



OPEN

Adaptability and sustainability of machine learning approaches to traffic signal control

Marcin Korecki

This study investigates how adaptable Machine Learning Traffic Signal control methods are to topological variability. We ask how well can these methods generalize to non-Manhattan-like networks with non-uniform distances between intersections. A Machine Learning method that is highly reliable in various topologies is proposed and compared with state-of-the-art alternatives. Lastly, we analyze the sustainability of different traffic signal control methods based on computational efforts required to achieve convergence and perform training and testing. We show that our method achieves an approximately seven-fold improvement in terms of CO₂ emitted in training over the second-best method.

Due to the continuous increase of the global population and rising migration from rural areas into urban environments, there emerges a necessity for the evolution of the concept of the city. As cities become bigger, denser, and more populated the solutions to problems such as traffic control that were previously efficient become ineffective¹. At the same time, new challenges caused by this paradigm shift exacerbate the limitations of the old model of the city. An answer to these issues is the new paradigm of the smart city, where the traditional infrastructure is combined with information and communications technology and coordinated via emerging digital technologies².

One of the key issues of the modern city is its environmental impact on the immediate surroundings as well as its citizens. Urban pollution's detrimental health effects³ as well as its negative impact on perceived happiness⁴ have been well-documented. As such, one of the main goals of the new concept of a city—a smart city is to allow for creation of technologies that can be integrated into the city infrastructure for greater efficiency and sustainability in order to create cities that are not only smart but also sustainable⁵.

A key contributor to the lack of urban sustainability is the common transportation model relying on motorized vehicles. The frequent congestion of traffic causes increased emissions, loss of time, and sound pollution. Moreover, noise and air pollution caused by vehicles are considered key indicators of pedestrian perception of urban spaces⁶. Therefore, improving the current transportation paradigm is of crucial importance for the development of sustainable smart cities⁷.

Some promising approaches addressing the negative effects of traffic are the introduction of electric vehicles and alternative modes of transportation⁸. However, traffic signal control remains the most widespread solution to limiting congestion and increasing the efficiency and sustainability of traffic. A recent trend in the sustainable smart city paradigm is to address traffic congestion with Machine Learning and Internet of Things technologies⁹.

Cities and even more so smart cities are highly dynamic environments, thus they require adaptable approaches. For instance, in order to make the street network more environmentally friendly or to avoid heat island effects¹⁰ the road network might need to be adjusted (for example to introduce more greenery¹¹). Similarly, a recent trend aims at rethinking the streets themselves to, among other things, reduce the number of cars allowed on them¹². Machine Learning shows great promise as a traffic signal control approach, yet the training of advanced models is computationally expensive which often translates into environmental costs¹³. Moreover, most proposed Machine Learning methods are trained on particular city topology and traffic flow and might not readily adapt to temporal and spatial changes. A model requiring retraining when the topology changes or the training of a separate model for different parts of the city would require significant resources and would not be sustainable. Thus a good traffic control model needs to be able to adapt to changes in the road network at least to some extent.

We note that most Machine Learning approaches to traffic signal control focus on idealized Manhattan-like grids which are not representative of the cities in, for example Europe (see Fig. 1)^{14,15}. It is important to investigate how well these models generalize to less idealized conditions of more heterogeneous networks.

Computational Social Science, ETH Zurich, Stampfenbachstrasse 48, 8092 Zurich, Switzerland. email: marcin.korecki@gess.ethz.ch

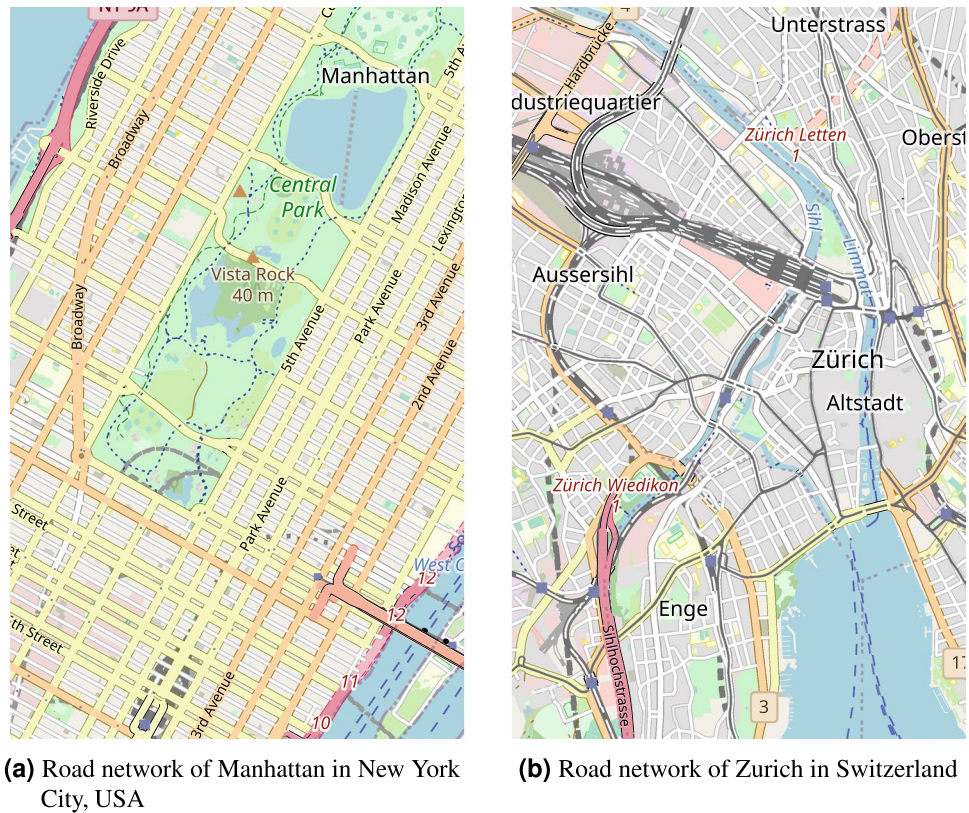


Figure 1. A comparison of the road networks of two large cities, note the difference between the grid like structure of Manhattan and the more chaotic network of Zurich¹⁶.

In this paper we introduce a novel, Machine Learning solution to the problem of traffic signal control, that does not require frequent and costly retraining, and compare it with state-of-the-art alternatives. We show that a clever design of the state space of a Reinforcement Learning (RL) algorithm, along with pretraining the algorithm on a simple scenario yields an adaptable and highly efficient model. In the interest of sustainability, we not only consider the efficiency of the compared methodologies but also their environmental impact. Furthermore, we study how well different traffic signal control methods adapt to changes in the road network topology and compare their results on non-Manhattan-like road networks. We conclude that the proposed method is able to outperform alternative classical and RL approaches in terms of average travel time achieved due to its length-invariant state design (able to account for heterogeneous road lengths) and analytic exploration. Moreover, the method boasts a superior environmental efficiency, do to employing a pretraining procedure.

Background

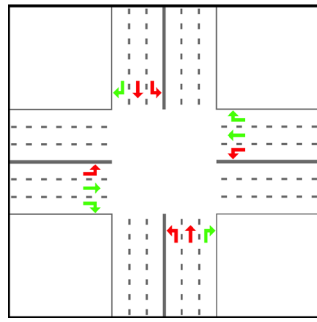
In this section, we formulate the description of the traffic signal control problem as well as define the Reinforcement Learning approach to it.

Glossary of terms.

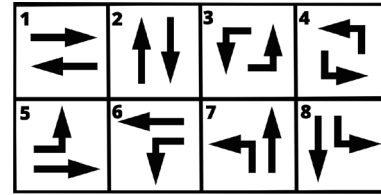
- **Movement:** Represents traffic flow moving through an intersection from an incoming lane to an outgoing lane. The movement from lane a to b is denoted as (a, b) . The intersection in Fig. 2a consists of 12 movements, each indicated with an arrow.
- **Phase:** Represents the set of all movements that are allowed at a given time (given green light). In Fig. 2a, the phase allows for 6 movements indicated in green and corresponding to phase 1 in Fig. 2b.
- **Movement Pressure:** Represents the imbalance between the number of cars of all incoming lanes of the movement and the number of cars on all outgoing lanes of the movement, adjusted for the maximum number of cars possible on the lane¹⁷, according to Eq. (1).

$$w(l, o) = \frac{x(l)}{x_{\max}(l)} - \frac{x(o)}{x_{\max}(o)}. \quad (1)$$

where $w(l, o)$ is the pressure of movement (l, o) , $x(l)$ represents the number of cars on incoming lane l , $x(o)$ the number of cars on the outgoing lane o and $x_{\max}(l)$ and $x_{\max}(o)$ corresponds to the maximum number of cars possible on lanes l and o .



(a) A representation of an intersection with 12 possible movements indicated by the arrows, and 8 possible phases. The green arrows indicate currently allowed movements, which correspond to phase 1 from Figure 2b.



(b) 8 possible phases for the intersection in Figure 2a. Right turn is considered to be allowed in all phases as it does not conflict with any movement. Note that only non conflicting movements can be allowed in a given phase.

Figure 2. A representation of an intersection with 12 possible movements indicated by the arrows, and 8 possible phases. The green arrows indicate currently allowed movements, which correspond to phase 1 from Fig. 2b. 8 possible phases for the intersection in Fig. 2a. Right turn is considered to be allowed in all phases as it does not conflict with any movement. Note that only non conflicting movements can be allowed in a given phase.

- **Intersection pressure:** represents the imbalance of cars on outgoing and incoming lanes across all the possible movements of the intersection, following Eq. (2).

$$P_i = \left| \sum_{(l,o) \in i} w(l,o) \right|, \quad (2)$$

Here P_i is the pressure of the intersection i and the rest follows the notation from Eq. (1).

Reinforcement learning for traffic signal control. Arguably, the most common Machine Learning approach to traffic signal control is Reinforcement Learning¹⁸. Each intersection of a given road network is associated with a Reinforcement Learning agent which observes the state of the intersection. The data representing the observed state at the particular decision step is fed to the agent, which then selects an action, that is the phase to be set. The goal of the agent is to select the actions so that some reward function is maximized. Depending on the design of the agent, the state and reward definitions might differ significantly leading to different results. A detailed review of Reinforcement Learning approaches to the traffic control problem can be found in¹⁹.

Related literature

In this section, we present some of the relevant literature on the topic of traffic signal control.

Conventional. The traditional method of controlling traffic lights in cities is to employ cyclic traffic plans, rotating over the phases²⁰. In more advanced versions of this approach, the cycles are enacted with offsets and the green times given to each phase may vary²⁰. An improvement over the cyclical plans is the SCOOT system (split, cycle and offset optimization technique), which is a data-driven method that is able to adapt to changing and unusual conditions²¹. Further improvements around adaptive methods included prioritization of public transport and privileged vehicles²². At the beginning of the 21st century the conventional methods have begun to be replaced by more intelligent and real-time measurement based solutions. An example of such a method would be the hierarchical control architecture—RHODES that decomposed the control problem into several sub-problems and included a module for prediction of traffic flow²³.

Due to the high variability of urban traffic conventional approaches are usually ineffective in avoiding the creation of traffic jams and have been consistently outperformed by the methods related below.

Self-organizing. The goal of the self-organizing approaches to traffic signal control is to achieve emergent optimization by allowing individual agents to control their behavior in a decentralized manner²⁴. The control mechanism is often designed so that negative interactions between agents are avoided and cooperation is maximized.

One self-organizing approach relies on an analytic model of traffic derived from queuing theory²⁵ and base decision making algorithm on two rules: an optimization and stabilization rule. The optimization rule relies on a short-term anticipation of future arrivals and calculating the minimum green time needed to clear a queue. The stabilization rule makes sure the algorithm avoids situations where a large queue overflows to the previous

intersection. A control scheme based on this method has been successfully deployed in Dresden, Germany, and Lucerne, Switzerland^{26,27}.

Reinforcement learning. Many recent traffic signal control algorithms rely on Deep Reinforcement Learning¹⁸. They differ in their state and reward description as well as the particular type of learning algorithm used. The main learning algorithms in Reinforcement Learning are value-function learning and policy gradient and they have both been applied to traffic signal control²⁸. An example of the value-function learning would be any algorithm that relies on a DQN (Deep Q-network), an example of a policy gradient method would be DDPG (Deep Deterministic Policy Gradient). The key difference between the two is that DQN outputs discrete actions, while DDPG is able to output continuous actions. Furthermore, DQN requires training of one neural network while DDPG, being an actor-critic algorithm requires at least two networks to be trained (the actor network and the critic network and often two additional target networks).

The design of the reward function has a crucial effect on the efficiency of the algorithm, which has been studied in some depth²⁹. One of the approaches to the reward design is to use the concept of pressure (Eq. 2)¹⁴. Similarly, the design of the state space is of great importance²⁹, depending on its design the algorithm might be limited to work only for intersections with a particular number of movements³⁰. An example of state description is to use the queue length, the number of vehicles, the waiting times and an image representation of the intersection³¹. Some research has also investigated how the state description can be structured, including prior knowledge to achieve better learning outcomes³⁰. An important distinction is whether the agents operate independently, or share some information in a centralized manner. In³², the authors compare different decentralized algorithms such as traditional Q-Learning and advantage actor critic in the context of traffic signal control. Some publications have highlighted the benefits of agents sharing information with neighbors³³ or sharing the neural network parameters in training and deployment¹⁴. One recently proposed method, GuidedLight, aimed to combine the benefits of Reinforcement Learning with analytic approach by injecting analytic knowledge into the exploration procedure of the Reinforcement Learning agent¹⁵.

An overview of the recent state-of-the-art Reinforcement Learning methods can be found in^{34,35}. These Reinforcement Learning approaches show great promise and convincing results in simulated homogeneous, grid-like environments. It is not clear, however, how well these models can adapt to changes in the topology of the road-network or how well they perform in non-Manhattan like settings.

Meta-reinforcement learning. A limitation of classical Deep Reinforcement Learning is that the learned models, while achieving good results in the scenarios they were trained on, tend to lack the ability to generalize to different scenarios. In a sense the neural networks at the core of Deep RL can overfit to the data that is provided to them in training. When they later, in deployment, encounter data that they have not been exposed to in training they might recommend actions that are not optimal. The main goal of task generalization is to design learning models that are able to generalize to unseen environments³⁶. Since in RL it has been a standard practice to train and test the models on the same environment it is not clear what is the models' (including the ones in traffic signal control) ability to generalize³⁷. Training and testing in different environments is one of the ways of addressing the issue and so we employ it in our experiments.

Another method that has been designed to address the task generalization is Meta-Learning. It deals with methods that can generalize well to an array of scenarios³⁸. It is worth noting that Deep Q-Learning (or value-function learning) has been shown to possess some Meta-Learning qualities if given access to a context variable that represents the past trajectory³⁹. Approaches that show increased adaptability to changing traffic flows have already been proposed. One way is to use a gradient-based model agnostic meta-learning⁴⁰ for periodically alternating global and individual adaptation and leveraging the experience gotten from learning different scenarios⁴¹. In a similar manner, better generalization can be achieved by leveraging model agnostic meta-learning and flow clustering⁴². A recent study has implemented a meta learning model-based framework to address the need to improve data efficiency and limit the number of interactions with the environment that are needed to learn an efficient model⁴³. Another approach to meta-learning has focused on incorporating both short and long term information in the learning process through the use of decentralized advantage actor-critic approach⁴⁴. On the other hand, a solution based on a meta learning spatial-temporal graph attention network along with a long short-term memory unit has been proposed to deal with neighbor information processing⁴⁵. Most of the meta-learning methods proposed in recent years can be characterized by a rather high complexity of the proposed models. The methods differ in the exact area they focus on (neighbor information, data efficiency etc.) as well as in the way they could be implemented in real world (centralized, decentralized). However, little attention has been given to adaptation to topological variability of the scenarios, which we aim to address in this paper.

Methods

In this section, we will describe in detail the implementation of the proposed agent, that controls the traffic signal at an intersection, and the experimental setup used to showcase and compare its results. As the basis for our proposed agent, we use the GuidedLight agent, which has been shown to perform favorably when compared to other Machine Learning methods as well as analytic self-organizing methods¹⁵. We have decided to focus on Deep Q-learning methods as they have been shown to achieve comparable results to other methods such as DDPG^{46,47}. Furthermore, DQN requires training of fewer deep neural networks as compared to DDPG (which needs an actor network, a critic network and often two additional target networks), since we are interested in decreasing the environmental impact of training the models it makes sense to focus on models which require less computational time to be trained.

We will introduce the features of the proposed agent's state, exploration and learning algorithm design. We will then introduce the novel pretraining procedure and summarize the proposed experimental design.

Length invariant state design. The usual state design of the Reinforcement Learning agent for traffic signal control includes the observations available to a single intersection. This most often consists of the number of cars in the incoming lanes, as well as the number of cars on the outgoing lanes along with the current phase of the intersection. Since the aim is to create an agent which can adapt well to a variety of different road network topologies, we wish to make the state as invariant to topological heterogeneity as possible.

Our aim is to make the agent perform well not only on intersections where the length of incoming and outgoing lanes is the same but also on ones where these lengths vary. In real-life conditions, especially in European cities (Fig. 1b), it is common that the distances and so lengths of roads between intersections are not uniform. Because of that, we propose to express the number of cars on in/outgoing lanes, not as an absolute number but rather the proportion of the lane that is occupied. A 50-m long lane with five 5-m cars on it would be covered in 50%, on the other hand, a 25-m long lane with the same cars would be 100% covered. Even though both lanes have the same number of cars waiting on them it makes more sense to prioritize the one that is full to avoid an overflow. Our agent needs to distinguish between such cases so that it can work well in situations where the lanes have non-uniform lengths. Hence, we introduce the length invariant state design where the occupancy of the lane is expressed in terms of coverage and not simply the absolute number of cars on it.

Analytic exploration. Like all Reinforcement Learning approaches, GuidedLight agent needs to explore its environment before the model becomes sufficiently effective. Exploration is more likely to happen in the early stages of training and the chance that it occurs decreases with time. Instead of exploring randomly (choosing random actions instead of actions suggested by the model), GuidedLight uses analytic exploration (α -exploration). Rather than taking random actions at times, it takes actions that would be selected in a given state by an analytic algorithm²⁵. This approach not only speeds up convergence but also leads to better results.

Deep Q-learning. GuidedLight agent implements the Deep Q-Learning approach. The learning process is based on a Q-function mapping state and action pairs to the expected reward. This function is learned using function approximation capabilities of a deep neural network. The mapping is based on the expected future reward as in Eq. (3).

$$Q^{new}(S_t, A_t) = Q(S_t, A_t) + \ell \cdot (R_t + \gamma \cdot \max_A Q(S_{t+1}, A) - Q(S_t, A_t)) \quad (3)$$

Here Q^{new} is the Q-value after an update for the state-action pair (S_t, A_t at time t), Q is the previous Q-value for the state-action pair, ℓ represents the learning rate and γ weights the effect of short-term vs. long-term gains. R_t is the reward at time t and $\max_A Q(S_{t+1}, A)$ is the estimate of the highest obtainable Q value if one starts from state S_{t+1} and selects optimal actions. In Deep Q-Learning, the Q-function is approximated using a Deep Q Network (DQN). The GuidedLight Agent implements the Double Deep Q-Network (DDQN) approach for greater learning stability⁴⁸.

GuidedLight agent. The agent controls the phase of a single intersection in the traffic network. GuidedLight design is as follows.

- **State:** consists of the percentage coverage of each of three equal segments of the incoming lanes, the coverage of the outgoing lanes and the current phase (one-hot-encoded). The separation of incoming lanes into segments was shown to improve results¹⁴.
- **Actions:** consist of all possible phases at the given intersection.
- **Reward:** based on the concept of pressure of the intersection as in Eq. (2) and following Eq. (4).

$$R_i = -P_i \quad (4)$$

where R is the reward of intersection i . By maximizing the negative of its pressure the agent balances the number of cars between its incoming and outgoing lanes which leads to maximization of throughput and minimization of average travel time¹⁷. An individual GuidedLight agent is deployed at each intersection, however, the agents share the same DQN and store and replay the memory from a shared buffer. This data sharing speeds up the convergence and leads to better results³⁰. Thus in a scenario with 16 intersections there would be 16 agents deployed. These agents, however, would share the same DQN and would add their experience to the same memory replay that would then be available to all other agents. The further details of the implementation of the GuidedLight agent are found in the following sections.

Pretraining. To study GuidedLight's generalization ability we train the model on a synthetic, uniform, Manhattan-like road network. The network is shown in Fig. 3a and the flow is the same as for scenario I (Table 1). The trained model is then tested on the non-uniform and more complex scenarios with different topologies and different traffic flows. The model only learns on the simple network and no further learning occurs in subsequent tests. This way we can assess GuidedLight's resilience to changing conditions.

Configuration	Arrival rate (vehicles/s)	Demand pattern (arrival rate variance)	Average road length (m)
I	0.388	0.3	447.23 ± 115.82
II	0.388	0.6	447.23 ± 115.82
III	0.416	0.3	447.23 ± 115.82
IV	0.416	0.6	447.23 ± 115.82
NY16	1.89	0.00008	397.42 ± 133.00
NY196	0.803	0.0011	425.79 ± 108.72

Table 1. Traffic flows and road lengths assumed in the simulation scenarios.

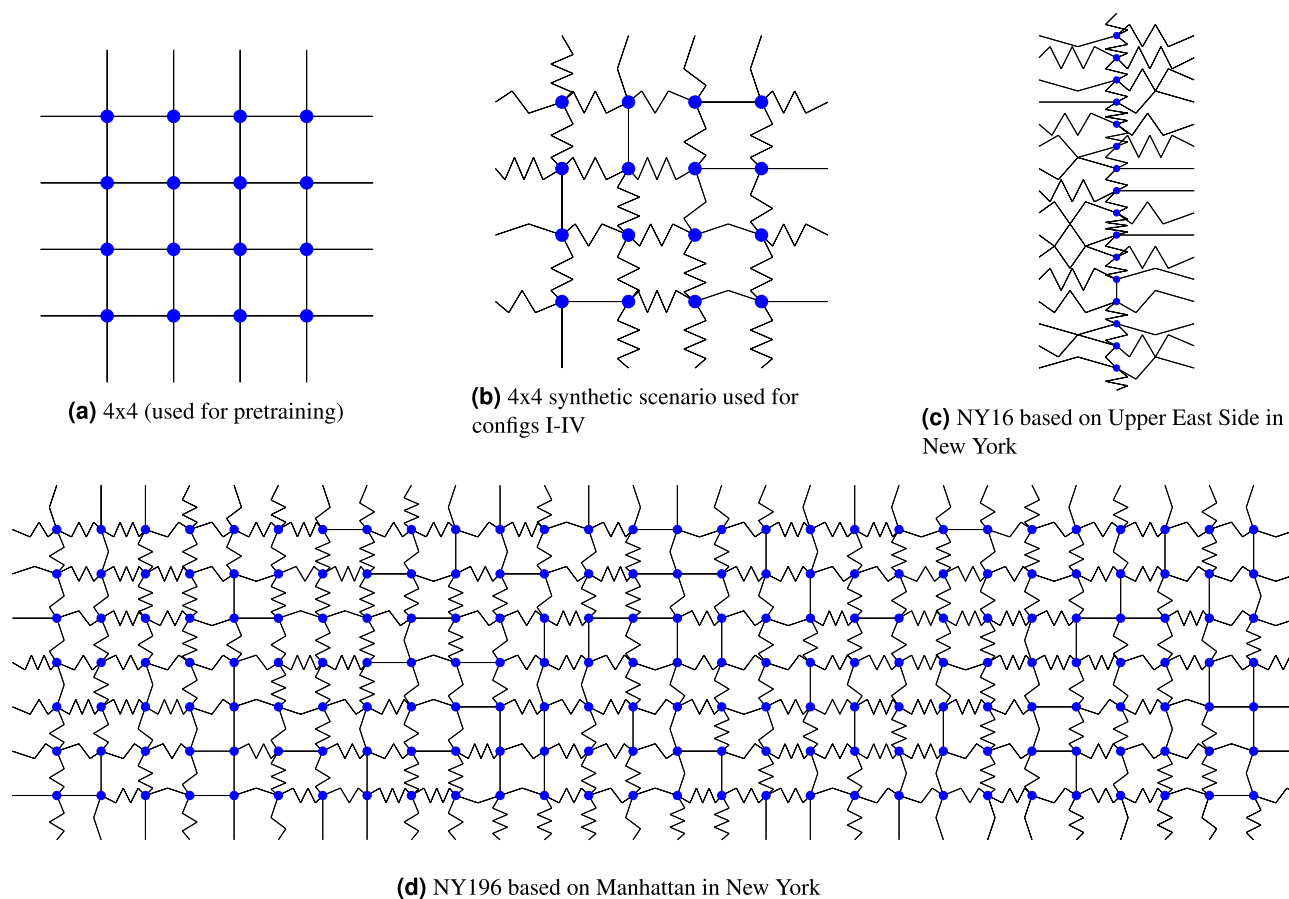


Figure 3. The four road networks used in the experiments, blue dots indicate intersections, black lines indicate roads. Note the varied lengths of the roads in (b–d).

Experiments. To test the GuidedLight agent's efficiency and performance we conduct a series of experiments. First, we compare the average travel time, which is considered to be a good indicator of overall performance^{14,31,41,42}, achieved by GuidedLight on six different simulated scenarios with the results of state-of-the-art alternatives, which we have chosen based on their good performance. We follow with an ablation study where we compare GuidedLight's performance on our most complex scenario—NY196—against versions of GuidedLight that do not use analytic exploration and/or the improved state design. Lastly, we conduct a comparison of the training times of each of the compared methods with their results on the NY196 scenario as well as convergence analysis for the learning algorithms. The non-pretrained, learning methods are trained on each scenario separately and the results of the best model tested on the corresponding scenario is reported.

The details of the road networks used for the experiments and their flows can be found in Table 1. A visualization of the road networks' topologies is presented in Fig. 3. The regular topologies of the networks have been disrupted by elongating the roads at random leading to a situation, where a single intersection has non-uniform lanes' lengths.

The NY16 scenario follows real-world traffic flow data from Upper East Side¹⁴ but with a modified network topology. Similarly, NY196 follows real-world data from Manhattan³³ with a modified topology of the network. We are especially interesting to test how the compared methods scale with the increase in the number of

Model	Avg. travel time (s)					
	I	II	III	IV	NY16	NY196
Fixed time	547.40	480.45	547.05	491.20	511.12	774.37
Random	502.17	466.43	509.30	485.51	424.41	765.45
Analytic	285.90	287.31	291.07	305.48	261.82	669.00
SimpleDQN	619.09	607.85	661.13	636.20	609.5	–
	(10.70)	(11.13)	(13.63)	(8.52)	(9.25)	
FRAP	560.82	520.75	538.33	526.14	534.84	–
	(22.61)	(15.10)	(37.22)	(22.36)	(16.48)	
PressLight	288.65	278.36	296.96	297.42	<u>243.84</u>	706.36
	(1.87)	(2.08)	(2.66)	(2.15)	(4.31)	(3.93)
Pretrained PressLight	289.77	280.43	300.17	314.26	287.98	697.71
GuidedLight	<u>274.36</u>	<u>270.03</u>	<u>289.25</u>	<u>293.46</u>	255.95	<u>660.58</u>
	(1.78)	(3.06)	(1.98)	(2.96)	(2.48)	(5.77)
Pretrained GuidedLight	271.44	265.55	277.07	285.84	<u>246.71</u>	647.45

Table 2. Average travel time of different traffic signal control methods for 4 configurations of 4×4 synthetic scenario and two real-world based scenarios. The values in brackets are standard deviations of the last 10 training epochs. Pretrained methods do not include standard deviations as they are trained and tested on different scenarios. Best results are presented in bold, second best are underscored. The NY196 scenario has not been run with FRAP and Simple DQN as the implementation of these agents is prohibitively slow and energy consuming for large scenarios⁵⁰.

intersection. Hence the NY196 scenario is of most interest to us as it represents a scale at which these methods could be deployed in real life.

The learning agents in this study take decisions every 10 s. When changing phases, 2-s all-red signal (allowing only right turns) is emitted. All scenarios are run for 1800 simulation steps corresponding to 30 min of real-time. Cityflow simulator⁴⁹ is used for all experiments. The learning agents are trained for 150 learning epochs, where one epoch is a full run of the 1800 simulation steps. The DQN used for GuidedLight and PressLight has two layers of 128 and 64 nodes, learning rate of 0.0005 with Adam optimizer is used for training. The training uses mini-batches of 64 samples, the memory is limited to 10000 data points and γ is set to 0.8. The parameters have been selected through grid search. The Simple DQN and FRAP agents use the parameters from⁵⁰.

Compared methods.

- **FixedTime:** static method, rotating through all possible phases one after another, included for comparison²⁰.
- **Random:** an agent choosing actions at random, a safety baseline below which no learning algorithm should fall.
- **Analytic:** an advanced analytic method based on short term traffic anticipation and self-organization, implements an optimization and stabilization rule⁵¹.
- **SimpleDQN:** a simple Reinforcement Learning method³² using a Deep Q-Network. Each agent uses a different network and the parameters are not shared.
- **FRAP:** a state-of-the-art Reinforcement Learning method³⁰, that uses state design invariant to symmetrical cases, thus reportedly being able to generalize to more diverse situations.
- **PressLight:** a state-of-the-art Reinforcement Learning method¹⁴. It uses a regular DQN (not the double DQN), random exploration., and absolute number of cars on incoming and outgoing lanes in the state.
- **Pretrained PressLight:** a PressLight model trained on the synthetic 4×4 scenario.
- **GuidedLight:** a Reinforcement Learning method described above, relying on analytic exploration and length invariant state description. Key differences between PressLight and GuidedLight are the exploration procedure, state design and the use of DDQN instead a DQN.
- **Pretrained GuidedLight:** a GuidedLight model trained on the synthetic 4×4 scenario.

The code used for all experiments and agents (except Simple DQN and FRAP which are available <https://github.com/mKafouros/PlanLight>) is available at https://github.com/mbkorecki/rl_traffic this repository.

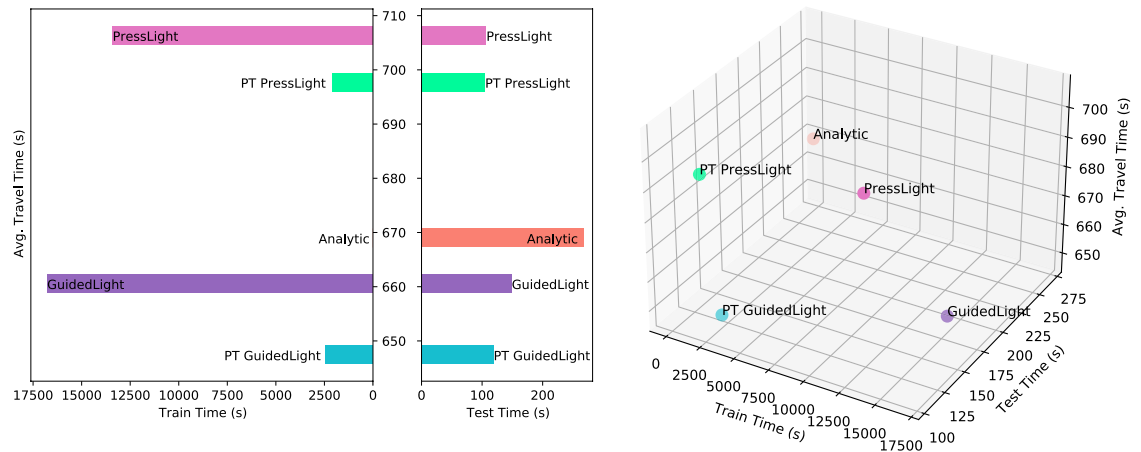
Results

In this section, we present the results of the experiments outlined in the previous section.

Overall performance. In Table 2 we present the results of the tested methods on the 6 experimental scenarios in terms of the average travel time achieved by the methods. The table shows that the best results for most settings are achieved by the Pretrained GuidedLight model. Only in the NY16 scenario does the result of the Pretrained GuidedLight share first place with PressLight. This might be caused by the fact that for the NY16 scenario each intersection has at most two neighbors. Due to that, the problem of overflow, which GuidedLight

Model	Avg. travel time (s)	
GuidedLight - α -exploration -% state	688.8	(9.33)
GuidedLight - α -exploration	682.97	(5.23)
GuidedLight -% state	684.66	(9.34)
GuidedLight	660.58	(5.77)

Table 3. Results of the ablation study on the NY196 scenario. Bold Indicates result with the lowest average travel time. -% state indicates a model using absolute number of vehicles instead of road length adjusted coverage, - α -exploration indicates a model using random exploration. The values in brackets indicate the standard deviations of the last 10 training epochs.



(a) A bar chart of the test and training times across average travel times for different methods. Note the difference in scale between the train and test times.

(b) A 3d projection of the different models in the train-test-travel times space.

Figure 4. Training and test times vs. the avg. travel time on the NY196 scenario. PT stands for pretrained.

addresses with its length invariant state design, is less pronounced as the negative interactions between neighbors are less likely. For all settings except for NY16 GuidedLight is the second best. It is also worth noting that the analytic algorithm's performance is exceptionally high and in most scenarios within 1–2% of the best result. Furthermore, we note that in 5 out of 6 scenarios the Pretrained PressLight achieves worse results than PressLight, only in NY196 is the Pretrained version better, but by a very small margin. The situation is different with GuidedLight, where the pretrained version is significantly superior to the non-pretrained version in all scenarios. Moreover, it is clear that the NY196 scenario, being the most complex one, shows the biggest differences between the algorithms. It is also apparent that two of the ML methods—SimpleDQN and FRAP—perform surprisingly bad on the tested scenarios. In fact the performance of both these methods is worse than the performance of the Random and Fixed Time methods. Thus, we do not include these methods in the training, test time and convergence experiments.

Ablation study. In Table 3, we report the results of the ablation study. We can see that the removal of the length invariant state description (GuidedLight-% state) leads to significantly worse results. The same is true of the removal of α -exploration. Finally, the removal of both of these features leads to the worst results. Interestingly, it seems that the removal of the length invariant state description also increases the standard deviation of the last 10 training epochs which can be interpreted as the training process being less stable. It is worth noting that GuidedLight without % state and α -exploration is still better than PressLight, showcasing the benefits of using Double Deep Q-Network.

Training/test times and convergence. In Fig. 4, we report the test and train times for the compared models as well as their results in terms of average travel time. All models were run on a cluster using AMD EPYC 7742 CPU and requiring at least 8 cores, 2 GeForce RTX 2080Ti GPUs and 2048 MB of RAM. As can be seen, the models have varied train time with GuidedLight requiring most training time and analytic requiring no training at all. On the other hand, at test time the analytic algorithm is most time-consuming.

It is important to note that the main source of the computational strain for the GuidedLight and the analytic algorithm is the calculation of the rates of arrival and departure. Due to the design of the simulation, computing these values requires some costly operations. In deployment, such values can be easily detected using inductive loops⁵². Thus the computational costs in the deployment of GuidedLight would be much closer, if not the same, as PressLight's. Similarly, the analytic test time cost would be lower in deployment.

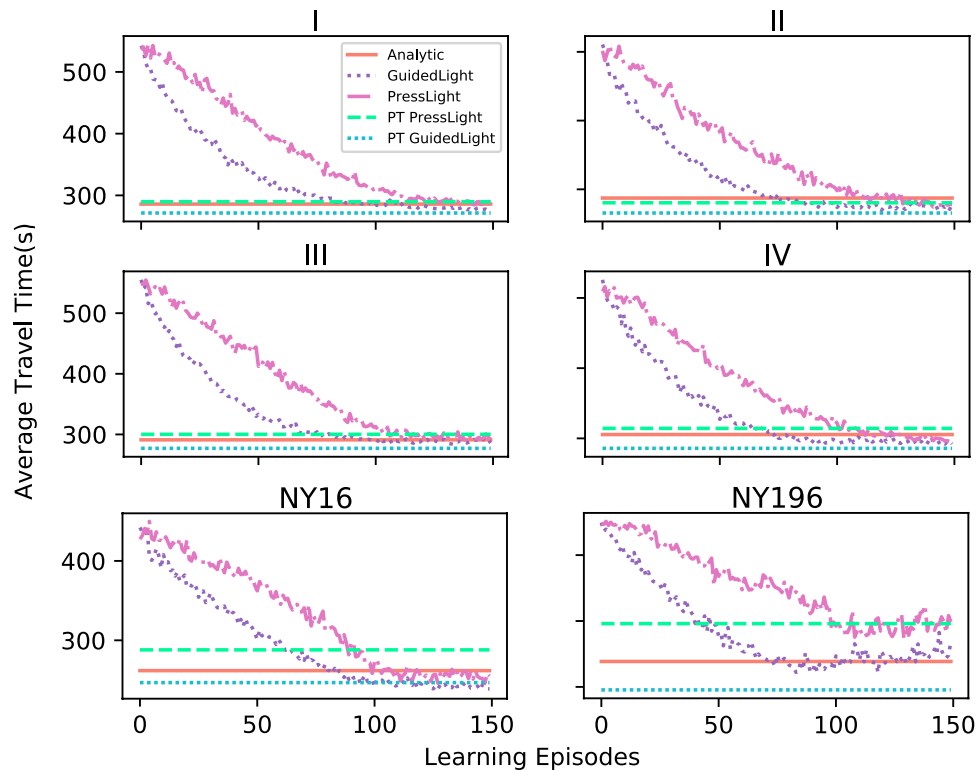


Figure 5. Average travel time achieved through learning epochs for PressLight and GuidedLight along with the results achieved by the non-trained methods: analytic and two pretrained (PT) models.

In Fig. 5, we report the convergence for training learning algorithms along with the results of the pretrained or non-learning approaches. We can see that for the synthetic scenarios the final results are similar for all compared methods. The largest differences occur in the NY196 scenario. It is interesting to note that the performance of the analytic algorithm in the NY196 is very high and that the GuidedLight trained on this scenario takes a around 80 epochs to converge to the analytic results and even the best model barely surpasses the analytic results. Lastly, it can be seen that GuidedLight converges faster and towards better results than PressLight.

Discussion and conclusion

In this paper, we proposed a Machine Learning agent for traffic signal control that achieves, in the majority of the scenarios studied, better results than both state-of-the-art Machine Learning and analytic alternatives. Moreover, we have shown that our agent shows high levels of topological adaptability. Namely, a model pretrained on a simple setting with uniform distances between intersections achieved results surpassing all compared alternatives when run on more complex scenarios with non-uniform distances between intersections. We showed that PressLight does not appear to adapt well to changes in topology, since the pretrained version of PressLight performed worse than PressLight. At the same time pretrained GuidedLight was shown to perform better than the non-pretrained version showcasing its topological adaptability. The success of the pretraining procedure in case of GuidedLight can be explained by its length invariant state design, which allows it to generalize its experience to different topological conditions. On the other hand PressLight is not able to perform such generalization as it will experience completely different states in different topological conditions. Our ablation study further confirmed the power of proposed improvements, both the length invariant state as well as the α -exploration approach. We also showcased the faster convergence of the GuidedLight agent over PressLight, which is due to the effects of α -exploration.

Moreover, we compared the training and test times of the studied methods. These values can be understood as indicators of the environmental impact of the methods as computational time translates into energy expenditure¹³. Using a Machine Learning CO₂ calculator⁵³ we can perform some high level approximation of the actual amount of carbon emitted during training. We assume the world average value of carbon efficiency of 475 g/kWh from 2019⁵⁴. Under these assumptions the training of GuidedLight on the NY196 scenario would emit 0.56 kg of CO₂ which is equivalent to burning 0.28 kg of coal or driving 2.26 km with an average car. On the other hand using the pretrained GuidedLight would yield only 0.08 kg of CO₂ corresponding to burning 0.04 kg of coal or driving 0.32 km with an average car. That is 7 times less CO₂ when using the pretrained model, a significant improvement. Furthermore the analytic solution would emit no CO₂ altogether as it does not require training. It is worth noting that depending on the country where the computational power is located and the potential offsets employed these values may vary significantly. These calculations stress the importance of limiting the

number of times the models need to be retrained and furthermore showcase how one can decrease the training costs by employing methods such as pretraining on a simpler scenario.

We noted the high training and test costs of GuidedLight but also noticed that the Pretrained GuidedLight's testing and training times are very low. Thus the proposed pretraining procedure not only allows for great adaptability but also increases the environmental friendliness of the method. Why train the model on a highly complex setting if it can adapt to it better after being trained on a simple, less demanding scenario. It is also important to note that the analytic method while performing worse than GuidedLight does not need any training. In some cases, it might be more beneficial and sustainable to deploy such a method even if there is a slight drop in efficiency (in terms of avg. travel time for instance).

We also note the apparent lack of effectiveness of the SimpleDQN and FRAP methods, which, while being consistent with the results in other work⁵⁰ (where both these methods perform worse than Fixed Time in real-world scenarios), is still surprising. This low performance is likely due to the methods focusing on fully individual agents, with no parameter sharing or communications. Such agents require much more training data to learn optimal behavior. These findings reflect the need for strong baselines that can be used to judge the performance of ML methods. In a similar vein it reflects the need of varied testing scenarios. For example FRAP has been shown to perform better than some baselines in some simple scenarios³⁰, but in more complex, realistic scenarios it appears to falter, as shown in this work.

With this work we hope to contribute to the trend of sustainability-conscious Machine Learning, which has recently seen growing interest. While we see the great potential of the Machine Learning for the particular problem of traffic signal control, exemplified by our results, we also believe it important to quantify and weight the environmental costs of such methods. Future directions for study include further investigating the pretraining procedure and identifying what properties of the smaller network used for pretraining lead to high adaptability of the model as well as considering models that are adaptable to disruptions in both topology as well as traffic flows.

In summary, we have presented and compared the topological adaptability and sustainability of some Machine Learning and analytic methods. We have shown the value of pretraining and intelligent state design as well as the benefits of injecting analytic knowledge into the Machine Learning models via exploration. Lastly, we were able to propose an adaptive and sustainable Machine Learning method—the Pretrained GuidedLight which boasts superior performance to the alternatives.

Data availability

The datasets generated and analyzed during this study are available from the corresponding author on reasonable request. The code used to run the experiments is available https://github.com/mbkorecki/rl_trafficere.

Received: 21 October 2021; Accepted: 21 September 2022

Published online: 06 October 2022

References

- Pellicer, S. *et al.* A global perspective of smart cities: A survey. In *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 439–444 (2013).
- Batty, M. *et al.* Smart cities of the future. *Eur. Phys. J. Spec. Top.* **214**, 481–518 (2012).
- Ali, M. U. *et al.* A systematic review on global pollution status of particulate matter-associated potential toxic elements and health perspectives in urban environment. *Environ. Geochem. Health* **41**, 1131–1162 (2019).
- Zheng, S., Wang, J., Sun, C., Zhang, X. & Kahn, M. E. Air pollution lowers Chinese urbanites' expressed happiness on social media. *Nat. Hum. Behav.* **3**, 237–243 (2019).
- Ahvenniemi, H., Huovila, A., Pinto-Seppä, I. & Airaksinen, M. What are the differences between sustainable and smart cities?. *Cities* **60**, 234–245 (2017).
- Piselli, C., Castaldo, V., Pigliautile, I., Pisello, A. & Cotana, F. Outdoor comfort conditions in urban areas: On citizens' perspective about microclimate mitigation of urban transit areas. *Sustain. Cities Soc.* **39**, 16–36 (2018).
- Silva, B. N., Khan, M. U. & Han, K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustain. Cities Soc.* **38**, 697–713 (2018).
- Hafizyar, M., Arsallan, A., Sabory, N. R., Danish, M. S. S. & Senjyu, T. *Smart and Sustainable Township: An Overview*, 65–80 (2020).
- Din, I. U., Guizani, M., Rodrigues, J. J., Hassan, S. & Korotaeov, V. V. Machine learning in the internet of things: Designed techniques for smart cities. *Futur. Gener. Comput. Syst.* **100**, 826–843 (2019).
- Akbari, H. *et al.* Local climate change and urban heat island mitigation techniques—the state of the art. *J. Civ. Eng. Manage.* **22**, 1–16 (2016).
- Wong, N. H., Tan, C. L., Kolokotsa, D. D. & Takebayashi, H. Greenery as a mitigation and adaptation strategy to urban heat. *Nat. Rev. Earth Environ.* **2**, 166–181 (2021).
- Von Schönfeld, K. C. & Bertolini, L. Urban streets: Epitomes of planning challenges and opportunities at the interface of public space and mobility. *Cities* **68**, 48–55 (2017).
- Green, A. I. *Commun. ACM* **63**, 54–63 (2020).
- Wei, H. *et al.* Presslight: Learning Max pressure control to coordinate traffic signals in arterial network. *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* **1**, 1290–1298 (2019).
- Korecki, M. & Helbing, D. Analytically Guided Reinforcement Learning for Green It and Fluent Traffic. *IEEE Access*, **10**, 96348–96358 <https://doi.org/10.1109/ACCESS.2022.3204057> (2021).
- OpenStreetMap contributors. Planet dump retrieved from <https://planet.osm.org>. howpublished <https://www.openstreetmap.org> (2021).
- Varaiya, P. Max pressure control of a network of signalized intersections. *Transp. Res. Part C Emerg. Technol.* **36**, 177–195 (2013).
- Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT press, 2018).
- Mannion, P., Duggan, J. & Howley, E. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. *Auton. Road Transport Support Syst.* **20**, 47–66 (2016).
- Urbanik, T. *et al.* Signal Timing Manual. NCHRP Report, p. 317 (2015).
- Hunt, P., Robertson, D., Bretherton, R. & Royle, M. C. The scoot on-line traffic signal optimisation technique. *Traffic Eng. Control* **23**, 25 (1982).
- Bell, M. G. Future directions in traffic signal control. *Transp. Res. Part A Policy Pract.* **26**, 303–313 (1992).

23. Mirchandani, P. & Head, L. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transp. Res. Part C Emerg. Technol.* **9**, 415–432 (2001).
24. Gershenson, C. Self-organizing traffic lights. *Complex Syst.* **16**, 29–53 (2004).
25. Lämmer, S. & Helbing, D. Self-control of traffic lights and vehicle flows in urban road networks. *J. Stat. Mech. Theory Exp.* **2008**, 1–36 (2008).
26. Lämmer, S. Selbst-gesteuerte Lichtsignalanlagen im Praxistest. *Straßenverkehrstechnik* **60**, 25 (2016).
27. Genser, A. Wirkungsanalyse selbst-steuerung (2020).
28. Mousavi, S. S., Schukat, M. & Howley, E. Traffic light control using deep policy-gradient and value-function based reinforcement learning (2017).
29. Zheng, G. *et al.* Diagnosing reinforcement learning for traffic signal control, 1 (2019).
30. Zheng, G. *et al.* Learning phase competition for traffic signal control. In *International Conference on Information and Knowledge Management, Proceedings 1963–1972* (2019).
31. Wei, H., Zheng, G., Yao, H. & Li, Z. Intellilight. 2496–2505 (2018).
32. Chu, T., Wang, J., Codecà, L. & Li, Z. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **21**, 1086–1095 (2019).
33. Wei, H. *et al.* Colight: Learning network-level cooperation for traffic signal control. In *International Conference on Information and Knowledge Management, Proceedings 1913–1922* (2019).
34. Wei, H., Zheng, G., Gayah, V. & Li, Z. A survey on traffic signal control methods. [arXiv:1904.08117](https://arxiv.org/abs/1904.08117) (arXiv preprint) (2019).
35. Wei, H., Zheng, G., Gayah, V. & Li, Z. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explor. Newsl.* **22**, 12–18 (2021).
36. Agarwal, R., Machado, M. C., Castro, P. S. & Bellemare, M. G. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. [arXiv:2101.05265](https://arxiv.org/abs/2101.05265) (arXiv preprint) (2021).
37. Cobbe, K., Klimov, O., Hesse, C., Kim, T. & Schulman, J. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, 1282–1289 (PMLR, 2019).
38. Hospedales, T., Antoniou, A., Micaelli, P. & Storkey, A. Meta-learning in neural networks: A survey (2020).
39. Fakoor, R., Chaudhari, P., Soatto, S. & Smola, A. J. Meta-q-learning (2020).
40. Finn, C., Abbeel, P. & Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *34th International Conference on Machine Learning, ICML 2017* **3**, 1856–1868 (2017).
41. Zang, X. *et al.* MetaLight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence* **34**, 1153–1160 (2020).
42. Zhang, H., Liu, C., Zhang, W., Zheng, G. & Yu, Y. GeneralLight: Improving environment generalization of traffic signal control via meta reinforcement learning. In *International Conference on Information and Knowledge Management, Proceedings 1783–1792* (2020).
43. Huang, X., Wu, D., Jenkin, M. & Boulet, B. Modellight: Model-based meta-reinforcement learning for traffic signal control. [arXiv:2111.08067](https://arxiv.org/abs/2111.08067) (arXiv preprint) (2021).
44. Yang, S. & Yang, B. A meta multi-agent reinforcement learning algorithm for multi-intersection traffic signal control. In 2021 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 18–25 (organizationIEEE, 2021).
45. Wang, M. *et al.* Meta-learning based spatial-temporal graph attention network for traffic signal control. *Knowl. Based Syst.* **20**, 109166 (2022).
46. Wu, H. Control method of traffic signal lights based on ddpq reinforcement learning. *J. Phys. Conf. Ser.* **1646**, 012077 (2020).
47. Pang, H. & Gao, W. Deep deterministic policy gradient for traffic signal control of single intersection. In *2019 Chinese Control And Decision Conference (CCDC)*, 5861–5866 (IEEE, 2019).
48. Van Hasselt, H., Guez, A. & Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30 (2016).
49. Zhang, H. *et al.* Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The Web Conference 2019—Proceedings of the World Wide Web Conference, WWW 2019* 3620–3624 (2019).
50. Zhang, H., Kafourous, M. & Yu, Y. Planlight: Learning to optimize traffic signal control with planning and iterative policy improvement. *IEEE Access* **8**, 219244–219255 (2020).
51. Lämmer, S., Donner, R. & Helbing, D. Anticipative control of switched queueing systems. *Eur. Phys. J. B* **63**, 341–347 (2008).
52. Bhaskar, L., Sahai, A., Sinha, D., Varshney, G. & Jain, T. Intelligent traffic light controller using inductive loops for vehicle detection. In *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, 518–522 (2015).
53. Lacoste, A., Luccioni, A., Schmidt, V. & Dandres, T. Quantifying the carbon emissions of machine learning. [arXiv:1910.09700](https://arxiv.org/abs/1910.09700) (arXiv preprint) (2019).
54. International Energy Agency. Global Energy & CO2 Status Report 2019—Analysis.

Acknowledgements

The author would like to express deep gratitude to the members of Computational Social Science group at ETH Zurich led by prof. Dirk Helbing for their support with special thanks to Dr. Vaiva Vasiliauskaite, Dr. Sachit Mahajan, Dr. Carina I. Hausladen, Javier Argota Sánchez-Vaquerizo and Andrea Musso for their invaluable help in reviewing and proofreading this manuscript. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement no. 833168).

Author contributions

M.K. conceived and conducted the experiment, analyzed the results and wrote the manuscript.

Competing interests

The author declares no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022