HAL
open science

# Beyond Traditional Simulators: Adopting Videogames for Autonomous Vehicle Testing

Abderrahmane Maaradji, Naila Bouchemal, Idir Smadhi, Ahmed Bouhraoua, Amayas Ghanine

## To cite this version:

Abderrahmane Maaradji, Naila Bouchemal, Idir Smadhi, Ahmed Bouhraoua, Amayas Ghanine. Beyond Traditional Simulators: Adopting Videogames for Autonomous Vehicle Testing. IEEE DASC 2023 : The 21st IEEE International Conference on Dependable, Autonomic & Secure Computing, Nov 2023, Abu Dhabi, United Arab Emirates. pp.0570-0575, 10.1109/DASC/PiCom/CBDCom/Cy59711.2023.10361295 . hal-04242049

**HAL Id: hal-04242049**

**https://hal.science/hal-04242049**

Submitted on 14 Oct 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Beyond Traditional Simulators: Adopting Videogames for Autonomous Vehicle Testing

Abderrahmane Maaradji, Naila Bouchemal, Idir Smadhi, Ahmed Bouhraoua, and Amayas Ghanine
*LyRIDS, ECE Engineering School*
Paris, France
firstname.lastname@ece.fr

***Abstract*—This article presents an ongoing research study investigating the potential of using Grand Theft Auto V (GTAV) as a simulation environment for testing edge cases in self-driving vehicles. Our hypothesis is that GTAV can offer a more accessible and cost-effective solution for testing edge cases compared to established simulation environments such as CARLA. In this work-in-progress study, we aim to assess the feasibility, scalability, and effectiveness of utilizing GTAV by simulating complex driving scenarios. We selected Openpilot, an open-source Advanced Driver-Assistance System (ADAS), to test these scenarios. Compared to CARLA, GTAV proved to be a better choice for testing realistic urban environments, dynamic weather, complex intersections, and detailed nighttime settings. Moreover, these simulations offered valuable insights into how Openpilot reacts to various driving conditions and how it could be enhanced to handle edge cases more efficiently.**

***Index Terms*—Self-driving vehicles, Openpilot, Edge-case scenario, GTAV, Simulation environment, CARLA**

## I. Introduction

Autonomous vehicles are often expected to revolutionize our commute by offering increased convenience, safety, and efficiency. At the core of these innovations are complex self-driving algorithms, the product of countless hours of research and development, which govern vehicle behavior. One such algorithm that stands out due to its maturity and distinct approach is Openpilot [1].

Openpilot, developed by Comma.ai, is an open-source, end-to-end self-driving system. Unlike modular systems that decompose the driving task into discrete components, Openpilot's deep neural network takes raw input from cameras and sensors and outputs steering, throttle, and brake actions directly. This end-to-end approach provides a unique perspective in the autonomous driving landscape.

The challenge, however, lies in adequately testing these algorithms to ensure they can handle the different situations that arise on the roads. This challenge escalates in the context of SAE International's classification of autonomous vehicles. This standard ranges from Level 0, where the car is entirely controlled by a human, to Level 5, which represents complete autonomy in all situations [2]. Currently, most "self-driving" cars available are at Level 2, including Openpilot.

Testing the higher levels of autonomy in vehicles is challenging, as they may encounter various real-world scenarios. One approach is to use simulation environments, such as CARLA [3], to create controlled conditions that allow for extensive testing without the potential hazards and costs associated with real-world testing.

However, these environments have their limitations. Despite the impressive fidelity of simulators like CARLA, they still struggle to replicate the complexity and unpredictability of real-world driving conditions, particularly the so-called edge cases that fall outside of normal operating conditions. It's these edge cases that often prove to be the most critical in testing autonomous driving algorithms [4]. In an attempt to address these challenges, this research explores the potential of using commercially available videogames as simulation environments. Specifically, we turn our attention to GTAV, a game renowned for its detailed and dynamic urban environment that mirrors the complexities of a real city.

Our hypothesis is that the richness of the GTAV environment might provide a more robust platform for testing edge-case scenarios than traditional simulators. Our research specifically focuses on testing Openpilot, comparing its performance in GTAV against the established CARLA simulator. In the following sections, we will present our methodology, the current preliminary results, and discuss future work.

## II. Related work

The vision of Autonomous Vehicles (AVs) is prompting a revolutionary change in the realm of transportation. However, this vision introduces a multitude of research challenges spanning technology and infrastructure to accommodate AVs, reliable V2X communication [5], [6], as well as the crucial need for comprehensive testing and simulation to ensure the reliability and safety of AVs. The last decade has seen intensified efforts dedicated to addressing these challenges, setting the stage for this novel era of transportation.

Advanced Driver-Assistance Systems (ADAS) represent an essential step in this evolution, encompassing various systems designed to automate, adapt, and enhance vehicle systems for safety and better driving. In particular, Autopilot and Openpilot stand as two of the most mature ADAS systems. Autopilot, developed by Tesla, and Openpilot, a project by Comma.ai, use a combination of radar-based sensors and cameras to achieve features such as Adaptive Cruise Control (ACC) and Lane Keeping (LK). Openpilot distinguishes itself as an end-to-end solution where a deep neural network processes raw sensor input to output driving commands [7].

In [4], authors highlight the role of simulation in the verification and validation of AVs and propose a high-level roadmap for future research. The testing and validation of such systems, especially those aspiring to higher autonomy levels, are of paramount importance. In this context, simulation platforms like CARLA, PTV Vissim, Autoware ROS, and DYNA4 have gained traction. Authors in [4] (cf. table VI) present an exhaustive comparison between simulation environments. CARLA [3] stands out as the defacto open-source simulator for AV research, providing a wide variety of urban and suburban environments.

Edge case scenarios have garnered significant attention. These scenarios, which represent unusual or extreme driving conditions, are critical for testing and improving the robustness of self-driving algorithms. However, they are inherently challenging to replicate in real-world testing due to their rarity and potential risk. Therefore, simulation environments have become an essential tool for exploring these edge-case scenarios in a safe and controlled manner. Several studies have highlighted the importance of edge case testing in autonomous driving. For instance, [4] emphasized the role of edge case scenarios in revealing the limitations of current self-driving technologies, while [8] proposed a framework for systematically generating and testing edge case scenarios in simulation environments.

While traditional simulation environments have been instrumental in testing edge-case scenarios, they often lack the complexity and diversity of real-world driving conditions. This is where video games, such as GTAV, present an intriguing alternative. Videogames offer immersive, dynamic, and unpredictable scenarios that closely resemble real-world situations. Their detailed environments are advantageous for testing edge-case scenarios in autonomous driving, enabling a broad spectrum of situations to be simulated. Utilizing videogames as simulation environments could expedite the development of self-driving algorithms due to their cost-effectiveness and scalability in generating diverse driving scenarios. Therefore, the exploration of video games like GTAV as a simulation environment for autonomous driving is a promising avenue for future research [9].

GTAV is a popular video game known for its realistic and detailed virtual environment. The game's environment includes a wide variety of urban and rural settings, diverse weather conditions, and dynamic traffic scenarios, making it a potentially rich resource for autonomous vehicle simulation. The potential of GTAV as a simulation environment for autonomous driving has been recognized in a few studies [10], [11]. Other studies highlighted using GTAV as data generation and collection for AV testing [12], [13].

## III. Methodology

This study aims to compare the usability and realism of two different simulation environments, GTAV and CARLA, for testing the Openpilot self-driving algorithm in edge-case scenarios. In this section, we first highlight the necessity for edge case testing with AV. We then present the Openpilot architecture with respect to the AV architecture. We finally, present our experimental setup and testing procedure.

### A. The Importance of edge case testing for AVs

Autonomous driving environments can be broadly classified into three main categories: rural, highway, and urban environments. Each environment is characterized by unique parameters to which the autonomous vehicle must integrate and adapt its driving. For instance, urban environments are often characterized by intersections, signalization, and vulnerable road users, while rural environments may feature elements such as trees, mountains, and bends.

The task of adapting to these diverse parameters is challenging. In several instances, autonomous vehicles have been involved in fatal accidents due to their inability to effectively handle certain driving conditions. For example, on March 18, 2018, in Arizona, a pedestrian was killed by an autonomous vehicle that was driving at a speed 20 km/h above the recommended limit during nighttime [14]. The vehicle's braking time was insufficient to avoid the accident, resulting in the death of a vulnerable road user. In another crash incident, the self-driving vehicle failed to detect an emergency vehicle with flashing lights in a timely manner [15].

Meteorological conditions have also been responsible for several AV accidents. A notable example occurred on May 7, 2016, in Florida, when an autonomous vehicle, crashed in a tractor-trailer (the AV's sensor is believed to have been dazzled by the sun) [16]. Tests conducted by other drivers have revealed that self-driving vehicles have difficulty navigating snow-covered roads [17].

Moreover, where driverless vehicles may encounter resistance from people such as the incident in San Francisco where protesters placed traffic cones on the hoods of these autonomous cars as a form of protest against their expansion on city streets [18].

By simulating autonomous driving in a game environment like GTAV, we can test several extreme cases, such as driving on a rural track at night in snowy weather or driving in the wrong direction on a highway. This allows us to evaluate the performance of autonomous driving algorithms in a wide range of challenging scenarios, which is crucial for improving their safety and reliability.

### B. Openpilot software architecture

Openpilot is an advanced driver-assistance system that has been designed with a modular architecture [19], [20], which includes several key components related to autonomous vehicle (AV) operation: perception, localization and path planning, and control (cf. Fig. 1).

- Perception: Openpilot uses a combination of sensors and machine learning algorithms to perceive the environment around the vehicle. This includes detecting other vehicles, pedestrians, and road signs, as well as estimating the vehicle's relative position and velocity.
- Localization and Path Planning: Once the environment is perceived, Openpilot uses a module called the "planner"

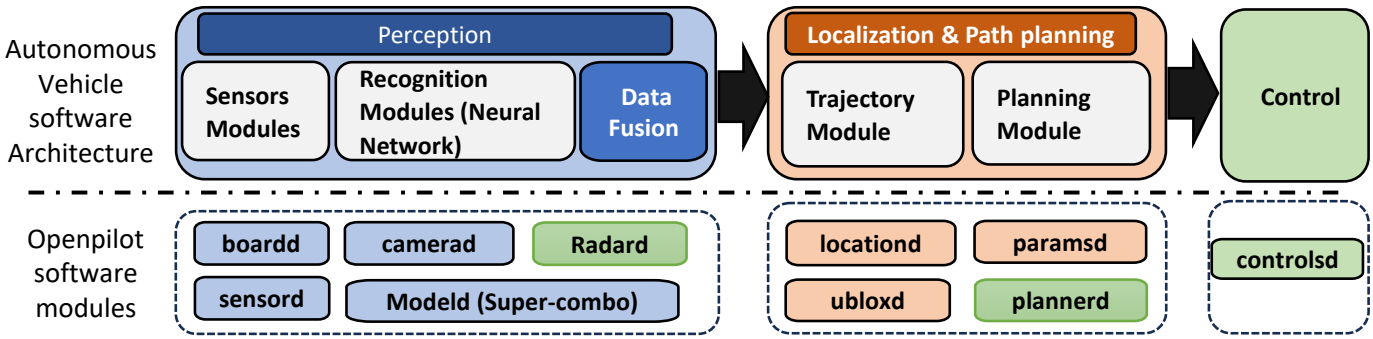


Fig. 1: Openpilot architecture w.r.t. to Autonomous Vehicle Overall Architecture.

to determine the vehicle's position within it and plan a safe path forward. This involves predicting the future positions of other road users and adjusting the planned path accordingly.

- Control: The final step in the process is controlling the vehicle to follow the planned path. Openpilot's control module, known as "controlsd", uses the planned path and the vehicle's current state to generate control commands for the vehicle's steering, throttle, and brakes.

### C. Experimental Setup

As shown in figure 2, the testing environment for this study was set up on an Ubuntu system (version 20.04). The GTAV game was installed using Wine, a compatibility layer that allows Windows applications to run on Linux operating systems like Ubuntu.

To interact with GTAV programmatically, we used ScriptHookV[1], a library that enables the game's native functions to be called from custom scripts. The necessary ScriptHookV files were placed in the GTAV installation folder, which allowed us to open a network socket for sending and receiving data from the game environment.

To facilitate the interaction with GTAV, we used DeepGTAV[2], a plugin for GTAV that provides a rich set of data from the game environment. We also used VPilot[3], a Python abstraction layer for DeepGTAV, which allowed us to control the game programmatically.

Using an Openpilot self-driving model version from the *littlemountainman/modeld* GitHub repository (available here https://github.com/littlemountainman/modeld), we adapted the Python script from the same repository in order to receive a video stream from GTAV. This video stream was then fed into the Openpilot self-driving model for processing[4].

Scenarios for testing the self-driving model were defined and launched using VPilotGUI[5], a graphical user interface for VPilot. This setup allowed us to simulate a wide range of driving conditions and scenarios in GTAV and evaluate the performance of Openpilot under these conditions.

[1]http://www.dev-c.com/gtav/scripthookv
[2]https://github.com/aitorzip/DeepGTAV
[3]https://github.com/aitorzip/VPilot
[4]https://github.com/idirsmadhi/OpenpilotGTAV
[5]https://github.com/hamouddd/VPilotGUI

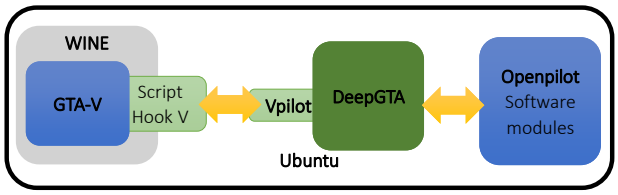


Fig. 2: Experimental setup architecture for GTAV.

In parallel to the GTAV setup, we also established an experimental environment using CARLA. CARLA was installed on the same Ubuntu system (version 20.04). For our testing procedure, we utilized CARLA's Python API to programmatically control the simulation environment and to feed sensor data to the Openpilot self-driving model. This setup allowed us to compare the performance of Openpilot in both GTAV and CARLA under similar edge-case scenarios.

### D. Quantitative evaluation of path prediction accuracy

In this work-in-progress paper, various trajectory similarity measures [21], [22], including Dynamic Time Warping (DTW), Frechet Distance [23], and others, were considered for evaluating the performance of Openpilot's path predictions in different edge-case scenarios. These sophisticated measures offer detailed insights into the alignment and similarity between predicted and ground truth trajectories, capturing complex variations and temporal distortions. However, for the sake of simplicity and practicality in this preliminary study, we opted to utilize the Point-to-Point (P2P) distance. The P2P distance provides a straightforward and intuitive measure of trajectory similarity, calculating the Euclidean distance between corresponding points in 2D space at each frame. Despite its simplicity, the P2P distance allows for a frame-by-frame assessment of the algorithm's path prediction accuracy, making it suitable for our current stage of research.

**Point-to-Point (P2PD) Distance (cf. Figure 3):**

Given two trajectories, $G = \{g_1, g_2, \ldots, g_n\}$ (for the ground truth path) and $P = \{p_1, p_2, \ldots, p_n\}$ (for the predicted path), where $n$ represents the number of frames.

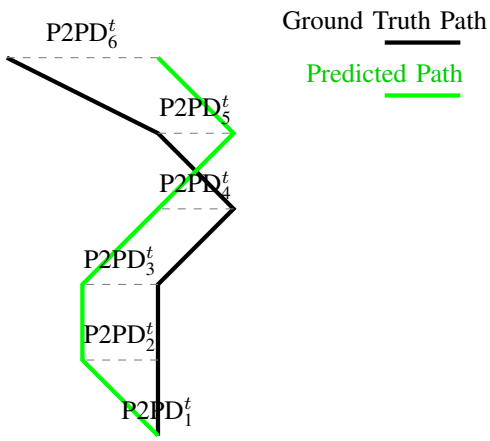


Fig. 3: Illustration of the Point-to-Point Distance (P2PD$^t$) between the Ground Truth Path and the Predicted Path at frame $t$. The dashed gray lines represent the P2PD$_{\mathrm{i}}^t$ distances at each $i$th sampled point in the frame $t$.

1. At each time step $t$ (frame $t$), along the Y-axis, sample points at equal intervals to define a set of sampling points for both trajectories.

2. For each frame $t$, calculate the Euclidean distance between the x coordinates of the corresponding sampling points in the trajectories:

$$\mathrm{P2PD}(G[t], P[t]) = \sum_{i=1}^{\text{num_samples}} \mathrm{P2PD}_t^i$$

where

$$\mathrm{P2PD}_t^i = g_{t_x}^i - p_{t_x}^i$$

and $g_{t_x}^i$ and $p_{t_x}^i$ are the x coordinates of the $i$th sampling point on the ground truth path and predicted path in frame $t$, respectively. num_samples represents the total number of sampling points along the Y-axis.

3. Repeat the calculation for all frames, from $t = 1$ to $t = n$.

4. Average the sum of the Basic Point-to-Point distances at each time step to obtain the total similarity measure:

$$\mathrm{P2PD}_{\text{total}} = \frac{1}{n} \sum_{t=1}^{n} \mathrm{P2PD}(G[t], P[t])$$

The Point-to-Point (P2PD) distance provides a frame-by-frame evaluation of the similarity between the predicted trajectory (P) and the ground truth trajectory (G), considering the x-coordinate distances of the sampled points along the Y-axis. This measure allows for a more detailed analysis of the algorithm's performance, considering the lateral accuracy of the predictions at each frame. It provides valuable insights into the model's ability to maintain accurate positioning in complex driving scenarios and edge cases.

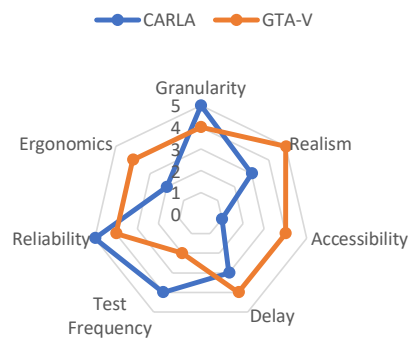


Fig. 4: Comparison between GTAV and CARLA simulation environments.

## IV. Preliminary Results

### A. AVs' simulation environments: GTAV Vs CARLA

Our preliminary results indicate that both GTAV and CARLA have their strengths and weaknesses as simulation environments for testing Openpilot.

As shown in figure 4, GTAV showed promising results in terms of usability and realism. The game's rich and dynamic environment provided a wide range of realistic driving scenarios, including several edge-case scenarios that are difficult to replicate in traditional simulators. This made GTAV a valuable tool for testing the robustness of Openpilot in diverse and challenging conditions.

On the other hand, CARLA excelled in terms of the granularity of scenarios. The simulator's comprehensive API allowed for a high level of control over the driving scenarios, making it possible to test Openpilot under very specific conditions. This was particularly useful for testing the algorithm's performance in edge-case scenarios, where small details can have a significant impact on the outcome.

### B. Preliminary edge-cases analysis and interpretation

In this work-in-progress paper, We will be presenting and discussing a few scenarios.

It's worth noting that certain scenarios would pose considerable challenges not only for autonomous driving systems but also for human drivers. These scenarios, characterized by adverse weather conditions, low visibility, or complex road layouts, are inherently difficult to navigate safely. Human drivers may also encounter difficulties in making accurate predictions and decisions in such situations, highlighting the complexity of these edge cases. This observation underscores the importance of developing robust autonomous driving systems capable of handling these challenging scenarios, as they can occur in real-world driving conditions.

Our preliminary testing focused on four specific edge-case scenarios: (i) pedestrian detection in a nighttime environment with heavy rain; (ii) reaction to sudden changes in brightness, such as during a storm or when exiting tunnels; (iii) reaction to obstacles on a winding road; and finally (iv) driving on a snowy road.

### C. Night Driving Scenario

In the night driving scenario, we conducted tests in both CARLA and GTAV environments to assess the performance of Openpilot in predicting trajectories under low-light conditions. However, our findings revealed that Openpilot struggled to accurately predict the correct trajectory in both environments, resulting in a significant decrease in trajectory precision (see Figure 5).

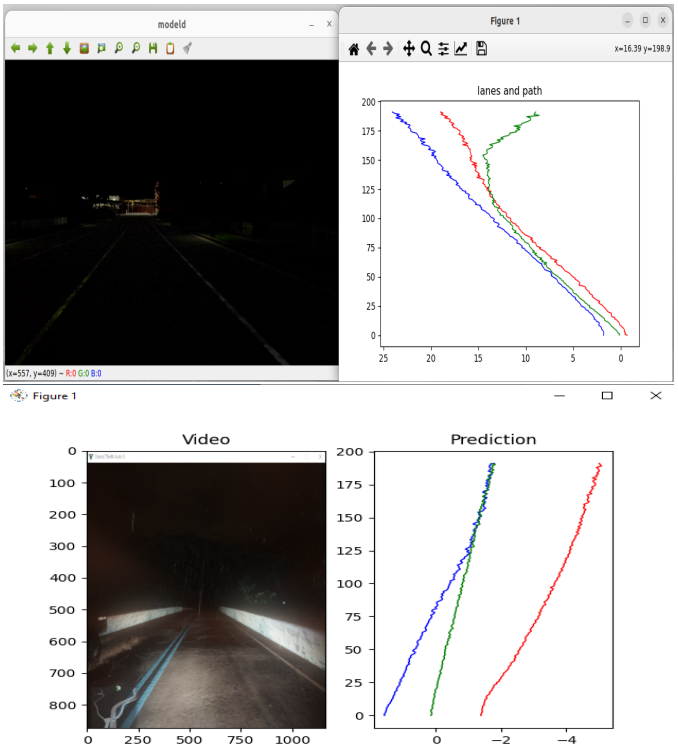


Fig. 5: Illustration of Openpilot's trajectory prediction in the night driving scenario (top: CARLA, bottom: GTAV).

### D. Challenging Weather Scenario

Under challenging weather conditions, such as rain and fog in CARLA and snowy rural roads at night in GTAV, the Openpilot algorithm faced difficulties in predicting correct trajectories (see Figure 6).

### E. Combined Scenario - Snowy Night Driving

In this combined scenario, we evaluated the Openpilot prediction algorithm on a rural road covered in snow during nighttime in the GTAV environment. The results were particularly challenging, as the algorithm struggled to accurately detect and track the road lines. The reduced visibility caused by the snowy weather and darkness seemed to significantly impact the algorithm's ability to perceive and respond to the road markings (see Figure 7).

In these extreme scenarios, the path prediction showed an oscillating detection of the road edges, indicating that the algorithm was struggling to accurately perceive and navigate

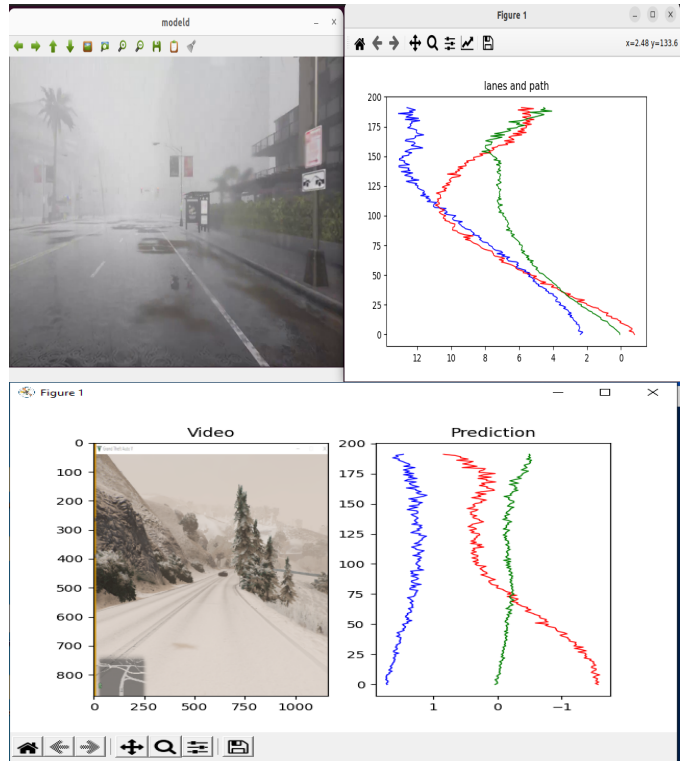


Fig. 6: Illustration of Openpilot's trajectory prediction under challenging weather conditions (top: CARLA, bottom: GTAV).

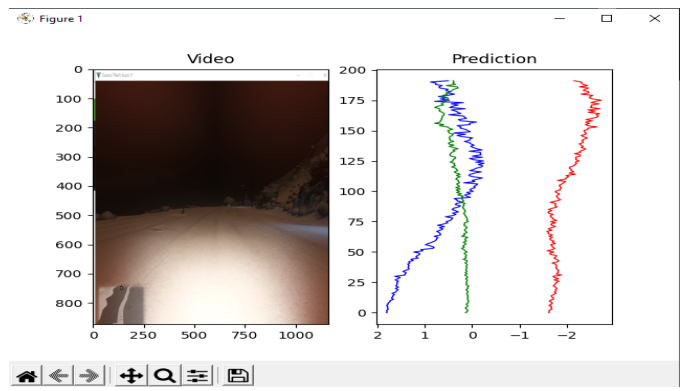


Fig. 7: Illustration of Openpilot's trajectory prediction in the combined scenario of snowy night driving.

the challenging conditions. Such a prediction could result in unpredictable and potentially unsafe driving actions.

These preliminary results highlight the importance of testing self-driving algorithms in a wide range of scenarios, particularly those that combine multiple challenging conditions. They also underscore the need for further research and development to improve the performance of autonomous driving algorithms in extreme scenarios.

## V. Conclusion and Future Work

Our preliminary results highlight the potential of using video games, specifically GTAV, as a simulation environment

for testing self-driving algorithms. GTAV excels in simulating realistic urban environments with intricate traffic patterns, dynamic weather conditions, and highly detailed nighttime settings. Additionally, GTAV models realistic pedestrian behavior and allows for diverse non-vehicular obstacles on roads. Its flexibility and scalability enable the creation of customized scenarios tailored to research needs. These advantages make GTAV a valuable choice for testing and training autonomous vehicle systems in a wide range of challenging scenarios.

These results also underscore the challenges that self-driving models face when dealing with complex, multi-factor edge-case scenarios. Specifically, we tested Openpilot on specific edge-case scenarios on both GTAV and CARLA simulation environments. These scenarios provided valuable insights into how the self-driving model responds to different driving conditions and how it can be improved to handle edge-case scenarios more effectively. In particular, we visually examined the predicted paths generated by Openpilot for various edge-case scenarios in both CARLA and GTAV environments. As a disclaimer, it is important to note that this qualitative discussion does not serve as a comprehensive evaluation of the Openpilot self-driving model itself, but rather aims to use it as a general path prediction capability in our simulation environments.

In future work, we will focus on refining the testing procedure and further exploring the potential of video games as a simulation environment for autonomous driving research. Specifically, we intend to quantitatively measure the Point-to-Point Distance (P2PD) using the annotated ground truth paths to provide a more rigorous and objective evaluation of the model's performance. We plan to test a wider range of edge-case scenarios and to further investigate the performance of Openpilot under these conditions. These results could potentially provide suggestions for enhancing the performance of self-driving technology in extreme scenarios by refining its perception, planning, and control modules.

## Acknowledgment

We would like to acknowledge the contributions of the authors to this study: Naila Bouchemal and Aberrahmane Maaradji contributed equally to the research efforts; Idir Smadhi focused on GTAV scenarios simulation; Ahmed Bouhraoua developed the scenario specification GUI and P2PD measure; and Amayas Ghanine contributed to CARLA scenarios simulation.

We also express our sincere gratitude to all the students who contributed to this study through study groups.

## References

[1] "COMMA Openpilot's Github repository." [Online]. Available: https://github.com/commaai/openpilot

[2] B. C. Zanchin, R. Adamshuk, M. M. Santos, and K. S. Collazos, "On the instrumentation and classification of autonomous cars," *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, vol. 2017-January, pp. 2631–2636, 11 2017.

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," *Proceedings of the 1st Annual Conference on Robot Learning, PMLR*, vol. 78, pp. 1–16, 11 2017.

[4] R. Chakra, "Exiting the Simulation: The Road to Robust and Resilient Autonomous Vehicles at Scale," 10 2022. [Online]. Available: https://arxiv.org/abs/2210.10876v1

[5] N. Bouchemal and S. Kallel, "Testbed of V2X infrastructure for autonomous vehicles," *Annales des Telecommunications/Annals of Telecommunications*, vol. 76, no. 9-10, pp. 731–743, 10 2021.

[6] N. Bouchemal and J. Y. Jun, "V2X Architecture for Autonomous Platoon Management in Urban Environment," *Proceedings of the International Conference on Computer, Information, and Telecommunication Systems, CITS 2021*, 2021.

[7] T. Sato, J. Shen, N. Wang, Y. J. Jia, X. Lin, and Q. A. Chen, "Security of Deep Learning based Lane Keeping System under Physical-World Adversarial Attack," 3 2020. [Online]. Available: https://arxiv.org/abs/2003.01782v1

[8] S. Ramakrishna, B. Luo, C. B. Kuhn, G. Karsai, and A. Dubey, "ANTI-CARLA: An Adversarial Testing Framework for Autonomous Vehicles in CARLA," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2022-October, pp. 2620–2627, 2022.

[9] H. Abbas, M. O'Kelly, A. Rodionova, and R. Mangharam, "Safe At Any Speed: A Simulation-Based Test Harness for Autonomous Vehicles," in *Cyber Physical Systems. Design, Modeling, and Evaluation. CyPhy 2017*, C. Lecture Notes in Computer Science(), vol 11267. Springer, Ed., 2019, pp. 94–106.

[10] M. Martinez, C. Sitawarin, K. Finch, L. Meincke, A. Yablonski, and A. Kornhauser, "Beyond Grand Theft Auto V for Training, Testing and Enhancing Deep Learning in Self Driving Cars," *arxiv.org*, 12 2017. [Online]. Available: http://arxiv.org/abs/1712.01397

[11] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to Learn: Autonomously Identifying Perception Failures for Self-Driving Cars," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3860–3867, 10 2018.

[12] M. Cao and R. Ramezani, "Data Generation Using Simulation Technology to Improve Perception Mechanism of Autonomous Vehicles," *Journal of Physics: Conference Series*, vol. 2547, no. 1, p. 012006, 7 2023.

[13] B. Hurl, K. Czarnecki, and S. Waslander, "Precise Synthetic Image and LiDAR (PreSIL) Dataset for Autonomous Vehicle Perception," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 6 2019, pp. 2522–2529. [Online]. Available: https://ieeexplore.ieee.org/document/8813809/

[14] "Death of Elaine Herzberg." [Online]. Available: https://en.wikipedia.org/wiki/Death_of_Elaine_Herzberg

[15] "Watch: Exclusive Tesla Footage Suggests Reasons for Autopilot Crashes." [Online]. Available: https://www.wsj.com/video/series/in-depth-features/watch-exclusive-tesla-footage-suggests-reasons-for-autopilot-crashes/2FBEE1CA-56E1-4ACC-92B6-DED638B531CE

[16] "Tesla driver dies in first fatal crash while using autopilot mode." [Online]. Available: https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk

[17] "Tesla's Full Self-Driving Struggles In Snow-Covered Edmonton." [Online]. Available: https://www.carscoops.com/2022/11/teslas-full-self-driving-struggles-in-snow-covered-edmonton/

[18] "Protesters develop novel way to build cone-sensus against driverless cars." [Online]. Available: https://www.theguardian.com/technology/2023/jul/07/san-francisco-autonomous-cars-protest-cone

[19] "How openpilot works in 2021 - comma.ai blog." [Online]. Available: https://blog.comma.ai/openpilot-in-2021/

[20] F. FONTANA, "Self-driving cars and Openpilot : a complete overview of the framework," Tech. Rep., 12 2021. [Online]. Available: https://www.politesi.polimi.it/handle/10589/181889

[21] J. Quehl, H. Hu, O. S. Tas, E. Rehder, and M. Lauer, "How good is my prediction? Finding a similarity measure for trajectory prediction evaluation," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-March, 2018.

[22] L. Zheng, H. Song, B. Li, and H. Zhang, "Generation of lane-level road networks based on a trajectory-similarity-join pruning strategy," *ISPRS International Journal of Geo-Information*, vol. 8, no. 9, 2019.

[23] J. Laconte, A. Kasmi, R. Aufrère, M. Vaidis, and R. Chapuis, "A survey of localization methods for autonomous vehicles in highway scenarios," 2022.