Temporal Based Deep Reinforcement Learning for Crowded Lane Merging Maneuvers

Luis Miguel Martínez Gómez[1], Iván García Daza[1], Miguel Ángel Sotelo Vázquez[1]

***Abstract*— In this paper, we propose a joint behavior and motion planning agent based on DRL (Deep Reinforcement Learning) intended for automated vehicles in a crowded merging scenario. The agent is trained using the PPO (Proximal Policy Optimization) algorithm, a state-of-the-art solution that ensures training stability and sample efficiency. We include temporal information in the observation of our agent to improve system stability. We have defined a simulated environment using the CARLA (Car Learning to Act) simulator, which handles the behavior of all other vehicles in the problem. We have performed a comparison between our temporal approach and a classic, distance-based one, both in terms of safety, smoothness and comfort. Results show that our proposed agent yields a smoother, safer experience, and prove the viability of interweaving both systems within the same agent.**

I. Introduction

Autonomous vehicles are expected to revolutionize the transportation industry and the mobility in our cities, but their successful deployment requires robust and reliable decision-making systems capable of dealing with complex and challenging situations in the road environment. As a matter of fact, in order to ensure that autonomous vehicles can safely and efficiently operate in various scenarios, behavior planning becomes a crucial aspect of their implementation, where behavior planning refers to the process of deciding the appropriate actions that an autonomous vehicle (AV) should take in response to its surroundings. Some studies have proposed rule-based methods, where decision-making algorithms are developed based on a set of predefined rules. For example, in [1] a rule-based approach is proposed for controlling connected vehicles on unsignalized intersections. Other studies have proposed Machine learning (ML) techniques for behavior planning in autonomous vehicles given the ability of artificial intelligence algorithms to learn from data to make decisions. Examples of this are Imitation Learning (IL) and Reinforcement Learning (RL), two of the main branches of learning-based approaches that have been successfully applied in the field of autonomous driving [20].

Imitation Learning, which aims at mimicking humans in driving tasks, has been applied in AV control strategies in some specific use cases, such as rural [8] and urban driving scenarios [16]. However, given that IL learns from human drivers (experts that provide the learning source) the performance of the learned policies is asymptotically limited and is extremely unlikely to surpass that of the experts.

[1]Department of Automatics, University of Alcalá, 28805, Spain. lmiguel.martinez@uah.es, ivan.garciad@uah.es, miguel.sotelo@uah.es

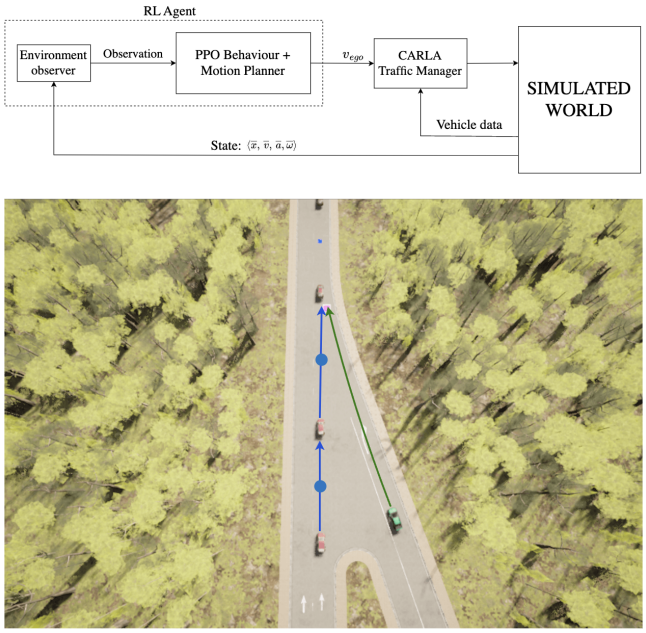

Fig. 1. Merging maneuver on crowded lane.

Reinforcement learning (RL) is gaining attention in behavior planning for autonomous vehicles given its ability to adapt to complex and dynamic environments and its potential to be scaled to many different types of driving situations. Most research results have been achieved in simulation by applying value-based RL algorithms, such as Deep Q-Networks (DQN) to intersection scenarios [11] and highway driving [12].

Some studies have trained an RL agent in a simulated environment and then deployed the agent in a real vehicle [5], and for a limited case, trained the agent directly in a real vehicle [4]. Actor-critic RL algorithms with more complex network structures have been developed and have achieved better control performance in autonomous driving [6]. In particular, state-of-the-art algorithms including soft actor-critic (SAC) [10] and twin delayed DDPG (TD3) [9] have been successfully implemented in AVs in many challenging scenarios, such as complex urban driving and high-speed drifting conditions.

Recent studies have also explored the combination of RL and Model Predictive Control (MPC), a control strategy that utilizes a model of the system to predict future behavior, in behavior planning for autonomous vehicles. In [15], RL is combined with MPC for building a decision making algo-

rithm intended for automated vehicles that have to negotiate with other possibly non-automated vehicles in intersections. The decision algorithm is separated into two parts: a high-level decision module based on RL, and a low-level planning module based on MPC. Similarly, the study by [18] combines RL and MPC for on-ramp merging applications on highways. The conclusion of this study is that the performance of the RL agent is worse than that of the MPC solution from the perspective of safety and robustness to out-of-distribution traffic patterns, i.e., traffic patterns which were not seen by the RL agent during training. Conversely, the performance of the RL agent is usually better than that of the MPC solution when it comes to efficiency and passenger comfort.

Although MPC has shown great potential in behavior planning for autonomous vehicles due to its ability to handle complex dynamics and constraints, a major limitation of MPC is the high computational cost, something that hinders real-time implementation. Similarly, RL has its own limitations. A major one is the computational or learning efficiency. As demonstrated in [14], model training consumes a remarkable amount of computational resources and time. The learning efficiency can be even worse when the reward signal generated by the environment is sparse. Hence, the design of the reward functions becomes another crucial and critical point. An additional limitation that stems from the combination of RL and MPC is the fact that two different models are needed in a kind of hierarchical structure (RL in the high-level and MPC in the low-level), thus contributing to increasing the computational burden.

In this paper, we propose a behavior planning system for autonomous driving tasks that is composed of a single RL-based agent that handles behavior decisions by means of the proper design of the reward function. Thus, optimal high-level decision making and tactical control actions become intertwined in a natural way as a consequence of the learning process. The proposed agent has been developed using PPO [7], an advanced policy gradient RL technique, and tested on challenging ramp merging scenarios on highways using the Carla simulator. Preliminary results indicate that a single RL agent can simultaneously deal with behavior planning and motion control in real time by leveraging a reward function that accounts for safety and comfort in challenging and complex driving scenarios learned from experience.

The rest of the paper is organized as follows: section II provides the technical details of the proposed method; section III describes the practical setup that has been used for running the experiments; section IV presents the main results achieved so far; finally, section V discusses the conclusions and future research steps.

II. Methodology

A. Problem description

The system proposed in this work aims to solve the merging maneuver in a safe and efficient manner, avoiding crashes between the ego vehicle and any other at all costs. As seen in figure 2, the simplest form of the merging problem can be seen from a high-level standpoint as one of four

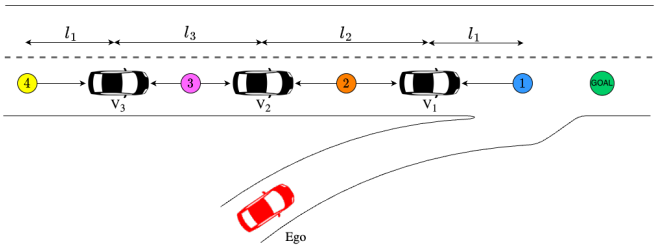

Fig. 2. Schematic representation of the merging maneuver.

choices: merge in front (1), merge between V_1 and V_2 (2), merge between V_2 and V_3 (3), or merge at the back (4). Once the high level process of choosing the destination point of the maneuver is finished, the ego vehicle needs to perform the corresponding low-level actions to achieve its goal, which are comprised of control commands for the ego vehicle. This kind of hierarchical distribution of choices is implemented in [17]. Our system, however, aims to integrate both processes within one architecture, by actuating over the speed reference of the ego vehicle based on a predefined observation. This interconnects the decision making aspect of the behavior planning stage of autonomous driving with the control actions that are performed by the ego vehicle.

However, the problem defined in figure 2 can get much more complex in real world scenarios, in which the main lane is crowded with several vehicles. We include such complexity in our system, so that the system has to perform an active choice on the merging slot that gives the best chances of success. Thus, the agent now has a higher amount of slots to merge into, which opens the possibility of including other elements that affect the criterion that the system uses to perform the maneuver.

B. Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a state-of-the-art Deep Reinforcement Learning (DRL) presented by OpenAI in 2017 [7]. PPO is of the policy gradient family, and it uses stochastic gradient ascent (SGA) on an estimated policy gradient to determine the update direction of the parameters of the neural network that defines the policy of the agent.

It employs the concept of trust region, which guarantees improvements by defining a safe search zone within the parameter space by calculating the expected advantage of an update with several approximations. The approximation error is bound by restricting the difference between the policies before and after updating. An in depth explanation of how the algorithm works can be found on the article cited before.

PPO has been shown to increase sample efficiency, since every iteration of the learning process is theoretically guaranteed to improve the q-value of any given state-action pair. The approximations that allow the calculation of the trust region in which the policy is guaranteed to improve eliminate the certainty of improvement, but in practice the algorithm has been applied to a wide array of domains with satisfying results. Training stability is also improved by limiting policy change with a clipping process of the surrogate objective

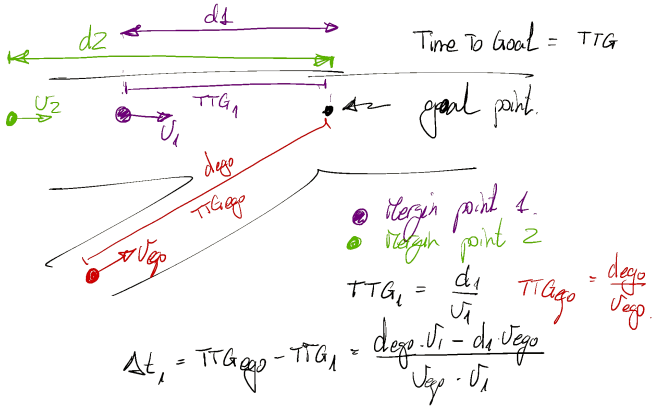

Fig. 3. Schematic representation of observation.

function that maximizes the advantage between the current policy and the updated one, which reduces the perceived improvement of policies that heavily diverge from the current one, and therefore produces an incremental refinement of the policy as the training goes on.

C. Reward function

The state space of the system includes the state of every vehicle in the simulated world, that is, all positions, velocities and accelerations, both linear and angular, of every actor in the simulation. However, the state of the system is refined into the following vector:

$$O = [\Delta t_1,\, \Delta t_1,\, \ldots,\, \Delta t_N,\, \Delta t_{N+1},\, v_{ego},\, v_1,\, \ldots\, v_N] \quad (1)$$

Where Δt_i represents the time differential of point i, i.e. each of the merging points defined as the mid-point between vehicles on the main lane, and v_j denotes the speed of vehicle j, and N represents the number of vehicles in the simulation other than the ego vehicle. In our experiments, we have set $N = 10$, which brings the total of cars in the simulation to 11.

The time differential is calculated as follows:

$$\Delta t_i = \Delta t_{\text{ego}} - \Delta t_{p_i} \quad (2)$$

$$\Delta t_{p_i} = \frac{d_i + d_{i+1}}{v_i + v_{i+1}} \quad (3)$$

where d_i denotes the distance between point i and the destination point, and v_i represents the velocity with which point i advances. Both the distance and speed of all possible merging points are calculated as the mean of the two surrounding vehicles. When dealing with points (1) and (4), a vehicle is assumed to be leading (trailing) 30 m ahead (behind) and with the same speed as the one being considered.

With this observation vector we provide temporal information to the agent, which has the task of minimizing the time differential for any of the points that are available for merging.

The reward function is defined as follows:

$$r = w_1 \cdot r_{\text{collision}} + w_2 \cdot r_{v_{\text{ego}}} + w_3 \cdot \sum_{i=1}^{N} r_{p_i} \quad (4)$$

where w_i represents the weight given to any particular term of the definition. We set $w = [1000,\, 2,\, 1]$, to heavily penalize collisions and modulate the speed term so that its effect is not overshadowed by the time differentials rewards.

The first term penalizes collisions between the ego vehicle and any other car and is defined as:

$$r_{\text{collision}} = \begin{cases} -1 & \text{if collision} \\ 0 & \text{else} \end{cases}$$

The second term penalizes slow speeds for the ego vehicle and is used as a deterrent for the vehicle to allow the convoy that drives through the main lane to pass and then complete the maneuver. It also promotes the ego vehicle to maintain its speed within the range defined by v_{min} and v_{max}. In our tests, we set v_{min} to 8m/s and v_{max} to 12m/s. It is obtained with the following definition:

$$r_{v_{\text{ego}}} = \begin{cases} \frac{v_{\text{ego}} - v_{\text{min}}}{v_{\text{max}} - v_{\text{min}}} & \text{if } v_{\text{ego}} < v_{\text{max}} \\ -1 & \text{if } v_{\text{ego}} > v_{\text{max}} \end{cases}$$

The third term of the reward function aims to minimize the time differentials according to the next expression:

$$r_{p_i} = \frac{1}{|\Delta t_i|} \cdot w_i \quad (5)$$

where w_i denotes the weight associated to spot i, and serves as a modulator of the frequency with which said spot will be chosen by the agent. On our tests, those weights were set to 0.1 for the first and last slots, and 1 for all others. r_{p_i} is clipped between 0 and 5 to prevent the reward value from exploding when the time differences are small.

III. Results and Discussion

A. Simulation environment

The environment in which the DRL agent will be trained derives from CARLA [3], a well-known simulation suite that allows for realistic and accurate traffic simulations. We have developed an OpenAI's Gym [2] custom environment that wraps the CARLA client into a package that can be interconnected with several established DRL libraries such as Stable Baselines3 [19], in which the PPO instance that was trained for this work is implemented.

The simulation has a period of $50\,\text{ms}$, which is also the control period for the underlying low level controllers that perform direct control actions (i.e. steer and pedal actuation). These PID controllers are managed by CARLA's Traffic Manager module, and take as reference the location of the goal and the action of the PPO agent. The decision period is set to $1\,\text{s}$, so that the controller has time to react to a sustained reference.

Figure 1 shows the interconnection of the RL agent and the modules that perform both low level actions and data acquisition from the world. The simulation sends the state of every car to the agent, which refines the information into an observation. This observation is used to calculate the action that the agent will take, which is then sent to the Traffic Manager module to manage the low level actions of both the ego vehicle and the vehicles on the main lane.

TABLE I

STATISTICAL PARAMETERS OF THE BEHAVIOUR OF THE EGO VEHICLE.

Measurement	Temporal reward function	Spatial reward function
Collision rate - 500 episodes (%)	17.82	28.71
Mean of 95 percentile of jerk ($\mathrm{m/s^3}$)	2.02	6.18
Mean of max jerk ($\mathrm{m/s^3}$)	3.48	10.52
Mean of 95 percentile of acceleration ($\mathrm{m/s^2}$)	2.39	2.45
Mean time of episode completion (s)	22.45	16.49

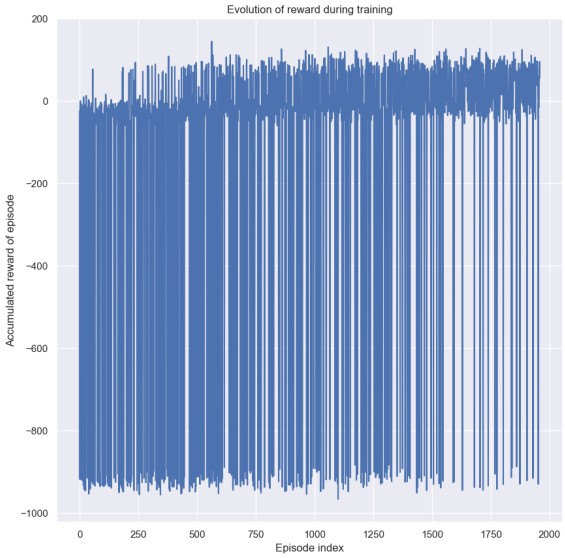

Fig. 4. Evolution of reward during training of temporal RL Agent.

B. Results

Figure 4 shows the evolution of the reward signal during the 50000 steps of training of the temporal agent. The result denotes the ability of the PPO agent to learn to maximize rewards in terms of both the speed and time differential terms. The graph also shows how the system learns to avoid collisions until the collision rate is reduced, but not completely eliminated. Further training has not improved collision rate, and shows signs of overfitting, in which the ego vehicle learns to accelerate up to a constant value of speed and maintain it regardless of any other stimuli in the reward function.

Table I shows a statistical comparison of a 100 episode run between the agent proposed in this work and one trained with the reward function defined in [13], trained on a PPO agent with the same configuration as the one shown here. As inferred from the statistical parameters, our agent has a reduced collision rate, mainly due to the inclusion of collision information on our reward function. The one implemented in the spatial agent relies on keeping the ego vehicle away of the other vehicles present on the simulation by penalizing close distances, but has no sense of collision, which prevents the agent from discerning between situations in which the maneuver is successful or the ego vehicle is too close to one vehicle and far from other. However, further studies are required to assess the viability of improving the collision rate of the temporal agent, either with a modified reward function that includes both spatial and temporal terms, or by designing a complementary system that identifies potential high-risk situations and modulates the action of the agent accordingly to better evade lateral or frontal collisions.

Jerk distribution tests considerably smoother with the temporal agent. As seen in table I, the mean of the 95th percentile across the 100 episode run falls close to acceptable limits in non steady-state maneuvers. Despite the system not having a sense of rate of change of either its output or the observations, it has learned a behavior that adapts to the other vehicles in the road, which have either constant or monotonous profile speeds. This allows the system to modulate its dynamics so that they become somewhat smooth inherently. However, even when including terms in its reward function that penalize large accelerations, the spatial agent exhibits greater values of both jerk and acceleration.

The only parameter in which the spatial agent has an advantage over the temporal one is swiftness in the completion of the maneuver; it is almost 6 seconds faster than its counterpart, which can be seen as a potential safety benefit since by taking less time to complete the maneuver there is a reduced chance of any risk hazard to materialize. However, if all other parameters are factored in –particularly collision rate and max jerk–, it is clear that this improvement in time comes at the cost of both actual and perceived safety, given the fact that this agent produces more collisions and the overall experience presents a much more jittery behavior.

Figure 5a) depicts the typical behavior of the temporal agent. At the beginning of the episode, it accelerates up to the lane's speed –configured to be $8\mathrm{m/s}$, and maintains that velocity until it detects a potential conflict with vehicle 06. The agent engages then in a modulation phase, in which it reacts to the changes in velocity of said vehicle to ensure that it will reach the merging point avoiding any hazards. In the final phase of the maneuver, the agent modulates the speed of the ego vehicle yet again to prevent it from merging too close to either vehicle 05 or 06. The agent ends the maneuver with an acceleration stage to adequate the speed of the ego vehicle to that of the main lane so that it can avoid a rear collision. The agent shows a small amount of ripple in the velocity signal, mostly due to noise in the low-level controller that executes the throttle and brake commands to follow the reference that our agent outputs.

This episode proves the temporal approach's success when refining the problem's state into an observation to feed into

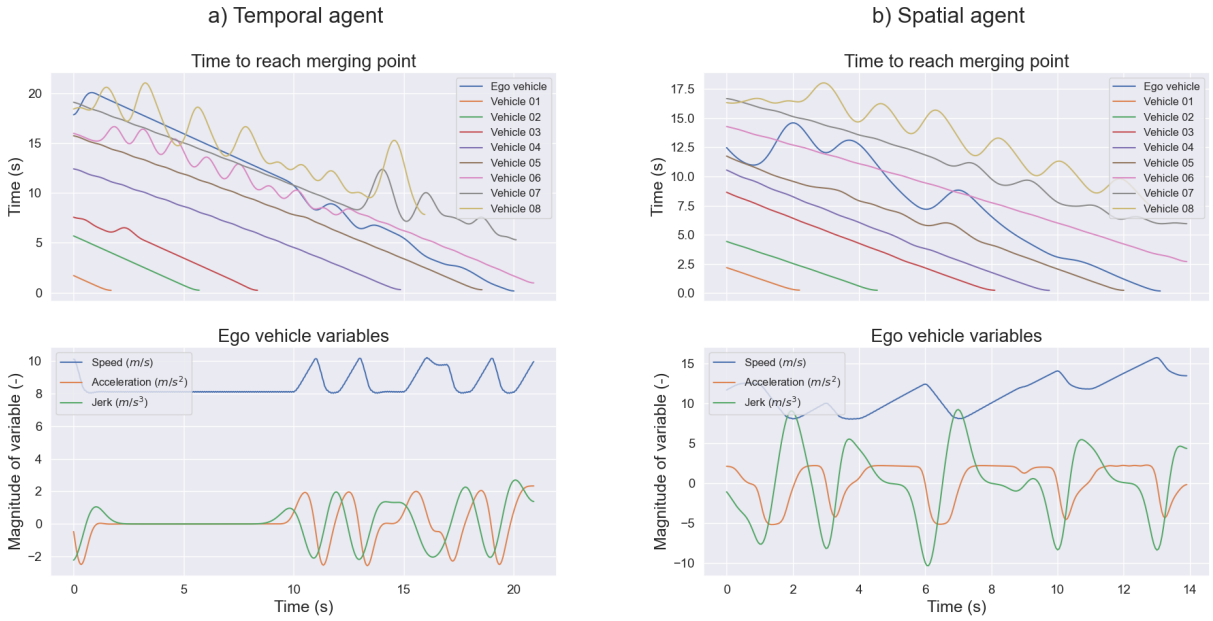

Fig. 5. Evaluation of an episode. a) Temporal agent. b) Spatial agent.

the RL agent. By processing the state into time differentials of merging slots, the system obtains a sense of avoidance of other vehicles, which is further accentuated by the collision term in the reward function. Moreover, the behavior planning section of our implementation shows promising performance, as it allows the agent to modulate its actions and adapt to road conditions.

Side b) of figure 5 includes an evaluation run of the spatial agent. It can be seen from the bottom subfigure that its behavior is much more aggressive than the temporal one, with higher values of speed, jerk and acceleration. Furthermore, there are barely any instants in which the agent maintains a stable velocity, and it shows no sign of adapting to the conditions of the main road. In a configuration considerably simpler than the one studied for the temporal agent –particularly in how vehicle 06 modulates its velocity–, the one with spatial information performs worse, completing the maneuver but with clearly unsafe parameters. In addition, even though the reward function was configured to prioritize merging in the mid-point between two reference cars, the agent does not manage to regulate its action to achieve this safety objective.

Another point of contention between the two agents refers to the information each of them employ to make the decision for the merging slot. While the temporal agent includes information about all the vehicles on the road, interwoven in the mid-point calculations that occur before the agent receives the observation, the spatial agent gets information only about its two closest vehicles. This reduces the adaptability of the system, since its observation only recognizes distances and velocities of the surrounding vehicles and it is unable to identify other potential better merging slots. The temporal agent requires more computational overhead to process all the information it receives from the world, and could potentially be at a disadvantage in scenarios with a large number of cars, but the trade-off in adaptability, smoothness and safety falls in its favour.

IV. Conclusions and Future Works

In this work, a PPO DRL agent has been trained to solve a merging maneuver in a crowded environment. The agent has been trained with a temporal based reward function, and a comparison has been carried out against a similar agent, trained with spatial information in its reward function.

Several conclusions can be extracted from the results discussed in the previous section. Chief among them is the viability of a combined architecture with both behavior and motion planning in a single system. It has been shown that such an agent can safely and timely complete the maneuver without a noticeable increase in system complexity, training cost or computation time.

Furthermore, we have proved how the inclusion of temporal information in the form of time differentials to the merging slots improves the stability and adaptability of the system while allowing the system to modulate its behaviour to that of the vehicles on the main road. In addition, it appears that, even without explicit terms to regulate the behavior in terms of acceleration and jerk, the system has learned to actuate in such a way that the experience inside the ego vehicle will not be perceived as unsafe or uncomfortable. We have also shown how including information on collisions in the reward function reduces the collision rate.

Several lines of research stem from the results shown in this work. The most important one is the extension of the intertwined proposed architecture to other complex scenarios, such as crowded highway cruising, roundabouts

or t-junctions. It is our belief that such an approach can be highly beneficial both in training ease, computational cost and performance. In this line, an all-in-one system, which integrates behavior and motion planning and low-level control could be worth studying to determine the incremental gains of implementing problem-wide systems.

We also propose a testing scenario in which the ego vehicle faces human drivers in a merging maneuver, instead of simulated agents. By performing the most intensive part of training in a computer-controlled setting and then adding human actors to the environment via driving simulation equipment we hope to achieve better performance in real world tasks, without having to invest in a large amount of human hours in the simulation.

ACKNOWLEDGMENT

This project has received funding from the Key Digital Technologies Program (HORIZON-KDT-JU-2021-2-RIA) for European Leadership Joint Undertaking under project proposal No 101096658 (A-IQ-Ready Project). This Joint Undertaking receives support from the European Union Horizon Europe research and innovation programme and Germany, Austria, Spain, Italy, Latvia, Belgium, Netherlands, Sweden, Finland, Lithuania, Czech Republic, Romania, Norway.

REFERENCES

[1] G. Lu, L. Li, Y. Wang, R. Zhang, Z. Bao, and H. Chen, "A rule based control algorithm of connected vehicles in uncontrolled intersection," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[2] G. Brockman, V. Cheung, L. Pettersson, *et al.*, *Openai gym*, 2016. eprint: `arXiv:1606.01540`.

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[4] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *British Machine Vision Conference (BMVC)*, 2017.

[5] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," in *British Machine Vision Conference (BMVC)*, 2017.

[6] A. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 29, 2017.

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *ArXiv*, vol. 1707.06347, Jul. 2017.

[8] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitski, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4693–4700.

[9] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, 2018.

[10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*, 2018.

[11] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *IEEE Int. Conf. on Robot. and Automat. (ICRA)*, 2018.

[12] P. Wang, C. Chan, and A. d. L. Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[13] P. Wang and C.-y. Chan, "Autonomous ramp merge maneuver based on reinforcement learning with continuous action space," *ArXiv*, vol. abs/1803.09203, 2018.

[14] E. Neftci and B. Averbeck, "Reinforcement learning in artificial and biological systems," *Natural Machine Intelligence*, vol. 1(3), pp. 133–143, 2019.

[15] T. Tram, I. Batkovic, M. Ali1, and J. Sjoberg, "Learning when to drive in intersections by combining reinforcement learning and model predictive control," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019.

[16] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *ArXiv*, vol. 2010.03118v4, 2020.

[17] S. Triest, A. Villaflor, and J. M. Dolan, "Learning highway ramp merging via reinforcement learning with temporally-extended actions," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2020, pp. 1595–1600.

[18] J. Lubars, H. Gupta, S. Chinchali, *et al.*, "Combining reinforcement learning with model predictive control for on-ramp merging," *ArXiv*, vol. 2011.08484v, 2021.

[19] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: `http://jmlr.org/papers/v22/20-1364.html`.

[20] J. Wu, Z. Huang, Z. Hu, and C. Lv, "Toward human-in-the-loop ai: Enhancing deep reinforcement learning via real-time human guidance for autonomous driving," *Engineering*, vol. 10.1016/j.eng.2022.05.017, 2022.