Learning Overtaking and Blocking Skills in Simulated Car Racing

Han-Hsien Huang, Tsaipei Wang
Department of Computer Science, National Chiao Tung University
Hsinchu City, Taiwan

***Abstract*—In this paper we describe the analysis of using Q-learning to acquire overtaking and blocking skills in simulated car racing games. Overtaking and blocking are more complicated racing skills compared to driving alone, and past work on this topic has only touched overtaking in very limited scenarios. Our work demonstrates that a driving AI agent can learn overtaking and blocking skills via machine learning, and the acquired skills are applicable when facing different opponent types and track characteristics, even on actual built-in tracks in TORCS.**

Keywords—Q-learning, TORCS, Car Racing

I. Introduction

Games have been popular topics in the study of artificial intelligence, and researchers keep on improving the performance of computers in various games. While board games such as chess and Go have been studied the most, a wide variety of other games have also attracted increasing attention of AI researchers in recent years. In many games, human players need to interact with non-player characters (NPCs), and the performance of these NPCs directly affect the playing experiences of human players. As NPCs with hard-coded behaviors (such as a set of rules) tend to be more predictable, it is natural to use learning to develop NPCs that are more competitive and more fun to play with.

The game discussed in this paper is simulated car racing. Such games provide many challenges for AI research that are different from traditional board games. For example, NPCs in car racing games, which we will call AI-cars in this paper, should be able to drive both fast and safely. However, these are two conditions that sometimes require opposite behaviors. These new challenges have resulted in many research works in this game. Among the many applications of AI in racing games, optimization of NPCs has attracted the most attention. Representative works include [1]-[5], where various computational intelligence techniques (fuzzy logic, neural networks, and genetic algorithms, to name a few) are used to optimize the actions given sensor inputs. The goal in [6] is to find an optimal path using evolutionary computation when the whole track is known. The transfer of driving skills between two racing games is studied in [7]. The imitation of driving behaviors of either other NPCs or human players is the focus of [8]-[10]. For cases when the complete track model is not provided to the NPC, [12] discusses a method for building track models from sensory data.

This paper focuses on the learning of overtaking and blocking skills. While these are important skills in car racing, they pose unique challenges because the driving agent needs to respond and adapt to the behaviors of other cars, at times needing to deviate from the optimal or safe route, therefore incurring extra danger of crashing. As a result, the planning and execution of these skills is significantly more complicated than attempting to drive well by oneself.

II. TORCS and Overtaking/Blocking Behaviors

Similar to most recent works on simulated car racing, we use TORCS (The Open Racing Car Simulator) [12] as the platform for our experiments. It is an open-source platform that realistically simulates Formula One contests. It provides dozens of car models, built-in tracks, built-in AI-cars, and many other details. Most importantly, a researcher can easily implement new AI-cars and analyze their behaviors in this framework, and performances of different AI-cars can be easily compared. Fig. 1 shows an example screenshot of TORCS. Most of the built-in AI-cars implement simple overtaking behaviors based on finding a "free lane" to pass the car ahead. However, they do not possess the more complicated overtaking skills in real races. In addition, none of these AI-cars possess programmed behaviors that will attempt to block opponent cars that attempt to overtake from behind. Instead, these AI-cars would just avoid collision with the overtaking car, sometimes even by yielding. As a result, we believe there are many interesting topics that we can study regarding these more complex driving behaviors.

Among the many research papers on TORCS and car racing games, [13] and [14] are two that mainly deal with overtaking. The main idea of [13] is to use Q-learning to acquire overtaking skills. The experiments contain two scenarios (overtaking on straight sections, and on tight bends using the strategy of break delay), two opponent types (always on centerline and random sideways moves), and two aerodynamic models. Performances are compared with Berniw, a built-in AI-car in TORCS, in terms of time to complete overtaking, top speed, success rate, and probability of crashing. Improvements over built-in Berniw are evident in all aspects, indicating the usefulness of even a simple learning algorithm like Q-learning for acquiring complex behaviors.

The focus of [14] is on overtaking while the opponent has a blocking strategy, which is not covered by [13]. Three blocking strategies are examined: limited blocking, slowly reactive, and fully reactive. The overtaking strategy is represented by a set of

This work is supported by the Ministry of Science and Technology of R.O.C. with grants No. 102-2221-E-009-150- and 103-2221-E-009-130-.

Figure 1. Example screenshot of TORCS.

TABLE I. CHARACTERISTICS OF TEST TRACKS

Name	Length (m)	Radius of Curvature (m)	Width (m)
LS	4408	65	10
R40	1251	40	10
R65	1408	65	10
R90	1565	90	10
R120	1753	120	10
R150	1942	150	10
CG track 3	2843	30-250	10
Ruudskogen	3274	30-750	11

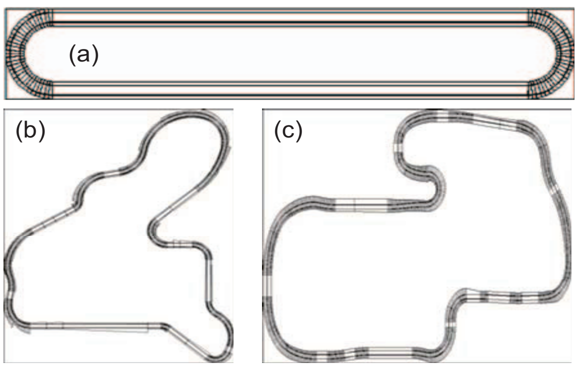

Figure 2. Test tracks. (a) The general shape of oval tracks used in training, (b) CG-track3, (c) Ruudskogen.

hand-crafted fuzzy logic rules. Experiments in [14] indicate that, while all the built-in AI-cars fail badly to overtake opponents with blocking strategies, AI-cars equipped with the overtaking strategies in this paper succeed most of the time. The experiments in [14] are only limited to long straight sections of race tracks, leaving some doubts as to whether the fuzzy-logic overtaking strategies are still useful in more realistic races.

Overall, both [13] and [14] deal with overtaking in very limited scenarios, and questions remain regarding whether those overtaking skills are applicable in more general situations. Generalization is an important issue in any learning system, as the system learned may be of little use if it only works well in scenarios it has faced during the learning process. However, this issue has never been discussed in the very limited literature about the learning of overtaking in car racing. Therefore, we have two major objectives in our study regarding overtaking in car racing: To demonstrate that machine learning can produce useful and sophisticated overtaking behaviors beyond those demonstrated in [13], and to analyze how well the learned skills remain applicable when the testing environments and settings differ from those used during the learning phase. When stating so, we want to note that our main objective regarding overtaking is not to improve upon the methods in [13], but to demonstrate capabilities of those methods that were not evident in [13].

In addition to the learning of overtaking skills, we also want to consider the perspective of the car in front regarding how it can prevent being overtaken by the car from behind. Specifically, we will investigate the learning of blocking skills from scratch as well as how well the learned skills can generalize to new environments.

III. TEST TRACKS

Table I summarizes the characteristics of the test tracks used in our experiments. The tracks used in training all have the general oval shape in Fig. 2(a). One of them is named LS (meaning "long and straight") and, with two long straight sections of 2000m each, is particularly used for the study of overtaking and blocking behaviors along straight sections. The five tracks R40, R65, R90, R120, and R150 also have similar oval shapes as LS, with the numbers representing the radii of curvature of the curvy sections. Their straight sections are only 500-m long, and they are used in the study of overtaking and blocking behaviors that can occur on curves.

All the oval tracks above have a width of 10 m. While widths of the built-in tracks in TORCS generally range between 10 m and 15 m, we choose the narrow end because it is more challenging for the overtaking car. As a matter of fact, we find through experiments that, when the tracks are 15-m wide, learning provides little advantage because an overtaking car with built-in overtaking strategy can overtake the opponent successfully most of the time, which is not the case in 10-m-wide tracks (see Table III below).

In addition to these simple oval tracks, we also use two built-in tracks in TORCS to test whether the learned behaviors work in realistic tracks. Fig. 2(b) and 2(c) display the two tracks selected, CG-Track3 and Ruudskogen, respectively. We choose these two tracks because of widths similar to those of our oval tracks and their moderate level of difficulty.

IV. EXPERIMENTS ON OVERTAKING

A. Base AI-Cars

As the learning here is focused only on overtaking, we need to have an AI-car to start with that is able to drive well when not overtaking other cars. The base AI-car we use is Berniw, a built-in AI-car in TORCS that drives on nearly optimal routes when alone. It also has some simple rule-based overtaking behaviors built-in. Our learning AI-cars use Berniw's mechanisms while driving along and for controlling acceleration and breaking. Machine learning is used to acquire steering behaviors when there is an opponent car ahead.

We also need to have an AI-car acting as the opponent (the car to be overtaken), and for this role we select BT, another built-in AI-car in TORCS that tends to run along track centerlines. Its maximum speed is limited to about 140-150 km/h to allow for many overtaking opportunities, which are necessary for the overtaking car to learn its behavior. In comparison, the top speed of the overtaking car is about 290 km/h. We implement three types of blocking behaviors for the

TABLE II. INTERVALS OF VARIABLES IN Q-LEARNING

Variable	Intervals
d_y (m)	[0 5), [5 10), [10 15), [15 20), [20 25), [25 30)
d_x (m)	(-∞ -10), [-10 -5), [-5 -3), [-3 -1), [-1 0), [0 1), [1 3), [3 5), [5 10), [10 ∞)
d_v (km/h)	[0 10), [10 20), [20 30), [30 50), [50 70), [70 100), [100 150), [150 200), [200 ∞)
p_x (m)	(-∞ -3), [-3 -2), [-2 -1), [-1 0), [0 1), [1 2), [2 3), [3 ∞)

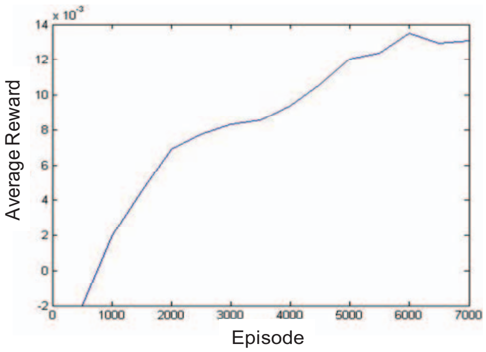

Figure 3. Learning curve of Q-learning on LS.

opponent: non-blocking, fully reactive (the opponent shifts immediately according to the relative cross-track positions of the two cars), and slowly reactive (similar to fully reactive but the opponent waits for one second before shifting). These three blocking strategies are abbreviated as NB, FR, and SR, respectively; strategies FR and SR are the same as those used in [14]. The original programmed behaviors (collision avoidance and yielding) of the opponent AI-cars in TORCS are disabled.

B. Settings of Q-learning

Q-learning is a simple reinforcement learning approach. It works by iteratively updating a Q-table where each entry, $Q(s,a)$, represents the value of taking action a at state s. Our setting of Q-learning is very similar to that in [13]. The update equation of the Q-values is

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha \times [R_{t+1} + \gamma \times (\max_a Q_t(s_{t+1},a)) - Q_t(s_t,a_t)]. \quad (1)$$

Here α is the learning rate (set to 0.5), R is the reward, and γ is the discount factor (set to 0.8).

There are only three possible actions by the learning AI-car: shifting left by 1 m, shifting right by 1 m, or staying on the current path. The states (a total of 4320) are combinations of intervals of the four variables listed in Table II. The following are the meanings of the variables: d_y is the along-track position difference of the opponent AI-car relative to the learning AI-car, d_x is their cross-track position difference, d_v is their speed difference, and p_x is the cross-track position of the learning AI-car. We use smaller intervals than those in [13] in order to model the behaviors at different states more precisely. If d_y becomes negative, the learning AI-car is ahead of the opponent. In this case the overtaking attempt is deemed successful and the learning stops. The reward used in the Q-learning process is +1 for each successful overtaking, −1 if the learning AI-car cannot overtake its opponent within a given time limit (currently set at 40 seconds) or if there is a crash that disables the car, and zero otherwise.

C. Learning to Overtake Different Opponents

Our first experiment focuses on learning overtaking behaviors on the straight sections of track LS; the learning AI-car is forbidden to overtake on the curves. Each episode is an opportunity for the learning AI-car to attempt to overtake its opponent. With a non-blocking opponent and a 40-second time limit, the average reward per step (shown in Fig. 3) approaches equilibrium after about 5000 episodes. This learning curve is consistent with the one in [13]. From now on we use the form $Q_{track,opponent}$ to represent the Q-learning AI-car trained on a particular track with a particular opponent. The three opponent types are represented as NB (non-blocking), FR (fully reactive) and SR (slowly reactive).

Next we apply each Q-learning AI-car (one for each blocker type) to the test of running with its targeted opponent type on track LS. Overtaking is only allowed on straight sections, and the time limit of 40 seconds starts when the velocity of the overtaking car drops below 160 km/h due to the existence of a car ahead (the opponent). Only those overtaking events that occur within this time limit are deemed successful. Table III contains the successful overtaking rates in 200 attempts. The results using the built-in Berniw as the overtaker are also included for comparison. The Q-learning AI-cars clearly outperform Berniw, therefore indicating the effectiveness of learning.

It is interesting to note here that, after inspecting the videos recorded during the trials, we find that our AI-cars exhibit the following maneuver mentioned in [14]: The overtaker can deceive its opponent with a blocking strategy by shifting one side first to attract the opponent to that side, and then quickly swing to the other side to complete overtaking. While this maneuver in [14] is hard-coded, it is acquired by Q-learning in our experiments. In Fig. 4 we illustrate this more "advanced" overtaking strategy using the recorded trajectory of our test. Fig. 4 contains a segment of the trajectories of the overtaking and opponent cars over a span of 14 seconds; the separation between adjacent markers is one second. The track segment shown here is approximately 650 m in distance. The overtaking car is initially at the outside (left) when coming out of the turn. It shifts toward the right edge of the track, attracting the opponent, whose blocking strategy is fully reactive, to shift to the right as well. The overtaking car then quickly shifts to the left and overtakes the opponent before the opponent can block it. Also shown in Fig. 4 are five video screenshots that depict the event temporally.

D. Skill Transferability with Different Blocker Types

Now that the Q-learning AI-cars can overtake their respective opponents successfully, our next question is whether the learned skills are still applicable when facing types of opponent which they are not trained with. In Table IV, we list the successful overtaking rates of all the three Q-learning AI-cars against all three types of blockers; the diagonal contains values in Table III. The setting of this experiment is the same

TABLE III. SUCCESSFUL OVERTAKING RATES OF Q-LEARNING AI-CARS ON TRACK LS

Overtaker	Opponent		
	NB	SR	FR
Q-learning	0.91	0.83	0.80
Berniw	0.14	0.11	0.00

TABLE IV. SUCCESSFUL OVERTAKING RATES OF Q-LEARNING AI-CARS FACING DIFFERENT OPPONENTS

Overtaker	Opponent		
	NB	SR	FR
$Q_{LS,NB}$	0.91	0.55	0.51
$Q_{LS,SR}$	0.81	0.83	0.78
$Q_{LS,FR}$	0.90	0.82	0.80

as that in the previous subsection. There is no learning in this experiment.

The results in Table IV are encouraging in that the Q-learning AI-cars clearly exhibit most of their skills when facing opponents of different blocking types. Even $Q_{LS,NB}$, trained with non-blocking opponents, can overtake more difficult blockers in more than half of the times. On the other hand, $Q_{LS,FR}$, the AI-car trained with the most challenging opponent, has even better success rates when facing less challenging opponents. These results demonstrate that the learned overtaking skills are not particular to one type of opponent. In subsequent overtaking experiments, we will only use the most challenging blocker (FR) as the opponent.

E. Learning to Overtake on Curves

While Q-learning AI-cars have demonstrated their overtaking abilities on straight track sections, in real contests they have to be able to simultaneously handle driving on curves and overtaking an opponent. This is the focus in the experiments in this and next subsections. Five tracks are used for the experiments here: R40, R65, R90, R120 and R150, covering radii of curvature that range from fairly tight (R40) to fairly smooth (R150) for simulated car racing. The straight sections are reduced to 500 m each compared to 2000 m for LS, making it very unlikely that an overtaking event can be initiated and completed within one straight section. Therefore, the overtaking can happen on straight sections, curve sections, or when entering or leaving curves, creating much more diverse scenarios for learning than in previous subsections.

Due to this diversity, in this and subsequent subsections on overtaking experiments, a different (and more realistic) method is used to evaluate the performance of the overtaking AI-car. Instead of providing a fixed number of overtaking opportunities, the overtaking car and its opponent are placed on the track to run continuously for a fixed amount of time. Three values are used for evaluation: total distance travelled, lap difference (indicating the number of successful overtaking), and accumulated damage, although the first two are quite correlated. For those who are not familiar with TORCS, a car is disabled when the accumulated damage value reaches 10000. The amount of time used here is 1500 seconds, which is about enough time to run 50-55 laps on these tracks. The results, averaged over 5 runs, are listed in Table V. For comparison, we also include results using the built-in Berniw as the overtaking car. We can see that our Q-learning AI-cars (in bold face) drastically outperform Berniw in all the tracks.

Since it is impossible to train AI-cars for each different curve in realistic tracks, we want to investigate how well a trained AI-car can perform on different tracks. Here we test all the 5 AI-cars trained on R40 to R150 on all the 5 tracks. The results are in Table VI where, to save space, we only list the lap differences. While the best performer on a track is mostly the AI-car trained on that track, performances of the other AI-cars are not far off. Even in the most extreme cases, namely $Q_{R40,FR}$ running on R150 and $Q_{R150,FR}$ running on R40, the resulting lap differences are at least 60% of the best values on the respective tracks.

F. Overtaking on Built-in TORCS Tracks

Finally, our AI-cars trained on the oval tracks are put to the

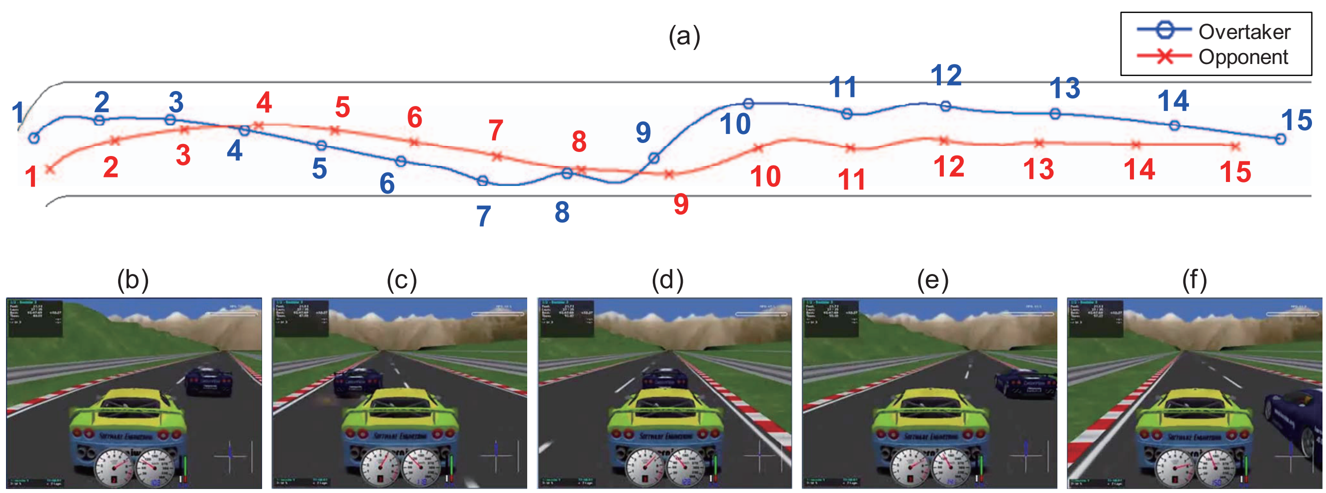

Figure 4. An example of learned overtaking maneuvers. (a) The trajectory of the Q-learning overtaker (blue) and the opponent (red) over a time span of 14 seconds. (b)-(f) The screen shots of steps of the overtaking event in the trajectory. (b) Overtaking car is behind the opponent, somewhat to the left. (c) The overtaking car shifts right. (d) Both the overtaking car and the opponent at the right side of the track. (e) The overtaking car shifts left. (f) The overtaking car passes the opponent.

test of running on built-in TORCS tracks. The two tracks CG-Track3 and Ruudskogen are selected because they are only moderately difficult and have widths similar to the tracks used for training. Even so, they contain curves of a wide range of curvatures as well as continuous curves, making them quite challenging.

The amount of time used here is 2400 seconds to allow for enough overtaking opportunities in these longer tracks. The results are listed in Table VII. The different Q-learning AI-cars all perform similarly, again indicating that they are not very particular to the tracks they are trained on. When compared with Berniw, the Q-learning AI-cars have two more successful overtaking events on CG-Track3 and one more successful overtaking event on Ruudskogen. While the differences are not as dramatic as those in Table V, the Q-learning AI-cars still clearly outperform Berniw.

V. Experiments on Blocking

A. Base AI-Cars

We use Berniw as the base AI-car for the blocker in the experiments on learning blocking behaviors. Similar to the experiments on overtaking, its top speed is limited to 140-150 km/h to allow many opportunities to be overtaken. The built-in crash avoidance and yielding behaviors are disabled. Before any learning, its behavior is identical to non-blockers in the previous section.

Four different AI-cars are used as the overtaking cars in the experiments here. Three of them are TORCS AI-cars with programmed overtaking behaviors: Lliaw, Tita, and BT. The last one is the AI-car with learned overtaking behaviors (from the previous section) against fully reactive blockers.

B. Learning to Block Different Opponents

The Q-learning settings for the blocking AI-car are the same as those used when learning overtaking behaviors; the only difference is that the sign of reward is opposite: −1 for each successful overtaking, +1 if the learning AI-car is not overtaken by its opponent within a given time limit (currently set at 60 seconds) or if there is a crash that disables the car, and zero otherwise. The time allowed is longer here so that the blocker can learn more diverse behaviors over different parts of the track.

Our first experiments here are on track LS. The results of successful overtaking rates over 200 opportunities are listed in Table VIII. Results for a non-blocking AI-car are also included for comparison. We can see that the blockers have acquired useful blocking strategies against these overtaking attempts, allowing successful overtaking in only about one third of the attempts. It is interesting to see that Q-learning overtakers have higher successful overtaking rates than built-in AI-cars. This is probably because the overtaking behaviors of the built-in AI-cars are more limited and predictable, making it easier to learn the blocking strategies against them.

C. Skill Transferability with Different Overtakers

Our next question is whether the learned skills are still applicable when facing types of opponent which they are not trained with. In Table IX, we list the successful overtaking

Table V. Overtaking Performance on Tracks with Different Radii of Curvature

Track	AI-Car	Total Dist. (km)	Lap Difference	Damage
R40	$Q_{R40,FR}$	68	10.0	407
	Berniw	60	0.0	1071
R65	$Q_{R65,FR}$	77	8.8	0
	Berniw	67	0.0	0
R90	$Q_{R90,FR}$	87	8.4	62
	Berniw	76	1.0	0
R120	$Q_{R120,FR}$	98	11.6	0
	Berniw	80	1.0	0
R150	$Q_{R150,FR}$	109	15.6	123
	Berniw	81	1.0	0

Table VI. Lap Differences of Q-learning AI-Cars on Tracks with Different Curvatures

	R40	R65	R90	R120	R150
$Q_{R40,FR}$	10.0	6.8	7.2	8.0	10.0
$Q_{R65,FR}$	7.6	8.8	7.2	10.0	9.6
$Q_{R90,FR}$	7.2	7.2	8.4	8.8	9.6
$Q_{R120,FR}$	7.6	7.0	8.2	11.6	10.0
$Q_{R150,FR}$	6.0	6.6	7.6	8.6	15.6

Table VII. Overtaking Performance on Built-In TORCS Tracks

Track	AI-Car	Distance (km)	Lap Difference	Damage
CG-Track3	$Q_{R40,FR}$	110	4.8	757
	$Q_{R90,FR}$	110	4.8	1177
	$Q_{R150,FR}$	111	5.0	873
	Berniw	103	3.0	166
Ruud-skogen	$Q_{R40,FR}$	121	6.6	418
	$Q_{R90,FR}$	122	6.8	534
	$Q_{R150,FR}$	122	7.0	396
	Berniw	116	6.0	1412

rates of all the three Q-learning blocking AI-cars against the three built-in AI-cars as overtakers. From here on, to avoid confusion with Q-learning overtakers, we use the form $B_{track,opponent}$ to represent a Q-learning blocker trained on a particular track against a particular opponent.

The diagonal in Table IX contains values in Table VIII and exhibits the most successful blocking (lowest successful overtaking rate) for each overtaker. While this is expected, the overall results are encouraging in that the degradation of blocking abilities when facing different overtakers is not significant, indicating the learned blocking behaviors are still mostly useful when facing different overtakers.

D. Learning to Block on Curves and TORCS Tracks

In this subsection, we test whether the blocking skills can be learned and applied to more complicated and diverse track environments. Here the Q-learning blockers are trained on the oval tracks R40, R90, and R150 against two different overtakers: Lliaw, which is somewhat more aggressive then Berniw, and the Q-learning overtakers trained on the corresponding tracks.

Table X lists the performance on the three oval tracks. The evaluation is based on the distance travelled by the opponent, the lap difference between the overtaking and blocking cars, and the accumulated damages over a time span of 1500 seconds; both the distances travelled and the lap differences are the smaller, the better.

TABLE VIII. SUCCESSFUL OVERTAKING RATES AGAINST Q-LEARNING BLOCKING AI-CARS ON TRACK LS

Blocker	Opponent (Overtaker)			
	Lliaw	Tita	BT	$Q_{LS,FR}$
Q-learning	0.28	0.29	0.31	0.37
Berniw (NB)	1.00	0.97	1.00	

TABLE IX. SUCCESSFUL OVERTAKING RATES AGAINST Q-LEARNING BLOCKING AI-CARS FACING DIFFERENT OVERTAKERS

Blocker	Opponent (Overtaker)		
	Lliaw	Tita	BT
$B_{LS,Lliaw}$	0.28	0.35	0.40
$B_{LS,Tita}$	0.36	0.29	0.43
$B_{LS,BT}$	0.34	0.33	0.31

TABLE X. BLOCKING PERFORMANCE ON TRACKS WITH DIFFERENT RADII OF CURVATURE

Track	Overtaker	Blocker	Distance (km)	Lap Difference	Damage
R40	LLiaw	$B_{R40,LLiaw}$	65	7.2	1448
	$Q_{R40,FR}$	$B_{R40,Q*}$	67	9.4	904
		FR	68	10.0	407
R90	LLiaw	$B_{R90,LLiaw}$	83	5.2	522
	$Q_{R90,FR}$	$B_{R90,Q*}$	87	7.6	1029
		FR	87	8.4	62
R150	LLiaw	$B_{R150,LLiaw}$	99	10.8	650
	$Q_{R150,FR}$	$B_{R150,Q*}$	103	14.0	1696
		FR	109	15.6	123

* Here Q represents the Q-learning overtaker trained on the same track with a fully-reactive blocker. For example, for $B_{R40,Q}$, Q is $Q_{R40,FR}$.

TABLE XI. BLOCKING PERFORMANCE ON BUILT-IN TORCS TRACKS

Track	Overtaker	Blocker	Distance (km)	Lap Difference	Damage
CG-Track3	LLiaw	$B_{R40,LLiaw}$	106	4.8	1237
		$B_{R90,LLiaw}$	106	4.4	416
		$B_{R150,LLiaw}$	107	4.8	1289
	$Q_{R40,FR}$	$B_{R40,Q*}$	109	5.0	1924
		FR	110	4.8	757
	$Q_{R90,FR}$	$B_{R90,Q*}$	110	5.0	2098
		FR	110	4.8	1177
	$Q_{R150,FR}$	$B_{R150,Q*}$	110	5.2	1627
		FR	111	5.0	873
Ruud-skogen	LLiaw	$B_{R40,LLiaw}$	120	6.0	254
		$B_{R90,LLiaw}$	121	6.0	0
		$B_{R150,LLiaw}$	120	6.0	506
	$Q_{R40,FR}$	$B_{R40,Q*}$	125	7.0	235
		FR	121	6.6	418
	$Q_{R90,FR}$	$B_{R90,Q*}$	125	7.0	29
		FR	122	6.8	534
	$Q_{R150,FR}$	$B_{R150,Q*}$	123	7.0	840
		FR	122	7.0	396

* Here Q represents the Q-learning overtaker trained on the same track with a fully-reactive blocker. For example, for $B_{R40,Q}$, Q is $Q_{R40,FR}$.

When comparing results obtained by training with Lliaw and Q-learning overtakers, we can see that it is more difficult to learn blocking skills against Q-learning overtakers as their behaviors are more diverse and less predictable. This observation is consistent with the results in Table VIII. We can also compare the performance of our Q-learning blockers and the fully reactive blocker. While the differences are not dramatic, the Q-learning blockers actually slightly outperform the fully reactive blocker on these tracks.

Finally, the Q-learning blockers trained on the oval tracks are tested on the two built-in TORCS tracks, and the performances are listed in Table XI. The main difference from Table X is that the blocking performances of Q-learning blockers are only on par or slightly worse than the performances of fully reactive blocker.

VI. CONCLUSIONS

In sum, this paper describes the application of a very simple machine learning method, Q-learning, to the task of learning overtaking and blocking behaviors in the simulated car racing game TORCS. Compared with previous experimental results on overtaking, we demonstrate that the AI-cars can acquire complex overtaking maneuvers through Q-learning, and the learned skills have good transferability to different opponent types and different track curvatures. These are all important issues affecting the usefulness of our method in more realistic settings. We also demonstrate that Q-learning can also be used to acquire useful blocking behaviors. Finally, our AI-cars are tested on two built-in tracks in TORCS and demonstrate that they still outperform hard-coded overtaking policies of Berniw in overtaking experiments, and have similar performances compared to fully-reactive blockers in blocking experiments.

While the works described in this paper clearly demonstrates the usefulness of machine learning in acquiring overtaking and blocking skills, there remain many interesting problems to be pursued in the future. First, the current learning process only considers the current state. However, an overtaking and blocking event involves a series of actions over some time span, so it should be better if we can use information from more time steps to make the decision. We are also interested in studying how to incorporate expert knowledge into the process to achieve more efficient learning as well as better results.

REFERENCES

[1] D. Perez, G. Recio and Y. Saez, "Evolving a Fuzzy Controller for a Car Racing Competition," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 263-270, 2009.

[2] E. Onieva, D. A. Pelta, V. Milanés, and J. Pérez, "A fuzzy-rule-based driving architecture for non-player characters in a car racing game," *Soft Computing*, vol. 15, pp. 1617-1629, 2011.

[3] E. Yee and J. Teo, "Evolutionary Spiking Neural Networks as Racing Car Controllers," *Proc. Int'l Conf. Hybrid Intelligent Systems*, pp. 411-416, 2011.

[4] L. Cardamone, D. Loiacono, P. L. Lanzi, "Evolving Competitive Car Controllers for Racing Games with Neuroevolution," *Proc. Annual conference Genetic and Evolutionary Computation*, pp. 1179-1186, 2009.

[5] K-J. Kim, J-H. Seo, J-G. Park and J-C. Na, "Generalization of TORCS car racing controllers with artificial neural networks and linear regression analysis," *Neurocomputing*, vol. 88, pp. 87-99, 2012.

[6] M. Botta, V. Gautieri, D. Loiacono and P. L. Lanzi, "Evolving the Optimal Racing Line in a High-End Racing Game," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 108-115, 2012.

[7] L. Cardamone, A. Caiazzo, D. Loiacono and P. L. Lanzi, "Transfer of Driving Behaviors Across Different Racing Games," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 227-234, 2011.

[8] L. Cardamone, D. Loiacono and P. L. Lanzi, "Learning drivers for TORCS through imitation using supervised methods," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 148-155, 2009.

[9] J. Muñoz, G. Gutierrez, A. Sanchis, "A human-like TORCS controller for the simulated car racing championship," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 473-480, 2010.

[10] J. Muñoz, G. Gutierrez, A. Sanchis, "Controller for TORCS created by imitation," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 271-278, 2009.

[11] J. Quadflieg, M. Preuss, O. Kramer, and G. Rudolph, "Learning the track and planning ahead in a car racing controller," *Proc. IEEE Conf. Computational Intelligence and Games*, pages 395-402, 2010.

[12] "The open racing car simulator website." [Online] Available: http://torcs.sourceforge.net/

[13] D. Loiacono, A. Prete, P. L. Lanzi and L. Cardamone, "Learning to Overtake in TORCS Using Simple Reinforcement Learning," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 1-8, 2010.

[14] E. Onieva, L. Cardamone, D. Loiacono and P. L. Lanzi, "Overtaking Opponents with Blocking Strategies Using Fuzzy Logic," *Proc. IEEE Symposium Computational Intelligence and Games*, pp. 123-130, 2010.