

## RESEARCH ARTICLE

# Analytically Guided Reinforcement Learning for Green It and Fluent Traffic

**MARCIN KORECKI<sup>1</sup> AND DIRK HELBING<sup>1,2</sup>**<sup>1</sup>ETH Zürich, Computational Social Science, ETH Zürich, 8092 Zürich, Switzerland<sup>2</sup>Complexity Science Hub Vienna, 1080 Vienna, Austria

Corresponding author: Marcin Korecki (marcin.korecki@gess.ethz.ch)

This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Program under Grant 833168.

**ABSTRACT** This study investigates various methods for autonomous traffic signal control. We look into different types of control methods, including fixed time, adaptive, analytic, and reinforcement learning approaches. Machine learning approaches are compared with the “analytic” approach, which is used as “gold standard” for performance assessment. We find that conventional machine learning approaches are better than the analytic approach, but require a lot more computer power. We, therefore, introduce a novel hybrid method called “analytically guided reinforcement learning” or shorter “ $\alpha$ -RL”. This approach is implemented in our “GuidedLight agent” and tends to outperform both, classical machine learning and the analytic approach, while largely improving convergence. This method is therefore suited as a “green IT” solution that improves environmental impact in a two-fold way: by reducing (i) traffic congestion and (ii) the processing power needed for the learning and operation of the traffic light control algorithm.

**INDEX TERMS** Green AI, complex systems, reinforcement learning, smart cities, sustainability, traffic light control.

## I. INTRODUCTION

Traffic congestion is one of the most widespread problems of cities today, leading to losses in productivity, avoidable CO<sub>2</sub> emissions, environmental pollution, and reduced quality of life. Along with the world's population growth and progressive urbanization, these problems are expected to amplify further.


While in the long term, a technological shift to less problematic forms of transportation is likely, traffic congestion will remain a challenge for the foreseeable future.

### A. CONTRIBUTION

Traffic light control is a complex optimization problem, which is NP-hard [1], i.e. not solvable exactly in real-time, if problems get realistically big. Hence, it is required to apply approximation methods. Such methods comprise, among others, fixed (time) scheduling [2], analytic methods [3], [4], [5], [6], [7], [8], adaptive methods [9], [10], [11], [12], [13],

[14], [15], [16], and genetic algorithms [17], [18]. Moreover, there exist methods which assume all or some of the vehicles in the network to be autonomous [19], [20]. Recently, with the spread of powerful machine learning (ML) and artificial intelligence (AI) applications, reinforcement learning (RL) approaches have attracted much interest [21], [22], [23], [24], [25]. However, while the feasibility of the RL approach got much attention, the related issues and limitations have not yet been investigated in full [3], [10]. Furthermore, the potential benefits of “hybrid” approaches, which combine analytic knowledge and RL methods, have not yet been explored in depth. Also, the assessment of the ecological footprint of machine learning approaches has been often neglected. Therefore, the main contribution of this paper is to:

- highlight the performance and limitations of machine learning approaches considering ecological issues,
- propose an improved, hybrid machine learning approach called “analytically guided reinforcement learning” or “ $\alpha$ -RL”, which converges much more quickly than conventional machine learning methods.

The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Ayyash .

In the following sections, we will present the background of the field and current state of the art, focusing on the comparison between adaptive and learning methods. We will also propose an analytic benchmark for machine learning methods. Finally, we will discuss the potential benefits of combining reinforcement learning (RL) and analytic approaches in a hybrid method (“ $\alpha$ -RL”).

## II. BACKGROUND

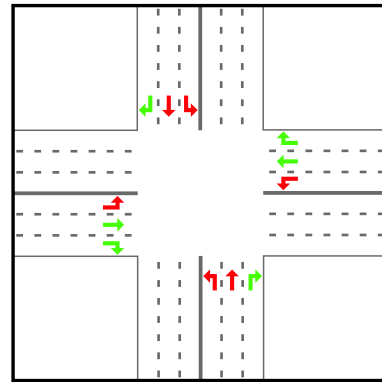
One of the simplest method of traffic signal control is fixed time scheduling [2], which is usually predefined and operated in a periodic way. For the sake of simplicity, it is sometimes furthermore assumed that the same amount of green time is assigned to each phase. This approach is obviously quite limited, but often considered as a baseline to compare the performance of various traffic light control approaches to. An adaptive extension of fixed time scheduling is able to select a traffic plan from a predefined list of plans in response to the respective traffic conditions [26], but the assumption of repetitive service patterns is usually still applied. In contrast, fully adaptive approaches are also possible, which respond to data from induction loops placed before and after intersections, that detect arriving and departing vehicles [27]. Such approaches do not rely on predefined plans, but rather adapt in real-time to the particular local traffic conditions. They, however, often lack coordination among intersections. Recently, a lot of interest has also been paid to employing data-driven machine learning approaches to traffic light control [28].

In the following section, we will introduce the traffic light control problem in more detail and typical solution approaches including fixed time, analytic, and reinforcement learning methods. For a comprehensive survey of different traffic control approaches, we recommend to read, for example, [29].

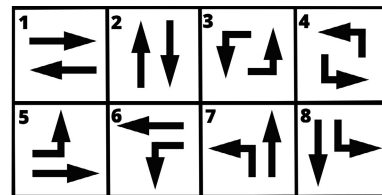
### A. GLOSSARY OF TERMS

Here we provide definitions of the key terms used in formalizing traffic intersections.

- **Approach:** a road crossing other roads at an intersection. There are “incoming approaches”, i.e. the ones through which cars arrive at the intersection, and “outgoing approaches” through which cars depart.
- **Lane:** a single approach can be subdivided into lanes. The lanes on the incoming approach are referred to as “incoming lanes” (short: “in-lanes”), the one on the outgoing approach as “outgoing lanes” (short: “out-lanes”).
- **Movement:** consists of an incoming approach and an outgoing approach, through which vehicles can move from in-lane(s) to out-lane(s). Usually three types of movements are considered: left turns, right turns and moving straight (“through traffic”).
- **Movement signal:** signal indicating whether the given movement is allowed (green) or not (red). The yellow signal indicates the change from green to red and, depending on national law, may allow or block the



**FIGURE 1.** Representation of an intersection with four approaches: North, West, East, South. There are 3 separate lanes on each approach: one for through traffic, one for turning left, and one for turning right. Here, the traffic lights are assumed to be in phase 1 as per the numbering introduced in Figure 2. Green arrows indicate movements that are allowed, while red arrows indicate movements that are disallowed in the current phase.



**FIGURE 2.** Possible phases to be selected from by a control mechanism, here, for intersections with four approaches: North, West, East, South. For all the phases, a right-turn from each approach is also assumed to be possible, when there are no conflicting traffic flows.

movement (in this work we assume the yellow signal allows for movement). A conditional green signal is usually assumed for the right turn, allowing for movement when there is currently no conflicting traffic.

- **Phase:** a combination of movement signals. A phase can only consist of no conflicting movement signals. A movement signal is conflicting, if a related movements crosses another movement.

### B. PROBLEM DESCRIPTION

In this paper we are looking for methods to change traffic lights at intersections such that the resulting traffic performance is as high as possible. To assess the performance, one often studies quantities such as the throughput and average travel time. The methods we are interested in should work for different traffic intensities. They should also work for a large number of intersections with a reasonable computational effort. In this connection, an important distinction to make is whether one attempts to optimize traffic flow locally on the level of single intersections or over extended parts of the entire road network. A network-wide approach requires much more computational resources than an intersection-based approach and is often practically intractable. In this paper, we will focus on local control approaches due to the focus on green IT and for the sake of comparability with previous pub-

lications such as [22], [23]. Note, however, that this does not exclude the possibility of coordination between neighboring intersections.

### III. RELATED WORK

In this section we will discuss relevant related work.

#### A. FIXED TIME CONTROL

A classical method of traffic control is to generate centralized schedules, which are imposed on all intersection in the city [2]. In its simplest form each intersection cycles through all its phases with no off-sets. Each intersection at a given time has the same phases, and each phase is given the same amount of time. We refer to this simplistic method as Fixed Time Control. More advanced versions of this method include the implementation of different green times periods for each phase and suitably calibrated off-sets [2].

#### B. ADAPTIVE METHODS

A typical adaptive method is able to select the next phase based on the current state of the intersection controlled. One of the simplest adaptive methods is “demand-based” control. This approach adapts its actions based on the “demand of a phase”, which is defined as the sum of the demands of all movements belonging to the phase. The “demand of a movement” corresponds to the number of cars that are present on all incoming lanes belonging to the movement.

#### C. SELF-ORGANIZATION

An important aspect in local traffic optimization is the avoidance of negative interactions between neighboring intersections. In general, a decision that is optimal at one intersection may cause sub-optimal traffic flows at neighboring intersections, for example, due to spill-over effects. To address this problem, the concept of self-organized traffic light control has been developed, which promotes a coordination among neighboring intersections [30].

A self-organizing system is a system where its adjacent elements interact in a way that gives rise to a collective behavior. This can be coordinated behavior over the entire system or extended parts of it. If the interactions are well chosen, the resulting self-organized system dynamics can perform extremely well. Therefore, the emphasis is to make the interactions between the individual system elements mutually positive (synergistic). In [10], it is demonstrated that a method, called “Self-Organizing Traffic Light” (SOTL), based on the above concepts, can reach significant improvements even over state-of-the-art methods to produce green waves, which attempt global traffic flow optimization by synchronizing traffic lights and supporting vehicle platoons that rarely need to stop [31].

#### D. ANALYTIC APPROACH

Analytic methods rely on models and formulas derived from a theory (e.g. queuing theory or traffic physics) and focus on

showing that the proposed control scheme locally optimizes the selected performance criterion.

A very effective analytic, adaptive approach, which relies on concepts from traffic physics as well as self-organization principles, has been proposed in [5]. The method consists of two elements: an optimization rule and a stabilization rule. The optimization rule (see Appendix) is based on the short-term anticipation of future arrivals of vehicles to the queue and on calculating the green time needed to clear the expected queue. A priority score is used by the optimization rule to select the movement or phase that needs to be switched to.

The stabilization rule overrides the optimization rule in situations when a queue has grown too large or some phases have not been activated for a long time [6]. This helps to prevent spill-over effects at neighboring intersections.

The short-term anticipation of this analytic approach promotes a self-organized coordination between flows and traffic lights at neighboring intersections. Due to the resulting self-organization, the two rules lead to a spontaneous emergence of green waves, much like in [30]. The method has been successfully implemented in real life settings in the cities of Dresden, Germany, and Lucerne, Switzerland [32], [33]. In the following, for the sake of simplicity, our implementation of the analytic method will use the optimization rule only, while the stabilization rule will be neglected, possibly at the cost of losing some performance. (We will focus on its role in a follow-up study.)

#### E. REINFORCEMENT LEARNING (RL)

Due to the complexity of traffic light optimization, many recent publications have proposed to use machine learning approaches. Instead of deriving analytic models, these propose to use an iterative, neural-network-based learning method, often called a “black box”, which is fed with lots of data. Significant success has been demonstrated by multi-agent deep reinforcement learning models, which we discuss below. We focus on models which, like the previously described approaches, optimize traffic flows locally on the level of a single intersection, mainly for the sake of comparison with previously published results [22], [23].

In the machine learning models, an “agent” represents an intersection of the road network. The agent is fed with data from observations of the environment and takes actions based on them. The agent is also given rewards that reflect the desirability of the actions it had taken [34]. The data included in the observations as well as the choice of the reward function may have a strong influence on the efficiency of the learning process.

In [22], a learning algorithm called “IntelliLight” uses the queue length, number of vehicles, waiting time and an image representation of the intersection as its state.

In [35], an analysis of the reward and state design in reinforcement learning is applied to traffic light control. Moreover, the “LIT” method is proposed to simplify the state description.

In [24], the authors propose “CoLight”, which uses graph attentional networks to facilitate communication between traffic lights. The method considers a spatial and temporal interaction of neighboring agents.

The state representation is studied in depth in [36] and a “FRAP” model is proposed. The model addresses the problem of limited adaptive potential of most learning approaches (e.g. a model trained with morning traffic may not adapt well to evening traffic, because the prevailing direction of traffic is reversed). It decides the competition between alternative phases based on demand. FRAP is able to achieve invariance to rotation and flipping. Moreover, FRAP can be applied to intersections with different numbers of incoming lanes as well as a different number of possible phases. FRAP shows very good performance (in terms of average travel times) for a simple, single intersection setting. However, in a realistic setting with many intersections its performance deteriorates.

Another learning algorithm is described in [23]. “PressLight” simplifies the state to consist only of cars on incoming and outgoing lanes and the current phase. The reward is the “pressure” at an intersection [3], which is explained in detail in subsection IV-B.

The PressLight method outperforms both IntelliLight and LIT in both synthetic and realistic scenarios in terms of average travel time. PressLight outperforms the FRAP model in scenarios with more than one intersection as well. PressLight’s performance appears to be comparable with CoLight although no direct comparison has yet been published.

The publications mentioned above achieve convincing results. With the help of computer simulations, it is shown that reinforcement learning has great potential to help mitigate the problem of traffic congestion. It is less clear, however, how the machine learning approaches perform compared to previous adaptive approaches, also in terms of the computational resources needed. Similarly, the environmental costs of training the RL models are often left unreported. This will be the focus of our further investigation.

## IV. METHODS

In this section we will specify the design of the GuidedLight agent implementing “analytically guided reinforcement learning” (short: “ $\alpha$ -RL”). We will specifically describe the basis of the  $\alpha$ -RL approach which, as we will see, combines the benefits of the analytic approach with those of machine learning.

### A. DEEP Q-LEARNING

In the approach called  $Q$ -learning, the agents’ decisions are guided by a  $Q$ -function, which takes the current state and an action as arguments and maps them to the reward space. The mapping is according to the Bellman equation based on the expected future rewards as in Equation 1, where  $Q^{new}$  is the  $Q$ -value after the update for the given state-action pair  $(S_t, A_t$  at time  $t$  in this case);  $Q$  is the old  $Q$ -value for the same state-action pair;  $l$  represents the learning rate;  $\gamma$  weights the

importance of long-term vs. short-term gains.  $R_t$  is the reward at time  $t$  and  $\max_A Q(S_{t+1}, A)$  is the estimate of the optimal future value, more specifically it is the estimate of the highest  $Q$  value that can be obtained starting from state  $S_{t+1}$  and taking optimal actions. The term in the square brackets is also known as temporal difference [34].

$$Q^{new}(S_t, A_t) = Q(S_t, A_t) + l * [R_t + \gamma * \max_A Q(S_{t+1}, A) - Q(S_t, A_t)] \quad (1)$$

In deep  $Q$ -learning, the  $Q$ -function is approximated using a deep neural network. For most complex problems, it is impossible to specify the  $Q$ -function explicitly (that is why one speaks of a “black box” approach). The  $Q$ -function is learned by the agent as the training process advances. In modern implementations, the agent stores the previous action  $A$ , current state  $S$ , and reward  $R$  in its memory. Mini-batches of these data triplets are sampled from the memory at intervals and used to learn the  $Q$ -function [37].

In deep- $Q$ -learning, a neural network referred to as Deep  $Q$ -Network (DQN) is used to approximate the  $Q$ -function that estimates the reward, given a state-action pair.

Our GuidedLight implements a more advanced version of the DQN known as Double Deep  $Q$ -Network (DDQN), to avoid the overestimation of the action values. This is done by leveraging two parallel DQNs, which are updated with a different frequency using “soft updates” (see [38] for details).

We implement the memory replay [37] and train the DDQN periodically with mini-batches sampled from the memory. While the data is generated on the individual level of every intersection, the memory is shared between all agents to speed up convergence, by increasing the number of training samples. The details of the DDQN and memory implementations can be found in Appendix .

### B. PRESSURE-BASED LEARNING

Drawing on the good results of [23], [25], and the theoretical background of [3], we incorporate a “pressure” concept in the reward design of “GuidedLight”. Intuitively, the pressure can be interpreted as an imbalance in the distribution of vehicles over the incoming and outgoing lanes of an intersection.

Specifically the pressure of an intersection is defined in Equation 2, where  $i$  denotes the intersection,  $l$  the incoming lane of a given movement and  $o$  the outgoing lane of the same movement.  $w(l, o)$  represents the pressure of a single movement from lane  $l$  to lane  $o$ . The pressure of a single movement is simply the difference between the number of cars on the incoming lane  $l$  and the outgoing lane  $o$ , weighted by the maximum number of cars possible on the corresponding lanes. This is summarized in Equation 3, where  $x(a)$  denotes the number of cars on lane  $a$  and  $x_{max}(a)$  denotes the maximum possible number of cars on that lane:

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



where

$$w(l, o) = \frac{x(l)}{x_{\max}(l)} - \frac{x(o)}{x_{\max}(o)}. \quad (3)$$

Based on the results in [3], we conjecture that, optimizing the pressure at the level of individual intersections leads to the global throughput also being optimized, under certain constraints. Thus, we expect, the emergence of coordination between the intersection as long as they are optimizing their individual pressures.

### C. ANALYTIC COMPONENT

Our main goal is to build on the benefits of the analytic approach in order to improve the efficiency and accuracy of our learning method. An area that can benefit from analytic insights is the exploration strategy chosen by our agent. In reinforcement learning, exploration is a key concept that allows the agent to learn more about its environment and avoid getting stuck in local optima. The learning methods mentioned in subsection III-E rely on the epsilon-greedy ( $\epsilon$ -greedy) exploration method [34] explained in the following.

#### 1) EPSILON-GREEDY EXPLORATION

In this approach, every time the agent acts, with probability  $\epsilon$  it chooses a random action rather than the action suggested by its  $Q$ -function. This probability is usually relatively high in the beginning, in order to allow the agent to thoroughly explore the state-action space and generate enough experience to train its deep  $Q$ -network. However, as the training progresses, the value of  $\epsilon$  is gradually decreased to some minimal value, which is typically greater than 0 in order to provide a chance of further exploration even to trained agents. The value of  $\epsilon$ , its decrease rate and minimum value must be specified in a problem-dependent way, see Appendix .

#### 2) ANALYTIC EXPLORATION

In this paper we propose an alternative, analytic exploration process, where the exploration of the agent is guided by the results of some analytic method. The design extends the epsilon-greedy approach. Every time the agent acts, there is a probability  $\epsilon$  of choosing a random action. However, there is also some probability  $\alpha$  that the chosen value will not be random but an action that would be chosen by some analytic approach. Hence the overall probability to deviate from the action proposed by the  $Q$ -function is  $\epsilon$  and then there is the probability  $\alpha$  that the action will be analytically derived.

The intuition behind this approach is that we inject knowledge from the analytic approach, in order to guide our exploration into areas of the state-action space that are likely to be performing highly. The analytic approach is based on implications of the physical laws underlying traffic flows (“traffic physics”), which can be expressed by precise mathematical formulas. In comparison, the data-driven reinforcement learning approach, is only able to provide approximate relationships.

By injecting precise analytic knowledge into the exploration, we hope to accelerate the convergence of our method as compared to alternative, “blind” (i.e. unguided) learning methods. We expect that, with our new approach, the agent needs to explore less states to find the optimal state-action pairs. Nevertheless, we still allow random exploration to make sure the agent does not get stuck in local optima. The  $\alpha$  value, much like the  $\epsilon$ , can also be specified to decrease over time. The interplay between the two values might have a significant influence on the convergence. Furthermore, by relying on analytic exploration, the agent could even be allowed to safely explore in deployment, as most of the exploring actions would not be random, but analytically motivated (and so more likely to be efficient).

Note that the analytic exploration can be understood in terms of the heuristic-exploration paradigm [39]. In our case, the exploration uses a problem-specific heuristic - that of an analytic model. In that sense it can be considered a concrete application of the general heuristic exploration approach, which has been shown to achieve good results for many problems [39].

The details of the analytic approach used for the analytically guided exploration in this paper can be found in Appendix . Note that, for simplicity, we have restricted ourselves to the optimization rule of the analytic self-control approach proposed in [5], while the stabilization rule has been neglected, here (which may lead to higher densities, as we will see).

### D. GuidedLight AGENT

In this subsection we summarize the design of the “GuidedLight” agent implementing the analytically guided exploration paradigm ( $\alpha$ -RL)

- **Agent:** An agent is a decision-making entity that represents a single intersection in the traffic network and controls the traffic lights at that intersection.
- **State:** The state of the agent, also referred to as “observations” according to [23], consists of the percentage coverage of vehicles on the incoming lanes. We use the percentage coverage of the lane, as it implicitly includes the length of the respective lane: Three cars of approximately 5 meters each on a 30 meters lane should be considered differently from such cars on a 300 meters long lane. Furthermore each incoming lane is divided into 3 segments of equal length: closest to the intersection, middle and furthest. Such an approach has been shown to give superior results as compared to the unsegmented approach [23]. Moreover the state includes the percentage coverage of cars on each of the outgoing lanes and the current phase at the intersection.
- **Actions:** The actions, from which the agent selects, consist of the possible phases for the given intersection.
- **Reward:** The reward  $R_i$  uses the pressure concept [3] and is equal to the negative of the pressure  $P_i$  defined in

Equation 2:

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

In other words, the reward follows Equation 4, where  $i$  represents the specific intersection. The negative is taken, as we aim at minimizing the pressure, which corresponds to maximizing its negative.

## V. SIMULATION EXPERIMENTS

The goal of our computer-based simulation experiments is to test the machine learning approaches described above against the fixed time and the analytic approach. We will, therefore, conduct simulation experiments in several virtual city environments and compare the results to each other. We will also specifically evaluate the number of learning episodes needed for the learning approaches to achieve convergence.

The specific details of all parameters used in our experiments can be found in Appendix .

### A. METHODS COMPARED

Our simulation experiments will compare the following methods:

- **Fixed Time:** A fixed traffic light schedule, where we give 10 seconds to each phase with a 2 seconds clearing phase in between the phases as described in [2]. The same order of phases is followed by all agents. Hence, at a given time all intersections have the same phase. This is obviously a low baseline which, however, has repeatedly been used to compare the relative performance of various reinforcement learning approaches.
- **Demand:** A simple adaptive method, which always chooses the phases with the highest demand as expressed by the number of cars on the incoming lanes.
- **Analytic:** A state-of-the-art analytic approach relying on the optimization rule described in [4] and [5]. The method calculates both, the phase to be chosen and the amount of green time to be given following the details in Appendix .
- **PressLight:** A popular reinforcement learning approach [23] with a reward based on pressure [3] and an action-state space similar to the description in subsection IV-D, but considering the number of vehicles instead of the percentage coverage. For the purpose of the study, the “PressLight agent” was re-implemented. The results obtained by our implementation were compared with the open-sourced PressLight implementation and were found to be well consistent. Small differences might occur due to the use of a larger neural network, the use of a newer version of the CityFlow simulator, smaller set-up times as well as a larger number of actions-phases available to the agent.
- **GuidedLight:** A reinforcement learning approach using analytic insights for exploration, as proposed in section IV of this paper.

All the agents in all scenarios have 8 actions to select from. The actions correspond to all 8 non-conflicting phases avail-

TABLE 1. Traffic flows assumed in the four synthetic simulation scenarios.

Configuration	Demand (variance)	Arrival rate (vehicles/s)
I	Flat (0.3)	0.388
II	Peak (0.6)	0.388
III	Flat (0.3)	0.416
IV	Peak (0.6)	0.416

able at a 12 movement intersection. Both learning methods (GuidedLight and PressLight) explore the environment with the same probability  $\epsilon$ . The main difference is that GuidedLight chooses an analytically derived action with probability  $\alpha$  while PressLight always chooses a random action when exploring.

### B. COMPUTER SIMULATIONS

As simulation environment we use CityFlow [40] due to the availability of a large number of synthetic and realistic scenarios as well as the higher computational efficiency compared to SUMO [41].

#### 1) SCENARIOS

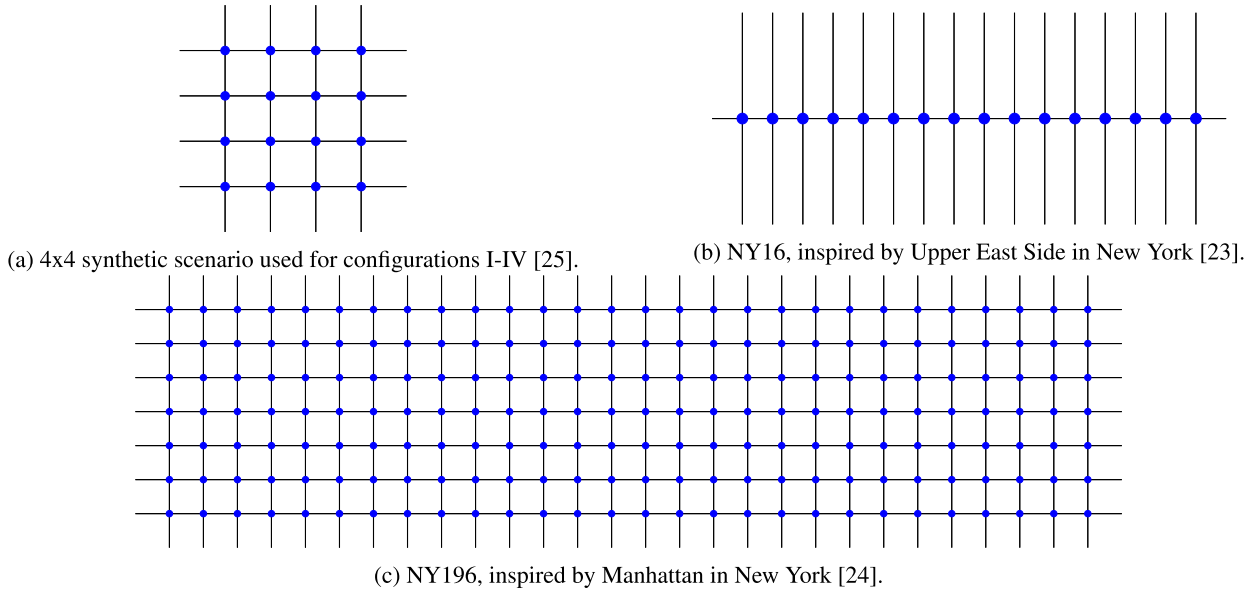
In our experiments we compare the aforementioned methods in a variety of scenarios. Four of them are based on synthetic configurations specified in Table 1, which follow the research design in [23]. The first setting is a 4 by 4 artificial road grid with 16 intersection agents. The distances between the intersections are assumed to be 100 meters.

At each intersection, the amount of vehicles turning left is set to 10%, the amount going straight to 60%, and the amount turning right to 30%. The specification of the synthetic traffic data follows [25].

The second simulation scenario is based on real-world traffic and a real world network: the 16 by 1 grid with 16 agents based on the 8th Avenue in Manhattan, New York. The road network is based on the road network data extracted from OpenStreetMap and flow data based on open-sourced taxi trip data as presented in [23]. The arrival rate is 1.886 vehicles/second with standard deviation of 0.009.

The third setting is also based on Manhattan, New York. However, it consists of 196 intersections of the Upper East Side. The vehicle flow, also based on taxi trip data, is set at 0.803 vehicles/second with a standard deviation of 0.0336. Since the taxi data provides only origin-destination data, the shortest path between two points is generated following [24].

In all scenarios, the vehicles arrive at the terminal edges of the road network. Moreover, the action frequency for the demand agents and learning agents is set to 10 simulation steps, where 1 simulation step corresponds to 1 second. If the phase is changed, a clearing phase is initiated first for a fixed time period of 2 seconds. During that time, only right turns are allowed, which is possible in all phases, following the custom in many countries. The traffic is bidirectional in all scenarios.



**FIGURE 3.** The five road networks used in the experiments, blue dots indicate intersections, black lines indicate roads.

The synthetic scenarios are included for better comparability with previously published learning methods [23]. The performance in the NY196 scenario is of greater interest to us, as it is realistically complex both, in terms of the road network and the traffic flows.

Each scenario is run for 1800 seconds, that is 30 minutes of real-world time. An “episode” is a full run of a simulation for the entire period of 1800 seconds, which corresponds to 1800 simulation steps.

### C. PERFORMANCE METRICS

The main performance metrics that we use for comparison are the average travel time (in seconds) and the throughput (in number of vehicles over the entire simulation period). The average travel time as well as throughput is calculated using the methods available in the CityFlow simulator [40].

For the machine learning methods, we present the minimum of the average travel time and maximum of the throughput along with the standard deviations in the last ten episodes of training, which can be treated as an indicator of the methods’ stability. We also provide data on the number of episodes needed for convergence of the learning methods. The learning methods are trained for 150 episodes.

### D. FURTHER ANALYSIS

In addition to studying the performance of different methods in different scenarios we also include an ablation study, where we validate the benefits of analytic exploration. We further investigate the influence of the  $\alpha$  parameter on the performance of the GuidedLight method.

Furthermore, we analyze the action space induced by the three best performing methods. We present a histogram of

the actions taken by the agent controlling the intersection to compare the similarity of the action space for different methods.

## VI. SIMULATION RESULTS

In this section we present the results of our simulation experiments described in the previous section.

### A. AVERAGE TRAVEL TIME AND THROUGHPUT

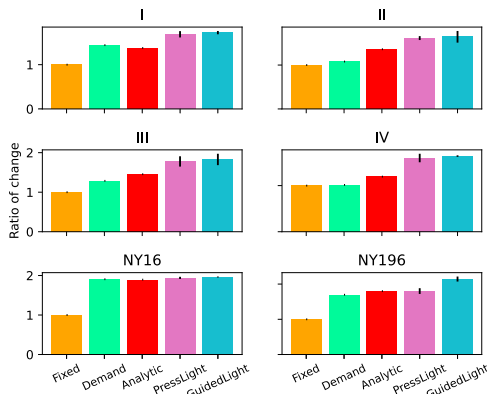
In Table 2 we can see the performance of the various traffic light control methods in terms of the average travel time and throughput for the four configurations of the synthetic scenario (I-IV) and the two real-world scenarios. As can be seen in the table, the GuidedLight method achieves the best results for all configurations.

If we compare the different approaches in Figure 4, we can see that the differences between the analytic and the GuidedLight approaches are especially significant for the synthetic scenarios and NY196. Interestingly, for the NY16 scenario, the simple Demand-based method is able to reach comparable results to the learning and analytic methods. Furthermore, the standard deviations of the PressLight method are higher than that of GuidedLight for the two realistic scenarios, suggesting that the training is less stable for PressLight than for GuidedLight.

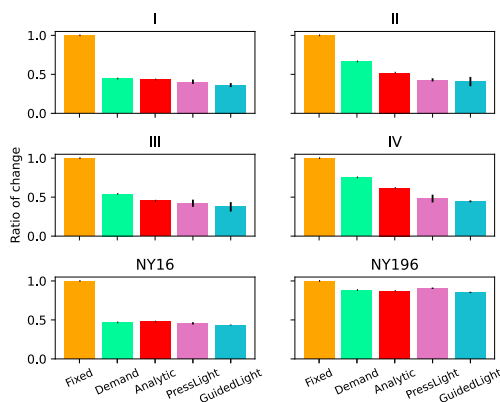
Similarly, by consulting Figure 5 we find that the travel time improvement over the Fixed Time method is significant for all methods tested. GuidedLight gives the best ratio of improvement in all scenarios. It is also worth noting that, for all methods, the travel time improvement over the Fixed Time method is lowest in the NY196 scenario.

**TABLE 2.** Average travel time and throughput of different traffic signal control methods for the four configurations of the 4 × 4 synthetic scenarios and two real-world scenarios. The values in brackets are standard deviations. Best results are presented in bold.

Model	Throughput (cars)						Avg. Travel Time (s)					
	I	II	III	IV	NY16	NY196	I	II	III	IV	NY16	NY196
Fixed	3056	3026	2968	3107	1387	1254	457	411	493	427	486	734
Demand	4404	3269	3821	3153	2644	2128	203	274	267	322	227	649
Analytic	4209	4118	4319	3713	2629	2257	198	214	224	264	232	640
PressLight	5154	4887	5278	4966	2691	2259	185	176	208	206	221	665
	(218)	(135)	(387)	(286)	(38)	(102)	(13)	(8)	(23)	(21)	(7)	(7)
GuidedLight	<b>5269</b>	<b>4975</b>	<b>5427</b>	<b>5095</b>	<b>2722</b>	<b>2687</b>	<b>165</b>	<b>167</b>	<b>185</b>	<b>190</b>	<b>211</b>	<b>626</b>
	(114)	(402)	(427)	(52)	(16)	(89)	(12)	(24)	(30)	(5)	(2)	(6)



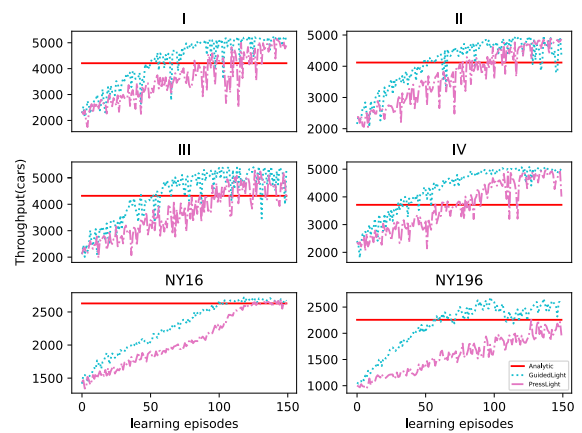
**FIGURE 4.** Throughput of various traffic light control methods relative to the throughput of fixed time control for the four configurations of the 4 × 4 synthetic scenarios and two real-world scenarios. Higher values are better.



**FIGURE 5.** Average travel time of various traffic light control methods relative to the average travel time for fixed time scheduling for the four configurations of the 4 × 4 synthetic scenarios and two real-world scenarios. Lower values are better.

**B. CONVERGENCE OF LEARNING METHODS**

Here, we compare the convergence of the machine learning approach PressLight to the  $\alpha$ -RL method GuidedLight. As can be seen in Figure 6, GuidedLight converges to a stable result significantly faster in all the tested settings than PressLight. This showcases the benefits of the analytically guided exploration ( $\alpha$ -RL). It is worth noting that the training of PressLight in most scenarios (I-IV) becomes unstable—unlike GuidedLight. This is likely due to the overestimation



**FIGURE 6.** Throughput achieved as a function of the number of learning episodes for the conventional machine learning method PressLight and the analytically guided method GuidedLight.

**TABLE 3.** Results of the ablation study on the NY196 scenario run to validate. the benefits of  $\alpha$ -exploration (with standard deviations determined over the last 10 training epochs).  $-\alpha$ -exploration indicates a model using random exploration.

Model	Throughput (cars)
GuidedLight	2687 ± 89
GuidedLight- $\alpha$ -exploration	2408 ± 108
PressLight+ $\alpha$ -exploration	2317 ± 92
PressLight	2259 ± 102

of the action values, which is avoided in GuidedLight by using the Double Deep Q-Network.

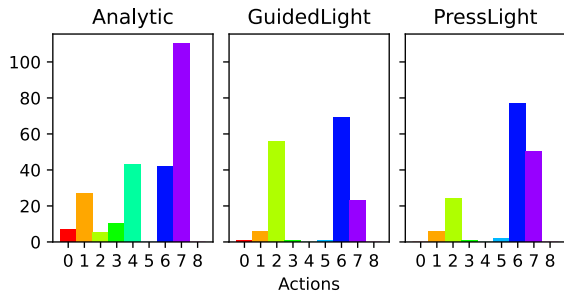
**C. ABLATION STUDY**

To validate the benefits of  $\alpha$ -exploration, we perform an ablation study. We compare the GuidedLight without  $\alpha$ -exploration with PressLight with  $\alpha$ -exploration and both normal GuidedLight and PressLight. In Table 3 we can see that indeed GuidedLight without  $\alpha$ -exploration performs much worse to GuidedLight with it. Similarly, PressLight with  $\alpha$ -exploration is superior to PressLight but not as good as GuidedLight, due to differences in DQN and state-space design (the use of DDQN instead of DQN and the use of percentage coverage instead of absolute number of vehicles in the state description).



**TABLE 4.** Throughput achieved by GuidedLight in the NY196 scenario with different starting and end values and changing rates of the  $\alpha$  parameter (with the standard deviations over the last 10 training epochs).

$\alpha$ start	$\alpha$ end	$\alpha$ change	Throughput (cars) $\pm$ std
1	0	-5e5	2687 $\pm$ 89
1	0	-5e6	2634 $\pm$ 96
0	1	+5e4	2665 $\pm$ 76
0	1	+5e5	2575 $\pm$ 59
0	1	+5e6	2440 $\pm$ 111
1	1	0	2408 $\pm$ 108



**FIGURE 7.** Histograms of actions taken by a single agent in the NY196 scenario according to each of the three methods. Different colors indicate different actions.

**D.  $\alpha$  PARAMETER STUDY**

In Table 4, we present the study of the effects of different values of the  $\alpha$  parameter, controlling the frequency of analytically guided exploration, on the results achieved by GuidedLight. We compare different starting and end values for the  $\alpha$  parameter, as well as different rates of change. It seems it is beneficial to initialize  $\alpha$  at 1 and decrease it to 0 gradually. Also, decreasing the  $\alpha$  parameter slowly appears to be favorable. Setting alpha to 1 and not decreasing the parameter corresponds to  $\epsilon$  exploration and obtains the worst results among the settings compared.

**E. ACTION SPACE ANALYSIS**

In order to further understand the differences and characteristics of the compared methods, we study the action space of the Reinforcement Learning and analytic methods. Actions correspond to the possible phases such as indicated in Figure 2. The agents using the analytic method favors action 7, PressLight appears to favor action 6 and 7 heavily, while GuidedLight favors actions 2 and 6. Furthermore, the analytic approach appears to select a greater variety of actions as compared to PressLight and GuidedLight. It is important to note that the analytic method selects more actions, as it adjusts the green time given to each action.

**VII. SUMMARY, CONCLUSIONS, DISCUSSION, AND OUTLOOK**

In this paper, we have compared different performance indicators of various adaptive traffic light control approaches and some alternative reinforcement learning methods. It turns out that the analytic method performs well, especially in

real-world inspired scenarios, and can, thus, serve as benchmark for novel reinforcement learning (RL) methods. We also note that the analytic method becomes less effective in highly congested traffic as in the synthetic scenarios, at least if the stabilization rule is neglected. Our results further show that  $\alpha$ -RL methods can significantly outperform the analytic approach after a sufficient number of learning episodes. Even though these results have been gained for somewhat idealized traffic scenarios, we expect qualitatively similar findings for irregular road networks and more complex traffic scenarios (as is to be shown in follow-up work).

The performance of the analytic method results from the use of mathematical formulas derived from traffic physics, which allow one to determine the green time needed to clear the entire vehicle queue, considering the arrivals of further vehicles based on a sophisticated short-term prediction. This mechanism also promotes coordinated traffic flows and emergent green waves, while not being restricted to repetitive service patterns.

Reinforcement learning misses analytic insight into the physical laws underlying traffic dynamics - it has to rely on guessing the dynamics based on traffic patterns that occurred in the past.  $\alpha$ -RL combines the good parts of RL and the analytic method, hence, outperforming both of these methods.

In summary we find that:

- In order to find superior solutions, one needs a “hybrid” approach, where the scientific knowledge behind the analytic approach is fed into the machine learning approach.
- Therefore, even in the age of Artificial Intelligence, analytic approaches remain important, but hybrid approaches are best.

**A. GREEN IT**

A recently highlighted issue in connection with the UN Sustainability Development Goals (SDGs) are the energy consumption and environmental footprint of technologies. While digital technologies contributed just about 3-5% to the world’s electricity consumption some years ago, the share is expected to grow beyond 20% by the year 2030 [42]. In some cities, the share of electricity spent on data centers is already higher than that.

These developments have caused a call for “green IT”, i.e. Information Technology solutions that have a low environmental footprint. This is of particular importance for machine learning methods [43], which are computationally quite expensive. Deep learning, including deep reinforcement learning, relies heavily on deep neural networks. This often requires vast amounts of GPU processing time, which translates into significant amounts of energy consumed.

It is, therefore, relevant to consider the ecological impact of reinforcement learning (RL) models used for traffic light control, since one of its goals is reducing emissions. It would be questionable to employ models to solve a problem, if they would actually exacerbate that problem. Some of the models we have mentioned take dozens of hours of training time

on a state of the art computational architecture until they converge [24], then reach a performance achieved by the analytic approach from the very beginning. Unfortunately, this is combined with limited generalization abilities to modified scenarios, for example, involving accidents or temporary building sites. Such typical disruptions of the regular operation would call for frequent retraining in order to avoid sub-optimal performance. The related ecological footprint should, hence, be taken into account, particularly considering the fact that highly performing analytic approaches exist, which are computationally cheap and environment-friendly.

At least it seems pressing to work on novel methods that use analytic knowledge, speed up convergence, and improve the ability to generalize. For these reasons, we have proposed a novel, hybrid machine learning method called “Guided-Light”, which combines the benefits of machine learning and analytic approaches by analytically guided exploration. We have shown that the proposed “ $\alpha$ -RL” method achieves considerably faster convergence than conventional reinforcement learning methods, leading to decreased training times. This in turn is expected to reduce the environmental footprint. Moreover, we were able to show that GuidedLight performs better than the analytic approach and better than the other reinforcement learning approaches studied in this paper. This applies particularly to the performance measures of average travel time and throughput. We believe that analytically guided machine learning would have benefits also in many other application areas, which are to be explored in the future.

#### APPENDIX. MODEL PARAMETERS

For both, PressLight and GuidedLight, we use a fully-connected neural network with two hidden layers of 128 and 64 hidden units, each. We use a learning rate of 0.0005, batch size of 64, a starting  $\epsilon$  of 1, a minimum  $\epsilon$  of 0.01.  $\epsilon$  is decreased by a factor of 0.00005 per each action taken by an agent. The size of the memory buffer is set to 100000, the discount factor to 0.999 and the soft update parameter to 0.0001. The network update frequency is set to 10. For GuidedLight, the value of  $\alpha$  is set to the value of  $\epsilon$  in each episode. Finally, for the analytic approach’s stabilization strategy we use  $T = 180$  and  $T_{max} = 240$ .

#### APPENDIX. ANALYTIC METHOD

In the study presented here, the analytic approach used to guide the exploration is the optimization rule proposed in [5]. We have selected this method because of its superior results among all analytic approaches we have tested. Specifically the analytic approach selects the next phase based on a priority score, which itself is related to the required green time  $\hat{g}$  to clear a given lane [5]. This value is derived from Equation 5, where  $N^{exp}(t)$  represents the expected number of cars that arrive at the intersection by time  $t$ ,  $N^{out}(t)$  the number of vehicles that depart from the upstream intersection by time  $t$ ,  $q^{max}$  is the maximum flow of the movement (“saturation

rate”), and  $\tau$  the set up time needed to switch phases:

$$N^{exp}(t + \tau + \hat{g}) = N^{out}(t) + \hat{g}q^{max} \quad (5)$$

The additional data needed to perform the analytic computations consist of the arrival and departure rates. This data is easily available to any intersection equipped with cameras, induction loops, or other suitable sensors. Therefore, the overhead of performing the analytic computations is negligible due to their low complexity, data availability and, lastly, because they are performed only with low frequency.

#### REFERENCES

- [1] G. Wünsch, “Coordination of traffic signals in networks,” Doctoral thesis, Dept. Mathematik und Naturwissenschaften, Technische Univ. Berlin, Berlin, Germany, 2008.
- [2] T. Urbanik, A. Tanaka, B. Lozner, E. Lindstrom, K. Lee, S. Quayle, S. Beard, S. Tsoi, P. Ryus, D. Gettman, S. Sunkari, K. Balke, and D. Bullock, “Signal timing manual,” Transp. Res. Board, Washington, DC, USA, NCHRP Rep. 812, 2015, p. 317.
- [3] P. Varaiya, “Max pressure control of a network of signalized intersections,” *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 177–195, Nov. 2013.
- [4] S. Lämmer and D. Helbing, “Self-control of traffic lights and vehicle flows in urban road networks,” *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 4, pp. 1–36, 2008.
- [5] S. Lämmer, R. Donner, and D. Helbing, “Anticipative control of switched queueing systems,” *Eur. Phys. J. B*, vol. 63, no. 3, pp. 341–347, Jun. 2008.
- [6] S. Lämmer and D. Helbing, “Self-stabilizing decentralized signal control of realistic, saturated network traffic,” *Traffic*, vol. 554, pp. 30–37, Sep. 2010.
- [7] C. Diakaki, M. Papageorgiou, and K. Aboudolas, “A multivariable regulator approach to traffic-responsive network-wide signal control,” *Control Eng. Pract.*, vol. 10, no. 2, pp. 183–195, Feb. 2002.
- [8] T. Le, P. Kovács, N. Walton, H. Vu, L. H. Andrew, and S. P. Hoogendoorn, “Decentralized signal control for urban road networks,” *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 431–450, Sep. 2015.
- [9] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti, “Revisiting street intersections using slot-based systems,” *PLoS ONE*, vol. 11, no. 3, pp. 1–9, Mar. 2016.
- [10] S. B. Cools, C. Gershenson, and B. D’Hooghe, “Self-organizing traffic lights: A realistic simulation,” in *Advances in Applied Self-Organizing Systems* (Advanced Information and Knowledge Processing), M. Prokopenko, Ed. London, U.K.: Springer, 2008, pp. 41–50.
- [11] C. Gershenson and D. A. Rosenbluth, “Self-organizing traffic lights at multiple-street intersections,” *Complexity*, vol. 17, no. 4, pp. 23–39, Mar. 2012.
- [12] C. Bielefeldt and F. Busch, “MOTION—A new on-line traffic signal network control system,” in *Proc. 7th Int. Conf. Road Traffic Monit. Control*, London, U.K., 1994, pp. 55–59, doi: 10.1049/cp:19940424.
- [13] T. Thunig, N. Kühnel, and K. Nagel, “Adaptive traffic signal control for real-world scenarios in agent-based transport simulations,” *Transp. Res. Proc.*, vol. 37, pp. 481–488, Jan. 2019.
- [14] M. Ramezani, N. De Lamberterie, A. Skabardonis, and N. Geroliminis, “A link partitioning approach for real-time control of queue spillbacks on congested arterials,” *Transportmetrica B, Transp. Dyn.*, vol. 5, no. 2, pp. 177–190, 2017.
- [15] B. Zhou, J. Cao, X. Zeng, and H. Wu, “Adaptive traffic light control in wireless sensor network-based intelligent transportation system,” in *Proc. IEEE 72nd Veh. Technol. Conf. (Fall)*, Sep. 2010, pp. 1–5.
- [16] M. Aslani, M. S. Mesgari, and M. Wiering, “Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events,” *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 732–752, Dec. 2017.
- [17] A. M. Turky, M. S. Ahmad, M. Z. M. Yusoff, and B. T. Hammad, “Using genetic algorithm for traffic light control system with a pedestrian crossing,” in *Rough Sets and Knowledge Technology*, P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang, Eds. Berlin, Germany: Springer, 2009, pp. 512–519.
- [18] T. Mao, A.-S. Mihaita, and C. Cai, “Traffic signal control optimization under severe incident conditions using genetic algorithm,” 2019, *arXiv:1906.05356*.

- [19] S. M. A. B. A. Islam, A. Hajbabaie, and H. M. A. Aziz, "A real-time network-level traffic signal control methodology with partial connected vehicle information," *Transp. Res. C, Emerg. Technol.*, vol. 121, Dec. 2020, Art. no. 102830.
- [20] Q. Guo, L. Li, and X. J. Ban, "Urban traffic signal control with connected and automated vehicles: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 313–334, Apr. 2019.
- [21] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 99, pp. 144–163, Jan. 2019.
- [22] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 2496–2505.
- [23] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "PressLight: Learning Max pressure control to coordinate traffic signals in arterial network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vol. 1, 2019, pp. 1290–1298.
- [24] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "CoLight: Learning network-level cooperation for traffic signal control," in *Proc. Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1913–1922.
- [25] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *Proc. AAAI*, 2020, pp. 3414–3421.
- [26] P. R. Lowrie, "SCATS, Sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic," Roads Traffic Authority NSW, Darlinghurst, NSW, Australia, 1990.
- [27] D. Helbing and A. Mazloumian, "Operation regimes and slower-is-faster effect in the control of traffic intersections," *Eur. Phys. J. B*, vol. 70, no. 2, pp. 257–274, Jun. 2009.
- [28] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems* (Autonomic Systems), T. McCluskey, A. Kotsialos, J. Müller, F. Klügl, O. Rana, and R. Schumann, Eds. Cham, Switzerland: Birkhäuser, 2016, pp. 47–66.
- [29] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A survey on traffic signal control methods," 2019, *arXiv:1904.08117*.
- [30] C. Gershenson, "Self-organizing traffic lights," *Complex Syst.*, vol. 16, no. 1, pp. 29–53, 2005.
- [31] N. H. Gartner and C. Stamatiadis, "Arterial-based control of traffic flow in urban grid networks," *Math. Comput. Model.*, vol. 35, nos. 5–6, pp. 657–671, 2002.
- [32] S. Lämmer, "Selbst-gesteuerte Lichtsignalanlagen im Praxistest," *Straßenverkehrstechnik*, vol. 60, no. 3, pp. 1–13, 2016.
- [33] A. Genser, M. Neuenschwander, and A. Kouvelas, "Wirkungsanalyse selbst-steuerung," ETH Zurich, 2020, doi: [10.3929/ethz-b-000456701](https://doi.org/10.3929/ethz-b-000456701).
- [34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
- [35] G. Zheng, X. Zang, N. Xu, H. Wei, Z. Yu, V. Gayah, K. Xu, and Z. Li, "Diagnosing reinforcement learning for traffic signal control," 2019, *arXiv:1905.04716*.
- [36] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, and Z. Li, "Learning phase competition for traffic signal control," in *Proc. Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1963–1972.
- [37] V. Mnih, K. Kavukcuoglu, D. Silver, A. Andrei Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [38] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 1–7.
- [39] N. A. Vien, N. H. Viet, S. Lee, and T. Chung, "Heuristic search based exploration in reinforcement learning," in *Computational and Ambient Intelligence*, F. Sandoval, A. Prieto, J. Cabestany, and M. Graña, Eds. Berlin, Germany: Springer, 2007, pp. 110–118.
- [40] H. Zhang, Y. Ding, W. Zhang, S. Feng, Y. Zhu, Y. Yu, Z. Li, C. Liu, Z. Zhou, and H. Jin, "CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 3620–3624.
- [41] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2582.
- [42] N. Jones, "How to stop data centres from gobbling up the world's electricity," *Nature*, vol. 561, no. 7722, pp. 163–166, 2018.
- [43] R. Schwartz, J. Dodge, N. Smith, and O. Etzioni, "Green AI," *Commun. ACM*, vol. 63, no. 12, pp. 54–63, Dec. 2020.



**MARCIN KORECKI** received the B.Sc. degree from the University of Groningen, and the M.Sc. degree in artificial intelligence from The University of Edinburgh. He is currently pursuing the Ph.D. degree with ETH Zürich. He specializes in machine learning, complex systems, and sustainable technology.



**DIRK HELBING** is Professor of Computational Social Science at the Department of Humanities, Social and Political Sciences and affiliate of the Computer Science Department at ETH Zurich as well as the AI Center. In January 2014 Prof. Helbing received an honorary PhD from Delft University of Technology (TU Delft). From June 2015 onwards, he was affiliated at the faculty of Technology, Policy, and Management of TU Delft for some years, where he led the PhD school in "Engineering Social Technologies for a Responsible Digital Future". He is also a member of the Complexity Science Hub Vienna.

• • •