

A SIMPLE FRAMEWORK TOWARDS VISION-BASED TRAFFIC SIGNAL CONTROL WITH MICROSCOPIC SIMULATION

Pan He

Department of Computer Science and Software Engineering
Auburn University, Auburn, AL 36849, USA
pan.he@auburn.edu

Quanyi Li

Senior Applied Scientist
Oxa, Oxford Business Park North Oxford, OX4 2FL, UK
quanyili0057@gmail.com

Xiaoyong (Brian) Yuan

Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634, USA
xiaoyon@clemson.edu

Bolei Zhou

Department of Computer Science
University of California, Los Angeles (UCLA), Los Angeles, CA 90095, USA
bolei@cs.ucla.edu

Word Count: 5160 words + 1 table(s) $\times$ 250 = 5410 words

Submission Date: November 27, 2025

ABSTRACT

Traffic signal control (TSC) is crucial for reducing traffic congestion leading to smoother traffic flow, reduced idle time, and mitigated CO2 emissions. In this paper, we explore the computer vision approach for TSC that modulates on-road traffic flows through visual observation. Unlike traditional feature-based approaches, vision-based methods depend much less on heuristics and predefined features, bringing promising potentials for end-to-end learning and optimization of traffic signals. Thus, we introduce a simple traffic simulation framework called TrafficDojo towards vision-based TSC and its benchmark by integrating the microscopic traffic flow provided in SUMO into the 3D driving simulator MetaDrive. This proposed framework offers a versatile traffic environment for in-depth analysis and comprehensive evaluation of traffic signal controllers across diverse traffic conditions and scenarios. We establish and compare baseline algorithms including both traditional and Reinforcement Learning (RL) approaches. This work sheds light on the design and development of vision-based TSC approaches and opens up new research opportunities¹.

Keywords: Traffic Signal Control; Reinforcement Learning; Traffic Simulation; Deep Learning

¹All the code and baselines have been made publicly available at https://panhe.org/projects/traffic_dojo.html

INTRODUCTION

As per the 2022 INRIX Global Traffic Scorecard annual report (*1*), a typical American driver experienced 51 hours in traffic congestion in 2022, resulting in the monetary loss of approximately \$869 in wasted time. A key strategy for mitigating traffic congestion is to develop new Traffic Signal Control (TSC) algorithms that can effectively coordinate traffic movements at intersections.

Traffic signals are typically controlled by fixed-time, actuated, or adaptive methods. Generally, these traditional methods must be specifically configured or tuned by experts for regions with various traffic conditions, which is labor-intensive and limits its large-scale application. Recently, Reinforcement Learning (RL) has been investigated as a method to optimize adaptive TSC policies for intersections. By learning from scratch through interaction with dynamic traffic environments, RL has demonstrated superior performance compared to traditional approaches (*2–4*). However, the current study of RL-based methods remains insufficient, as they operate under the strong assumption that all high-level features and observations—such as queue length, waiting time, volume, delay, speed, vehicle positions, phase duration, and congestion—are readily available (*5*). These high-level heuristics are often difficult to extract from raw sensor inputs. Connected and Autonomous Vehicles (CAVs), leveraging Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, can enhance TSC by providing heuristic information on the speed and location of surrounding vehicles, thereby reducing reliance on camera-based systems. However, they (*6, 7*) are unsuitable for environments where human-driven vehicles dominate. Such scenarios necessitate cameras for sensing and perception algorithms to detect and respond to these vehicles accurately. It is fair to assert that RL-based TSC needs further exploration to determine if its policy remains robust when operating with noisy high-level features derived from cameras.

Additionally, manually engineered high-level features may not comprehensively represent traffic scenarios or align with specific reward designs, hindering RL-based approaches from identifying the optimal solution. In contrast, the end-to-end policy that directly takes raw images as input has the potential to learn a better feature extractor and boost the performance of the TSC policy. Unfortunately, there is a limited body of work exploring vision-based TSC methods, and many existing studies focus on training TSC policies using simplified or toy top-down snapshots (*3, 8*). Moreover, the 2D nature of the simulation limits the ability to study multiview camera setups in a 3D environment. This paper aims to explore end-to-end TSC policy with visual inputs.

Although these aforementioned questions are essential for accelerating the research and deployment of vision-based TSC methods, we cannot tackle them using existing research infrastructures or traffic simulators. Popular traffic simulators, such as Flow (*11*), Simulation of Urban MObility (SUMO) (*12*), CityFlow (*13*), RESCO (*14*), and CBLab (*15*), have been introduced to simulate diverse traffic scenarios. However, none of them support sensor simulation and thus can not serve as the test-bed to investigate the impact of the high-level feature estimation for vision-based TSC methods. A potential solution is to use CARLA (*16*) with the SUMO bridge where the images are rendered by CARLA and traffic flows are generated by SUMO. Nevertheless, this solution is not efficient because CARLA is too slow to render images to train RL algorithms. As a result, it may take several days or even weeks to collect enough samples for training a TSC policy.

In this work, we develop an open-source traffic simulation framework, TrafficDojo, for fast-prototyping and comparing TSC algorithms. Our TrafficDojo seamlessly integrates MetaDrive (*17*) and SUMO simulators into a unified simulation system where traffic is controlled by SUMO while MetaDrive serves as a rendering engine that provides multi-view visual observations. Compared with CARLA (*18*), which achieves roughly 45 FPS with a single camera on low-quality settings

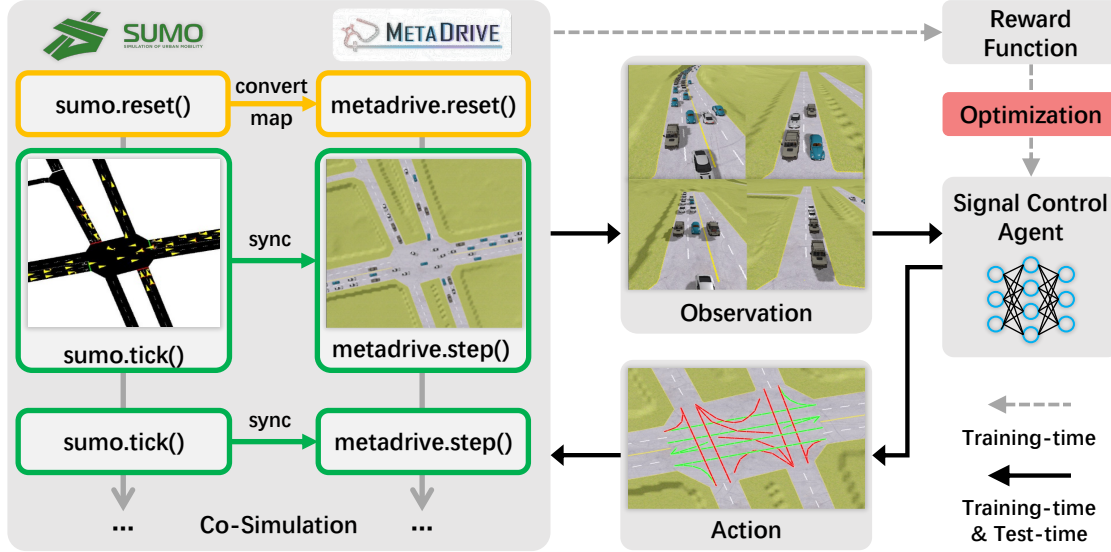


FIGURE 1: Overview of TrafficDojo. TrafficDojo supports generating rich 3D visual scenarios from SUMO maps, leveraged by a visual rendering engine of MetaDrive. At each time step, TrafficDojo implements a synchronization mechanism to direct the synchronous creation, updating, and removal of vehicles and pedestrians between SUMO and MetaDrive. TrafficDojo thus provides a Gym interactive environment tailored for traffic signal control, equipped with the capability to capture visual data from sensors like RGB cameras and LIDARs positioned at a traffic intersection. Additionally, it directly connects to the common RL training platforms such as Stable-Baseline3 (9) and RLLib (10) where a wide range of RL algorithms can be evaluated.

in a moderately complex scene, MetaDrive provides a more favorable balance between rendering fidelity and simulation efficiency. Notably, MetaDrive can sustain up to 300 FPS on a standard PC while maintaining high-quality visuals and a lightweight footprint of only about 100 MB per instance (17). Also, it is lightweight and can be easily parallelized. To facilitate the applicability of standard RL libraries, we follow the standard OpenAI Gym interface and provide a TSC environment that enables capturing visual data from RGB cameras. The system is programmed in Python and will be open-sourced.

In summary, the main contributions of the paper are:

- We develop TrafficDojo, a traffic signal simulation framework that bridges MetaDrive with microscopic simulation and facilitates the research of vision-based TSC policies. The modular-based design enables customizable configurations of visual sensors and map topologies.
- We benchmark various vision-based adaptive TSC approaches across two standard traffic scenarios and offer a comprehensive set of evaluation metrics reflecting quality of service and environmental impact.
- We investigate the vision-based adaptive TSC approaches and provide valuable insights into their model design and development and demonstrate that using a pre-trained foundational model as a feature extractor for vision-based TSC approaches is highly effective, surpassing traditional feature-based methods.

Experimental results highlight the promising potential of end-to-end learning and control

for vision-based TSC in simulated environments. We hope this approach will open up new research opportunities for both the reinforcement learning and transportation communities.

RELATED WORK

In this section, we delve into recent research that aligns with our work, specifically focusing on traffic signal control, simulation environments, and road network generation and descriptions.

Traffic Signal Control

TSC strategies, categorized into fixed-time, actuated, and adaptive control strategies (5), aim to optimize signal timings for efficient traffic flow. Fixed-time methods (19) rely on historical data to establish predetermined signal control sequences, assuming stable traffic conditions over a specific period, typically the past five minutes. Actuated signal methods adapt to real-time traffic data from sensors like pressure plates and loop detectors, via determining phase and duration based on current demand and follow predefined logical rules. Adaptive methods (20) optimize signal timing and excel at automatically adjusting to unexpected traffic changes by monitoring and addressing performance gaps.

Recent advancements in TSC leverage RL methods that adapt to real-time traffic conditions through trial-and-error (3, 8, 21), unlike non-RL approaches relying on specific assumptions. Traditional RL methods often use low-dimensional, hand-crafted features as state representations, limiting performance (5). RL-based methods can generally be categorized by reward, state, and action designs. Typical rewards include queue length, waiting time, change of delay, speed, number of stops, throughput, frequency of change, accident avoidance, and pressure (5). Many state representations have been employed to describe the traffic conditions, such as queue length, waiting time, volume, delay, speed, positions of vehicles, phase duration, and congestion. In terms of action designs, several common choices exist, such as establishing the current phase duration (22), determining the ratio of the phase duration relative to a predetermined total cycle duration (23), transitioning to the next phase in a predetermined cyclic phase sequence (8, 24, 25), and choosing the phase to transition to from a set of available phases (26, 27). We refer the readers to (28) for a comprehensive overview of RL algorithms. However, recent deep RLs have made processing high-dimensional input data like images feasible, offering new TSC possibilities. This paper uses visual data to compare RL-based TSC algorithms against traditional controllers.

Simulation Environments

Traffic simulation includes three main categories: macroscopic, microscopic, and mesoscopic. Macroscopic approaches describe traffic flow at an aggregate level, focusing on parameters like traffic speed, flow, and density and large-scale road simulation. Microscopic approaches model individual road users, such as vehicles, and their interactions with the environment, including acceleration, braking, lane-changing, and traffic signals. Traffic3D (29) leverages Unity's graphical and physics engines but is limited to a narrow range of road traffic scenarios. Sumounity (30), a concurrent work, integrates SUMO with Unity to enhance traffic simulation capabilities. SUMO, a popular microscopic simulator, supports 2D simulations of vehicles, public transport, and pedestrians, with extensions like CityFlow (13) and FLOW (11) enabling diverse traffic scenarios. SUMO also allows co-simulation with 3D platforms like CARLA (16) and Unity 3D. We focus on SUMO for its ability to simulate long-term traffic scenarios, despite its limitation in generating synthetic sensor data. Mesoscopic approaches are intermediate approaches bridging macroscopic and mi-

croscopic. We refer the readers to (31) for a comprehensive review. In this paper, we aim at developing the benchmark and library for vision-based TSC.

Road Network Generation and Descriptions

Effectively acquiring and modeling road networks is a crucial yet challenging task in traffic simulation. While digital representations of real-world road networks are increasingly available, their direct integration in simulations often require multi-step procedure and refinements, requiring domain expertise for seamless incorporation. CityEngine (32) proposes a procedural modeling approach in the generation of 3D urban environments. MITSIM describes the semantics of a road network using terms *node*, *link*, *segment*, and *lane*. VISSIM (33) introduces *link* and *connector* to represent the topology of a road network. OpenSCENARIO defines an open standard for describing the dynamic content of driving scenarios. OpenDRIVE (34) standardizes the logical road description between different driving simulators. OpenCRG (35) is developed to describe 3D geometries such as inclination, pitch, and yaw angles of the roads. CommonRoad (36) leverages lanelets (37) to describe road networks. SCENIC (38) introduces a new probabilistic programming language for describing driving scenarios. Using procedural generation and importing real-world data, MetaDrive (17) can create an infinite number of scenarios. A comprehensive surveys is suggested in (39). In this paper, we utilize MetaDrive’s scenario definitions to represent traffic scenarios. Additionally, we integrate MetaDrive as the rendering engine due to its lightweight nature and efficiency.

PRELIMINARIES

In this section, we provide the preliminary knowledge about traffic signal control tasks.

Terminologies

We use a standard intersection with four approaches as an illustrative example to demonstrate the key concepts. These concepts can be extended to intersections with various topological structures, such as varying numbers of approaches and skewed intersections.

- **Approaches** refer to the road segments or lanes directing traffic into or out of an intersection, specifying the distinct directions from which road users approach or depart. In a typical intersection, there are four incoming approaches, namely North, South, West, and East. Accordingly, there are outgoing approaches on which vehicles can leave the intersections.
- **Traffic Movement** describes the various ways in which road users navigate through an intersection. Each movement corresponds to a specific path or direction a vehicle, pedestrian, or cyclist can take within the intersection. Some examples of traffic movements include a left turn, right turn, through movement, or U-turn.
- **Traffic Signals**, commonly known as traffic lights, serves as mechanisms for regulating traffic flows at intersections. They determine when specific traffic movements are permitted or required to stop, guaranteeing the safe and efficient movement of vehicles and pedestrians. The standard components include red, green, and yellow lights.
- **Phases** describes the distinct operations of a signalized intersection, equipped with multiple lanes for incoming and outgoing roads. Each phase represents a specific combination of traffic movements. Phases are identified as “in conflict” when simultaneous assignment is not feasible, often due to intersecting traffic movements. A signal controller

is responsible for assigning non-conflicting phase sequences to optimize the long-term objective function. We direct readers to (5) for illustrative examples depicting both non-conflicting and conflicting phases, and commonly used phases.

- **Green Time**, also referred to as the duration of the green phase, designates the period during which traffic can safely pass through the intersection. The actual duration of this phase is determined by the traffic conditions in the studied area and the space available for vehicles waiting in the queues. To ensure the safety of vehicles and pedestrians, it is critical to minimize travel time and simultaneously prevent any skipping of the green phase. Therefore, there is often a minimum green time associated to an intersection.

Traffic Signal Control as Markov Decision Process

Reinforcement Learning trains an agent how to act to maximize a given numerical reward function. In our TSC setting, RL will train the agent to map situations to a favorable action to control the traffic light. It is modeled as a Markov Decision Process (MDP) defined as $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}_a, \mathcal{R}, \gamma\}$, where $\mathcal{S}$ is the *state space*, $\mathcal{A}$ is the *action space*, $\mathcal{P}_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ defines the probability of transitioning from state s at time t to state s' at time $t + 1$ given action a . $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the *reward function*, and $\gamma \in [0, 1]$ is a discount factor. At each timestep t , an agent relies on a state $s_t \in \mathcal{S}$ to take an action $a_t \in \mathcal{A}$, which is driven by a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The next state s_{t+1} is then determined by the transition probability $\Pr(s_{t+1} \mid s_t, a_t)$ while the agent received a reward $r(s_t, a_t) \in \mathcal{R}$. To learn the decision-making process, we leverage the accumulated award $G_T = \sum_{t=0}^T \gamma^t r(s_t, a_t)$ to find the optimal policy π^* such that G_T is maximized. Popular RL approaches generally include value-based, policy-gradient, and actor-critic approaches.

The TSC can be formulated as an MDP using RL approaches. The *State Space* $\mathcal{S}$ describes the state of incoming traffic and current phase information. Specifically, the state of incoming traffic is determined by the sensing capability and ranges that are assumed in the environment. Our TSC agent representing the state via directly mapping RGB images (with a resolution of 256×256) to actions for controlling the traffic signal. This leads to an end-to-end learning, that might eliminate the need for pre-specification of traffic environment features, such as vehicle density and queue length. At each timestep, the agent chooses an action, i.e., changing to a signaling phase for assigning right-of-passage (green light), from the *Action Space* $\mathcal{A}$. Figure 2 illustrates a visual example of observations and actions. A mandatory yellow phase is enforced for a specified duration if the chosen phase differs from the current phase, so as to obey the the real-world phase transition rules. The *Transition Function* $\mathcal{P}_a$ is determined by traffic progression models for simulated environments, i.e., MetaDrive and SUMO. The *Reward Function* $\mathcal{R}$ reflects the objectives or interests of traffic agents. In this paper, we focus on optimizing traffic throughput and reducing travel time or delay of vehicles. Therefore, we adopt the popular diff waiting time as the reward, which measures the change of waiting time of vehicles (40).

OVERVIEW OF TRAFFICDOJO

In this section, we introduce an open-source traffic simulator, called TrafficDojo for developing and benchmarking vision-based adaptive TSC algorithms. We describe the key designs of TrafficDojo and introduce its executing logic for TSC tasks.

Overview

Both SUMO and MetaDrive have been chosen as key simulation tools within our TrafficDojo framework. MetaDrive, specifically, stands out as a lightweight and flexible framework built on Panda3D. It offers a precise physics simulation and supports a range of sensory inputs. Notable is MetaDrive’s remarkable balance between visual rendering and physical simulation, achieving up to 300 FPS on a standard PC with a single instance of 100 MB in size (17). TrafficDojo seamlessly integrates MetaDrive, leveraging its distinctive features while maintaining comparable simulation speed. However, MetaDrive exhibits limitations in its ability to comprehensively analyze diverse traffic flow conditions, primarily due to its original design focusing on short-term self-driving scenarios from the perspective of the ego vehicle. Hence, we have seamlessly integrated SUMO (12), an open source, highly portable, microscopic traffic simulation platform, into our TrafficDojo framework. This choice is attributed to SUMO’s exceptional capability in efficiently managing large-scale networks and accurately simulating realistic traffic flows. Significantly, SUMO utilizes the TraCI (Traffic Control Interface) API for dynamic traffic modeling and quick customization of traffic scenarios.

We make efforts in bridging MetaDrive with SUMO for large-scale traffic simulation. While CARLA offers co-simulation with SUMO, it sees limited adoption in the TSC community due to the intricacy of creating custom maps and using them to run simulations in CARLA. We offer a more practical and efficient choice for researchers to train and test their algorithms and protocols, focusing on the downstream TSC application. Specifically, our research developments equip TrafficDojo with several key features:

- a map manager to facilitate the seamless transformation of road network configurations and the support of importing maps and managing traffic flow.
- a synchronization manager to direct the synchronous creation, updating, and removal of vehicles and pedestrians between SUMO and MetaDrive.
- a traffic signal agent manager takes charge of traffic signal controllers at intersections, responsible for retrieving critical traffic information and making transitions between traffic phases. It also provides an interface to adjust the behaviors and actions of traffic controllers, facilitating the exploration of diverse TSC approaches.
- an observer function linked to traffic signals, capturing visual information from a specific viewpoint. This functionality enhances data collection capabilities, providing valuable insights into traffic dynamics and contributing to a more comprehensive understanding of the traffic environment through the utilization of visual data.

Observing the Traffic Situation

Our framework allows flexible sensor placement (e.g., LIDAR, RGB cameras, depth cameras) and fine-tuning of parameters like field of view and mounting height. It also features an observer function that dynamically adjusts sensors based on traffic signal location and surrounding environment. This enables the strategic placement of multiple cameras, each precisely directed towards the approaching lanes. By doing so, it effectively captures and monitors traffic from those lanes. The heights of street light poles vary depending on the area they serve, which in turn affects the mounting height of the cameras. According to (41), cameras are typically mounted at heights ranging from 30 to 60 feet (9 to 18 meters). Therefore, in this paper, we heuristically place each camera at a height of 10 meters and set its pitch angle to be -35° . Figure 2 presents exemplary illustrations of the multiview camera setup (left) and the corresponding bird’s-eye view (right).

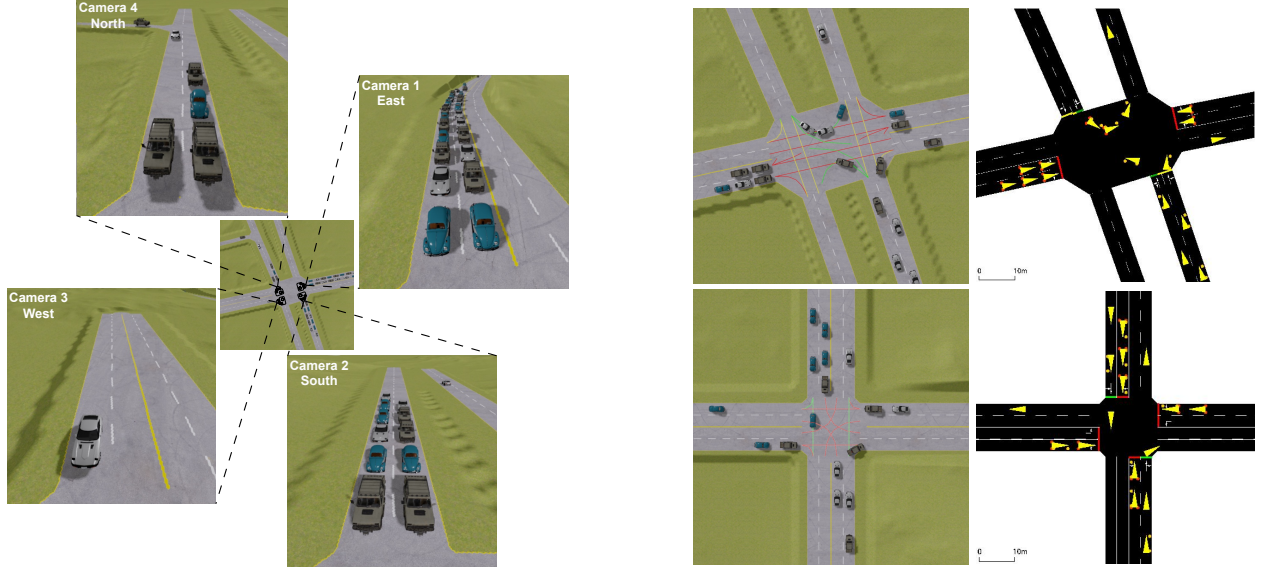


FIGURE 2: (Left) The multi-view observation setup at an intersection. It involves cameras positioned at a height of 10 meters, each focusing on the respective approaching lanes. The camera height can be adjusted to provide images in different views. (Right) The coordination of traffic signals and participants between MetaDrive and SUMO is achieved through synchronization. The status of traffic lights for specific links is represented by green, yellow, and red lines. The left image is the rendered image from MetaDrive and the right image is the SUMO visualization.

Generation of Road Networks

We provide the map manager to generate rich 3D scenarios from SUMO maps. It translates the SUMO format for scenario definitions into a unified description, providing precise geometric details for map objects such as driving roads, sidewalks, shoulders, lanes, and crossings. By harnessing OpenStreetMap data, which offers detailed road network models, including layouts, intersections, traffic rules, and geographical features, TrafficDojo enables the design and evaluation of TSC approaches using real-world geographical data. It's crucial to note that in cases where the conversion encounters errors, such as confusing connections or sidewalk overlapping, particularly in large-scale urban networks sourced from OpenStreetMap, manual refinement is still necessary via tools such as SUMO-GUI for customized maps.

Synchronization of Traffic Signals and Participants

Inspired by CARLA co-simulation, we developed a synchronization manager to efficiently handle the dynamic creation and removal of vehicles and pedestrians while seamlessly updating their positions between SUMO and MetaDrive. Furthermore, our framework provides an interface for controlling and synchronizing the status of traffic signals between both. A visual example is shown in Figure 2 (right). The traffic signal manager takes charge of controllers at intersections. At each fixed-time step, the controller selects a non-conflicting phase to receive the right-of-passage (green light). If the selected phases differ from the currently active ones, a mandatory yellow phase is triggered for a predefined duration, such as two seconds. Additionally, a minimum green time is ensured for all phases.

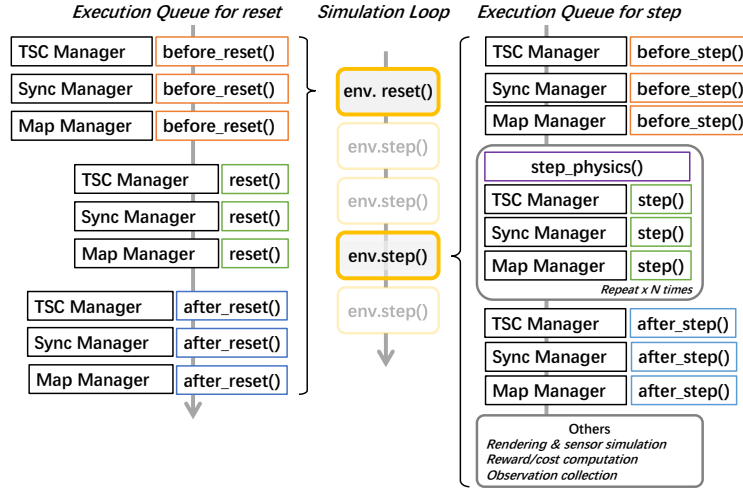


FIGURE 3: The execution logic of TrafficDojo. Multiple managers are registered and executed in specific orders for managing traffic signal controllers, synchronizations, and maps.

The Execution Logic

At every simulation step, TrafficDojo oversees the simulator’s status, managing static objects like maps and dynamic objects’ attributes such as positions, headings, and velocities of vehicles, along with traffic signal status. Figure 3 depicts TrafficDojo’s execution logic, with several defined key functions for each manager.

- *before_reset*. It involves the specification of essential preparation functions required before initializing the manager including tasks like cleaning up traffic objects and detaching the map from the scene.
- *reset*. This step generates objects according to some predefined rules such as initializing and attaching the map to the scene and initializing/restarting simulators.
- *before_step*. It is used to configure manager actions for tasks such as communication with SUMO and setting the light status in our specific context.
- *step*. This function preprocesses actions for specified agents, e.g., TSC agents, and performs the underlying physics simulation using the Bullet engine.
- *after_step*. It allows post-processing, calculation, and information collection after advancing the simulation.

It is important to note that the inclusion of each key step is optional and depends on the specific requirements of a given task. In our framework, managers serve as comprehensive controllers for almost everything, including maps, vehicles, pedestrians, traffic lights, etc.

EXPERIMENTS

In this section, we establish different baseline TSC controllers and compare their performance with feature- and vision-based TSC algorithms in different datasets.

Benchmark Algorithms

Baseline controllers: 1) *FixedTime* controls the traffic signal according to a predefined time duration and phase order, whether traffic is present or not. 2) *Self-Organizing Traffic Lights (SOTL)* (42), is a cycle-based dynamic phase length TSC based on self-organizing principles. 3) *MaxPres-*

sure (43) is designed to create an acyclic, dynamic phase length TSC, via modeling vehicles in lanes as a substance flowing through a pipe and implements control to maximize the relief of pressure between incoming and outgoing lanes.

RL controllers: 1) *Deep Q Network (DQN)* (44) is a powerful algorithm that combines deep neural networks with Q-learning to learn optimal policies in complex environments. 2) *A2C*, short for Advantage Actor Critic, presents a synchronous variant of the A3C method (45). Unlike A3C, A2C is synchronous and deterministic, waiting for each actor to complete its experience segment before updating, subsequently averaging across all actors. 3) *Proximal Policy Optimization (PPO)* (46) integrates concepts from A2C and TRPO.

We explore four distinct model variants. 1) *Feature-based RL approaches (DQN-F, PPO-F, A2C-F)* employ predefined features from SUMO, such as lane density and queue length. 2) *Vision-based RL approaches with multi-view cameras (DQN-MV, PPO-MV, A2C-MV)*: In contrast to feature-based variants, these models utilize multiple cameras to capture visual information, as shown in Figure 2. We limit our exploration to straightforward methods for fusing features from multi-view cameras with a direct concatenation operation. 3) *Vision-based RL approaches with a single BEV camera (DQN-BEV, PPO-BEV, A2C-BEV)*: Using a BEV camera to observe the traffic intersection. 4) *Vision-based RL approaches utilizing a single BEV camera and a frozen foundational model for feature extraction (DQN-BEV*, PPO-BEV*, A2C-BEV*)*: Addressing the challenges of learning directly from high-dimensional images (3), these models use a frozen foundational model for feature extraction. Inspired by CLIP (47), we replace feature extractors in DQN-BEV, PPO-BEV, and A2C-BEV with the pretrained ViT-H-14-quickgelu model from OpenCLIP (48). This enhancement builds policies on robust features, resulting in the modified DQN-BEV*, PPO-BEV*, and A2C-BEV*.

Evaluation Metrics

We evaluate TSC algorithm thoroughly using the following metrics: 1) Average travel time, calculating the average duration spent by each vehicle traveling the network including both waiting and actual travel time; 2) Throughput, normalized on an hourly basis, counts successfully completed trips; 3) Queue length, the sum of vehicles waiting on lanes; 4) Average delay, calculated as $1 - \frac{\sum_{i=1}^n v_i}{n \cdot v_{max}}$, characterizing the difference between mean vehicle speed and maximum speed limit, where n is the number of vehicles on the lane, v_i is the mean speed of vehicle i , and v_{max} is the maximum allowed speed; 5) Accumulated waiting time, the sum of vehicles' waiting time across lanes over a specific interval (e.g., 1000); 6) CO2 emissions, evaluating environmental impact by computing the total emitted CO2 by vehicles.

Experiment Setup

We evaluated two representative traffic scenarios: a synthetic two-way intersection and a real-world TAPAS Cologne intersection (49) with calibrated traffic demands. Simulations ran for 3,600 seconds, with our map manager converting road networks into suitable representations. Key settings included a 10-second minimum green time, a 5-second action interval, and a mandatory 2-second yellow phase before green transitions. All baseline models were implemented in our TrafficDojo. RGB images from multi-view and BEV sensors are configured with a resolution of 256×356 . Experiments were conducted on a server with two AMD EPYC Milan 7713 CPUs and four NVIDIA RTX A6000 GPUs. Results are based on the best-performing models selected from 200 training epochs, with evaluations averaged over five runs, reporting means and standard deviations. All

code and baseline implementations have been publicly released to help readers better understand the details of TrafficDojo.

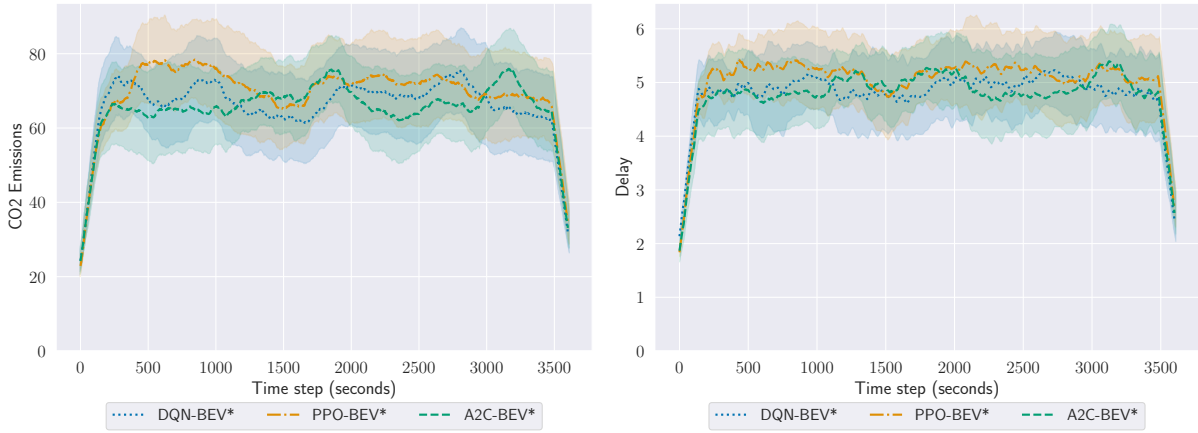


FIGURE 4: Evaluation of the trained policies from various RL controllers in the two-way single intersection, with a duration of 3,600 seconds. The curves are smoothed using a moving average with a window size of 50. **(Left)** Evaluation of CO2 emissions. **(Right)** Evaluation of average delay.

Performance Evaluation

We provide a summary of the experimental results shown in Table 1. When trained from scratch, vision-based RL approaches, especially DQN-BEV, PPO-BEV, A2C-BEV, are better or competitive compared to feature-based methods (DQN-F, PPO-F, A2C-F) in terms of performance metrics. However, current feature-based approaches depend on the precise extraction and analysis of aggregated traffic features, such as vehicle density and queue length, which hinders the development of end-to-end solutions. In contrast, vision-based approaches excel in end-to-end learning, despite the challenge of directly inferring these aggregated features from images.

More impressively, the use of a frozen foundational model as a feature extractor for vision-based TSC approaches significantly improves the performance of all baseline RL policies. Specifically, in the two-way single intersection scenario, DQN-BEV*, PPO-BEV*, and A2C-BEV* achieve CO2 emissions of 67.15, 70.02, and 66.35 g/sec, respectively. This represents a substantial reduction in emissions by 7.93%, 138.98% and 6.99% compared to the original DQN-BEV, PPO-BEV, and A2C-BEV models. Moreover, PPO-BEV* improves the model convergence of PPO-BEV by providing a robust feature representation; without this enhancement, PPO-BEV would perform poorly. All of these findings highlight the importance of discriminative features in learning superior RL policies, exemplified by the use of foundational models.

We observe a similar trend in the Cologne 1x1 scenario, reducing emissions from DQN-BEV (48.15), PPO-BEV (50.62) and A2C-BEV (51.00) to DQN-BEV* (46.27), PPO-BEV* (49.40) and A2C-BEV* (43.64), representing a relative reduction of 3.91%, 2.41%, and 14.43%, respectively. We observe consistent improvements in other metrics by transitioning from feature-based approaches to vision-based approaches.

TABLE 1: Comparison across different controllers using travel time, throughput, delay, accumulated waiting, queue (sec), and CO2 emission (g/sec). For each RL method, we **bold** the one with the best performance. For all baseline controllers and RL controllers with different feature extractors, we underline the method with the best performance.

Metric	Two-way single intersection					
	Travel Time↓	Throughput↑	Delay↓	Accumulated Waiting↓	Queue↓	CO2 Emissions↓
FixedTime	95.11 ±2.62	1879.80 ±18.74	5.57 ±0.08	19480.37 ±2417.03	60.28 ±2.29	190.01 ±7.11
SOTL	80.51 ±6.32	2429.40 ±29.58	5.88 ±0.13	5009.64 ±1672.86	45.86 ±6.27	151.31 ±16.53
MaxPressure	44.08 ±0.89	2484.60 ±2.30	4.86 ±0.10	345.40 ±24.23	11.20 ±0.37	56.54 ±1.27
DQN-F	52.35 ±0.62	2478.20 ±3.42	5.02 ±0.06	338.06 ±10.50	15.16 ±0.30	73.92 ±1.41
DQN-MV	76.13 ±5.73	2225.20 ±421.93	5.73 ±0.15	7646.32 ±12560.97	36.33 ±10.89	141.94 ±17.47
DQN-BEV	51.01 ±0.73	2475.00 ±2.00	5.07 ±0.04	334.35 ±6.96	14.88 ±0.24	72.93 ±0.96
DQN-BEV*	50.10 ±0.46	2481.40 ±3.21	4.88 ±0.08	282.07 ±13.08	13.37 ±0.50	67.15 ±1.85
PPO-F	49.75 ±0.76	2476.40 ±3.51	4.95 ±0.05	307.79 ±9.14	13.54 ±0.36	67.53 ±1.10
PPO-MV	100.88 ±0.57	2221.00 ±1.73	5.73 ±0.02	4255.39 ±24.30	47.42 ±0.27	171.21 ±0.92
PPO-BEV	100.01 ±0.55	2120.00 ±1.41	5.59 ±0.02	4356.62 ±19.99	46.59 ±0.33	167.33 ±1.05
PPO-BEV*	51.00 ±0.80	2477.20 ±3.27	5.12 ±0.05	337.32 ±7.30	14.26 ±0.21	70.02 ±1.03
A2C-F	50.66 ±1.20	2471.40 ±3.51	5.05 ±0.12	340.90 ±27.45	14.85 ±0.67	72.86 ±2.33
A2C-MV	98.30 ±0.35	2350.60 ±4.72	5.92 ±0.01	4172.72 ±24.62	48.79 ±0.10	180.39 ±0.30
A2C-BEV	50.87 ±0.39	2475.40 ±2.30	5.04 ±0.11	324.97 ±13.33	14.58 ±0.31	70.99 ±1.27
A2C-BEV*	48.67 ±0.64	2480.00 ±1.58	4.87 ±0.07	277.71 ±6.92	13.22 ±0.22	66.35 ±1.01
Metric	Cologne 1x1					
	Travel Time↓	Throughput↑	Delay↓	Accumulated Waiting↓	Queue↓	CO2 Emissions↓
FixedTime	165.47 ±3.33	1614.80 ±34.02	4.70 ±0.02	35008.23 ±2972.30	90.85 ±1.22	303.39 ±1.74
SOTL	99.20 ±6.20	1930.20 ±16.39	4.68 ±0.09	5667.02 ±1179.61	32.02 ±2.13	118.32 ±6.07
MaxPressure	44.12 ±0.40	1996.20 ±2.39	2.86 ±0.08	112.05 ±12.92	4.69 ±0.29	46.34 ±0.82
DQN-F	50.28 ±1.69	1994.80 ±2.17	3.12 ±0.07	299.14 ±19.96	6.26 ±0.26	50.77 ±1.02
DQN-MV	50.63 ±1.81	1981.60 ±10.19	3.41 ±0.11	389.67 ±90.84	8.55 ±0.76	57.89 ±2.24
DQN-BEV	44.78 ±1.43	1991.60 ±4.93	2.81 ±0.05	114.32 ±29.87	5.14 ±0.58	48.15 ±1.81
DQN-BEV*	43.83 ±0.72	1997.60 ±0.55	2.74 ±0.05	94.78 ±16.43	4.39 ±0.24	46.27 ±0.99
PPO-F	48.14 ±0.43	1997.00 ±0.00	2.97 ±0.02	156.14 ±17.06	5.73 ±0.18	49.72 ±0.76
PPO-MV	48.44 ±0.74	1997.00 ±0.00	3.01 ±0.03	206.09 ±24.93	6.65 ±0.21	52.63 ±0.64
PPO-BEV	47.71 ±0.55	1996.80 ±0.45	3.00 ±0.05	141.80 ±15.19	5.93 ±0.19	50.62 ±0.35
PPO-BEV*	47.19 ±0.32	1997.00 ±0.00	2.95 ±0.03	122.43 ±8.22	5.51 ±0.12	49.40 ±0.55
A2C-F	42.38 ±0.42	1996.60 ±0.89	2.56 ±0.01	85.57 ±8.50	3.88 ±0.02	44.72 ±0.19
A2C-MV	47.64 ±0.16	1997.00 ±0.71	3.03 ±0.02	160.98 ±14.74	6.03 ±0.17	50.65 ±0.50
A2C-BEV	47.90 ±0.27	1996.80 ±0.45	3.00 ±0.03	184.74 ±10.40	6.17 ±0.10	51.00 ±0.21
A2C-BEV*	43.30 ±0.60	1998.20 ±1.30	2.55 ±0.03	89.23 ±9.51	3.71 ±0.15	43.64 ±0.53

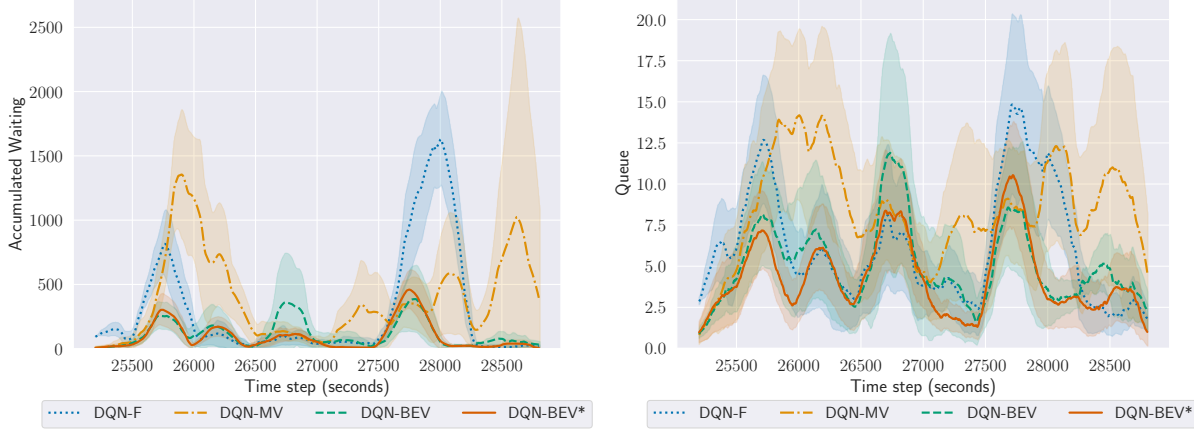


FIGURE 5: Evaluation of the trained policies from model variants of DQN in the Cologne 1x1 scenario, with a duration of 3,600 seconds and a start time of the 25,200_{th} second. The curves are smoothed using a moving average with a window size of 50. **(Left)** Evaluation of the accumulated waiting time. **(Right)** Evaluation of average queue length.

Ablation Study and Visualization

Comparison of models with BEV and multi-view camera observations. It is highlighted that vision-based RL approaches with a BEV camera demonstrate a clear advantage over multiview cameras. This observation is expected, given that our fusion of features from multiview cameras involves a straightforward concatenation that might ignore useful context information for coordinated control. Additionally, the poor performance of PPO-MV in the two-way single intersection scenario suggests a likely failure in achieving model convergence. We encourage researchers to explore more sophisticated fusion mechanisms to mitigate performance gaps between BEV and multi-view camera observations.

Visualizations of different RL controllers. We further analyzed and plotted the results of CO2 emissions and delay for policies trained by different RL controllers—specifically DQN-BEV*, PPO-BEV*, and A2C-BEV*—at the two-way single intersection. As depicted in Figure 4, when leveraging discriminative features from foundational models, A2C-BEV* demonstrates slightly superior performance compared to DQN-BEV* and PPO-BEV*. Both DQN-BEV* and A2C-BEV* adopt effective policies that promptly minimize traffic delays and CO2 emissions at critical time points, such as at 0, 900, 1800, and 2700 seconds, where traffic patterns change, leading to superior overall performance.

Visualizations of model variants. We evaluate the policies learned from various model variants of a single RL algorithm, using DQN as an example. Specifically, we compare DQN-F, DQN-MV, DQN-BEV, and DQN-BEV*. As shown in Figure 5, DQN-BEV* exhibits negligible fluctuations throughout the evaluation period. All model variants exhibit similar policies at certain timesteps (as evidenced by similar curves and trends), with superior models showing better performance at critical points to minimize sudden increases.

DISCUSSIONS AND CONCLUSIONS

Our current TrafficDojo has several limitations. Firstly, it exclusively supports and evaluates single-agent TSC tasks. However, it is highlighted that our framework is designed with flexi-

bility in mind, and future enhancements are planned to extend its capabilities to accommodate multi-agent TSC tasks where scenarios involve multiple traffic intersections. This incurs additional sensing and computational cost in capturing and processing visual data from the simulated environment. Furthermore, while transitioning from feature-based methods to vision-based methods contributes to a reduction in metrics such as traffic delay and CO2 emissions and facilitates progress towards end-to-end learning, it is imperative to acknowledge that the sim-to-real gap are still observable. Promising directions include works such as (50, 51).

In this paper, we have developed a holistic traffic simulation framework towards vision-based TSC with microscopic simulation. We have provided a benchmark for training and testing TSC approaches with comprehensive evaluation metrics. Experimental results highlight the promising potential of the end-to-end learning and control for vision-based TSC. It opens up new research opportunities for both reinforcement learning and transportation communities.

ACKNOWLEDGMENTS

We thank the Panda3D community and the developers of open-source projects such as SUMO-RL and RenderPipeline for their valuable support. This research is supported by the ORAU Ralph E. Powe Junior Faculty Enhancement Award, the NAIRR Pilot Project (NAIRR250417), and Lambda’s Research Grant Program.

REFERENCES

1. Pishue, B., 2022 – INRIX Global Traffic Scorecard. In *INRIX (December 2021)*, 2022.
2. Mei, H., X. L. Lei, L. Da, B. Shi, and H. Wei, *LibSignal: An Open Library for Traffic Signal Control*, 2022.
3. Tran, T. V. and M. Sartipi, Revisiting Pixel-based Traffic Signal Controls using Reinforcement Learning with World Models. In *The Workshop on Artificial Intelligence for Social Good at The 37th AAAI conference on artificial intelligence*, 2023.
4. Islam, Z., M. Abdel-Aty, A. Goswamy, A. Abdelraouf, and O. Zheng, Effect of signal timing on vehicles' near misses at intersections. *Scientific reports*, Vol. 13, No. 1, 2023, p. 9065.
5. Wei, H., G. Zheng, V. Gayah, and Z. Li, A Survey on Traffic Signal Control Methods. *arXiv preprint arXiv:1904.08117*, 2019.
6. Chen, H. and T. Z. Qiu, Distributed dynamic route guidance and signal control for mobile edge computing-enhanced connected vehicle environment. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 8, 2021, pp. 12251–12262.
7. Song, L., *Impacts of Connected and Autonomous Vehicles on Deep Reinforcement Learning Controlled Intersection Systems*. Ph.D. thesis, The University of North Carolina at Charlotte, 2022.
8. Van der Pol, E. and F. A. Oliehoek, Coordinated Deep Reinforcement Learners for Traffic Light Control. *Proceedings of learning, inference and control of multi-agent systems (at NIPS 2016)*, Vol. 8, 2016, pp. 21–38.
9. Raffin, A., A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, Stable-Baselines3: Reliable Reinforcement Learning Implementations. *The Journal of Machine Learning Research*, Vol. 22, No. 1, 2021, pp. 12348–12355.
10. Liang, E., R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, RLLib: Abstractions for Distributed Reinforcement Learning. In *International conference on machine learning*, PMLR, 2018, pp. 3053–3062.
11. Wu, C., A. Kreidieh, K. Parvate, E. Vinitzky, and A. M. Bayen, Flow: Architecture and Benchmarking for Reinforcement Learning in Traffic Control. *arXiv preprint arXiv:1710.05465*, Vol. 10, 2017.
12. Lopez, P. A., M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, Microscopic Traffic Simulation using SUMO. In *2018 21st international conference on intelligent transportation systems (ITSC)*, IEEE, 2018, pp. 2575–2582.
13. Zhang, H., S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, and Z. Li, CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. In *The world wide web conference*, 2019, pp. 3620–3624.
14. Ault, J. and G. Sharon, Reinforcement Learning Benchmarks for Traffic Signal Control. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.
15. Liang, C., Z. Huang, Y. Liu, Z. Liu, G. Zheng, H. Shi, K. Wu, Y. Du, F. Li, and Z. J. Li, CBLab: Supporting the Training of Large-scale Traffic Control Policies with Scalable Traffic Simulation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4449–4460.

16. Dosovitskiy, A., G. Ros, F. Codevilla, A. Lopez, and V. Koltun, CARLA: An Open Urban Driving Simulator. In *Conference on robot learning*, PMLR, 2017, pp. 1–16.
17. Li, Q., Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, MetaDrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 45, No. 3, 2022, pp. 3461–3475.
18. CARLA Team, *CARLA 0.8.0 Release*. <https://carla.org/2018/03/27/release-0.8.0/>, 2018, accessed: 2025-11-20.
19. Miller, A. J., Settings for Fixed-Cycle Traffic Signals. *Journal of the Operational Research Society*, Vol. 14, No. 4, 1963, pp. 373–386.
20. Hunt, P., D. Robertson, R. Bretherton, and R. Winton, SCOOT-a Traffic Responsive Method of Coordinating Signals. *Transport and Road Research Lab., Crowthorne, UK*, 1981.
21. Ducrocq, R. 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, 2023, pp. 192–206.
22. Aslani, M., M. S. Mesgari, and M. Wiering, Adaptive Traffic Signal Control with Actor-critic Methods in a Real-world Traffic Network with Different Traffic Disruption Events. *Transportation Research Part C: Emerging Technologies*, Vol. 85, 2017, pp. 732–752.
23. Casas, N., Deep Deterministic Policy Gradient for Urban Traffic Light Control. *arXiv preprint arXiv:1703.09035*, 2017.
24. Pham, T. T., T. Brys, M. E. Taylor, T. Brys, M. M. Drugan, P. Bosman, M.-D. Cock, C. Lazar, L. Demarchi, D. Steenhoff, et al., Learning Coordinated Traffic Light Control. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-13)*, IEEE, 2013, Vol. 10, pp. 1196–1201.
25. Mannion, P., J. Duggan, and E. Howley, An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control. *Autonomic road transport support systems*, 2016, pp. 47–66.
26. Arel, I., C. Liu, T. Urbanik, and A. G. Kohls, Reinforcement Learning-based Multi-agent System for Network Traffic Signal Control. *IET Intelligent Transport Systems*, Vol. 4, No. 2, 2010, pp. 128–135.
27. Zheng, G., X. Zang, N. Xu, H. Wei, Z. Yu, V. Gayah, K. Xu, and Z. Li, Diagnosing Reinforcement Learning for Traffic Signal Control. *arXiv preprint arXiv:1905.04716*, 2019.
28. Arulkumaran, K., M. P. Deisenroth, M. Brundage, and A. A. Bharath, A Brief Survey of Deep Reinforcement Learning. *arXiv preprint arXiv:1708.05866*, 2017.
29. Garg, D., M. Chli, and G. Vogiatis, Traffic3d: A rich 3d-traffic environment to train intelligent agents. In *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part V 19*, Springer, 2019, pp. 749–755.
30. Pechinger, M. and J. Lindner, Sumonity: Bridging SUMO and Unity for Enhanced Traffic Simulation Experiences. In *SUMO Conference Proceedings*, 2024, Vol. 5, pp. 163–177.
31. Boxill, S. A. and L. Yu, An Evaluation of Traffic Simulation Models for Supporting ITS Development. *Houston, TX: Development Centre for Transportation Training and Research, Texas Southern University*, 2000.
32. Parish, Y. I. and P. Müller, Procedural Modeling of Cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 301–308.

33. Fellendorf, M., VISSIM: A microscopic Simulation Tool to Evaluate Actuated Signal Control including Bus Priority. In *64th Institute of transportation engineers annual meeting*, Springer, 1994, Vol. 32, pp. 1–9.
34. Dupuis, M., M. Strobl, and H. Grezlikowski, OpenDRIVE 2010 and Beyond – Status and Future of the de facto Standard for the Description of Road Networks. In *Proc. of the Driving Simulation Conference Europe*, 2010, pp. 231–242.
35. ASAM, *OpenCRG*. <https://www.asam.net/standards/detail/opencrg/>, 2023, [Online; accessed 1-December-2023].
36. Althoff, M., M. Koschi, and S. Manzing, CommonRoad: Composible Benchmarks for Motion Planning on Roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 719–726.
37. Bender, P., J. Ziegler, and C. Stiller, Lanelets: Efficient Map Representation for Autonomous Driving. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, IEEE, 2014, pp. 420–425.
38. Fremont, D. J., T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, Scenic: A Language for Scenario Specification and Scene Generation. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2019, pp. 63–78.
39. Ma, J., X. Che, Y. Li, and E. M.-K. Lai, Traffic Scenarios for Automated Vehicle Testing: A Review of Description Languages and Systems. *Machines*, Vol. 9, No. 12, 2021, p. 342.
40. Alegre, L. N., *SUMO-RL*. <https://github.com/LucasAlegre/sumo-rl>, 2019.
41. Fullerton, I. J., J. H. Kell, and M. K. Mills, *Traffic Detector Handbook*. United States. Federal Highway Administration, 1990.
42. Cools, S.-B., C. Gershenson, and B. D’Hooghe, Self-Organizing Traffic Lights: A Realistic Simulation. *Advances in applied self-organizing systems*, 2013, pp. 45–55.
43. Varaiya, P., The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections. In *Advances in dynamic network modeling in complex transportation systems*, Springer, 2013, pp. 27–66.
44. Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-Level Control Through Deep Reinforcement Learning. *nature*, Vol. 518, No. 7540, 2015, pp. 529–533.
45. Mnih, V., A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, Asynchronous Methods for Deep Reinforcement Learning. In *International conference on machine learning*, PMLR, 2016, pp. 1928–1937.
46. Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
47. Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision. In *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.
48. Ilharco, G., M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, *OpenCLIP*, 2021, if you use this software, please cite it as below.

49. Varschen, C. and P. Wagner, Mikroskopische Modellierung der personenverkehrsnachfrage auf Basis von Zeitverwendungstagebüchern. *Integrierte Mikro-Simulation von Raum-und Verkehrsentwicklung. Theorie, Konzepte, Modelle, Praxis*, Vol. 81, 2006, pp. 63–69.
50. Zhou, Y., M. Simon, Z. Peng, S. Mo, H. Zhu, M. Guo, and B. Zhou, SimGen: Simulator-conditioned Driving Scene Generation. *arXiv preprint arXiv:2406.09386*, 2024.
51. Da, L., M. Gao, H. Mei, and H. Wei, Prompt to Transfer: Sim-to-Real Transfer for Traffic Signal Control with Prompt Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, Vol. 38, pp. 82–90.