sensors

MDPI

Article

Weakly Supervised Reinforcement Learning for Autonomous Highway Driving via Virtual Safety Cages

Sampo Kuutti [1,*], Richard Bowden [2] and Saber Fallah [1]

[1] Connected and Autonomous Vehicles Lab, University of Surrey, Guildford GU2 7XH, UK; s.fallah@surrey.ac.uk

[2] Centre for Vision Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK; r.bowden@surrey.ac.uk

* Correspondence: s.j.kuutti@surrey.ac.uk

Abstract: The use of neural networks and reinforcement learning has become increasingly popular in autonomous vehicle control. However, the opaqueness of the resulting control policies presents a significant barrier to deploying neural network-based control in autonomous vehicles. In this paper, we present a reinforcement learning based approach to autonomous vehicle longitudinal control, where the rule-based safety cages provide enhanced safety for the vehicle as well as weak supervision to the reinforcement learning agent. By guiding the agent to meaningful states and actions, this weak supervision improves the convergence during training and enhances the safety of the final trained policy. This rule-based supervisory controller has the further advantage of being fully interpretable, thereby enabling traditional validation and verification approaches to ensure the safety of the vehicle. We compare models with and without safety cages, as well as models with optimal and constrained model parameters, and show that the weak supervision consistently improves the safety of exploration, speed of convergence, and model performance. Additionally, we show that when the model parameters are constrained or sub-optimal, the safety cages can enable a model to learn a safe driving policy even when the model could not be trained to drive through reinforcement learning alone.

Keywords: machine learning; neural networks; reinforcement learning; advanced driver assistance; autonomous vehicles; vehicle control; safety

check for updates

Citation: Kuutti, S.; Bowden, R.; Fallah, S. Weakly Supervised Reinforcement Learning for Autonomous Highway Driving via Virtual Safety Cages. *Sensors* **2021**, *21*, 2032. https://doi.org/10.3390/s21062032

Academic Editor: Enrico Meli

Received: 19 February 2021
Accepted: 11 March 2021
Published: 13 March 2021

1. Introduction

Autonomous driving has gained significant attention within the automotive research community in recent years [1–3]. The potential benefits in improved fuel efficiency, passenger safety, traffic flow, and ride sharing mean self-driving cars could have a significant impact on issues such as climate change, road safety, and passenger productivity [4–6]. Deep learning techniques have been demonstrated to be powerful tools for autonomous vehicle control, due to their capability to learn complex behaviours from data and generalise these learned rules to completely new scenarios [7]. These techniques can be divided in two categories based on the modularity of the system. On one hand, modular systems divide the driving task into multiple sub-tasks such as perception, planning, and control and deploy a number of sub-systems and algorithms to solve each of these tasks. On the other hand, end-to-end systems aim to learn the driving task directly from sensory measurements (e.g., camera, radar) by predicting low-level control actions.

End-to-end approaches have recently risen in popularity, due to the ease of implementation and better leverage of the function approximation of Deep Neural Networks (DNNs). However, the high level of opaqueness in DNNs is one of the main limiting factors in the use of neural network-based control techniques in safety-critical applications, such as autonomous vehicles [8–11]. As the DNNs used to control the autonomous vehicles become deeper and more complex, their learned control policies and any potential safety issues within them become increasingly difficult to evaluate. This is made further challenging by the complex environment in which autonomous vehicles have to operate in, as it is impossible to evaluate the safety of these systems in all possible scenarios they may encounter once deployed [12–14]. One class of solutions to introduce safety in machine learning enabled autonomous vehicles is to utilise a modular approach to autonomous driving, where the machine learning systems are used mainly in the decision making layer, whilst low-level control is handled by more interpretable rule-based systems [15–17]. Alternatively, safety can be guaranteed through redundancy for end-to-end approaches, where machine learning can be used for vehicle motion control with an additional rule-based virtual safety cage acting as a supervisory controller [18,19]. The purpose of the rule-based virtual safety cage is to check the safety of the control actions of the machine learning system, and to intervene in the control of the vehicle if the safety rules imposed by the safety cages are breached. Therefore, during normal operation, the more intelligent machine learning-based controller is in control of the vehicle. However, if the safety of the vehicle is compromised the safety cages can step in and attempt to bring the vehicle back to a safe state through a more conservative rule-based control policy.

This work extends our previously developed Reinforcement Learning (RL) based vehicle following model [20] and virtual safety cages [21]. We make important extensions to our previous works, by integrating our safety cages into the RL algorithm. The safety cages not only act as a safety function enhancing the vehicle's safety, but are also used to provide weak supervision during training, by limiting the amount of unnecessary exploration and providing penalties to the agent when breached. In this way, the vehicle can be safe during training by avoiding collisions when the RL agent takes unsafe actions. More importantly, the efficiency of the training process is improved, as the agent converges to an optimal control policy with less samples. We also compare our proposed framework on less safe agents with smaller neural networks, and show significant improvement in the final learned policies when used to train these shallow models. Our contributions can be summarised as follows:

- We combine the safety cages with reinforcement learning by intervening on unsafe control actions, as well as providing an additional learning signal for the agent to enable safe and efficient exploration.
- We compare the effect of the safety cages during training for both models with optimised hyperparameters, as well as less optimised models which may require additional safety considerations.
- We test all trained agents without safety cages enabled, in both naturalistic and adversarial driving scenarios, showing that even if the safety cages are only used during training, the models exhibit safer driving behaviour.
- We demonstrate that by using the weak supervision from the safety cages during training, the shallow model which otherwise could not learn to drive can be enabled to learn to drive without collisions.

The remainder of this paper is structured as follows. Section 2 discusses related work and explains the novelty of our approach. Section 3 provides the necessary theoretical background for the reader and describes the methodology used for the safety cages and reinforcement learning technique. The results from the simulated experiments are presented and discussed in Section 4. Finally, the concluding remarks are presented in Section 5.

2. Related Work

2.1. Autonomous Driving

A brief overview of relevant works in this field is given in this Section. For a more in-depth view of deep learning based autonomous driving techniques, we refer the interested readers to the review in [7]. One of the earliest works in neural control for autonomous driving was Pomerleau's Autonomous Land Vehicle In a Neural Network (ALVINN) [22], which learned to steer an autonomous vehicle by observing images from a front facing camera, using the recorded steering commands of a human driver as training data. Among the first to adapt techniques such as ALVINN to use deep neural networks, was NVIDIA's PilotNet [23]. PilotNet was trained for lane keeping using supervised learning with a total of 72 h of recorded human driving as training data.

Since then, these works have inspired a number of deep learning techniques, with imitation learning often being the preferred learning technique. For instance, Zhang et al. [24] and Pan et al. [25] extended the popular Dataset Aggregation (DAgger) [26] imitation learning algorithm to the autonomous driving domain, demonstrating that autonomous vehicle control can be learned from vision. While imitation learning based approaches have shown important progress in autonomous driving [27–30], they present limitations when deployed in environments beyond the training distribution [31]. These driving models relying on supervised techniques are often evaluated on performance metrics on pre-collected validation datasets [32], however low prediction error on offline testing is not necessarily correlated with driving quality [33]. Even when demonstrating desirable performance during closed-loop testing in naturalistic driving scenarios, imitation learning models often degrade in performance due to distributional shift [26], unpredictable road users [34], or causal confusion [35] when exposed to a variety of driving scenarios.

However, RL-based techniques have shown promising results for autonomous vehicle applications [36–38]. These RL approaches are advantageous for autonomous vehicle motion control, as they can learn general driving rules, which can also adapt to new environments. Indeed, many recent works have utilised RL for longitudinal control in autonomous vehicles with great success [39–43]. This is largely due to the fact that longitudinal control can be learned from low-dimensional observations (e.g., relative distances, velocities), which partially overcomes the sample-efficiency problem inherent in RL. Moreover, the reward function for RL is easier to define in the longitudinal control case (e.g., based on safety distances to vehicles in front). For these reasons, we focus on longitudinal control and extend on our previous work on RL-based longitudinal control in a highway driving environment [20].

2.2. Safety Cages

Virtual safety cages have been used in several cyber-physical systems to provide safety guarantees when the controller is not interpretable. The most straightforward application of such safety cages is to limit the possible actions of the controller to ensure the system is bounded to a safe operational envelope. If the controller issues commands that breach the safety cages, the safety cages step in and attempt to recover the system back to a safe state. This type of approach has been used to guarantee the safety of complex controllers in different domains such as robotics [44–47], aerospace [48], and automotive applications [49–51]. Heckemann et al. [18] suggested that these safety cages could be used to ensure the safety of black box systems in autonomous vehicles by utilising the vehicle's sensors to monitor the state of the environment, and then limiting the actions of the vehicle in safety-critical scenarios. Demonstrating the effectiveness of this approach, Adler et al. [49] proposed five safety cages based on the Automotive Safety Integrity Levels (ASIL) defined by ISO26262 [52] to improve the safety of an autonomous vehicle with machine learning based controllers. Focusing on path planning in urban environments, Yurtsever et al. [53] combined RL with rule-based path planning to provide safety guarantees in autonomous driving. Similar approaches have also been used for highway driving, by combining rule-based systems with machine learning based controllers for enhanced driving safety [54,55].

In our previous work [21], we developed two safety cages for highway driving, and demonstrated these safety cages can be used to prevent collisions when the neural network controllers make unpredictable decisions. Furthermore, we demonstrated that the interventions by the safety cages can be used to re-train the neural networks in a supervised learning approach, enabling the system to learn from its own mistakes and further making the controller more robust. However, the main limitation of the safety cage approach was that the re-training happened in an offline manner, where the learning was broken down into three stages: (i) supervised training, (ii) closed-loop evaluation with safety cages, and (iii) re-training using the safety cage interventions as labels for supervised learning.

Here, we extend on this approach by utilising the safety cages to improve the safety of a RL based vehicle motion controller, and using the interventions of the safety cages as weak supervision which enables the system to learn to drive more safely in an online manner. Weak supervision has been shown to improve the efficiency of exploration in RL [56] by guiding the agent towards useful directions during exploration. Here, the weak supervision enhances the exploration process in two ways; the safety cages stop the vehicle from taking unsafe actions thereby eliminating the unsafe part of the action space from the exploration, while also maintaining the vehicle in a safe state and thereby reducing the amount of states that need to be explored. Reinforcement learning algorithms often struggle to learn efficiently at the beginning of training, since initially the agent is taking largely random actions, and it can take a significant amount of training before the agent starts to take the correct actions which are needed to learn its task. Therefore, by utilising weak supervision to guide the agent to the correct actions and states, the efficiency of the early training stage can be improved. We show that eliminating the unsafe parts of the exploration space improves convergence during training, which can be a significant advantage considering the low sample efficiency of RL. Furthermore, we show that the safety cages eliminate the collisions that would normally happen during training, which could be a further advantage of our technique, should the training occur in a real-world system where collisions are undesirable.

3. Materials and Methods

3.1. Reinforcement Learning

Reinforcement learning can be formally described as a Markov Decision Process (MDP). The MDP is denoted by a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the the action space, $\mathcal{P}$ denotes the state transition probability model, and $\mathcal{R}$ is the reward function [57]. As shown in Figure 1, at each time step t, the agent takes an action a_t from the possible set of actions $\mathcal{A}$, according to its policy π which is a mapping from states s_t to actions a_t. Based on the action taken in the current state, the environment then transitions to the next state s_{t+1} according to the transition dynamics $p(s_{t+1}|s_t, a_t)$ as given by the transition probability model $\mathcal{P}$. The agent then observes the new state s_{t+1} and receives a scalar reward r_t according to the reward function $\mathcal{R}$. The aim of the agent in the RL setting is to maximise the total accumulated returns R_t:

$$R_t = \sum_{i=t}^{\infty} \gamma^{(i-t)} r(s_i, a_i) \tag{1}$$

where $\gamma \in [0,1]$ is the discount factor used to prioritise immediate rewards over future rewards.

3.2. Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG) [58] extends the Deterministic Policy Gradient algorithm by Silver et al. [59] by utilising DNNs for function approximation. It is an actor-critic based off-policy RL algorithm, which can scale to high-dimensional and continuous state and action spaces. The DDPG uses the state-action value function, or Q-function, $Q(s, a)$, which estimates the expected returns after taking an action a_t in state s_t under policy π. Therefore, given a state visitation distribution ρ^π under policy π in environment E the Q-function is denoted by:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_{i \geq t}, s_{i>t} \sim E, a_{i>t} \sim \pi}[R_t | s_t, a_t] \tag{2}$$

The Q-function can be estimated by the Bellman equation for deterministic policies as:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r(s_t, a_t) + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1}))] \tag{3}$$

As the expectations depend only on the environment, the critic network can be trained off-policy, using transitions from a different stochastic policy with the state visitation distribution ρ^β. The parameters of the critic network θ^Q can then be updated by minimising the critic loss $\mathcal{L}_Q$:

$$\mathcal{L}_Q = \mathbb{E}_{s_t \sim \rho^\beta, r_t \sim E}[(Q(s_t, a_t|\theta^Q) - y_t)^2] \tag{4}$$

where

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \pi(s_{t+1})|\theta^Q) \tag{5}$$

The actor network parameters θ^π are then updated using the policy gradient [59] from the expected returns from a start distribution J with respect to the actor parameters θ^π:

$$\begin{aligned} \nabla_{\theta^\pi} J &\approx \mathbb{E}_{s_t \sim \rho^\beta}[\nabla_{\theta^\pi} Q(s, a|\theta^Q)|_{s=s_t, a=\pi(s_t|\theta^\pi)}] \\ &= \mathbb{E}_{s_t \sim \rho^\beta}[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\pi(s_t)} \nabla_{\theta^\pi} \pi(s|\theta^\pi)|_{s=s_t}] \end{aligned} \tag{6}$$

For updating the networks, mini-batches are drawn from a replay memory $\mathcal{D}$, which is a finite sized buffer storing state transitions $e = [s_t, a_t, r_t, s_{t+1}]$. To avoid divergence and improve stability of training, DDPG utilises target networks [60], which copy the parameters of the actor and critic networks. These target networks, target actor $\pi'(s|\theta^{\pi'})$ and target critic network $Q'(s, a|\theta^{Q'})$, are updated slowly based on the learned network parameters to improve stability:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta' \tag{7}$$

where $\tau \ll 1$ is the mixing factor, a hyperparameter controlling the speed of target network updates.

To encourage the agent to explore the possible actions for continuous action spaces, noise is added to the actions of the deterministic policy $\pi(s_t|\theta^\pi)$. This exploration policy $\pi^e(s_t)$, samples noise from a noise process $\mathcal{N}$ which is added to the actor policy:

$$\pi^e(s_t) = \pi(s_t|\theta_t^\pi) + \mathcal{N} \tag{8}$$

Here, the chosen noise process $\mathcal{N}$ is the Ornstein-Uhlenbeck process [61], which generates temporally correlated noise for efficient exploration in physical control problems.

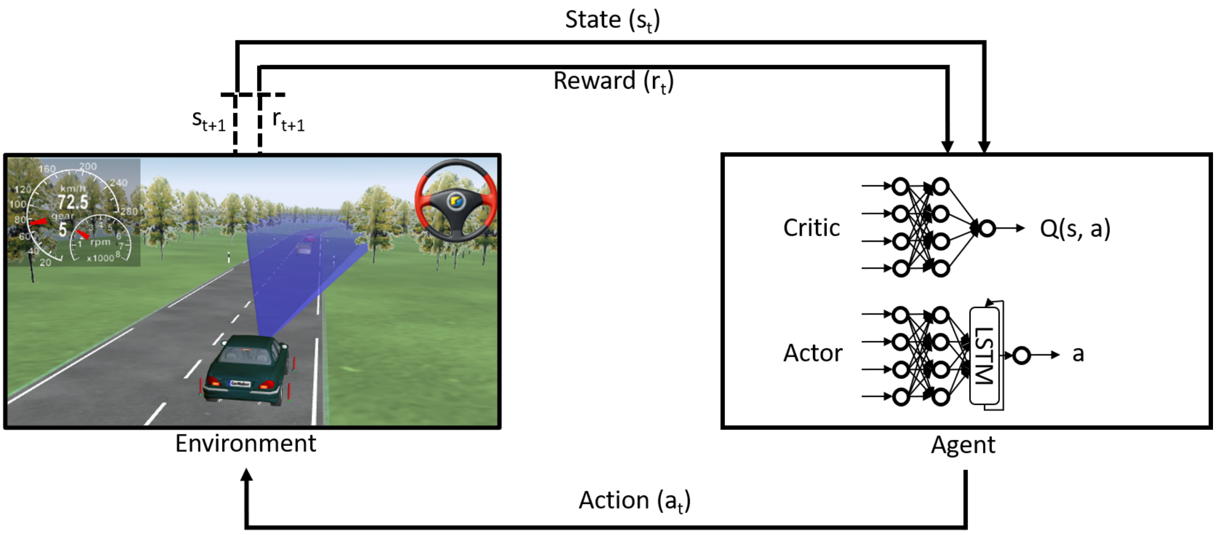

Figure 1. Reinforcement learning process.

3.3. Safety Cages

Virtual safety cages provide interpretable rule-based safety for complex cyber-physical systems. The purpose of these safety cages is to limit the actions of the system to a safe operational envelope. The simple way to achieve this, would be to limit the upper or lower limits of the system's action space. However, by using run-time monitoring to observe the state of the environment, the safety cages can dynamically select the control limits based on the current states. Therefore, the system can be limited in its possible courses of action when faced with a safety-critical scenario, such as a near-accident situation on a highway. We utilise our previously presented safety cages [21], which limit the longitudinal control actions of a vehicle based on the Time Headway (TH) and Time-To-Collision (TTC) relative to the vehicle in front. The TH and TTC metrics represent the risk of potential forward collision with the vehicle in front, and are calculated as:

$$TH = \frac{x_{rel}}{v} \tag{9}$$

$$TTC = \frac{x_{rel}}{v_{rel}} \tag{10}$$

where x_{rel} is the distance between the two vehicles in m, v is the velocity of the host vehicle in m/s, and v_{rel} is the relative velocity between the two vehicles in m/s.

The TTC and TH metrics were chosen as the states monitored by the safety cages as they represent the risk of potential collision with the vehicle in front, thereby providing effective safety measurements for our vehicle following use-case. We utilise two metrics as the TTC and TH provide complimentary information; the TTC measures time to a forward collision assuming both vehicles continue at their current speeds, whilst TH measures distance to the vehicle in front in time and makes no assumptions about the lead vehicle's actions. For example, when the host vehicle is driving significantly faster than the vehicle in front, as the distance between the vehicles gets closer the TTC approaches zero and correctly captures the risk of a forward collision. However, in a scenario where both vehicles are driving close to each other but at the same speed, the TTC will not signal a high risk of collision even though in this scenario if the lead vehicle begins to break, the two vehicles would be in a likely collision. In such a scenario, the two vehicles will have a low headway, therefore monitoring the TH will correctly inform the safety monitors of a collision risk.

The risk levels for both safety cages are as defined in [21], where the aim was to identify potential collisions in time to prevent them, whilst minimising unnecessary interventions on the control of the vehicle. The different risk levels and associated minimum braking values are illustrated in Figure 2. For each safety cage, there are three risk levels for which the safety cages will enforce a minimum braking value on the vehicle, with higher risk levels using increased rate of braking relative to the associated safety metric. When the vehicle is in the low risk region, no minimum braking is necessary and the RL agent is in full control of the vehicle. The minimum braking values enforced by the safety cages can be formally defined as shown in (11)–(12). The braking value is normalised to the range [0, 1] where 0 is no braking and 1 is maximum braking value. In this framework, both safety cages provide a recommended braking value, which is then compared to the current braking action from the RL agent. The final braking value used for the vehicle motion control, b, is then chosen as the largest braking value between the two safety cages and the RL agent as given by (13).

$$b_{TH} = \begin{cases} 0.0, & \text{for TH} > 1.6 \\ -0.5TH + 1.0, & \text{for } 1.0 < \text{TH} \leq 1.6 \\ -1.0TH + 1.5, & \text{for } 0.5 < \text{TH} \leq 1.0 \\ 1.0, & \text{for TH} \leq 0.5 \end{cases} \tag{11}$$

$$b_{TTC} = \begin{cases} 0.0, & \text{for TTC} > 2.5 \\ -0.5TTC + 1.25, & \text{for } 1.5 < \text{TTC} \leq 2.5 \\ -1.0TTC + 2.0, & \text{for } 1.0 < \text{TTC} \leq 1.5 \\ 1.0, & \text{for TTC} \leq 1.0 \end{cases} \quad (12)$$

$$b = max(b_{TH}, b_{TTC}, b_{RL}) \quad (13)$$

where b_{TH} is the minimum braking value from the TH safety cage, b_{TTC} is the minimum braking value from the TTC safety cage, and b_{RL} is the current braking value from the RL agent.

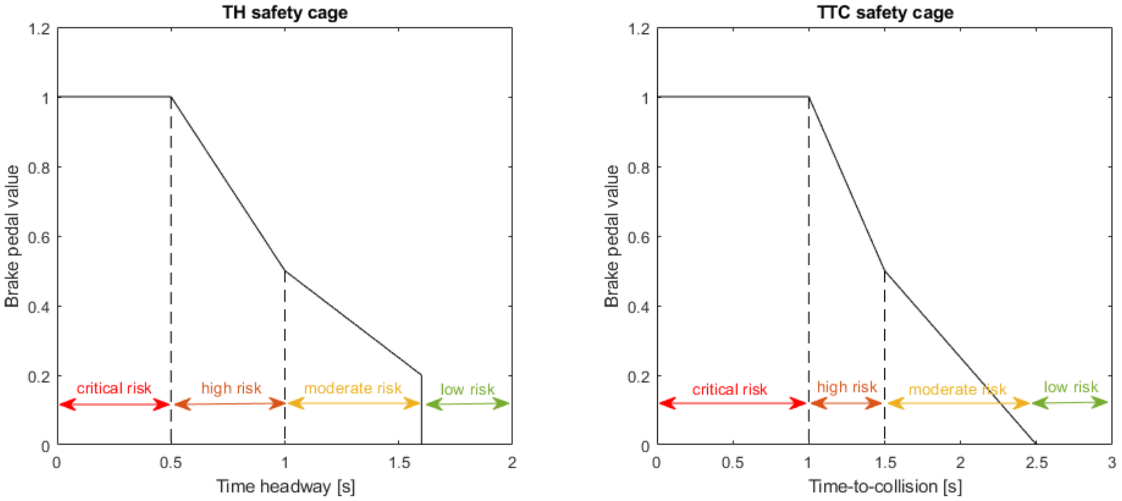

Figure 2. Risk-levels and minimum braking values for each safety cage.

3.4. Highway Vehicle Following Use-Case

The vehicle following use-case was framed as a scenario on a straight highway, with two vehicles travelling in a single lane. The host vehicle controlled by RL is the follower vehicle, and its aim is to maintain a 2 s headway from the lead vehicle. The lead vehicle velocities were limited to $v_{lead} \in [17, 40]$ m/s, and coefficient of friction values were chosen from the range [0.4, 1.0] for each episode. During training, the lead vehicle's acceleration was limited to $\dot{v}_{lead} \in [-2, 2]$ m/s^2, except for emergency braking manoeuvrers, which occurred on average once an hour, which used an acceleration in the range $\dot{v}_{lead} \in [-6, -3]$ m/s^2. The output from the RL agent is the gas and brake pedal values, which are continuous action values used to control the vehicle. As in [20], a neural network is used to estimate the longitudinal vehicle dynamics, by inferring the vehicle response to the pedal actions from the RL agent. This neural network acts as a type of World Model [62], providing an estimation of the simulator environment. This has the advantage that the neural network can be deployed on the same GPU as the RL network during training, thereby speeding up training time significantly. The World Model was trained with 2,364,041 time-steps from the IPG CarMaker simulator under different driving policies combining a total of 45 h of simulated driving. This approach was shown in [20] to speed up training by up to a factor of 20, compared to training with the IPG CarMaker simulator. However, to ensure the accuracy of all results, we also evaluate all trained policies in IPG CarMaker (Section 4.1).

3.5. Training

The DDPG model is trained in the vehicle following environment for 5000 episodes, where each episode lasts up to 5 min or until a collision occurs. The simulation is sampled at 25 Hz, therefore each time-step has a duration of 40 ms. The training parameters of the DDPG were tuned heuristically, and the final values can be found in Table 1. The critic uses a single hidden layer, followed by the output layer estimating the Q value. The actor network utilises 3 hidden feedforward layers, followed by a Long Short-Term Memory (LSTM) [63] and then the action layer. The actor network outputs the vehicle control action, for which the action space is represented by a single continuous value $a_t \in [-1, 1]$, where positive values represent the use of the gas pedal and negative values represent the use of the brake pedal. The observations of the agent are composed of 4 continuous state-values, which are the host vehicle velocity v, host vehicle acceleration $\dot{v}$, relative velocity v_{rel}, and time headway TH, such that $s_t = [v, \dot{v}, v_{rel}, TH]^T$. To enable the LSTM to learn temporal correlations, the mini-batches for training were sampled as consecutive time-steps, with the LSTM cell state reset between each training update. To encourage the agent to learn a safe vehicle following policy, a reward function based on its current headway and headway derivative was defined in [20] based on the reward function by Desjardins & Chaib-Draa [64], as shown in Figure 3. The agent gains the maximum reward when it is close to the target headway of 2 s, whilst straying further from the target headway results in smaller rewards. The headway derivative is used in the reward function to encourage the vehicle to move towards the target headway, by giving small positive rewards as it moves closer to the target and penalising the agent when it is moving further away from the target region. For further comparison, we compare training the model with an additional penalty for breaching the safety cages, such that the final reward is given as follows:

$$r_t = r_{th} + r_{sc} \tag{14}$$

where r_t is the reward for time-step t, r_{th} is the headway based reward function as shown in Figure 3, and r_{sc} is the safety cages penalty equal to -0.1 if the safety cage is breached and 0 otherwise.

Table 1. Network hyperparameters.

Parameter	Value
Mini-batch size	64
Hidden neurons in feedforward layers	50
LSTM units	16
Discount factor γ	0.99
Actor learning rate η_π	10^{-4}
Critic learning rate η_q	10^{-2}
Replay memory size	10^6
Mixing factor τ	10^{-3}
Initial exploration noise scale	1.0
Max gradient norm for clipping	0.5
Exploration noise decay	0.997
Exploration mean μ	0.0
Exploration scale factor θ	0.15
Exploration variance σ	0.2

The episode rewards during training can be seen in Figure 4, where three models are compared. The three models are DDPG only, DDPG+SC which is DDPG with safety cages, and DDPG+SC (no penalty) which is the DDPG with safety cages but without the r_{sc} penalty. As can be seen, the DDPG+SC model has lower rewards at the beginning of training as it receives additional penalties compared to the other two models. However, after the initial exploration the DDPG+SC is the first model to reach the optimal rewards per episode (∼7500 rewards), demonstrating improved convergence. Comparing the DDPG+SC models with and without penalties from the safety cages shows the model with the penalties converges to the optimal solution sooner, suggesting the penalty improves convergence during training. An additional benefit of the safety cages here is the safety of exploration, as the DDPG model collided 30 times during training, whilst the DDPG+SC model had no collisions during training. However, it can be seen that all three models converge to the same level of performance, therefore no significant difference in the trained policies can be concluded from the training rewards alone.

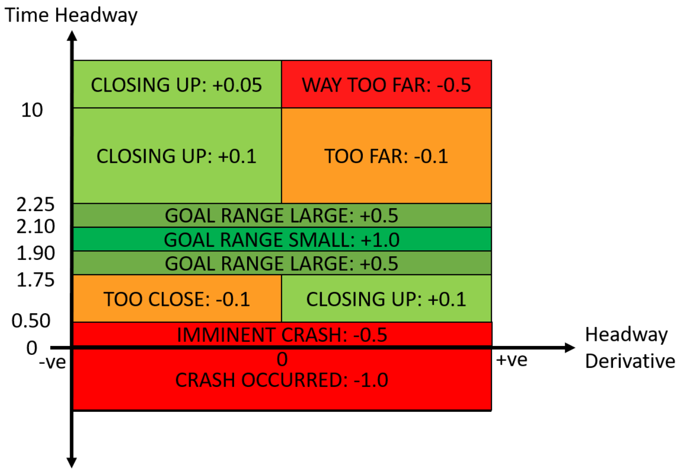

Figure 3. Time headway based reward function for vehicle following.

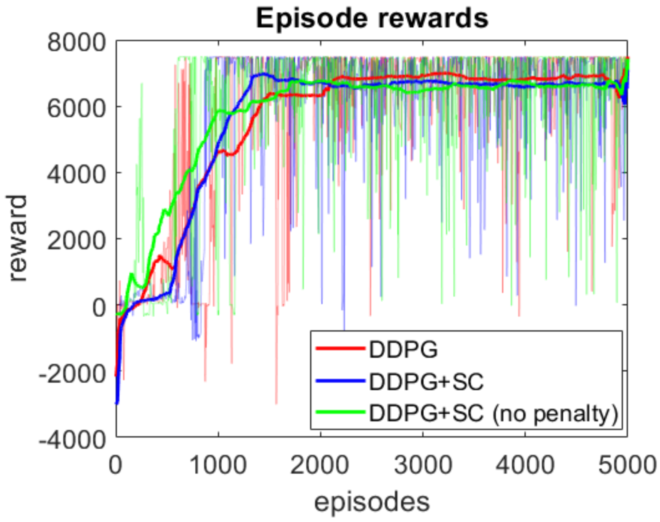

Figure 4. Episode rewards during DDPG training. Darker lines represent the moving average reward plot, whilst the actual reward values are seen in the transparent region.

As an additional investigation of the effect of the safety cages on less safe control policies, we train two further models utilising smaller neural networks with constrained parameters. These models use the same parameters as in Table 1, except they only have 1 single hidden layer with 50 neurons and no LSTM layer. We refer to these models as Shallow DDPG and Shallow DDPG+SC. It should be noted that the parameters of these models were not tuned for better performance, and indeed sub-optimal parameters were chosen on purpose to enable better insight into the effect of the safety cages in unsafe systems. The episode rewards for the two shallow models during training are shown in Figure 5. As can be seen, these two models have a more significant difference in training performance. The Shallow DDPG struggles to learn a feasible training policy, whilst the Shallow DDPG+SC learns to drive without collisions, although at a lower level of overall performance compared to the deeper models.

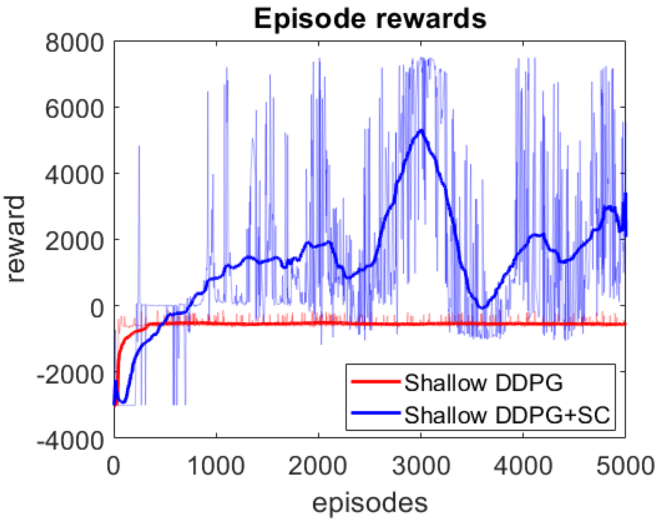

Figure 5. Episode rewards during DDPG training of shallow models. Darker lines represent the moving average reward plot, whilst the actual reward values are seen in the transparent region.

4. Results

To investigate the performance of the learned control policies, we evaluate the vehicle follower models in various highway driving scenarios. We utilise two types of testing for this evaluation. Naturalistic testing tests the control policies in typical driving scenarios, giving an idea of how the control policies perform in everyday driving. Adversarial testing utilises an adversarial agent to create safety-critical scenarios, showing how the vehicle performs in dangerous edge cases where collisions are likely to occur. The controller performance in both types of scenario is important, since most driving scenarios on the road fall into naturalistic driving the controller must be able to drive efficiently and safely in these scenarios, however the controller must also be able remain safe in dangerous edge cases in order to avoid collisions. To enable better analysis of the performance of the RL-based control policies, no safety cages are used during testing so the vehicle follower models must depend on their own learned knowledge to keep the vehicle safe. This also enables better understanding on the effect of using the safety cages during training on the final learned control policy.

4.1. Naturalistic Testing

For the naturalistic driving, similar lead vehicle behaviours were used to those during training, with velocities in the range [17, 40] m/s and acceleration [−2, 2] m/s^2. The exception to this was the harsh braking manoeuvres which occurred, on average, once an hour with deceleration [−6, −3] m/s^2. At the start of the episode, the coefficient of friction is randomly chosen in the range $[0.4, 1.0]$ and each episode lasts until 5 min has passed or a collision occurs. For each driving model, a total of 120 test scenarios were completed, totalling up to 10 h of testing. All driving for these tests occurred in the IPG CarMaker simulation environment to ensure accuracy of the results. Two types of baselines are provided for comparison; the IPG Driver is the default driver in the CarMaker Simulator and A2C is the Advantage Actor Critic [65] based vehicle follower model in [20].

The results from the naturalistic driving scenarios are summarised in Table 2. The table shows the RL based models outperform the default IPG Driver, with the exception of the shallow models. The results demonstrate that both DDPG based models outperform the previous A2C-based vehicle follower model. However, comparing the DDPG and DDPG+SC models shows the benefit of using the safety cages during RL training. While in most scenarios the two models have similar performance (the mean values seen are approximately equal), the minimum headway by the DDPG+SC during testing is higher, showing it can maintain a safer distance from the lead vehicle. However, as both models can maintain a safe distance without collisions this difference is not significant by itself.

Therefore, investigating the difference between the Shallow DDPG and Shallow DDPG+SC models provides further insight into the role the safety cages play in supervision during RL training. Similar to the training rewards, the shallow models show a more extreme difference between the two models. The results show the Shallow DDPG model without safety cages fails to learn to drive safely, whilst the Shallow DDPG+SC model avoids collisions safely, although it comes relatively close to collisions with a minimum time headway at 0.79 s. This shows the benefit of the safety cages in guiding the model towards a safe control policy during training.

Table 2. 10-hour driving test under naturalistic driving conditions in IPG CarMaker.

Parameter	IPG Driver	A2C [20]	DDPG	DDPG+SC	Shallow DDPG	Shallow DDPG+SC
min. x_{rel} [m]	10.737	7.780	15.252	13.403	0.000	5.840
mean x_{rel} [m]	75.16	58.01	58.19	58.24	41.45	59.34
max. v_{rel} [m/s]	13.90	7.89	10.74	9.33	13.43	6.97
mean v_{rel} [m/s]	0.187	0.0289	0.0281	0.0271	4.59	0.0328
min. TH [s]	1.046	1.114	1.530	1.693	0.000	0.787
mean TH [s]	2.546	2.007	2.015	2.015	1.313	2.034
collisions	0	0	0	0	120	0

4.2. *Adversarial Testing*

Utilising machine learning to expose weaknesses in safety-critical cyber-physical systems has been shown to be an effective method for finding failure cases effectively [66,67]. We utilise the Adversarial Testing Framework (ATF) presented in [34], which utilised an adversarial agent trained through RL to expose over 11,000 collision cases in machine learning based autonomous vehicle control systems. The adversarial agent is trained through A2C [65] with a reward function r_A based on the inverse headway:

$$r_A = min\left(\frac{1}{TH}, 100\right) \tag{15}$$

This reward function encourages the adversarial agent to minimise the headway and make collisions happen, while capping the reward at 100 ensures that the reward does not tend to infinity as the headway reaches zero.

As this lead vehicle used in the adversarial testing can behave very differently to those seen during training, this testing focuses on investigating the models' generalisation capability as well as their response to hazardous scenarios. Each DDPG model is tested under two different velocity ranges; the first limits the lead vehicle's velocity to the same as the training scenarios with $v_{lead} \in [17, 40]$ m/s, and the second uses a lower velocity range which enables the ATF to expose collisions more easily at a velocity range of $v_{lead} \in [12, 30]$ m/s. For each model, 3 different adversarial agents were trained, such that results can be averaged between these 3 training runs. The minimum episode TH during training can be seen for both deep models over the 2500 training episodes in Figures 6 and 7. These tests show that both deep models can maintain a safe distance from the lead vehicle even when the lead vehicle is attempting to cause collisions intentionally. Although a slight difference in the two models can be seen, as the DDPG+SC model has a slightly higher headway on average as well as significantly less variance. However as both deep models remain at a safe distance from the adversarial agent, these models can be considered safe even in safety-critical edge cases. Comparing the two shallow models in Figures 8 and 9, a more significant difference can be seen. While both models are worse in performance than the deep models, the Shallow DDPG is significantly easier to exploit than the Shallow DDPG+SC model. The Shallow DDPG model continues to cause collisions during the adversarial testing, whilst the Shallow DDPG+SC model remains at a safer distance. In the training conditions, the Shallow DDPG+SC remains relatively safe, with no decrease in the minimum headway during the training of the adversarial agent, although it can be seen that the variance increases as the training progresses. In the lower velocity case, the Shallow DDPG+SC still avoids collisions, but the adversarial agent is able to reduce the minimum headway significantly better. This shows that the safety cages have helped the model learn a significantly more robust control policy, even when the model uses sub-optimal parameters. Without the additional weak supervision from the safety cages, it can be seen that these shallow models would not have been able to learn a reasonable driving policy. Therefore, the weak-supervision by the safety cages can be used to train models with sub-optimal parameters. In addition, for models with optimal parameters they provide improved convergence during training and slightly improved safety in the final trained policy.

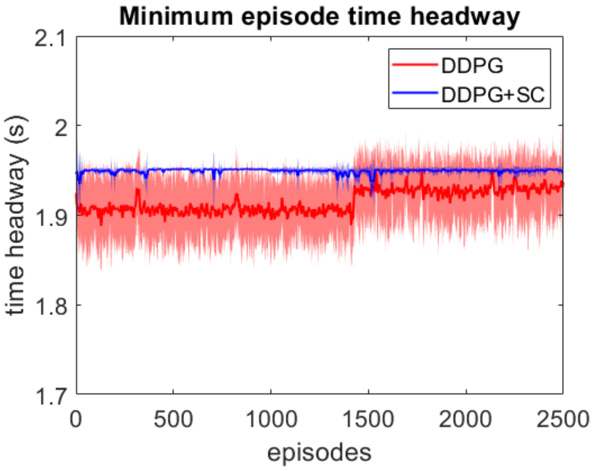

Figure 6. Comparison of the deep vehicle following agents' minimum TH per episode over adversarial training runs with lead vehicle velocity limits $v_{lead} \in [17, 40]$ m/s. Averaged over 3 runs, with standard deviation shown in shaded colour.

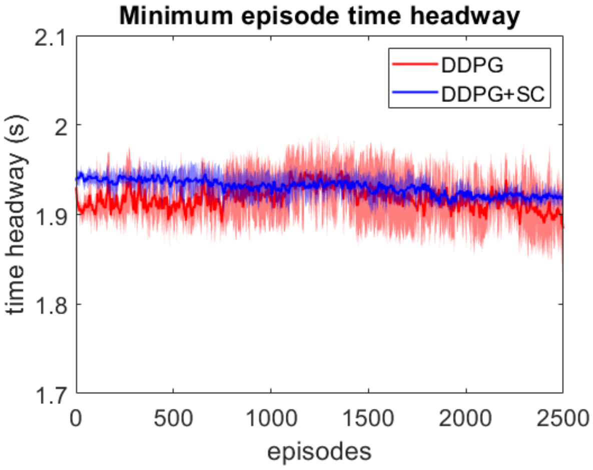

Figure 7. Comparison of the deep vehicle following agents' minimum TH per episode over adversarial training runs with lead vehicle velocity limits $v_{lead} \in [12, 30]$ m/s. Averaged over 3 runs, with standard deviation shown in shaded colour.

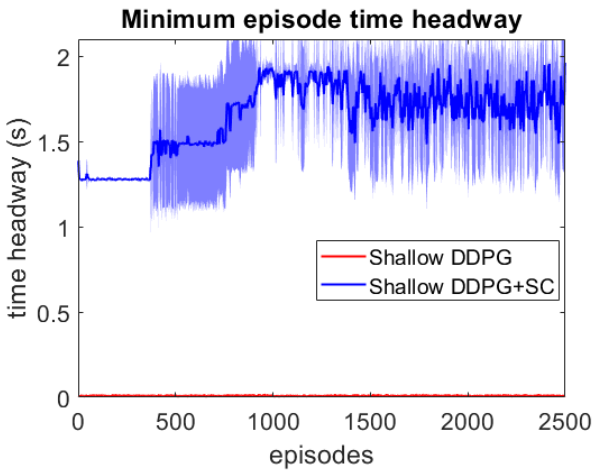

Figure 8. Comparison of the shallow vehicle following agents' minimum TH per episode over adversarial training runs with lead vehicle velocity limits $v_{lead} \in [17, 40]$ m/s. Averaged over 3 runs, with standard deviation shown in shaded colour.

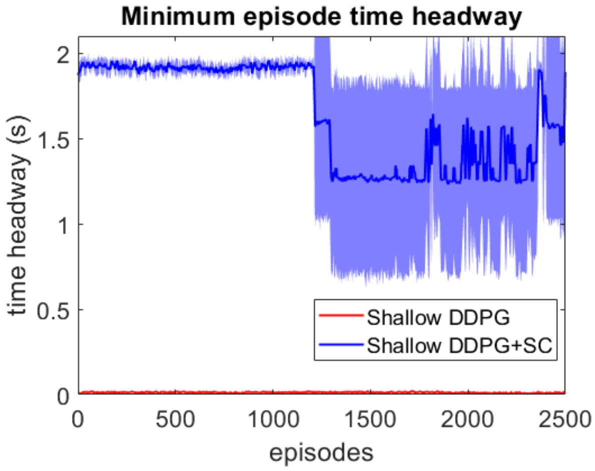

Figure 9. Comparison of the shallow vehicle following agents' minimum TH per episode over adversarial training runs with lead vehicle velocity limits $v_{lead} \in [12, 30]$ m/s. Averaged over 3 runs, with standard deviation shown in shaded colour.

5. Conclusions

In this paper, a reinforcement learning technique combining rule-based safety cages was presented. The safety cages provide a safety mechanism for the autonomous vehicle in case the neural network-based controller makes unsafe decisions, thereby enhancing the safety of the vehicle and providing interpretability in the vehicle motion control system. In addition, the safety cages are used as weak supervision during training, by guiding the agent towards useful actions and avoiding dangerous states.

We compared the model with safety cages to a model without them, and show improvements in safety of exploration, speed of convergence, and the safety of the final control policy. In addition to improved training efficiency, simulated testing scenarios demonstrated that even with the safety cages disabled, the model which used them during training has learned a safer control policy by maintaining a minimum headway of 1.69 s in a safety-critical scenario, compared to 1.53 s without safety cage training. We additionally tested the proposed approach on shallow models with constrained parameters, and showed that the shallow model with safety cage training was able to drive without collisions, whilst the shallow model without safety cage training collided in every test scenario. These results demonstrate that the safety cages enabled the shallow models to learn a safe control policy while otherwise the shallow models were not able to learn a feasible driving policy. This showed that the safety cages add beneficial supervision during training, enabling the model to learn from the environment more effectively.

Therefore, this work provides an effective way to combine reinforcement learning based control with rule-based safety mechanisms not only to improve the safety of the vehicle, but also incorporating weak supervision in the training process for improved convergence and performance.

This work opens up multiple potential avenues for future work. The use-case in this study was a simplified vehicle following scenario. However, extending the safety cages to consider both longitudinal and lateral control actions, as well as potential objects on other lanes, would allow the technique to be applied to more complex use-cases such as urban driving. Moreover, comparing the use of the weak supervision for different use-cases or learning algorithms (e.g., on-policy vs. off-policy RL) would help with understanding the most efficient use of weak supervision in reinforcement learning. Furthermore, extending the reinforcement learning agent to use more high dimensional inputs, such as images, would allow investigation into how the increased speed of convergence helps in cases where the sample inefficient reinforcement learning algorithms struggle. Finally, using the safety cages presented here in real-world training could better demonstrate the benefit in both safety and efficiency of exploration, compared to the simulated scenario presented in this work.

Author Contributions: Conceptualization, S.K., R.B. and S.F.; methodology, S.K.; software, S.K.; validation, S.K.; formal analysis, S.K.; investigation, S.K.; resources, S.K.; data curation, S.K.; writing—original draft preparation, S.K.; writing—review and editing, R.B. and S.F.; visualization, S.K.; supervision, R.B. and S.F.; project administration, R.B. and S.F.; funding acquisition, R.B. and S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the U.K.-Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R512217.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Eskandarian, A.; Wu, C.; Sun, C. Research advances and challenges of autonomous and connected ground vehicles. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 683–711. [CrossRef]
2. Montanaro, U.; Dixit, S.; Fallah, S.; Dianati, M.; Stevens, A.; Oxtoby, D.; Mouzakitis, A. Towards connected autonomous driving: Review of use-cases. *Veh. Syst. Dyn.* **2018**, *57*, 779–814. [CrossRef]
3. Kuutti, S.; Fallah, S.; Bowden, R.; Barber, P. *Deep Learning for Autonomous Vehicle Control: Algorithms, State-of-the-Art, and Future Prospects*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2019; pp. 1–80.
4. Eskandarian, A. *Handbook of Intelligent Vehicles*; Springer: London, UK, 2012; Volume 2.
5. Thrun, S. Toward robotic cars. *Commun. ACM* **2010**, *53*, 99–106. [CrossRef]
6. *Research on the Impacts of Connected and Autonomous Vehicles (CAVs) on Traffic Flow: Summary Report*; Department for Transport: London, UK, 2017.
7. Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S. A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 712–733. [CrossRef]
8. Borg, M.; Englund, C.; Wnuk, K.; Duran, B.; Levandowski, C.; Gao, S.; Tan, Y.; Kaijser, H.; Lönn, H.; Törnqvist, J. Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. *arXiv* **2018**, arXiv:1812.05389.
9. Burton, S.; Gauerhof, L.; Heinzemann, C. Making the case for safety of machine learning in highly automated driving. *International Conference on Computer Safety, Reliability, and Security*; Springer: Cham, Switzerland, 2017; pp. 5–16.
10. Varshney, K.R.; Alemzadeh, H. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *Big Data* **2017**, *5*, 246–255. [CrossRef]
11. Rajabli, N.; Flammini, F.; Nardone, R.; Vittorini, V. Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review. *IEEE Access* **2021**, *9*, 4797–4819. [CrossRef]

12. Kalra, N.; Paddock, S.M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part Policy Pract.* **2016**, *94*, 182–193. [CrossRef]
13. Wachenfeld, W.; Winner, H. The new role of road testing for the safety validation of automated vehicles. In *Automated Driving*; Springer: Cham, Switzerland, 2017; pp. 419–435.
14. Koopman, P.; Wagner, M. Challenges in autonomous vehicle testing and validation. *SAE Int. J. Transp. Saf.* **2016**, *4*, 15–24. [CrossRef]
15. Xu, Z.; Tang, C.; Tomizuka, M. Zero-shot deep reinforcement learning driving policy transfer for autonomous vehicles based on robust control. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2865–2871.
16. Wang, J.; Zhang, Q.; Zhao, D.; Chen, Y. Lane change decision-making through deep reinforcement learning with rule-based constraints. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN) Budapest, Hungary, 14–19 July 2019; pp. 1–6.
17. Arbabi, S.; Dixit, S.; Zheng, Z.; Oxtoby, D.; Mouzakitis, A.; Fallah, S. Lane-Change Initiation and Planning Approach for Highly Automated Driving on Freeways. *arXiv* **2020**, arXiv:2007.14032.
18. Heckemann, K.; Gesell, M.; Pfister, T.; Berns, K.; Schneider, K.; Trapp, M. Safe automotive software. *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 167–176.
19. Vom Dorff, S.; Böddeker, B.; Kneissl, M.; Fränzle, M. A fail-safe architecture for automated driving. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 828–833.
20. Kuutti, S.; Bowden, R.; Joshi, H.; de Temple, R.; Fallah, S. End-to-end Reinforcement Learning for Autonomous Longitudinal Control Using Advantage Actor Critic with Temporal Context. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 2456–2462.
21. Kuutti, S.; Bowden, R.; Joshi, H.; de Temple, R.; Fallah, S. Safe Deep Neural Network-driven Autonomous Vehicles Using Software Safety Cages. In Proceedings of the 2019 20th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL), Manchester, UK, 14–16 November 2019; Springer: Cham, Switzerland, 2019; pp. 150–160.
22. Pomerleau, D.A. Alvinn: An autonomous land vehicle in a neural network. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Denver, CO, USA, 27–30 November 1989; pp. 305–313.
23. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.; Zhang, J.; et al. End to end learning for self-driving cars. *arXiv* **2016**, arXiv:1604.07316.
24. Zhang, J.; Cho, K. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv* **2016**, arXiv:1605.06450.
25. Pan, Y.; Cheng, C.A.; Saigol, K.; Lee, K.; Yan, X.; Theodorou, E.; Boots, B. Agile autonomous driving using end-to-end deep imitation learning. *arXiv* **2018**, arXiv:1709.07174.
26. Ross, S.; Gordon, G.; Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the 2011 International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 627–635.
27. Codevilla, F.; Müller, M.; López, A.; Koltun, V.; Dosovitskiy, A. End-to-end driving via conditional imitation learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1–9.
28. Bansal, M.; Krizhevsky, A.; Ogale, A. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv* **2018**, arXiv:1812.03079.
29. Wang, D.; Devin, C.; Cai, Q.Z.; Yu, F.; Darrell, T. Deep object centric policies for autonomous driving. *arXiv* **2018**, arXiv:1811.05432.
30. Hecker, S.; Dai, D.; Van Gool, L. End-to-end learning of driving models with surround-view cameras and route planners. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 435–453.
31. Codevilla, F.; Santana, E.; López, A.M.; Gaidon, A. Exploring the limitations of behavior cloning for autonomous driving. In Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9329–9338.
32. Xu, H.; Gao, Y.; Yu, F.; Darrell, T. End-to-end learning of driving models from large-scale video datasets. In Proceedings of the 2017 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2174–2182.
33. Codevilla, F.; Lopez, A.M.; Koltun, V.; Dosovitskiy, A. On offline evaluation of vision-based driving models. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 236–251.
34. Kuutti, S.; Fallah, S.; Bowden, R. Training adversarial agents to exploit weaknesses in deep control policies. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 108–114.
35. De Haan, P.; Jayaraman, D.; Levine, S. Causal confusion in imitation learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019;pp. 11693–11704.
36. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.A.; Yogamani, S.; Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *arXiv* **2020**, arXiv:2002.00444.
37. Wu, J.; Wei, Z.; Li, W.; Wang, Y.; Li, Y.; Sauer, D. Battery thermal-and health-constrained energy management for hybrid electric bus based on soft actor-critic DRL algorithm. *IEEE Trans. Ind. Informatics* **2021**, *17*, 3751–3761. [CrossRef]

38. Wu, J.; Wei, Z.; Liu, K.; Quan, Z.; Li, Y. Battery-involved Energy Management for Hybrid Electric Bus Based on Expert-assistance Deep Deterministic Policy Gradient Algorithm. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12786–12796. [CrossRef]
39. Puccetti, L.; Rathgeber, C.; Hohmann, S. Actor-Critic Reinforcement Learning for Linear Longitudinal Output Control of a Road Vehicle. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 2907–2913.
40. Chae, H.; Kang, C.M.; Kim, B.; Kim, J.; Chung, C.C.; Choi, J.W. Autonomous braking system via deep reinforcement learning. In Proceedings of the 2017 IEEE Intelligent Transportation Systems Conference (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
41. Zhao, D.; Xia, Z.; Zhang, Q. Model-Free Optimal Control Based Intelligent Cruise Control with Hardware-in-the-Loop Demonstration [Research Frontier]. *IEEE Comput. Intell. Mag.* **2017**, *12*, 56–69. [CrossRef]
42. Li, G.; Görges, D. Ecological Adaptive Cruise Control for Vehicles With Step-Gear Transmission Based on Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 4895–4905. [CrossRef]
43. Huang, Z.; Xu, X.; He, H.; Tan, J.; Sun, Z. Parameterized batch reinforcement learning for longitudinal control of autonomous land vehicles. *IEEE Trans. Syst. Man, Cybern. Syst.* **2017**, *49*, 730–741. [CrossRef]
44. Kurien J.; Nayak P.P.; Williams B.C. Model Based Autonomy for Robust Mars Operations. In Proceedings of the 1st International Conference Mars Society, Boulder, CO, USA, 13–16 August 1998.
45. Crestani, D.; Godary-Dejean, K.; Lapierre, L. Enhancing fault tolerance of autonomous mobile robots. *Robot. Auton. Syst.* **2015**, *68*, 140–155. [CrossRef]
46. Haddadin, S.; Haddadin, S.; Khoury, A.; Rokahr, T.; Parusel, S.; Burgkart, R.; Bicchi, A.; Albu-Schäffer, A. On making robots understand safety: Embedding injury knowledge into control. *Int. J. Robot. Res.* **2012**, *31*, 1578–1602. [CrossRef]
47. Kuffner, J.J., Jr.; Anderson-Sprecher, P.E. Virtual Safety Cages for Robotic Devices. US Patent 9,522,471, 21 January 2016. .
48. Polycarpou, M.; Zhang, X.; Xu, R.; Yang, Y.; Kwan, C. A neural network based approach to adaptive fault tolerant flight control. In Proceedings of the 2004 IEEE International Symposium on Intelligent Control, Taipei, Taiwan, 4 September 2004; pp. 61–66.
49. Adler, R.; Feth, P.; Schneider, D. Safety engineering for autonomous vehicles. In Proceedings of the 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W), Toulouse, France, 28 June–1 July 2016; pp. 200–205.
50. Jackson, D.; DeCastro, J.; Kong, S.; Koutentakis, D.; Ping, A.L.F.; Solar-Lezama, A.; Wang, M.; Zhang, X. Certified Control for Self-Driving Cars. In Proceedings of the DARS 2019: 4th Workshop On The Design And Analysis Of Robust Systems, New York, NY, USA, 13 July 2019.
51. Pek, C.; Manzinger, S.; Koschi, M.; Althoff, M. Using online verification to prevent autonomous vehicles from causing accidents. *Nat. Mach. Intell.* **2020**, *2*, 518–528. [CrossRef]
52. *ISO 26262: Road Vehicles-Functional safety;* International Organization for Standardization: Geneva, Switzerland, 2011.
53. Yurtsever, E.; Capito, L.; Redmill, K.; Ozguner, U. Integrating Deep Reinforcement Learning with Model-based Path Planners for Automated Driving. *arXiv* **2020**, arXiv:2002.00434.
54. Likmeta, A.; Metelli, A.M.; Tirinzoni, A.; Giol, R.; Restelli, M.; Romano, D. Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving. *Robot. Auton. Syst.* **2020**, *131*, 103568. [CrossRef]
55. Baheri, A.; Nageshrao, S.; Tseng, H.E.; Kolmanovsky, I.; Girard, A.; Filev, D. Deep reinforcement learning with enhanced safety for autonomous highway driving. *arXiv* **2019**, arXiv:1910.12905.
56. Lee, L.; Eysenbach, B.; Salakhutdinov, R.; Finn, C. Weakly-Supervised Reinforcement Learning for Controllable Behavior. *arXiv* **2020**, arXiv:2004.02860.
57. Sutton, R.S.; Barto, A.G. *Introduction to Reinforcement Learning*; MIT Press: Cambridge, MA, USA, 1998; Volume 135.
58. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
59. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic Policy Gradient Algorithms. In Proceedings of the Internation Conference on Machine Learning (ICML), Beijing, China, 21–26 June 2014. .
60. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
61. Uhlenbeck, G.E.; Ornstein, L.S. On the theory of the Brownian motion. *Phys. Rev.* **1930**, *36*, 823. [CrossRef]
62. Ha, D.; Schmidhuber, J. World models. *arXiv* **2018**, arXiv:1803.10122.
63. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
64. Desjardins, C.; Chaib-Draa, B. Cooperative adaptive cruise control: A reinforcement learning approach. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1248–1260. [CrossRef]
65. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the 2016 International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
66. Corso, A.; Moss, R.J.; Koren, M.; Lee, R.; Kochenderfer, M.J. A Survey of Algorithms for Black-Box Safety Validation. *arXiv* **2020**, arXiv:2005.02979.
67. Riedmaier, S.; Ponn, T.; Ludwig, D.; Schick, B.; Diermeyer, F. Survey on Scenario-Based Safety Assessment of Automated Vehicles. *IEEE Access* **2020**, *8*, 87456–87477. [CrossRef]