

Investigating the use of unseen simulators for evaluating self-driving car models

Harry Collins, s1664050

Supervised by:

Seán Mullery, Sligo Institute of Technology

Abstract

This work was focused on the use of simulators for training self-driving cars, a topic currently featured prominently in machine learning research and legislative discussions. The work aimed to evaluate the possibility of using unseen simulators to improve self-driving car models by comparing the performance of different simulator-trained models on unseen simulators. The project was unable to find previous work on the use of multiple simulators to train self-driving car models. The project successfully installed unsupported simulation software, CARLA and DeepDrive to a cloud computing platform leveraging cost-effective and scalable computing power. This opens up a potentially new area of connected and autonomous vehicle research in the use of unseen simulators to evaluate models' ability to generalise to reality which could increase real-world performance and decrease development costs.

Introduction

Training autonomous vehicles can be a very expensive and challenging endeavour. The first challenge is collecting the thousands or millions of hours of driving data needed to train an autonomous driver. The second is verifying the safety of the trained autonomous drivers. The high cost and small quantity of real-life data has made its use alone for training infeasible. The lower costs and increased model performance from larger quantities of data have necessitated the heavy use of simulated data for both the training and testing of autonomous driver models. However there remains questions concerning the ability of models to transfer from simulated to real environments.

This work attempts a preliminary analysis of how self-driving machine learning models cope with the differences between different simulators. The goal is to make inferences on the ability of the tested models to handle the differences between simulators and reality.

The basis of the project is the hypothesis that if a simulator-trained model can generalise well to an unseen simulator, it can generalise well to reality. Similarly, this proposed cross simulator testing could help to better understand how simulator design choices affect model creation and real-life performance.

Currently the use of multiple different simulating softwares in the design and creation of self-driving car models has been unexplored in research.

Literature Review

Training autonomous vehicles can be a very expensive and challenge endeavour. Current challenges facing autonomous driving companies and legislators today include the large amount of training data required and how to verify the safety of an autonomous driver. The most obvious way to assess safety is to test drive autonomous vehicles in real traffic, observe their performance and statistically compare the results to humans [3]. However as shown in [3], over 5 billion driven miles would be necessary in order to verify that an autonomous car was statistically 20% safer than a human; To have only an 80% chance of correctly verifying that AI driver is safer (known in statistics as power) requires over 11 billion miles; 100 vehicles driving 24 hours a day averaging 25 mph would take over 500 years to log that much distance[3].

Simulators are far cheaper and convenient for both generating training data and verifying the performance of AI driving models. Servers can drive millions of miles a day at a fraction of the per distance cost. Through simulators it is possible to design and generate emergency scenarios to verify the system generates the correct responses [6]; These types of scenarios are known as edge cases and are infrequent in real driving data.

Physical testing of self-driving car models is complex and limited to the end of the development cycle. This makes prototype design, implementation, intensive testing in the simulation circuit the most effective way to verify and validate the design idea[5].

The ability to partially verify the performance of models and autonomous systems in the simulation phases of design can reduce risks and costs. This is critical to feasibility of producing safe, and affordable autonomous vehicles.

In designing simulators, one must account for all aspects of the physical system in order to simplify the transition from virtual to real-world scenarios[5]. Simulators are only a simplification or representation of reality. Self-driving car models have to be able to generalise in order to handle the differences between the simulator and reality.

Currently there is no research into how well simulator trained models can generalise to handle the differences between simulators. The simulator’s physics engines use different approximations and equations to represent the physical world. Similarly, simulator’s graphical choices produce object representations with differing levels of detail and realism. In the case of perception based simulators (models trained using camera technology for example), the models’ object classification methods need to use features that are also present in their real-world counterparts.

In the authors opinion, evaluating how simulator-trained models generalise to unseen simulators can provide insights into the problem of training models optimised to bridge this simulator-reality gap.

Method

Two self-driving simulation environments, CARLA [2] and DeepDrive [4] were set up on Google’s cloud computing platform. There were two major challenges associated with this step. Firstly, the simulators are designed to be run locally with little or no support for running on cloud platforms. Secondly, nvidia tesla GPU’s which make up most of the GPUs in computing servers do not have display ports. These challenges meant GUI’s were mostly unavailable and therefore several workarounds had to be used to verify the results. Steps to replicate the installations can be found on the project’s github¹[1].

High performing Conditional Imitation Learning tensorflow models (CIL) for both DeepDrive and CARLA CIL model were then benchmarked on their respective simulators.

Then attempts were made to migrate the trained agents from one simulator to another. Modification

to the tensorflow agents were made with reference to the tutorials and series on tensorflow and autonomous vehicles from the website pythonprogramming². This involved separating the tensorflow agents from the custom learning environments that interface the tensorflow agents with their respective simulators. This was challenging as the code used to write the agent assumed the presence of the simulator specific learning environments which were implemented differently for both simulators.

While it was possible to normalise the sensory inputs to the agent. Replicating the environmental structure passed to the agent which governs how the agent transforms the output of its neural network to simulator controls is still ongoing.

Results

Simulators Running in the Cloud

While the simulators CARLA [2] and DeepDrive [4] were not built to run on a cloud computing platform, it was possible to run most of the simulator’s functionalities with the methods described in the project’s README³ [1]

Examples of the simulators running in the cloud are shown in figures 1-3.

The result was the ability to leverage powerful computing hardware such as Nvidia Tesla p100s to run the simulation and machine learning software. This was necessary as the CARLA CIL model alone required 5.5 GB of GPU memory.



Figure 1: Carla simulator agent in rain with pedestrians and traffic

¹<https://github.com/hcollins345/PP-evaluating-ai-driving-simulators>

²<https://pythonprogramming.net/tensorflow-introduction-machine-learning-tutorial/>

³<https://github.com/hcollins345/PP-evaluating-ai-driving-simulators>



Figure 2: Agent stopped at traffic lights

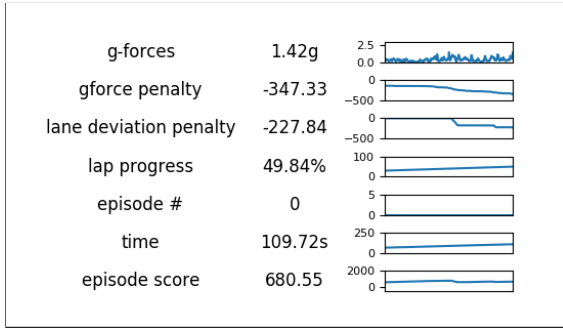


Figure 3: Monitoring status of agent in DeepDrive simulator

Benchmarking

The benchmarking results for the DeepDrive simulator can be seen in figure 4 below. 34 runs were conducted with the final score for each run consisting of the distance of the route completed minus penalties for excessive lane deviation and g-forces.

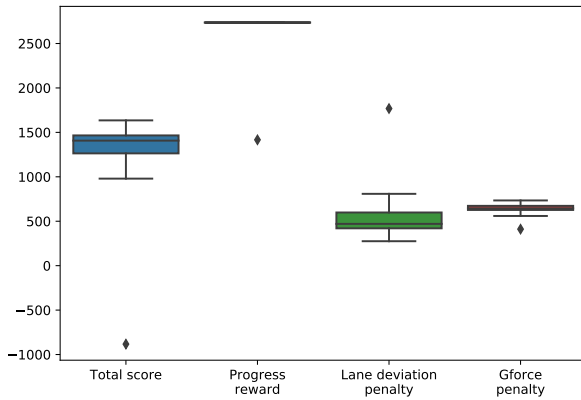


Figure 4: Benchmarking results for Deepdrive agent in DeepDrive simulator

23 runs were conducted for the CARLA benchmarking. The test suite consisted of 4 sub-tests, as examples, test 1 consists of driving straight forward while test 2 involves a right hand turn at a junction.

To give an indication of the variation, the driven kilometres required to complete each task for each trial were plotted in figure 5.

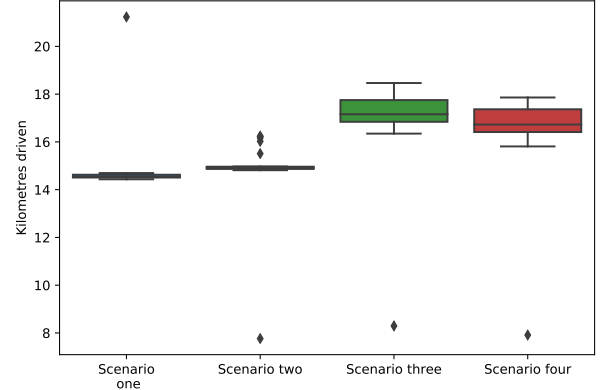


Figure 5: Benchmark results showing kilometres driven to complete each task for Carla agent in Carla simulator

Discussion

Variable vs Fixed time step

The DeepDrive simulator runs only with variable-time step. This means that as computing loads change, the time between frames, the number of frames per second and the number of control inputs from the driving agent per second vary throughout the run. This is to keep the simulator running in sync with real world time. ie. if an in-game object moves at 3 m/s, it will move 3 metres every real-life second.

The alternative is to use fixed-time steps, where the simulator time is desynced from the real world will take time to fully render every frame and wait for every input from the driving agent. This has the benefit of making results fully reproducible.

For the benchmarks, CARLA has both options but variable-time was chosen to help align the results with the DeepDrive simulator.

Comparing benchmarking results.

The results for the benchmarking tests in figures 4 and 5 obviously differ. This is due to the different configuration of the test suites used in the two simulators.

For example the one large test in DeepDrive suite vs the four smaller tests in CARLA. As another example, in the DeepDrive tests, the metric for lane deviation is an analog value that continually increases proportional to the amount the agent moves out of its lane. In contrast, the CARLA suite merely records whether or not the agent move out of its lane by a specified cut-off value as a binary value.

Both simulators are built on the Unreal Engine and have similar metric available. However the test suites for both simulators would need to be rewritten in order to adequately compare. An alternative option would be to install a third simulator used solely for

evaluating the performance of models trained in the other two simulators.

Conclusion

The literature review failed to find any work exploring the use multiple simulators in the training of self-driving cars.

The use of unseen simulators to aid in creating more robust self-driving car models appears to be novel area of research. The ability to migrate simulators to cloud computing platforms makes this research approach accessible to researchers without high-end, large memory GPU desktops.

Future work, finalising the model transfers and equalising the evaluation suite for the models, is required to determine the merits of this proposed area of research.

-
- [1] Collins, H. 2019. hcollins345/PP-evaluating-ai-driving-simulators: Personal Pursuit Submission. (Sep. 2019). DOI:<https://doi.org/10.5281/ZENODO.3453314>.
 - [2] Dosovitskiy, A. et al. 2017. CARLA: An Open Urban Driving Simulator. (Nov. 2017).
 - [3] Kalra, N. and Paddock, S.M. 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*. 94, (Dec. 2016), 182–193. DOI:<https://doi.org/10.1016/J.TRA.2016.09.010>.
 - [4] Quiter, C. and Ernst, M. 2018. Deepdrive/deepdrive: 2.0. (Mar. 2018). DOI:<https://doi.org/10.5281/ZENODO.1248998>.
 - [5] Rosique, F. et al. 2019. A systematic review of perception system and simulators for autonomous vehicles research. MDPI AG.
 - [6] Wotawa, F. et al. 2018. Quality assurance methodologies for automated driving. *e & i Elektrotechnik und Informationstechnik*. 135, 4-5 (Aug. 2018), 322–327. DOI:<https://doi.org/10.1007/s00502-018-0630-7>.