2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)
July 11-17, 2021. Nagoya, Japan

IV2021 Workshop Proceedings

2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops) | 978-1-6654-7921-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/IVWorkshops54471.2021.9669204

Precise Control for Deep Driving using Dual Critic based DRL Approaches

Surbhi Gupta, Gaurav Singal and Deepak Garg

Abstract— **Autonomous driving problems related to vehicle control using deep reinforcement learning (DRL) techniques, are still unsolved. DRL approaches have achieved notable results, its dependability on reward functions and defining the type of control actions are dominating factors of the objective, that controls its success. Several DRL approaches applied in the past consider a finite set of available actions to be controlled by the agent hence, it performs sharp actions. While real driving requires precision control capabilities that tend to apply safer and smoother actions. For incorporating such precision control capabilities, this paper considers the driving problem as a continuous control problem. For this, the gym-highway environments are used as these environments are controllable and customizable to simulate diverse driving scenarios. The simulation setup for parking is updated to resemble the complex scenario and for highway driving a novel reward function is designed to handle continuous actions. Dual critic based DRL approaches are applied as these approaches have shown remarkable performance in robotic locomotion control problems. The video results demonstrate the way different policies fulfil the objective.**

I. INTRODUCTION

Self-driving car has gained the attraction of people. The first self-driving car was developed in the 1980s by Ernst Dickmanns [1]. The winner of 2005, DARPA Grand Challenge, Stanley, used one of the Artificial Intelligence techniques (machine learning) for navigation [2]. Nowadays technology used to make the car autonomous and make a decision as a human does is based on deep learning models. Deep learning became the choice of researchers and industries practising autonomous driving because of its auto feature extraction capabilities and the outperforming results of these models on perception tasks related to classification, detection, and regression approaches practising autonomous driving. At present, deep learning models are also being used for control tasks related to robot movement control [3], drone flight control [4], vehicle movement control [5]. For these tasks, deep learning models are used with reinforcement learning techniques that have shown breakthrough results in the control domain as reinforcement learning involves learning while interacting with the environment with the notion of reward feedback. When it comes to driving a vehicle, an artificial agent is expected to act the way humans act. According to the Driver and Vehicle Standards Agency (DVSA), most people take 67 hours, this includes learning drive lessons and practising [6]. Hence on average, driving skills require a month and a couple of days of human efforts. While deep reinforcement learning (DRL) agents can be trained in a couple of days to learn driving.

Past work applied DRL for driving dexterity, has shown the possibility to see the driverless car. With these artificial agents, autonomous vehicles are supposed to be less error/hazard-prone than a human as there are many factors that cause human distraction and such distractions never apply to an artificial agent. Despite the driving capabilities, these autonomous agents were reported in several scandals of accidents [7], [8] and proved that technology is not yet ready for driving on public road and unable to get highest grading of SAE J3016 standard [9]. Even after technological advancements, human trust, and reliability concerns disallow them to ride driverless. To win human reliability concern, reported work must show the acute control of the vehicle and collision-free ride.

Driving on highway considered in literature with deep learning + supervised learning in [10] (DSL) is limited to the detection part. Though [11], [12] consider control aspects, [11] needs labelled data and [12] controls only steering angle. Deep learning + reinforcement learning (DRL) based approach applied in [13] needs demonstration data from human, [14] considers only single vehicle running on the road, [15] considers cyber-physical attacks on sensor values, and [13], [16]–[19] considers only discrete action hence are limited to apply on a finite small set of actions. To deal with unavoidable crashes that may occur due to the fault of other vehicles, the agent needs to be sensitive to collision and penalized according to the level of injury incurred due to collision. For this, it is essential to consider control over minor action. For incorporating precision control capabilities deep driving agents need to consider continuous action values. Limited work has considered the highway and parking-like scenarios with continuous actions that are addressed in this paper. The following points illustrate the contribution of the paper:

- We have modelled the driving control problem as a continuous control problem to achieve precise control over driving decisions in highway roundabout driving

All authors belongs to Bennett University, Greater Noida, U.P., India
Surbhi Gupta: *email* gsurbhi1993@gmail.com *ORCID* 0000-0002-8356-1836, Gaurav Singal: *email* gauravsingal789@gmail.com *ORCID* 0000-0001-7570-6292, Deepak Garg: *email* deepakgarg108@gmail.com *ORCID* 0000-0002-7243-3599

and for parking cars in the parking lot by the update in reward function.

- For parking scenario, advancements were made in the pre-existing gym-parking simulation environment. The original platform considers only a single vehicle that is controlled by the agent (ego-vehicle) and in each episode, it is initiated from the centre of parking (Fig 4a). Updated environment simulates multiple vehicles and ego vehicle is instantiated from any side of the parking lot to be realistic (Fig 4b). Also, the number of vehicles in the parking lot in each episode varies in the updated environments. Also, the reward function proposed in our work for parking includes an auxiliary reward term for successful parking. For highway driving, a reward function is designed to handle continuous actions as the original environment reward function was designed to handle discrete actions.
- We have applied advanced DRL actor-critic approaches that consider two critics instead of one to improve driving manoeuvre in dense environments. The results on the proposed updated environment have shown successful driving policy on parking lot and highway.

This paper is organized in the following manner: Section II provides detail on the literature for driving task using deep learning with reinforcement learning and describes the background of DRL approaches, details on optimal vs. robust policy, and precision control. Section III illustrates how dual critic based DRL approaches are applied in the updated environment by considering continuous actions for driving on highway and parking system. This section also illustrates the changes done in the reward function for precision control. Section IV outlines the simulation parameters, with results, and discussion. Finally, Section V gives the concluding remarks.

II. Related Work

This section provides detail on deep learning combined with reinforcement learning for autonomous driving, sets the background, details on optimal vs. robust policy, and precision control.

A. *Autonomous Driving Tasks considered in Deep Learning + Reinforcement Learning (DRL)*

Wulfmeier et.al. [13] had considered the urban driving scenario and approached it with inverse RL. While collecting expert data, the conventional obstacles: car, tree, pedestrian, and unconventional obstacles: narrow underpass, steep ramp, bollards, etc. were considered. Sallab et.al. [14] approached the lane-keeping scenario with the attention, DQN, and RNN method. The video simulations show that the road is not occupied by any other vehicle/pedestrian. Hence the complete walkway passage is available for driving. Edouard et. al. [17] had considered the dense traffic that may appear while driving and also approached an attention-based architecture with DQN to learn driving behaviour at intersections by considering nearby vehicles. Aidin et.al. [15] have incorporated robustness in autonomous vehicle (AV) by analysing the state estimation process and focused on the reliability of the data used in decision making. As a safety mechanism to data injection attack, they proposed an LSTM based NN and used the RL algorithm for the true state estimation based on the leading vehicle's speed.Yesmina et. al. [16] also considered an urban autonomous driving scenario and proposed a multi-step Temporal Difference (TD) approach. Edouard et. al [18] considered the non-linear system where the task is also to drive fast and keeping the Lane with uncertain participant's behaviour. Lin et.al. [19] considered robust adaptive control problem with dynamical Bicycle model (linear dynamics) for lane-keeping application and proposed a model predictive control algorithm with interval predictor for state prediction. Wang et.al. [20] also highlighted the requirement of fine control in autonomous driving as considered in our work. They have designed a reward function, network architecture and used DDPG for fine control.

Lane changing behaviour was learned for a multi-lane straight road in [21] using Intelligent Driver Model (IDM) car-following model and only lateral movement was controlled using the RL approach (DDPG) and with the proposed reward function. Only simple lane geometry was considered and control of both lateral and longitudinal using RL was left as future work that is considered in our work. Chen et.al [22] considered driving at a complex roundabout environment and worked on state representation for driving using DDQN, TD3, and SAC. They have used image-based input and learned low dimensional representation using Variational Auto Encoder. The control commands (action) used for driving are not specified. As they have applied DDQN that is useful for discrete action space and SAC, and TD3 are applicable for Continuous actions hence, the considered action space is unclear. We have considered the continuous actions to incorporate precision control capabilities on parking and highway driving. Also, the high-level visionary perception was considered in [20], [22] while, our work incorporates the control capabilities based on the low-level raw sensory input.

B. *Background of DRL approaches*

The algorithms proposed in DRL fall under three categories: Policy-based, Value-based, and Actor critic [23]. This paper considers actor-critic approaches SAC [24] and TD3 [25] as these approaches outperform the other actor-critic approaches. For goal-based environment HER [26] is used with these approaches. Soft actor critic (SAC) is an entropy-based method that favours stochastic policy and is robust to modelling error. Twin Delayed Deep Deterministic policy gradient (TD3) addresses the overestimation bias in actor critic setting using clipped Double Q-Learning and dual critics. SAC and TD3 were applied to physics-based tasks. Hindsight experience replay (HER) allows to apply off-policy DRL algorithms to goal-based tasks by selecting goal with one of the four proposed strategies. The results have shown that future strategy outperforms hence our work considers this strategy.

C. Optimal Vs. Robust Policy

Being optimal is different from being robust. An optimal policy may refer to achieving the best performance or minimizing cost whereas a robust policy need not be optimal but it chooses actions that are on the safer side and avoids catastrophic events. In autonomous driving, rather than reaching an optimal policy, getting a robust policy is of great interest as a robust policy will intend to avoid collision thereby aimed to reduce road traffic. Also, a robust policy will not cost the challan penalties that may incur due to the breakage of traffic rules.

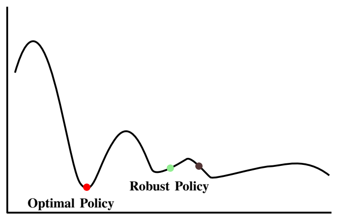


Fig. 1: Difference between Optimal and Robust policy

Being optimal most of the time will be beneficial but scarifying optimality for robustness even some time can evade from the disastrous event. Driving fast corresponds to the optimal policy as it saves the travelling time while driving at average speed may represent a safer policy as it is easy to control and take sudden safer actions at the slow driving car than that runs at fast speed. Fig 1 further graphically illustrates the point where an optimal or robust policy may lie. There may exist a single policy that optimizes the performance (maximizing return or minimizing cost). While there may exist multiple policies that may not be optimal but attains robustness to different scenarios. For example, when driving on highways, it's important to keep a safe distance from other vehicles hence, a robust policy will be the one that follows such a rule of thumb. On the other hand, when driving to park the car in the parking lot, the car should be parked close to other vehicles to save your car from being theft. Hence when to maintain distance and when not is the situation-dependent.

D. Precision Control

DRL previously applied to autonomous driving considers the discrete action space such as turning to the right, left, moving forward. When defining the actions in a discrete manner the right refers to the $+90^\circ$ and the left refers to -90° complete rotation of the vehicle. In realistic driving, a human seldom performs such complete rotations and most of the turns (U-turn, turning the car back, driving on a rotatory lane) require intermediate steering angle control. Even turning to the right can be accompanied by controlling the steering at different angles depending on the situation at the turn as shown in Fig 2. Also, the turning angle differs from the steering angle by a large margin. The turning angle in the range [-90,+90] may correspond to a steering angle of range [-30,+30] as the steering angle refers to the wheel angle from the front of the vehicle. In reality, if a person says to turn complete left it corresponds to a turning angle of -90° while in steering angle it means -30° of control. Precision control refers to driving the vehicle at such intermediary angles or acceleration values. Though deep reinforcement learning techniques can manage such continuous control, still applied with discretized actions in the past for driving. Those that have considered the continuous control had acknowledged only for quiet straightforward scenarios.

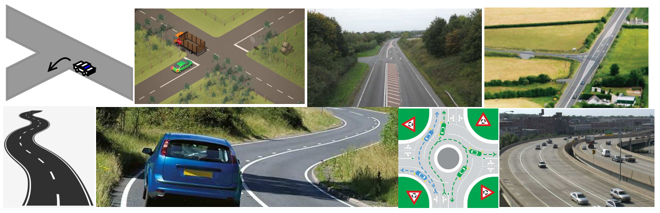


Fig. 2: Different types of road scenarios: Top row shows the lanes where discrete actions (such as turning to left, right, moving straight) can be performed. The bottom row shows the lane where continuous actions (such as turning to various angles) can be performed.

III. Dual Critic DRL Approaches for Precision Control Deep Driving

This section provides the detail of approaches that are applied to analyse the precise control capabilities of DRL. To analyse such capabilities, this paper considers the driving scenario for parking in the parking lot and driving on the highway. The simulation setup provided in OpenAI gym [27] for parking is too simplistic and for driving at the roundabout on the highway considers only discrete actions such as turning left, and right. Because of this, the past methods that had applied DRL algorithms in driving using the gym-highway platform had not considered such a scenario. This paper provides a more advanced setup by considering other parked vehicles in the parking lot and by considering the continuous actions for both environments.

A. Preliminaries

In deep reinforcement learning, the agent interacts with the environment defined in MDP notation (S, A, R, p) where, S, and A shows the state and action spaces respectively, R shows the reward function and p shows the transition probability. Every episode of the interaction starts with the initial state $s_0 \in S$. For some scenarios, the initial state is fixed for each episode while for others it is sampled from the initial state distribution of the environment. While interacting with the environment, the agent observes a state of the environment $s \in S$ and accordingly selects the action $a \in A$. Based on the environmental dynamics and unknown transition probabilities $p(s'|s,a)$, the agent is transitioned to another state s' of the environment and receives a reward $r(s,a)$. The agent's objective $J(\pi)$ is to maximize the expected return using the learned policy π.

$$J(\pi) = \max \sum \mathbb{E}_{(s,a)}[r(s,a)] \tag{1}$$

where reward at each time comes from the reward function R.

B. Learning the controller for Parking

Learning in DRL is derived by the reward function based on the action selected by the policy in a state. The wrong choice or inappropriate function may converge to a policy that still needs improvement. As high positive rewards aid in learning good actions, low positive rewards help in choosing neutral actions rather than those that lead to bad. Negative rewards have their specific significance that helps in avoiding vulnerable/offensive actions. It is difficult to get the data that can be used to learn the action that leads to negativity as such data can be acquired by performing the collisions/accidents in real. Instead of depending on the real data, it is easy to work in simulation and get such data without incurring any real harm.

First 6
位置(x,y),
速度(vx, vy)和
航向(cosα, sinα)
α是与x正方向的夹角

middle 6
已实现的目标Ga

last 6
期望的目标Gd

The state defined in the gym-parking environment is 18 dimensional. First 6D representing the position (x,y), velocity (v_x, v_y) and heading direction (cosα,sinα) where α is the angle from positive x-direction. The middle 6D shows the achieved goal G_a and the last 6D shows the desired goal G_d (Landmark features). Position values are scaled by 1/100, velocities by 1/5. Actions cover both the lateral and longitudinal control of the vehicle by controlling steering angle θ and acceleration a_k. For precision control, we defined both as continuous variables. The reward function is as follows:

$$R = -\sqrt{(|G_a - G_d|.W_r)} \tag{2}$$

where W_r shows the reward weights. The episode is successfully terminated if the timeout is reached or a vehicle is crashed or the goal is achieved i.e. $R > -r_{sg}$. where r_{sg} represents the success goal reward. The environment simulates only a single vehicle called an ego vehicle which is to be parked in the landmarked position (Fig 4a). In each episode, the ego vehicle starts from the centre position.

Fig 4b shows the update done in the environment to simulate other parked vehicles. Also in each episode of the updated environment ego vehicle may appear at any position within the corner boxes shown in the figure as in real, a car used to start parking from the entry point, not from the middle. The number of other vehicles N to appear in the episode is not fixed and varies in each episode. N is sampled from uniform distribution and $N \in [0, spots)$. In the updated environment the ego vehicle can observe a $18+2n$D vector augmented with 2D features of other n nearby vehicles. The initial 6D shows the state of the ego vehicle $(x, y, v_x, v_y, cos\alpha, sin\alpha)$, next (n)*2D shows the position (x,y) of the n neighbouring vehicles of ego vehicle, next 6D shows the achieved goal and last 6D shows the desired goal features. The original observation of (18D) can also get back by setting $n = 0$. To learn a robust policy that avoids collision we have augmented the reward function with other information. If the ego vehicle successfully drive to the landmarked location without collision then

$$R = R + I_s \tag{3}$$

Where, I_s is set to 1 if the vehicle is parked successfully, 0 otherwise. In case the ego vehicle collides with the other vehicle then a penalty is applied. The motivating example for incorporating the penalty is: if a car strikes with another vehicle and only the scratch appear due to collision then a small penalty should be given. If collision breaks up the headlight then it should be more penalize. Hence the penalty should depend on the type of collision or the overlapping between two vehicles. Based on this idea we penalize as per the overlapping between two vehicles. Hence in case of collision:

$$R = R - I_c * over(v_e, v_o) \tag{4}$$

Where I_c is set to 1 if the vehicle is crashed, 0 otherwise. $over(v_e, v_o)$ shows the collision penalty function. It is used

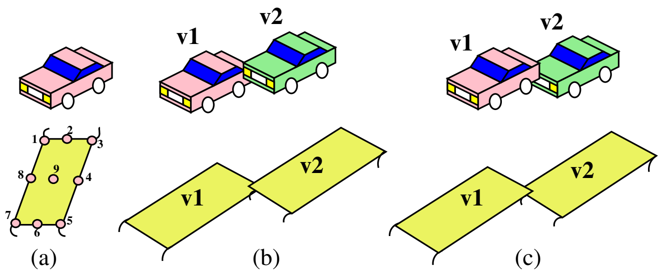


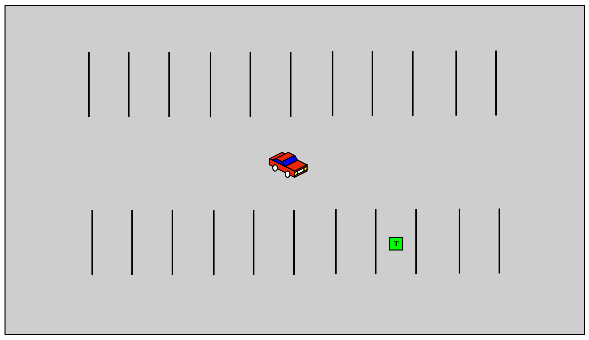
Fig. 3: Estimating collision based on vehicle overlapping to set penalty

to estimate the intersected area between ego vehicle v_e and other vehicles v_o. To estimate the value of this function 9 points (as shown in Fig 3 (a)) were drawn onto the vehicle v1. As the collision can be caused either by the front vehicle or by the rear vehicle (Fig 3 (b,c) top row). In such a case either a point of v1 will be overlapped on v2 or vice versa (Fig 3 (b,c) bottom row). This overlapped value is defined by the $over(.)$ function. Hence more overlapping would cost more negative penalty and the minor collision would lead to a minor penalty. The second term of Eq. 3 and Eq. 4 is called the auxiliary reward term as these are applied on successful reaching or collision. The reward function considered in the original environment does not include these terms. In this paper, we have applied SAC and TD3 with HER. Results on different experiments and with updated reward function are discussed in Section IV-B.

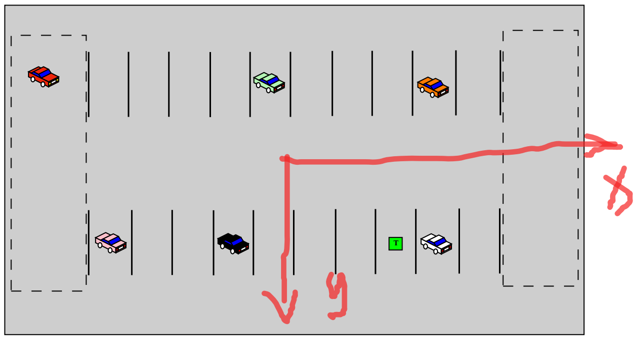
C. Learning the controller for Highway Driving

We have considered the continuous action space for precise control of actions while driving on the highway-roundabout-like scenario. The task is to drive on the highway as fast as possible, without changing the lane and without any collision. Agent state includes the information of ego vehicle and its neighbouring n vehicles. For each vehicle, observation includes a binary indicator variable that shows the presence of a vehicle, its position (x, y), and velocity (v_x, v_y). The action includes vehicle lateral and longitude control and includes 5 actions (turning left, turning right, idle, driving fast, driving slow). The reward function defined for discrete actions is as follows:

$$R = r_c * (I_c) + \frac{r_{hs} * s_a}{\max(s_c - 1, 1)} + r_{lc} * (I_l) \tag{5}$$



(a) The gym-parking environment considers single vehicle to be parked on the landmarked position (green box). In each episode the car is initiated from the centre.



(b) The proposed updated parking environment that considers multiple already parked vehicles with the ego vehicle. In each episode, the car may start from any corner side area (dotted box).

Fig. 4: gym-parking environment Original Vs. Updated

where, r_c: collision reward, I_c binary indicator variable (1:if vehicle crashed, 0 otherwise), r_{hs}: high speed reward, s_a: closest speed allowed to a given speed, s_c: vehicle speed count, r_{lc} : reward on changing the lane, I_l: binary indicator variable (1:if lane is changed, 0 otherwise).

We have considered controlling of steering angle and acceleration actions as continuous variables and updated the reward function accordingly as follows:

$$R = r_c * (I_c) + \frac{r_{hs} * s_a}{\max(s_c - 1, 1)} + r_{lc} \tag{6}$$

$$r_{lc} = 1 - (lat/w_{l_e})^2 \tag{7}$$

where lat shows the latitude estimated using the vehicle's position (x,y) on the ego vehicle's lane and w_{l_e} shows the width of the lane initially chosen for ego vehicle. Instead of using a fixed lane changing reward, in the continuous environment, it is dependent on the latitude position of the vehicle. Such reward (Eq 7) models the lane-keeping task. The episode is terminated whenever timeout is reached or the ego vehicle is crashed. In this paper, we have applied SAC and TD3. Results of different experiments are discussed in Section IV-B.

IV. Simulation Results and Discussion

This section provides the details of experimental parameters and shows the results with discussion.

A. Simulation Parameters

For both simulation the common parameter Steering Angle $\theta \in [-45°, +45°]$, Acceleration Range $a_k \in [-5, +5] m/sec^2$, no. of neighbouring vehicles n is set to 4. For parking environment simulation, the value of parameters are set as
reward weight
W_r = [1, 0.3, 0, 0, 0.02, 0.02], r_{sg} = 0.12 and $spots$=15*2. For highway environment r_c=-1, r_{hs} = 0.2, r_{lc} = -0.05, s_c = 3. The minimum and maximum speed allowed for vehicle are 20m/s and 30m/s respectively. A vehicle has perception distance of 180m hence, can observe the position of other vehicle if fall within perception distance.

For each algorithm, three experiments were done and their mean and variance are shown in the performance graph. In all experiments, a neural network consists of three dense layers with 256 nodes and Relu non-linearity, layer normalization, batch size of 256, replay buffer of 10^6, learning rate=$3*10^{-4}$, and γ=0.99. For SAC in average or discounted reward setting, τ is set to 0.005, and training frequency to 1. For TD3, target_noise_clip=0.5, target_policy_noise=0.2, policy_delay=2, train_freq=100 and Gaussian action noise= $\mathcal{N}$(0,0.2) is considered.

B. Results and Discussion

The goal of each experiment is to find the algorithm that has more precision control capabilities for easier as well as in a complicated scenario. Fig 5 shows the return obtained during training. Though all training graphs are almost the same, the trajectories followed by them are different. Testing trajectories followed by them are shown in the video link[1]. Some experimented approaches even after converging, did not show parking to the landmark and collide with other vehicles so sooner.

We have analysed all models trained on the simple environment when tested on the updated environment without any nearby vehicle's information. SAC+HER has shown outstanding results. The trajectory followed does not use other unoccupied areas for parking, the vehicle rides smoothly without any vibrations, and shows safe parking even the vehicles are parked on both sides of the landmarked location. TD3+HER trained model ride shows vibrations in the moving vehicle, uses other unoccupied areas of parking. Its trajectories are showing iterative behaviour near the landmarked area. SAC uses discounted return that was used to avoid infinite sum in non-episodic tasks. Since car parking is an episodic task and episode completed either on collision or successful parking. Hence we experimented with SAC+HER in an average reward setting. The trajectories followed, do not use other unoccupied parking areas, showing no vibrate behaviour, capable to park if the vehicle is at either side of the landmarked location but cannot be parked if vehicles are parked at both sides. For our next experiment, we have trained the SAC+HER model on the updated environment with the observation of 4 nearby vehicles. The testing results were unexpected and showing that instead of parking on the landmark, the agent learned to collide with the first appearing

[1]https://tinyurl.com/y2z5yejt

nearby vehicle. This happened due to the issue in the reward function equations Eq. 3, Eq 4. After analysing the return from several trajectories performed by the learned model, we found that it happened because the agent achieves more return when it collides with a nearby vehicle than coming on the landmark. Hence, it preferred to collide with a nearby vehicle. We updated these equations as follow and shown successful learned agent in videos.

$$R = R + I_s * 10 \tag{8}$$

$$R = R - I_c * over(v_e, v_o) * 10 \tag{9}$$

Experiments show the generalization capabilities of dual critic based actor-critic algorithms SAC and TD3 and give insight that even training in a simple environment the model can handle unknown complex situations as well. Though, if trained in the complex (proposed updated) environment will show more success than trained in the simple environment as reflected from tabular results.

Table I shows the performance averaged over a thousand episodes during the testing of each approach. In the table, "Steps" shows the average no. of steps taken to complete each trajectory. "Return" shows the average return of thousand episodes. "Success, Collision, and Timeout" show whether the trajectory results in successful completion (reach landmark) or episode termination due to collision or timeout respectively. "Collision near landmark" indicates that the agent reached near the landmark and eventually collides. "SNLC" represents Stop near landmark without collision and shows that the agent reached just near the landmark without colliding. In table, SACHER + Updated_env results were shown for the updated reward equations (Eq.8, Eq.9). The tabular results reflect the training on the proposed updated environment leads to comparatively more successful episodes completion with few collisions.

Fig 5b shows the training results for driving on the highway with continuous actions. Since the actions are continuous, hence agent may drive out of the lane in any direction with varying acceleration. From the graph, it is clear that TD3 converges to the policy that attains low return and two versions of SAC (discounted and average) performed almost the same. TD3 is showing large variance across three instances of experiments. Despite getting some return, and converging earlier during training, testing results of all approaches show that with continuous actions agent has not learned the driving skill, and instead of crossing roundabout, it maintains itself to be near its initial location.

In order to improve the results for highway driving we have updated Eq. 6 in following manner.

$$R = r_c * (I_c) + r_{hs} * r_s + f_t \tag{10}$$

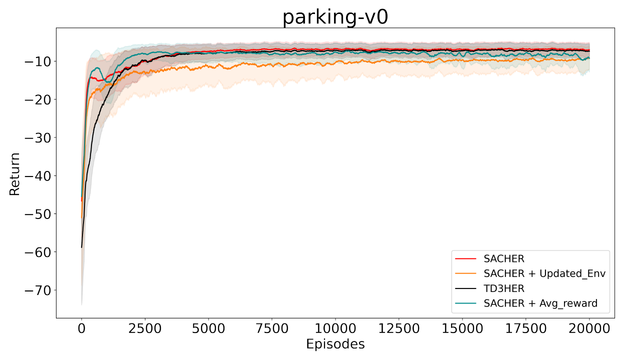
Where r_s is the reward for driving at high speed, which is estimated by a linear mapping of the vehicle's speed from expected high-speed range $(15, 30)$ to $(0, 1)$. The rightmost term of Eq. 6 is replaced with a target following function f_t, which enables the agent to learn driving as it sets the auxiliary goals for the agent to accomplish before reaching the final destination goal. To set the auxiliary goals, initially a route is estimated for the agent to follow. The route defines the ordered list of lanes that the agent needs to follow to reach the goal lane. Each lane of the route sets the target lane for the agent.

$$f_t = \begin{cases} -o_l, & \text{if ego vehicle is out of the lane} \\ r_{ga}, & \text{otherwise} \end{cases} \tag{11}$$

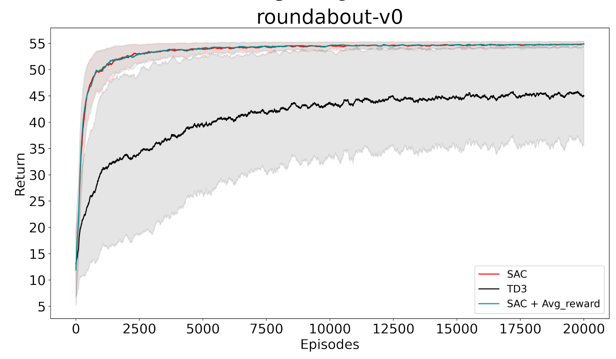
Where, o_l defines the penalty for moving out of the lane and r_{ga} shows the reward/penalty for learning auxiliary goal. For experiment, we have set o_l to 100. To estimate the value of r_{ga}, distance of vehicle (v_e) from target lane (l_t) is estimated and represented by d. The agent is penalized more if it is far from l_t.

$$r_{ga} = \begin{cases} 1, & \text{if ego is on } l_t \\ 0.5, & \text{else if ego is moving towards } l_t \\ -d, & \text{else if ego is on different lane \& } d \leq \eta \\ -\eta, & \text{otherwise} \end{cases}$$

Where η sets the distance threshold and is set to 10. With this update in the reward function and with $r_c = -50$, $r_{hs} = 0.5$, $a_k \in [-15, 15]$ and increased simulation time, agent performs better driving on highway than earlier. The video results show the behaviour of the agent. Hence the results show the capabilities of DRL algorithms that can learn if the simulation is settled in the correct manner. All video shows the agent's performance in five episodes.



(a) parking-v0



(b) roundabout-v0

Fig. 5: Performance graph on episode basis return

TABLE I: Performance in Parking Environment

Approach	Success	SNLC	Timeout	Collision	Collision near landmark	Steps	Return
TD3HER	535	76	9	376	4	20	-9.2
SACHER	539	126	7	315	13	15.7	-7.3
SACHER + Avg_reward	508	44	17	415	16	18	-9.5
SACHER + Updated_env	823	47	15	114	1	22.5	-2.7

V. Conclusion and Future Direction

In this paper, dual critic based DRL approaches SAC and TD3 are applied for driving in the parking lot and on highway roundabout scenario. The customization is done in the simulation setup and in the reward function to consider continuous action control. The training graph shows that advanced approaches quickly learn to drive. The video results also show the generalization capabilities of DRL dual critic based actor-critic approaches from the simple driving environment to a complex environment. The negative outcome is observed from video results for driving on a roundabout scenario with precision control and further improved results with updated reward function are also shown in videos. In the future, policy learned for parking can also be assessed with other moving vehicles on the parking lot, and for roundabout driving robust DRL approaches can be used for further improvement.

Acknowledgement

This work was presented at the Workshop on "Intelligent Transportation Systems, Intelligent Vehicles and Advanced Driver Assistant Systems for Unstructured Environments (WS05), IV2021"

References

[1] E. D. Dickmanns and V. Graefe, "Dynamic monocular machine vision," *Machine vision and applications*, vol. 1, no. 4, pp. 223–240, 1988.

[2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[3] C. Liu, J. Gao, Y. Bi, X. Shi, and D. Tian, "A multitasking-oriented robot arm motion planning scheme based on deep reinforcement learning and twin synchro-control," *Sensors*, vol. 20, no. 12, p. 3515, 2020.

[4] Z. Hu, K. Wan, X. Gao, Y. Zhai, and Q. Wang, "Deep reinforcement learning approach with multiple experience pools for uav's autonomous motion planning in complex unknown environments," *Sensors*, vol. 20, no. 7, p. 1890, 2020.

[5] S. Kuutti, R. Bowden, Y. Jin, and S. Barber, Phil anwrtxd Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[6] D. Line, "How long does it take to learn to drive?" https://tinyurl.com/y9qxeunz, 2013.

[7] G. N. . M. Limited, "uber-self-driving-car-kills-woman-arizona-tempe," https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe, 2018.

[8] S. Gless, E. Silverman, and T. Weigend, "If robots cause harm, who is to blame? self-driving cars and criminal liability," *New Criminal Law Review*, vol. 19, no. 3, pp. 412–436, 2016.

[9] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," *SAE Standard J*, vol. 3016, pp. 1–16, 2014.

[10] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue *et al.*, "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.

[11] G. Katz, A. Roushan, and A. Shenoi, "Supervised learning for autonomous driving," 2017.

[12] V. Nikulin, A. Podusenko, I. Tanev, and K. Shimohara, "Regression-based supervised learning of autosteering of a road car featuring a delayed steering response," *International Journal of Data Science and Analytics*, vol. 7, no. 2, pp. 149–163, 2019.

[13] M. Wulfmeier, D. Z. Wang, and I. Posner, "Watch this: Scalable cost-function learning for path planning in urban environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2089–2095.

[14] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[15] A. Ferdowsi, U. Challita, W. Saad, and N. B. Mandayam, "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 307–312.

[16] Y. Jaafra, J. L. Laurent, A. Deruyver, and M. S. Naceur, "Robust reinforcement learning for autonomous driving," 2019.

[17] E. Leurent and J. Mercat, "Social attention for autonomous decision-making in dense traffic," *arXiv preprint arXiv:1911.12250*, 2019.

[18] E. Leurent, Y. Blanco, D. Efimov, and O.-A. Maillard, "Approximate robust control of uncertain dynamical systems," *arXiv preprint arXiv:1903.00220*, 2019.

[19] E. Leurent, D. Efimov, and O.-A. Maillard, "Robust-adaptive interval predictive control for linear uncertain systems," *arXiv preprint arXiv:2007.10401*, 2020.

[20] S. Wang, D. Jia, and X. Weng, "Deep reinforcement learning for autonomous driving," *arXiv preprint arXiv:1811.11329*, 2018.

[21] P. Wang, H. Li, and C.-Y. Chan, "Continuous control for automated lane change behavior based on deep deterministic policy gradient algorithm," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1454–1460.

[22] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2765–2771.

[23] S. Gupta, G. Singal, and D. Garg, "Deep reinforcement learning techniques in diversified domains: A survey," *Archives of Computational Methods in Engineering*, pp. 1–40, 2021.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[25] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv preprint arXiv:1802.09477*, 2018.

[26] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, pp. 5048–5058, 2017.

[27] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.