Risk-Aware Reward Shaping of Reinforcement Learning Agents for Autonomous Driving

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Risk-Aware Reward Shaping of Reinforcement Learning Agents for Autonomous Driving

Lin-Chi Wu[1], Zengjie Zhang[1*], Sofie Haesaert[1], Zhiqiang Ma[2*], and Zhiyong Sun[1]

Abstract—Reinforcement learning (RL) is an effective approach to motion planning in autonomous driving, where an optimal driving policy can be automatically learned using the interaction data with the environment. Nevertheless, the reward function for an RL agent, which is significant to its performance, is challenging to determine. The conventional work mainly focuses on rewarding safe driving states but does not incorporate the awareness of risky driving behaviors of the vehicles. In this paper, we investigate how to use risk-aware reward shaping to leverage the training and test performance of RL agents in autonomous driving. Based on the essential requirements that prescribe the safety specifications for general autonomous driving in practice, we propose additional reshaped reward terms that encourage exploration and penalize risky driving behaviors. A simulation study in OpenAI Gym indicates the advantage of risk-aware reward shaping for various RL agents. Also, we point out that proximal policy optimization (PPO) is likely to be the best RL method that works with risk-aware reward shaping.

Index Terms—reinforcement learning, reward shaping, autonomous driving, motion planning, risk awareness.

I. INTRODUCTION

Autonomous driving is dedicated to developing an automatic driving system that can substitute human drivers with equivalent performance. An autonomous driving policy should not only avoid severe events like collisions or rule violations but also be aware of upcoming risky events and take action in advance. A typical human driving pattern has been abstracted into a decision-making system that contains four layers: route planning, behavioral layer, motion planning, and local feedback control [1]. The decision maker in each layer is designed to accomplish the driving task subject to certain rules defined at different levels. For example, a route planner is restricted by urban terrain and road conditions. In the behavioral layer, the vehicles should follow certain traffic rules. Also, motion planning of a vehicle is concerned with collision avoidance with obstacles and other vehicles. In industrial applications, however, autonomous driving is categorized into six levels of behaviors [2], [3]. According to the National Highway Traffic Safety Administration in the United States, the autonomous system product that is permitted on the road is of level 2

[4], which means that the system still requires additional assistance from a human expert who perceives the traffic condition behind the wheel.

Reinforcement learning (RL) provides an effective way to automatically learn a driving policy from the interaction with the environment, such as deep-Q-network (DQN) [5], deep deterministic policy gradient (DDPG) [6], and proximal policy optimization (PPO) [7]. RL does not require human demonstrations but the data obtained from the interaction with the environment during the training process [6], [8]. The core idea of learning to drive is about trial and error like the agent playing a game with the environment. Corresponding to the multi-layer design of an autonomous driving system mentioned above, hierarchical RL is popularly applied by decomposing the entire driving task into several subtasks [9], [10]. The decomposition of the driving task is heuristic and usually requires engineering experience and knowledge from experts. Proper reward shaping allows the agent to perform better in complex scenarios [11], [12]. Besides, heuristic models, such as motion primitives, are also used to simplify the learning process [13], [14]. Different from heuristic RL, end-to-end RL uses a deep neural network to learn an overall driving policy that directly maps vision perception to control commands [15], [16]. A review of RL methods used to solve autonomous driving problems can be found in [6]. However, ensuring a risk-free system, i.e., one that does not make harmful or unethical decisions, is still challenging due to the difficulty of describing risky driving behaviors. As a result, one may find that RL-based vehicles often generate aggressive and reckless driving behaviors since the reward does not fully incorporate the potential risks of these behaviors.

Risks usually refer to events that are not fatal to the system for the current moment but may lead to severe consequences in the future. Fig. 1 illustrates several examples of risky events obtained from the *CarRacing-V0* environment provided by OpenAI Gym. Fig. 1a shows the case where the car is driving close to the edge of the road, which makes it more likely that the car runs out of the road, compared to driving in the center of the road. Fig. 1b shows that an obstacle is in front of the car but it still takes a while until the collision actually happens. Fig. 1c indicates that the car is close to the turn where the car is under the threat of vision occlusion. Different from fatal events like collisions, autonomous vehicles under risk are still in safe conditions, but they may run into fatal events in the short future. Vehicles under risk still have the chance to take conservative actions to prevent possible fatal events. Thus, how to be aware of risks and taking proper

[1]Lin-Chi Wu, Zengjie Zhang, Sofie Haesaert, and Zhiyong Sun are with the Department of Electrical Engineering, Eindhoven University of Technology, Netherlands. `l.wu@student.tue.nl`, {`z.zhang3, s.haesaert, z.sun`}`@tue.nl`.

[2]Zhiqiang Ma is with Research Center for Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China, `zhiqiangma@nwpu.edu.cn`.

* Common corresponding authors.

actions to prevent fatal events is an effective way of avoiding severe accidents. Risk-aware technologies have been widely applied to robotic systems, such as estimating external disturbances [17], [18], predicting the likelihood of collisions [19], and handling instantaneous perception [20]. It is very recently that risk-awareness has become an important topic for autonomous driving [21]. Existing work has integrated the risk-awareness procedure into model predictive control (MPC) to solve the motion planning problem for autonomous driving [22]. In [23], risk awareness has been encoded as rule-based specifications. In [24], a stochastic uncertainty model is constructed to quantify risks. However, risk awareness has not been introduced to reinforcement learning (RL) for the training of autonomous driving policies. This can be achieved by encoding the risk-aware specifications into the rewards of the RL agent. Nevertheless, this is a heuristic process where empirical knowledge of reinforcement learning is needed.

<div align="center">

(a) close to edge   (b) obstacle in front   (c) close to the turn
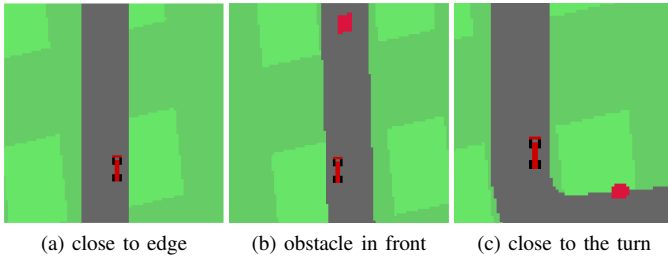
Fig. 1: The illustration of risky events.

</div>

In this paper, we investigate how to use risk-aware reward shaping to improve the capability of RL-trained autonomous vehicles to avoid risky driving behaviors while ensuring sufficient exploration of the environment. Additional reward terms are introduced to encourage explorative driving directions and penalize risky driving behaviors. The reshaped reward is applied to three different agent models, namely a DQN agent, a DDPG agent, and a PPO agent, to validate its general applicability. A comparison study in the OpenAI Gym environment validates that the agents trained with the reshaped reward show better training performance and higher test scores than the ones with the non-reshaped reward. The different effects of the reshaped reward terms on different agents are also addressed, which supports that PPO is the most suitable method for the reshaped reward in the studied case. In general, the main contributions of this paper are summarized as follows.

- Encoding risk-aware specifications into RL for autonomous driving via reward shaping.
- Comparison studies to show to which RL approach the reshaped reward is most effective.

The rest of the paper is organized as follows. Sec. II provides the preliminary knowledge of this paper. Sec. III presents our proposed reward-shaping method. Sec. IV conducts a simulation study to validate the efficacy of the reshaped risk-aware reward. Sec. V concludes the paper.

II. PRELIMINARIES AND PROBLEM STATEMENT

This section introduces the preliminary knowledge of this paper, including the formulation of reinforcement learning and the OpenAI Gym simulation environment. Then, we formulate the main problem to be studied in the paper.

A. Reinforcement Learning (RL)

Motion planning in autonomous driving can be solved using RL through the formulation of a Markov Decision Process (MDP). An MDP is a tuple $M = (S, A, \mathscr{F}, \mathscr{R})$, where S and A are the spaces of states and actions of the agent, $\mathscr{R} : S \times A \to \mathbb{R}$ is the reward function, and $\mathscr{F}(s'_t|s_t, a_t)$ is a distribution that denotes the state transition of the environment, where $s_t, s'_t \in S$ and $a_t \in A$ are, respectively, the current state, the successive state, and the action of the environment at certain time t, where $t = 0, 1, \cdots, T$ and $T \in \mathbb{R}^+$ is the length of the trajectory. The objective of RL is to solve an optimal policy $\pi(s_t|a_t)$ such that the following accumulated reward $J(\pi) = \mathop{\mathsf{E}}_{a \sim \pi} \left(\sum_{t=0}^{T} \gamma^t \mathscr{R}(s_t, a_t) \right)$ is maximized, where $0 < \gamma < 1$ is a discount factor. The optimal policy can be solved with either off-policy methods, such as DQN and DDPG, or on-policy methods like PPO. DQN and DDPG use deep neural networks to approximate the Q-value function $Q^*(s, a) = \max_\pi \mathop{\mathsf{E}}_{a \sim \pi} (\mathscr{R}(s, a))$, $s \in S$, which represents the expected future reward for the current policy π at a certain state $s \in S$. Both approaches approximate the Q-value function using the Bellman equation, $Q^*(s, a) = \mathop{\mathsf{E}}_{s' \sim \mathscr{F}} \left[(\mathscr{R}(s, a) + \gamma \max_{a'} Q^*(s', a') \right]$ which renders an iterative update law. The details of DQN and DDPG methods can be referred to in [25]. Different from them, PPO solves the optimal policy using a policy-gradient search method [26].

B. Problem Statement

In this paper, we consider an autonomous driving task performed in the *CarRacing-V0* environment provided by the OpenAI Gym box2D environment sets [27]. The environment renders a racing car that can run through a 2D map, as shown in Fig 2. More details about this map can be referred to in [28]. In this environment, the car is required to drive along a track with the following requirements satisfied.

Autonomous driving requirements:
- The vehicle should drive along the track and stay inside the track. It fails to survive when driving out of the track.
- The vehicle should avoid collisions with any obstacles showing up on the track. It fails when hitting an obstacle.

An MDP $M = (S, A, \mathscr{F}, \mathscr{R})$ is defined to train an RL agent to achieve these requirements. The state space S is the set of all images showing a bird's view of a car at the bottom middle of the picture with a pixel dimension of $(96, 96)$ in RGB format, saved in the format of grayscale. The action set A contains five discrete actions, namely *no action*, *steer left*, *steer right*, *accelerate*, and *brake*. The state transition $\mathscr{F}$ is prescribed by the OpenAI Gym environment. The main problem to be solved
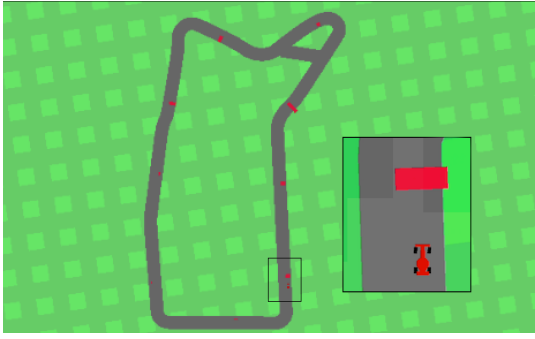
Fig. 2: A screenshot of the *CarRacing-V0* environment, where the track is marked as gray, the green area is the grassland around the track, the red blocks denote the obstacles, and the racing car is marked with a red body with black wheels.

in this paper is to define the reward function $\mathscr{R}$, such that the solved policy π achieves the above-mentioned autonomous driving requirements.

III. REWARD SHAPING FOR AUTONOMOUS DRIVING

This section presents our main methods of risk-aware reward shaping of RL agents. We first extract the essential requirements for autonomous driving from practical experience. Then, we introduce the additional reshaped reward terms based on the encouragement of explorative driving intentions and the penalization of risky driving behaviors.

A. Essential Principles of Reward Shaping

As addressed in Sec. II-B, the essential objective of the problem is to train an RL agent that drives along the track without collisions with the obstacles on the track. The encoding of safety requirements, such as collision avoidance, has been well-studied in the existing RL methods. However, risky driving behaviors are rarely incorporated by the conventional reward-shaping approaches. For autonomous driving, a risk may not be fatal but should not last for a long time. An example of a risky event might be when the car drives too close to the edges of the track. In this case, the car is still safe but it has a higher possibility of running out of the track than driving in the center of the track. Thus, a proper action of the vehicle is to drive back to the center of the track instead of maintaining the original direction. In the meantime, RL requires that the agent performs a reasonable balance between the exploration of the environment and the exploitation of the historical interaction with the environment. Getting stuck in the visited states is also a risk that prevents the agent from sufficiently exploring the environment. Incorporating driving risks into motion planning is promising to effectively improve the reliability of driving. Specifically, positive rewards should be added for explorative driving intentions that lead to unvisited states. On the contrary, risky driving behaviors should receive mild negative rewards to encourage risk mitigation. These principles are explained in detail as follows.

B. Encouragement of Explorative Driving Intensions

The training of the RL agent should encourage the exploration of the car in the environment. Specifically, the car should try to sufficiently explore the environment by driving along new paths. This is a practical principle to avoid the agent being stuck in the visited states. The OpenAI Gym provides an internal API to check whether a state has been visited in the history. We add a positive reward $r_t^{\mathrm{exp}} > 0$ whenever the current state is recognized as *unvisited*. The principle is summarized as follows.

Encouragement principle:

- The agent receives a positive reward r_t^{exp} when the agent is visiting the new path.

C. Penalty of Risky Driving

Alongside encouragement, penalties are also needed to discourage dangerous or risky driving behaviors. In the literature, penalties are mostly raised for collision avoidance since it is the biggest concern of autonomous driving. Similar to most of the existing work, we impose a considerable negative reward $r_t^{\mathrm{obs}} < 0$ on the agent when it collides with an obstacle on the track. Moreover, we also consider the case where the car goes out of the track and drive into the grassland. This case is a risk but not a fatal event since the agent can still survive and come back to the track within a finite time. Therefore, different from the collisions with the obstacles, we penalize the agent for going out with a mild negative reward $r_t^{\mathrm{risk}} < 0$. We also imposed a soft penalty of $r_t^{\mathrm{alive}} < 0$ for each step where the agent remains inside the track, hoping to stimulate the agent to earn a positive reward by exploring new paths. The penalty principles are summarized as follows.

Penalization principle:

- A considerable penalty $r_t^{\mathrm{obs}} < 0$ is given to the agent when the car collides with the obstacle.
- A mild penalty $r_t^{\mathrm{risk}} < 0$ is given when the agent encounters risky events, i.e., getting close to the edges.
- A mild penalty $r_t^{\mathrm{alive}} < 0$ is added for each episode step.

D. Liveness Reshaping

Based on the encouragement and the penalization principles introduced in Sec. III-B and Sec. III-C, we propose the following risk-aware reward function for the training of an autonomous driving agent,

$$\mathscr{R}(s_t) = r_t^{\mathrm{exp}} + r_t^{\mathrm{obs}} + r_t^{\mathrm{risk}} + r_t^{\mathrm{alive}}, \tag{1}$$

where the reward terms $r_t^{\mathrm{exp}}, r_t^{\mathrm{obs}}, r_t^{\mathrm{risk}}, r_t^{\mathrm{alive}}$ are only added to (1) when their corresponding principles in Sec. III-B and Sec. III-C are true.

The liveness of the agent, i.e., the number of steps of each episode should also be carefully determined. If the liveness of the agent is too short, the agent may be *short-sighted* as to lose the capability to evaluate risks. For example, an agent with short liveness tends to drive along the diagonal direction of the track (white arrow in Fig. 3) instead of the parallel

direction along the track bank (red arrow in Fig. 3) since it can not distinguish between the possible consequences of the two directions. In other words, it can not foresee the risk of going out of track after a longer time duration. In this sense, the length of liveness N_{eps} should be properly selected. Moreover, the duration of risks should also be limited. Therefore, we propose the following termination conditions to prescribe the liveness of the car.

Termination conditions:
- The car collides with an obstacle.
- The episode reaches the number limit N_{eps} of steps.
- The risky event lasts for more than a fixed time T_{risk}.
- The accumulated reward reaches a positive value R_{up}.
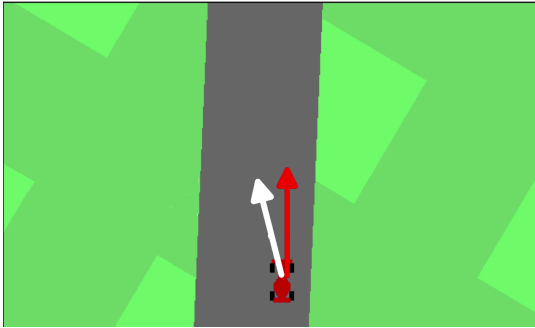- The accumulated reward reaches a negative value R_{down}.

Fig. 3: Short liveness may produce an agent that drives along the diagonal direction of the track (white arrow) instead of moving straightforwardly along the track (red arrow).

IV. EXPERIMENTAL STUDIES

In this section, the performance of the agents with the reward shaping is evaluated in the *CarRacing-V0* simulation experiment. We will also discuss the strengths and weaknesses of different RL agents with the reshaped reward function. The code of the experimental studies is available at [29].

A. Experiment Configuration

The experiment study is run on a Micro-Star International (MSI) laptop with a 12th Intel i7-12700H CPU and an NVIDIA GeForce RTX 3070 Laptop GPU. The initial positions of the car are randomly sampled. Ten possible initial conditions are shown in Fig. 4 as examples. The randomness of the initial locations can help the agent to explore the environment sufficiently. The sampling rate of the environment is 50 frames per second, and the action is performed at each frame sampling. Therefore, the average sampling time of the environment is $0.02\,\text{s}$.

To address the importance of each reward term in (1), we make a comparison between the default and the reshaped reward functions. The values of the reward terms for the two reward functions are presented in Table I. Compared to the default reward function, the reshaped reward function enlarges the absolute values of r_t^{exp} and r_t^{risk}, aiming at valuing more on the encouragement of exploration and the penalization on
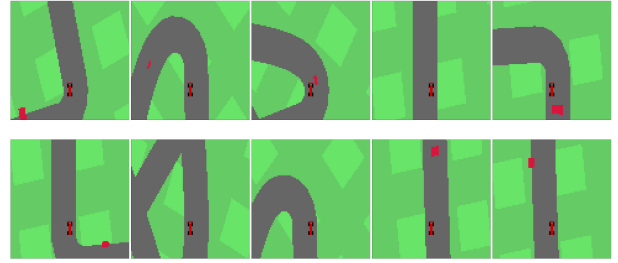

Fig. 4: The random initial conditions of training and test.

driving risks. Besides, the reshaped-reward agent is given a larger episode size to make it longer-sighted. The collision-avoidance and the liveness reward r_t^{obs} and r_t^{alive} are set the same for both reward functions since their functionalities have been well studied in the literature. Also, for both cases, the risk tolerance time is $T_{\text{risk}} = 5\,\text{s}$, and the reward limits are $R_{\text{up}} = 3000$ and $R_{\text{down}} = -400$. The comparison study is performed on three different RL agents, namely a DQN agent, a DDPG agent, and a PPO agent.

TABLE I: The default and the reshaped reward functions

reward function	r_t^{exp}	r_t^{obs}	r_t^{risk}	r_t^{alive}	N_{eps}
default	+1	-600	-1	-1	700
reshaped	+1.4	-600	-200	-1	1200

B. Training Results

The training performance of the two reward functions and three RL agents is illustrated in Fig. 5. In both figures, we notice that PPO reaches the highest training scores. It also achieves the largest performance increase from the default reward function to the reshaped reward function, among the three agents. This reveals that PPO is very sensitive to exploration encouragement and risk penalties, probably because of the usage of policy gradients. The DQN agent has lower training curves than the PPO. Its greedy search strategy makes it short-sighted, lacking the capability of exploration. DDPG has the lowest training rewards but with the most steady changes. This is because the actor-critic structure ensures steady learning progress which, however, costs time to achieve high training scores. The training results indicate that PPO is likely to be the best RL method for this autonomous driving study. However, a conclusion about the efficacy of reward shaping by comparing the two figures can not be drawn, since the two figures are defined by two different rewards. Further test study is needed to address the efficacy of reward shaping.

C. Test Results

In the test study, all trained agents were experimented on the same track separately and iterated for 1000 episodes. For each test episode, the initial position of the car is also randomly sampled. There is no limitation on the number of steps. Instead, a test episode ends when the cumulative reward reaches the limits $R_{\text{up}} = 3000$ and $R_{\text{down}} = -400$ which are
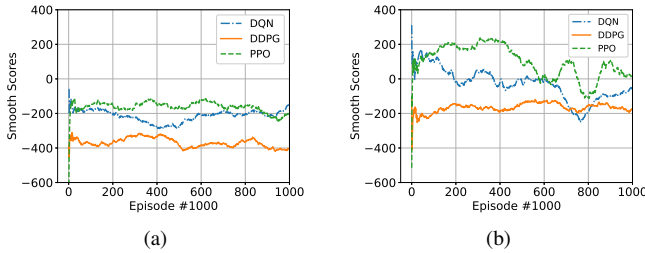
Fig. 5: The training curves of the three agents: (a). the default reward function. (b). the reshaped reward function.

the same as the training process. For each test episode, we use T_{srv} and R_{cum} to represent the total time of the episode and the corresponding cumulative reward, respectively. Then, we use the following three scores to evaluate the test performance of the three agents trained using the default reward and the reshaped reward functions.

- The out-of-track (o²t) counts N_{o^2t}: the number of instants when the car drives out of the track;
- The out-of-edge (oe) counts N_{oe}: the number of instants when the car drives close to the edge;
- The surviving time T_{srv} (s): the total surviving time of the car over the 1000 episodes.

The test scores of the three agents are presented in Table II.

TABLE II: The test scores of the three agents.

Scores	Reward Functions	DQN	DDPG	PPO
N_{o^2t}	default	0	771	721
	reshape	0	769	5
N_{oe}	default	47	3172	5323
	reshape	42	3156	3177
T_{srv}	default	1628	1278	1654
	reshape	1693	1328	3706

From Tab. II, we can see that the reshaped reward in general leads to a longer survival time T_{srv} and fewer o²t and oe counts. A longer survival time and fewer o²t counts represent the safer execution of the autonomous driving task, while fewer oe counts indicate the better handling of risky events. Therefore, the overall results reveal the advantage of the proposed reward-shaping method based on encouragement and penalization for all three agents. Also, we can notice that the DQN agent achieves the best overall performance among all three agents. Meanwhile, PPO achieves the largest performance increase after reward shaping, which is similar to the training result discussed in Sec IV-B. The DDPG agent achieves mediocre performance which does not change much after reward shaping, which is consistent with the training results.

Nevertheless, only the test scores are not sufficient to address the quality of the trained policies of the agents. We further investigate the action counts of the agents in the test to find out whether the agents are taking effective actions instead of lazy actions like *no action*. Fig. 6 shows the total number

of different actions that the three agents have taken during the test study. It can be noticed that the reshaped reward makes the DQN agent more conservative since it simply always takes *no action* for all test steps. Therefore, the policy of the DQN agent is invalid even though it achieves the best scores in Tab. II. This implies a negative effect of the reshaped reward on DQN. On the contrary, PPO shows a more practical driving strategy and becomes more aggressive after reward shaping, since the number of *braking* increases a lot. This indicates that it is taking practical actions to accomplish the driving task. More *braking* actions mean that the PPO agent tries to prevent the car from going off the track or colliding with obstacles immediately when the agent accelerates. Different from DQN and PPO, a DDPG agent corresponds to a more even distribution of different actions, and this distribution does not change much after the reward shaping. To sum up, we can conclude that the encouragement of exploration and the penalization of risky driving positively contribute to the performance promotion of PPO, but negatively affect DQN. Meanwhile, their influence on DDPG is very limited.
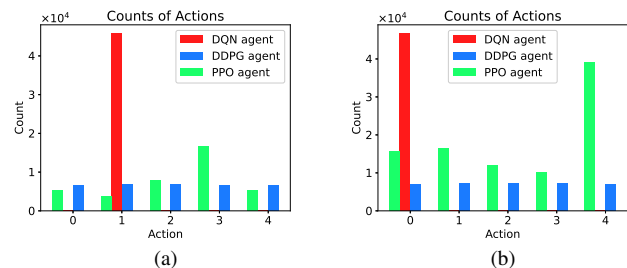

Fig. 6: The counts of actions of the three RL agents where the number 0 to 5 denote the discrete actions *no action*, *steer left*, *steer right*, *accelerate*, and *brake*, and (a) is for the default reward function and (b) is for the reshaped reward function.

D. Discussion of Reward Shaping for Three RL Agents

In this subsection, we summarize the results we have seen in the experimental studies and discuss how the proposed reward-shaping method can be extended for general autonomous driving applications. The results indicate that the DQN algorithm exhibits the lowest performance compared to the other algorithms. This is mainly because its greedy-search-based policy is *short-sighted*, which makes it unsuitable to incorporate exploration- or risk-related rewards. As a result, the trained DQN agent cannot establish a steady policy for navigating the racing track and is prone to remain in the same position while taking no valid actions. Conversely, the DDPG algorithm, which employs continuous action spaces, can effectively learn to navigate the track by accruing positive rewards through successful road traversal. However, DDPG may require an extended number of episodes to achieve a decent performance due to its actor-critic structure. The effects of the exploration encouragement and the risk penalization are not obvious if the training is not sufficiently long. By contrast,

the PPO algorithm exhibits excellent performance and great score promotion after the proposed reward shaping. This is mainly because the proximity guarantee of the policy updates ensures that the agent has steady policy promotions. Therefore, PPO is the most promising RL method that works with risk-aware reward shaping. Nevertheless, a drawback of PPO is that the agent halts the forward decisions too frequently, resulting in the agent moving forward at a lower speed than the DDPG agent. This is mainly due to the aggressiveness of the learning strategy of PPO. Additional reward terms can be introduced to penalize aggressive actions to make the agent more conservative.

V. Conclusion

This study investigates how to use risk-aware reward shaping to improve the performance of three widely-used RL agents, namely DQN, DDPG, and PPO, in solving the autonomous driving problem in a simulated environment. We propose a risk-aware reward-reshaping function to encourage exploration and discourage driving risks. PPO implies the biggest promise to work with the proposed risk-aware rewards. Nevertheless, risk-aware reward shaping still requires heuristic knowledge to penalize the aggressive driving strategies of the PPO agents, which is a challenging topic. We will also incorporate additional rewards in complex forms, such as temporal logic specifications, to solve more complicated autonomous driving tasks.

References

[1] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2020.

[2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] S. International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *SAE international*, vol. 4970, no. 724, pp. 1–5, 2018.

[4] N. H. T. S. Administration. The evolution of automated safety technologies. [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety#resources

[5] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, "Autonomous braking system via deep reinforcement learning," in *2017 IEEE 20th International conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 1–6.

[6] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[7] Y. Wu, S. Liao, X. Liu, Z. Li, and R. Lu, "Deep reinforcement learning on autonomous driving policy with auxiliary critic network," *IEEE transactions on neural networks and learning systems*, 2021.

[8] B. Osiński, A. Jakubowski, P. Zięcina, P. Miłoś, C. Galias, S. Homoceanu, and H. Michalewski, "Simulation-based reinforcement learning for real-world autonomous driving," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 6411–6418.

[9] C.-C. Hsu, L.-W. Kang, S.-Y. Chen, I.-S. Wang, C.-H. Hong, and C.-Y. Chang, "Deep learning-based vehicle trajectory prediction based on generative adversarial network for autonomous driving applications," *Multimedia Tools and Applications*, vol. 82, no. 7, pp. 10 763–10 780, 2023.

[10] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[11] R. Gutiérrez-Moreno, R. Barea, E. López-Guillén, J. Araluce, and L. M. Bergasa, "Reinforcement learning-based autonomous driving at intersections in carla simulator," *Sensors*, vol. 22, no. 21, p. 8373, 2022.

[12] F. Ye, X. Cheng, P. Wang, C.-Y. Chan, and J. Zhang, "Automated lane change strategy using proximal policy optimization-based deep reinforcement learning," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1746–1752.

[13] M. V. Pedrosa, T. Schneider, and K. Flaßkamp, "Learning motion primitives automata for autonomous driving applications," *Mathematical and Computational Applications*, vol. 27, no. 4, p. 54, 2022.

[14] M. Sheckells, T. M. Caldwell, and M. Kobilarov, "Fast approximate path coordinate motion primitives for autonomous driving," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 837–842.

[15] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 537–547, 2020.

[16] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, 2020.

[17] Z. Zhang, M. Leibold, and D. Wollherr, "Integral sliding-mode observer-based disturbance estimation for euler–lagrangian systems," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2377–2389, 2019.

[18] Z. Zhang, D. Wollherr, and H. Najjaran, "Disturbance estimation for robotic systems using continuous integral sliding mode observer," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 14, pp. 7946–7966, 2022.

[19] Z. Zhang, K. Qian, B. W. Schuller, and D. Wollherr, "An online robot collision detection and identification scheme by supervised learning and bayesian decision theory," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1144–1156, 2020.

[20] C. Li, Z. Zhang, A. Nesrin, Q. Liu, F. Liu, and M. Buss, "Instantaneous local control barrier function: An online learning approach for collision avoidance," *arXiv preprint arXiv:2106.05341*, 2021.

[21] X. Huang, A. Jasour, M. Deyo, A. Hofmann, and B. C. Williams, "Hybrid risk-aware conditional planning with applications in autonomous vehicles," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3608–3614.

[22] T. Kim, H. Lee, and W. Lee, "Physics embedded neural network vehicle model and applications in risk-aware autonomous driving using latent features," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4182–4189.

[23] T. Nyberg, C. Pek, L. Dal Col, C. Norén, and J. Tumova, "Risk-aware motion planning for autonomous vehicles with safety specifications," in *2021 ieee intelligent vehicles symposium (iv)*. IEEE, 2021, pp. 1016–1023.

[24] M. Khonji, J. Dias, R. Alyassi, F. Almaskari, and L. Seneviratne, "A risk-aware architecture for autonomous vehicle operation under uncertainty," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, pp. 311–317.

[25] H. Tan, "Reinforcement learning with deep deterministic policy gradient," in *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*. IEEE, 2021, pp. 82–85.

[26] T. Zhang, C. Yuan, and Y. Zou, "Research on the algorithm of constant force grinding controller based on reinforcement learning ppo," *The International Journal of Advanced Manufacturing Technology*, pp. 1–14, 2023.

[27] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. (2016) OpenAI gym. Publisher: arXiv Version Number: 1. [Online]. Available: https://arxiv.org/abs/1606.01540

[28] Woodcock, Mike, "An improvement of gym." [Online]. Available: https://github.com/NotAnyMike/gym

[29] L.-C. Wu and Z. Zhang, "Risk-Aware Reward Shaping of Reinforcement Learning Agents for Autonomous Driving," Aug. 2023. [Online]. Available: https://github.com/zhang-zengjie/code_2023_iecon_shaping_wu