IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 March 2023, accepted 5 April 2023, date of publication 12 April 2023, date of current version 17 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3266644

RESEARCH ARTICLE

How Well Do Reinforcement Learning Approaches Cope With Disruptions? The Case of Traffic Signal Control

MARCIN KORECKI[1], **DAMIAN DAILISAN**[1], **AND DIRK HELBING**[1,2]
[1]Computational Social Science, ETH Zürich, 8092 Zürich, Switzerland
[2]Complexity Science Hub Vienna, 1080 Vienna, Austria

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

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme through the Distributed Intelligence & Technology for Traffic & Mobility Management (DIT4TraM) Project under Grant 953783.

ABSTRACT Data-driven and machine-learning-based methods are increasingly used in attempts to master the challenges of the world. But are they really the best approaches to manage complex dynamical systems? Our aim is to gain more insights into this question by studying various popular reinforcement learning methods for traffic signal control, namely in disrupted scenarios characterized by significant, unpredictable variations. The results are expected to be relevant in subject areas ranging from traffic physics to transportation theory, from dynamics in networks to complex systems, from control theory to self-organization, and from adaptive heuristics to machine learning.

INDEX TERMS Traffic networks, reinforcement learning, self-organization, signal control, disruptions, benchmark.

I. INTRODUCTION

Around the world, the digital revolution is reshaping societies [1]. Data-driven approaches using machine learning (ML) and artificial intelligence (AI) are increasingly commonplace. In particular, they are considered to be good solutions for systems that are too complex to understand. To fit typical system behaviors and account for the complexity of some tasks, machine learning approaches may adjust thousands, millions, or even billions of parameters [2]. One popular application area is ''smart cities'', where such approaches are used, among others, to optimize and automate processes [3], [4], [5]. An application to traffic signal control seems logical but comes with particular challenges, as traffic flows in urban road networks are largely variable and hard to predict. Furthermore, traffic light control is an NP-hard optimization problem [6], [7], [8], which for reasonably sized cities cannot be solved exactly in real-time. While optimal solutions for average or ''typical'' traffic situations can be found offline, these will not be strictly optimal for any actual traffic situation [9], [10], [11], [12]. Hence, traffic signal optimization often uses heuristic algorithms for dynamic adaptation [13], [14]. In this connection, machine learning (ML) approaches, particularly reinforcement learning (RL) methods, have recently gained great popularity [15], [16], [17], [18].

Reinforcement learning (RL) algorithms operate traffic controllers in ways that are difficult to attain using traditional control approaches. When dealing with large traffic systems, decentralized approaches offer desirable qualities such as efficiency, adaptability, reduced cost, scalability, and resilience [19]. However, introducing decentralization also necessitates the need for coordination among agents [20].

In some algorithms, coordination arises as an emergent property resulting from self-organization, as demonstrated in [14], [21], and [18]. In comparison, RL-based methods can employ several ways to embed coordination into the learning process, such as passing messages [22], communicating observations between neighboring intersections [21], sharing parameters or states [23], or agent actions. In particular,

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

studies of RL approaches that leverage Deep Learning present results with significant improvements of traffic performance in the specific scenarios they have been trained on [15], [16], and [17]. However, related to RL's impressive performance is a tendency to overfit its training scenarios [24], [25]. While this can be in favor of deployment scenarios, when the traffic flows vary little compared to the training conditions, deviations from the normal case can result in sub-optimal traffic performance.

Despite such issues, one may expect that the use of machine learning methods is becoming or likely to become the state-of-the-art. So, how well are such methods performing as compared to other, perhaps considerably simpler methods? In this paper, given the large range of machine learning and traffic signal approaches, we cannot give a final answer. We rather propose some scenarios and techniques, which allow one to compare the performance of diverse approaches and to standardize procedures. In this connection, we study several benchmarks, various reinforcement learning approaches reported in the scientific literature, and some extensions of them, based on the approach of ''Pre-Training''.

Our work attempts to answer the question: how resilient are various decentralized algorithms for traffic signal control to disruptions in the traffic network? In contrast to many previous publications, we investigate scenarios that are disrupted. This serves to reflect the reality of traffic systems, the performance of which is affected by accidents and building sites every day. We simulate these events by randomly closing down links (i.e., sections) of the road network studied. So, we are interested in how ''resilient'' traffic signal control is to random disruptions, i.e., how well the approaches can handle such scenarios.

We evaluate several traffic control algorithms in multiple synthetic and real-world simulation scenarios, including RL-based methods, demand-driven and analytical approaches, and naive baselines (such as random and cyclical control schemes). In particular, we measure the steady-state effects of disruptions on the average travel times of vehicles in the system. In scenarios where RL-based methods fail to learn an effective policy, we Pre-Train them on a simple traffic system and apply the learned method to a more complex one. We demonstrate that this results in better performance than direct training.

II. BACKGROUND

In this section, we briefly review the different types of traffic control algorithms used in this study. We also provide an overview of the commonly used terms in the traffic control literature that the reader will encounter in the following sections.

A traffic network is a collection of **roads** that cross other roads at **intersections**. Roads that are bi-directional are represented using two **links**, one for each direction of the road. At an intersection, links that lead to the intersection are called *incoming*, while those that lead away from it are called *outgoing*. A link can be further subdivided into **lanes**. One can then consider pairs of incoming lane(s) and outgoing lane(s) as **movements**. Typically, we are interested in three types of movements: left turns, right turns, and straight movements. Traffic lights then regulate traffic flow by determining which combinations of non-conflicting movements, or **phases**, can pass through the intersection at a given time. Traffic lights assign a **green time** to an activated phase, which is the duration of the corresponding phase's movements, during which cars are allowed to progress.

Intersections can employ the use of control plan **schedules**, wherein the cycle of phase activations are repeated after a duration called the **cycle length**. For a given control plan, **splits** refer to the portions of the cycle length allocated to the various phases. The control cycles of intersections are often operated with a certain time shift, which can benefit coordination. The corresponding delay is referred to as cycle **offset**.

A. CONVENTIONAL CONTROL

The simplest method employed in traffic signal control uses centrally determined schedules of green times, phase activation cycles, and cycle lengths [26]. In practice, these schedules may be pre-optimized for ''typical'' traffic patterns, using offline tools [27], [28], [29]. Coordination across intersections is achieved by introducing a cycle offset to the time schedule of successive intersections. A limitation of using pre-timed schedules is that they are highly inflexible and do not respond to real-time traffic variations or disruptions in the traffic flow network.

Online implementations of adaptive algorithms such as SCOOT [13] offer more control over timing parameters such as split, cycle, and offsets, and can better respond to variations in traffic conditions in real-time. Another approach would be to remove the constraint of cycling through the phases in a pre-set order, and to use instead suitable criteria to select the next phase and green time duration. A simple adaptive method that follows the aforementioned approach is *demand-based control*, where the activated phase is chosen by measuring the sum of demands of all vehicles involved in movements of the given phase.

Actuated and adaptive control provides some flexibility by only activating traffic phases for movements that have vehicles detected by detectors. Phases are subject to minimum and maximum green times, and their durations can be extended, reduced, or even skipped according to the detected demand. However, demand-responsive actuated control alone cannot guarantee system-wide optimization.

B. SELF-ORGANIZING CONTROL

Self-organization approaches follow a different traffic control paradigm. In the traffic control context, an *agent* refers to the traffic signal controllers for each intersection in the network. Here, a set of rules determines how agents interact with each other, and these interactions result in some emergent dynamics in a decentralized manner [30]. One popular algorithm,

which is based on self-organization principles combined with queuing theory can control traffic using just two rules: an optimization and a stabilization rule [14], [31]. The interaction of these two rules across multiple intersections in the network results in the spontaneous emergence of *green waves*, which is a result of coordinating the phase activations of successive intersections along a corridor. This algorithm has been successfully implemented, for example in Dresden, Germany [32], and Lucerne, Switzerland [33].

C. REINFORCEMENT LEARNING

In reinforcement learning (RL), an agent learns to perform complex tasks through repeated interactions with its environment. This approach is mathematically formulated as a *Markov Decision Process* (MDP) [34], with the tuple $\langle \mathcal{S}, \mathcal{A}, R, P \rangle$. Herein, $\mathcal{S} \subseteq \mathbb{R}^n$ is the set of all possible states of the environment (partially or fully observable) in n dimensions. $\mathcal{A} \subseteq \mathbb{R}^m$ is the m-dimensional action space. $R \in (\mathbb{R}^n, \mathbb{R}^m) \rightarrow \mathbb{R}$ is the reward function that determines the "reward" for the state s' obtained by the agent after taking action a in state s. P is the transition probability function. The goal of maximizing the cumulative reward function allows an agent to learn the appropriate action a to take, given a certain state s [35].

One approach to solving the MDP is called Q-learning [36]. It uses a function $Q : s \times a \rightarrow \mathbb{R}$ to map state and action pairs to the reward space. This equation estimates the *quality* of the current state from the perspective of the expected rewards for possible future states. The Q-values are then iteratively updated using the equation

$$Q^{\text{new}}(s, a) = (1 - l)Q(s, a) + l \left[r + \gamma \max_{a'} Q\left(s', a'\right) \right], \tag{1}$$

where l is a learning rate. The discount factor γ controls the importance of immediate compared to future rewards. Learning such a function can become computationally demanding when large state and action spaces are considered. Thus, the development of Deep RL employs *deep neural networks* (DNN) to approximate the function $Q(s, a)$ [37], [38]. This has shown the ability to exceed human performance in complex tasks [39], [40], [41] and in traffic control [15], [16], [17], [18]. Unlike supervised learning, deep RL does not rely on labeled datasets. Training data are rather generated from interactions of the agents with their environment, where the agent may take random actions (exploration) or choose the action $a = \max_{a'} Q\left(s', a'\right)$ that maximizes the Q-function (exploitation).

In the following, we use the double deep Q-network (DDQN) [42], which employs two Q-networks that are updated with different frequencies via *soft updates*. We use a multi-agent RL paradigm with a single shared DDQN network for all of the agents. We also use the memory replay introduced in [39], but collect samples from all agents into a single memory replay buffer, following [17].

D. RESILIENCE

Resilience can be defined as a system's ability to withstand, respond to, and recover from disruptions [43]. Typically, this involves evaluating the evolution of a performance metric over time [44] in the form of a resilience curve. More formally, a resilience curve shows the evolution of a performance metric that maps system states to a scalar value throughout a scenario [45]. Such a curve provides multiple insights into how a system responds to disruptions, for example, the rate at which system performance degrades or recovers, the depth of impact, whether or not a system can restore its performance once the disruption is removed, and how quickly.

For transportation systems, disruptions can come in two flavors: a sudden change in traffic volume (demand side) or a change in network topology (supply side). In urban traffic systems, these two aspects of disruptions are strongly interlinked [46]. The transportation systems are prone to many types of disruptions such as natural disasters (earthquakes, floods [47]) or road works or accidents. An explicit quantification of resilience in the context of urban road networks can be found in [48]. In this work, we focus on the longer-term modifications to the network and look at the steady state resulting from various control algorithms in different traffic scenarios after disruptions, and comparing them to the undisrupted performance.

III. METHODS

In this section, we report the details of our conducted simulation experiments. We specify the flows and road networks used in the simulated scenarios and explain the logic behind the simulated disruptions and the Pre-Training procedure. Lastly, we present the methods compared in the experiments.

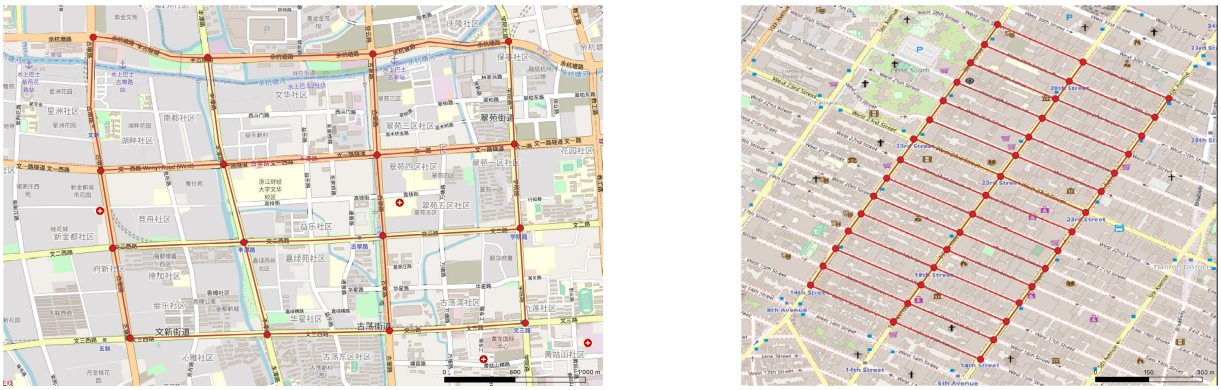
A. EXPERIMENTS

Two series of experiments have been run for this study. The first series is referred to as the **Benchmark Experiments**, where we showcase the performance of different traffic signal control methods in the proposed benchmark scenarios. In this connection, our study aims to introduce a set of diverse scenarios along with a method of disrupting them.

The second series of experiments are referred to as **Pre-Training Experiments**. They focus on the Reinforcement Learning approach to traffic signal control and showcase the benefit of using simple scenarios for Pre-Training these learning methods. All the experiments are conducted in the CityFlow simulator [50], using open boundaries and stochastic Poisson-process for vehicle arrivals. Each scenario is run for 3600 steps, corresponding to an hour of real time. The clearing all-red phase is set to 5 seconds for all simulation models, and is only activated when the phase signal changes (there is no yellow/orange phase).

TABLE 1. Details of the scenarios used in the experiments. The number of vehicles refers to the total number of vehicles that have their starting time within the scenario's run time.

	2x2	4x4 Het	Hangzhou	NY48	NY48 double	NY48 quad
Number of vehicles	14400	12117	2983	2824	5648	11296
Median starting time (sec)	1799.5	1842.0	1600	1781	1781	1781
Arrival rate (vehicles/sec)	4	3.37	0.83	0.78	1.57	3.14



(a) Gudang Residential District in Hangzhou, China. (b) Manhattan Upper East Side, New York City, USA.

FIGURE 1. Maps of the cities used for the real-world data-based scenarios: Hangzhou and NY48 [49]. Red lines indicate the roads considered in the simulated scenarios.

B. SCENARIOS

The proposed scenarios represent a diverse set of road network topologies and traffic flows on which one can comprehensively test different traffic signal control methods. The idea is that some methods might perform better under particular conditions (e.g., low traffic vs. high traffic). Thus, to test a given method and its robustness, one should compare its performance across various flow and topological conditions.

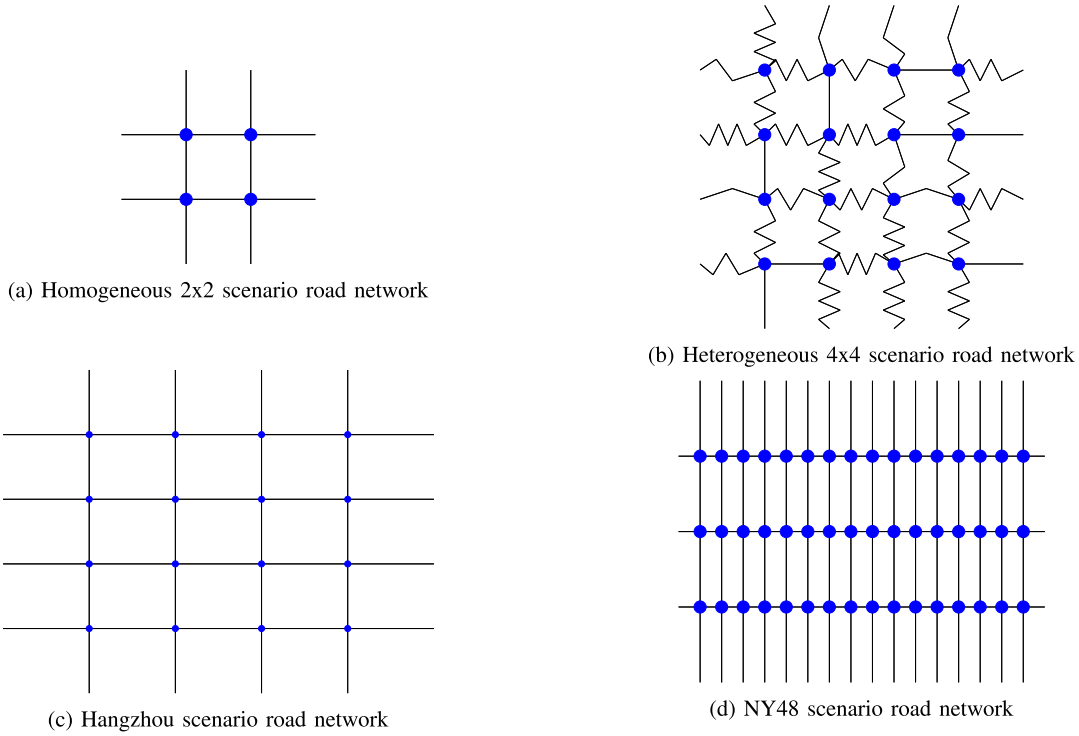
Table 1 presents the details of the flows for the six proposed scenarios. The number of vehicles and their arrival rate varies from low (Hangzhou scenario) to high (2×2 and NY48 quad). In the 2 × 2 scenario, only straight movements occur (in its disrupted version the turning movements are also added to reroute around the affected link). The Hangzhou scenario is (for most methods) easy and uncongested due to low arrival rates and long roads. In the NY48 scenario, even though the arrival rates are not extremely high, there is significant local congestion occurring (due to the short length of some roads). As an additional stress test of all control algorithms, we took the NY48 scenario and generated two additional synthetic scenarios with double and quadruple ('quad') arrival rates.

Fig. 1 illustrates the areas of the cities upon which the two real-world data scenarios are based (Hangzhou and NY48). Considering both, realistic and synthetic scenarios, we are able to represent a greater variety of flows and topologies. Fig. 2 represents the idealized road networks underlying our simulation scenarios. As can be seen from Fig. 2, the scenarios have either homogeneous road lengths (2 × 2, Hangzhou) or heterogeneous road lengths (4 × 4, NY48).

C. DISRUPTIONS

The simulated disruptions are a key concept of this paper. We propose a simple and intuitive method of generating disruptions by removing links from the road network. The underlying logic is that accidents, construction sites, or any other disruptive events will likely affect the entire link and result in either partial or complete blockage. For the sake of simplicity, we assume that any disruption makes the link completely useless for traffic. Thus, a single disruption affects all lanes of the link (where we consider a link to be unidirectional). These disruptions necessarily affect the flow dynamics in the scenario. Fig. 3 exemplifies how this is implemented in our experiments. Every car with a path that traverses the disrupted link is rerouted via the shortest alternative path. If multiple alternative paths are available, one of them gets assigned at random assuming a uniform distribution. A disrupted link with only one alternative path will congest that single alternative path more than a link with two alternative paths, where the traffic will be distributed across the two alternative (assuming the same flows for both links). Therefore, for the sake of consistency, we only allow for the disruption of links that allow two alternative paths (otherwise in some samplings where more links with one alternative path are selected there would be larger congestion).

We quantify the disruptions as a ratio z of disrupted links to the number of agents (controlling traffic signals at intersections) in a given scenario. We choose this manner of quantification as we want to investigate the disruptions' effects on the agents and their ability to control traffic in the



(a) Homogeneous 2x2 scenario road network

(b) Heterogeneous 4x4 scenario road network

(c) Hangzhou scenario road network

(d) NY48 scenario road network

FIGURE 2. Road networks of the scenarios used in the simulation experiments. Black lines represent links, blue dots represent intersections. The generation of the heterogeneous street networks in (b) is adapted from [21].

TABLE 2. Average path length for various scenarios, reported in terms of the number of links passed (a link is a unidirectional connection between two nodes representing intersections). The average path length is the average length of the route of each vehicle in a given scenario in terms of links. The index z in Dis_z refers to the number of disrupted links in a given scenario. Specifically, z is the ratio of disrupted links compared to the total number of agents in a scenario. In parentheses, we present the standard deviations of the path lengths.

	Avg. path length (links)						
	Dis_0	$\text{Dis}_{0.0625}$	$\text{Dis}_{0.125}$	$\text{Dis}_{0.1875}$	$\text{Dis}_{0.25}$	$\text{Dis}_{0.5}$	Dis_1
2x2	3 (0)	–	–	–	3.25 (0.66)	–	–
4x4 Het	4.91 (2.17)	5.03 (2.33)	5.15 (2.49)	5.29 (2.70)	5.40 (2.82)	–	–
Hangzhou	4.65 (2.15)	4.74 (2.29)	4.82 (2.41)	4.90 (2.53)	5.01 (2.72)	–	–
NY48	10.00 (4.63)	10.28 (4.88)	10.57 (5.17)	10.92 (5.54)	11.40 (6.01)	13.30 (8.00)	18.28 (12.76)
NY48 double	10.00 (4.63)	10.29 (4.90)	10.47 (5.05)	10.91 (5.53)	11.49 (6.13)	13.42 (8.01)	17.53 (12.04)
NY48 quad	10.00 (4.63)	10.27 (4.89)	10.53 (5.13)	10.87 (5.48)	11.15 (73)	13.53 (8.21)	17.96 (13.04)

system. Thus, we link the disruption to the number of agents explicitly. Moreover, a single disruption directly affects one agent's downstream link and another agent's upstream link. Hence, when our ratio z is equal to 1, all agents in the system are directly affected by disruptions, but not all links (meaning that each agent has at least one of its links affected by disruption).

We performed simulations for varying disruption levels for each of the scenarios and generate 100 replications for each scenario (i.e., each disruption level–scenario pair). The disrupted links were chosen uniformly at random from the set of all the links that allow for two alternative paths around them. All methods were trained and evaluated on each of the disrupted scenarios.

Table 2 presents the quantified effects of the disruptions on the flows of the studied scenarios. In the experiments, we study six levels of disruptions, which we represent in terms of the ratio of disrupted links to the number of intersections in a given scenario. As can be seen in most cases, the more disruptions occur, the longer the average path length of vehicles in the system (due to the required detours) and the higher the standard deviation of the path lengths.

It is worth noting that the disruptions were always local. Hence, their effects were initially also local, but they could eventually spread and affect the entire system.

D. PRE-TRAINING

Pre-Training is a well-known method in Machine Learning that allows one to train a method on some data and then apply the method successfully to different data. It might be employed if the data are hard to acquire or training on the full dataset is expensive. This approach can also avoid overfitting

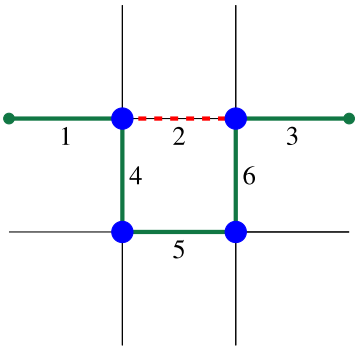


FIGURE 3. Representation of the effects of a disruption in the 2 × 2 scenario. Here, link 2 becomes disrupted. As a result, any vehicle with a path going through link 2 (e.g., 1 – 2 – 3) uses a modified path leading around the disrupted link 2 (e.g., 1 – 4 – 5 – 6 – 3).

by training a method on more general data before applying the method learned to more specific cases.

Pre-Training has recently been employed for traffic signal control with promising results [21]. In our particular setting, the Pre-Training is motivated by the weak performance of the Deep RL methods in the NY48 scenario. Accordingly, the better-performing Deep Learning method (GuidedLight) is trained on the smaller scenarios—4 × 4 Hom and 4 × 4 Het —and then applied to all disrupted scenarios. The 4 × 4 Hom scenario has the same arrival rates as 4 × 4 Het but has homogeneous and smaller road links. We select it for here as it has achieved good results as a Pre-Training scenario in previous research [21]. The method only learns on the smaller scenarios, and there is no learning in deployment.

E. COMPARED METHODS

We implement several control methods within the simulation framework of the CityFlow simulator. To simplify the implementation of some of the algorithms, we define an action interval of 10 seconds as a hyperparameter, which constrains the green time of each selected phase to 10 seconds. This still allows agents to give more than 10 seconds of green time in total to a single phase (without triggering the all-red clearing phase), as they can choose to select the given phase again in the following action interval. The constraint is that the green time will be in increments of 10. Two algorithms are excluded from this constraint, as their implementation explicitly prescribes the exact duration of green times of the activated phase. In our Benchmark Experiment, we compare the following methods:

- **Random**: Each agent chooses its actions at random.
- **Cyclical**: Each agent chooses its actions based on a fixed cycle. The green time is varied and depends on the number of vehicles awaiting service. Each agent starts with a phase chosen at random.
- **Demand**: A simple adaptive method, where the agent selects the phase, which has the highest demand.
- **Analytic+**: An adaptive, self-organizing method relying on optimization and stabilization rules [14]. Green times are varied.
- **PressLight**: A Deep Reinforcement Learning method relying on Q-learning [17].
- **GuidedLight**: A Deep Reinforcement Learning method relying on Q-learning and heuristic exploration [18].
- **Pre-Trained GuidedLight**:
  - 4 × 4 **Het**: The GuidedLight method Pre-Trained on the heterogeneous 4 × 4 scenario (with variable lengths of street sections).
  - 4 × 4 **Hom**: Same, but Pre-Trained on the homogeneous 4 × 4 scenario (where the road network is a square grid and the road lengths are of uniform lengths (300m) and therefore on average shorter than 4 × 4 Hom (447.23 ± 115.82m)). This shares the flows with 4 × 4 Het, whereas the road sections are homogeneous and shorter than in 4 × 4 Het. The scenario has been used for Pre-Training with good results in [21].

These methods were chosen for our simulation experiments as they represent a diverse set of approaches to traffic signal control. The **Random** method is presented as a low-performance bound, as no well-working method should, on average, perform worse than random control. The **Cyclical** approach represents the still widely used cyclic (periodic) approach to traffic signal control, where the demand determines the (fixed) green time splits. **Demand** and **Analytic+** are two methods, whose actions are algorithmically driven by the variable demand in an adaptive way. Lastly, the **GuidedLight** and **PressLight** represent the recently more and more popular RL approaches.

For the RL methods, we implement the agents following [17], [18], [21]. States observed by the agents at the intersections are vehicle occupancies of the incoming and outgoing lanes and the one-hot encoded vector representing the previous action. We define the reward received by the agent at intersection i as the negative *pressure* $-P_i$:

$$r_i = -P_i \tag{2}$$

$$= -\left| \sum_{(l,o)\in i} \frac{x(l)}{x_{\max}(l)} - \frac{x(o)}{x_{\max}(o)} \right|. \tag{3}$$

The pressure is the difference between the number of vehicles x on the incoming l and outgoing o lanes at the intersection, weighted by the capacities $x_{\max}$ of the lanes. We choose negative pressure as the reward since it has been confirmed to work well in prior work [17], [18]. Furthermore, it can function well in a decentralised setting, where the agent only has access to local information. The non-pretrained RL methods are trained on a given scenario with no disruptions and tested on all the levels of disruption (and all the samples for each level) for the given scenario.

IV. RESULTS

In this section we present our results of the two series of experiments. The **Benchmark Experiments** highlight the various traffic networks and demand flow configurations that we use in order to demonstrate the ability of a control

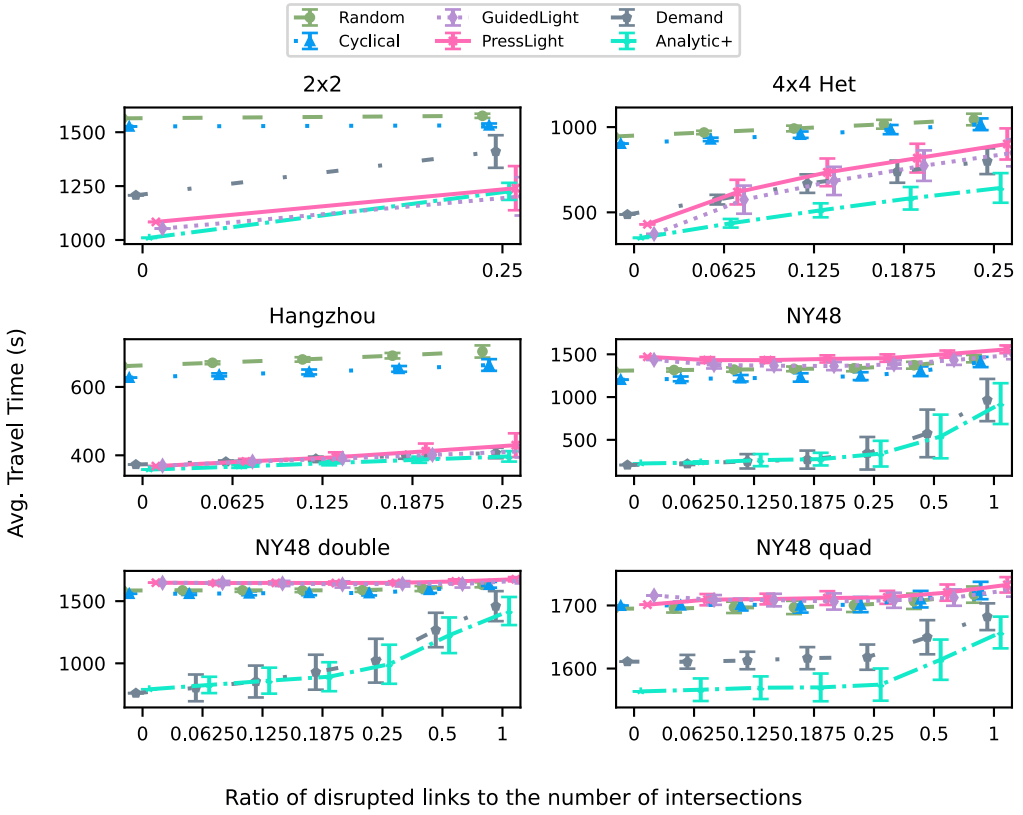


FIGURE 4. Results for the average travel times in our benchmark simulation experiments. Error bars indicate the standard deviations of the 100 disrupted scenarios simulated for each disruption level. Lower values are better.

algorithm to manage traffic flows. This is followed by the **Pre-Training Experiments** that are addressing some limitations of Deep RL algorithms, which prevent them from learning effective control policies in some traffic scenarios.

A. BENCHMARK RESULTS

The results of the **Benchmark Experiments** are presented in Fig. 4. In all scenarios, the **Analytic+** method performs better than all other methods. In some scenarios (Hangzhou, NY48, NY48 double), some methods perform similarly well, while in other scenarios (4 × 4 Het, NY48 quad), it is clearly superior to all other methods. We also note that the **Random** method is the worst-performing approach in almost all scenarios, as expected.

Furthermore, in all scenarios, **Analytic+** performs better than all other methods. In some scenarios (Hangzhou, NY48, NY48 double), a couple of methods perform similarly well, while in other scenarios (4 × 4 Het, NY48 quad), **Analytic+** is clearly superior to all other methods. We also note that the **Random** method is the worst-performing approach in almost all scenarios, as expected. Modulating the green times in cyclic scheduling leads to better performance, such that **Cyclical** outperforms **Random**. Interestingly and unexpectedly, in the three NY48 scenarios the two Reinforcement Learning methods perform poorly, at a level comparable to random. We will discuss the weak performance of the learning methods and how to overcome it with Pre-Training in Section V.

The low performance is especially striking since **GuidedLight** reaches a level comparable to **Analytic+** in the 2 × 2, 4 × 4, and Hangzhou scenarios. Moreover, **GuidedLight** typically outperforms **PressLight**.

For the scenarios themselves, we note that increases in the disruption level decrease the performance of all control methods. This trend is particularly pronounced in the $D_{0.5}$ and D_1 NY48 scenarios.

B. PRE-TRAINING RESULTS

The results of the **Pre-Training Experiments** are reported in Fig. 5. As we can see, the best-performing learning method across all scenarios except Hangzhou is **GuidedLight** 4 × 4 **Hom**, which performs at a comparable level as **Analytic+** in the 4 × 4, Hangzhou and NY48 scenarios. The performance in NY48 quad, for smaller disruptions, is not as good as **Analytic+**, but significantly better than the results of the non-Pre-Trained version. The question of why the Pre-Trained **GuidedLight** 4 × 4 is successful will be addressed in the next section.

V. DISCUSSION

In this work, we study traffic scenarios that can test the performance of an algorithm under various traffic conditions. The 2 × 2 scenario only has through movements, which is

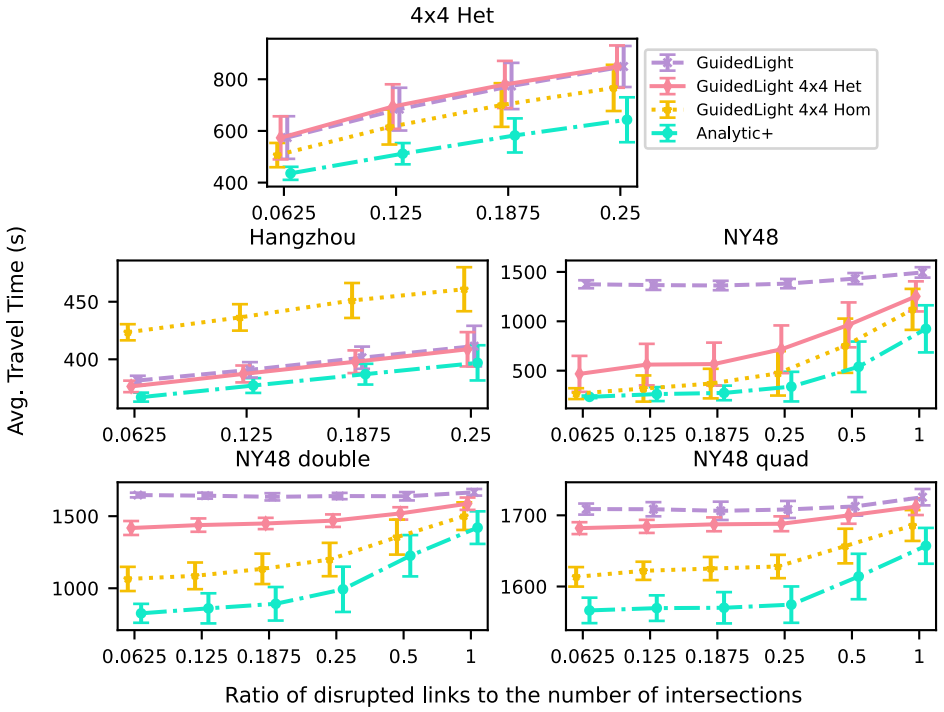


FIGURE 5. Results for the average travel times in the Pre-Training experiments on all scenarios except 2 × 2 (2 × 2 is omitted due to having only one disruption level). Lower values are better. Analytic+ results are shown for comparison. The error bars indicate the standard deviations over the 100 disrupted scenarios for each disruption level.

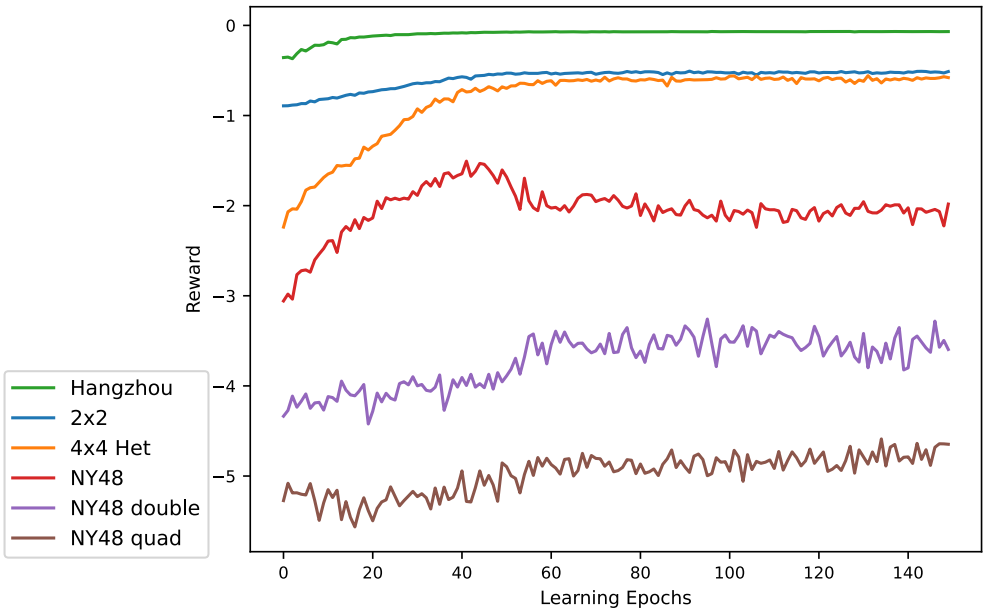


FIGURE 6. Convergence in the training of the machine learning algorithm behind the **GuidedLight** approach for each scenario. The drop in reward for the NY48 training, around epoch 40, corresponds to the exploration rate dropping to its minimum value.

quite easy to handle by adaptive methods that can skip phases, but a problem for cyclic algorithms. The 4 × 4 Het scenario is more challenging to learn, as the algorithm must be able to handle a variety of topologies, while still maintaining control across a larger network. The Hangzhou and NY48 (and its variants) serve as tests of real-world traffic scenarios with inhomogeneous traffic flows in the road network.

Our study extends the benchmark scenarios to settings involving disruptions, which are relevant for the resilience of traffic signal control. The results indicate a significant dif-

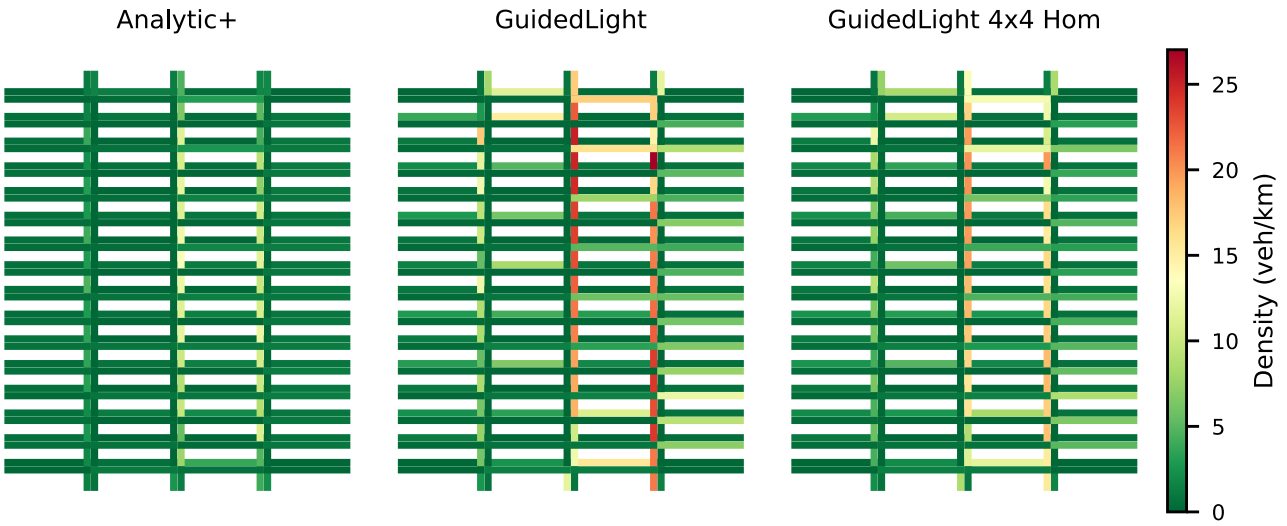


FIGURE 7. Average vehicle density on the links of the NY48 scenario. Most traffic flows are running in North- and South-bound directions on the right-most and center roads, respectively. The Analytic+ method is able to prevent traffic buildup on the horizontal roads (left). However, without Pre-Training the GuidedLight approach (center) causes congestion, which is represented by red, orange, and yellow colors. Pre-Training on the homogeneous 4 × 4 network allows for GuidedLight to mitigate congestion, but still not at the level of performance of Analytic+ (right).

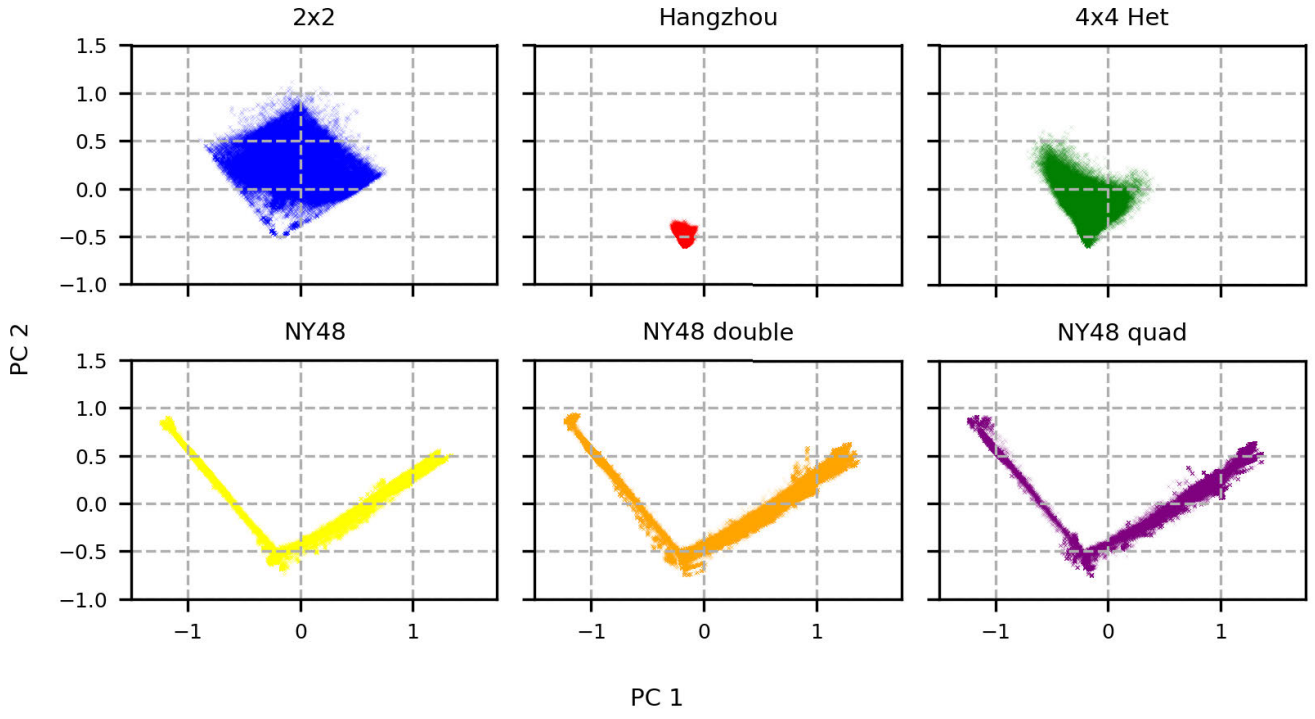


FIGURE 8. Projection of the states that the **GuidedLight** method encountered during the training of each scenario. The states are projected into a 2D plane using Principal Component Analysis (PCA) [52]. The ratio of variance explained by the two Principal Components (PC) is 0.30 for PC1 and 0.13 for PC2. The PCs were computed for the combined state data from all scenarios to allow for comparison across different scenarios. Then the states from each scenario were separately projected into the shared PC space.

ference in how various algorithms can deal with unexpected, local changes in the flow dynamics.

The behavior of the two RL methods, **PressLight** and **GuidedLight**, is of particular interest, especially in the NY48 scenario. Both methods can learn (converge during training) at a comparable level in the 2 × 2, 4 × 4, and Hangzhou scenarios. However, in all three NY48 scenarios, both RL methods perform at a level similar to **Random** control. Learning curves for **GuidedLight** show that, whereas heuristic exploration allows the agents to learn a good policy during the early stages of training, these are eventually "forgotten" at the later stages (learning epochs) (see Fig. 6). This effect occurs due to the NY48 scenarios being highly congested locally (see Fig. 7).

One reason could be that, in congested scenarios, the training signal provided to the RL method is not meaningful. In fact, in many congested situations, congestion persists, no matter what action is taken. Hence, little can be learned from such scenarios. This finding is in accordance with the known inability of some Deep RL methods to learn in congested situations [51].

Our Pre-Training Experiments investigate the performance of RL methods, which are Pre-Trained in a certain scenario before being deployed (without further training) to a different scenario. Interestingly, we find that the **GuidedLight** 4×4 **Hom**, Pre-Trained on the 4×4 homogeneous scenario, performs better than the same method Pre-Trained on the 4×4 heterogeneous scenario. The good performance of the **GuidedLight** 4×4 **Hom** and **GuidedLight** 4×4 **Het** is especially apparent in the NY48 scenarios. While the reinforcement learning methods trained on the NY48 scenario had trouble converging, the same method when Pre-Trained on either of both 4×4 scenarios, performed at a level closer to the best-performing method, **Analytic+** (and in the case of **GuidedLight** 4×4 **Hom** on practically the same level). We also note that **GuidedLight** 4×4 **Het** performs worse than **GuidedLight** 4×4 **Hom** in all scenarios except Hangzhou. This is likely because the 4×4 Hom scenario has shorter links but the same flows. Thus, the 4×4 Hom scenario is more difficult to learn, but also more relevant. This shows that a good Pre-Training scenario needs to be challenging enough for the learnings generated by the RL method to be informative, but not so difficult as to make the learning signal meaningless due to congestion.

We further analyze the states encountered by **GuidedLight** during the training phase (Fig. 8) using Principal Component Analysis (PCA) [52]. We note that in the scenarios, where **GuidedLight** is able to converge (2×2, 4×4 Het and Hangzhou), the states form blob-like clusters, while in the NY48 scenarios, the states are spread out in a v-shape (see Fig. 8). From this, it follows that the states encountered in training are significantly different across the traffic scenarios. However, if one looks at the states encountered in the 4×4 Het scenario, we see that these states are also observed in the 2×2 and Hangzhou scenarios, which can explain why Pre-Training on the 4×4 scenario yields good performance on the other two scenarios. Moreover, learnings from the 4×4 Het network (where traffic demands are almost the same on all links) seem to be more relevant to complex signal control than the states learned by the agent directly trained on the NY48 scenarios.

A further finding from our Pre-Training Experiments is that some Deep Q-Learning algorithms (such as **GuidedLight**) are actually surprisingly good at meta-learning, even without any explicit meta-learning techniques being implemented. In fact, in our study, we show that a rather simple DDQN-based method, Pre-Trained on one scenario, can perform very well in different scenarios across varying disruption levels. Thus, a Pre-Trained DDQN is clearly able to adapt in some way, even to situations that were not explicitly included in the training environment. This confirms the recent findings reported in [53] for the TD3 Q-Learning algorithm.

We would also like to highlight the environmental benefits of Pre-Training (which have been studied in [21]. This is because training on smaller scenarios takes less computational resources (time and energy) than training on large, complex traffic scenarios. At the same time, in many cases, it can be expected to lead to comparable or even better results. Overall, this allows one to save time and energy, while reaching better performance.

VI. CONCLUSION

A. A NEW BASELINE

Last but not least, we point to the superiority of the self-organizing **Analytic+** method, which does not need any training and can be deployed to any scenario with ease. This method is performing best in all the scenarios studied here. Only **GuidedLight** 4×4 and **Demand** reach similar performance in some, but not all scenarios.

The great performance of the **Analytic+** approach is perhaps a bit unexpected. However, it can be explained as follows:

(1) The method uses short-term predictions based on the inflows into the neighboring road sections, specifically the incoming links. This assumes technology that does not only measure outflows, but also inflows.
(2) The method exploits exact mathematical relationships from traffic physics or queuing theory, which a machine learning approach may only approximate if it manages to figure out the relationship at all. The consideration of inflows allows for perfectly tailored green times, which avoid stops for arriving cars. In such a way, local coordination is enabled, which promotes the emergence of green waves.
(3) By clearing long queues before returning to travel time minimization, the stabilization rule prevents spill-over effects. In such a way, it is avoided that congestion extends into upstream intersections and beyond. Therefore, the formation of congestion patterns, which expand over large areas, is largely prevented. This keeps traffic fluid as long as possible.

The above also explains the great performance of the **Analytic+** method in adapting to disruptions in our simulation experiments. Thus, we believe the **Analytic+** method can serve as a good baseline, which machine learning and other methods can and should be compared against. It is clearly a more difficult baseline to beat than random, fixed time, or cyclical approaches and, thus, offers better insights into how well a given method really performs.

B. GENERAL IMPLICATIONS

Machine learning and AI have become approaches that are increasingly applied to solve problems of all kinds. They are also being proposed to solve security, sustainability, resilience, and health issues, to mention just a few of the grand

challenges humanity is trying to address. All those problems, however, concern complex dynamical systems, for which traffic flows in urban street networks are a good example. Like many other complex dynamical systems, traffic flows suffer from

- limited resources (space and flow capacity, for example);
- non-linear network interactions;
- heterogeneous/diverse behavior;
- feedback and delay effects;
- great variability;
- randomness;
- largely unpredictable dynamics;
- disruptions;
- limits to control;
- an exact real-time optimization not feasible due to computational complexity.

As mentioned before, such kinds of features are also typical for many other systems making up our world. It is, therefore, expected that many of our findings will extend to other complex dynamical systems as well, including material flow and supply networks as well as our economy [54], [55], [56], [57]. Accordingly, machine learning approaches, even though comfortable and powerful, may not always be the best methods to solve such problems. Suitable adaptive approaches, which flexibly respond to short-time predictions of local needs, may often perform better, based on feedback rather than control, and coordination rather than optimization. In other words, certain analytical approaches, which are based on understanding the underlying system logic and dynamics (i.e., based on disciplinary knowledge, not just data-driven methods), may outperform machine learning approaches for at least two reasons:

(1) machine learning comes up with approximate solutions (and only, if it converges well),
(2) machine learning takes time, while the world may change in the meantime, for example, due to frequent disruptions.

We should, therefore, not ''put all eggs in one basket'', i.e., not rely on generic machine learning approaches alone. Overall, it is expected that hybrid approaches, which combine analytical and machine learning approaches, perform best [18]. Therefore, future work might explore new ways to integrate elements of the **Analytic+** method into machine learning approaches, which are specially tailored to the challenges of traffic signal control.

REFERENCES

[1] D. Helbing, *Next Civilization: Digital Democracy and Socio-Ecological Finance-How to Avoid Dystopia and Upgrade Society by Digital Means*. Cham, Switzerland: Springer, 2021.
[2] X.-W. Chen and X. Lin, ''Big data deep learning: Challenges and perspectives,'' *IEEE Access*, vol. 2, pp. 514–525, 2014.
[3] I. U. Din, M. Guizani, J. P. C. Rodrigues, S. Hassan, and V. V. Korotaev, ''Machine learning in the Internet of Things: Designed techniques for smart cities,'' *Future Gener. Comput. Syst.*, vol. 100, pp. 826–843, Nov. 2019.
[4] S. Nosratabadi, A. Mosavi, R. Keivani, S. Ardabili, and F. Aram, ''State of the art survey of deep learning and machine learning models for smart cities and urban sustainability,'' in *Proc. Int. Conf. Global Res. Educ.* Cham, Switzerland: Springer, 2020, pp. 228–238.
[5] Z. Ullah, F. Al-Turjman, L. Mostarda, and R. Gagliardi, ''Applications of artificial intelligence and machine learning in smart cities,'' *Comput. Commun.*, vol. 154, pp. 313–323, Mar. 2020.
[6] W. Hatzack and B. Nebel, ''The operational traffic control problem: Computational complexity and solutions,'' in *Proc. 6th Eur. Conf. Planning*, 2014, pp. 1–10.
[7] D. Renfrew and X.-H. Yu, ''Traffic signal optimization using ant colony algorithm,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Brisbane, QLD, Australia, 2012, pp. 1–7, doi: 10.1109/IJCNN.2012.6252852.
[8] S. Timotheou, C. G. Panayiotou, and M. M. Polycarpou, ''Towards distributed online cooperative traffic signal control using the cell transmission model,'' in *Proc. 16th IEEE Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 1737–1742.
[9] J. D. C. Little, ''The synchronization of traffic signals by mixed-integer linear programming,'' *Oper. Res.*, vol. 14, no. 4, pp. 568–594, Aug. 1966.
[10] N. H. Gartner, S. F. Assman, F. Lasaga, and D. L. Hou, ''A multi-band approach to arterial traffic signal optimization,'' *Transp. Res. B, Methodol.*, vol. 25, no. 1, pp. 55–74, 1991.
[11] T. Urbanik, A. Tanaka, B. Lozner, E. Lindstrom, K. Lee, S. Quayle, S. Beaird, S. Tsoi, P. Ryus, and D. Gettman,'' *Signal Timing Manual*, vol. 1. Washington, DC, USA: Transp. Res. Board, 2015.
[12] Y. Wang, X. Yang, H. Liang, and Y. Liu, ''A review of the self-adaptive traffic signal control system based on future traffic environment,'' *J. Adv. Transp.*, vol. 2018, pp. 1–12, Jun. 2018.
[13] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle, ''The scoot on-line traffic signal optimisation technique,'' *Traffic Eng. Control*, vol. 23, no. 4, pp. 1–10, 1982.
[14] 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.
[15] H. Zhang, C. Liu, W. Zhang, G. Zheng, and Y. Yu, ''GeneraLight: Improving environment generalization of traffic signal control via meta reinforcement learning,'' in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1783–1792.
[16] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, ''MetaLight: Value-based meta-reinforcement learning for traffic signal control,'' in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1153–1160.
[17] 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.
[18] M. Korecki and D. Helbing, ''Analytically guided reinforcement learning for green it and fluent traffic,'' *IEEE Access*, vol. 10, pp. 96348–96358, 2022.
[19] P. G. Balaji and D. Srinivasan, ''Multi-agent system in urban traffic signal control,'' *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 43–51, Nov. 2010.
[20] L. Busoniu, R. Babuska, and S. B. De, ''A comprehensive survey of multiagent reinforcement learning,'' *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Feb. 2008.
[21] M. Korecki, ''Adaptability and sustainability of machine learning approaches to traffic signal control,'' *Sci. Rep.*, vol. 12, no. 1, pp. 1–12, 2022.
[22] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, ''Multiagent reinforcement learning for urban traffic control using coordination graphs,'' in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science), W. Daelemans, B. Goethals, and K. Morik, Eds. Berlin, Germany: Springer, 2008, pp. 656–671.
[23] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, ''Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.
[24] M. Hardt, B. Recht, and Y. Singer, ''Train faster, generalize better: Stability of stochastic gradient descent,'' in *Proc. Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 1225–1234.
[25] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, ''A study on overfitting in deep reinforcement learning,'' 2018, *arXiv:1804.06893*.
[26] *Highway Capacity Manual*, Nat. Acad. Sci., Nat. Res. Council Publication, Washington, DC, USA, 2000, p. 1328.

[27] S. P. Venglar, P. Koonce, and T. Urbanik, II, "PASSER TM III-98 application and user's guide," Texas Transp. Inst., Texas A&M Univ., College Station, TX, USA, 1998.
[28] D. Husch and J. Albeck, "Trafficware synchro 6 user guide," TrafficWare, Albany, CA, USA, 2004.
[29] D. K. Hale, "Traffic network study tool," United States Version, McTrans Center, Univ. Florida, Gainesville, FL, USA, Tech. Rep. TRANSYT-7F, 2005.
[30] C. Gershenson, "Self-organizing traffic lights," 2004, *arXiv:nlin/0411066*.
[31] S. Lämmer and D. Helbing, "Self-stabilizing decentralized signal control of realistic, saturated network traffic," Santa Fe Inst., Santa Fe, NM, USA, 2010.
[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," IVT, ETH Zürich, Zürich, Switzerland, Tech. Rep., 2020, doi: 10.3929/ethz-b-000456701.
[34] R. Bellman, "A Markovian decision process," *J. Math. Mech.*, vol. 4, pp. 679–684, Jan. 1957.
[35] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[36] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[37] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Dec. 1989.
[38] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.
[39] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[40] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, and T. A. Lillicrap, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018.
[41] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 2074–2081.
[42] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.
[43] *Disaster Resilience: A National Imperative*, The National Academies Press, Washington, DC, USA, 2012.
[44] I. Linkov and B. D. Trump, *The Science and Practice of Resilience*. Cham, Switzerland: Springer, 2019.
[45] C. Poulin and M. B. Kane, "Infrastructure resilience curves: Performance measures and summary metrics," *Rel. Eng. Syst. Saf.*, vol. 216, Dec. 2021, Art. no. 107926.
[46] M. Akbarzadeh, S. Memarmontazerin, S. Derrible, and S. F. S. Reihani, "The role of travel demand and network centrality on the connectivity and resilience of an urban street system," *Transportation*, vol. 46, no. 4, pp. 1127–1141, Aug. 2019.
[47] I. G. Kasmalkar, K. A. Serafin, Y. Miao, I. A. Bick, L. Ortolano, D. Ouyang, and J. Suckale, "When floods hit the road: Resilience to flood-related traffic disruption in the San Francisco bay area and beyond," *Sci. Adv.*, vol. 6, no. 32, Aug. 2020, Art. no. eaba2423.
[48] A. A. Ganin, M. Kitsak, D. Marchese, J. M. Keisler, T. Seager, and I. Linkov, "Resilience and efficiency in transportation networks," *Sci. Adv.*, vol. 3, no. 12, 2017, Art. no. e1701079.
[49] (2021). *OpenStreetMap Contributors*. [Online]. Available: https://www.openstreetmap.org
[50] 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.*, 2019, pp. 3620–3624.
[51] J. Laval and H. Zhou, "Congested urban networks tend to be insensitive to signal settings: Implications for learning-based control," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24904–24917, Dec. 2022.
[52] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, no. 4, pp. 37–52, 1986.
[53] R. Fakoor, P. Chaudhari, S. Soatto, and J. A. Smola, "Meta-q-learning," 2019, *arXiv:1910.00125*.
[54] D. Helbing, "Economics 2.0: The natural step towards a self-regulating, participatory market society," *Evol. Institutional Econ. Rev.*, vol. 10, no. 1, pp. 3–41, Jun. 2013.
[55] D. Helbing, S. Lämmer, T. Seidel, P. Šeba, and T. Płatkowski, "Physics, stability, and dynamics of supply networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, Dec. 2004, Art. no. 066116.
[56] S. Lämmer, H. Kori, K. Peters, and D. Helbing, "Decentralised control of material or traffic flows in networks using phase-synchronisation," *Phys. A, Stat. Mech. Appl.*, vol. 363, no. 1, pp. 39–47, 2006.
[57] T. Seidel, J. Hartwig, L. Richard Sanders, and D. Helbing, *An Agent-Based Approach to Self-organized Production*. Berlin, Germany: Springer, 2008.

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 the philosophy of AI.

DAMIAN DAILISAN received the B.S. degree in applied physics and the M.S. and Ph.D. degrees in physics from the University of the Philippines Diliman, in 2015, 2017, and 2020, respectively. He is currently a Postdoctoral Researcher with ETH Zürich. His research interests include phase transitions in agent-based cellular automata traffic models, data science, and deep RL.

DIRK HELBING received the Ph.D. degree (Hons.) from the Delft University of Technology (TU Delft), in January 2014. In June 2015, he was with the Faculty of Technology, Policy, and Management, TU Delft, for some years, where he led the Ph.D. School in "Engineering Social Technologies for a Responsible Digital Future." He is currently a Professor of computational social science with the Department of Humanities, Social and Political Sciences and the Department of Computer Science, ETH Zürich, and the ETH AI Center. He is also a member of the Complexity Science Hub Vienna.