A Deep Reinforcement Learning Agent for Traffic Intersection Control Optimization

Deepeka Garg, Maria Chli and George Vogiatzis

***Abstract*— The efficiency of traffic flows in urban areas largely depends on signal operation. The state-of-the-art traffic signal control strategies are not able to efficiently deal with varying or over-saturated conditions. To optimize the performance of existing traffic signal infrastructure, we present an end-to-end autonomous intersection control agent, based on Deep Reinforcement Learning (DRL). In the recent years, DRL has emerged as a powerful tool, solving control problems involving sequential decision making and demonstrating unprecedented success in complex settings. Our DRL traffic intersection control agent configures the traffic signal regimes based *solely* on live photo-realistic camera footage. We demonstrate that our agent consistently, significantly outperforms state-of-the-art fixed (pre-defined) and adaptive (induction loop-based) signal control methods under a wide range of ambient conditions, by increasing the traffic throughput and decreasing the intersection traversal time for individual vehicles.**

I. INTRODUCTION

Traffic congestion is a serious problem, associated with substantial economic and environmental costs. Poorly-managed traffic signals are known to be among the major causes of traffic congestion in urban traffic networks [1]. Over the years, as the variability and unpredictability of traffic have outpaced the capabilities of traffic light systems on predefined, fixed time plans - the most common means of intersection control - to operate efficiently, significant effort has been devoted towards online adaptive traffic signal control. In contrast to pre-timed fixed signal control which repeats a preset signal control regime, online adaptive signal control is capable of responding to the presence of vehicles at the intersection by adjusting the regime according to changing traffic patterns in real time. Nevertheless, the state-of-the-art adaptive control mechanisms are unable to deal with congestion efficiently enough. These mechanisms are either based on application-specific heuristics [2] or they rely on systems which fail to replicate the traffic flow accurately - as the real-world traffic phenomena include highly-stochastic driving dynamics, such as sudden accidents blocking the flow. For effective utilization of existing traffic signal control systems, it is critically important to carry out their optimization using automated agents, capable of learning, self-configuration and self-optimization.

Since the 1990s, Reinforcement Learning (RL) is considered as a direct approach to adaptive optimal control of non-linear systems. RL agents accomplish tasks by monitoring the environment through perception, influencing it by applying actions and learning by observing the outcomes of the actions [3]. Deep Reinforcement Learning (DRL) (a mechanism combining reinforcement and deep learning), in the recent years, has emerged as a powerful solution for control problems involving sequential decision-making; demonstrating unprecedented success in complex, dynamic and high-dimensional settings such as Atari games [4], among others. To accomplish a certain task, the DRL agent continuously interacts with its intended environment and it *learns* the set of environment features that are significant in each problem. Establishing the correlation between the actions taken by the agent and their subsequent effect on the environment is the most fundamental aspect of this interaction.

Fig. 1: A view of Traffic3D's Graphical Display

In a previous proof-of-concept paper [5], we demonstrated the efficacy of a DRL agent in optimizing traffic through a road intersection. In the current work, our DRL agent optimizes the traffic signal regimes under varying ambient conditions (such as different traffic densities, different vehicle types, different weather and lighting conditions) perceived *solely* using high-quality photo-realistic visual traffic data. To perceive the pertaining situation of the intersection traffic environment holistically, we use a deep neural network (DNN). Breakthrough developments in the field of deep learning have made it possible to learn intricate feature representations directly from high-dimensional raw data such as images, videos and audio [6]. DRL applied to visual traffic data from an intersection eliminates the need for pre-determined hand-crafted features describing the environment

The authors are based at the Computer Science Department, School of Engineering and Applied Science, Aston University, Birmingham, United Kingdom, e-mail: [gargd, m.chli, g.vogiatzis]@aston.ac.uk.

state. In addition, it enables optimisation of decision making based on visual features (e.g. vehicle type and approach speeds) that would otherwise be impossible or impractical using other methods of traffic data collection (such as induction loops). The nature of the visual input data used (raw pixels) and possible extensions (combining different camera views to optimize traffic through multiple junctions) make it a very powerful resource, which we fully exploit in the current work through an appropriately trained and tuned DRL system. Furthermore, in the current work, we utilise the Macroscopic Fundamental Diagram (MFD) [7] alongside measuring junction travel time, to assess our DRL agent's performance based on established transportation metrics - traffic density (veh/km), traffic flow (veh/hr) and speed (km/hr). Our results demonstrate that our agent successfully optimizes the navigation of vehicles through intersections in varying traffic conditions such as traffic densities, weather and lighting.

II. RELATED WORK

Here, we briefly introduce the conventional signal control methods, followed by incorporation of reinforcement learning (RL) methods for signal control. Lastly, we summarize state-of-the-art traffic simulation platforms.

A. Traditional Signal Control Methods

Current traffic signal control systems operate either in pre-defined fixed mode, adaptive/actuated mode, or in combination of the two. The pre-defined signal control regimes are configured based on historical traffic data. Traffic data is collected by continuously monitoring the traffic networks using sensors (such as loop detectors) and subsequently drawing inference on key information, such as establishment of critical junctions and their congestion trends. However, with real-world traffic phenomena exhibiting highly-stochastic dynamics, their irregularities cannot be *a priori* anticipated based *solely* on historical data. An alternative is online traffic monitoring (e.g. using inductive loop sensors) and subsequently configuring the signal regimes in real time.

Table I summarizes some of the more-widely used methods in adaptive traffic signal control. The sensors can be located upstream of stop lines at the entrance of link, or downstream from the previous junction. A variety of sensors is used to monitor traffic. The most commonly used sensors are Loops (underground vehicle detection) and Microwave Detectors (above-ground vehicle detection). 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 is of key importance in these approaches. For instance, SCOOT is known to lose its capability to faithfully detect traffic if 15% of the loop sensors are faulty. Buried under the road, the loop detectors have 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.

B. Reinforcement Learning-based Signal Control Methods

The classical traffic modeling and analysis tools used by transportation planning agencies (summarized in Table I) struggle to provide tractable policies to optimize the performance of traffic signal infrastructure. Due to their complexity (involving uncertainly, imprecision and randomness), traffic dynamics form a complex, non-linear system and are required to be modeled with complex dynamical systems. This inspired the research community to move towards utilizing better-suited methods for real-time traffic management, including learning-based paradigms such as Reinforcement Learning (RL) [3]. RL was first applied to traffic light control in the 1990s, with the first techniques limited to tabular-Q learning [17]. Table II compares the recent work on traffic signal dynamic control using DRL. Most of the previous works on DRL-based traffic light control [12], [13], [14], [16] does not consider high-dimensional perpetual inputs. The state representation they use consists of hand-crafted traffic features; a Boolean-valued vector specifying the presence of a vehicle at the intersection. Along with this Boolean vector, some research works also include a vector recording vehicles' speed. We believe that these vectors do not render the complete traffic state representation such as types of vehicles and their exact position. Furthermore, all these studies rely on SUMO [18] to simulate traffic. We believe that certain aspects of SUMO (such as its random incident management protocol [19]) trade away the inherent complexities of vastly stochastic traffic dynamics for tractability of analysis. This hinders the applicability of the resulting solutions to a real world setting.

Mousavi et al. [15] used raw pixels to analyse the prevailing situation of the junction. We believe that their research work takes a step in the right direction, but the simulation tool they used to test and evaluate their research approach; SUMO, is not photo-realistic and as previously discussed, does not incorporate the full complexity of urban traffic [19]. This is likely to limit their agent's ability to track all significant features and its readiness to be transferred to a real environment. Furthermore, their work does not explore agent's performance in variable environments such as varying traffic densities and it is unlikely to reliably be able to respond to different vehicle types (such as emergency vehicles) or weather conditions. In contrast, we configure the signal regimes under varying traffic conditions in real time *solely* based on live photo-realistic camera footage (complete 3D-picture of the pertaining traffic situation). Our simulation platform; Traffic3D [20] gives the learning agent a natural, unstructured and interactive environment to operate on.

C. Traffic Simulation

Researchers in the area of traffic and transportation maintain that simulations provide a safe, controlled and accelerated environment for protocol development. The most prominent state-of-the-art traffic simulators often fail to deliver

Control Technique	Traffic Data	Control System	Optimizing Performance Metric
SCAT [8]	Online data (from stop-line downstream detectors)	Centralized	Junction throughput, travel time
SCOOT [8]	Online data (from upstream detectors)	Centralized	Delay, stops and congestion
UTOPIA [9]	Online data (from upstream detectors)	Centralized	Delay and stops
MOVA [10]	Online data (from a single upstream detector)	Decentralized	Delay, congestion and stops
OPAC [11]	Online data (from upstream detectors)	Decentralized	Delay and stops
Our study	**Online data (from cameras)**	**Centralized**	**Traffic throughput, junction travel-time**

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

Research Study	State Space	Reward	Simulator
Van der Pol and Oliehoek [12]	Position of vehicles	Teleport, wait time, stop, switch and delay	SUMO
Genders and Razavi [13]	Position & speed of vehicles	Cumulative delay	SUMO
Gao et al. [14]	Position & speed of vehicles	Cumulative wait time	SUMO
Mousavi et al. [15]	Raw pixels	Cumulative delay	SUMO
Liang et al. [16]	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 II: Summary of recent deep reinforcement learning-based traffic light control research studies.

important functionalities that are fundamental to authentic traffic simulation. They lack in realism, richness, diversity and perception challenges of the real-world [19]. To address the discrepancy between simulations and real-world traffic characteristics, we created a traffic micro-simulation tool, Traffic3D [20], [21]. Traffic3D allows us to have 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 e.g. we precisely calibrate important traffic parameters including faithful modelling of complex physical interactions between the transportation entities based on mass, friction and other forces (such as gravity).

Furthermore, from our research perspective, to optimize the performance of traffic signal infrastructure using DRL, using simulations is the only reliable way. Since the learning agent has no prior knowledge of its environment, it is bound to have a large amount of interactions with the environment to learn a suitable policy to effectively optimize the signal regimes. This makes it infeasible to train an agent in the real world (due to economic and safety concerns), which persuaded us to develop our high-quality 3D traffic simulation platform. Traffic3D, rich in content and structure, effectively reproduces real-world dynamic and diverse traffic scenarios; encompassing adequate physical and visual behavior of traffic entities (such as photo-realism, faithful simulation of individual vehicle behavior and precise physics of vehicle movement). In [21], we discussed the merits and demerits of Traffic3D compared to other state-of-the-art traffic simulation platforms. Fig. 1 shows an example of our simulator's graphical display.

III. Background and Notation

In this section, we discuss the underlying concepts involving our signal control agent's implementation - Deep Reinforcement Learning.

A. Deep Reinforcement Learning

In a typical RL setting [22], an agent learns to act in an unseen physical environment by interacting with it. The agent improves its learning by receiving a 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 analyse the quality of action taken. A standard RL framework is modelled mathematically as a Markov Decision Process (MDP), 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 and 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 : $S \times A \times S \mapsto \mathbb{R}^+$ 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 the optimal policy π^*

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

A deep neural network (π_θ) [6] with parameters θ in the high-dimensional RL settings represents policy π^*. The agent aims to learn θ^* that achieves the highest expected episodic return,

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

1) Policy Gradient Reinforcement Learning: Neural Network-based function approximation [6], to map input traffic state to a traffic signal control action, is essential for RL to be effective in large high-dimensional state spaces. A dominant approach has been value-function approximation, in which the action-selection policy is implicitly represented as a 'greedy' policy with respect to the estimated values (for instance, as the policy which selects the action with highest estimated value in each state). The value function approach is known to work well in many applications, but it has many limitations such as it cannot efficiently learn stochastic policies and it is less-effective in high-dimensional action spaces [23]. In the current work, we explore an alternative approach to function approximation, a policy-based approach (known as Policy Gradient). Instead of estimating the value function, a stochastic policy is directly estimated using an independent function approximator (a neural network), 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_{PG}),

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

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

Our research approach centers around integrating a reinforcement learning algorithm (i.e. Policy Gradient) with a deep convolutional neural network (DCNN), which directly learns from RGB images to configure effective signal regimes. We use a DCNN in our research, as DCNNs have yielded a substantial performance boost for various visual-based tasks [6].

IV. Our Autonomous Intersection Signal Control Protocol

In this section, we set out our formulation for the autonomous traffic signal control problem.

A. Problem Formulation

Vision-based traffic signal control requires a mapping from visual sensory signals to configuration of signal regimes. Our agent's goal essentially is to learn an end-to-end mapping from photo-realistic images (depicting a 3D traffic setting) to an action (controlling signals). It does so using a DRL model, which aims at learning an effective policy function π^* with a representation of current state s_t as input, to produce the probability distribution over the action space A.

B. Learning Environment Setup

We define the key ingredients of our DRL environment setup including the observation, action spaces and reward signals.

1) Observations: We utilized real world traffic videos as a reference and meticulously studied real world traffic dynamics to create physically intelligent and photo-realistic 3D-traffic intersection scenarios. In our traffic simulator; Traffic3D, material, texture, light and scale work in synergy to make digital content look as close to a real scene as possible [20], [21].

For our signal control optimization task, we define a traffic environment E, consisting of a four-way road intersection (shown in Fig. 1). Traffic flows in directions - east (E), west (W), north (N) and south (S). A four-way junction consists of four traffic lights, all managed by a single signal control agent which decides the configuration of these traffic lights based on prevailing traffic situation in and around the intersection. Our learning agent perceives the current situation of the traffic using visual signals i.e. RGB images taken by a camera placed at the intersection. To economise on memory and accelerate the training process, we resize the images to a compact resolution of 100 x 100. This is highlighted to aid in fully appreciating our results; downsizing the images causes degradation in their quality and may adversely affect the effective recognition of objects within the traffic environment.

2) Action Space: We define a fixed set of discrete action space A, deciding the configuration of traffic signal regimes. For a four-way intersection (shown in Fig. 1) with four traffic lights $< L_1, L_2, L_3, L_4 >$, the action space includes four actions $< A_1, A_2, A_3, A_4 >$ such that each action controls each traffic light $< A_1 \mapsto L_1, A_2 \mapsto L_2, A_3 \mapsto L_3, A_4 \mapsto L_4 >$. Essentially, the agent decides which directions of travel get a green light and for how long. We have a minimum green signal time of 5 seconds, while maximum green signal time is decided by our agent depending on the traffic density and the type of vehicles around; e.g. in the presence of an emergency vehicle, our agent switches the signals to prioritize its movement through the intersection while ensuring safety for all vehicles.

3) Reward Design: Transportation engineering literature reflects that both delay and throughput are the suitable measures to indicate the overall state of the traffic [24]. Throughput and delay are inversely proportional to each other and optimizing delay also optimizes throughput and vice versa. In the current work, our agent's focus is to optimize the traffic throughput across a junction and consequently decreasing the junction traversal time and delay for vehicles, a task for which two reward signals would suffice: (1) a success reward for vehicles safely passing through the junction; and (2) a penalty for vehicles waiting at the junction.

C. Training Protocol

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

$$a_t \sim \pi(s_t|\theta) \quad (4)$$

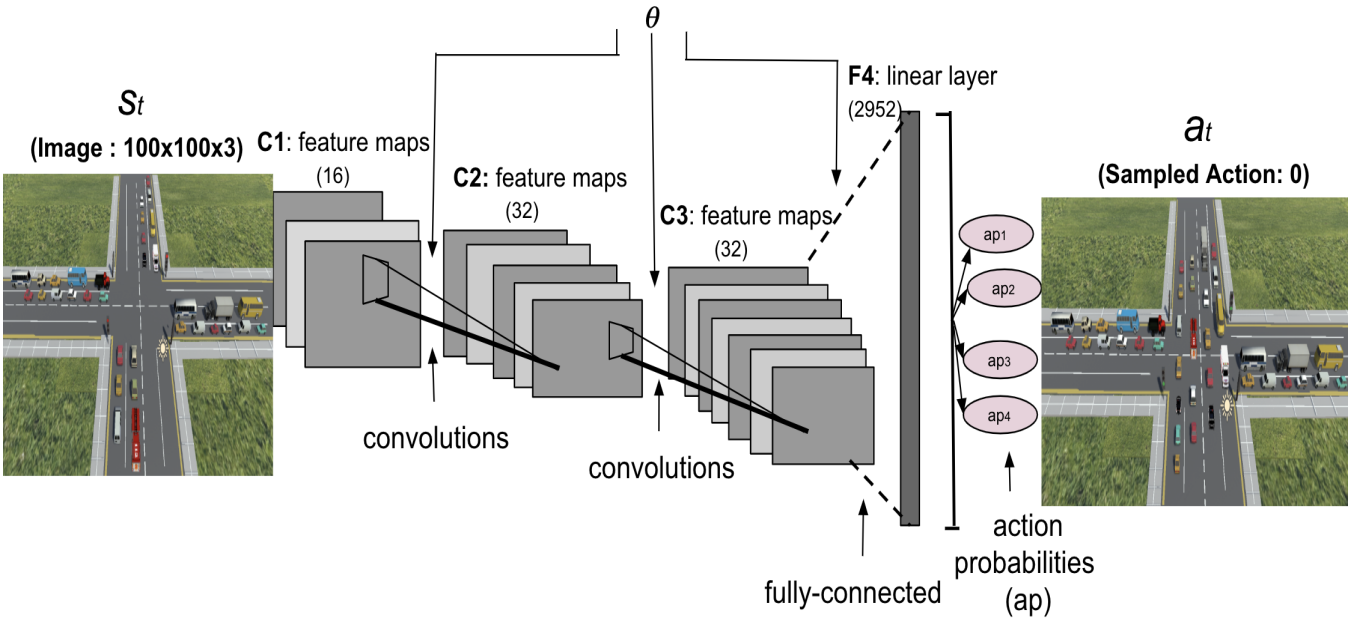


Fig. 2: Our Learning Agent's Network Architecture.

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. 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, 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, to encourage the actions leading to good outcomes and discourage less desirable ones.

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

D. Network Architecture

Our DCNN is composed 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 a RMSProp optimizer [25] of learning rate 0.001. As illustrated in Fig. 2, the network takes an RGB image as input of the current traffic situation and produces the action probabilities as output, from which an action deciding on the current configuration of the traffic signal regime is sampled. Our traffic scene input image (shown in Fig. 2) gives our deep learning network (DNN) an opportunity to explore many more traffic features than just counting the vehicles (as conventionally done by induction loops).

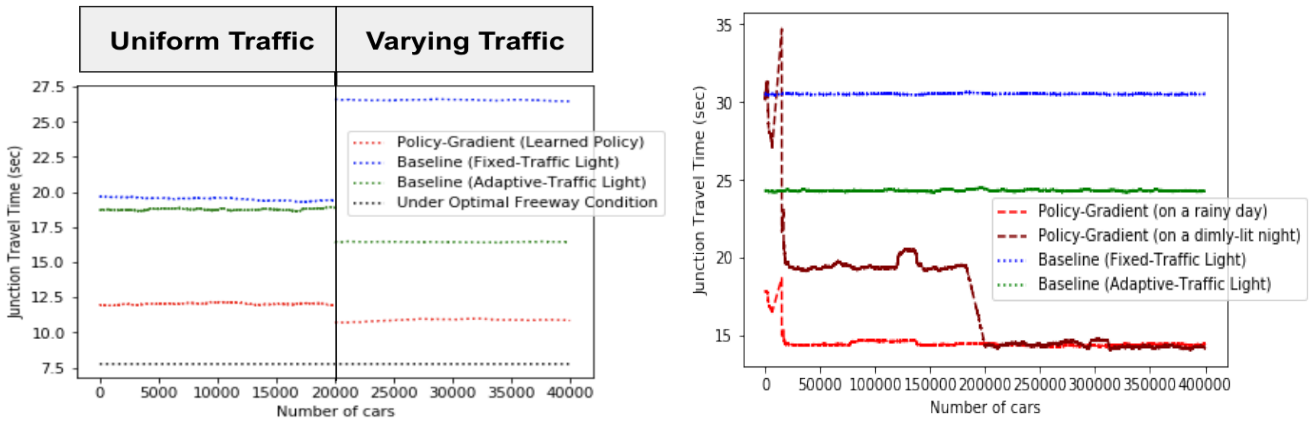
V. Experiments and Results

Our experiments are based on the learning network architecture (shown in Fig. 2), described in Section IV-D. To validate our research findings, we perform comparisons against the following baselines: (1) standard, non-adaptive signal control, (2) induction loop-based adaptive signal control for a single junction (as shown in Fig. 1). Multi-junction optimisation is an interesting extension, which we leave for future work. We evaluate our research findings based on two performance metrics; (1) junction travel time (i.e. the time period between the vehicle reaching the junction stop-line and the vehicle reaching the end of the junction), and (2) the macroscopic fundamental diagram (MFD) [7].

A. Performance in uniform and varying traffic density

We conducted a training experiment, based on architecture shown in Fig. 2. Our agent learns an effective policy approximately half a million time-steps into training, based on a reward design of +1/car passing through the intersection and -1/car stopping at the start of the intersection (stop-line). The policy achieves levels of throughput that significantly outperform our baselines - for brevity we did not include the training plot here.

1) Experiment Description: We use the learned policy (i.e. our trained agent) to perform evaluation experiments to demonstrate the efficacy of our agent in two settings: (1) Uniform and constant traffic generation in all directions. (2) Varying and random traffic generation in each of the directions. In the uniform traffic generation scheme, cars are spawned with a constant uniform density distribution (1000 cars/hour/lane), while in the varying traffic generation scheme, cars spawned follow a variable random density distribution ranging between very high traffic arrival rates



(a) (b)

Fig. 3: Graphs depicting cars' junction travel time versus number of cars observed from the start of the experiment. We plot the moving average of 100 cars' junction travel time. The lower the junction travel time, the better. (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.

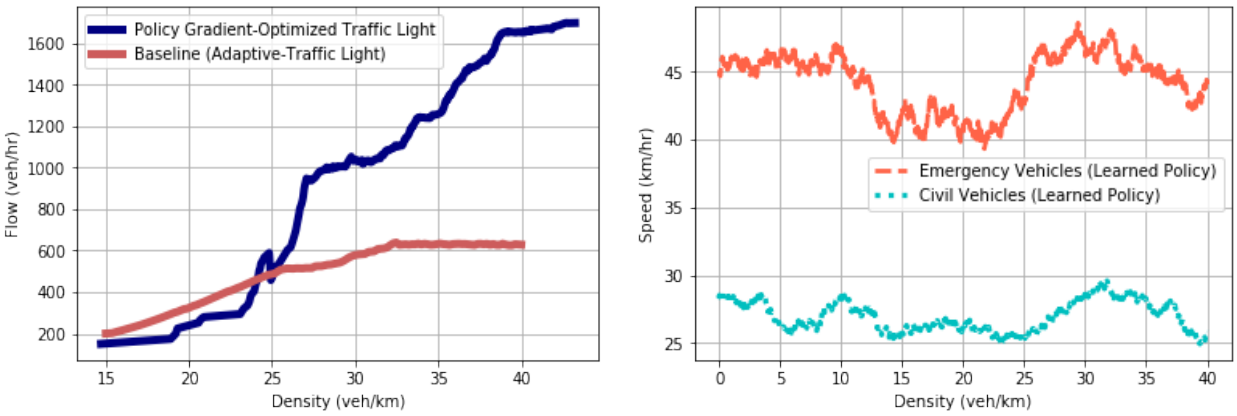
at some instants (5000 cars/hour/lane) and no traffic at all at other instants, i.e. the situation where vehicles are spawned on a road leading to the intersection. Our agent decides which lane should be given green signal and for how long. The cars can either go straight or turn right (path selection probability is parameterizable in our simulator [21]). After a fixed minimum green signal time of 5 seconds, the agent may autonomously decide whether to switch to red depending on the traffic around the intersection.

2) Results: To evaluate our DRL agent's performance in terms of reduced junction traversal time for individual vehicles, we use junction travel time versus number of cars metric. The graphs shown in Fig. 3 depict our agent's performance based on average junction travel time (y-axis) over the number of cars (x-axis). We plot 100 cars' junction travel time's moving average. The lower the junction travel time, the better. The number of cars shown in x-axis represent 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). N.B. the x-axis represents the total number of cars being spawned into the scene in the complete evaluation phase (i.e. at $x = 20,000$, we do not mean that there are 20,000 vehicles in the scene, but that 20,000 vehicles have navigated through the junction from the start of the experiment).

Ideally, the optimal delay for an individual vehicle is no delay at all. In order to create a worthwhile benchmark against which we compare our research approach, 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. This is plotted as the Under Optimal Freeway Condition line. As shown in the Fig. 3(a), our agent performs significantly better than both fixed and adaptive baselines under both uniform and varying traffic schemes. It is noteworthy that our trained agent's performance does not significantly change with uniform and varying traffic density distribution. We believe that using visual sensors to detect traffic is highly beneficial. As observed during the training, our agent is able to effectively detect the presence of cars even when they have not arrived at the junction and are far from it but within the camera range. The agent is also able to infer cars' speed based on their positions in consecutive frames, which equips it to timely decide the configuration of traffic signal regimes to minimize delay for individual cars. This is in contrast to the adaptive systems used in real world (such as induction loops) to detect traffic, which have some grave limitations in terms of narrow operational range and poor detection of small vehicles, among others. Also, the fixed and adaptive systems work on a maximum green time period set by the user. As reported in the literature, maximum green time is usually set between 90-120 seconds [26], this can lead to setting the green light for longer than it is needed.

B. Impact of different lighting and adverse weather conditions

1) Experiment Description: We aim at creating an agent that is robust to different variants of its environment. Our simulator; Traffic3D allows us to realistically simulate different weather (such as rain and snow) and lighting (such as dimly-lit night) conditions with photo-realistic lighting and texture. To establish the resilience of our agent in dim lighting and adverse weather conditions, we simulated a dimly-lit night scene and a rainy day scene within our traffic environment. Apparently the rain and dim-lighting degrade the quality of the visual data, however our agent is still able to learn to optimize the traffic flows, which can be seen in figure 3(b) (shown in red and maroon).



(a) (b)

Fig. 4: The MFDs demonstrating our DRL (Policy Gradient) agent's performance vs adaptive traffic signal control baseline baseline. (a) Density vs Flow of all vehicles. (b) Density vs Speed of emergency vehicles against civil vehicles.

2) Results: The performance metric (junction travel time) used here is same as the previous section. Rain does not have much of a negative impact on our agent's performance, our agent quickly learns to optimize the movement of vehicles through the junction (red) on a rainy day. However, when exposed to a dimly-lit night, our agent starts to learn slowly as the dark pixels significantly reduce the visibility. But it eventually reaches its peak performance (maroon).

Our agent, when exposed to both adverse lighting and weather conditions, performs significantly better than both fixed and adaptive baselines, which marks our agent's resiliency to different conditions it is exposed to.

C. Macroscopic Fundamental Diagram

In this paper, we consider an intersection shown in Fig. 2 and we collect downstream traffic data from our DRL-optimized signalized traffic junction. We derive the relationships between macroscopic variables; (1) density and flow, (2) density and speed. We collect vehicle count and speed of individual vehicles after the vehicles have safely passed through the junction. We then calculate density (i.e. number of vehicles per kilometer) by dividing the vehicle count by average distance travelled by vehicles. We estimate each individual vehicle's speed by dividing the distance travelled by time taken. We then compute flow (i.e. number of vehicles per hour) by multiplying density with average speed of all the vehicles passing through the junction in a time window of 15 seconds. We sample density, speed and flow every 15 seconds. Within our traffic simulator, the elapsed-time metrics correspond to the actual real world time metrics.

1) Experiment Description: In this set-up, we perform two sets of experiments (both based on intersection and architecture shown in Fig. 2); (1) To demonstrate relationship between Density and Flow for all the vehicles, irrespective of their type and corresponding relevance (by relevance we highlight the importance of public transport and emergency vehicles over civil vehicles). For this experiment, we contrast our DRL-optimized junction's performance against the adaptive junction controlled by induction loop. (2) To demonstrate relationship between Density and Speed for emergency vehicles (such as police cars, ambulances and fire trucks) and civil vehicles. For this experiment, we associate with emergency vehicles a higher positive reward of +5/emergency vehicle for passing through the junction and a higher negative reward of -5/emergency vehicle for waiting at the stop-line. The reward design of all other vehicles remain; +1/car passing and -1/car waiting. After training our agent to prioritize the movement of emergency vehicles over civil vehicles, we collected the macroscopic traffic variables to plot the relationship between density versus emergency and civil vehicles' speed.

2) Results: As shown in Fig. 4(a), our DRL-optimized junction (blue) allows more efficient movement of vehicles as compared to adaptive traffic signal control baseline (pink). At the critical density (i.e. the maximum number of vehicles a road segment can effectively accommodate), the cumulative flow of the adaptive baseline system is much lower than the cumulative flow of our DRL-optimized traffic signal control. This establishes our agent's competency in facilitating vehicles' rapid navigation through the intersection.

For the emergency versus civil vehicles experiment, for the given densities, we plot the moving average speed of 100 emergency vehicles versus civil vehicles. It can be inferred from Fig. 4(b), our DRL agent is able to successfully prioritize the navigation of emergency vehicles over civil vehicles.

VI. Summary of Results

Our simulation results show that our DRL agent, equipped with real-time visual traffic data is able to to significantly optimize the movement of vehicles through the road intersection in different traffic situations. To affirm our agent's efficacy, we also derive the relationship between macroscopic traffic variables. We visualized our DCNN to know where exactly it sees while making a decision, but due to brevity

we did not include results here. Our agent significantly outperforms the state-of-the-art fixed and adaptive systems. It is also able to prioritize the movement of emergency vehicles, resulting in their significantly higher speed as compared to the speed of civil vehicles. However, our DRL agent takes a long[1] time to learn to effectively configure the signal regimes. We are currently exploring alternative architectural designs for Traffic3D, alongside transfer learning techniques for DRL [27] to boost its performance.

VII. Conclusion and Future Work

This paper presents an end-to-end trainable DRL agent to autonomously optimize the performance of traffic signal control systems. Our agent is able to effectively perceive the traffic situation in and around an intersection, *solely* using visual sensory data captured in real-time. It continuously modifies the traffic signal regimes, as per changing observations. Compared to the state-of-the-art traffic simulation tools, our experimentation environment is significantly 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 it in the real world.

Going forward, we intend to tackle one of the major challenges in efficient signal control; creating meaningful coordination between multiple signal control systems. Specifically, our future work focuses on creating multiple agents to optimize traffic through collaboration and coordination between multiple intersections, to further improve the quality of traffic signal control systems.

References

[1] M. R. Jabbarpour, H. Zarrabi, R. H. Khokhar, S. Shamshirband, and K.-K. R. Choo, "Applications of computational intelligence in vehicle traffic congestion problem: a survey," *Soft Computing*, vol. 22, no. 7, pp. 2299–2320, 2018.

[2] A. Kouvelas, D. Triantafyllos, and N. Geroliminis, "Hierarchical control design for large-scale urban road traffic networks," in *97th Annual Meeting of the Transportation Research Board (TRB 2018)*. Transportation Research Board (TRB), 2018, pp. 18–04 080.

[3] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19–22, 1992.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[5] D. Garg, M. Chli, and G. Vogiatzis, "Deep reinforcement learning for autonomous traffic light control," in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2018, pp. 214–218.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[7] N. Geroliminis, C. F. Daganzo *et al.*, "Macroscopic modeling of traffic in cities," in *Transportation Research Board 86th Annual Meeting*, no. 07-0413. No. 07-0413, 2007.

[8] J. Luk, "Two traffic-responsive area traffic control methods: Scat and scoot," *Traffic engineering & control*, vol. 25, no. 1, 1984.

[9] V. Mauro and C. Di Taranto, "Utopia," in *Control, computers, communications in transportation*. Elsevier, 1990, pp. 245–252.

[1]Indicatively, it takes 5 days to train our DRL agent for half a million frames while rendering the graphics on a machine with Nvidia GTX-1080 GPU and Intel Xeon(R) Core-16 CPU.

[10] J. Peirce and P. Webb, "Mova control of isolated traffic signals-recent experience," in *Third International Conference on Road Traffic Control, 1990.* IET, 1994, pp. 110–113.

[11] N. H. Gartner, F. J. Pooran, and C. M. Andrews, "Implementation of the opac adaptive control strategy in a traffic signal network," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001, pp. 195–200.

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

[13] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," *arXiv preprint arXiv:1611.01142*, 2016.

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

[15] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intelligent Transport Systems*, vol. 11, no. 7, pp. 417–423, 2017.

[16] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," *arXiv preprint arXiv:1803.11115*, 2018.

[17] T. L. Thorpe and C. W. Anderson, "Traffic light control using sarsa with three state representations," Citeseer, Tech. Rep., 1996.

[18] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility," in *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, vol. 42, 2011.

[19] A. Pell, A. Meingast, and O. Schauer, "Trends in real-time traffic simulation," *Transportation research procedia*, vol. 25, pp. 1477–1484, 2017.

[20] D. Garg, M. Chli, and G. Vogiatzis, "Traffic3d: A new traffic simulation paradigm," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 2354–2356.

[21] ——, "Traffic3d: A rich 3d-traffic environment to train intelligent agents," in *International Conference on Computational Science*. Springer, 2019, pp. 749–755.

[22] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[23] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[24] P. Chakroborty and A. Das, *Principles of transportation engineering*. PHI Learning Pvt. Ltd., 2017.

[25] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[26] A. Warberg, J. Larsen, and R. M. Jørgensen, "Green wave traffic optimization-a survey," 2008.

[27] T. Carr, M. Chli, and G. Vogiatzis, "Domain adaptation for reinforcement learning on the atari," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1859–1861.