Autonomous Traffic Signal Control using Deep Reinforcement Learning

Deepeka Garg

A thesis submitted in fulfilment of the requirements

for the degree of Doctor of Philosophy

in the

Department of Computer Science

ASTON UNIVERSITY

MARCH, 2020

ASTON UNIVERSITY

Department of Computer Science

Autonomous Traffic Signal Control using Deep Reinforcement Learning

Deepeka Garg

Doctor of Philosophy, 2020

Abstract

Traffic signals provide one of the primary means to administer conflicting road traffic flows. The efficiency of road transportation systems significantly depends on signal operation. The state-of-the-art signal control strategies are unable to efficiently and autonomously adapt to changing traffic flow patterns. In this thesis, we present an autonomous traffic signal control system, in which each intersection independently computes effective signal regimes to optimize traffic flows through that intersection at all instants-based *solely* on *live* camera footage. Our signal control system is trained via Deep Reinforcement Learning (DRL). In recent years, DRL has emerged as a powerful paradigm for control optimization problems by autonomously discovering effective control policies. Our signal control agent perceives the traffic situation around an intersection through visual sensory data and continuously modifies the traffic signal regimes in real time, as per the changing traffic observations. The contributions of this thesis are summarised as; (1) A truly adaptive signal control agent, that effectively tailors its signal control decisions to changing traffic patterns and significantly outperforms the conventional signal control methods (both fixed and adaptive) in single and multi-intersection scenarios. (2) This thesis, for the first time, by using transfer learning, empirically demonstrates vision-based signal control agent's high generalizability and accelerated learning skills on newly-encountered traffic conditions (such as prioritizing the navigation of emergency vehicles, handling adverse weather and lighting conditions). (3) Additionally, this thesis presents the first application of attention-visualization to illustrate the interpretation of DRL agents' signal control decisions, while highlighting the benefits of using visual traffic data from CCTV cameras for signal control over the conventional traffic data collection methods such as induction loops.

Keywords: Deep Reinforcement Learning; Computer Vision; Intelligent Transportation System; Autonomous Signal Control.

Acknowledgements

The invaluable support of friends, family and colleagues made it possible for me to complete this work and I am sincerely grateful to all of them.

First, I would like to express my gratitude to my supervisors; Dr Maria Chli and Dr George Vogiatzis for their guidance and their precious time that they invested in me throughout this journey.

Second, I would like to thank my lab-mates/friends for all the fun get-together and fruitful discussions. A special thanks to Noa, Thomas, Arezoo and Aamir.

Also, I would like to thank my qualifying report examiners; Dr Luca Rossi and Dr Elizabeth Wanner, and my final PhD examiners; Dr Yordon Raykov and Dr N. Kemal Ure for their time and helpful advice.

It would be unfair to not acknowledge my husband Akshat Agarwal, who supported me unconditionally during the last two years of my PhD in many different ways; from discussing new ideas to just being there listening and understanding. This work would not have been the same without him.

Finally, I would like to thank my parents and siblings for their constant support. Especially, my mom for coming to stay with me when things were difficult in the UK.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Traffic management is a critical task with significant economic and environmental repercussions. Urbanization and motorization have caused an imbalance between demand and supply of transportation infrastructure, leading to traffic congestion and problems such as travel delays, road accidents and environmental degradation, among others. Traffic congestion is a serious problem, costing substantially to drivers in terms of wasted fuel and time. Among others, in the urban road networks, inadequate traffic signal timings are one of the repeated causes of congestion (Chin et al., 2004). This thesis explores the application of a popular machine learning paradigm; Reinforcement Learning (Sutton & Barto, 2011) (used for learning effective control policies by interacting with complex environments) in the field of traffic and transportation, particularly for autonomous signal control.

At a road intersection, operation of traffic signal infrastructure is administered by a *signal timing plan*. This timing plan defines the sequence in which the traffic light phases (i.e. green signal) must be activated and the corresponding duration of each phase. Widely-used conventional signal control methods are based on simple protocols that follow preset/predefined signal control regimes for preset time intervals. Fixed-time/preset signal (Koonce & Rodegerdts, 2008) regimes are typically based on historically recorded traffic data. The time interval may alter based on the peak or quiet hours, but signals are not otherwise optimized. However, over the years the variability and unpredictability of traffic have outpaced the capabilities of preset signal control methods to operate efficiently. Consequently, the transportation research community shifted its focus towards adaptive traffic signal control (ATSC). In contrast to preset signal control, adaptive signal control is capable of adjusting its regimes online in real-time as per the changing traffic patterns. Some of the more-widely used ATSC systems include SCATS (Sims & Dobinson, 1980), SCOOT (Hunt, Robertson, Bretherton, & Royle, 1982), PRODYN (Henry, Farges, & Tuffal, 1984)

and OPAC (Gartner, Pooran, & Andrews, 2001). However, these signal control methods are not truly adaptive and are primarily designed to be reactive to slow long-term variations in traffic flows and not to random short-term fluctuations in traffic patterns (e.g. a sudden road segment blockage caused by an accident).

Real-world traffic phenomena form a complex, non-linear system, including highly-stochastic driving dynamics (such as sudden accidents blocking the flow). To exert true, real-time, adaptive control, signal control systems' optimization is needed to be carried out by automated agents capable of self-learning, self-configuration and self-optimization. Since the 1990s, Reinforcement Learning (RL) is considered as a direct approach to optimal adaptive control of non-linear systems involving sequential decision-making (Sutton, Barto, & Williams, 1992). Unlike conventionally used signal control methods, RL agents do not depend on heuristic assumptions, instead, they monitor the environment through perception, influence it by applying actions and *learn* the optimal control by observing the outcomes of actions. RL was first applied to traffic signal control in the 1990s, with the first techniques limited to tabular Q-learning (Thorpe & Anderson, 1996). Traditional RL methods suffered from limited scalability and optimality in practice. However, in recent years, deep learning paradigms (such as deep neural networks (DNNs)) (LeCun, Bengio, & Hinton, 2015) have proven their effectiveness in significantly improving the performance of RL methods. Deep Reinforcement Learning (DRL) (a mechanism combining reinforcement and deep learning) emerged as a powerful paradigm; demonstrating unprecedented success in complex and dynamic settings such as Atari games (Mnih et al., 2013), among others. DRL facilitates end-to-end learning (i.e. a direct mapping from sensory inputs to action outputs) and eliminating the need for hand engineering of task-specific features by domain experts. To accomplish a particular task, the DRL agent learns the set of environmental features that are significant in each task. These agents derive efficient representations of high-dimensional, raw sensory data (such as videos and images) and subsequently utilize these to generalize the past experience to new unseen situations.

Effective signal control with multiple competing traffic flows altering dynamically (often non-periodically) through the day, is a challenging job for the conventional signal control methodologies. The goal of this thesis is to develop a signal control methodology that can provide an effective signal control in the face of complex, imprecise traffic environment. This thesis presents a Deep Reinforcement Learning (DRL)-based signal control method, to optimize traffic flows (that

fluctuate dynamically through the day) through intersections in real time-based *solely* on *live* camera feed. Having the ability to visually perceive the prevailing traffic state gives our signal control agent an opportunity to extensively process its environment and subsequently, learn intricate feature representations. This enables our agent to make signal control decisions, based on 3D-view of the traffic environment (including vehicles' type, their precise positions and corresponding approach speeds) that would otherwise be impossible/impractical to carry out using popular traffic data collection methods (such as induction loops and microwave detectors (Coifman, 2006)).

1.1 Research Questions and Main Contributions

The aim of this thesis is to contribute to the field of traffic and transportation, in particular signal control optimization. We address the following research questions:

Question 1: Does the existing simulation platforms (both transportation-based and in general) support human-like learning (e.g. based on realistic visual input)?

Question 2: Can traffic signals be controlled in real time such that the signal regimes can be effectively tailored to dynamically changing traffic situations, by *solely* using *live* camera feed?

Question 3: Can traffic flows be optimized through multiple road intersections *solely* based on *live* camera feed, to increase the efficiency of signal control infrastructure at a network-level?

Question 4: Can the signal control agents generalize well to different newly-encountered traffic situations by transferring their previously-learned skills/knowledge, without having to train them from scratch every time they encounter a new (never seen before) traffic situation?

Question 5: Given the prevailing traffic conditions, can DRL agents' signal control decisions (i.e. configured signal phase in a certain traffic situation) be interpreted?

We address each of the above-listed research questions and contribute to the advancement of the state-of-the-art traffic signal optimization, as follows:

Contribution 1: To address *Question 1*, we built a gamified traffic simulation platform; Traffic3D. The goal of Traffic3D is to provide a fast, cheap and scalable proxy for real-world traffic environments by creating physically, visually intelligent traffic simulations. This contribution has

been published in the conferences; AAMAS-2019 (International Conference on Agents and Multi-Agent Systems) (Garg, Chli, & Vogiatzis, 2019b) and ICCS-2019 (International Conference on Computational Science) (Garg, Chli, & Vogiatzis, 2019c).

Contribution 2: To address *Question 2*, we developed a Deep Reinforcement Learning (DRL)-based signal control agent to optimize the flow of traffic through intersections under a wide range of ambient conditions (such as dynamically varying traffic densities, vehicle types, weather and lighting conditions) perceived *solely* using *live* camera feed. This contribution has been published in the conferences; ICITE-2018 (IEEE International Conference on Intelligent Transportation Engineering) (Garg, Chli, & Vogiatzis, 2018) and ITSC-2019 (IEEE International Conference on Intelligent Transportation) (Garg, Chli, & Vogiatzis, 2019a).

Contribution 3: To address *Question 3*, we devised a system of multiple, coordinating traffic signal control agents. This thesis presents the first application of multi-agent deep reinforcement learning (DRL) to achieve traffic optimization through multiple road intersections *solely* based on raw pixel input from *live* CCTV cameras. This contribution has been accepted (to appear) in the conference ITSC-2020 (IEEE International Conference on Intelligent Transportation).

Contribution 4: To address *Question 4*, we used Transfer Learning so that when encountering new (visually different) traffic situations (such as different intersection layouts, traffic densities, weather and lighting conditions), our single signal control agent leverages previously-learned knowledge; accumulated across a series of experiences, to optimize traffic flows.

Contribution 5: To address *Question 5*, we implemented a specialised visual explanation technique to interpret a certain signal control decision, given the prevailing traffic condition which is visually perceived by the signal control agent. This contribution is part of our accepted paper (to appear) in the conference; ITSC-2020 (IEEE International Conference on Intelligent Transportation).

1.2 Thesis Outline

This thesis consists of eight chapters, outlined below:

Chapter 1 introduces the problem of ineffective signal control strategies within the urban road networks and the potential solution-based on Deep Reinforcement Learning.

Chapter 2 includes the relevant background and mathematical notation describing our signal control agents' implementation in different scenarios (such as single-intersection and multi-intersection). In this chapter, we also introduce the pertinent literature in the domain of signal control optimization.

Chapter 3 introduces our rich and extensible 3D-traffic environment; Traffic3D to train autonomous agents in a high-dimensional complex traffic environment.

Chapter 4 presents a Deep Reinforcement Learning (DRL)-based signal control agent, with the ability to exert real-time, adaptive control.

Chapter 5 demonstrates the optimization of traffic flows through multiple intersections to achieve network-level coordination between individually operating signal control agents.

Chapter 6 explores transfer learning to facilitate the development of autonomously operating signal control agents that can effectively cope with newly-encountered traffic situations.

Chapter 7, by using a visualization technique, effectively reasons about signal control agents' signal regime decisions, while demonstrating the potential of the learning algorithm used and validating the benefits of visual data-based signal control optimization approach.

Chapter 8, concludes the research undertaken in this thesis, highlighting the relevant future research lines to further enhance the quality of our road transportation systems.

Chapter 2

Background

In this chapter, we briefly introduce signal control-based related work, followed by a description of various underlying concepts involving our autonomous signal control agent's implementation.

2.1 A Brief Introduction of Signal Control-based Related Work

Conventional signal control methods (details provided in Sec. 4.1.1) independently optimizing traffic flows through one intersection at a time, operate on pre-programmed *signal regime plans* (Sims & Dobinson, 1980; Hunt et al., 1982; Henry et al., 1984; Gartner et al., 2001). The phase time interval may change based on the peak or quiet hours, but they are not otherwise optimized. However, over the years, as the volatility of traffic patterns outpaced the effectiveness of pre-programmed signal control methods, interdisciplinary methods such as Reinforcement Learning (RL)-based are being studied to adaptively configure signal regimes. There exists a large body of work on RL-based adaptive signal control (discussed in detail in Sec. 4.1.2), however, the majority of recent studies are conducted using relatively simplified traffic state information-based on hand-crafted traffic features (i.e. a vector specifying the presence of vehicles at an intersection and their respective speed information) (Van der Pol & Oliehoek, 2016; Genders & Razavi, 2016; Gao, Shen, Liu, Ito, & Shiratori, 2017; Liang, Du, Wang, & Han, 2018). Our signal control method, in contrast, is end-to-end trainable and utilizes *live* visual inputs, rendering an extensive representation of the prevailing traffic state (including flows, types of vehicles, weather conditions, etc.) to decide the configuration of signal regimes. Close to our work, (Mousavi, Schukat, & Howley, 2017; Jeon, Lee, & Sohn, 2018) used a visual representation of the traffic environment for signal control. However, the simulation environment used in both the research studies does not include the visual complexities of urban traffic (Pell, Meingast, & Schauer, 2017), which impedes

the purpose of using visual traffic data for signal control. In contrast, our dynamic 3D-traffic simulation paradigm; Traffic3D (Garg et al., 2019b, 2019c) renders a natural and unstructured traffic environment for the signal control agents to operate on. Merits and demerits of Traffic3D compared to other state-of-the-art simulation platforms is discussed in Sec. 3.1.

Only a handful of studies address signal control optimization through multiple intersections (Sec. 5.1) (Wiering, 2000; El-Tantawy, Abdulhai, & Abdelgawad, 2013; Chu, Qu, & Wang, 2016; Aziz, Zhu, & Ukkusuri, 2018). Most of these research studies implement value function-based approaches (Q-learning) for traffic optimization. Value function-based methods are often criticized for being unstable and in practice are difficult to use. For instance, they are inclined towards finding deterministic policies, whereas, in a dynamic environment like traffic, an effective policy is expected to be stochastic (Sutton, McAllester, Singh, & Mansour, 2000). In contrast, we use a different method (i.e. actor-critic RL (Konda & Tsitsiklis, 2000)) for autonomous signal control through a network of intersections based *solely* on *live* camera footage. This thesis, for the first time, by using transfer learning, empirically evaluates our signal control agents' generalizability and transferability skills to newly-encountered traffic conditions, including prioritization of emergency vehicles' navigation through the intersections, handling adverse weather and lighting conditions. To our knowledge, transfer learning for a vision-based signal control task has not been previously explored, details of this are provided in Sec. 6.1. Furthermore, this work presents the first application of attention-visualization (details provided in Sec. 7.1), to illustrate the interpretation of our agents' signal control decisions, while highlighting the benefit of using visual traffic data for signal control over conventional traffic data collection methods (such as induction loops and microwaves (Koonce & Rodegerdts, 2008)).

2.2 Reinforcement Learning (RL)

A fundamental problem faced by autonomous agents while interacting with an extensive environment is sequential decision making. Reinforcement Learning (RL); a popular machine learning paradigm, is particularly useful in situations where data arrives in a continuous, sequential manner and the agent is required to adapt its behavior in real time. In a typical RL setting, an agent learns to achieve a goal. It does so by interacting dynamically with its environment and trying different actions in different situations. The agent improves its learning by receiving scalar feedback

from the environment. The basic RL loop demonstrating this interaction encompasses an agent receiving environment observations, selecting actions to maximize a reward signal and receiving feedback from the environment to evaluate the quality of action taken.

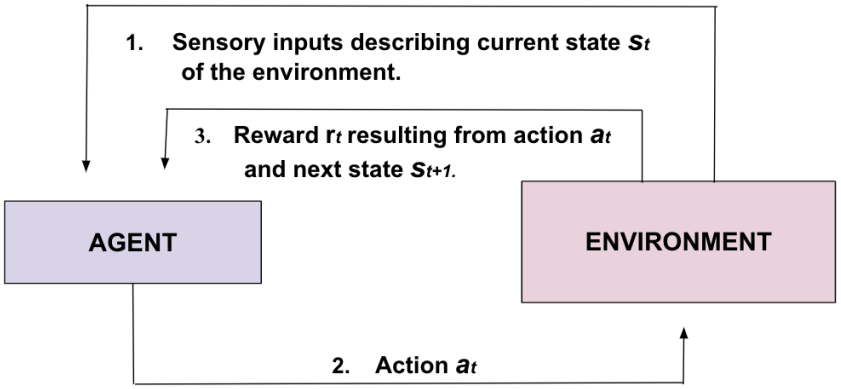


FIGURE 2.1: Basic Reinforcement Learning Mechanism.

Fig. 2.1 outlines the basic RL mechanism, demonstrating the interaction between the agent and the environment. An agent has a repertoire of possible actions and it learns to map situations to actions to maximize a numerical reward signal. A standard RL framework is mathematically modelled as a Markov Decision Process (MDP). A Markov Decision Process (MDP) is a discrete-time stochastic control process, that is defined using a tuple $< S, A, T, R, \gamma >$, where S and A are the state and action spaces, respectively. $\gamma \in (0, 1)$ denotes the discount factor, which models the relevance of immediate rewards over the future rewards. After observing a state, an agent working under the policy $\pi : S \mapsto A$ produces an action. Given current state s_t and action a_t, the transition function $T : S \times A \times S \mapsto \mathbb{R}^+$ determines the distribution of the next state s_{t+1}. The reward function R is determined by $R : S \times A \mapsto \mathbb{R}$. An episode $\tau \sim \mathcal{M}$ with horizon H is a sequence of state, action, reward $(s_0, a_0, r_0, \ldots, s_H, a_H, r_H)$ at every time-step t. The discounted episodic return of τ is determined by $R_t = \sum_{t=0}^{H} \gamma^t r_t$. Given the agent's policy π, the expected episodic return is defined by $E_\pi[R_\tau]$. The expected episodic return is maximized by optimal policy π^*

$$\pi^* = \arg\max_{\pi} E_{\tau \sim \mathcal{M}, \pi}[R_\tau]. \tag{2.1}$$

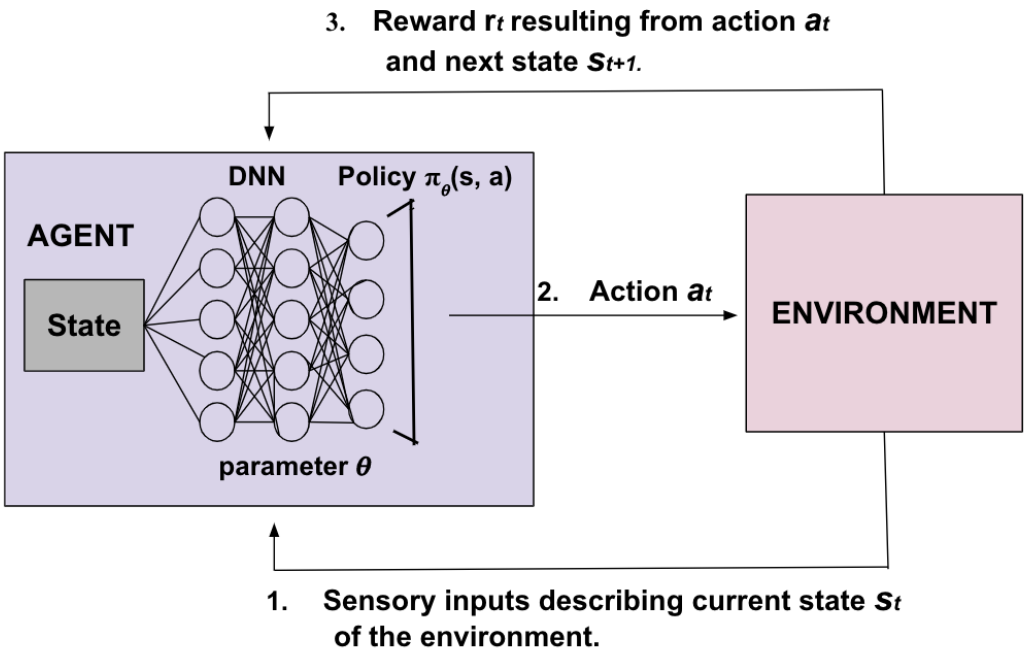


FIGURE 2.2: Deep Reinforcement Learning Mechanism.

2.3 Deep Reinforcement Leaning (DRL)

Initially, RL methods were limited in their ability to effectively process the environment in its raw form. RL agents relied on careful engineering and considerable domain expertise to transform the raw environmental data (such as images) into a suitable representation/feature vector, using which the agent processes/understands its environment. For instance, studying the game-playing strategies of an expert player, observing which actions lead to winning the game and constructing features from these insights. However, the breakthrough advancements in the field of deep learning have made it possible for RL agents to automatically learn the intricate environmental feature representations directly from high-dimensional raw data such as images and videos (He, Zhang, Ren, & Sun, 2016).

Deep learning methods are representation-learning methods consisting of multiple levels of representation, composed of non-linear modules such that each transforms the representation at one level (beginning from the raw input such as images and videos) into a representation at a higher, more abstract level. Fundamentally, deep learning consists of computational models (such as deep neural networks), composed of multiple processing layers to learn representations of data with multiple levels of abstraction. For instance, for an image classification task, the initial neural network layers detect more generic features (such as edges and color blobs), while the later layers

progressively detect more specific features. Training a deep neural network with sufficient data has bee shown to lead to learning of significantly better data representations than hand-crafted features (LeCun et al., 2015). In particular, deep convolutional neural networks have yielded a substantial performance boost for various visual-based tasks (Krizhevsky, Sutskever, & Hinton, 2012). Fig. 2.2 illustrates an end-to-end trainable DRL mechanism. In high-dimensional RL settings, a deep neural network with parameters θ represents policy π (i.e. π_θ). The agent aims to learn θ^* achieving the highest expected episodic return,

$$\theta^* = \arg\max_{\theta} E_{\tau\sim\mathcal{M},\pi}[R_\tau]. \tag{2.2}$$

2.4 Reinforcement Learning Methods

This section highlights different RL methods:

2.4.1 Q-learning

Q-learning (Watkins & Dayan, 1992) is a model-free RL method, which does not build the model of the environment's transition and reward functions. On the contrary, it directly estimates the value of taking an action a in state s, such that $Q - value$ of the s, a-pair is represented as $Q(s, a)$. Q-learning is an *off-policy* algorithm; a class of algorithms that uses a different policy for estimating Q-values than for selecting an action. Q-learning updates the Q-values of the current s, a-pair using the greedy policy to estimate the Q-value of the optimal policy of the next s, a-pair. The agent following the traditional Q-learning, uses a lookup table of s, a-pairs and iteratively updates the Q-value estimates using;

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha\left[r_t + \gamma[\max_{\alpha'} Q_t(s_{t+1}, \alpha'; \theta_t)] - Q_t(s, a)\right] \tag{2.3}$$

Eq. 2.3 represents the difference between the current estimate of the s, a-pair and the actual value of the s, a-pair. However, the true value of the s, a-pair is not *a priori* known, the agent instead uses the current reward signal and the maximizing Q-value of the next state as a proxy for the true value. This method is known as tabular Q-learning and it works fine in small domains. However, many real-world problems have large state (S), action (A) spaces and enumeration over s, a-pairs is not feasible. A straightforward solution to this problem is function approximation, in

which Q-value is no longer an entry in an $|S|$ X $|A|$ table, instead it is a function parameterized using weights θ. These weights can be updated using gradient descent methods (Ruder, 2016), to minimize the mean squared error between the current estimate of $Q(s, a)$ and the target, which is defined as the true Q-value of the s, a-pair under policy π; $Q^{\pi}(s, a)$. The gradient descent update can be computed by taking the derivative of the mean square error (MSE);

$$MSE(\theta) = \sum_{s \in S} P(s) \Big[Q^{\pi}(s, a; \theta^*) - Q_t(s, a; \theta_t) \Big]^2 \tag{2.4}$$

where $P(s)$ is the sampling distribution or the probability of visiting state s under policy π.

The derivative is represented as;

$$\frac{\partial}{\partial \theta_t} MSE(\theta) = 2 \Big[Q^{\pi}(s, a; \theta^*) - Q_t(s, a; \theta_t) \Big] \frac{\partial}{\partial \theta_t} Q_t(s, a, ; \theta_t) \tag{2.5}$$

As targets are not directly observable, a proxy is used for targets, given by the reward in the current time-step and a discounted estimate of the next state's best Q-value using the current Q-function approximation Q_t;

$$Q^{\pi}(s, a; \theta^*) \approx r_t + \gamma \Big[\max_{\alpha'} Q_t(s_{t+1}, \alpha'; \theta_t) \Big] \tag{2.6}$$

The Q-value is an expected discounted cumulative reward for taking an action a in state s and following the policy π afterwards.

2.4.2 Policy Gradient Reinforcement Learning

Neural Network-based function approximation is essential for RL to be effective in large high-dimensional state spaces. A dominant approach has been value-function approximation (discussed above). The value function approach is known to work well in many applications, but it has some limitations such as it cannot efficiently learn stochastic policies and it is less-effective in high-dimensional state/action spaces (Sutton et al., 2000). In this thesis, we explore an alternative policy-based approach to function approximation (known as Policy Gradient).

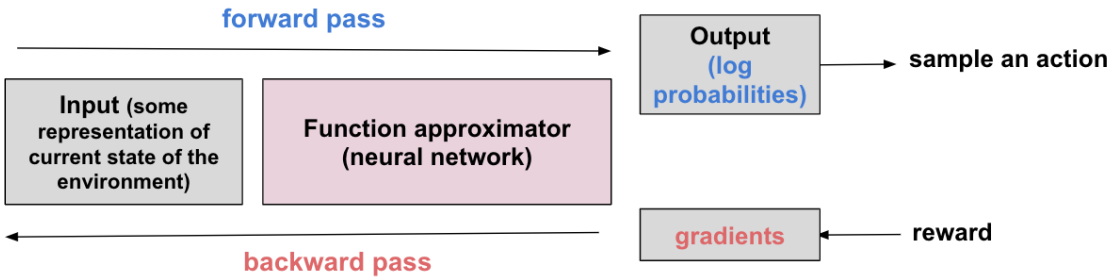


FIGURE 2.3: Policy Gradient Reinforcement Learning Mechanism.

Fig. 2.3 demonstrates the policy gradient pipeline. Instead of estimating the value function, a stochastic policy is directly estimated using an independent function approximator (such as a neural network), whose input is some representation of the current state of the environment (s_t), it generates as output action selection probabilities (a_t), and whose weights are the policy parameters. In the forward pass, the neural network computes the probability distribution of the pre-defined actions, from which an action is sampled. Based on the implemented action and received rewards, gradients are computed in the backward pass. The objective stated in equation 2.2 can be achieved using policy gradient RL by stepping in the direction of $E[R_\tau \nabla log \pi(\tau)]$. This gradient can be converted into a surrogate loss function (L_{PG}),

$$L_{\text{PG}} = E[R_\tau \log \pi(\tau)] = E\left[R_t \sum_{t=0}^{H} \log \pi(a_t|s_t)\right] \tag{2.7}$$

such that the gradient of L_{PG} is equal to policy gradient.

2.4.3 Actor-Critic Reinforcement Learning

In actor-critic RL, the actor is the policy $\pi^\theta(a \mid s)$ with parameters θ, based on which actions are estimated, while the critic computes value functions to help the actor in learning. Action and value function are estimated using function approximators and the gradient is estimated from trajectories sampled from environment. R_t is replaced by an expression equivalent to $Q(s_t, u_t) - b(s_t)$, where $b(s_t)$ contributes in reducing the variance. If R_t is replaced by $A(s_t, u_t)$, then $b(s_t) = V(s_t)$. $R(t)$ can also be replaced by the *temporal difference* error; $r_t + \gamma V(s_{t+1}) - V(s)$, which is an unbiased estimate of $A(s_t, u_t)$ (Konda & Tsitsiklis, 2000).

2.5 Multi-agent Reinforcement Learning (MARL)

This thesis extends to multi-agent RL for implementing coordination and cooperative learning among independently operating RL-based signal control agents. To accomplish this task, a network of signal control agents is considered, forming a multi-agent system. In this multi-agent setting, the goal is to train signal control agents to effectively participate in optimizing traffic flows at a global network level. Here, we consider the multi-agent extension of the Markov Decision Process (MDP); which is defined by a tuple $E =< S, U, P, r, Z, O, n, \gamma >$, where n agents (represented by $a \in A \equiv [1, ..., n]$) act in the environment E. The true state of the environment is represented as $s \in S$. At each time-step, each agent independently, simultaneously chooses an action $u^a \in U$, forming a joint action space $u \in U \equiv U^n$, which produces a transition in the environment (represented by $P(s\prime \mid s, u) \colon S \times U \times \mapsto [0, 1]$). For their individual selected actions, the agents receive their individual rewards; $r(s, u) \colon S \times U \mapsto \mathbb{R}$ and $\gamma \in (0, 1)$ denotes the discount factor. Given the real-world traffic complexity, we consider a partially observable traffic settings, where each agent acts on its local observations $z \in Z$ (based on the observation function $O(s, a) \colon S \times A \mapsto Z$). Each agent depends on action-observation history (represented by $\tau^a \in T \equiv (Z \times U)^*$), based on which it conditions a stochastic policy $\pi^a(u^a \mid \tau^a) \colon T \times U \mapsto [0, 1]$. The discounted return is denoted by $R_t = \sum_{t=0}^{\infty} \gamma^l r_{t+l}$. The agents' goal is to learn a policy that maximizes their joint expected discounted returns.

2.6 Interpretable Signal Control via Deep Neural Network (DNN) Visualization

Deep learning models are known to offer insights that go well beyond human understanding (Mnih et al., 2015). We analyse our signal control agents' decision-making through a specialised visualization technique involving the visual attention of vision-based inputs (Chapter 7). The core idea of visualizing DNNs is to adaptively realize the most-relevant features as per the input data. For the DNN visualization, we implement Grad-CAM (Selvaraju et al., 2017), which facilitates the visual explanations of DNNs using gradient-based localization (i.e. localization of visual evidence in an image). This localization technique generates explanation for a CNN-based model without the need for any architectural reforms or re-training. Gradients flowing into the final convolutional

layer generate a coarse localization, highlighting the important areas in the image leading to a certain neural network output. Via Grad-CAM, a heatmap is produced on top of the input image, depicting the critical areas that dominate a certain decision. Grad-CAM was previously applied to produce visual explanations for a variety of CNN-model families; (1) image classification (CNNs with fully-connected layers), (2) image captioning (CNNs to achieve structured outputs) and (3) visual question answering (CNNs with multimodal inputs). To our knowledge, in this work, Grad-CAM is applied for the first time to produce visual explanations for a signal control optimization task.

2.7 Summary

In this chapter, we briefly introduced relevant signal control-based literature. We will further discuss the pertinent literature (based on the contributions enlisted in Sec. 1.1) in-depth in the following chapters. In this chapter, we also presented the necessary background contributing to the implementation of our signal control agents in both; single junction and multiple junction scenarios. The topics we discussed in this chapter include deep reinforcement learning, types of reinforcement learning methods (Q-learning, policy gradient and actor-critic), multi-agent reinforcement learning for coordinated multi-intersection signal control and DNN visualization to interpret agents' signal control decisions. More specific implementation details are provided in the following chapters.

Chapter 3

A New Simulation Paradigm: Traffic3D

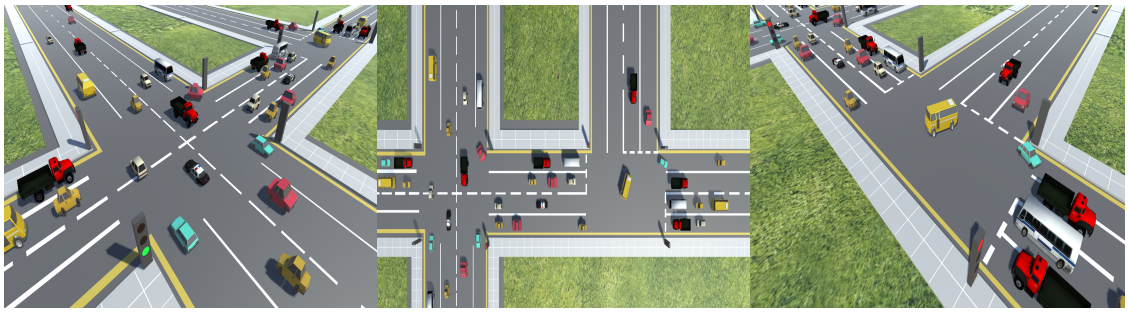
In this chapter, we address the *research question 1* (outlined in Sec. 1.1). A longstanding goal of the artificial intelligence research community is to devise robust agents that demonstrate human-like intelligence by autonomously performing tasks in real-world settings. However, training agents to act effectively in the real world entails challenges that go well beyond existing supervised learning tasks such as object recognition. To be able to perform well in a dynamic physical environment, an agent is required to have a significant amount of extensive interaction with its environment so that it can explore, learn and effectively adapt. Reinforcement learning (RL) (Sutton & Barto, 2011) paradigms hold the promise of allowing autonomous agents to learn to accomplish arbitrary tasks. Mimicking the fundamentals of human learning, RL agents learn by interacting with their environment and observing the outcome of these interactions in the form of positive or negative feedback. However, these agents are slow to train, since they have no prior knowledge of their intended environment and they are bound to have a large number of interactions with the environment to learn suitable policies. To avoid expense, risk and disruption associated with extensive real-world experimentation, computer simulations are considered as a viable alternative for development and evaluation of new ideas and algorithms. Although there has been widespread use of simulations to pursue research in road transportation, however, the existing simulation platforms are limited and fail to deliver critical physical and visual functionalities that are fundamental to authentic traffic simulation (Pell et al., 2017).

The RL agents learning in simulated environments depicting a high degree of realism are known to learn features, which are generalizable to their corresponding real-world environments (Sadeghi & Levine, 2016). However, creating realistic simulation environments can be cumbersome. Some of the commonly-encountered issues when creating these simulation platforms include; (1) Developing simulation environments with high-definition graphical rendering, to ensure

FIGURE 3.1: A view of Traffic3D's multi-intersection graphical display.

that digital content looks as close as possible to a physical scene. (2) Authentic physics support to model complex physical interactions between environment entities based on mass, friction and gravity. (3) Encompassing diversity of parameters and settings, to ensure generalizability and stability of the trained agent (domain randomization). Trained agents tend to exhibit good performance under ideal conditions (such as well-illuminated surroundings) but as the input conditions changes, the agents' performance often degrades. (4) Furthermore, it needs to be ensured that simulation assumptions about the agent's access to the environment are realistic; while operating in real-world settings, the agent will have limited sensoric and effectoric capabilities. For e.g. an agent controlling an autonomous vehicle cannot see the entire road network at any given instant.

To bridge the gap between simulations and real-world traffic dynamics, we created a traffic micro-simulation tool, Traffic3D (Garg et al., 2019b, 2019c) (illustrated in Fig. 3.1, 3.2, 3.3). The goal of Traffic3D is to push forward research in human-like learning (e.g. based on reliable visual input). Traffic3D provides a fast, cheap and scalable proxy for real-world traffic environment, including a diverse and extensible range of dynamic scenes which are both visually and physically realistic; to accurately simulate transportation entities and their emergent properties. Traffic3D renders traffic dynamics that are being generated through the simulation, encompassing all the emergent properties of the traffic entities, without making any explicit assumptions or aggregated models of these properties. For openness and research collaborations, Traffic3D is publicly available (including complete documentation and installation guidelines) at https://traffic3d.org/.



(A) (B) (C)

FIGURE 3.2: An example of different views (camera orientations) of a multi-intersection setting in Traffic3D.

3.1 Related Work

Computer simulations are increasingly being used to train and evaluate intelligent agents, with a large volume dedicated to games-based benchmarks such as the ATARI learning environment (Bellemare, Naddaf, Veness, & Bowling, 2013), DeepMind Lab (Beattie et al., 2016), TorchCraft (Synnaeve et al., 2016) and VizDoom (Kempka, Wydmuch, Runc, Toczek, & Jaśkowski, 2016). Though these platforms contributed in successfully developing and demonstrating the performance of RL agents (Mnih et al., 2013), these benchmarks are not photo-realistic and unsuitable to train agents to perform real-world tasks such as autonomous transportation. More Realistic Environments include House3D (Wu, Wu, Gkioxari, & Tian, 2018), CHALET (Yan et al., 2018), AI2THOR (Kolve et al., 2017). These environments provide 3D-rendered indoor house scenes. Using these environments, Gordon et al. (Gordon et al., 2018) endeavoured in effectively solving the task of visual question answering in interactive environments. Zhu et al. (Zhu et al., 2017) successfully implemented vision-based robot navigation by mapping sensory signals to motion commands. In contrast to these paradigms, we focus on the simulation of the urban traffic environment.

Traffic optimization being an established field, several traffic simulators exist. Pell et al. (Pell et al., 2017) thoroughly reviewed popular traffic simulation environments. The review acknowledges that currently-used traffic simulation tools are unable to adequately deliver critical functionalities that are fundamental to faithful traffic simulation. The existing traffic models lack in detail and flexibility. A detailed network model with efficient real-time traffic information collection capabilities is necessary to simulate heterogeneous transportation networks. Reinforcing the consensus among the computer vision research community that robust computer vision is key to the

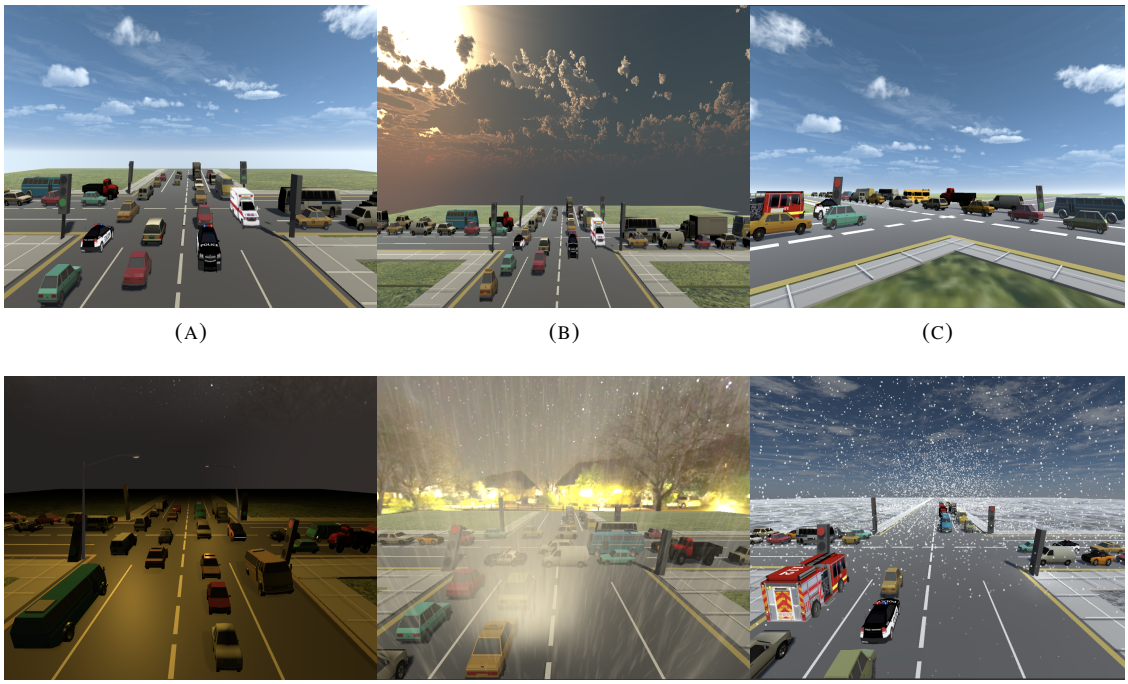
prevalence of autonomous transportation, the SYNTHIA (Ros, Sellart, Materzynska, Vazquez, & Lopez, 2016) dataset was introduced. SYNTHIA consists of a synthetic collection of a diverse set of urban traffic-specific photo-realistic images. We believe that due to its fundamentally different objectives during its creation, SYNTHIA does not support transportation infrastructure optimization. SYNTHIA's documentation does not provide any indication about important vehicle-related functionalities such as effectively handling the physics of vehicles. An urban driving game simulator, Grand Theft Auto (Richter, Hayder, & Koltun, 2017), is known to lack the flexibility required to pursue learning-based research simulations. Another driving simulator, TORCS (Wymann et al., 2000), is predominantly a racing simulator. It does not represent the complexities of urban driving efficiently.

In Table 3.1, we summarize the capabilities of a few widely-used traffic and deep learning-based simulation environments over important simulation characteristics; photo-realistic graphical rendering, 3D nature of objects, simulation physics and flexibility to customize the simulation environment according to the requirements of the application. Table 3.1 reflects that no single simulator supports comprehensive traffic-based research and analysis. To address this gap, traffic dynamics in our simulation platform; Traffic3D are generated using agent-based simulation model, yielding visually rich and physically precise traffic scenarios.

3.2 Our Simulation Environment: Traffic3D

The main contribution of this chapter is our gamified simulator; Traffic3D. Traffic3D consists of a diverse range of traffic scenes, including a variety of photo-realistic vehicles (such as emergency vehicles, personal and public transport vehicles) and street furniture (sidewalks and traffic lights). Scenes include, from the clear view of a sunny day to the blurry dimly-lit night, a rainy and a snowy day (as shown in Figure 3.3). Real-world traffic images are used as a reference to create 3D-traffic scenes with near-photorealistic lighting and texture. As a microscopic traffic simulation environment, Traffic3D configures behavior of every vehicle independently; emulating real-world

[1] no proper reactive control to random incidents like collisions between vehicles.
[2] does not support simulation of autonomous vehicles and does not prioritize public transport.
[3] unrealistic lane-closing behavior.
[4] restrictions in customizing delay.
[5] limited sensor suite.
[6] does not support road intersection simulation.
[7] information not available.



(A) (B) (C)

(D) (E) (F)

FIGURE 3.3: Different views of an intersection in Traffic3D . (A) A clear sky scene. (B) An evening scene. (C) A clear sky scene (2-way junction) . (D) A night scene. (E) A rainy scene. (F) A snowy scene

traffic dynamically. Traffic3D is a cross-platform simulation environment. It supports different operating systems, including Windows, Linux and Mac OS X.

To facilitate control optimization, we developed a framework to enable an agent to perform actions in a 3D-traffic environment and subsequently, perceive the outcomes of its actions. Our framework includes integration of Unity engine with deep learning python support (illustrated in Fig 3.4). It consists of two components; (1) a Unity game engine, and (2) a light-weight Python API. Traffic scenes are created within the Unity game engine with functionality to train intelligent agents in Python. These two components interact in a client-server manner to facilitate seamless bilateral communication between the traffic environment and the agent. The server implements the learning agent, while the traffic simulation environment acts as a client. This client-server architecture allows for a generic system, in which other analysis and learning platforms (such as Matlab, R and Julia) can be conveniently plugged-in instead of Python. Fig. 3.4 illustrates the traffic environment and learning agent's bilateral interaction. The socket data exchange mechanism we implemented to facilitate interaction between the simulation and the deep learning python module works accurately and seamlessly. However, deep learning training is typically long. To

Environment	Suitable for Traffic Simulation	Photo-Realistic	3D	Physics	Customizable
SUMO (Pell et al., 2017)	Yes	No	No	Yes (with restrictions[1])	Yes (with restrictions[2])
VISSIM (Pell et al., 2017)	Yes	Yes	Yes	Yes (with restrictions[3])	Yes (with restrictions[4])
TORCS (Wymann et al., 2000)	Yes	Yes	Yes	Yes (with restrictions[5])	Yes (with restrictions[6])
Virtual KITTI (Gaidon, Wang, Cabon, & Vig, 2016)	Yes	Yes	Yes	No[7]	Yes
CHALET (Yan et al., 2018)	No	Yes	Yes	Yes	Yes
AI2-THOR (Kolve et al., 2017)	No	Yes	Yes	Yes	Yes
ATARI (Bellemare et al., 2013)	No	No	No	No	No
DeepMind Lab (Beattie et al., 2016)	No	No	Yes	No	Yes
Traffic3D	**Yes**	**Yes**	**Yes**	**Yes(with high degree of realism)**	**Yes(fully)**

TABLE 3.1: Comparison between different traffic-based and deep learning-based simulation environments.

accelerate training, Traffic3D allows control of the frame rate and speed of the simulation without affecting its quality. Collecting data using an accurate simulator is still faster and safer than doing so directly from the physical world (Zhu et al., 2017).

3.3 Traffic3D's Properties

In this section, we discuss the key simulation properties of Traffic3D. This section highlights Traffic3D's potential in creating relatively natural-looking and realistically-operating traffic environment with believable visuals and physics.

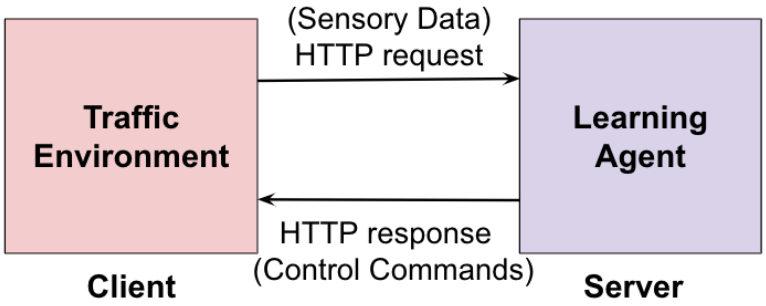


FIGURE 3.4: Our traffic environment and python API forming a client-server system.

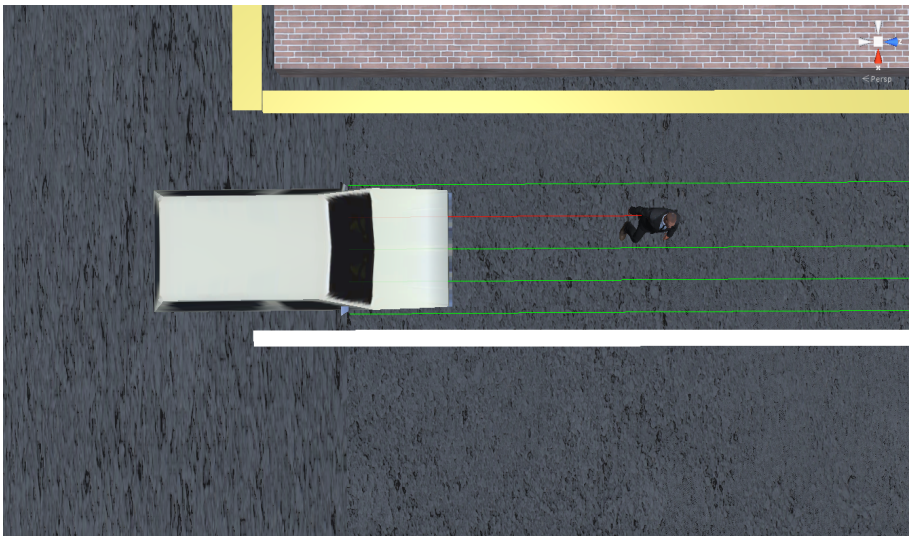


FIGURE 3.5: An illustration of raycast sensor for collision avoidance.

3.3.1 Traffic3D's Physical Properties

To ensure high-precision simulation, Traffic3D is supported by NVIDIA PhysX SDK[1] that allows effective modelling of complex physical interactions between the transportation entities based on mass, friction and other forces (such as gravity). Vehicles are independently tuned to react appropriately accordingly to their input parameters and also to the presence of other vehicles/traffic infrastructure. Vehicles exhibit progressive slow-down and acceleration. Vehicles are equipped with appropriate sensors to avoid collisions, overtake and give way to high-priority (emergency) vehicles.

[1] https://developer.nvidia.com/physx-sdk

3.3.2 Traffic3D's Visual Properties

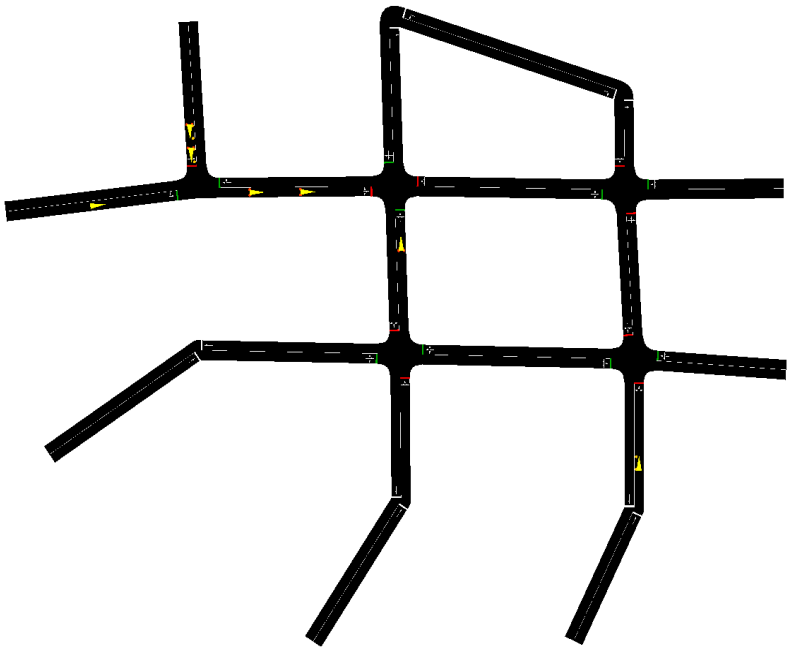
Traffic3D provides high-definition graphical rendering to potentially facilitate the transfer of developed models to real-world settings with no or minimum pre-training. Traffic3D supports real-time global illumination such that light, material, texture and scale work in synergy to make digital content look as close to a real scene as possible. The lighting techniques used in Traffic3D create immersive scene lighting, mimicking the real-world effects. Objects' shadows can be dynamically cast on the scene, adding further realism to the environment.

3.3.3 Traffic3D's Sensors

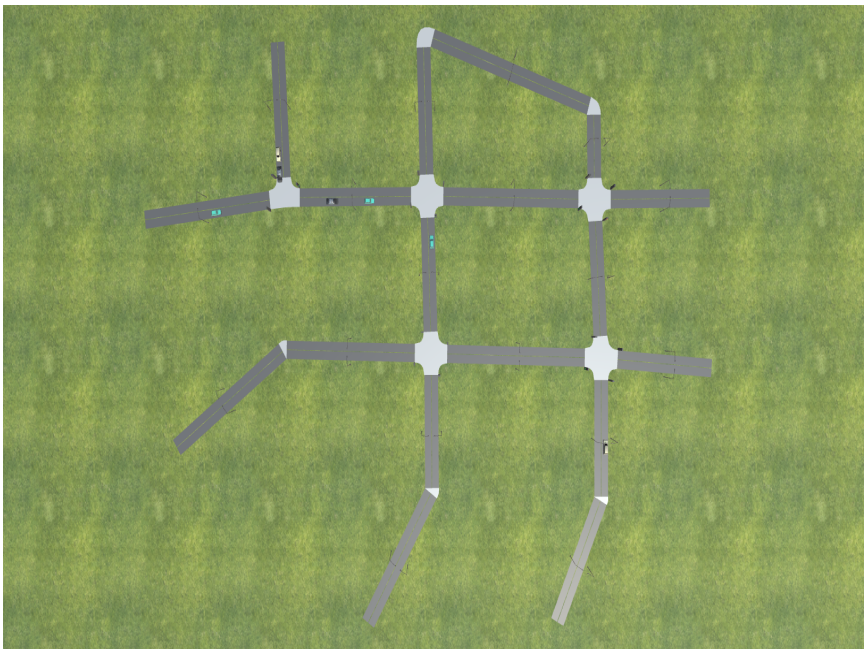
Traffic3D facilitates the fast, inexpensive, limitless, diverse and photo-realistic collection of traffic data for various research purposes such as online reinforcement learning and supervised learning. Vehicles in Traffic3D are well-equipped to behave realistically. They have appropriate sensors (i.e. ray-cast sensors which can be customized to use as an equivalent to Lidar and microwaves) and programmed behaviour to avoid collisions, overtake and give way to emergency vehicles (illustrated in Fig. 3.5). Cameras in Traffic3D replicate the operation of real-world cameras; supporting capturing of photo-realistic images and videos with proper field-of-view and depth-of-view. To study the effects of occlusions in the traffic environment, which is common in real-world traffic scenarios; vehicles and street furniture that are not currently being seen by the camera can have their rendering disabled. Multiple cameras can be deployed at the same time within a scene to perceive different aspects of the scene and their views can be combined in different ways. For example, in addition to the main view of the complete scene, when simulating a car, camera output to show a rear-view mirror footage can also be simulated separately, if needed.

3.3.4 Co-Simulation with SUMO

Traffic3D facilitates the simulation of traffic scenarios using external traffic simulators such as SUMO (Behrisch, Bieker, Erdmann, & Krajzewicz, 2011) (illustrated in Fig. 3.6). Traffic3D allows both road network and traffic density/distribution configuration/control through SUMO. It is also possible to use SUMO's physical abilities to configure properties of vehicles within Traffic3D (such as maximum speed, acceleration and mobility models).



(A)

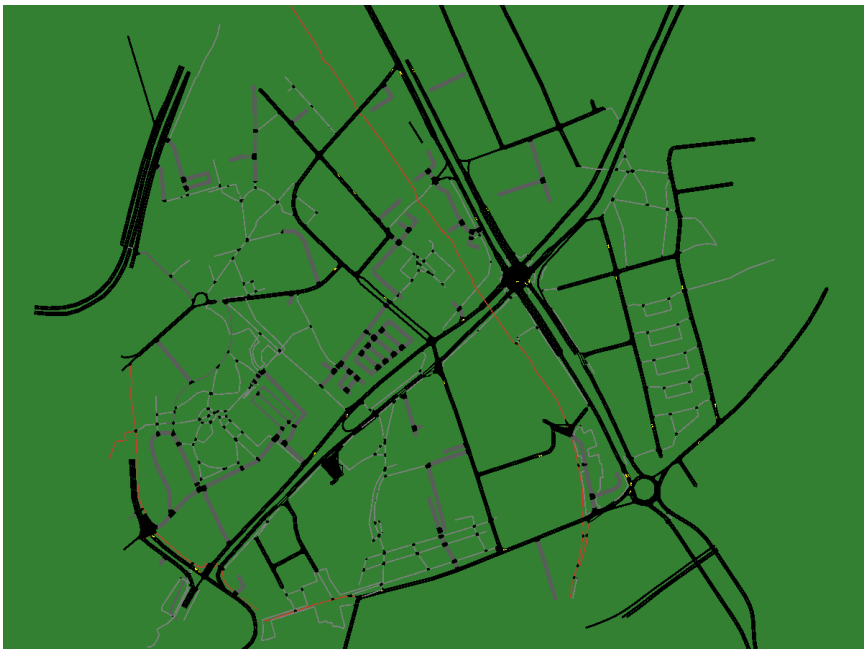


(B)

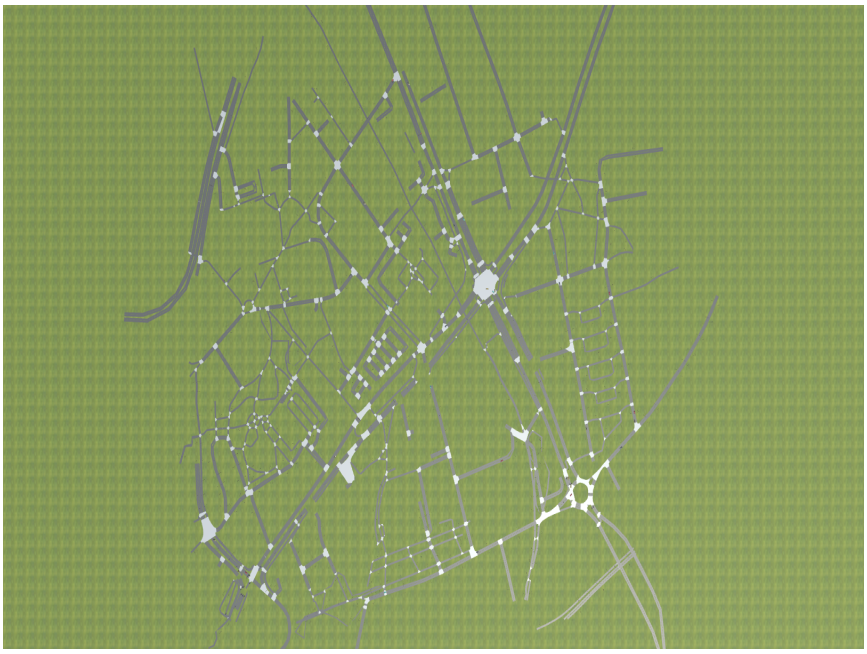
FIGURE 3.6: An illustration of co-simulation with SUMO. (A) A road network simulated in SUMO. (B) Equivalent road network transferred from SUMO to Traffic3D.

3.3.5 Real-world Road Maps

Via an interface with SUMO, real-world digital road maps can be imported in Traffic3D. This makes it is possible to create precise models of complex real-world road networks, including the number of lanes, accurate locations of traffic lights and permitted traffic movements. Real-world



(A)



(B)

FIGURE 3.7: An illustration of real-world road network around Aston University, Birmingham. (A) In SUMO. (B) In Traffic3D.

road maps are illustrated in Fig. 3.7.

3.3.6 Traffic3D's Diversity of Parameters

To ensure stability and generalizability of agents trained using Traffic3D to different variants of the environment, Traffic3D supports plug-n-play architecture to facilitate seamless creation and switching between scenarios during the run-time without interrupting the simulation or causing frame-rate hiccups. For example, traffic density can be dynamically varied independently, or around different times of the day (rush and quiet hours), as well as different weather conditions (e.g. rain, snow) can be varied during the run-time. This allows studying quantitatively the impact of different times and weather conditions on the learning agent's behavior. The current level of diversity offered by Traffic3D includes (1) vehicle number, types, models, colors and sizes, (2) vehicles' trajectory configuration and speed, (3) pedestrians, (4) different road layouts and surface textures, (5) variety of street furniture, (6) different lighting and weather conditions and (7) different camera position and orientation to comprehensively capture a wide range of possible aspects of a traffic scene.

3.3.7 Traffic3D's Reusability

Traffic3D consists of a library of pre-created traffic environments. In addition, it offers complete flexibility over the creation/deployment of new simulation scenarios; allowing complete customization over vehicle spawning, vehicle routing, road layouts and street furniture placement. Apart from creating simulation scenes manually, which can be tedious and less intuitive, Traffic3D also allows its users to programmatically generate scenes. Any of the traffic elements (such as vehicles) available within the project can be programmatically placed at any position and their physical (such a vehicle's size, mass and its speed) and visual behavior (such as a vehicle's texture) can also be programmatically configured.

3.4 Summary

In this chapter, we introduced our novel traffic simulation platform; Traffic3D. We built Traffic3D to create visually and physically realistic traffic simulation models to develop and test new technology with the goal of its eventual deployment in the real world. Traffic3D provides a high level of complexity (sensory, social and cognitive). To realistically train autonomous agents, Traffic3D supports unique simulation features, including complex physical phenomenon, creation of

relevant content (such as traffic objects with appropriate background), photo-realism, comprehensibility, robustness, adaptability, partial observability challenges and inexpensive collection of diverse training data. While most of the presently-used traffic simulation environments are pertinent to transportation-specific research, Traffic3D's capabilities transcend beyond the realm of transportation. Traffic3D can facilitate research across multiple directions, including, but not limited to semantic segmentation, 3D navigation visual question answering and neural attention mechanisms. As per the application under consideration, the required level of complexity can be conveniently simulated.

Chapter 4

Deep Reinforcement Learning-based Autonomous Traffic Signal Control using *Live* Camera Feed

In this chapter, we address the *research question 2* (outlined in Sec. 1.1) by using the Reinforcement Learning method; *Policy Gradient* (described in Sec. 2.4.2). A road intersection is a shared physical space; access to this common resource must be granted intelligently to optimize the flow of traffic while ensuring the safe passage of vehicles. In urban areas, the efficiency of road transportation systems significantly depends on the signal operation. Ever since their advent at the end of 19^{th} century, traffic lights have been used as the primary mode to grant vehicles access to the intersections, however, their benefits tail off when they fail to adapt to dynamically changing traffic flows (Priemer & Friedrich, 2009). Furthermore, existing signal control methods operate on pre-specified models of the traffic environment (Koonce & Rodegerdts, 2008). The purpose of having these pre-specified traffic models is to effectively visualize the picture of the present and imminent traffic conditions. These models are required to be constructed by the domain experts and must be generic enough to cover a variety of traffic conditions, as it is impractical to have a separate model to independently demonstrate each potential traffic situation. However, a generalized traffic model may not be able to reliably reflect the vast range of traffic flow patterns. For instance, TRANSYT, one of the popular adaptive signal control methods, only uses platoon-dispersion model to determine the arrival pattern of vehicles (Manar & Baass, 1996), irrespective of the prevailing traffic conditions.

A truly adaptive agent; exerting real-time signal control, must directly respond to the actual

traffic conditions without any pre-specification of the traffic environment. RL (Sutton et al., 1992) is a successful paradigm that obviates the need for pre-specification of an environment model. The environment RL agents operate in is not known in advance. Instead, the agents monitor their environment through perception, influence it by implementing actions and learn by observing the outcomes of their actions. Deep neural networks (DNNs) have further enhanced the learning power of RL agents; allowing end-to-end learning from raw sensory data and eliminating the need for hand-engineered features describing the prevailing state of the environment (Mnih et al., 2015). To accomplish a particular task, the DRL agent constantly interacts with its environmental and *learns* the set of environment features that are significant in each task. In this chapter, we address the problem of congestion around the road intersections. We present a DRL-based signal control agent that effectively optimizes traffic through intersections with multiple competing traffic flows altering dynamically and non-periodically through the day. Our signal control agent *solely* operates on *live* camera feed to optimize traffic flows through intersections in real time. Our empirical results reflect that our vision-based signal control agent is able to extensively process the traffic environment to learn it's intricate feature representations. This allows our agent to take signal control decisions, based on 3D-view of the traffic environment (including vehicles' type, their precise positions and corresponding approach speeds) that would otherwise be tedious to explore/exploit using popular traffic data collection methods (such as induction loops and microwave detectors (Coifman, 2006)). We compare our DRL-based signal control approach against baseline methods, including conventional signal control (fixed and adaptive). Our empirical evaluations reveal that our DRL-based signal control methodology led to a significant performance boost in comparison to the baseline methods.

4.1 Related Work

In this section, we summarize state-of-the-art conventional signal control methods, followed by incorporation of reinforcement learning (RL) methods for signal control.

4.1.1 Conventional Signal Control

Conventional signal control methods operate on pre-programmed signal regimes. These methods either follow a fixed-time/preset mode, an adaptive/actuated mode, or a combination of the two.

In the fixed mode, traffic networks are continuously monitored using sensors (such as induction loops) to collect important traffic data (such as traffic density on a certain road segment) and subsequently draw inference on congestion trends of the intersections. Based on this traffic information, the fixed/preset signal regimes are retrospectively configured. In this mode, typically the same duration of time is allocated to each phase in each cycle. However, with real-world traffic phenomena exhibiting highly-stochastic dynamics, the irregularities in traffic flow patterns cannot be *a priori* anticipated-based solely on historical data. An alternative is online traffic monitoring and subsequently configuring signal regimes in real time. In contrast to preset signal control, in actuated signal control (summarized in Table 4.1), actual traffic flows approaching the intersections are monitored online using sensors to reactively allow variations in phase durations. The sensors can be located upstream of stop lines at the entrance of a road link, or downstream from the previous junction. A variety of sensors is used to monitor traffic. The most commonly used sensors can be categorized as (a) underground vehicle detectors (such as induction loops (Coifman, 2006)), and (b) above-ground vehicle detectors (such as microwaves (Coifman, 2006)). The former detect the presence of vehicles by measuring the change in inductance when vehicles move over the loop and the latter detect the presence of vehicles anywhere within the field of vision as long as a vehicle is moving faster than 2-3mph. However, sensor reliability and accuracy are key concerns in these popular traffic detection approaches. For instance, vehicles taking sharp turns at an intersection can be missed by induction loops, resulting in unreliable traffic density estimation (Rhodes, Bullock, Sturdevant, Clark, & Candey Jr, 2005). Furthermore, buried under the road, the loop detectors have a narrow operational range and can be easily damaged by heavy vehicles or road deterioration. Microwave-based sensors are unable to detect slow-moving vehicles and can be easily obstructed by overhanging objects such as trees. Although optimizing phase durations has shown improvement in the performance of fixed-signal control, it does not take the prevailing traffic state adequately into account (loops have vehicle counting/speed monitoring limited functionality), leading to less than efficient regulation of dynamic traffic flows. Using RL may be advantageous here, as RL-trained signal control agents can *learn* to make effective signal control decisions precisely based on the true prevailing state of the traffic environment.

Control Technique	Traffic Data	Control System	Optimizing Performance Metric
SCAT (Sims & Dobinson, 1980)	Online data (from stop-line downstream detectors)	Centralized	Junction throughput, travel time
SCOOT (Hunt et al., 1982)	Online data (from upstream detectors)	Centralized	Delay, stops and congestion
UTOPIA (Mauro & Di Taranto, 1990)	Online data (from upstream detectors)	Centralized	Delay and stops
MOVA (Peirce & Webb, 1994)	Online data (from a single upstream detector)	Decentralized	Delay, congestion and stops
OPAC (Gartner et al., 2001)	Online data (from upstream detectors)	Decentralized	Delay and stops
Our study	**Online data (from cameras)**	**Centralized**	**Traffic throughput, junction travel-time**

TABLE 4.1: Summary of techniques used for adaptive traffic signal control.

Research Study	State Space	Reward	Simulator
Van der Pol and Oliehoek (Van der Pol & Oliehoek, 2016)	Position of vehicles	Teleport, wait time, stop, switch and delay	SUMO
Genders and Razavi (Genders & Razavi, 2016)	Position & speed of vehicles	Cumulative delay	SUMO
Gao et al. (Gao et al., 2017)	Position & speed of vehicles	Cumulative wait time	SUMO
Mousavi et al. (Mousavi et al., 2017)	Raw pixels	Cumulative delay	SUMO
Jeon et al. (Jeon et al., 2018)	Raw pixels	Number of waiting vehicles	VISSIM
Liang et al. (Liang et al., 2018)	Position & speed of vehicles	Cumulative wait time	SUMO
Our study	**Raw pixels**	**+1/car passing through the junction and -1/car waiting at the stop line**	**Our Simulator (Traffic3D)**

TABLE 4.2: Summary of recent DRL-based traffic light control research studies.

4.1.2 Reinforcement Learning-based Signal Control

The classical traffic modelling and analysis tools used by transportation planning agencies (summarized in Table 4.1) struggle to provide tractable policies for signal control infrastructure optimization. Traffic dynamics (including imprecision and uncertainty) form a non-linear, complex spatio-temporal system and are required to be modelled using complex dynamical systems. This inspired the transportation research community to move towards utilizing better-suited methods for real-time traffic management, including learning-based paradigms such as Reinforcement Learning (RL) (Sutton & Barto, 2011). To achieve greater real-time responsiveness and constantly optimize actual traffic flows, RL was first applied to signal control in the 1990s, with the initial experimentation limited to tabular Q-learning (Thorpe & Anderson, 1996). Traditional RL methods suffered from limited scalability and optimality in practice. Deep neural networks (DNNs), in recent years, have proven their effectiveness by significantly improving the performance of traditional RL methods (Mnih et al., 2013).

The majority of recent research on DRL-based adaptive signal control (summarized in Table 4.2) is conducted using relatively simplified traffic state information based on hand-engineered traffic features (i.e. a vector specifying the presence of vehicles at the intersection and their respective speed information). These features do not render a true picture of the traffic environment. In contrast, our signal control methodology is end-to-end-trainable and signal control decisions are taken *solely*-based on *live* camera feed rendering an extensive representation of the prevailing traffic state (including key traffic information such as flows, types of vehicles, weather and lighting conditions, etc.). Close to our work, (Mousavi et al., 2017; Jeon et al., 2018) used a visual representation of the traffic environment for signal control. However, the simulation environment used in both the research studies does not include the visual complexities of urban traffic (such as vehicles' 3D positions, their orientations and presence of clutter/complex background), which impedes the purpose of using visual traffic data for signal control. In contrast, our dynamic 3D-traffic simulation paradigm; Traffic3D (Garg et al., 2019b, 2019c) renders a natural and unstructured traffic environment for our signal control agent to operate on. Furthermore, most of the above-mentioned studies implement value-function based (Q-learning) approaches for traffic optimization. Q-learning is known to have worked well in many applications, but it suffers from limitations such as an inclination towards finding deterministic policies. However, in a highly dynamic environment (such as traffic), an effective policy is expected to be stochastic. In contrast,

in the current work, we implement two other popular RL paradigms; policy gradient (Sutton et al., 2000) and actor-critic (Konda & Tsitsiklis, 2000).

4.2 Autonomous Traffic Signal Control Methodology

In this section, we describe the implementation of the signal control agent, including state, action, reward specifications.

4.2.1 Problem Formulation

The objective of this work is to develop an efficient, fully-actuated agent that learns to control traffic signals in real time-based solely on *live* footage of the traffic situation of the area the signals affect. Our agent directly maps RGB images (describing the prevailing traffic state) to actions (controlling the traffic signals), demonstrating end-to-end learning for real-time adaptive signal control.

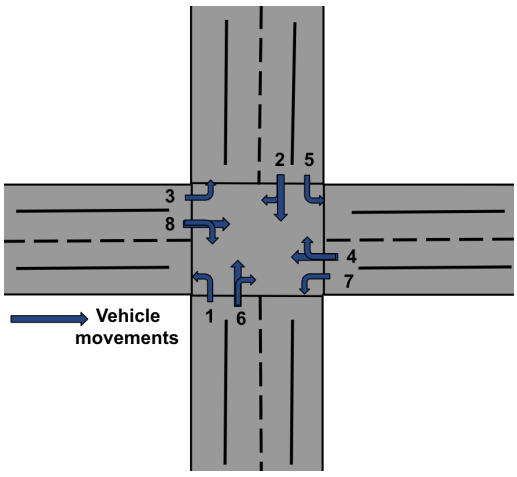


FIGURE 4.1: Possible Signal Phases.

4.2.2 Traffic Model Simulation

As stated previously (in Chapter 3), due to economic and safety concerns, an agent cannot be trained via DRL to autonomously control traffic signals in the real world. Simulation is considered as a safe, cost-effective, controlled tool catalyzing protocol development. All the experiments presented in this chapter are conducted using our traffic simulation environment; Traffic3D (Garg et al., 2019b, 2019c). We created 3D-intersection scenarios (illustrated in Fig. 3.3) with microscopic traffic properties. Based on real-world traffic specifications (Koonce & Rodegerdts, 2008),

our traffic environment (including weather and lighting conditions) and vehicle behavior (including distribution of maximum speeds, lane and car-following mobility model etc.) are configured.

4.2.3 Traffic Movement Simulation

Traffic movement is defined as the vehicles navigating across an intersection (from an entrance lane to an exit lane). In this chapter, we illustrate the agent's performance on four-legged standard intersections. We define a set of possible vehicle movements (Koonce & Rodegerdts, 2008). Based on the set of admissible vehicle movements, *signal phases* are configured (illustrated in Fig. 4.1). Vehicles follow the fundamental rules of motion (based on their mass, friction and other forces such as gravity) and react appropriately to their input parameters to navigate through the road networks. Vehicle spawn rate is regulated to reproduce real traffic data obtained at different times of day (such as AM rush hours, mid-day quiet hours and PM rush hours) ranging between very high traffic arrival rates at some instants (5000 cars/hour/lane) and no traffic at all at other instants (i.e. a situation where no cars are spawned on a road leading to the intersection). Vehicles can either go straight or turn right/left. Route selection probability is fully parameterizable in our simulator.

4.2.4 Learning Environment Setup: MDP Settings

At each MDP time-step, a signal control agent interacts with the traffic environment every t seconds (i.e the agent senses the prevailing traffic state using the *live* camera-feed, based on which it selects a certain signal phase configuration and implements it for t seconds). The smaller the t, the more often the agent will be asked to make a signal control decision. In this work, to ensure greater adaptiveness, we set t to 5s for all our experiments, including the baselines. It implies that at each MDP step, we have a minimum green signal time duration of 5s. After 5s elapses, based on the prevailing state of the traffic, the agents may decide to have the same signal phase configuration or change it. Real-world minimum/maximum signal time durations dictated by traffic regulation rules can also be conveniently accommodated by our simulation model. Following are the MDP settings for our signal control agent, including state, action spaces and reward design.

State Space

Our signal control agent operates *solely* on *live* camera footage to achieve signal control in real time. The agent visually perceives the current state of the traffic environment in and around the intersection it is controlling. For faster computation, we downsize the input images to a compact resolution of 100 x 100, having experimentally verified that this does not impair our agent's decision making.

Action Space

At each MDP time-step, our signal control agent selects one of the available phases (illustrated in Fig. 4.1), which is implemented for a duration of t seconds. We define a set of discrete actions A such that each computed action corresponds to each phase. For instance, an action a_1 corresponds to a phase p_1 (i.e. $< a_1 \mapsto p_1 >$). At each MDP time-step, given the current state of the traffic, the signal control agent's goal is to select the signal phase that best serves the existing traffic demand.

Reward Design

As reflected by transportation engineering literature, both delay and throughput are considered as acceptable metrics to evaluate the overall state of the traffic (Chakroborty & Das, 2017). Throughput and delay are inversely proportional to each other and optimizing one also optimizes the other. In this chapter, we focus on optimizing the traffic throughput across the intersections and subsequently, reducing the intersection traversal time for vehicles. We define two reward functions for this task (can be implemented individually or in combination): (1) a positive success reward (i.e. +1) for every civil vehicle passing safely through the intersection, and (2) a penalty (i.e. -1) for every civil vehicle waiting at the start of the intersection. Besides civil vehicles, we also include near-photorealistic emergency vehicles (such as ambulances, police cars and fire-trucks) in our experiments. We associate a higher reward of (i.e. +5) for their passing through the intersection and a higher penalty of (i.e. -5) for their waiting at the start of the intersection.

4.2.5 Learning Protocol

To explicitly learn an effective policy $\pi_\theta(a|s)$ via DRL that implicitly maximizes reward over all policies, our signal control agent is supported by a deep convolutional neural network (DCNN) as

a non-linear function approximator. An action a at time t can be drawn by:

$$a_t \sim \pi(s_t|\theta) \tag{4.1}$$

where θ denotes the model parameters and s_t is the 100 x 100 x 3 RGB image representing the current observation of the traffic environment. Based on the implemented actions and predefined reward function, the rewards are observed and gradients are computed, as per Eq. 4.2

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta log\pi_\theta(a_t^i|s_t^i)\right)\left(\sum_{t=1}^{T}r(s_t^i,a_t^i)\right) \tag{4.2}$$

where $J(\theta)$ denotes the loss function, T = 100 and N = 10.

A local maximum in $J(\theta)$ is searched by ascending the gradient of the policy with respect to parameters θ. $\nabla_\theta J(\theta)$ is the policy gradient and α is a step-size parameter. The policy is updated in the direction of the gradient (illustrated in Eq. 4.3) to encourage the actions leading to good outcomes and discourage less desirable ones.

$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta) \tag{4.3}$$

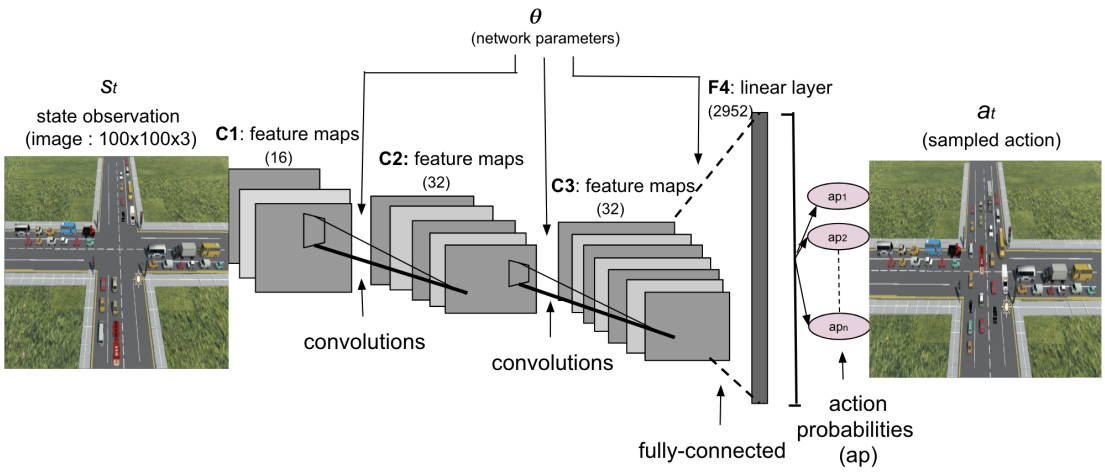
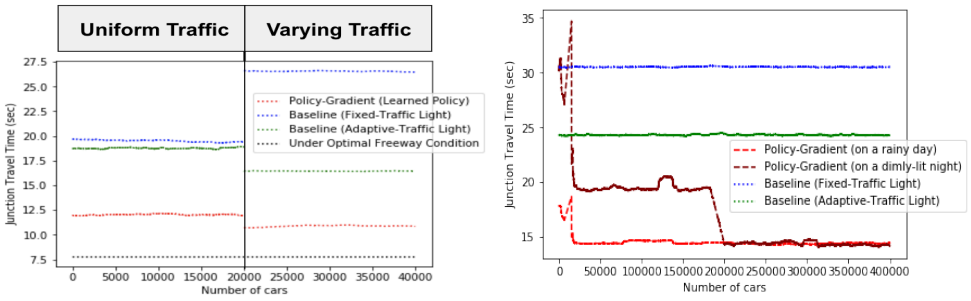


FIGURE 4.2: Signal Control Agent's Network Architecture.

4.2.6 Network Architecture

As Convolutional Neural Networks have demonstrated unprecedented success in accomplishing visual tasks (such as image classification and object recognition) (Krizhevsky et al., 2012), we



(A) (B)

FIGURE 4.3: Graphs depicting main experiments' results; cars' average junction travel time versus number of cars observed from the start of the experiment. (A) Uniform and constant traffic density (left), varying and random traffic tensity (right) based on a learned policy vs. fixed and adaptive traffic signal control baselines. (B) Signal optimization training plot on a rainy day and dimly-lit night vs. fixed and adaptive traffic signal control baselines.

use a deep convolutional neural network (DCNN) to implement our vision-based signal control agent. Our DCNN comprises of three convolutional layers (C1 with 16 output channels, C2 with 32 output channels and C3 with 32 output channels) and one fully-connected layer (F4 with 2952 neurons). We train this network with an RMSProp optimizer (Tieleman & Hinton, 2012) with a learning rate of 0.001. As illustrated in Fig. 4.2, the network takes an RGB image as input (depicting the current traffic state) and produces action probabilities as output (from which an action deciding the configuration of signal regimes is sampled). All our experiments in this chapter are based on this network architecture.

4.3 Main Experiments and Results

The goal is to develop a signal control agent that can independently optimize traffic flows through intersections in varied conditions. All our experiments (in the current and the following sections) are based on the network architecture described in Sec. 4.2.6 and illustrated in Fig. 4.2. Traffic environment specifications, including traffic model and flow details are outlined in Sec. 4.2.2 and Sec. 4.2.3 respectively. In these set of experiments, we select the following two performance metrics to evaluate our autonomous signal control strategy;

Junction Travel-Time is defined as the time interval between vehicles arriving at the junction stop-line and reaching at the end of the junction. We take the moving average of 100 vehicles' junction travel-time. Lower journey travel-time indicates better signal control.

Macroscopic Fundamental Diagram (MFD) is an established macroscopic transportation set of metrics - traffic density (veh/km), traffic flow (veh/hr) and speed (km/hr) (Geroliminis & Daganzo, 2008).

We compare our research findings against the following conventional widely-used signal control methods:

Standard (non-adaptive) signal control: follows the signal control policy that uses predefined signal phase regimes (widely used for steady traffic conditions) (Koonce & Rodegerdts, 2008).

Induction loop-based (adaptive) signal control: a loop detects approaching vehicles along each incoming lane, and an electronic impulse is sent to the signal circuit - switch the red signal to green.

Following is the set of main experiments analysing our DRL agent's performance:

4.3.1 Performance in uniform (constant) and varying (random) traffic density

Our DRL-based signal control agent learns an effective policy to optimize the flow of vehicles through an intersection after approximately half a million time-steps into training. We use the trained agent (i.e. learned policy) to demonstrate its efficacy in two different test settings (1) Uniform traffic generation in all directions. The vehicles are spawned with a fixed density distribution (1000 vehicles/hour/lane). (2) Varying traffic generation in each of the directions. The vehicles spawned follow a variable density distribution ranging between very high vehicle arrival rates at some instants (5000 vehicles/hour/lane) and no vehicles at other instants.

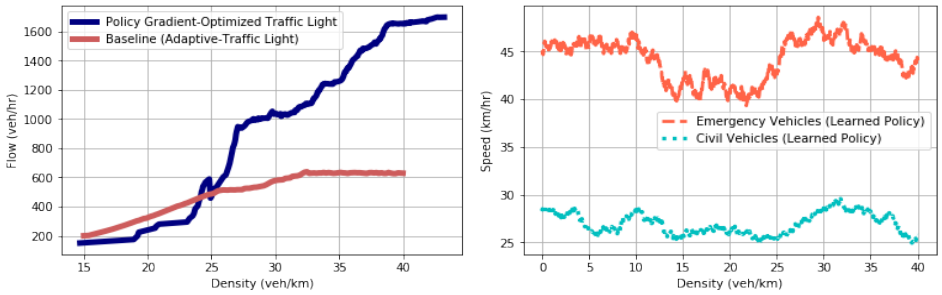
The graph shown in Fig. 4.3 (A) demonstrates our agent's performance based on average junction travel-time (y-axis). The number of cars (x-axis) represents the total number of cars spawned into the traffic scene during the evaluation phase (i.e. 20,000 cars each in uniform traffic scheme and varying traffic scheme). Ideally, the optimal delay for an individual vehicle is no delay at all. In order to create a more challenging benchmark to compare our research approach with, we consider junction travel-time under freeway optimal conditions, i.e. where each vehicle is able to travel through the junction as soon as it arrives at the start of the junction with no delay. This is plotted as 'Under Optimal Freeway Conditions'. Our signal control agent achieves a level of performance, significantly outperforming the conventionally-used signal control (adaptive/non-adaptive) baseline methods. These baseline methods fail to continuously modify the signal regimes based on the dynamically changing traffic flow patterns, as there is no

learning involved. As seen in Fig. 4.3 (A), our trained agent's performance does not significantly change with uniform and varying traffic density distribution. We attribute this performance to our agent's high adaptiveness to prevailing traffic conditions, as it sets the green signal for the minimum time the regulations stipulate (e.g. 5s in our experiments) and extends it in the following time-step, if needed, or else switches the signal phase depending on the prevailing traffic distribution. In contrast, as reported in the transportation literature (Chakroborty & Das, 2017), for real-world signal settings, maximum green time is usually set between 90-120 seconds, this may lead to implementing the green light for longer than it is needed.

4.3.2 Impact of adverse weather and dim-lighting conditions on the performance of our DRL-based signal control agent

Adverse conditions (such as bad weather) are known to be amongst the major causes leading to degradation in the efficiency of road networks; affecting the safety and mobility of travellers. The present-day transportation infrastructure is primarily designed to operate in normal conditions (such as a dry clear day) and is generally unable to effectively adapt to handle adverse conditions. Furthermore, from our vision-based signal control research perspective, real-time visual detection of traffic flows can be negatively affected by adverse weather and dim-lighting conditions. In our work, for the first time, we attempt to evaluate the extent to which our DRL-based signal control agent (operating on visual traffic data) is robust to unfavourable variations in its ambient conditions (weather and lighting). Our simulation platform; Traffic3D allows dynamic illumination to authentically simulate (illustrated in Fig. 3.3) different moments of the day (such as a sunny morning, evening dusk and a dimly-lit night) and weather conditions (such as rain and snow).

Figure 4.3 (B) records the performance of our agent in various lighting and weather settings and contrasts it to the baselines. This graph includes the learning phase, to illustrate the DRL agent's performance both during training for the new conditions and after. For our experiment, we consider rain of 10 mm/h. To sustain high image quality, we take into account the raindrops falling on the camera lens and have a mechanism in place to remove them. Heavy rain of 10 mm/h does not negatively affect our agent's ability to interpret the fundamental traffic scene (i.e. general junction layout and vehicle distribution). However, when exposed to a dimly-lit night, our agent starts to learn slowly as the dark pixels significantly impair visibility and the agent's ability to effectively perceive the traffic state. After processing an adequate number of data samples, the agent



(A) (B)

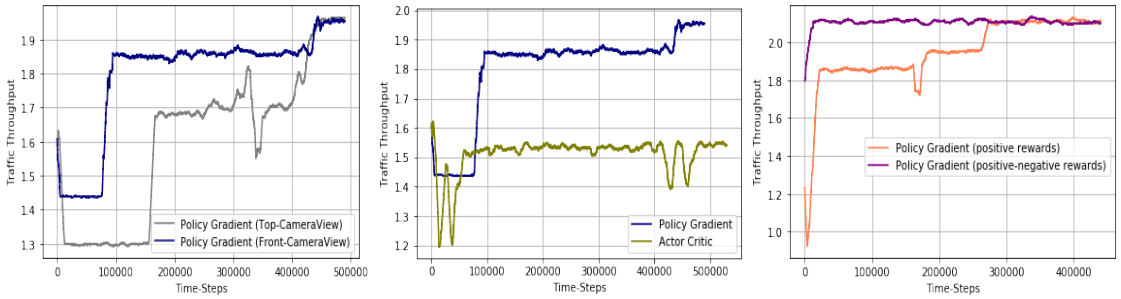
FIGURE 4.4: The MFDs demonstrating our DRL-trained (Policy Gradient) signal control agent's performance vs adaptive traffic signal control baseline. (A) Density vs Flow of all vehicles. (B) Density vs Speed of emergency vehicles against civil vehicles.

eventually reaches its peak performance and is able to effectively operate on the distorted pixels caused by low-lighting. Our vision-based DRL agent significantly outperforms both adaptive and non-adaptive baselines in adverse weather and lighting conditions as these methods do not adapt to the changing external conditions (such as weather and lighting).

4.3.3 Macroscopic Fundamental Diagram (MFD) (Geroliminis & Daganzo, 2008)

Here, we derive the relationships between established traffic macroscopic variables; (1) density and flow, and (2) density and speed. To accomplish this task, we collect downstream traffic data from our DRL-optimized signalized road intersection. After collecting the required traffic data (vehicle count and speed), we compute traffic density (i.e. number of vehicles per kilometer) and flow (i.e. number of vehicles per hour). We perform two sets of experiments in this setup (1) Demonstration of the relationship between density and flow for all vehicles, irrespective of their type and corresponding relevance (by relevance we highlight the significance of public transport and emergency vehicles over civil vehicles). (2) Demonstration of the relationship between traffic density and speed of emergency vehicles (such as ambulances, police cars and fire trucks) versus civil vehicles. After training the agent to prioritize the movement of emergency vehicles over civil vehicles, we collect the relevant macroscopic traffic variables and plot the relationship between combined density (including civil and emergency vehicles) versus emergency and civil vehicles' speed.

As shown in Fig. 4.4 (A), DRL-optimized intersection allows more efficient movement of vehicles as compared to induction loop-based adaptive traffic signal control. At the critical density



(A) (B) (C)

FIGURE 4.5: Graphs depicting our sensitivity analysis experiments' results (Sec. 4.4). (A) Top-Camera view vs. Front-Camera view. (B) Learning algorithm; Policy Gradient vs. Actor-Critic (C) Positive rewards vs. Positive-Negative rewards

(i.e. the maximum vehicle traffic that a road segment can effectively accommodate), the cumulative traffic flow through our DRL-optimized traffic signal control system is much higher than the cumulative traffic flow through the loop-induced adaptive baseline system. This density and flow relationship establishes our agent's competence in facilitating vehicles' swift navigation through the intersection-based on the extensive state of the traffic perceived using visual stimuli. Fig. 4.4 (B) demonstrates that our DRL-based signal control agent maximizes the cumulative reward by quickly switching the signal phases to prioritize the movement of emergency vehicles over civil vehicles (reflected by emergency vehicles' high cumulative speed as compared to civil vehicles' speed), given that emergency vehicles are associated with higher rewards than the civil vehicles (reward regime outlined in Sec. 4.2.4).

From the above results, it is perceivable that DRL-optimized vision-based signal control improves urban mobility; leading to better utilization of the existing transportation infrastructure.

4.4 Sensitivity Analysis of Important DRL Parameters: Experiments and Results

When applying DRL to optimize traffic flows through road intersections, knowing *a priori* which combination of - algorithm (whether an actor-only (Sutton et al., 2000) or actor-critic (Konda & Tsitsiklis, 2000)), reward signal and camera orientation (used to capture the visual input data) will yield a successful/sustainable signal control policy is virtually impossible. We conduct the

following sensitivity analysis to assess the robustness of our signal control agent to variation in pertinent DRL parameters used in Sec. 4.3.

As reflected by transportation engineering literature (Chakroborty & Das, 2017), both *traffic throughput* and *junction travel-time* (being inversely proportional to each other) are considered as viable measures to indicate the overall state of the traffic. To maximize the traffic throughput, it is paramount to decrease the average time a vehicle spends in an intersection (Au, Quinlan, Stiurca, Zhu, & Stone, 2010). To have further diversification in our research studies, the performance of our signal control agent is evaluated based on traffic throughput in the following set of experiments.

Traffic Throughput is defined as the number of vehicles that manage to pass through the intersection at each time-step. We take the moving average of 100 time-steps. A higher throughput corresponds to a larger number of vehicles passing through the intersection; indicating superior signal control.

Following is the set of our sensitivity experiments:

4.4.1 Top-Camera view versus Front-Camera View

Surging development of vision-based autonomous agents (i.e. agents operating on visual inputs) has pushed the need to devise agents that can process/understand their environment from different physical perspectives. When operating on visual inputs captured using different camera orientations, these agents may produce conflicting outcomes (P. Wang, 2016). In this experiment, we explore how different camera orientations can affect our signal control agent's ability to perceive the traffic environment and subsequently, regulate the traffic flows. Given an intersection, along with its front-view (45^o angle) (illustrated in Fig. 3.2 (A)), we also consider its top-view (illustrated in Fig. 3.2 (B)) and compare the learning curves. The signal control agent operates with a policy gradient (PG)-based RL method and a positive-only reward regime (i.e. +1/vehicle passing through the intersection) for both camera angle settings.

As seen in Fig. 4.5 (A), the agent learning from front-view camera images demonstrates more stable learning as compared to the agent learning from top-view images, however, both the agents reach their peak performance around the same time. This revelation reflects that vision-based autonomous agents' perception/understanding of their environment depends on the camera angle with which the input data is captured. However, as the agent rigorously explores its environment

and gains a breadth of experience, it is able to counteract the impact of less-effective camera orientation.

4.4.2 Policy-Gradient (Sutton et al., 2000) versus Actor-Critic (Konda & Tsitsiklis, 2000) Reinforcement Learning

To investigate the effectiveness of popular RL methods for our autonomous signal control task, along with implementing a policy gradient RL (outlined in Sec. 2.4.2) agent, we also implement an actor-critic RL (outlined in Sec. 2.4.3) agent. As our signal control agent demonstrated better performance when operating on visual input data captured with front camera-view (shown in Fig. 4.5 (A)), in this experiment, we consider front camera-view of the traffic environment and a positive-only (i.e. +1/vehicle passing through the intersection) reward regime.

Fig. 4.5 (B)) demonstrates the learning curves for both the RL algorithms. For the same MDP settings, policy gradient RL agent learns much better than the actor-critic agent. Policy-Gradient (actor only) methods are known to be more resilient to fast-changing non-stationary environments, where a critic (from actor-critic RL) is incapable to keep up with the time-varying nature of the environment and consequently, is unable to provide any useful information to the actor; thereby negating the advantages of actor-critic RL algorithm (Grondman, Busoniu, Lopes, & Babuska, 2012).

4.4.3 Positive Rewards versus Positive-Negative Rewards

Designing RL agents' reward signals that elicit desired behavior in an uncertain environment is considered a challenging task (Dewey, 2014). In this sensitivity analysis experiment, we evaluate the effectiveness of two different reward signal regimes (1) positive-only reward regime (i.e. +1/vehicle passing through the intersection), and (2) positive-negative reward regime (i.e. +1/vehicle passing through the intersection and -1/vehicle waiting at the start of the intersection). As our policy gradient-based signal control agent (illustrated in Fig. 4.5 (B)) indicated better performance when operating on visual input data captured with front camera-view (illustrated in Fig. 4.5 (A)), in this experiment, we capture the traffic environment with the front camera-view and implement a policy gradient-based signal control.

Fig. 4.5 (C) compare the throughput performance of the agent in each of the reward regimes. The agent with both positive and negative rewards learns much faster than the one with only

positive rewards. As per our observation, with just the positive rewards as the feedback, the agent may get stuck in local optima. In contrast, having both positive and negative feedback helps the agent to determine more effective policies to optimize traffic on a global level.

As deduced from the above sensitivity analysis experiments' results, to effectively optimize the traffic flows through intersections, the combination of visual traffic data captured with the front-camera view, policy gradient RL algorithm and positive-negative rewards work effectively. We have used this combination of DRL parameters to conduct experiments in Sec. 4.3.

4.5 Summary

In this chapter, we presented an end-to-end trainable, fully-actuated autonomous traffic signal control agent. To fairly validate our vision-based research idea, we used a novel traffic simulation environment; Traffic3D (Garg et al., 2019b, 2019c) to conduct our experiments. Compared to the popular state-of-the-art traffic simulation tools, our simulation platform is relatively more realistic (in terms of both physical and visual properties), adequately capturing the reality of traffic scenarios. We believe that the ability to train our signal control agent in a realistic environment is key in making it possible to deploy the agent in real world settings. Our simulation results demonstrate that our signal control agent can effectively perceive the traffic situation in and around an intersection using visual sensory data captured in real time. It continuously modifies the traffic signal regimes, as per changing observations and is significantly more robust than existing signal control methods. In contrast to the conventionally-used signal control methods that are more effective in settings with single dominant traffic flow, our signal control agent optimizes multiple competing traffic flows that dynamically alter through the day.

Chapter 5

Multi-agent Deep Reinforcement Learning for Traffic Optimization through Multiple Road Intersections using *Live* Camera Feed

This chapter addresses the *research question 3* (outlined in Sec. 1.1), using *Actor-Critic* RL algorithm (described in Sec. 2.4.3). In Chapter 4, we presented a fully-adaptive signal control agent that directly responds to the actual traffic conditions around a single intersection; achieving effective signal control in the face of complex, imprecise traffic environments. However, RL agents learn not only by trial-and-error but also through collaboration/cooperation among each other. Many RL tasks (such as autonomous driving and robotic manipulation) can be naturally modelled as cooperative multi-agent systems. RL agents attempting to single-handedly solve these tasks perform poorly, as their joint state and action spaces grow exponentially, leading to dimensionality explosion. Using RL to achieve coordinated behavior among agents operating in an environment is considered beneficial for achieving robustness and generality (Sen, Sekaran, Hale, et al., 1994).

In this chapter, to further establish the resilience of our DRL-based signal control approach, we investigate the utilization of multi-agent DRL to real-time adaptive signal control through a *network* of road intersections. The current work, for the first time, establishes network-level coordination between multiple DRL-based signal control agents operating on visual traffic data. The agents *solely* operate on camera feed to optimize an aggregate of traffic flows through a network of

Research Study	State Space	Algorithm	Reward Scheme (described in Sec. 5.2.7)
Wiering (Wiering, 2000)	Cumulative wait time	Q-learning	Global
Kuyer et al. (Kuyer, Whiteson, Bakker, & Vlassis, 2008)	Position of vehicles	Q-learning	Local
Arel et al. (Arel, Liu, Urbanik, & Kohls, 2010)	Delay	Q-learning	Local
Pol et al. (Van der Pol & Oliehoek, 2016)	Position of vehicles	Q-learning	Global
Chu et al. (Chu, Wang, Codecà, & Li, 2019)	Cumulative delay & total number of vehicles on each lane	A2C	Global
Our study	**Raw pixels**	**Actor-critic**	**Local**

TABLE 5.1: Summary of relevant reinforcement learning-based multi-intersection traffic signal control optimization research studies.

intersections). Incorporating the concepts and perspectives from recent work in the field of multi-agent planning (Kraemer & Banerjee, 2016), (Foerster, Farquhar, Afouras, Nardelli, & Whiteson, 2018), to achieve network-level coordination between individually operating local signal control agents, we apply an *actor-critic* (Konda & Tsitsiklis, 2000) RL approach. We implement a centralised critic that enables global learning, while each actor's execution is local. To achieve network-level optimality, the centralised critic operates on all available state information (i.e. concatenation of local states of the collaborating signal control agents). In contrast, each actor (i.e. each participating signal control agent) operates exclusively on its limited local observation of the environment. We compare our proposed centralised learning method against baseline methods: (1) fully-decentralised learning (outlined in Sec. 5.3.1), (2) fully-independent learning (outlined in Sec. 5.3.2), and (3) loop-induced signal control (outlined in Sec. 5.3.3). Our experiment-based evaluations reveal that our research approach leads to a positive emergence of coordinated behavior between individual signal control agents; resulting in significant performance improvement over above-mentioned baseline methods.

5.1 Related Work

Only a handful of studies address signal control optimization through multiple intersections. In (Wiering, 2000), tabular Q-learning is applied to each intersection in a multiple intersection traffic setting. This work is further extended in (Chu et al., 2016), in which traffic regions are dynamically clustered to improve observability. In (Aziz et al., 2018), both Q-learning and SARSA are used, with traffic state observability enhanced using neighbourhood information sharing. Tantawy et al. (El-Tantawy et al., 2013) implemented a heuristic communication between tabular Q-learning-based intersection control agents, in which each message consisted of the estimated neighboring agents' signal control policies. Chu et al. (Chu & Wang, 2017) used the max-sum communication for Q-learning-optimized intersections, in which each message signified the impact of the neighbouring intersection on each local Q-value. Most of these research studies implement value function-based approaches (Q-learning) for multi-intersection signal control. Value function-based methods are often criticized for being unstable and in practice are difficult to use. They are inclined towards finding deterministic policies, whereas, in a dynamic environment like traffic, an effective policy is expected to be stochastic (Sutton et al., 2000).

Closest to our work (Chu et al., 2019), traffic is optimized through a network of intersections in a decentralized fashion. The authors devise a fully-decentralized multi-agent signal control method. In each local agent's state observation information, observations and fingerprints of neighboring agents are included such that each local agent is more aware of regional traffic distribution. In contrast, we implement centralized critic and decentralised actors to perform centralised learning and decentralised execution. Furthermore in (Chu et al., 2019), handcrafted traffic state features (i.e. cumulative delay of first vehicle and number of vehicles approaching an intersection within 50m range to the intersection) are used. Our multi-intersection signal control methodology is end-to-end trainable and, to our knowledge, is the first to depend *solely* on camera feed for traffic optimization in real time.

5.2 Our Autonomous Multi-Intersection Signal Control Methodology

In this section, we describe the complete implementation of our signal control agents in multi-intersection settings, including the MDP (state, action, reward) specifications.

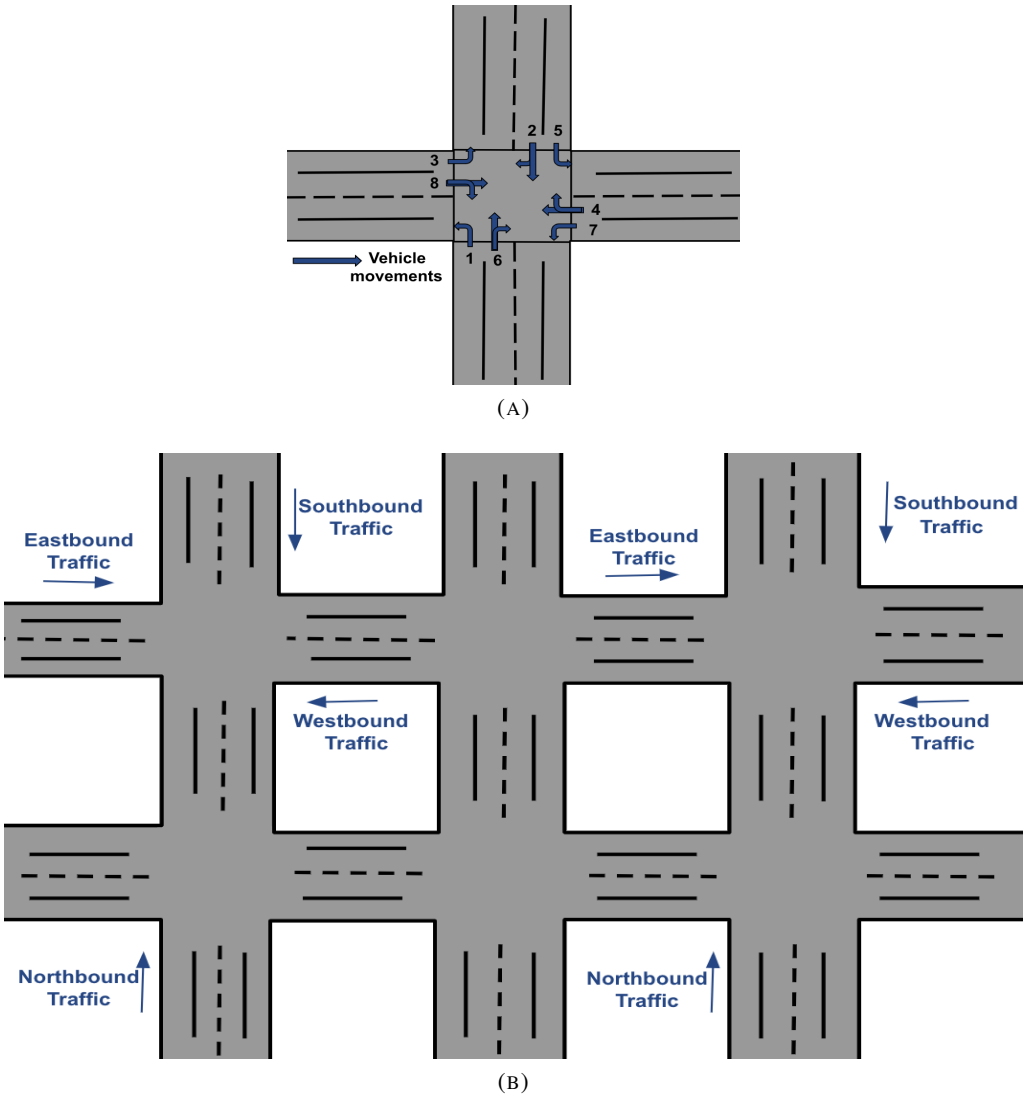


FIGURE 5.1: (A) Possible Signal Phases and Vehicle Movements. (B) An illustration of Intersection Grid.

5.2.1 Problem Formulation

Under our research methodology, we consider a multi-intersection road network scenario in which a set of signal control agents act in parallel. Each signal control agent controls one intersection in the network by directly mapping RGB images (describing the prevailing traffic state) to actions (controlling the traffic signals). Our goal is to achieve effective coordination of agents' actions such that their joint utility is maximized.

5.2.2 Traffic Model Simulation

All the experiments presented in this paper are conducted using our novel traffic simulation environment; Traffic3D (Garg et al., 2019b, 2019c). We simulate 3D four-way intersection scenarios with microscopic traffic properties. Instead of modelling the aggregate behaviour of traffic flow, each vehicle is simulated individually to captures its visual and physical properties with high-degree of realism. Vehicles follow the fundamental rules of motion (based on their mass, friction and other forces such as gravity) and react appropriately to their input parameters, to navigate through the network. Vehicle spawn rate is regulated to mimic different times of day (such as AM/PM rush hours and mid-day quiet hours).

5.2.3 Traffic Movement Simulation

Traffic movement is defined as the vehicles navigating across an intersection (from an entrance lane to an exit lane). Based on real-world guidelines, we define a set of possible, non-conflicting vehicle movements to allow their safe passage through the intersections (illustrated in Fig. 5.1 (A)) (Koonce & Rodegerdts, 2008). Signal phases are configurable and it is possible to have simultaneous execution of more than one phases. Vehicles can either go straight or turn right/left, route selection probability is parameterizable in our simulator. Fig. 5.1 (B) illustrates our multi-intersection network grid. Each intersection is a four-way intersection.

5.2.4 Learning Environment Setup: MDP Settings

At each MDP time-step, concurrently operating signal control agents interact with the traffic environment every t seconds (i.e the agents sense the prevailing state of the traffic environment using the visual data, based on which they configure traffic signals in real time for t seconds). The smaller the t, the more often the agents will be asked to make signal control decision. In the current work, to ensure greater adaptiveness, we set t to 10s, which implies that at each MDP step, we have a minimum green signal time duration of 10s. After 10s elapses, based on the prevailing state of the traffic, the agents may decide to have the same signal phase configuration or change it. Real-world minimum/maximum signal time durations dictated by traffic regulation rules can also be conveniently accommodated by our simulation model. Following are the MDP settings for our signal control agent, including state, action spaces and reward design.

State Space

Each actor (i.e local signal control agent) operates *solely* on camera footage to achieve signal control in real time. Actors only perceive the current state of the traffic environment in and around the intersections that they are controlling. In contrast, the centralised critic operates on the global state of the traffic (i.e. concatenation of local observations of all actors). For faster computation, we downsize the input images to a compact resolution of 100 x 100, having experimentally verified that this does not impair our agents' decision making.

Action Space

At each MDP time-step, each signal control agent selects one of the available phases, to be implemented for a duration of t seconds. Based on the set of admissible vehicle movements, *signal phases* are configured (illustrated in Fig. 5.1 (A)) (Koonce & Rodegerdts, 2008). We define a set of discrete actions A such that each computed action corresponds to each phase. For instance, an action a_1 corresponds to a phase p_1 (i.e. $< a_1 \mapsto p_1 >$). At each MDP time-step, given the current state of the traffic, the signal control agents share the common goal to select the signal phases that best serves the existing traffic demand.

Reward Design

To evaluate/optimize the overall efficiency of road networks, both delay and throughput are considered as acceptable metrics (Chakroborty & Das, 2017). In our work, we focus on optimizing joint traffic throughput across a network of intersections and subsequently, reducing the average time a vehicle spends in an intersection. To accomplish this task, we define two reward functions: (1) a success reward of +1 for every vehicle passing safely through an intersection; and (2) a penalty of -1 for every vehicle waiting at the start of an intersection.

5.2.5 Network Architecture

Our actor-critic network framework is illustrated in Fig. 5.2. Given the nature of input data (i.e. vision-based), both our actor and critic networks comprise three convolutional layers (*Conv1*, *Conv2* and *Conv3*). Along with the convolutional layers, our critic network includes a linear layer (*Linear4*). In contrast, in our actor network, we use long-short term memory (*LSTM*) as the last hidden layer to memorize a short history. Traffic flows form a complex spatial-temporal structure,

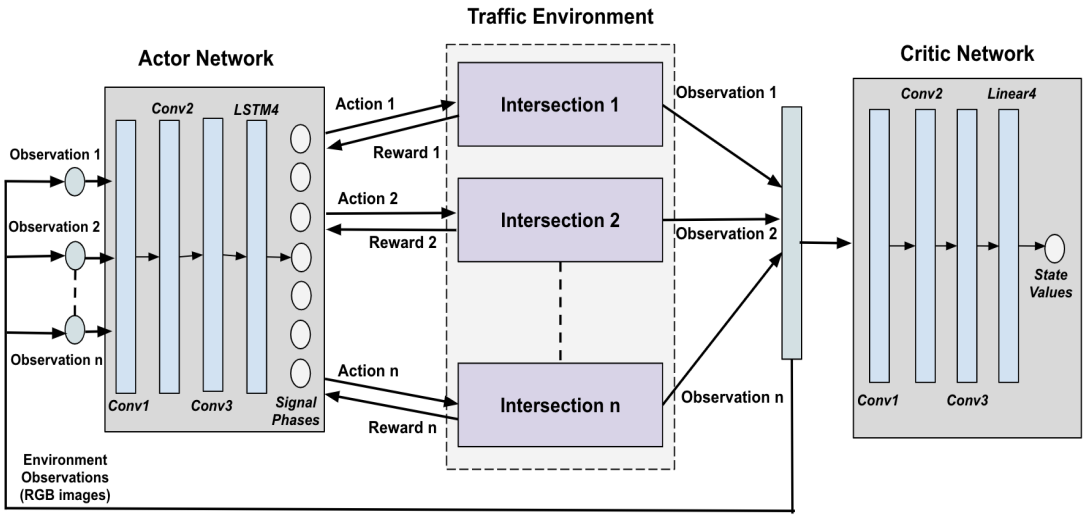


FIGURE 5.2: Our Multi-Intersection *Actor-Critic* Network Framework. We use network parameter sharing (described in Sec. 5.2.6) to implement one *actor* and one *critic* network, which is shared by all the agents.

resulting in non-stationary MDP if the agents do not have access to any previous data to rely on. *LSTM* networks provide an implicit memory that improves performance in partially-observable environments. As seen in Fig. 5.2, the actor network takes an RGB image as input (depicting the current traffic state of a signal control agent) and produces action probabilities as output (from which an action deciding the signal phase is sampled). The critic network takes an RGB image as input (depicting the current traffic states of all participating signal control agents) and produces state values as output.

5.2.6 Network Parameter Sharing

Agents may learn successful policies more efficiently using parameter sharing, as it allows simultaneous learning, based on all agents' experiences. Furthermore, parameter sharing enables the large-scale application of the proposed multi-intersection optimization approach, as it is infeasible to have a separate actor and critic network for each intersection in a multi-intersection scenario. In the current work, to improve learning efficiency and economise on training time, the agents are allowed to share parameters among each other, i.e. we implement one actor network and one critic network, which are shared by all the agents (illustrated in Fig. 5.2). However, the agents still demonstrate their respective independent behaviors, as each agent receives different observations based on the prevailing traffic situation in and around the intersection it is controlling.

5.2.7 Single Agent Credit Assignment in a Multi-Agent Environment

One of the primary challenges of multi-agent environments is marginalization of each agent's individual contribution towards a global reward. In a recent multi-intersection signal control implementation (Chu et al., 2019), at each time-step, all signal control agents receive the same global reward (i.e. total aggregated reward through a network of intersections); keeping them oblivious to their true individual contribution towards network-level traffic optimization. In contrast, in the current work, the agents operating under both, our proposed centralised learning method (Sec. 5.2.8) and baseline methods (Sec. 5.3) are allowed to observe their individual local rewards. From our research perspective, deducing each signal control agent's individual reward is fairly straightforward. Almost all real-world traffic intersections are equipped with induction loops or rely on cameras, which are used to count the vehicles. Since our reward signal includes traffic throughput; thus deducing each signal control agent's independent contribution towards the global network reward is possible.

5.2.8 Centralised Signal Control Learning Protocol (our method)

Within urban road networks, following a decentralised framework, any local signal control agent might be susceptible to *myopic* signal control decisions, that work effectively locally but fail to globally optimize traffic on the network level. To avert this possibility we implement an actor-critic approach such that our critic is centralised; conditioning on the combined observations of all the actors to output a consensual value estimate. While each actor (i.e. each signal control agent) acts independently-based on its private, local observation of the traffic environment without knowing the state of other actors (illustrated in Fig. 5.2). Our actor network (shared to all actors) represents the policy π, parameterized by θ. Given a team of actors (i.e. signal control agents) consisting of N agents, let $\mathbf{u} = \{u_1, ..., u_N\}$ represents all agents' actions and $\mathbf{o} = \{o_1, ..., o_N\}$ represents all agents' observations. The gradient of the expected return for an agent i, $J(\theta) = \mathbb{E}[R]$ is represented as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\mathbf{s}\sim\rho,\mathbf{u}\sim\pi(\mathbf{s})}[\nabla_\theta \log \pi_\theta(\mathbf{u}|\mathbf{o}) V^\pi(\mathbf{o})], \tag{5.1}$$

where, V^π represents a centralised critic network that takes as input the concatenated state information of all the active agents and outputs a centralised state value (i.e it produces a single

state-value function after considering the observations of all agents). Based on the following equation, the policy is updated:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta), \tag{5.2}$$

where, α is a step-size parameter.

5.3 Baselines for Comparison

We compare our multi-intersection signal control strategy (described in Sec. 5.2.8) with both RL-based and conventional signal control-based methods.

5.3.1 Fully-Decentralized Signal Control using Augmented State and Local Rewards (Chu et al., 2019)

In contrast to our centralised signal control methodology (outlined in Sec. 5.2.8), here we implement a completely decentralized protocol for traffic optimization through a network of intersections. Signal control agents communicate with one another in the absence of any central controller. At each MDP time-step, each signal control agent independently executes an action based on its local information and the information shared by its neighbors, which increases the observability of each local agent. This approach, using information sharing, aims at diffusing local state observations of each agent across the network of intersections.

5.3.2 Fully-Independent Signal Control using Local State and Local Rewards (Konda & Tsitsiklis, 2000)

A straightforward method to implement actor-critic DRL for autonomous signal control is to have each signal control agent control its individual intersection by independently learning its own policy and the corresponding state-value function. Learning is independent in this setup, without any central controller or interaction between local agents. At each MDP time-step, both actor and critic networks operate on the same local observations.

5.3.3 Loop-Induced Signal Control (no learning involved) (Koonce & Rodegerdts, 2008)

Lastly, we compare our research findings against the standard induction loop-based adaptive signal control (Koonce & Rodegerdts, 2008). In loop-induced adaptive signal control, a loop detects approaching vehicles along each incoming lane that are idling overhead (within 50m to the junction) and an electronic impulse is sent to the signal circuit - to switch the red light to green.

All our baseline methods use same configuration of signal regimes, illustrated in Fig. 5.1 (A)).

5.4 Evaluation Metrics

We define the following performance metrics used to evaluate our research findings;

5.4.1 Traffic Throughput

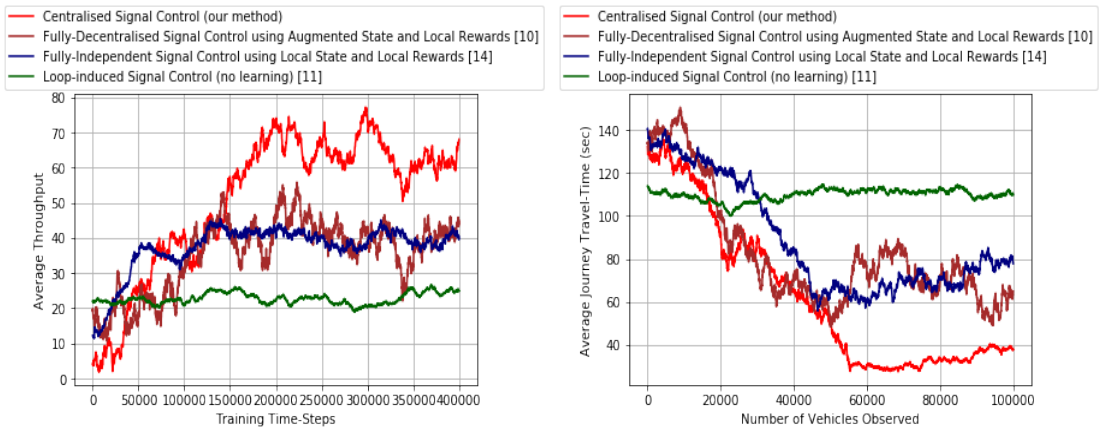
At each MDP time-step, traffic throughput gives the aggregate number of vehicles that manage to pass through the network of intersections. Higher throughput corresponds to a larger number of vehicles passing through the intersections; indicating a superior multi-intersection signal control method.

5.4.2 Journey Travel-Time

At each MDP time-step, journey travel-time is defined as the time interval between vehicles arriving at an intersection stop-line and reaching the end of the intersection. Lower journey travel-times indicates a better multi-intersection signal control method.

5.5 Experiments and Results

We simulated multi-intersection traffic environment with time-variant traffic flows. A view of the traffic environment used for experimentation is illustrated in Fig. 3.1. To our knowledge, this is the first study considering the optimization of traffic flows through multiple intersections based on near-photorealistic visual traffic data. In this section, we empirically investigate the performance of our multi-intersection signal control strategy (outlined in Sec. 5.2.8) in contrast to various



(A) (B)

FIGURE 5.3: Graphs demonstrating our centralised signal control method vs baselines (fully-decentralised, fully-independent and loop-induced signal control). (A) Average Throughput. (B) Average Journey Travel Time.

relevant baseline methods (outlined in Sec. 5.3). The evaluation metrics used in all experiments are outlined in Sec. 5.4.

5.5.1 Centralised signal Control (our method)

Following the framework illustrated in Fig. 5.2 and discussed in Sec. 5.2.8, we implement centralised learning of decentralised policies. At each MDP time-step, our centralised critic network acts on the global traffic state i.e. concatenation of local states of all signal control agents. In contrast, every actor (i.e. every signal control agent) acts on its individual local observation of the prevailing traffic state in and around the intersection under consideration. As seen in Fig. 5.3 (A) and Fig. 5.3 (B), average traffic throughput (red line) is highest and average junction travel-time (red line) is lowest using our centralised signal control strategy. Centralised critic acting on the global traffic state observation can perceive the overall state of the traffic environment at the network level (i.e. around the network of intersections). With respect to the value function (outputted by centralised critic network), agents efficiently determine the jointly optimal actions (i.e. signal control agents harmonize to maximize the total return). Furthermore, since the centralised critic is aware of the traffic distribution at the network level, this mitigates the known non-stationarity problem of multi-agent environments. Our results signify that multiple agents operating in the same environment do not always require to have an explicit communication amongst themselves to learn coordinated behaviors. This implies that collaboration/cooperation is still possible without

information sharing among the agents.

5.5.2 Fully-Decentralized Signal Control using Augmented State and Local Rewards

In this setup, based on the method discussed in Sec. 5.3.1, we implement fully-decentralised learning of agent (actor) policies. Each local agent has access to augmented state information, including regional traffic distribution and cooperative strategy, i.e. observations and fingerprints (current policy) of neighboring agents. At each MDP time-step, both actor and critic networks receive as inputs, this augmented state information. As seen in Fig. 5.3 (A) and Fig. 5.3 (B), both average traffic throughput (brown line) and average junction travel time (brown line) get worse in a decentralised learning scenario. This indicates that having access to information from neighboring agents is not always beneficial, as it can interfere in learning. Furthermore, agents having access to their individual rewards in a decentralised multi-agent environment can become greedy and tend not to sacrifice for the greater good. Therefore, having a centralised controller with a wider picture of the environment may be useful. In general, finding a globally optimal solution for multiple agents operating with partial information of their environment without any central overseer is considered intractable (Bernstein, Givan, Immerman, & Zilberstein, 2002).

5.5.3 Fully-Independent Signal Control using Local State and Local Rewards

In this setup, based on the method discussed in Sec. 5.3.2, the learning is completely independent without any central controller or any communication between the agents. Agents only operate on their local observations. At each MDP time-step, both actor and critic networks are fed with the same local observations of the traffic environment. As seen in Fig. 5.3 (A) and Fig. 5.3 (B), independent learning results in both, average traffic throughput (blue line) and average junction travel time (blue line) getting as worse as in the case of decentralised learning. This indicates that agents learning independently-based on their local field-of-view are susceptible to myopic decisions which leads to overall inferior performance of the signal network.

Loop-induced signal control (green line) performs the worst in all cases, as this method fails to; (1) extensively view the traffic environment due to induction loops' narrow operational range, and (2) continuously modify agents' traffic optimization decisions-based on the dynamically changing traffic flow patterns, as there is no learning involved.

5.6 Summary

Multi-agent systems are increasingly finding applications across a wide range of domains, including robotics, autonomous driving and telecommunications, among others. Majority of these tasks involve sequential decision-making and require agents to learn behaviours online. A significant part of the research on multi-agent learning is based on reinforcement learning methods. RL can provide natural and robust coordination between agents in multi-agent systems. In this chapter, we examined the factors that can influence, either positively or negatively, the dynamics of our DRL-based learning protocol in a multi-agent setting. We introduce a novel formulation of signal optimization task to facilitate dynamic configuration of effective signal regimes in a multiple-intersection scenario. To our knowledge, this work presents the first application extending DRL methods to optimize traffic through multiple intersections-based *solely* on visual traffic data, without hand-crafted traffic state features. We demonstrate a centralised controller that is able to bring about a principled learning strategy between the signal control agents, resulting in positive emergence of cooperative behavior among them in a scenario where each agent has access only to the partial state of the traffic environment.

Chapter 6

Transferable Vision-based Traffic Signal Control using Deep Reinforcement Learning

In this chapter, we address the *research question 4* (outlined in Sec. 1.1) using a transfer learning-based approach. Teaching an agent to autonomously act in an unseen, dynamic 3D-environment is a challenging endeavour. Agents are required to process and derive effective representations of their high-dimensional environment. Deep Reinforcement Learning agents evolved significantly over the past few years; exhibiting their potential in achieving mastery in solving complex games such as Go and Atari. However, generalizability is still considered as a prominent problem in training these agents to become self-sufficient to the vast environment (such as urban traffic) they are exposed to. To be able to generalize well and sustain good performance in an unseen diverse environment, an agent must generalise past experiences to new situations and be resilient to (1) low-level variations in the environment such as texture, colour and shapes of objects and (2) high-level variations such as a completely different environment layout. Transfer learning exhibits the possibility of knowledge transfer between tasks such that an agent's previously-acquired knowledge/experience while performing a task within either a simulator or the natural environment can be effectively reused to perform other tasks. Evaluation of an autonomous agent's knowledge transfer skills is crucial, as a DRL agent is only capable of acting effectively if it can abstract the environment's fundamental features through its individual experience.

Furthermore, a DRL agent can be slow to learn. Since it has no prior knowledge of its intended environment, it is bound to have a large number of interactions with the environment to learn a

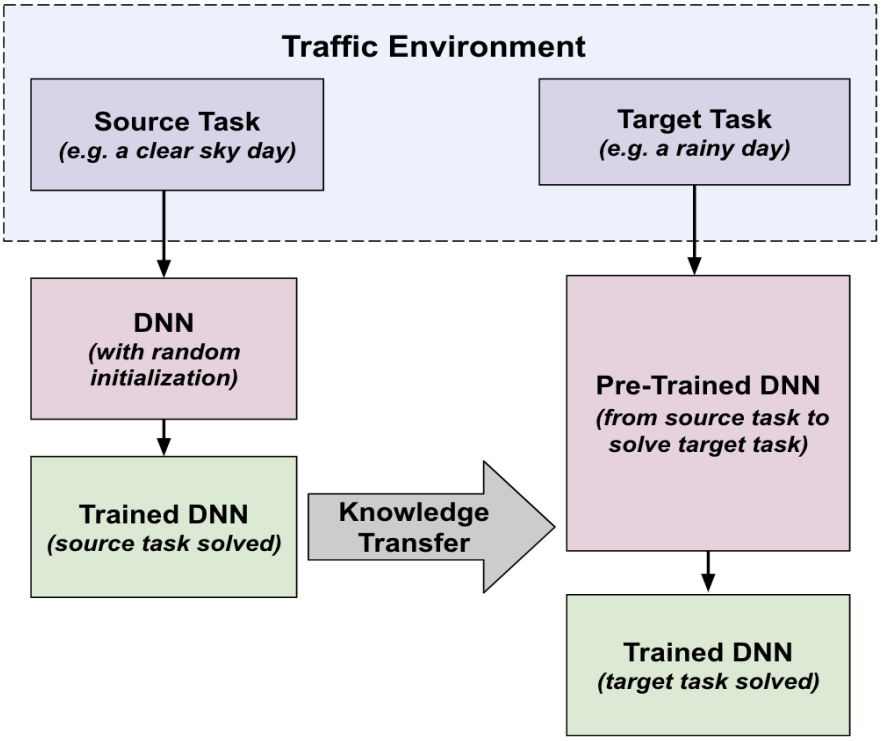


FIGURE 6.1: Transfer Learning Mechanism Pipeline (pre-trained DNN (Deep Neural Network) from source task to solve target task).

suitable policy. However, this problem can be alleviated by utilizing previously-learned knowledge and transferring it to solve a new related task. Transfer learning can alleviate the need for training an agent from the beginning, such that an agent can build on from what it already knows. In the current work, our goal is to build a *generalist* signal control agent that can independently operate in a rich traffic environment with dynamically-changing (visually different) conditions (such as varying weather and lighting conditions), without re-learning from the beginning for each condition. Our transfer learning pipeline is illustrated in Fig. 6.1. Our empirical results demonstrate our agent's high transferability/generalizability and accelerated-learning skills. When encountering a new traffic situation, our single signal control agent leverages previously-learned knowledge accumulated across a series of experiences to optimize traffic flows.

6.1 Related Work

Knowledge transfer is a key technique employed to facilitate the development of autonomous agents that operate effectively in multiple-related environments, such that they can efficiently

transfer their previously-acquired knowledge to new unseen situations. Transfer learning was pioneered to achieve dimensionality reduction by transferring knowledge from a generative to a discriminative model (Hinton & Salakhutdinov, 2006). Since then it is being studied extensively by the AI community (Taylor & Stone, 2011; Silver, Yang, & Li, 2013). However, transfer learning to ensure reliable generalizability and accelerated-learning for a vision-based signal control agent has not been previously explored. To our knowledge, the current work, for the first time demonstrates the use of transfer learning to devise a vision-based signal control agent that can sustain effective performance when exposed to various dynamic traffic situations (such as sudden degradation in weather and lighting conditions), while optimizing on learning time. The use of transfer learning makes our agent readily deployable from simulation to real-life, as well as across junction layouts, weather and lighting conditions.

6.2 Our Signal Control Agent's Knowledge Transfer Methodology

In this section, we describe our signal control agent's knowledge transfer protocol, including MDP settings - state, action, reward specifications.

6.2.1 Problem Formulation

Here, the objective is to develop a transferable, fully-actuated agent that learns to autonomously control traffic signals in real time based-*solely* on *live* footage of the traffic situation around the area the signals affect. Traffic distribution, lighting and weather conditions are variable. The signal control agent should be able to continue operating effectively. Under our transfer learning research methodology, our signal control agent learns to perform effectively in newly-encountered traffic conditions by leveraging knowledge gained from former experiences.

6.2.2 Traffic Model Simulation

A practical approach is to deploy the agents to the real world after training them in simulation. All the experiments presented in this chapter are conducted using our novel traffic simulation environment; Traffic3D (Garg et al., 2019b, 2019c). For the current work, we use a variety of 3D-traffic scenes, including a clear day, a rainy day, a snowy day and dimly-lit night (illustrated in Fig. 3.3).

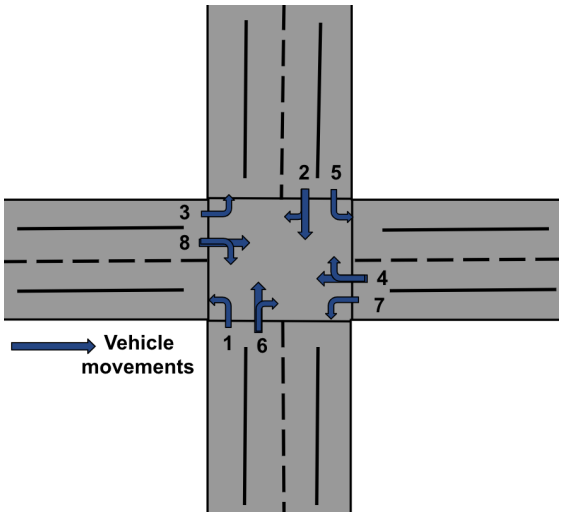


FIGURE 6.2: Possible Signal Phases and Vehicle Movements.

6.2.3 Traffic Movement Simulation

Traffic movement is defined as the vehicles navigating across an intersection (from an entrance lane to an exit lane). In this chapter, we illustrate the agent's transfer performance on four-legged standard intersections under varying ambient conditions. We define a set of admissible vehicle movements, based on which *signal phases* are configured (illustrated in Fig. 6.2) (Koonce & Rodegerdts, 2008). Vehicle spawn rate is regulated to mimic different times of day (such as AM rush hours and mid-day quiet hours and PM rush hours). Vehicles can either go straight or turn right/left. Route selection probability is parameterizable in our simulator.

6.2.4 Learning Environment Setup: MDP Settings

For our transfer learning experiments, at each MDP time-step, the signal control agent interacts with the traffic environment every t seconds. To ensure greater adaptiveness, we set t to 5 seconds, which implies that we have a minimum green signal time of 5 seconds, while the maximum green signal time is decided by our agent depending on the prevailing traffic distribution around the intersection. Following are the MDP settings for our signal control agents, including state, action spaces and reward design.

State Space

Our transfer signal control agent operates *solely* on *live* camera footage to achieve signal control in real time. The agent visually perceives the current state of the traffic environment in and around the

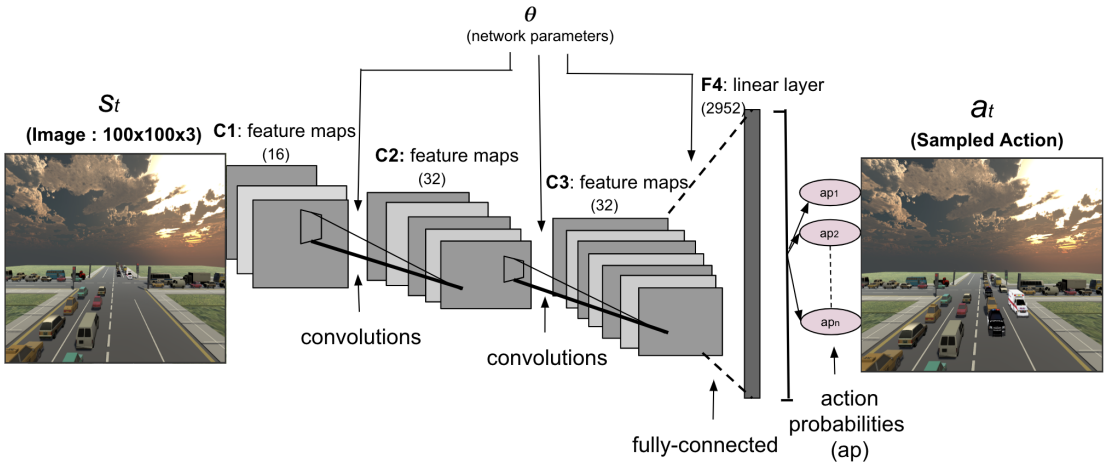


FIGURE 6.3: Signal Control Agent's Transfer Learning (for fine-tuning) Network Architecture.

intersection it is controlling. For faster computation, we downsize the input images to a compact resolution of 100 x 100.

Action Space

Our signal control agent selects one of the available phases (P) (illustrated in Fig. 6.2, based on real-world intersection lane rules (Koonce & Rodegerdts, 2008)), which is implemented for a duration of t seconds. We further define a discrete action space (A) such that each computed action corresponds to each phase. For instance, an action a_1 corresponds to a phase p_1; $< a_1 \mapsto p_1 >$. At each time-step, given the current state of traffic, the goal of our signal control agent is to select the signal phase that maximizes the traffic throughput.

Reward Design

Throughput and delay (the acceptable traffic state evaluation metrics) are inversely proportional to each other and optimizing one also optimizes the other. In this chapter, we focus on optimizing the traffic throughput across the intersections and subsequently, reducing the intersection traversal time and delay for vehicles; a task for which we use a combination of two reward functions (1) a success reward of +1 for every civil vehicle passing safely through the intersection and (2) a penalty of -1 for every civil vehicle waiting at the start of the intersection. Besides civil vehicles, we also include emergency vehicles (such as ambulances, police cars and fire-trucks) in

our experiments. We associate a higher reward of +5 for their passing through the intersection and a higher penalty of -5 for their waiting at the start of the intersection.

6.2.5 Source Task Learning Protocol

To explicitly learn an effective policy $\pi_\theta(a|s)$ via DRL that implicitly maximizes reward over all the policies, our signal control agent is supported by a deep convolutional neural network (DCNN) for a non-linear function approximation. An action a at time t can be drawn by:

$$a_t \sim \pi(s_t|\theta) \tag{6.1}$$

where θ denotes the model parameters and s_t is the 100 x 100 x 3 RGB image representing the current observation of the traffic environment. Based on the implemented actions and predefined reward function, the rewards are observed and gradients are computed, as per Eq. 6.2,

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\Big(\sum_{t=1}^{T}\nabla_\theta log\pi_\theta\big(a_t^i|s_t^i\big)\Big)\Big(\sum_{t=1}^{T}r\big(s_t^i,a_t^i\big)\Big) \tag{6.2}$$

where $J(\theta)$ denotes the loss function, T = 100 and N = 10.

A local maximum in $J(\theta)$ is searched by ascending the gradient of the policy with respect to parameters θ. $\nabla_\theta J(\theta)$ is the policy gradient and α is a step-size parameter. The policy is updated in the direction of the gradient (illustrated in Eq. 6.3) to encourage the actions leading to good outcomes and discourage the less desirable ones.

$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta) \tag{6.3}$$

6.2.6 Transfer Learning Protocol

To evaluate our vision-based agent's generalizability, we define (a) a source task and (b) a target task. We train our agent to solve the source task (based on the learning protocol outlined in Sec. 6.2.5) and transfer its acquired knowledge from the source task to solve the target task. To do so, we initialize our target task's deep neural network (DNN) with our pre-trained source task's CNN's parameters (illustrated in Fig. 6.1). The agent is then tuned to solve the target task, based

on Eq. 6.4,

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\Big(\sum_{t=1}^{T\times i}\nabla_\theta log\pi_\theta\big(a_t^i|s_t^i\big)\Big)\Big(\sum_{t=1}^{T\times i}r\big(s_t^i,a_t^i\big)\Big) \tag{6.4}$$

where T = 10 and N = 10. The policy is updated in the direction of the gradient based on Eq. 6.3.

6.2.7 Network Architecture

Our deep CNN comprises of three convolutional layers (C1 with 16 output channels, C2 with 32 output channels and C3 with 32 output channels) and one fully-connected layer (F4 with 2952 neurons). We train this network with an RMSProp optimizer (Tieleman & Hinton, 2012) with a learning rate of 0.001. As illustrated in Fig. 6.3, the network takes an RGB image as input (depicting the current traffic state) and produces action probabilities as output (from which an action deciding the configuration of signal regimes is sampled). All our experiments (source and transfer) are based on this network architecture.

6.3 Transfer Learning to Build a Self-Sufficient (visually-intelligent) Autonomous Traffic Signal Control Agent: Experiments and Results

Reinforcing the consensus amongst AI research community that achieving generality with a single agent/model is the fundamental aspect of intelligence, we in this section, use our pre-trained signal control agent to optimize traffic flows in various newly-encountered (never seen before) situations. In these set of experiments, we use junction travel time as the performance evaluation metric:

Junction Travel-Time is defined as the time interval between vehicles arriving at the junction stop-line and vehicles reaching at the end of the junction. We take the moving average of 100 vehicles' junction travel-time. Lower journey travel-time indicates better signal control.

Following is the set of our transfer learning experiments:

6.3.1 Generalizability to different vehicle types/models

Here, our agent learns to prioritize the traversal of emergency vehicles (such as police cars, fire engines and ambulances) through the intersection. We conduct two experiments in this setup

(1) With transfer from the source task (signal control on a clear day, outlined in Sec. 4.3.1); in the target task experiment, we train our agent to effectively recognise and respond to the presence of emergency vehicles by reusing previously-learned knowledge from the source task. The source experiment only included the civil vehicles. (2) Without transfer; we initialize our agent with random neural network parameters to prioritize navigation of emergency vehicles. As seen in Fig. 6.4(A), the agent equipped with an overall understanding of the traffic environment (via transfer learning) quickly learns to prioritize emergency vehicles' swift movement through the intersection. In contrast, training the agent with random parameters to prioritize navigation of emergency vehicles demonstrated relatively slow learning.

6.3.2 Generalizability to a dimly-lit night

Since our signal control agent perceives its environment using vision, we believe it is essential to check our agent's agility when subjected to dim-lighting (illustrated in Fig. 3.3 (D)). Our experiments in this set-up include (1) With transfer from the source task (signal control, including emergency vehicles, outlined in Sec. 6.3.1); in the target task experiment, we reuse a previously-learned policy from the source task. (2) Without transfer; we train our agent with random neural network initializations on a dimly-lit night. As seen in Fig. 6.4 (B), the agent relying on previously-acquired skill-set learns to minimize the junction travel time for individual vehicles almost instantaneously. In contrast, the agent with the random neural network initializations learns slowly. The target experiment agent's basic understanding of the traffic scene and its ability to learn a clearly structured topology in the regular lattice of pixels from the visual input data, allows it to quickly adapt to the changing lighting conditions. Moreover, there is no visual obstruction (such as snow) in the scene which can hinder the agent's capability to reuse the previously-learned feature representation of the traffic environment.

6.3.3 Generalizability to a rainy day

Here, our agent learns to optimize traffic flows in the presence of rain (illustrated in Fig. 3.3 (E)). For these experiments, we consider rain of 10 mm/h. To sustain high image quality, we take into account the raindrops falling on the camera lens and have a mechanism in place to remove them. In this setup, we conduct two experiments (1) With transfer from the source task (signal control, including emergency vehicles and dim-lighting, outlined in Sec. 6.3.2); in the

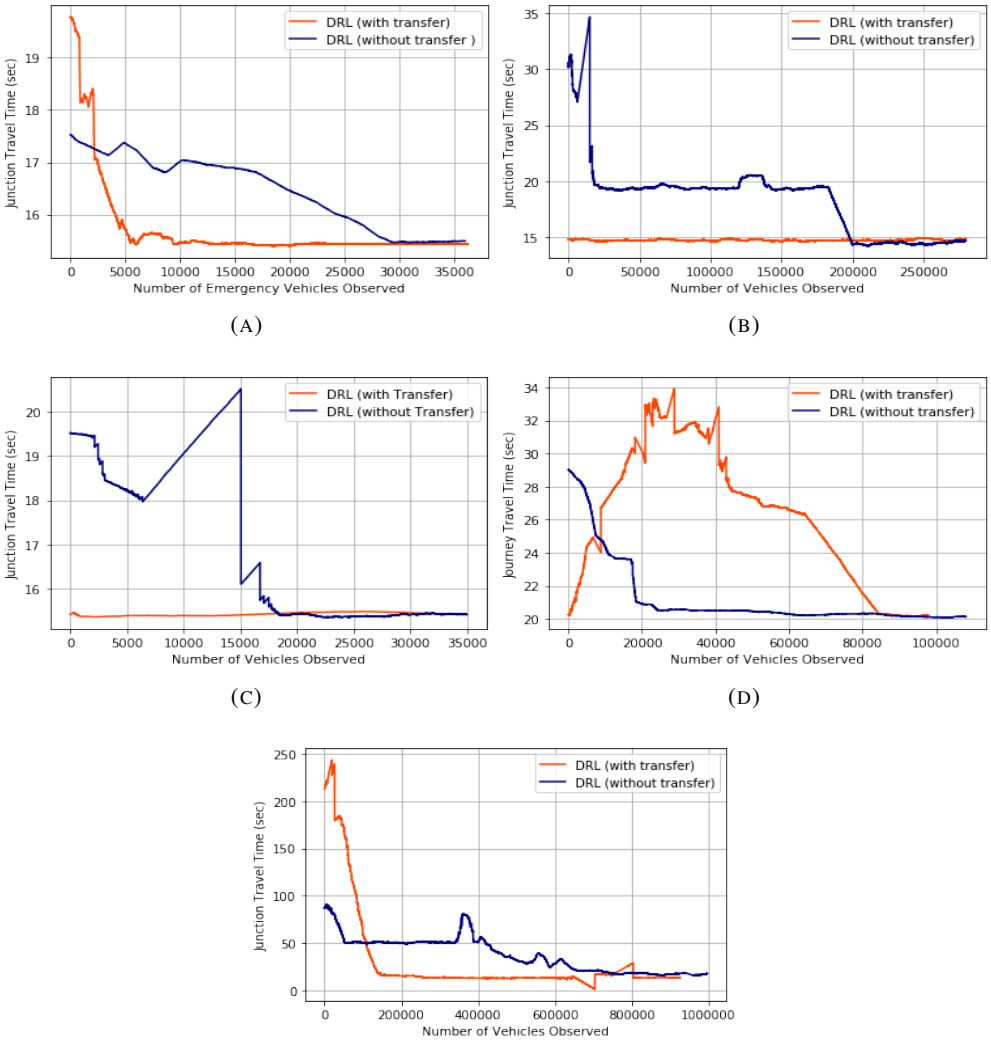


FIGURE 6.4: Graphs depicting our signal control agent's performance based on cumulative junction travel time (y-axis) over the total number of vehicles observed during the training (x-axis). The lower the junction travel time, the better. We compare DRL approach for traffic optimization; with (red line) and without (blue line) transfer learning. Our Learning curves showing vehicles' junction travel time include traffic simulation experiments; (A) In the presence of emergency vehicles. (B) On a dimly-lit night. (C) On a rainy day. (D) On a snowy day. (E) Around a different junction layout.

target task experiment, we reuse a previously-learned policy from the source task. (2) Without transfer; we initialize our agent with random neural network parameters to optimize the flow of traffic on a rainy day. As seen in the graph of Fig. 6.4 (C), the agent making use of learned policy reduces junction travel time for individual vehicles almost instantaneously. Heavy rain of 10 mm/h has little/no effect on our agent's ability to interpret the fundamental traffic scene (i.e. general junction layout and vehicle distribution). In contrast, the agent initialized with random

neural network parameters does not have any pre-existing knowledge to build on, in consequence, it learns relatively slowly.

6.3.4 Generalizability to a snowy day

Here, our agent learns to optimize traffic flows in the presence of snow (illustrated in Fig. 3.3 (F)). In this setup, we conduct two experiments (1) With transfer from the source task (signal control, including emergency vehicles, as well as dim-lighting and rain, outlined in Sec. 6.3.3); in the target task experiment, we reuse a previously-learned policy from the source task. (2) Without transfer; we initialize our agent with random neural network parameters to optimize the traffic flows on a snowy day. The results shown in Fig. 6.4 (D) indicate negative transfer. The agent learning via transfer learning performs worse than the agent using the random initializations. We attribute this performance to the fact that snow, being opaque in nature, causes visibility degradation and occlusion; significantly modifying the agent's visual input. This affects the agent's prior understanding of the traffic scene and its object localization potential; leaving no points of visual reference from formerly-possessed knowledge. In contrast, the agent with random initializations begins learning in the presence of snow and gradually learns to optimize the flow of traffic. This type of experiment informs us of the requirement to pre-train agents for snowy scenes before deployment.

6.3.5 Generalizability to a different junction layout

Here, we establish the ease of deployment of our single signal control agent to new junctions with varied topologies/structures. Our experiments in this set-up include (1) With transfer from the source task; in the target task experiment (signal control on a 4-way junction, illustrated in Fig. 3.3 (A)), the agent reuses the previously-learned policy from the source task (signal control on a 2-way junction, illustrated in Fig. 3.3 (C)). (2) Without transfer; the agent is trained with random neural network initializations on a 4-way junction. The difference between the 2-way and 4-way junctions includes an altogether different junction layout (2-way junction has two traffic lights and 4-way junction has four traffic lights). The results of these experiments are shown in Fig. 6.4 (E). Initially, the agent equipped with a learned policy starts worse than the agent with random initializations, but it learns an effective policy to optimize traffic flows much faster.

Owing to the pre-trained agent's understanding of the basic traffic entities such as vehicles, lane-markings, it adapts its behavior to the varied junction layout. In contrast, the agent using the random initializations devotes considerable time to exploring the traffic environment from the beginning, slowly learns its intrinsic feature representation, before it subsequently optimizes the traffic flows through the intersection. This is an indication that it is not only feasible but also desirable to re-use a previously trained agent on a new intersection layout. Traffic3D, our visual simulation environment, can act as a suitable ground for training such agents prior to physical deployment.

6.4 Summary

The ability to transfer knowledge between tasks in a vast environment such as traffic has the potential to scale up the domain of reinforcement learning. Combination of deep reinforcement learning and transfer learning naturally decomposes a complex sequential decision-making task into a series of relatively less complicated sub-tasks. In this chapter, we examined the benefits of reusing the previously-acquired knowledge to solve new-related tasks in high-dimensional settings. We experimentally evaluate our agent's generalisation performance and robustness to newly-encountered traffic situations. Our research findings reflect that our agent can successfully transfer the previously-acquired knowledge to effectively generalize to different traffic situations(such as different traffic densities, the presence of vehicles of different appearance, road behaviour and priorities, diverse road layouts/geometry, as well as variations in lighting and weather conditions). In our experience, the agent is unable to unlearn the previously-acquired knowledge about its environment and relearn the dynamics of a new environment if there are any obstructions or modifications in the frame of reference of environmental observations. This impedes the agent's ability to comprehend inputs between source task and target task in the same way.

Chapter 7

Interpretable Signal Control: Analysis of Deep Reinforcement Learning Agent's Performance

In this chapter, we address the *research question 5* (outlined in Sec. 1.1), using a visualization technique; Grad-CAM (Gradient Weighted Class Activation Mapping) outlined in Sec. 2.6. Over the recent years, deep neural networks (DNNs) astoundingly improved the state-of-the-art and outperformed humans in various empirical tasks, ranging from speech recognition, computer vision to training agents to autonomously play complex games (LeCun et al., 2015; Mnih et al., 2015). Despite these impressive successes, there has been a pervasive issue involving interpretability of deep neural network architectures. DNNs act as black-box models and theoretical guarantees demonstrating the viability of these models are scarce. There are open questions around inferring the decisions taken by these models, the stability of training as well as potential design methodology of stable architectures. Especially, to produce intelligent agents that can be successfully taken out of the laboratory and employed in real world, an intuitive and coherent explanation of DNNs is of great importance. Considering house price estimation system-based on attributes (such as location, road conditions and the number of bedrooms), it is desirable to have some attention drawn to the factors influencing the house price prediction. For instance, whether or not the surrounding roads increase the values of the houses.

DRL methods utilize DNNs as function approximators, which are known to generalize well to high-dimensional input data (such as visual traffic data) but, at the cost of turning into black boxes. In this chapter, we take a step towards explainable *Artificial Intelligence* and provide an

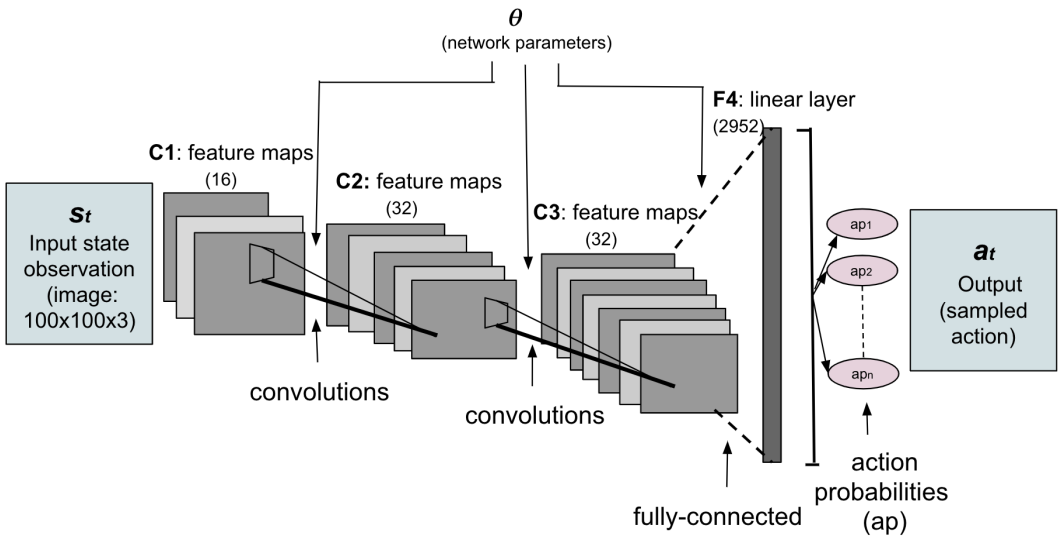


FIGURE 7.1: Our DRL-based (single-intersection) Signal Control Agent's Network Architecture.

inference of our DNN model used for signal control. Our goal is to identify the parts/regions of the input that are more significant to the corresponding output. Our visualization results demonstrate that our signal control agent is able to achieve traffic optimization based on features that are rarely exploited by conventional traffic data collection methods (such as induction loops (Koonce & Rodegerdts, 2008)).

7.1 Related Work

Interpretability of deep models is an extensively recognised, but not yet solved problem. In recent years, many methods have been proposed to interpret deep learning architectures (DNNs) (Koh & Liang, 2017; Lundberg & Lee, 2017; Zhang, Nian Wu, & Zhu, 2018; Chen, Chen, Ren, Huang, & Zhang, 2019). However, most of these methods are developed for classification tasks (Liu, Xuan, Zhang, Stylianou, & Pless, 2019). In contrast, in this thesis, we interpret the signal regime decisions taken by DRL agents to accomplish effective signal control.

Some of the previous research works have explored DRL agents' visualization in game-based benchmarks. Wang et al. (J. Wang, Gou, Shen, & Yang, 2018) created a system; DONViz to provide interpretation information about DQN models in the form of bar, line and pie charts. Greydanus et al. (Greydanus, Koul, Dodge, & Fern, 2017) explored the utility of visual stimuli in making decisions in the Atari domain (Bellemare et al., 2013) using saliency maps. Douglas et al. (Douglas et al., 2019) attempted in enhancing the understandability of their DRL model to identify

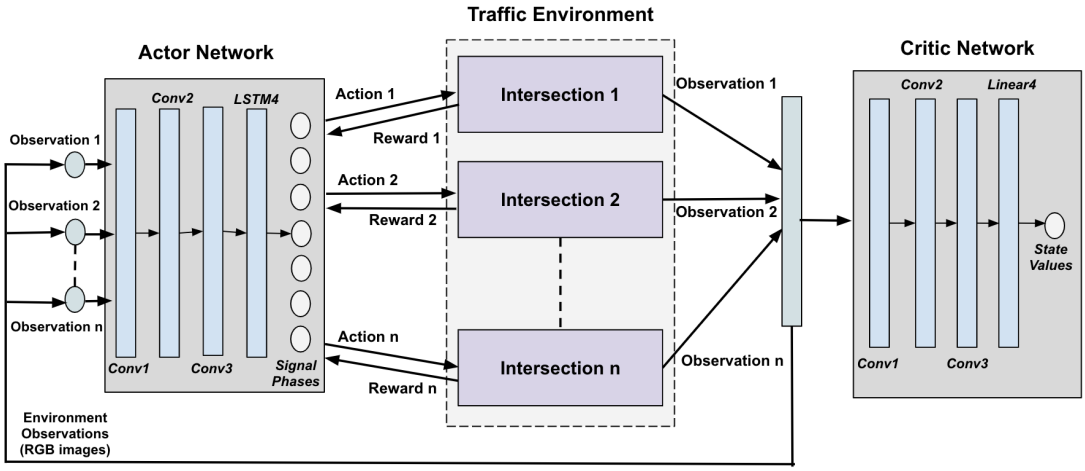


FIGURE 7.2: Our DRL-based (multi-intersection) Signal Control Agent's Network Architecture.

the input regions dominating the computation of the corresponding output using a perturbation-based saliency approach in Pommerman benchmark (Resnick et al., 2018). To our knowledge, no other research work exists that visualizes DRL agents used for autonomous signal control. Our visualization results reflect that DRL applied to visual traffic data from road intersections eliminates the need to have pre-determined hand-engineered features describing the traffic environment.

7.2 Our Visualization Methodology

Our visualization methodology is based on Grad-CAM (Gradient-weighted Class Activation Mapping) (Selvaraju et al., 2017). Our method takes as inputs - a pre-trained network (i.e. pre-trained signal control agent) and an image (depicting the traffic environment). The output is produced in the form of an attention map (i.e. a heatmap). Our Grad-CAM based visualization method makes use of the gradient information flowing into the last convolutional layer of the pre-trained CNN to determine the importance of each neuron for making a certain signal control decision. To obtain localization map for a particular signal control phase regime decision p, Grad-CAM method first computes the gradient of the score y^p (before softmax) with respect to the feature maps A^k;

$$g_p(A^k) = \frac{\partial y^p}{\partial A^k} \tag{7.1}$$

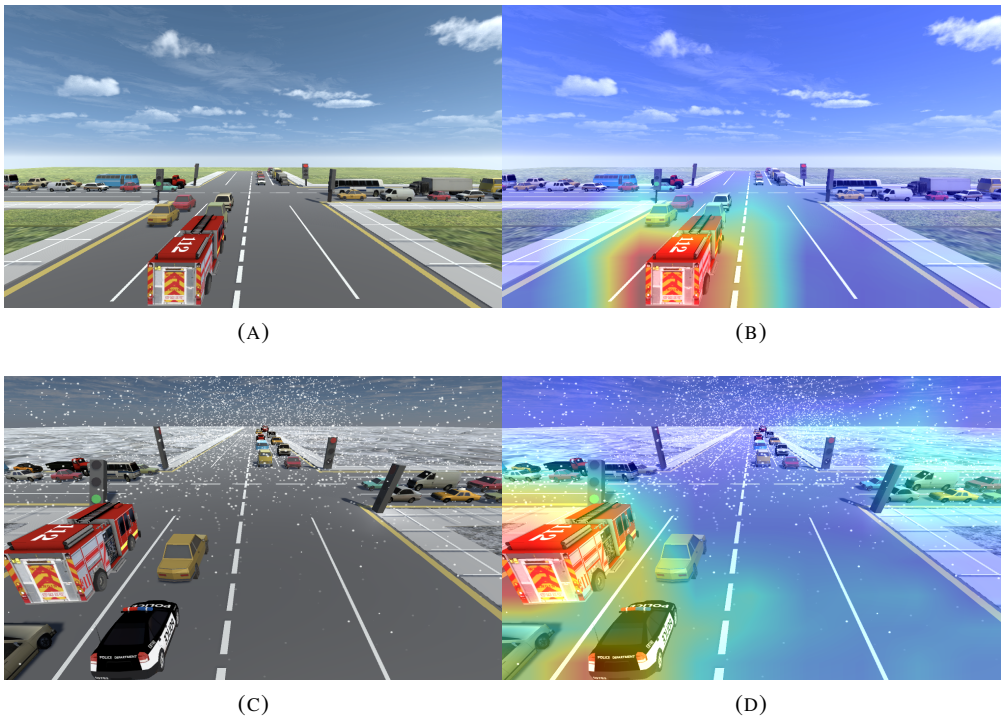
where k is the channel index. Then, the gradients are averaged as the neural importance weight α_k^p in each channel;

$$\alpha_k^p = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^p}{\partial A_{i,j}^k} \tag{7.2}$$

where (i, j) and Z are the spatial index and spatial resolution of the feature map respectively. Finally Grad-CAM is a weighted sum of feature maps (followed by a ReLU operator);

$$H_{Grad-CAM}^p = ReLU(\sum_k \alpha_k^p A^k) \tag{7.3}$$

This gives Grad-CAM implementation, in which the heatmap produced is of the same size as feature maps.



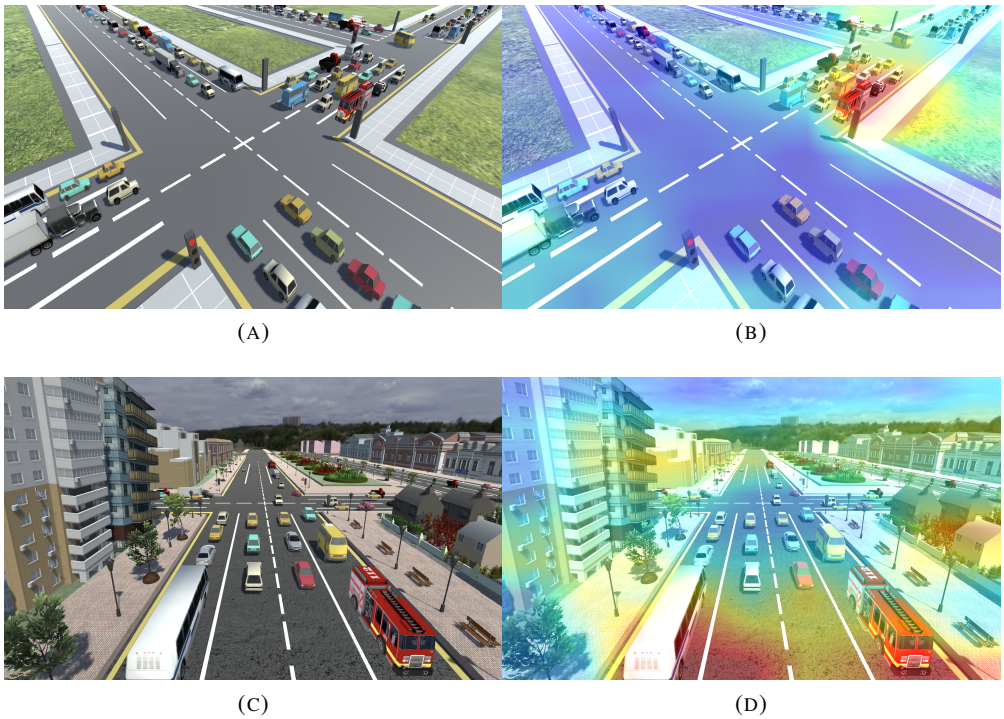
(A) (B)

(C) (D)

FIGURE 7.3: Images depicting attention-visualization in the presence of emergency vehicles (A) Original image on a clear day. (B) Grad-CAM activation-firetruck. (C) Original image on a snowy day. (D) Grad-CAM activation-police car and firetruck.

7.3 Experiments and Results

In this section, we conduct experiments to demonstrate which parts of an image (depicting the prevailing state of the traffic environment) influence a certain signal regime decision.



(A) (B)

(C) (D)

FIGURE 7.4: Images depicting attention-visualization on a multi-junction scenario (A) Original image on a multi-intersection setting. (B) Grad-CAM activation-firetruck and ambulance. (C) Original image on an enhanced multi-intersection setting. (D) Grad-CAM activation-firetruck and public bus.

7.3.1 Attention-Visualization (single-intersection) on a Clear Day

In this experiment, we visualize the last CNN layer of our pre-trained signal control agent's neural network (illustrated in Fig. 7.1). This agent is trained to optimize traffic flows through a single intersection on a clear day (illustrated in Fig. 3.3 (A)). As seen in Fig. 7.3 (B), our DRL-based signal control agent prioritizes the movement of emergency vehicles on a clear sky day through the intersection by configuring the signal regimes accordingly.

7.3.2 Attention-Visualization (single-intersection) on a Snowy Day

Even the smallest of perturbations in the visual input is known to significantly distort the feature embeddings and consequently affect the output of a neural network (Zheng, Song, Leung, & Goodfellow, 2016). To investigate our signal control agent's stability against natural distortions in the visual input, we train our DRL agent to optimize the flow of traffic in the presence of snow. In this experiment, we again visualize the last CNN layer of our pre-trained signal control agent's neural network (illustrated in Fig. 7.1). This agent is trained to optimize traffic flows through a

single intersection on a snowy day (illustrated in Fig. 3.3 (F)). Our results indicate that our agent is able to maintain its good performance on a snowy day (shown in Fig. 7.3 (D)). The performance of our agent on a clear day and a snowy day are comparable, as the agent operating on a snowy day begins its learning in the presence of snow and gradually, through its repertoire of experience, learns to counteract the effect of snow and optimize the flow of traffic in the presence of snow.

7.3.3 Attention-Visualization (multi-intersection) on a Clear Sky Day

Further to ensure our signal control agent's scalability, we evaluate our agent's performance on a multi-intersection setting (shown in Fig. 7.4 (A)). In this experiment, we visualize the last CNN layer of our pre-trained multi-intersection signal control agent's actor-network (illustrated in Fig. 7.2). Owing to our agent's continuous interaction with the traffic environment, our agent is able to sustain its good performance on a more complex intersection layout and process the traffic environment appropriately (shown in Fig. 7.4 (B)).

In another experiment, we use a more enhanced multi-intersection setting (illustrated in Fig. 3.1) to draw inference on our agent's signal control decisions. As seen in Fig. 7.4 (D), our DRL-based signal control method shows activation around emergency and public vehicles to prioritizes their movement through the intersections.

Our visualization research findings reflect that DRL applied to visual traffic data enables signal control-based on key traffic features (such as vehicle type and their relevance). It also validates the benefit of using visual traffic data in providing more flexibility (e.g. larger detection areas) than typically used induction loops. Furthermore, visualizing the output of DNN paradigms has the potential of breaking barriers in machine learning research. We expect that our DNN visualization will help transportation engineers gain further trust in applying deep learning paradigms to autonomous transportation.

7.4 Summary

To have them integrated into the real-world settings, it is paramount to build trustworthy autonomous agents with a high degree of transparency and explainability capabilities such that these agents have the ability to reflect why they predict what they predict. Also, reliable interpretability

of these agents is expected to increase human acceptance of the existing neural-nets-based black-box approaches. In this chapter, we advanced towards explainable *Artificial Intelligence* to address the increasing need for producing human interpretable explanations of autonomous agents' operation. We translate our DRL agent's signal control decisions in a human-understandable and human-verifiable form. We implement a localization mechanism; Grad-CAM to produce visual explanations of our signal control agents' decisions. Grad-CAM has been previously used to visually interpret decisions made by deep models in the domains, including image classification, image captioning and visual question answering. To our knowledge, this work is the first to interpret DRL-based signal control agents' decisions. Our visualization results demonstrate the faithfulness of our signal control methodology and can help non-machine learning experts to understand what our signal control agents' beliefs are while making signal regime decisions. We conclude this chapter on the note that a truly autonomous agent should not just be intelligent, but it should also be able to reason about its beliefs and decisions so that it can be trusted and subsequently, deployed in real-world settings.

Chapter 8

Conclusion and Future Work

Teaching an agent to operate autonomously, effectively in a complex, unfamiliar environment to accomplish a certain goal has been the *Artificial Intelligence* community's longstanding goal. In this thesis, we develop a deep reinforcement learning agent (DRL) to enhance the performance of already existing traffic signal control infrastructure. Our DRL agent autonomously configures traffic signal regimes in real time based on the actual prevailing traffic situation. To mimic human-like learning, our signal control agent operates *solely* on *live* camera feed. We tested our research approach in a variety of traffic scenarios. To realistically frame our research problem and demonstrate the applicability of our DRL-based signal control agent to dynamically varying diverse traffic conditions, we created a novel traffic simulation environment; Traffic3D. Our simulation environment is significantly richer in terms of both physical and visual properties and adequately captures the characteristics of real-world traffic scenarios. Ability to train signal control agents in a realistic environment is critical in making it possible to deploy them in real-world traffic settings. In conclusion, in this thesis, we demonstrate DRL is promising in effectively achieving autonomous signal control. In this chapter, we conclude this thesis by outlining our main contributions and avenues for future research.

8.1 Contributions

We summarize the main contributions of this thesis as follows;

8.1.1 A physically and visually intelligent traffic simulation environment

In Chapter 3, we presented our new traffic simulation environment; Traffic3D. The motivation to create this simulation tool is to fairly validate our vision-based signal control research approach.

The goal of Traffic3D is to create traffic simulations with a high degree of realism. Our novel traffic micro-simulation platform supports unique simulation features such as near-photorealism, complex physical phenomenon, inexpensive collection of diverse traffic data, real-world partial observability challenges and python support for deep learning applications.

8.1.2 An adaptive signal control agent to optimize traffic flows through single intersections

In Chapter 4, we introduced an end-to-end trainable DRL-based signal control agent that continuously modifies the traffic signal regimes online, as per the changing traffic observations. Equipped with the ability to perceive the prevailing traffic conditions extensively using high-dimensional visual inputs, our signal control agent is able to sustain good performance in different traffic situations, including varying intersection topologies, traffic densities/distribution, vehicle types, weather and lighting conditions.

8.1.3 Traffic optimization through multiple intersections

In Chapter 5, we presented the first application extending DRL methods to optimize traffic through multiple intersections-based *solely* on visual traffic data, without hand-crafted traffic state features. Our multi-intersection signal control method led to positive emergence of cooperative behavior among individual signal control agents.

8.1.4 A generalizable and transferable signal control agent

In Chapter 6, we implemented transfer learning to ensure our signal control agent's generalizability and accelerated-learning around new traffic situations. Our agent is able to interpret and extract salient features from complex high-dimensional traffic environment, to optimize the movement of vehicles in newly-encountered traffic situations.

8.1.5 Interpretation of signal control decisions (analysis of DRL performance)

In Chapter 7, we visualized the last convolutional layer of the trained DNN to demonstrate which parts of the visual input influence a certain signal regime decision. Our visualization research findings reflect that our signal control agent, apart from prioritizing the swift movement of traffic-based on the prevailing traffic demand captured by wide-range cameras, it also attends to different

types of vehicles (prioritizes the traversal of emergency vehicles). DRL applied to visual traffic data enables our agent to configure signal regimes based on key traffic features (such as vehicle type and their relevance) that would otherwise be impossible or impractical using conventionally-used traffic data collection methods (such as induction loops, which counts the number of vehicles on every lane).

8.2 Future Work

DRL has proven to be a powerful machine learning paradigm that has achieved significant success across a variety of control problems. This thesis covered various research activities, including creating a 3D-traffic simulation platform to devise intelligent agents, DRL-based autonomous single-intersection and multi-intersection signal control, transfer learning in signal control and analysis of DRL performance. There are different open venues for further research; some related to autonomous signal control and others useful in enhancing the performance of DRL agents in general. The research activities undertaken in this thesis can be further explored in the following ways;

8.2.1 Our traffic simulation environment

We created a new 3D-traffic simulation tool; Traffic3D to validate our research approach. While most of the presently-used traffic simulation environments are pertinent to transportation-specific research, Traffic3D's capabilities transcend beyond the realm of transportation. The research community can use Traffic3D as a learning environment to conduct research across multiple directions, including, but not limited to semantic segmentation, 3D-navigation, visual question answering and neural attention mechanisms. As per the application under consideration, the required level of complexity can be conveniently simulated. In addition to Traffic3D's current level of functionality (discussed in Chapter 3), Unity game engine's set of properties can be used to develop more realistic and increasingly complex traffic scenarios in the future. For instance, traffic lights can be configured to allow pedestrians' safe passage through intersections and parallel simulation of multiple copies of the traffic environment can be executed to improve sample efficiency.

8.2.2 Self-sufficient signal control agent

Our DRL-based signal control agent trained using a diverse set of training data which we generated using our simulation tool; Traffic3D gains a breadth of experience and effectively optimizes the movement of vehicles through intersections. However, generalizability is a prominent issue in training these agents to become self-sufficient to the vast environment they are exposed to. As future work, potential efforts can be made to explore more effective transfer learning techniques to develop truly self-sufficient signal control agents that can effectively operate in the vast environment, including the snowy scene. As shown in Chapter 6, our signal control agent demonstrated negative transfer on a snowy scene.

8.2.3 Enhanced multi-intersection signal control

Although our centralised multi-intersection signal control methodology worked well (shown in Chapter 5), but as the number of agents/intersections increases, the centralised critic's state input dimensionality increases exponentially and is susceptible to single-point-of-failure. As future work, more effective algorithms can be explored to address the limitations arising from centralisation.

8.2.4 Transfer learning in multi-intersection scenarios

Future research can be pursued to implement transfer learning methods in multi-intersection traffic scenarios, such that a single signal control agent-based on the proposed signal control strategy (outlined in Chapter 5) can effectively optimize traffic flows around different network topologies under varying ambient (weather and lighting) conditions.

8.2.5 Network-to-network knowledge transfer

In Chapter 6, we demonstrated the use the knowledge transfer to achieve signal control in newly-encountered traffic situations around single intersections. However, transfer of previously-learned knowledge can also be carried out between a network of intersections. As future work, we outline another direction of implementing transfer learning in signal control, in which knowledge from one network of intersections is transferred to another with varied topology.

8.2.6 Embedding structures to accelerate signal control agents' training

After interpreting DRL agents' signal control decisions using attention-visualization (presented in Chapter 7), as future work, high-dimensional traffic state representation (raw pixels) can be mapped into a low-dimensional vector space. By doing this, instead of processing high-dimensional visual inputs, the neural network will specifically pay attention to the salient environmental features that influence the signal control decisions. We expect this to optimize the signal control agents' training time.

References

Arel, I., Liu, C., Urbanik, T., & Kohls, A. G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, *4*(2), 128–135.

Au, T.-C., Quinlan, M., Stiurca, N., Zhu, J., & Stone, P. (2010). Planning for improving throughput in autonomous intersection management. In *Icaps.*

Aziz, H. A., Zhu, F., & Ukkusuri, S. V. (2018). Learning-based traffic signal control algorithms with neighborhood information sharing: An application for sustainable mobility. *Journal of Intelligent Transportation Systems*, *22*(1), 40–52.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., ... others (2016). Deepmind lab. *arXiv preprint arXiv:1612.03801*.

Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D. (2011). Sumo–simulation of urban mobility. In *The third international conference on advances in system simulation (simul 2011), barcelona, spain* (Vol. 42).

Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, *47*, 253–279.

Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, *27*(4), 819–840.

Chakroborty, P., & Das, A. (2017). *Principles of transportation engineering*. PHI Learning Pvt. Ltd.

Chen, R., Chen, H., Ren, J., Huang, G., & Zhang, Q. (2019). Explaining neural networks semantically and quantitatively. In *Proceedings of the ieee international conference on computer vision* (pp. 9187–9196).

Chin, S.-M., Franzese, O., Greene, D. L., Hwang, H.-L., Gibson, R., et al. (2004). *Temporary losses of highway capacity and impacts on performance: Phase 2* (Tech. Rep.). United States. Dept. of Energy. Office of Scientific and Technical Information.

Chu, T., Qu, S., & Wang, J. (2016). Large-scale traffic grid signal control with regional reinforcement learning. In *2016 american control conference (acc)* (pp. 815–820).

Chu, T., & Wang, J. (2017). Traffic signal control by distributed reinforcement learning with

min-sum communication. In *2017 american control conference (acc)* (pp. 5095–5100).

Chu, T., Wang, J., Codecà, L., & Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*.

Coifman, B. (2006). Vehicle level evaluation of loop detectors and the remote traffic microwave sensor. *Journal of transportation engineering*, *132*(3), 213–226.

Dewey, D. (2014). Reinforcement learning and the reward engineering principle. In *2014 aaai spring symposium series.*

Douglas, N., Yim, D., Kartal, B., Hernandez-Leal, P., Maurer, F., & Taylor, M. E. (2019). Towers of saliency: A reinforcement learning visualization using immersive environments. In *Proceedings of the 2019 acm international conference on interactive surfaces and spaces* (pp. 339–342).

El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2013). Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1140–1150.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). Counterfactual multi-agent policy gradients. In *Thirty-second aaai conference on artificial intelligence.*

Gaidon, A., Wang, Q., Cabon, Y., & Vig, E. (2016). Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4340–4349).

Gao, J., Shen, Y., Liu, J., Ito, M., & Shiratori, N. (2017). Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. *arXiv preprint arXiv:1705.02755*.

Garg, D., Chli, M., & Vogiatzis, G. (2018). Deep reinforcement learning for autonomous traffic light control. In *2018 3rd ieee international conference on intelligent transportation engineering (icite)* (pp. 214–218).

Garg, D., Chli, M., & Vogiatzis, G. (2019a). A deep reinforcement learning agent for traffic intersection control optimization. In *2019 ieee intelligent transportation systems conference (itsc)* (pp. 4222–4229).

Garg, D., Chli, M., & Vogiatzis, G. (2019b). Traffic3d: A new traffic simulation paradigm. In *Proceedings of the 18th international conference on autonomous agents and multiagent*

systems (pp. 2354–2356).

Garg, D., Chli, M., & Vogiatzis, G. (2019c). Traffic3d: A rich 3d-traffic environment to train intelligent agents. In *International conference on computational science* (pp. 749–755).

Gartner, N. H., Pooran, F. J., & Andrews, C. M. (2001). Implementation of the opac adaptive control strategy in a traffic signal network. In *Itsc 2001. 2001 ieee intelligent transportation systems. proceedings (cat. no. 01th8585)* (pp. 195–200).

Genders, W., & Razavi, S. (2016). Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint arXiv:1611.01142*.

Geroliminis, N., & Daganzo, C. F. (2008). Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, *42*(9), 759–770.

Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., & Farhadi, A. (2018). Iqa: Visual question answering in interactive environments. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4089–4098).

Greydanus, S., Koul, A., Dodge, J., & Fern, A. (2017). Visualizing and understanding atari agents. *arXiv preprint arXiv:1711.00138*.

Grondman, I., Busoniu, L., Lopes, G. A., & Babuska, R. (2012). A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1291–1307.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Henry, J.-J., Farges, J. L., & Tuffal, J. (1984). The prodyn real time traffic algorithm. In *Control in transportation systems* (pp. 305–310). Elsevier.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Hunt, P., Robertson, D., Bretherton, R., & Royle, M. C. (1982). The scoot on-line traffic signal optimisation technique. *Traffic Engineering & Control*, *23*(4).

Jeon, H., Lee, J., & Sohn, K. (2018). Artificial intelligence for traffic signal control based solely on video images. *Journal of Intelligent Transportation Systems*, *22*(5), 433–445.

Kempka, M., Wydmuch, M., Runc, G., Toczek, J., & Jaśkowski, W. (2016). Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *Computational intelligence and games (cig), 2016 ieee conference on* (pp. 1–8).

Koh, P. W., & Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1885–1894).

Kolve, E., Mottaghi, R., Gordon, D., Zhu, Y., Gupta, A., & Farhadi, A. (2017). Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*.

Konda, V. R., & Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in neural information processing systems* (pp. 1008–1014).

Koonce, P., & Rodegerdts, L. (2008). *Traffic signal timing manual.* (Tech. Rep.). United States. Federal Highway Administration.

Kraemer, L., & Banerjee, B. (2016). Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, *190*, 82–94.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kuyer, L., Whiteson, S., Bakker, B., & Vlassis, N. (2008). Multiagent reinforcement learning for urban traffic control using coordination graphs. In *Joint european conference on machine learning and knowledge discovery in databases* (pp. 656–671).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Liang, X., Du, X., Wang, G., & Han, Z. (2018). Deep reinforcement learning for traffic light control in vehicular networks. *arXiv preprint arXiv:1803.11115*.

Liu, X., Xuan, H., Zhang, Z., Stylianou, A., & Pless, R. (2019). Visualizing how embeddings generalize. *arXiv preprint arXiv:1909.07464*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Manar, A., & Baass, K. G. (1996). Traffic platoon dispersion modeling on arterial streets. *Transportation Research Record*, *1566*(1), 49–53.

Mauro, V., & Di Taranto, C. (1990). Utopia. In *Control, computers, communications in transportation* (pp. 245–252). Elsevier.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.

Mousavi, S. S., Schukat, M., & Howley, E. (2017). Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, *11*(7), 417–423.

Peirce, J., & Webb, P. (1994). Mova control of isolated traffic signals-recent experience. In *Third international conference on road traffic control, 1990.* (pp. 110–113).

Pell, A., Meingast, A., & Schauer, O. (2017). Trends in real-time traffic simulation. *Transportation research procedia*, *25*, 1477–1484.

Priemer, C., & Friedrich, B. (2009). A decentralized adaptive traffic signal control using v2i communication data. In *2009 12th international ieee conference on intelligent transportation systems* (pp. 1–6).

Resnick, C., Eldridge, W., Ha, D., Britz, D., Foerster, J., Togelius, J., ... Bruna, J. (2018). Pommerman: A multi-agent playground. *arXiv preprint arXiv:1809.07124*.

Rhodes, A., Bullock, D. M., Sturdevant, J., Clark, Z., & Candey Jr, D. G. (2005). Evaluation of the accuracy of stop bar video vehicle detection at signalized intersections. *Transportation research record*, *1925*(1), 134–145.

Richter, S. R., Hayder, Z., & Koltun, V. (2017). Playing for benchmarks. In *Proceedings of the ieee international conference on computer vision* (pp. 2213–2222).

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3234–3243).

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sadeghi, F., & Levine, S. (2016). Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of*

the ieee international conference on computer vision (pp. 618–626).

Sen, S., Sekaran, M., Hale, J., et al. (1994). Learning to coordinate without sharing information. In *Aaai* (Vol. 94, pp. 426–431).

Silver, D. L., Yang, Q., & Li, L. (2013). Lifelong machine learning systems: Beyond learning algorithms. In *Aaai spring symposium: Lifelong machine learning* (Vol. 13, p. 05).

Sims, A. G., & Dobinson, K. W. (1980). The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on vehicular technology*, *29*(2), 130–137.

Sutton, R. S., & Barto, A. G. (2011). Reinforcement learning: An introduction.

Sutton, R. S., Barto, A. G., & Williams, R. J. (1992). Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, *12*(2), 19–22.

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (pp. 1057–1063).

Synnaeve, G., Nardelli, N., Auvolat, A., Chintala, S., Lacroix, T., Lin, Z., ... Usunier, N. (2016). Torchcraft: a library for machine learning research on real-time strategy games. *arXiv preprint arXiv:1611.00625*.

Taylor, M. E., & Stone, P. (2011). An introduction to intertask transfer for reinforcement learning. *Ai Magazine*, *32*(1), 15.

Thorpe, T. L., & Anderson, C. W. (1996). *Traffic light control using sarsa with three state representations* (Tech. Rep.). Citeseer.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, *4*(2), 26–31.

Van der Pol, E., & Oliehoek, F. A. (2016). Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*.

Wang, J., Gou, L., Shen, H.-W., & Yang, H. (2018). Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE transactions on visualization and computer graphics*, *25*(1), 288–298.

Wang, P. (2016). *Learning and exploiting camera geometry for computer vision* (Unpublished doctoral dissertation). Duke University.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292.

Wiering, M. (2000). Multi-agent reinforcement learning for traffic light control. In *Machine learning: Proceedings of the seventeenth international conference (icml'2000)* (pp. 1151–1158).

Wu, Y., Wu, Y., Gkioxari, G., & Tian, Y. (2018). Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*.

Wymann, B., Espié, E., Guionneau, C., Dimitrakakis, C., Coulom, R., & Sumner, A. (2000). Torcs, the open racing car simulator. *Software available at http://torcs. sourceforge. net*, *4*, 6.

Yan, C., Misra, D., Bennnett, A., Walsman, A., Bisk, Y., & Artzi, Y. (2018). Chalet: Cornell house agent learning environment. *arXiv preprint arXiv:1801.07357*.

Zhang, Q., Nian Wu, Y., & Zhu, S.-C. (2018). Interpretable convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 8827–8836).

Zheng, S., Song, Y., Leung, T., & Goodfellow, I. (2016). Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4480–4488).

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., & Farhadi, A. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 ieee international conference on robotics and automation (icra)* (pp. 3357–3364).