ADDRESSING THE INEFFICIENCIES IN CARLA SIMULATOR

By

Ziheng Chen

Senior Thesis in Computer Engineering

University of Illinois at Urbana-Champaign

Advisor: Professor Ravishankar K. Iyer

May 2023

Abstract

Despite the constant refinement of autonomous vehicles (AVs), their safety remains a critical issue. Many autonomous vehicle systems are developed with the help of simulations since computer simulation is a cheap and straightforward approach to generating valuable datasets. Furthermore, as a validation method, simulation-based benchmarks are widely used to determine the performance of autonomous models. In addition, AV research was also commonly conducted on computer hardware with simulation tools for its advantages. However, as we discover in this work, state-of-the-art AV simulators, such as CARLA [4], still fail to simulate the key aspects that can affect the AV's safety, especially under adverse weather conditions., which might produce suboptimally trained systems.

Simulation tools possess innate disadvantages such as incomplete physics simulation and lack of weather-specific effect in rendered camera images, which can lead to a discrepancy between simulation and the real world. Due to the discrepancy, the simulator often underrepresents driving challenges in real life, especially under complex physics, limited lighting, and emergency scenarios. Therefore, the simulation can fundamentally limit the accuracy and safety of trained algorithms because of the inaccurate generated data. Therefore, this thesis aims to provide augmentation solutions to make the CARLA Simulator more accurately represent the real world:

1. This thesis introduces friction to actor vehicles in rainy and icy scenarios.
2. This thesis will compare the effectiveness of the above modifications by comparing the behaviors of the vehicle with and without modifications and other metrics.
3. Our evaluation shows that in the worst case, compared to the existing CARLA implementation, the accident rate could rise from 49% to 98% with our modifications with adverse scenarios, leading to a wholly different yet perilous trajectory from a designed scenario.

As a result, this thesis aims to push better and more accurate open-sourced simulation tools that spark further safer AV research.

Subject Keywords: Autonomous Vehicles; performance simulations

Acknowledgments

I am deeply indebted to Professor Ravishankar Iyer and Shengkun Cui for their invaluable patience and feedback.

Contents

1. Introduction

The safety of AVs has always been at the center of the discussion about their use. Despite an enormous amount of data required and hours of training time [1], DMV accident reports pointed out the persisting insufficiency of autonomous systems in practice [2]. For example, from July 2021 to May 2022, the National Highway Traffic Safety Administration collected 273 crashes involving Tesla's Autopilot or the Full Self Driving mode, which includes five fatal crashes [3]. To consistently perform well in different adverse situations, the AV first needs to learn to behave in various scenarios, relying heavily on datasets containing such scenarios. Generating diverse datasets using computer simulation has been advantageous for its speed, cost, and safety. As a result, computer simulation is commonly used in the industry to support the development and benchmarking of AV systems.

One crucial usage for computer simulation is to sample datasets from the distribution's tail, i.e., in scenarios with unexpected faults or in adverse weather that are less common in the real world. Rain is a typical example of adverse weather, and it could severely affect tire physics, causing the automotive accident rate to increase along with rainfall intensity [5]. In the case of training the AV to perform steadily under conditions like adverse weather and system faults [6], the AV must be trained on datasets from such conditions to learn the correct perception and decision-making. For its ease of access and low overhead cost, computer simulation tools like CARLA can be easily selected for generating datasets in adverse conditions. For this purpose, the simulator's accuracy is demanded since the AV can only learn suboptimally from datasets that do not resemble the real world.

However, simulators also possess innate disadvantages, with the most critical factor being the authenticity or the level of resemblance to the real world. For example, the CARLA simulator improved significantly across generations. It can reproduce photo-realistic sceneries with visual effects incorporating real-world physics. Still, its representation of adverse weather is only in visual effects that the camera sensors can capture and do not affect physics, such as friction[4]. In other words, the amount of rain only affects the perception of AVs without affecting the friction of vehicles, which leads to an astonishing disconnection between the simulation and the real world. As a result, the simulator's output deviates from the expected behavior in the real world.

Moreover, suppose the computer-generated dataset does not consider variable friction. In that case, that dataset is not representative of the natural world, and the AV trained on such a dataset might require less braking distance in production, leading to accidents. Therefore, not incorporating physics and weather effects does not benefit the robustness of the AV as much as it potentially could.

In addition, much well-known research striving to improve the safety of an autonomous vehicle system brought much attention to the issue. For example, an effort has been made to efficiently perturb driving maneuvers of traffic participants and hardware signals to find rare cases that create safety violations [7]-[8]. However, most research, including those listed above, was built and heavily relied on the dataset generated using simulation software like the CARLA Simulator, highlighting the importance of reducing the gap between simulation and the real world.

This thesis will evaluate the effectiveness of the simulations generated by the CARLA simulator. More specifically, it introduces varying friction with weather parameters to the simulator. First, this thesis briefly covers previous efforts in friction models and explains why the statistical model is used in the implementation. This thesis then suggests an augmentation to the simulation pipeline to modify the friction of vehicles due to changes following different weather parameters, which produces a simulation that more accurately follows real-world physics. Finally, after explaining the metrics used for measurement, this thesis analyzes the data collected in various weather parameters and scenarios. It examines the magnitude of the difference the friction augmentation makes under the effect of various weather.

1.1 Motivating Example

In the ghost cut-in scenario, a selected testing scenario used later, an NPC vehicle (blue vehicle) aggressively cuts in front of the ego vehicle (red vehicle) from the top. Given the same icy weather condition, prior to any modification to the simulation pipeline, the pre-trained ego vehicle detects the incoming NPC vehicle and brakes in time to avoid a collision, as shown in Figure 1 [27]. The ego vehicle then re-accelerates after the NPC vehicle moves out of its way.

However, after modifying the pipeline, friction is reduced to 32.2% of its original value with the same icy weather parameter. As a result, when the NPC vehicle cuts in front of the AV, it completely loses control and spins out of the way, as shown in Figure 2. Although the loss of control does not result in a collision between the two vehicles, it poses a severe safety threat since it can collide with other potential participating vehicles on a populated street. This motivating example shows that the modified friction can render a completely different outcome from a designed scenario. Furthermore, it successfully demonstrated that the modification proposed in this thesis could generate a more realistic behavior under icy weather.

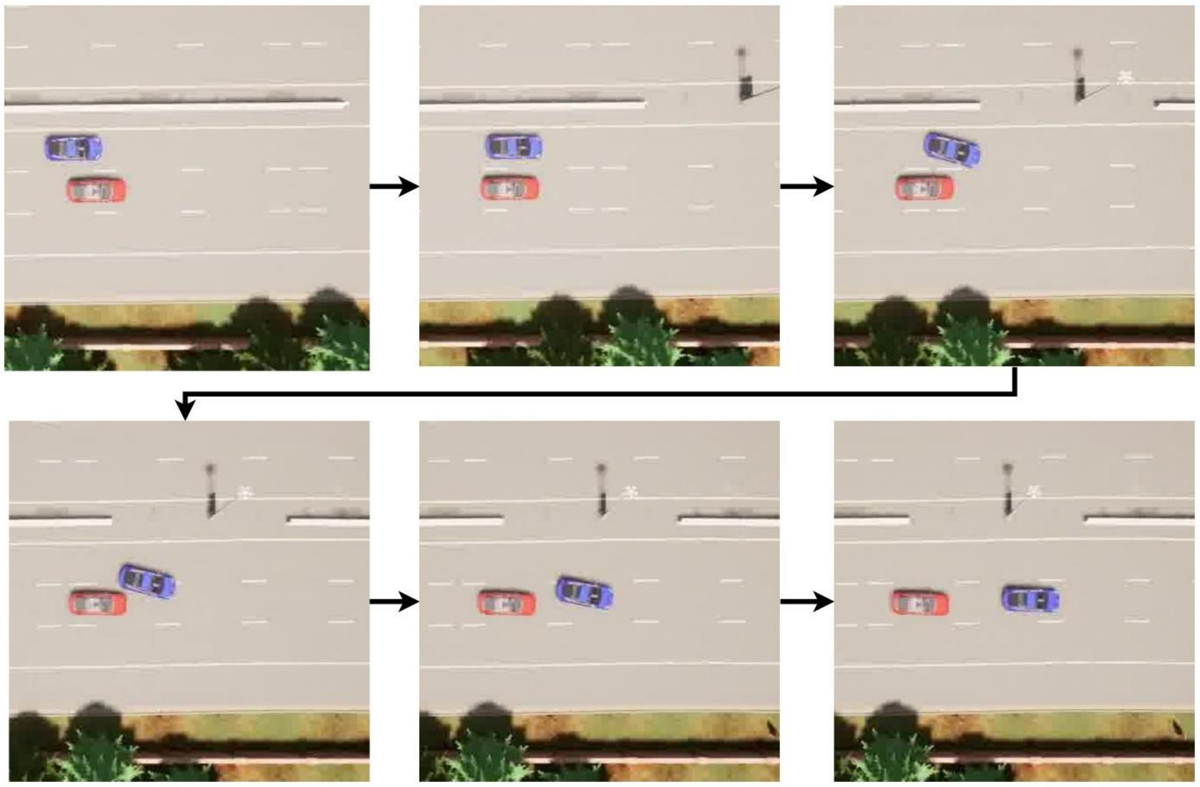


Figure 1: Vehicle trajectories in the cut-in scenario with original friction. A pre-trained network [27] controls the ego vehicle (red) to brake to react to the cutting vehicle (blue, NPC vehicle) from the top lane.

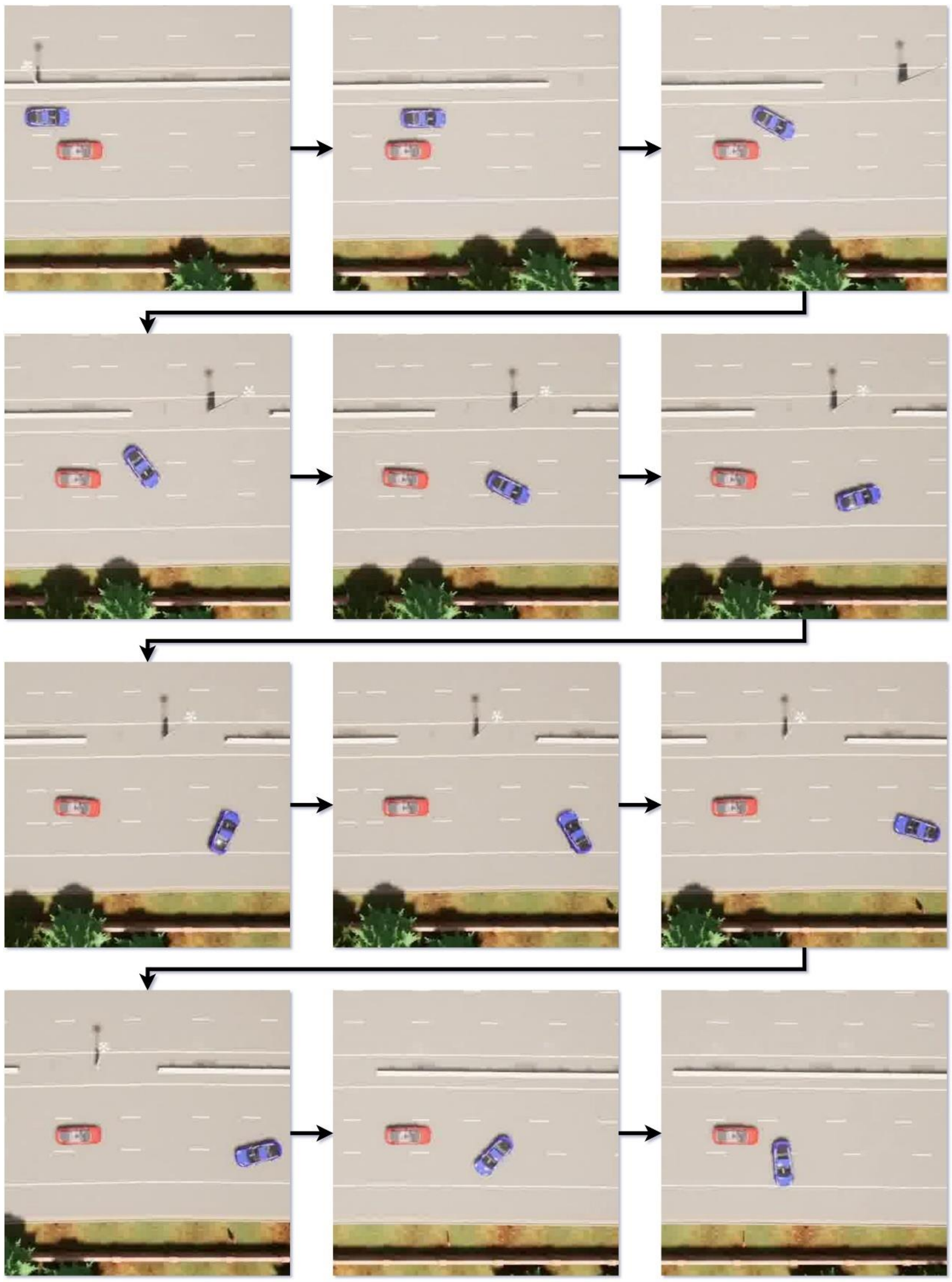


Figure 2: Vehicle trajectories in the cut-in scenario with reduced friction. The cutting vehicle (blue, NPC) loses control and slides out of the lane. A pre-trained network [27] controls the ego vehicle (red) to brake to counter unexpected behavior. Friction is reduced, leading to a completely different behavior.

2. Related Work

The friction physics between the vehicle's wheels and the pavement's surface has been thoroughly studied in various weather, and the change due to weather has been measured using braking distance [9]. Although the braking distance is undoubtedly a vital safety metric, this thesis does not find it relevant. Instead, this thesis intends to model the friction in a direct approach for its explainability when it is already natively supported, which is also directly related to the braking distance. Although the friction reduction in coefficient of friction was used in calculations explaining the braking distance, the values were often collected from historical studies. They lacked variability concerning the rainfall intensity, therefore not ideal for implementation into the simulator.

Independently, from the measured braking distance, people have also calculated the reduction of friction in the form of coefficient between the wheel and the wet and dry road under rainfall [10]. Still, the results were averages and, similar to before, also lacked mathematical models to estimate the friction of coefficient in varying wetness of the road.

Decoupled from braking distance, of the studies focusing specifically on the friction coefficient, people have also found explicit physical friction models for the tires in different weather [11]-[12]. However, such models were highly complex and depended on many more parameters. As a result, not only are the models computationally expensive, but the equations also consider many more variables. Moreover, most such parameters can only be assumed since they are not provided in the simulator, adding uncertainty. For these reasons, it is not practical to implement friction using explicit physical models.

Unlike the previous complex models, statistical methods exist to depict the relationship between friction and weather parameters, such as the thickness of ice and precipitation content on the road. The statistical friction models developed by the Finnish Meteorological Institute are based on "statistical analysis utilizing road weather station data" and marked clear correlation between measured friction and road weather parameters [13]. Meanwhile, the statistical methods also guarantee accuracy [14]. Besides interpretability, such methods are adequate for this thesis since they are lightweight and can easily predict the reduction in friction under various weather. Hence, this thesis relies on the above statistical method to model the friction with various amounts of rain and ice on the road.

From a different perspective, besides studying the physics and statistical models of friction, there are also existing solutions that try to integrate the effect of weather on the level of simulation, i.e., including more features in the simulator directly for realistic output. In addition, other simulators support the testing of AVs under varying weather conditions, such as the amount of rain [19]. However, upon closer examination, although such simulators support modifications to the base tire coefficient of friction, they do not have a built-in friction model.

Compared to the other simulators, the CARLA simulator provides a superset of the functionalities of other simulators. Therefore, the CARLA simulator is not only suited for end-to-end testing but also possesses more realistic core physics, such as the suspension and the center of mass of vehicles [20], since it is built with the Unreal Engine. Therefore, CARLA was the object of study of this thesis.

From the above studies, this thesis focuses on calculating the reduction of friction concerning the weather parameters and reducing friction by modifying the pipeline of the CARLA simulator. After running multiple scenarios under numerous weather conditions, this thesis examines the effect of such modifications using the metrics collected by a widely used benchmark [21].

3. Methodology

3.1 Reducing Friction

3.1.1 Native Friction Model

It is explained in the previous section that the core physics model is supported well in the CARLA simulator. More specifically, in the CARLA simulator, each vehicle is created from a blueprint, and the friction of each tire of a vehicle blueprint is set to a pre-defined value that holds constant and unaffected by the weather or other conditions unless changed explicitly. For example, the friction of a wheel in the simulator is a physical property of a tire. It affects a vehicle's velocity and angular velocity in the internal physics engine [22]. A tire's default friction (selected friction value) is 3.5, a "value suitable for most vehicle use cases." However, CARLA allows it to be adjusted manually; Decreasing the friction value causes a vehicle to be slippery, and increasing it causes a vehicle to have more grip, suitable for "a racing vehicle with slick tires" [23]. By modulating this friction value, we emulate the impact of reduced friction under weather parameters.

3.1.2 Modified Friction Model

Supporting adjustable tire friction does not automatically lead to more realistic simulation; The CARLA simulator can reproduce realistic and convincing visual effects under various weather, but they are only "visuals that the camera sensors can capture" and have no effect on the friction of tires [4]. This thesis modifies friction to produce more realistic simulations using the listed friction reduction formula below. This thesis considers two kinds of weather, i.e., rainy and icy, and assumes that only one weather can be present simultaneously. Therefore, friction is a function of weather parameters: precipitation deposits and wetness in rainy weather and ice thickness in icy weather. The descriptions of relevant weather parameters are listed in Table 1.

The measured friction must be calculated before being multiplied by the original tire friction to calculate the reduced friction in CARLA. From the statistical method, the measured friction on dry and clear roads is 0.8, which can be reduced to 0.1 under slippery situations. It is important to note that the measured friction does not refer to the friction coefficient but a ratio that indicates the degree to which original friction is reduced. For simplicity, this thesis assumes that friction keeps constant on a dry road, i.e., the measured friction without any impact from the weather is set to 1.0. In addition, the terminal ratio as the amount of rain and ice approaches maximum was respectively set to 0.4 and 0.15. Although different from the suggested value from the statistical method, such values were supported by previous practical studies [9], [11].

Table 1: Descriptions of relevant weather parameters

Parameter	Data Type	Range	Description
cloudiness	float	0 - 100	0 represents a clear sky, and 100 one completely covered with clouds.
precipitation	float	0 - 100	0 represents no precipitation at all, and 100 a heavy rain.
precipitation_deposits	float	0 - 100	0 represents no puddles, and 100 represents the road entirely capped with water. Puddles are created with static noise, which will always appear at the exact locations.
wind_intensity	float	0 - 100	Controls the strength of the wind with values from 0, no wind at all, to 100, a strong wind. The wind does affect rain direction and leaves from trees, so this value is restricted to avoid animation issues.
fog_density	float	0 - 100	Fog concentration or thickness. It only affects the RGB camera sensor.
wetness	float	0 - 100	Wetness intensity. It only affects the RGB camera sensor.
Ice_thickness (newly added)	float	0 - 100	From 0, no ice at all, to 100, the road is covered with 2cm of ice.

More specifically, the friction was reduced using the following models:

$$\mathrm{f}\,(\mathrm{w},\mathrm{p}) = \left[\exp\left(\frac{-0.916\ \times\ \mathrm{w}}{100}\right) \times \left(1 - \frac{\mathrm{w}}{100}\right)^3 \times 0.6 + 0.4 - 0.1 \times \frac{\mathrm{p}}{100}\right] \times \dot{f} \tag{3.1}$$

$$\mathrm{f}\,(\mathrm{i}) = \left[\exp\left(\frac{-1.89711 \times i}{100}\right) \times \left(1 - \frac{i}{100}\right)^3 \times 0.85 + 0.15\right] \times \dot{f} \tag{3.2}$$

$$\mathrm{f} = \text{friction}$$

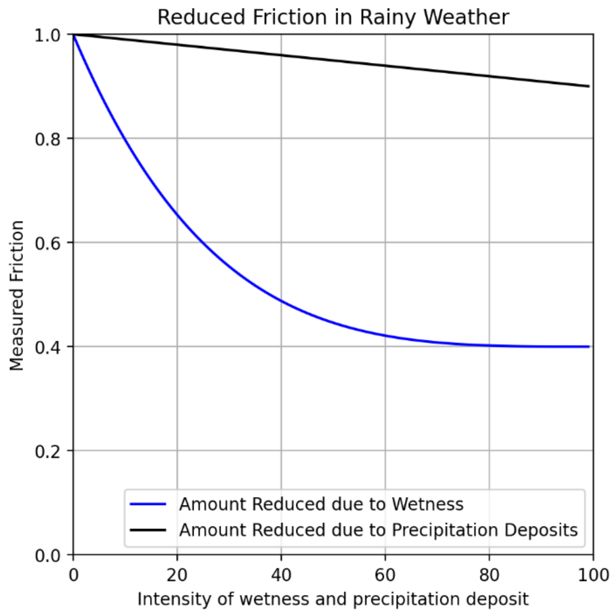
$$\mathrm{w} = \text{wetness}$$

$$\mathrm{p} = \text{precipitation deposits}$$

$$\mathrm{i} = \text{ice thickness}$$

$$\dot{f} = \text{original friction}$$

Equation 3.1 describes how tire friction is calculated depending on the original friction, wetness, and precipitation deposits. Equation 3.2 describes how the friction is calculated depending on the original tire friction and the ice thickness. Both equations are designed to have an exponential decay nature, and the scalars are used to keep the starting and terminal measured friction to desired values. In rainy weather, although the measured friction follows an exponential decay concerning the wetness, it is also affected by a slight linear offset from the precipitation deposit shown in Figure 3. The amount of puddle also indirectly contributes to the wetness of the road and can also prevent firm contact between the tire and the road, further decreasing the measured friction [26]. A linear model for the effect of puddles is assumed due to the limited description in the literature. In Figure 4, the measured friction follows a steeper exponential decay in icy weather concerning the ice thickness, almost pushing the ice thickness to have a binary effect. It intuitively makes sense as the road gets slippery when ice is present, no matter the thickness.



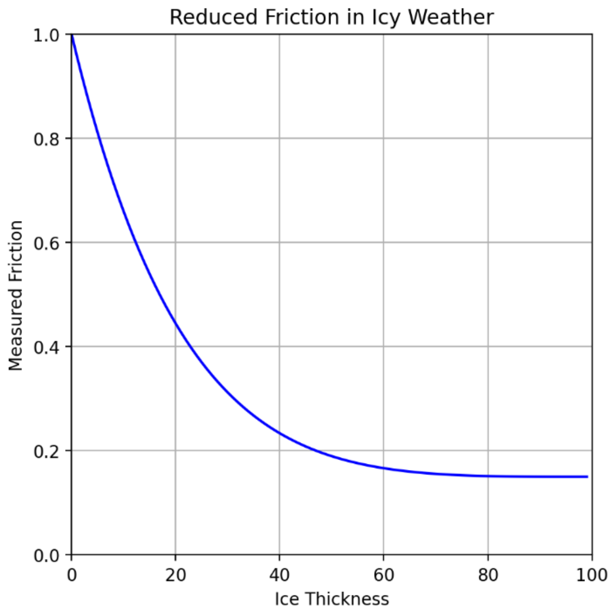


Figure 3 (Left): Measured friction as a function of wetness and precipitation deposit in rainy weather. The reduced friction can be calculated by multiplying the original tire friction by the measured friction. The x-axis corresponds to the weather parameters of wetness and precipitation deposits.

Figure 4 (Right): Measured friction as a function of ice thickness in icy weather. The reduced friction can be calculated by multiplying the original tire friction by the measured friction.

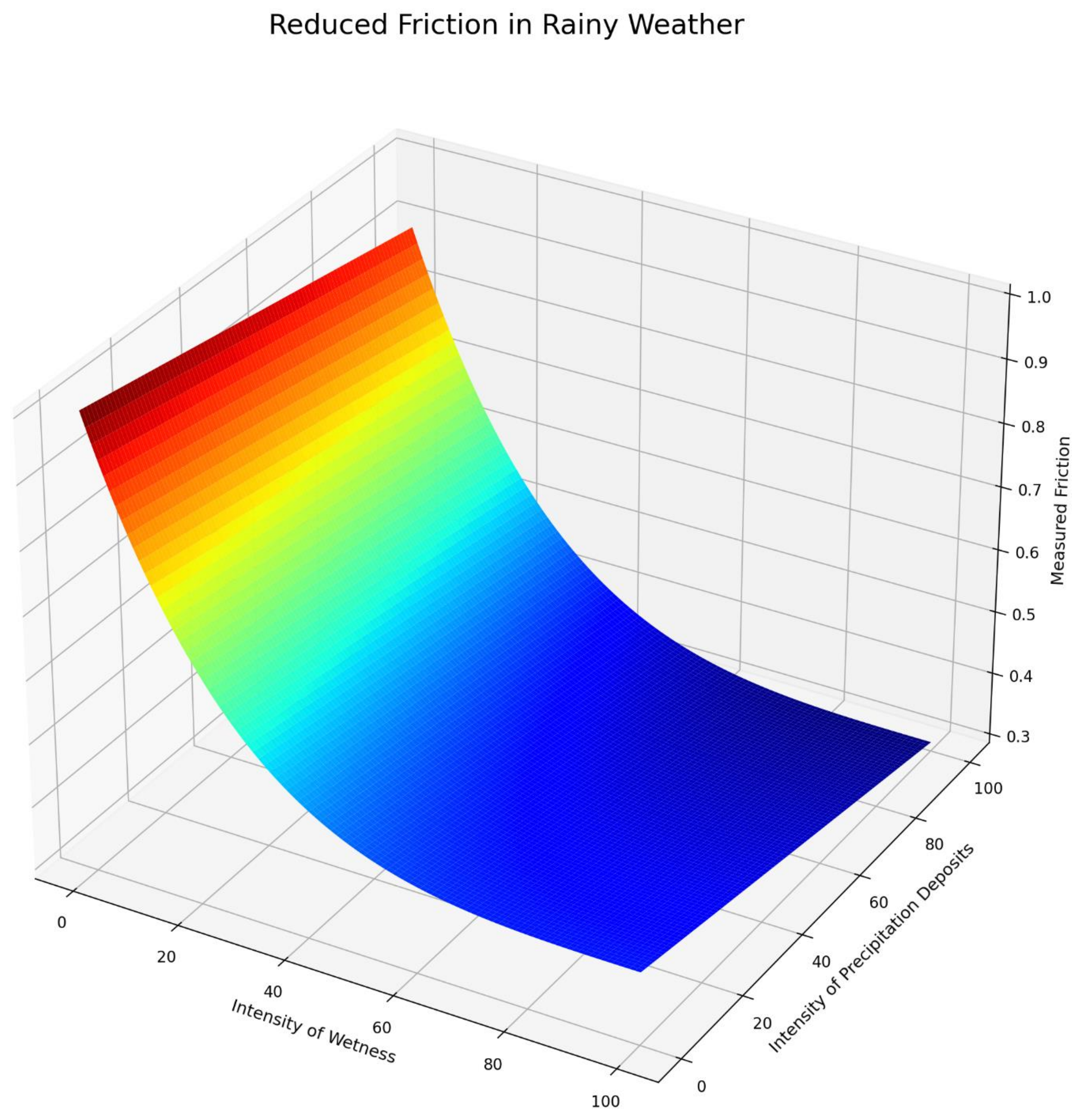


Figure 5: Measured friction in rainy weather with exponential decay from wetness and linear decay from precipitation deposits.

Figure 5 further visualizes the measured friction in rainy weather in a 3D plot. It shows a slight offset from the precipitation deposit. The z-axis represents the measured friction. By fixing the wetness, increasing the precipitation deposit decreases the measured friction by a small coefficient.

Following the model of friction above, the thesis examines the six scenarios with an increasing amount of rain, along with five scenarios with an increasing amount of ice thickness. The number in a weather parameter name indicates the intensity of the weather, e.g., rain_20 represents when the rain is at a 20% intensity. The six rainy scenarios are shown in Table 2 in the order of increasing intensity. The amount of fog and wind also increases incrementally concerning the intensity of the rain to make the scenarios as realistic as possible. It is important to note that the data from a baseline scenario without any friction modification was also collected.

Similarly, the data from 5 scenarios with an increasing amount of ice thickness is shown in Table 3. The wetness also increases concerning the thickness of the ice to mimic the amount of reflection from the road. The tables above also include calculated friction ratios corresponding to the weather parameters.

Table 2: Rainy weather parameters and resulting friction ratios

Weather Parameter Name	Cloudiness	Precipitation	Precipitation Deposits	Wetness	Fog Density	Wind Intensity	Ice Thickness	Measured Friction
rain_0	20	0	0	0	0	10	0	**1**
rain_20	20	20	20	20	5	20	0	**0.63**
rain_40	40	40	40	40	10	30	0	**0.45**
rain_60	60	60	60	60	15	40	0	**0.36**
rain_80	80	80	80	80	20	50	0	**0.32**
rain_100	100	100	100	100	30	70	0	**0.3**

Table 3: Icy weather parameters and resulting friction ratios

Weather Parameter Name	Cloudiness	Precipitation	Precipitation Deposits	Wetness	Fog Density	Wind Intensity	Ice Thickness	Measured Friction
icy_0	20	0	0	0	0	10	0	**1**
icy_10	20	0	0	10	0	10	10	**0.66**
icy_30	20	0	0	30	0	10	30	**0.31**
icy_70	20	0	0	70	0	10	70	**0.16**
icy_100	20	0	0	100	0	10	100	**0.15**

3.1.2 Friction Implementation in Pipeline

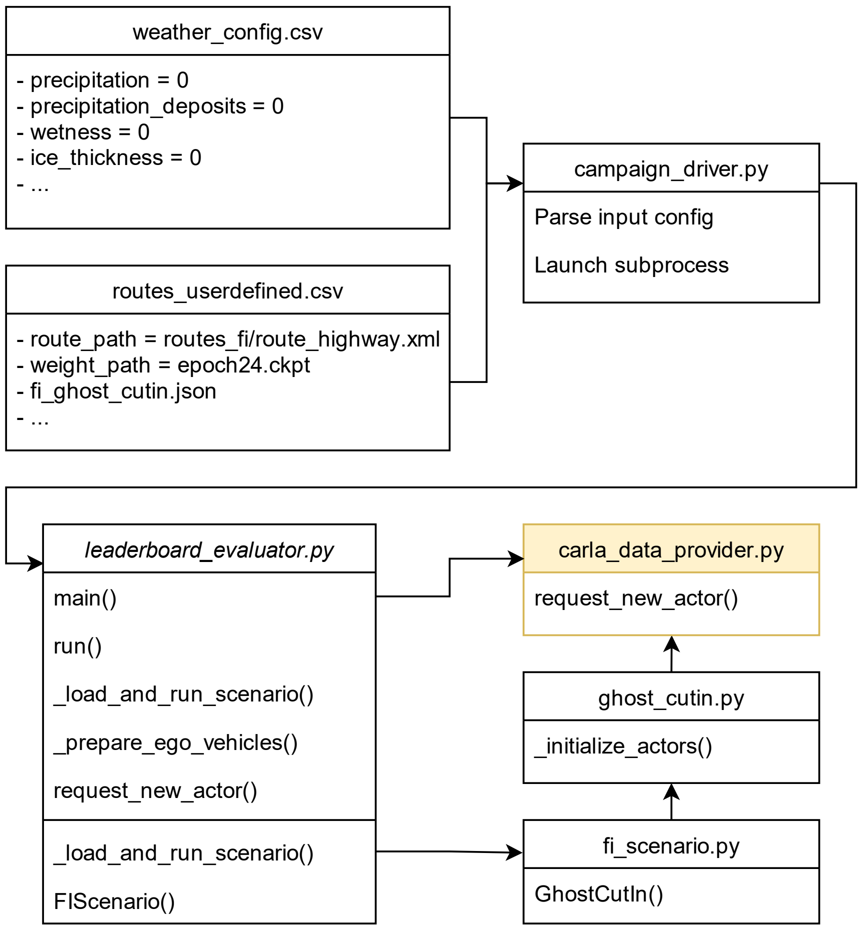
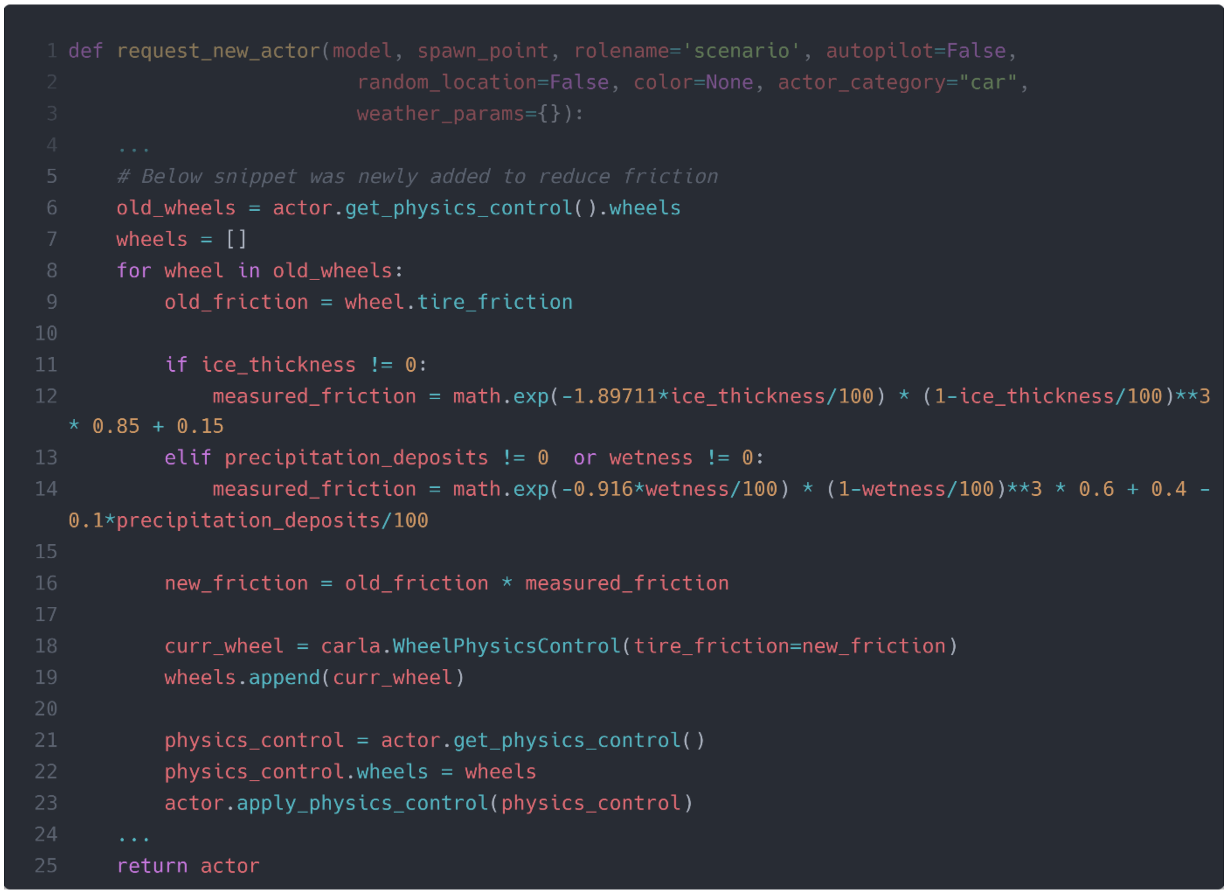


Figure 6: Overview of the pipeline of spawning vehicles for a scenario.

After solidifying a statistical friction model, this thesis suggests an implementation to modify friction for the existing pipeline, conditioning on weather parameters [21]. As illustrated by Figure 6, both weather and route configuration files are required in order to run a simulation. The pipeline is modified to pass the configuration files through the pipeline to `campaign_driver.py` and `leaderboard_evaluator.py`, in which the pipeline uses `carla_data_provider.py` to spawn the ego vehicle using the weather configuration file. Meanwhile, `leaderboard_evaluator.py` also loads the scenario, which requests new actors in carla_data_provider.py using the weather configuration. In both cases, the function `CarlaDataProvider.request_new_actor()` is used. Therefore, the weather parameters are carried through the pipeline when creating the ego vehicle and other NPC vehicles to the `request_new_actor()` function.

Figure 7 shows the modification to the `request_new_actor()` function closely. From line 6 to 23, the added script first get the original friction of the tires, then calculates the corresponding measured friction between line 13 and 14, then multiply the measured friction by the original friction of the tires to get the reduced friction before updating it in line 18 to 23 since it is assumed that rainy and icy weather cannot appear together, only of line 13 and 14 can be reached.

Another minor change to the pipeline is the addition of ice thickness to the weather parameters since the CARLA simulator does not natively support ice on the road. However, in icy weather, we assume that only the ice thickness impacts the scenarios, so using a single parameter to represent the adverseness of the icy weather suffices.



```
 1 def request_new_actor(model, spawn_point, rolename='scenario', autopilot=False,
 2                         random_location=False, color=None, actor_category="car",
 3                         weather_params={}):
 4     ...
 5     # Below snippet was newly added to reduce friction
 6     old_wheels = actor.get_physics_control().wheels
 7     wheels = []
 8     for wheel in old_wheels:
 9         old_friction = wheel.tire_friction
10
11         if ice_thickness != 0:
12             measured_friction = math.exp(-1.89711*ice_thickness/100) * (1-ice_thickness/100)**3
   * 0.85 + 0.15
13         elif precipitation_deposits != 0  or wetness != 0:
14             measured_friction = math.exp(-0.916*wetness/100) * (1-wetness/100)**3 * 0.6 + 0.4 -
   0.1*precipitation_deposits/100
15
16         new_friction = old_friction * measured_friction
17
18         curr_wheel = carla.WheelPhysicsControl(tire_friction=new_friction)
19         wheels.append(curr_wheel)
20
21         physics_control = actor.get_physics_control()
22         physics_control.wheels = wheels
23         actor.apply_physics_control(physics_control)
24     ...
25     return actor
```

Figure 7: Modified function used to spawn a vehicle, modified from [21].

3.1.3 Inconsistent Behavior of Scenarios

Due to the modification to the pipeline shown in Figure 7, the complexity of the pipeline increases, and the amount of time required to spawn both the ego vehicle and NPC vehicle increases. Therefore, the time between the two vehicles' spawning increases, producing an unwanted artifact.

The sequence of the cut-in scenario is shown in Figure 8 as a tree, which indicates how the NPC vehicle behaves. Each tree level consists of parallel tasks, and a path from top to bottom indicates a series of tasks in order. For example, "Start Driving" and "Merge Lane" are parallel to each other, and "Stand Still" is an ancestor of the task "Waiting for end position." The leaves of the tree indicate that the NPC vehicle always keeps its velocity much higher than the ego vehicle and cuts in front of the ego vehicle from behind as soon as it reaches a specific range with the ego vehicle, regardless of the state of the ego vehicle, e.g., its velocity. When the NPC vehicle cuts in, the ego vehicle would have more time to accelerate due to the increase in the difference between spawn times. As a result, the NPC vehicle would always cut into the ego vehicle prematurely, causing a collision, as shown in Figure 9. The problem does not exist in the other two tested scenarios since adding friction does not directly cause an accident in the baseline weather.

As a result, the data is only collected from the start until the premature collision in the cut-in scenario. Not only is the collected data much shorter than usual, but it also does not serve as a valuable training dataset that can be used to train a more robust AV system since the NPC vehicle always runs into the ego vehicle, and the ego vehicle will get into an accident no matter what it does; The three front-facing cameras could barely detect the collision. In addition, these collisions are purely an artifact from the modified pipeline, and it causes inconsistent behavior among baseline. For example, the "icy_0" weather parameter is considered a baseline since no ice is on the ground; therefore, the friction is not reduced. Therefore, it is expected to behave like in Figure 10, where the blue NPC vehicle cuts in at the right time. Instead, however, it produces a collision, shown in Figure 9, without fixing the scenario.

Subsequently, the artifact is addressed by offsetting the delay in spawning the NPC vehicle by decreasing the "distance" in the "Merge Lane" action in Figure 8. As a result, the NPC vehicle only cuts in when it gets closer to the AV than before, and delaying the cut-in also lets the NPC vehicle reach a higher velocity. Modifying the cut-in scenario further improves the consistency with the baseline behavior before any modification to the pipeline, as shown in Figure 10, leaving the AV more time to react.

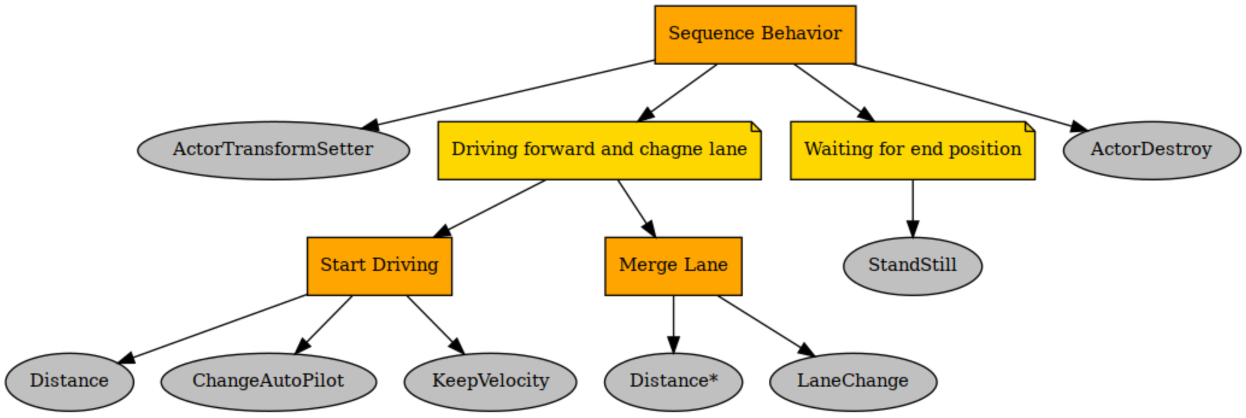


Figure 8: Visualization of pyTree structure of the cut-in scenario.

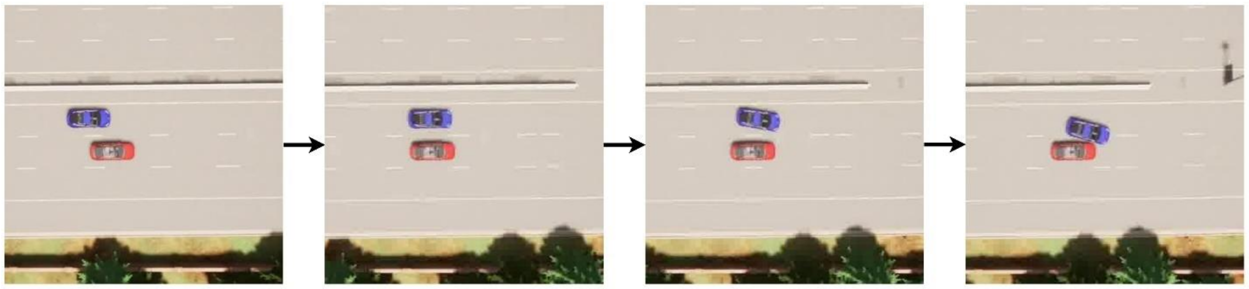


Figure 9: Unfixed cut-in scenario with the artifact. Without changing the scenario, the NPC vehicle (blue) cuts in front of the ego vehicle (red) prematurely, leaving the ego vehicle no time to react.

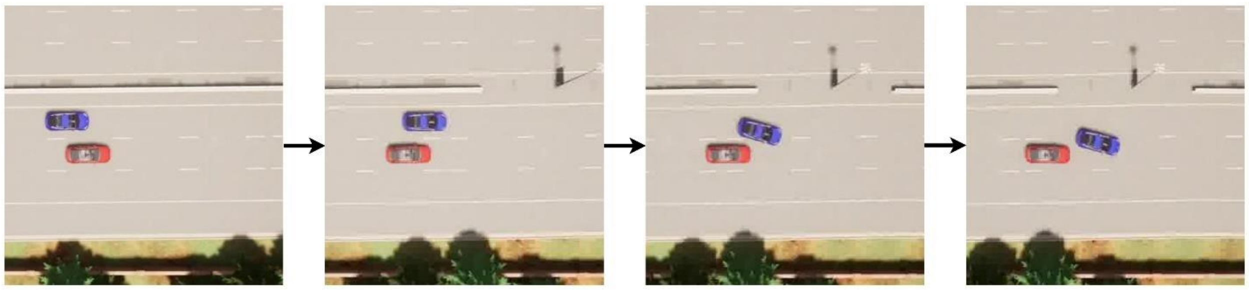


Figure 10: Fixed cut-in scenario. Baseline behavior with no reduction in friction. After delaying the cut-in timing, the actor vehicle (blue) cuts in front of the ego vehicle (red) at the right moment.

Table 4: Average Completion Rate and Score for Rainy Baseline Weather Parameter

Scenario Description	Average Completion Rate Rainy	Average Score Rate Rainy
original	100.00	100.00
unfixed	0.00	55.79
fixed	100.00	100.00

Table 5: Dynamic time warping (DTW) of all metrics for Rainy Baseline Weather Parameter

Scenario Description	x	y	v	cvip	steer	brake	throttle
original	0	0	0	0	0	0	0
unfixed	3.24	517.4	45.32	609.7	0.08	3.04	3.55
fixed	0.25	1.66	2.98	7.95	0.16	2.24	1.11

Table 4 shows the cut-in scenario's average completion rate and score for the "rainy_0" baseline weather parameter. The friction in original, unfixed, and fixed scenarios is all unmodified. The original scenario represents data collected before any pipeline modification or friction. Unfixed represents data collected after modification to the friction pipeline but not the scenario itself. The fixed scenario represents data collected after modification to the friction pipeline and the scenario.

Table 4 reflects the artifact as the unfixed scenario has a 0 completion rate due to an early accident from a premature collision. However, after the fix, the average completion rate and score stay consistent with the original scenario.

Furthermore, the DTW values of the metrics are calculated and shown in Table 5, showing the importance of fixing the scenario. DTW is an algorithm for measuring the similarity between two temporal sequences [25]. As expected, when comparing the recorded data from the original scenario against itself, all of the collected metrics are 0. Also, a significant gap exists between the unfixed and original scenario in DTW metrics values, especially in velocity and cvip. However, the gap quickly reduces after fixing the scenario, indicating much higher consistency with the original data collection.

4. Results

4.1 Scenario Description

1. **Ghost Cut-In**: Figure 11 shows this scenario's expected driving behavior. The user-controlled ego vehicle follows a lane at a constant speed. Another reckless NPC vehicle suddenly cut in front of the ego vehicle from the left while slowing down, leaving minimal time for the ego vehicle to react. The user-controlled ego vehicle should brake immediately and stop if necessary to avoid a crash.

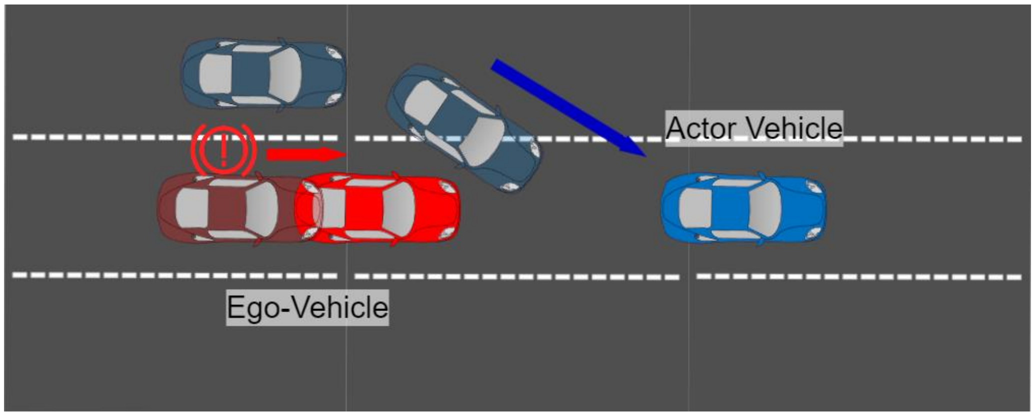


Figure 11: Ghost cut-in scenario. Red car: ego vehicle, Blue Car: NPC vehicle.

2. **Lead slowdown**: Figure 12 shows this scenario's expected driving behavior. The user-controlled ego vehicle follows a leading car driving down a given road. At some point, the leading car has to slow down and finally stop. The ego vehicle has to brake accordingly to avoid a collision.

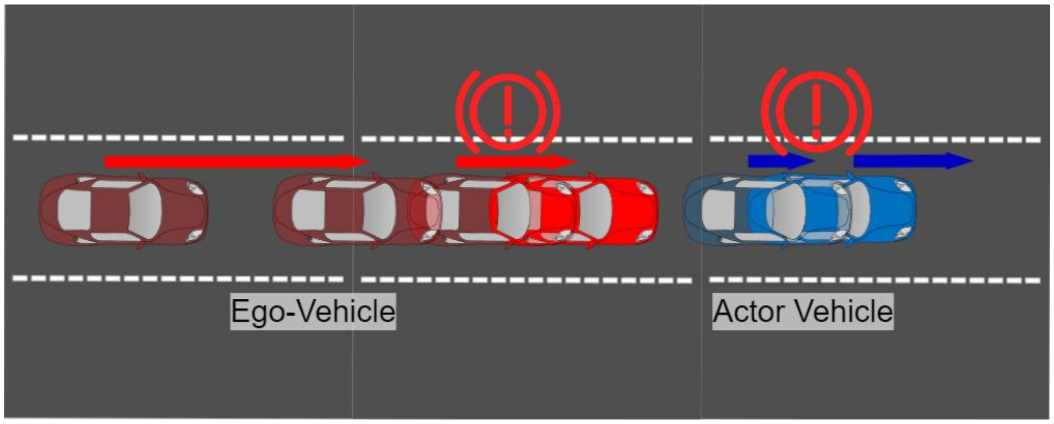


Figure 12: Lead slowdown scenario. Red car: ego vehicle, Blue Car: NPC vehicle.

3. **Front accident**: Figure 13 shows this scenario renders an uncommon driving behavior in which the user-controlled ego vehicle follows a lane at a constant speed. Two NPC vehicles suddenly crash into each other ahead of the ego vehicle while slowing down. The user-controlled ego vehicle should brake immediately and stop if necessary to avoid a crash.

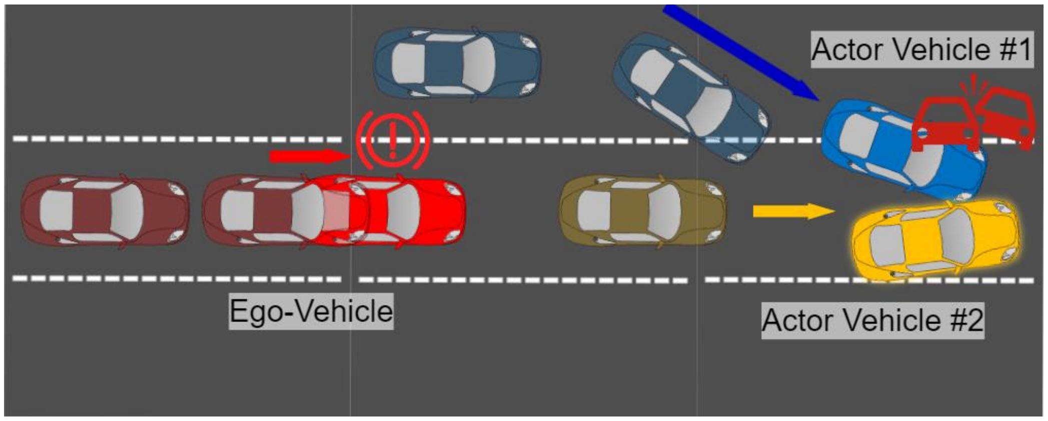


Figure 13: Front accident scenario. Red car: ego vehicle, Blue Car: NPC vehicle #1, Yellow Car: NPC vehicle #2.

4.2 Metrics Description

Table 6: Descriptions of recorded metrics

Metric	Description	Data Type
x	The horizontal location (left/right) of the vehicle	float
y	The moving direction of the vehicle	float
v	The velocity of the vehicle	float
cvip	The distance from the ego vehicle to the actor vehicle	float
steer	The moving direction of the vehicle	float
brake	The deceleration of the vehicle	boolean
throttle	The acceleration of the vehicle	float
Completion Ratio	The percentage of trials not resulting in an accident. The status of the simulation is completed.	float
Average Score	The average of safety violations (infractions). An infraction results in a score of less than 100.	float

The telemetry data of the AV is collected using metrics including x, y, and v, all measured in meters. Each CARLA actor in the form of a vehicle, including the ego vehicle and NPC vehicles, is expressed in a bounding box, and x and y indicate the center of that bounding box. They can directly express the physical location of the AV and can be used to study the trajectory of the AV.

The cvip is the measurement between the centers of the bounding boxes of the ego and the NPC vehicles, measured in meters. A small cvip value means the two vehicles are physically close and could be a leading indicator of a collision. Conversely, since the centers of the bounding boxes are measured, it is possible to have a collision with a cvip value larger than zero.

The control data of the AV is collected using metrics including steer, brake, and throttle. The steer is represented by a value from -1 to 1, with -1 being the left-most steer and 1 being the right-most steer. Similarly, for throttle, 0 indicates no throttle, and 1 indicates full throttle. Brake is a binary value where 0 means no brake, and 1 means full brake. The control data represents the microscopic changes the ego vehicle makes that cannot be easily observed from the physical locations.

Finally, the CARLA Challenge provides the completion ratio and average score [21]. They detect collision and lane control and are used to benchmark the performance of the ego vehicle.

4.3 Experimental Setup

Our experimental setup uses an AMD Ryzen 5600X CPU with 32 GB of RAM with NVIDIA RTX 3060ti GPU. All simulations are run on CARLA 0.9.10.1, an Unreal Engine-based simulator [24].

In all collected data, the ego vehicle is controlled by a pre-trained visual-based network [27]. The visual input is collected from three front-facing cameras, each facing left, center, and right. An example of visual input is shown in Figure 14.

The testing campaign includes all weather parameters introduced in Tables 2 and 3 and all scenarios introduced in section 4.1. All metrics in Table 6 are recorded in the campaign to monitor the vehicle's behavior. In addition, the results with reduced friction and original friction are collected for comparison, with the dataset from original friction as a baseline. Given the simulator's non-deterministic nature, minor deviations exist across campaigns even with the same weather parameter in the same scenario. Therefore, to ensure generality, for a weather parameter and scenario pair, the simulation runs 100 times, and the median of the 100 runs is calculated and used in the later processing to avoid outliers.

For example, among all the weather parameters and scenario pairs, the "ghost cut-in" scenario is paired with the "rain_20" weather parameter, and the friction is reduced according to equation 3.1. The procedure repeats 100 times, and all metrics, such as x, y, and cvip, are collected 100 times. Finally, the simulation process repeats itself again but with original friction.

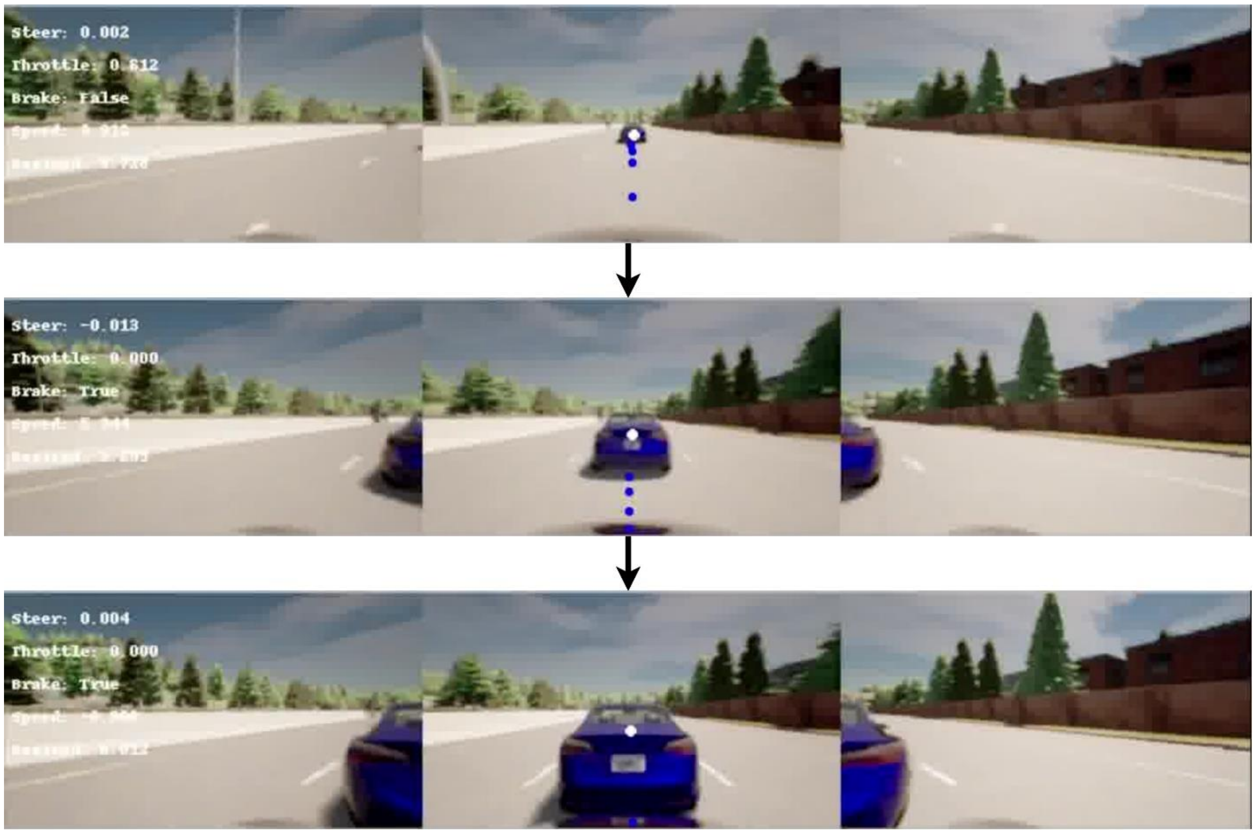


Figure 14: Selected camera input frames in lead slowdown scenario using icy_0 weather parameter.

4.4 Comparisons

This section focuses on the effect of reducing friction on AV performance while examining the metrics shown in Table 6. The outcomes are diverse, but the behavior follows a few modes, and trends can be observed.

4.4.1 Effect of Implementation with Same Weather Parameters

The effect of the implementation is first examined in the same weather parameter in each scenario, with the only difference being the reduced friction.

The first common phenomenon with reduced friction is delayed action. Due to reduced friction, vehicles take longer to accelerate and reach the state of action. For example, Figure 15 shows a cut-in scenario with the thickest ice. The orange line represents the velocity of the ego vehicle with original friction, and the blue is reduced friction. In both cases, the ego vehicle gets up to speed by timestamp 300 before braking for the cutting vehicle around timestamp 500. After the cut-in vehicle passes, the ego-vehicle accelerates again. The distance between the ego and NPC vehicles is similar, with distinct peaks and troughs in red boxes. However, with reduced friction, the trace is delayed for a constant duration, as indicated by the blue arrow. However, with reduced friction, the trace gets delayed for a constant duration, as shown in the arrow around timestamp 100.

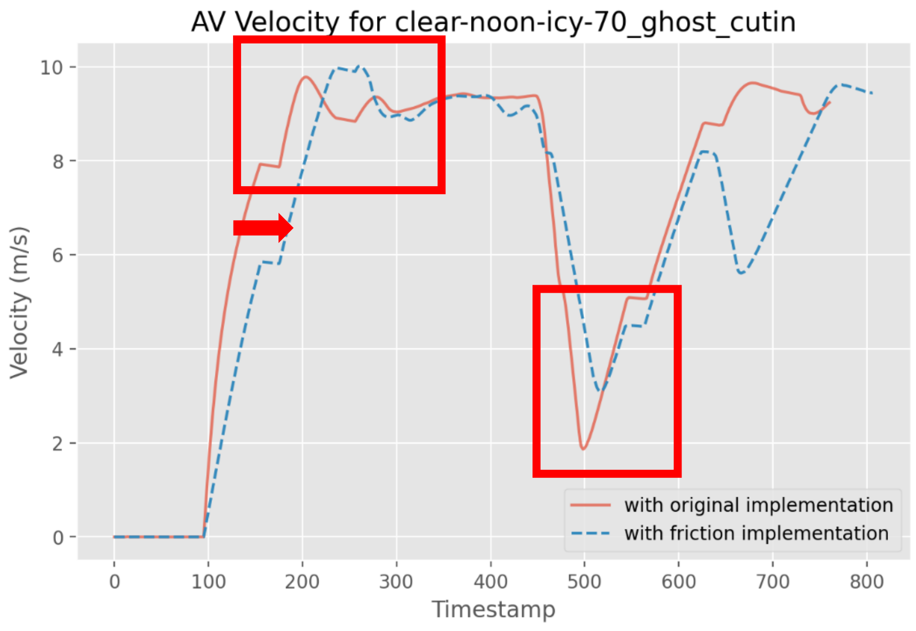


Figure 15: In the cut-in scenario, the distance between the ego and the NPC vehicle in icy weather. Reducing friction causes a delay in motion.

Besides the delay, another common phenomenon with reduced friction is that the modified friction will render an unexpected and different outcome from a designed scenario.

Figure 16 shows the distance between the ego and NPC vehicles in the lead slowdown scenario where the ice is the thickest. Starting at timestamp 200, the ego-vehicle at a higher velocity brake for the slow lead vehicle, which causes the cvip to decrease dramatically. Due to reduced friction, the first half of the plot shows a delay in time explained above. More importantly, the ego vehicle could not brake as efficiently when friction is reduced, and the distance between the two vehicles is smaller, as indicated by the red box. With original friction, the ego-vehicle stops in time and continues to finish the simulation. However, with reduced friction, we can see a similar delay in time during braking. However, instead of following the cvip in the original implementation, the ego-vehicle did not stop in time and crashed into the lead vehicle, ending the scenario prematurely, shown in the red box, causing the dashed blue line to end after the collision since the simulation is terminated.

In this case, reducing friction can result in a collision, a different outcome from a designed scenario. As a result, the ego vehicle could not brake as efficiently and eventually collided with the NPC vehicle.

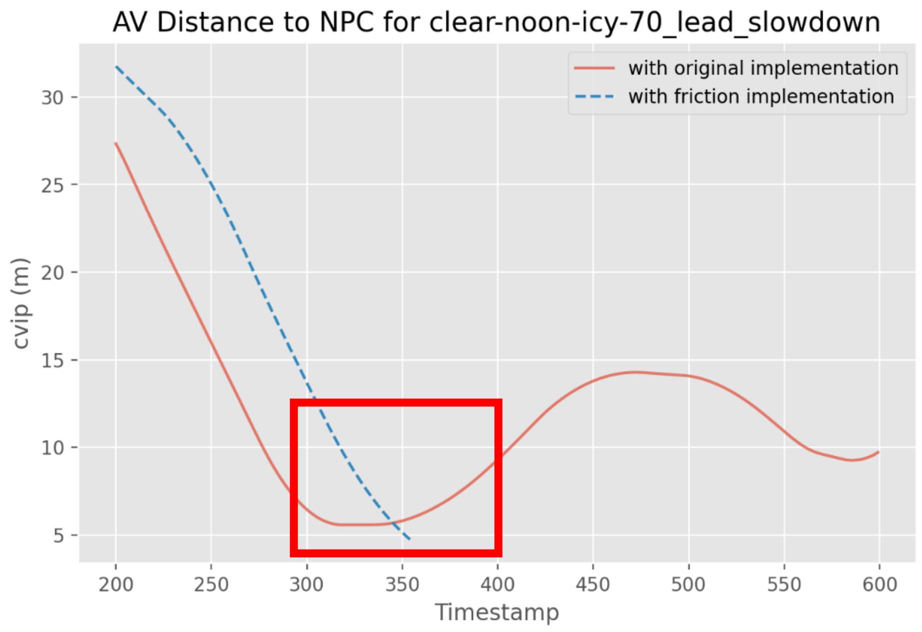


Figure 16: In the cut-in scenario, the distance between ego and the NPC vehicle in rainy weather. It results in a collision.

Besides problems in acceleration and deceleration, another common phenomenon with reduced friction is the more effortful and drastic control signal exerted by the ego vehicle. For example, shown in Figure 17, in the front accident scenario, when the ice is thickest, the ego vehicle executes steering more often and to a larger magnitude towards the later half of the trial, shown in the red box.

This phenomenon can be seen more clearly in Figure 18, which indicates the density of the steering signal issued by the ego vehicle. With original friction, the distribution is more centered at 0 since the ride is mostly smooth. With modified friction, on the other hand, the distribution of steering becomes more spread out, indicating a more drastic control signal since the reduced friction makes the ego vehicle less responsive, requiring it to adjust its direction of driving more often.

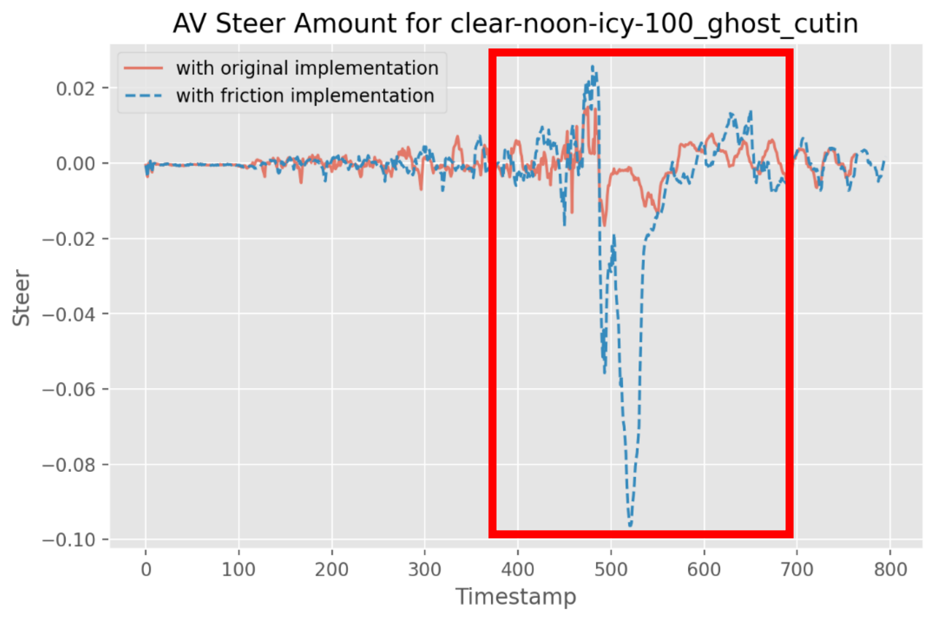


Figure 17: More drastic changes in the steering of the ego vehicle in rainy weather in the cut-in scenario.

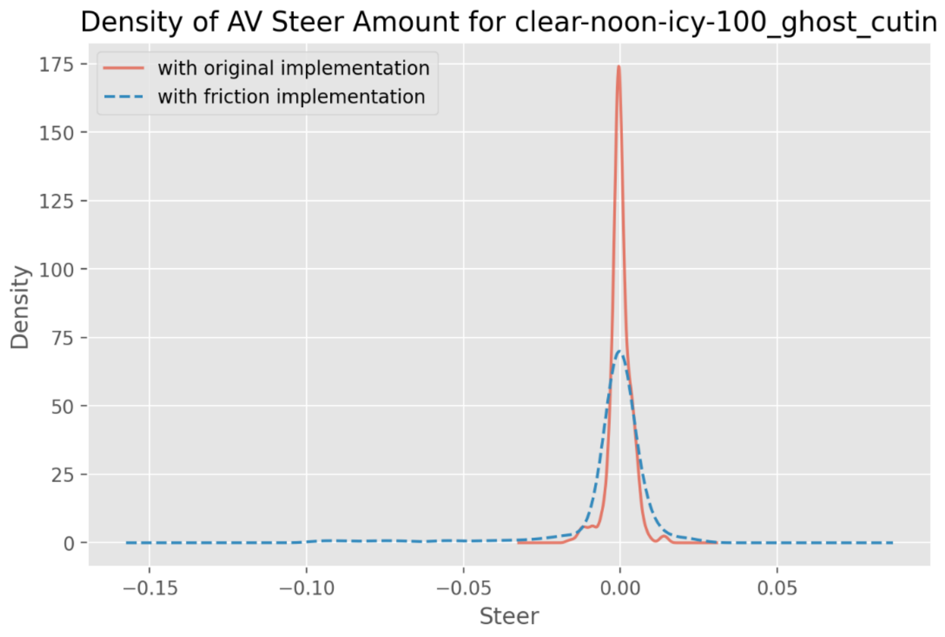


Figure 18: Distribution of steering showing more drastic changes in the steering of the ego vehicle in rainy weather in the front accident scenario.

4.4.2 Effect of Weather Parameters in New Implementation

Figure 19 shows the distance between the vehicles as the ice thickness increases in the lead slowdown scenario. As friction decreases, the vehicle's behavior gets increasingly delayed, following the trend discussed in the previous section. Increasing the ice thickness from 70 to 100 decreased the friction ratio by 0.6%, which explains the almost identical velocities of the two weather parameters.

More importantly, with increasing friction, the minimum distance between the two vehicles decreases due to ineffective braking, as shown in the red box. Further increasing the ice thickness causes the ego vehicle to collide with the NPC vehicle due to even less effective braking. Therefore, increasing the adverseness of the weather, in this case, the thickness of ice, can significantly increase the danger of the scenario. Increasing the ice thickness to the maximum can lead to accidents.

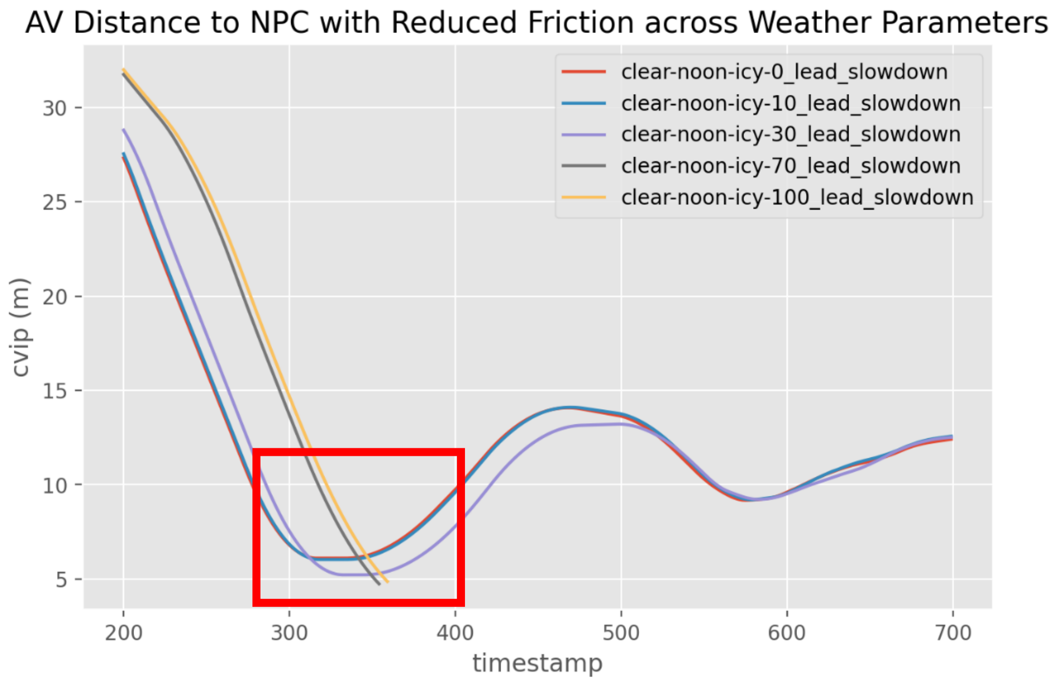


Figure 19: Distance between two vehicles in icy weather in the lead slowdown scenario.

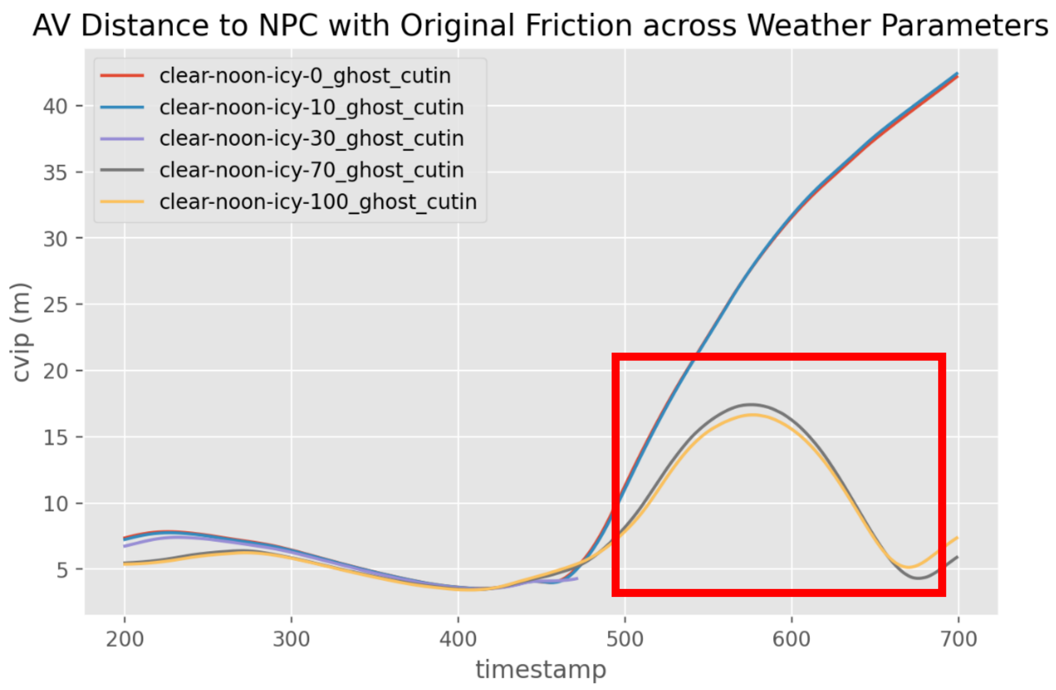


Figure 20: Distance between two vehicles in icy weather in the ghost cut-in scenario.

On the other hand, besides accidents, reducing friction can also lead to even more unexpected outcomes, as shown in Figure 20, which depicts the outcome from the motivating example.

Normally with almost unchanged friction, after timestamp 400, the ego vehicle can re-accelerate after braking and catch up to the NPC vehicle ahead, which causes the distance between the two vehicles to start to increase as fast towards the end. However, in the red box, It is common for the NPC vehicle to lose control, as shown in the motivating example ultimately. In this case, there is no NPC vehicle to catch up, and the distance between the vehicles does not increase since the ego-vehicle does not brake and gets passed by the NPC. Therefore, Figure 20 shows how reducing friction can cause the designed scenario to perform unexpectedly.

4.4.3 Quantitative Evaluation

In addition to the above demonstration, the differences caused by reducing friction can be shown numerically. All completion rates and average scores across rainy and icy weather from the three scenarios are shown in Table 7, and the negatively impacted scenarios are bolded. Generally, compared with the original friction, it is more likely for an accident to occur when friction is reduced.

More specifically, the effect is the most devastating when the friction is reduced to the maximum in icy weather parameters. In this weather, all vehicles are prone to lose control when performing a designated action, and the ego vehicle is most likely unable to brake effectively. As a result, when the ice is thickest in a lead slowdown scenario, it is guaranteed to have an accident with reduced friction, whereas there would be no accident initially. Correspondingly, the average score drops from 100% to 33.7%.

Accidents in rainy scenarios do not happen as frequently as in their icy counterpart. For example, in icy_30 and rain_100 weather parameters, the frictions are about the same. However, due to the hints from the visual input, e.g., poor visibility, the ego vehicle tends to drive conservatively, avoiding accidents, shown in the completion rate in the ghost cut-in scenario in the above weather parameters.

Besides the above metrics, since the trajectories are time series, dynamic time warping (DTW) is used to study further the difference made by friction reduction. All metrics recorded in scenarios and weather parameters with and without friction modification and the DTW of the metrics are calculated and shown in Table 8 to illustrate the effect of reducing friction on vehicle telemetry and control.

Out of all weather parameters, "rain_0" and "icy_0" are used as a baseline for comparison since these weather parameters do not change the friction; therefore, the vehicle behaviors largely remain unchanged, and the baseline entries are bolded in Table 8. For example, in icy weather in the front accident scenario, the DTW of all metrics stays below 1.0 according to the baseline, indicating a minimal change. However, as ice thickness increases, all DTW values increase due to reasons explained in the previous section, and DTW values of velocity and the cvip especially increase significantly. Therefore, it suffices to show that the added friction significantly impacted the AV's behavior, even though it is not reflected in the average completion rate and score.

Another example is in rainy weather in the cut-in scenario. The DTW of cvip should be about 0.8 according to the baseline "rain_0"; with increasing rain intensity, the DTW values also increase. When rain is the heaviest, the DTW of cvip increases to above 200, indicating a massive difference between the temporal sequence of cvip with and without friction. Therefore, the DTW values suggest that reducing friction can heavily affect the vehicle's behavior even if not easily observable.

Table 7: Average Score and Completion Rate for All Scenarios and Weather Parameters

Weather Parameter Name	Scenario Name	Average Score with Original Friction	Average Score with Reduced Friction	Average Completion Rate with Original Friction	Average Completion Rate with Reduced Friction
rain_0	Ghost Cut-In	100.00	100.00	100.00	100.00
rain_20	Ghost Cut-In	100.00	100.00	100.00	100.00
rain_40	Ghost Cut-In	100.00	100.00	100.00	100.00
rain_60	Ghost Cut-In	100.00	100.00	100.00	100.00
rain_80	Ghost Cut-In	100.00	100.00	100.00	100.00
rain_100	Ghost Cut-In	100.00	100.00	100.00	100.00
rain_0	Front Accident	100.00	100.00	100.00	100.00
rain_20	**Front Accident**	**100.00**	**41.00**	**100.00**	**76.10**
rain_40	**Front Accident**	100.00	100.00	**100.00**	**66.11**
rain_60	**Front Accident**	**100.00**	**0.00**	100.00	100.00
rain_80	**Front Accident**	**100.00**	**0.00**	100.00	100.00
rain_100	*Front Accident*	*6.00*	*34.00*	*58.90*	*87.55*
rain_0	Lead Slowdown	100.00	100.00	100.00	100.00
rain_20	Lead Slowdown	100.00	100.00	100.00	100.00
rain_40	Lead Slowdown	100.00	100.00	100.00	100.00
rain_60	Lead Slowdown	100.00	100.00	100.00	100.00
rain_80	Lead Slowdown	100.00	100.00	100.00	100.00
rain_100	Lead Slowdown	100.00	100.00	100.00	100.00
icy_0	Ghost Cut-In	100.00	100.00	100.00	100.00
icy_10	Ghost Cut-In	100.00	100.00	100.00	100.00
icy_30	**Ghost Cut-In**	**51.00**	**2.00**	**80.28**	**60.35**
icy_70	*Ghost Cut-In*	*60.00*	*100.00*	*83.91*	*100.00*
icy_100	*Ghost Cut-In*	*60.00*	*100.00*	*99.59*	*100.00*
icy_0	Front Accident	100.00	100.00	100.00	100.00
icy_10	**Front Accident**	**100.00**	**3.00**	**100.00**	**60.36**
icy_30	Front Accident	100.00	100.00	100.00	100.00
icy_70	Front Accident	100.00	100.00	100.00	100.00
icy_100	Front Accident	100.00	100.00	100.00	100.00
icy_0	Lead Slowdown	100.00	100.00	100.00	100.00
icy_10	Lead Slowdown	100.00	100.00	100.00	100.00
icy_30	Lead Slowdown	100.00	100.00	100.00	100.00
icy_70	**Lead Slowdown**	**100.00**	**0.00**	**100.00**	**33.67**
icy_100	**Lead Slowdown**	**100.00**	**0.00**	**100.00**	**33.67**

Table 8: Dynamic time warping (DTW) of all metrics with and without reduced friction

Weather Parameter	Scenario	x	y	v	cvip	steer	brake	throttle
rain_0	**Lead Slowdown**	**0.01**	**0.72**	**0.67**	**0.32**	**0.01**	**1.00**	**0.41**
rain_20	Lead Slowdown	0.06	3.66	0.98	13.75	0.02	0.50	0.89
rain_40	Lead Slowdown	0.02	1.23	1.46	2.63	0.02	1.58	1.49
rain_60	Lead Slowdown	0.03	1.40	1.97	2.68	0.05	1.50	1.27
rain_80	Lead Slowdown	0.02	1.39	3.38	3.45	0.03	1.22	1.53
rain_100	Lead Slowdown	0.12	1.51	1.80	9.14	0.03	1.50	1.48
rain_0	**Ghost Cut-In**	**0.00**	**0.26**	**0.05**	**0.87**	**0.00**	**0.00**	**0.05**
rain_20	Ghost Cut-In	0.05	1.48	4.03	9.78	0.04	2.24	0.68
rain_40	Ghost Cut-In	0.09	2.44	1.49	4.54	0.05	0.00	0.92
rain_60	Ghost Cut-In	0.11	1.64	3.07	5.10	0.03	0.87	1.11
rain_80	Ghost Cut-In	0.26	1.44	1.59	5.06	0.03	2.12	1.87
rain_100	Ghost Cut-In	0.25	1.57	1.36	203.96	0.03	1.22	1.38
rain_0	**Front Accident**	**0.03**	**0.84**	**0.39**	**0.66**	**0.02**	**0.50**	**0.35**
rain_20	Front Accident	4.50	455.82	85.38	444.55	0.33	5.00	2.92
rain_40	Front Accident	1.80	355.26	45.66	448.48	0.18	3.54	3.11
rain_60	Front Accident	0.17	2.10	16.81	237.21	0.09	3.57	3.16
rain_80	Front Accident	0.03	1.90	83.15	130.84	0.09	3.81	3.25
rain_100	Front Accident	4.29	337.84	5.09	47.96	0.05	3.28	2.21
icy_0	**Lead Slowdown**	**0.01**	**0.76**	**0.55**	**0.23**	**0.00**	**1.22**	**0.38**
icy_10	Lead Slowdown	0.02	1.11	1.66	0.84	0.02	2.00	0.80
icy_30	Lead Slowdown	0.04	1.82	4.06	9.21	0.04	1.00	1.10
icy_70	Lead Slowdown	9.58	984.81	80.28	185.55	0.10	9.33	8.73
icy_100	Lead Slowdown	10.19	967.95	77.93	199.31	0.10	9.00	9.99
icy_0	**Ghost Cut-In**	**0.02**	**1.13**	**0.34**	**1.14**	**0.01**	**0.50**	**0.20**
icy_10	Ghost Cut-In	0.04	1.42	0.69	1.06	0.03	0.00	0.63
icy_30	Ghost Cut-In	1.95	459.88	46.36	491.79	0.07	2.83	2.36
icy_70	Ghost Cut-In	0.60	1.74	10.34	286.06	0.23	3.00	2.32
icy_100	Ghost Cut-In	0.49	1.44	7.27	262.66	0.32	1.41	2.08
icy_0	**Front Accident**	**0.01**	**0.69**	**0.31**	**0.46**	**0.01**	**0.00**	**0.19**
icy_10	Front Accident	4.24	467.00	84.09	341.81	0.12	0.71	3.38
icy_30	Front Accident	0.91	2.33	6.63	87.65	0.10	7.16	2.85
icy_70	Front Accident	0.28	1.98	75.22	148.66	0.33	1.41	4.07
icy_100	Front Accident	0.17	2.08	76.52	117.54	0.26	3.61	3.00

5. Conclusion

This thesis aims to address the insufficiency of the CARLA simulator by improving its realism. It first studies the existing literature on the friction of tires in adverse weather and selects a statistical method that models friction in rainy and icy weather. The CARLA pipeline is then modified according to the model, and the results are generated with various weather parameters across rain and ice intensities.

Our quantitative analysis shows that it is determined that reducing the friction of tires, according to a mathematical model, affects the behavior of the AV. Reducing friction introduces a delay in time, causing collisions, and requires more effortful controls, leading to deviation from the original metrics. In extreme cases, altering a scenario's expected behavior can significantly affect the vehicle's behavior, causing a loss of control.

Some limitations of our work include limited scenarios and weather parameters. For example, this thesis only evaluated the performance in three designed scenarios distilled from everyday practices, but this thesis does not represent countless traffic scenarios in the real world. Similarly, although the performance of AV is measured in a range of weather parameters, the used weather parameters are only a subset of a continuous spectrum of weather from the real world. Therefore, for the above reasons, the results in this thesis are far from conclusive.

Furthermore, sometimes reducing friction could increase the safety of the AV, as shown italicized in Table 7, which should result from unclear data representation or the average completion rate that does not accurately reflect the behavior in that scenario. Therefore, another potential improvement is adding more participating vehicles to make the scenario more realistic. This way, when an NPC vehicle loses control, it collides with another. Therefore, it can more likely cause an accident with the ego vehicle, which the average completion rate will then reflect.

This thesis shows that CARLA is insufficient in rendering realistic output under adverse weather conditions, especially in rainy and icy weather. We hope the results in this thesis can motivate more realistic simulations in the future, ultimately producing a safe AV system that can be trustworthy to perform in all weather and scenarios.

6. Future Work

6.1 Advanced Friction Model

The friction model used in this thesis is general. It provides a reasonable friction estimate, but there are no physical calculations as evidence. Therefore, a future direction after this thesis is finding a more advanced and accurate friction model.

6.2 Camera Modification

Besides friction, the effect of adverse weather can also affect the sensors. The effects that weather, namely precipitation, fog, and lightning, have on the various types of sensors, which often consist of lidar, GPS, camera, and radar used in autonomous vehicles, have been systematically studied [15]. The effect has been graphically presented as technical issues in sun glare and fog [16] – [17]. Still, often the metric of effect has been measured in the change in the electromagnetic spectrum and principle of operation of the sensors. Although the results can contribute to creating more realistic scenarios [18], the studies focused on physical fundamentals. It is impractical to make rain more realistic by modifying the physical properties of sensors since it is often not exposed to the outside of the simulator. Doing so requires much more low-level knowledge of the simulator and, more importantly, relies on assumptions.

On the other hand, adverse weather can also affect the camera input, mainly when the camera input controls the AV. Therefore, improving the realism of CARLA simulation also requires augmenting the camera input. The industry has mature solutions to generate accurate camera input in rainy weather. Such solutions involve physically spraying water over the camera during data collection [18] and utilizing computational fluid dynamics for realistic renderings that obey water droplets' physics. However, the methods were company-faced and resource-heavy, sometimes requiring a weather laboratory for physical testing.

Therefore, besides modifying the friction, a future direction can be modifying the camera input in the CARLA pipeline. Since cameras often have further focal length, the raindrops can be out of focus to ensure realism. Therefore, a concept for future work can be overlaying a pre-processed rain filter with a low-pass filter, shown in Figure 21.

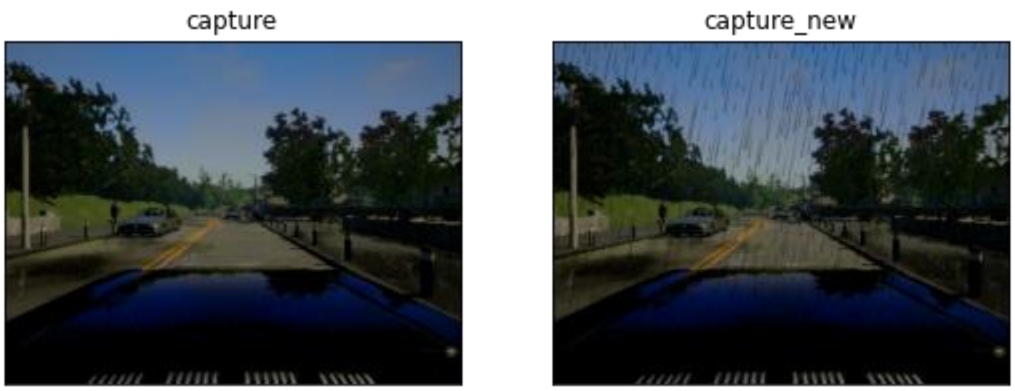


Figure 21: Before and after adding an overlay.

References

[1] Christianson, P. (2020, September 30). *Billions of Miles of Data: The Autonomous Vehicle Training Conundrum*. Cloudfactory. https://blog.cloudfactory.com/autonomous-vehicle-training-conundrum

[2] (2023, May 5). *AUTONOMOUS VEHICLE COLLISION REPORTS*. State of California Department of Motor Vehicles. https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-collision-reports/

[3] Boudette, N. E., Metz, C., & Ewing, J. (2022, June 15). *Tesla Autopilot and Other Driver-Assist Systems Linked to Hundreds of Crashes*. The New York Times. https://www.nytimes.com/2022/06/15/business/self-driving-car-nhtsa-crash-data.html

[4] "Weather," https://carla.readthedocs.io/en/latest/core_world/#weather

[5] Pang, Ming-Bao & Ren, Bo-Ning. (2017). Effects of rainy weather on traffic accidents of a freeway using cellular automata model. Chinese Physics B. 26. 108901. 10.1088/1674-1056/26/10/108901.

[6] McFarland, M. (2023, January 17). *Tesla-induced pileup involved driver-assist tech, government data reveals*. CNN Business. https://www.cnn.com/2023/01/17/business/tesla-8-car-crash-autopilot/index.html

[7] G. Li *et al*., "AV-FUZZER: Finding Safety Violations in Autonomous Driving Systems," *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, Coimbra, Portugal, 2020, pp. 25-36, doi: 10.1109/ISSRE5003.2020.00012.

[8] Saurabh Jha, Subho Banerjee, Timothy Tsai, Siva K. S. Hari, Michael B. Sullivan, Zbigniew T. Kalbarczyk, Stephen W. Keckler, Ravishankar K. Iyer. ML-Based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection. IEEE Xplore Digital Library, 2019 DOI: 10.1109/DSN.2019.00025

[9] Kordani, Ali & Rahmani, Omid & Nasiri, Amir & Boroomandrad, Sid. (2018). Effect of Adverse Weather Conditions on Vehicle Braking Distance of Highways. Civil Engineering Journal. 4. 46. 10.28991/cej-030967.

[10] Hartman, J., Marzbani, H., Alam, F., Atapourfard, M., & Nakhaie Jazar, G. (2018). Friction coefficient of pneumatic tires and bitumen roads. In *Nonlinear Approaches in Engineering Applications* (pp. 209–276). Springer. https://doi.org/10.1007/978-3-319-69480-1_8

[11] Liu, X., Cao, Q., Wang, H., Chen, J., & Huang, X. (2019). Evaluation of Vehicle Braking Performance on Wet Pavement Surface using an Integrated Tire-Vehicle Modeling Approach. *Transportation Research Record, 2673*(3), 295–307. https://doi.org/10.1177/0361198119832886

[12] X. Claeys, Jingang Yi, L. Alvarez, R. Horowitz, C. C. de Wit and L. Richard, "Tire friction modeling under wet road conditions," *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, Arlington, VA, USA, 2001, pp. 1794-1799 vol.3, doi: 10.1109/ACC.2001.945994.

[13] "Evaluation of FMI's new forecast model of road surface friction," https://sirwec.org/wp-content/uploads/2022/04/Quebec-D-21.pdf

[14] Hippi, Marjo & Juga, Ilkka & Nurmi, Pertti. (2010). A statistical forecast model for road surface friction.

[15] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo and M. A. Kaafar, "The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car," in *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 103-111, June 2019, doi: 10.1109/MVT.2019.2892497.

[16] Zhang, Y., Carballo, A., Yang, H., and Takeda, K., "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," <i>ISPRS Journal of Photogrammetry and Remote Sensing</i>, vol. 196, pp. 146–177, 2023. doi:10.1016/j.isprsjprs.2022.12.021.

[17] Yoneda, Keisuke & Suganuma, Naoki & Yanase, Ryo & Aldibaja, Mohammad. (2019). Automated driving recognition technologies for adverse weather conditions. IATSS Research. 43. 10.1016/j.iatssr.2019.11.005.

[18] Vargas, Jorge & Alsweiss, Suleiman & Toker, Onur & Razdan, Rahul & Santos, Joshua. (2021). An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. Sensors (Basel, Switzerland). 21. 10.3390/s21165397

[19] A. Best, S. Narang, L. Pasqualin, D. Barber and D. Manocha, "AutonoVi-Sim: Autonomous Vehicle Simulation Platform with Weather, Sensing, and Traffic Control," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, USA, 2018, pp. 1161-11618, doi: 10.1109/CVPRW.2018.00152.

[20] Kaur, Prabhjot & Taghavi, Samira & Tian, Zhaofeng & Shi, Weisong. (2021). A Survey on Simulators for Testing Self-Driving Cars.

[21] "2020 CARLA Challenge", https://github.com/bradyz/2020_CARLA_challenge

[22] "Instance Variables," https://carla.readthedocs.io/en/latest/python_api/#instance-variables_84

[23] "Tire configuration," https://carla.readthedocs.io/en/latest/tuto_content_authoring_vehicles/

[24] "CARLA Documentation", https://carla.readthedocs.io/en/0.9.10/

[25] Zhang, J. (2020, February 1). *Dynamic Time Warping*. Towardsdatascience. https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd

[26] Weiss, P. (2004, November 10). *Piddly Puddle Peril: Little water pools foil road friction*. ScienceNews. https://www.sciencenews.org/article/piddly-puddle-peril-little-water-pools-foil-road-friction

[27] Dian Chen, Brady Zhou, Vladlen Koltun, & Philipp Krähenbühl. (2019). Learning by Cheating.