DHM2020
L. Hanson et al. (Eds.)

doi:10.3233/ATDE200009

Learning Individual Driver's Mental Models Using POMDPs and BToM

Amena DARWISH [1] and H. Joe STEINHAUER [2]
University of Skövde, School of Informatics, 541 28 Skövde, Sweden

Abstract. Advanced driver assistant systems are supposed to assist the driver and ensure their safety while at the same time providing a fulfilling driving experience that suits their individual driving styles. What a driver will do in any given traffic situation depends on the driver's mental model which describes how the driver perceives the observable aspects of the environment, interprets these aspects, and on the driver's goals and beliefs of applicable actions for the current situation. Understanding the driver's mental model has hence received great attention from researchers, where defining the driver's beliefs and goals is one of the greatest challenges. In this paper we present an approach to establish individual drivers' temporal-spatial mental models by considering driving to be a continuous Partially Observable Markov Decision Process (POMDP) wherein the driver's mental model can be represented as a graph structure following the Bayesian Theory of Mind (BToM). The individual's mental model can then be automatically obtained through deep reinforcement learning. Using the driving simulator CARLA and deep Q-learning, we demonstrate our approach through the scenario of keeping the optimal time gap between the own vehicle and the vehicle in front.

Keywords. Driver Mental Model, Partially Observable Markov Decision Processes (POMDPs), Bayesian Theory of Mind (BToM), Reinforcement Learning

1. Introduction

Advanced driver assistance systems [1] (ADAS) are supposed to aid a car's driver to drive safely and comfortably in various driving situations. For example, the adaptive cruise control [1] (ACC) will dynamically adjust the vehicle's speed in order to keep a safe distance to the vehicle in front. However, human drivers have individual driving styles [2-4], which are based on their previous experience, cultural inheritance and personality traits and will also differ depending on the driving environment. Personalization of ADAS systems is thus of interest for car manufactures in order to enhance the individual's driving experience and satisfaction [5, 6]. With car sharing becoming more popular, the car's driver will change more frequently and a quick adaptation based on relatively little data is needed to personalize the driving experience and to ensure traffic safety.

In cases where the ADAS' actions are not to the individual driver's satisfaction [7], the driver might get annoyed and agitated which might negatively affect their driving capabilities and put them and other traffic participants at risk. Another risk is that the driver becomes preoccupied by trying to counteract the ADAS which takes away their

[1] Corresponding Author, Email: amena.darwish@his.se.
[2] Corresponding Author, Email: joe.steinhauer@his.se.

attention from the traffic situation, or that the driver might simply switch the system off and consequently also loses all its safety features [8, 9].

Hence, efforts should be made to build driver assistance systems that are capable of adapting to the individual driver's preferences within different road and traffic situations. This means that the ADAS should be able to learn the driving style and preferences of the individual driver and adapt to them as long as traffic safety is ensured. The challenges for this endeavor are manifold. Machine learning usually takes time and needs a lot of data, which is not an option when the situation requires that learning takes place online during actual driving in real traffic and driver satisfaction requires that the recognizable learning phase is kept to an absolute minimum.

In recent years, deep reinforcement learning has shown to achieve high performance in many different tasks, ranging from game playing [10, 11] to autonomous driving [12, 13]. The advantage of deep reinforcement learning is that the learner is able to learn from experience and can continue to learn while already in use. For the personalizable ADAS this would mean that on top of the normal ADAS functions, a learning module can be deployed that will, based on the driver's reaction to the ADAS' actions, learn how to better accommodate the driver.

In this paper we will illustrate this approach by focusing on one function within the ADAS, namely the adaptive cruise control (ACC) and the important scenario of keeping an optimal time gap between the own car (ego vehicle) and the car in front (lead vehicle). This distance is usually calculated with regard to the travel speed and given in seconds. A gap of two-three seconds is the usually recommended distance. This time gap is estimated to account for the reaction time of the ego driver to any changes in behavior of the lead vehicle. However, the actual distance that a driver prefers to keep will be influenced by their personality and different aspects of the environment. For example, preferences might depend on if the driving takes place in rural or urban areas, peak or off-peak traffic, dry or wet roads, night or day time conditions with street lights or in darkness, the driver's stress level, the reason for traveling (e.g. cruising for leisure or hurry to the hospital) etc.

In order to apply a time gap that safely accounts for the driver's reaction time, we need to take into account that today's cars free the driver of certain driving tasks, e.g. through dynamic cruise control and lane keeping. When less attention is needed to focus on the driving task a driver's attention might wander to other areas, such as selecting music, attending children, using the telephone, etc. This would call for an increased time gap [14]. However, the engaged and active driver who is fully focused on the traffic situation and, for example, planning to overtake the lead vehicle as soon as possible, might prefer and be safe with a shorter time gap.

The challenge that we will address within this work, is how the ACC can automatically learn the driver's preferences regarding the time gap to the lead vehicle. For this we consider driving to be a Partially Observable Markov Decision Process [15] (POMDP) wherein the individual's driving style depends on their individual mental model for driving. We regard this mental model to be a graph structure that can be represented by the Bayesian Theory of Mind (BToM) e.g. [16, 17]. This makes it possible that the individual's graph structure can then be automatically obtained through deep reinforcement learning. This approach has previously only been used on aggregated drivers' data using a Gaussian learning algorithm where it has led to promising results in mimicking realistic driver behavior in certain situations [18].

In this paper, we provide a novel approach of using deep Q-learning on individual driver's data to personalize the driving experience which we will demonstrate for the time gap scenario through implementation of an assistant for the ACC (the ACC assistant (ACCA)). The ACCA will based on the driver's reactions learn to suggest the adjustment of the time gap in accordance with driver preferences.

2. Method

We regard driving to be a Partially Observable Markov Decision Process (POMDP) as illustrated in Figure 1. It means that the environment, which at time t is in state s_t, will only be partially observed, these observations o_t, are described by the probability function $P(o_t|s_t)$ based on the observations o_t at time t, the driver makes a decision for an action a_t which will be carried out and leads to changes in the environment in the next time step (t+1) leaving the environment in state s_{t+1}. What impact the action a_t has on the environment is determined by another probability function $P(s_{t+1}|s_t, a_t)$.

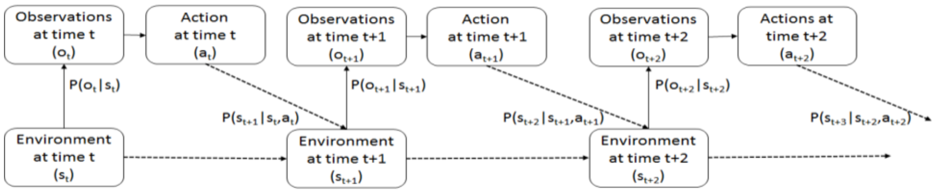


Figure 1. Driving illustrated as a Partially Observable Markov Decision Process (POMDP). The environment at time t is partially observable. Based on these observations o_t the driver will decide on an action a_t. The action will with probability $P(s_{t+1}|s_t, a_t)$ influence the next state of the environment s_{t+1}.

We argue that the driver's mental model that represents the grounds on which the driver is taking his or her decision for an action can be represented through a graph structure using the Bayesian Theory of Mind (BToM), which is commonly used to infer the mental reasoning and mental states of an observed agent by using probabilistic inverse planning [16, 18]. This graph structure captures the driver's knowledge about driving, their beliefs about what actions are applicable in what situation and their goals which will influence which of the applicable actions they will decide on in the given situation. Different individuals will have different graph structures which depend on their experiences, character traits, cultural background, etc. In our case, the observed agent is the vehicle's driver and the observer is the ADAS that needs to understand the driver's mental model in order to be of better assistance to them. In practice this means that the ADAS should be able to map any given state of the environment to the most likely action the driver will take. The "mind of the driver" as the ADAS sees it according to the Theory of Mind [17], denotes the belief that is recursively inferred using a probability distribution over the environment states.

$$b_{t+1} \propto P(o_{t+1}|\, s_{t+1})\, P(s_{t+1}|s_t\,, a_t)\;\, b_t \tag{1}$$

The belief b_{t+1} (at time point t+1) is calculated based on the probability of the observations o_{t+1} given the state of the environment s_{t+1} and on the belief b_t that with probability $P(s_{t+1}|s_t\,, a_t)$ that this state s_{t+1} will have occurred given that action a_t has been executed in the previous state s_t .

Based on the Bellman equation [19] the formula for the quality Q of a state-action pair can be calculated as by [20]:

$$Q\left(s_t, a_t\right) = \alpha\left(r\left(s_t, a_t\right) + \gamma \max_a Q\left(s_{t+1}, a\right)\right) \tag{2}$$

Where the quality, Q of applying action a_t in state s_t is calculated recursively by the sum of the immediate reward $r\left(s_t, a_t\right)$ and the maximum reward that can be obtained in the future over action space $\max_a Q\left(s_{t+1}, a\right)$. α Defines the learning rate and γ represents the discounting factor for the importance of rewards achievable in the future.

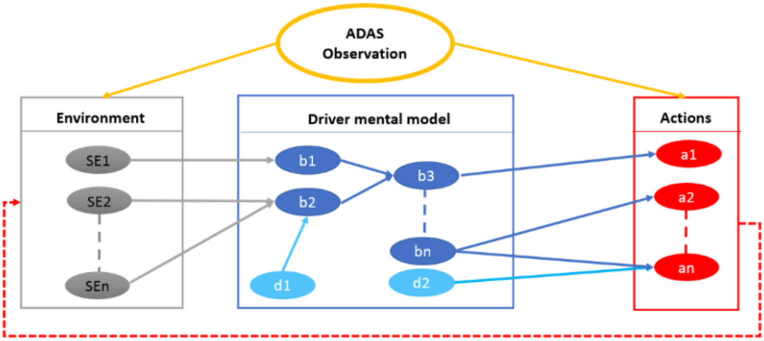


Figure 2. The driver's mental model expressed through the Bayesian Theory of Mind (BToM) (in blue), the input to the mental model will be the environment that can be represented through all observable elements (SEs) in the environment (in grey). The output from the mental model is the driver's action which is one of many possible actions to choose from (in red). The action will influence the environment (represented through the dashed line). The combination of these three results in a structure similar to a deep artificial neural network (ANN). The ADAS is represented as the observer who can observe the observable elements in the envoronment as well as the driver's reaction, but to whom the driver's mental model is a black box.

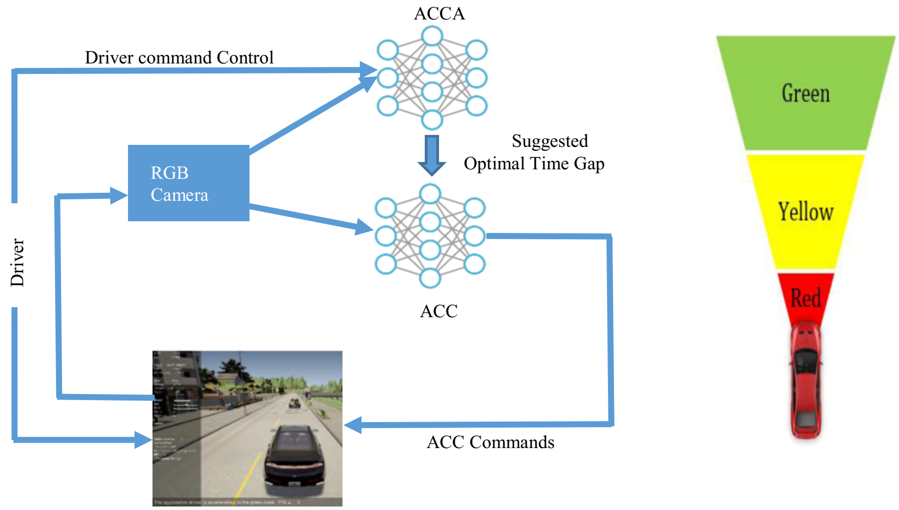
To update the beliefs we will apply deep Q-learning [21, 22] and regard the graph structure of the mental model to be the hidden layers of a deep ANN as shown (in blue) in Figure 2. The input to this network is the state of the environment, provided by an RGB camera image and the output is the action that the driver is most likely to take in this state. The individual's mental model will hence be expressed by the weights assigned to the connections between the nodes in the network, which will be learned through deep reinforcement learning.

Using an ANN in that way to approximate the POMDPs output is a common approach [23]. Here we use TensorFlow to implement equation (2) and use two different deep Q-learning networks, one to obtain $Q(s_t, a_t)$ and $Q(s_{t+1}, a)$ over the action space, respectively. For the implementation we use the Xception model [24] provided in Keras using RGB camera images with 720x1280 pixels as input. For the simulation we use CARLA (CAR Learn to Act) which is an open source simulator for autonomous driving [25]. Specifically, we use Scenario Runner [26] v.0.9.8, which also is an open source repository containing traffic scenario definitions and an execution engine for CARLA. In CARLA we can simulate the driving environment, choose from different weather and lightning conditions, and include other traffic participants. CARLA provides different sensors for the car, among them the RGB camera images (720x1280 pixels), that will provide the input to the deep Q-learning algorithm.

Initially, our task is to personalize the ACC to the individual driver's preferences, which will be learned from the driver's reactions to the ACC's driving. CARLA, however, does not provide ACC functions that we could build upon. Hence, in order to build an ACC we use the deep reinforcement learning approach for deep Q-learning inspired by [27] and [28, 29]. Once the ACC is in place it will be set to drive the car and a program mimicking an individual driver will intervene through either breaking or accelerating whenever the time gap is not to the driver's liking.

The ACCA that learns the individual driving style, gets as input the environment through the RGB camera image as well as the actions that the ACC performs on this input together with the driver's feedback. The feedback is coupled to the driver's actions, which will be to accelerate if the time gap that is kept by the ACC is too wide, or to break if the time gap is considered to be too small. The ACCA will, based on this, learn the driver's individual preferences. As output it will provide suggestions to the ACC on how to adjust the time gap.

The ACC itself has learned to always follow the lead vehicle in a safe distance, through keeping it in the green zone (see the illustration in Figure 3b) but when the personalized suggestions from the ACCA are added, the ACC will act according to these suggestions instead. This can result in reducing the time gap to an extent where the lead vehicle will be allowed in the yellow zone just adjacent to the red zone. Note, however, that it is not possible for the ACCA to suggest a time gap that would allow the lead vehicle into the red zone.



(a) (b)

Figure 3. (a) After the system is trained, the RGB image is fed to the ACC and the ACCA alike, the ACCA knows the current time gap (which is part of the environment) and suggests an adjustment to the ACC. The ACC will adjust the time gap accordingly. Note, the ACCA cannot suggest time gaps that are outside safety parameters. (b) An illustration of the different time-gap zones considered in our approach. When the lead vehicle is within the green zone the time gap is considered to be long enough for safe following in all circumstance, when the lead vehicle is in the yellow zone the driver should be actively engaged in driving. The lead vehicle is not allowed in the red zone at all. If the time gap to the lead vehicle becomes larger than the green zone it will be considered to be free driving.

3. Results

The training takes place on the Town01 map in CARLA. The initial distance between ego vehicle and lead vehicle is set to 15 m, the lead vehicle travels with 8 m/s and the initial time gap is set to 5 seconds. We train for 100 episodes; each episode is comprised of 15 steps, unless calculations within steps take unusually long and the episode is ended through a time out after 30 seconds. During training, the ACC learns how to maximize the cumulative sum of rewards (return) over episodes. It receives -200 for a collision with the lead vehicle, +10 when the lead vehicle is kept in the green zone, and -10 if the lead vehicle has entered the red zone. A similar approach has been used by [27] and is formally expressed in equation (3).

$$r = 200\, r_{collision} + 10\, r_{greenzone} + r_{yellowzone} + 10\, r_{redzone} + 0.2\, v_{log}^2 + 0.1 \qquad (3)$$

Where $r_{collision}$ = -1, $r_{colorzone}$ = +1 (within the yellow and green zone) and $r_{colorzone}$ = -1 within the red zone, v_{log} represents the longitudinal speed of the ego vehicle (measured in m/s) and 0.2* v_{log}^2 is needed to prevent the ego vehicle from driving unnecessarily slow, the last summand (0.1) is used to prevent the car from stopping.

How quickly and accurately the algorithm will learn depends, among others, also on the quality of input from the environment. We expect the learning to take longer when sight is poor and hence have chosen to use two different environmental conditions that are available in CARLA. One represents daytime conditions with clear weather, the other sunset illumination on wet roads. The graphs, presented in Figure 4, show that the algorithm learns how to keep an appropriate time gap. They also show, as expected, that the weather condition does influence the learning based on the RGB camera input and that in clear weather conditions a better outcome is received.

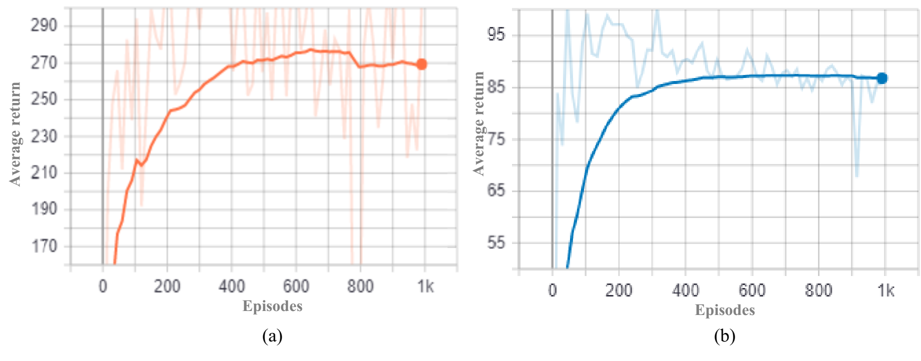


Figure 4. (a) Average return for the ACC in clear weather. (b) Average return for the ACC in wet sunset weather.

For the ACCA, the deep Q-learning algorithm can choose between three different actions that it will suggest to the ACC: increase the time gap by 0.2 s, decrease the gap by 0.2 s (as long as the new time gap will still be within safety parameters) or keep the time gap as it is. During learning the reward for the ACCA is defined as presented in equation (4).

$$r = 200\, r_{collision} - \sum_{zone=1}^{2} r_{colorzone}\, r_{driver} \left| r_{diff} \right| + 0.2\, v_{log} + 0.1 \qquad (4)$$

With $r_{collision}$ = -1, $r_{colorzone}$ = 1 if the ego vehicle is in the respective color zone, otherwise $r_{colorzone}$ = 0, r_{driver} = 1 when the driver controls the ego vehicle otherwise r_{driver} = 0, r_{diff} represents the difference between the recommended time gap and the predicted optimal time gap, unless there is no difference then r_{diff} =10. v_{log} is the longitudinal speed of the ego vehicle measured in m/s, and also here 0.2* v_{log} is needed to prevent the ego vehicle from being unnecessarily slow as well as the last summand (0.1) is needed to prevent the car from stopping. This assures that the ACCA tries to maximize the return for each episode by keeping the time gap to the driver's satisfaction and does not crash into the lead vehicle.

We test our approach for two different driving styles, one describing a conservative driver who wants to follow the lead vehicle at a distance that in all circumstances would be considered to be safe and would place the lead vehicle into the green zone (illustrated in Figure 3 (b)); the other describing a more aggressive driver who wants to be as close to the lead vehicle as possible which would bring it within the red zone.

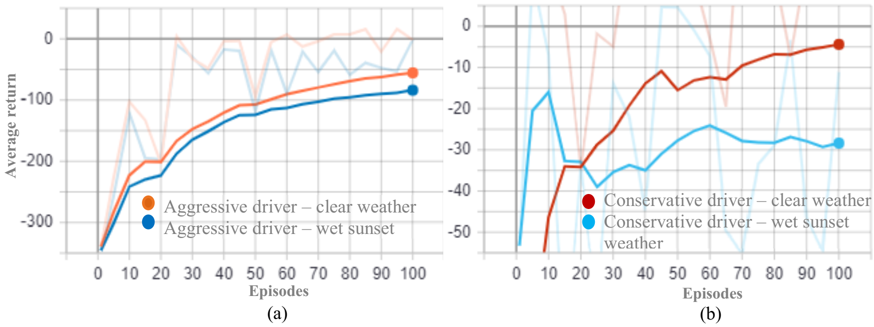


Figure 5. (a) Average return for the aggressive diver. (b) Average return for the conservative driver.

The results presented in Figure 5 show also here that the weather conditions have a significant influence when it comes to the conservative driver. They also show that it is in general harder for the ACCA to please the aggressive driver, due to the fact that the aggressive driver would want to follow the lead vehicle too closely which is not possible with the given safety parameters. However, the weather seems not to influence the learning as much for the aggressive driver. This gives rise to the speculation the ACCA learns just to keep the gap as short as possible and is not analyzing the RGB images of the environment as well as it does for the conservative driver.

To validate our results, we use the success rate that is commonly used in games and robotic domains [30] and describes how many times the agent is able to reach the goal. Here the goal is to suggest a time gap that is accepted by the driver where acceptance means that the driver will not adjust the gap through accelerating or breaking manually during the considered time interval. Table 1 shows the success rate for both our driver types for the clear weather condition.

Table 1. Success and fail rate for the aggressive and the conservative driver and the average time gap in all cases respectively as well as the difference in time gap (in seconds) between average success and average fail for the clear weather condition.

Driver style	Success rate	Average time gap when successful	Fail rate	Average time gap when failed	Difference in time gap
Aggressive	61.91%	1.25	38.09%	2.57	**1.32**
Conservative	63.46%	4.31	36.54%	6.07	**1.76**

4. Discussion

In this study we looked at driving from the perspective of the ADAS, which is supposed to predict the driver's actions from observations of the environment and the driver. The main contribution in our work is hence the approach to use POMDPs wherein the driver's mental model can be represented as a graph structure that follows the BToM and can, together with the input from the environment and the output (action) that the driver choses, be regarded as a deep ANN. For this deep ANN the weights that account for the individual driver's mental model can be learned by deep reinforcement learning (in this case deep Q-learning). Note, that the novelty of our approach lies in the combination of these techniques and their application to personalize the driving experience for the individual driver.

From an applicational point of view the personalization of the driving experience, where the ADAS learns the preferences of the individual driver while at the same time ensuring traffic safety is an important task for the automobile industry. Achieving personalization requires data-driven approaches and requires that the system is able to adapt quickly to the driver's needs and habits and keeps adapting while the driver's behavior and preferences evolve as well over time and with experience [5, 6]. Hence, the system should be able to adapt in real time which means on the bases of very little data. This exceeds previous work done for example for aircraft autopilots e.g. [31] or to model human driving on a general, not individual, level [32]. Our work describes personalization for the relatively simple driving scenario of following a lead vehicle. In reality the system should be able to adapt to many different scenarios in parallel (e.g. overtaking, turning, approaching traffic light, approaching zebra crossing).

So far, validation of the system has been done by looking at the success rate. Further research is needed to see if this can be further improved when using a different reward function. Once the system is tested within a real car and against a human driver further validation will be done by the drivers through questionnaires about the driving experience and satisfactions.

Future work will include to broaden the driving scenario where individual preferences are also dependent on the type of road, the amount of surrounding traffic, and the in-car situation. There is increasing work on autonomously driving cars, several of which are implemented in CARLA with very good results (see for example [27]). Approaches similar to them could replace the ACC in our example and represent a complex driver assistant system that then will, following the approach presented here, become personalizable through deep Q-learning.

Acknowledgement

This work has been supported by VINNOVA, the Swedish Government Agency for Innovation Systems, proj. "Intention Recognition for Real-time Automotive 3D situation awareness (IRRA)", in the funding program FFI: Strategic Vehicle Research and Innovation (DNR 2018-05012)

References

[1] American Automobile Association. Advanced driver assistance technology names: AAA's recommendation for common naming of advanced safety systems. AAA News Room, January, 2019.
[2] Witt M, Wang L, Fahrenkrog F, Kompaß K, Prokop G. Cognitive Driver Behavior Modeling: Influence of Personality and Driver Characteristics on Driver Behavior. In: Stanton N, editor. Advances in Human Aspects of Transportation. AHFE 2018, 751-763. Springer, Cham.
[3] Bellet T, Bornard JC, Richard B, Laverdure. Use of a cognitive simulation model of the driver to support the Virtual Human Centred Design (V-HCD) of ADAS and automated vehicles, 2018. In: Graz Symposium Virtual Vehicle: Artifical Intelligence Meets Model-Centric Design.
[4] Chen H, Zhao F, Huang K, Tian Y. Driver behavior analysis for advanced driver assistance system. In: 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), 2018, May, 492-497. IEEE.
[5] Hasenjäger M, Heckmann M, Wersing H. A Survey of Personalization for Advanced Driver Assistance Systems. IEEE Transactions on Intelligent Vehicles, 2019, 5(2), 335-344.
[6] Yi D, Su J, Hu L, Liu C, Quddus MA, Dianati M, Chen WH. Implicit personalization in driving assistance: State-of-the-art and open issues. IEEE Transactions on Intelligent Vehicles, 2019.
[7] Strand N, Nilsson J, Karlsson IC, Nilsson L. Interaction with and use of driver assistance systems: a study of end-user experiences. In: Proceedings of the 18th ITS World Congress. Washington, DC: ITS America, 2011.
[8] Aziz T, Horiguchi Y, Sawaragi T. Towards understanding the development of driver's mental model of a Lane Departure Warning system while driving. Journal of Advanced Mechanical Design, Systems, and Manufacturing, 2014, 8(3), JAMDSM0032-.
[9] Heikoop DD, de Winter JC, van Arem B, Stanton NA. Psychological constructs in driving automation: a consensus model and critical comment on construct proliferation. Theoretical Issues in Ergonomics Science, 2016, 17(3), 284-303.
[10] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013, Dec 19.
[11] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S. Human-level control through deep reinforcement learning. Nature, 2015, 518, 529-533.
[12] Kiran BR, Sobh I, Talpaert V, Mannion P, Sallab AA, Yogamani S, Pérez P. Deep reinforcement learning for autonomous driving: A survey. arXiv preprint arXiv:2002.00444, 2020, Feb 2.
[13] Sallab AE, Abdou M, Perot E, Yogamani S. Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017, (19), 70-76.
[14] Salvucci DD. Predicting the effects of in-car interface use on driver performance: An integrated model approach. International Journal of Human-Computer Studies, 2001, 55(1), 85-107.
[15] Bai H, Cai S, Ye N, Hsu D, Lee WS. Intention-aware online POMDP planning for autonomous driving in a crowd. In: 2015 IEEE international conference on robotics and automation (icra), 2015, May, 454-460. IEEE.
[16] Sukthankar G, Geib C, Bui HH, Pynadath D, Goldman RP, editors. Plan, activity, and intent recognition: Theory and practice. Newnes, 2014, Mar 3.
[17] Doherty M. Theory of mind: How children understand others' thoughts and feelings. Psychology press, 2008, Aug 26.
[18] Eilers M, Möbus C, Tango F, Pietquin O. The learning of longitudinal human driving behavior and driver assistance strategies. Transportation research part F: traffic psychology and behavior, 2013, 21, 295-314.
[19] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press, 2018, Oct 19.
[20] Stekolshchi, R. How does the Bellman equation work in Deep RL? 2020, Towards Data Science.
[21] Hausknecht M, Stone P. Deep recurrent q-learning for partially observable mdps. In: 2015 AAAI Fall Symposium Series, 2015, Sep 23.
[22] Egorov M. Deep reinforcement learning with pomdps, 2015.

[23] Hauskrecht M. Value-function approximations for partially observable Markov decision processes. Journal of artificial intelligence research, 2000, 13, 33-94.
[24] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, 1251-1258.
[25] Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V. CARLA: An open urban driving simulator. arXiv preprint arXiv:1711.03938, 2017, Nov 10.
[26] GitHub. 2020. Carla-Simulator/Scenario_Runner. [online] Available at: <https://github.com/carla-simulator/scenario_runner> [Accessed 13 May 2020].
[27] Chen J, Li SE, Tomizuka M. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. arXiv preprint arXiv:2001.08726, 2020, Jan 23.
[28] Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang J, Zhang X. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016, Apr 25.
[29] Dharmawan W, Nambo H. End-to-End Xception Model Implementation on Carla Self Driving Car in Moderate Dense Environment. In: Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference, 2019, Dec 21, 139-143.
[30] Zhang F, Leitner J, Milford M, Upcroft B, Corke P. Towards vision-based deep reinforcement learning for robotic motion control. arXiv preprint arXiv:1511.03791, 2015, Nov 12.
[31] Wickens CD, McCarley JS, Alexander AL, Thomas LC, Ambinder M, Zheng S. Attention-situation awareness (A-SA) model of pilot error. Human performance modeling in aviation, 2008, 213-39.
[32] Bellet T, Bailly-Asuni B, Mayenobe P, Banet A. A theoretical and methodological framework for studying and modelling drivers' mental representations. Safety Science, 2009, 47(9), 1205-1221.