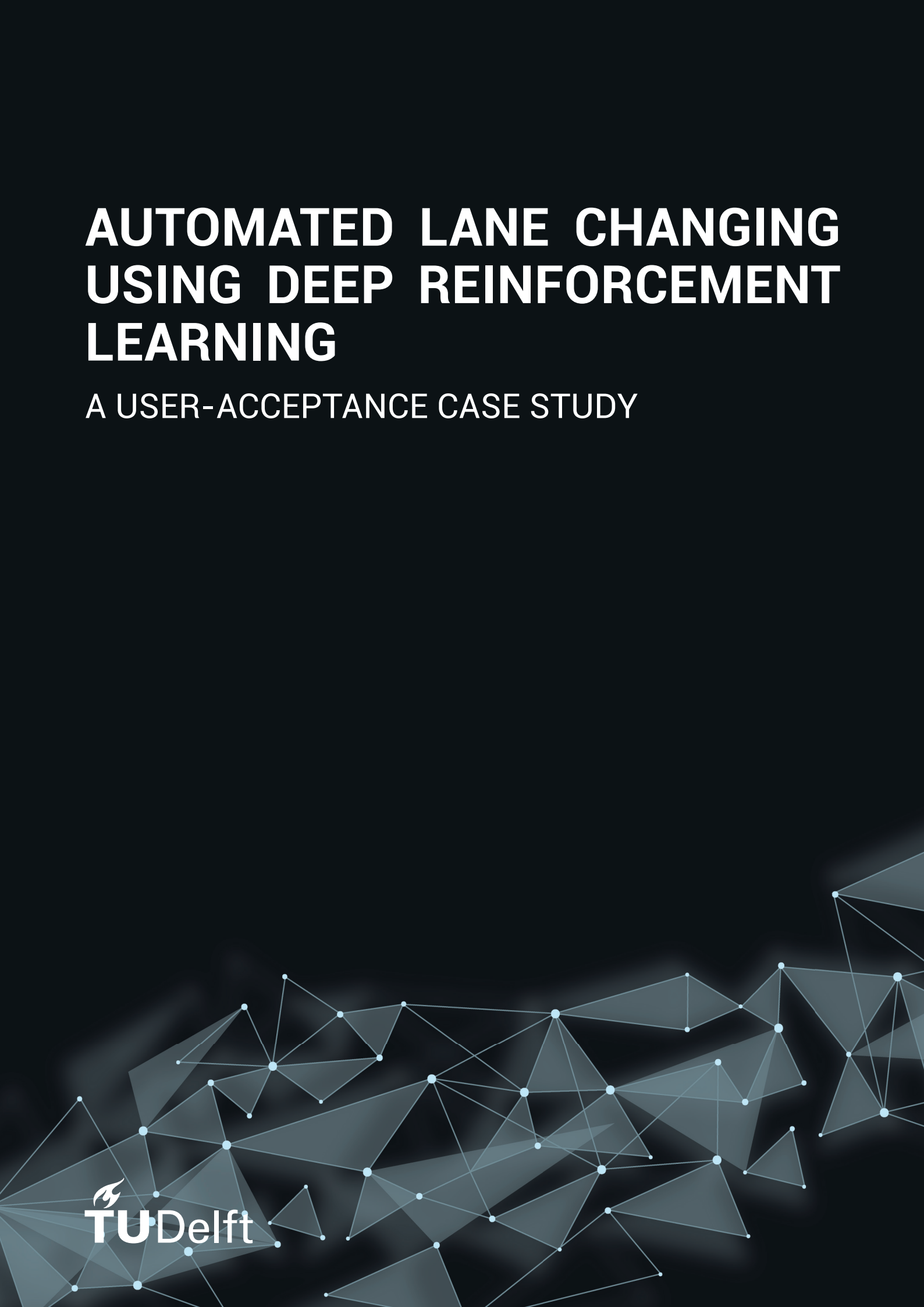
AUTOMATED LANE CHANGING USING DEEP REINFORCEMENT LEARNING

A USER-ACCEPTANCE CASE STUDY



TU Delft

AUTOMATED LANE CHANGING USING DEEP REINFORCEMENT LEARNING: A USER-ACCEPTANCE CASE STUDY

by

Daniël van den Haak

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of

Master of Science
in Mechanical Engineering

To be defended on February 23rd, 2021

Daniël van den Haak: *Automated lane changing using Deep Reinforcement Learning: a user-acceptance case study* (2021)

The work in this thesis was made in the:

Human-Robot Interaction section
Department of Cognitive Robotics (CoR)
Faculty of Mechanical, Maritime and Materials Engineering
Delft University of Technology

Student number:	4284429	
Supervisors:	Dr.ir. J.C.F. (Joost) de Winter	CoR, TU Delft
	Dr.ir. P. (Pavlo) Bazilinskyy	CoR, TU Delft
Assessment committee:	Dr. D. (Dimitra) Dodou	BmechE, TU Delft
	Dr.ir. M. (Meng) Wang	T&P, TU Delft

Abstract — Lane change decision-making is an important challenge for automated vehicles, urging the need for high performance algorithms that are able to handle complex traffic situations. Deep reinforcement learning (DRL), a machine learning method based on artificial neural networks, has recently become a popular choice for modelling the lane change decision-making process, outperforming various traditional rule-based models. So far, performance has often been expressed in terms of achieved average speed, absence of collisions or merging success rate. However, no studies have investigated how humans will react to the resulting behavior as potential occupants. This study addresses this research gap by validating a self-developed DRL-based lane changing model (trained using proximal policy optimization) from a technology acceptance perspective through an online crowdsourcing experiment. Participants ($N = 1085$) viewed a random subset of 32 out of 120 videos of an automated vehicle driving on a three-lane highway with varying traffic densities featuring our proposed model or a baseline policy (i.e. a state-of-the-art rule-based model, MOBIL). They were tasked to press a response key if the decision-making was deemed undesirable and subsequently rated the vehicle's behavior along four acceptance constructs (performance expectancy, safety, human-likeness and reliability) on a scale of 1 to 5. Results showed that the proposed model caused a significantly lower amount of disagreements and was rated significantly higher on all four acceptance constructs compared to the baseline policy. Moreover, considerable differences between individual disagreement rates were observed for both models. Our findings offer prospects for the practical application of DRL-based lane change models in a use-case scenario, depending on the user. Further research is necessary to examine whether these observations hold in other (more complex) traffic situations. Additionally, we recommend combining DRL with other modelling techniques that allow for personalization of behavioral parameters, such as imitation learning.

Keywords — automated lane changing, automated vehicles, deep reinforcement learning, artificial neural networks, proximal policy optimization (PPO), technology acceptance, crowdsourcing, MOBIL

◆

ACKNOWLEDGEMENTS

After many months of work, building a simulator environment in Unity from scratch, training a neural network model and conducting a crowdsourcing experiment that 1373 people participated in, it is a genuine pleasure to express my deep gratitude to my supervisors Joost de Winter and Pavlo Bazilinskyy for their excellent guidance, valuable feedback and enthusiasm throughout this thesis project. It was really heartwarming to see them being so actively involved in this project, always helping no matter the time of the day. This result would never have been accomplished without their expertise and the freedom they provided me with to explore all things interesting to me.

Furthermore, my gratitude goes out to all my family and friends for their patience and support throughout this unusual year working from home.

Daniël van den Haak
Delft, February 2021

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AV	automated vehicle
SAE	Society of Automotive Engineers
AI	artificial intelligence
ML	machine learning
NN	neural network
ANN	artificial neural network
RL	reinforcement learning
DRL	deep reinforcement learning
MDP	Markov decision process
TPRO	trust region policy optimization
PPO	Proximal Policy Optimization
GAE	generalized advantage estimator
TAM	Technology Acceptance Model
CTAM	Car Technology Acceptance Model
UTAUT	Unified Theory of Acceptance and Use of Technology
MAVA	multi-level model of automated vehicle acceptance
MOBIL	minimizing overall braking induced by lane change
ACC	Adaptive Cruise Control
TTC	time-to-collision

1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

With the rapid development of automated vehicle (AV) technology, the automotive transportation sector is expected to undergo a fundamental transformation in the next decades. Vehicle automation holds a significant potential in the improvement of mobility, as it is expected to make driving more comfortable and sustainable while improving traffic safety and efficiency [1, 2]. We can already see the implementation of semi-automated features in consumer cars, with some self-driving systems being able to drive under limited conditions with human supervision (SAE level 2). To fulfill higher levels of autonomy (SAE level 3+), intelligent vehicles need to learn how to make correct decisions in a safe and timely manner in order to reach human-like reliability [3].

Automated lane change decision-making on highways is one relevant area that has become one of the most thoroughly studied topics of automated driving. Improper lane changes and merges account for about 5% of crashes and 0.5% of road fatalities, making it a highly safety relevant topic [4].

A large number of studies have been conducted on automated lane changing with different approaches towards modelling the decision-making process. Traditional rule-based models, such as the lane change model by Gipps [5], the MOBIL model by Kesting et al. [6] and the potential field-based model by Ji et al. [7] follow a set of predefined rules which are applied to specific traffic situations or states. As this fundamentally requires that the correct outcome to each situation has to be explicitly defined, it means that rule-based models lack the capacity to deal with undefined situations. In a complex and interactive situation such as a traffic environment - where humans may behave unpredictably or even irrationally - more robust, efficient, and adaptive algorithms are essential.

An alternative approach is the use of machine learning (ML) methods. Deep reinforcement learning (DRL), is a branch of ML that utilizes deep neural networks to learn behavior through trial-and-error and interaction with the environment, without the need for human driving data. Recently, the use of DRL has led to breakthroughs in different fields of AI, solving complex decision-making problems such as surpassing the performance of human professional players in Atari [8], beating a world champion in Go [9], performing movement tasks in robotics [10, 11] or achieving human-like locomotion in 3D simulation [12]. Research has shown that DRL can be used effectively in the domain of automated lane changing as well, with promising results.

1.2 RELATED WORK

In general, there are two main approaches towards implementing a DRL-based lane change modelling framework: end-to-end learning or split hierarchical control.

In end-to-end learning the model is responsible for all aspects of the lane change maneuver, including environment perception, planning, decision making and executing the movement [13–15]. Hence, this increases the task complexity as the model is also responsible for the operational control of the vehicle (i.e. steering and velocity control). Even though end-to-end learning is praised for its holistic concept, it has various shortcomings [16]. Firstly, it requires a relatively large amount of training data due to the task complexity, meaning it is computationally heavy and takes a long time to train. This is especially cumbersome if the model has to be modified and the training process has to be redone. Secondly, the necessary neural network size, black box nature of the system and resulting amount of inputs-output pairs make adaptions and validation difficult. More specifically, generalizing an end-to-end model to the real world would result in limited performance if the fundamental data distribution and physics are not sufficiently captured in the simulated environment. Lastly, the task complexity may simply be too high for the neural network to capture.

An alternative approach that is widely applied is splitting the framework into several subsequent control-layers. Such a hierarchical structure consists of a high-level DRL-based decision-making module that delegates the operational execution to one or multiple lower-level control layer(s). In that case, the policy only has to learn high-level interactive lane change behavior, simplifying the problem. This makes modification considerably more accessible, and by extension, allows for integration of control systems that have already been proven to work in the real world. Lee and Choi [17] proposed a system where rule-based models were responsible for car-following, lane change trajectory generation and steering. In a more elaborate approach by Duan et al. [18] and Shi et al. [19], the low-level control layers were also based on neural networks, which were trained using separate reward functions and test episodes. A split approach also provides design freedom. For example, in a study by Jiang et al. [20], a separate recurrent neural network was used to infer the probability of surrounding vehicles being cooperative, which was then used an input in the DRL-based decision-making model.

1.2.1 Modelling variations to accelerate training

A considerable body of literature contributed by combining DRL with other modelling techniques with the intention to accelerate the training process. Li et al. [21] combined DRL with an evolutionary learning algorithm. Zhang et al. [22] guided exploration during training by deriving a surprise-based intrinsic reward relative to the expected model behavior. Liu et al. [23] applied the technique of 'behavioral cloning' by adding several episodes of human demonstration to assist the training process.

Yet one of the most popular and effective measures is the incorporation of so-called safety verification: explicitly defined restrictions that disallow the model from performing actions that are irrelevant or unsafe. One of the challenges that specifically arise in DRL is the trade-off between exploration and exploitation [24]. Especially in the beginning of the training phase, the agent is still learning the consequences of all its choices and considerable time may be wasted trying to unnecessarily explore irrelevant actions (e.g.

driving off the road or performing a lane change when a neighbouring vehicle is too close). Several works used safety verification in their methods [21–23, 25–28], where it is also called safety checking or safe exploration.

1.2.2 Evaluation of DRL-based lane change models

Various performance metrics have been used to evaluate DRL-based lane change models. Most commonly, studies reported average velocities, percentage or number of collisions, number of lane changes or reward signal plots [17–23, 26, 28–30]. Other studies reported success rates for specific traffic scenarios such as mandatory merging [27, 31] or the macroscopic effect on traffic flow [32].

Some studies directly compared DRL to rule-based methods. In an exploratory study, Alizadeh et al. [33] compared the MOBIL model against a trained DRL agent in lane change scenarios where Gaussian noise (0%, 5% and 15%) was applied on the observed input states and found DRL to have a higher average reward score with fewer collisions. In a study by Hoel et al. [34], it was found that a DRL based agent maintained a higher average speed compared to a rule-based reference model (MOBIL and IDM) in an overtaking task. Wang et al. [25] proposed a DRL-based decision-making module with rule-based trajectory constraints and found higher average speed and safety rate (proportion of episodes without collision) values compared to a rule-based policy.

1.3 RESEARCH GAP

Collectively, the cited literature so far suggests that DRL is a promising technique for modelling lane change decision-making in AVs. Above all, numerous contributions have been made regarding the improvement of the training process. Nevertheless, no studies so far have investigated whether human occupants would react positively to the resulting lane change behavior.

As a matter of fact, researchers have long regarded the concept of user acceptance as an important factor in the development process of AVs. Using theoretical models such as Technology Acceptance Model (TAM) [35] or Unified Theory of Acceptance and Use of Technology (UTAUT) [36] as a baseline, researchers have developed conceptual models in the context of AV acceptance [37–40] or conducted field- and survey-based studies [41–46] to explain people's intention to use various types of AVs. According to Osswald et al. [37], "being able to predict user acceptance would be helpful in the development process to build appropriate systems and avoid issues that affect the acceptance of a system."

Considering the fact that its fundamental working principles are based on the mathematical optimization of a numerical reward that indirectly reflects drivers' goals as opposed to other methods which directly utilize human driving data (e.g. rule-based methods, supervised learning, imitation learning, etc.), the resulting policy may potentially be to the dislike of users or other traffic participants. For this reason, we argue it is especially interesting to evaluate DRL-based lane changing models from a user acceptance perspective.

1.4 AIM OF THIS RESEARCH

The aim of this research is two-fold. The first aim is to design, develop and successfully train a lane changing model using DRL. A three-lane highway environment is built using Unity, a 3D engine, where the spatiotemporal traffic distribution is randomly generated every episode based on real traffic flow data of the A4 highway in the Netherlands. The lane change model follows a hierarchical approach, where the high-level decision-making policy is based on a neural network along with a low-level vehicle control module. A safety verification module is also added using a discrete action mask. The network policy is trained using Proximal Policy Optimization (PPO) [47].

The second aim, which is our novel contribution, is to investigate whether people agree with the decision-making and how they rate the proposed model based on four acceptance constructs from literature. Our proposed model is statistically compared to the rule-based model MOBIL [6], which is regarded as a baseline. Through an online crowdsourcing experiment, we let respondents view a random series of 32 one-minute long video demonstrations of both models from a first-person perspective in two different weather conditions. In spite of the difference between models being the main point of examination, we included the two different weather conditions (clear sunny weather and misty weather with bad visibility) to investigate the robustness of acceptance ratings in a wider range of possible weather conditions. During each video, the respondents were tasked to press a response key whenever they did not agree with the vehicle's decision-making, followed by a questionnaire in which they rated the vehicle behavior's effectiveness, safety, human-likeness and reliability on a five-point Likert scale. It is assumed that the amount of disagreements and four acceptance ratings as measured in this study provide a reliable estimation regarding the respondents' general acceptance of each model.

1.5 OVERVIEW

The remainder of this work is organized as follows. In Chapter 2, several theoretical foundations and empirical evidence with respect to the proposed model and acceptance assessment will be provided. This includes a driver behavior modelling framework, an in-depth description of MOBIL, a detailed description of (deep) reinforcement learning, the working principles of PPO and an empirical review of the four acceptance constructs used in this study. Chapter 3 showcases the process behind the design of the simulator in Unity (i.e. the environment in which the lane changing agent is trained), specifically focusing on traffic generation and highway design. Chapter 4 describes the working elements of the lane changing agent, including the overall algorithm structure, observation space, action space, reward function and separate vehicle control module (which is part of every vehicle). Chapter 5 and Chapter 6 describe the experiment method and results, respectively. Finally, the results are discussed and future research recommendations are provided in Chapter 7.

2 THEORETICAL FOUNDATION

2.1 DRIVER BEHAVIOR MODELING FRAMEWORK

Driver behavior modelling is a complex task that has primarily emerged to "predict driving maneuvers, driver intent, vehicle and driver state, and environmental factors, to improve transportation safety and the driving experience as a whole" [48]. One such model that is used as a reference framework in this study is that of Michon [49], which classifies driving behavior on time-scale levels. He argues that there are three levels of skill and control while driving a vehicle:

1. The strategical (planning) level defines the general planning of a trip including destination planning, route planning, modal choice, time management and evaluation of costs and risks. Being the top level of control, it has the highest time constant varying from minutes to hours.

2. The tactical (maneuvering) level consists of maneuvers such as obstacle avoidance, turning, overtaking or lane changing. Time constants in this level are typically in seconds.

3. The operational (control) level which consists of near instantaneous actions such as pushing the gas pedal, steering or shifting gears and are executed in sequences. This level has the shortest time constant, typically measured in milliseconds.

A comprehensive understanding of Michon's model takes into account the various levels of control, the flow of information between those levels and the influence of the outside environment. As Figure 2.1 shows, there is a top-down flow of decision making, meaning that the performed maneuvers at the tactical level serve to complete the driving task(s) set at the upper strategical level. Likewise, maneuvers at the tactical level are composed of action sequences at the operational level. There are also feedback loops: based on the outcome of certain events or decisions, the driver may decide to redefine decisions at either the same, or upper levels. Lastly, there is the surrounding environment, constraining decision-making across all levels.

What this model implies is that all levels should be reflected in the lane change model. To elaborate, lane changes should be safe and comfortable, but also improve efficiency and traffic flow. These goals are discussed in-depth later, in Section 4.4.

2.1.1 A rule-based lane changing model: MOBIL

Even though traditional rule-based methods have their limitations, they are based on the decision-making process of human drivers and therefore they provide valuable insights when it comes to designing another model. One such model that is well-documented is MOBIL [6]. Consider the situation in Figure 2.2, where v_c and a_c represent the velocity

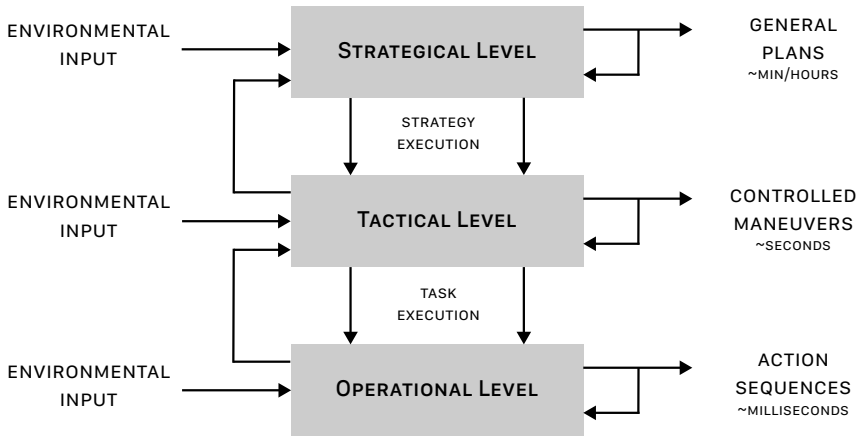


Figure 2.1: The hierarchical information flow structure according to Michon [49].

and acceleration of the ego-vehicle c that is planning a lane change to the left, respectively. The neighbouring are denoted o and n, representing the 'old' successive vehicle in the original lane and the 'new' successive vehicle in the target lane, respectively. The tilde represents any variable in case a lane change occurs. The model's working principles are based on two distinct and independent criteria (i.e. rules): a safety criterion and incentive criterion. If both criteria are satisfied, a lane change is initiated.

SAFETY CRITERION The safety criterion checks whether a lane change is feasible in terms of safety by considering the effect on the longitudinal acceleration of the successive vehicle in the target lane:

$$\tilde{a}_n \geq -b_{\text{safe}} \tag{2.1}$$

where b_{safe} is the safety limit. This ensures that the deceleration of successive vehicle in the target lane will not exceed a given threshold as a result of a lane change.

INCENTIVE CRITERION The incentive criterion determines whether a lane change results in the improvement of the ego-vehicle's traffic situation:

$$\tilde{a}_c - a_c + p\left(\tilde{a}_n - a_n + \tilde{a}_o - a_o\right) > \Delta a_{\text{th}} \tag{2.2}$$

where $p \in [0, 1]$ is the politeness factor and Δa_{th} is the switching threshold. The first part calculates the advantage from an egocentric point of view (i.e. the gain in ego-vehicle acceleration) whereas the second part determines the gain (or loss) of the two vehicles that are directly affected, weighted by a factor p. In essence, a lane change is only viable if the egocentric acceleration gain minus the weighted sum of the losses of the old (i.e. original lane) and new (i.e. target lane) successive vehicles is larger than a given threshold value, provided a lane change were to occur. Note that this rule is valid for symmetric passing rules (e.g. roads in the United States). The asymmetric variant, where

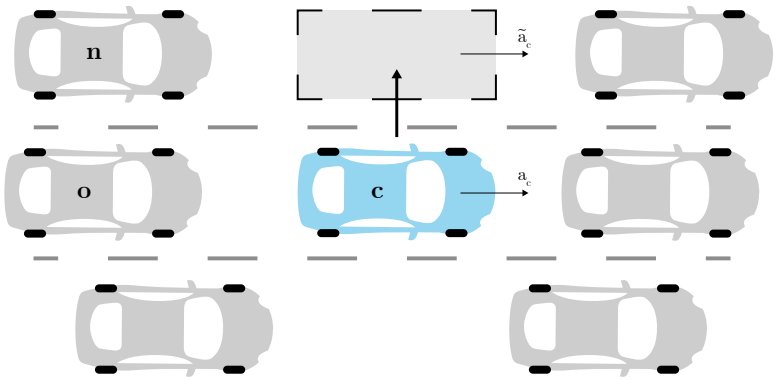


Figure 2.2: MOBIL traffic situation [6]. The ego-vehicle is denoted by c, whereas o and n denote the successive vehicles in the 'old' and 'new' lane, respectively.

a 'keep right' policy is prevalent and overtakes in the right-hand lane(s) are forbidden (e.g. roads in most of Europe), works as follows:

$$\text{left} \rightarrow \text{right}: \quad \tilde{a}_c^s - a_c + p\,(\tilde{a}_o - a_o) > \Delta a_{\text{th}} - \Delta a_{\text{bias}} \tag{2.3}$$

$$\text{right} \rightarrow \text{left}: \quad \tilde{a}_c - a_c^s + p\,(\tilde{a}_n - a_n) > \Delta a_{\text{th}} + \Delta a_{\text{bias}} \tag{2.4}$$

$$\text{where} \quad a_c^s = \begin{cases} \min(a_c, \tilde{a}_c) & \text{if } v_c > \tilde{v}_{\text{lead}} > v_{\text{crit}} \\ a_c & \text{otherwise} \end{cases} \tag{2.5}$$

where Δa_{bias} is a set parameter which should be larger than Δa_{th} [6] and a_c^s is determined using the passing rule. The passing rule influences the acceleration in the right-hand lane only if there is no uncongested traffic ($\tilde{v}_{\text{lead}} > v_{\text{crit}}$) and the ego-vehicle is faster than the leading vehicle in the left-hand lane ($v_c > \tilde{v}_{\text{lead}}$). A suitable value for v_{crit} is 60 km/h [6].

It should be noted that all values are calculated using a car-following model, whereas MOBIL is only responsible for taking lane change decisions.

2.2 REINFORCEMENT LEARNING

Reinforcement learning (RL) is a method in which software agents learn by means of finding an optimal behavioral policy that maximizes the expected value of a cumulative reward [24, 50]. In essence, the learner is not explicitly told what to do, but instead must discover policy online through interacting with the environment and find out what actions yield the highest reward. In an iterative fashion, the agent takes an action based on the current state, which affects not only the immediate reward, but also the next state and, through that, all subsequent rewards.

In any RL problem, the learner and decision-maker is referred to as the *agent*. All things the agent interacts with is the *environment*, which happens in a sequence of discrete time-steps $t = 0, 1, 2, 3...$ At each time step t, the agent goes through a cycle called an *experience*: it first perceives the environment's *state* $S_t \in \mathcal{S}$, then performs an *action* $A_t \in \mathcal{A}(S_t)$ which subsequently results in a *reward* $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$[1] and the new state

1 One can use R_t or R_{t+1} to denote the next reward due to A_t. Unfortunately, both conventions are widely used in literature. In this work, R_{t+1} is used to emphasize that the next reward and state are jointly determined from A_t.

S_{t+1}. Here $\mathcal{S}$ is the set of possible states (i.e. *state space*), $\mathcal{A}(S_t)$ the set of actions available (i.e. *action space*) in state S_t and $\mathcal{R} \subset \mathbb{R}$ the set of all possible rewards. See Figure 2.3. This cycle gives rise to a sequence called a *trajectory*:

$$\tau = S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, ... \tag{2.6}$$

At some point, the trajectory ends at a final time step T, where it is in the *terminal state*. Trajectories that start at the initial state and end in the terminal state (i.e. $t = 0, ..., T$) are called *episodes*. Naturally, most agent-environment interactions can be broken down in episodes. In these so-called *episodic tasks*, there can always be different outcomes - where T is a random variable that varies from episode to episode - but all episodes end in the terminal state and begin independently of how the previous episode ended. In the special case of $T = \infty$, the agent-environment interaction can not naturally be broken down in identifiable episodes, at which point they are called *continuing tasks*.

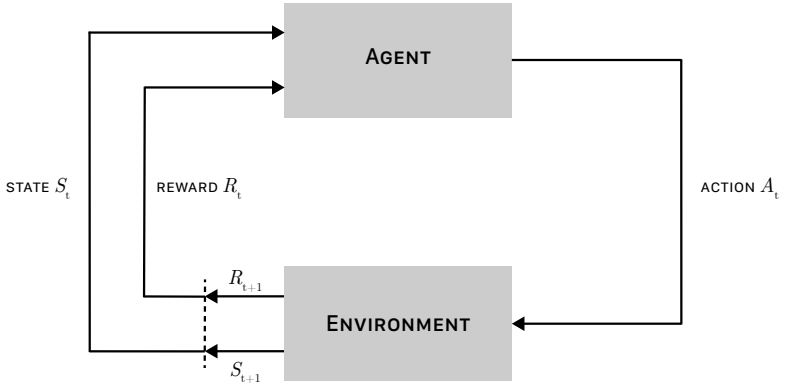


Figure 2.3: The agent-environment feedback framework in reinforcement learning [24].

2.2.1 Markov decision process

Typically, an RL problem is formally defined as a Markov decision process (MDP). If the state and action spaces are finite, which is often the case for RL problems, then it is called a *finite* Markov decision process. An MDP follows the assumption that the next state only depends on the current state and action representation, i.e. it satisfies the 'Markov property' [24, 50]. Therefore, given any state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, the probability of transitioning into a next state $s' \in \mathcal{S}$ with reward $r \in \mathcal{R} \subset \mathbb{R}$ is denoted as[2]

$$p(s', r \mid s, a) = \Pr\left\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\right\} \tag{2.7}$$

For a finite state set $\mathcal{S}$, the transition function can be described by a matrix $\mathbf{P}$. If the reward function R is described as

$$R_t = R(S_t, A_{t-1}) \tag{2.8}$$

then a finite MDP is a 4-tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{P}, R \rangle$.

2 The notations s, a and r are general representations of state, action and reward regardless of time step t.

The *policy* π maps the probabilities of the agent selecting each possible action a based on its observation of the environment state s. More specifically, $\pi(a|s)$ describes the probability distribution of selecting action $A_t = a$ for state $S_t = s$, i.e. it effectively defines the behavior of the agent at t. The distribution is defined as

$$\pi(a \mid s) = \Pr\{A_t = a \mid S_t = s\} \quad \longrightarrow \quad [0, 1] \tag{2.9}$$

2.2.2 Optimization objective

As is stated in the introduction of this section, the objective of RL is to find an optimal policy π^* that maximizes the expected cumulative reward, which is mathematically formalized in this subsection. In general, the goal is to maximize the *return* G_t:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T \tag{2.10}$$

However, this definition is problematic in the case of continuing tasks where $T = \infty$. For this, the concept of discounting is introduced. The *discounted return* is defined as the sum of future discounted rewards:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{T} \gamma^k R_{t+k+1} \tag{2.11}$$

where $\gamma \in [0, 1]$ is a set parameter, called the *discount rate*. Sometimes, a state may yield a low immediate award, but is followed by other states that subsequently yield a long-term high award. The discount rate determines the present value of future rewards: as γ approaches 0, the agent will tend to consider immediate rewards like R_{t+1}. Conversely, if the value is closer to 1, the agent will consider future rewards more strongly; it will become more farsighted. Notice that as long as $\gamma \neq 1$, G_t will always have a finite value in case $T = \infty$.

The *expected return* $J(\pi)$ is defined as

$$J(\pi) = \int_{\tau} p(\tau \mid \pi) G_t = \mathbb{E}_{\tau \sim \pi}[G_t] = \mathbb{E}_{\tau \sim \pi}\left[\sum_{k=0}^{T} \gamma^k R_{t+k+1}\right] \tag{2.12}$$

where $\mathbb{E}_{\tau \sim \pi}[\cdot]$ denotes the expected value of a variable given that the trajectories τ are sampled while following policy π. This finally leads to the mathematical definition of the RL *optimization problem* which is to be solved:

$$\pi^* = \arg\max J(\pi) \tag{2.13}$$

where π^* is the optimal policy that is aimed to be obtained by altering the policy such that the expected return is maximized. Note that the terms expected cumulative reward and expected return are often used interchangeably.

2.2.3 Value functions

The *value function* V^π determines the value of a state s under policy π, which is informally defined as the expected return when starting in state s and successfully following policy π thereafter. Formally, it is defined as

$$V^\pi(s) = \mathbb{E}_{\tau\sim\pi}\left[G_t \mid S_t = s\right] = \mathbb{E}_{\tau\sim\pi}\left[\sum_{k=0}^{T}\gamma^k R_{t+k+1} \;\middle|\; S_t = s\right] \tag{2.14}$$

The reward function can be regarded as the primary basis for altering the policy; without rewards, there can be no value and value is estimated using rewards. However, the value function is arguably as important. The agent performs actions based on the amount of value, not the immediate reward, because these actions obtain the highest reward in the long term [24].

Similarly, there is the *action-value function* Q^π, which depends on both the state and the action. It returns the expected return starting from s, taking the action a and thereafter following policy π:

$$Q^\pi(s,a) = \mathbb{E}_{\tau\sim\pi}\left[G_t \mid S_t = s, A_t = a\right] = \mathbb{E}_{\tau\sim\pi}\left[\sum_{k=0}^{T}\gamma^k R_{t+k+1} \;\middle|\; S_t = s, A_t = a\right] \tag{2.15}$$

Another way of interpreting Q^π is that is determines the 'quality' of an action while in a particular state.

2.2.4 Optimization methods

So far, it is assumed that the value functions can be estimated by computing expectations over the whole state-space and storing them in state-action pair tables. The policy can then be updated accordingly. However, this 'brute force' method is expensive and generally impractical unless the MDP is finite and small [24]. For this reason, most RL methods involve function approximation in order to optimize the policy.

In general, there are two main categories for policy optimization: 1) *value-based optimization* and, 2) *policy-based optimization* [50].

Value-based optimization: Q-learning

Value-based optimization algorithms are focused on optimizing the value functions, usually $Q(s,a)$. One of the most important and significant algorithms in this category is *Q-learning*, which is often also the term used when talking about this category [24, 51]. Through the definition in Equation 2.13, we can define the optimal action-value function $Q^*(s,a)$ as

$$\begin{aligned} Q^*(s,a) &= \max_\pi \mathbb{E}_{\tau\sim\pi}\left[G_t \mid S_t = s, A_t = a\right] \\ &= \max_\pi \mathbb{E}_{\tau\sim\pi}\left[\sum_{k=0}^{T}\gamma^k R_{t+k+1} \;\middle|\; S_t = s, A_t = a\right] \end{aligned} \tag{2.16}$$

which returns the expected return starting in state s, taking action a and thereafter following the optimal policy π^*. Put simply, the optimal action a^* at state s is the action a that returns the highest Q-value:

$$a^* = \arg\max_a Q^*(s, a) \tag{2.17}$$

In Q-learning, the Q-value function is updated every transition from a non-terminal state s:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{2.18}$$

where α is a set parameter, the *learning-rate*, determining the relative weight of the new update. It is important to note that the experience generated is 'off' (not following) the target policy, which is why these methods are often referred to as *off-policy* methods [24, 51]. Off-policy methods allow reusing past experiences during learning which is often done by storing experiences in a replay buffer $\mathcal{D}$.

Policy-based optimization: Policy gradient

Policy-based methods focus on optimizing the policy directly without approximating or learning an action-value function (e.g. Q) [52]. This is why these methods are labelled as *on-policy* algorithms. Given the definition in Equation 2.12, the general goal is to optimize π in order to maximize $J(\pi_\theta)$ for some policy with parameters θ, which is often done using gradient-based (a.k.a. *policy gradient*) methods.

Policy gradient methods attempt to estimate the gradient $\nabla_\theta J(\pi_\theta)$ using a variant of an estimator with the general form

$$\hat{g} = \hat{\mathbb{E}} \left[\nabla_\theta \log \pi_\theta(a \mid s) \Psi_t \right] \tag{2.19}$$

which is obtained by differentiating the *loss function*

$$\mathcal{L}^{PG}(\theta) = \hat{\mathbb{E}} \left[\log \pi_\theta(a \mid s) \Psi_t \right] \tag{2.20}$$

Here θ indicates the policy parameters and Ψ_t is a function dependent on the specific method. Examples vary from the cumulative reward of the trajectory, a state-action value function Q^π or an advantage function A^π. The optimization problem can be solved using gradient ascent, i.e. adjusting θ in the direction of $\nabla_\theta J(\pi_\theta)$. The policy is updated according to the following step:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\pi_\theta) \tag{2.21}$$

where α is the previously introduced learning rate, indicating the strength of the gradient ascent update. Unlike off-policy methods, on-policy do not store past experiences in a replay buffer and discards a batch of experiences once it has been used in a gradient update.

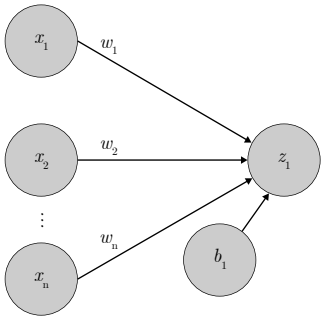


Figure 2.4: A single-output neural network.

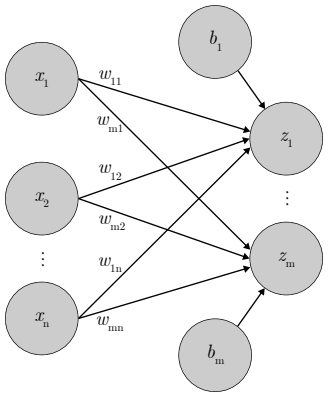


Figure 2.5: A multiple-output neural network.

2.3 DEEP REINFORCEMENT LEARNING

One way of approximating value functions or policies is through the use of deep neural networks, which is called deep reinforcement learning (DRL) and this enables optimization through gradient-based methods. Usually, the neural network parameters of a policy are denoted by θ whereas those of a value function by ϕ.

2.3.1 Artificial neural networks

An artificial neural network (ANN), as the names suggests, consists of a network of *neurons* (sometimes called nodes), as an abstract representation of a real brain. A neuron is formalized with numerical input(s) and output(s).

Working principles of a single neuron

Consider one neuron with n inputs $x_1, x_2, ..., x_n$ (see Figure 2.4). The signals being passed through the inputs are aggregated as in

$$z = w_1x_1 + w_2x_2 + ... + w_nx_n + b = \sum_{i=1}^{n} w_ix_i + b \tag{2.22}$$

where w_i represent the *weights* which determine the importance of a specific input signal and b is the *bias* which shifts the output of the neuron. Each neuron contains an activation function $f(z)$, which determines the output value: if the aggregated signal that is passed through the activation function is strong enough, then the neuron will output a high output value. Otherwise, the output will be a low value.

(Multiple) neuron layers

A fully connected layer with n inputs and m outputs is called a *dense* layer and commonly represented by a matrix multiplication:

$$\begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{2.23}$$

or

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{2.24}$$

where $\mathbf{W} \in \mathbb{R}^{m \times n}$ is a matrix representing the weights and $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ are vectors representing the output, input and bias respectively.

Multilayer perceptron

In addition to multiple inputs and outputs, one can add layers of neurons between the input and output layers called hidden layers. Models with such added hidden layers, called *multilayer perceptrons*, are able to fit more complex data compared to a single dense layer and therefore have a stronger learning capability [53]. For DRL, the input layer represents the perceived state variables whereas the output layer represents the agent's action probabilities (discrete action space), action magnitude (continuous action space) or the Q-values per action, depending on the method (see Section 2.2.4). The amount of hidden layers and their corresponding sizes are set parameters. The neural network's parameters θ or ϕ determine all layer's weights and biases and are suspect to change during training.

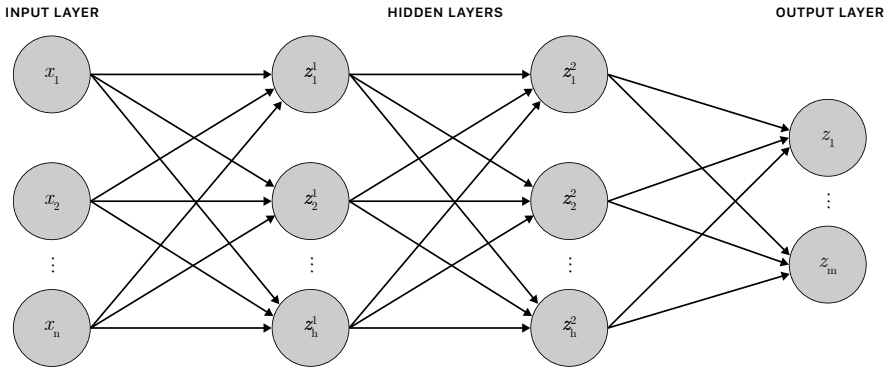


Figure 2.6: A multilayer perceptron with two hidden layers.

2.3.2 Proximal Policy Optimization (PPO)

PPO [47] is a state-of-the-art policy gradient ascent method for DRL. It was developed as an algorithm seeking to attain the data efficiency and reliability of another algorithm called trust region policy optimization (TPRO) [54], while retaining simplicity and ease of use.

PPO and TPRO are based on the following gradient estimator:

$$\hat{g} = \hat{\mathbb{E}}_t \left[\nabla_\theta \log \pi_\theta(A_t \mid S_t) \hat{A}_t \right] \tag{2.25}$$

where $\hat{A}_t$ represents an estimator of the *advantage function* at time step t. For TPRO and PPO, the advantage is estimated using the truncated generalized advantage estimator (GAE) [55]:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{2.26}$$

$$\text{where} \quad \delta_t = R_t + (\gamma\lambda)V_\phi^{\pi_\theta}(S_{t+1}) - V(S_t) \tag{2.27}$$

where $V_\phi^{\pi_\theta}(s)$ is a value-function estimator with parameters ϕ which is constantly updated during training, $\lambda \in [0,1]$ is a *smoothing* parameter and T is the *time-horizon* parameter which is much smaller than the episode length. Note that both time-horizon and the time step of a terminal state are conventionally denoted by T. The index t in this case represents the time index in $[0,T]$, within a given length-T trajectory segment. The incorporation of T ensures that GAE is a finite-horizon estimator that does not look beyond T time steps back. Similarly to γ, λ contributes to a bias-variance tradeoff and indicates how much the estimator should rely on estimated versus actual values (see Equation 2.26) [55].

In basic words, $\hat{A}_t$ calculates how much better or worse a chosen action is compared to what was expected at time step t. However, as this advantage function is often a noisy estimate, there is a high probability of destroying the policy if one keeps performing gradient ascent on a batch of experiences with a large α, while a small α on the other hand, may be too conservative and prevent learning [54].

As opposed to the $\log \pi_\theta(A_t \mid S_t)$ function, TPRO uses the following (surrogate) loss function:

$$\mathcal{L}^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(A_t \mid S_t)}{\pi_{\theta_{\text{old}}}(A_t \mid S_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[l_t(\theta)\hat{A}_t\right] \tag{2.28}$$

However, without a constraint, L^{CLI} would lead to an excessively large policy update if an action is much more probable under π_θ compared to the old policy $\pi_{\theta_{\text{old}}}$, resulting in the proposed clipped surrogate loss function in PPO [47]:

$$\mathcal{L}^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(l_t(\theta)\hat{A}_t,\ \text{clip}\left(l_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \tag{2.29}$$

where ϵ is a parameter. The first term retains the surrogate loss function from TPRO as described in Equation 2.28 whereas second term prevents potentially large policy updates (which could ruin the policy) by clipping the probability ratio $l_t(\theta)$ between the bounds $[1-\epsilon, 1+\epsilon]$. Finally, a minimum of the clipped and unclipped loss functions is taken, resulting in a final lower bound on the unclipped objective.

The PPO algorithm is summarized in Algorithm 2.1.

2.4 RELEVANT FACTORS AFFECTING USER ACCEPTANCE

This section provides some background information regarding the acceptance assessment experiment, supported by a brief analysis of existing research. As stated before, AV acceptance has been studied extensively in the past, with results ranging from the development of theoretical models specifically focusing on AVs [37–40], to several field- and survey-based studies [41–46].

The model that is used as the underlying foundation for our analysis is the multi-level model of automated vehicle acceptance (MAVA) by Nordhoff et al. [39], which in turn is based on two other proven models: UTAUT [36] and Car Technology Acceptance Model (CTAM) [37]. MAVA is empirically supported by 124 studies and specifically designed for highly automated vehicles (SAE levels 4 and 5). It presents a process-oriented view on AV acceptance, where it is modelled as a sequential decision-making

Algorithm 2.1: PPO

Hyperparameters: epsilon ϵ, minibatch size M, time-horizon T, epochs K
Input: initial policy parameters θ_0, initial value function parameters ϕ_0

for $k = 0, 1, 2, ...$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ on the environment for T time steps.

Compute discounted rewards $\hat{R}_1, ..., \hat{R}_T$.

Compute advantage estimates $\hat{A}_1, ..., \hat{A}_T$ based on $V_\phi^{\pi_\theta}$.

Update policy by maximizing the surrogate loss function $\mathcal{L}^{CLIP}(\theta)$:

$$\theta_{k+1} = \arg\max_\theta \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(l_t(\theta)\hat{A}_t,\ \text{clip}\big(l_t(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_t \right)$$

typically via K of steps minibatch M stochastic gradient descent with Adam [56].

Fit value function by regression mean-squared error:

$$\phi_{k+1} = \arg\min_\phi \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi^{\pi_\theta}(S_t) - \hat{R}_t \right)^2$$

typically via K steps of stochastic descent.

process. Given the nature of this study (i.e. the assessment of a DRL-based AV highway system), we will focus on the second stage, namely the "favourable or unfavourable attitude towards AVs on the basis of the evaluation of the instrumental domain-specific, symbolic-affective and moral-normative characteristics of AVs" [39].

As becomes apparent from reading literature, there are numerous documented (and possibly undocumented) *acceptance constructs* in this stage. As this research is focused on evaluating the behavioral functionality of a DRL lane changing model and not on AV acceptance in general, only some relevant factors are explored in this work. That is, only constructs that are directly influenced by the model's functional behavior are discussed and later assessed in the experiment, although the authors acknowledge the importance of all remaining factors.

PERFORMANCE EXPECTANCY Performance expectancy, otherwise often referred to as perceived usefulness, is defined as the degree to which a person believes that using the system will provide benefits in task performance [36, 39]. According to Venkatesh et al. [36], this construct is the strongest predictor of intention to use a product. Both Nordhoff et al. [41] and Choi and Ji [46] found a significant correlation between performance expectancy and the participants' intention to use an AV. Jing et al. [40] state that "AV acceptance is intensely predicted by perceived usefulness", as reflected in all literature that was collected in their analysis. In this work, the 'task' is assumed to be reaching a destination as fast and safely as possible, i.e. performance expectancy will be measured by how *efficient* the agent is at performing its task.

PERCEIVED SAFETY Perceived safety, like performance expectancy, is also a domain-specific characteristic of AVs and another key determinant of AV acceptance [39]. While it is assumed to be unlikely that developers will be able to establish AVs to be fully safe before their public release [57], several field studies suggest that safety is of high importance for drivers. Castritius et al. [45] evaluated an automated truck platooning system and report that safety concerns predominated in pre-test drive interviews whereas safety was highlighted as the main advantage of the system in post-test drive interviews. Both Xu et al. [43] and Cho et al. [58] report significant correlations between perceived safety and intention to use an AV. One should note that this factor is of subjective nature as opposed to objective safety.

TRUST Trust, indicated by how trust-able, reliable and dependable a system is, was not valued in early acceptance models. Yet, it is a fundamental factor affecting human-machine interaction [59, 60]. It is important to note that trust not only directly predicts acceptance, but also indirectly affects it through the other constructs perceived usefulness, perceived ease of use or perceived safety [40, 43, 46].

HUMAN-LIKENESS Human-likeness in AVs has often been linked with increased user trust and acceptance [61–64]. Research conducted by Griesche et al. [65] showed that drivers prefer an automated vehicle driving style that is similar to their own. In fact, human-likeness is a subject of a larger debate that questions how automated vehicles should behave, as AVs that are too 'perfectly' designed may not align with the expectations of (other) road users [66]. Oliveira et al. [67] investigated the effect of human-like driving behavior on user trust and acceptance but found no significant differences between a human-like and machine-like driving style. They recommend a balanced approach and argue that automated vehicles should be designed with both human- and machine-like behavioral features in order to harness the advantages of automation while still retaining a degree of user comfort due to familiarity. Since DRL utilizes no (naturalistic) driving data for optimization, it is arguably important to investigate the degree of human-likeness in the agent's final behavior.

3 ENVIRONMENT DESIGN

This chapter provides an overview of the simulator environment in which the agent operates and further highlights the process behind constructing the environment, specifically focusing on how the surrounding traffic state is measured and generated. Several metrics and notations are introduced to describe the individual state of vehicles and the average state of traffic as a whole in Section 3.1. Through the use of real measurement traffic data (see Section 3.2) in combination with pseudo-random number generators, a function to randomly generate the spatiotemporal distribution of the traffic per (training) episode is discussed in Section 3.3. Finally, Section 3.4 explains the concept of simulation episodes, the fundamental structure behind the environment simulator.

A three lane, 5 km long straight highway is designed in Unity using free assets. Each lane is 3.5 m wide and clearly marked. The road boundaries have 2 m wide shoulders for clearance in addition to either a metal guardrail or concrete wall. Other arbitrary surrounding elements such as trees, buildings or other roads are purely for aesthetic purposes and have no effect on the agent or other traffic. Figure 3.1 shows an impression of the environment.

Unity's standard coordinate system is a left-handed Cartesian coordinate system measured in meters where the x-axis corresponds to the lateral direction, the z-axis to the longitudinal direction and the y-axis to the vertical direction. To preserve consistency, this coordinate system will also be used in the remainder of this study. Lanes will be denoted by index k. Furthermore, the index α will be used to denote a vehicle, where $\alpha - 1$ denotes a leading/preceding vehicle and $\alpha + 1$ a successive/lagging vehicle.

Figure 3.1: An impression of the 3D simulator environment. The agent is shown performing a lane change towards the middle lane.

3.1 TRAFFIC FLOW MODELLING

The behavior of all surrounding traffic has a direct impact on the agent environment. This section considers all the variables with respect to the flow of traffic in the simulator that are used to build a realistic traffic situation. Both microscopic- and macroscopic parameters will be introduced in this section.

3.1.1 Microscopic parameters

Microscopic parameters describe the behavior of individual vehicles and how they interact with surrounding traffic. Realize that these variables are also especially important for the vehicle control module that is discussed later in this work in Section 4.5.

Following the vehicle index notation described above, x_α and z_α are used to denote a vehicle's position whereas v_α and a_α are used to denote the longitudinal velocity and acceleration of vehicle α, respectively. Note that position is globally defined in the world frame, while velocity and acceleration are in the longitudinal direction of the local vehicle frame.

For vehicle α, the *time headway* (or just *headway*) Δt_α is defined as the time difference between the front bumpers of vehicles a and $a-1$. From that, we can define the microscopic spatial quantities *distance headway*

$$d_\alpha = v_\alpha \Delta t_\alpha \tag{3.1}$$

and *distance gap* (or just *gap*)

$$s_\alpha = d_\alpha - l_{\alpha-1} \tag{3.2}$$

where $l_{\alpha-1}$ denotes the length of vehicle $\alpha-1$. See Figure 3.2. Likewise, the *time-gap* is the time difference between the front bumper of vehicle α and rear bumper of vehicle $\alpha-1$.

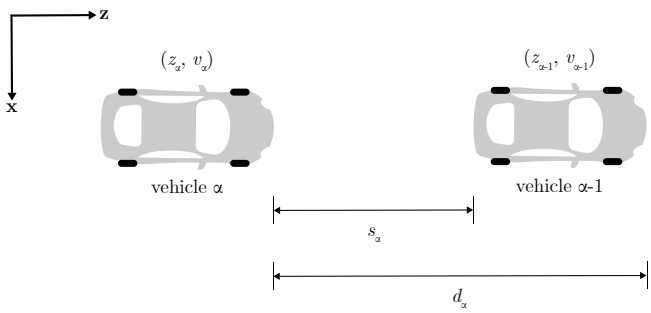


Figure 3.2: Environment coordinate system, index notation and microscopic traffic quantities.

3.1.2 Macroscopic parameters

Whereas microscopic parameters describe the states of individual vehicles, macroscopic parameters describe average state of traffic. There are three macroscopic metrics of importance in this work: mean speed V, traffic flow Q and density ρ.

The *traffic flow* (or *intensity*) is defined as the number of vehicles ΔN passing through a point within a time interval Δt:

$$Q(z,t) = \frac{\Delta N}{\Delta t} \tag{3.3}$$

Usually, it is given in units of vehicles per hour (veh/h). It should be noted that the inverse of traffic flow is the *time mean of the headways* $\Delta \bar{t}_\alpha$, the average time headway between ΔN vehicles:

$$\Delta \bar{t}_\alpha = \frac{1}{Q(z,t)} \tag{3.4}$$

The *mean speed* is the average speed of the number of vehicles ΔN that passes through the point during a time interval Δt:

$$V(z,t) = \bar{v}_\alpha = \frac{1}{\Delta N} \sum_k \sum_{\alpha=\alpha_0}^{\alpha_0+\Delta N-1} v_\alpha \tag{3.5}$$

Here $V(z,t)$ is used to denote macroscopic speed in order to distinguish it from the microscopic speed v_α of single vehicles. The traffic *density* expresses the amount of vehicles per unit length and can simply be estimated from the previous two metrics:

$$\rho(z,t) = \frac{Q(z,t)}{V(z,t)} \tag{3.6}$$

3.2 HIGHWAY DATA

As a means of constructing a realistic traffic template, actual highway traffic data is used as a reference. The Dutch National Traffic Database NDW [68] collects nationwide traffic data across the Dutch road system including the highway section used in this study, which is made available as open data. The data of four separate sensors along a three-lane highway are chosen to be included in this study. Over the span of 2019, the hourly mean speed V and traffic flow Q are obtained, which are plotted in Figure 3.3. The raw data and the exact locations of the sensors (A4 highway, Netherlands) can be found in Appendix A. It should be noted that the legal speed limit was 120 km/h during the time in which the measurements were taken as opposed to the limit of 100 km/h at the time of writing this thesis.

Based on the traffic data, three templates of varying traffic flow are defined and used in this study (see Table 3.1). Note that the traffic data gives no insight into traffic flow of each individual lane. In order to simulate fewer and faster vehicles on left lanes and vice-versa, we have taken the liberty to define V and ρ per lane separately. The parameters of the total highway sample are still in reference with the traffic flow data. The mean speed distributions have a standard deviation of 2.5 in lane 1 and 5 in lanes 2 and 3. Furthermore, lane 1 contains 20% of all highway traffic, lane 2 contains 35% and lane 3 45%.

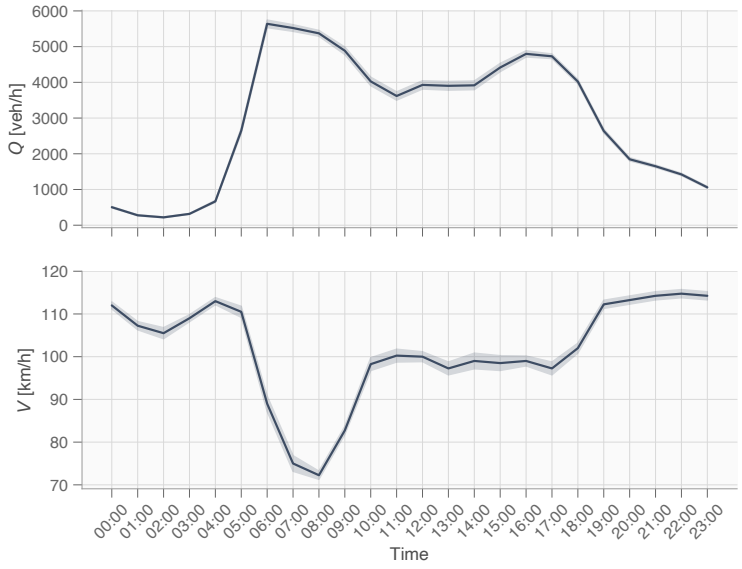


Figure 3.3: Hourly mean speed and traffic intensity across 4 measurement points on the A4 highway. The transparent boundaries represent the standard deviations.

Table 3.1: Traffic templates

	Template 1 ($Q = 1500$)		**Template 2** ($Q = 2500$)		**Template 3** ($Q = 3500$)	
	$V(SD)$	ρ	$V(SD)$	ρ	$V(SD)$	ρ
Lane 1	120 (2.5)	3	120 (2.5)	5	120 (2.5)	6
Lane 2	114 (5)	5	110 (5)	8	100 (5)	12
Lane 3	110 (5)	7	105 (5)	11	90 (5)	18
Total	114 (-)	15	110 (-)	24	100 (-)	36

3.3 INITIAL TRAFFIC STATE GENERATION

Using the traffic flow templates and pseudo-random number generators, we can generate each vehicle's initial longitudinal position and their initial and desired speeds at the start of a simulation episode. The stochastic element of each initial state helps the agent explore a wide range of states, contributing to a higher degree of generalization. It should be noted that traffic flow is assumed to be uncongested in this study, as congested traffic behaves significantly different in term of traffic flow and speed [69, 70].

SPEED DISTRIBUTION The speed distribution determines the initial and desired speed of each vehicle. A common assumption in traffic flow theory is that speed distribution on a highway can be modelled according to a normal distribution, as long as the traffic flow is uncongested [71, 72]. Moreover, Helbing et al. [73] and Treiber and Kesting [70] provide empirical evidence that the speed distribution in each lane has a distinct normal distribution with possible differences in mean speed between lanes.

As such, the speed distribution in this work is modelled as a 3-dimensional multivariate normal distribution. The distributions are assumed to be uncorrelated, i.e. the speeds in an arbitrary lane do not influence the speeds in other lanes.

HEADWAY DISTRIBUTION For random, uncongested traffic flow, it turns out that time headways can be represented by an exponential distribution [69, 74]. By taking the product of time headway Δt_α and speed v_α of a vehicle, the distance headway d_α can be calculated and through that, a spatial distribution of all cars on the highway is determined.

TRAFFIC GENERATION FUNCTION The eventual function for generating the spatiotemporal traffic distribution is highlighted in Algorithm 3.1. All the input metrics are either a set constant or defined in the traffic templates as defined in Table 3.1. For each lane, the mean time headway $\Delta \bar{t}_\alpha$ and lane density ρ_k are first determined from the lane traffic flow Q_k. Using Gaussian and exponential random number generators, each vehicle's velocity and time headway is then generated, from which the initial position is then also calculated in a cumulative fashion, as shown in line 9. Note that the exponential random number generator has a minimum output Δt_{min}, which is set to 2 s in this work, based on the well-known recommendation of 'holding a 2-second gap to the leading vehicle'. As shown in line 3, a random distance between 0 and 20 m (uniformly distributed) is added to the first vehicle's initial position in order to prevent all first vehicles from starting next to each other.

It should be noted that the used probability distributions are statistical approximations of traffic situations that were specifically measured in the cited studies and are by no means an accurate representation of the many possible traffic situations that occur on real life highways. Yet, we do argue that this method is sufficient enough for generating a varied traffic environment in the scope of this study.

Algorithm 3.1: GenerateTraffic

Input: per lane mean speed V_k, standard deviation σ_k, lane traffic density ρ_k, lane traffic flow Q_k, minimum time headway Δt_{min}

Output: initial position vector $\mathbf{z}_{\text{in}}$ and speed vector $\mathbf{v}_{\text{in}}$

1 **for** $k = 0$ **to** 3 **do**
2 $\quad \Delta \bar{t}_\alpha = 3600 \frac{1}{Q_k}$
3 $\quad z \sim \text{random.Uniform}(0, 20)$
4 $\quad v \sim \text{random.Gaussian}(V_k, \sigma_k)$
5 $\quad \Delta t \sim \text{random.Exponential}(\Delta \bar{t}_\alpha, \Delta t_{\text{min}})$
6 $\quad$ **for** $\alpha = 0$ **to** ρ_k **do**
$\quad\quad$ // Add initial position and velocity to the respective vectors
7 $\quad\quad \mathbf{z}_{\text{in}}[\alpha] = z$
8 $\quad\quad \mathbf{v}_{\text{in}}[\alpha] = v$
$\quad\quad$ // Determine new values
9 $\quad\quad z = z + \frac{v}{3.6} \Delta t$
10 $\quad\quad v \sim \text{random.Gaussian}(V_k, \sigma_k)$
11 $\quad\quad \Delta t \sim \text{random.Exponential}(\Delta \bar{t}_\alpha, \Delta t_{\text{min}})$
12 **return** $\mathbf{z}_{\text{in}}, \mathbf{v}_{\text{in}}$

3.4 SIMULATION EPISODES

As explained in Section 2.2, RL agents accumulate experiences in so-called trajectories or episodes: simulation loops that start from initial states and end at the time step T of a terminal state. All highway environment traffic simulations follow this episode-based structure.

At the start of each episode, an array containing the initial traffic states are generated according to the function described before in Algorithm 3.1. The agent is then 'spawned' in the middle lane as either the second, third or fourth car on the highway at the specified initial speed. All other surrounding traffic is then spawned in a similar fashion. These *passive vehicles* feature a static policy that prevents the execution of lane changes. Simply put, a policy that continuously outputs a lane keeping action, regardless of state. Furthermore, they attempt to reach a desired speed v_{des} that is identical to the initial speed. After the episode is initialized, it is simulated until a terminal state occurs, which happens when one of the following criteria is satisfied:

- The agent reaches the end of the highway, i.e. $z = 5000$. For agent training episodes, a different 8 km long highway environment was used (explanations for this follow later in Section 5.2).
- The agent collides with other vehicles or road boundaries, i.e. the metal guardrails or concrete walls.
- The episode time limit is reached, provided it is set. Normally, this limit only exists in training episodes where it has a specific purpose. Details and explanations for this follow later in Section 5.2.

Due to the highway environment having a finite length, all passive vehicles that manage to reach the physical end of the highway before the episode is terminated will be instantaneously moved to the starting position of the highway (i.e. $z = 0$) where it will have the same velocity. If one or more passive vehicle collide, the vehicles in question will simply be removed for the remainder of the episode, although this did not occur unless the lane change agent was involved.

4 AGENT DESIGN

This chapter discusses the relevant design elements within the context of a DRL modelling framework. The overall framework is described in Section 4.1. Whereas, as stated before, the 3D environment is developed in Unity, the DRL agent is implemented using Unity's machine learning toolkit, ML-agents[1] [75]. All the following RL elements are discussed in-depth: environment state observation (Section 4.2), the policy action space (Section 4.3) and the reward function (Section 4.4). In addition, Section 4.5 describes the operational vehicle control module that is responsible for steering, velocity control and lane change trajectory generation. As stated before, this is part of every vehicle in the simulator, but it is included in this chapter for better readability.

4.1 OVERALL AGENT MODELLING FRAMEWORK

As stated before, a hierarchical approach is adopted where the neural network policy is solely responsible for environment perception and high-level decision-making whereas a so-called *vehicle control module* is delegated responsibility for operational control (i.e. steering, throttle control and trajectory generation). The overall framework structure is described in Figure 4.1. Note that the figure is similar to Figure 2.3, but that the vehicle control module is displayed as a separate block in order to emphasize that the policy is only responsible for the decision-making and that it can not (directly) alter the environment. Instead, the vehicle control module is directed by the chosen action A_t and subsequently creates the next state S_{t+1}. Furthermore, the agent can force a new episode which resets the environment (as previously described in Section 3.4). This cycle is evaluated every time step Δt.

4.1.1 Updating frequency

Unity computes (physics) calculations in the FixedUpdate() function, which by default, is called at 50 Hz, i.e. every 0.02 seconds. However, using the function provided by Unity ML-agents, RequestDecision(), the DRL agent experience cycle (as depicted in Figure 2.3) can be evaluated at a different rate. In this work, an experience (i.e. a sequence of $[S_t, A_t, S_{t+1}, R_{t+1}]$) is evaluated every time step $\Delta t = 0.1$ s if the agent is lane keeping. The reasoning behind this is as follows:

- Choosing a lower frequency reduces the amount of experience arrays to be processed per unit of physical time, thereby reducing the computational cost to process a training episode at the expense of a lower resolution. This is especially advantageous with complex algorithms such as PPO, which performs expensive gradient ascent/descent two times per update (Algorithm 2.1).

1 https://github.com/Unity-Technologies/ml-agents

- Not evaluating an experience cycle during a lane change prevents the agent from choosing another action, thereby preventing the agent from aborting a lane change once it is initiated, which is by design.

- There is evidence of DRL algorithms collapsing in environments where Δt is close to zero [76].

- It would be difficult to capture the long-term consequences of strategic lane change decisions if trajectories are computed within a time horizon that provides little to no new information. In slower paced control tasks, such as strategy games, it was found a larger time step resulted in higher performance, presumably due to agents then "having a greater capacity to learn associations between more temporally distant states and action" [77]. In this work, it is assumed that the environment state will not drastically change (i.e. provide no new information) at a rate of 50 Hz and that a lower value is needed to capture the strategic value of efficient lane change decision-making.

- A policy solution may be found that relies on a system being able to control at such a high frequency. Realistically, it may simply be unnecessary to evaluate a lane change decision at such a fast rate.

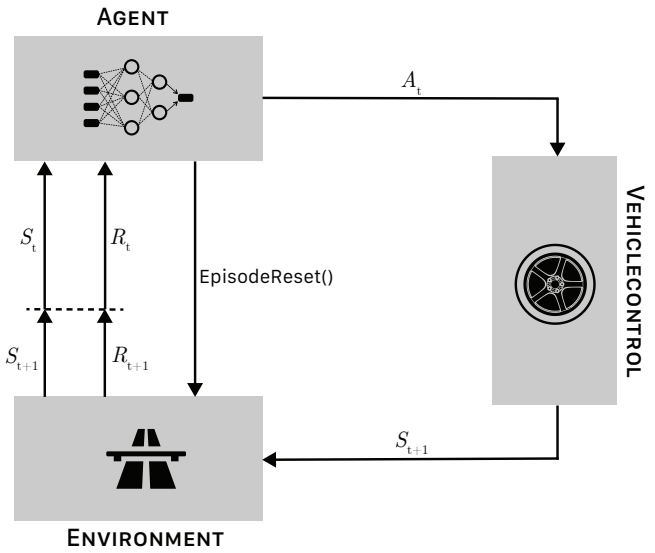


Figure 4.1: The overall structure of the DRL lane change modelling framework.

4.2 OBSERVED STATE

As is the case with many automated vehicle technologies, the state of the environment that the agent can observe is dependent on the integrated sensors. That is, only measurements within the agent's range d_{meas} are observed, making the state in this problem partially observable. The observed state $S_t = [s_1 ... s_{17}] \in \mathcal{S}$ consists of both ego-vehicle as well as surrounding vehicle information. The full input observation state vector including descriptions is summarized in Table 4.1.

AGENT SPEED The agent's speed s_1, is normalized between the minimum and maximum speeds $v_{\max}$ and $v_{\min}$ km/h respectively. In this work, we set $v_{\min} = 80$ km/h and $v_{\max} = 120$ km/h.

CURRENT LANE States $s_{2...4}$ describe the agent's current lane and are elements in an enumeration encoded in the *one-hot* style. If the agent is in a lane, a value of 1 (which can be interpreted as True) is used to represent the state corresponding to that lane. Otherwise, it is simply 0 (or False). Note that since the environment consists of three lanes, this enumeration is also three elements long.

OBSERVATION GRID States $s_{5...16}$ describe the surrounding vehicles' relative positions and velocities, mapped in a grid-like formation. The grid is comprised of the leading vehicle in the left-hand lane, followed by the successive vehicle in the left-hand lane, the leading and successive vehicles in the current lane and the leading and successive vehicles in the right-hand lane respectively. In order to mimic the typical scanning range of a LIDAR sensor, all vehicles within a range of $d_{\text{meas}} = 200$ meters will be included [78]. Relative distances are normalized between 0 and this 200 meter limit whereas the relative velocities are normalized between the minimum and maximum speeds $v_{\min}$ and $v_{\max}$ respectively. If a vehicle is outside of the measuring range, it will be classified as a vehicle driving at this range with no speed difference, i.e. the values 1 and 0 will be assigned to relative position and velocity respectively.

WARNING TO GO RIGHT State s_{17} indicates whether the agent has space to return to a slower right-hand lane if it is driving in the left-most overtake lane. The value is a boolean that is True if there is both space in front and behind in the right-hand lane, otherwise it is False. Having space in front is defined as the time gap to the preceding vehicle being at least 3 seconds and the time-to-collision (TTC) being more than 20 seconds (provided it is positive). Conversely, having space behind is defined as the space gap to the successive vehicle being more than 60% of that vehicle's spacing policy, i.e. $s_{\alpha+1} > 0.6 s_{\text{des}}$. Note that these parameters are arbitrarily chosen through trial-and-error and can easily be changed for more aggressive or conservative behavior.

As can be seen, all states are either normalized floats or booleans, as is recommended for faster convergence during training. If any state exceeds its boundaries, it is clamped to either $[0, 1]$ or $[-1, 1]$, depending on the state.

Table 4.1: Input observation state vector S_t of the neural network

State	Description	
s_1	normalized speed	$\frac{v_i - v_{\min}}{v_{\max} - v_{\min}}$
$s_{2...4}$	current lane	True/False
s_5	relative longitudinal position of vehicle $\alpha - 1$ in lane $k - 1$	$\frac{\Delta z_\alpha}{d_{\text{meas}}}$
⋮		
s_{16}	relative velocity of vehicle $\alpha + 1$ in lane $k + 1$	$\frac{\Delta v_\alpha}{v_{\max} - v_{\min}}$
s_{17}	move to right	True/False

4.3 ACTION SPACE

Since the neural network is used for lane change decision-making, a discrete action space is used in this work. Hence, each action refers to a different set of lateral and longitudinal control references that is then executed through the vehicle control module. Some works such as that by Liu et al. [23] use two separate action space branches where the lateral- and longitudinal dimensions are regulated independently. In this work however, only one branch is used that controls both dimensions simultaneously. The three possible actions are described in Table 4.2 and correspond to left lane change, keep lane and right lane change respectively where the leading vehicle in the target lane is designated as the following target. The agent is designed to always pursuit a desired speed of 120 km/h, i.e. $v_{\text{des}} = v_{\text{max}}$.

4.3.1 Discrete action masking

When using a discrete action space, it is possible to prevent certain actions from being chosen for the next decision. In Unity ML-agents, this mechanism is called *action masking*. As previously discussed in Chapter 1, several works successfully used comparable techniques in their methods to prevent unsafe or irrelevant actions in order to accelerate training, where it is commonly called *safety verification/checking* or *safe exploration* [21–23, 25–28].

Action masking not only benefits the agent during training. Additionally, due to the nature of DRL, there is always a possibility where a fully trained agent still chooses to engage in dangerous behavior. This is especially possible when the inference environment is different compared to the training environment. This risk can additionally be mitigated through the use of action masking even after the model has been trained.

Two action masks are implemented in this work:

1. If the agent is in the left-most lane, then a_1 is unavailable. Similarly, if the agent is in the right-most lane, then a_3 is unavailable. This prevents the agent from driving off the highway.
2. If either the TTC falls below 1 second or the space gap with any car in an adjacent lane falls below 2 meters, then a lane change towards that lane is unavailable. Note that these values are arbitrarily chosen (with trial-and-error) in order to accelerate learning and are by no means considered hard-constraints on what is widely considered safe or unsafe. Colliding and/or dangerous behaviour is still possible.

Table 4.2: Output action space for the neural network

action	longitudinal	lateral
a_1	follow vehicle $\alpha - 1$ in lane $k - 1$	left lane change to $k - 1$
a_2	follow vehicle $\alpha - 1$ in lane k	keep current lane k
a_3	follow vehicle $\alpha - 1$ in lane $k + 1$	right lane change to $k + 1$

4.4 REWARD FUNCTION

Since the goal of RL is to generate the policy that maximizes the reward, the reward function has a significant effect on the resulting behavior. For this reason, it is supposed to reflect the following driver goals:

1. Drive safely.
2. Drive as fast as possible.
3. Drive efficiently.
4. Drive socially.

The reward function is defined as follows:

$$R_t = \begin{cases} -1 & \text{if collided} \\ 0.01 r_{\text{vel}} + 0.05 r_{\text{o}} - 0.01 r_{\text{ld}} - 0.01 r_{\text{lc}} - 0.05 r_{\text{danger}} & \text{otherwise} \end{cases} \tag{4.1}$$

where

$$r_{\text{vel}} = \frac{v - v_{\text{min}}}{v_{\text{max}} - v_{\text{min}}} \quad \in \quad [0, 1] \tag{4.2a}$$

$$r_{\text{o}} = \begin{cases} 1 & \text{if overtaking from the left} \\ -1 & \text{if overtaking from the right} \end{cases} \tag{4.2b}$$

$$r_{\text{ld}} = \begin{cases} 1 & \text{if unnecessarily driving on overtake lane} \\ 0 & \text{otherwise} \end{cases} \tag{4.2c}$$

$$r_{\text{lc}} = \begin{cases} 1 & \text{if lane changing} \\ 0 & \text{otherwise} \end{cases} \tag{4.2d}$$

$$r_{\text{danger}} = \begin{cases} 1 & \text{if tailgating or cutting off} \\ 0 & \text{otherwise} \end{cases} \tag{4.2e}$$

Even though it is previously stated that the agent manages a different update frequency compared to the Unity simulator, the reward is evaluated every Unity update loop (i.e. every 0.002 s). This is done to capture important events during lane changes such as tailgating. The numerical reward values are then simply summarized until they are added to the next experience. In summary, the agent can collect the optimal reward through driving as fast and right-sided as possible with minimum lane changing, dangerous driving and colliding with other cars or the environment. The reward function was designed according to the following reasoning:

- By normalizing the velocity reward, zero positive rewards are collected when the agent drives at the minimum velocity and consequently the maximum value when driving at the desired (i.e. maximum) velocity.
- The negative reward for lane changing is to discourage unnecessary lane changing, forcing the agent to only perform lane changes when it estimates it could lead to a potential future reward (i.e. an overtake leading to an increase in speed).
- Similarly, a negative reward is assigned for dangerous behavior. In this work, tailgating is defined as the space gap s_α to the preceding vehicle being lower than

60% of the desired spacing s_{des} (see Equation 4.5). Furthermore, if a following vehicle has to emergency brake (defined as a braking torque of 500 N/m or higher) or the gap is lower than 60% of the spacing as a result of an agent lane change, then the behavior is classified as cutting off.

- Social driving is accomplished by an overtaking reward and left driving penalty. Overtaking is positively rewarded if it is done on a left lane, whereas overtaking from the right results in an equal negative reward. Driving on the left is only classified as unnecessarily if $s_{17} = \text{True}$ (see Section 4.2). This approach is slightly similar to that of [29, 30], except that this study utilizes TTC instead of time-gap in order to account for both relative distance and velocity.

The choice of all reward weights were determined using mostly trial-and-error, as there is no standardized method for designing the reward function apart from some best practice rules.

4.5 VEHICLE CONTROL MODULE

By default, vehicles in Unity are controlled through three inputs: motor torque, braking torque and steering angle. The so-called vehicle control module is responsible for regulating inputs and consists of three distinct features: longitudinal velocity control using ACC, trajectory generation and trajectory tracking control. As previously stated, all vehicles in the simulator possess this module, including the agent.

4.5.1 Velocity control

Similar to a real traffic scenario on a highway, each vehicle is supposed to adjust its velocity to the surrounding traffic. This (longitudinal) velocity control can be achieved using an Adaptive Cruise Control (ACC) system or a car-following model, enabling the (automated) maintenance of a desired velocity v_{des} as well as a desired following gap s_{des} of vehicle α with respect to a preceding vehicle $\alpha - 1$.

Since acceleration is proportional to motor torque in the wheels [79], a torque based model that is similar to the acceleration based model of van Arem et al. [80] can be used in this work. Imagine the situation in Figure 3.2. In every update loop, two different torque signals are computed: a reference torque signal T_{ref} and a gap-control torque signal T_{gc}. The reference torque is designed to reach a defined desired velocity v_{des} as in

$$T_{\text{ref}} = k_r \left(v_{\text{des}} - v_\alpha \right) \tag{4.3}$$

where k_r is a constant speed error factor. The gap-control torque is designed to preserve a defined inter-vehicle distance (or gap) s_{des} between the ego-vehicle and preceding vehicle and is based on the preceding vehicle's torque, relative velocity and inter-vehicle gap error. It is defined as

$$T_{\text{gc}} = k_T T_{\alpha-1} + k_v \left(v_{\alpha-1} - v_\alpha \right) + k_d \left(s_\alpha - s_{\text{des}} \right) \qquad \text{if} \quad z_{\alpha-1} - z_\alpha \leq d_{\text{meas}} \tag{4.4}$$

where k_T, k_v and k_d are constants. If a preceding vehicle is not within a pre-defined measuring distance d_{meas}, then T_{gc} will have no value as a means of limiting the computational cost.

The desired gap s_{des} is determined through the spacing policy. Zhou and Peng [81] formulated a velocity-based spacing policy based on human driving data which will be used in this work. It is defined as

$$s_{\text{des}} = 3 + 0.0019v_\alpha + 0.0448v_\alpha^2 \tag{4.5}$$

The torque is then determined by selecting the most restrictive torque signal:

$$T_\alpha = \min\left(T_{\text{ref}}, T_{\text{gc}}\right) \tag{4.6}$$

The input motor- and braking torques are then determined from T_α and simultaneously clipped using the set parameters $T_{\text{max,motor}}$ and $T_{\text{max,brake}}$ as in:

$$T_{\text{motor}} = \text{clip}\left(T_\alpha, 0, T_{\text{max,motor}}\right) \tag{4.7}$$
$$T_{\text{brake}} = \text{clip}\left(T_\alpha, T_{\text{max,brake}}, 0\right) \tag{4.8}$$

preventing the system from exhibiting unstable and/or unrealistic behavior. Note that when T_α is negative the vehicle brakes whereas a positive value leads to acceleration; simultaneous braking and acceleration is impossible.

4.5.2 Lane change trajectory

Typically, lane change trajectories can be modelled with mathematical equations, i.e. 'ideal trajectories' [82]. Examples vary from sinusoidal [83] or (quintic) polynomial [84] functions to more advanced dynamic mathematical systems like the approach used by Xu et al. [85]. Some works also opted to fit a curve to a large set of collected human driving data [86–88].

For the sake of preserving simplicity, a quintic polynomial function is used to model the (spatial) lane change trajectory. Since a quintic polynomial is twice continuously differentiable, it holds the advantage of having a continuous curvature profile, resulting in smooth steering without the need for sudden adjustments [84, 89]. Consider a road vehicle on a lane of width w where z represents the longitudinal direction and $x(z)$ is the lateral position expressed as a function of z, as in

$$x(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4 + a_5 z^5 \tag{4.9}$$

If a lane change maneuver has a length z_{lc} and width x_{lc} (i.e. the lane width w), the unknown constants in Equation 4.9 can be solved by introducing the following boundary conditions:

$$\begin{cases} x = 0, \ \dfrac{\mathrm{d}x}{\mathrm{d}z} = 0, \ \dfrac{\mathrm{d}^2x}{\mathrm{d}z^2} = 0 & \text{at } z = 0 \\ x = x_{\text{lc}}, \ \dfrac{\mathrm{d}x}{\mathrm{d}z} = 0, \ \dfrac{\mathrm{d}^2x}{\mathrm{d}z^2} = 0 & \text{at } z = z_{\text{lc}} \end{cases} \tag{4.10}$$

Solving this boundary condition problem results in the trajectory

$$x(z) = x_{\text{lc}}\left[10\left(\frac{z}{z_{\text{lc}}}\right)^3 - 15\left(\frac{z}{z_{\text{lc}}}\right)^4 + 6\left(\frac{z}{z_{\text{lc}}}\right)^5\right] \tag{4.11}$$

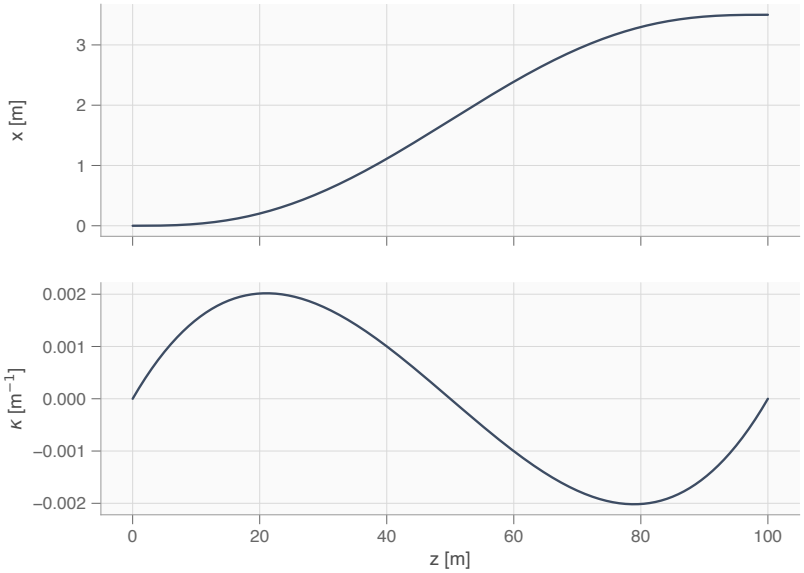


Figure 4.2: The lateral position and curvature during a 100 m long lane change maneuver on 3.5 m wide lane(s). Here, $x = 0$ corresponds to the center of the original lane.

Furthermore, the curvature $\kappa(z)$ [90] is defined as

$$\kappa(z) = \frac{\left| \left(\mathrm{d}^2 x / \mathrm{d}z^2 \right) \right|}{\left(1 + \left(\mathrm{d}x / \mathrm{d}z \right)^2 \right)^{3/2}} \tag{4.12}$$

To illustrate, both the resulting lateral trajectory and curvature of a 100 m long, 3.5 m wide lane change are plotted against longitudinal position in Figure 4.2. Given that the constraints are met, the trajectory can be applied to any lane change segment with an arbitrary starting location and orientation in the simulator by transforming from the coordinate frame shown in Figure 4.2. Naturally, a trajectory towards the right lane can simply be obtained by multiplying Equation 4.11 with -1.

To prevent a constant lane change length z_{lc} for all situations, the lane change length will be defined as a function of lane change duration t_{lc} which is a constant and the velocity at the time of initiating the lane change maneuver v_{lc}:

$$z_{\mathrm{lc}} = t_{\mathrm{lc}} v_{\mathrm{lc}} \tag{4.13}$$

4.5.3 Trajectory tracking module

With the trajectories for lane changing and lane keeping (i.e. a straight line in the middle of the lane) defined, the car follows these reference trajectories using the Stanley method [91]. Referring to the situation in Figure 4.3, the Stanley method is a nonlinear steering control method that considers the the cross track error e_{ct} measured laterally from the center of the front axle of the vehicle to the trajectory point (x_p, z_p) and the heading error θ_e of the vehicle with respect to the path. The control law is given as

$$\delta = \theta_e + \arctan\left(\frac{k e_{\mathrm{ct}}}{v_\alpha} \right) \tag{4.14}$$

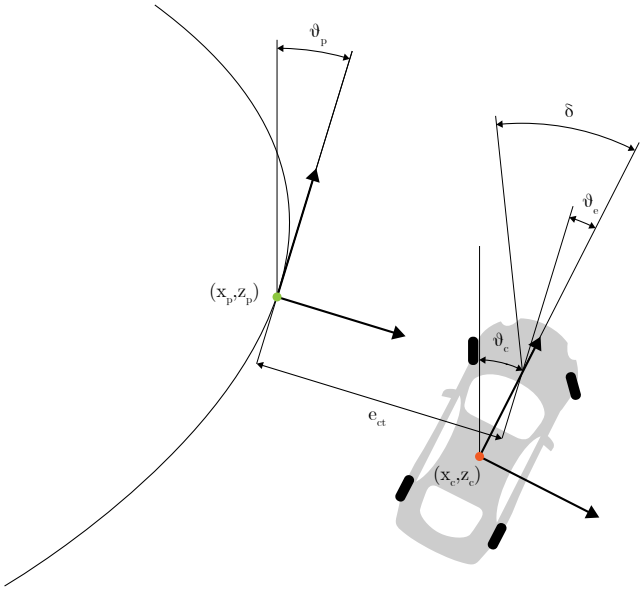


Figure 4.3: The Stanley method steering geometry [92].

where k is the manually tuned gain parameter and v_α the longitudinal velocity of the vehicle. The heading error can be determined as

$$\theta_e = \theta_c - \theta_p \tag{4.15}$$

where θ_p is the heading of the trajectory and θ_c the heading of the vehicle. The Stanley method control law is proven to exponentially converge to zero cross track error and, if sufficiently tuned, performs well at high speeds compared to other popular methods such as the Pure Pursuit method [92].

4.5.4 Parameter tuning and resulting behavior

This section highlights all the chosen vehicle control parameters along with a short motivation. The parameter values can be found in Table 4.3.

VELOCITY CONTROL The velocity control aspect features the most parameters that need to be tuned. Although literature provides reference values for the error factors [80], different values are adopted in this study since Unity utilizes torque signals instead of accelerations. Therefore, these parameters were manually tuned until reasonable car following behavior and velocity control was achieved without unrealistic accelerations and overshooting. Like stated before, acceleration is a linear function of torque [79]. To verify this and allow for comparison with other literature, a linear regression analysis is conducted on approximately 2000 driving data samples representing several scenarios such as regular driving, heavy and medium braking and overtaking using the parameters described in this section. As expected, a significant regression equation was found $F(1, 1925) = 1.1059 \times 10^4, p < 0.001$, with an R^2 of 0.846. Torque can be expressed in acceleration through:

$$a_\alpha = -0.2212 + 0.0047 T_\alpha \tag{4.16}$$

The maximum motor- and braking torques used for clamping ensure a maximum acceleration and deceleration of approximately 4.5 m/s² and -5 m/s², respectively. Note that the regression analysis was performed with these clamping parameters already implemented.

LANE CHANGE TRAJECTORY Tuning the lane change parameters is relatively straightforward. The width is simply equal to the lane width of the highway environment. When it comes to lane change duration, researchers have reported varying mean values, such as 3.91 s [93] or 4.60 s [94]. At first glance, the lane change duration in this work appears to be relatively short compared to the cited empirical evidence, although it is certainly not uncommon considering that the data in these studies follows a log-normal distribution with standard deviations of 2.34 s and 2.30 s, respectively.

TRAJECTORY TRACKING The trajectory tracking module only featured a single parameter: the gain parameter k. Since no reference values can be obtained from literature, this parameter was manually tuned until satisfactory steering behavior was achieved that converged fast enough without overshoot or oscillation.

Table 4.3: Vehicle control parameters

Parameter	Symbol	Value
Speed error factor	k_r	250 Ns/m²
Torque error factor	k_T	1
Speed error factor	k_v	60 Ns/m²
Gap error factor	k_d	30 N/m²
Measuring range	d_{meas}	200 m
Max motor torque	$T_{\text{max,motor}}$	1000 N
Max braking torque	$T_{\text{max,brake}}$	-1000 N
Lane change time	t_{lc}	2.5 s
Lane change width	x_{lc}	3.5 m
Gain parameter	k	0.4 rad/s

5 METHOD

As mentioned, the experimental part of this study consists of two main parts: the development of the DRL highway agent and the subsequent acceptance assessment. In both parts, the DRL-trained model, from now on referred to as the neural network (NN) model, is compared against a baseline model. Firstly, the baseline rule-based model will be introduced in Section 5.1. The agent training and inference method can be found in Section 5.2. The acceptance assessment method, which is the main contribution of this study, can be found in Section 5.3.

5.1 IMPLEMENTATION OF BASELINE MODEL

The baseline model used in this work is MOBIL, as described in Section 2.1.1. Similarly to the DRL-based model, MOBIL outputs a decision every $\Delta t = 0.1$ s while lane keeping and follows $v_{\text{des}} = 120$ km/h. As mentioned before, Unity uses torque as opposed to acceleration as input, meaning that similarly to the velocity control module (see Section 4.5.1), model parameters had to be adapted to fit the simulators working principles. This was achieved by replacing all acceleration terms with their respective torque equivalents in Equation 2.1 and Equation 2.3.

The values of the MOBIL parameters can be found in Table 5.1. The three torque values are different compared to the values used by both Kesting et al. [6] and Alizadeh et al. [33], which were approximately $b_{\text{safe}} = -800$ Nm, $\Delta T_{\text{th}} = 70$ Nm and $\Delta T_{\text{bias}} = 110$ Nm. It should be mentioned that the authors initially used these values as well, but that the resulting MOBIL behavior exhibited an abnormally high lane change rate (e.g. regularly exceeding 22 lane changes per 200 s episodes) and caused several collisions. In order to facilitate a fair comparison, the chosen maximum safe braking torque was made identical to the threshold at which the DRL agent was penalized during training and the threshold and bias torques were adapted to result in behavior that was less oscillatory.

Table 5.1: MOBIL parameters

Parameter	Symbol	Value
Politeness factor	p	0.5 (balanced)
Changing threshold	ΔT_{th}	200 Nm
Bias for right lane	ΔT_{bias}	300 Nm
Maximum safe braking torque	b_{safe}	-500 Nm

5.2 AGENT TRAINING AND INFERENCE

Given the agent and the environment described in the previous chapters, the model was trained using Tensorboard on a Windows 10 system with a Ryzen 3600 CPU clocked at 4.2Ghz, 16GB RAM and a Nvidia RTX 2060 Super GPU. A special 8 km long highway environment without any aesthetic elements was created specifically for training (a lower computational load enables faster training). The longer highway length was chosen in order to ensure that all long-term effects of lane changes were captured.

Each training episode lasted 10000 Unity update steps (corresponding to 200s of time) unless the episode was terminated early due to a collision. This time-based termination was favoured over a spatial termination policy (e.g. terminate the episode after x km) in order to prevent the agent from 'misusing' the system by driving in slow lanes; episodes would then take longer to complete, thereby accumulating more points as opposed to driving as fast as possible and finishing the episode earlier. Training was terminated after 5-million time steps, a value which was arbitrarily chosen, albeit large enough in order to ensure that the agent has enough time to explore and converge to a solution. To further accelerate the training process, all simulations were run at 100 times the original time rate.

In addition to the training session, inference trials of 1-million time steps (i.e. 100,000 seconds) were run for all three traffic templates (see Table 3.1) per model.

5.2.1 Proximial Policy Optimization (PPO)

In this work, PPO was used as the reinforcement learning algorithm. More detailed information regarding the working principles can be found in Section 2.3.2. It should be noted that Unity's implementation of PPO uses a so-called 'experience buffer' for storing the discounted returns and estimated advantages, which is first filled prior to updating the policy through gradient descent as opposed to the vanilla implementation which performs updates every T time steps. The used hyperparameters can be found in Table 5.2. Due to the computational complexity, long training times and large set of possible hyperparameter combinations, all values were determined through trial-and-error and informal searches as opposed to a grid search.

Table 5.2: Training hyperparameters

Hyperparameter	Symbol	Value
Minibatch size	M	64
Buffer size	-	4096
Learning rate	α	1e-5
Beta	β	0.05
Epsilon	ϵ	0.2
Time horizon	T	256
Number of epochs	K	10
Discount factor	γ	0.99
GAE parameter	λ	0.95

5.2.2 Evaluation metrics

For training, the following metrics were collected: cumulative reward, mean normalized velocity, number of lane changes per episode and cumulative number of collisions. In the inference trials, mean normalized velocity and number of lane changes were measured. We report descriptive statistics (mean, standard deviation, minimum, maximum) for all metrics per traffic condition across both trials.

5.3 ACCEPTANCE ASSESSMENT

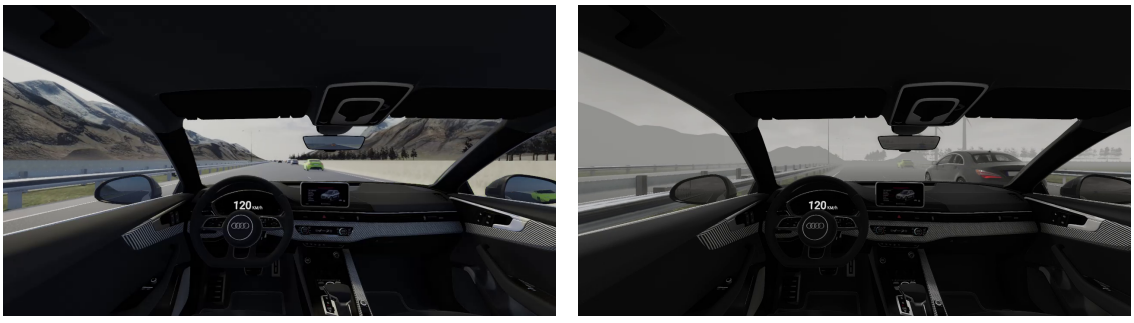
User acceptance of the system was quantitatively measured in a crowdsourcing experiment where participants had to perform a key-pressing task complemented by a questionnaire while watching demonstration videos of both the NN model and MOBIL. The questionnaire is based on the constructs of theoretical models from existing technology acceptance literature (as described in Section 2.4) and the items are adapted to fit the system's context.

5.3.1 Questionnaire design

Table 5.3 highlights all the items per construct. All construct questionnaire items were measured on a 5-point Likert scale (1 = *disagree strongly*, 2 = *disagree a little*, 3 = *neither agree nor disagree*, 4 = *agree a little* and 5 = *agree strongly*).

5.3.2 Videos

Four simulations were run as a 2×2 combination of model (NN and MOBIL) and weather condition (clear sunset and mist[1], creating a total of 4 experimental conditions. Each simulation was run in 2-minute long episodes and recorded for 30 minutes, resulting in exactly 1 hour of video per lane change model. These recordings were subsequently cut into equal 1-minute long segments (i.e. half an episode). All video were 1280 pixels wide and 720 pixels high (720p) and filmed from the first person view of a passenger in the driver seat with working mirrors and speedometer (see Figure 5.1).



(a) In the sunset condition

(b) In the mist condition

Figure 5.1: Snapshots of videos used in the experiment.

1 The misty weather was achieved using the Unity 'Fog' setting in post-processing, with a value of 0.03, obstructing the visual range beyond 40 m.

5.3.3 Procedure

Participants were recruited online and performed the experiment remotely via the crowdsourcing service Appen (https://appen.com/), where they would login and then see our study listed among other experiments. They would then self-enroll for the study. Participants from all countries were allowed to participate. Each participant was only allowed to complete the experiment once. A payment of USD 0.25 was offered for the completion of the experiment.

At the beginning of the study, the contact information of the researchers was provided, followed by a short description of the purpose of the study which was described as "*to evaluate the overall acceptance of an automated vehicle system changing lanes on a highway*". Participants were then informed that they were free to contact the investigator for questions, that they had to be at least 18 years old and that the study would take approximately 30 minutes. A short informative section about anonymity and voluntary participation then followed. Participants provided informed consent via a dedicated questionnaire item. The research was approved by the Human Research Ethics Committee of the Delft University of Technology.

Next, the participants were asked to answer questions regarding their sociodemographic information (gender, age, nationality), primary mode of transportation, driving frequency and mileage in the last 12 months and involve in accidents. The full Appen questionnaire can be found in Appendix C.

The participants were then redirected to another website with the video experiment via a clickable link. The following instructions were provided:

> You will watch 32 1-minute long videos of an automated vehicle driving on a highway in various weather conditions. The vehicle is always in automated mode and will either change lanes (left/right) or stay in the lane. The speed limit is 120 km/h. The left lane should only be used for overtaking. Please try to imagine as if you are an occupant during a normal ride in the vehicle. During each video press the key 'F' whenever you feel the car changes lanes when it should not change lanes, or when it does not change lanes when it should change lanes. You can press the key as many times as you want per video. The window of your browser should be at least 1300px wide and 800px tall. Press 'C' to proceed to the first video.

Each participant was randomly assigned a subset of 32 out of 120 videos which were a 2×2 combination of model (NN vs MOBIL) and weather condition (clear sunset vs mist), i.e. each participant watches an equal amount of videos per model and weather condition. This means that there was a approximate 26.7% chance that a participant was assigned a specific video. The subset of 32 videos was presented in random order in four batches of 8. After each batch, the participants were prompted with a message indicating their progress, such as: "You have now completed 16 videos out of 32. When ready press 'C' to continue to the next batch." Each video is followed by the acceptance questionnaire with the four items presented in random order. After completing the experiment, a unique code string was presented to the participants which they were required to enter into the Appen questionnaire as proof for participating in order to receive their compensation.

Table 5.3: Acceptance constructs, their items and sources

Construct		Items
Performance expectancy	PE	The vehicle's driving behavior is efficient.[a]
Safety	S	The vehicle's driving behavior is safe.[b]
Human-likeness	HL	The vehicle's driving behavior is human-like.[c]
Reliability	R	The vehicle's driving behavior is reliable.[d]

Note: all items are rated on a five-point Likert scale, i.e. 1 = *disagree strongly*, 2 = *disagree a little*, 3 = *neither agree nor disagree*, 4 = *agree a little* and 5 = *agree strongly*.
[a] Modified from Venkatesh et al. [36]
[b] Modified from Osswald et al. [37]
[c] Custom item
[d] Modified from Choi and Ji [46]

5.3.4 Data-analysis

Descriptive statistics (mean, standard deviation) of the number of key responses and questionnaire scores were calculated per model and weather condition, aggregated over all videos. Statistical differences between lane change models (NN and MOBIL) and weather condition (sunset and mist) were assessed using two-way repeated-measures ANOVA. Additionally, relationships between participant age and gender and number of disagreements (all videos aggregated) were investigated using a Spearman correlation analysis whereas correlations between all five acceptance metrics were assessed using a Pearson correlation analysis.

Descriptive statistics (mean, standard deviation) of the same metrics were calculated on the video-level (aggregated over all participants) to investigate differences between individual videos, separated by model. Correlations between number of disagreements and questionnaire scores on the video-level were investigated using a Pearson correlation analysis.

For each video, the average number of cumulative key presses were calculated as a function of time, aggregated over all participants. Time segments with particularly sharp increases in presses were visually analyzed.

The safety of all lane changes was evaluated and categorized in terms of 'urgency', 'severity' and minimum TTC according to the method by Lee et al. [95]. Correlations between these metrics and percentage of participants that pressed the button at least once during that lane change were investigated using a Spearman correlation analysis.

All data analysis was performed in Python using the packages `pandas` and `scipy.stats`.

6 RESULTS

Similarly to Chapter 5, this chapter also consists of two distinct parts in order to preserve consistency. The agent training and inference results can be found in Section 6.1. The acceptance assessment results, which is the main contribution of this study, can be found in Section 6.2.

> **Note**
> The Unity simulator source files, including all the written code and the trained neural network can be found at https://github.com/danielvdhaak/DRL-highway.

6.1 AGENT TRAINING AND INFERENCE

In this section we first showcase the training process of the highway agent. As stated in the previous chapter, the model was trained over 5-million steps whereas several inference trials of 1-million steps each were performed afterwards.

6.1.1 Training results

Figure 6.1 plots the cumulative reward, mean normalized velocity, number of lane changes per episode and total number of collisions against time during training. The lines in the first three plots (cumulative reward, normalized velocity and lane changes per episode) are smoothed for a clearer overview using a moving average filter of 10 data points wide through the function `pandas.DataFrame.rolling()`. The agent starts with a random policy and zero learning, explaining the low cumulative rewards, abnormally large amount of lane changes and presence of collisions at the start. Several observations can then be made about the training process.

Firstly, as is reflected by Figure 6.1a, the cumulative reward rapidly increases between the 50K-250K period, generally indicating that the agent is learning. This is complemented by a sharp decline in the number of lane changes and the absence of any collisions after this period. Note that the presence of the discrete action mask (i.e. *safe learning*) does not prevent the possibility of colliding with other vehicles as demonstrated by the 6 collisions in this time period.

What follows is a period (250K-1M) where the resulting policy mainly focuses on car-following with a low lane changing rate (see Figure 6.1c). The absence of collisions demonstrates that the agent has learned to avoid colliding with other vehicles. As the environment randomly selects a traffic template, there is a subsequent random chance that the agent will be impeded by a preceding car. This explains the high variance of the cumulative reward and the normalized velocity during this period.

After 1M steps, the amount of lane changes increases slightly. Furthermore, the cumulative reward slowly variance decreases. These facts indicate that the agent has learned to consistently and successfully overtake without relying on the environment's random traffic seed for a high reward.

6.1.2 Inference results

Table 6.1 shows the descriptive statistics (mean, standard deviation, min and max values) of the normalized velocity per episode and lane changes per episode of the inference trials across all three traffic templates per model. Neither model caused any collisions.

In general, both models were able to maintain a velocity close to the desired value, with mean values ranging between 0.98 and 0.80. However, the results indicate that the NN model was able to maintain a higher velocity on average at the expense of fewer average lane changes compared to the MOBIL model. The NN model data also exhibits a lower variance and spread, indicating more consistent behavior.

Logically, mean normalized velocity decreases as traffic becomes denser while the number of lane changes increases, presumably due to a larger amount of vehicles potentially slowing down the agent. Furthermore, the variance of both metrics increases with increasing traffic flow.

Table 6.1: Inference trial descriptive statistics

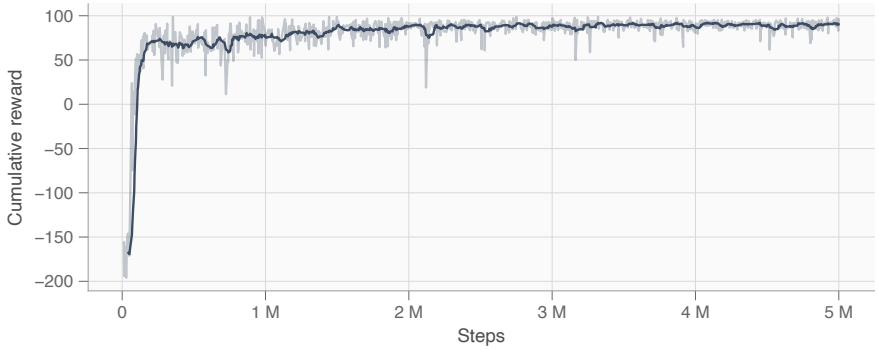
Model	**Traffic flow**	**Normalized velocity**				**Lane changes**			
	Q	M	SD	min	max	M	SD	min	max
NN	1500	0.98	0.02	0.72	0.99	2.79	1.45	0	9
	2500	0.93	0.08	0.57	0.90	4.31	1.36	1	9
	3500	0.89	0.13	0.25	0.98	5.23	1.45	1	10
MOBIL	1500	0.95	0.05	0.73	0.98	4.89	1.98	1	9
	2500	0.90	0.09	0.59	0.98	7.82	2.64	1	14
	3500	0.80	0.15	0.19	0.98	12.7	3.36	1	21

Note: data was gathered over 1 million time steps (i.e. 100,000 seconds) of driving per condition.

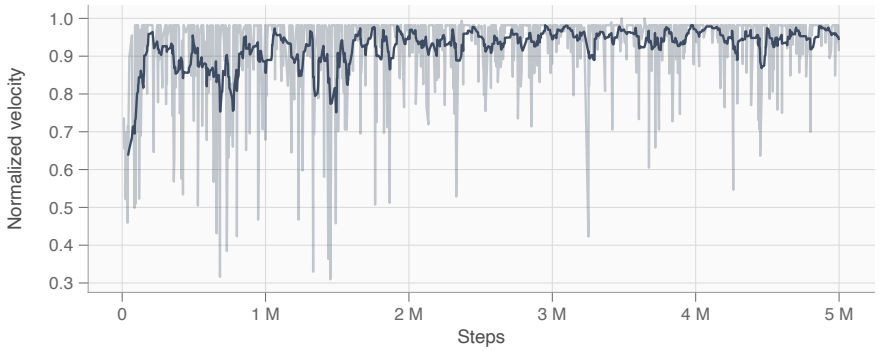
6.2 ACCEPTANCE ASSESSMENT

6.2.1 Participants

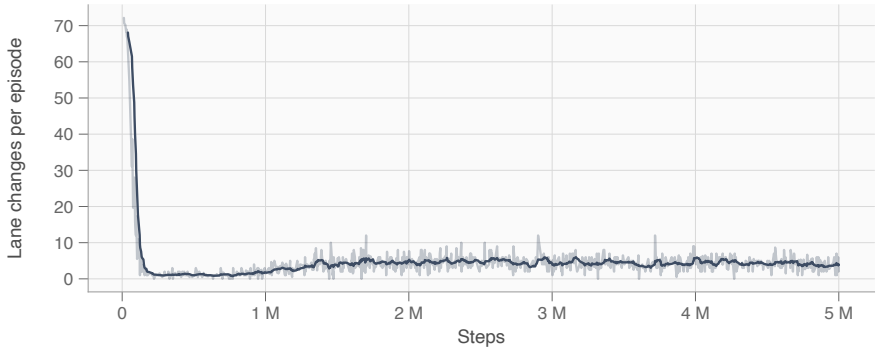
A total of 1373 people participated in this study between 12 November and 21 December 2020. The survey received a satisfaction rating of 3.9 ($n = 118$) on a scale from 1 ('very dissatisfied') to 5 ('very satisfied'). Prior to the data analysis, a strict screening procedure was adopted in order to filter out participants that did not complete the experiment properly. Participants who indicated they did not read the instructions, indicated they were under 18 years of age or who did not fully complete the task were removed. If a person completed the survey more than once from the same IP address, only the first response was kept. If a person cheated the system by filling in the Appen survey



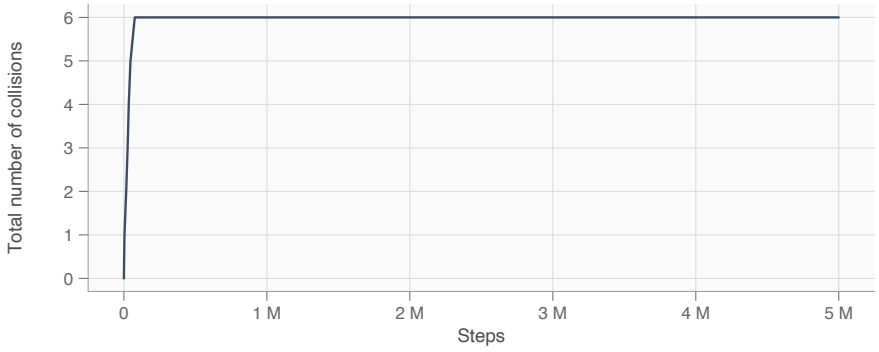
(a) Cumulative reward



(b) Normalized velocity



(c) Lane changes per episode



(d) Total number of collisions

Figure 6.1: Agent evaluation metrics during training. The transparent plots represent the raw data whereas the solid plots are smoothed.

multiple times by reusing the code that was received after completing the task, only the first response was kept. If a person was suspected to suffer from delayed video playback, carried out a different key-pressing task[1] or pressed the response key an anomalous number of times (more than 100 times per video), their key-pressing data for the video in question was discarded, but the questionnaire responses were kept.

In total, 288 (21%) participants were removed due to filtering, leaving $N = 1085$ participants from 69 different countries. On average, there were 31.7 (SD = 0.5) as opposed to 32 responses available per participant, presumably as a result of data loss due to server issues. This means that were was a 26.4% chance that a participant responded to a specific video. Of the total of 34,432 responses across 120 videos, the mean (SD) amount of responses per video was 287 (37.5). A total of 795 entries were removed from the key-pressing data due to delayed video feedback, wrong task execution or excessive pressing. A total of 1122 entries were removed from the key-pressing data as a result too many presses.

The participants had a mean age of 37.3 years (SD = 11.4 years). A total of 666 participants were male, 416 were female and 3 preferred not to respond. On average, the age of obtaining a driver's license was 21.5 years (SD = 4.8 years); 260 respondents provided an invalid or no answer to this question, presumably due to not having a driver's license. The vast majority of respondents used a private vehicle for primary transport ($n = 644$), followed by public transportation ($n = 244$), walking or cycling ($n = 105$), motorcycle ($n = 74$), other ($n = 8$) and 10 provided no response. The average time to complete the study was 61.9 minutes (SD = 17.5 minutes).

The three most represented countries were Venezuela ($n = 478$), United States ($n = 53$) and Russia ($n = 50$). As stated, Venezuela is represented considerably more often than any other country.

6.2.2 Analyses at the individual level

Table 6.2 shows the descriptive statistics (mean and standard deviation) of the number of key responses and four questionnaire scores (performance expectancy, safety, human-likeness and reliability) per participant, separated by model and/or weather condition and aggregated over all videos per condition.

The mean (SD) number of button presses were 3.72 (7.61), 5.09 (9.04), 4.15 (7.90) and 4.69 (8.74) for the NN model, MOBIL model, sunset weather condition and misty weather condition, respectively. Figure 6.2 shows the variation in mean button presses per individual, separated by model. It can be observed that there is a greater variability for MOBIL mean button presses as well as higher outlier values. Furthermore, a small group of participants (indicated by the outliers) pressed the button a high amount of times on average across both models. Follow-up analysis ($n = 1063$) reveals a weak Spearman correlation between age and average amount of button-presses ($r_s = 0.16, p < 0.001$), i.e. older participants pressed a higher amount of times on average. There was no significant correlation ($p > 0.01$) found between gender and number of button presses or the questionnaire items.

The mean (SD) performance expectancy, safety, human-likeness and reliability ratings were 3.87 (0.80), 3.84 (0.80), 3.93 (0.79), 3.87 (0.79) and 4.12 (0.72), 4.05 (0.75), 4.12

1 The data of 22 participants was obtained during a pilot study where the button-pressing task was different.

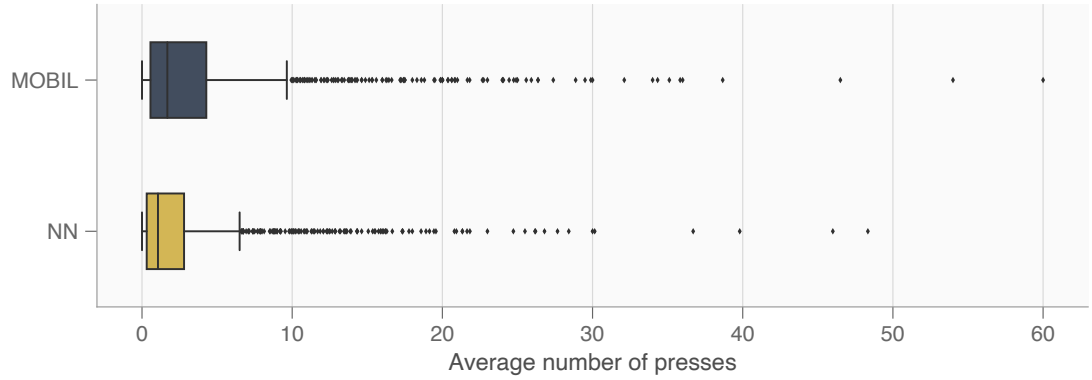


Figure 6.2: Average number of button presses distribution per model. Results are aggregated over all videos per participant ($n = 1063$).

Table 6.2: Acceptance construct ratings and button presses descriptive statistics ($n = 1085$)

Model	**Weather condition**	**PE** Mean (*SD*)	**S** Mean (*SD*)	**HL** Mean (*SD*)	**R** Mean (*SD*)	**nr. presses** Mean (*SD*)
MOBIL	sunset	3.90 (0.82)	3.90 (0.82)	3.96 (0.81)	3.92 (0.80)	5.00 (9.22)
	mist	3.84 (0.86)	3.78 (0.88)	3.90 (0.83)	3.82 (0.85)	5.23 (10.29)
	total	3.87 (0.80)	3.84 (0.80)	3.93 (0.79)	3.87 (0.79)	5.09 (9.04)
NN	sunset	4.20 (0.71)	4.18 (0.74)	4.19 (0.72)	4.19 (0.71)	3.20 (7.25)
	mist	4.03 (0.80)	3.92 (0.87)	4.04 (0.80)	3.98 (0.83)	4.28 (9.55)
	total	4.12 (0.72)	4.05 (0.75)	4.12 (0.72)	4.09 (0.73)	3.72 (7.61)
total	sunset	4.05 (0.71)	4.04 (0.72)	4.07 (0.71)	4.05 (0.70)	4.15 (7.90)
	mist	3.93 (0.77)	3.85 (0.82)	3.97 (0.77)	3.90 (0.79)	4.69 (8.74)
	total	3.99 (0.71)	3.95 (0.74)	4.02 (0.72)	3.98 (0.72)	4.41 (7.89)

Note: scores per condition are aggregated over all videos per participant. PE = performance expectancy, S = safety, HL = human-likeness and R = reliability. PE, S, HL and R response options were 1 = *disagree strongly*, 2 = *disagree a little*, 3 = *neither agree nor disagree*, 4 = *agree a little* and 5 = *agree strongly*.

(0.72), 4.09 (0.73) for the MOBIL and NN model, respectively. If separated by weather condition, mean ratings were higher in the sunny weather condition compared to the misty condition, i.e. the mean (*SD*) scores were 4.05 (0.71), 4.04 (0.72), 4.07 (0.71), 4.05 (0.70) and 3.93 (0.77), 3.85 (0.82), 3.97 (0.77), 3.90 (0.79) for the sunny and misty conditions, respectively. Figure 6.3 shows the total counts and relative proportions of the five Likert scales across all four acceptance constructs, separated by model. Whereas the proportions of the Likert scales "neither agree nor disagree" and "agree" are fairly similar across both models, there is a noticeable difference when it comes to the other scales. NN model videos received a relatively larger proportion of "strongly agree" votes whereas MOBIL model videos received a larger proportion of "strongly disagree" and "disagree" votes, although the differences are not dramatic.

Differences between models and weather conditions

A two-way repeated-measures ANOVA was conducted to examine the effect of lane change model and weather condition on the occurrence of disagreements and the four

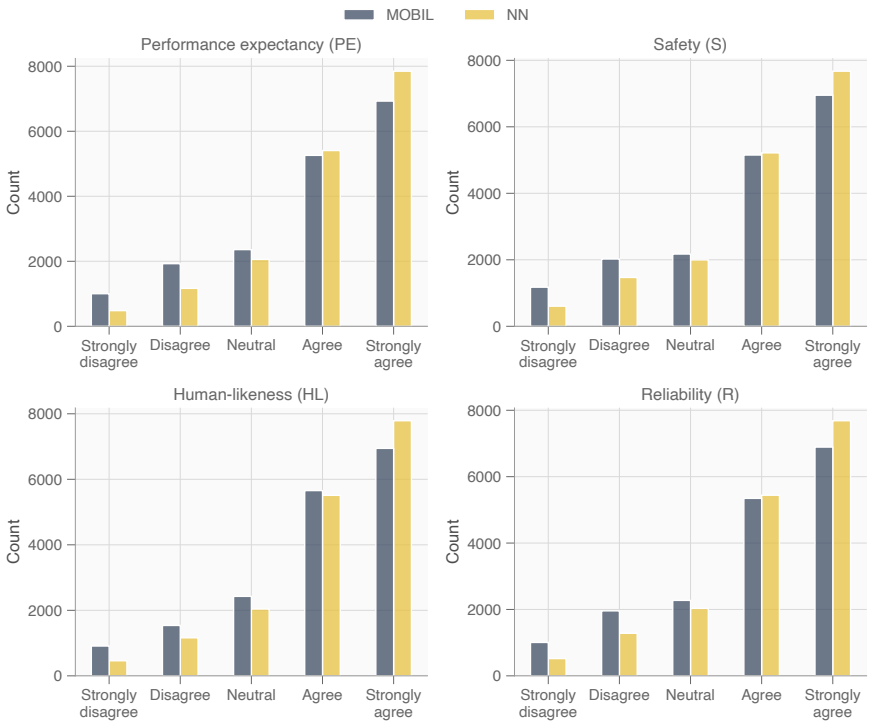


Figure 6.3: Distribution of acceptance construct scores across all responses separated per model ($n = 34432$).

questionnaire scores (see Table 6.3). Results show that the utilized lane change model had a significant ($p < 0.001$) effect on all five metrics (i.e. number of key presses and the four questionnaire scores). We also found a significant effect for weather condition on number of presses ($p = 0.001$) and all four questionnaire scores ($p < 0.001$). There were significant model $\times$ weather interaction effects for the four acceptance constructs performance expectancy, safety, human-likeness and reliability ratings ($p < 0.001$) but not for the number of key presses ($p = 0.02$). Note that the model had the strongest effect on all five variables, followed by weather conditions and then the interaction effect. The effects are also noticeably weaker for the number of key presses compared to the four questionnaire scores. Furthermore, the effects of weather are strongest on safety and reliability ratings, but weaker on performance expectancy and human-likeness.

Post-hoc pairwise comparisons were conducted using paired t-tests with Bonferroni correction. All comparisons can be found in Table 6.4. Results showed that the NN model in the sunset caused significantly fewer disagreements among participants $t(1084) = -8.81, -8.53, -5.30, p < 0.001$ and that it was rated significantly higher ($p < 0.001$) in terms of performance expectancy $t(1084) = 16.63, 18.15, 11.68$; safety $t(1084) = 15.37, 19.13, 14.60$; human-likeness $t(1084) = 13.69, 15.99, 10.81$; and reliability $t(1084) = 15.85,\ 18.46, 13.35$ compared to MOBIL in sunset, MOBIL in mist and NN in mist, respectively. The NN mist condition was rated significantly higher compared to MOBIL in sunset and mist for the majority of metrics as well; only the difference in safety rating ($t(1084) = 0.89, p = 0.37$) and number of key presses ($t(1084) = -2.95, p = 0.003$) with MOBIL in sunset was non-significant. Finally, it was found that the MOBIL model in sunset significantly outperformed ($p < 0.001$) the MOBIL model in mist across all the four acceptance constructs but not in terms of key presses ($t(1084) = -1.14, p = 0.25$).

Table 6.3: Two-way repeated-measured ANOVA results between conditions ($n = 1085$)

Variable	Effect	df	F	p-value
PE	model	(1,1084)	245.69	< 0.001
	weather	(1,1084)	99.45	< 0.001
	model:weather	(1,1084)	32.26	< 0.001
S	model	(1,1084)	192.73	< 0.001
	weather	(1,1084)	189.72	< 0.001
	model:weather	(1,1084)	41.14	< 0.001
HL	model	(1,1084)	165.03	< 0.001
	weather	(1,1084)	93.85	< 0.001
	model:weather	(1,1084)	27.50	< 0.001
R	model	(1,1084)	211.50	< 0.001
	weather	(1,1084)	148.91	< 0.001
	model:weather	(1,1084)	37.70	< 0.001
nr.	model	(1,1084)	54.54	< 0.001
presses	weather	(1,1084)	10.78	0.001
	model:weather	(1,1084)	4.84	0.02

Note: scores per condition are aggregated over all videos per participant. PE = performance expectancy, S = safety, HL = human-likeness and R = reliability.

Table 6.4: Pairwise comparison t-test results between conditions ($n = 1085$)

Pair		PE	S	HL	R	presses
		t	t	t	t	t
MOBIL sunset	MOBIL mist	4.28*	7.58*	4.23*	6.30*	-1.14
	NN sunset	-16.63*	-15.37*	-13.69*	-15.85*	8.81*
	NN mist	-6.44*	-0.89	-4.54*	-3.32*	2.95
MOBIL mist	MOBIL sunset	-4.28*	-7.58*	-4.23*	-6.30*	1.14
	NN sunset	-18.15*	-19.13*	-15.99*	-18.46*	8.53*
	NN mist	-10.37*	-7.87*	-8.30*	-9.01*	3.78*
NN sunset	MOBIL sunset	16.63*	15.37*	13.69*	15.85*	-8.81*
	MOBIL mist	18.15*	19.13*	15.99*	18.46*	-8.53*
	NN mist	11.68*	14.60*	10.81*	13.35*	-5.30*
NN mist	MOBIL sunset	6.44*	0.89	4.54*	3.32*	-2.95
	MOBIL mist	10.37*	7.87*	8.30*	9.01*	-3.78*
	NN sunset	-11.68*	-14.60*	-10.81*	-13.35*	5.30*

Note: scores per condition are aggregated over all videos per participant. PE = performance expectancy, S = safety, HL = human-likeness and R = reliability.
* = significant at $p < 0.00125$. All significance levels are adjusted using a Bonferroni correction.

Correlations among experiment metrics

A Pearson correlation analysis was conducted to investigate possible relationships between questionnaire construct scores and disagreement at the participant level (all 120 videos aggregated). A visualization of the correlation matrix can be seen in Figure 6.4.

It was found that number of key presses was significantly (albeit weakly) correlated with all four acceptance constructs at the level of participants ($n = 1063$)[2], i.e. $r_p = -0.11, p < 0.001$ for performance expectancy, $r_p = -0.11, p < 0.001$ for safety, $r_p = -0.10, p = 0.001$ for human-likeness and $r_p = -0.11, p < 0.001$ for reliability.

Substantial correlations were found between questionnaire constructs. Performance expectancy scores correlated significantly with safety $r_p = 0.95, p < 0.001$, human-likeness $r_p = 0.88, p < 0.001$ and reliability $r_p = 0.97, p < 0.001$. Safety scores correlated significantly with human-likeness $r_p = 0.84, p < 0.001$ and reliability $r_p = 0.97, p < 0.001$. Finally, we found a significant correlation between human-likeness and reliability $r_p = 0.87, p < 0.001$.

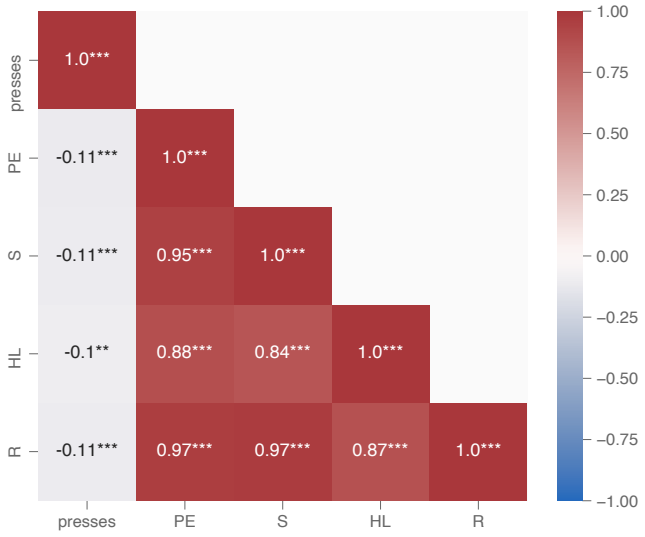


Figure 6.4: Pearson correlations between questionnaire items and number of button-presses on the individual level ($n = 1063$). * significant at $p < 0.01$, ** significant at $p < 0.005$, *** significant at $p < 0.001$.

6.2.3 Analysis at the video level

Mean questionnaire item scores and number of key presses were calculated on the video level, i.e. by aggregating over all participants per video. In terms of mean (*SD*) key presses, NN videos received 3.18 (1.06) whereas MOBIL got 4.36 (1.62). Figure 6.5 highlights the average number of key presses, separated by model. The overall smaller spread suggests that the NN model performed more consistently across its video subset

2 As described in Section 6.2.1, some participants' key-pressing data was filtered due to internet connection issues, originating from a pilot study or excessive pressing.

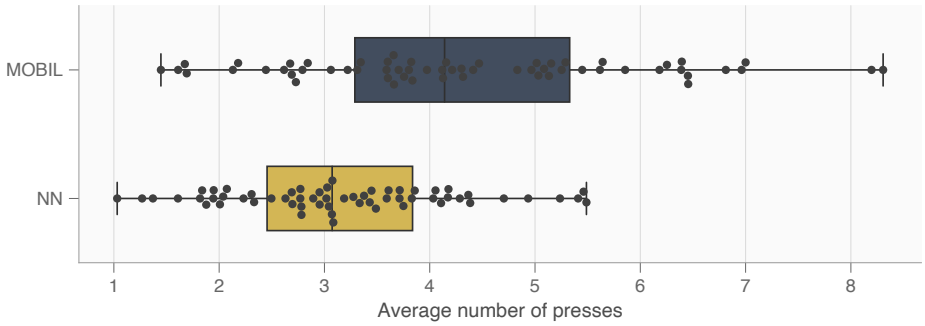


Figure 6.5: Average number of button presses distribution per video, separated by model ($n = 120$). Results are aggregated over all participants per video. The grey dots indicate the raw data points.

compared to the MOBIL model. The NN model received mean (SD) questionnaire scores of 4.14 (0.25) on performance expectancy, 4.07 (0.28) on safety, 4.14 (0.22) on human-likeness and 4.11 (0.26) on reliability whereas MOBIL received 3.91 (0.31), 3.89 (0.37), 3.97 (0.25) and 3.91 (0.35), respectively.

The results of the Pearson correlation analysis on the video level ($n = 120$) can be found in Figure 6.6. Similarly to the correlation on the participant-level, there were substantial positive correlations between the four questionnaire items ($r_p > 0.97, p < 0.001$). Moreover, considerable correlations existed between number of presses and performance expectancy ($r_p = -0.85, p < 0.001$), safety ($r_p = -0.82, p < 0.001$), human-likeness ($r_p = -0.83, p < 0.001$) and reliability ($r_p = -0.83, p < 0.001$).

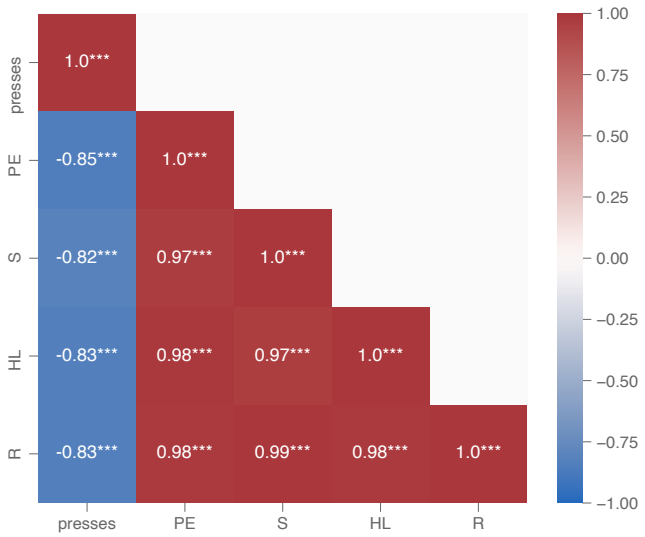


Figure 6.6: Pearson correlations between questionnaire items and number of button-presses on the video level ($n = 120$). * significant at $p < 0.01$, ** significant at $p < 0.005$, *** significant at $p < 0.001$.

6.2.4 In-depth analysis of button-pressing behavior

The mean number of cumulative button presses were calculated for all 120 videos. We repeat that all participants viewed a random subset of 32 videos, so each video was viewed by a unique sample of participants. The mean cumulative button-presses as a function of time for the 9 most-pressed and 9-least pressed videos for both models (each represented by 60 videos) can be seen in Figures 6.7-6.8. The same plots for all videos in descending order can be found in Appendix B. Judging from the sorted ranking, it can be observed that the videos with high amounts of button presses generally featured MOBIL. The translucent error bands and grey areas represent the 95% confidence intervals and parts where the agent performed a lane change, respectively. The lane change of each direction is represented by the letters 'L' or 'R' at the bottom. The confidence intervals were calculated using percentile bootstrap resampling.

It can be seen that for both the most-pressed and least-pressed videos, lane changing decisions generally had a substantial effect on the participants' propensity to press the button. To illustrate, the plots show larger increases in mean cumulative button presses during lane changes or in time segments immediately followed by lane changes (which can presumably be explained by delayed reaction) for both models as compared to most other parts where the agent is lane keeping. Furthermore, the lane keeping segments are characterized by either steady (albeit less considerable) button-pressing (e.g. videos 42 or 107) or almost no button pressing (e.g. videos 11 or 91). Some videos exhibit a larger degree of within-subjects variation than other as indicated by the varying confidence intervals.

With Appendix B as a reference, video segments with relative sharp increases in cumulative presses were visually analyzed with the intention to investigate why participants pressed the response key. To be brief, the explanations per model are summarized below.

MOBIL

- There were multiple cases where the vehicle engages in multiple lane changes within a short time span, only to return to that same lane some seconds later. To illustrate, in the corresponding time plots of videos 21, 22, 30, 54, 60 and 91 in Appendix B, we can observe this phenomenon as indicated by the lane change segments in opposite directions with relatively steep increases in cumulative presses.
- There was a considerable amount of cases where a slow vehicle was driving in front of the vehicle followed by a lane change to the right as opposed to overtaking from the left. From there, the vehicle would either (1) end up 'stuck' behind slower traffic, sometimes without being able to go left; or (2) overtake the slower vehicle from the right-hand side. Notable examples of where this occurred are videos 11, 13, 27, 30, 41, 47, 48 and 60. For reference, see the corresponding plots in Appendix B, where sharp increases in average presses can be observed during and following the period after the right lane change.
- Lane changes in the right direction sometimes cut off approaching vehicles. A more in-depth analysis regarding this problem follows later in this section.
- In some cases the agent was hesitant and waited before committing to an overtake.

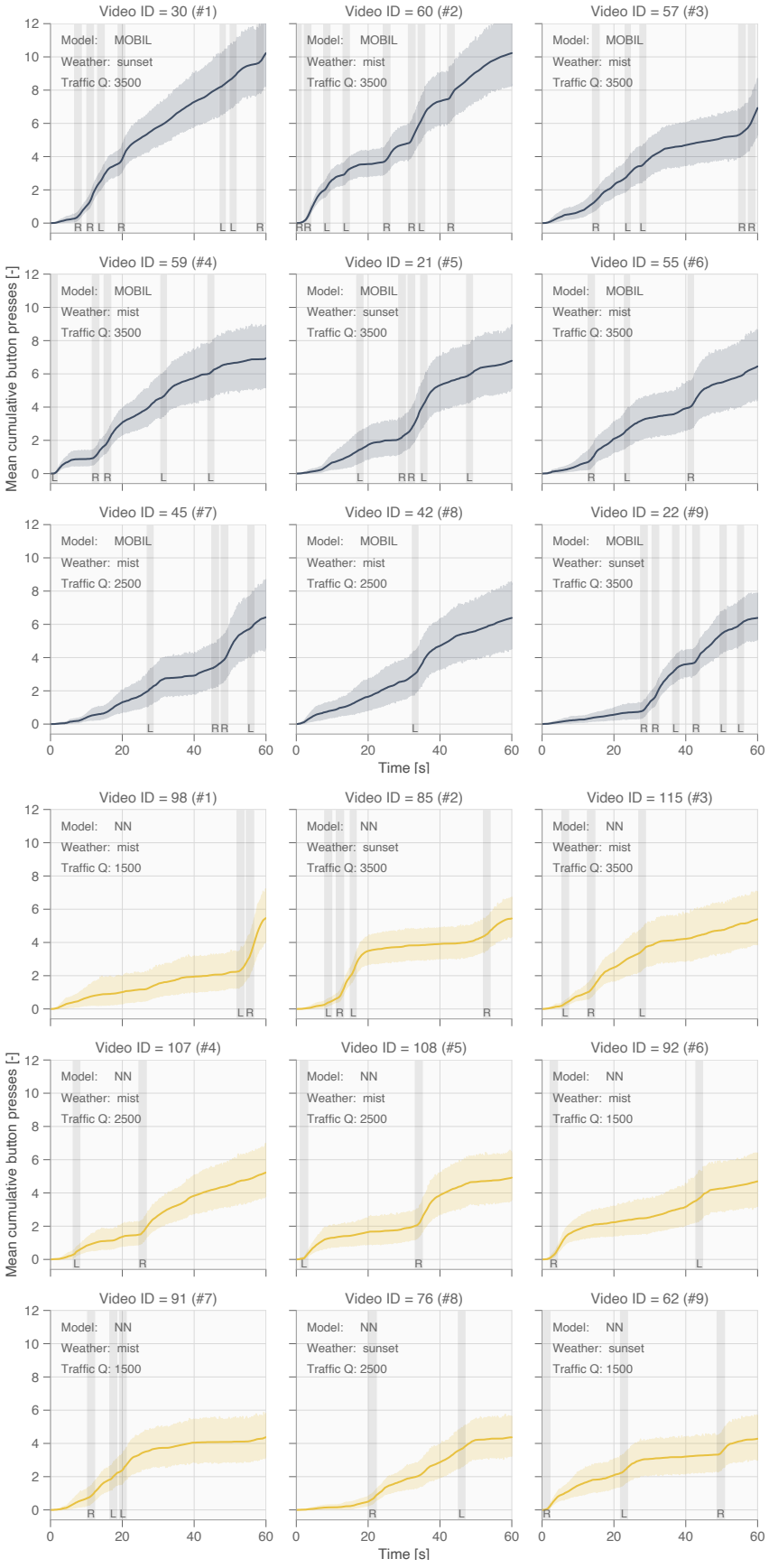


Figure 6.7: Presses (mean, 95% CI) as a function of time for the 9 most-pressed videos per model.

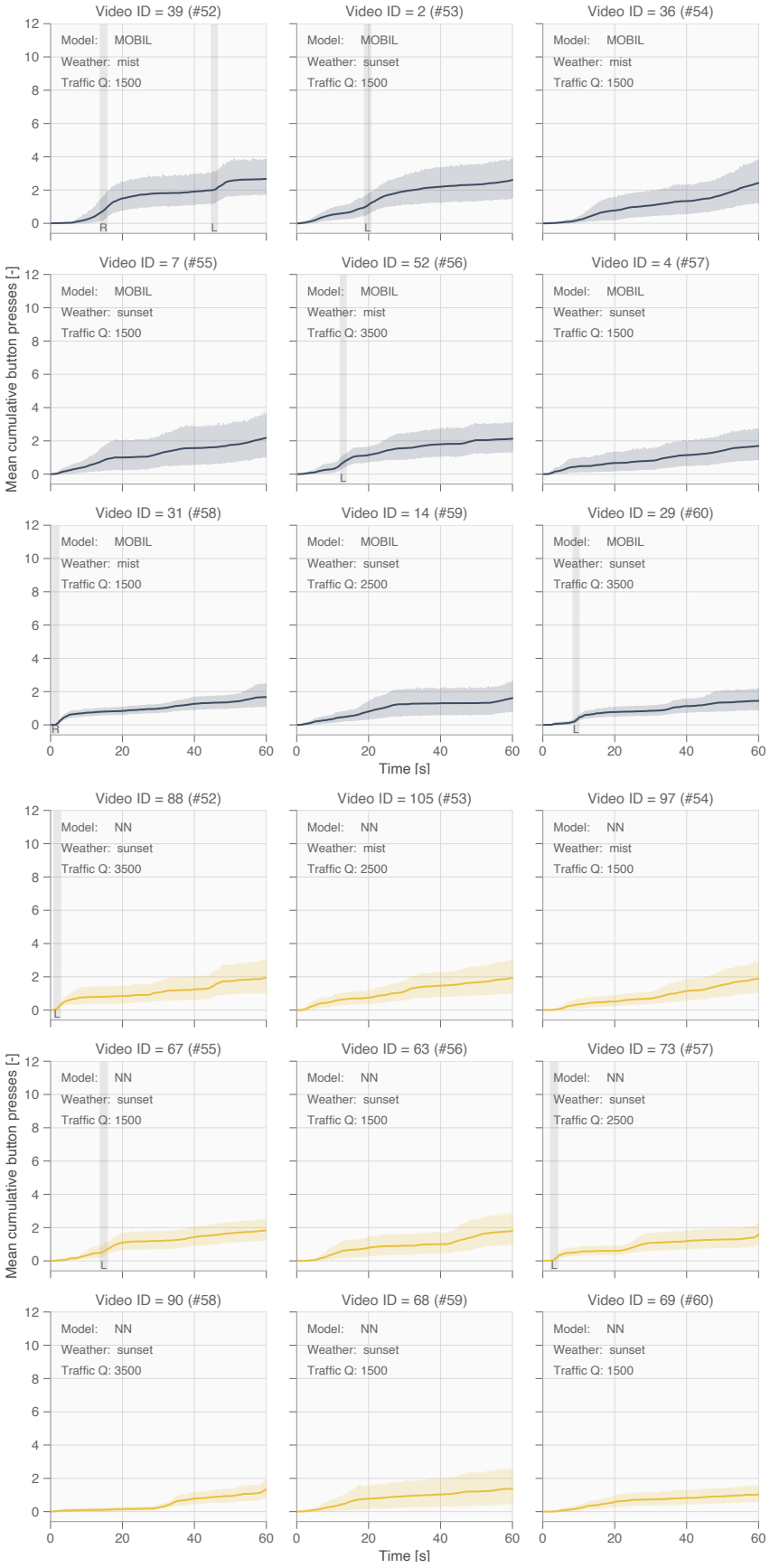


Figure 6.8: Presses (mean, 95% CI) as a function of time for the 9 least-pressed videos per model.

Neural network (NN)

- Similarly to MOBIL, there were also some cases where the NN agent executed a lane change only to go back shortly after. It should be noted that this happened to a considerably smaller extent, namely in videos 85 and 98. Judging from the sorted plots in Appendix B, these videos featured the highest press rates of all NN videos.
- Some participants did not agree with the vehicle changing back to the right-hand lane(s), despite having enough space. Notable examples are videos 84 and 108.

Possible in-depth reasons for these phenomena are explored in Chapter 7.

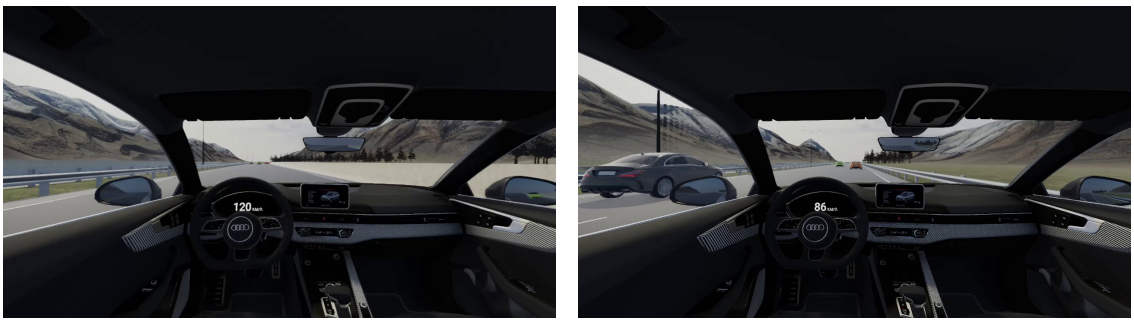


Figure 6.9: Video snapshots of a case where the MOBIL agent lane changed to the right and ended up following slow traffic in lane 3.

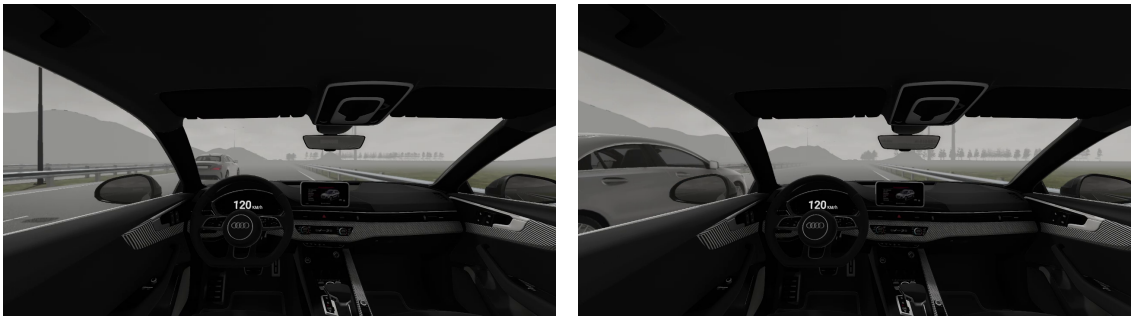


Figure 6.10: Video snapshots of a case where the MOBIL agent engaged in right-sided overtaking.

6.2.5 Analysis of lane changes

All lane changes were categorized in terms of 'urgency' and 'severity' according to the method used by Lee et al. [95]. They used TTC to classify the urgency of lane-changes on a 4-point scale, indicating how soon the lane change was needed, i.e. 1 = non-urgent (TTC $>$ 5.5 s), 2 = urgent (5.5 s $\geq$ TTC $>$ 3 s), 3 = forced (TTC $\leq$ 3 s), and 4 = critical (physical contact occurred in the same lane or emergency maneuvers were required). Severity was classified based on the presence of vehicles in the so-called proximity zone (time-gap $\leq$ 0.3 s behind and space-gap 1.2 m in-front of the agent) and the time to reach the rear-end of the proximity zone T_r if a vehicle is in the fast approach zone (0.3 s $<$ time-gap $\leq$ 1.6 s), rated on a 7-point scale, i.e. 1 (vehicle is in FAZ where $T_r >$ 5.0 s or no vehicle present), 2 (3.0 s $< T_r \leq$ 5.0 s), 3 (1.0 s $< T_r \leq$ 3.0 s), 4 ($T_r \leq$ 1.0 s), 5 (vehicle is in PZ), 6 (emergency maneuvers occurred) and 7 (physical contact occurred). Note that both these zones refer to areas in the target lane, where points 1-5 are measured at the moment the lane change maneuver is initiated. Whereas urgency describes the

traffic situation in the agent's current lane, severity was used indicate whether a vehicle in the target lane was cut off or tailgated. In addition to the TTC value that was used for determining urgency, which is calculated during lane change initiation, the minimum TTC throughout the entire maneuver (for both front and rear vehicles) was calculated in order to determine the level of danger. This 'danger' value was calculated by taking into account the relative space-gaps and velocities in the longitudinal direction only. These values classified using the same 4-point scale as with the urgency rating.

Of the total 234 lane changes made in all 120 videos, 132 (56%) were made by MOBIL and 102 (44%) by the NN model. Table 6.5 presents the frequency distributions for urgency, severity and minimum TTC respectively, for both lane change models. Lane changes performed by both the MOBIL- and NN agent were most commonly rated non-urgent (98% and 100%, respectively), level 1 severe (70% and 95%, respectively) and level 1 dangerous (77% and 95%, respectively). There were no lane changes rated 3 (forced) or higher and 6 severe or higher. On average, MOBIL performed lane changes that were rated higher on urgency (1.022 vs 1.000) , severity (2.21 vs 1.12) and danger (1.23 vs 1.05) compared to the NN agent. There was a notably high amount of lane changes rated level 5 severe performed by MOBIL, indicating cases of possible tailgating or cut-offs.

In an attempt to relate button pressing behavior during lane changes with traffic data, the percentage of participants who pressed the button at least during a lane change once were calculated for all lane changes. Even though it is realistically possible that participants pressed the button after the lane change segment, the time window is still restricted to the lane change segment itself due to the occurrence of some successive lane changes in short time windows. As shown in Figure 6.11, there are some data points which were classified with high levels of danger or severity that had a high percentage of presses. A Spearman correlation analysis was performed on the lane change data per model. It was found that for MOBIL ($n = 132$), relative button-pressing percentages significantly correlated with severity $r_s = 0.38, p < 0.001$ whereas no significant correlations were found with level of urgency $r_s = 0.04, p = 0.67$ or level of danger $r_s = -0.12, p = 0.18$. For the NN model ($n = 102$), no significant correlations were found between button-pressing percentages and severity $r_s = 0.01, p = 0.92$ or danger $r_s = 0.06, p = 0.53$ (no Spearman correlations were computed with urgency as all values were constant).

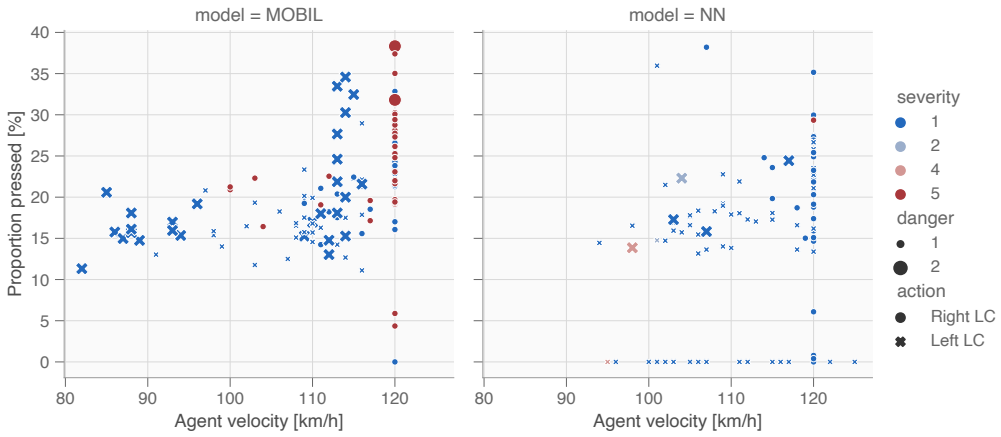


Figure 6.11: Relationship between button-pressing and traffic data.

Table 6.5: Lane change urgency, severity and minimum TTC frequency distributions.

	MOBIL		NN	
	Frequency	Percentage	Frequency	Percentage
Urgency				
1 (non-urgent)	129	98%	102	100%
2 (urgent)	3	2%	0	0%
3 (forced)	0	0%	0	0%
4 (critical)	0	0%	0	0%
Severity				
1	92	70%	97	95%
2	0	0%	2	2%
3	0	0%	0	0%
4	0	0%	2	2%
5	40	30%	1	1%
6	0	0%	0	0%
7	0	0%	0	0%
Danger (minimum TTC)				
1 (non-urgent)	102	77%	97	95%
2 (urgent)	30	23%	5	5%
3 (forced)	0	0%	0	0%
4 (critical)	0	0%	0	0%

7 DISCUSSION

7.1 MAIN FINDINGS

This study used deep reinforcement learning (DRL) to develop a automated lane changing model in Unity and measured the participants' acceptance through a crowdsourcing experiment. In our method, participants were required to watch a sequence of videos from a first person perspective and press a response key whenever a moment of disagreement occurred, after which they rated the video in question using a four item-long questionnaire, allowing us to examine the technology acceptance of various lane changing models in varying weather conditions. Overall, agent training was successful and the model's behavior was rated relatively high in terms of efficiency ($M = 4.12$), safety ($M = 4.05$), human-likeness ($M = 4.12$) and reliability ($M = 4.09$) on a scale from 1 to 5 while simultaneously causing varying levels of disagreements ($M = 3.72$, $SD = 7.61$) among participants. Furthermore, it was rated higher on average across all four acceptance constructs while simultaneously causing fewer disagreements compared to the baseline MOBIL model in both weather conditions, although the effects were moderate.

7.2 AGENT TRAINING AND INFERENCE

Training and inference results show that Proximal Policy Optimization (PPO) can effectively be used to yield lane changing behavior on a 3-lane highway that is reasonable, safe and consistent without any collisions with surrounding traffic. The discrete action space of the agent allowed for a hierarchical approach in which the trained neural network was only responsible for high-level decision-making whereas low-level car-control and maneuver execution was executed by other, proven control methods. This allows for flexible behavioral parameter tuning, such as changing to a different car-following model or trajectory generation model for example. In addition, we used a discrete action mask to guide exploration and accelerate the training process.

The inference results showed that the NN model was able to maintain a higher speed on average with fewer lane changes compared to MOBIL in all three traffic densities. Moreover, the variance of both metrics was lower, indicating a more consistent performance. This is in accordance with the notion that DRL outperforms MOBIL, which has been documented in other works before [33, 34]. Performance seemed to decrease as the traffic density increased for both models. This was also to be expected, as more surrounding traffic increases the probabilities that a slower car impedes the agent while also obstructing possible overtakes, which would subsequently result in a lower average velocity and the need for more lane changes.

7.3 BUTTON-PRESSING

According to the repeated-measures ANOVA and paired *t*-tests, there were significantly fewer disagreements with the driving behavior of the proposed NN model compared to the MOBIL model. Descriptive statistics showed a higher variation in individual pressing behavior regarding MOBIL with respect to the NN. Furthermore, a small group of participants exhibited a high amount of disagreements with regards to both models. On the video-level, we can observe fewer presses on average with a smaller spread as well, suggesting that NN model was more consistent in behavior across its video subset.

7.3.1 Individual differences

We assume that the individual differences in disagreement can be explained by at least three human factors: differences in driving style, attitude and experience with AVs.

According to Taubman-Ben-Ari et al. [96], there is exist multiple driving styles that are classified through multiple facets or scales, such as reckless driving, patient and careful driving or angry and hostile driving. Considering the fact that drivers prefer an automated driving style that reflects their own [65], button-pressing could in part be influenced by the participants trying to impose their own behavioral intent. Indeed, Zhang et al. [97] found substantial differences in velocity, trajectory and initiation timing between the lane changes of an aggressive driver versus a conservative driver. Inter-driver differences when car-following have also been empirically documented using naturalistic driving data [98, 99].

Attitude is another pre-defining factor, defined as the psychological tendency that describes the degree to which an individual likes and dislikes a particular entity [100]. It has often raised particular attention from researchers and has been certified to have a strong effect on AV acceptance [101–104]. Kyriakidis et al. [104] conducted an international study among 5000 respondents and found a high spread in responses to fully automated AVs, stating that "some people were clearly against automated driving while others would enjoy it." Given the documented prevalence of large variety in attitudes towards AVs, the possibility that attitude influenced individual rating behavior should be considered.

The influence of AV experience on acceptance has been investigated by various authors. Castritius et al. [45] conducted interviews and questionnaires before and after experiencing a drive in a highly automated vehicle, stating that before the experience "concerns predominated among drivers", subsequently finding "a clear increase in acceptance after experience with the system in real traffic" and that the concerns did not materialize. Eden et al. [105] found a similar effect regarding safety concerns in a study concerning an automated shuttle. Dikmen and Burns [106] also found a strong connection between experience and attitude towards AVs. They investigated the experiences of 162 Tesla Model S drivers who possessed significant experience in using the highly automated features Autopilot and Summon (both highly automated systems), stating that these technologies "were not considered to be particularly risky" in spite of the high frequencies of automation failures. It should be noted that their sample consisted of early adopters of technology and were by no means representing the general population. As AV technology is still relatively new in the automotive industry and our daily lives, we

cannot rule out that individual differences in rating behavior are affected by varying degrees of AV experience within the participant sample.

7.3.2 Button-pressing behavior

Multiple observations can be made regarding button-pressing behavior in the time dimension. First and foremost, the occurrence of lane changes generally had a strong effect on cumulative button presses as the time plots indicated relatively sharp increases in average responses during and around lane change events for both models. Lane keeping segments were characterized by either steady increases in average presses (albeit to a lesser degree compared to lane change events) or little to no key responses. Moreover, videos with high lane change frequencies also featured higher traffic densities, which is consistent with the agent inference trial results. Videos that contained the highest amount of button presses generally featured more lane changes compared to videos with fewer button presses.

In-depth (visual) analysis revealed several behavioral characteristics that can ultimately be attributed to the fundamental principles of each model. In case of MOBIL, the model exhibited various cases of frequent (inefficient) lane changing in short time-spans and lane changes that resulted in right-handed overtakes or the agent ending up behind another slow vehicle. We believe this behavior highlights the inherent weakness of a rule-based approach like MOBIL. To illustrate, MOBIL enforces a keep-right policy by changing right as soon as safety- and incentive criteria are met regardless of the presence of other traffic in the vicinity. Unless the criteria to change left (which would lead to the correct choice) were met earlier, this keep-right policy led to unfavourable choices where the agent ended up behind other slow traffic and having to change back again or committing a right-sided overtake in case there was enough space on the right. Note that there exists no build-in feature that prevents such short-seeing behavior, despite Kesting et al. [6] arguing that the model "reflects realistically far-seeing and anticipative driving behavior." In other cases, the agent cut-off upcoming traffic, exclusively when performing right-hand lane changes. This can be attributed to the fact that MOBIL considers the expected braking acceleration of the follower in the same lane instead of the right lane. One can make the case that parameter values were not restrictive enough. For example, increasing the safety criterion threshold could technically prevent cut-offs. However, the used values in this work are already considerably stricter compared to the default values in Kesting et al. [6]. Besides, more restrictive parameters would also result in less decisive behavior in other situations.

In contrast, the NN model demonstrated adaptive behavior, only changing lanes if it was strategically profitable in the long-term. Although inefficient lane changing also occurred for the DRL agent, it happened to a considerably lesser extent. To summarize, MOBIL inherently lacks the strategic capabilities and dynamic parameter tuning that fits the situation, whereas DRL learned optimal behavior for each state through extensive trial-and-error.

7.3.3 Button-pressing during lane changes

Analysis of the urgency, severity and minimum TTC of all lane changes showed that the majority was classified as non-urgent and non-severe. There was a relatively large

proportion of level 5 severe lane changes (30%) performed by MOBIL. However, as could be seen in Figure 6.11, even though the level 5 severe rated lane changes generally led to a moderate proportion (20% or more) of participants pressing the button at least once, almost all lane changes were classified as non dangerous with a minimum TTC of 5.5 seconds or higher. Furthermore, it was found that all of these lane changes were from left to right and almost all occurred while driving 120 km/h, which is consistent with the findings from the visual analysis. Note that even though high levels of severity, danger or urgency could justify high levels of disagreement (as demonstrated by high proportions of presses for some data points), it did not guarantee that a large proportion of participants would press the button nor was it the only reason as there were multiple cases with low severity or levels of danger that resulted in high pressing rates. We did find significant correlations between severity and pressing percentages for MOBIL.

7.4 ACCEPTANCE CONSTRUCT RATINGS

We found that respondents considered the resulting driving behavior of both the NN and MOBIL model efficient, safe, human-like and reliable in both weather conditions. Out of the four questionnaire item scores, the NN model was rated highest on efficiency and human-likeness, followed by reliability and safety whereas the MOBIL model was rated highest on human-likeness, followed by reliability, efficiency and then safety. Even though the differences between the mean questionnaire scores were marginal, safety ratings were the lowest for both models. Indeed, safety has been a major point of concern in other AV acceptance studies as well [43, 45].

Repeated-measures ANOVA and paired t-tests yielded statistical evidence that the decision-making behavior of our proposed model was rated significantly higher compared to MOBIL in both weather conditions. Despite the novelty of our testing method, we argue that these findings support the general hypothesis that ML methods such as DRL are more suitable for controlling driving tasks than more traditional, rule-based methods like MOBIL, which could explain the increasing popularity of ML methods among researchers in general. Studies by Alizadeh et al. [33] and Hoel et al. [34] already conducted the same comparison and in both cases found the DRL model to be superior to MOBIL. Even though their simulator environments, agent designs and testing methods are different to what is used in this study, their results support the above-mentioned hypothesis.

Differences in construct ratings between weather conditions were also found to be significant, albeit with a considerably smaller effect compared to the difference in models. Weather condition effects were strongest on safety and reliability ratings, but weaker on performance expectancy and human-likeness. This is expected, as bad visibility could pose issues from a safety or trust standpoint while leaving performance unaffected. Moreover, one should not underestimate the effects of poor visibility and the effect it has on driver behavior. Studies reported that weather condition affects driver speed choice, headway and overtaking frequency [107] and that reduced visibility (due to fog) leads to a decrease in self-reported comfort when the headway remained unchanged [108]. As both models do not take into account external conditions such as weather and/or visibility in their behavior, one can argue that adaptions are needed on the behavioral end in order for these models to be robust in different weather conditions.

7.5 CORRELATION ANALYSIS

It was found that the four questionnaire items scores (performance expectancy, safety, human-likeness and reliability) correlated strongly ($r_p > 0.8$) with each other. Reliability, a sub-item of trust, was found to correlate strongly with performance expectancy in other studies [43, 59]. Xu et al. [43] also found strong correlations between safety and performance expectancy and between safety and trust. Note that positive correlations between construct ratings could likely also be explained by a common cause (i.e. participants giving similar ratings due to general acceptance towards the model or AVs in general), or methodological factors (i.e. items with identical response options that cluster on the same component). After all, survey items are known to be statistically unreliable [109]. Another possibility is the effect of national differences which has been documented on other works [104], although this effect was not examined since the cross-national differences fall outside of the research scope.

There was a weak (positive) correlation between the propensity to disagree and age, but not between disagreement and gender. Some literature exists that empirically documented the difference in preferred automated driving style between age groups and gender [110], but this effect was not captured in this study. Furthermore, the correlation coefficients could have been attenuated due to the sample consisting mainly of young people (median age = 36). It is also possible that disagreement increased with age because older participants took the task more seriously. In a crowdsourcing study featuring a similar button-pressing task, [111] could not rule out the possibility that certain skewed demographic factors (nationality, age and gender) influenced the differences in button-pressing rates between participant groups.

Moreover, the (negative) correlations between number of presses and the acceptance constructs were significant but weak (< 0.3) on the individual-level, possibly suggesting that disagreement wouldn't necessarily lead to non-acceptance for all individuals. However, substantial (negative) correlations (> 0.8) were found between number of presses and the acceptance constructs on the video-level, which suggests that number of presses could provide an indication for (general) acceptance. This is not unfeasible, since humans prefer AVs that drive similarly to their own driving style [65].

7.6 STUDY STRENGTHS AND LIMITATIONS

A limitation of our simulator environment (and therefore our study) are several simplifications with respect to traffic realism. Firstly, the surrounding traffic in our simulator did not engage in lane changing behavior and therefore could be considered unrealistic. Secondly, a static trajectory generation module was used, with fixed start and end points as soon as the lane change is initiated, not allowing adaptions in case a dangerous situation arises during a lane change maneuver, prompting the need for dynamic modules [85, 112]. Lastly, the highway environment did not feature curved roads like real highways.

In the context of crowdsourcing research and its apparent advantage of enabling the recruitment of a diverse sample on an international scale, the participants' demographic data suggested that the respondents were mainly young people with a relatively large group originating from Venezuela. It is possible that the overall findings may be

influenced by these sociodemographic factors. To elaborate, vehicle owners with the highest interest in AVs are between the ages of 18 and 37 [104]. Still, a large sample was obtained, contributing to the validity of our findings. It is important to note that these results should be interpreted carefully, as the experiment was conducted online with no supervision from the researchers, i.e. some unfiltered data could have come from participants that did not take the task seriously. In summary, what crowdsourcing experiments lack in environmental control it makes up for in participant recruitment, sample diversity and data gathering efficiency and one can argue it is a powerful tool to explore certain hypotheses [113, 114].

The nature of the survey task presents several methodological issues as well. It is unknown whether key presses occurred due to actual disagreement with the model's behavior or methodological factors such as task engagement. Expectancy is another factor: people may feel pressed to find conflicts and be extra critical. Moreover, the threshold for pressing may differ between participants, where some may press only in case of considerable conflicts whereas others press at the slightest difference in preference. One can not rule out language barriers either, further jeopardizing the validity of our results. Another shortcoming is the arbitrary nature of our exclusion criteria for situations of abuse. Even though others advocate not to exclude participants [114], some extreme cases with abnormally high amounts of button presses could not be ignored.

Lastly, more clarifications are needed regarding the causal determinants for button-pressing behavior (both in general in specifically during lane changes) and the observed individual differences. Although some hypotheses are provided regarding the causes of button presses (unnecessary lane changing, safety issues), we believe that disagreement or feelings of conflict in this experiment could be the result of a combination of different factors and that this reflects the multidimensional highly complex nature of AV acceptance. Regarding the observed difference between individual responses, even though ample evidence is provided that could theoretically explain the reported individual differences, it is speculative in nature and further research is required to understand the factors underlying this divergence in responses.

7.7 FUTURE RESEARCH RECOMMENDATIONS

Judging from our findings, it can be argued that DRL proves to be a sufficient method for developing a lane change model for use in AVs in the context of the four used acceptance constructs. Even though, as stated before, some observations were made regarding the causes of disagreements with the decision-making, further research should be conducted in this area. Furthermore, the relationship between decision-making disagreements and the acceptance constructs and to which extent disagreements pose an issue towards acceptance should be further explored. After all, one can argue that the behavior does not have to be completely identical to that of humans [65].

Considering the individual differences, one of the most important areas of improvement would be parameter individualization, i.e. personalization of the models behavior according to the user's preference. Further (qualitative) research would be needed to clarify the needs of different human drivers and passengers in the context of automated lane changing on highways, similar to the mixed-methods approach that was

conducted in [45]. Reinforcement learning can be combined with imitation learning or supervised learning in order to harness the robustness of reinforcement learning while enabling personalization. In this case, the agent would train in a parallel fashion through self-exploration while at the same time attempting to copy the behavior of a human demonstrator. This combined approach would result in a model that exhibits both machine- and human-like behavior which is recommended in other literature [67]. Reinforcement learning has already been combined with imitation learning in other studies [115]. Note that the hierarchical approach in our method allows for personalization in other control levels as well (e.g. more comfortable driving or personalized driving trajectories on the operational level).

7.8 CONCLUSION

This study successfully trained an agent to perform automated lane changes using deep reinforcement learning (PPO), replicating the results that were found in preceding literature. We showed that our proposed model was able to maintain a higher velocity on average at the cost of fewer lane changes. In addition, we found new contributing evidence that an automated lane change model based on NN was rated higher than the rule-based MOBIL model in a user-acceptance context. To illustrate, our proposed NN model received higher ratings for performance expectancy, safety, human-likeness and reliability while simultaneously provoking fewer instances of disagreement. Weather condition (i.e. visibility) had a weaker effect which mainly influenced safety and reliability ratings. Moreover, there were substantial individual differences in rating behavior.

We believe these findings have important implications for researchers and car manufacturers that wish to design highly automated lane change decision-making models that replace the human driver. Firstly, it suggests that DRL is a feasible method for developing a lane change model in the context of acceptance. However, even though the system may be to the liking of human occupants, there are still bound to be varying degrees of disagreement and that more research is needed to uncover the reasoning behind this mechanism. Moreover, the observed individual differences serve as a reminder that a 'one size fits all' approach may not work in the context of AV algorithms, urging the need for personalized behavioral driving models. Finally, more adaptive capabilities are needed in order to make these models robust in varying weather conditions, especially in terms of perceived safety.

BIBLIOGRAPHY

[1] Jonas Meyer, Henrik Becker, Patrick M. Bösch, and Kay W. Axhausen. Autonomous vehicles: The next jump in accessibilities? *Research in Transportation Economics*, 62: 80–91, June 2017. ISSN 0739-8859. doi:10.1016/j.retrec.2017.03.005.

[2] Francesco Biondi, Ignacio Alvarez, and Kyeong-Ah Jeong. Human–Vehicle Cooperation in Automated Driving: A Multidisciplinary Review and Appraisal. *International Journal of Human–Computer Interaction*, 35(11):932–946, July 2019. ISSN 1044-7318. doi:10.1080/10447318.2018.1561792.

[3] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and Decision-Making for Autonomous Vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210, 2018. doi:10.1146/annurev-control-060117-105157.

[4] Guo Feng, Brian Wotring, and Jonathan Antin. Evaluation of Lane Change Collision Avoidance Systems Using the National Advanced Driving Simulator. Technical Report DOT HS 811 332, National Highway Traffic Safety Administration, Washington DC, May 2010. URL https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/analyses20of20rear-end20crashes20and20near-crashes20dot20hs2081020846.pdf.

[5] Peter G. Gipps. A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological*, 20(5):403–414, October 1986. ISSN 0191-2615. doi:10.1016/0191-2615(86)90012-3.

[6] Arne Kesting, Martin Treiber, and Dirk Helbing. General Lane-Changing Model MOBIL for Car-Following Models. *Transportation Research Record*, 1999:86–94, January 2007. doi:10.3141/1999-10.

[7] Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints. *IEEE Transactions on Vehicular Technology*, 66(2):952–964, February 2017. ISSN 1939-9359. doi:10.1109/TVT.2016.2555853.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015. doi:10.1038/nature14236.

[9] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017. ISSN 1476-4687. doi:10.1038/nature24270. URL https://www.nature.com/articles/nature24270.

[10] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364, May 2017. doi:10.1109/ICRA.2017.7989381.

[11] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3389–3396, May 2017. doi:10.1109/ICRA.2017.7989385.

[12] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, S. M. Ali Eslami, Martin Riedmiller, and David Silver. Emergence of Locomotion Behaviours in Rich Environments. *arXiv:1707.02286 [cs]*, July 2017.

[13] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *arXiv:1604.07316 [cs]*, April 2016. URL http://arxiv.org/abs/1604.07316.

[14] Pin Wang, Ching-Yao Chan, and Arnaud de La Fortelle. A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, April 2018. doi:10.1109/IVS.2018.8500556.

[15] Meha Kaushik, Vignesh Prasad, K Madhava Krishna, and Balaraman Ravindran. Overtaking Maneuvers in Simulated Highway Driving using Deep Reinforcement Learning. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1885–1890, June 2018. doi:10.1109/IVS.2018.8500718. ISSN: 1931-0587.

[16] Tobias Glasmachers. Limits of End-to-End Learning. *arXiv:1704.08305 [cs, stat]*, April 2017. URL http://arxiv.org/abs/1704.08305.

[17] Junho Lee and Jun Won Choi. May I Cut Into Your Lane?: A Policy Network to Learn Interactive Lane Change Behavior for Autonomous Driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4342–4347, October 2019. doi:10.1109/ITSC.2019.8917434.

[18] Jingliang Duan, Shengbo Eben Li, Yang Guan, Qi Sun, and Bo Cheng. Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data. *IET Intelligent Transport Systems*, 14(5):297–305, 2020. ISSN 1751-9578. doi:10.1049/iet-its.2019.0317.

[19] Tianyu Shi, Pin Wang, Xuxin Cheng, Ching-Yao Chan, and Ding Huang. Driving Decision and Control for Automated Lane Change Behavior based on Deep Reinforcement Learning. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2895–2900, October 2019. doi:10.1109/ITSC.2019.8917392.

[20] Shenghao Jiang, Jiying Chen, and Macheng Shen. An Interactive Lane Change Decision Making Model With Deep Reinforcement Learning. In *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 370–376, November 2019. doi:10.1109/ICCMA46720.2019.8988750.

[21] Tingting Li, Jianping Wu, and Ching-Yao Chan. Evolutionary Learning in Decision Making for Tactical Lane Changing. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1826–1831, October 2019. doi:10.1109/ITSC.2019.8916888.

[22] Songan Zhang, Huei Peng, Subramanya Nageshrao, and Eric Tseng. Discretionary Lane Change Decision Making using Reinforcement Learning with Model-Based Exploration. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 844–850, December 2019. doi:10.1109/ICMLA.2019.00147.

[23] Dapeng Liu, Mattias Brännstrom, Andrew Backhouse, and Lennart Svensson. Learning faster to perform autonomous lane changes by constructing maneuvers from shielded semantic actions. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1838–1844, October 2019. doi:10.1109/ITSC.2019.8917221. ISSN: null.

[24] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018. ISBN 978-0-262-03924-6.

[25] Junjie Wang, Qichao Zhang, Dongbin Zhao, and Yaran Chen. Lane Change Decision-making through Deep Reinforcement Learning with Rule-based Constraints. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, July 2019. doi:10.1109/IJCNN.2019.8852110.

[26] Subramanya Nageshrao, H. Eric Tseng, and Dimitar Filev. *Autonomous Highway Driving using Deep Reinforcement Learning*. March 2019.

[27] Fei Ye, Xuxin Cheng, Pin Wang, Ching-Yao Chan, and Jiucai Zhang. Automated Lane Change Strategy using Proximal Policy Optimization-based Deep Reinforcement Learning. *arXiv:2002.02667 [cs, eess]*, May 2020.

[28] Mustafa Mukadam, Akansel Cosgun, Alireza Nakhaei, and Kikuo Fujimura. Tactical Decision Making for Lane Changing with Deep Reinforcement Learning. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, US, December 2017.

[29] S. Aradi, T. Becsi, and P. Gaspar. Policy gradient based reinforcement learning approach for autonomous highway driving. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 670–675, 2018. doi:10.1109/CCTA.2018.8511514.

[30] Tamás Bécsi, Szilárd Aradi, Árpád Fehér, János Szalay, and Péter Gáspár. Highway environment model for reinforcement learning. *IFAC-PapersOnLine*, 51(22):429–434, 2018. ISSN 2405-8963. doi:https://doi.org/10.1016/j.ifacol.2018.11.596. 12th IFAC Symposium on Robot Control SYROCO 2018.

[31] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv:1610.03295 [cs, stat]*, October 2016. URL http://arxiv.org/abs/1610.03295.

[32] Long Wang, Fangmin Ye, Yibing Wang, Jingqiu Guo, Ioannis Papamichail, Markos Papageorgiou, Simon Hu, and Lihui Zhang. A Q-learning Foresighted Approach to Ego-efficient Lane Changes of Connected and Automated Vehicles on Freeways. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1385–1392, October 2019. doi:10.1109/ITSC.2019.8917036.

[33] Ali Alizadeh, Majid Moghadam, Yunus Bicer, Nazim Kemal Ure, Ugur Yavas, and Can Kurtulus. Automated Lane Change Decision Making using Deep Reinforcement Learning in Dynamic and Uncertain Highway Environment. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1399–1404, October 2019. doi:10.1109/ITSC.2019.8917192.

[34] C. Hoel, K. Wolff, and L. Laine. Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2148–2155, November 2018. doi:10.1109/ITSC.2018.8569568.

[35] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Management Science*, 35(8):982–1003, August 1989. ISSN 0025-1909. doi:10.1287/mnsc.35.8.982.

[36] Viswanath Venkatesh, Michael G. Morris, Gordon B. Davis, and Fred D. Davis. User Acceptance of Information Technology: Toward a Unified View. *MIS Q.*, 27 (3):425–478, 2003. doi:10.2307/30036540.

[37] Sebastian Osswald, Daniela Wurhofer, Sandra Trösterer, Elke Beck, and Manfred Tscheligi. Predicting information technology usage in the car: towards a car technology acceptance model. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 51–58, October 2012. ISBN 978-1-4503-1751-1. doi:10.1145/2390256.2390264.

[38] Sina Nordhoff, Bart van Arem, and Riender Happee. Conceptual Model to Explain, Predict, and Improve User Acceptance of Driverless Podlike Vehicles. *Transportation Research Record*, 2602(1):60–67, January 2016. doi:10.3141/2602-08.

[39] Sina Nordhoff, Miltos Kyriakidis, Bart van Arem, and Riender Happee. A multi-level model on automated vehicle acceptance (MAVA): a review-based study. *Theoretical Issues in Ergonomics Science*, 20(6):682–710, November 2019. ISSN 1463-922X. doi:10.1080/1463922X.2019.1621406.

[40] Peng Jing, Gang Xu, Yuexia Chen, Yuji Shi, and Fengping Zhan. The Determinants behind the Acceptance of Autonomous Vehicles: A Systematic Review. *Sustainability*, 12(5):1719, January 2020. doi:10.3390/su12051719.

[41] S. Nordhoff, J. C. F. de Winter, Ruth Madigan, Natasha Merat, B. van Arem, and R. Happee. User acceptance of automated shuttles in Berlin-Schöneberg: A questionnaire study. *Transportation Research. Part F: Traffic Psychology and Behaviour*, 58, 2018. ISSN 1369-8478. doi:10.1016/j.trf.2018.06.024.

[42] Sina Nordhoff, Joost de Winter, Miltos Kyriakidis, Bart van Arem, and Riender Happee. Acceptance of Driverless Vehicles: Results from a Large Cross-National Questionnaire Study. *Journal of Advanced Transportation*, 2018:1–22, April 2018. ISSN 0197-6729. doi:10.1155/2018/5382192.

[43] Zhigang Xu, Kaifan Zhang, Haigen Min, Zhen Wang, Xiangmo Zhao, and Peng Liu. What drives people to accept automated vehicles? Findings from a field experiment. *Transportation Research Part C: Emerging Technologies*, 95:320–334, October 2018. ISSN 0968-090X. doi:10.1016/j.trc.2018.07.024.

[44] Jan C. Zoellick, Adelheid Kuhlmey, Liane Schenk, Daniel Schindel, and Stefan Blüher. Assessing acceptance of electric automated vehicles after exposure in a realistic traffic environment. *PLOS ONE*, 14(5):1–23, 2019. ISSN 1932-6203. doi:10.1371/journal.pone.0215969.

[45] Sarah-Maria Castritius, Heiko Hecht, Johanna Möller, Christoph J. Dietz, Patric Schubert, Christoph Bernhard, Simone Morvilius, Christian T. Haas, and Sabine Hammer. Acceptance of truck platooning by professional drivers on German highways. A mixed methods approach. *Applied Ergonomics*, 85:103042, May 2020. ISSN 0003-6870. doi:10.1016/j.apergo.2019.103042.

[46] Jong Kyu Choi and Yong Gu Ji. Investigating the Importance of Trust on Adopting an Autonomous Vehicle. *International Journal of Human–Computer Interaction*, 31 (10):692–702, October 2015. ISSN 1044-7318. doi:10.1080/10447318.2015.1070549.

[47] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv:1707.06347 [cs]*, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv: 1707.06347.

[48] Najah Abu Ali and Hatem Abou-zeid. Driver Behavior Modeling: Developments and Future Directions. *International Journal of Vehicular Technology*, 2016:1–12, December 2016. doi:10.1155/2016/6952791.

[49] John A. Michon. A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? In Leonard Evans and Richard C. Schwing, editors, *Human Behavior and Traffic Safety*, pages 485–524. Springer US, Boston, MA, 1985. ISBN 978-1-4613-2173-6. doi:10.1007/978-1-4613-2173-6_19.

[50] Zihan Ding, Yanhua Huang, Hang Yuan, and Hao Dong. Introduction to Reinforcement Learning. In Hao Dong, Zihan Ding, and Shanghang Zhang, editors, *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pages 47–123. Springer, Singapore, 2020. ISBN 9789811540950. doi:10.1007/978-981-15-4095-0_2.

[51] Yanhua Huang. Deep Q-Networks. In Hao Dong, Zihan Ding, and Shanghang Zhang, editors, *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pages 135–160. Springer, Singapore, 2020. ISBN 9789811540950. doi:10.1007/978-981-15-4095-0_4.

[52] Ruitong Huang, Tianyang Yu, Zihan Ding, and Shanghang Zhang. Policy Gradient. In Hao Dong, Zihan Ding, and Shanghang Zhang, editors, *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pages 161–212. Springer, Singapore, 2020. ISBN 9789811540950. doi:10.1007/978-981-15-4095-0_5.

[53] Jingqing Zhang, Hang Yuan, and Hao Dong. Introduction to Deep Learning. In Hao Dong, Zihan Ding, and Shanghang Zhang, editors, *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pages 3–46. Springer, Singapore, 2020. ISBN 9789811540950. doi:10.1007/978-981-15-4095-0_1.

[54] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization. *arXiv:1502.05477 [cs]*, April 2017. URL http://arxiv.org/abs/1502.05477.

[55] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv:1506.02438 [cs]*, October 2018. URL http://arxiv.org/abs/1506.02438. arXiv: 1506.02438.

[56] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

[57] Nidhi Kalra and Susan M. Paddock. Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? In *Driving to Safety*, How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?, pages 1–16. RAND Corporation, 2016. URL https://www.jstor.org/stable/10.7249/j.ctt1btc0xw.1.

[58] Yujun Cho, Jaekyu Park, Sungjun Park, and Eui S. Jung. Technology Acceptance Modeling based on User Experience for Autonomous Vehicles. *Journal of the Ergonomics Society of Korea*, 36(2):87–108, 2017. ISSN 1229-1684. doi:10.5143/JESK.2017.36.2.87.

[59] John D. Lee and Katrina A. See. Trust in Automation: Designing for Appropriate Reliance. *Human Factors*, August 2016. doi:10.1518/hfes.46.1.50_30392. URL https://journals.sagepub.com/doi/10.1518/hfes.46.1.50_30392. Publisher: SAGE PublicationsSage UK: London, England.

[60] Mahtab Ghazizadeh, Yiyun Peng, John D. Lee, and Linda N.G. Boyle. Augmenting the Technology Acceptance Model with Trust: Commercial Drivers' Attitudes towards Monitoring and Feedback. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56(1):2286–2290, September 2012. ISSN 2169-5067. doi:10.1177/1071181312561481. URL https://doi.org/10.1177/1071181312561481.

[61] Rui Fu, Zhen Li, Qinyu Sun, and Chang Wang. Human-like car-following model for autonomous vehicles considering the cut-in behavior of other vehicles in mixed traffic. *Accident Analysis & Prevention*, 132:105260, November 2019. ISSN 0001-4575. doi:10.1016/j.aap.2019.105260.

[62] Jae-gil Lee, Ki Joon Kim, Sangwon Lee, and Don Shin. Can autonomous vehicles be safe and trustworthy? Effects of appearance and autonomy of unmanned driving systems. *International Journal of Human-Computer Interaction*, 31:682–691, July 2015. doi:10.1080/10447318.2015.1070547.

[63] Adam Waytz, Joy Heafner, and Nicholas Epley. The mind in the machine: Anthropomorphism increases trust in an autonomous vehicle. *Journal of Experimental Social Psychology*, 52:113–117, May 2014. ISSN 0022-1031. doi:10.1016/j.jesp.2014.01.005.

[64] Sarvesh Kolekar, Joost de Winter, and David Abbink. Human-like driving behaviour emerges from a risk-based driver model. *Nature Communications*, 11(1): 4850, September 2020. ISSN 2041-1723. doi:10.1038/s41467-020-18353-4. URL https://www.nature.com/articles/s41467-020-18353-4.

[65] Stefan Griesche, Eric Nicolay, Dirk Assmann, Mandy Dotzauer, and David Käthner. Should my car drive as I do? What kind of driving style do drivers prefer for the design of automated driving functions? In *Braunschweiger Symposium*, volume 10, pages 185–204, February 2016. ISBN 978-3-937655-37-6.

[66] Pavlo Bazilinskyy, Tsuyoshi Sakuma, and Joost de Winter. *The 'Turing test' of automated driving*. August 2020.

[67] Luis Oliveira, Karl Proctor, Christopher Burns, and Stewart Birrell. Driving Style: How Should an Automated Vehicle Behave? *Information (Switzerland)*, 10:1–20, June 2019. doi:10.3390/info10060219.

[68] Nationale databank wegverkeergegevens - open data, 2020. URL https://www.ndw.nu/pagina/nl/103/datalevering/120/open_data/.

[69] Lily Elefteriadou. Flow, Speed, Density, and Their Relationships. In Lily Elefteriadou, editor, *An Introduction to Traffic Flow Theory*, Springer Optimization and Its Applications, pages 61–91. Springer, New York, NY, 2014. ISBN 978-1-4614-8435-6. doi:10.1007/978-1-4614-8435-6_3.

[70] Martin Treiber and Arne Kesting. *Traffic Flow Dynamics*. Springer, Berlin, Heidelberg, 1st edition, 2013. ISBN 978-3-642-32460-4.

[71] Donald S. Berry and Daniel M. Belmont. Distribution of Vehicle Speeds and Travel Times. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 589–602. The Regents of the University of California, 1951.

[72] Richard J. Salter. Traffic speed distributions and estimation. In Richard J. Salter, editor, *Highway Traffic Analysis and Design*, pages 131–145. Macmillan Education UK, London, 1996. ISBN 978-1-349-13423-6. doi:10.1007/978-1-349-13423-6_14.

[73] Dirk Helbing, Ansgar Hennecke, Vladimir I. Shvetsov, and Martin Treiber. Micro- and macro-simulation of freeway traffic. *Mathematical and Computer Modelling*, 35 (5):517–547, March 2002. ISSN 0895-7177. doi:10.1016/S0895-7177(02)80019-X.

[74] Richard J. Salter. Headway distributions in highway traffic flow. In Richard J. Salter, editor, *Highway Traffic Analysis and Design*, pages 107–124. Macmillan Education UK, London, 1976. ISBN 978-1-349-06952-1. doi:10.1007/978-1-349-06952-1_12.

[75] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A General Platform for Intelligent Agents. *arXiv:1809.02627 [cs, stat]*, May 2020. URL http://arxiv.org/abs/1809.02627.

[76] Corentin Tallec, Léonard Blier, and Yann Ollivier. Making Deep Q-learning methods robust to time discretization. *arXiv:1901.09732 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1901.09732.

[77] Alex Braylan, Mark Hollenbeck, Elliot Meyerson, and Risto Miikkulainen. Frame Skip Is a Powerful Parameter for Learning to Play Atari. In *AAAI Workshop: Learning for General Competency in Video Games*, pages 10–11, 2015.

[78] Motaz Khader and Samir Cherian. An Introduction to Automotive LIDAR. Technical Report SLYY150A, Texas Instruments, Dallas, TX.

[79] Hermann Winner. Adaptive Cruise Control. In Azim Eskandarian, editor, *Handbook of Intelligent Vehicles*, pages 613–656. Springer, London, 2012. ISBN 978-0-85729-085-4. doi:10.1007/978-0-85729-085-4_24.

[80] Bart van Arem, Cornelie J. G. van Driel, and Ruben Visser. The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, December 2006. ISSN 1558-0016. doi:10.1109/TITS.2006.884615.

[81] Jing Zhou and Huei Peng. Range policy of adaptive cruise control vehicles for improved flow stability and string stability. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):229–237, June 2005. ISSN 1558-0016. doi:10.1109/TITS.2005.848359.

[82] Nathaniel H. Sledge and Kurt M. Marshek. Comparison of Ideal Vehicle Lane-Change Trajectories. *SAE Transactions*, 106:2004–2027, February 1997. ISSN 0096736X, 25771531. doi:10.4271/971062.

[83] Kurt Enke. Possibilities for improving safety within the driver vehicle environment control loop. In *Proceedings of International Technological Conference on Experimental Safety Vehicles*, pages 789–802, 1979.

[84] Winston Nelson. Continuous-curvature paths for autonomous vehicles. In *1989 International Conference on Robotics and Automation Proceedings*, pages 1260–1264, May 1989. doi:10.1109/ROBOT.1989.100153.

[85] Guoqing Xu, Li Liu, Yongsheng Ou, and Zhangjun Song. Dynamic Modeling of Driver Control Strategy of Lane-Change Behavior and Trajectory Planning for Collision Prediction. *Intelligent Transportation Systems, IEEE Transactions on*, 13: 1138–1155, September 2012. doi:10.1109/TITS.2012.2187447.

[86] Bin Zhou, Yunpeng Wang, Guizhen Yu, and Xinkai Wu. A lane-change trajectory model from drivers' vision view. *Transportation Research Part C: Emerging Technologies*, 85:609–627, December 2017. ISSN 0968-090X. doi:10.1016/j.trc.2017.10.013.

[87] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125, November 2018. doi:10.1109/ITSC.2018.8569552. ISSN: 2153-0017.

[88] Wen Yao, Huijing Zhao, Franck Davoine, and Hongbin Zha. Learning lane change trajectories from on-road driving data. In *2012 IEEE Intelligent Vehicles Symposium*, pages 885–890, June 2012. doi:10.1109/IVS.2012.6232190. ISSN: 1931-0587.

[89] Iakovos Papadimitriou and Masayoshi Tomizuka. Fast lane changing computations using polynomials. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 1, pages 48–53, June 2003. doi:10.1109/ACC.2003.1238912.

[90] James Stewart. *Calculus Early Transcendentals*. Cengage Learning, 7th edition, 2012. ISBN 978-0-538-49781-7.

[91] Gabriel M. Hoffmann, Claire J. Tomlin, Michael Montemerlo, and Sebastian Thrun. Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing. In *2007 American Control Conference*, pages 2296–2301, July 2007. doi:10.1109/ACC.2007.4282788.

[92] Scott Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Eng, Daniela Rus, and Marcelo Jr. Perception, Planning, Control, and Coordination for Autonomous Vehicles. *Machines*, 5(6):1–54, February 2017. doi:10.3390/machines5010006.

[93] Xuesong Wang, Minming Yang, and David Hurwitz. Analysis of cut-in behavior based on naturalistic driving data. *Accident Analysis & Prevention*, 124:127–137, March 2019. ISSN 0001-4575. doi:10.1016/j.aap.2019.01.006.

[94] Tomer Toledo and David Zohar. Modeling Duration of Lane Changes. *Transportation Research Record*, 1999(1):71–78, January 2007. ISSN 0361-1981. doi:10.3141/1999-08.

[95] Suzanne E. Lee, Erik C.B. Olsen, and Walter W. Wierwille. A Comprehensive Examination of Naturalistic Lane-Changes. Technical Report DOT HS 809 702, American Psychological Association, Blacksburg, VA, 2004. URL http://doi.apa.org/get-pe-doi.cfm?doi=10.1037/e733232011-001.

[96] Orit Taubman-Ben-Ari, Mario Mikulincer, and Omri Gillath. The multidimensional driving style inventory—scale construct and validation. *Accident Analysis & Prevention*, 36(3):323–332, May 2004. ISSN 0001-4575. doi:10.1016/S0001-4575(03)00010-1.

[97] Yifan Zhang, Qian Xu, Jianping Wang, Kui Wu, Zuduo Zheng, and Kejie Lu. A Learning-based Discretionary Lane-Change Decision-Making Model with Driving Style Awareness. *arXiv:2010.09533 [cs]*, October 2020. URL http://arxiv.org/abs/2010.09533.

[98] Saskia Ossen, Serge P. Hoogendoorn, and Ben G. H. Gorte. Interdriver Differences in Car-Following: A Vehicle Trajectory–Based Study. *Transportation Research Record*, 1965(1):121–129, January 2006. ISSN 0361-1981. doi:10.1177/0361198106196500113.

[99] Saskia Ossen and Serge Hoogendoorn. Heterogeneity In Car-Following Behavior: Theory And Empirics. *Transportation Research Part C Emerging Technologies*, 19, April 2011. doi:10.1016/j.trc.2010.05.006.

[100] Alice H. Eagly and Shelly Chaiken. *The psychology of attitudes*. The psychology of attitudes. Harcourt Brace Jovanovich College Publishers, Orlando, FL, US, 1993. ISBN 0-15-500097-7 (Hardcover).

[101] William Payre, Julien Cestac, and Patricia Delhomme. Intention to use a fully automated car: Attitudes and a priori acceptability. *Transportation Research Part F: Traffic Psychology and Behaviour*, 27:252–263, November 2014. ISSN 1369-8478. doi:10.1016/j.trf.2014.04.009.

[102] Thomas Alexander Sick Nielsen and Sonja Haustein. On sceptics and enthusiasts: What are the expectations towards self-driving cars? *Transport Policy*, 66:49–55, August 2018. ISSN 0967-070X. doi:10.1016/j.tranpol.2018.03.004.

[103] Ahmadreza Talebian and Sabyasachee Mishra. Predicting the adoption of connected autonomous vehicles: A new approach based on the theory of diffusion of innovations. *Transportation Research Part C: Emerging Technologies*, 95:363–380, October 2018. ISSN 0968-090X. doi:10.1016/j.trc.2018.06.005.

[104] M. Kyriakidis, R. Happee, and J. C. F. de Winter. Public opinion on automated driving: Results of an international questionnaire among 5000 respondents. *Transportation Research Part F: Traffic Psychology and Behaviour*, 32:127–140, July 2015. ISSN 1369-8478. doi:10.1016/j.trf.2015.04.014.

[105] Grace Eden, Benjamin Nanchen, Randolf Ramseyer, and Florian Evéquoz. Expectation and Experience: Passenger Acceptance of Autonomous Public Transportation Vehicles. September 2017. ISBN 978-3-319-68058-3. doi:10.1007/978-3-319-68059-0_30.

[106] Murat Dikmen and Catherine M. Burns. Autonomous Driving in the Real World: Experiences with Tesla Autopilot and Summon. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Automotive'UI 16, pages 225–228, New York, NY, USA, October 2016. Association for Computing Machinery. ISBN 978-1-4503-4533-0. doi:10.1145/3003715.3005465.

[107] Markku Kilpeläinen and Heikki Summala. Effects of weather and weather forecasts on driver behaviour. *Transportation Research Part F: Traffic Psychology and Behaviour*, 10(4):288–299, July 2007. ISSN 1369-8478. doi:10.1016/j.trf.2006.11.002.

[108] Felix Wilhelm Siebert and Fares Lian Wallis. How speed and visibility influence preferred headway distances in highly automated driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, 64:485–494, July 2019. ISSN 1369-8478. doi:10.1016/j.trf.2019.06.009.

[109] J Philippe Rushton, Charles J Brainerd, and Michael Pressley. Behavioral Development and Construct Validity: The Principle of Aggregation. *Psychological Bulletin*, 94(1):18–38, 1983.

[110] Franziska Hartwich, Matthias Beggiato, and Josef F. Krems. Driving comfort, enjoyment and acceptance of automated driving – effects of drivers' age and driving style familiarity. *Ergonomics*, 61(8):1017–1032, August 2018. ISSN 0014-0139. doi:10.1080/00140139.2018.1441448.

[111] Pavlo Bazilinskyy, Yke Eisma, Dimitra Dodou, and Joost de Winter. Risk perception: A study using dashcam videos and participants from different world regions. *Traffic Injury Prevention*, May 2020. doi:10.1080/15389588.2020.1762871.

[112] Da Yang, Shiyu Zheng, Cheng Wen, Peter J. Jin, and Bin Ran. A dynamic lane-changing trajectory planning model for automated vehicles. *Transportation Research Part C: Emerging Technologies*, 95:228–247, October 2018. ISSN 0968-090X. doi:10.1016/j.trc.2018.06.007. URL http://www.sciencedirect.com/science/article/pii/S0968090X18305898.

[113] Michael Buhrmester, Tracy Kwang, and Samuel D. Gosling. Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data? *Perspectives on Psychological Science*, 6(1):3–5, January 2011. ISSN 1745-6916. doi:10.1177/1745691610393980.

[114] Matthew J. C. Crump, John V. McDonnell, and Todd M. Gureckis. Evaluating Amazon's Mechanical Turk as a Tool for Experimental Behavioral Research. *PLOS ONE*, 8(3):e57410, 2013. ISSN 1932-6203. doi:10.1371/journal.pone.0057410.

[115] Akira Kinose and Tadahiro Taniguchi. Integration of imitation learning using GAIL and reinforcement learning using task-achievement rewards via probabilistic graphical model. *Advanced Robotics*, 34(16):1055–1067, August 2020. ISSN 0169-1864. doi:10.1080/01691864.2020.1778521.

A A4 HIGHWAY TRAFFIC DATA

This appendix contains the raw traffic data (hourly intensity $q(t)$ and hourly mean velocity $V(t)$) averaged across four data points [68] on the A4 highway in the Netherlands. All data has been averaged over the year 2019 with Saturdays, Sundays and holidays excluded. See Table A.1 and Table A.2.

The sensors are located close to interchange Burgerveen. See Figure A.1 for a detailed view on the sensor locations. Only the traffic headed north has been included in the data. This part of the highway has three lanes.

Figure A.1: Locations of the traffic sensors

Table A.1: Hourly intensity $q(t)$ across 4 measurement points in veh/h

	0041hr l0205ra	0041hr l0209ra	0041hr l0214ra	0041hr l0218ra
00:00	516.3	513.6	475.2	513.6
01:00	286.1	284.8	254.2	285.4
02:00	227	226.2	199.8	226.5
03:00	323.3	322.9	297	325.4
04:00	680.5	680.9	623.1	688.1
05:00	2663.8	2671.9	2535.5	2717.9
06:00	5677.6	5669.2	5484.3	5723.8
07:00	5551.8	5545.9	5385.3	5599.9
08:00	5410.7	5393.8	5253.2	5432.6
09:00	4943.7	4924.3	4727.3	4943
10:00	4092.1	4073.1	3849.2	4091.7
11:00	3675.6	3663.2	3443.8	3688.1
12:00	3986.3	3974.3	3754	3999
13:00	3965.7	3950.9	3720.4	3974.1
14:00	3976.9	3962	3735.5	3991.8
15:00	4467.6	4457.1	4243.1	4493.5
16:00	4837.2	4821.7	4669	4862.2
17:00	4755.5	4746.9	4636.1	4779.5
18:00	4056.1	4040.6	3932	4056.1
19:00	2676	2662.6	2555	2662.8
20:00	1871	1862.8	1779.2	1871.8
21:00	1670.4	1663.8	1597.6	1670.6
22:00	1441.4	1433.2	1373.9	1439.7
23:00	1076.9	1070.5	1017.2	1072.5

Table A.2: Hourly mean velocity $V(t)$ across 4 measurement points in km/h

	0041hr l0205ra	0041hr l0209ra	0041hr l0214ra	0041hr l0218ra
00:00	113	111	112	112
01:00	108	106	108	107
02:00	106	104	107	105
03:00	109	108	110	109
04:00	113	112	114	113
05:00	111	109	112	110
06:00	90	88	91	87
07:00	77	74	76	73
08:00	73	72	73	71
09:00	84	82	83	82
10:00	99	97	100	97
11:00	101	99	102	99
12:00	101	99	101	99
13:00	99	96	98	96
14:00	101	98	100	97
15:00	100	97	100	97
16:00	100	98	100	98
17:00	99	96	98	96
18:00	103	101	103	101
19:00	113	111	113	112
20:00	114	112	114	113
21:00	115	113	115	114
22:00	116	114	115	114
23:00	115	114	115	113

B VIDEO PRESSING DATA

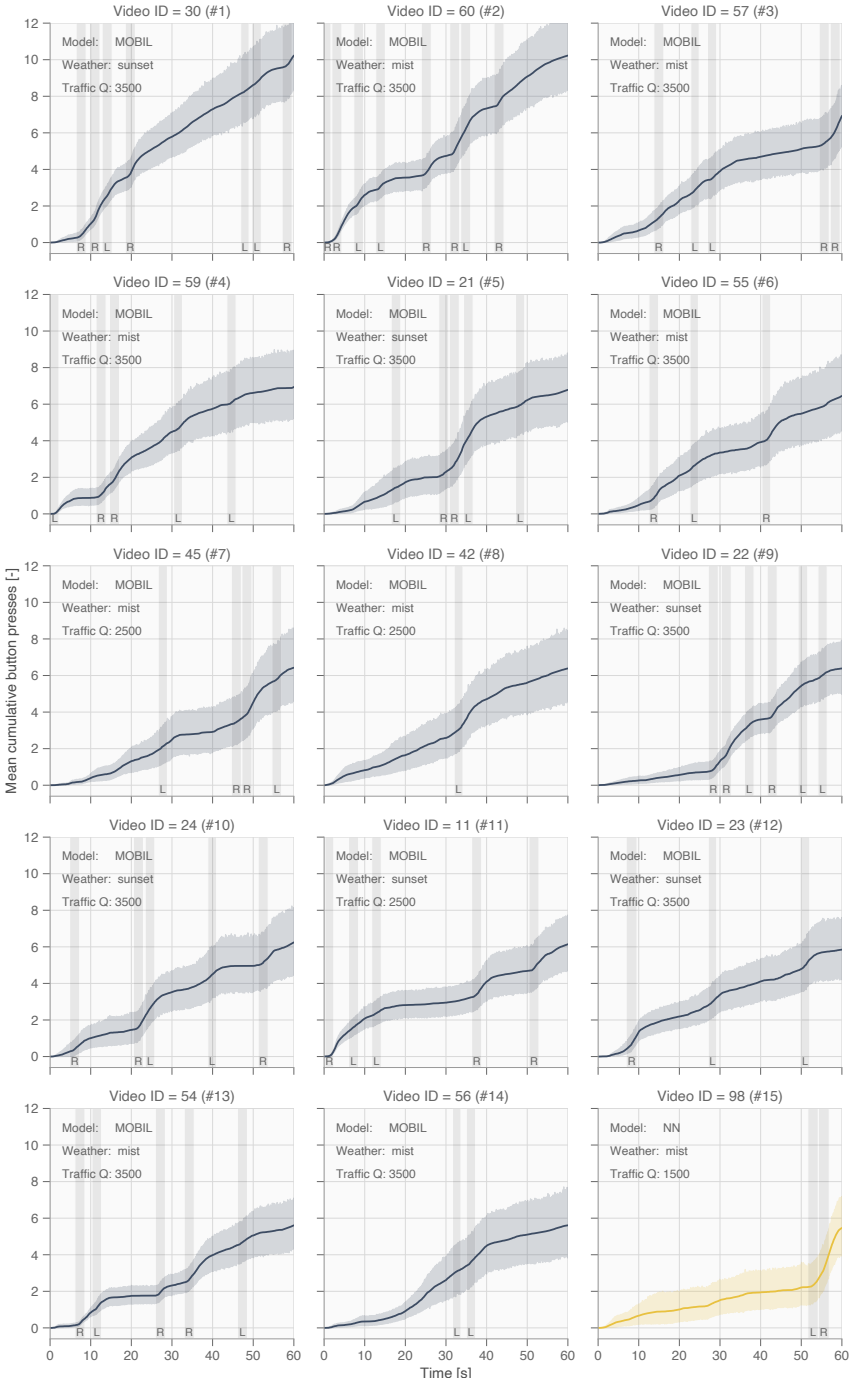


Figure B.1: Average cumulative button presses over time ranked 1 to 15

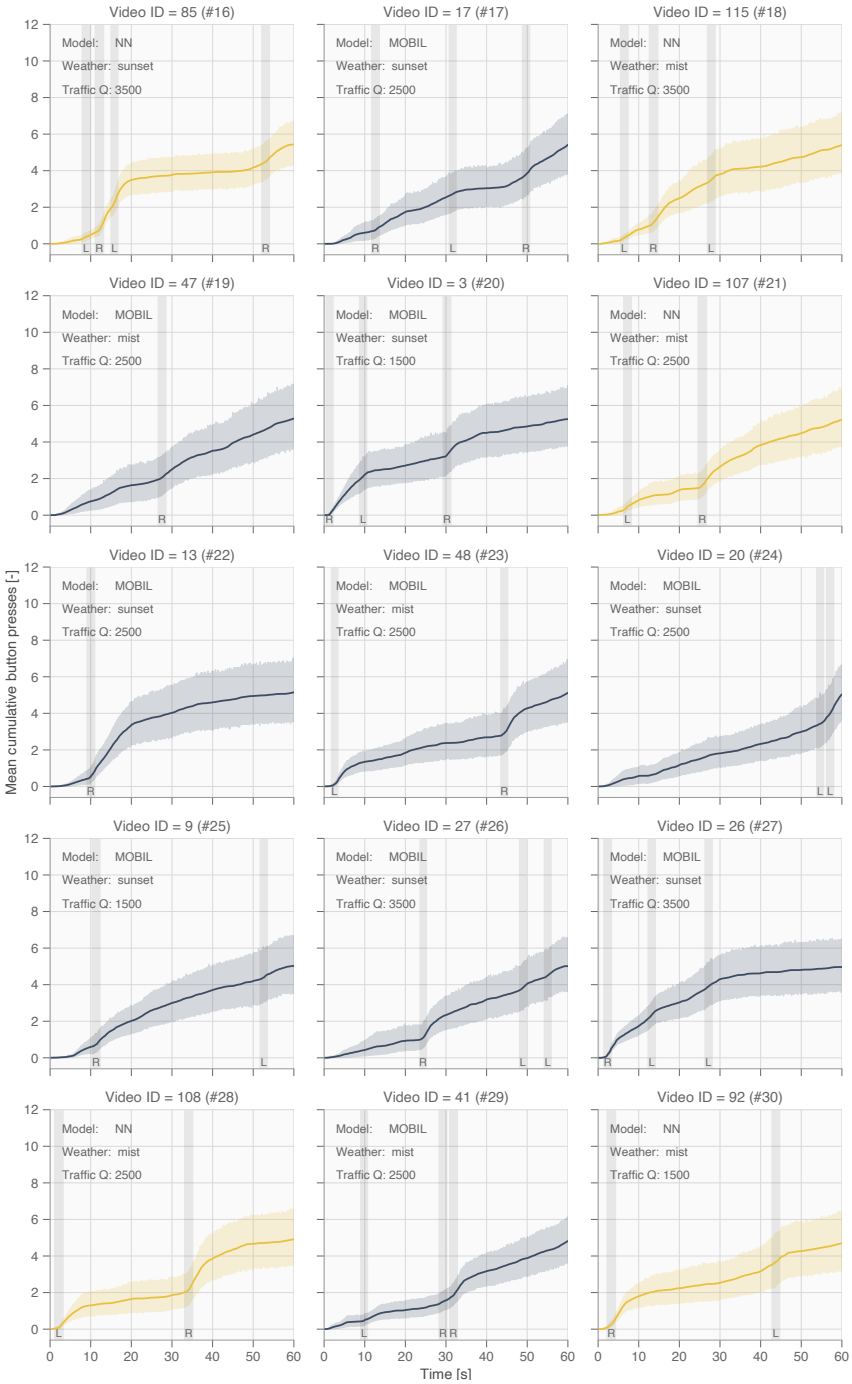


Figure B.2: Average cumulative button presses over time ranked 16 to 30

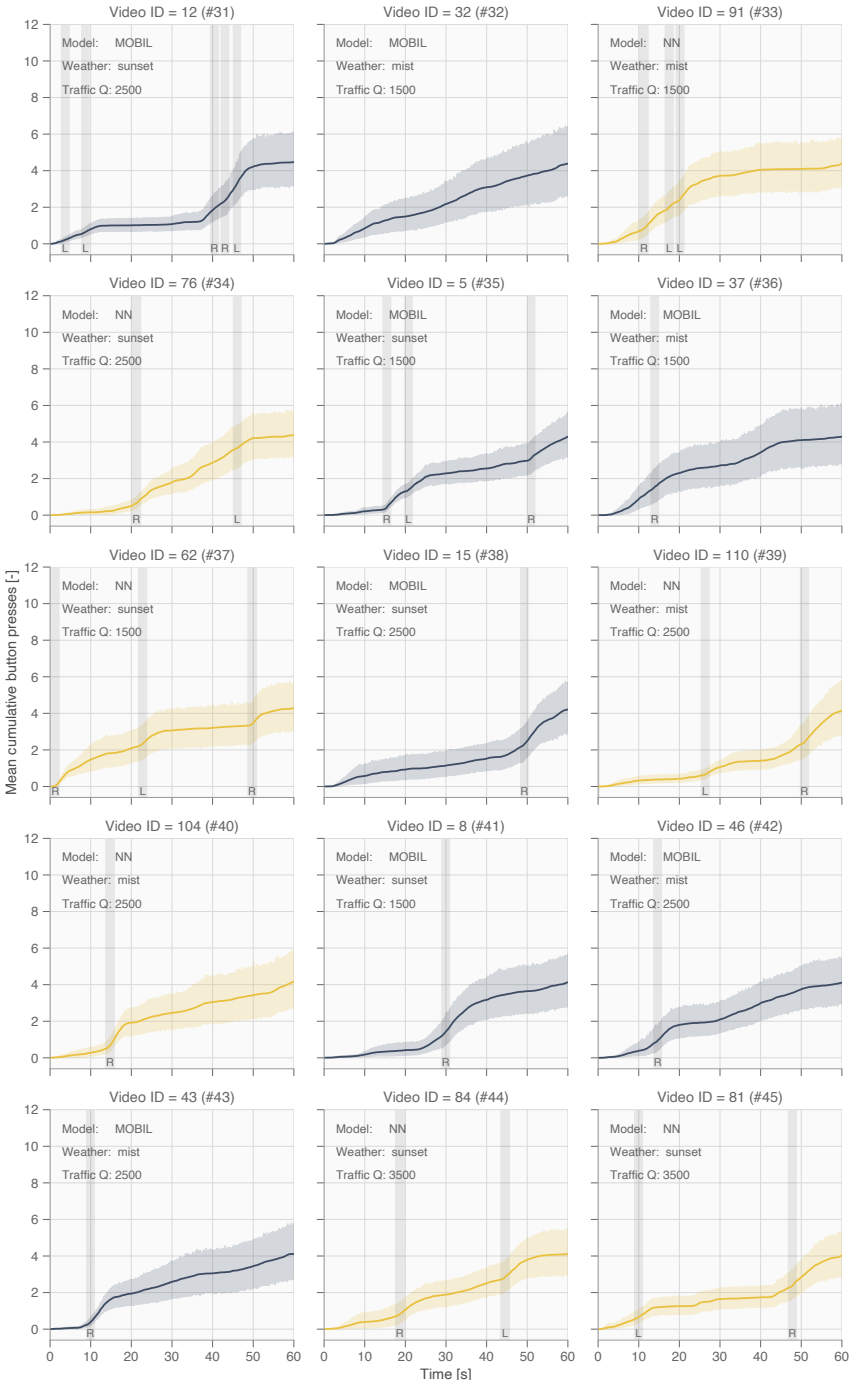


Figure B.3: Average cumulative button presses over time ranked 31 to 45

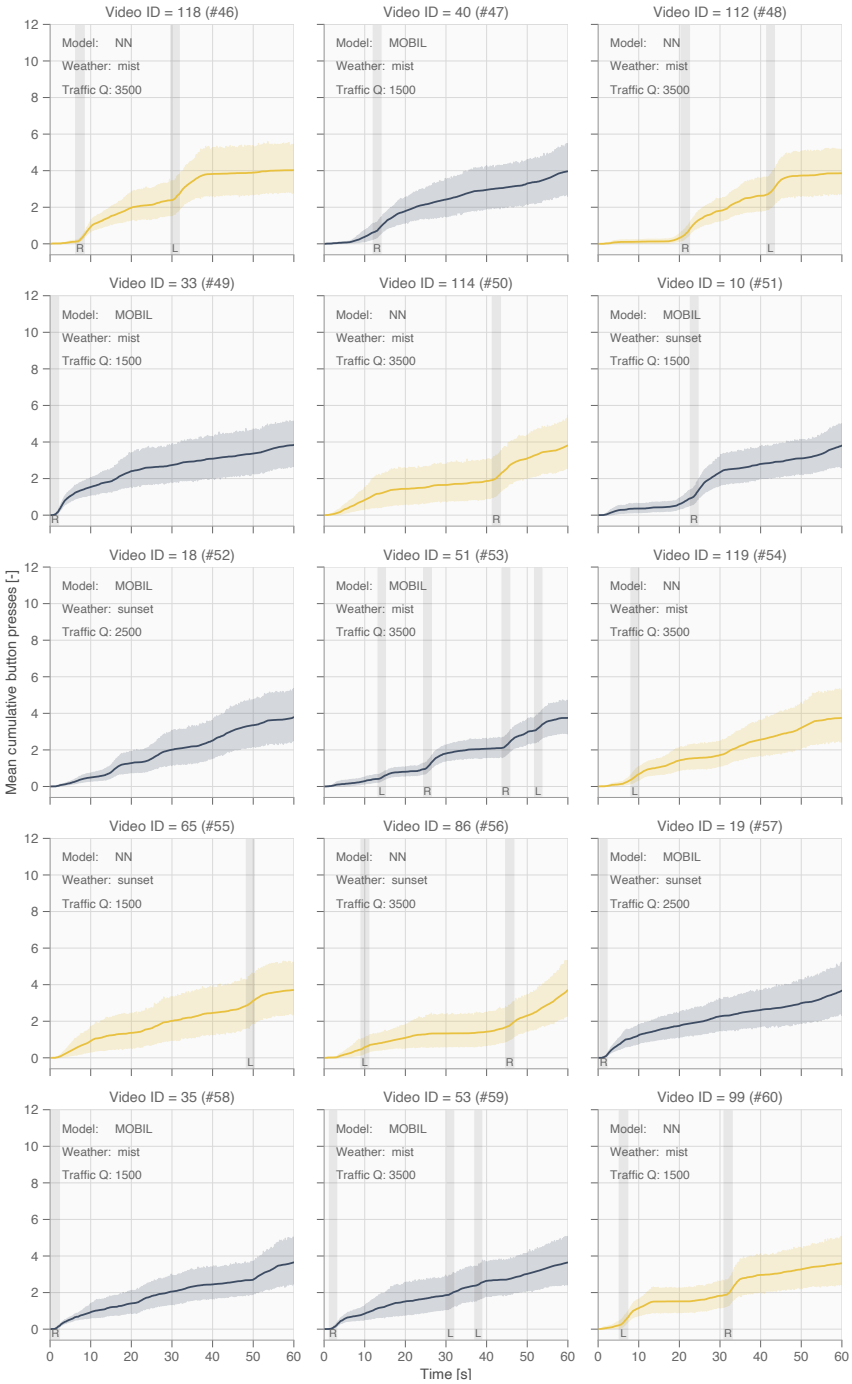


Figure B.4: Average cumulative button presses over time ranked 46 to 60

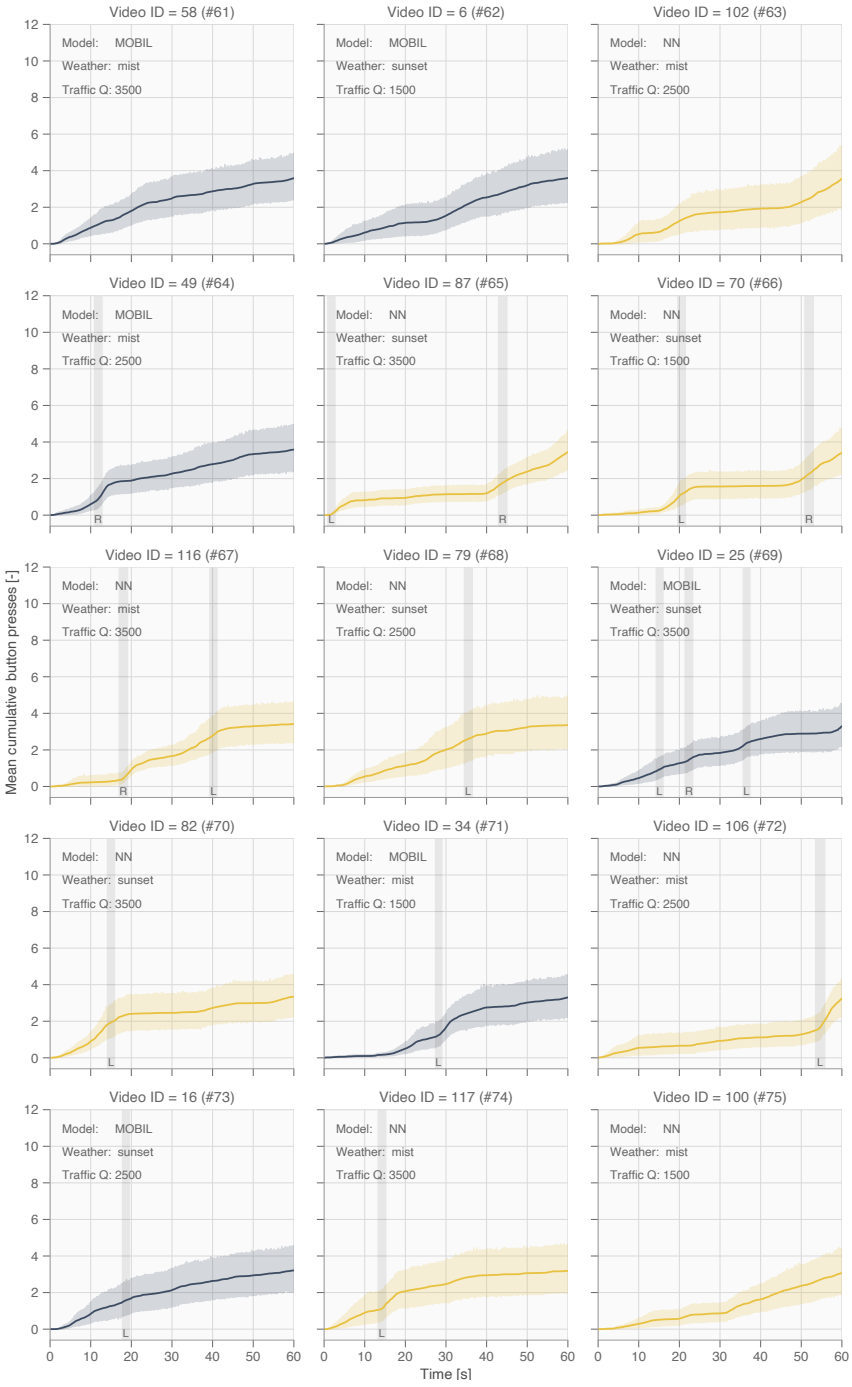


Figure B.5: Average cumulative button presses over time ranked 61 to 75

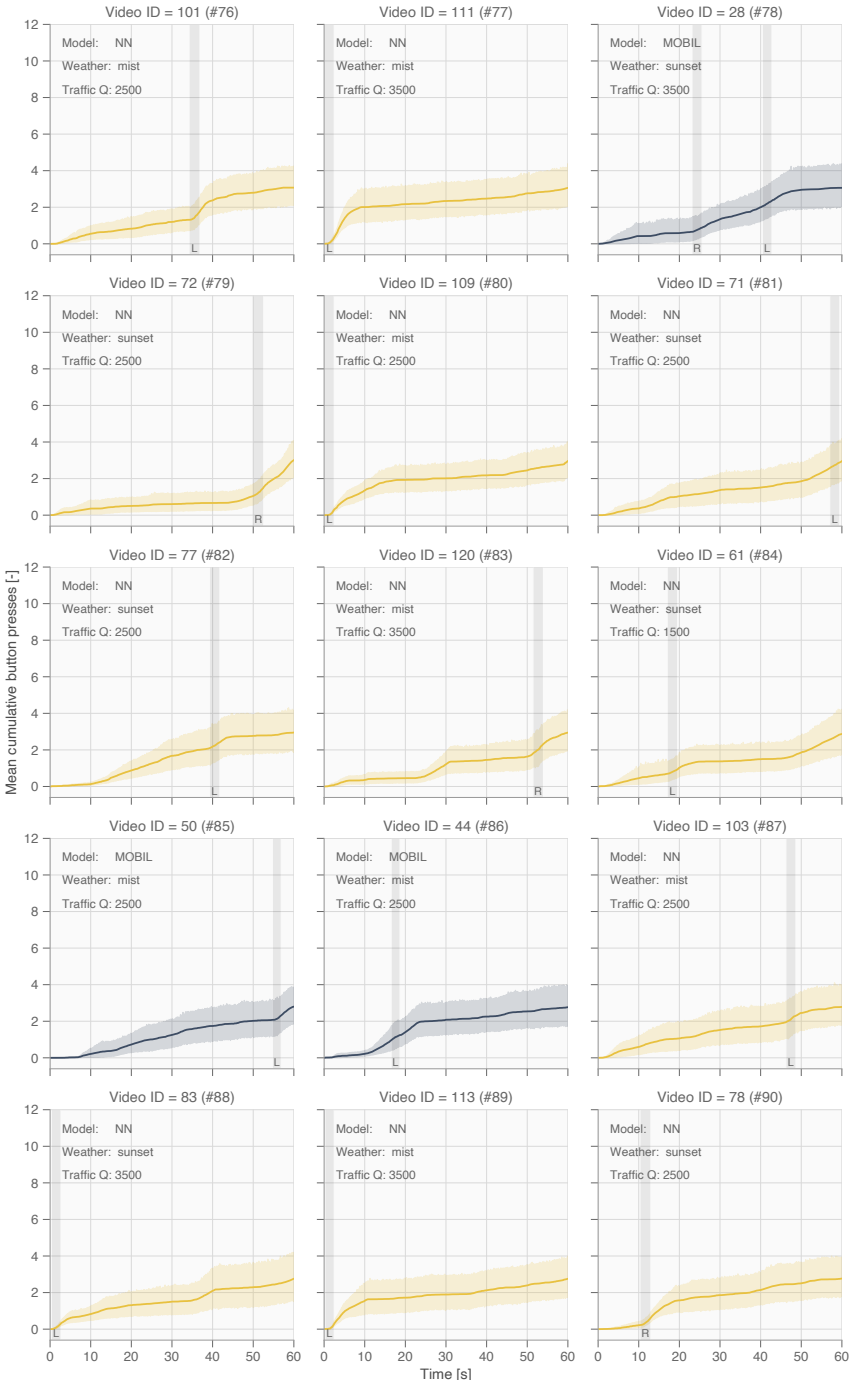


Figure B.6: Average cumulative button presses over time ranked 76 to 90

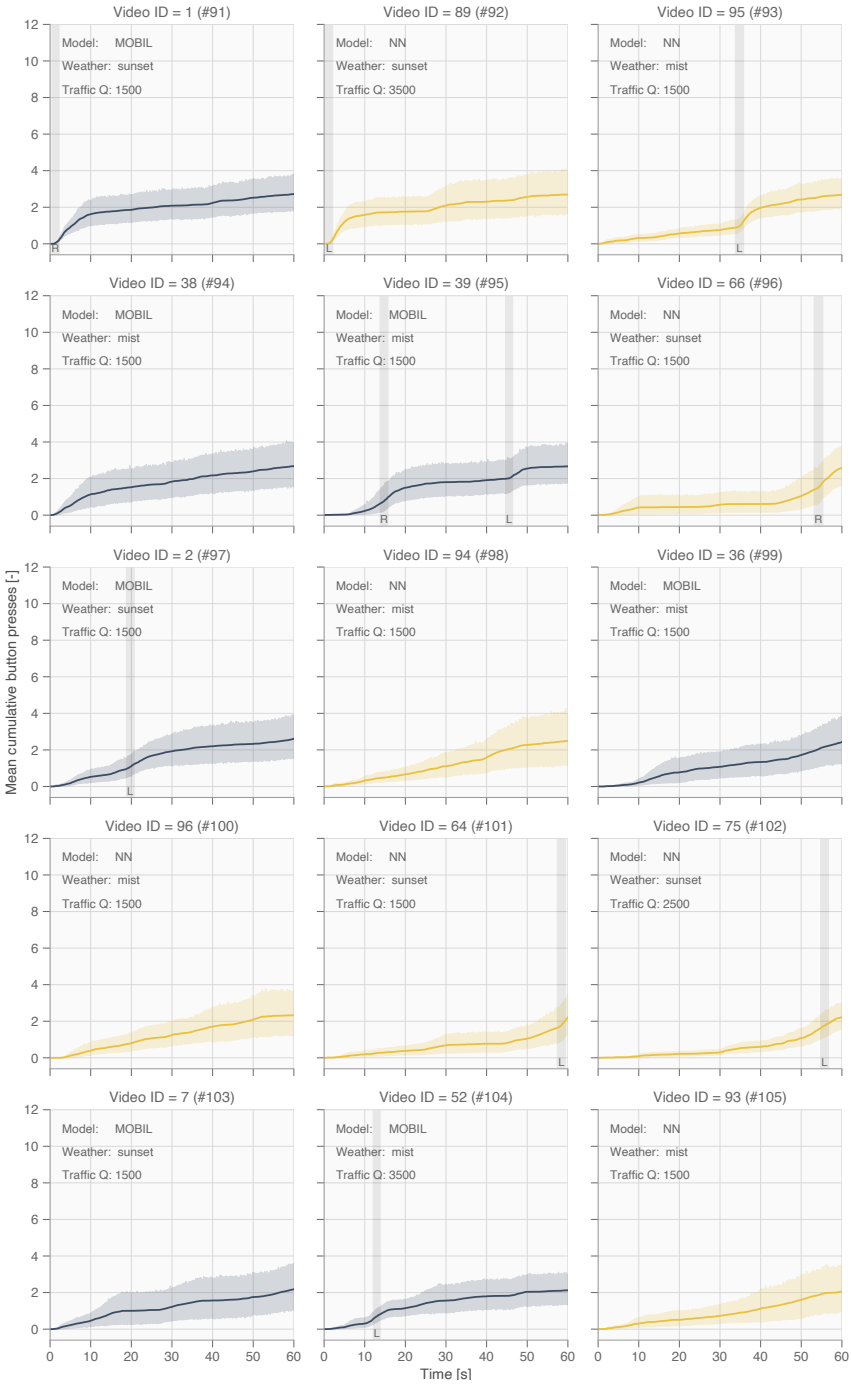


Figure B.7: Average cumulative button presses over time ranked 91 to 105

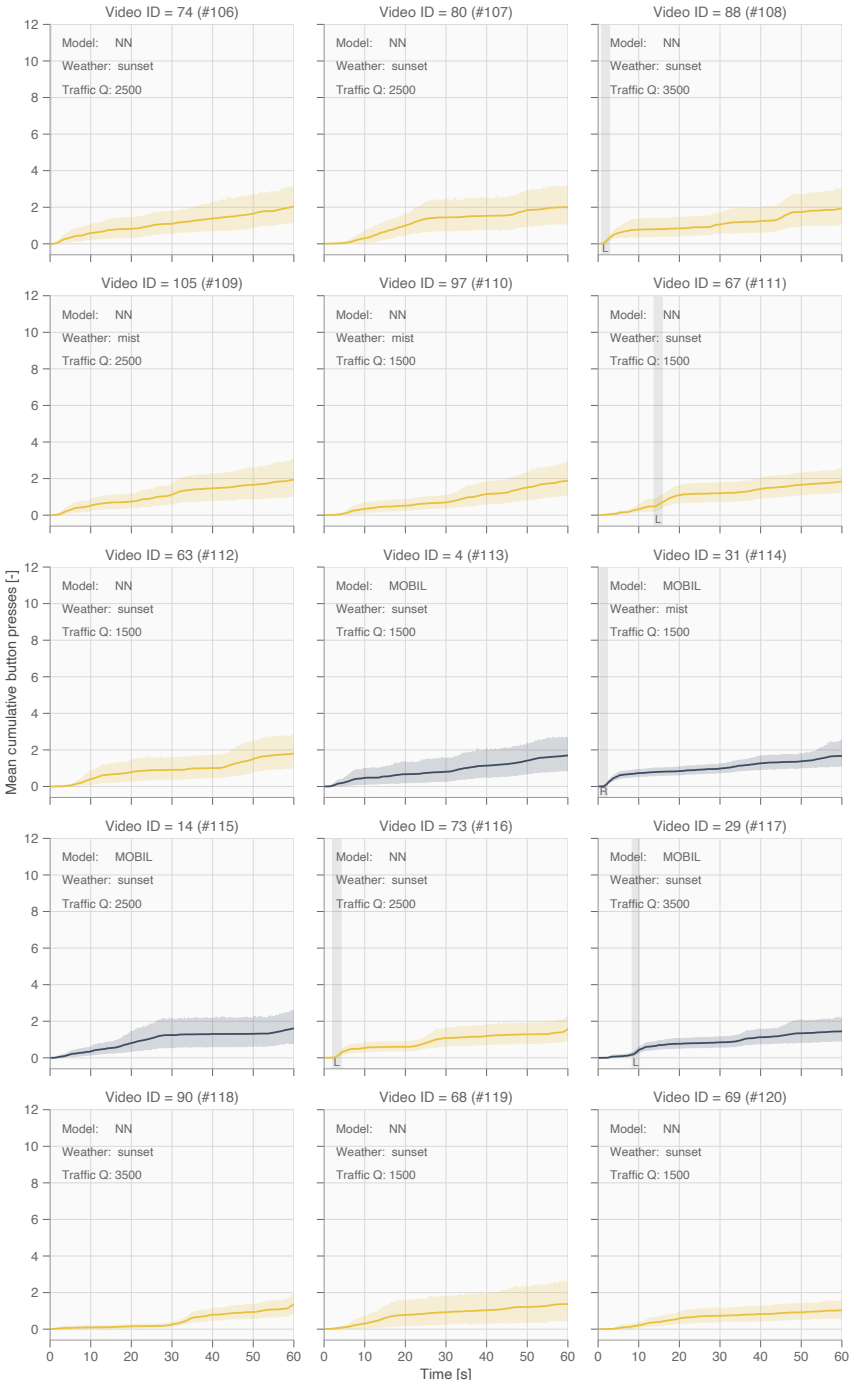


Figure B.8: Average cumulative button presses over time ranked 106 to 120

C APPEN QUESTIONNAIRE

Automated Lane Changes

Instructions ▲

You are invited to participate in a research study entitled "Automated lane changes". The study is being conducted by Daniel van der Haak, master student Mechanical Engineering at the Delft University of Technology, The Netherlands. He is supervised by Dr.ir. Joost de Winter and Dr. Pavlo Bazilinskyy of the Department of Cognitive Robotics, Delft University of Technology, The Netherlands. Contact: d.j.vandenhaak@student.tudelft.nl (mailto:d.j.vandenhaak@student.tudelft.nl).

The purpose of this research is to evaluate the overall acceptance of an automated vehicle system changing lanes on a highway. Your participation in this study may contribute to a better understanding of the acceptance of automated vehicles, and the creation of lane changing models.

You are free to contact the investigator at the above email address to ask questions about the study. You must be at least 18 years old to participate. The survey will take approximately 30 minutes of your time. In case you participated in a previous survey of one of the researchers of this study, your responses may be combined with the previous survey. The information collected in the survey is anonymous. Participants will not be personally identifiable in any research papers arising from this study. If you agree to participate and understand that your participation is voluntary, then continue. If you would not like to participate, then please close this page. Before the study starts, the images will be preloaded. This may take a few minutes depending on your Internet connection.

General questions

Have you read and understood the above instructions? (required)

- ○ Yes
- ○ No

What is your gender? (required)

- ○ Male
- ○ Female
- ○ I prefer not to respond

What is your age? (required)

In which type of place are you located now? (required)

- ○ Indoor, dark
- ○ Indoor, dim light
- ○ Indoor, bright light
- ○ Outdoor, dark

- ○ Outdoor, dim light
- ○ Outdoor, bright light
- ○ Other
- ○ I prefer not to respond

If you answered 'Other' in the previous question, please decribe the place where you located now below.

Which input device are you using now? (required)

- ○ Laptop keyboard
- ○ Desktop keyboard
- ○ Tablet on-screen keyboard
- ○ Mobile phone on-screen keyboard
- ○ Other
- ○ I prefer not to respond

If you answered 'Other' in the previous question, please decribe your input device below.

At which age did you obtain your first license for driving a car or motorcycle?

What is your primary mode of transportation (required)

- ○ Private vehicle
- ○ Public transportation
- ○ Motorcycle
- ○ Walking/Cycling
- ○ Other
- ○ I prefer not to respond

On average, how often did you drive a vehicle in the last 12 months? (required)

- ○ Every day
- ○ 4 to 6 days a week
- ○ 1 to 3 days a week
- ○ Once a month to once a week
- ○ Less than once a month
- ○ Never
- ○ I prefer not to respond

About how many kilometers (miles) did you drive in the last 12 months? (required)

- ○ 0 km / mi

- ○ 1 - 1,000 km (1 - 621 mi)
- ○ 1,001 - 5,000 km (622 - 3,107 mi)
- ○ 5,001 - 15,000 km (3,108 - 9,321 mi)
- ○ 15,001 - 20,000 km (9,322 - 12,427 mi)
- ○ 20,001 - 25,000 km (12,428 - 15,534 mi)
- ○ 25,001 - 35,000 km (15,535 - 21,748 mi)
- ○ 35,001 - 50,000 km (21,749 - 31,069 mi)
- ○ 50,001 - 100,000 km (31,070 - 62,137 mi)
- ○ More than 100,000 km (more than 62,137 mi)
- ○ I prefer not to respond

How many accidents were you involved in when driving a car in the last 3 years? (please include all accidents, regardless of how they were caused, how slight they were, or where they happened) (required)

- ○ 0
- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ More than 5
- ○ I prefer not to respond

How often do you do the following?: Becoming angered by a particular type of driver, and indicate your hostility by whatever means you can. (required)

- ○ 0 times per month
- ○ 1 to 3 times per month
- ○ 4 to 6 times per month
- ○ 7 to 9 times per month
- ○ 10 or more times per month
- ○ I prefer not to respond

How often do you do the following?: Disregarding the speed limit on a motorway. (required)

- ○ 0 times per month
- ○ 1 to 3 times per month
- ○ 4 to 6 times per month
- ○ 7 to 9 times per month
- ○ 10 or more times per month
- ○ I prefer not to respond

How often do you do the following?: Disregarding the speed limit on a residential road. (required)

- ○ 0 times per month
- ○ 1 to 3 times per month

- ○ 4 to 6 times per month
- ○ 7 to 9 times per month
- ○ 10 or more times per month
- ○ I prefer not to respond

How often do you do the following?: Driving so close to the car in front that it would be difficult to stop in an emergency. (required)

- ○ 0 times per month
- ○ 1 to 3 times per month
- ○ 4 to 6 times per month
- ○ 7 to 9 times per month
- ○ 10 or more times per month
- ○ I prefer not to respond

How often do you do the following?: Racing away from traffic lights with the intention of beating the driver next to you. (required)

- ○ 0 times per month
- ○ 1 to 3 times per month
- ○ 4 to 6 times per month
- ○ 7 to 9 times per month
- ○ 10 or more times per month
- ○ I prefer not to respond

How often do you do the following?: Sounding your horn to indicate your annoyance with another road user. (required)

- ○ 0 times per month
- ○ 1 to 3 times per month
- ○ 4 to 6 times per month
- ○ 7 to 9 times per month
- ○ 10 or more times per month
- ○ I prefer not to respond

How often do you do the following?: Using a mobile phone without a hands free kit. (required)

- ○ 0 times per month
- ○ 1 to 3 times per month
- ○ 4 to 6 times per month
- ○ 7 to 9 times per month
- ○ 10 or more times per month
- ○ I prefer not to respond

Experiment

You will be asked to leave Appen to participate in the rating task. You will need to open the link below. Do not close this tab. In the end of the experiment you will be given a code to input in the next

question on this tab. Please take a note of the code. Without the code, you will not be able to receive money for your participation. All videos will be downloaded before the start of the experiment. It may take a few minutes. Please do not close your browser during that time.

Open **this link (https://lane-change-crowdsourcing.herokuapp.com/)** to start experiment.

Type the code that you received at the end of the experiment. (required)

Miscellaneous questions

In which year do you think that most cars will be able to drive fully automatically in your country of residence? (required)

Please provide any suggestions that could help engineers to build safe and enjoyable automated cars.

Test Validators

/

TU Delft