2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Learning faster to perform autonomous lane changes by constructing maneuvers from shielded semantic actions

Dapeng Liu[1,2,3], Mattias Brännström[1], Andrew Backhouse[1], and Lennart Svensson[2]

Abstract— **This paper introduces a new method to solve tactical decision making problems for highway lane changes. In the system design, reference sets for low level controllers are employed to formulate semantic meaningful actions used by reinforcement learning algorithm. Safety is ensured by preemptively shielding the Markov decision process (MDP) from unsafe actions. This frees the agent to focus on learning how to interact efficiently with the surrounding traffic. By introducing human demonstration with supervised loss as better exploration strategy, the learning process and initial performance are boosted further.**

I. INTRODUCTION

Autonomous driving has the potential to improve vehicle safety, reduce congestion problems and lower the cost of transportation [1]. However, for the foreseeable future, autonomously driven vehicles will need to co-exist and interact with human-driven vehicles. Even if it would be desirable to replace all human-driven vehicles, there are so many of them that the process would likely take decades to complete. Moreover, it will take time to develop self-driving vehicles with the ability to drive from any point A to point B in all weather, road and traffic conditions, adding additional time to the co-existence of self-driving vehicles and other road users.

Interacting with other road users, ranging from pedestrians to bicyclists, cars, trucks, and animals is a difficult task. Even though there are traffic rules available, all road users are individuals that interpret and follow them in different ways. Some will always create space for other road users while others can be more obstructive and even reduce maneuver space. This leads us to the question: How should self-driving vehicles interact with other road users? Naturally, this question is hard to answer as autonomous vehicles should interact with individuals that might have different preferences.

Generally speaking, there are a few properties that are desirable, e.g. safety, efficiency and social acceptance. Safety and efficiency is relatively easy to define, e.g. avoid accidents, maximize traffic flow and fuel efficiency. Social acceptance is more difficult to define, which leads us to a demand of exploring actual interactions between road users in the real world. When interacting with others, it needs to be safe and preferably the interaction ability should improve over time for increased traffic flow, social acceptance and improved fuel economy.

During the past few years, many planning and decision making frameworks have been proposed, ranging from model based methods to end-to-end learning [2] [3] [4] [5]. Trajectories that are safe under modelling assumptions can be generated using e.g. model predictive control (MPC) [3]. Learning methods have the advantage of quickly being able to mimic complex human interaction in normal traffic conditions [5]. Attempts have been made to combine learning methods and model predictive control (MPC) by learning high level semantic actions, e.g. *change lane*, and then execute them safely with MPC [4]. One of the remaining challenges is to further improve the learning rate in order to achieve efficient learning in real-time. Specifically, this is very challenging when aiming to safely and efficiently interact in dense traffic scenarios, where a small change in the trajectory can potentially lead to a crash.

The main contribution of this paper is a framework that improves the learning rate for decision-making and trajectory planning in autonomous driving by combining control algorithms with a recently proposed technique, safe reinforcement learning via shielding [6]. In this setting we use shielding to restrict the learning to semantic actions that can be executed safely and comfortably. The main advantage of the proposed framework is that the neural network does not have to learn what is safe or how to drive comfortably, enabling it to fully focus on learning the social interaction. Additionally, as learning space is restricted to safe actions, there are less actions to choose from and hence the learning rate is further improved. Human demonstration is added to the framework to improve the learning rate for complex interactions.

The paper is organized as follows. Section II outlines the system architecture, a lane change problem and our application of reinforcement learning via shielding. Section III describes the simulation environment and results. Conclusions and future work are discussed in Section IV.

II. SYSTEM ARCHITECTURE AND PROBLEM FORMULATION

The architecture proposed in this paper derives from the premise that:

- interaction between road users is difficult to model;

This work is supported by Zenuity AB, Vinnova, and AI Innovation of Sweden.

[1] Zenuity, Gothenburg, Sweden. name.surname@zenuity.com
[2] Electrical Engineering Department, Chalmers University of Technology, Gothenburg, Sweden. name.surname@chalmers.se
[3] AI Innovation of Sweden, Gothenburg, Sweden.

- safety requirements for self-driving vehicles are high;
- there is a need to quickly learn how to interact.

Reinforcement learning is known to be inefficient when it comes to solving problems where the performance requirements are high such as safety in our case [4]. The reason is that the negative reward for an accident would have to be several magnitudes higher than for all other actions, making gradients highly unstable and hard to learn. Moreover, even if it could be learned, the requirements on safety are going to be so tough that collecting safety evidence based on end-to-end arguments alone is not feasible [7]. In an attempt to overcome this problem, shielding the learning task from selecting unsafe actions has been proposed [6]. Thus, learning to select efficient actions is separated from the task of ensuring that the selected action is safe.

The task of ensuring that selected actions are safe can be achieved either by constraining or modifying the action post-selection or by explicitly disallowing unsafe actions to begin with. The type of actions can be either low-level control signals such as requested steer rate and brake torque, trajectories or semantic actions which describe a control action relative to the environment. There exist very good vehicle control algorithms which are model dependent and can convert trajectories or semantic actions into actuation signals, using e.g. MPC. Comparing with a set of semantic actions, trajectories allow for considerable greater degree of freedom in motion planning. However, it comes with the cost that it becomes more difficult to prove that any trajectory satisfying a set of constraints is safe.

Establishing which semantic actions are safe is easier as the set is finite. Semantic actions have the additional benefit of being readily understandable and reflect how human directions are given. If the semantic actions are sufficiently granular they can provide sufficient degrees of freedom. For instance, a decision to change lane can be broken down into a set of instructions such as ”turn on the indicators” followed by ”keep the vehicle to the left in the lane”, ”wait until there is sufficient space in adjacent lane” and finally ”commit lane change”. Additionally, if the actions are sufficiently granular, the response becomes more predictable which makes it easier for a model-free approach to learn how to utilize them.

A. System Architecture

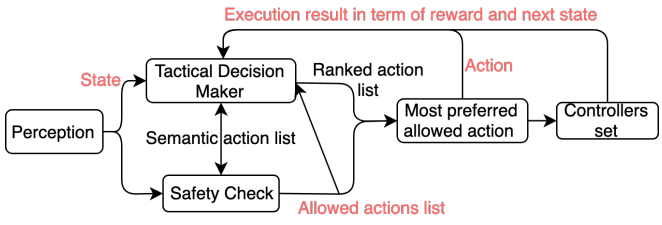


Fig. 1. System architecture. Words in red colour are Markov decision process related information.

Based on the analysis above, we propose an architecture shown in Fig. 1. The perception module provides object level kinematic information of the host and surrounding vehicles to a tactical decision maker and a safety planner. As the focus in this paper is on planning and decision making, we make the assumption that the perception module produces ideal sensor estimates. Sensor limitations could be dealt with by adding safety margins [3], but that is beyond the scope of this paper.

The tactical decision maker and the safety check have a shared list of semantic actions. The safety check has the responsibility to evaluate each action and determine which is safe while the tactical planner only has the responsibility of achieving its goals by selecting among the safe actions. Finally the selected action is provided to an actuation module which converts the chosen action into control signals. In this paper, the set of actions is designed to deal with the narrow problem of performing lane changes. Actions are formulated as instructions to activate and deactivate turn indicators, longitudinal control actions to maintain time gaps with respect to different constellations of other vehicles, and lateral control actions to keep current lane or perform a lane change. The longitudinal controllers are designed to be safe and robust in terms of limiting space error, speed, and acceleration [8]. Further details for these semantic actions will be addressed in Section II-C.2. The safety constraints for these actions are described in Section II-C.3.

In order for the tactical decision maker to learn appropriate actions, a reinforcement learning approach is selected to solve this problem.

B. Reinforcement Learning

Reinforcement learning is a rapidly evolving field of machine learning in which an agent learns to find an optimal policy of choosing actions which maximize its expected return. To promote long-term planning, the return is described by the sum of future discounted rewards. In this paper, reinforcement learning is applied which can be modeled as a Markov Decision Process (MDP) and double deep Q-learning (DDQN) [9] is used to find the optimal policy.

The MDP is defined by a tuple $\langle S, A, R, T, \gamma \rangle$, in which S denotes the states set; A, the actions set; R, the reward function $R(s, a)$; T, the transition functions $T(s, a, s') = P(s'|s, a)$; and $\gamma \in [0, 1]$ is the discount factor. In each state $s \in S$, the agent chooses an action $a \in A$ according to its policy π. After transition to the new state $s' \in S$, a reward $r = R(s, a)$ is received. Let $t \in \{0, 1\}$ denote whether the MDP terminates, the experience tuple is defined as $\langle s, a, r, s', t \rangle$ used for experience replay.

Using the Q-learning algorithm, the action-value function represents the expected accumulated rewards given the policy for each action, defined as:

$$Q(s, a) = \mathbb{E}[r + \gamma \max_{a'} Q(s', a')|s, a, \pi], \quad (1)$$

a' is the according action in state s'. When the action-value function is approximated by a deep network with parameters

$\boldsymbol{\theta}$, it can be represented by $Q(s, a; \boldsymbol{\theta})$. To find the optimal action-value function $Q^*(s, a)$ and optimal policy π^*, the DDQN algorithm minimizes the temporal difference (TD) loss which is defined as

$$l_{td} = (Q(s, a; \boldsymbol{\theta}) - (r + \gamma Q(s', a'; \boldsymbol{\theta'})))^2, \tag{2}$$

where

$$a' = \arg\max_{a \in A} Q(s', a; \boldsymbol{\theta}). \tag{3}$$

In this paper, the neural network used for action-value function approximation has three hidden layers, with 64, 256, 32 neurons accordingly.

C. Collision Free MDP Formulation

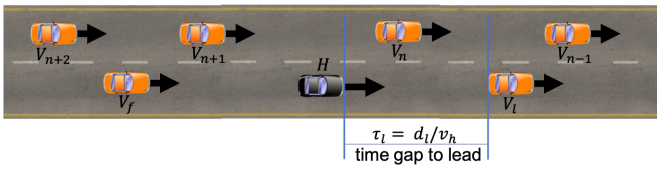


Fig. 2. Scenario, host, and surrounding vehicles.

1) State space: In order for the MDP to infer the intentions of the surrounding vehicles, it needs a short observation history to monitor how other vehicles respond to actions. Therefore, the state s_t contains a list of k observations from nearest past time instances, $s_t = \{O_t, O_{t-1}, ..., O_{t-k+1}\}$. In this paper, three time instances, i.e. $k = 3$, $s_t = \{O_t, O_{t-1}, O_{t-2}\}$ are chosen where each observation is separated by 0.5 seconds.

As shown in Table I, each time instance has information about the host vehicle, from o_1 to o_8, and target vehicles, indexed with i. To set appropriate rewards, the last executed action is included in the state as well as the number of times the host has been overtaken. To monitor interactions, the state vector includes the turn indicator as well as information from the lead vehicle V_l, the following vehicle V_f, and 4 vehicles in the target lane. Here, V_n is the nearest vehicle ahead of host as shown in Fig. 2.

TABLE I

CONTENT OF OBSERVATIONS, HOST AND SURROUNDING VEHICLES

observation	Description
o_1	lateral distance to target lane
o_2	heading angle
o_3	steering wheel angle
o_4	speed
o_5	longitudinal accelerations
o_6	last action
o_7	turn indicator status, on or off
o_8	number of overtake
o_{i1}	longitudinal distance to host
o_{i2}	lateral distance to host
o_{i3}	relative speed to host
o_{i4}	relative accelerations to host

2) Action space: When facing design choices of actions, a hierarchical way of thinking is beneficial. By separating low-level vehicle control tasks and high-level tactical decisions, the learning focus could be on interaction between road users. Hence, each action has a set of control references of both longitudinal and lateral, for the controllers to execute. In total, there are 7 actions, in Table II.

For longitudinal control, two aspects are considered, desired host vehicle set speed v_s and desired time gaps τ to surrounding vehicles. For example, in Fig. 3, action A_2, the host vehicle should follow both V_l and V_n with desired time gap τ_l and τ_n, and drive at desired speed v_s if possible. The controllers calculates the required accelerations a_{τ_l}, a_{τ_n}, and a_{v_s} accordingly. Since it is the responsibility of host to not hit the vehicles in front, then final acceleration is selected by $\min_a\{a_{\tau_l}, a_{\tau_n}, a_{v_s}\}$. The time gap used in action are $\tau_l = 0.5s$, $\tau_i = 0.7s, i \in \{n, n-1, n+1, n+2\}$, and $v_s = 25m/s$. Each action is executed $\Delta t = 0.5s$.

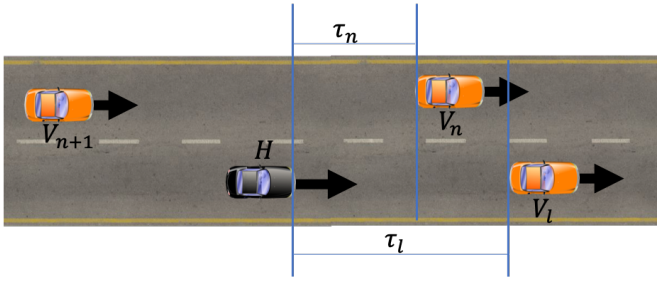


Fig. 3. Control reference set.

Laterally, relative distance to the center of lane, d_{lat} is used by controllers, d^c_{lat} for current lane, and d^t_{lat} for target lane. For the first 5 actions, the lateral target is set to keep the current lane d^c_{lat} , while A_6 is to commit lane change d^t_{lat}. Last action is to turn on/off turn indicator.

TABLE II

ACTION SPACE OF CONTROL REFERENCES

Action	Longitudinal	Lateral
A_1	following lead V_l $\min_a\{a_{\tau_l}, a_{v_s}\}$	original lane d^c_{lat}
A_2	following V_l and V_n $\min_a\{a_{\tau_l}, a_{\tau_n}, a_{v_s}\}$	original lane d^c_{lat}
A_3	following V_l and V_{n-1} $\min_a\{a_{\tau_l}, a_{\tau_{n-1}}, a_{v_s}\}$	original lane d^c_{lat}
A_4	following V_l and V_{n+1} $\min_a\{a_{\tau_l}, a_{\tau_{n+1}}, a_{v_s}\}$	original lane d^c_{lat}
A_5	following V_l and V_{n+2} $\min_a\{a_{\tau_l}, a_{\tau_{n+2}}, a_{v_s}\}$	original lane d^c_{lat}
A_6	commit lane change following V_l and new lead in target lane $\min_a\{a_{\tau_l}, a_{\tau_l^{new}}, a_{v_s}\}$	target lane d^t_{lat}
A_7	turn indicator status, on or off motion control follows last valid action	

3) Shielding to ensure safety: In this work, the safe reinforcement learning via shielding framework proposed

by [6] is adopted. In the paper, two ways to define shield are discussed, preemptive and post-posed shielding. The key difference is that preemptive shielding changes MDP by removing the dangerous actions in each state, while the post-posed shielding maps the dangerous actions to the most preferred available actions in a ranked action list.

In our work, we combine both preemptive and post-posed shielding. Ranked actions from Q-learning and shielding are used to take only the safe action by the preference of Q-value. However, in addition, the estimation of the action-value function takes into account which actions are allowed and is updated by avoiding overestimation from disqualified actions. This is performed by adding the list A_{safe} of safe actions in state s' into the experience tuple $\langle s, a, r, s', t, A_{safe}\rangle$.

When updating the TD loss function in Eq (2), the TD target must only consider allowed actions. Therefore Eq (3) needs to be updated as follows:

$$a' = \underset{a \in A_{safe}}{\arg\max}\, Q(s', a; \boldsymbol{\theta}). \tag{4}$$

This change is needed since changing the available actions in the next state s', is changing the MDP. Consequently, the agent will never encounter an experience tuple containing dangerous actions, nor using actions not available in the next state to evaluate in TD loss. In this MDP, the agent learns each action with no ambiguity in the results, and the action value reflects the behaviour model of the surrounding vehicles.

To determine which actions are in A_{safe} each action needs to be evaluated to determine whether it can lead to a collision. For the actions A_1 to A_5 which adapt speed to keep a time gap, safety can be ensured by choosing a safe controller such as in e.g. [8] and therefore for this problem they are always allowed. However, committing to a lane change might be unsafe. There exist methods to formally determine whether committing to a lane change is safe [3]. Formal proof of safety is outside the scope of this paper and instead a simple approach is adopted where lane changes are disallowed if they will lead to time gaps $\tau_{margin} \leq 0.5s$ or require other vehicles to decelerate by more $a_{min} = -3m/s^2$ to avoid collisions.

4) Reward function setups: Since the safety issue is taken care of by shielding, the reward design is focused on lane change efficiency and comfort. To achieve lane change with comfort, four rewards are engineered with respect to lateral distance, speed, jerk, and a counter for being surpassed.

The first reward is given by

$$r_{lat} = (|d_{lat}|/d_{max} - 1)^2, \tag{5}$$

where d_{max} is the maximum possible distance to the central of target lane and is designed to penalize being in the wrong lane. The second reward is given by

$$r_{speed} = \begin{cases} 1 & v_n \geqslant 2 \\ v_n - 1 & 1 \leqslant v_n < 2 \\ 0 & -1 \leqslant v_n < 1 \\ v_n + 1 & -2 \leqslant v_n < -1 \\ 2(v_n + 1) & v_n < -2, \end{cases} \tag{6}$$

Where $v_n = (v_{host} - v_{average}/c_{range})$ is the normalized host speed, $v_{average}$ is the fixed average speed on the road, c_{range} is a constant value for a speed range. The numbers of intervals are arbitrarily chosen to encourage faster speed. The third reward is given by

$$r_{jerk} = \begin{cases} 1/(j_{max} - c_j) & |j| \geqslant j_{max} \\ 1/(|j| - c_j) & j_{min} \leqslant |j| < j_{max} \\ 0 & |j| < j_{min}, \end{cases} \tag{7}$$

where j is the jerk of host vehicle, j_{min} is the minimum jerk where passenger starting to feel uncomfortable, and j_{max} is the limitation beyond which passenger feels very uncomfortable. c_j is a constant for tuning. Acceleration of host vehicle is bounded by controllers, thus not used for comfort reward. Meanwhile, large jerk mainly comes from changing actions, which the agent should learn to avoid. Finally, the last reward is given by

$$r_{overtake} = 0.9^{n_o} r_{lat}, \tag{8}$$

where n_o is a counter and increases once ever time, the host get surpassed by a vehicle in target lane. This reward is designed to punish host slows down too much and blocking the traffic. The total reward is given by

$$r_{total} = \lambda_{lat} r_{lat} + \lambda_v r_{speed} + \lambda_j r_{jerk} + \lambda_o r_{overtake}. \tag{9}$$

In reward design, $\lambda_{lat} = 1$, $\lambda_v = 0.15$, $\lambda_j = 1$, $\lambda_o = 1$, λ_v is lowered for in the dense traffic, the speed is largely determined by the traffic flow. Yet λ_j and λ_o is set to one, as they balance the trade off between comfort and lane change efficiency. Other parameters in reward are, $d_{max} = 4.625\ m$, $v_{average} = 20\ m/s$, $c_{range} = 2$, $j_{min} = 0.6\ m/s^2$, $j_{max} = 3.5\ m/s^2$, $c_j = 4$.

D. Deep Q-learning from Demonstration (DQfD)

While DDQN algorithm achieves good performance in many applications, it normally take millions of training steps to acquire reasonably good policies [10]. This is typically due to the random exploration in the learning phase [11]. To increase the learning rate, human demonstration can be used as shown in [12][13][14]. In this paper, we use the DQfD method from [10] to achieve better exploration.

While logs from manually-driven vehicles are readily available in the automotive industry, to limit the scope of this paper, a simulator is used to create training data and demonstrate results. In this setting, logs are created by allowing a human to perform lane changes within the same simulator as which the agents are trained. The experience

tuples $\langle s, a, r, s', t, A_{safe}\rangle$, from human play has been saved into human experience buffers similar to the experience buffer created by self learning agents. The human actions, though generally sub-optimum with respect to rewards, can serve as a better way in explorations, and be used for offline learning.

Besides better exploration, it is possible to add a supervised loss to promote human's choice of action a_E given state s and suppressing other actions, based on a margin function $l(a_E, a)$,

$$l_s = \max_{a \in A}[Q(s,a) + l(a_E, a)] - Q(s, a_E), \tag{10}$$

where

$$l(a_E, a) = \begin{cases} 0 & \text{if } a = a_E \\ positive & \text{otherwise.} \end{cases} \tag{11}$$

In total, the final loss function will be a linear combination of TD loss l_{td} and supervised loss l_s,

$$l_{total} = l_{td} + \lambda_s l_s. \tag{12}$$

III. Simulation environment and results

In this section, firstly the simulation environment is introduced. Then the results are shown, which makes a comparison between self-learning, learning from human demonstration and a combination of both. Finally, the performance is compared between with and without shielding.

A. Simulation details

The simulation environment is designed to challenge the vehicle's ability to interact and therefore the surrounding traffic has been selected to mimic dense traffic with tight time gaps. Each vehicle has an intrinsic probability to yield to a turn indicator which is set to 80%. However, they will not yield immediately and each vehicle has a random reaction time before they yield and is given in Table III.

To increase the diversity of the scenario, each target vehicle is initialized with time gap τ, desired time gap τ_d, initial speed v, and desired speed v_s. These are all sampled from random distributions as shown in Table III. Different initial and desired values make the scenario more dynamic, so that the target vehicles are not static but are in pursuit of their targets. To simulate different driving styles, the parameter μ in the sliding mode control [15] for each target vehicle is also drawn from a random distribution.

Each scenario used for training and evaluation is generated with a specific random seed number. To show generalized performance and ensure fair comparisons, the evaluation for all agents is performed on 100 fixed random episodes which are not included in the training sets. The learning rate for all agents are set to be the same, $lr = 10^{-4}$.

TABLE III
Target vehicle behaviour distribution

Parameter	Distribution
reaction time	$t_r \sim \mathcal{U}(0.5, 1.5)\ s$
initial time gap	$\tau \sim \mathcal{U}(0.8, 1.2)\ s$
desired time gap	$\tau_d \sim \mathcal{U}(0.8, 1.2)\ s$
initial speed	$v \sim \mathcal{U}(18, 22)\ m/s$
desired speed	$v_s \sim \mathcal{U}(22, 24)\ m/s$
fierce of control	$\mu \sim \mathcal{U}(0.5, 0.9)$

B. Agents with shielding and self learning

As shown in Fig. 4a, the performance of the agent with self learning and shielding is shown. The total length of training is 250 thousand steps. In each step, a batch of 32 experience tuples is sampled from experience buffer with prioritization and importance weights. Random exploration is performed in the first 10% of the training length, from 100% randomness to 2%, with linear decay. The rest of the training, agent remains 2% random exploration. After around 150 thousands training steps, the agent can solve almost all 100 episodes in the evaluation set. At this point, the Q value for host vehicle has converged and the performance has roughly converged as well. The remaining variance is possibly due to the learning rate is not changed during the whole training.

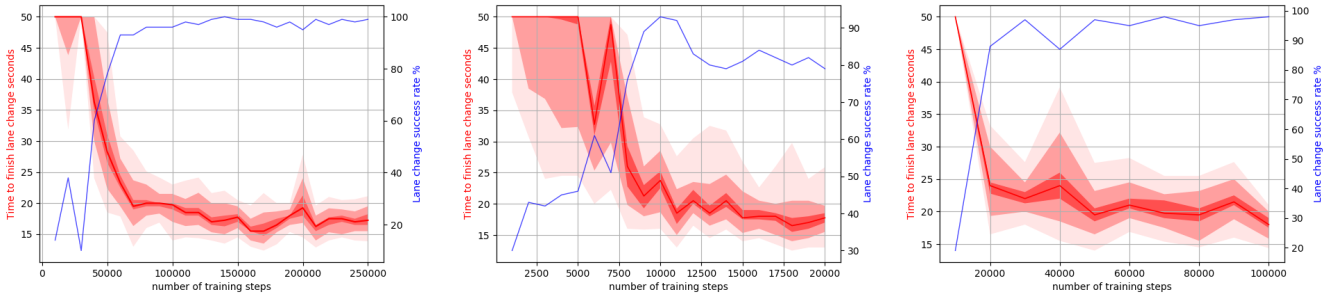
C. Agents with shielding and demonstrations

To perform learning from human demonstration, a human player is asked to play 100 random episodes. These are saved in a human experience buffer. In training, both TD loss and supervised loss in Eq. (12) are used, where $\lambda_s = 0.001$ and $l(a_E, a) = 0.8$. Here λ_s is tuned smaller than in [10] to avoid over fitting as the number of samples is limited. In Fig. 4b, it can be seen that the performance which the agent with demonstration achieves in 10 thousand training steps is roughly equivalent to performance which the self-learning algorithm achieves after 100 thousand steps.

More importantly, the result shows that it is possible to apply off-policy learning and achieve good performance with a relatively small number of samples. With only 100 episodes as demonstration, the agent learns to generalize well and solve 95 out of 100 episodes in test episodes. This could be beneficial when applied to real traffic data. However, with too few experience samples, over-fitting occurs and the lane change success rate starts to drop after 10 thousand training steps.

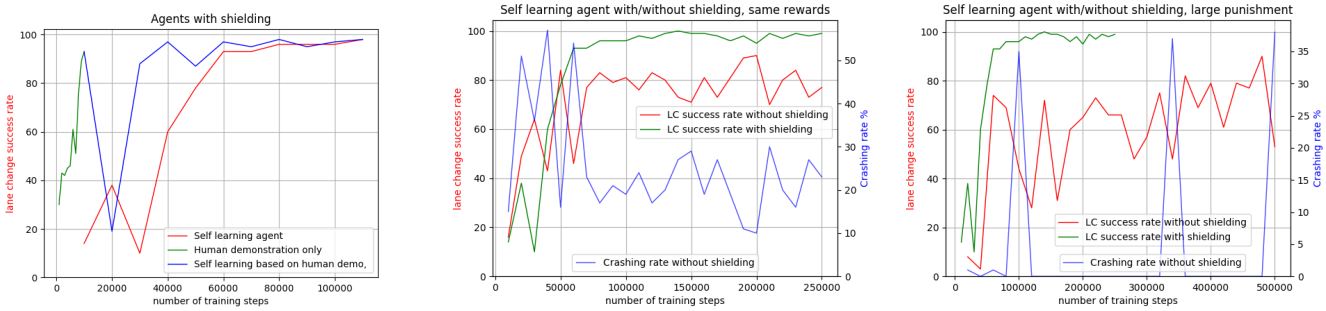
D. Combining human demonstration and self learning

In Fig. 4c, the performance of the agent combining self learning and demonstration is shown. The agent in Fig. 4b at 10k training step is employed when initialize interaction with environment. In the beginning of training, the performance has a quick drop as shown the first evaluation, which is due to intensive random exploration leading the agent to many unseen states and actions, large TD errors of these state action pairs resulting in radical changes in Q value thus



(a) Self-learning agent (b) Agent with only human demonstration (c) Self-learning Agent after training on human demonstration

Fig. 4. Comparing performance (a) Self-learning agent; (b) Only human demonstration with TD loss and supervised loss. (c) Self-learning agent based on human demonstrations. Notice that scale of training steps are largely different between agents for different learning efficiency . Time to finish lane change in seconds are marked with red. Median value in 100 episodes is shown the solid red line. Coloured regions are corresponding to percentile 25-75, 35-65, 45-55, from light to dark.



(a) Compare agents with shielding (b) w/o shielding, same level reward (c) w/o shielding, 100 times larger punishment

Fig. 5. (a) Agent with human demonstration has the highest learning rate, combined with self learning, agent can learn faster than only self learning (b) Agent without shielding cannot learn avoid crashing, the crashing rate is around 20%; (c) Reward engineering cannot solve the crashing problem, even with much larger punishment. Agent behaves more conservatively, but as the training continues, large crashing rate appears every now and then, probably due to catastrophic forgetting.

the driving policy. As only one human player is strongly biased in policy, in addition, human play is more likely to be sub-optimal with respect to reward designed in the MDP, therefore, the initial performance becomes worse than human demonstration. However, it can be fixed with more diversity in human demonstrations and scenarios. The performance quickly becomes better than only human demonstration, and still faster than only with self learning. The learning rate comparison is in Fig. 5a.

E. Comparing with and without shielding

In Figures 5b and 5c, a comparison of lane change success rate is made between with and without shielding. Additionally the crash rate of the self-learning agent without shielding is added. Two agents are trained without shielding. For both, an additional reward of -5 is given when a cut-in causes a time gap of less than 0.5s. Additionally for the agent shown in 5c, all the other rewards are reduced by a factor of 100 such that $\lambda_{lat} = 0.01$, $\lambda_v = 0.01$ in Eq. (9). Neither self-learning agent can solve the crashing problem and both agents learn slower to perform successful lane changes. By designing very large punishment, agent behaves more conservatively in learning and crashing rate is low most of time, as in Fig. 5c. However, there are sudden bursts of high crashing rate. It is probably due to catastrophic forgetting [16], where, as the learning goes on, the proportion of samples containing bad behaviours becoming too small.

IV. Conclusions

In this work, a new method for lane change decision making is proposed. RL algorithms with shielding and human demonstration are designed to learn faster without compromising safety. Control references as actions reduce the action space, and focus on learning interactions. With shielding, safety is separated from learning which makes it faster to train. Furthermore, human demonstrations as part of off-policy learning, can boost the learning speed and initial performance with better explorations. In future work, the system and algorithm can be applied to learn interactions from real traffic data.

References

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Trans-*

portation Research Part A: Policy and Practice, vol. 77, pp. 167–181, 2015.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson, "Lane change maneuvers for automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 18, no. 5, pp. 1087–1096, 2017.

[4] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] N. Kalra and S. Paddock, "How many miles of driving would it take to demonstrate autonomous vehicle reliability," *Driving to Safety*, 2016.

[8] J. P. Maschuw, G. C. Keßler, and D. Abel, "Lmi-based control of vehicle platoons for robust longitudinal guidance," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 12 111–12 116, 2008.

[9] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[10] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband *et al.*, "Deep q-learning from demonstrations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped DQN," in *Advances in neural information processing systems*, 2016, pp. 4026–4034.

[12] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.

[13] M. E. Taylor, H. B. Suay, and S. Chernova, "Integrating reinforcement learning with human demonstrations of varying ability," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 617–624.

[14] K. Subramanian, C. L. Isbell Jr, and A. L. Thomaz, "Exploration from demonstration for interactive reinforcement learning," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 447–456.

[15] T. Tram, A. Jansson, R. Grönberg, M. Ali, and J. Sjöberg, "Learning negotiating behavior between cars in intersections using deep q-learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3169–3174.

[16] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.