Published online: 1 November 2024 DOI: 10.54254/2977-3903/13/2024137

A lightweight, easy-integration reward shaping study for progress maximization in Reinforcement Learning for autonomous driving

*Hongze Fu [1], Kunqiang Qing [2, a, *]*

[1] School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, China
[2] Automotive Software Innovation Center (Chongqing), Chongqing, China

a. kunqiang@pusan.ac.kr
* Corresponding author

Abstract. This paper addresses the challenge of sample efficiency in reinforcement learning (RL) for autonomous driving, a domain characterized by long-term dependencies and complex environments. While RL has shown success in various fields, its application to autonomous driving is hindered by the need for numerous samples to learn effective policies. We propose a novel, lightweight reward-shaping method called room-of-adjust to maximize learning progress. This approach separates rewards into continuous tendency rewards for long-term guidance and discrete milestone rewards for short-term exploration. Our method is designed to be easily integrated with other approaches, such as efficient representation, imitation learning, and transfer learning. We evaluate our approach on a hill-climbing task with uneven surfaces, which simulates the spatial-temporal reasoning required in autonomous driving. Results show that our room-of-adjust reward shaping achieves near-human performance (81.93%), whereas other reward shaping and progress maximization methods struggle. When combined with imitation learning, the performance matches human levels (97.00%). The Study also explores the method's effectiveness in formulating control theory, such as 4-wheel independent drive (4WID) systems. With reduced spatial-temporal reasoning, reward shaping can match human performance (89.7%). However, control theory cannot be trained together with complicated spatial-temporal progress maximization.

Keywords: Reinforcement Learning, Autonomous Driving, Proximal Policy Optimization (PPO), End to end learning, Reward Shaping

1. Introduction

Autonomous driving is a long-term dependent issue with a complex environment, requiring agents with spatial-temporal reasoning. Reinforcement learning has achieved numerous successes from trial and error. However, one of the critical challenges to adopting RL in autonomous driving is sample efficacy. RL's learning process requires numerous samples to learn a reasonable policy. Such an issue is complex due to sparse and delayed reward, high-dimension of observation, action, and state space in autonomous driving problems.

Many methods have been proposed to maximize RL's progress in previous works. Efficient representation: Due to inherent deficiency in sampling, Reinforcement Learning in autonomous driving requires low-dimensional state and action representations to succeed [1]. Raw data gives finer resolution but condensed abstracted data reduces the complexity. Thus a mid-level representation is mostly adopted [2, 3], compresses exploration space with expert-prior knowledge, and reduces action space by adopting predefined skills. Chae et al. [4] designed a collision-only memory to pool experience effectively and facilitate a control policy specifically for breaking scenarios. The world model is compressed to facilitate fast training of the generative recurrent neural networks [5]. However, efficient representations require task-specific prior knowledge that needs a universal approach. RL+IL: Imitation Learning aims to mimic expert behavior for a given task. It suffers from data mismatch but is powerful at generalizing tasks. Reinforcement learning is more robust because it utilizes sequential actions instead of instant ones [6]. Combining IL with RL, Lu et al. [7] successfully address more challenging scenarios of autonomous driving, reducing the likelihood of collision. Nilaksh et al. [8] introduce a barrier function for safe and stable RL training. However, according to [9], human expert dataset scarcity, temporal difference, and safety issues. Also, efficient IL methods like Behaviour Cloning are unable to copy complex human demonstrations, but powerful strategies like Direct Policy Learning and Inverse Reinforcement

Learning are challenging to train, Transfer Learning: uses the prior knowledge obtained from the source domain to guide the inference in current tasks. It also effectively addresses the domain gap between diverse driving environments [10-12]. The Study of [13] shows that excluding excessive training time, RL shows strong transfer learning capability in adapting to different racing tracks. In order to resolve the data scarcity of edge cases, Some researchers adopt the simulation of crash cases to generalize collision avoidance [14], indicating its transfer learning capability. However, the need for well-labeled complex scenario data and the poor performance of unsupervised transfer learning require further methodology improvement [15].

In this paper, the author endeavors to design a lightweight, easy-implement reward-shaping method for boosting progress that can be easily integrated with other approaches. Reward shaping can boost progress or enable safe training by carefully designing reward functions [16]. For safety training, some works propose to boost RL performance by calculating and penalizing rule-based risk behaviors [17], which are not imminently fatal but could lead to severe outcomes. LSTM [18] predicts behavior and a potentially fatal accident at the same time and minimizes accident prediction [19] using a rule-based safety module to regulate RL output and data-driven safety module to facilitate safe RL training. Traditional methods often highlight travel distance and target speed for progress maximization to provide dense rewards [20, 21].

Our approach focuses on room to adjust for a prolonged time and space horizon, arguing that providing the agent with room to adjust can facilitate better performance. Our method separates the reward function into continuous tendency and discrete milestone rewards. Specifically, the continuous tendency reward serves as a guideline for the correct direction for a long time horizon, The sparse milestone reward recognizes the agent's behavior for this short duration and allows the agent to explore freely during that time.

We researched hill-climbing the asks on uneven surfaces, which requires long-term dependent actions with spatial-temporal similar to other autonomous driving tasks. Our room-of-adjust reward shaping performed human-like performance (81.93%), while other reward shaping and progress maximization performed poorly. Human performance can be matched with other progress maximization, specifically Imitation Learning (97.00%). In increased-complexity action spaces like 4WID, the room-of-adjust method showed increased procedural performance but cannot exploit its potential to facilitate more significant progress.

2. Relevant basic principles

2.1. Reinforcement Learning

Reinforcement learning (RL) is concerned with the Markov Decision Process (MDP), which can be formulated in $\mathcal{M} = (S, \mathcal{A}, \mathcal{F}, \mathcal{R})$ where S and A are, respectively, the spaces of agent states and actions. $S * A \to \mathcal{R}(s, a, s')$ denotes the Reward function. $\mathcal{F}(s_{t+1}, s_t, a_t)$ serves the probabilistic state transition distribution, in which $s_{t+1}, s_t \in S$, and $a_t \in A$. The objective of the RL can be formulated as formulating optimal policy $\pi(s_t, a_t)$ such that cumulative reward $\mathcal{J}(\pi) = \sum_{t=t}^{T}(\gamma * \mathcal{R}_t)$ is maximized. Γ is the discount factor that emphasizes imminent, low-risk rewards.

2.2. Proximity Policy Optimization

Proximity policy optimization (PPO), proposed by [22] is an on-policy algorithm that directly optimizes policy function through a first-order trust-region optimization method. Given the θ parameterized policy $\pi(s_t, a_t)$ and the trajectories $\mathcal{J}(s_t, a_t)$, PPO optimizes the $\pi(s_t, a_t)$ as follows. We formulate the Advantage Function as follows: $\mathcal{A}_t(s_t, a_t) = \mathcal{J}_t(s_t, a_t) - V_t(s_t, a_t)$ in which $\mathcal{J}_t$ is the expected return starting from s_t, taking action a_t and then following the current policy π. V_t is the averaged expected return starting from s_t and following the current policy π. In other words, the Advantage function describes the relative performance gain if an action a_t is performed. Importance function $\mathcal{P}(\theta) = \pi(s_t, a_t)/\pi(s_t, a)$ denotes the probability of selecting action a_t under the current policy. Then, the policy π is updated via $\pi_{new} = \pi_{old} + \min(\mathcal{P}(\theta) * \mathcal{A}_t, \text{clip}(\mathcal{P}(\theta), 1 - \xi, 1 + \xi) * \mathcal{A}_t)$, which is updated by importance-weighted advantage with proximity clipping term. The clipped term keeps the update in a trust region, allowing stable updates.

2.3. Reward shaping

Reward shaping is a technique inspired by animal training to make supplemental incentives or penalties to guide an efficient agent learning process. According to [23], scarce environmental rewards fail to facilitate effective policy in domains that contain subgoals or human-level reasoning. Reward shaping can be formulated into: $R'(s, a, s') = R(s, a, s') + F(s, a, s')$ In which R' is the shaped reward function, R is the original reward function and F is the shaping term incorporating human knowledge. Human knowledge can be categorized into domain knowledge and world knowledge. Domain knowledge is task-specific knowledge that can provide ambiguous but heuristic guidance across different states. For example, when the humanoid agent is learning to walk, domain knowledge can be provided by putting feet on the ground alternatively. World knowledge is some state transitions regulated by real-world physics that can be provided as rules to regulate undesired behavior.

For example, autonomous driving agents should learn to avoid collisions or collision-possible situations. By integrating human knowledge, reward shaping can significantly accelerate learning, improve sample efficiency, and guide the agent toward more human-like or interpretable solutions.

3. Materials and methods

3.1. Simulation platforms

Unity3D is a real-time 3D development platform with a rendering and physics engine. Allowing high-fidelity sensors and real-life physics with complex agents and tasks. MLAgent is an open-source project that enables games and simulations to serve as environments for training intelligent agents. Agents can be trained using reinforcement, imitation, neuroevolution, or machine learning methods. Both training and evaluation can be completed in the games.

3.2. Experiment task

To directly demonstrate progress maximization as travel for autonomous driving, instead of a traditional race-around-track and traffic simulation environments, we design this hill-climbing task on uneven surfaces, as shown in Figure 1. Like other autonomous driving, agents also perform long-term dependent tasks with spatial-temporal understanding. Moreover, the performance of the whole episode can be reflected in travel distance. In other words, the travel distance can directly reflect the agent's performance in progress maximization.

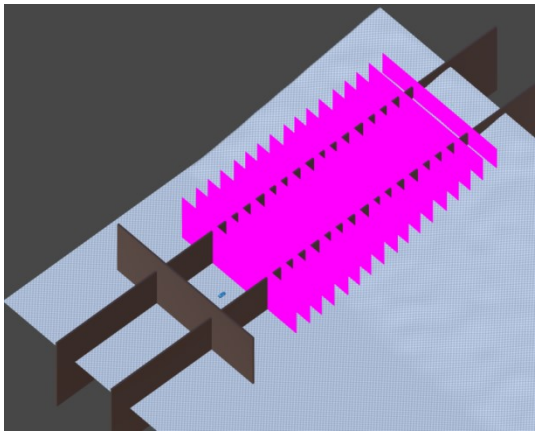

Figure 1. Demonstration of hill-climbing task on uneven surfaces.

The terrain is set with constant graduation, and a Perlin-noise-introduced surface is used. Perlin-noise enables the surface to be both low-traction and consistent across the terrain. Formulation of surface and terrain is afterward.

3.2.1. Perlin Noise surface

PerlinNoise was initially adopted for mountainous terrain generation to generate a low-adhesion surface by reducing the noise scale, as depicted in Figure 2(a).

Perlin noise is generated by assigning a random gradient vector and a random height to each integer coordinate. The gradient possesses a unit length and a random direction. This random height determines the height of the integer coordinate, and for non-integer coordinates, it is upsampled, considering the nearest integer coordinate. The Perlin Noise algorithm is pseudo-random. The author aims to generate Formula 1 to enhance the terrain's robustness, which brings true randomness into the terrain.

$$\sigma(x, y) = f\{x + \varepsilon(0, \Delta)\} \tag{1}$$

Where f{a} is a generated pseudo, and ε(x) is a generated randomized actual number from 0 to Δ.

3.2.2. Procedural generated terrain

The height of each point is calculated using the following formula:

$$H(x,y) = \frac{\sigma(x,y)}{\rho} + \delta(x,y) \tag{2}$$

Where H(x, y) is the height of each point, σ(x, y) is the height variation created by PerlinNoise, ρ is the magnitude of PerlinNoise, and δ is the predefined Heightmap. This formula produces a consistent gradient slope interspersed with randomized bumps. Figure 2(b) presents the algorithmically generated Falloff Map. Additionally, Figure 2(c) illustrates the algorithmically generated Perlin Noise

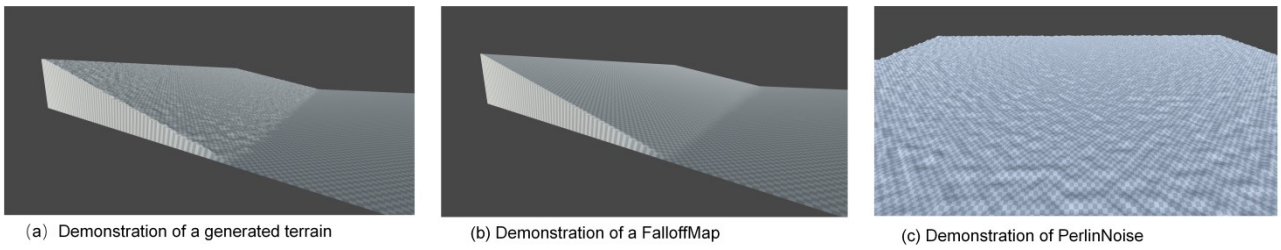

Figure 2. Demonstration of procedural terrain generation; (a) Demonstration of a generated terrain; (b) Demonstration of a FalloffMap; (c) Demonstration of a PerlinNoise Generated flat terrain.

3.3. Agent configuration

3.3.1. Environment perception

In our work, the agent perceives the environment with LiDAR sensors. Tags can also be detected to differentiate whether this object is a "Wall" or "Checkpoint." The position of 6 LiDAR sensors is distributed evenly with 60-degree intervals. The positions of the "Wall," "Checkpoints," and LiDAR were displayed in Figure 3 relative to the car's initial spawn position. In addition to sensors, we also provided the NN network with additional distance data of its position compared to the next checkpoint.

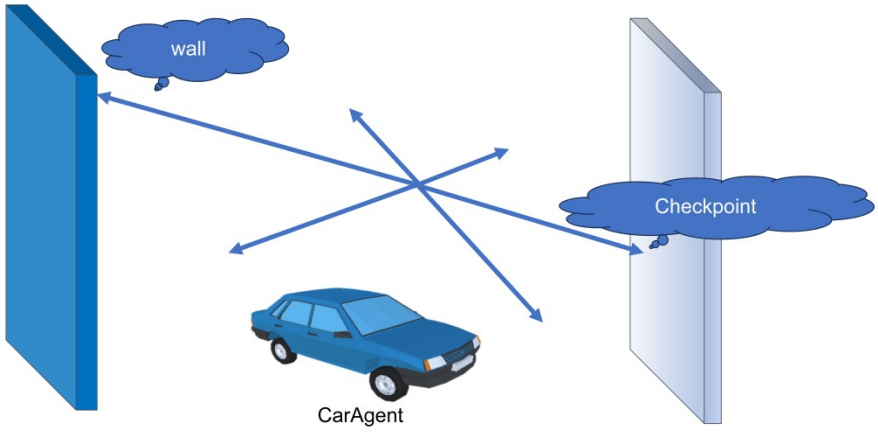

Figure 3. Positioning of ray perceptron 3D sensor.

3.3.2. Related physics

The terrain is uneven, so wheel suspension and friction characteristics are essential. Such behavior can be formulated as follows:

3.3.2.1. Suspension character

$$T_{force} = \alpha T_{depth} * \beta T_{speed} (\text{if } T_{depth} > T_{depthmax} \text{ then } T_{depth} = T_{depthmax}) \tag{3}$$

Formula 3 represents that the live travel speed and travel depth of suspension affect the force output of the suspension. In our settings, the α is set to 0.4 and β 0.8 as default, in line with the recommendation of the unity environment

3.3.2.2. Friction character

$$f = \begin{cases} E_{out} * [-(f_{in}/E_{in})^3 + (f_{in}/E_{in})^2 + (f_{in}/E_{in})] \text{ if } f_{in} < E_{in} \\ -2 * (A_{out} - E_{out}) * \frac{f_{in}-E_{in}}{A_{in}-E_{in}}^3 + 3 * (A_{out} - E_{out}) * \frac{f_{in}-E_{in}}{A_{in}-E_{in}}^2 + E_{out} \text{ if } E_{in} < f_{in} < A_{in} \\ A_{out} \text{ if } f_{in} > A_{in} \end{cases} \quad (4)$$

The wheel's forward and sideways friction is calculated in a three-stage manner, with different rolling and sliding friction combinations. The first stage $f_{in} < E_{in}$ represents that friction force is proportional to the given slip force, representing pure rolling friction, and the linear velocity of the wheels is identical to the agent's velocity. The second stage, $E_{in} < f_{in} < A_{in}$, sees more and more participation in sliding friction, and the third stage, $f_{in} > A_{in}$, is purely made of sliding friction force, with the linear velocity of the wheels irrelevant.

In our settings, these parameters are set as Table 1, which aligns with the recommendation of the Unity platform.

Table 1. Friction parameter.

	E_{in}	About	A_{in}	A_{bout}
Tyre Forward grip	0.4	1	0.8	0.5
Tyre Sideways grip	0.2	1	0.5	0.75

3.4. Room-of-adjust reward shaping formulation

In this paper, we consider an autonomous driving task to cross hill-climbing task on uneven surfaces. To efficiently maximize the progress in such a task, the agent should perform the following actions; details of these guidelines can be found in Chapter 3.6, Human Baselines. 1) Focus on Speed on Level Ground; 2) Choose the Easiest Path and Maintain Forward Momentum; 3) Reassess and Adapt

The agent must differentiate between different action spaces: position, speed, and tire grip. For example, in the same position, if the agent possesses speed, its best choice is to keep its momentum and only drive forward and adjust regarding the tire grip. However, if the agent does not possess speed, the best choice is to try to reassess and find a better path forward.

Thus, our reward-shaping method should follow these principles: 1) Provide long-term directional guideline; 2) Recognize positive progress; 3) Encourage time and space exploration, which might sacrifice short-term progress; 4) Enable the formation of the initial strategy.

Our room-of-adjust reward shaping method consists of the three systems, as depicted in Figure 4. Checkpoint system with increasing intervals, as depicted in Figure 5, travel distance encouragement with low reward values, and punishment of hitting the walls.

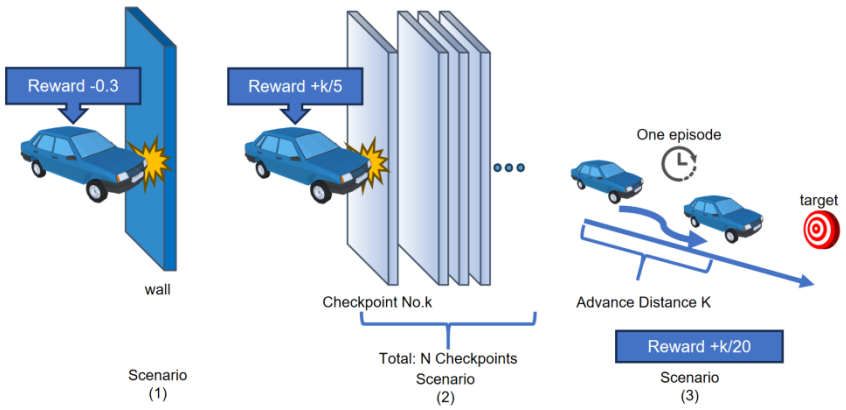

Figure 4. Reward function illustration.

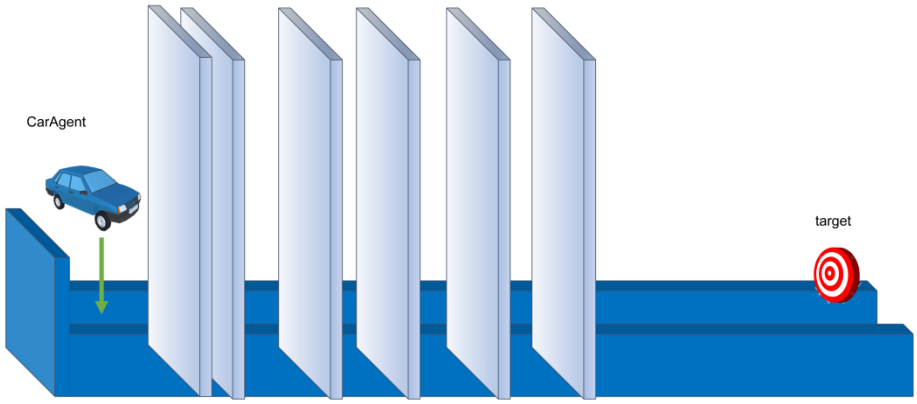

Figure 5. Positioning of checkpoints.

Most of the agent's reward comes from the checkpoint system, and its intervals effectively satisfy principle 3. The agent will only be recognized for its progress when reaching checkpoints and can choose its own strategy during intervals. We also designed it in an increasing interval order. The initial phase's close interval helps the agent learn an imperfect strategy by quickly gaining rewards, satisfying principle 4, as illustrated in Figure 4.

The incremental travel distance encouragement directly indicates the task's goal, serving principle 1. As the agent can be shown with such guidelines, it does not need to understand the environment in a long-term horizon semantically. In other words, it helps the observation space to focus on current short-term tasks. Also, encouraging travel distance helps the agent limit the scope of reassessment and Adaptation.

The determination of the value loss for hitting the wall requires careful tuning. Excessive punishment can initially lead to training stalling, as the strategy may favor not moving to minimize risk. An intense punishment may slow the training without significantly impacting the desired outcome. Therefore, the author opted for a milder punishment to satisfy purpose 3.

3.5. Training configuration

This section describes experiments designed to validate room-of-adjust reward shaping's performance. We first experimented against state-of-the-art methods like Traditional RL, Target-speed-oriented Reward shaping, and Imitation Learning. Then, we tested its performance in combination with state-of-the-art method imitation learning. We also discuss its effectiveness in formulating ambiguous control policies for 4WID.

Hyperparameters. Table 2 outlines essential hyperparameters for the reinforcement learning (RL) framework. Batch_size determines the number of experiences used per gradient descent update, while buffer_size indicates the number of stored experiences, including observations, actions, and rewards before model updates commence. The neural network architecture is defined by hidden_units, specifying units per fully connected layer, and number_layers, determining hidden layer depth after input. At the same time, time_horizon governs the number of steps gathered per agent before inclusion in the buffer, influencing data frequency and volume for learning dynamics.

Table 2. Hyperparameter Configurations.

Network parameter	Value	Algorithm parameter	Value
batch_size	1024	gamma	0.95
buffer_size	5120	epsilon	0.3
hidden_units	264	number_epoch	5
number_layers	3	learning_rate	0.00035
time_horizon	264		

The learning_rate controls the step size in gradient descent for the PPO algorithm. number_epoch specifies how many times the complete experience buffer is cycled through during gradient descent, and epsilon sets the permissible deviation between old and new policies during updates. Gamma is the discount factor for future rewards and is pivotal in RL scenarios.

3.6. Baselines

3.6.1. Human baseline

The environment comprises even ground and gradient hills with uneven surfaces. The following behaviors are preferred. Human demonstration of such behavior can be regarded as our training target and best-case scenario based on multiple trials by human experts.

1)Focus on Speed on Level Ground: Prioritize speed development on flat surfaces. Maximizing velocity in favorable conditions enhances the agent's capacity to navigate challenging terrain. 2) Choose the Easiest Path and Maintain Forward Momentum: Opt for the most straightforward, flattest route available and maintain forward momentum. Leveraging its initial speed, the agent significantly improves its chances of successfully crossing terrain obstacles. 3) Reassess and Adapt: Upon exhausting its initial speed, the agent should consider reversing course to explore alternative routes or focus on regaining momentum to surmount the obstacle terrain effectively.

3.6.2. Speed-maintaining reward shaping

As implemented in [21], the agent is encouraged to reach the target in the given time window. As in our environment, a faster speed is always encouraged at any given moment; we designed speed-maintaining reward shaping to penalize speed under 1m/s and not reward any speed above 1m/s. We tested that this strategy can achieve better performance than only rewarding speed and not penalizing, as the termination travel distance is 35.67, compared to 22.284. Thus, all our following evaluations will be based on the penalizing method.

3.6.3. Traditional RL

We trained RL without reward shaping to evaluate the strength of our room-of-adjust reward shaping. The agent will only receive a reward based on its travel distance.

3.6.4. Imitation learning

Behavior cloning [24] is an efficient imitation tool capable of imitating the demonstrator immediately without interacting with the environment. The agent receives as training data both the encountered states and actions of the demonstrator and then uses a classifier or regressor to replicate the expert's policy.

Generative Adversarial Imitation Learning [25] harnesses generative adversarial training to fit distributions of states and actions defining expert behavior. GAIL is computationally less efficient than Behaviour Cloning but does not suffer from action-drifting issues.

We introduced both Behaviour Cloning and Generative Adversarial Imitation Learning in addition to RL, with the study weight bias set to 1:1:1. Imitation learning is conducted across the whole training process, as we conducted experiments that phasing out imitation learning will cause significant performance deficit and cannot be recovered.

4. Results

Upon completing the training process, we evaluate the performance of each strategy by running 50 tests on the same hill-climbing task in an uneven-surface environment.

The overall performance of progress maximization can be directly indicated by termination distance. In terms of specific behaviour during the process, Trajectory of the car is the indication of path curiosity and environment awareness of easier terrain. Torque serves as an indicator of the agent’s ability to gain momentum on flat ground and to reverse, both of which are aspects of environmental awareness. Speed can be cross-compared to evaluate each strategy's ability to preserve and regain velocity, indicating the stability of the strategy over prolonged time and space horizons.

As mentioned in Sector 3.5, the results are divided into three groups: The first group, which consists of Table 3, Figures 6, 7, 8 and 9, are compared to evaluate effectiveness in progress maximization. The second group, which consists of Table 4, Figures 9 and 10 are compared to evaluate parallel integration capability with imitation learning and the third group, consists of Table 5, Figure 9 and 11 are demonstrated to evaluate the capability of control theory formation.

In Figure 6, Standard RL reached 29.71% of human performance. While some trajectories still ended early due to wall collisions, we observed mild route curiosity. Speed degradation was temporarily mitigated in some instances, but the overall

trend of deceleration remained irreversible, with no ability to regain momentum. RL with speed-maintaining reward shaping achieved 27.82% of human performance in Figure 7. However, trajectories often terminated prematurely due to wall collisions and exhibited limited route exploration. Notably, this method showed no capability for reversing or regaining speed. In Figure 8, The combination of RL and imitation learning unexpectedly resulted in deliberate reverse crashes, with no successful forward progress observed. Our proposed method is presented in Figure 9, RL with room-of-adjust reward shaping, significantly outperformed the previous approaches, achieving 81.93% of human performance. This method demonstrated great route curiosity, indicating an ability to select optimal pathways. Importantly, it showed successful instances of regaining speed after losing momentum. The integration of RL, imitation learning, and room-of-adjust reward shaping yielded the best results in Figure 10, reaching 97.5% of human performance. This approach exhibited improved route curiosity compared to RL with room-of-adjust reward shaping alone. It not only demonstrated successful speed recovery but also maintained a consistent speed of 10 m/s. RL with 4WID reward shaping and turning capabilities achieved 54.26% of human 4WID performance in Figure 11. This method showed mild route curiosity and some ability to regain speed, similar to the RL with room-of-adjust reward shaping approach.

Table 3. Overall performance.

Strategy	Average Travel Distance
Human baseline	120.03
RL+Speed-maintaining reward shaping	35.67
RL	33.395
RL+Imiation learning	-19.633
RL+ room-of-adjust reward shaping (our method)	98.35

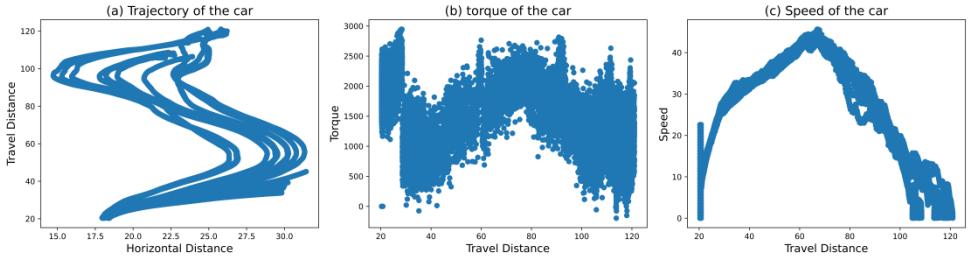

Figure 6. Training Results of RL.

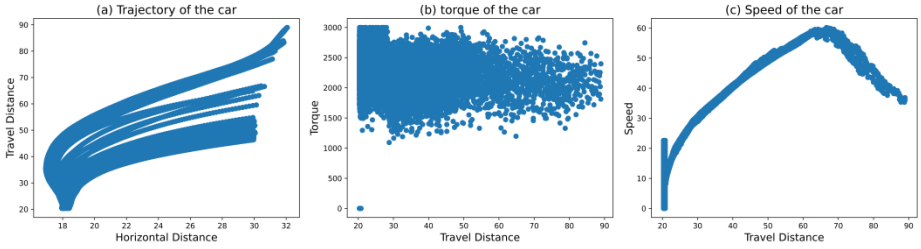

Figure 7. Training Results of RL+Speed-maintaining reward shaping.

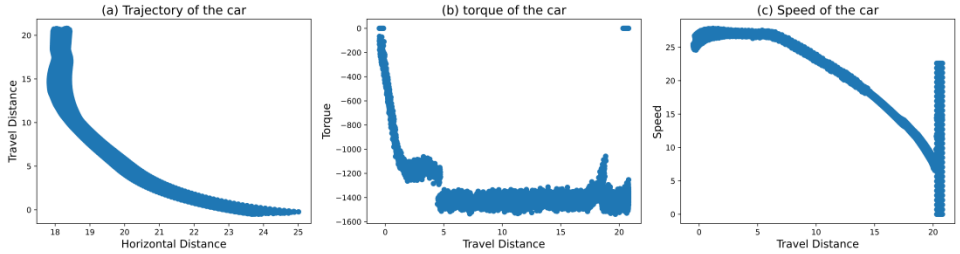

Figure 8. Training Results of RL+ Imitation Learning.

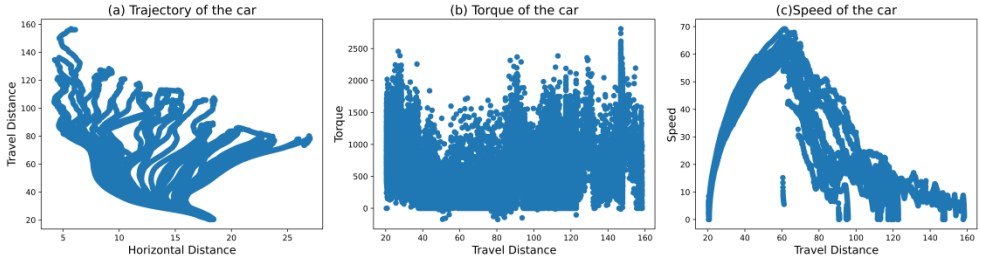

Figure 9. Training Results of RL+ room-of-adjust reward shaping.

Table 4. Overall performance with Imitation Learning.

Strategy	Average Travel Distance
Human baseline	120.03
RL+ room-of-adjust reward shaping(our method)	98.35
RL	33.395
RL+Imiation learning	-19.633
RL+Imiation learning+ room-of-adjust reward shaping	117.03

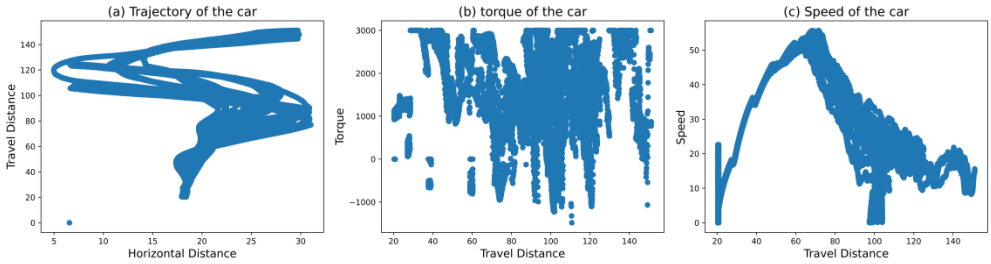

Figure 10. Training Results of RL+ room-of-adjust reward shaping+ Imitation learning.

Table 5. Overall performance with 4WID.

Strategy	Average Travel Distance
Human 4WID With turning	120.03
Human 4WID Without turning	108.78
RL reward shaping With turning	98.35
RL 4WID reward shaping With turning	59.80
RL reward shaping Without turning	84.56
RL 4WID reward shaping Without turning	97.63

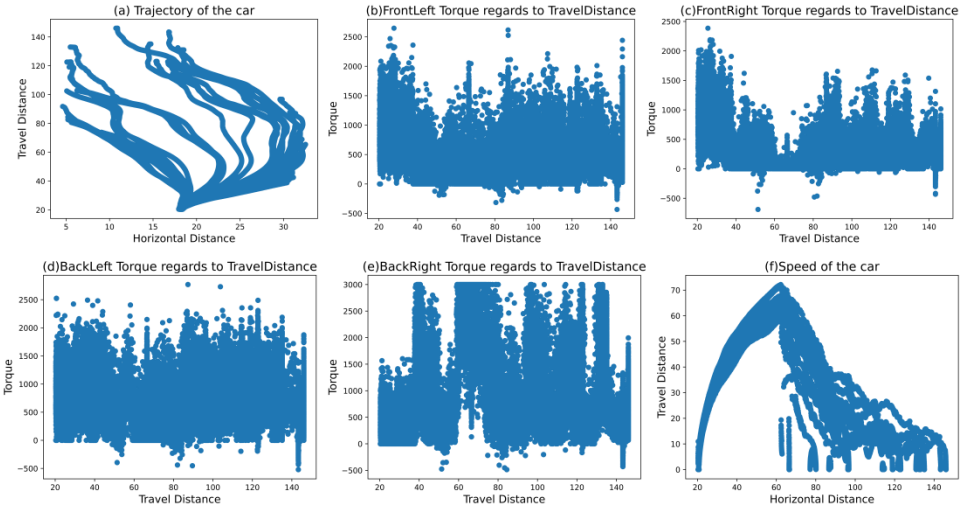

Figure 11. Training results of RL 4WID reward shaping With turning (a) Travel distance; (b) Speed regards to travel distance; (c) FrontLeft wheel power regards to travel distance.

5. Discussion

5.1. Effectiveness in progress maximization

In general, we can achieve 81.6% of the human baseline by empowering RL with our room-the-of-adjust method, while traditional RL can only achieve 27.07% of human performance. Other reward shaping method shows poor performance, Speed-maintaining reward-shaping RL shows marginal improvement in our scenario compared to RL, and imitation learning failed to learn, as shown in Table 3.

For our method, RL+ room-of-adjust reward shaping: Figure 9. Before position 60 on flat surfaces, the agent utilizes more average torque than low-adhesion gradient surfaces, and it is done to build more speed and store additional inertia, aiding the car in overcoming obstacles with limited power and traction. The agent strives to maintain speed as much as possible when initial speed is gained from level grounds, and it is because the agent believes that the initial speed provides the car with an extra capability for travel, and once stopped willingly or due to a collision, this additional capability cannot be regained.

However, after losing its initial speed, the network abandons the previous strategy of maintaining speed (in our sampling results, 1.16% of the torque is distributed backward); the author believes it is because it believes the speed built in this process is not significant enough to make a difference. Instead, when coming to obstacles, the agent tried reversing and advancing as the negative torque in Figure 7. shown. By humanly observing the agent's behavior after training, the scale of the agent's behavior is limited to within the length of the car itself.

In conclusion, the actions performed by the agent successfully learned Guideline 1: Focus on Speed on Level Ground and Guideline 2: Choose the Easiest Path and Maintain Forward Momentum, but Guideline 3: Reassess and Adapt remains limited to a small scale.

The agent can achieve guideline 1 for the RL baseline, but the maximum speed is limited to 45.66m/s instead of 74.57m/s. For guideline 2, about 75% of the agents crashed early into the wall, indicating their inability to find a path. Guideline 3 is almost imperceptible as little to no torque is allocated backward.

For RL+Speed-maintaining reward shaping, compared to the RL baseline, the agent can achieve stable termination distance, and the speed and torque distribution is more centralized, as an explicit reward function is dedicated to improving it. However, similar to the RL baseline, it cannot achieve guidelines 2 and 3. This strategy deliberately hits the wall to terminate early before losing speed, which the author argues as a byproduct of speed-maintaining reward shaping. Such deliberate termination also hampers its ability to explore terrain and choose the best path forward.

For RL+Imitation learning, the agent failed to learn and deliberately crashed into the wall behind. The author poses that it is because the penalty given by imitation learning makes the agent argue that terminating the process early actually preserves the reward. Such an issue is common in other scenarios, as an imperfect and inconclusive demonstration cannot facilitate training.

5.2. Parallel integration capability with imitation learning

Integrating our room-of-adjust reward shaping with imitation learning, the agent overcomes the struggle of imitation learning in training and outperforms room-of-adjust reward shaping alone, with 97.00% of human performance.

In terms of specific behaviors, For guideline 1, the agent shows a much straighter trajectory on flat surfaces; however, the torque distribution is much more inconsistent, resulting in a slower top speed. For guideline 2, the agent demonstrated bigger adjustments. Most importantly, for guideline 3, the agent reallocated reverse torque on a more frequent basis and allowed it to regain speed after being stuck. Also, it learned additional capability of maintaining speed not below 10m/s, as the author argues it can improve the overall cross terrain capability for any given time period, and it is worth of sacrificing route selection, as more speed is naturally bad for changing routes.

5.3. Tests on generating control theory

4WID is a novel electric vehicle that reduces mechanical components, such as differentials and half shafts, and has unique control dynamic advantages [26,27].

4WID systems preserve better control potential but are harder to exploit: Four-Wheel Independent Drive (4WID) system's control potential is demonstrated in many types of research. The potential of the Four-Wheel Independent Drive (4WID), demonstrated by [28], indicates that it can follow trajectories better than the two-wheel model. In comparison to Two-Wheel Steering (2WS) and Four-Wheel Drive (4WD) vehicles, the optimal controller in Four-Wheel Independent Steering and Four-Wheel Independent Drive (4WIS4WID) vehicles exhibits superior performance, leading to a reduction in maximum lateral deviation, as illustrated in [29]

We tested reward shaping's capability to tackle the 4WID control theory. Route planning or turning is an important but intricate part of progress maximization in our environment. When conducted without route planning(turning), 4WID can achieve human-like performance (89.7%). However, in combination with planning, 4WID performed poorer than traditional human input control. In conclusion, reward shaping is capable of generating control theory for 4WID but is unable to train a sophisticated route planning progress maximization at the same time.

The reason why, in combination with planning, 4WID performed poorer than traditional human input control is analyzed as follows: Evidence emerges indicating that the RL 4WID consistently attains speeds exceeding 60 more frequently, accumulating more inertia. However, this heightened speed does not necessarily translate into a greater travel distance; as Figure 11 depicted, the best examples (4 out of 48) are in line with the performance of RL human control, but it suffered from a more significant inconsistency, showing an inability to conclude these best examples. The average and variance are much worse than those of RL human control, as illustrated in Table 5. The underlying reason for this discrepancy lies in the complexity introduced by the control dimensions, which tends to overshadow the progressive improvements

The RL 4WID Strategy also showed increased curiosity in choosing the routes. Figure 12 is a recomposition of Figure 9(A) and Figure 11(A) by removing the vertical axis. By analyzing the horizontal trajectory in Figure 13, the author found the RL 4WID would tend to diversify the routes more; the reasons behind this are unclear. The author hypothesized that due to the PPO algorithm's struggle to find a policy that converges and the cumulative policy fluctuates, it is more likely to exploit other choices due to PPO's stochastic update.

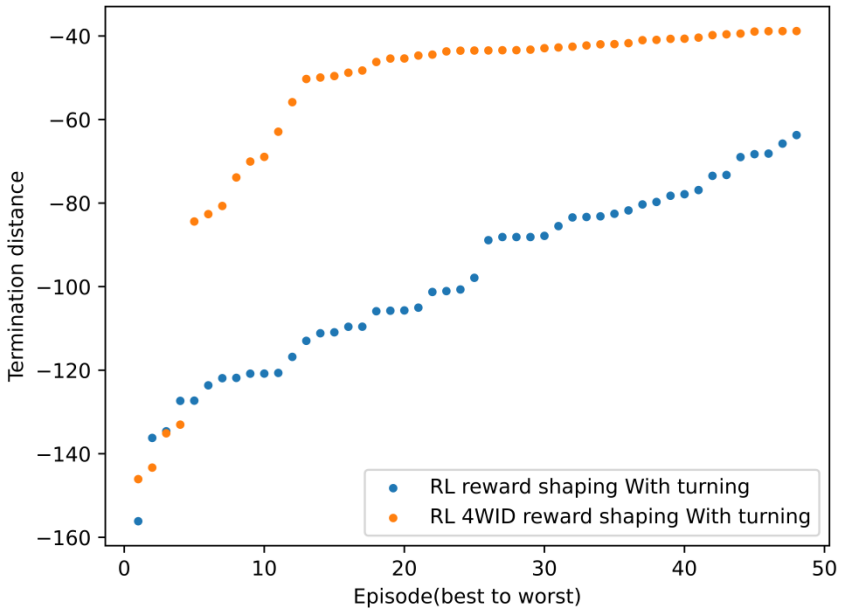

Figure 12. Termination travel distance for 2 Strategies.

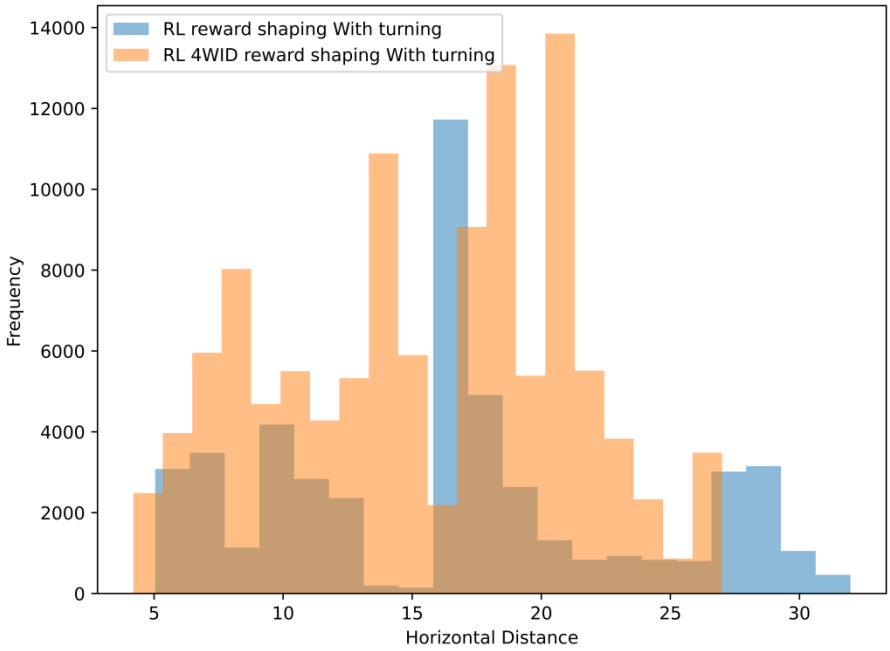

Figure 13. Horizontal trajectory distribution comparison.

The author posits that the performance decline observed in Reinforcement Learning (RL) can be attributed to its struggle in real-world-simulated environments characterized by high-dimensional continuous state and action spaces, as discussed in [30]. Policy-based algorithms like Proximal Policy Optimization (PPO), as highlighted in [31], heavily rely on perceiving and representing complex tasks within deep decision-making models. However, as suggested by [32], PPO only achieves significant convergence with a simplified representation of actions, thereby squandering its control potential.

6. Conclusion

In this Study, the author implemented a room-of-adjust reward shaping method for a prolonged time and space horizon as a lightweight, easy-integrate progress maximization method for RL. We conducted an experiment on the hill-climbing task on uneven surfaces, which requires the agent to perform spatial-temporal awarded action for a prolonged time to achieve progress maximization. Our methods show significant improvement compared to state-of-the-art methods and reached 81.93% human performance. In addition, parallel implementation with imitation learning can see agents perform similarly to humans (97.00%). Reward shaping can also be implemented to generate control theory, like 4WID, to achieve human-like performance (89.7%) but without sophisticated route planning progress maximization capability.

Acknowledgement and Appendices

Author Contributions:

Hongze Fu: Methodology, Conceptualization, Experiment, Writing-Original Draft & Editing;
Kunqiang Qing: Writing-review, Conceptualization.
All authors have read and agreed to the published version of the manuscript.
Datasets are available on request.

References

[1] Trumpp, R., Büchner, M., Valada, A., Caccamo, M. (2023). Efficient Learning of Urban Driving Policies Using Bird'View State Representations. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), pp. 4181–4186.

[2] Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23, 4909–4926.

[3] Wang, L., Liu, J., Shao, H., Wang, W., Chen, R., Liu, Y., Waslander, S. L. (2023). Efficient reinforcement learning for autonomous driving with parameterized skills and priors. arXiv preprint arXiv:2305.04412.

[4] Chae, H., Kang, C. M., Kim, B., Kim, J., Chung, C. C., Choi, J. W. (2017). Autonomous braking system via deep reinforcement learning. In Proceedings of the 2017 IEEE 20th International conference on intelligent transportation systems (ITSC), pp. 1–6.
[5] Ha, D., Schmidhuber, J. (2018). Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31.
[6] Chekroun, R., Toromanoff, M., Hornauer, S., Moutarde, F. (2023). Gri: General reinforced imitation and its application to vision-based autonomous driving. *Robotics*, 12, 127.
[7] Lu, Y., Fu, J., Tucker, G., Pan, X., Bronstein, E., Roelofs, R., Sapp, B., White, B., Faust, A., Whiteson, S. (2023). Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7553–7560.
[8] Nilaksh, A. R., Agrawal, S., Jain, A., Jagtap, P., Kolathaya, S. (2024). Barrier Functions Inspired Reward Shaping for Reinforcement Learning. arXiv preprint arXiv:2403.01410.
[9] Le Mero, L., Yi, D., Dianati, M., Mouzakitis, A. (2022). A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23, 14128–14147.
[10] Hnewa, M., Radha, H. (2020). Object detection under rainy conditions for autonomous vehicles: A review of state-of-the-art and emerging techniques. *IEEE Signal Processing Magazine*, 38, 53–67.
[11] Jehanzeb Mirza, M., Masana, M., Possegger, H., Bischof, H. (2022). An efficient domain-incremental learning approach to drive in all weather conditions. arXiv e-prints, arXiv:2204.08817.
[12] Mohammed, A. S., Amamou, A., Ayevide, F. K., Kelouwani, S., Agbossou, K., Zioui, N. (2020). The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review. *Sensors*, 20, 6532.
[13] Chiba, S., Sasaoka, H. (2021). Effectiveness of transfer learning in autonomous driving using model car. In Proceedings of the Proceedings of the 2021 13th International Conference on Machine Learning and Computing, pp. 595–601.
[14] Akhauri, S., Zheng, L. Y., Lin, M. C. (2020). Enhanced transfer learning for autonomous driving with systematic accident simulation. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5986–5993.
[15] Liu, X., Li, J., Ma, J., Sun, H., Xu, Z., Zhang, T., Yu, H. (2023). Deep transfer learning for intelligent vehicle perception: A survey. Green Energy and Intelligent Transportation, 100125.
[16] Randløv, J., Alstrøm, P. (1998). Learning to Drive a Bicycle Using Reinforcement Learning and Shaping. In Proceedings of the ICML, pp. 463–471.
[17] Wu, L. -C., Zhang, Z., Haesaert, S., Ma, Z., Sun, Z. (2023). Risk-aware reward shaping of reinforcement learning agents for autonomous driving. In Proceedings of the IECON 2023 - 49th Annual Conference of the IEEE Industrial Electronics Society, pp. 1–6.
[18] Lv, K., Pei, X., Chen, C., Xu, J. (2022). A safe and efficient lane change decision-making strategy of autonomous driving based on deep reinforcement learning. *Mathematics*, 10, 1551.
[19] Niu, J., Hu, Y., Jin, B., Han, Y., Li, X. (2020). Two-stage safe reinforcement learning for high-speed autonomous racing. In Proceedings of the 2020 IEEE international conference on Systems, Man, and Cybernetics (SMC), pp. 3934–3941.
[20] Abouelazm, A., Michel, J., Zoellner, J. M. (2024). A Review of Reward Functions for Reinforcement Learning in the context of Autonomous Driving. arXiv preprint arXiv:2405.01440.
[21] Lee, H., Jeong, J. (2023). Velocity range-based reward shaping technique for effective map-less navigation with LiDAR sensor and deep reinforcement learning. *Frontiers in Neurorobotics*, 17, 1210442.
[22] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
[23] Vamplew, P., Smith, B. J., Källström, J., Ramos, G., Rădulescu, R., Roijers, D. M., Hayes, C. F., Heintz, F., Mannion, P., Libin, P. J. (2022). Scalar reward is not enough: A response to silver, singh, precup and sutton (2021). *Autonomous Agents and Multi-Agent Systems*, 36, 41.
[24] Mariusz, B. (2016). End to end learning for self-driving cars. arXiv:1604.07316.
[25] Ho, J., Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.
[26] Peng, H., Chen, X. (2022). Active safety control of X-by-wire electric vehicles: A survey. *SAE International Journal of Vehicle Dynamics, Stability, and NVH*, 6, 115–133.
[27] Tong, Y., Li, C., Wang, G., Jing, H. (2022). Integrated path-following and fault-tolerant control for four-wheel independent-driving electric vehicles. *Automotive Innovation*, 5, 311–323.
[28] Li, R., Yu, Y., Sun, Y., Lu, Z., Tian, G. (2022). Trajectory following control for automated drifting of 4WID vehicles; 0148-7191; *SAE Technical Paper*.
[29] Li, B., Du, H., Zhang, B. (2019). Path planning for autonomous vehicle in off-road scenario. In Path Planning for Autonomous Vehicles - Ensuring Reliable Driverless Navigation and Control Maneuver; IntechOpen.
[30] Dulac-Arnold, G., Mankowitz, D., Hester, T. (2019). Challenges of real-world reinforcement learning. arXiv preprint arXiv:1904.12901.
[31] Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., Silver, D., Graepel, T. (2017). A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30.
[32] Sun, X., Zhou, M., Zhuang, Z., Yang, S., Betz, J., Mangharam, R. (2023). A benchmark comparison of imitation learning-based control policies for autonomous racing. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), pp. 1–5.