Towards Explainable Deep Reinforcement Learning for Traffic Signal Control

Lincoln V. Schreiber [1] **Gabriel de O. Ramos** [1] **Ana L. C. Bazzan** [2]

Abstract

Deep reinforcement learning has shown potential for traffic signal control. However, the lack of explainability has limited its use in real-world conditions. In this work, we present a Deep Q-learning approach, with the SHAP framework, able to explain its policy. Our approach can explain the impact of features on each action, which promotes the understanding of how the agent behaves in the face of different traffic conditions. Furthermore, our approach improved travel time, waiting time, and speed by 21.49%, 27.97%, 20.87%, compared to fixed-time traffic signal controllers.

1. Introduction

With the fast increase in urbanization levels, traffic congestion has become a major problem to society, environment, and economy. According to (Schrank et al., 2019), 2017 data suggest that traffic congestions imposed a cost of 179 million dollars on the U.S. economy. A practical approach to alleviating this problem has been adaptive traffic signal control (ATSC) (Bazzan & Klügl, 2013). The use of ATSC has significant advantages since it allows the reuse of existing cities' infrastructure, thus representing a cost-effective approach as compared to other alternatives. Further, drivers' signaled traffic culture is already widely accepted and understood, making its adoption simpler and faster.

In the literature, many works employ deep neural networks combined with RL techniques (such as deep Q-leaning and deep policy gradient) to optimize decision making (Wei et al., 2019). However, such methods can be seen as black boxes, since the learned policies are not easily understandable or explainable. This lack of explainability represents a major concern in real-world scenarios like traffic (Gunning & Aha, 2019). Nevertheless, currently, only a few works investigate deep RL explainability in traffic signal control (Ault et al., 2019; Rizzo et al., 2019).

The area of explainable AI (XAI) has become widely active in recent years. The idea here is to promote the production of explanatory models, which have high performance but are still explainable (Gunning & Aha, 2019). One framework introduced in recent years is SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017), which is gaining popularity, providing explanations that experts can verify.

The SHAP framework can provide explanations for any model since it is model-agnostic. In particular, SHAP values unify feature attribution models under a single solution. SHAP treats models as a black box, providing explanations based on the model's behavior for different inputs around the data point without going into the model's internal information (such as parameters).

Motivated by this challenge, in this work, we introduce a deep RL approach for traffic signal control that, together with SHAP, enables explaining the learned policy. The idea is to validate the policy behind an agent capable of optimizing the traffic signal. Our agent uses deep Q-learning and is simulated in CityFlow, with a flow of vehicles based on real-world data. To demonstrate that the agent can reduce an intersection's congestion, we compare our agent against three fixed time baselines using different metrics.

The main contributions of this work can be enumerated as follows: (1) A model capable of optimizing traffic in an intersection; (2) A first study towards the use of SHAP to locally explain the impact of features on all possible actions in a given state; (3) An investigation of possibilities and limitations of this kind of explainability in the context of deep RL-based traffic signal control. To the best of our knowledge, this is the first approach to use SHAP to provide local explanations in the context of traffic signal control.

2. Method

2.1. Problem Formulation

Our testing environment consists of a single, isolated intersection with four approaches, as shown in Fig. 1. Here, vehicles are allowed to advance in a straight line and to turn left or right. We model the single RL agent as a fixed-phasing

[1]Graduate Program in Applied Computing, Universidade do Vale do Rio dos Sinos, São Leopoldo, RS, Brazil [2]Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brazil. Correspondence to: Lincoln V. Schreiber <lincolnschreiber@gmail.com>, Gabriel de O. Ramos <gdoramos@unisinos.br>, Ana L. C. Bazzan <bazzan@inf.ufrgs.br>.

Proceedings of the LatinX in AI (LXAI) Research workshop at ICML 2021.

controller with 8 movements. This means the phases order is given and does not change. The agent is then only responsible for defining the *duration* of each phase.

To develop an RL-based traffic signal controller, we characterize the problem as an MDP described as follows. Actions represent the green time duration to be set for the next phase (ranging from 0 to 56 seconds). Hence, the agent selects actions by the end of each phase.

When the agent performs an action, it observes the reward and new state at the next decision point. States are composed of the current phase's queue length, the next phase, and all other phases.The reward defined as the weighted sum (50%) of two features. The first feature represents the total number of waiting vehicles at the intersection. The second feature denotes the difference between the previous and current number of waiting vehicles at current phase. Reward is normalized in the interval $[-1, 1]$.

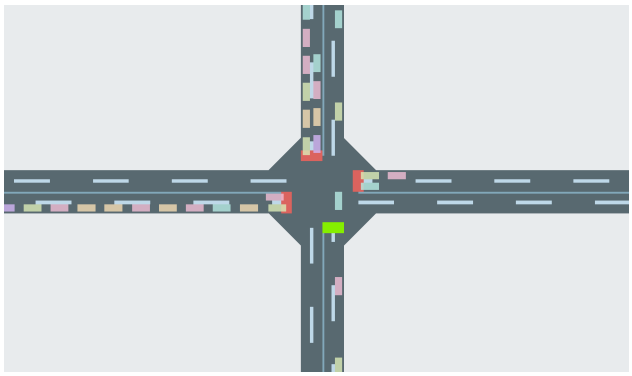


Figure 1. Road network used in our tests.

2.2. Proposed Algorithm

The RL approach used here is Deep Q-Learning (DQN) (Mnih et al., 2015), in which a neural network estimates the agent policy based on Q-values. The neural network receives as input the features that describe the state and provides as output an estimate of each action's Q-value. Based on the network output, the agent chooses the action with the highest Q-value for the given state.

Our neural network architecture consists of six fully connected layers. The input layer has three nodes. The first hidden layer has 128 nodes. The three next hidden layers have 512 nodes each. The output layer has 57 nodes. As for the activation functions, we employ ReLU for all except the output layer, which we employ linear activation function. The network was trained using Adam optimizer with MSE loss function. To improve the algorithm's performance, we used a proportional experience replay scheme (Schaul et al., 2015),where experiences are sampled uniformly at random with a probability that is proportional to their rewards.

3. Experimental Results

3.1. Methodology

We simulated our scenario using CityFlow simulator (Tang et al., 2019). Using a real-world traffic instance from Hangzhou, China (Wei et al., 2019) captured using surveillance cameras (*hangzhou_1x1_bc-tyc_18041608_1h*[1] dataset name). The intersection is four-way, with a speed limit equal to 11.11 meters per second (i.e., 40 kilometers per hour). Each segment is 300 meters long. In this sense, we can estimate the minimum travel time through the entire intersection to be about 54.10 seconds. Also, by definition, each green signal is followed by a 3-second yellow signal and a 2-second all red signal.

To measure and compare the performance of the agent, we chose three different metrics available in the literature (Wei et al., 2019), namely average travel time, average waiting time, and average speed score. In order to better assess performance, our model has been trained ten times.

As a baseline to compare our model's efficiency, we have chosen the FixedTime scheme proposed by (Miller, 1963). Such an scheme has pre-determined cycle and phase time plan. We use the terms FT15, FT30, and FT45 to represent the FixedTime scheme with a duration of 15, 30, and 45 seconds, respectively.

3.2. Optimization and Traffic Control Results

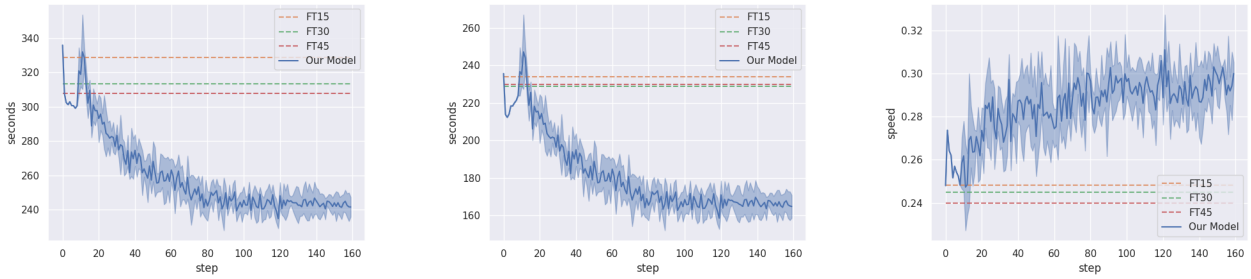
A complete study on buffer size, batch size, gamma, epsilon, number of episodes, and pre-training was performed. Our best model was the one trained for 160 episodes, with a replay buffer with 10240 experience tuples, and batch size of 2048. The discount factor was $\gamma = 0.9$ and the exploration rate was $\epsilon = 1$, with an exponential decay rate 0.97 and a minimum value of 0.01. The pre-training was set with 8 episodes.

The Table 1 presents the overall results for our method and the baselines concerning all metrics. Figures 2a, 2b, and 2c show, respectively, the average travel time, average waiting time, and average speed score along episodes for our method and for the baselines. Since each experiment was repeated 10 times, the figures also show shaded lines to represent the standard deviation.

3.3. Model Explainability

To investigate how each state feature impacts a given state's possible actions, as a case study, we consider the state represented by the features described below. As discussed in Section 2.1, a state is represented by a tuple with three features, which we conveniently refer to as F0, F1, and F2.

[1]Obtained from https://traffic-signal-control.github.io/.



(a) (b) (c)

Figure 2. Average travel time along episodes (a), Average waiting time along episodes (b), Average speed score along episodes (c). The shaded line represents the standard deviation.

Table 1. Results obtained. (1) Average Travel Time; (2) Average Waiting Time; (3) Average Speed Score. The last row presents the average improvement of our method as compared to the baselines.

ALG.	(1)	(2)	(3)
FT15	328.44	233.99	0.2482
FT30	313.38	228.64	0.2448
FT45	307.76	229.75	0.2399
MODEL	**241.62** $\pm$ 5.70	**164.70** $\pm$ 6.07	**0.30** $\pm$ 0.005
IMPROV.	23.60%	28.63%	22.82%

F0 equals 58 (i.e., 58 vehicles are in the queues of the phase for which the agent will currently select the action). F1 equals 6 (i.e., 6 vehicles are waiting on the next phase's queues). F2 equals 19 (i.e., 19 vehicles are waiting on the queues of the remaining phases). Given the state described, the agent decided to take action 43, meaning that the current phase will remain open for 43 seconds.

Fig. 3 shows the impact of each feature (vertical axis) on the possible actions (a local explanation). As seen, the figure has 57 lines with colors representing the different actions, from blue lines denoting actions close to 0 seconds to purple lines representing actions close to 56 seconds. The horizontal axis indicates the contribution level (i.e., the approximate Q-value) of the actions. The Q-value of an action is then based on the impact of these three features. In this sense, the action whose line ends most to the right is the one with the highest Q-values and is chosen by our model (as discussed in previous sections).

To understand how the model can be explained, consider the purple dotted line, which represents the action of 43 seconds. Start looking from bottom to top in Fig. 3. Starting from the base value (which is the value that would be predicted if we did not know any features for the current output (Lundberg & Lee, 2017)), this line is leaning to the right under F0; it means a positive impact on the Q-value.

At F1, the line leans even further to the right. The rationale here is that, as the current phase has much more waiting vehicles (58) than the next (6), it is better to keep the current phase green for a longer time. As a consequence, the reward to be received by the agent tends to be higher. Intuitively, this means that the agent prefers to avoid long queues. For the last feature, F2, the line remains leaning to the right, but a little bit less. Thus, as seen, all features positively impacted an action of 43 seconds, as this would more rapidly decrease the total number of vehicles waiting at the intersection.

Consider the blue dotted line, which represents the action of 8 seconds. Feature F0 negatively impacts this action, since 8 seconds is a small-time window to sufficiently reduce the queue. On the other hand, the action still represents a positive impact on the subsequent phases, as evidenced by the leaning-to-the-right behavior of features F1 and F2. However, by observing this line's behavior as a whole, it becomes clear that this is not a good action in this case.

3.4. Discussion

Following Fig. 3, to optimize this specific case, we would have to choose an action representing a long enough duration to decrease the queues on current phase, but not overly long so that the queues on other phases increases too much. By observing the figure, it becomes clear that this is precisely the behavior followed by our model.

The fact that a given feature has a positive or negative impact on the actions as described above does not mean that it will equally impact all states. In fact, all we can explain is that when we are in a given state, the features have the impact described on the actions based on their values, and with this, we can understand the importance of that feature locally. However, in the present work, we cannot justify an increase or decrease in another state by the value that the feature has in that state.

Even considering that the model is getting it right and obtaining excellent results in the metrics, we can identify outliers in the agent's behavior. The leftmost line in Fig. 3, corre-

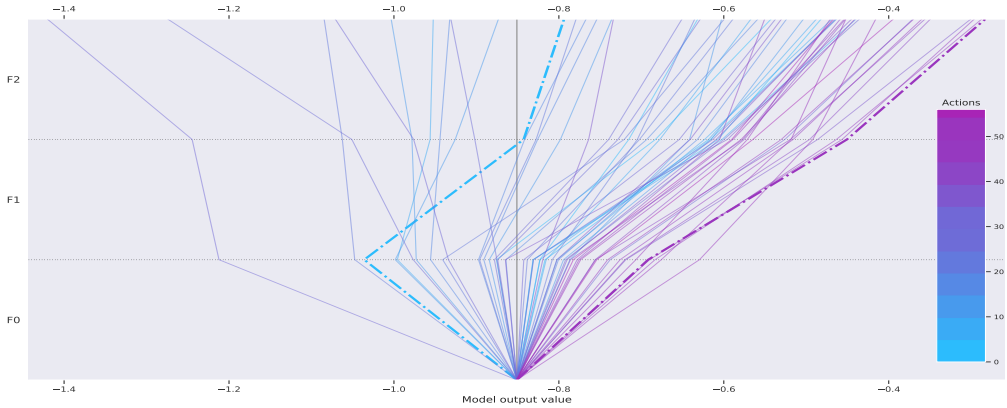


Figure 3. All features' contribution over actions in a state. Interpret the plot from bottom to top.

sponds to the action of 45 seconds, which is close to the action of 43 seconds chosen by the agent.

We present some hypotheses that may explain this behavior: (1) The simulation uses real data to generate vehicles' flow, and it is generated for 3600 seconds. Because of this, we may not observe all combinations of states and actions a sufficient number of times, which may lead to outliers; (2) The stochastic nature of the experience replay algorithm can be a limiting factor for learning, since it often chooses tuples that have already been used several times and not choosing some that have only been through training a few times; (3) The hyper-parameters were optimized empirically to get the best results in the metrics with respect to minimizing traffic congestions. However, these values do not necessarily represent the best option with respect to explainability.

4. Concluding Remarks

In this paper, we proposed a way to explain a reinforcement learning-based agent's policy capable of optimizing traffic at an intersection. Using the advantages of Deep Q-learning, the agent optimized the traffic in a simulated intersection based on real data. It was able to find policies that reduced the average travel time and average waiting time, whereas increasing the vehicles' average speed at the intersection. The simulation results proved that the agent was better than the fixed time baselines.

Building upon our preliminary results, using SHAP, we observed the impact (contribution) of each feature to the agent's actions. Moreover, we could understand the importance of that feature locally, and demonstrate the consistency in the logic of the model, even under the discussed limitations and issues. Our approach can be used to enhance the understanding of policies, thus increasing trustworthiness and safety. In future work, we will investigate improved explanation presentations and test continuous action space.

References

Ault, J., Hanna, J., and Sharon, G. Learning an interpretable traffic signal control policy, 2019.

Bazzan, A. L. and Klügl, F. Introduction to intelligent systems in traffic and transportation, 2013.

Gunning, D. and Aha, D. DARPA's explainable artificial intelligence (XAI) program. *AI Magazine*, 40(2):44–58, 2019.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions, 2017.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning, 2015.

Rizzo, S. G., Vantini, G., and Chawla, S. Reinforcement learning with explainability for traffic signal control. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3567–3572. IEEE, 2019.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay, 2015.

Schrank, D., Eisele, B., and Lomax, T. Urban mobility report 2019. Technical report, Texas A&M Transportation Institute, 2019.

Tang, Z., Naphade, M., Liu, M.-Y., Yang, X., Birchfield, S., Wang, S., Kumar, R., Anastasiu, D., and Hwang, J.-N. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proc. of CVPR-*, pp. 8797–8806, 2019.

Wei, H., Zheng, G., Gayah, V., and Li, Z. A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117*, 2019.