# Fully-Autonomous, Vision-based Traffic Signal Control: from Simulation to Reality

Deepeka Garg
Aston University
Birmingham, United Kingdom
gargd@aston.ac.uk

Maria Chli
Aston University
Birmingham, United Kingdom
m.chli@aston.ac.uk

George Vogiatzis
Aston University
Birmingham, United Kingdom
g.vogiatzis@aston.ac.uk

## ABSTRACT

Ineffective traffic signal control is one of the major causes of congestion in urban road networks. Dynamically changing traffic conditions and live traffic state estimation are fundamental challenges that limit the ability of the existing signal infrastructure in rendering individualized signal control in real-time. We use deep reinforcement learning (DRL) to address these challenges. Due to economic and safety constraints associated with training such agents in the real world, a practical approach is to do so in simulation before deployment. Domain randomisation is an effective technique for bridging the *reality gap* and ensuring effective transfer of simulation-trained agents to the real world. In this paper, we develop a fully-autonomous, vision-based DRL agent that achieves adaptive signal control in the face of complex, imprecise, and dynamic traffic environments. Our agent uses live visual data (i.e. a stream of real-time RGB footage) from an intersection to extensively perceive and subsequently act upon the traffic environment. Employing domain randomisation, we examine our agent's generalisation capabilities under varying traffic conditions in both the simulation and the real-world environments. In a diverse validation set independent of training data, our traffic control agent reliably adapted to novel traffic situations and demonstrated a positive transfer to previously unseen real intersections despite being trained entirely in simulation.

## KEYWORDS

Intelligent Transportation Systems; Autonomous Signal Control; Deep Reinforcement Learning.

**ACM Reference Format:**
Deepeka Garg, Maria Chli, and George Vogiatzis. 2022. Fully-Autonomous, Vision-based Traffic Signal Control: from Simulation to Reality. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), online, May 9–13, 2022*, IFAAMAS, 9 pages.

## 1 INTRODUCTION

Road traffic congestion remains a major problem around the world, resulting in significant economic and environmental repercussions. One of the most effective ways to mitigate traffic congestion is by intelligently managing the signal infrastructure. Current signal control systems operate either on fixed time frames (Webster method [1]) or use in-road sensors (inductive loops [2]) to extend or shorten green signals when needed. Widely-used adaptive signal control methods (such as SCOOT [3] and SCAT [4]) largely rely on manually-designed signal phase plans. These plans are designed to be dynamically selected according to the volume of the traffic detected by inductive loops. The loop sensors are commonly placed close to the intersection and are not activated until vehicles pass through them, providing only partial information on the traffic conditions. Consequently, the signals are unable to perceive and react to changing traffic patterns in real time. Transportation operators often have to manually override signal phase decisions to keep up with the evolving traffic conditions. Currently, no tool exists that achieves autonomous signal control optimised for a junction's specific geometrical layout and dynamically changing traffic distribution.

Deep Reinforcement Learning (DRL) is one of the most prominent subfields in AI, holding the promise of enabling agents to learn sophisticated behaviors automatically while making decisions in real time. One of the most enticing possibilities that DRL (a mechanism combining reinforcement and deep learning) presents is the ability to train the agents to perform tasks solely from raw sensory inputs, while the traditional RL methods relied on predetermined environment features for decision making. DRL has enabled agents to learn sensory perception and control in an 'end-to-end' fashion (i.e. directly mapping from sensory inputs to action outputs) eliminating the need for hand engineering of task-specific features by domain experts [5]. DRL involving learning visual features and a control policy jointly (end-to-end) has been successfully applied to several domains ranging from sophisticated video games [6, 7] to robotics [8, 9] and transportation infrastructure optimization [10–12]. While DRL agents can learn complex control policies from raw sensory data, they suffer from poor generalizability. Devising agents that can generalize well to a wide range of environmental variations and bridge the gap between simulated and real-world environments, is a significant challenge.

In this paper, we develop end-to-end trainable signal control agents that respond to the actual traffic conditions in real time. In essence, our signal control agents generate and execute signal phases based on the prevailing traffic state. They learn to adjust their signal control strategy based on the feedback they get from the traffic environment. *Domain randomization* [13] is employed to enhance the generalization capabilities and subsequently achieve a robust distributional shift of the signal control agents we create. The idea is to expose the agent to as many possible variations of a traffic setting in order to make it invariant to factors such as junction layout, traffic distribution, background, illumination and camera viewpoint. We show that this *domain randomization* approach leads to a significant performance boost as the agent's vision-based perception becomes invariant not only to particular conditions but also to the training

*Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, online.*

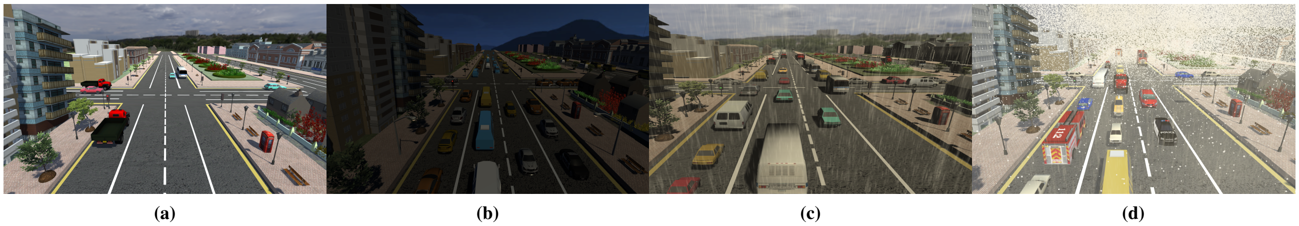


**Figure 1: Examples of visual domain randomisation in our signal control experiments. (a) A clear sky scene. (b) A night scene. (c) A rainy scene. (d) A snowy scene, all generated in Traffic3D (www.traffic3d.org), our simulator.**

domain. Our results (Fig. 6) demonstrate that signal control policies learned entirely in simulation, transfer effectively to previously unseen real-world intersections. This marks the significance of realistic simulation environments in training real-world-deployable DRL agents, alleviating the requirement for tedious data collection in the physical world, as well as of risky and costly on-site training.

## 2 RELATED WORK

Over the years, traffic signals have evolved from being pre-timed (fixed time given to all green phases based on historical traffic demand without considering potential fluctuations in traffic flow patterns) [14, 15] to adaptive - using loop sensors (real-time traffic demand is used to configure green phase duration) [3, 16]. Inductive loops detect the presence of passing vehicles, prompting the signals to allow the queuing vehicles to pass through. Adopting adaptive signal control has helped in reducing bottlenecks around intersections during peak times. However, this method heavily relies on hand-crafted rules which fail to address the dynamic traffic flows effectively enough. Conventionally-used inductive loops have a narrow operational range as they only gather traffic data (vehicle density) in their immediate area. Alternatively, we use cameras to have a wider coverage of traffic and enhance the quality of traffic detection. Roadside surveillance cameras have already emerged as powerful tools in effectively enforcing speed limits and reducing road fatalities. Furthermore, unlike inductive loops which can only detect traffic volume, cameras can effectively detect a large array of different objects (including cars, motorcycles, buses, vans and pedestrians) to extensively perceive the traffic environment.

Real-world traffic phenomena are characterised by highly-stochastic dynamics. To increase signal efficiency, signals must be constantly monitored and frequently adjusted to regulate the dynamic traffic flows. RL enables greater real-time responsiveness and constant optimization of actual traffic flows. RL agents are inherently adaptive and are capable of responding to changes in the environment. The majority of research on RL-based adaptive signal control ([17–20]) 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). However, real-world traffic evolution is influenced by many factors (such as different road users - pedestrians and cyclists, accidents, weather and road conditions). These features, while being crucial, are not considered in state-of-the-art signal control. In contrast, our signal control methodology is 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 vision-based signal control, [21, 22] used simple 2D-visual representations of the traffic environment, ignoring the visual complexities of urban traffic and did not show the effectiveness of their technique on real data. Our signal control agent is exposed to a rich traffic simulation environment [23, 24] (illustrated in Fig. 1) and achieves remarkable performance on previously unseen real-life images (see Fig. 6).

## 3 METHOD AND NOTATION

### 3.1 Reinforcement Learning

In a basic RL setting [25], an agent *learns* to achieve a goal by dynamically interacting with an uncertain environment. A standard RL framework is mathematically modelled as a Markov Decision Process (MDP), which is defined as 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 $\tau$ is determined by $R_t = \sum_{t=0}^{H} \gamma^t r_t$. Given the agent's policy $\pi$, the expected episodic return is defined by $E_\pi[R_\tau]$. The expected episodic return is maximized by optimal policy $\pi^*$

$$\pi^* = \arg\max_{\pi} E_{\tau\sim\mathcal{M},\pi}[R_\tau]. \tag{1}$$

A deep neural network $(\pi_\theta)$ with parameters $\theta$ in high-dimensional RL settings represents policy $\pi^*$. The agent aims to learn $\theta^*$ that achieves highest expected episodic return,

$$\theta^* = \arg\max_{\theta} E_{\tau\sim\mathcal{M},\pi}[R_\tau]. \tag{2}$$

### 3.2 Policy-based Reinforcement Learning

Neural Network-based function approximation [26], for mapping input traffic state to a traffic signal control action, is essential for RL to be effective in high-dimensional large state spaces. Instead of implementing a dominant value function-based off-policy RL

(Q-learning [27]), we explore an alternative on-policy RL (Policy Gradient) [28] for our signal control task.

The value function-based methods approximate the state-value function or state-action value function (i.e. how rewarding each state is or state-action pair is) and the policy is implicitly derived from the learned value function [29]. In contrast, policy-based methods directly update the policy parameters (i.e. a vector of probabilities to conduct actions under a specific state) along the direction to maximize a predefined objection (for e.g. average expected reward) [28]. One of the main advantages of policy-based methods over value-based methods is that they can learn stochastic policies (i.e. they keep exploring potentially more rewarding actions), while value-based algorithms are inclined towards learning deterministic policies. In real-world environments such as traffic settings that are characterised by uncertainty, an effective policy must be stochastic [28]. Prior work on autonomous signal control demonstrated policy-based RL's superior performance over value-based RL [30].

In this work, we directly estimate a stochastic policy using an independent function approximator (DNN), whose input is some representation of the current state of the environment ($s_t$), it generates as output action selection probabilities (from which an action $a_t$ is sampled) and whose weights are the policy parameters. The objective stated in Eq. 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 ($L_{\text{PG}}$);

$$L_{\text{PG}} = E[R_\tau \log \pi(\tau)] = E\left[R_t \sum_{t=0}^{H} \log \pi(a_t|s_t)\right] \tag{3}$$

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

# 4 OUR AUTONOMOUS TRAFFIC SIGNAL CONTROL METHODOLOGY

In this section, we describe our signal control agent's implementation, including the MDP settings; state, action, reward specifications.

## 4.1 Problem Definition

Our goal is for our agent to learn a real-world-deployable signal control policy by leveraging diverse traffic data gathered in a visually realistic traffic simulator. In this paper, we develop a 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. To ensure reliable transfer to real-world traffic settings, we progressively train our signal control agent on diverse traffic conditions (such as adverse weather and lighting conditions) in simulation.

## 4.2 Traffic Model Simulation

DRL agents require millions of samples (i.e. interactions with the environment) to learn meaningful policies. Although data gathered in the real world will provide precise signals about the dynamics of the traffic environment, it may suffer from lack of visual diversity as it is costly to gather comprehensive data (i.e. traffic distribution on clear sky, snow, rain, evening and dimly-lit nights, various junction configurations) in the real world. In consequence, simulation is deemed as a safe, cost-effective and controlled tool to train DRL agents. In this work, we train our signal control agent in a variety of complex traffic conditions created using an open-source multi-agent road transportation-based simulation environment with a visual element; Traffic3D (https://traffic3d.org/) [23, 24]. Traffic3D is capable of creating realistic traffic scenarios including extreme traffic and ambient conditions. Situations such as crashes and obstacles do occur and form part of the agent's training. The signal control agent, therefore, learns to deal with them.

## 4.3 Traffic Movement Simulation

Traffic movement is defined as the vehicles navigating across an intersection (from an entrance lane to an exit lane). In this paper, we trained an agent on four-legged standard intersections. We define a set of admissible vehicle movements, eight standard *signal phases* and safety rules (e.g. the minimum prescribed time before signal phase changes) as per the Traffic Signs Manual by the Department for Transport (UK) [31]. In the simulation environment utilised, Traffic3D, 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. To mimic real-world traffic trends, simulations are initialised to reproduce real traffic data obtained at different times of day.

# 5 LEARNING ENVIRONMENT SETUP: MDP SETTINGS

Our simulated traffic environment is illustrated in Fig. 1. At each MDP time-step, the 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 decides 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 (i.e. configuration of signal phases). Following are the MDP settings for our signal control agent; including state, action spaces and reward design.

## 5.1 State Space

Our agent directly maps RGB images (depicting the prevailing traffic state) to actions (controlling the traffic signals), demonstrating end-to-end learning without *any* pre-specification of traffic environment features (such as vehicle density, type, etc). 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. Furthermore, the smaller resolution of the images helps the agent to generalize better to new settings; images containing fewer details of the traffic environment prevent overfitting. Our results in Sec. 7 verify this.

## 5.2 Action Space

While policy-based DRL can handle both continuous and discrete action spaces, a few prior control optimization research works have shown that discrete action spaces work much better [32, 33]. This is because discretization of actions makes learning a good control policy potentially simpler. Therefore, for our signal control task, 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, our signal control agent selects one of the available phases to be implemented for a duration of $T$ seconds (e.g. $5s$). This implies that at each MDP

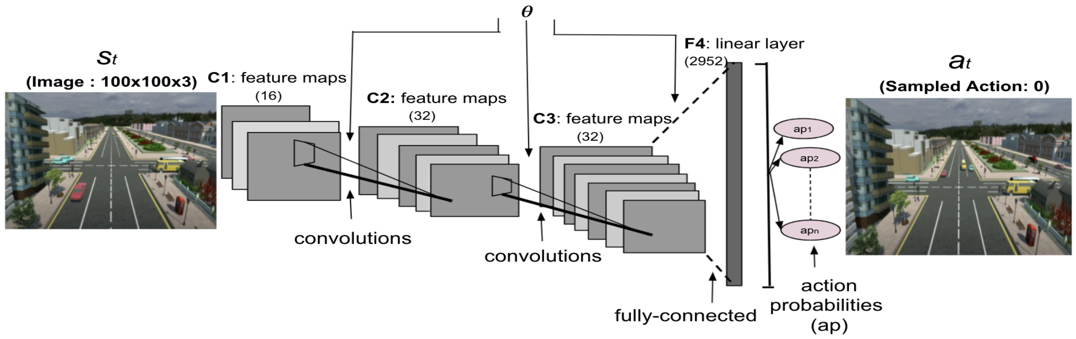


**Figure 2: Our Signal Control Agent's Network Architecture.**

step, a green signal is implemented for a minimum time duration of $T$ seconds. After $T$ seconds elapse, based on the state perceived, the agent may decide to have the same signal phase or change it. Minimum/maximum signal time durations dictated by traffic regulations are conveniently accommodated by our decision making.

## 5.3 Reward Design

Both delay and throughput are often-used metrics to evaluate/optimize the overall state of the traffic. Throughput and delay are inversely proportional to each other and optimizing one also optimizes the other. In this paper, 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 define two reward functions: (1) a positive success reward (e.g. +1) for every civil vehicle passing safely through the intersection; and (2) a penalty (e.g. -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) and public transport vehicles (such as buses) in our experiments. We associate a higher reward of (e.g. +5) for their passing through the intersection and a higher penalty of (e.g. -5) for their waiting at the intersection.

## 5.4 Learning Protocol

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

$$a_t \sim \pi(s_t|\theta) \tag{4}$$

where, $\theta$ denotes the model parameters and $s_t$ is the $100x100x3$ 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. 5,

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta log\pi_\theta(a_t^i|s_t^i)\right)\left(\sum_{t=1}^{T}r(s_t^i,a_t^i)\right) \tag{5}$$

where $J(\theta)$ denotes the loss function.
where $T = 100$, $N = 10$. A local maximum in $J(\theta)$ is searched by ascending the gradient of the policy with respect to parameters $\theta$. $\nabla_\theta J(\theta)$ is the policy gradient and $\alpha$ is a step-size parameter. The policy is updated in the direction of the gradient (Eq. 6) to encourage actions leading to good outcomes and discourage less desirable ones.

$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta) \tag{6}$$

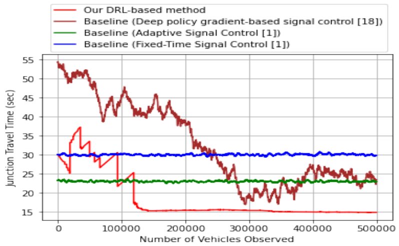


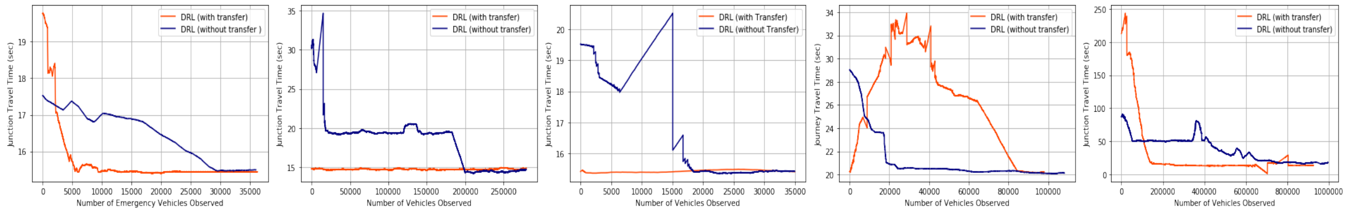
**Figure 3: Quantitative results demonstrating our autonomous traffic signal control agent's performance during training against the baselines; fixed-time [1], adaptive [1] and RL-based [21] signal control.**

## 5.5 Network Architecture

In the current work, we employ a deep neural network with a small number of hidden layers. Additionally, we use batch normalization to prevent overfitting. Batch Normalization is a widely used regularization technique that enables more stable and faster training with improved convergence and generalization of deep neural networks (DNNs) [34]. In this work, we use a convolutional neural network (CNN) as CNNs exploit the advantage of spatial coherence in visual data. Our deep learning network comprises three convolutional layers and one fully-connected layer. This network architecture yields positive signal control in varied traffic conditions.

## 5.6 Domain Randomisation

Domain randomization has been previously used to successfully transfer simulation-trained RL agents to the real world [13, 35, 36]. In this work, to reduce the *reality gap* between the simulated and real-world environments, we modify the basic version of our traffic simulation environment to the distribution of many simulations in order to foster effective skill transfer. The wider the simulation settings variation, the more likely the agent is to capture the real-world dynamics. Fig. 1 depicts some examples of these altered environments. Different aspects of the traffic environment such as



(a) (b) (c) (d) (e)

**Figure 4: Graphs depicting our signal control agent's performance based on average junction travel time (y-axis) over the total number of vehicles observed during the training (x-axis). We compare our DRL approach for traffic optimization; with (red line) and without (blue line) transfer learning. Our learning curves showing vehicles' junction travel time include experiments; (a) In the presence of emergency vehicles. (b) On a dimly-lit night. (c) On a rainy evening. (d) On a snowy day. (e) On a random junction with different (never seen before) geometry.**

lighting or weather conditions are modified to force the agent to learn the essential features i.e. objects of interest. The intuition behind applying domain randomization to our signal control task is that by altering various aspects of our simulated environment (e.g. different weather and lighting conditions), we produce a signal control policy that is less likely to overfit to a certain simulated environment and more likely to successfully transfer to the real-world traffic settings. Our results are aligned with this intuition, reflecting the emergence of an effective real-world (as shown on physical CCTV images) transferable signal control policy trained using only simulator-generated data (Fig.6).

## 5.7 Domain Randomization Protocol

As effective generalization is essential to RL agents' real-world deployment, in this work, we focus on solving the problem of generalization between traffic scenes that visually differ from each other via domain randomization. Domain randomization methods use data from a source domain to improve the performance of the learned model on a target domain. To ensure our vision-based agent's generalizability to dynamically changing traffic conditions both in simulated and real-world settings, we define (a) a source domain and (b) a target domain. To achieve domain randomization, we train our agent to act in the source domain (based on the learning protocol outlined in Sec. 5.4) and reuse its acquired knowledge from the source domain to learn to effectively operate in the target domain. We initialize our agent's convolutional neural network (CNN) in the target domain with our agent's pre-trained CNN parameters from the source domain. The agent is then tuned to operate in the target domain, based on Eq. 7,

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T\times i}\nabla_\theta log\pi_\theta\left(a_t^i|s_t^i\right)\right)\left(\sum_{t=1}^{T\times i} r\left(s_t^i, a_t^i\right)\right) \tag{7}$$

where T = 10 and N = 10 and the policy is updated in the direction of the gradient based on Eq. 6.

## 5.8 Network (Signal Policy) Visualization

Saliency maps are amongst the most popular techniques used to interpret the decisions made by neural networks. Our visualization methodology is based on Grad-CAM (Gradient-weighted Class Activation Mapping) [37]. 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 a localization map for a particular signal control phase regime decision $p$, the Grad-CAM method first computes the gradient of the score $y^p$ (before softmax) with respect to the feature maps $A^k$;

$$g_p(A^k) = \frac{\partial y^p}{\partial A^k} \tag{8}$$

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

$$\alpha_k^p = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^p}{\partial A_{i,j}^k} \tag{9}$$

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);

$$H_{Grad-CAM}^p = ReLU(\sum_k \alpha_k^p A^k) \tag{10}$$

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

# 6 EXPERIMENTS AND RESULTS

The goal of this paper is to optimize the performance of existing traffic signal infrastructure using DRL. However, when applying DRL for signal control, ascertaining *a priori* the empirical settings that will yield a successful/sustainable signal control policy, is virtually impossible. Hence, we conducted a set of *sensitivity analysis* experiments to assess the robustness of our signal control agent to variation in pertinent empirical settings such as the RL algorithm (actor-only [28] or actor-critic [38]), reward signal and camera orientation (used to capture visual input data). Our sensitivity analysis research findings reflected that to effectively optimize the traffic flows through intersections, a combination of visual traffic data captured with a

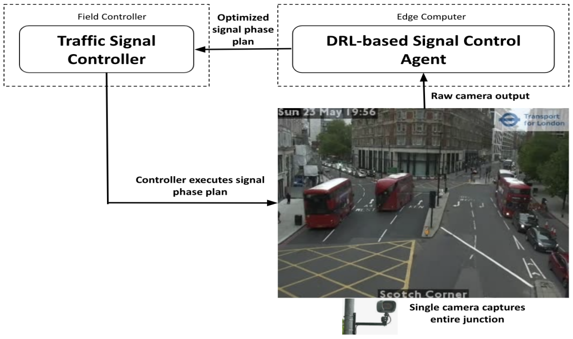

Field Controller
Traffic Signal Controller
Optimized signal phase plan
Edge Computer
DRL-based Signal Control Agent
Raw camera output
Sun 23 May 19:56
Transport for London
Controller executes signal phase plan
Scotch Corner
Single camera captures entire junction

**Figure 5: Real-World Deployment Schema at Scotch Corner, London (using existing TfL camera infrastructure).**

front-camera view, policy-based RL algorithm and positive-negative rewards worked effectively in a wide range of traffic situations. For brevity, we omit our sensitivity analysis results in the current paper. We use this combination of empirical settings in all our simulation-based experiments. While we have no control over camera location or angle for experiments carried out on real-life footage, this does not appear to have adverse effects on the performance of our agent.

We categorize our experiments as (1) DRL-based autonomous signal control. (2) Domain Randomization to ensure the agent's generalizability to environment variations. All our experiments are based on the network architecture described in Sec. 5.5 and illustrated in Fig. 2. Traffic environment specifications, including traffic model and flow details are outlined in Sec. 4.2 and Sec. 4.3, respectively.

## 6.1 DRL Autonomous Traffic Signal Control

This experiment is conducted on a clear day setting (illustrated in Fig. 1(a)). We select the following performance metric 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 vehicles reaching at the end of the junction. The longer a vehicle is forced to wait at the start of the junction, the higher its junction travel-time will be. We take the moving average of 100 vehicles' junction travel-time to capture their long-range trend. Lower journey travel-time indicates better signal control.

We compare our research findings against the following conventional and RL-based baselines:

***Standard (non-adaptive) signal control*** [1]: follows the signal control policy that uses predefined signal phase regimes (widely used for steady traffic conditions).

***Induction loop-based (adaptive) signal control*** [1]: a loop detects approaching vehicles along each incoming lane and an electronic impulse is sent to the signal circuit - to switch the red signal to green.

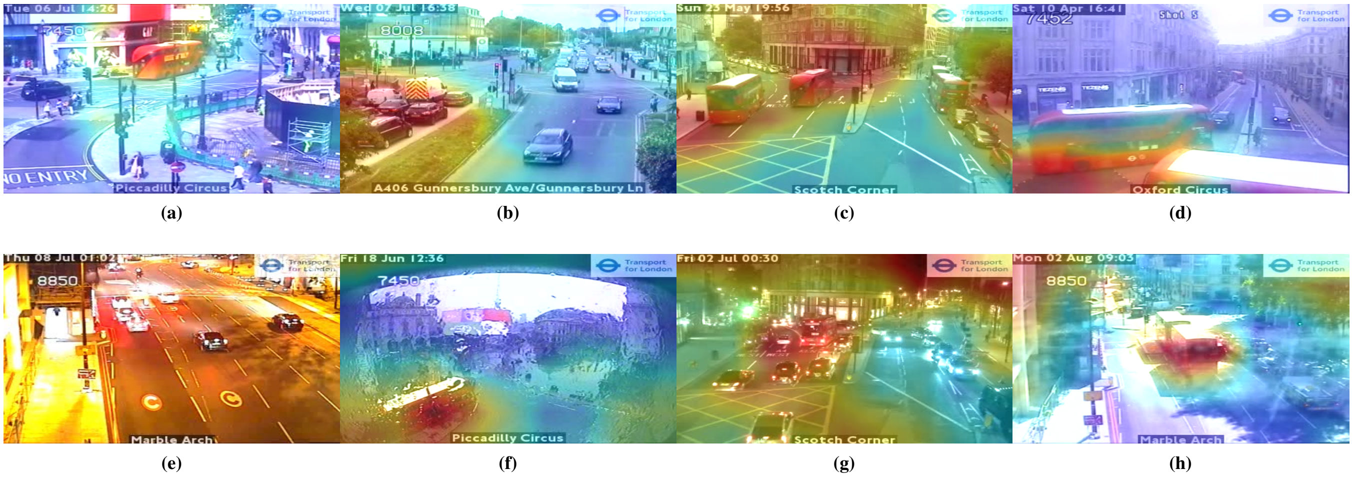
***Deep policy gradient-based (adaptive) signal control*** [21]: a policy gradient algorithm for vision-based traffic signal control (close to our work). The state and action specifications are similar to our proposed method (outlined in Sec. 5). While depending on visual input, the signal control agent is being trained on simplistic/non-realistic camera footage following a less diversified and rigorous approach, hindering its deployability to real settings. The reward signal is based on total cumulative delay (for further details, see [21]). Another vision-based signal control [22] is a value-based approach while our *sensitivity analysis* experiments demonstrated more-effective signal control using policy-based methods.

Our signal control agent's training graph, including the average junction travel time of the total number of vehicles observed during the simulation is shown in Fig. 3. Our signal control agent significantly outperforms both conventional (fixed-time and adaptive) and RL-based signal control methods; intelligently adjusting signals to different traffic situations. Also, as compared to the other DRL-based approach [21] which demonstrates high variance in learning, our method demonstrates faster and more stable (with sustainable policy) learning. We believe that the use of cumulative delay in the baseline [21] leads to inferior performance, as this metric is ambiguous and it does not inform the agent about delays faced by individual vehicles. Fixed-time and loop-induced signal control methods perform the worst. These methods fail to timely modify agents' traffic optimization decisions as per the dynamically changing traffic flow patterns, as there is no learning involved. However, even a learning-based DRL agent relying on vehicle count as the traffic state information performs comparably to loop-induced signal control. We note that using visual traffic data to optimize signals has several benefits including detection of vehicles' type, precise position of vehicles and estimation of speed of vehicles based on their position in consecutive frames.

## 6.2 Domain Randomization Experiments

The main objective of applying domain randomization is to provide enough variability in the simulation environment at training time so that the agent is able to generalize to real-world settings at testing time. Following is the set of our domain randomization experiments;

*6.2.1 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 associate a higher positive reward (+5) for every emergency vehicle's traversal through the intersection and a higher negative reward (-5) for every emergency vehicle waiting at the intersection. We conduct two experiments in this setup: (1) With knowledge transfer from the source domain (signal control on a clear day, outlined in Sec. 6.1); in the target domain 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 knowledge transfer; we initialize our agent with random neural network parameters to prioritize navigation of emergency vehicles. In both transfer and non-transfer experiments, we use a mixture of civil and emergency vehicles in the ratio 10:1. As seen in Fig. 4 (a) (red), the agent equipped with an overall understanding of the traffic environment (from the source task the agent learns to optimize traffic flows on a clear sky day after approx. 22000 time-steps into training) quickly learns (after approx. 5000 time-steps into fine-tuning) to prioritize emergency vehicles' swift movement through the intersection via transfer learning. In contrast, training our agent with random parameters to prioritize navigation of emergency vehicles demonstrated relatively slow learning (blue).



(a) (b) (c) (d)

(e) (f) (g) (h)

**Figure 6: Images demonstrating attention visualization real-world intersections in London city on clear, smoggy, rainy, night and distorted scenes (Marble Arch, Piccadilly Circus, Gunnersbury Lane, Scotch Corner and Oxford Circus obtained from live TfL cameras). There is clear attention on emergency and public transport vehicles and the lane with higher traffic density (in the absence of public transport vehicles); obstructions (e.g. raindrops on camera), adverse lighting and smog do not affect the network's performance.**

*6.2.2 Dimly-lit night.* Since our signal control agent perceives its environment using vision, we believe it is important to validate its agility when subjected to dim lighting (illustrated in Fig. 1 (b)). Our experiments in this set-up include; (1) With transfer from the source domain (signal control including emergency vehicles, outlined in Sec. 6.2.1); in the target domain experiment, we reuse a previously-learned policy from the source domain. (2) Without transfer; we train our agent from scratch with random neural network initializations on a dimly-lit night. As seen in Fig. 4 (b), the agent relying on previously-acquired skill-set (red) learns to minimize the junction travel time for individual vehicles almost instantaneously. In contrast, the agent with the random neural network initializations (blue) takes longer to learn. 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 (from source task the agent learns to optimize traffic flows after approx.27000 time-steps into training and fine-tuning), allows it to quickly adapt to the changing lighting conditions.

*6.2.3 Rainy evening.* Here, our agent learns to optimize traffic flows in the presence of rain (illustrated in Fig. 1 (c)). For these experiments, we simulate in Traffic3D rain of $10mm/h$. In this setup, we conduct two experiments: (1) With transfer from the source domain (signal control including emergency vehicles and dim-lighting, outlined in Sec. 6.2.2); in the target domain experiment, we reuse a previously-learned policy from the source domain. (2) Without transfer; we initialize our agent with random neural network parameters to optimize the flow of traffic on a rainy evening. As seen in the graph of Fig. 4 (c), the agent making use of learned policy (red) learns to reduce junction travel time for individual vehicles almost instantaneously. A heavy rain of $10mm/h$ has little/no effect on our agent's ability to interpret the fundamental traffic scene (from source task the agent learns to optimize traffic flows after approx.$27K$ time-steps into training). In contrast, the agent initialized with random neural network parameters (blue) does not have any pre-existing knowledge to build on, in consequence, it learns relatively slowly.

*6.2.4 Snowy day.* Here, our agent learns to optimize traffic flows in the presence of snow (illustrated in Fig. 1 (d)). In this setup, we conduct two experiments; (1) With transfer from the source domain (signal control including emergency vehicles, as well as dim lighting and rain, outlined in Sec. 6.2.3); in the target domain experiment, we reuse a previously-learned policy from the source domain. (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. 4 (d) indicate initial negative transfer as agent learning via transfer learning (red) performs worse than the agent using the random initializations (blue). 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 fewer 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 as per the need to pre-train agents for snowy scenes prior to deployment.

*6.2.5 Different Junction Layout.* Here, we establish the ease of deployment of our signal control agent to new junctions with varied topologies/structures. Our experiments in this set-up include: (1) With transfer from the source domain (signal control including emergency vehicles, as well as dim-lighting and rain, outlined in Sec. 6.2.4); in the target domain experiment, the agent reuses the previously-learned policy from the source domain. (2) Without transfer; the agent is trained with random neural network initializations to optimize traffic flows through a new (visually different) junction. The difference between the junction layouts in the source (4-legged

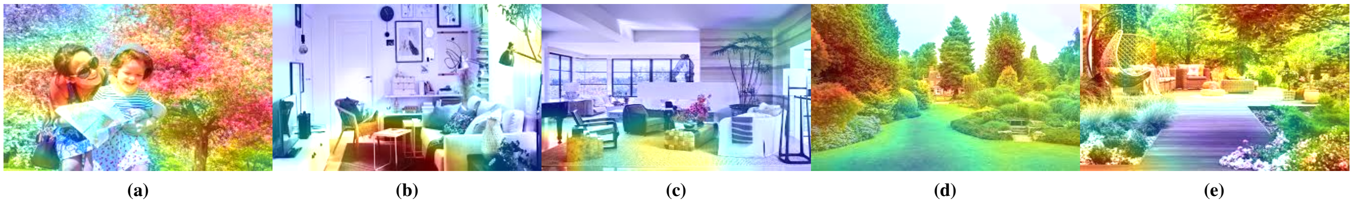


**Figure 7: Images demonstrating attention visualization on unrelated (non-traffic) scenes, after the network has been trained on a cross-junction scene. Areas of significant attention, visualised in red, appear in random places in these images.**

junction) and target (2-legged junction) domains is that the 4-legged junction has four traffic lights and the 2-legged junction has two traffic lights. The results of these experiments are shown in Fig. 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.

## 7 DEPLOYMENT AND EVALUATION IN THE REAL-WORLD

Our vision-based signal control system is real-world deployable without the need for expensive infrastructure upgrades. For example, Transport for London publishes real-time footage from its network of traffic cameras in the city of London (www.tfljamcams.net). Illustrated in Fig. 5, is the proposed execution of DRL-based signal control on a real-world intersection (in this case Scotch Corner, London). Our signal control agent will sense the environment in real time using raw camera footage. It will then process this information and determine a signal optimization policy to move the traffic through the intersection as efficiently as possible. Lastly, via software integration/API, the agent will send commands to the controller to implement the optimized signal phase plan. At settings monitored by multiple cameras, it is possible to combine all streams to obtain an extensive state of the intersection and apply our signal control DRL algorithm on the combined input space.

Aiming to evaluate the deployment readiness of our vision-based signal control method, we demonstrate the attention visualization of our signal control policy (trained entirely on simulated footage of a four-legged intersection on Traffic3D) on TfL CCTV images of intersections in London; Piccadilly Circus, Gunnersbury Lane, Scotch Corner, Marble Arch and Oxford Circus (illustrated in Fig. 6), which have significantly different layouts. While no actions are taken in this experiment, the attention visualisation demonstrates that our policy is able to accurately recognise different vehicles on real intersections (i.e. public transport and emergency vehicles to give them priority access of the intersection). Additionally, Figs. 6(e)-(h) demonstrate the successful attention visualization of our signal control policy on real-world intersections, on the scenes affected by heavy rain, night-time lighting and distorted camera output. In the absence of public transport and emergency vehicles, attention can be seen on the lane with higher traffic density (around the vehicles closer to the intersection), in line with the training the agent has experienced. Testing our signal control policy on varied real-world traffic data provides us with a close proxy of our DRL-based agent's performance in real-world traffic settings. While even the smallest of perturbations to an agent's input state representation can lead to undesirable outcomes, the use of domain randomization counters this issue by exposing the agent to a variety of settings during training. Fig. 6 demonstrates our agent's ability to transfer from simulation to real-world settings comprising visual traffic data captured with camera angles we have no control over, different weather conditions (clear and rain), lighting (day and night) and types of intersection layouts to which the agent has never been exposed to during its training phase. This strongly indicates that our agent does not overfit to the training data and is robust to distributional shift.

To further validate the efficacy of our DRL-based signal control agent, we applied attention visualisation in the same way to a set of unrelated (non-traffic) scenes (www.gettyimages.co.uk). This is common practice in attention visualisation-based system verification; it confirms that the trained policy is only acting upon relevant features [39, 40]. Our signal control policy trained on traffic scenes shows attention at random places on the unrelated scenes ( Fig. 7). This strongly emphasizes that our agent learns to recognise and act upon features that are relevant to the traffic optimization task.

## 8 CONCLUSION

We presented a vision-based, end-to-end trainable autonomous traffic signal control agent. Our agent optimizes traffic based *solely* on *live* visual traffic data, without hand-crafted traffic state features. Our agent, which has been trained with domain randomisation, achieves individualized signal control that autonomously adapts to varying junction types, traffic distribution, weather and lighting conditions, both in simulation and the real world. Using attention visualization, we advance towards explainable AI and translate our agent's signal control decisions into a human-understandable form. This further helps us gain insight into our end-to-end (jointly learning perception and control) signal control approach. We further highlight the importance of using simulations to train autonomous agents by demonstrating that the agent trained entirely on simulated scenes employing domain randomization produces a signal control policy that can be successfully transferred to the real world with no pre-training.

In the future, we intend to deploy multi-agent RL to control networks of intersections, exploring formal logic and probabilistic verification of our signal control agent and the underlying simulation.

## REFERENCES

[1] Peter Koonce and Lee Rodegerdts. Traffic signal timing manual. Technical report, United States. Federal Highway Administration, 2008.

[2] Janusz Gajda, Ryszard Sroka, Marek Stencel, Andrzej Wajda, and Tadeusz Zeglen. A vehicle classification based on inductive loop detectors. In *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No. 01CH 37188)*, volume 1, pages 460–464. IEEE, 2001.

[3] PB Hunt, DI Robertson, RD Bretherton, and M Cr Royle. The scoot on-line traffic signal optimisation technique. *Traffic Engineering & Control*, 23(4), 1982.

[4] Arthur G Sims and Kenneth W Dobinson. The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on vehicular technology*, 29(2):130–137, 1980.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[6] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[7] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

[8] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019.

[9] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020.

[10] Deepeka Garg, Maria Chli, and George Vogiatzis. Deep reinforcement learning for autonomous traffic light control. In *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pages 214–218. IEEE, 2018.

[11] Deepeka Garg, Maria Chli, and George Vogiatzis. A deep reinforcement learning agent for traffic intersection control optimization. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4222–4229. IEEE, 2019.

[12] Deepeka Garg, Maria Chli, and George Vogiatzis. Multi-agent deep reinforcement learning for traffic optimization through multiple road intersections using live camera feed. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[13] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[14] Fo Vo Webster. Traffic signal settings. Technical report, 1958.

[15] Alan J Miller. Settings for fixed-cycle traffic signals. *Journal of the Operational Research Society*, 14(4):373–386, 1963.

[16] Aleksandar Stevanovic. *Adaptive traffic control systems: domestic and foreign state of practice*. Number Project 20-5 (Topic 40-03). 2010.

[17] Elise Van der Pol and Frans A Oliehoek. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, 2016.

[18] Wade Genders and Saiedeh Razavi. Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint arXiv:1611.01142*, 2016.

[19] Juntao Gao, Yulong Shen, Jia Liu, Minoru Ito, and Norio Shiratori. Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. *arXiv preprint arXiv:1705.02755*, 2017.

[20] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. Deep reinforcement learning for traffic light control in vehicular networks. *arXiv preprint arXiv:1803.11115*, 2018.

[21] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, 11(7):417–423, 2017.

[22] Hyunjeong Jeon, Jincheol Lee, and Keemin Sohn. Artificial intelligence for traffic signal control based solely on video images. *Journal of Intelligent Transportation Systems*, 22(5):433–445, 2018.

[23] Deepeka Garg, Maria Chli, and George Vogiatzis. Traffic3d: A new traffic simulation paradigm. 2019.

[24] Deepeka Garg, Maria Chli, and George Vogiatzis. Traffic3d: A rich 3d-traffic environment to train intelligent agents. In *International Conference on Computational Science*, pages 749–755. Springer, 2019.

[25] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 2011.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[27] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[28] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[29] Leemon Baird and Andrew W Moore. Gradient descent for general reinforcement learning. *Advances in neural information processing systems*, pages 968–974, 1999.

[30] LA Prashanth and Shalabh Bhatnagar. Reinforcement learning with average cost for adaptive control of traffic lights at intersections. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1640–1645. IEEE, 2011.

[31] Traffic signals manual, 2019.

[32] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[33] Wojciech Jaśkowski, Odd Rune Lykkebø, Nihat Engin Toklu, Florian Trifterer, Zdeněk Buk, Jan Koutník, and Faustino Gomez. Reinforcement learning to run… fast. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 155–167. Springer, 2018.

[34] Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. *arXiv preprint arXiv:1809.00846*, 2018.

[35] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

[36] Katie Kang, Suneel Belkhale, Gregory Kahn, Pieter Abbeel, and Sergey Levine. Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight. In *2019 international conference on robotics and automation (ICRA)*, pages 6008–6014. IEEE, 2019.

[37] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[38] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[39] Harsh Panwar, PK Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Prakhar Bhardwaj, and Vaishnavi Singh. A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images. *Chaos, Solitons & Fractals*, 140:110190, 2020.

[40] Ho-Taek Joo and Kyung-Joong Kim. Visualization of deep reinforcement learning using grad-cam: how ai plays atari games? In *2019 IEEE Conference on Games (CoG)*, pages 1–2. IEEE, 2019.