A driving profile recommender system for autonomous driving using sensor data and reinforcement learning

Christos Chronis
Harokopio University of Athens
Athens, Greece
chronis@hua.gr

Christos Sardianos
Harokopio University of Athens
Athens, Greece
sardianos@hua.gr

Iraklis Varlamis
Harokopio University of Athens
Athens, Greece
varlamis@hua.gr

Dimitrios Michail
Harokopio University of Athens
Athens, Greece
michail@hua.gr

Konstantinos Tserpes
Harokopio University of Athens
Athens, Greece
tserpes@hua.gr

ABSTRACT

The design of algorithms for autonomous vehicles includes a wide range of machine learning tasks including scene perception by the visual input from cameras and other sensors, monitoring and prediction of the driver and passengers' state, and others. The aim of the present work is to study the task of personalizing the driving experience in an autonomous vehicle, taking into account the particularities and differences of each person in how he/she perceives the vehicle's velocity. For this purpose, we employ the Actor-Critic Reinforcement Learning technique in order to automatically select the best driving mode during driving. The input to the actor-critic model comprises the driver's stress and excitement, which are affected by the route conditions, and the vehicle velocity and angular velocity. The output at each step is the best mode for each driver, which better balances stress, excitement, and route completion time. The whole setup is simulated and tested within the Carla open-source simulator for autonomous driving research.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Personalization**.

KEYWORDS

personalization, autonomous driving, reinforcement learning, simulation environment

ACM Reference Format:
Christos Chronis, Christos Sardianos, Iraklis Varlamis, Dimitrios Michail, and Konstantinos Tserpes. 2021. A driving profile recommender system for autonomous driving using sensor data and reinforcement learning. In *25th Pan-Hellenic Conference on Informatics (PCI 2021), November 26–28, 2021, Volos, Greece*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3503823.3503830

PCI 2021, November 26–28, 2021, Volos, Greece

ACM ISBN 978-1-4503-9555-7/21/11…$15.00
https://doi.org/10.1145/3503823.3503830

1 INTRODUCTION

The last few years we are witnessing the rapid development and adoption of electric vehicles in our daily lives and in parallel to this development, the quick rise of interest for autonomous driving. Companies like Tesla and Waymo have shown some really impressive examples of autonomous driving in real road conditions. Waymo has already created a fleet of vehicles that operate as taxis in some US cities [14], WeRide did the same in Guangzhou, China [4] and Tesla has announced that is launching a similar service soon.

For increasing trust to autonomous vehicles and boosting their public acceptance, researchers and companies are developing new models, methods and solution, which are tested using different forms of experimentation such as on-road, on a simulated test bed, in a living lab, etc. Since the financial and organisational cost for performing experiments on living labs and on real roads are high, researchers usually rely on simulation software in order to train, fine-tune and evaluate their models. The key factor in the full adoption of autonomous vehicles are humans. Humans however are a subjective factor since their desire, mood, age and of course psychological and physical condition can affect the final experience from the use of autonomous vehicles. It is thus important to monitor human state, along with the vehicle and environment condition and consequently adapt the autonomous vehicle behavior to human preferences. The key enabler to this behaviour personalisation is the proper detection of the driver state, which can be expressed as stress, excitement, etc. In all cases, this has to be combined with the proper interpretation of traffic rules in order to guarantee the safe operation of the vehicle and the correct management of the low level controllers of the vehicle that operate the acceleration pedal, the breaks and the steering wheel.

In this work, we focus on the personalisation aspect of autonomous driving, assuming that the autopilot of the vehicle takes all the necessary decisions that relate to car navigation and traffic rules. As a result, the proposed personalisation approach takes control only of the vehicle acceleration/deceleration. In order to implement the personalisation of the autonomous vehicle behavior, we assume a modular pipeline in which an autopilot is responsible for setting the parameters of the vehicle controller and more than one autopilots are available, corresponding to different modes of driving the vehicle (e.g. aggressively, conservatively, etc). In our experimental evaluation, we rely on the driving simulation environment of Carla simulator [1], which offers a long list of vehicle sensors and controllers, information about the vehicle route, and a configurable autopilot that can simulate different driving profiles. The driving mode personalisation employs an actor-critic reinforcement learning model, which learns to choose the best driving mode (action) for each driver during the execution of the route, based on the road's and driver's condition (state). The extensive evaluation performed in the Carla simulation environment demonstrates that having a module that dynamically changes driving modes based on the driver's condition and the vehicle's state and environment, results in lower stress and higher excitement for different driver profiles.

Section 2 that follows highlights the main research works on the personalisation of autonomous driving, and briefly introduces our choices for this task. Section 3 introduces the simulation environment employed in this study. Section 4 provides the implementation details with emphasis on the deep learning architectures that have been employed. and summarizes the results we achieved and Section 6 concludes the paper.

2 RELATED WORK

The initial thought behind the personalisation of autonomous driving is that a driver feels more comfortable in a vehicle that moves in a similar way to the one he usually drives. However, researches in simulated environments have shown that this only applies to younger individuals, whereas people over 65 do not feel as comfortable when they experience the way they were driving [5, 12].

A different study that used a simulator [2] found that people tend to choose a driving profile, which they believe matches their driving style, but which is usually less aggressive than their actual style.

The empirical results highlight the difficulty of finding the optimal driving profile for each driver, and the various subjective factors that emanate from human nature make it difficult for an autonomous vehicle to learn the driving style. They also raise the need for further intervention so that the driving style of the autonomous vehicle to always adapt to the driving conditions and the state of the driver. This is in agreement to the definition of personalisation provided by Adomavicius & Tuzhilin [1], who define it as an endless cyclical process. In the context of autonomous driving, this process consists of i) understanding the driver (e.g. by observing the driving behavior), ii) enabling the driver to customize the vehicle's functions and finally, iii) measuring the impact of the current vehicle behavior to the driver, so that it can adapted as needed.

Nava et al. [10] were the first to point out the need for continuous on-line learning with the aim of personalizing ACC (adaptive cruise control), and employed a Reinforcement Learning technique to form a personalized model, which learns by interacting with the environment.

The two machine learning alternatives that can be employed for the automatic control of autonomous vehicles are the *modular pipeline* and the *end-to-end learning* [7].

[1]https://carla.org/

The modular pipeline offers complete control and understanding of why the vehicle exhibits a particular behavior, since each module (e.g. low-level perception, scene parsing, path planning, behavior controller) in the pipeline is trained and evaluated individually. This specialization allows more targeted interventions on the tasks that demonstrate unexpected behaviors, without interfering with critical systems of the vehicle that can cause safety complications (i.e. the vehicle control system). Reinforcement Learning (RL) is a popular technique for training ML models for playing video games [9], where the agent learns to solve the problem in an optimal way, after a large number of repetitions. RL training requires a reliable reward system, a lot of learning time and can not be applied in real conditions, but only in simulations since it learns by making mistakes, which can be fatal in real conditions. The main competitor of RL in autonomous driving, Imitation Learning, trains the model by observing the human driver and imitating its behavior [3]. The major disadvantage of imitation learning is the difficulty of model generalisation, since models trained on a highway, will not behave well within the city limits or on a country road, and even different weather or lighting conditions may affect its performance.

In the absence of a living lab, we did all the development and evaluation in a simulated environment. So, we rely on reinforcement learning to train a model that dynamically adapts the vehicle's driving mode according to the driver's and vehicle's conditions. More specifically, we employ a Model-Free approach which is more appropriate for the very complex and highly interactive driving task [8] than the Model-Based one. We use a Policy Optimization technique, the Actor-Critic one, which has been used by many researchers in autonomous driving [6, 11, 13].

3 THE SIMULATION ENVIRONMENT

All the experiments have been performed within the Carla driving simulation environment, which allows to quickly create different scenarios, have access to all the vehicle data, and easily attach specialized sensors and cameras to the vehicle and collect more data. Carla offers a highly parametric behavior agent, the autopilot, which can drive the vehicle within the simulated environment, paying respect to all the road safety rules. The parametric autopilot allows to simulate different driving behaviors (driving modes), which correspond to how different drivers behave in various traffic conditions. Carla offers different driving environments, from city road networks with crossroads and traffic lights, to rural roads and highways. Pedestrians, vehicles, traffic lights, speed limits, etc., can be added to the world to provide a more realistic setup for the experiments, or removed completely for simplicity.

3.1 Vehicle state monitoring, scene perception and the autopilot

Carla offers a wide range of sensors that can provide useful feedback concerning the vehicle state. In addition, the information about a route (i.e. the waypoints it contains) is always available through the autopilot. The visualisation of the forthcoming route segment, is used as an additional input to the personalisation module, as shown in Figure 1. The Proportional Integral Drivative (PID) controller that is used to control steering and velocity, can be configured to have specific speed and steering limits. In order to simulate different driving behaviors, we configure more than one autopilots, each one having different PID controller limits, thus resulting in a less or more aggressive driving behavior.

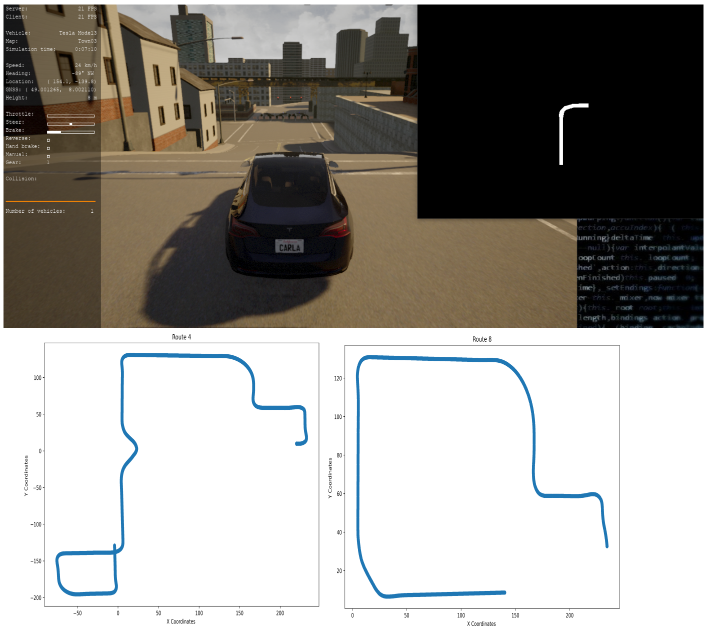

Figure 1: A visualization of the forthcoming vehicle route segment in Carla (left) and two sample routes (right), the complex route 4 and the simple route 8.

3.2 Driver state monitoring

Two main factors that affect the driver's experience from autonomous driving are stress and excitement. Stress increases when drivers are exposed to risky and stressing situations and in general decreases when the vehicle is moving at lower speeds. Excitement has been found to play a very important role in the driving experience, and is on the opposite edge of stress, since if the vehicle slows down to reduce stress, and keeps moving at very slow speeds for a long time, this can reduce the driver's excitement, making him/her bored and in some cases brink drowsiness and sleepiness [15]. In live-labs or real-world experiments, it would be possible to use galvanic sensors for measuring sweating, heart rate sensors, and questionnaires, in order to monitor the driver's state. However, the simulation environment does not provide any method for measuring the driver's stress and excitement, so we decided to simulate driver's stress and excitement based on measurements of speed, acceleration, angular velocity, speed limits, number of violations, etc. We developed three types of drivers ('easily stressed', 'normal', and 'resilient') that have different limits of sensitivity to the velocity and angular velocity of the vehicle. Their stress and excitement level in relation to the vehicle's velocity and angular velocity are simulated using sigmoid and skewed normal curves as depicted in Figure 2.

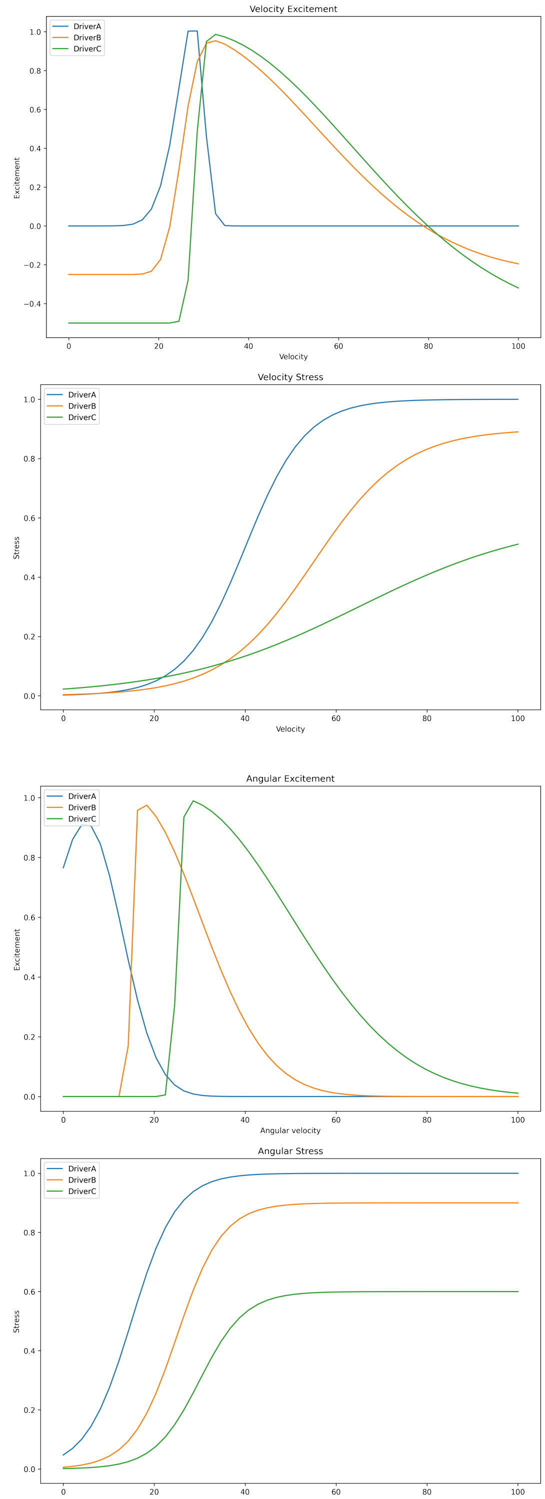

Figure 2: The simulated curves of driver stress and excitement of the three sample drivers A, B, C.

4 THE PROPOSED APPROACH

4.1 The Actor Critic algorithm

The Actor Critic algorithm [16] used in this study consists of two neural networks, the actor and the critic. The actor network is based on the policy-method and is responsible for the actions of the agent in the dynamic environment, while the critic network, which is value-based, is responsible for judging how good were the actions that came from the decision of the actor network. The actor network maps each state to a corresponding action by giving as output a probability distribution corresponding to each action. The critic network maps each state to its corresponding Q-value, which represents the quality (value) of the state. The weights of the two neural networks are updated in order to maximize an objective function J_θ whose gradient is defined as follows:

$$\nabla_\theta J_\theta = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{T}\nabla_\theta log\pi_\theta(\alpha_t|s_t)Q(s_t,\alpha_t) \tag{1}$$

where $J(\theta)$ is the objective function that depends on θ, m is the number of episodes executed, π is a policy parametrized by θ, which means that when θ varies the policy will be affected. The actor critic technique updates weights at every step and not at the end of the episode, so α_t is the action taken at step t and s_t is the state at the same step. $Q(s_t,\alpha_t)$ is the quality of the action α_t and $\pi_\theta(\alpha_t|s_t)$ is the probability that defines the action at step t. As shown in Equation 1 the updates take into account the probability of the action as defined by the actor, and the quality and value of the state as defined by the critic network.

4.2 The implementation of the Actor Critic neural network

The sensors used as input to the Actor Critic model at each step are the velocity and angular velocity of the vehicle, the vertical acceleration, the speed limit, the driver's stress and excitement scores as given by the equations depicted in Figure 2. In addition, we feed the black and white image depicting the forthcoming segment of the vehicle route (as shown in Figure 1).

The extended network architecture is depicted in Figure 3. The left output layer, which corresponds to the Actor takes the input and generates two probabilities that in our case correspond to choosing a less or more aggressive driving mode respectively. The right output layer of the Actor Critic model corresponds to the Critic and outputs a single value that sums up all the expected future rewards from the action. Using the actor output probabilities (and the current driving mode) we choose a driving mode each time. During training, the action is chosen at random based on the two probabilities (which sum up to 1) whereas during inference, the action with the highest probability is chosen. The quality of an action is evaluated using a reward function that jointly checks the driver's excitement and stress as a result of this action. Since the objective is to keep driver's stress low and driver's excitement high, we reward a step with 1 if the stress is low and excitement is high. We also penalize the agent when we have a crash or the vehicle fails to terminate the route on time. The objective is to maximize the reward at the end of each episode. The actions and rewards at each step are collected in a buffer and are used to update the network weights in a batch, when the episode is completed. Tensorflow 2.0 and Keras 2.0 have been employed for creating the network. We employed the log of probability of the action multiplied by the difference of the total reward minus the action reward, as a loss function for the actor model, and the Huber loss as the loss function for the critic model. In back-propagation we employed the sum of the two loss functions and Adam optimizer with learning rate 1e-3.

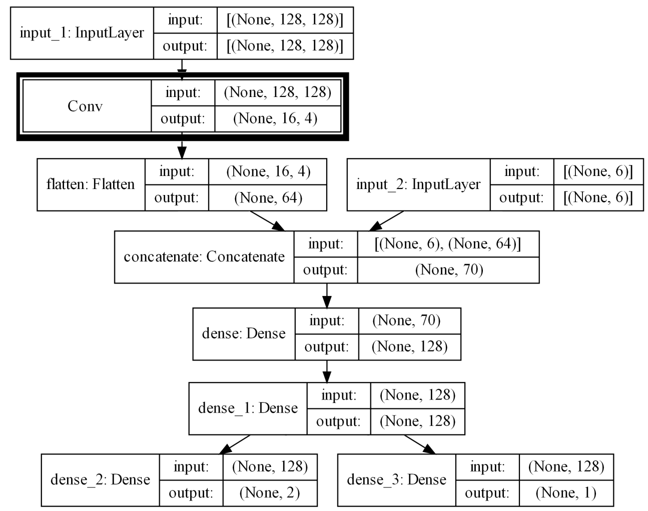

Figure 3: The architecture of the Actor Critic model, along with the additional layers (on the left branch) for handling the image that depicts the forthcoming section of the route.

5 EXPERIMENTAL SETUP AND RESULTS

In order to evaluate our personalisation algorithm, we employed Town 3 in Carla simulator, which contains a lot of turns, wide and narrower streets in an urban setup, as well as a circular highway around the town. This allows the model to be tested in varying route conditions where straight lines interchange with 45 to 180 degree turns, as shown in the two sample routes of Figure 1. We employed the three driver profiles (A, B, C), who get stressed in a different way (easily stressed, normally stressed, low stress) and excited (low excitement, medium excitement, high excitement) from high velocities and angular velocities. We also define three autopilot setups (AP1, AP2, AP3) that correspond to different limits in the PID controller and consequently to three different driving modes (conservative, normal, aggressive). For the training of the RL model we assume that in each episode the vehicle is performing a random route within the town. During the route we change the driving mode based on the actor-critic model probabilities. This results to models that in general choose the conservative mode during turns and the aggressive mode in the straights and highways, of course with a different mix, depending on the driver profile. The final RL model has been trained for several hours on an AMD Ryzen 9 5900X (12 cores) CPU with 32 RAM, using a RTX 3070 OC 8GB graphics card. All experiments have been performed in real-time in Carla, so the training time corresponds to real driving time and the training was interrupted when there was no improvement

Table 1: Results on different routes, using a single driver mode for each driver (manual) or by changing profiles during the route (RL).

route #	mode	stress			excitement			route completion time			stress out of limits		
		A	B	C	A	B	C	A	B	C	A	B	C
1	manual	0.10	0.10	0.08	0.41	0.49	0.50	149.53	86.19	86.18	1341	794	609
1	RL	0.17	0.06	0.05	0.58	0.33	0.15	104.77	102.94	102.26	322	59	13
2	manual	0.09	0.09	0.07	0.41	0.48	0.48	189.13	108.94	108.91	390	256	205
2	RL	0.16	0.06	0.06	0.61	0.3	0.16	131.56	129.59	128.88	309	128	87
3	manual	0.08	0.08	0.07	0.41	0.5	0.46	169.55	96.95	97.03	275	193	139
3	RL	0.14	0.05	0.05	0.58	0.25	0.12	119.37	117.97	117.39	210	71	43
4	manual	0.12	0.13	0.09	0.39	0.53	0.52	173.83	100.63	100.7	814	525	354
4	RL	0.19	0.08	0.07	0.53	0.26	0.1	127.27	125.77	125.13	471	203	141
5	manual	0.12	0.14	0.1	0.39	0.52	0.5	127.33	74.83	74.6	569	423	289
5	RL	0.19	0.09	0.07	0.51	0.24	0.08	95.08	93.82	93.63	344	172	98
6	manual	0.09	0.09	0.07	0.41	0.46	0.47	101.26	59.23	59.18	252	142	123
6	RL	0.15	0.06	0.05	0.61	0.27	0.11	72.52	71.56	70.98	122	77	36
7	manual	0.11	0.11	0.08	0.4	0.51	0.47	76.77	45.6	45.43	363	183	97
7	RL	0.17	0.06	0.05	0.54	0.27	0.06	56.55	55.56	55.65	207	40	18
8	manual	0.1	0.1	0.08	0.41	0.48	0.47	110.47	64.55	64.7	344	226	191
8	RL	0.17	0.07	0.06	0.57	0.27	0.09	80.29	78.55	78.46	239	94	58
9	manual	0.07	0.07	0.06	0.43	0.47	0.47	162.0	93.08	93.17	180	104	92
9	RL	0.14	0.05	0.05	0.64	0.31	0.17	110.31	108.34	107.55	131	61	22
Average	manual	0.1	0.1	0.08	0.41	0.49	0.48	139.99	81.11	81.1	503	316	233
Average	RL	0.16	0.06	0.06	0.57	0.28	0.12	99.75	98.23	97.77	262	101	57

to the episode reward for 50 episodes or when 700 episodes are reached. Finally, we kept the model with the highest reward so far. In order to evaluate the approach, we choose 9 random routes of varying complexity[2] and use each driver's RL model for choosing the most appropriate driving mode at each step. We repeat the same route three times, one for each driver profile (i.e. A, B and C) in order to evaluate the ability of the RL to choose the proper driving mode in each condition. The correct choice is expected to balance between stress and excitement and in the same time to have an impact on the completion time of the route. A baseline method for comparison is the manual assignment of a certain driving mode to each driver during the whole route. More specifically we assign the conservative driving mode (i.e. AP1) to the most easily stressed driver (i.e. driver A), the normal profile (AP2) to driver B and the aggressive profile (AP3) to driver C who is not easily stressed and prefers high velocities. The performance of the RL personalisation technique, along with the baseline manual choice of driving mode is depicted in Table 1, for each of the nine routes. We measure the average stress and excitement of all steps (4500 steps per route) and the total time needed to complete the route for every driver profile. We also provide the average stress, excitement and total time for all the nine routes. The aim of the RL model (and the respective reward) was to keep driver's stress below 0.2 and excitement above 0.4, so we also depict the number of steps in which the driver's stress was higher than this threshold.

The results show that the average execution time for all the routes has been reduced by 29% for the easily stressed driver (driver A) by a small increase (i.e. 0.06) in the average stress. Also, the stress of the other two drivers has been slightly decreased, and similarly did the average route completion time. This better control of stress values is obvious when we compare the RL method with the manual one in the average number of steps, in which stress succeeded the desired threshold of 0.2. For all the three drivers this number decreased: by 47% for driver A and by 70% and 76% for drivers B and C respectively. This is more evident in routes of higher complexity (i.e. with more and more sharp turns) such as route #4, which is depicted in Figure 1.

In the box plots depicted in Figure 4, which compare the module's behavior for drivers A (left two plots) and C (right two plots), it is obvious that the excitement values for driver 0 are increasing both for high and low complexity routes (increase more in the latter case) with a small increasing effect on stress, which is still bellow the desired threshold. In all cases the comparison is against the manual setup, which assigns a single profile for the whole route.

6 CONCLUSIONS

Our research on the task of personalizing the driving mode of an autonomous vehicle based on the driver's state and the road conditions showed that reinforcement learning is a promising solution, which can improve the driver's comfort by reducing stress and increasing excitement. The next steps of our work comprise adding more information from the Carla environment (front and rear camera), and evaluating more complex scenarios that engage pedestrians and other vehicles. This will increase the complexity of the task and will require more interventions to the autopilot on steering control.

[2]route 4 in the middle of Figure 1 corresponds to a complex route with many sharp turns, whereas route 8 on the right is a simple route with few turns

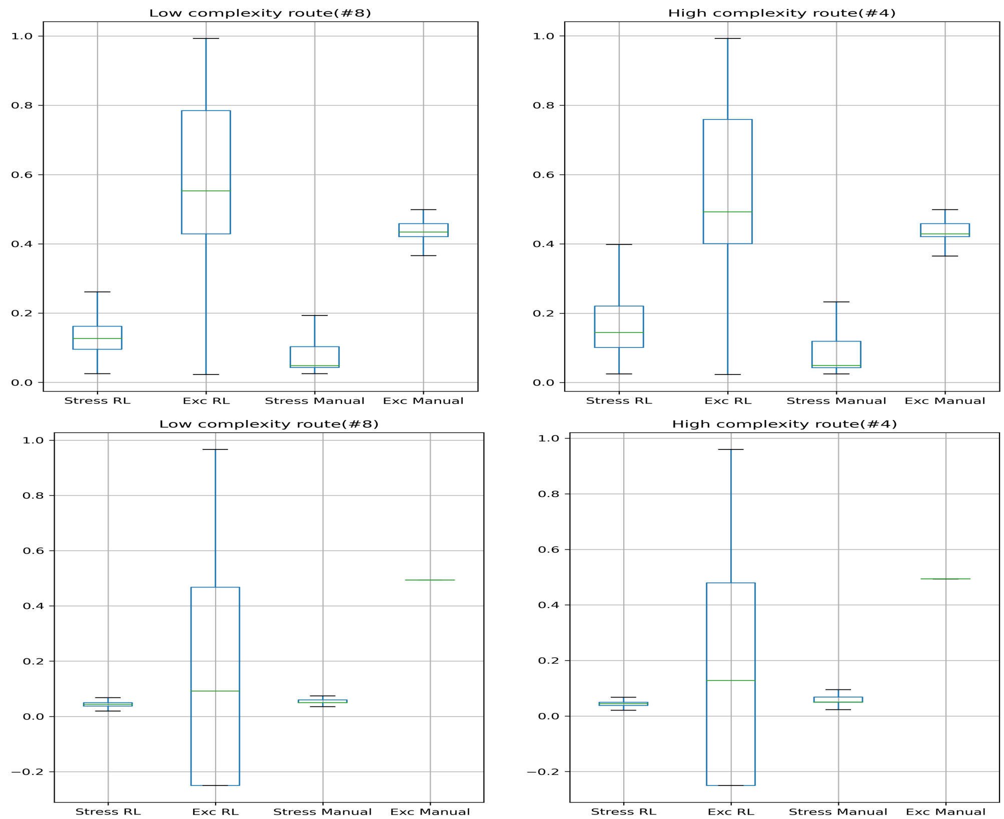

Figure 4: The stress and excitement box plots using the RL and the manual profile assignment. The two plots on the left are for driver A and the two plots on the right for driver C.

ACKNOWLEDGMENTS

This research was supported by TEACHING, a project funded by the EU Horizon 2020 research and innovation programme under GA n. 871385

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Personalization technologies: a process-oriented perspective. *Commun. ACM* 48, 10 (2005), 83–90.

[2] Chandrayee Basu, Qian Yang, David Hungerman, Mukesh Sinahal, and Anca D Draqan. 2017. Do you want your autonomous car to drive like you?. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI.* IEEE, 417–425.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).

[4] Jingchen Dai, Ruimin Li, Zhiyong Liu, and Shichao Lin. 2021. Impacts of the introduction of autonomous taxi on travel behaviors of the experienced user: Evidence from a one-year paid taxi service in Guangzhou, China. *Transportation Research Part C: Emerging Technologies* 130 (2021), 103311.

[5] Andre Dettmann, Franziska Hartwich, Patrick Roßner, Matthias Beggiato, Konstantin Felbel, Josef Krems, and Angelika C Bullinger. 2021. Comfort or not? Automated driving style and user characteristics causing human discomfort in automated driving. *International Journal of Human–Computer Interaction* 37, 4 (2021), 331–339.

[6] Jingliang Duan, Yangang Ren, Fawang Zhang, Yang Guan, Dongjie Yu, Shengbo Eben Li, Bo Cheng, and Lin Zhao. 2021. Encoding Distributional Soft Actor-Critic for Autonomous Driving in Multi-lane Scenarios. *arXiv preprint arXiv:2109.05540* (2021).

[7] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. 2020. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision* 12, 1–3 (2020), 1–308.

[8] Yongshuai Liu, Avishai Halev, and Xin Liu. 2021. Policy Learning with Constraints in Model-free Reinforcement Learning: A Survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence.*

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[10] Dario Nava, Giulio Panzani, Pierluigi Zampieri, and Sergio M Savaresi. 2019. A personalized Adaptive Cruise Control driving style characterization based on a learning approach. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC).* IEEE, 2901–2906.

[11] Maryam Savari and Yoonsuck Choe. 2021. Online Virtual Training in Soft Actor-Critic for Autonomous Driving. In *2021 International Joint Conference on Neural Networks (IJCNN).* IEEE, 1–8.

[12] Svenja Scherer, André Dettmann, Franziska Hartwich, Timo Pech, Angelika C Bullinger, and Gerd Wanielik. 2015. How the driver wants to be driven-modelling driving styles in highly automated driving. In *7. Tagung Fahrerassistenzsysteme.*

[13] Wenjie Song, Shixian Liu, Yujun Li, Yi Yang, and Changle Xiang. 2020. Smooth Actor-Critic Algorithm for End-to-End Autonomous Driving. In *2020 American Control Conference (ACC).* IEEE, 3242–3248.

[14] Peter Stopher, Tassio B Magassy, Ram M Pendyala, Devon McAslan, Farah Najar Arevalo, Thaddeus Miller, et al. 2021. *An Evaluation of the Valley Metro–Waymo Automated Vehicle RideChoice Mobility on Demand Demonstration, Final Report.* Technical Report. United States. Department of Transportation. Federal Transit Administration.

[15] Tobias Vogelpohl, Matthias Kühn, Thomas Hummel, and Mark Vollrath. 2019. Asleep at the automated wheel—Sleepiness and fatigue during highly automated driving. *Accident Analysis & Prevention* 126 (2019), 70–84.

[16] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. 2016. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763* (2016).