2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

An Ontology-based Intelligent Traffic Signal Control Model

Saeedeh Ghanadbashi[1] and Fatemeh Golpayegani[2]

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564962

Abstract— **Reinforcement Learning (RL) can enhance the adjustment of the traffic signals' phases to improve the traffic flow. RL methods use ontologies and reasoning to enrich the controllers' domain knowledge, enabling them to interpret the traffic data, and ultimately improving their performance.**

Various RL methods are proposed for signal controllers with assumptions such as operating in non-stochastic environments with a predictable traffic flow and observing the fine-grained information of all vehicles. Such methods have not examined the robustness of the trained RL controllers' action selection when deployed in dynamic environments with partial detection of vehicles. However, in the real world, not all vehicles can be detectable, and not all events can be predicted.

In this paper, we propose an Ontology-based Intelligent Traffic Signal Control (OITSC) model that augments the RL controllers' observation using an environment ontology model, which improves their action selection particularly in dynamic, partially observable environments with stochastic traffic flow. The decreased vehicles' waiting time in various traffic scenarios with partial detection of vehicles, noisy sensor data, and unexpected traffic events shows that the performance of the controllers is significantly improved in all tested RL algorithms (i.e., Q-learning, SARSA, and Deep Q-Network).

I. INTRODUCTION

Intelligent Traffic Signal Control (ITSC) systems have recently received increasing attention as a way of improving the traffic flow and decreasing traffic congestion [1]. ITSC combines traditional Traffic Signal Control (TSC) with communication technology using an array of sensors and Artificial Intelligence (AI) to coordinate traffic signal phases according to real-time traffic. The Internet of Things places such technology in the driver's seat instead of setting it up at each traffic signal controller. It could either be external to the vehicle (e.g. an application) or integrated into it (e.g. Dedicated Short Range Communications (DSRC) radios) allowing vehicle tracking (i.e., position, speed, etc.), thus providing a great amount of data that can be used in the state representation of an intersection [2]. Most traditional ITSC systems assume that every vehicle can be observed by the signal controller agents. However, in real-world scenarios, only a portion of vehicles are equipped with such technologies (i.e., partial detection of vehicles) [3]. Also, the controller agents do not always obtain the information of vehicles accurately (i.e., information uncertainty) because the data collected from sensors might be noisy (e.g., observation noise) [4]. So, the controllers' observation is often based on incomplete, ambiguous, and noisy data (i.e., partial observability), which may affect their decision-making.

[1]Saeedeh Ghanadbashi is a PhD Candidate at the Department of Computer Science, University College Dublin, Dublin, Ireland saeedeh.ghanadbashi@ucdconnect.ie

[2]Fatemeh Golpayegani is an Assistant Professor with the Department of Computer Science, University College Dublin, Dublin, Ireland fatemeh.golpayegani@ucd.ie

Moreover, in ITSC systems, it is not possible to predict all the events, and consequently facing unforeseen situations is inevitable (i.e., dynamic and stochastic traffic flow). Due to unexpected events such as traffic accidents, disabled vehicles, adverse weather conditions, spilled loads, and hazardous materials, the traffic demand varies from time to time which increases the travel times [5]. So, the controller agents should be able to handle emerging requirements and adapt to their ever-changing environment on a real-time basis [6]. In such situations, effective incident management (e.g., considering how to move traffic during an emergency) can decrease congestion, secondary crashes, improve roadway safety, and decrease traffic delays. For instance, in Fig. 3, at intersection 5, an unexpected event occurs when an incident happens, and it is expected that the system manages the traffic in a way that the ambulance could get to that location as soon as possible.

Therefore, partial observability, dynamism, and stochastic traffic flow are important characteristics of the environment that must be taken into account when designing an ITSC system.

Reinforcement Learning (RL) is a type of machine learning technique that enables the controller agents to learn a policy (i.e., mapping between states and actions) that would maximise the total cumulative reward. This usually requires the RL controller agent to apply actions to the same environment many times to discover the optimal actions (i.e., action selection) [7]. RL techniques such as Q-learning, SARSA (State–Action–Reward–State–Action), and Deep Q-Network (DQN) are useful for partial detection of vehicles [8], [3], [2] and stochastic traffic network [9] as they do not require comprehensive theoretical modelling of the underlying environment dynamics. However, in the early stages of the deployment, when the detection rate of vehicles is low, RL algorithms may not perform well and execute a wrong update because of a high degree of unpredictability of the environment and detected vehicle patterns [8]. This may result in major traffic congestion in real-world environments. Also, RL controller agents cannot observe the reward when operating in a partially observable environment. Hence, when the real environment is significantly different from the simulated one, the controllers trained in the simulator are not robust and cannot adapt to such environments [3]. Finally, most of the current RL algorithms are evaluated in deterministic or non-stochastic environments with a fixed traffic flow pattern, and characteristics such as dynamism and unpredictable traffic flow patterns are neglected. In the literature, ontological knowledge is used to enrich the semantics of the raw sensor data and provide reasoning capabilities [10], [11], however, the concerns related to partial detection of vehicles, noisy sensor data, and unexpected traffic events are not addressed.

In this paper, we present a novel Ontology-based Intelligent Traffic Signal Control (OITSC) model that enables the RL signal controllers to augment their observation and improve their action selection. OITSC models the traffic signal controllers as autonomous RL agents with access to the environment ontology model, which is used to sample observation and augment it by inferring implicit information. The augmented observation will be then used to facilitate their adaptation to unpredictable changes in their environment. Various scenarios are tested to evaluate the performance of the OITSC model. The decreased network average waiting time shows that the performance of all the RL controllers investigated in this paper (i.e., Q-learning, SARSA, and DQN) can be significantly improved under partial observability, dynamism, and stochastic traffic flow using the OITSC model.

The paper is organised as follows. Section II presents a review of relevant literature and Section III provides the required background knowledge. Section IV briefly describes the problem statement. Section V presents our model and how it works. In Section VI, the case study is defined and the results are analysed in Section VII. Finally, our conclusion and future works are drawn in Section VIII.

II. RELATED WORK

To tackle the partial detection of vehicles problem in ITSC systems, various RL algorithms are used such as Q-learning, Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), Actor-Critic with Kronecker-Factored Trust-Region (ACKTR) [8], Deep Q-learning [3], and Connectionist RL [2]. RL algorithms have high deviation under partial detection of vehicles especially in off-policy learning algorithms such as Q-learning and DQN, in which rapid Q-value updates in a noisy environment sometimes lead to a "bad update". However, policy gradient methods such as the A2C algorithm try to improve their policy based on their approximation of policy gradient and have better performance at each update [8].

Also, proposed approaches to evaluate traffic conditions using traffic volume and average speed of vehicles do not work as well when the analysed data does not represent the unexpected/unforeseen traffic situation and cannot deal with accompanied uncertainties [12]. In [13], the authors propose RL signal controllers for dynamic and stochastic traffic micro-simulation. However, this model works with constant traffic signal phase duration and if the action space or state space is changed, the model needs to be retrained. In [14], the authors evaluate the adaptability of two commonly used AI algorithms to optimize the traffic signalisation with instant stochastic traffic demand. They conclude that a deterministic algorithm such as Fuzzy Logic cannot cope with the stochastic traffic flow because it computes the solution based on the fixed/non-adaptive computational procedure. However, meta-heuristic-based algorithms such as Genetic Algorithm (GA) have shown better performance and determine a near-optimum solution with partial knowledge about the traffic flow pattern. The reason is that GA uses the mutation mechanism to have a wider exploration in solution space.

In this paper, our goal is to improve the RL signal controller agents' action selection, even at a low detection rate of vehicles and in the presence of unexpected events.

Ontology provides user-contributed, augmented intelligence, and machine-understandable semantics of data. Some AI techniques enhance agents' learning process by providing semantic models and augmented feedback loop to optimize their overall accuracy [15]. We take inspiration from this research area and use ontological knowledge to address partial detection of vehicles and adapt to unexpected events in the TSC context. In [16], the authors model the TSC context using a fuzzy ontology for reusing knowledge and firing suitable fuzzy rules using a fuzzy inference engine. The approach proposed in [10], enriches the semantics of the raw sensor data and infers contextual description by an ontology-based model using context-aware attributes. Although ontology has been used in the context of TSC, however, the concerns related to environments' partial observability, dynamism, and stochastic traffic flow are not addressed yet. In this paper, the signal controllers use ontology and RL to obtain a better understanding of their environment and augment their observation. We show that the performance of several RL algorithms including Q-learning, SARSA, and DQN can be improved in such environments by ontological knowledge.

III. BACKGROUND

This section briefly presents the required background information.

A. Markov Decision Process

A Markov Decision Process (MDP) is a discrete-time stochastic control process that provides a mathematical framework to describe an environment in RL. MDP is a 5-tuple (S, A, T, R, γ), where S is a set of states called the state space, A is a set of actions called the action space, probabilistic transition function $T(s, a, s') = \Pr\left(s^{t+1} = s' \mid s^t = s, a^t = a\right) \in [0, 1]$ is the probability that action a in state s at time step t will lead to state s' at time step $t+1$, $R(s, a, s') \in \mathbb{R}$ is the reward value the agent gets from the environment after doing action a in state s which takes the environment to the new state s', and $0 \leq \gamma < 1$ is a discount rate that shows how much the future rewards contribute to the total reward [7].

Partially Observable MDPs: Most real-world problems are not completely captured by MDPs, and exhibit at least some degree of partial observability. A Partially Observable MDP (POMDP) $(\Omega, S, A, R, T, O, \gamma)$ is an MDP with two additional components: the possibly infinite set Ω of observations, and the $O : S \rightarrow \Omega$ function that produces observations o based on the unobservable state s of the process (i.e., a set of conditional observation probabilities). At each time, the agent receives an observation $o \in \Omega$ which depends on the new state of the environment, s', and on the just taken action, a, with probability $O(o, a, s')$. Learning is difficult in these settings due to partial observability because the same observation may be obtained from two different states, each requiring a different optimal action [7].

B. Reinforcement Learning

RL is a trial-and-error method in which the agents learn through interaction with the environment by sequential decision-making in such a way that an agent chooses an action according to its policy, which is subsequently sent to the environment, then the environment moves to a new state and an immediate reward will be sent to the agent. The objective is to optimize the expected reward for the agent in the underlying MDPs.

Q-learning is a fundamental RL method in which the agent computes the Q-value that estimates discounted cumulative reward of actions as displayed in Equation 1.

$$Q(s,a) = Q(s,a) + \rho[R(s,a,s') + \gamma \max Q'(s',a') - Q(s,a)] \quad (1)$$

$Q(s,a)$ is the current stored Q-value for applying action a in state s and $\max Q'(s',a')$ is the maximum expected future reward. The learning rate ρ determines to which degree the new information overrides the old one [7].

SARSA is an on-policy RL technique that learns Q-values relative to the current policy it follows. The DQN algorithm uses a Deep Neural Network (DNN), and the input of the neural network would be the state that the agent is in and the targets would be the Q-values of each of the actions [7].

RL agents choose actions based on the exploration-exploitation dilemma in which an exploration (i.e., exploring action space) and exploitation (i.e., performing the best action) scheme used to compute a policy that maximises the payoff. ε is the percentage dedicated to the exploration.

C. Ontology

An ontology is a description of concepts, properties, relations, and axioms. By using a hierarchical approach, an ontology can describe all relations between concepts such as the inverse relation between successor and predecessor, multiple relations, and conditional relation. The domain and range of a relation determine what kind of instances it can be used for (i.e., domain) and what kind of values it can have (i.e., range). For example, in the TSC context, “Vehicle” (i.e., domain) “hasType” (i.e., relation) and can be an “Emergency” one (i.e., range). In [17], Semantic Sensor Network (SSN) ontology is proposed to describe sensor resources, and the data they collect as observations. It bridges the gap between low-level data streams coming from sensors in real-time and high-level concepts used by agents to interpret an observation (i.e., high-level semantic representations).

In this paper, to model the concepts in the TSC context, we use the ontology proposed in [18] (see Fig. 1). The inference rules are expressed in Semantic Web Rule Language (SWRL) and there are two methods of reasoning when using inference rules [19]:

- **Forward reasoning:** It starts from state observation and applies inference rules to extract more facts until reaches the goal. For example, we can conclude from “A” and “A implies B” to “B”.
- **Backward reasoning:** It starts from the goal and chaining through inference rules to find the required facts that support the goal. For example, we can conclude from “not B” and “A implies B” to “not A”.

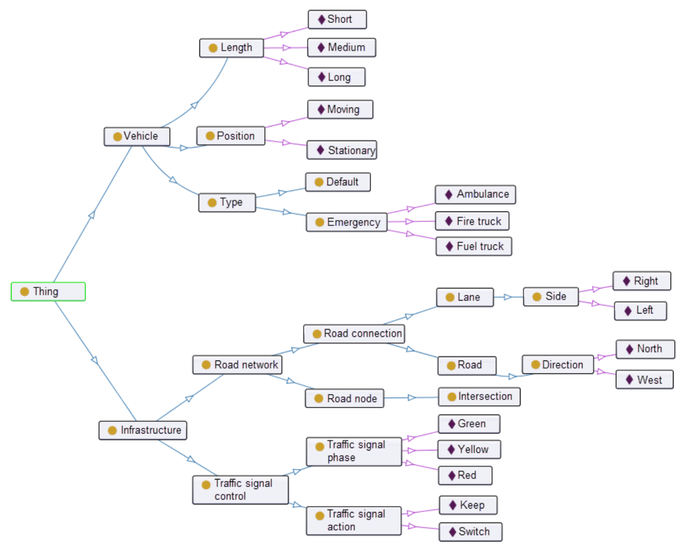

Fig. 1. Ontology for traffic signal control.

Concepts’ weight: When designing an ontology, some concepts might appear more important than others in a context. The importance of concepts may be considered as a graded phenomenon. The judgement regarding the importance of concepts is based on background knowledge related to concepts and relations that are represented by an ontology (i.e., ontology-based knowledge representation). Assigning weights to concepts (i.e., concept weighting) is a technique used in various methods of data analysis in which the concepts’ weight determines their importance [20].

IV. PROBLEM STATEMENT

Most of ITSC systems proposed in recent years are still relying on vehicle state and quantity information that is either very difficult to obtain, or that requires expensive equipment, and it cannot be expected that every user is able or agrees to get the proposed technology/device [2]. Therefore, the system we are considering includes detectable and undetectable vehicles. The detectable vehicles are equipped with communication devices (e.g., DSRC) that can communicate with the signal controller at an intersection. Undetectable vehicles are not equipped with such devices, so the signal controller is not able to detect them.

In this system, the environment is **partially observable** and can be modelled as a POMDP since the signal controller has only access to the information communicated by the detectable vehicles and yet has to decide the current phase at the intersection to minimize the delay for both vehicle types [3] (e.g., see Fig. 3 intersection 2). Moreover, to resemble a real-world traffic system with non-recurrent congestion, we include **dynamic and stochastic traffic flow** by considering unexpected events such as the arrival of important vehicles (i.e., ambulances, fuel trucks, and trailer trucks) in the system. The duties of the emergency vehicles such as ambulances are to transport medical equipment, critical patients, and necessary medicines in time. Heavy vehicles such as trailer trucks can increase the average travel time and the number of traffic accidents, and potentially reduce traffic safety because of their length and size, and acceleration/deceleration characteristics. So, during such unexpected events, the signal controller should change the traffic signal phase from red to green to make clearance for important vehicles' paths automatically.

A. Parameter Modelling

The signal controllers are modelled as RL agents that receive reward and state observation from the environment and take actions accordingly. State, actions, and reward are as follows:

1) *State:* We model information of each state $s^t_{g_i}$ for the signal controller g_i at time step t as Equation 2. The details of the parameters are listed in Table I.

$$s^t_{g_i} = \{\Phi^t_{g_i}, e^t_{g_i}, q^t_{l_i}, d^t_{l_i}, k^t_{v_i}, z^t_{v_i}, w^t_{v_i}\} \tag{2}$$

We assume that the controller agent's partial observation $s^t_{g_i}$ does not contain enough information for the states of all vehicles at the intersection.

2) *Actions:* The action space $A = \{1, 2, 3\}$ is the set of all traffic signal phases (i.e., 1, 2, and 3 indicating green, yellow, and red respectively). Taking action a means selecting an appropriate traffic signal phase for the next time step, which will be "Keep" or "Switch" the traffic signal phase accordingly.

3) *Reward*: The immediate reward the controller agent g_i gets from the environment at time step t after doing action a is calculated as follows:

$$r^t_{g_i} = \sum w^{t-1}_{v_i} - \sum w^t_{v_i} \tag{3}$$

Since the delay increases over time, the reward is always a negative value. So, the RL agent tries different signal control schemes and eventually converges to an optimal scheme that yields a maximum reward (i.e., minimum average waiting time).

V. ONTOLOGY-BASED INTELLIGENT TRAFFIC SIGNAL CONTROL MODEL

We present a new Ontology-based Intelligent Traffic Signal Control (OITSC) model that enables the signal controller agents to augment their observation and consequently improve their performance when an unexpected event occurs while operating in a partially observable, dynamic, and stochastic environment. In the OITSC model, RL agents extract important concepts using an ontology-based schema

TABLE I
DETAILS OF STATE REPRESENTATION.

Parameter	Information
$\Phi^t_{g_i}$	Yellow, red, and green *traffic signal phases* for the intersection monitored by the controller g_i at time step t.
$e^t_{g_i}$	Current *traffic signal phase elapsed time*, the time duration from the start of the current traffic signal phase up to now for the intersection monitored by the controller g_i at time step t.
$q^t_{l_i}$	Current *lane queue*, the number of vehicles waiting in each lane divided by the lane capacity (i.e., the length of the lane in meters divided by the sum of the vehicle length and the minimum gap between the two vehicles) for lane l at the intersection monitored by the controller g_i at time step t.
$d^t_{l_i}$	Current *lane density*, the number of vehicles in each lane divided by the lane capacity for lane l at the intersection monitored by the controller g_i at time step t.
$k^t_{v_i}$	The *type of vehicle* (i.e., default, ambulance, fuel truck, and trailer truck) for vehicle v at the intersection monitored by the controller g_i at time step t.
$z^t_{v_i}$	The *position of a vehicle* along the lane, the distance from the front bumper to the start of the lane for vehicle v at the intersection monitored by the controller g_i at time step t.
$w^t_{v_i}$	The *waiting time of a vehicle*, the number of seconds that a vehicle has a speed of less than 0.1 m/s for vehicle v at the intersection monitored by the controller g_i at time step t.

populated via abstracting TSC concepts. They sample the observed data based on concepts' importance and infer implicit observation data based on explicit observation. The augmented observation, concepts' importance, and inference rules will be used to improve the RL agents' performance. RL agents are then enabled to augment their reward function with an efficiency rate using concepts' weight. The components of the OITSC model are shown in Fig. 2 and described in detail in the following subsections.

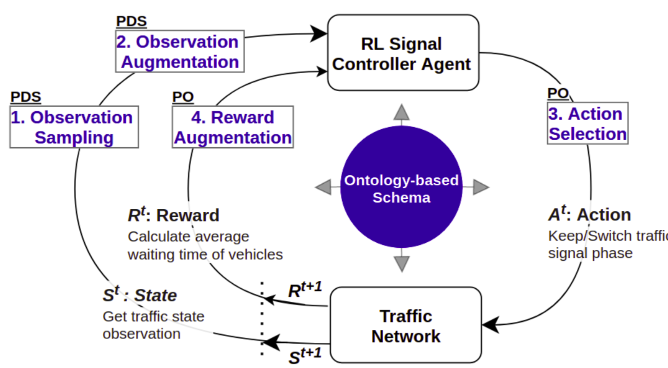

Fig. 2. Ontology-based Intelligent Traffic Signal Control (OITSC) model. **PO** refers to the partial observability, and **PDS** refers to the dynamic, stochastic traffic flow (see Section IV).

A. Ontology-based Schema

Each agent represents its observation using a schema described by an ontology. This schema is used when a semantic description is needed (for example, when interpreting an unexpected/unforeseen event is required) and is composed of surrounding concepts and relations between concepts perceived by the agent. For example, the concept

"Vehicle" and its relation "hasPosition" can be defined. These relations enable inheritance between concepts and automated reasoning. We define $L^t_{g_i}$ as the schema describing the data observed by agent g_i at time step t. C represents the set of concepts, and M represents the set of relations over these concepts (see Equation 4).

$$L^t_{g_i} = \{C^t_{g_i}, M^t_{g_i}\} \tag{4}$$

The importance of each observation $s^t_{g_i}$ is determined based on the importance of the concepts $C^t_{g_i}$ involved. The concept weighting function in Equation 5 is used to quantify the degree of importance of each concept $x \in C^t_{g_i}$ in a context using an iweighting indicator [21]:

$$iw_c(x) = 1/|M(x)| \sum_{m \in M(x)}^{|M(x)|} iw^{(x,y)}_{M_m} \tag{5}$$

To compute concepts' weight, the relations are initially weighted manually by ontology engineers during the ontology development process, and $iw_c(x)$ is calculated based on the average importance weights of the relations $m \in M^t_{g_i}$ of domain concept x constrained by their particular range y. For example, in the TSC context, a "Highest" importance weight can be assigned to a vehicle of the emergency type. There are five degrees of importance weights for the relations, which can be converted to numerical values using predefined mappings (see Table II): "Lowest", "Low", "Middle", "High", and "Highest".

TABLE II
RELATIONS IN TRAFFIC SIGNAL CONTROL ONTOLOGY.

Domain	Relation	Range	Importance weight	Numerical value
Vehicle	hasPosition	Moving	Lowest	0
		Stationary	Low	1
	hasLength	Short	Lowest	0
		Medium	Lowest	0
		Long	Middle (trailer truck)	2
	hasType	Default	Lowest	0
		Emergency	High (fuel truck)	3
			Highest (ambulance)	4

B. Observation Sampling

In partially observable TSC environments, agent g_i needs to sample observation $s^t_{g_i}$ to reduce the impact of observed noisy data on its action selection process [22]. Moreover, when the agent uses more complicated state representation in RL algorithms, it requires a huge number of learning samples [23] and sampling observation can extract important information from the state representation. In our sampling method, observation data $s^t_{g_i}$ is selectively sampled at different sampling rates proportional to concepts' weights $C^t_{g_i}$. So, observation data of concept x with higher iweighting indicator $iw_c(x)$ could be sampled at a higher rate $\lfloor iw_c(x) \times u/n \rceil$ than observation data of concept y with lower importance weight $iw_c(y)$. The sampled observation $\overline{\mathrm{s}^{\mathrm{t}}_{\mathrm{g_i}}}$ is formed as follows:

$$\overline{\mathrm{s}^{\mathrm{t}}_{\mathrm{g_i}}} = \{x_1, \ldots, x_{\lfloor iw_c(x) \times u/n \rceil}, y_1, \ldots, y_{\lfloor iw_c(y) \times u/n \rceil}\}$$
$$\text{if } iw_c(x) + iw_c(y) = n \text{ and sample size} = u \tag{6}$$

For instance, consider two types of vehicles "Stationary" and "Moving" at intersection 7 (see Fig. 3). By computing the iweighting indicator, the agent recognises that the stationary vehicles on the roads r_{NS} and r_{WE} have higher importance than the moving vehicles, so the waiting time of stationary vehicles could be sampled at $8/10$th sample size and the waiting time of moving vehicles could be sampled at $2/10$th sample size.

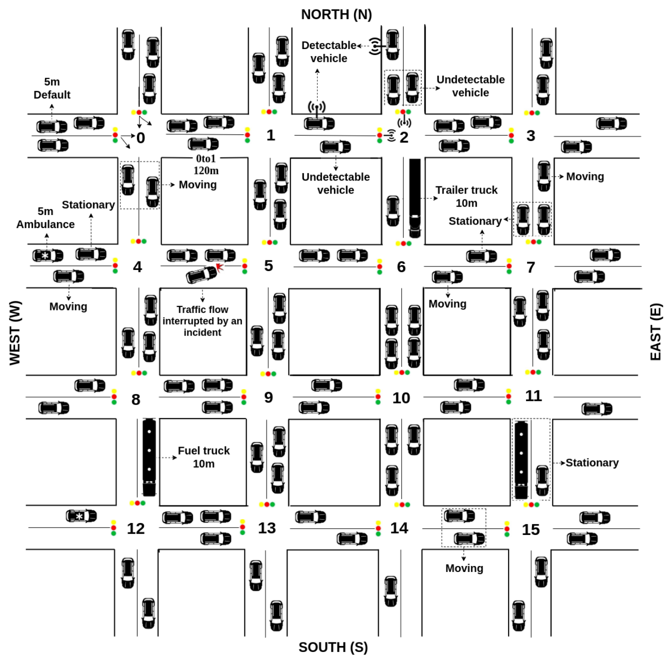

Fig. 3. Simulated traffic network.

C. Observation Augmentation

When operating in partially observable environments, agent g_i extracts implicit observation data of concept x by applying inference rules to the existing relations over explicit observation data of a similar concept y. In other words, the automatic inference mechanism over the ontology rules deduces the augmented observation for the concepts with partial observability. As an example, at intersection 2 in Fig. 3, the agent cannot observe the waiting time of undetectable vehicles in the environment, so it replaces the waiting time of stationary vehicles in the road r_{NS} with the current traffic signal phase elapsed time e^t_2 at time step t according to the inference rules in Table III.

TABLE III
AN EXAMPLE OF INFERENCE RULES, OBSERVATION AUGMENTATION.

Inference rules
TrafficSignalControl($?i$)[1], Intersection($?s$), Road($?r$), Lane($?l$), consistOf($?r, ?l$), Vehicle($?a$), isOn($?a, ?l$), atIntersection($?i, ?s$), isRegulatedBy($?r, ?i$), hasPosition($?a, Stationary$), hasElapsedTime($?i, ?e$) $- >$ hasWaitingTime($?a, ?e$)

D. Action Selection

When operating in an uncertain environment with varying unpredictable traffic flow patterns caused by accidents or other unexpected events, agents dedicate a percentage of their exploration to actions that have more rewards for the most important concepts in the environment. The automatic inference mechanism over the ontology rules deduces the desirable action for the concepts with the highest importance weights. Agent g_i uses a hybrid policy consisting of the proposed ontology-based policy π_{OITSC} and RL policy π_{RL}. Agent g_i can also switch between these two policies (see Equation 7). The percentage function $\alpha(t)$ is a parameter that depends mostly on the occurrence of unexpected events at time step t. In our experiments, since the unexpected events occur per 2 minutes the $\alpha(t)$ is 0.5.

$$\pi(a \mid s) = \alpha(t) \times \pi_{\text{OITSC}}(a \mid s) + (1 - \alpha(t)) \times \pi_{\text{RL}}(a \mid s) \quad (7)$$

As an example, at intersection 4 in Fig. 3, the agent computes the importance weight of the road $iw_c(r_i)$ using the iweighting indicator $\sum_{v_i \in r_i} iw_c(v_i)$ for vehicles v_i on each road r_i (importance weights are listed in Table II), then selects the action with the highest reward for the road with the highest importance weight. For instance, by computing the iweighting indicator, agent g_4 recognizes that r_{WE} has a higher importance than the other road, and selects "Switch" as the next traffic signal action, using the inference rules listed in Table IV, because the current traffic signal phase for road r_{WE} is red.

TABLE IV

AN EXAMPLE OF INFERENCE RULES, ACTION SELECTION.

Inference rules
TrafficSignalControl($?i$), Intersection($?s$), Road($?r_1$), Road($?r_2$), atIntersection($?i, ?s$), DifferentFrom($?r_1, ?r_2$), isRegulatedBy($?r_1, ?i$), isRegulatedBy($?r_2, ?i$), hasHigherImportance($?r_1, ?r_2$), hasTrafficSignalPhase($?r_1, red$), hasTrafficSignalPhase($?r_2, green$) $->$ isNextTrafficSignalAction($?i, Switch$)

E. Reward Augmentation

Reward shaping allows a reward function to be modified to provide a more frequent feedback on appropriate behaviors and improve the RL agents' action selection. We can have $r' = r + f$ where r is the output from the original reward function R, f represents the additional reward from a shaping function F, and r' is the signal given to the agent by the augmented reward function R' [24]. Using an ontology-based schema, we augment the reward with an efficiency rate of actions (see Equation 9). Efficient actions are those that with equal or fewer observations, they can make more changes (e.g., decrease waiting time) in the environment when compared to other valid actions in the current state. Suppose we have a set of k actions $\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_k$. Each action changes n information items of state (output) while observing m information items of state (input). Let us consider an input matrix $X = [x_{ij}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, k]$ and an output matrix $Y = [y_{ij}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, k]$. The q-th line (i.e. X_q and Y_q) of these matrices thus shows quantified inputs/outputs of action a_q. The Efficiency rate (E) of the action can then be expressed as:

$$E = \frac{\text{weighted sum of outputs}}{\text{weighted sum of inputs}} = \frac{\sum_{i=1}^{n} iw_c(i) \times y_{iq}}{\sum_{j=1}^{m} iw_c(j) \times x_{jq}} \quad (8)$$

$$r'_{g_i} = r^t_{g_i} + E \quad (9)$$

For example, at intersection 15, Fig. 3, "Switch" to green phase for road r_{NS} has a higher efficiency rate than "Keep" the green phase for the road r_{WE}, because the former action reduces the waiting time of a fuel truck with "High" importance weight.

VI. EVALUATION

This section presents the evaluation setting and discusses the evaluation scenarios.

A. Traffic Signal Control Simulator

For evaluation purposes, we use a microscopic real-time traffic simulator, Simulation of Urban MObility (SUMO) [25]. The whole simulated traffic network is a $750m \times 750m$ area. Each intersection is a $300m \times 300m$ area (see Fig. 3). At each intersection, we have two double-lane incoming roads and two double-lane exit roads. The road length is 120 meters. The vehicles in incoming roads from west to east are allowed to take right-turn and pass through traffic. The vehicles in incoming roads from north to south are allowed to take left-turn and pass through traffic. The minimum gap between the two vehicles is 2.5 meters. We have four types of vehicles in the simulation: default, ambulance, fuel truck, and trailer truck. The length of default and ambulance vehicles is 5 meters and the length of fuel truck and trailer truck is 10 meters. The default vehicles arrive in the environment following a random process, and the arrival rate of every lane is the same, one per second. The arrival rate of other types of vehicles is according to scenarios discussed in Section VI-B. Vehicles are discarded if they could not be inserted. For all types of vehicles, the max speed is 55.55 m/s, which is equal to 200 km/h. Also, the max accelerating acceleration is 2.6 m/s^2 and the decelerating acceleration is 4.5 m/s^2. SUMO uses the Krauss following model, which guarantees safe driving on the road. The transition phase is defined as 2 seconds of the yellow phase for lanes that have previously received the green phase. The minimum duration of the green phase is set to 5 seconds and the maximum one is set to 100 seconds. The number of simulation seconds ran before learning begins is set to 300 seconds. The number of simulated seconds on SUMO is set to 1,000 seconds. The simulation seconds between actions are set to 5 seconds. We perform the simulation through 10 runs for each scenario. The goal in our network is to maximise the reward in each run by modifying the traffic signals' phases. The simulation results show the average values obtained from 10 runs and are compared to the baseline algorithms. We used the

[1] In Semantic Web Rule Language (SWRL), variables are indicated using the standard convention of prefixing them with a question mark.

interface to instantiate RL environments with SUMO for TSC provided by [26] to interact with the traffic signal-controlled intersections. We followed the default network parameters as shown in Table V.

TABLE V
PARAMETER SETTINGS OF THE BASELINE ALGORITHMS.

Parameter	Value
ε	0.05 (Q-learning, SARSA, DQN)
γ	0.99 (Q-learning), 0.95 (SARSA), 0.99 (DQN)
ρ	1e−1 (Q-learning), 1e−9 (SARSA), 1e−3 (DQN)

B. Evaluation Scenarios and Criteria

Evaluation scenarios: Four scenarios were defined to evaluate the performance of the OITSC model. The basic setting for all our scenarios includes generating five important vehicles (i.e., ambulances, fuel trucks, and trailer trucks) at random roads as unexpected/unforeseen situations per 2 minutes. The further details of each scenario are as follows:

- **Scenario 1:** This scenario is based on the basic setting we explained above. This will test the OITSC model's performance to adapt to such unexpected events.
- **Scenario 2:** The waiting time of 50% of vehicles will be corrupted. This will test the OITSC model's performance to handle the noisy observation data.
- **Scenario 3:** The controller agents cannot observe the waiting time of 20% of vehicles. This will test the OITSC model's performance in handling the partial detection of vehicles problem.
- **Scenario 4:** The controller agents cannot observe the reward for 20% of the vehicles. This will test the OITSC model's performance under a partially observable environment.

Performance metric: The average waiting time of vehicles is used as a performance criterion to measure the efficiency of the methods (see Equations 10).

$$w_{g_i}^t = 1/|v_i| \sum w_{v_i}^t \tag{10}$$

Baselines: We compare the results obtained from the baseline RL controllers to OITSC-enabled RL controllers. Q-learning, SARSA, and DQN are selected as three baseline RL algorithms in this paper.

VII. RESULTS ANALYSIS AND DISCUSSION

The results report the average waiting time for all types of vehicles in 10 runs in each scenario. The results show that the OITSC model significantly decreases the average waiting time of vehicles compared to all baseline algorithms (see Fig. 4 and Table VI).

From Fig. 4, vehicles' waiting time in scenarios 2, 3, and 4 is higher than scenario 1, because in the first three scenarios agents' partial observability can lead to faulty reward that decreases agents' actions selection accuracy. However, in scenario 1 (i.e., less complicated scenario), due to the unexpected/unforeseen situation (i.e., the arrival of important vehicles at intersections), agents are uncertain about the best action and because there are only two possible actions "Switch" or "Keep", even random action selection can result in appropriate action selection.

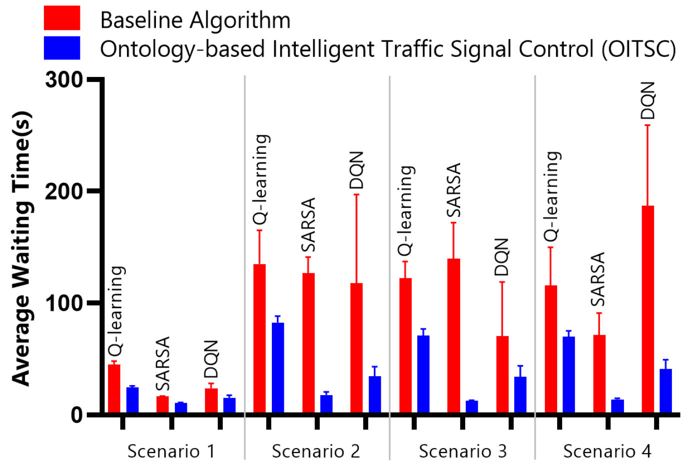

Fig. 4. The average waiting time of vehicles – The OITSC model and the three baselines.

TABLE VI
THE PERCENTAGE DECREASE IN AVERAGE WAITING TIME – THE OITSC MODEL AND THE BASELINE ALGORITHMS.

Scenario	Type of vehicle	OITSC-enabled			
		Q-learning	SARSA	DQN	Average
1	Default	51%	35%	35%	40%
	Ambulance	82%	89%	87%	86%
	Fuel truck	69%	78%	90%	79%
	Trailer truck	71%	68%	36%	58%
	All	68%	67%	62%	66%
2	Default	39%	86%	71%	65%
	Ambulance	31%	90%	66%	62%
	Fuel truck	43%	91%	63%	66%
	Trailer truck	21%	94%	79%	65%
	All	33%	90%	70%	64%
3	Default	42%	91%	51%	61%
	Ambulance	50%	97%	64%	70%
	Fuel truck	6%	96%	53%	52%
	Trailer truck	39%	97%	48%	61%
	All	34%	95%	54%	61%
4	Default	39%	81%	78%	66%
	Ambulance	64%	96%	83%	81%
	Fuel truck	30%	93%	84%	69%
	Trailer truck	57%	91%	80%	76%
	All	47%	90%	81%	73%
Average	All	45%	85%	67%	-

From Table VI, we observe that the OITSC model decreases the waiting time of ambulances in scenarios 3 and 4, fuel trucks in scenario 2, or both of them in scenario 1 more than the other types of vehicles. For example, in scenario 1, the results show 86% decrease in waiting time of ambulances and 79%, 58%, and 40% decrease in waiting time of fuel trucks, trailer trucks, and default vehicles respectively. This is because when important vehicles enter the intersecting roads and we assign the "Highest" and "High" importance weights to the relation "hasType" and its domain "Vehicle" and range "Emergency" (see Table II), consequently, a higher priority is given to the road that has the most important vehicles.

Whole model performance: Finally, the results show 85%, 67%, and 45% decrease in average waiting time for

SARSA, DQN, and Q-learning algorithms respectively for all types of vehicles across all scenarios (see Table VI). It is shown that the OITSC model reduces average waiting time in on-policy learning algorithms such as SARSA more than Q-learning and DQN algorithms because off-policy learning algorithms such as Q-learning and DQN experience more random actions and this can alleviate the negative effects of partial detection of vehicles and stochastic traffic flow to some extent.

VIII. Conclusion and Future Work

Existing studies on using RL agents with Intelligent Traffic Signal Control (ITSC) often lack pragmatic considerations concerning real-world problems, especially for the traffic system infrastructure and traffic flow behavior. This includes constraints imposed by partial detection of vehicles, noisy sensor data, and unexpected traffic events. In this paper, a novel Ontology-based Intelligent Traffic Signal Control (OITSC) model is proposed to address these research gaps.

It is worth mentioning that the OITSC model proposed in the paper is the first attempt to show that using ontology has merit and significant benefits in RL algorithms, however further research is needed to make this model more practical. This paper can be extended in several directions. The OITSC model's performance depends on the accuracy and completeness of the used ontology. Operating in dynamic environments requires the ontology to evolve and update frequently. Ontology evolution techniques can be used to address this issue. Additionally, we have assigned numerical values to the importance degrees of the relations based on experiments that help to set these values in the TSC case study. However, initial values' computation, the transition from discrete qualitative values to quantitative values, and their impact on the overall process if under or overestimated will be further examined in future works. Another future direction would be to further develop the system to achieve multi-agent coordination so that the RL signal controller agents will be able to communicate with each other through their ontologies.

References

[1] J. M. Bailey, F. Golpayegani, and S. Clarke, "CoMASig: A Collaborative Multi-Agent Signal Control to support senior drivers," in *Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1239–1244.

[2] C. Geortay *et al.*, "Master's thesis: Partially detected intelligent traffic signal control using connectionist reinforcement learning," Master's thesis, Université de Liège, Liège, Belgique, 2020. [Online]. Available: https://matheo.uliege.be/handle/2268.2/9068

[3] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Using reinforcement learning with partial vehicle detection for intelligent traffic signal control," *Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 22, no. 1, pp. 404–415, 2020.

[4] C. Wu, Z. Ma, and I. Kim, "Multi-agent reinforcement learning for traffic signal control: Algorithms and robustness analysis," in *Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2020, pp. 1–7.

[5] M. Riad and F. Golpayegani, "Run-time norms synthesis in multi-objective multi-agent systems," *International Workshop on Coordination, Organizations, Institutions, Norms and Ethics for Governance of Multi-Agent Systems (COINE), AAMAS*, 2021.

[6] S. Ghanadbashi and F. Golpayegani, "Using ontology to guide reinforcement learning agents in unseen situations: A traffic signal control system case study," *Applied Intelligence (APIN)*, 2021.

[7] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[8] R. Zhang, R. Leteurtre, B. Striner, A. Alanazi, A. Alghafis, and O. K. Tonguz, "Partially detected intelligent traffic signal control: Environmental adaptation," in *International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2019, pp. 1956–1960.

[9] F. Zhu and S. V. Ukkusuri, "Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 41, pp. 30–47, 2014.

[10] M. Swarnamugi and R. Chinnaiyan, "Context—aware smart reliable service model for intelligent transportation system based on ontology," in *International Conference on Recent Innovations in Computing (ICRIC)*. Springer, Cham, 2020, pp. 23–30.

[11] F. Golpayegani, I. Dusparic, and S. Clarke, "Using social dependence to enable neighbourly behaviour in open multi-agent systems," *Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 3, pp. 1–31, 2019.

[12] M. Kalinic and J. M. Krisp, "Fuzzy inference approach in traffic congestion detection," *Annals of Geo-information Science (GIS)*, vol. 25, no. 4, pp. 329–336, 2019.

[13] W. Genders and S. Razavi, "Policy analysis of adaptive traffic signal control using reinforcement learning," *Computing in Civil Engineering*, vol. 34, no. 1, p. 04019046, 2020.

[14] M. K. Tana, H. S. E. Chuo, K. G. Lim, R. K. Y. Chin, S. S. Yang, and K. T. K. Teo, "A comparison study of deterministic and metaheuristic algorithms for stochastic traffic flow optimization under saturated condition," *ICTACT Journal on Soft Computing (IJSC)*, vol. 10, no. 3, 2020.

[15] J. A. Motta, L. Capus, and N. Tourigny, "Vence: A new machine learning method enhanced by ontological knowledge to extract summaries," in *Science and Information (SAI)*. IEEE, 2016, pp. 61–70.

[16] M. Keyarsalan and G. A. Montazer, "Designing an intelligent ontological system for traffic light control in isolated intersections," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 8, pp. 1328–1339, 2011.

[17] A. Haller, K. Janowicz, S. J. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, "The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation," *Semantic Web*, vol. 10, no. 1, pp. 9–32, 2019.

[18] P. Morignot and F. Nashashibi, "An ontology-based approach to relax traffic regulation for autonomous vehicle assistance," *Computing Research Repository (CoRR)*, 2012. [Online]. Available: http://arxiv.org/abs/1212.0768

[19] T. Sharma, N. Tiwari, and D. Kelkar, "Study of difference between forward and backward reasoning," *International Journal of Emerging Technology and Advanced Engineering (IJETAE)*, vol. 2, no. 10, pp. 271–273, 2012.

[20] R. Belohlavek and J. Macko, "Selecting important concepts using weights," in *International Conference on Formal Concept Analysis (ICFCA)*. Springer, 2011, pp. 65–80.

[21] A. Mazak, B. Schandl, and M. Lanzenberger, "iweightings: Enhancing structure-based ontology alignment by enriching models with importance weighting," in *Complex, Intelligent and Software Intensive Systems (CISIS)*. IEEE, 2010, pp. 992–997.

[22] M. Vallejo and D. W. Corne, "Evolutionary algorithms under noise and uncertainty: A location-allocation case study," in *Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–10.

[23] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A survey on traffic signal control methods," *Computing Research Repository (CoRR)*, 2019. [Online]. Available: arXiv preprint arXiv:1904.08117

[24] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *Transactions on Intelligent Transportation Systems (T-ITS)*, 2021.

[25] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic traffic simulation using SUMO," in *Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2018, pp. 2575–2582.

[26] L. N. Alegre, "SUMO-RL," https://github.com/LucasAlegre/sumo-rl, 2019.