S-RAF: A Simulation-Based Robustness Assessment Framework for Responsible Autonomous Driving

Daniel Omeiza[1*], Pratik Somaiya[1], Jo-Ann Pattinson[3], Carolyn Ten-Holter[1], Marina Jirotka[1], Jack Stilgoe[4], Lars Kunze[1,2]

[1]University of Oxford, UK,
[2]University of the West of England,
[3]University of Leeds,
[4]University College London
{danielomeiza, pratik, lars}@robots.ox.ac.uk, carolyn@cs.ox.ac.uk, j.m.pattinson@leeds.ac.uk, marina.jirotka@cs.ox.ac.uk, j.stilgoe@ucl.ac.uk

Abstract

As artificial intelligence (AI) technology advances, ensuring the robustness and safety of AI-driven systems has become paramount. However, varying perceptions of robustness among AI developers create misaligned evaluation metrics, complicating the assessment and certification of safety-critical and complex AI systems such as autonomous driving (AD) agents. To address this challenge, we introduce Simulation-Based Robustness Assessment Framework (S-RAF) for autonomous driving. S-RAF leverages the CARLA Driving simulator to rigorously assess AD agents across diverse conditions, including faulty sensors, environmental changes, and complex traffic situations. By quantifying robustness and its relationship with other safety-critical factors, such as carbon emissions, S-RAF aids developers and stakeholders in building safe and responsible driving agents, and streamlining safety certification processes. Furthermore, S-RAF offers significant advantages, such as reduced testing costs, and the ability to explore edge cases that may be unsafe to test in the real world.

Introduction

Responsible AI (RAI) development has gained increased attention lately as the development and application of sophisticated AI technologies continue to increase immeasurably with threats of harm to society (Allen and Weyl 2024). Beyond the assessment of technical capabilities (e.g., accuracy), the potential threats these technologies pose highlight the need for a more collective assessment of their *purpose* and *process* to mitigate associated risks while harnessing their potential for good. Of late, deep tech corporations, e.g., autonomous vehicle companies, seem to operate in silos regarding the communication of new knowledge, safety test cases, and reports. Many develop their own metrics and benchmarks for assessing their technologies, thus, making it difficult for regulators and other parties of interest to have a fair assessment of these technologies across board. We, therefore, advocate for the development of accessible frameworks for easier and more objective assessment of safety-critical responsible AI principles such as robustness, and environmental sustainability, among others.

We conceptualise RAI as the conscious effort in designing, developing, and deploying artificial intelligence (AI) systems in a way that maximises their benefits while minimising their risks to people and society at large.

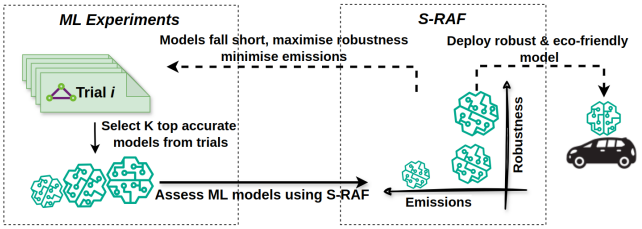

Figure 1: Overview of S-RAF. Trained agents/models from the ML trials are selected and passed to S-RAF for comprehensive robustness and CO_2 emission assessment, and are ranked accordingly.

If we are developing artefacts to act with some autonomy, then 'we had better be quite sure that the purpose put into the machine is the purpose which we really desire' as quoted in (Dignum 2019). In this light, the development of a framework for assessing each dimension of RAI emerges as a pivotal tool for *responsible innovation,* a broader subject guiding the development of technologies that align with societal needs and values (Stilgoe, Owen, and Macnaghten 2020; Jirotka et al. 2017). While purposes are very hard to technically account for, processes might be amenable. Thus, we developed a robustness assessment framework (S-RAF) that focuses on assessing how AD agents handle unforeseen situations during operation (Marcus 2020). AD is a safety critical domain with reports of accident cases (Reed 2024; Lavrinc 2016). Hence, there is a need for responsible innovation involving rigorous assessment of safety-critical RAI principles such as robustness.

While there might be a finite list of RAI principles in the academic literature, as a first step, we have focused on robustness (a safety critical concept in AD) and examined how efforts to improve robustness impact carbon emission,

*Corresponding author.

an indicator for environmental sustainability in this paper. We tested three state-of-the-art AD models to demonstrate how RAF can be applied in practical settings. This encompasses the assessment of robustness and the investigation of trade-offs with sustainability and transparency (see Fig. 1).

While there are a few accessible developments around RAI, most of the efforts are limited to guidelines, conceptual frameworks, and toolkits. The existing toolkits are mainly targeted toward regular machine learning models and are impracticable for assessing complex goal-based systems such as AD agents. Moreover, they also mainly support single modality input (either RGB sets or text/audio sequence) and/or require structured input types. In line with the notion of the 'veil of ignorance' (Rawls 2020), where impartial decisions are made without knowledge of one's social position, S-RAF aims to embody a similar principle by ensuring unbiased robustness assessments of complex AD agents.

The Need for RAI Indicators

We focus on robustness and sustainability indicators in this section.

Robustness One of the core principles of RAI is robustness (Floridi 2021). In contrast to symbolic AI systems, deep learning models often deployed in AD undergo training with extensive datasets, and their complex structures, which may consist of millions or billions of parameters, are not collectively readily interpretable by humans. Consequently, it is currently impractical to offer comprehensive assurances regarding the accurate performance of neural networks when they encounter input data significantly divergent from what was seen during training. Nonetheless, numerous AI applications (including AD) have critical security or safety implications, necessitating the ability to assess the systems' resilience when confronted with unforeseen events, whether they arise naturally or are deliberately induced by malicious actors (Berghoff et al. 2021). There are many methods of adversarial attacks such as those created using generative models like GAN, VAE (Bowles et al. 2018), and those created by adversarial perturbations like LBFGS (Szegedy et al. 2013), FGSM (Goodfellow, Shlens, and Szegedy 2014), Deep Fool (Moosavi-Dezfooli, Fawzi, and Frossard 2016). On the other hand, multiple works exist addressing this question of robustness against such attacks (Zhang et al. 2020). Some works (Huang et al. 2017; Singh et al. 2019) have leveraged formal verification to provide robustness guarantees. These methods are faced with the limitations of scalability to the large neural networks often used in practice. Moreover, the defence provided is usually against artificial attacks at the pixel level. Hence, our notion of robustness in this paper is the ability of the AD agent to maintain a consistent behaviour or gracefully handle unforeseen events that might **naturally** occur in its operating environment. We seek to address questions such as, what happens when a sensor of an autonomous vehicle (AV) malfunctions? What happens when it is occluded by dirt or other materials? What happens when there is a sudden serious decline in weather conditions? These conditions are serious and can lead to fatal accidents when not effectively handled. For instance, when a camera fails, one expects the other cameras to make up for this failure, and do so without constituting safety risk or violating traffic rules. While the primary focus of this paper is on robustness, researchers have drawn connections between robustness and other indicators, especially sustainability. With many AI companies following the scaling law—which suggests that increasing the number of model parameters leads to improved performance—the demand for computational resources continues to grow. This escalating resource consumption consequently results in higher carbon dioxide CO_2 emissions.

Carbon Emission According to Strubell et al. (Strubell, Ganesh, and McCallum 2019), the training process of a single deep learning natural language processing (NLP) model on a GPU can result in the emission of approximately $\sim 272,155kg$ of CO_2, comparable to the lifetime emissions of five cars. Similarly, Google's AlphaGo Zero generated 96 metric tonnes of CO_2 during 40 days of training, equivalent to 1,000 hours of air travel or the carbon footprint of 23 American households (Van Wynsberghe 2021). The environmental impact of recent Generative AI models (e.g., GPT-3, ChatGPT, DALL-E, etc.) is even more concerning. Efforts to build more robust, multi-tasking models are observed to negatively impact environmental sustainability. Other than train time emissions, applications like ChatGPT, handling 11 million requests per hour, emit 12.8 thousand metric tons of CO_2 annually, 25 times higher than GPT-3's training emissions (Chien et al. 2023). This has implications for AD, where such models are increasingly integrated. Research has assessed AI's energy use and CO_2 emissions (Strubell, Ganesh, and McCallum 2019; Lacoste et al. 2019), focusing on the carbon impact of various Graphics Processing Units (GPUs). We adopt this approach and report CO_2 emissions of benchmark AD agents by tracking their software processes.

Previous RAI Efforts

RAI Frameworks The development of RAI framework (including tools) is essential for all the different stages of AI development and deployment. Several efforts have been channelled toward developing RAI guidelines and frameworks over the years. Based on the survey by (Berman, Goyal, and Madaio 2024), the efforts include the provision of guidance for problem formulation and procurement decisions for the appropriate AI system for the given use case as equally argued in (Coston et al. 2023). It also includes ethical considerations for designing the systems, e.g., checklists (Lifshitz and McMaster 2020), and procedures for enabling participatory design. In the actual machine learning workflow, dataset collection and training processes benefit from established ethical guidelines (Rhem 2023). Some of the frameworks are also useful for model post-training activities, e.g., for fairness assessment (Agarwal et al. 2018), model-card (Mitchell et al. 2019), and datasheet (Gebru et al. 2021) for model training documentation, dataset details, and model reporting. Another use case is the facilitation of an effective AI system auditing process (Krafft et al. 2021). These efforts are limited to data-driven models.

Other than the CARLA Leaderboard (wayve 2023), which is mainly performance-focused, there seems to be a dearth of accessible frameworks and tools in AD that are focused on RAI principles. Our work contributes to the efforts by defining metrics for and providing a software tool for assessing the robustness of AD agents and understanding connections with carbon emissions.

RAI Tools RAI tools have been developed for use by AI practitioners, of which a few focus on robustness (e.g., ART (Nicolae et al. 2018), FoolBox (Rauber, Brendel, and Bethge 2017), RobustBench (Croce et al. 2020)), explainability (e.g., Google What-if tool (Wexler et al. 2019), Captum (Kokhlikyan et al. 2020), etc), a handful on sustainability (e.g., CodeCarbon (Courty et al. 2024) and Eco2AI (Budennyy et al. 2022)), fairness (e.g., IBM Fairness 360 (Bellamy et al. 2019), Fairlearn (Microsoft and Contributors 2019), etc). Our work spans robustness to include CO_2 emission tracking and supports complex multi-modal AD agents.

In summary, building responsible AD agents requires additional efforts beyond adversarial robustness for models with single input modality, and should be done without trading off environmental sustainability. This is the gap that S-RAF potentially seeks to fill.

S-RAF: Robustness Indicators

We consider an agent to be robust if it can sustain performance in the presence of environmental disturbances, measurement noise, and data drift without infractions of traffic rules or collisions.

Robustness against Environmental Disturbances

Robustness against environmental disturbances is paramount to ensure the safe and reliable operation of vehicles. There are different types of disturbances peculiar to different sensor types:

i. Camera Occlusion Environmental materials such as dirt, leaves, and snow accumulation on sensors, among others can cause camera occlusion. Camera occlusion occurs when the camera's field of view is obstructed, leading to incomplete or inaccurate perception data. In this disturbance situation, we assume that the disturbance occurs directly on the camera lens.

Formally, let $I_{\text{original}}(x, y)$ represent the original image captured by the camera, where x and y denote the spatial coordinates of the image. Image occlusion can be represented by introducing an occlusion mask $M(x, y)$ that indicates the regions of the image affected by occlusion. The occluded image $I_{\text{occluded}}(x, y)$ is defined as:

$$I_{\text{occluded}}(x, y) = I_{\text{original}}(x, y) \cdot (1 - M(x, y))$$

ii. LiDAR Occlusion Similar to cameras, occlusion can arise in 3-dimensional (3D) Light Detection and Ranging (LiDAR) from environmental factors, including leaves, snow, dirt, etc. Such environmental elements have the potential to obscure the sensor's enclosure, leading to instances of occlusion where specific regions of the observed scene become inaccessible or exhibit data incompleteness.

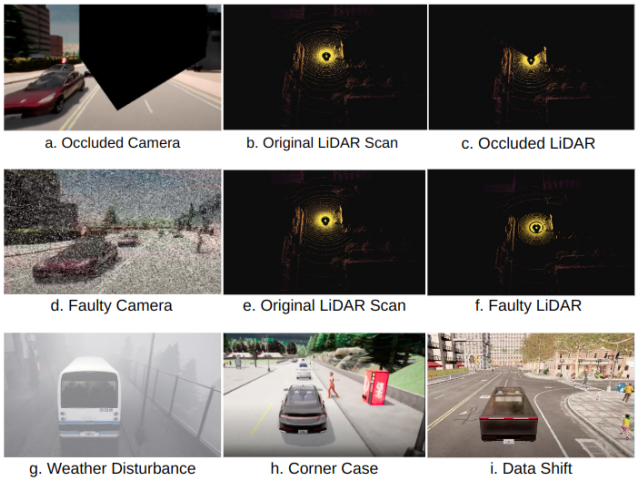

Figure 2: The first row shows samples of camera occlusion (Fig. 2a) and LiDAR occlusion (normal: Fig. 2b, occluded: Fig. 2c). 2nd row shows faulty sensors for a camera (Fig. 2d) and for LiDAR (normal: Fig. 2e, faulty with missing rays: Fig. 2f). Third row shows a rainy weather (2g), a jaywalker (Fig. 2h), and data drift example, a chaotic scene (Fig. 2i).

If the sensor data is represented as $S(x, y, z)$, where x, y, and z are spatial coordinates of each LIDAR point, then occluded data can be written as:

$$S_{\text{occluded}}(x, y, z) = S_{\text{original}}(x, y, z) \cdot (1 - M(x, y, z))$$

where $M(x, y, z)$ denotes the occluded section.

Fig. 2(c) illustrates an instance of occlusion observed in LiDAR data.

iii. Weather Disturbances Adverse weather conditions such as heavy rain, fog, snow, or glare can introduce noise and distortions in sensor data, affecting the vehicle's ability to perceive its surroundings accurately. Raindrops on camera or LiDAR sensors can scatter light and cause reflections, leading to blurred or obscured images. CARLA simulator already provides a simulation of these different weather conditions, and we leveraged this in our experiment.

Robustness against Sensor Errors

Sensor noise resulting from hardware faults, calibration issues, or sensor drifts can lead to erroneous readings and misinterpretations of the environment, consequently impacting critical tasks such as obstacle detection, and planning. Detecting and mitigating sensor errors is crucial for ensuring the robustness and accuracy of AD systems.

i. Camera Error Errors associated with faulty camera sensors, dead sensor elements, inaccurate pixel interpretation process, and intermittent electrical interference result in poor camera outputs (modelled with salt and pepper noise). This type of noise manifests as sparsely occurring white and black pixels in images, affecting the visual quality and integrity of the captured data. High-amplitude intermittent electrical interference, such as arcing on electrical contacts,

can also contribute to the occurrence of salt and pepper noise in camera outputs. We add salt and pepper noise to each pixel independently following a probability p. Where p is the probability that an arbitrary pixel ($I_{\text{original}}(x, y)$) gets assigned 0 and $1-p$ for 255. Understanding the sources of salt and pepper noise is crucial for developing effective noise removal techniques to enhance the quality of images captured by cameras in AD systems.

ii. LiDAR Error 3D LIDARs are incorporated with multiple channels to enable a higher vertical field of view (FOV), where each channel is a distinct laser beam. One example of hardware failure in 3D LIDARs is channel failure, where a particular beam stops functioning. To simulate such a fault, we remove sensor readings corresponding to arbitrarily selected channels. See examples of noisy and faulty sensor outputs in Fig. 2.

iii. Other Sensors Errors In addition to cameras and LiDARs, we introduced random noise, drawn from a uniform distribution, to the sensor data coming from positional measurement-related sensors such as; the Global Navigation Satellite System (GNSS), the Inertial Measurement Unit (IMU), and the speedometer. These sensors are naturally susceptible to faults due to hardware limitations, signal interference, or environmental conditions. Hence, assessing driving agents' robustness along these dimensions is important. Given sensor data S, noised sensor data $\hat{S}$, is obtained as: $\hat{S} = S + n \; ; \; n \sim U(-N, N)$ where n represents added noise, U denotes uniform distribution, and N controls the magnitude of the noise.

Robustness against Corner Cases

i. Corner Cases Corner cases represent extreme situations that may not be encountered in regular traffic but have the potential to challenge the capabilities of AD systems. This includes situations where an actor violates traffic rules, e.g., a pedestrian crossing outside of a crosswalk, debris on the road, etc.

It could also be the data drift effect, which occurs when there's a divergence between the test (or production) data distribution and the train data distributions. This potentially leads to model degradation and decreased accuracy. This could be a collective change in the behaviour of actors within a region, e.g., from being conservative to aggressive, as the population density of actors increases in the region. It could also be the fading away of traffic signs. By monitoring data drift and implementing strategies to adapt models to changing data patterns, AD systems can maintain their effectiveness in diverse and evolving environments. Robustness can be ensured by utilising techniques such as drift detection, model re-calibration, and ongoing performance evaluation to ensure that the models remain robust and effective in the face of fluctuating environmental variables. We leverage CARLA's flexibility in defining environments and pedestrian behaviour to create a combination of these situations. We can create data shift scenarios if the details, e.g., the source of training data and its distribution, are included in the datasheet.

S-RAF: Carbon Emission Indicator

It is challenging to obtain an accurate measure of the overall CO_2 autonomous vehicles emits to the environment. However, we can estimate how much of CO_2 the AI model that powers the vehicles constitutes at inference time. We estimated this by obtaining the carbon intensity (CI) value for the region from which the model is run and multiplying this by the amount of energy (E) the process running the model used up: CO_2 emissions (in Kg CO_2Eq.) $= \text{CI} \times E$.

The Carbon Intensity (CI) of the consumed electricity is calculated as a weighted average of the emissions from the different energy sources that are used to generate electricity, including fossil fuels and renewables. In (Courty et al. 2024), fossil fuels coal, petroleum, and natural gas are associated with specific carbon intensities based on a known standard amount of CO_2 emitted for each kilowatt-hour of electricity generated. This is based on publicly available charts. Renewable or low-carbon fuels include solar power, hydroelectricity, biomass, geothermal, etc. The nearby energy grid contains a mixture of fossil fuels and low-carbon energy sources, called the Energy Mix. Based on the mix of energy sources in the local grid, the Carbon Intensity of the electricity consumed is calculated. The nearby grid is determined based on the location of the compute resource.

E is the energy consumed by the computational infrastructure (both GPU and RAM) expressed in kilowatt-hours. See (Courty et al. 2024) for further implementation details.

Experiment

Traffic Setup

The traffic setup was done in CARLA simulator (Dosovitskiy et al. 2017). For the experiment reported in this paper, we created a complicated route and spawned multiple actors to roam around the town. The route consists of different road structures, including junctions and intersections (based on the NHTSA (National Highway Traffic Safety Administration 2007) topology). Actors include different forms of vehicles, from cyclists to trucks, interacting in different forms with the agent being tested. This traffic setup is similar to those used in the previous CARLA AD Challenge (wayve 2023).

Driving Agent Details

We selected three trained agents, LBC (Chen et al. 2020), NEAT (Chitta, Prakash, and Geiger 2021), and InterFuser (Shao et al. 2023), from the 2020, 2021, and 2022 CARLA Challenges, respectively based on the leaderboard results. We selected one agent from each year using two steps (i) sorting by driving scores (ii) selecting the top agent that provided enough details for easy reproducibility.

The authors of the selected agents were quite transparent by providing model weights, code, and other materials useful for running S-RAF. We refer to this as *process-based transparency*, where detailed information about the processes involved in the agent's development stage is made available. Other forms of transparency important for responsible autonomous driving include *output-based transparency* where the agent provide human-understandable ex-

			Process						
Model	**Out. (2)**	**Mtd.**	IC	Code	CD	DS	MW	MC	TR
LBC	✗	E2E	✗	✓	✓	✓	✓	✓	✓
NEAT	✗	E2E	✓	✓	✓	✓	✓	✓	✓
InterFuser	✗	E2E	✓	✓	✓	✓	✓	✓	✓

Table 1: Qualitative details on transparency of agents. Out.: Output; Mtd: Method; E2E: End-to-End; IC: Interpretability Considerations (i.e., whether authors implemented interpretability techniques); CD: Code Documentation; MW: Model Weights; MC: Model Card; DS: Datasheet; TR: Technical Report.

planations for their predictions, and *method-based transparency* which relates to the architecture used, e.g., modular or end-to-end (Omeiza et al. 2021). A modular architecture is assumed to offer better transparency. We examined the available technical materials (e.g., code base, research paper, etc) of the selected agents to see how they align with these different transparency dimensions. As this is mainly a qualitative assessment and quite subjective, we do not consider it an indicator in S-RAF. See Table 1 for details.

Regular Driving Score

We used the driving score metric to assess the driving performance of the agents. The driving score $D_i = R_i P_i$ is the product of the route completion and the infraction penalty. Here R_i is the percentage of completion of the $i-th$ route and P_i is the infraction penalty incurred during the $i-th$ route.

We track different types of infractions (e.g., collision with a vehicle, running a red light, etc.) in which the agent was involved. The infraction penalty score aggregates all of the infractions as a geometric series along a particular route. Agents start with an ideal 1.0 base penalty score, which is reduced each time an infraction is committed. Infraction penalty P_i was computed as: $\prod_{l}^{|W|} (P_l)^{|W_l|}$, where W is a set of infractions, l is an infraction type in W, and $|W_l|$ is the number of W_l infractions that occurred, and P_l is the cost for infraction type l.

Robustness Driving Score

We ran one route multiple times, introducing each of the different types of disturbances in each run. For example, if the traffic disturbance was weather and there were three weather types, then the route would be run three times corresponding to the three different weather types. The run condition, in this case, is 'weather'. When a route has been run for the desired number of times:

1. The driving scores (i.e., after penalties have been applied) for the runs are grouped based on the type of traffic disturbances introduced (condition). For example, weather driving scores [...], camera noise driving scores [...], ...
2. The driving score for each type of traffic disturbance/-condition j is obtained by selecting the minimum score across all runs k under condition j. from the runs under this condition: $D_i^j = \min_k(D_{i,k}^j)$. For example, if a route were run in n different weather conditions, the driving score for condition weather would be the least driving score obtained after running the agent in the n different weather conditions. In the situation that more than one routes were used, the score from the route that produced the lowest score for the selected traffic disturbance type would be used.
3. A robustness ratio is computed per condition. Robustness ratio for $j-th$ condition is the ratio of the driving score obtained when the disturbance is introduced and the driving score obtained in the normal/regular driving case where no disturbance was introduced, that is, $s_j = \frac{D_i^j}{D_i}$
4. A robustness driving score for which agents are ranked is computed by taking the mean of all D_i^j. $RDS = \frac{1}{m}\sum_j^m D_i^j$ where m is the total number of conditions.

Estimating Carbon Emissions

For each run, we estimate the amount of carbon (in Kg CO_2 Eq.) emitted as a result of running the agent. Note that only the system process running the agent is tracked. We provide the average CO_2 emission for the runs as well as the average CO_2 emission per second. The emissions estimation procedure is based on the codecarbon (Courty et al. 2024) implementation explained in the previous section. Code is available at https://github.com/cognitive-robots/rai-leaderboard.

Results

Robustness

From Table 2 and Fig. 3, we observe that Interfuser had the highest driving score and overall robustness driving score. This is not surprising as it has the highest number of sensors for accurate perception and planning. Camera seems to be the most intolerant to faults as we observe the greatest decline when the camera was noised. Cameras are very important sensors for navigation. In fact, some AD companies are beginning to only rely on cameras for navigation. Robustness readings need to be put in perspective with average route completion (ARC) and driving score. For instance, while LBC seems to have impressive robustness scores for many of the runs, it should be noted that the agent was only able to complete 28% of each route on average. Its driving

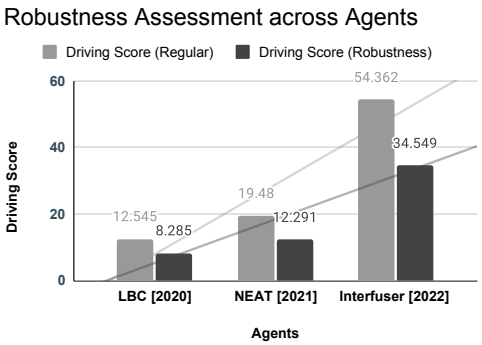

Figure 3: All agents' performance dropped with disturbances. CARLA's NPC Agent was excluded as it doesn't rely on sensor data.

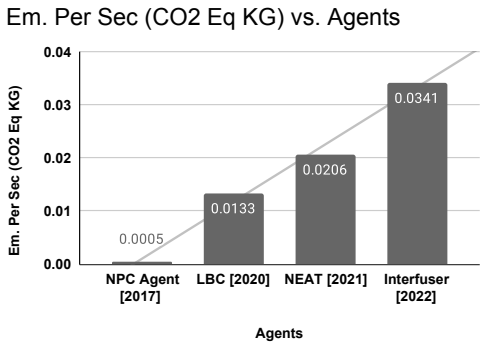

Figure 4: We observe increased emissions as the agents get more robust over the years, with NPC Agent constituting significantly low emissions.

	Robustness against Environmental Disturbances						Robustness against Sensor Faults				
Models	DS (%)	RDS (%)	CO	LO	Wth.	Drift	Cam.	LiD	GNSS	IMU	Spdm
NPC Ag.	26.697	-	-	-	-	-	-	-	-	-	-
LBC	12.545	8.285	0.241	-	0.216	**0.999**	**0.472**	-	**0.999**	0.693	**0.999**
NEAT	19.48	12.291	0.145	-	**0.723**	0.978	0.001	-	0.57	**0.999**	0.999
Interfuser	**54.362**	**34.549**	**0.932**	0.29	0.239	**0.999**	0.239	0.804	0.546	**0.999**	0.668

Table 2: Performances of Previous CARLA Challenge Agents from 2020 to 2022, including CARLA'S NPC Agent. DS: Driving Score, RDS: Robustness Driving Score, AEPS: Average Emission Per Seconds (Kg CO_2 Eq.), AEPR: Average Emissions Per Route (Kg CO_2 Eq.), Cam.: Camera, CO: Camera Occlusion, LO: LiDAR Occlusion, Spdm: Speedometer, ARC: Average Route Completion (only on conditioned routes D_i^j), ASTPR: Average Simulation Time Per Route. Hyphens (-) means sensor is absent.

	Env. Sustainability		Simulation Details	
Models	AEPS	AEPR	ARC(%)	ASTPR(s)
NPC Ag.	**0.0005**	**0.090**	38.139	**189.300**
LBC	0.0133	4.636	28.222	336.085
NEAT	0.0206	7.468	32.932	356.936
Interfuser	0.0341	12.418	**52.188**	367.017

Table 3: Performances Analysis Contd. AEPS: Average Emission Per Seconds (Kg CO_2 Eq.), AEPR: Average Emissions Per Route (Kg CO_2 Eq.), ARC: Average Route Completion (only on conditioned routes D_i^j), ASTPR: Average Simulation Time Per Route.

score is as well low. This information tells us that irrespective of the impressive robustness scores for many of the sensors, LBC doesn't seem to be a safe and performant agent.

Carbon Emissions

It is challenging to assess how much emission a vehicle in its entirety contributes to the environment. However, we can estimate how much emission the underlying AI model that controls navigation constitutes. Hence, we estimated the amount of CO_2 emitted when the AI model is run over time. For fair comparisons, we estimated average emissions per second (AEPS). From Fig. 4 and Table 3, we see that the default CARLA NPC Agent that uses the classical navigation algorithms (non-machine learning) constitutes the lowest CO_2 emission. This agent doesn't utilise sensors, it rather uses ground truth information for planning. Hence, no robustness scores were assigned. The agent was useful to benchmark CO_2 emissions. As AD agents get more sophisticated (with increased robustness), the amount of CO_2 they emit increases.

Discussion and Limitations

We have drawn attention to the need for accessible frameworks with RAI indicators, starting with robustness. These frameworks serve as safety guardrails for AD agent development. We took a socio-technical approach in this work by first establishing the importance and the need for robustness indicators and then proposed a technical framework that of-

fers metrics for robustness and environmental sustainability assessment (using CO_2 emission as proxy). Our framework (S-RAF) focused on streamlining AD agents development process rather than the purpose for which the agent is developed. While the Responsible Research and Innovation (RRI) frameworks (e.g., AREA framework (Jirotka et al. 2017) underscore the importance of the purpose inputted into a machine, assessing such purposes is nearly intractable. S-RAF thus focuses on enhancing development processes to achieve safe and responsible driving agents. S-RAF improves over the conventional AI agents assessment approach that focuses on improving prediction accuracy with limited examples of critical edge cases by quantifying different aspects of robustness and environmental sustainability.

S-RAF's robustness metric takes into account core safety critical sensors in an AV and questions the capability of the agent when an individual or a combination of these sensors malfunctions or when their sensing range is limited due to environmental factors. These cases are hard to obtain in the real world. But with S-RAF, we can simulate these cases affordably. Our results indicate an increasing trend in robustness and safe driving capability over the years. With the breakthrough in generative AI, where generalisable models are being developed, we believe that AD agents would witness a tremendous increase in capabilities. However, we advocate RAI principles and the use of S-RAF in the process.

The emission of green gases by vehicles have impacts on human safety (Ogur and Kariuki 2014). With S-RAF, AI developers have the option to discard models that emit excessive amounts of CO_2. The limitation with S-RAF regarding sustainability is that it does not factor in the emissions caused during the manufacturing process of hardware components, e.g., electric vehicle batteries (Ji et al. 2012) which are notable for constituting CO_2 emissions at manufacturing time. We saw that CO_2 emissions increase as the agents get more performant. No surprises as this adheres to the scaling law (the larger the better). Should the development of extremely large models be stopped? This is an ongoing debate. But again, we encourage developers to consider RAI principles and also optimise for low S-RAF CO_2 emissions.

We have only tested S-RAF on synthetic driving data from CARLA, this is one limitation of this work. Also, when S-RAF assesses agents, it indirectly emits CO_2 as it runs the agents. However, this run is only for a limited time compared to when the models are already deployed and run for an extended period. Thus, S-RAF's use is justifiable as it helps to prevent the deployment of unsafe models. Nuances exist between robustness and resilience in AI, we shall draw the connections in future work. Another limitation of this work is that we have only tested on a handful of AD agents due to the scarcity of open-source AD agents. Hence, we have created a challenge website[1] that encourages developers to submit their agents for assessment and receive feedback for improvements. This challenge would yield more examples from which we can further validate S-RAF while respecting the privacy and copyright agreements of the agent/model owners. That said, we encourage developers and corporate entities to embrace all transparency dimensions highlighted in this paper (process-based, method-based, and output-based) to facilitate easy assessment and potentially, increased safety.

[1]https://carla-rai-challenge.github.io/

Conclusion

We have argued for the need for an accessible framework for assessing AD agents. We developed S-RAF, a framework aimed at guiding developers in building responsible AD agents that are robust and environmentally friendly. S-RAF can equally support AV regulators and other authorised stakeholders in assessing AD agents. Through an experiment, we showed how indicators composed in S-RAF were developed, and we tested these indicators on benchmark AD agents in CARLA simulator. We saw improvements in robustness over time (from 2020 to 2022), while the amount of CO_2 emissions has increased as the models got sophisticated over the years. Lastly, we discussed the implications of these findings and future research agenda.

Acknowledgements

This work was supported by the EPSRC RAILS project (grant reference: EP/W011344/1), Amazon Web Services (AWS), the Embodied AI Foundation, and EvalAI.

References

Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; and Wallach, H. 2018. A reductions approach to fair classification. In *International conference on machine learning*, 60–69. PMLR.

Allen, D.; and Weyl, E. G. 2024. The Real Dangers of Generative AI. *Journal of Democracy*, 35(1): 147–162.

Bellamy, R. K.; Dey, K.; Hind, M.; Hoffman, S. C.; Houde, S.; Kannan, K.; Lohia, P.; Martino, J.; Mehta, S.; Mojsilović, A.; et al. 2019. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5): 4–1.

Berghoff, C.; Bielik, P.; Neu, M.; Tsankov, P.; and Von Twickel, A. 2021. Robustness testing of AI systems: a case study for traffic sign recognition. In *Artificial Intelligence Applications and Innovations: 17th IFIP WG 12.5 International Conference, AIAI 2021, Hersonissos, Crete, Greece, June 25–27, 2021, Proceedings 17*, 256–267. Springer.

Berman, G.; Goyal, N.; and Madaio, M. 2024. A Scoping Study of Evaluation Practices for Responsible AI Tools: Steps Towards Effectiveness Evaluations. *arXiv preprint arXiv:2401.17486*.

Bowles, C.; Chen, L.; Guerrero, R.; Bentley, P.; Gunn, R.; Hammers, A.; Dickie, D. A.; Hernández, M. V.; Wardlaw, J.; and Rueckert, D. 2018. Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*.

Budennyy, S. A.; Lazarev, V. D.; Zakharenko, N. N.; Korovin, A. N.; Plosskaya, O.; Dimitrov, D. V.; Akhripkin, V.; Pavlov, I.; Oseledets, I. V.; Barsola, I. S.; et al. 2022. Eco2ai: carbon emissions tracking of machine learning models as the first step towards sustainable ai. In *Doklady Mathematics*, volume 106, S118–S128. Springer.

Chen, D.; Zhou, B.; Koltun, V.; and Krähenbühl, P. 2020. Learning by cheating. In *Conference on Robot Learning*, 66–75. PMLR.

Chien, A. A.; Lin, L.; Nguyen, H.; Rao, V.; Sharma, T.; and Wijayawardana, R. 2023. Reducing the Carbon Impact of Generative

AI Inference (today and in 2035). In *Proceedings of the 2nd Workshop on Sustainable Computer Systems*, 1–7.

Chitta, K.; Prakash, A.; and Geiger, A. 2021. Neat: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15793–15803.

Coston, A.; Kawakami, A.; Zhu, H.; Holstein, K.; and Heidari, H. 2023. A validity perspective on evaluating the justified use of data-driven decision-making algorithms. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 690–704. IEEE.

Courty, B.; Schmidt, V.; Luccioni, S.; Goyal-Kamal; MarionCoutarel; Feld, B.; Lecourt, J.; LiamConnell; Saboni, A.; Inimaz; supatomic; Léval, M.; Blanche, L.; Cruveiller, A.; ouminasara; Zhao, F.; Joshi, A.; Bogroff, A.; de Lavoreille, H.; Laskaris, N.; Abati, E.; Blank, D.; Wang, Z.; Catovic, A.; Alencon, M.; Stéchĺy, M.; Bauer, C.; de Araújo, L. O. N.; JPW; and MinervaBooks. 2024. mlco2/codecarbon: v2.4.1.

Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2020. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*.

Dignum, V. 2019. *Responsible artificial intelligence: how to develop and use AI in a responsible way*, volume 2156. Springer.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.

Floridi, L. 2021. Establishing the rules for building trustworthy AI. *Ethics, Governance, and Policies in Artificial Intelligence*, 41–45.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Huang, X.; Kwiatkowska, M.; Wang, S.; and Wu, M. 2017. Safety verification of deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, 3–29. Springer.

Ji, S.; Cherry, C. R.; J. Bechle, M.; Wu, Y.; and Marshall, J. D. 2012. Electric vehicles in China: emissions and health impacts. *Environmental science & technology*, 46(4): 2018–2024.

Jirotka, M.; Grimpe, B.; Stahl, B.; Eden, G.; and Hartswood, M. 2017. Responsible research and innovation in the digital age. *Communications of the ACM*, 60(5): 62–68.

Kokhlikyan, N.; Miglani, V.; Martin, M.; Wang, E.; Alsallakh, B.; Reynolds, J.; Melnikov, A.; Kliushkina, N.; Araya, C.; Yan, S.; et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Krafft, P.; Young, M.; Katell, M.; Lee, J. E.; Narayan, S.; Epstein, M.; Dailey, D.; Herman, B.; Tam, A.; Guetler, V.; et al. 2021. An action-oriented AI policy toolkit for technology audits by community advocates and activists. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 772–781.

Lacoste, A.; Luccioni, A.; Schmidt, V.; and Dandres, T. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Lavrinc, D. 2016. This is how bad self-driving cars suck in rain. Accessed: Jul., 2020.

Lifshitz, L. R.; and McMaster, C. 2020. Legal and Ethics Checklist for AI Systems. *SciTech Lawyer*, 17(1): 28–34.

Marcus, G. 2020. The next decade in AI: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*.

Microsoft; and Contributors. 2019. Fairlearn.

Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I. D.; and Gebru, T. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, 220–229.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

National Highway Traffic Safety Administration. 2007. Pre-Crash Scenario Typology. https://www.nhtsa.gov/sites/nhtsa.gov/files/pre-crash_scenario_typology-final_pdf_version_5-2-07.pdf. Accessed: 16-04-2023.

Nicolae, M.-I.; Sinn, M.; Tran, M. N.; Buesser, B.; Rawat, A.; Wistuba, M.; Zantedeschi, V.; Baracaldo, N.; Chen, B.; Ludwig, H.; et al. 2018. Adversarial Robustness Toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069*.

Ogur, E.; and Kariuki, S. 2014. Effect of car emissions on human health and the environment. *International Journal of Applied Engineering Research*, 9(21): 11121–11128.

Omeiza, D.; Webb, H.; Jirotka, M.; and Kunze, L. 2021. Explanations in autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(8): 10142–10162.

Rauber, J.; Brendel, W.; and Bethge, M. 2017. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*.

Rawls, J. 2020. *A theory of justice: Revised edition*. Harvard university press.

Reed, B. 2024. Who's responsible when an autonomous car crashes? Accessed: Apr. 24, 2024.

Rhem, A. J. 2023. Ethical use of data in AI Applications. In *Ethics-Scientific Research, Ethical Issues, Artificial Intelligence and Education*. IntechOpen.

Shao, H.; Wang, L.; Chen, R.; Li, H.; and Liu, Y. 2023. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, 726–737. PMLR.

Singh, G.; Gehr, T.; Püschel, M.; and Vechev, M. 2019. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL): 1–30.

Stilgoe, J.; Owen, R.; and Macnaghten, P. 2020. Developing a framework for responsible innovation. In *The Ethics of Nanotechnology, Geoengineering, and Clean Energy*, 347–359. Routledge.

Strubell, E.; Ganesh, A.; and McCallum, A. 2019. Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Van Wynsberghe, A. 2021. Sustainable AI: AI for sustainability and the sustainability of AI. *AI and Ethics*, 1(3): 213–218.

wayve. 2023. CARLA Autonomous Driving Challenge. Accessed: Aug. 5, 2024.

Wexler, J.; Pushkarna, M.; Bolukbasi, T.; Wattenberg, M.; Viégas, F.; and Wilson, J. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics*, 26(1): 56–65.

Zhang, C.; Liu, A.; Liu, X.; Xu, Y.; Yu, H.; Ma, Y.; and Li, T. 2020. Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity. *IEEE Transactions on Image Processing*, 30: 1291–1304.