SUMO User Conference 2023
Conference paper
https://doi.org/10.52825/scp.v4i.222

Published: 29 June 2023

Challenges in Reward Design for Reinforcement Learning-based Traffic Signal Control: An Investigation using a CO_2 Emission Objective

Max Eric Henry Schumacher[1] [https://orcid.org/0000-0003-0486-3498],
Christian Medeiros Adriano[1] [1] [https://orcid.org/0000-0003-2588-9937], and
Holger Giese[1] [https://orcid.org/0000-0002-4723-730X]

[1] Hasso-Plattner Institute, University of Potsdam, Germany

Abstract: Deep Reinforcement Learning (DRL) is a promising data-driven approach for traffic signal control, especially because DRL can learn to adapt to varying traffic demands. For that, DRL agents maximize a scalar reward by interacting with an environment. However, one needs to formulate a suitable reward, aligning agent behavior and user objectives, which is an open research problem. We investigate this problem in the context of traffic signal control with the objective of minimizing CO_2 emissions at intersections. Because CO_2 emissions can be affected by multiple factors outside the agent's control, it is unclear if an emission-based metric works well as a reward, or if a proxy reward is needed. To obtain a suitable reward, we evaluate various rewards and combinations of rewards. For each reward, we train a Deep Q-Network (DQN) on homogeneous and heterogeneous traffic scenarios. We use the SUMO (Simulation of Urban MObility) simulator and its default emission model to monitor the agent's performance on the specified rewards and CO_2 emission. Our experiments show that a CO_2 emission-based reward is inefficient for training a DQN, the agent's performance is sensitive to variations in the parameters of combined rewards, and some reward formulations do not work equally well in different scenarios. Based on these results, we identify desirable reward properties that have implications to reward design for reinforcement learning-based traffic signal control.

Keywords: Traffic Signal Control, Reinforcement Learning, Reward Modeling, Pollutant Emissions

1 Introduction

Deep reinforcement learning (DRL) is a data-driven approach that holds promise for improving traffic signal control (TSC), because DRL can learn to adapt to changing traffic demands [1]–[3]. To achieve this, a DRL agent interacts with its environment and learns to take actions that maximize a cumulative scalar reward. By doing so, the agent can optimize the flow of traffic and improve the overall efficiency of the system.

In TSC, the actions correspond to changes in traffic lights and rewards correspond to traffic flow metrics (e.g., average vehicle speed, braking accelerations, and queuing lengths at intersections). However, in real-world applications of DRL, the agent's reward should also reflect the users' goals, which in TSC could be, to minimize traffic delays [4], [5] and CO_2 emissions [6], [7]. Nonetheless, it is not obvious how to select reward formulations that are also effective in satisfying users' goals. This is an open and challenging research problem known as the ”agent alignment problem”. The optimization goal of minimizing travel time in the context of TSC is challenging due to the influence of various external factors, such as free flow speed and current congestion level, which are beyond the agent's immediate control [8]. While this makes travel time an ineffective reward in practice [4], it is also not obvious which traffic flow metrics are guaranteed to be effective to this goal. There are several studies that combine traffic metrics as rewards for DRL agents [4], [5], [9]. Similarly, there is a growing body of research on training DRL agents to minimize pollutant emissions in TSC [6], [7]. However, these DLR-based approaches provide limited insight into how the convergence curves of traffic metrics behave relative to CO_2 emissions during the training of DRL agents. This information is important for designing reward functions that are effectively aligned with the users' goals. We investigate how to bridge this knowledge gap by performing a systematic study of the reward design space, which comprises single-metric rewards, combined-metrics and their corresponding parameterizations (weights in a linear function). For each candidate reward, we train a Deep Q-Network (DQN) [10] on two traffic scenarios, one with homogeneous traffic and one with heterogeneous traffic. To evaluate the various reward model formulations, we adopt the SUMO (Simulation of Urban MObility) simulator and its default emission model (Handbook Emission Factors For Road Transportation - HBEFA 3.1) [11]. Our evaluation consist of measurements of convergence curves of the agent's reward and the corresponding CO_2 emissions, producing the following results:

1. a CO_2 emission-based reward is inefficient for training a DQN agent,
2. only a few single-metric rewards were capable of minimizing CO_2 emissions,
3. metrics that individually did not produce effective reward formulations, were, when combined, successful in minimizing CO_2 emissions,
4. and, even when there exists an effective instance of a combined reward (e.g., a combination of queue and brake), there are still variations (i.e., from different parameterizations) of those same traffic flow metrics that produce ineffective rewards.

These results generalize both under homogeneous and heterogeneous traffic flow scenarios. Based on these results, we generated two contributions in the form of systematic analyses.

1. *Property-based analysis* of convergence curves. This analysis generates explanations for the cases of insufficient alignment between the agent's reward model and the CO_2 emission goal. The explanations consist of a paradigmatic classification of the reward models through orthogonal categories defined by two properties. *Informativeness* captures how well the agent approximates the given proxy reward, and *expressiveness* reflects how strong episode rewards correlate with episode CO_2 emission levels.
2. *Sensitivity analysis* of the challenges to align combined reward models with CO_2 emission goals. This analysis shows that alignment has two levels of sensitivity: the choice of traffic flow metrics, and the parameterization of these metrics in a linear reward formulation.

The remainder of the paper is organized as follows. In Section 2, we present the problem of agent alignment and its impact on TSC and emissions. We contextualize our work in relation to DRL for TSC, and for minimization of pollutant emissions (Section 3).

The approach and experimental setup is detailed in Section 4, while the corresponding results are presented in Section 5. The analyses of these results in terms of contributions, implications, and threats to validity are discussed in Section 6. Finally, we offer our conclusions and ideas for future work in Section 7.

2 Foundations

Deep reinforcement learning (DRL) is a popular approach that combines deep neural networks with reinforcement learning to enable agents to learn optimal behavior in complex environments. However, ensuring that the goals of the system align with the goals of the user is a critical challenge in DRL systems. Misaligned goals can result in unintended and potentially harmful outcomes that undermine the users' goals. This section examines the challenges that make alignment difficult in DRL systems and describes how reward models align with user goals.

2.1 The Agent Alignment Problem

The AI alignment problem [12] consists of finding ways to ensure that, quoting [13]: ”... these [machine learning] models capture our norms and values, understand what we mean or intend, and, above all, do what we want”. In other words, it involves matching agent rewards and users' goals regarding behavior [14], intent [15], incentive [16], inner and outer alignment [17], and instruction alignment [18]. Behavior alignment consists of producing predictions for given inputs, whereas intent looks at more general specification that cover different desired behaviors. Incentive alignment studies how rewards induce desired behaviors, whereas inner and outer alignment deals with partitioning the alignment in scopes that present specific dynamics. Instruction alignment consists of communicating human intent as a sequence of instructions that must be learned. These various definitions of alignment make specification, measurement, and evaluation challenging.

Therefore, a more pragmatic approach is to look at the failure of the agent to align with the user's goals (misalignment). Misalignment can have unintended consequences that are counterproductive (optimize against the users' goals), futile (no effect on users' goals), or simply could jeopardize users' goals (suboptimal behavior). Additionally, misalignment in DRL can increase the chances of reward hacking [19], [20]. For instance, in the case of a game boat race, an agent maximized a reward by indefinitely hitting a nearby target without ever concluding the race [21] – violating what the user intended.

One can argue for a proper definition for the user's goal and how it should be reflected on the reward model; however, this is still challenging, as evident in the many recent AI failure cases reported in the ”Artificial Intelligence Incident Database”[1]. In other words, there is no perfect alignment [15]. Instead, one needs to specify models that satisfy the conditions of being sufficiently *meaningful* and *precise* to steer the process of achieving user goals (e.g., reducing CO_2 emissions) by optimizing traffic flow metrics. For that, one needs a systematic way to evaluate how reward models align with user goals. Our approach presented in this paper is to partition the alignment specification problem into two metrics that allow to *express* a *meaningful* goal, and *inform* precisely enough how this goal can be achieved.

[1] https://incidentdatabase.ai/

2.2 Alignment Challenges

Partial observability in the form of hidden states (inherent to DRL environments) make alignment more difficult to achieve by preventing the agent from observing all the effects of its actions (in particular the delayed ones).The hidden states can result both from misspecified (wrong) [22] and underspecified (incomplete) [23] models. In the context of DRL, wrong or incomplete models can cause the agent to show good convergence curves at training time, but present unexpected behaviors after deployment. This can have consequences for the safety and cost of applications like autonomous vehicles and robotics.

Delayed and stochastic effects of actions are challenges when performing credit assignment, i.e., determining how each action contributed to achieve the users' goal. While delay and stochasticity cannot be eliminated, as they are properties of the environment, one can have reward models that are less sensitive to these factors. In the case of emissions, one can compare how different traffic flow metrics (e.g., average speed versus queue length) relate to changes in CO_2 emissions.

2.3 Deep Q-Network

Q-Learning is a popular reinforcement learning algorithm that helps agents make decisions based on rewards in their environment. It involves estimating the action-value function, which maps a state and action to the expected future rewards. In tabular Q-learning, the action-value function is represented as a table, but this becomes impractical for large or continuous state and action spaces [24]. Function approximation can solve this problem by representing the function using a neural network or another approximator.

Neural Fitted Q-Iteration (NFQ) [25] is an extension of tabular Q-learning with function approximation, improving scalability to large state-action spaces. However, NFQ uses a fixed dataset; thus, it is susceptible to overfitting on the training data. To mitigate this problem, Deep Q-Network (DQN) was introduced [10]. DQN builds on NFQ and introduces two key components: the experience replay buffer and the target network. The replay buffer stores the agent's experiences that can be retrieved for updating the Q-value estimates. The target network is used to set the TD targets, which are calculated based on the immediate reward and discounted future returns. Finally, our choice for DQN relied on its simplicity (off-policy and model-free), as it would allow to establish a comparison baseline for more sophisticated approaches like Proximal Policy Optimization algorithms (PPO) [26].

3 The State of the Art

This section introduces the topic of reward modeling in deep reinforcement learning (DRL) and its application to traffic signal control (TSC).

3.1 Reward Modeling

Reward modeling consists of learning to achieve specific user goals without requiring human feedback [14]. It has become a popular approach that precludes manually solving the credit assignment problem (e.g., via reward shaping [27]). However, because designed rewards can still be tampered by a learning agent [19], one still has to evaluate how alignment is done via reward modeling. This gives rise to the **Optimal**

Reward Problem - ORP [28], which aims to reduce the alignment problem to a reward modeling problem. This might involve defining intrinsic or extrinsic rewards [29]. The intrinsic reward constraints the agent on *how it can learn*, whereas the extrinsic reward instruments the user's goal by steering the agent on *what it can learn*[2]. We translate these intuitions respectively into two convergence properties named *informativeness* and *expressiveness* (formalized in Section 6.1).

3.2 Reward Models for Traffic Signal Control

Minimizing travel time is the main goal of a TSC policy. However, because travel time is affected by a multitude of factors and actions with delayed effects [8], traffic engineers rely on proxy reward metrics, like average waiting time, average intersection speed, or total braking acceleration. Accordingly, in DRL, various combinations for a reward models were investigated: queue length and delay in [5], queue length and pressure in [4], stop time and average speed and time lost [9], and many others (see Table-5 in [8]). We extend this family of work by combining more metrics (vehicle speed, brake acceleration) and evaluating their impact on CO_2 emissions.

3.3 Pollutant Emissions in Traffic

Traditionally, the first solutions comprised non-DRL control (both with SUMO [31], [32] and other simulators [33]–[35]). More recently, DRL-based TSC approaches to minimize pollutant emissions have been investigated [6], [7]. However, these DLR-based approaches provide limited understanding about the relationship between metrics for emissions and traffic flow, in particular, regarding how the convergence curves of metrics behave during the training of DRL agents. Without a proper understanding of this relationship, one is hindered in the task of reward modeling for aligning the agent's reward with CO_2 emission goals in TSC. Therefore, to bridge this gap, we investigated various formulations for a linear reward function based on traffic flow metrics, and computed the corresponding CO_2 emissions using SUMO's provided emission model from the Handbook Emission Factors For Road Transportation (HBEFA 3.1) [11].

3.4 Deep Reinforcement Learning for Traffic

The specification of the DRL approach goes beyond the choice of reward function: one needs to choose an algorithm and how to model the state-space. Among the many DRL algorithms to have been adopted [8], the DQN [10] algorithm has been one of the most popular choices (Table-1 in [3]). The adoption of DQN for TSC stems from its relative simplicity of having discrete actions, while still providing good convergence behavior [36].

Concerning the **state-space**, the traffic environment has been modeled at various levels of resolution, from coarse (flow) to fine (vehicle speed and position) [8], resulting in tabular discretized metrics [37], and image representations [36]. We opted for a lane segment level resolution and discretized metrics because studies could not show better results when adopting higher resolution [38] or more complex state representations [5].

[2] This could involve curiosity-driven exploration [30], which attributes credit based on the novelty of the state-action pair, usually measured by some information theoretic metric, e.g., entropy, mutual-information, or KL-divergence.

4 Methodology

In this section, we outline the methods used to study CO_2 emissions produced at signalized intersections. Our approach builds on the principles of reinforcement learning, where an agent learns to make decisions based on the interaction with its environment. We outline the traffic simulation scenario in Section 4.1 and formulate the reinforcement learning task in Section 4.2, defining the states, actions, and rewards used by the agent. Finally, in Section 4.3, we provide a detailed description of the experimental setup, including the neural network architecture, hyperparameters, and the setup of the traffic environment, used to train and evaluate the DQN algorithm.

4.1 Traffic Scenarios

We propose a scenario that comprises a controlled intersection (shown in Fig. 1), featuring two incoming and two outgoing lanes. The intersection allows two types of phases: either green or yellow in the north-south direction (NSG, NSY); or green or yellow in the east-west direction (WEG, WEY). In both cases, the orthogonal direction is set to red. Fig. 1 illustrates the intersection in NSG phase.

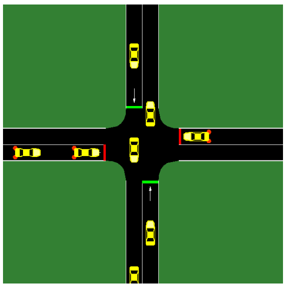


Figure 1. Screenshot (SUMO GUI) of a signalized intersection with four lanes.

We combine this infrastructure with traffic flows as shown in Fig. 2, consisting of two types: a time-varying Bernoulli distribution, and a traffic flow that remains constant throughout the simulation. At each second and on each road (north-south, west-east, etc.), a car is released into the simulation with a probability of p. Each traffic demand combined with the signalized intersection infrastructure gives rise to one scenario: a *heterogeneous traffic scenario*, using the time-varying demand, and a *homogeneous traffic scenario* (using the fixed demand).

For the heterogeneous traffic scenario, depicted in blue in Fig. 2, we deliberately chose a peak traffic volume of $p = 0.25$ – the maximum probability of releasing a car. This level of peak traffic makes the scenario challenging, as it exceeds the maximum intersection throughput and causes congestion temporarily. In contrast, the homogeneous traffic flow, depicted in red in Fig. 2, has a fixed probability of releasing a car with a value of $p = 0.2$. This value represents the maximum intersection throughput, ensuring that the flow remains steady throughout the simulation.

4.2 The Reinforcement Learning Task

Traffic signals play a critical role in ensuring safe and efficient traffic flow at intersections. Fixed pre-timed controllers are often insufficient in optimizing traffic flow, as traffic volume and driving behavior vary widely. Adaptive traffic signal control (ATSC) provides a solution, which uses electrical sensors and sets signals based on the data, adapting

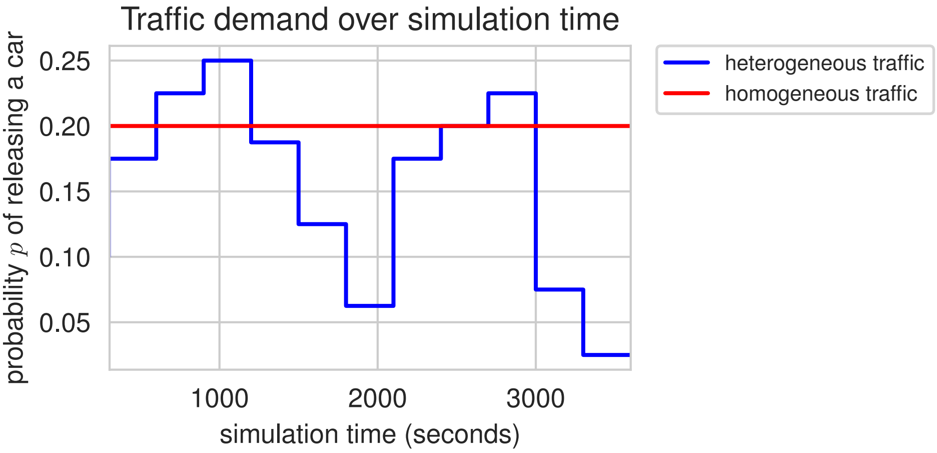
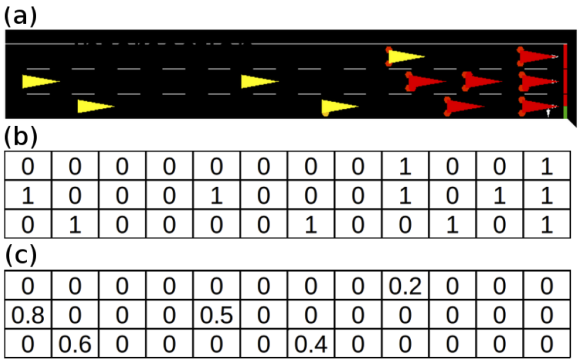


Figure 2. The probabilities of releasing cars into the system over simulation time.

to the current traffic situation. One of the simplest methods to achieve ATSC is actuated signal control, which triggers a specific signal based on sensory data gathered around the intersection. Reinforcement learning (RL) is a possible solution to obtain a program for ATSC. The output of the RL algorithm, the agent's policy, becomes the desired ATSC program, which works fully automated, and can be scaled. ATSC with DRL has achieved outstanding results, outperforming conventional methods in many situations. The agent repeatedly collects state information, acts, and updates its policy with a scalar reward, while being trained in safe or simulated environments. For the remainder of this Section, we will assume the environment described in Section 4.1 and specify the components of the reinforcement learning problem, the states, actions and rewards of the agent.

The agent's state or observation is a representation of the environment that the agent perceives at any given time, including relevant information that the agent can use to take actions that maximize its rewards. In the case of traffic signal control with reinforcement learning, the DTSE (Discrete Traffic State Encodings) state [39] is a commonly used representation that consists of two 2D matrices. The first matrix is a binary position matrix that encodes the presence or absence of a vehicle at each intersection, as depicted in Fig. 3 (b). The second matrix is a normalized velocity matrix that tracks the average speed of the vehicles on a given segment, as depicted in Fig. 3 (c).



(b)

0	0	0	0	0	0	0	0	1	0	0	1
1	0	0	0	1	0	0	0	1	0	1	1
0	1	0	0	0	0	1	0	0	1	0	1

(c)

0	0	0	0	0	0	0	0	0.2	0	0	0
0.8	0	0	0	0.5	0	0	0	0	0	0	0
0	0.6	0	0	0	0	0.4	0	0	0	0	0

Figure 3. Example of simulated traffic (a) with corresponding Boolean- (b) and Real-valued DTSE vectors (c). Image source: [39]

Our approach uses DTSE representations, which capture the position and speed of vehicles – key factors in determining CO_2 emissions. This allows the agent to make informed decisions about when to change traffic lights to achieve the goal of our RL task, which is to minimize CO_2 emissions.

The agent's actions are determined by the traffic scenario, as described previously. That is, the agent takes action every Δt (in seconds) and chooses from the set of allowed phases $\mathcal{A} = \{NSG, WEG\}$. Additionally, on each phase change, a yellow transition phase (NSY or WEY) is induced to ensure safety. In contrast to our approach, the agent could also cycle through a pre-defined sequence or operate in non-fixed intervals. However, using a fixed action interval with a set of allowed phases provides a balance between flexibility and difficulty, as non-fixed intervals make the problem harder, and a pre-defined sequence limits the agent's options.

The agent's reward is composed of one or multiple of the following average traffic metrics, aggregated over all lanes: queuing length (queue reward), vehicle speed (speed reward), braking acceleration (brake reward), and CO_2 emission rates (emission reward). Additionally, we provide linear combinations of average queuing length and braking acceleration (queue+brake reward) as well as queuing length and speed metrics (queue+speed reward).

4.3 Experimental Setup

Each experiment uses one of the intersection scenarios described in Section 4.1, with either heterogeneous or homogeneous traffic. Each training run uses simulations that last for 3600 seconds (simulation time), and the agent interacts in intervals of $\Delta t = 5s$, resulting in 720 steps $t = 1, \ldots, 720$ per episode. At episode termination, the simulation is reset, and the agent continues training. For a phase-switch, we selected a yellow time to of $t_{yellow} = 2s$. The agents observe DTSE features with speed and position information. To compute DTSE features, we split each road into 30 segments (segments of length $c \approx 8.33m$). Table 1 summarizes this general setup.

The DQN agent uses a Multi-Layer Perceptron (MLP) with two hidden layers as the neural network, each containing 64 neurons, and a linear output layer with four neurons (one for each action). We use the Adam optimizer [40] for mini-batch gradient descent, with a batch size of 64 and an initial learning rate of $\alpha = 1e-4$. To explore the environment, the agent begins with 100% exploration (ϵ = 1) and gradually decreases exploration linearly to 10% over the first third of training. The replay buffer holds up to 2000 samples, and learning begins after the first episode (720 steps of initial experience). The target network is updated every $C = 10000$ (steps), and the agent's discount factor is $\gamma = 0.99$, which captures long-term rewards. Hyperparameters and training setup are summarized in the second section of Table 1.

5 Results

This section is organized into three parts. In Section 5.1, we evaluate the suitability of CO_2 emission rates as a reward. In Section 5.2, we compare the performance of agents trained on proxy rewards to those trained on a CO_2 reward. Finally, in Section 5.3, we examine how different combinations of reward parameters impact agent's alignment.

Table 1. Environment, hyperparameter and training setup.

Parameter	Value	Description
episode length	$3600s$	episode length in seconds
Δt	$5s$	interval in which the agent interacts in seconds
T	720 steps	number of agent-environment interactions in an episode
t_{yellow}	$2s$	yellow transition time for phase switches
$\mathcal{A}$	$\{NSG, WEG\}$	action space of the agent
$\mathcal{S}$	DTSE	the agent's observable state space
c	$8.33m$	length of a DTSE segment
optimizer	Adam	optimizer
α	$1e^{-4}$	learning rate
batch-size	64	mini-batch size
buffer-size	2000	size of the replay buffer
learning starts	720 steps	number of steps of initial exploration without learning
C	10000 steps	update interval for the target-network of DQN
γ	0.99	discount factor

5.1 CO_2 Emissions as Reward

In Fig. 4 we show the performance of two DQN agents: one agent was trained on a speed reward, and the other agent was trained on the CO_2 emission reward. The solid line represents the median episode emission rate in g/h, and the shaded area shows the 95% confidence intervals. Our results demonstrate that while the agent trained on the CO_2 emission reward does improve in the first episode of training, it converges to a higher emission rate than the agent trained on the speed reward, and does not show any further improvement over time.

These findings suggest that training with the CO_2 reward leads to suboptimal behavior, as the agent is constrained in maximizing this reward and fails to learn an effective policy for minimizing CO_2 emissions. In contrast, the agent trained on the speed reward is able to converge to a better policy for emission minimization, ultimately achieving a lower emission rate.

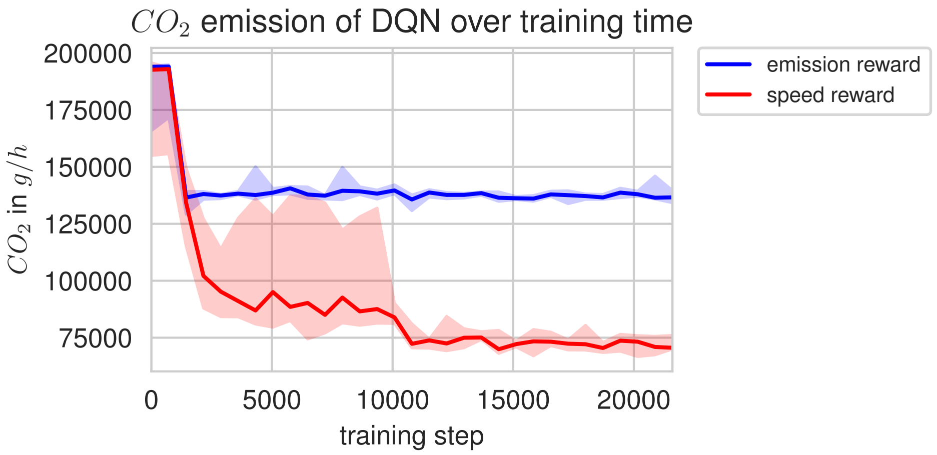


Figure 4. Two agents' performance on minimizing CO_2 emissions by following distinct formulations of cumulative reward. The blue agent has an emission-based reward formulation, whereas the red agent has a speed-based formulation.

5.2 Proxy Rewards for CO_2 Minimization

To explore alternative approaches to incentivizing emission reduction, we investigate the use of various proxy rewards in this section. Specifically, we analyze the performance of DQN agents trained on rewards based on queue lengths, braking accelerations, average speed, CO_2 emissions, a combination of queue length and braking acceleration, and a combination of queue length and average speed.

We present the results of our experiments in Fig. 5. This figure summarizes the performance of each agent on the different reward models, with each subplot showing the CO_2 emission rates (red line), proxy reward (green line), and maximum observed proxy rewards (dotted yellow line). In addition, the shaded areas in each subplot represent 95% confidence intervals for the emission rates and rewards.

Based on the results presented in Fig. 5, we observe that the DQN agent trained on the CO_2 emission reward converged to a suboptimal policy after one episode, resulting in comparatively high emission levels. Similar behavior was observed for the DQN agent trained on the queue reward, which achieved a reduction in CO_2 emissions, but at suboptimal levels. The agent trained on the brake reward had a positive correlation between CO_2 emissions and the episode reward, leading to no reduction in CO_2 emissions.

Good emission performance was achieved by agents using a speed reward and a combined queue and brake reward, denoted as queue-brake reward. The DQN agent trained on the speed reward achieved a relatively low CO_2 emission rate, while also achieving the highest speed reward among all agents. The DQN agent trained on the queue-brake reward achieved the lowest CO_2 emission levels so far, showing a negative correlation with CO_2 emissions (see Table 2).

Overall, the queue-brake reward was the most effective in reducing CO_2 emissions, while the speed reward was effective in achieving a relatively low CO_2 emission rate and high speed reward. Conversely, the emission and queue rewards resulted in suboptimal emission levels.

We calculated the degree of association between the episode CO_2 emissions and episode rewards as a measure of the behavior of the agent alignment (see Table 2). For that, we adopted the Kendall-*tau* rank correlation coefficient[3].

Table 2. Kendall-*tau* correlations (τ) between episode rewards and episode emissions. All values were statistically significant (*p-value* $\leq$ 0.05).

reward	τ	*p-value*
emission	-1.000	2.7e-91
speed	-0.832	8.0e-64
queue	-0.361	2.8e-13
brake	0.505	1.5e-24
queue-brake	-0.893	2.9e-73

5.3 Sensitivity to Reward Parameters

In this experiment, we explored the impact of reward parameter combinations on the performance of a DQN agent in managing traffic flow with the aim of minimizing CO_2

[3] The Kendall-*tau* coefficient is a non-parametric statistic that quantifies the strength and direction of association between two variables without assuming any specific distribution [41].

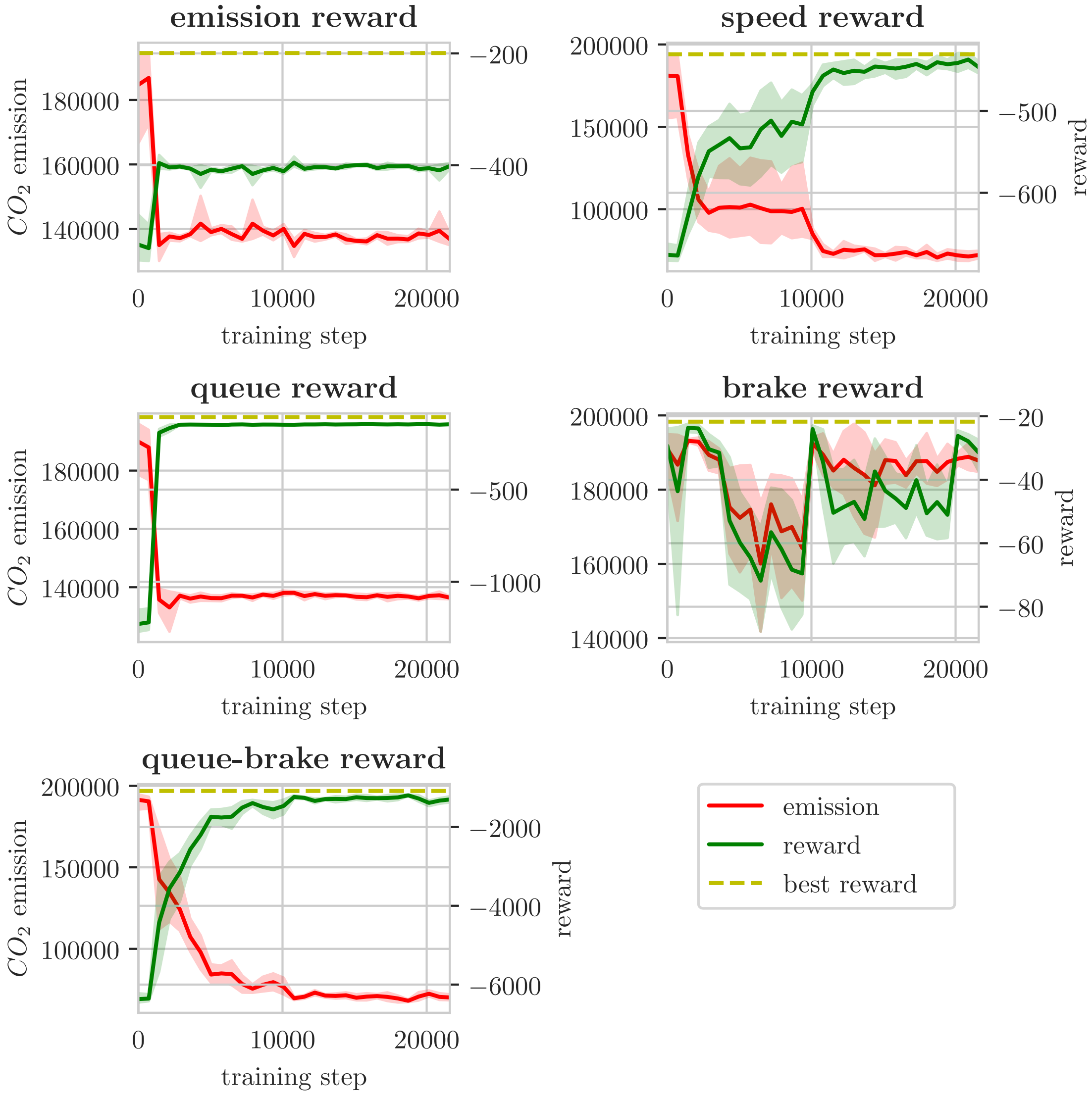


Figure 5. CO_2 emission rates in g/h (red) and absolute episode emission reward (green) of DQN agents over training time. The solid lines depict the median values, while the shades depict 95% confidence intervals. Each reward is combined to a "best reward" (yellow) that corresponds to the highest value on this reward that was observed among all agents (trained with various rewards).

emissions. We varied the ratio of queue and brake in a combined reward, and queue and speed in a combined reward. For each combination of metrics, we trained six DQN agents for 21800 steps and evaluated their performance in both heterogeneous and homogeneous traffic scenarios.

In Fig. 6 we show the average episode CO_2 emissions in g/h (y-axis) and weightings of queue and brake reward (x-axis). We observed that a combination of both queue and brake reward was necessary to achieve the lowest CO_2 emissions.

Interestingly, we found that the combination ratio of $(queue, brake) = (0.5, 0.5)$ provided the best performance across both traffic scenarios. Additionally, we observed that combinations close to $(queue, brake) = (1.0, 0)$ or $(0, 1.0)$ demonstrated similar performance to those combinations. This suggests that the agent focuses on only one reward parameter, which does not lead to the best outcome.

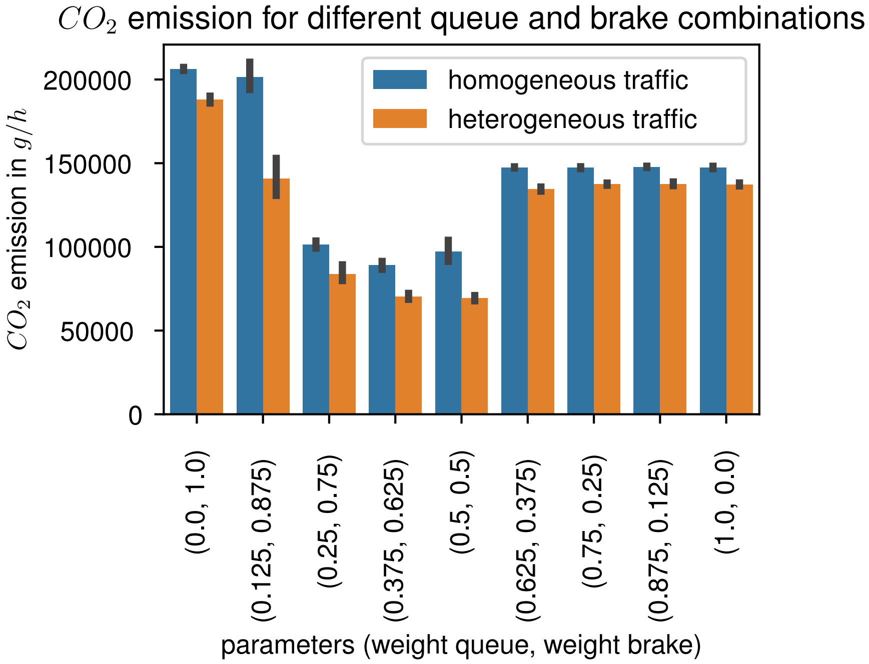


Figure 6. Episode CO_2 emission rates in g/h (y-axis) for DQN agents trained with various weighted combinations of queue and brake reward (x-axis).

For a combined queue speed reward, we would expect to see a similar trend in terms of the combination of rewards required to achieve good performance. However, as shown in Fig. 7, we observed that the level of queue metric must be zero (or close to zero) to achieve good performance.

Overall, our findings highlight the importance of reward parameter selection in training agents to optimize traffic flow and minimize CO_2 emissions.

6 Discussion

Next we discuss the results in terms of general properties for reward models, implications for modeling, and the threats to the validity of our results.

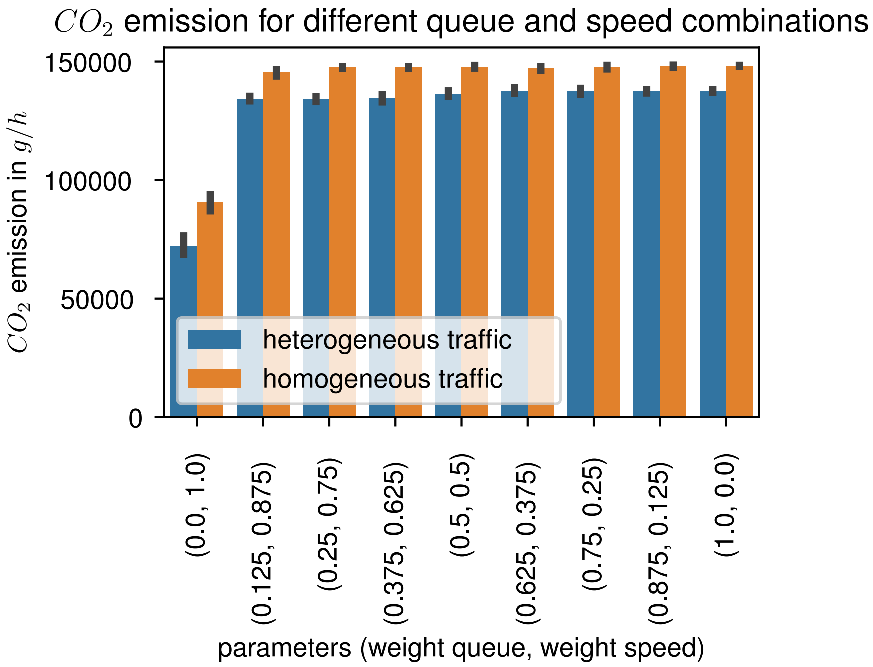


Figure 7. Episode CO_2 emission rates in g/h (y-axis) for DQN agents trained with various weighted combinations of queue and speed reward (x-axis).

6.1 Informativeness and Expressiveness

The two convergence curves shown in Section 5 correspond to: (1) how well the reward model *informs* the agent towards achieving the user's goal (CO_2) and (2) how well the reward model *expresses* the behavior of the emission. We call these two properties of the reward model *informativeness* and *expressiveness*. In other words, if the agent fails to converge to the optimal reward, we deem the reward model uninformative (see queue and emission reward in Fig. 5). Meanwhile, if the agent optimizes in the wrong direction, in our case positive correlation between reward and emissions (see brake reward in Table 2), then the reward model is not expressive.

These two properties are important because together they indicate if the agent is sufficiently aligned to the user's goals (minimizing CO_2 emissions). The judgment of sufficient alignment depends on how informative and expressive a reward model is. This is challenging because *informativeness* and *expressiveness* are continuous metrics based, respectively, on the measures of distance (from optima) and correlation (between reward and goal). Therefore, for the purpose of illustration and discussion, we assumed two arbitrary thresholds, which we introduce next.

Informativeness. A reward model (R_{mod}) is informative ($I(R_{mod}) = 1$) if the distance between the reward at convergence (R_{con}) and the optimal reward (R_{opt}) is smaller than δ. Formally, we have

$$I(R_{mod}) = \begin{cases} 1 & \text{if } (|R_{con} - R_{opt}| < \delta) \\ 0 & \text{otherwise} \end{cases}, \qquad (1)$$

where R_{con} is the episode reward and R_{opt} is R_{con} of the best performing agent regarding that reward.

Expressiveness. A reward model (R_{mod}) is *expressive* ($E(R_{mod}) = 1$) if the correlation ($Corr$) between the sequence of the agent's episode rewards (R) and the corresponding episode CO_2 emissions (G) has a certain direction (positive or negative) and its magnitude is above a certain strength (ρ). The correlation should be negative ($\in [-1, \rho]$) if the user's goal G has to be minimized, otherwise positive ($\in [\rho, 1]$).

We formalized E for the case where G has to be minimized.

$$E(R_{mod}) = \begin{cases} 1 & \text{if } (Corr(R, O) \in [-1, \rho]) \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where the magnitude ρ depends on the use case. For the purpose of illustration and discussion, we set next $|\rho| \geq 0.30$, which corresponds to at least a medium strength correlation [42] and be negative (as it minimizes emissions), hence, the threshold becomes $Corr(\cdot) \in [-1, -0.30]$.

Applying these formulas (Eq. (1) and Eq. (2)) as threshold criteria for classification, we populated a Venn diagram (Fig. 8) with the results from Section 5. The intersection area shows the reward models that are both expressive and informative, hence, they are considered to be sufficiently aligned with users' goals (minimize CO_2 emissions). Only the brake reward is considered not expressive, whereas queue, emission, and queue-speed rewards are considered non-informative. Next, we discuss the implications of this classification.

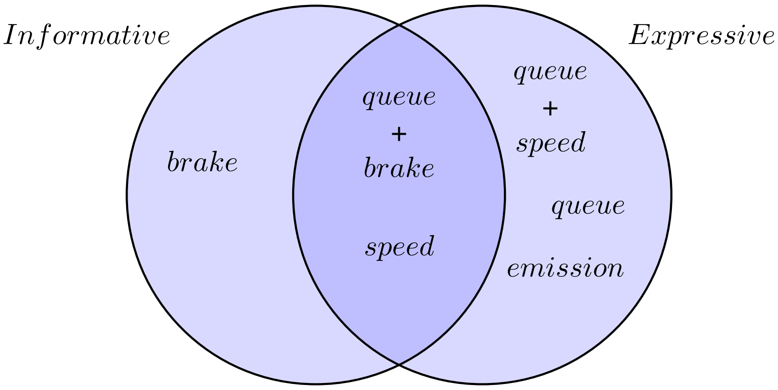


Figure 8. Classification of the rewards in terms of their *informativeness*, *expressiveness* and *alignment* (intersection).

6.2 Implications

Independent properties. The informativeness property did not necessarily imply expressiveness, and vice versa. Therefore, one has to monitor both properties while designing reward models. This is an additional requirement that involves a careful study of the thresholds that lead to agent alignment – satisfying users' goals. **Uninformativeness detection.** Many of the reward models that were deemed uninformative showed a very early convergence to a local minimum, e.g., queue reward and emission reward – their green curves follow a step-like function (Fig. 5). This suggests that the reward models provided a target that was too easy to learn; in other words, the agent is overfitting to the data collected in the first epoch.

Combining metrics. The design of the reward model should therefore incorporate metrics that make learning more challenging, for instance, with properties that are less correlated with emissions (lower *expressiveness*). This might explain why a combination of brake (low *expressiveness*) and queue (low *informativeness*) produced a sufficiently aligned agent, minimizing CO_2 emissions. Looking at the convergence curve of the proxy reward, green curve in Fig. 5, we can see diminishing returns over time, which suggests an increasing degree of difficulty for the agent to learn better policies as the training progresses. In other words, relative to the queue reward model, adding a brake metric made the learning more difficult. Conversely, adding the queue metric to the brake reward model provided the *expressiveness* that was missing.

Complementary properties. However, looking for complementary properties is not enough. Take, for instance, speed and queue metrics. Although the speed reward model is complementary to the queue reward model regarding *informativeness* and *expressiveness* – speed reward has higher correlation with emissions than the queue reward (see Table 2) – the combination of queue-speed did not produce an aligned agent. As we can see in Fig. 8, queue+speed convergence is categorized as *expressive*, but not *informative*. This is confirmed by the sensitivity analysis of the parameter weights for speed+queue combined reward (see Fig. 7).

Reward parameterization. Choosing the right traffic metrics to combine is not enough. One still has to decide on the weights that each metric should have in the reward model. While for the queue-brake reward we showed an optimum region (see Fig. 6), there is no guarantee that the combination of other metrics would present the same global optimum. This is important to design methods that systematically and efficiently look for the optimal parameterization. The shape of this parameterization space determines how informative and expressive a reward model should be to be considered aligned to the users' goal. Because a search in this space could be seen as a balance between exploitation (following an informative signal) and exploration (expressing desired behavior), one has to decide how to measure these properties. We note that assuming that these properties have uniform values during training is not realistic.

Property uncertainty. Defining how expressive or how informative a reward is might require new properties, for instance, properties that evaluate the uncertainty in the learning (convergence) process. The brake reward model illustrates this case, where there is larger than 10% variance in reward (green curve in Fig. 5) in the second half of training. This makes it challenging to decide how many training steps to execute or when training should stop, because slightly different stopping points could produce very different policies. Ideally, an engineer would like to know about the trade-off between reward model simplicity (only use the brake metric) and the risk of suboptimal rewards (high uncertainty at convergence).

6.3 Threats to Validity

Threats to validity [43], [44] act in ways that can hinder the reproducibility of the experimental results and corresponding interpretations.

Internal Validity evaluates if the causes of the measured effects can be attributed to our experimental design decisions [45]. In our case, we chose a benchmark (the best reward across agents - dotted lines in Fig. 5) and a set of proxy reward metrics (*speed*, *brake*, *queue*). We computed the effects on CO_2 emissions by varying the weights of metrics on combined reward models (e.g., X-axis in Fig. 6). When we claim that a given reward model is more or less informative or expressive, we are interpreting a measurement, i.e., the effect of a parameterization choice, that can still be confounded by what we did not control for, i.e, the other metrics not included in the given reward model, which might still indirectly affect the CO_2 emissions. To improve internal validity, we suggest more extensive simulations with more complex scenarios, for instance, by including real-world data.

External Validity discusses the situations in which the research outcomes might not generalize beyond the current experimental setup [45] comprised by both dataset and models parameters. Concerning the dataset, we showed similar results in two distinct scenarios of traffic demand. Although this might be a straightforward mitigation of the external validity threat (Fig. 6), a recent survey [2] reported that only seven out of 21 studies evaluated their models under distinct traffic scenarios. With respect to parameterization, we showed that certain pairs of weights for the queue-break reward produced suboptimal CO_2 emissions (see the extremes of the bar chart in Fig. 6). This highlights the challenge to generalize the combined reward results across a range of parameter values.

Construct Validity concerns the situations for which the performance indicators (thresholds) do not measure the actual concepts (constructs) [45]. This might happen because of bias in data generation, incorrect definitions, or inappropriate analysis methods (see Statistical Conclusion Validity). In our study, the mismatch between thresholds and the convergence properties (constructs) can happen through misspecification of the reward model and the properties themselves. One example is mistakenly deeming a reward model to be informative or expressive enough, when it is not. The reason for the mistake could be an inappropriate threshold or a reward model that is incomplete. To mitigate that, we specified convergence properties that are independent of the traffic signal control domain, but can be easily instantiated by choosing classification thresholds that are meaningful to what a user consider to be a sufficiently aligned agent reward model.

Statistical Conclusion Validity concerns the violations in the assumptions of the adopted statistical methods [45]. One example of possible violation is wrong assumption of normal data distributions. As we worked with small samples of reward outcomes, we adopted a non-parametric method (Kendall-tau) to compute the correlations, which we reported with their corresponding p-values (Table 2). Regarding conclusions about categorization within the two properties (*informativeness* and *expressiveness*), although we specified thresholds that were appropriate to discriminate among convergence curves, we did not take into account the inherent uncertainty in the convergence curves. A possible improvement could be to incorporate uncertainty measurements to the convergence analysis, for instance, the reward variance at the late training stages (so to ideally minimize it).

7 Conclusion

In the theory of bounded rationality [46], agents are bounded in their learning by the quality of the information they can access. We investigated this essential limitation in terms of the reward model, which we evaluated concerning the agent's alignment with the users' goals. Our main result is that, for the agent alignment with the goal of minimizing CO_2 emissions, it is *necessary* that the corresponding reward model formulation be both *expressive* and *informative*.

7.1 Results and Contributions.

Results. We showed that not all reward models are sufficiently aligned with users' goals (e.g., the models outside the intersection set depicted in Fig. 8). These results were reproduced in two distinct traffic scenarios. The sufficiently aligned reward models shared the characteristic of being both *informative* and *expressive*. However, the result from queue+speed indicated that to determine if an agent is aligned, it is not *sufficient* to look at the properties of the single-metric based rewards. Only after combining the individual traffic flow metrics into a properly parameterized reward formulation, one can ultimately assess the agent alignment (again by evaluating its convergence properties).

Contributions. We provided two systematic analyses: (1) a property-based paradigmatic classification for explaining the failure of an agent to align with users' goals and (2) a sensitivity analysis for explaining the challenges of aligning combined reward models with CO_2 emission goals.

7.2 Future work

Towards principles for reward model selection. We showed that combining complementary metrics worked to some extent. However, some outcomes are still counter-intuitive, i.e., we do not know how to predict good and bad combinations based on the properties of single-metrics rewards. This is critical, because one still has to rely on post hoc explanations (as we showed), instead of relying on principles to prioritize reward model combinations systematically.

Reproducibility in more challenging scenarios. A natural step is to reproduce our findings in more complex situations, for instance, incorporating real-data[4] to the simulations and a larger set of traffic flow metrics. Besides creating opportunities to falsify our current claims, we could explore more challenging questions like the effects of partial observability and confounding in reward modeling for agent alignment in TSC.

Alternative convergence property formulations. In order to evaluate non-linear relationships, we plan to study *expressiveness* in terms of mutual information or metrics like Wasserstein distance. Concerning *informativeness*, we plan to look at methods that incorporate variance as a criterion of quality of convergence.

Data availability statement

To promote the reproducibility of our results, we made our experimental setup, source code, and dataset publicly available[5], which further facilitates extension of our results and serves as a baseline for comparing different approaches w.r.t. reward modeling. Note that the convergence properties are independent of the choice of traffic flow metrics, therefore, they can be instantiated for TSC environments with different sensors and state-space configurations.

Author contributions

Max Eric Henry Schumacher contributed with the software implementation, performing the experiments for data/evidence collection, and writing the original draft. Christian Medeiros Adriano contributed with the conceptualization, methodology, and writing the original draft. Holger Giese contributed with the conceptualization and supervision.

Competing interests

The authors declare that they have no competing interests.

[4] This would also help bridge the gap in which real-world traffic data is still a minority among traffic simulation studies [2]

[5] https://github.com/EricSchuMa/reward-design-TSC. Our repository extends the publicly available OpenAI Gym interface for SUMO [47].

Funding

The authors are grateful for the funding and infrastructure provided by the Hasso-Plattner Institute.

References

[1] C. Louw, L. Labuschagne, and T. Woodley, “A comparison of reinforcement learning agents applied to traffic signal optimisation,” in *SUMO Conference Proceedings*, vol. 3, 2022, pp. 15–43.

[2] H. Wei, G. Zheng, V. Gayah, and Z. Li, “Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation,” *ACM SIGKDD Explorations Newsletter*, vol. 22, no. 2, pp. 12–18, 2021, Publisher: ACM New York, NY, USA.

[3] A. Haydari and Y. Yilmaz, “Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–22, 2020, ISSN: 1558-0016. DOI: 10.1109/TITS.2020.3008612.

[4] H. Wei, C. Chen, G. Zheng, *et al.*, “Presslight: Learning max pressure control to coordinate traffic signals in arterial network,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1290–1298.

[5] G. Zheng, X. Zang, N. Xu, *et al.*, “Diagnosing reinforcement learning for traffic signal control,” *arXiv preprint arXiv:1905.04716*, 2019.

[6] J. Kim, S. Jung, K. Kim, and S. Lee, “The real-time traffic signal control system for the minimum emission using reinforcement learning in v2x environment,” en, *Chemical Engineering Transactions*, vol. 72, pp. 91–96, Jan. 2019, ISSN: 2283-9216. DOI: 10.3303/CET1972016. [Online]. Available: https://www.cetjournal.it/index.php/cet/article/view/CET1972016 (visited on 09/21/2022).

[7] A. Haydari, M. Zhang, C.-N. Chuah, and D. Ghosal, “Impact of deep rl-based traffic signal control on air quality,” in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, ISSN: 2577-2465, Apr. 2021, pp. 1–6. DOI: 10.1109/VTC2021-Spring51267.2021.9448639.

[8] H. Wei, G. Zheng, V. Gayah, and Z. Li, “A Survey on Traffic Signal Control Methods,” *arXiv:1904.08117 [cs, stat]*, Jan. 2020, arXiv: 1904.08117. [Online]. Available: http://arxiv.org/abs/1904.08117 (visited on 01/16/2022).

[9] A. C. Egea, S. Howell, M. Knutins, and C. Connaughton, “Assessment of Reward Functions for Reinforcement Learning Traffic Signal Control under Real-World Limitations,” in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, ISSN: 2577-1655, Oct. 2020, pp. 965–972. DOI: 10.1109/SMC42975.2020.9283498.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[11] D. Krajzewicz, M. Behrisch, P. Wagner, R. Luz, and M. Krumnow, “Second Generation of Pollutant Emission Models for SUMO,” en, in *Modeling Mobility with Open Data*, M. Behrisch and M. Weber, Eds., Series Title: Lecture Notes in Mobility, Cham: Springer International Publishing, 2015, pp. 203–221, ISBN: 978-3-319-15023-9 978-3-319-15024-6. DOI: 10.1007/978-3-319-15024-6_12. [Online]. Available: http://link.springer.com/10.1007/978-3-319-15024-6_12 (visited on 11/21/2022).

[12] E. Yudkowsky, “The AI alignment problem: why it is hard, and where to start,” *Symbolic Systems Distinguished Speaker*, 2016.

[13] B. Christian, *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020.

[14] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg, "Scalable agent alignment via reward modeling: A research direction," *arXiv preprint arXiv:1811.07871*, 2018.

[15] P. Christiano, *"clarifying ai alignment"*, 2018. [Online]. Available: https://ai-alignment.com/clarifying-ai-alignment-cec47cd69dd6.

[16] T. Everitt, R. Carey, E. D. Langlois, P. A. Ortega, and S. Legg, "Agent incentives: A causal perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 11 487–11 495.

[17] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, and S. Garrabrant, "Risks from learned optimization in advanced machine learning systems," *arXiv preprint arXiv:1906.01820*, 2019.

[18] L. Ouyang, J. Wu, X. Jiang, *et al.*, *Training language models to follow instructions with human feedback*, arXiv:2203.02155 [cs], Mar. 2022. [Online]. Available: http://arxiv.org/abs/2203.02155 (visited on 12/08/2022).

[19] T. Everitt, M. Hutter, R. Kumar, and V. Krakovna, "Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective," *Synthese*, vol. 198, no. Suppl 27, pp. 6435–6467, 2021.

[20] M. Cohen, M. Hutter, and M. Osborne, "Advanced artificial agents intervene in the provision of reward," *AI Magazine*, vol. 43, no. 3, pp. 282–293, 2022.

[21] J. Clark and D. Amodei, *Faulty reward functions in the wild*, Dec. 2016. [Online]. Available: https://openai.com/blog/faulty-reward-functions/.

[22] J. Skalse and A. Abate, "Misspecification in inverse reinforcement learning," *arXiv preprint arXiv:2212.03201*, 2022.

[23] A. D'Amour, K. Heller, D. Moldovan, *et al.*, "Underspecification presents challenges for credibility in modern machine learning," *Journal of Machine Learning Research*, 2020.

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

[25] M. Riedmiller, "Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method," in *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16*, Springer, 2005, pp. 317–328.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, Citeseer, vol. 99, 1999, pp. 278–287.

[28] J. Sorg, R. L. Lewis, and S. Singh, "Reward design via online gradient ascent," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23, Curran Associates, Inc., 2010. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2010/file/168908dd3227b8358eababa07fcaf091-Paper.pdf.

[29] S. Singh, R. L. Lewis, and A. G. Barto, "Where do rewards come from," in *Proceedings of the annual conference of the cognitive science society*, Cognitive Science Society, 2009, pp. 2601–2606.

[30] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," in *Seventh International Conference on Learning Representations*, 2019, pp. 1–17.

[31] K. Belhassine, J. Renaud, L. Coelho, and V. Turgeon, "Signal priority for improving fluidity and decreasing fuel consumption," in *SUMO Conference Proceedings*, vol. 3, 2022, pp. 159–169.

[32] J. E. L. Quichimbo, J.-A. Moreno-Perez, E. Lorenzo-Sáez, *et al.*, "Estimation of green house gas and contaminant emissions from traffic by microsimulation and refined origin-destination matrices: A methodological approach," in *SUMO Conference Proceedings*, vol. 1, 2020, pp. 27–37.

[33] B. De Coensel, A. Can, B. Degraeuwe, I. De Vlieger, and D. Botteldooren, "Effects of traffic signal coordination on noise and air pollutant emissions," en, *Environmental Modelling & Software*, vol. 35, pp. 74–83, Jul. 2012, ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2012.02.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364815212000576 (visited on 09/06/2021).

[34] Y. Zhang, X. Chen, X. Zhang, G. Song, Y. Hao, and L. Yu, "Assessing effect of traffic signal control strategies on vehicle emissions," en, *Journal of Transportation Systems Engineering and Information Technology*, vol. 9, no. 1, pp. 150–155, Feb. 2009, ISSN: 1570-6672. DOI: 10.1016/S1570-6672(08)60050-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570667208600501 (visited on 09/06/2021).

[35] H. Rakha, M. Van Aerde, K. Ahn, and A. Trani, "Requirements for evaluating traffic signal control impacts on energy and emissions based on instantaneous speed and acceleration measurements," en, *Transportation Research Record*, vol. 1738, no. 1, pp. 56–67, Jan. 2000, Publisher: SAGE Publications Inc, ISSN: 0361-1981. DOI: 10.3141/1738-07. [Online]. Available: https://doi.org/10.3141/1738-07 (visited on 09/06/2021).

[36] X. Liang, X. Du, G. Wang, and Z. Han, "Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019, arXiv:1803.11115 [cs, stat], ISSN: 0018-9545, 1939-9359. DOI: 10.1109/TVT.2018.2890726. [Online]. Available: http://arxiv.org/abs/1803.11115 (visited on 02/15/2023).

[37] L. Prashanth and S. Bhatnagar, "Reinforcement learning with average cost for adaptive control of traffic lights at intersections," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2011, pp. 1640–1645.

[38] W. Genders and S. Razavi, "Evaluating reinforcement learning state representations for adaptive traffic signal control," *Procedia computer science*, vol. 130, pp. 26–33, 2018.

[39] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," *arXiv preprint arXiv:1611.01142*, 2016.

[40] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv, Tech. Rep. arXiv:1412.6980, Jan. 2017, arXiv:1412.6980 [cs] type: article. DOI: 10.48550/arXiv.1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 08/29/2022).

[41] M. G. Kendall, "Rank correlation methods.," 1948. [Online]. Available: https://archive.org/details/rankcorrelationm0000kend.

[42] H. Akoglu, "User's guide to correlation coefficients," *Turkish Journal of Emergency Medicine*, vol. 18, no. 3, pp. 91–93, 2018, ISSN: 2452-2473. DOI: https://doi.org/10.1016/j.tjem.2018.08.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452247318302164.

[43] D. T. Campbell and T. D. Cook, "Quasi-experimentation," *Chicago, IL: Rand Mc-Nally*, 1979.

[44] W. R. Shadish, T. D. Cook, and D. T. Campbell, *Experimental and quasi-experimental designs for generalized causal inference.* Houghton, Mifflin and Company, 2002.

[45] R. J. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014.

[46] H. A. Simon, “Bounded rationality,” *Utility and probability*, pp. 15–18, 1990.

[47] L. N. Alegre, *SUMO-RL*, `https://github.com/LucasAlegre/sumo-rl`, 2019.