budhanilakantha to Gaushala (L)	603	8	8	23	3	141

Table 6: Classified Traffic Volume - Day 3 (01/16/2026)

Movement	Moto	Micro	Bus	Carrier	Truck	Car
Basundhara to Gaushala (T)	1413	10	91	77	14	288
Basundhara to Budhanilakantha (L)	647	4	18	33	4	125
Basundhara to Teaching (R)	220	17	8	7	0	65
Gaushala to Basundhara (T)	1430	24	81	66	20	318
Gaushala to Teaching (L)	543	10	15	11	1	152
Gaushala to Budhanilakantha (R)	273	1	7	18	0	61
Teaching to Basundhara (L)	426	21	10	9	0	98
Teaching to Budhanilakantha (T)	667	17	14	18	2	215
Teaching to Gaushala (R)	307	3	16	8	0	84
Budhanilakantha to Basundhara (R)	366	3	14	18	4	94
Budhanilakantha to Teaching (T)	759	22	15	17	0	288
Budhanilakantha to Gaushala (L)	663	5	9	16	1	134

Table 7: Motorcycle Speed Data(1)

Basundhara Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
1	9.89	35.61
2	11.11	40.00
3	8.34	30.00
4	7.00	25.20
5	12.08	43.45
6	8.54	30.74
7	9.46	34.05
8	9.46	34.05
9	9.55	34.40
10	11.29	40.62
11	8.86	31.90
12	9.89	35.61
13	10.61	38.19
14	6.67	24.00
15	6.67	24.00
16	17.50	63.00
17	8.59	30.90
18	10.61	38.19
19	12.28	44.21
20	7.53	27.10
21	7.08	25.46
22	8.86	31.90
23	8.34	30.00
24	6.67	24.00
25	7.61	27.39
26	7.78	28.00
27	8.34	30.00
28	10.94	39.38
29	10.45	37.61
30	10.15	36.53
31	11.59	41.70
32	11.29	40.65
33	10.15	36.53
34	10.45	37.61
35	9.59	34.51
36	10.50	37.80
37	11.11	40.00
38	9.21	33.16
39	10.45	37.61
40	9.46	34.05
41	7.95	28.64
42	12.96	46.66
43	11.11	40.00
44	12.96	46.66
45	11.62	41.85
46	13.46	48.46
47	9.86	35.49
48	13.21	47.55
49	11.47	41.31
50	7.69	27.70

Gaushala Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
51	8.24	29.65
52	8.14	29.30
53	8.64	31.11
54	8.54	30.74
55	10.15	36.53
56	8.75	31.50
57	7.61	27.39
58	8.98	32.31
59	7.95	28.64
60	8.75	31.50
61	9.59	34.53
62	11.47	41.31
63	10.94	39.38
64	11.55	41.60
65	10.94	39.38
66	10.78	38.76
67	12.96	46.66
68	10.45	37.61
69	8.05	28.96
70	11.11	40.00
71	10.45	37.61
72	12.96	46.66
73	12.28	44.21
74	7.95	28.64
75	10.45	37.61
76	10.61	38.19
77	12.08	43.45
78	9.93	35.71
79	10.21	36.75
80	12.21	43.99
81	11.66	42.00
82	9.91	35.67
83	12.66	45.60
84	9.55	34.40
85	12.21	43.97
86	9.59	34.50
87	13.21	47.55
88	12.28	44.21
89	10.45	37.61
90	12.28	44.21
91	12.28	44.21
92	12.08	43.45
93	11.11	40.00
94	10.61	38.19
95	10.61	38.19
96	8.75	31.50
97	11.11	40.00
98	7.36	26.53
99	10.94	39.38
100	8.98	32.31

Table 8: Motorcycle Speed Data(2)

Budanilakantha Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
101	9.09	32.72
102	4.55	16.36
103	7.81	28.13
104	7.25	26.09
105	6.25	22.50
106	7.08	25.47
107	8.52	30.69
108	7.98	28.73
109	10.41	37.50
110	5.78	20.76
111	11.36	40.91
112	8.83	31.76
113	7.81	28.13
114	4.78	17.20
115	7.21	25.96
116	6.70	24.10
117	14.70	52.94
118	10.56	38.03
119	7.01	25.24
120	6.95	25.00
121	7.81	28.13
122	8.52	30.69
123	8.34	30.00
124	9.61	34.61
125	7.98	28.73
126	6.41	23.08
127	9.26	33.34
128	7.58	27.28
129	7.50	27.00
130	6.81	24.55
131	6.53	23.48
132	6.20	22.31
133	11.11	40.00
134	6.95	25.00
135	5.75	20.69
136	5.50	19.79
137	5.21	18.75
138	5.00	18.00
139	6.58	23.69
140	5.21	18.75
141	4.50	16.21
142	6.66	24.00
143	8.93	32.15
144	4.68	16.83
145	8.34	30.00
146	6.03	21.69
147	7.25	26.09
148	5.44	19.56
149	5.55	20.00
150	6.50	23.38

Teaching Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
151	15.25	54.88
152	12.50	45.00
153	7.02	25.28
154	8.93	32.14
155	13.03	46.88
156	8.45	30.41
157	10.25	36.89
158	10.78	38.80
159	9.61	34.61
160	8.56	30.83
161	6.95	25.00
162	7.81	28.13
163	8.34	30.00
164	7.81	28.13
165	7.91	28.48
166	7.44	26.79
167	7.63	27.44
168	8.45	30.41
169	8.68	31.25
170	8.80	31.69
171	15.25	54.88
172	12.50	45.00
173	11.58	41.66
174	11.58	41.66
175	8.56	30.83
176	7.53	27.11
177	8.23	29.60
178	8.01	28.85
179	6.65	23.94
180	6.45	23.20
181	11.36	40.91
182	9.76	35.15
183	8.23	29.60
184	6.58	23.69
185	5.63	20.28
186	6.38	22.96
187	7.81	28.13
188	7.91	28.48
189	7.53	27.11
190	8.56	30.83
191	7.53	27.11
192	8.80	31.69
193	10.59	38.14
194	9.19	33.09
195	7.91	28.48
196	15.88	57.15
197	14.88	53.58
198	12.00	43.20
199	8.61	31.00
200	11.71	42.16

Table 9: Car Speed Data(1)

Basundhara Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
1	8.20	29.50
2	8.00	28.80
3	8.00	28.80
4	8.00	28.80
5	5.43	19.51
6	9.34	33.60
7	6.59	23.71
8	12.45	44.80
9	7.81	28.13
10	12.00	43.20
11	12.93	46.53
12	8.20	29.50
13	7.30	26.29
14	8.84	31.84
15	9.89	35.58
16	7.00	25.20
17	8.40	30.24
18	7.81	28.13
19	8.61	31.01
20	8.61	31.01
21	9.60	34.56
22	9.34	33.60
23	10.50	37.80
24	12.45	44.80
25	11.61	41.79
26	13.66	49.20
27	13.44	48.39
28	10.50	37.80
29	8.20	29.50
30	8.40	30.24
31	7.30	26.29
32	7.81	28.13
33	7.00	25.20
34	5.01	18.05
35	10.19	36.65
36	8.40	30.24
37	10.68	38.41
38	8.40	30.24
39	8.84	31.84
40	10.84	39.03
41	7.46	26.88
42	8.61	31.01
43	6.73	24.19
44	8.61	31.01
45	13.44	48.39
46	10.84	39.03
47	9.60	34.56
48	9.08	32.69
49	9.60	34.56
50	9.08	32.69

Gaushala Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
51	8.11	29.19
52	6.98	25.11
53	8.34	30.00
54	8.83	31.76
55	6.83	24.55
56	10.35	37.24
57	10.35	37.24
58	6.39	22.97
59	11.54	41.54
60	6.83	24.55
61	9.67	34.84
62	10.71	38.58
63	6.25	22.50
64	8.11	29.19
65	6.83	24.55
66	7.90	28.43
67	6.13	22.04
68	6.00	21.60
69	8.58	30.86
70	7.69	27.69
71	9.09	32.72
72	10.35	37.24
73	13.64	49.09
74	11.54	41.54
75	9.09	32.72
76	6.66	24.00
77	6.25	22.50
78	7.90	28.43
79	7.31	26.34
80	7.31	26.34
81	6.25	22.50
82	6.83	24.55
83	6.53	23.48
84	6.66	24.00
85	7.31	26.34
86	7.90	28.43
87	6.39	22.97
88	5.89	21.17
89	5.36	19.29
90	6.25	22.50
91	5.89	21.17
92	5.45	19.64
93	6.53	23.48
94	5.55	20.00
95	6.53	23.48
96	6.39	22.97
97	7.31	26.34
98	6.98	25.11
99	9.67	34.84
100	8.01	28.85

Table 10: Car Speed Data(2)

Budanilakantha Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
101	8.83	31.75
102	9.88	35.53
103	7.45	26.80
104	10.75	38.71
105	9.03	32.48
106	10.04	36.14
107	8.88	31.93
108	10.74	38.66
109	9.60	34.54
110	12.40	44.64
111	8.66	31.19
112	7.56	27.21
113	8.63	31.08
114	7.33	26.38
115	7.98	28.70
116	8.63	31.08
117	8.40	30.24
118	7.69	27.66
119	9.13	32.88
120	10.40	37.45
121	9.00	32.40
122	7.25	26.10
123	9.40	33.85
124	6.45	23.24
125	8.05	28.99
126	9.85	35.44
127	11.89	42.79
128	10.16	36.59
129	9.69	34.89
130	11.05	39.79
131	9.35	33.66
132	11.35	40.86
133	10.50	37.80
134	10.33	37.18
135	8.93	32.14
136	11.70	42.11
137	11.89	42.79
138	8.40	30.24
139	7.69	27.66
140	8.63	31.08
141	7.98	28.70
142	10.76	38.75
143	9.56	34.41
144	8.19	29.49
145	10.41	37.49
146	11.58	41.66
147	10.60	38.14
148	9.08	32.66
149	11.25	40.50
150	9.41	33.89

Teaching Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
151	9.08	32.69
152	7.81	28.13
153	9.34	33.60
154	9.89	35.58
155	7.64	27.49
156	11.59	41.71
157	11.59	41.71
158	7.15	25.74
159	9.93	35.72
160	7.64	27.49
161	10.84	39.03
162	10.94	39.39
163	9.11	32.80
164	9.08	32.69
165	7.64	27.49
166	8.84	31.84
167	6.86	24.69
168	6.73	24.19
169	9.60	34.56
170	8.61	31.01
171	10.19	36.65
172	11.59	41.71
173	9.28	33.39
174	8.93	32.13
175	10.19	36.65
176	7.46	26.88
177	7.00	25.20
178	8.84	31.84
179	8.20	29.50
180	8.20	29.50
181	7.00	25.20
182	7.64	27.49
183	7.30	26.29
184	7.46	26.88
185	8.20	29.50
186	8.84	31.84
187	7.15	25.74
188	6.59	23.71
189	6.00	21.60
190	7.00	25.20
191	6.59	23.71
192	6.11	22.00
193	7.30	26.29
194	6.23	22.40
195	7.30	26.29
196	7.15	25.74
197	8.20	29.50
198	7.81	28.13
199	10.84	39.03
200	8.98	32.31

Table 11: Bus Speed Data(1)

Basundhara Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
1	8.86	31.90
2	11.25	40.53
3	8.47	30.50
4	7.79	28.03
5	7.29	26.25
6	8.11	29.20
7	8.00	28.80
8	12.00	43.20
9	9.76	35.15
10	9.14	32.91
11	9.14	32.91
12	11.08	39.88
13	9.60	34.56
14	9.00	32.40
15	8.94	32.19
16	11.08	39.88
17	12.00	43.20
18	10.29	37.03
19	10.86	39.13
20	7.80	28.09
21	11.53	41.48
22	9.44	33.99
23	8.25	29.73
24	8.60	30.95
25	8.35	30.05
26	11.60	41.75
27	9.25	33.32
28	9.93	35.75
29	8.73	31.41
30	8.00	28.80
31	9.44	33.99
32	8.35	30.05
33	6.26	22.54
34	9.44	33.99
35	8.53	30.68
36	8.53	30.68
37	10.40	37.43
38	8.11	29.20
39	10.66	38.40
40	9.44	33.99
41	8.11	29.20
42	9.60	34.56
43	10.66	38.40
44	9.76	35.15
45	6.48	23.30
46	10.29	37.03
47	6.55	23.56
48	8.35	30.05
49	8.00	28.80
50	8.36	30.09

Gaushala Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
51	7.90	28.43
52	6.53	23.48
53	8.65	31.15
54	9.00	32.40
55	7.50	27.00
56	7.50	27.00
57	10.00	36.00
58	5.40	19.44
59	7.95	28.63
60	9.30	33.48
61	6.90	24.84
62	8.04	28.93
63	6.53	23.48
64	7.50	27.00
65	9.58	34.46
66	9.00	32.40
67	6.53	23.48
68	9.00	32.40
69	9.00	32.40
70	7.04	25.31
71	12.86	46.29
72	6.43	23.15
73	5.85	21.04
74	7.04	25.31
75	10.71	38.58
76	9.58	34.46
77	11.84	42.64
78	8.49	30.56
79	7.90	28.43
80	11.54	41.54
81	8.34	30.00
82	7.14	25.71
83	12.50	45.00
84	8.19	29.46
85	7.14	25.71
86	9.58	34.46
87	10.22	36.81
88	7.50	27.00
89	8.04	28.93
90	13.64	49.09
91	8.34	30.00
92	6.53	23.48
93	9.58	34.46
94	8.19	29.46
95	9.48	34.09
96	11.25	40.50
97	8.83	31.76
98	6.34	22.81
99	5.43	19.51
100	7.14	25.71

Table 12: Bus Speed Data (2)

Budanilakantha Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
101	7.83	28.18
102	7.35	26.45
103	6.55	23.56
104	6.93	24.93
105	5.21	18.79
106	8.93	32.15
107	6.15	22.14
108	7.98	28.73
109	7.35	26.48
110	6.25	22.50
111	8.73	31.40
112	7.50	27.00
113	12.50	45.00
114	5.29	19.01
115	5.21	18.75
116	10.90	39.24
117	9.86	35.52
118	10.56	38.03
119	7.01	25.24
120	6.95	25.00
121	7.81	28.13
122	8.53	30.69
123	8.34	30.00
124	9.61	34.61
125	7.98	28.73

Teaching Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
126	6.41	23.08
127	9.26	33.34
128	7.58	27.28
129	5.14	18.49
130	5.86	21.09
131	9.61	34.61
132	5.86	21.09
133	9.38	33.75
134	5.60	20.15
135	12.50	45.00
136	11.72	42.19
137	15.00	54.00
138	8.73	31.40
139	10.41	37.50
140	3.95	14.21
141	5.29	19.01
142	6.19	22.25
143	5.60	20.15
144	4.94	17.76
145	7.08	25.48
146	6.35	22.89
147	7.08	25.48
148	5.36	19.29
149	9.38	33.75
150	5.86	21.09

Table 13: Carrier Speed Data(1)

Basundhara Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
1	9.64	34.71
2	9.15	32.95
3	11.49	41.36
4	11.25	40.50
5	10.19	36.68
6	9.56	34.41
7	10.39	37.39
8	7.30	26.28
9	7.20	25.93
10	6.35	22.86
11	9.28	33.38
12	13.50	48.60
13	11.74	42.26
14	11.43	41.11
15	8.01	28.86
16	10.80	38.88
17	10.59	38.11
18	10.83	38.96
19	8.26	29.74
20	7.63	27.44
21	10.00	36.00
22	8.18	29.45
23	9.11	32.81
24	9.40	33.84
25	8.34	30.03
26	10.76	38.75
27	10.03	36.08
28	9.95	35.83
29	6.61	23.80
30	9.13	32.85
31	8.20	29.53
32	8.76	31.53
33	8.00	28.80
34	10.03	36.10
35	9.49	34.15
36	10.33	37.16
37	9.61	34.62
38	7.89	28.40
39	8.89	31.99
40	9.00	32.40
41	8.66	31.20
42	8.89	31.99
43	13.00	46.80
44	9.00	32.40
45	10.98	39.49
46	8.35	30.09
47	10.18	36.63
48	10.18	36.63
49	8.26	29.72
50	7.55	27.18

Gaushala Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
51	8.22	29.60
52	8.91	32.09
53	7.80	28.09
54	8.46	30.46
55	8.60	30.99
56	6.24	22.46
57	8.76	31.53
58	6.84	24.63
59	6.75	24.29
60	9.21	33.17
61	9.25	33.29
62	11.35	40.84
63	9.25	33.29
64	7.56	27.23
65	7.70	27.74
66	6.94	24.96
67	12.58	45.29
68	9.08	32.65
69	8.48	30.53
70	9.48	34.10
71	9.03	32.49
72	9.71	34.95
73	7.07	25.46
74	8.74	31.45
75	8.45	30.44
76	4.46	16.05
77	4.88	17.55
78	5.49	19.78
79	4.76	17.13
80	5.58	20.06
81	5.28	18.98
82	5.65	20.35
83	5.83	20.96
84	6.29	22.65
85	6.29	22.65
86	6.61	23.80
87	6.10	21.94
88	7.80	28.08
89	5.20	18.72
90	9.26	33.34
91	10.00	36.00
92	10.00	36.00
93	10.41	37.50
94	10.88	39.14
95	11.20	40.30
96	9.39	33.81
97	11.96	43.06
98	10.15	36.54
99	9.13	32.84
100	8.18	29.41

Table 14: Carrier Speed Data (2)

Budanilakantha Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
101	5.39	19.40
102	8.68	31.25
103	7.56	27.21
104	8.09	29.10
105	10.19	36.69
106	5.94	21.36
107	8.09	29.10
108	8.68	31.25
109	10.65	38.35
110	7.69	27.66
111	9.19	33.09
112	9.01	32.45
113	7.44	26.79
114	12.03	43.26
115	5.58	20.05
116	9.54	34.33
117	8.04	28.93
118	10.61	38.20
119	10.61	38.20
120	7.70	27.74
121	7.81	28.13
122	4.89	17.61
123	5.99	21.54
124	9.52	34.28
125	7.43	26.75

Teaching Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
126	6.41	23.09
127	5.75	20.70
128	13.25	47.69
129	6.55	23.59
130	7.71	27.78
131	8.84	31.80
132	8.84	31.80
133	9.83	35.37
134	8.13	29.25
135	11.95	43.01
136	10.88	39.18
137	11.50	41.39
138	6.49	23.34
139	5.80	20.89
140	9.24	33.24
141	13.13	47.25
142	9.24	33.28
143	10.84	39.00
144	5.34	19.21
145	6.76	24.35
146	8.88	31.93
147	6.76	24.35
148	8.31	29.90
149	9.79	35.23
150	8.75	31.49

Table 15: Micro Bus Speed Data

Basundhara Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
1	8.68	31.25
2	9.98	35.90
3	11.78	42.40
4	10.38	37.36
5	6.79	24.46
6	8.36	30.10
7	8.52	30.68
8	8.85	31.84
9	11.16	40.18
10	7.44	26.79
11	7.21	25.96
12	8.91	32.11
13	10.19	36.69
14	10.90	39.25
15	11.72	42.19
16	7.10	25.57
17	7.21	25.96
18	11.16	40.18
19	7.95	28.60
20	9.01	32.45
21	8.52	30.68
22	8.09	29.10
23	7.69	27.66
24	8.68	31.25
25	10.41	37.50
26	7.10	25.57
27	7.69	27.66
28	7.63	27.45
29	8.52	30.68
30	8.22	29.60
31	7.10	25.57
32	10.19	36.69
33	11.44	41.16
34	10.41	37.50
35	7.81	28.13
36	9.38	33.75
37	7.21	25.96
38	9.91	35.68
39	9.98	35.90
40	12.34	44.40
41	7.21	25.96
42	7.33	26.36
43	7.81	28.13
44	5.04	18.15
45	5.04	18.15
46	4.84	17.40
47	7.56	27.21
48	6.70	24.11
49	6.51	23.44
50	7.76	27.94

Gaushala Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
51	8.22	29.60
52	5.15	18.55
53	6.25	22.50
54	8.22	29.60
55	8.85	31.84
56	7.95	28.60
57	8.22	29.60
58	6.51	23.44
59	6.09	21.91
60	8.52	30.68
61	8.52	30.68
62	6.25	22.50
63	7.21	25.96
64	13.03	46.88
65	9.76	35.15
66	8.52	30.68
67	7.10	25.57
68	6.25	22.50
69	5.79	20.84
70	7.33	26.36
71	7.81	28.13
72	7.81	28.13
73	6.70	24.11
74	5.45	19.63
75	8.22	29.60
76	8.68	31.25
77	8.52	30.68
78	6.89	24.81
79	6.70	24.11
80	7.10	25.57
81	5.33	19.17
82	6.42	23.11
83	6.60	23.78
84	6.70	24.11
85	9.01	32.45
86	5.65	20.33
87	7.44	26.79
88	5.71	20.57
89	8.68	31.25
90	8.85	31.84
91	10.65	38.35
92	7.81	28.13
93	8.85	31.84
94	6.25	22.50
95	6.60	23.78
96	7.44	26.79
97	6.16	22.20
98	6.38	22.95
99	8.48	30.50
100	11.14	40.10

Table 16: Truck Speed Data

Basundhara Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
1	6.76	24.35
2	7.69	27.66
3	8.68	31.25
4	8.53	30.68
5	6.96	25.08
6	9.38	33.75
7	8.21	29.57
8	6.34	22.80
9	9.19	33.09
10	6.25	22.50
11	8.35	30.04
12	11.09	39.90
13	7.44	26.76
14	9.26	33.35
15	10.51	37.85
16	7.21	25.96
17	7.69	27.66
18	12.03	43.26
19	9.69	34.86
20	6.51	23.44
21	8.44	30.38
22	10.90	39.25
23	7.05	25.39
24	8.23	29.60
25	10.04	36.13
26	8.53	30.68
27	6.25	22.50
28	11.54	41.55
29	9.20	33.11
30	7.81	28.11
31	10.53	37.88
32	6.34	22.79
33	6.51	23.46
34	7.10	25.58
35	7.34	26.41
36	8.21	29.56
37	8.21	29.58
38	10.45	37.64
39	7.21	25.96
40	4.69	16.88
41	7.95	28.60
42	9.19	33.09
43	7.10	25.58
44	6.51	23.44
45	9.76	35.15
46	11.72	42.19
47	4.65	16.72
48	8.69	31.28
49	10.25	36.89
50	8.20	29.51

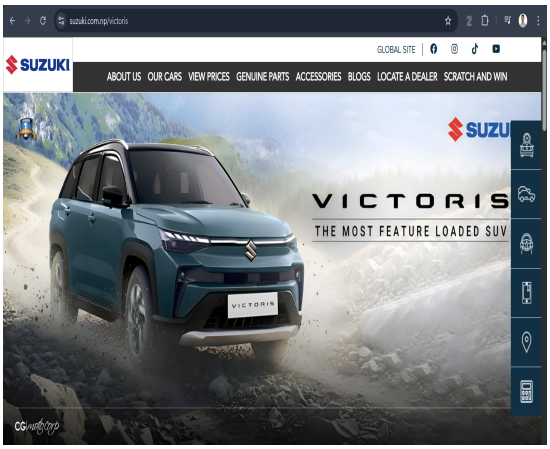
Gaushala Leg (Outgoing Edge)
11-16 JAN (2026)-7:00–12:30 PM

ID	Speed (m/s)	Speed (km/h)
51	10.55	37.97
52	7.30	26.26
53	10.55	37.96
54	10.55	37.96
55	9.78	35.19
56	7.56	27.22
57	9.84	35.40
58	8.29	29.85
59	10.23	36.83
60	4.99	17.96
61	6.96	25.08
62	5.28	18.99
63	10.25	36.91
64	9.08	32.67
65	6.76	24.34
66	7.36	26.49
67	5.83	20.99
68	6.88	24.75
69	5.39	19.39
70	7.10	25.55
71	9.46	34.08
72	6.27	22.60
73	5.38	19.36
74	7.29	26.25
75	10.85	39.08
76	5.76	20.76
77	5.13	18.45
78	11.54	41.53
79	10.25	36.91
80	13.23	47.61
81	6.36	22.91
82	9.74	35.04
83	7.71	27.75
84	10.76	38.74
85	8.49	30.56
86	10.55	37.97
87	12.30	44.29
88	8.36	30.11
89	7.76	27.95
90	9.71	34.97
91	10.23	36.83
92	7.76	27.95
93	7.81	28.13
94	7.46	26.85
95	6.78	24.41
96	8.18	29.45
97	9.51	34.25
98	7.89	28.38
99	6.84	24.61
100	6.90	24.84

A.5 Vehicle Dimensions Reference

The physical dimensions assigned to each vehicle type in the SUMO micro-simulation model correspond to representative vehicle models observed at Narayan Gopal Chowk. The images presented below illustrate these vehicle models for each class identified at the intersection, and their length, width, and height were extracted from official manufacturer and brand specification sheets.

(a) Car — front quarter view



(b) Car — side profile with length annotation

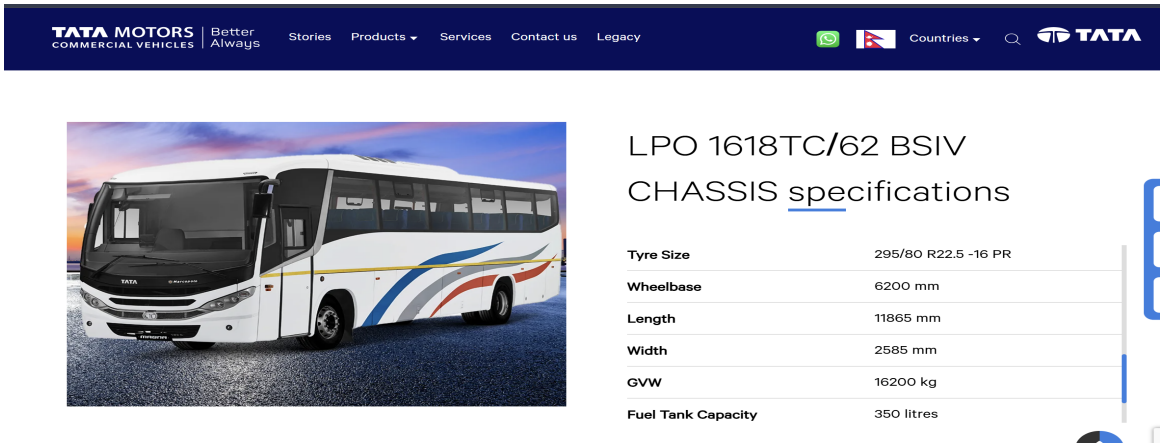
Wheelbase (mm)	2380
Height (mm)	1520
Width (mm)	1490
Length (mm)	3530

(c) Car manufacturer dimensions

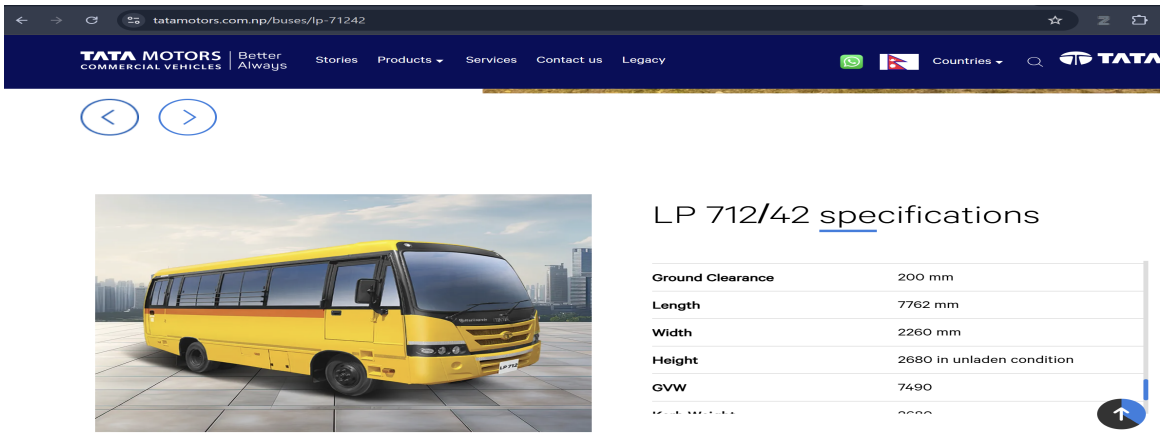
Length (mm)	4360
Height (mm)	1655
Width (mm)	1795
Wheelbase (mm)	2600

(d) Car manufacturer dimensions

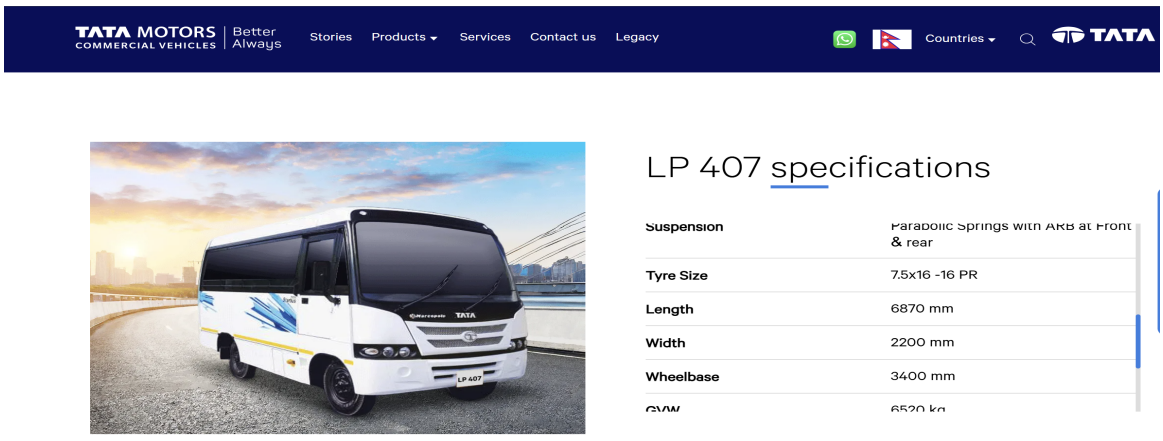
Figure 1: Category ’B’ vehicle Dimensions



(a) Bus — full side profile with length annotation

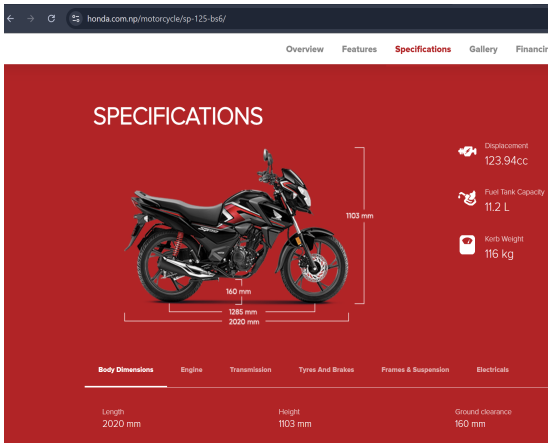


(b) Bus — front view with width annotation

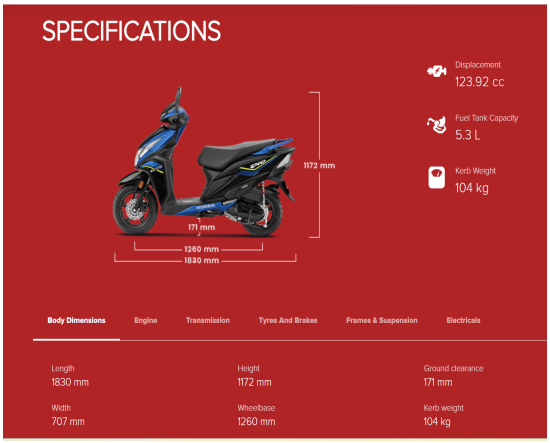


(c) Bus — scaled pixel reference against lane width

Figure 2: Category 'C' Vehicle Dimensions



(a) Motorcycle



(b) Scooter

Length (mm)	2020
Height (mm)	1103
Width (mm)	750

(c) Motorcycle manufacturer dimensions

Length (mm)	1832
Height (mm)	1172
Width (mm)	707

(d) Scooter manufacturer dimensions

Figure 3: Category 'A' Vehicle Dimensions

(a) Carrier

(b) Micro Bus

Length (mm)	2100
Height (mm)	1861
Width (mm)	1500

(c) Carrier manufacturer dimensions

Length (mm)	5380
Height (mm)	2285
Width (mm)	1880

(d) Hiace manufacturer dimensions

Figure 4: Category 'B' Vehicle Dimensions

ANNEX B: CALIBRATION AND VALIDATION RESULTS

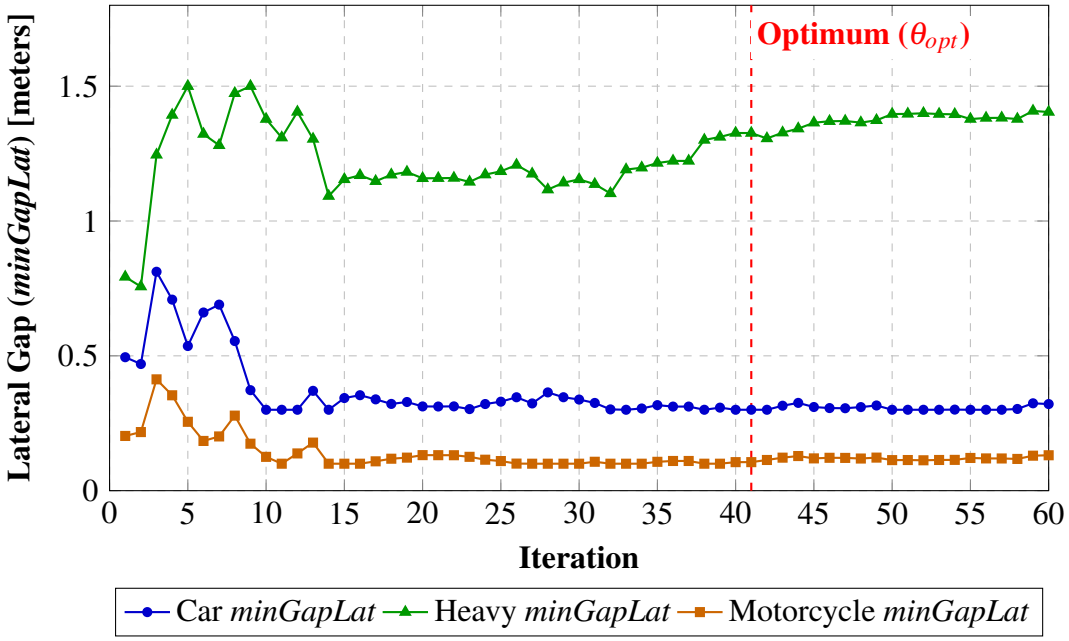


Figure 5: Trajectory of LatGap Parameter for different Vehicle Class

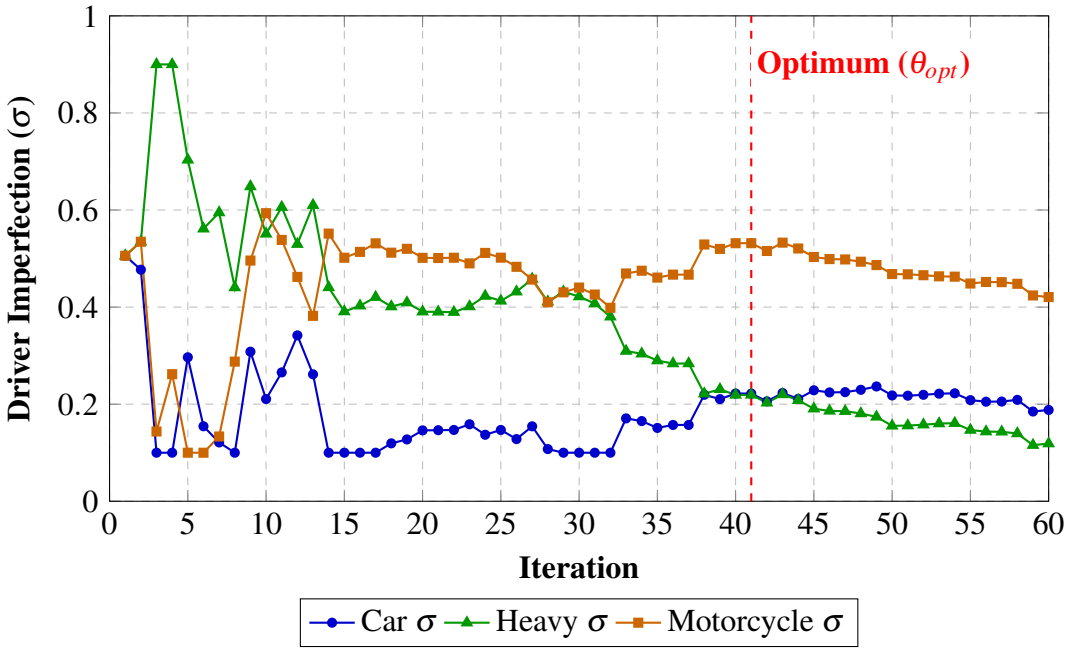


Figure 6: Trajectory of Sigma Parameter for different Vehicle Class

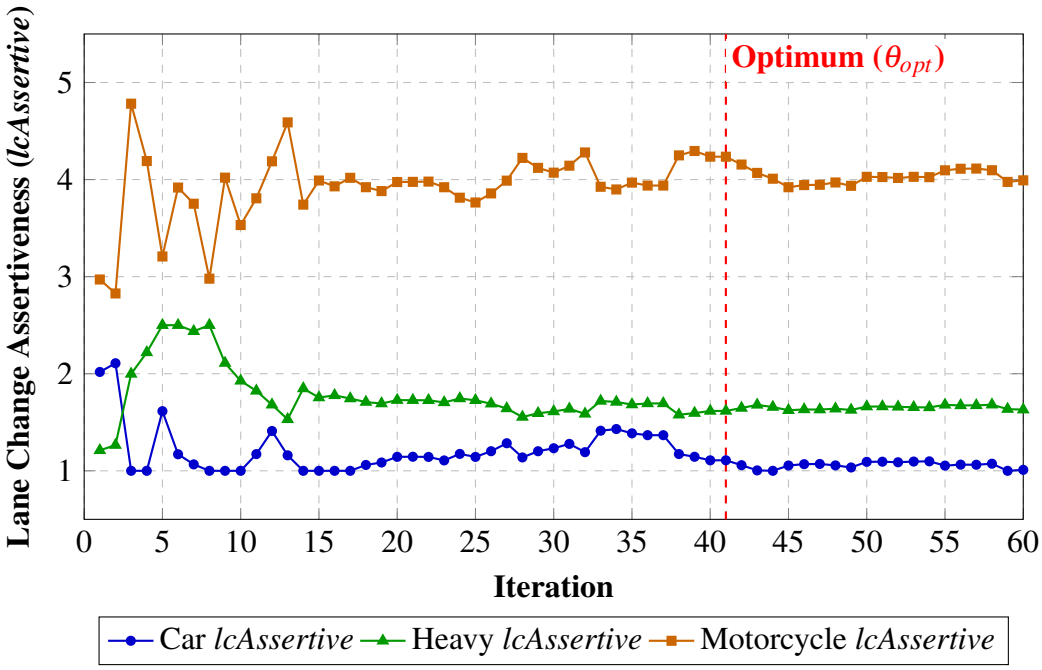


Figure 7: Trajectory of lcAssertive Parameter for different Vehicle Class