Application of Multi-Agent Deep Reinforcement Learning to Optimize Real-World Traffic Signal Controls

Maxim Friesen, Tian Tan, Jürgen Jasperneite, *Senior Member, IEEE*, and Jie Wang

Abstract—Increasing traffic congestion leads to significant costs associated by additional travel delays, whereby poorly configured signaled intersections are a common bottleneck and root cause. Traditional traffic signal control (TSC) systems employ rule-based or heuristic methods to decide signal timings, while adaptive TSC solutions utilize a traffic-actuated control logic to increase their adaptability to real-time traffic changes. However, such systems are expensive to deploy and are often not flexible enough to adequately adapt to the volatility of today’s traffic dynamics. More recently, this problem became a frontier topic in the domain of deep reinforcement learning (DRL) and enabled the development of multi-agent DRL approaches that could operate in environments with several agents present, such as traffic systems with multiple signaled intersections. However, most of these proposed approaches were validated using artificial traffic grids. This paper therefore presents a case study, where real-world traffic data from the town of Lemgo in Germany is used to create a realistic road model within VISSIM. A multi-agent DRL setup, comprising multiple independent deep Q-networks, is applied to the simulated traffic network. Traditional rule-based signal controls, currently employed in the real world at the studied intersections, are integrated in the traffic model with LISA+ and serve as a performance baseline. Our performance evaluation indicates a significant reduction of traffic congestion when using the RL-based signal control policy over the conventional TSC approach in LISA+. Consequently, this paper reinforces the applicability of RL concepts in the domain of TSC engineering by employing a highly realistic traffic model.

Index Terms—Deep Reinforcement Learning, Traffic Signal Control, Deep Q-Network, Multi-Agent, VISSIM, LISA+

I. INTRODUCTION

The development and expansion of transportation systems is linked to a growing economy and public welfare, as transport supports and facilitates these by means of delivering accessibility [1]. Unfortunately, mobility has evolved to such extents that it started to generate negative effects, such as congestion and environmental pollution. Even small and medium-sized cities experience unprecedented congestion problems due to an immense rise in road users that quickly outpace infrastructural expansion. Signaled intersections have been identified as

common bottlenecks [2] that both researchers and the industry have been trying to improve by introducing new signal control approaches to maximize the performance of Traffic Signal Control (TSC) systems.

Conventional traffic signal controls utilize rule-based methods, such as fixed-time schemes or heuristic approaches and often have to be manually tuned. Adaptive Traffic Signal Controls (ATSC) employ detectors to measure the real-time traffic demand and implement more responsive and dynamic control strategies. Commercial ATSC systems provide many different tools to design, evaluate and deploy solutions for optimizing control policies and signal coordination. They are however expensive to employ and maintain, while requiring experienced traffic engineers to operate, configure and tune the systems for optimal performance.

In recent years, Reinforcement Learning (RL) emerged as a promising alternative to developing ATSCs based on real-world traffic measurements. RL does not rely on rule-based policies, heuristic assumptions or human experience such as conventional traffic control approaches. Instead, it autonomously uses its own experience from interacting with the environment to maintain a parametric model that learns to optimally control complex traffic systems. Traditional RL approaches lack the scalability and optimality to handle large and complex state-action spaces in practice, as it is the case with most traffic grid environments. New developments in Deep Reinforcement Learning (DRL) can cope with such tasks and increase the learning capacities for more complex applications by employing neural networks.

A. Motivation

Intelligent transportation systems are no longer exclusive to metropolitan areas and big cities. Small and medium-sized cities can similarly benefit from digital, intelligent solutions, as shown through various initiatives in the old hanseatic town of Lemgo, Germany. Driven by the Institute for Industrial IT (inIT) at the Technische Hochschule Ostwestfalen-Lippe (TH OWL) and the Fraunhofer IOSB-INA, various smart city solutions are being piloted in Lemgo as part of research projects and a living lab approach [3], [4]. By deploying a variety of traffic related IoT technologies, such as vehicle counting sensors, the main arterial roads in Lemgo have been identified to exhibit traffic congestion issues during peak-hour traffic. Developing new approaches to mitigate these problems have therefore been a topic of high interest and

Maxim Friesen is with the Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe, 32657 Lemgo, Germany. Tian Tan and Jie Wang are with the Department of Civil and Environmental Engineering, Stanford University, CA 94305, USA. Jürgen Jasperneite is with the Fraunhofer IOSB-INA, 32657 Lemgo, Germany. This work is funded by the Ministerium für Wirtschaft, Innovation, Digitalisierung und Energie des Landes Nordrhein-Westfalen (MWIDE NRW) under ref. no. 34.03.08-004/2018-001. Email: maxim.friesen@th-owl.de, tiantan@stanford.edu, juergen.jasperneite@iosb-ina.fraunhofer.de, jjewang@stanford.edu.

importance. Among local municipalities and other industrial partners, the private company Stührenberg GmbH [5] is an active consortium member in one of these projects. They develop state-of-the-art road traffic technology, such as traffic signal controllers, which are employed at all major signaled intersections in Lemgo. Through their partnership, the internal control logic of individual TSCs at key intersections in Lemgo are made accessible for research purposes.

DRL in the context of TSCs has become a promising new approach to improve signal controls in a cost-efficient manner. Initially it was focused on optimizing single intersections however, real-world applications in traffic grid environments typically involve multiple interconnected junctions. Thus, multi-agent DRL approaches had to be considered when trying to apply reinforcement learning to real-world traffic grid scenarios. There has already been extensive research in the area of large scale AI-based TSCs [6]–[9]. DRL algorithms, such as Advantage Actor Critic (A2C) and Deep Q-Networks (DQN), were adopted to develop multi-agent frameworks for the application in ATSCs. However, the main focus in these studies typically went into re-approaching algorithmic problems to overcome previous shortcomings. The simulated traffic grids that were employed to validate these approaches were typically based on over-simplified grid layouts and lack traffic data obtained from the real world. While these artificial road and traffic models sufficiently prove the validity of their basic assumptions, it is uncertain whether and how well the proposed DRL agents would function in the real world. To further extend the applicability of such approaches beyond these idealized road models, a more realistic simulated environment is required that introduces additional complexity to the state-action space by employing a heterogeneous road network with more realistic traffic volumes and driver behaviors. Since Lemgo is already offering an extensive city-wide network of IoT sensors for capturing traffic state information, and provides access to the existing signal control infrastructure through consortium partners, the town was identified as an excellent case study location to benchmark the applicability of multi-agent DRL approaches in a close to real-world environment.

B. Contribution

This paper presents a case study in which the TSC performance of Multi-Agent Independent DQN (MA-IDQN) is evaluated against a classical rule-based approach with a simulated real-world traffic network. Two interconnected intersections in the town of Lemgo in Germany are modeled in the VISSIM traffic simulator. The studied route is pictured in Figure 4, with $N1$ and $N2$ denoting the modeled nodes. The existing signal head configuration is recreated and its control logic is trained by employing MA-IDQN. The currently employed rule-based signal controls used by the city are provided in LISA+, which is a traffic engineering software developed by the German company Schlothauer & Wauer [10]. It facilitates the development, simulation and evaluation of traffic-actuated signal controls and allows exporting them into industry standard formats for real-world or simulator integration. The key contributions of this work can be summarized as follows:

- A realistic simulation environment of two intersections in the town of Lemgo was modeled by employing real-world traffic data.
- Traffic signal controls within the model were trained with MA-IDQN.
- Real-world traffic signal controls were incorporated into the model as a performance baseline.
- The RL-based MA-IDQN controls were evaluated against the rule-based real-world signal controls as part of multiple numerical experiments conducted within the simulation model.

We are therefore the first to compare RL-based solutions with real-world traffic signal controls. Existing work tends to only use over-simplified baselines, e.g. longest-queue-first or rule-based controls to compare with RL solutions. However, that is not fair as we first have to investigate the potential performance benefits of RL methods against real-world signal controls currently in use. Thus this work offers a unique case study that can provide a deeper insight into the applicability of RL in the area of TSC.

C. Structure

This paper is structured as follows. Section II provides preliminary information about traffic congestion in general, discusses its root causes and mitigation strategies and describes the traffic situation along the studied route in the town of Lemgo. Subsequently, the research approach in this paper is distinguished from other related work. Section III introduces relevant machine learning approaches that are used in this paper and discusses their application in the context of traffic signal controls. Section IV first presents an overview over the used simulation and modeling tools, continues to define relevant terms and designations and describes the modeling and validation processes of the simulated road network by aggregating real-world traffic information. Section V elaborates the implementation and setup of the MA-IDQN agents including the definition of the state, action and reward functions. Section VI-A further details the simulator and agent configurations along with their parametrization during training. The evaluation setup, execution and results are presented and interpreted in section . Lastly, Section VIII summarizes this paper's research approach and concludes its findings.

II. BACKGROUND

According to the US Federal Highway Administration (FHWA), traffic congestion relates to an excess of automobiles on a segment of roadway at a certain time that result in slower than normal speeds [2]. While there is no global consensus, the most common definition for traffic congestion is a state of traffic where the travel demand exceeds the road capacity [11]. Other definitions include the travel-delay perspective, where congestion occurs when high vehicle densities disturb normal traffic flow and result in excess travel time delays [2]. The FHWA categorizes the seven root causes of traffic congestion into traffic incidents, work zones, bad weather conditions, fluctuations in normal traffic, large gatherings due to special events, physical bottlenecks and traffic control operations such

as signal lights [2]. Congestion often results from more than one of these root causes. Some events can cause others to occur, e.g. high congestion levels or bad weather can lead to a higher chance of traffic accidents. As a result, treating individual causes can alleviate or prevent the effects of others. However, most of these causes occur irregularly and are hard to avoid or predict. Traffic-influencing events such as traffic accidents are almost entirely unpredictable. Bad weather can be predicted to a certain extent but not avoided. Work zones are necessary for infrastructure maintenance and expansion. Traffic demand fluctuations on a day-to-day basis, or in the case of special events can be predicted, but require additional traffic flow management measures in place to balance the traffic load across alternative routes [12]. This in turn requires additional control infrastructure that is able to reroute traffic dynamically. Removing physical bottlenecks involves modifying or expanding the existing road infrastructure, which comes at great financial costs and is often restricted by dense urban cores. This typically only leaves traffic control operations such as signal lights with additional room for improvements.

While inefficient signal controls only cause around 5% of traffic congestion, they also represent one of the few causes that can be optimized in a cost-effective manner [2], [13]. In the relevant example of Lemgo, there is no simple option of infrastructure expansion or potential for advanced traffic management strategies, as the congested route in question lies in a densely urbanized environment and offers no immediate alternative routes for traffic load balancing approaches. Since traffic jams are mostly observed at the signaled intersections $N1$ and $N2$ and there are no further road features that may severely restrict the traffic flow, these congested intersections are assumed to fall under the cause of both poorly timed signal controls and physical bottlenecks. These nodes and their signal controls therefore represent an important research subject when looking for ways to alleviate traffic congestion along that route.

A. Related Work

To the best of our knowledge there are no other studies where VISSIM and LISA+ were utilized to evaluate the performance of trained DRL agents against rule-based traffic signal controls employed in the real world. Qadri et al. [14] provide a comprehensive survey on recent publications utilizing traffic simulators in the context of AI-controlled TSCs. Few of the listed works that employ VISSIM are actually studying DRL approaches, and instead focus on genetic algorithms, dynamic programming or particle swarm optimization. Moreover, the road models in VISSIM or other simulators such as SUMO are either used to validate and test the performance of different algorithmic approaches and hyperparameter configurations, or they are compared to signal controls generated in TSC optimization tools similar to LISA+, such as SYNCHRO [15] or VISTRO [16]. However, unlike in this work, they were not previously deployed in the real world, but instead specifically generated for the purpose of their research.

LemgoRL [17] was introduced as a benchmark tool in a related research context as this paper. Node $N2$ was modeled

in SUMO and the Lemgo road network was validated using similarly obtained data sets. The approach intends to provide a benchmark environment to test and evaluate a variety of RL algorithms in a realistic simulation environment. However, it focuses only on a single intersection and utilizes LISA+ to integrate a logic unit that ensures compliance with safety and regulatory requirements, instead of implementing the real-world TSC for direct performance comparison.

III. MACHINE LEARNING FOR TRAFFIC SIGNAL CONTROL

Fixed-time traffic signal controls, such as TRANSYT [18], have traditionally been implemented by using rule-based policies that are derived from heuristics or expert experience. More sophisticated adaptive strategies, such as SCOOT [19] or SCATS [20], use decentralized approaches that solve optimization problems locally or globally when coordinating multiple intersections. These methods view traffic networks as uncertain systems, where dynamic optimization is realized by adjusting signal timings and sequences on the basis of measured state variables as feedback. These include delay time, queue length or traffic flow and are used to address the stochastic characteristics of the transportation system. More recently, new unconventional approaches based on AI started to emerge. They include the use of fuzzy logic [21], [22], evolutionary algorithms [23], [24], neural networks [25], [26] and swarm intelligence [16]. RL is one major subfield of machine learning and has become increasingly relevant for ATSC in recent years [27].

A. Q-Learning

In RL, learners gather state information from an environment in which they execute actions to maximize their cumulative rewards. Environments are modeled as discrete-time Markov Decision Processes (MDPs) [28] and are described by states, actions, transition probabilities and rewards. A policy $\pi(a_t|s_t)$ maps states $s \in S$ to the probabilities of selecting individual actions $a \in A(s)$ at time t . The goal is finding an optimal policy that maximizes the expected reward over the planning horizon T . The state value function $v_\pi(s_t)$ for all $s \in S$ under a policy π at time t is defined by

$$v_\pi(s_t) = \mathbb{E}_\pi \left[\sum_{k=0}^T \gamma^k r_{t+k+1} \mid s_t \right], \quad (1)$$

where $\mathbb{E}_\pi[\cdot]$ is the expected value for the given policy and state. Similarly, the state-action value function $q_\pi(s_t, a_t)$ for all $s \in S$ and $a \in A(s)$ under policy π at time t is defined by

$$q_\pi(s_t, a_t) = \mathbb{E}_\pi \left[\sum_{k=0}^T \gamma^k r_{t+k+1} \mid s_t, a_t \right]. \quad (2)$$

An optimal policy $\pi_*(s) = \operatorname{argmax}_a q_*(s, a)$ maximizes the state-value function $v_*(s_t) \doteq \max_{\pi} v_\pi(s)$ and action-value function $q_*(s_t, a_t) \doteq \max_{\pi} q_\pi(s_t, a_t)$ for all $s \in S$ and $a \in A(s)$ at time t . The optimal Q-function q_* must further satisfy [29]:

$$q_*(s_t, a_t) = \mathbb{E} \left[r_{t+1} + \gamma \max_{a_{t+1}} q_*(s_{t+1}, a_{t+1}) \right], \quad (3)$$

where action a_{t+1} for the next state s_{t+1} maximizes the expected reward when following the optimal policy.

Q-learning is one of the most widely employed RL approaches [30]. It iteratively updates the Q-value for each state-action pair according to:

$$\underbrace{Q_\pi(s_t, a_t)}_{\text{new Q-value}} \leftarrow (1 - \alpha) \underbrace{Q_\pi(s_t, a_t)}_{\text{old Q-value}} + \alpha \underbrace{\left(r_{t+1} + \gamma \max_{a_{t+1}} Q_\pi(s_{t+1}, a_{t+1}) \right)}_{\text{learned Q-value}}, \quad (4)$$

with $0 < \alpha < 1$ and $0 \leq \gamma < 1$, until the function converges to the optimal Q-function q_* which satisfies Eq. 3, whereby α represents the learning rate, while γ is the discount factor.

B. Deep Q-Networks

Since Q-values are iteratively updated with each time step, the amount of time required to converge to an optimal Q-function q_* increases with the number of states to traverse. To efficiently apply Q-learning in large state-action spaces, the optimal Q-function can be approximated by a Deep Neural Network (DNN) resulting in a Deep Q-Network [31] that estimates Q-values based on a policy π_θ . The mean squared error of the network's predicted Q-value $Q(s_t, a_t)$ and the target Q-value $r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ is computed as the loss function. The network's weights and parameters θ are updated periodically through stochastic gradient descent and back-propagation with the objective of minimizing the loss function and eventually approaching the optimal Q-function q_* and optimal policy π_* .

C. Multi-Agent Independent Deep-Q Networks

Traffic-grid signal controls and similar applications are multi-agent by nature, as there are several distributed agents that have to perform independent decisions based on local state observations to optimize a global overall goal, e.g. reducing congestion in the traffic grid. These kind of use cases are addressed with Multi-Agent Reinforcement Learning (MARL) frameworks [32]. One popular MARL approach is Independent Q-Learning (IQL). It extends Q-learning with multi-agent settings [33], where each agent learns an independent Q-function and treats other agents as part of the environment. When complemented with DQNs, each agent independently trains a Q-network [34].

IV. SIMULATION MODEL DESIGN

A. Evaluation of Traffic Simulators

With the abundance of computational power, many complex microscopic simulation models have been developed in the past decades. Qadri et al. [14] provide a comprehensive literature review of currently relevant simulation tools in the context

Fig. 1: Number of recent TSC related publications using micro-simulation tools [14].

of traffic signal controls. Figure 1 illustrates the number of recent publications that utilized certain tools. It shows that VISSIM [35], SUMO [36], PARAMICS [37], CORSIM [38] and AIMSUN [39] represent the most referenced micro-simulation tools in recent literature. Other works also find these tools to be the most popular and widely established solutions [40], [41]. To find the most suitable traffic simulator for this study, a set of requirements is defined below and subsequently used for comparing the available tools.

- *Country* – The country of origin should preferably be Germany, since the case study is based there too. Simulators developed in Germany are more likely to provide the required tool set to realistically model the network geometries and traffic conditions found in the town of Lemgo.
- *System* – A space-continuous underlying system is preferred over a discrete approach, as the increased runtime performance and computational efficiency achieved with discrete approaches is just a secondary concern. Increased granularity of the simulated traffic and measured output of the signal control performance is of higher priority.
- *Open* – An open source traffic simulator is preferred over commercial products as its source code is adaptable and free to use.
- *Interface* – The availability of interfaces that allow the execution of external code is essential. This includes the ability to control the simulation parameters remotely and retrieve or change state information in the simulation. Detector data has to be retrieved and signal states have to be changed during runtime.
- *Python* – The available interfaces should use libraries or APIs compatible with Python, as the MA-IDQN implementation is developed in it.
- *LISA* – The LISA+ virtual signal controller that is executing the real-world signal controls has to be linked with the simulated signal network infrastructure through necessary available plugins.
- *GUI* – The availability of a GUI simplifies the modeling process.

	AIMSUN [39]	CORSIM [38]	PARAMICS [37]	SUMO [36]	VISSIM [35]
Country	USA	USA	UK	GER	GER
System	C	D	D	C	C
Open	×	×	×	✓	×
Interface	✓	✓	✓	✓	✓
Python	✓	×	×	✓	✓
LISA+	✓	×	×	✓	✓
GUI	✓	✓	✓	✓	✓

System: C = Space-Continuous; D = Space-Discrete

TABLE I: Comparison of the most commonly used micro-simulation tools.

B. Tool Selection

Table I provides an overview of the relevant tools with respect to the defined requirements. Based on the results, SUMO and VISSIM were singled out as the most suitable solutions for our use case, which is further corroborated by Figure 1, highlighting their frequent use in related work.

VISSIM is a multi-modal commercial traffic simulator, developed by PTV (Planung Transport Verkehr) in Germany [35] and is a global market leader for commercial traffic flow simulation software. It belongs to the most comprehensive traffic simulators in the world, with a variety of sophisticated road and vehicle modeling tools, enabling the creation of highly realistic traffic models. SUMO (Simulation of Urban Mobility) [36] on the other hand is a community-driven, open-source traffic simulator created by the German Aerospace Center to facilitate traffic modeling approaches in research studies.

A recent study by Liu et al. [27] indicates that most DRL related research utilizes SUMO. However, one of the key objectives in this study was the creation of a highly realistic road and traffic model, whereas much of the referenced research work use over-simplified modeling approaches. Industrial providers of traffic control solutions, such as our partner Stührenberg who developed and provided the real-world traffic signal controls, typically employ VISSIM, as it provides a wide set of integrated tools to generate fine-grained models, covering all aspects of multi-modal traffic, network geometries and sophisticated signal controls. Moreover, it offers a streamlined user experience that merges all tools and functions into a single interface, while SUMO's simulator, network editor and other features are separate modules. Since we closely worked with industrial partners and the work in this paper was strongly oriented towards practical applications, VISSIM was consequently deemed to be the best option for creating a traffic model that would be the most representative of the real world.

C. Integration of Real-World Signal Controls

The TSCs that are employed at the studied intersections were modeled in LISA+ [10]. LISA+ provides full compatibility to VISSIM by compiling the signal control logic into a

Fig. 2: Exemplary depiction of node $N1$ for the definition of relevant terms.

standalone server. The server maintains a virtual LISA+ traffic signal controller as a RESTful service that is accessed by VISSIM during the simulation. At runtime, VISSIM simulates its traffic model and sends all vehicle triggered detector data to the LISA+ virtual controller. The controller simulates the exported control logic and processes the detector data to calculate the corresponding signal switch commands for each time step. The LISA+ signal control plans for both nodes are not interconnected and therefore run autonomously.

D. Definition of Terms

Throughout the following sections certain terms are repeatedly used when referring to TSC related topics. The following list and Figure 2 provide the respective term definitions:

- **Node** – A node is an intersection where multiple roadways meet.
- **Link** – A roadway meeting with a node or interconnecting two nodes with each other is referred to as a link.
- **Lane** – A link consists of a set of lanes. This set can contain a single or multiple lanes. In this paper, lanes are denoted by the cardinal direction their link is oriented towards. Lanes originating from or exiting towards the northern direction are labeled n . Similarly, eastern, southern and western lanes are indicated by e , s and w . All incoming lanes are additionally numbered clock-wise beginning at the northern lane with n_1 .
- **Traffic Movement** – The direction vehicles take when entering and exiting an intersection is denoted by the incoming and outgoing lanes, e.g. $n_2 \rightarrow e_o$.
- **Phase** – Describes the state of signal heads that are mapped onto traffic movement. In this paper, signals for green and red are denoted by G_l and R_l , where l is the lane number affected by the respective movement signal.
- **Stage** – Denotes a combination of phases across some or all signaled incoming lanes for a specific time period that have no conflicting traffic movements. In Figure 2, the depicted exemplary stage is $\{G_1, G_2, G_3, R_4, R_5, R_6, R_7, R_8\}$.

Fig. 3: Modeled road network in VISSIM.

- *Signal Plan* – Specifies a stage sequence including details about the individual stages and their duration.
- *Interstage interval* – Interstages describe the time between the end and start of two stages. The yellow and all-red times are referred to as interstage intervals and allow vehicles to stop or safely pass the intersection shortly before or during a stage transition.
- *Cycle Time* – The cycle time specifies the duration of a stage or a sequence of stages.

E. Model Design

1) *Road Network*: Figure 3 shows the modeled road network in VISSIM. Minor access roads were disregarded as there was no traffic data for incoming and outgoing vehicles available. From real-world observations and measurements, the traffic volumes on these side roads were found to be negligible when compared to the vehicle flow along the main route. The entirety of the modeled road network is located within city bounds. As per the German traffic code, all lanes have therefore been assigned a maximum vehicle speed of 50 km/h. Additionally, reduced speed zones were placed on turns in both nodes to reflect a more realistic driving behavior within intersections.

2) *Signal Head & Detector Placement*: To enable separate signal controls for individual lanes, e.g. for the case of green arrows that allow single lanes to turn right or left, each lane was modeled as a separate link. Consequently, each lane at both nodes was assigned a separate signal head that can be separately actuated, either by MA-IDQN or the LISA+ controls. Vehicle detectors were placed according to the original signal plan layouts.

3) *Vehicle Volumes*: Only nodes *N1* and *N2*, their incoming links and their interconnecting link are relevant for this study. Consequently, six vehicle inputs had to be defined to generate traffic flow on all incoming node links. Figure 4 illustrates the placement of the vehicle inputs *V1* to *V6*. The link that interconnects nodes *N1* and *N2* does not need a vehicle input as it is saturated by traffic coming from either of the other six vehicle inputs.

The traffic density was modeled after the peak-hour volumes when the network is most saturated and prone to congestion. To identify the real-world peak-hour traffic volume, image and radar-based vehicle counting sensors were utilized. These were installed along various incoming links at the studied nodes. A one week interval limited to 5 business days was chosen to identify the highest average hourly vehicle volume at each link. Once the peak-hour was identified to be between

Fig. 4: Nodes under study, vehicle input placements and specified evaluation routes in the town of Lemgo.

07:00-08:00 AM, the hourly vehicle volume for each vehicle input was specified based on previous dedicated measurements conducted at each intersection. These include manual measurements and the deployment of a video collection unit that overlooks intersections and gathers data about vehicle quantities, types and their movement direction. Lastly, the average hourly vehicle volume during business days between 07:00-08:00 AM was calculated across a one month period and is listed in Table II. The vehicle spawn times in VISSIM are defined stochastically. The specification of an hourly volume results in average time gaps between spawning vehicles. These time gaps are derived from the Poisson distribution.

An average vehicle composition was derived by utilizing the information gathered from the video measurements. Excluding bicycles and motorcycles, the vehicle composition for the vehicle inputs was defined as follows: 96.99% cars, 1.17% trucks and 1.84% buses. The default vehicle lengths provided by VISSIM for the individual vehicle types were kept, as they are representative of common vehicle models in Germany, namely: 3,75m to 4,76m for cars, 10,22m for trucks and 12,4m for buses.

4) *Vehicle Routing & Behavior*: VISSIM utilizes vehicle routing rates to determine how vehicles behave when entering a junction or approaching an exit. These routing rates follow a random distribution and determine the probability of a vehicle moving from one link to another. Based on the previous measurements conducted at both nodes, the turning rates for all legal direction changes were calculated as per Tables III and IV.

VISSIM employs the psycho-spacing car-following models by Wiedemann [42] and the lane-changing model by Sparmann [43]. Since peak-hour traffic situations are the focus in this study, aggressive driving such as speeding or running red lights were disregarded. Vehicles were set to comply with the traffic laws, while the lane-changing aggressiveness was left on default. Thus, drivers comply with the speed limits, but tend to drive partially above or below the limit to reflect a more realistic traffic dynamic.

Vehicle Input	V1	V2	V3	V4	V5	V6
Vehicles / Hour	696	501	412	500	71	119

TABLE II: Used hourly vehicle volumes for the vehicle inputs in the simulation model.

	From North	From South	From West	From East
To East	41.51 %	16.81 %	49.47 %	-
To West	42.09 %	31.09 %	-	52 %
To South	16.4 %	-	7.83 %	0 %
To North	-	52.1 %	42.7 %	48 %

TABLE III: Vehicle turning rates at node $N1$.

	From North	From South	From West	From East
To East	41.21 %	38.03 %	82.57 %	-
To West	58.13 %	19.72 %	-	90.31 %
To South	0.66 %	-	0.82 %	0.96 %
To North	-	42.25 %	16.61 %	8.73 %

TABLE IV: Vehicle turning rates at node $N2$.

F. Limitations

The modeled road network does not include pedestrians, bicycles, motorcycles, emergency vehicles, traffic incidents, weather effects or other traffic influencing factors due to a lack of real-world data for the corresponding design and validation.

V. MA-IDQN IMPLEMENTATION

For the employed MA-IDQN setup the individual agents and their value networks are represented by standard DQNs. The used DNNs are regular feed-forward networks, where the input and output layers are formed by the dimensions of the state and action spaces respectively. For the hidden layers, the rectified linear activation function (ReLU) [44] is used while the network training is executed with the Adam optimizer [45]. The algorithms were implemented in Python relying on TensorFlow libraries [46]. To interconnect the IDQN agents with the VISSIM runtime environment a tailored interface was developed by interacting with VISSIM's Component Object Model (COM) API.

A. State Definition

The state representation in this study relies on vehicle queue lengths. Modern DRL approaches for ATSC applications commonly utilize queue lengths for their state representation. Various studies have shown that queue length is considered to be a superior state definition compared to other measures [47]–[49]. Authors experimenting with alternatives typically addressed the problems of partial observability. Measuring certain metrics reliably in the real world is much more challenging than in a simulation, where full observability is always given. For the sake of simplicity, full observability is assumed in this study. If A_{N_j} is a set of all incoming lanes l_i at node N_j then the local states for agent j at time t are given by

	Actions	Dimension
Node $N1$ / Agent 1	$a_1 = \{G_1, G_2, R_3, R_4, G_5, G_6, R_7, R_8\}$ $a_2 = \{R_1, R_2, G_3, G_4, R_5, R_6, G_7, G_8\}$ $a_3 = \{G_1, G_2, R_3, R_4, R_5, R_6, R_7, R_8\}$ $a_4 = \{R_1, R_2, G_3, G_4, R_5, R_6, R_7, R_8\}$ $a_5 = \{R_1, R_2, R_3, R_4, G_5, G_6, R_7, R_8\}$ $a_6 = \{R_1, R_2, R_3, R_4, R_5, R_6, G_7, G_8\}$ $a_7 = \{G_1, G_2, G_3, R_4, R_5, R_6, R_7, R_8\}$	7
Node $N2$ / Agent 2	$a_1 = \{G_1, G_2, R_3, R_4, G_5, G_6, R_7, R_8\}$ $a_2 = \{R_1, R_2, G_3, G_4, R_5, R_6, G_7, G_8\}$ $a_3 = \{G_1, G_2, R_3, R_4, R_5, R_6, R_7, R_8\}$ $a_4 = \{R_1, R_2, G_3, G_4, R_5, R_6, R_7, R_8\}$ $a_5 = \{R_1, R_2, R_3, R_4, G_5, G_6, R_7, R_8\}$ $a_6 = \{R_1, R_2, R_3, R_4, R_5, R_6, G_7, G_8\}$	6

TABLE V: Defined action space for nodes $N1$ and $N2$.

$$s_{t,j} = \{ \rho_{l_i}(t) \}_{l_i \in A_{N_j}}, \quad (5)$$

where $\rho_{l_i}(t)$ is the queue length function that gives the the total length of queued vehicles in lane l_i at time t . It is given in meters and measures the length of all queued vehicles including the gap between them. A vehicle is considered to be in a queue if it is less than 500m from the stop line and its speed is below 15 km/h. To limit the state space and improve training efficiency, the queue length for each lane is normalized to an interval of $[0, 1]$ and rounded up to 2 decimal places. The normalization denominator was set to 5 and was empirically derived from the simulated average vehicle lengths and vehicle gaps in meters. Values outside the specified interval were clipped. Each node in this study has 8 incoming lanes on which the queue length is measured. This results in a local state dimension of 8 for each node.

B. Action Definition

The action space for each agent was defined with a set that contains all legal stage combinations. They were selected based on the physically available signal head setup at the corresponding real-world intersections. For example, only a single right-turn signal head is present at node $N1$ that can independently switch to green (action a_7 as per Table V, refer to Figure 6). As a result, there were no stages defined where right-turn lanes might receive a green signal, while the neighboring lanes stay red. This approach intentionally limits an agent's action space and possibly diminishes training efficiency. However, by doing so, it allows comparing the signaling performance of MA-IDQN to conventional traffic controls using LISA+. Only when both evaluated TSC approaches have the same underlying signal head configuration to choose their stage combinations from, a fair comparison can be carried out. Thus, the actions were defined as shown in Table V. Each action is a stage that contains the phases for all signal heads at the corresponding node. The stage assignment always starts north and proceeds clockwise lane by lane. Figure 6 and 5 exemplary illustrate the stage assignment for actions a_7 and a_2 at nodes $N1$ and $N2$ respectively.

Fig. 5: Exemplary stage assignment of action a_2 at node N_2 as per Table V (taken from VISSIM network editor).

Fig. 6: Exemplary stage assignment of action a_7 at node N_1 as per Table V (taken from VISSIM network editor).

C. Reward Definition

Similar to the state representation, the vehicle queue length was selected to be integrated with the reward function. If A_{N_j} is a set of all incoming lanes l_i at node N_j then the local reward for agent j at time t is given by

$$r_{t,j} = \sum_{l_i \in A_{N_j}} \rho_{l_i}(t), \quad (6)$$

where $\rho_{l_i}(t)$ is the queue length function that gives the total length of queued vehicles in lane l_i at time t . The queue lengths for all lanes $l_i \in A_{N_j}$ are summed, negated, normalized and clipped to fit into an interval of $[-100, 0]$, whereas the normalization denominator was chosen similar to how it was selected for the state representation.

D. Agent Cooperation

MA-IDQN solves the scalability issues of centralized learning but leads to non-stationary environments from an agent's point of view. With only partial observability between agents, they formulate each others behavior in the environment as part of the entire system's dynamics. This becomes an issue when the concept of experience replay [50] is combined with multi-agent approaches. A learning agent's replay buffer may become obsolete if it can no longer correctly represent the current environment's dynamics. One solution for this problem is limiting the replay buffer to very short and recent batches [51], which in turn results in a lower sample efficiency and agent stability. Alternatively, cooperation may be introduced by sharing state information and reward functions between agents [6], [7]. By augmenting the state space of all other agents, the learning problem can be reduced to a regular single-agent problem, albeit in a larger environment. In this paper, a simplified approach of the state and reward sharing strategy proposed by Tianshu et al. [6] was employed. Figure 7 illustrates the resulting architecture for our studied road model. Following the suggested approach, nodes N_j and N_k are neighbors if they are connected by a link. Each node is controlled by an independent agent that maintains its own value network Q_θ . To introduce cooperation, the cooperative state and reward for agent j are given by $\tilde{s}_{t,j} = s_{t,j} + \omega s_{t,k}$ and $\tilde{r}_{t,j} = r_{t,j} + \omega r_{t,k}$ respectively, where $0 \leq \omega \leq 1$ is the cooperative discount factor that weakens the neighboring agent's state and reward signals.

VI. PARAMETRIZATION & TRAINING

A global reward was specified as a benchmark value to indicate the training efficiency across the entire road network during hyperparameter tuning. For that purpose, the local reward functions of both nodes were summed as indicated in Figure 7. By evaluating the convergence of the average global reward from both agents, the following parameter configuration was derived.

A. Simulator Configuration

The episode length was set to 800 simulation seconds. The episode length was deemed to be short enough to get reasonable training times and long enough to generate representative congestion levels. The control interval, which determines how often the signal stages are updated and consequently how often the simulation environment interacts with the IDQN agents, was set to 10 simulation seconds. The LISA+ signal controls employed at both nodes exhibit a minimum cycle time of 10 seconds, thus a similar minimum was selected for the RL-based controls. The green-yellow and all-red interstage intervals were both set to 3 simulation seconds in compliance with the German federal traffic code that prescribes a 3 second minimum interval if the speed limit is 50 km/h [52]. All possible signal stages were specified as per Table V. The simulation seed was incremented for subsequent episodes and reset when training was restarted to ensure comparability between training runs with different hyperparameter configurations.

Fig. 7: MA-IDQN architecture during training when employing the cooperative discount factor for state and reward sharing between agents.

B. Agent Configuration

The training was conducted for 75,000 steps, with a replay buffer of 15,000 samples and a mini batch size of 64 samples. One step corresponds to 10 seconds simulated time in VISSIM. The value network was updated every 4 steps and the target network was synchronized every 500 steps. The DNN was configured to have 2 layers with 256 neurons each, and 1 layer with 128 neurons. A constant learning rate of $1e-5$ was employed, while the discount factor was set to 0.99. For the employed ϵ -greedy policy a start value of 1.0, a stop value of 0.01 and a linear decay of size 0.0015 was selected.

C. Cooperative Discount

State and reward sharing was suggested to mitigate the effects of a non-stationary environment and improve the learning performance of the multi-agent setup. The resulting architecture for a two node setup is illustrated in Figure 7. Figure 8 further displays the observed average global reward taken from both agents during training when using different cooperative discount factors. A convergence towards 0 implies reduced global queue lengths and improved learning performance. It is evident that no cooperation between agents yielded the best results. A value of $\omega = 1$ led to divergence which was expected, as the cooperative discount factor was intended to weaken the neighboring signals and provide more stability. However, even with a discount factor of 0.5 and 0.75, the training efficiency was degraded and convergence impeded. This could be attributed to the scale of the simulation model. Since only two neighboring intersections were investigated, the model was lacking additional buffer zones that are typically found in larger traffic grids. Thus, the agents were limited in their action space and were forced to optimize performance along a horizontal arterial. In related

work [6], [7], the artificially generated simulation models were designed to be multi-dimensional. The Lemgo road network does not exhibit such features and the majority of vehicles flow along a single horizontal arterial. This could indicate that coordination might not always be helpful or necessary on a city or area-level, depending on the underlying road network layout. These experimental results pose interesting research questions that should be addressed in future studies. Overall, these findings suggest that the two nodes were not affecting each other as much as initially expected and that the effects of non-stationarity could be disregarded for this case study. Consequently, the cooperative discount factor was set to $\omega = 0$.

D. Training

The general workflow for the training process can be described as follows. For each training step, the simulation is executed for the duration of the 10 second control interval, after which the states and rewards for each agent are retrieved. Each new experience tuple, containing the previous state, selected action, reward and new state, for both agents is stored in the corresponding replay buffer. Based on the new state and the current ϵ -greedy policy, either an action with the highest expected reward or a random action is selected from the value network. The agents for both nodes are fed with the corresponding selected actions. Each agent then moves through several phase transitioning functions to ensure compliance with the prescribed green-yellow and all-red interstage intervals, after which the simulation is continued and the process is repeated. Every 4 steps, the value network is trained 10 times with mini batches of 64 samples taken from the replay buffer. Once an episode of 800 seconds is finished, the agents update their ϵ -greedy policy, VISSIM is terminated and the simulation is restarted with a new seed. The target network is synchronized every 500 steps. After 75,000 steps the training is concluded and the trained value network is saved. For inference during the evaluation, the model is loaded and actions are chosen based on the retrieved states by inferring from the trained state-action pairs in the loaded value network.

E. Limitations

VISSIM offers no functionality to run in a minimized mode. Parallelization is not feasible, as the required standard licensing limits the user to 4 simultaneous instances. Due to the design of the runtime environment and COM API, the time it takes to restart and reinitialize the simulation environment with each new episode is considerable. This resulted in significantly increased training times. Thus, parameter tweaking was a time consuming task and reduced the amount of different configurations that could be tested within the given time frame.

VII. NUMERICAL EXPERIMENTS

A. Evaluation Setup

To quantify the performance of the trained MA-IDQN models, a direct comparison to the implemented LISA+ signal

Fig. 8: Training results with different cooperative discounts ω .

controls was conducted. 18 simulation runs were executed that lasted 3600 simulation seconds each. The selected evaluation length was based on the identified peak-hour traffic time interval for which the model was trained. The parameter configuration described in Section VI-A was applied for the simulator and IDQN agents. The LISA+ signal plans designed for morning peak-hours were deployed for each node. Both control approaches were simulated using the same road network model and same incrementing simulation seeds.

For both nodes, the queue lengths on each incoming lane were captured by utilizing inbuilt VISSIM evaluation tools. Additionally, four routes that cross both nodes were specified, along which various travel time measurements were conducted. As indicated by checkpoints $C1$, $C2$, $C3$ and $C4$ in Figure 4 and Table VIII, these include $R_1 : C4 \rightarrow C1$ moving from west to east, $R_2 : C1 \rightarrow C4$ leading from east to west, $R_3 : C2 \rightarrow C4$ starting north-east and moving west and $R_4 : C3 \rightarrow C1$ beginning north-west and going east. Each specified route was set to have an exact length of 1000m. The south-bound routes for both nodes were disregarded because very few vehicles arrive on these links or turn onto them as shown in Tables III and IV. After 18 simulation runs each, the measurement aggregates were computed across all captured data.

1) *Queue Length*: The measured queue lengths for both nodes are presented in Table VI and VII. For each node, all incoming lanes are listed separately. The lane numbering begins north and rotates clockwise. n_1 is the left lane coming from the north, n_2 is the right lane, e_3 is the top lane coming from

the east and so on. $n_1 \rightarrow s, w$ indicates that the lane carries traffic that is coming from the north and is moving either south or west. Refer to Figure 2 for further clarification. For each lane, the average queue length, average maximum queue length and average queue stops are listed. These values are average aggregates from all 18 simulation runs. The average queue length is measured in meters and includes gaps between vehicles. A vehicle is considered to be queued, if its speed goes below 15 km/h and is positioned less than 500 meters from the stop line. The average maximum queue length indicates the average length of the longest observed queue for each full simulation run. The average queue stops denote the average number of vehicles that stopped and were considered to be queued for each simulation run. Lastly, the total maximum queue length is listed. It indicates the longest total queue length that was observed from all simulation runs. This metric is representative of potential worst case scenarios and provides a better understanding of the performance limits and maximum capacities. For MA-IDQN, the percentage differences to the corresponding LISA+ results are listed to quantify the change in performance. Figures 9 and 10 depict the average queue lengths over time for nodes $N1$ and $N2$ respectively. They are sampled every second and average the queue lengths from all lanes within one node, aggregated across all simulation runs.

2) *Travel Time & Vehicle Delay*: Table VIII presents the travel time and delay results for the four specified routes in Figure 4. The average travel time indicates the average time individual vehicles required to move from one specified

checkpoint to the other. The average vehicle delay describes the difference between the actual travel time and a hypothetical optimal travel time if no other traffic would be present and all signals were set to green. The average waiting time denotes the average waiting time that individual vehicles accumulated in queues when traveling through the road network. All values are aggregates from 18 simulation runs.

B. Evaluation Results

1) *Queue Length*: The results for node $N1$ in Table VI show that MA-IDQN significantly reduced the average queue lengths by at least 50% across all lanes. Similarly, the average maximum queue lengths were decreased by over 50% along the main arterial $e_4 \rightarrow w$ and $w_7 \rightarrow e, s$. This is further reinforced by the measured total maximum queue lengths for all simulation runs. Figure 9 indicates less queue length oscillations with MQ-IDQN due to shorter control intervals. The more distinct queue length spikes with the LISA+ controls stem from the queued traffic along the $w \leftrightarrow e$ arterial whenever the $s \leftrightarrow n$ direction receives green time. Because of longer control intervals, the maximum queue lengths are increased. MA-IDQN was assigned a minimum control interval of 10 seconds for all stages and directions, which made it more responsive to the traffic conditions and allowed for faster stage changes. Depending on the traffic conditions, the length of individual stages could be extended indefinitely by skipping a stage change for another 10 seconds each time the control interval passed. The rule-based LISA+ controls on the other hand were more limited. The morning peak-hour signal plan that was employed for node $N1$ has a fixed cycle time of 120 seconds within which phases are given prescribed lengths with varying reserves. These reserves represent the additional time a phase can occupy if certain conditions are met, e.g. when a queue length threshold is exceeded. However, these reserves have to be aligned with the other phases. The $w \rightarrow e$ and $e \rightarrow w$ phases have a minimum length of 55 and 38 seconds respectively. Consequently, phases for south- and north-bound directions have to be set such that they do not interfere with the phase timings of the $w \leftrightarrow e$ direction, while still having to fit into the total 120 seconds cycle time. The LISA+ controls generally allow the prolongation of certain phases beyond the given reserves. However, this is realized by simply skipping the next interfering phase entirely if no traffic is detected on the corresponding lane. This also means the current phase is kept until the end of the current cycle and the skipped phase is forced to wait for the next cycle until it can be initiated again.

The results for node $N2$ in Table VII reflect the findings from node $N1$. Due to the low quantity of vehicles that come from or turn to the southern lanes, the resulting change in performance for these directions was partially disregarded. Traffic on $w_8 \rightarrow n$ and $w_8 \rightarrow e, s$ stands out with slightly worsened results which likely relates to the definition of the reward function. The rewards were designed to only consider the average queue length across all lanes of a node, but not the individual lane-level queue length changes. Consequently, the agent might not observe any problems and will not take

action as long as the overall averaged queue length is not deteriorating. Injecting additional lane-level queue variations within the reward function may be addressed in future studies, but was not considered in this paper, as the main focus lay on improving travel times on the macro-level. Another reason for the reduced performance on the highlighted traffic movements could be the result of stage bias in the LISA+ controls. In MA-IDQN all possible stages/actions are weighted the same, since there is no stage that is more likely to be picked during training. Depending on the traffic conditions, the trained value network configures its weights accordingly to find an optimal policy. The LISA+ controls on the other hand use fixed cycle times and certain phases are allocated more time than others. In this case, node $N2$ has a fixed cycle time of 60 seconds, with prescribed phase timings of 28 seconds for lanes coming from the west or east. Phases for north- and south-bound lanes are set to 11 and 12 seconds. Thus, the LISA+ controls allocate more green time to $w_8 \rightarrow n$ and $w_8 \rightarrow e, s$ by default while the MA-IDQN controls do not prefer any direction and switch stages based entirely on the given queue lengths for each lane, resulting in partially deteriorated performance.

When comparing the average queue lengths for all lanes of node $N1$ in Figure 9, with those of node $N2$ in Figure 10, it becomes evident that the MA-IDQN performance is more consistent, regardless of the environment, as the average queue lengths are approximately the same. The reduction in the average queue lengths for LISA+ is a consequence of reduced traffic volumes on the north- and south-bound lanes (refer to the vehicle volumes in Table II). Moreover, the cycle time and individual phase durations in the LISA+ signal plan for node $N2$ are shorter and enable faster stage switching. This makes it come closer to the MA-IDQN performance which employs a similarly short control interval of 10 seconds. However, it still suffers from the same inflexibility as the employed signal plan in node $N1$ with regard to the prescribed phase durations along the $w \leftrightarrow e$ arterial that ultimately result in increased maximum queue lengths.

While the overall queue lengths were reduced for both nodes, the queue stops increased for some lanes. This is a consequence of the more responsive switching behavior of MA-IDQN. Since the LISA+ controls keep their phases for longer periods of time, the overall queue lengths increase but the amount of stopped cars decreases, as the green times are also longer. MA-IDQN switches stages much more frequently and thus increases the chances of facing a red phase when approaching the intersection. However, it also switches to green more quickly once the vehicle comes to a stop and is considered queued. Many of the drawbacks of the rule-based controls modeled in LISA+ seemed to stem from its inability to flexibly adjust phase lengths to account for changing traffic dynamics. Remodeling the given rule-based signal controls with LISA+ to incorporate shorter overall phase lengths was out of the scope of this study, as it would call for extensive modeling knowledge by a trained traffic engineer and require further manual calibration. However, it would represent a valuable additional baseline to compare the RL-based controls against and could be addressed in future studies to potentially further validate our previously observed findings.

Rule-based / LISA+								
Lane / Direction	$n_1 \rightarrow s, w$	$n_2 \rightarrow e$	$e_3 \rightarrow n$	$e_4 \rightarrow w$	$s_5 \rightarrow n, e$	$s_6 \rightarrow w$	$w_7 \rightarrow e, s$	$w_8 \rightarrow n$
Avg. Queue Length (m)	12.64 ± 2.02	8.45 ± 1.30	8.11 ± 1.16	20.87 ± 2.02	4.13 ± 0.97	1.35 ± 0.74	10.33 ± 1.36	10.39 ± 2.37
Avg. Max. Queue Length (m)	119.73 ± 31.25	99.87 ± 29.32	77.60 ± 14.21	118.49 ± 29.62	42.25 ± 8.48	22.95 ± 9.69	104.72 ± 25.31	97.71 ± 41.35
Avg. Queue Stops	170 ± 22	116 ± 18	170 ± 18	246 ± 18	58 ± 9	29 ± 7	189 ± 15	157 ± 27
Total Max. Queue Length (m)	190.59	159.73	100.08	192.99	56.77	53.03	160.23	211.29
RL-based / MA-IDQN								
Lane / Direction	$n_1 \rightarrow s, w$	$n_2 \rightarrow e$	$e_3 \rightarrow n$	$e_4 \rightarrow w$	$s_5 \rightarrow n, e$	$s_6 \rightarrow w$	$w_7 \rightarrow e, s$	$w_8 \rightarrow n$
Avg. Queue Length (m)	5.58 ± 0.59	4.07 ± 0.55	3.29 ± 0.69	3.57 ± 0.59	0.81 ± 0.32	0.22 ± 0.14	3.15 ± 0.54	5.17 ± 1.23
%-Difference	-55.85%	-51.83%	-59.43%	-82.89%	-80.39%	-83.70%	-69.51%	-50.24%
Avg. Max. Queue Length (m)	53.64 ± 16.32	50.26 ± 13.60	55.70 ± 9.42	54.34 ± 10.46	23.38 ± 6.48	14.50 ± 3.62	51.75 ± 10.58	64.24 ± 18.43
%-Difference	-55.20%	-49.67%	-28.22%	-54.14%	-44.66%	-36.82%	-50.58%	-34.25%
Avg. Queue Stops	220 ± 17	146 ± 12	170 ± 21	180 ± 18	56 ± 8	26 ± 6	162 ± 18	174 ± 21
%-Difference	+29.41%	+25.86%	0%	-26.83%	-3.44%	-10.34%	-14.29%	+10.83%
Total Max. Queue Length (m)	101.38	85.08	73.20	74.69	36.55	22.66	79.22	111.49
%-Difference	-46.81%	-46.74%	-26.86%	-61.30%	-35.62%	-57.27%	-50.56%	-47.23%

TABLE VI: Node N_1 queue length comparison of LISA+ and MA-IDQN controls.

Rule-based / LISA+								
Lane / Direction	$N_1 \rightarrow S, W$	$N_2 \rightarrow E$	$E_3 \rightarrow W, N$	$E_4 \rightarrow S$	$S_5 \rightarrow N, E$	$S_6 \rightarrow W$	$W_7 \rightarrow E, S$	$W_8 \rightarrow N$
Avg. Queue Length (m)	7.33 ± 1.22	4.76 ± 1.16	16.12 ± 1.38	>0.01 $>\pm 0.01$	0.61 ± 0.22	0.03 ± 0.05	5.66 ± 0.84	0.49 ± 0.23
Avg. Max. Queue Length (m)	59.90 ± 9.45	49.21 ± 12.68	128.94 ± 12.36	4.63 ± 1.20	22.02 ± 6.89	8.53 ± 4.07	71.02 ± 13.75	20.53 ± 5.95
Avg. Queue Stops	144 ± 19	93 ± 17	484 ± 27	3 ± 2	40 ± 4	7 ± 2	213 ± 21	27 ± 7
Total Max. Queue Length (m)	83.04	73.39	146.80	5.29	36.52	18.15	101.10	30.10
RL-based / MA-IDQN								
Lane / Direction	$N_1 \rightarrow S, W$	$N_2 \rightarrow E$	$E_3 \rightarrow W, N$	$E_4 \rightarrow S$	$S_5 \rightarrow N, E$	$S_6 \rightarrow W$	$W_7 \rightarrow E, S$	$W_8 \rightarrow N$
Avg. Queue Length (m)	3.18 ± 0.37	2.01 ± 0.43	9.58 ± 1.22	0.01 ± 0.01	0.50 ± 0.15	0.05 ± 0.04	4.88 ± 0.46	0.79 ± 0.48
%-Difference	-56.62%	-57.77%	-40.57%	–	-18.03%	–	-13.78%	+61.22%
Avg. Max. Queue Length (m)	45.34 ± 8.58	35.22 ± 9.11	93.99 ± 17.37	6.80 ± 3.02	19.93 ± 4.94	9.19 ± 3.76	61.39 ± 16.88	22.39 ± 8.99
%-Difference	-24.31%	-28.43%	-27.11%	–	-9.49%	–	-13.56%	+9.06%
Avg. Queue Stops	168 ± 12	117 ± 14	407 ± 30	3 ± 1	40 ± 5	10 ± 3	246 ± 13	47 ± 7
%-Difference	+16.67%	+25.81%	-15.91%	–	0%	–	+15.49%	+71.07%
Total Max. Queue Length (m)	58.59	58.15	134.14	13.52	31.24	17.39	115.79	45.13
%-Difference	-29.44%	-20.77%	-8.62%	–	-14.46%	–	+14.53%	+49.93%

TABLE VII: Node N_2 queue length comparison of LISA+ and MA-IDQN controls.

Fig. 9: Average queue length of all incoming lanes at node $N1$ for LISA+ and MA-IDQN controls. Fig. 10: Average queue length of all incoming lanes at node $N2$ for LISA+ and MA-IDQN controls.

Rule-based / LISA+				
Route	$R_1 :$ $C4 \rightarrow C1$	$R_2 :$ $C1 \rightarrow C4$	$R_3 :$ $C2 \rightarrow C4$	$R_4 :$ $C3 \rightarrow C1$
Avg. Trvl. Time (s)	98.72 ± 2.11	124.51 ± 2.38	108.29 ± 2.21	118.38 ± 3.33
Avg. Veh. Delay (s)	29.96 ± 2.18	55.67 ± 2.35	37.22 ± 2.22	47.08 ± 3.36
Avg. Wait. Time (s)	17.78 ± 1.81	38.21 ± 2.19	24.13 ± 1.76	33.05 ± 2.61
RL-based / MA-IDQN				
Route	$R_1 :$ $C4 \rightarrow C1$	$R_2 :$ $C1 \rightarrow C4$	$R_3 :$ $C2 \rightarrow C4$	$R_4 :$ $C3 \rightarrow C1$
Avg. Trvl. Time (s)	88.73 ± 1.57	90.80 ± 1.47	100.11 ± 1.41	92.26 ± 1.62
%-Difference	-10.11%	-27.07%	-7.55%	-22.06%
Avg. Veh. Delay (s)	19.97 ± 1.63	22.07 ± 1.41	29.11 ± 1.45	21.20 ± 1.46
%-Difference	-33.34%	-60.36%	-21.79%	-54.97%
Avg. Wait. Time (s)	8.17 ± 0.87	9.24 ± 0.80	15.22 ± 1.12	9.51 ± 1.02
%-Difference	-53.05%	-75.82%	-36.92%	-71.23%

TABLE VIII: Delay comparison of LISA+ and MA-IDQN controls for different routes as per Figure 4.

2) *Travel Time & Vehicle Delay*: The captured travel time and delay metrics along the specified routes provide a good representation of the traffic conditions from the perspective of individual vehicles. Drivers typically do not notice if a queue length is halved but rather pay attention to the time required to get from point A to B. The results show improvements for all routes in all categories. The largest change can be observed for routes that exhibit high vehicle volumes. For example, routes R_1 and R_2 that travel along the main arterial $C4 \leftrightarrow C1$ both indicate significant improvements, however R_2 for direction $C1 \rightarrow C4$ exhibits a more extreme change. This is due to the fact that the traffic flow during morning peak-hour intervals is mostly directed westwards, as most commuters try to reach nearby cities west of Lemgo. Due to the increased vehicle load on the $C1 \rightarrow C4$ axis, the larger impact of the MA-IDQN performance benefits and its improved responsiveness generated greater results. R_4 stands out with the second highest performance increases. Since R_4 starts north of node $N2$, vehicles that travel on that route are expected to be queued up longer with the LISA+ controls. As previously mentioned, node $N2$ uses a short green time of 12 seconds for its northern lanes, while allocating the rest of its 60 second cycle time to the other lanes coming from east or west. This corresponds to only 20% total green time during a stage cycle. Consequently, vehicles are likely to spend more time in

a queue when approaching node N_2 from the north. With MA-IDQN, no direction receives longer green times and instead all incoming lanes are treated with the same priority. For this reason, R_4 experienced massive improvements in travel time and vehicle delays.

VIII. CONCLUSION

In this paper, MA-IDQN was employed to learn and execute traffic signal controls. The agents were trained on a realistic traffic simulation model of a road network in the town of Lemgo. The studied route consists of two interconnected signaled intersections. The town's extensive traffic sensor infrastructure was used to validate and feed the simulation model with real-world data. Conventional rule-based signal controls, currently in use at both of the studied intersections, were integrated into the simulation model and served as a performance baseline during evaluation of the RL-based controls. MA-IDQN exhibited significantly reduced queue lengths, waiting times and travel times throughout the whole real-world road network when compared to conventional rule-based controls implemented in LISA+. The performance improvements were mainly attributed to MA-IDQN being trained with a rather short control interval and no bias towards certain stage selections. The compared LISA+ controls employ prescribed phase timings that can slightly vary in their duration, depending on whether certain vehicle thresholds were exceeded. They rely on fixed cycle times within which all phases are allocated a minimum green time of up to 5 times the length of the MA-IDQN control interval. This rigid setup did not allow for much flexibility and resulted in similar green and red times for all directions throughout the whole simulation run. Conversely, the MA-IDQN approach could freely decide whether to switch to a new stage or prolong the current selection after each control interval. It meant that no direction was being prioritized by longer predefined minimum green times. Consequently, the more responsive signal control setup could immediately adapt to the real-time traffic situations without having to abide to a fixed schedule. This demonstrates that RL represents a feasible approach for learning and executing adaptive signal controls. The RL based control model was trained with minimal human intervention and was able to outperform a rule-based control approach which was manually modeled and calibrated. Overall, the results of this paper show that the applicability of the tested RL mechanisms is not limited to simplified artificial road models, but that RL-based TSCs also achieve promising results in more complex and realistic simulation environments when being directly compared to rule-based controls from the real world. Future studies are necessary to account for public transportation, pedestrians, emergency vehicles, traffic-influencing incidents, weather effects, unideal road conditions, abnormal driving behavior and other factors, so the general volatility of traffic dynamics can be modeled in even greater detail and reinforce our findings in this paper.

ACKNOWLEDGMENT

The authors of this paper would like to thank the companies PTV AG, Stührenberg GmbH, SCHLOTHAUER & WAUER

Ingenieurgesellschaft für Straßenverkehr mbH, the Fraunhofer IOSB-INA Institute and the Department of Civil and Environmental Engineering at Stanford University for supporting the work published in this paper.

REFERENCES

- [1] P. H. Bovy and I. Salomon, "Congestion in Europe: Measurements, spatial patterns, policies," Delft University of Technology, Tech. Rep., 1998.
- [2] U.S. Department of Transportation - Federal Highway Administration, "Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation," Texas Transportation Institute, Tech. Rep., sep 2005.
- [3] InIT and Fraunhofer IOSB-INA, "Lemgo Digital," 2020. [Online]. Available: <https://lemgo-digital.de/index.php/de/>
- [4] Fraunhofer IOSB-INA, "Projekt KI4LSA: Künstliche Intelligenz für LichtSignalAnlagen," 2021. [Online]. Available: <https://www.iosb-ina.fraunhofer.de/de/geschaeftsbereiche/maschinelles-lernen/forschungsthemen-und-projekte/projekt-KI4LSA.html>
- [5] Stührenberg GmbH, "Stührenberg - moderne verkehrstechnik," 2021. [Online]. Available: <https://www.stuehrenberg.de/de/start.html>
- [6] T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2020.
- [7] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai *et al.*, "Cooperative deep reinforcement learning for large-scale traffic grid signal control," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2687–2700, 2020.
- [8] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *30th Conference on Neural Information Processing Systems*, 2016.
- [9] N. Casas, "Deep Deterministic Policy Gradient for Urban Traffic Light Control," 2017.
- [10] M. Barthauer and B. Friedrich, "Connecting microscopic traffic simulation and LISA+ external signal control," *Transportation Research Procedia*, vol. 27, pp. 420–427, 2017.
- [11] M. Aftabuzzaman, "Measuring traffic congestion - A critical review," in *30th Australasian Transport Research Forum*, 2007.
- [12] A. Bhandari, V. Patel, and M. Patel, "A survey on traffic congestion detection and Re-routing strategies," in *IEEE 2nd International Conference on Trends in Electronics and Informatics*, 2018, pp. 42–44.
- [13] A. Koźlak and D. Wach, "Causes of traffic congestion in urban areas. Case of Poland," *SHS Web of Conferences*, vol. 57, p. 01019, 2018.
- [14] S. M. Qadri, M. A. Gökçe, and E. Öner, "State-of-art review of traffic signal control methods: challenges and opportunities," *European Transport Research Review*, vol. 12, no. 1, 2020.
- [15] S. Chen and D. J. Sun, "An Improved Adaptive Signal Control Method for Isolated Signalized Intersection Based on Dynamic Programming," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 4, pp. 4–14, 2016.
- [16] P. W. Shaikh, M. El-Abd, M. Khanafer, and K. Gao, "A Review on Swarm Intelligence and Evolutionary Algorithms for Solving the Traffic Signal Control Problem," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [17] A. Müller, V. Rangras, T. Ferfers, F. Hufen, L. Schreckenberger *et al.*, "Towards Real-World Deployment of Reinforcement Learning for Traffic Signal Control," in *IEEE International Conference on Machine Learning and Applications*, 2021.
- [18] D. I. Robertson, "TRANSYT method for area traffic control," *Traffic Engineering and Control*, vol. 10, pp. 276–281, 1969.
- [19] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. C. Royle, "The SCOOT on-line traffic signal optimization technique," *Traffic Engineering and Control*, vol. 23, pp. 190–192, 1982.
- [20] P. Lowrie, "The Sydney Coordinated Adaptive Traffic System – Principles, Methodology, Algorithms," in *IEEE International Conference on Road Traffic Signaling*, 1982, pp. 67–70.
- [21] Y. Bi, X. Lu, Z. Sun, D. Srinivasan, and Z. Sun, "Optimal Type-2 Fuzzy System for Arterial Traffic Signal Control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 3009–3027, 2018.
- [22] M. H. Khooban and A. Liaghat, "A time-varying strategy for urban traffic network control: A fuzzy logic control based on an improved black hole algorithm," *International Journal of Bio-Inspired Computation*, vol. 10, no. 1, pp. 33–42, 2017.

- [23] H. Asadi, R. T. Moghaddam, N. S. Pour, and E. Najafi, "A new nondominated sorting genetic algorithm based on the regression line for fuzzy traffic signal optimization problem," *Scientia Iranica*, vol. 25, no. 3E, pp. 1712–1723, 2018.
- [24] Y. Zhang and Y. Zhou, "Distributed coordination control of traffic network flow using adaptive genetic algorithm based on cloud computing," *Journal of Network and Computer Applications*, vol. 119, pp. 110–120, 2018.
- [25] A. K. Ikiriwate, D. D. Perera, S. M. Samarakoon, D. M. Dissanayake, and P. L. Rupasinghe, "Traffic Density Estimation and Traffic Control using Convolutional Neural Network," *International Conference on Advancements in Computing*, pp. 323–328, 2019.
- [26] A. Niranjani and G. Mishra, "ETAS - Efficient Traffic Signal Assistance System Using Deep Learning," *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCNT 2018*, 2018.
- [27] X. Y. Liu, S. Borst, Z. Ding, and A. Walid, "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey," in *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [28] M. L. Puterman, *Markov Decision Processes - Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.
- [29] R. E. Bellman and S. E. Freyfus, *Applied Dynamic Programming*. United States Air Force Project RAND, 1969.
- [30] C. J. C. H. Watkins and P. Dayan, "Q-Learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.
- [31] M. Sewak, *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*. Springer, 2019.
- [32] J. N. Foerster, "Deep Multi-Agent Reinforcement Learning," Ph.D. dissertation, University of Oxford, 2018.
- [33] M. Tan, "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents," *10th International Conference on Machine Learning*, pp. 330–337, 1993.
- [34] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus *et al.*, "Multi-agent cooperation and competition with deep reinforcement learning," in *PLoS ONE*, vol. 12, no. 4, 2017.
- [35] J. Barcelo, "Microscopic Traffic Flow Simulator VISSIM," in *Fundamentals of Traffic Simulation (International Series in Operations Research & Management Science)*, 2010, ch. 2.
- [36] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO - Simulation of Urban Mobility," in *The Third International Conference on Advances in System Simulation*, 2011, pp. 55–60.
- [37] G. D. Cameron and G. I. Duncan, "PARAMICS - Parallel microscopic simulation of road traffic," *Journal of Supercomputing*, vol. 10, no. 1, pp. 25–53, 1996.
- [38] L. E. Owen, Y. Zhang, L. Rao, and G. McHale, "Traffic flow simulation using CORSIM," *Winter Simulation Conference Proceedings*, vol. 2, pp. 1143–1147, 2000.
- [39] J. Casas, J. L. Ferrer, D. Garcia, J. Perarnau, and A. Torday, "Traffic simulation with AIMSUN," in *Fundamentals of Traffic Simulation*, 2010, ch. 5, pp. 173–232.
- [40] D. Kobbeloer, *Dezentrale Steuerung von Lichtsignalanlagen in urbanen Verkehrsnetzen*. Universität Kassel, 2007.
- [41] M. Saidallah, A. El Fergougui, and A. E. Elalaoui, "A comparative study of urban road traffic simulators," *MATEC Web of Conferences*, vol. 81, 2016.
- [42] R. Wiedemann, *Simulation des Strassenverkehrsflusses*, 8th ed. Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, 1974.
- [43] U. Sparmann, "Spurwechselvorgänge auf zweispurigen BAB-Richtungsfahrbahnen," *Forschung, Straßenbau und Straßenverkehrstechnik*, vol. 263, 1978.
- [44] A. F. M. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv*, no. 1, 2018.
- [45] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [46] Google Brain Team, "TensorFlow: An end-to-end open source machine learning platform." [Online]. Available: <https://www.tensorflow.org/>
- [47] S. Toubhi, M. A. Babram, T. Nguyen-Huu, N. Marilleau, M. L. Hbid *et al.*, "Adaptive Traffic Signal Control: Exploring Reward Definition for Reinforcement Learning," *Procedia Computer Science*, vol. 109, pp. 513–520, 2017.
- [48] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC)," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, 2013, pp. 1140–1150.
- [49] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.
- [50] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.
- [51] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-agent reinforcement learning in sequential social dilemmas," *International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 464–473, 2017.
- [52] Bundesministeriums der Justiz, "§ 37 Wechsellichtzeichen, Dauerlichtzeichen und Grünpfeil, Randnummer 17, Punkt IX," in *Allgemeine Verwaltungsvorschrift zur Straßenverkehrs-Ordnung (VwV-StVO)*, 2009.

Maxim Friesen received his B.S. degree in electrical engineering in 2017 and the M.S. degree in information technologies in 2021, both from the Technische Hochschule Ostwestfalen-Lippe, Lemgo, Germany. His current research interests include smart city and Internet of Things technologies, distributed industrial communication systems, network resource management and applied machine learning algorithms, such as reinforcement learning and deep learning.

Tian Tan received his B.S. degree in telecommunications engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2012, and the M.S. degree from Stanford University, Stanford, USA, in 2015, where he is currently pursuing the Ph.D. degree with the Civil and Environmental Engineering Department (with a Ph.D. minor in computer science). His current research interests include machine learning and algorithms, such as reinforcement learning, deep learning, deep generative models, and statistical learning theory.

He is a Senior Member of the IEEE and a member of the IEEE Industrial Electronics Society.

Jürgen Jasperneite received his Dr.-Ing. degree in electrical engineering and information technology from the Otto-von-Guericke-University Magdeburg, Magdeburg, Germany, in 2002. He is a full professor of computer networks at the Technische Hochschule Ostwestfalen-Lippe and the founding director of the Institute for Industrial Information Technologies and of the Fraunhofer IOSB-INA in Lemgo, Germany. He is one of the main initiators of the Centrum Industrial IT, which is Germany's first science-to-business center in the field of industrial automation.

He is a Senior Member of the IEEE and a member of the IEEE Industrial Electronics Society.

Jie Wang received his B.S. degree from Shanghai Jiao Tong University, Shanghai, China, his two M.S. degrees from Stanford University, Stanford, and University of Miami, Coral Gables, USA, and his Ph.D. from Stanford University. He is an Adjunct Professor with Stanford University and Executive Director of the Stanford Center for Sustainable Development and Global Competitiveness. His current research interests include information analytics and knowledge acquisition for new business and industry development, enterprise modeling, smart manufacturing, smart infrastructures and smart city, computational learning and big data for engineering systems and environmental informatics.