

AD²: Analysis and Detection of Adversarial Threats in Visual Perception for End-to-End Autonomous Driving Systems

Supplementary Material

Ishan Sahu^{1,2}, Somnath Hazra¹, Somak Aditya¹, Soumyajit Dey¹

¹ Indian Institute of Technology Kharagpur ² TCS Research, India

{ishan.sahu@kgpian, somnathhazra@kgpian, saditya@cse, soumya@cse}.iitkgp.ac.in

A. Background

Perception Systems in Autonomous Driving

An AD system rely on the information provided by on-board sensors, which allow to describe the state of the vehicle, its environment and other actors [15]. Perception systems process data from these sensors individually or through data fusion. They may include classical sensor data processing algorithms, or machine learning / deep learning models trained for specific perception tasks. Such systems have the goal to discern both static objects (road and lane markings, road signs, traffic lights, etc.) and dynamic objects (vehicles, pedestrians, etc.) in the environment. It is also responsible for its own localization (position, linear and angular velocities, acceleration, orientation).

Cameras are the most common sensors and are available in the market in a wide range of configurations in resolution, frame rate, sensor size, and optics parameters. They are low cost and provide a range of information including spatial, dynamic, and semantic. They are affected by light and weather conditions.

RADAR (Radio detection and ranging) works using high frequency electromagnetic waves and their reflection from different objects. Its performance is independent of light and weather conditions [15]. However, it is affected by reflectivity of different materials. Metals amplify radar signals, whereas other materials like wood are virtually transparent.

LIDAR (Light Detection and Ranging) is an active ranging technology that calculates distance to objects by measuring round trip time of a laser pulse [15]. They are useful in creating a highly accurate digital maps using 3D point clouds. However, they suffer from several drawbacks: low vertical resolution, sparse measurements with gaps between layers, poor detection of dark and specular objects, and are affected by weather conditions.

IMU (Inertial Measurement Unit) consists of accelerometer to measure linear acceleration, gyroscope to measure

orientation and angular velocity, and sometimes a magnetometer for heading reference. They are affected by magnetic disturbances and time-variant sensor biases and measurement noise [19].

Speedometer is used to measure the instantaneous speed of the vehicle.

GNSS (Global Navigation Satellite System) refers to any satellite constellation that provides global positioning, navigation, and timing services [9]. They suffer from atmospheric interference, and availability limitations [13].

Autonomous Driving Agents in CARLA

Several autonomous driving agents have been developed and proven on CARLA simulator [8] as part of CARLA Autonomous Driving Leaderboard 1.0 used in CARLA AD challenges. This leaderboard platform evaluates the driving proficiency of autonomous agents in realistic traffic situations.

All driving agents meant for the leaderboard take sensor data as input and provide control signals such as steer, throttle, brake, handbrake (optional) as output. Block diagram representation of a generic AD agent is shown in Figure 1. Internally the driving agent can have different architectures and internal input/output from perception to planning to controller submodules. Agents can have varying



Figure 1. Simplified autonomous driving agent showing expected inputs and outputs.

sensor stack within the allowed limits. There are individual maximum limits on different sensors – RGB (Red green blue) camera: 4, LIDAR: 1, RADAR: 2, GNSS: 1, IMU: 1, Speedometer: 1. The position of these sensors on the ego vehicle can also vary.

We briefly discuss two top ranked end-to-end autonomous driving agent that have been proven in the leader-

board 1.0 benchmark. They will be the focus of our adversarial attack evaluation study.

Transfuser [5] uses multi-modal fusion transformer on image and LIDAR inputs to incorporate global context and pairwise interactions into the feature extraction layers. Using several transformer modules fusion of RGB images and LIDAR representations are performed to yield a 512 dimensional feature vector output. This feature vector constitutes a compact representation of the environment that encodes the global context of the scene. This is then passed to GRU based waypoint prediction network that predicts the differential waypoints of the ego vehicle. These waypoints are then used by the controller to generate steer, throttle and brake values. Figure 2 shows a simplified block diagram of their architecture.

Interfuser [20] is an interpretable sensor fusion transformer, in which information from multi-modal multi-view sensors is fused, which also provides intermediate interpretable features. Transformer encoder is used to fuse tokens from different sensors. Then, three types of queries are made to the transformer decoder: waypoint queries, density map queries, and traffic rule query. These outputs are then provided to three corresponding prediction headers to predict waypoints, object density map, and traffic rule respectively. In the end a safety controller is applied to determine steer, throttle and brake commands. Safety controller was designed by the authors to address safety concerns in complex traffic situations. This architecture is depicted in Figure 3.

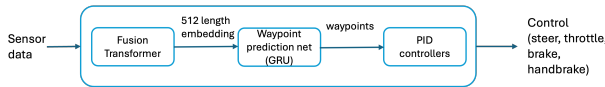


Figure 2. Simplified block diagram for Transfuser agent.

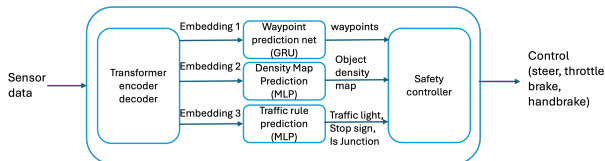


Figure 3. Simplified block diagram for Interfuser agent.

Evaluation of Autonomous Driving Agents

Performance of autonomous driving agents and its subsystems can be carried out in real world or through simulation [4, 12, 21]. In case of real world evaluation, AD agent is deployed in a real vehicle and its capabilities are tested on real physical roads with real obstacles and driving scenarios. However, this is expensive and may not be feasible for ensuring exhaustive coverage of scenarios. Simulation allows validation on almost infinite variability of possible

scenarios that could be encountered [6]. Simulation based evaluation can be carried out in different ways as described below:

- Open loop (or offline) evaluation: In open loop evaluation, expert trajectories are recorded in datasets with the corresponding sensor data. The agent under evaluation is then used to predict trajectories with the same sensor information which is compared with the expert ground truth [1].
 1. Model level evaluation: Autonomous driving system may be composed of different subsystems and models. Each model is treated as an independent unit of computation, and it is fed with input data meant for it. The model predicts values that are compared to its specific ground truth labels, which serve as an oracle. For example, testing of object detection model in modular AD systems.
 2. System level evaluation: Overall system is tested on the recorded dataset with expert trajectory as the ground truth. Thus, the individual model's prediction errors become less meaningful. As such, failing tests are characterized in terms of the misbehavior of the whole system in response to the AD's predictions.
- Closed loop (or online) evaluation: Open loop evaluations are insufficient [12]. In closed loop evaluation, the driving agent is placed in a dynamic simulation environment. Its predictions and decisions have an immediate effect on the overall system behavior.

In our work, we focus on comparison of closed loop performance of the state-of-the-art end-to-end AD systems under adversarial attack with respect to their performance under normal conditions.

Related Work

We discuss recent work that are closely related to our subproblem of adversarial attack evaluation on complete AD systems. Table 1 compares our work with such literature.

Few attacks and defense methods on regression-based driving models are analyzed in [7]. The driving models considered here are simpler. They are trained and evaluated offline only on images dataset collected using a front camera. Attack goal considered in the paper is also restrictive: attack is considered successful, if the predicted steer angle deviation for a given input frame is greater than specified adversarial threshold, without considering if that actually leads to traffic violations.

Adversarial attacks on end-to-end AD systems are relatively new and there are limited studies on this topic. In [23], authors devised white box attacks to deviate the vehicle outside the lane by perturbing the input image. They have shown that these attacks are more effective than random noise attacks. However, the study was conducted on a simpler end-to-end driving model proposed by the au-

Paper	Type of AD System	Sensor configuration of AD System	Sensor under attack	Attack Type	Evaluation Type	Attack strategy	Evaluation Metrics
[7]	Simple regression driving models	Camera only	Camera	White-box and black box	Open loop on recorded dataset	Each camera image in the dataset considered individually	Steering angle deviation
[23]	End-to-end	Camera only	Camera	White-box	Closed-loop without traffic	Online attack at each timestep	Steering angle deviation
[22]	End-to-end	Camera only	Camera	White-box	Open loop on recorded dataset	Attack optimized for each batch of dataset	L2 error of planned trajectory, collision rate
Our work	End-to-end SOTA [5, 20]	Multimodal	Camera	Restricted black box	Closed loop with traffic	Online continuous attack at each timestep, intermittent timesteps	Driving performance that includes traffic infractions

Table 1. Comparison of our adversarial evaluation with those in existing literature. Here we only include work which actually adversarially attack a complete autonomous driving system and analyze the impact.

thors themselves which only had camera sensor for perception. Simulations were conducted using Udacity and Gazebo simulator. The adversary also had complete white box access to the AD system. The effectiveness of their attacks on state-of-the-art agents is unknown. Authors in [22] propose an attack scheme for end-to-end autonomous driving model through module wise noise injection. They assume access to different sub-modules for noise injection as well as for optimization of noise using gradient based methods, that is, complete white box access.

For completeness and to clarify differentiation, we also briefly mention ANTI-CARLA [17]. This was proposed as an automated testing framework in CARLA for simulating weather conditions (e.g. heavy rain) and sensor faults (e.g. camera occlusion) with the goal of finding driving scenarios that may lead to failure of the system. This work, however, focuses on generating test cases given some conditions on the environment. It does not deal with adversarial attacks.

B. Attack Evaluation on End-to-End AD Systems

Due to the space constraints in the main paper, we provide additional details on our results here. We first perform the evaluation under normal conditions without any adversarial attack on the AD system and note the baseline performance by monitoring the ego vehicle. Subsequently, we evaluate the effect of poltergeist [11], SNAL [3], and ESIA [14]. The time taken by the ego vehicle to complete the same route may differ in different simulation runs. Even a slight change in predicted control action of the AD system at any one timestep may cause its own future course and be-

haviour to change as it also influences the behaviour of surrounding dynamic elements in the environment. Also, there is some randomness involved in simulation execution [2]. We, therefore, focus on macro level indicators of driving performance such as traffic violations and route completion percentage to compute driving score as discussed before.

In Table 2 of the main paper, we present the degradation in performance of Transfuser and Interfuser AD agents under Poltergeist attack as compared to their performance without any attacks. The first row for each agent with ‘None’ adversarial attack is its baseline performance when there is no attack during the entire driving task. Then we have rows corresponding to its driving evaluation under poltergeist attack with different attack intervals d . For each case, we measure driving score DS , infraction penalty P , route completion R , and mean \pm standard deviation of $|L_{dev}|$. In an ideal case, the AD system would score 100 in DS , 1.0 in P , and 100 in R . L_{dev} , the deviation from lane centre should also be small which would indicate that the vehicle kept to its lane and did not drift out of it. In the table, we give mean \pm standard deviation of absolute value $|L_{dev}|$ for the entire driving period. The arrows next to the quantities indicate their respective desired direction of change from attacker’s perspective. The attacker would want to prevent the AD vehicle from completing its route (reduce route completion) and increase the number of traffic infractions (decrease in infraction penalty), which would consequently result in lower driving score. Similarly, Table 3 and Table 4 in the main paper give the results for corresponding SNAL attack, and ESIA evaluation experiments.

AD system	Attack	Attack Parameters	Infraction Details							
		Attack interval d	Route Completion Test (\downarrow)	Outside Route Lanes Test (\uparrow)	Collision Test (\uparrow)	Running Red Light Test (\uparrow)	Running Stop Sign Test (\uparrow)	In Route Test	Agent Blocked Test	Timeout
Transfuser	None	None	100%	0%	0	0	0	S	S	S
	Poltergeist	1	64.83%	3.16%	11	3	0	S	F	S
		4	100%	0%	3	0	0	S	S	S
Interfuser	Poltergeist	11	100%	0%	2	0	0	S	S	S
		None	74.24%	0%	0	0	0	S	S	F
		1	9.17%	45.41%	2	1	0	S	F	S
		4	74.66%	0%	1	1	0	S	S	F
		11	100%	0%	0	1	0	S	S	S

Table 2. Infraction details of the driving agents under poltergeist attack. d is the interval in which attack is carried out. If the first attack was at timestep t , then the next attack would be at timestep $t + d$ and so on. S: Success, F: Failure. For reference, ideally route completion should be 100%, outside route lanes test should be 0%, different infractions should be 0, and the agent should succeed in the remaining tests. Arrows next to the quantities indicate their respective desired direction of change from attacker’s perspective. Attack ‘None’ implies baseline.

AD system	Attack	Attack Parameters		Infraction Details							
		$\epsilon (l_\infty)$	Attack interval (d)	Route Completion Test (\downarrow)	Outside Route Lanes Test (\uparrow)	Collision Test (\uparrow)	Running Red Light Test (\uparrow)	Running Stop Sign Test (\uparrow)	In Route Test	Agent Blocked Test	Timeout
Transfuser	None	None	None	100%	0%	0	0	0	S	S	S
	SNAL	4	1	100%	0%	3	0	0	S	S	S
		4	4	38.04%	1.64%	7	0	0	S	F	S
		4	11	100%	0%	1	0	0	S	S	S
		8	1	100%	0%	5	1	0	S	S	S
		8	4	100%	0%	3	0	0	S	S	S
		8	11	100%	0%	1	0	0	S	S	S
Interfuser	None	None	None	74.24%	0%	0	0	0	S	S	F
	SNAL	4	1	100%	0%	0	1	0	S	S	S
		4	4	100%	0%	0	1	0	S	S	S
		4	11	100%	0%	1	1	0	S	S	S
		8	1	100%	0%	1	0	0	S	S	S
		8	4	100%	0%	1	1	0	S	S	S
		8	11	100%	0%	1	1	0	S	S	S

Table 3. Infraction details of the driving agents under SNAL attack. ϵ is the maximum perturbation that the attacker can introduce. d is the interval in which attack is carried out. If the first attack was at timestep t , then the next attack would be at timestep $t + d$ and so on. S: Success, F: Failure. For reference, ideally route completion test should be 100%, outside route lanes test should be 0%, different infractions should be 0, and the agent should succeed in the remaining tests. Arrows next to the quantities indicate their respective desired direction of change from attacker’s perspective. Attack ‘None’ implies baseline.

Infraction Details of Adversarial Evaluations

Table 2 in this supplementary material provides the details of traffic infractions recorded in each setting along with out-

come of few tests. Route completion test gives the percentage of route that that agent was able to complete. The values in outside route lanes test signifies the percentage of the

AD system	Attack	Attack Parameters		Infraction Details							
		Severity	Attack interval (d)	Route Completion Test (↓)	Outside Route Lanes Test (↑)	Collision Test (↑)	Running Red Light Test (↑)	Running Stop Sign Test (↑)	In Route Test	Agent Blocked Test	Timeout
Transfuser	None	None	None	100%	0%	0	0	0	S	S	S
	ESIA	low	1	100%	0%	4	0	0	S	S	S
		low	4	100%	0%	4	0	0	S	S	S
		low	11	100%	0%	1	0	0	S	S	S
		med	1	69.9%	0%	0	0	0	S	S	F
		med	4	100%	0%	0	0	0	S	S	S
		med	11	100%	0%	3	0	0	S	S	S
		high	1	0.26%	0%	0	0	0	S	F	S
		high	4	100%	0%	0	0	0	S	S	S
		high	11	100%	0%	3	0	0	S	S	S
Interfuser	None	None	None	74.24%	0%	0	0	0	S	S	F
	ESIA	low	1	100%	0%	0	1	0	S	S	S
		low	4	100%	0%	0	0	0	S	S	S
		low	11	100%	0%	1	0	0	S	S	S
		med	1	100%	0%	0	0	0	S	S	S
		med	4	100%	0%	1	1	0	S	S	S
		med	11	100%	0%	0	1	0	S	S	S
		high	1	100%	0%	1	0	0	S	S	S
		high	4	100%	0%	3	1	0	S	S	S
		high	11	100%	0%	0	0	0	S	S	S

Table 4. Infraction details of the driving agents under ESIA *ad* is the interval in which attack is carried out. If the first attack was at timestep t , then the next attack would be at timestep $t + d$ and so on. S: Success, F: Failure. For reference, ideally route completion test should be 100%, outside route lanes test should be 0%, different infractions should be 0, and the agent should succeed in the remaining tests. Arrows next to the quantities indicate their respective desired direction of change from attacker’s perspective. Attack ‘None’ implies baseline.

agent’s total driving that was outside its supposed lane in the route. The numbers in collision test, running red light test, and running stop sign test gives the count of each of the respective traffic violations that have occurred. Failure in in-route test implies that the vehicle went off-route by more than 30 m. If vehicle controlled by AD system gets blocked by any static or dynamic object for more than 180 s, then it fails the agent blocked test. Timeout failure happens when the system takes too long to complete the route.

Similarly, Table 3 and Table 4 give the results for corresponding SNAL attack and ESIA evaluation experiments.

C. Adversarial Attack Detection Baselines

Adversarially perturbing images introduces artifacts that lead to their detection. In order to study the complexity of the detection task we also experiment with several baseline methods. This also provides a motivation for our approach. **Focus measure operators** [16]. They are used in the computation of the focus level for every pixel of an image and is the main step in traditional shape-from-focus techniques for recovering 3D shapes [16]. A variety of algorithms have

been proposed in the literature. They use different working principles: gradients, laplacians, wavelet transforms, image statistics, discrete cosine transforms, etc. Variance of Laplacian (LAP4) is one of such measure which is defined for point (i, j) in image I as

$$\phi_{i,j} = \sum_{(i,j) \in \Omega(x,y)} (\Delta I(i,j) - \bar{\Delta I})^2 \quad (1)$$

where ΔI is the image Laplacian obtained by convolving I with the Laplacian mask, $\bar{\Delta I}$ is the mean value of the image Laplacian within neighbourhood $\Omega(x, y)$.

In the experiments, laplacian based operators have shown best overall performance at normal imaging conditions but are most sensitive to noise [16]. They are proven to be very useful in detecting blurs in images.

Kernel PCA (KPCA) [10]. Kernel Principal Component Analysis employs non-linear kernels to enhance separability between in-distribution data and out-of-distribution data. This has been proposed to overcome the failure of PCA on features obtained using deep neural networks (DNNs). Given, feature representation \mathbf{z} from a DNN, two different

feature mappings have been proposed.

- Cosine Mapping followed by PCA (CoP): $\phi_{\cos}(\mathbf{z})$
- Cosine and RFFs (Random Fourier Features) Mapping followed by PCA (CoRP): $\phi_{RFF}(\phi_{\cos}(\mathbf{z}))$

where

$$\phi_{\cos}(\mathbf{z}) = \frac{\mathbf{z}}{\|\mathbf{z}\|} \quad (2)$$

$$\phi_{RFF}(\mathbf{z}) = \sqrt{\frac{2}{M}} [\phi_1(\mathbf{z}), \dots, \phi_1(\mathbf{z})], \phi_i(\mathbf{z}) = \cos(\mathbf{z}^T \omega_i + u_i) \quad (3)$$

Here, ω_i are i.i.d sampled from Fourier transform using kernel k , and u_i are i.i.d sampled from Uniform distribution $\mathcal{U}(0, 2\pi)$. Then, PCA is executed on mapped features for computing the non-linear principal components and corresponding reconstruction errors. Reconstruction errors have been shown to be very useful in detecting out-of-distribution data.

CyberDet [18]. CyberDet uses kth order differences computed from the input image fed to a ResNet based binary classifier to discriminate between attacked and genuine images. The method is claimed to be agnostic of the attack method, and of the data used for train or inference. It has been shown effective against few white-box attacks on the perception system of an autonomous robot.

D. Potential Negative Societal Impacts

Like any defense or detection based methods, public knowledge of detection methods may usher in both positive and negative impacts. While, many may benefit from it, potential attackers may exploit that itself further to come up with better attack. Our goal is that we have made it slightly more non-trivial to come up with easy attacks to cause failures in the AD systems.

References

- [1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020.
- [2] Greg Chance, Abanoub Ghobrial, Kevin McAreavey, Séverin Lemaignan, Tony Pipe, and Kerstin Eder. On determinism of game engines used for simulation-based autonomous vehicle verification. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20538–20552, 2022. Publisher: IEEE.
- [3] Erh-Chung Chen, Pin-Yu Chen, I Chung, Che-Rung Lee, and others. Steal now and attack later: Evaluating robustness of object detection against black-box adversarial attacks. *arXiv preprint arXiv:2404.15881*, 2024.
- [4] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2024.
- [5] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023.
- [6] Iván García Daza, Rubén Izquierdo, Luis Miguel Martínez, Ola Benderius, and David Fernández Llorca. Sim-to-real transfer and reality gap modeling in model predictive control for autonomous driving. *Applied Intelligence*, 53(10):12719–12735, 2023.
- [7] Yao Deng, Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–10, 2020.
- [8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st annual conference on robot learning*, pages 1–16, 2017.
- [9] EUSPA. What is GNSS | EU Agency for the Space Programme, 2025.
- [10] Kun Fang, Qinghua Tao, Kexin Lv, Mingzhen He, Xiaolin Huang, and JIE YANG. Kernel PCA for out-of-distribution detection. In *The thirty-eighth annual conference on neural information processing systems*, 2024.
- [11] Xiaoyu Ji, Yushi Cheng, Yuepeng Zhang, Kai Wang, Chen Yan, Wenyuan Xu, and Kevin Fu. Poltergeist: Acoustic adversarial machine learning against cameras and computer vision. In *2021 IEEE symposium on security and privacy (SP)*, pages 160–175, 2021.
- [12] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
- [13] Niels Joubert, Tyler GR Reid, and Fergus Noble. Developments in modern gnss and its impact on autonomous vehicle architectures. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 2029–2036. IEEE, 2020.
- [14] Wenhao Liao, Sineng Yan, Youqian Zhang, Xinwei Zhai, Yuanyuan Wang, and Eugene Fu. Is your autonomous vehicle safe? understanding the threat of electromagnetic signal injection attacks on traffic scene perception. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 27464–27472, 2025.
- [15] Enrique Marti, Miguel Angel de Miguel, Fernando Garcia, and Joshue Perez. A review of sensor technologies for perception in automated driving. *IEEE Intelligent Transportation Systems Magazine*, 11(4):94–108, 2019.
- [16] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432, 2013.
- [17] Shreyas Ramakrishna, Baiting Luo, Christopher B. Kuhn, Gabor Karsai, and Abhishek Dubey. ANTI-CARLA: An adversarial testing framework for autonomous vehicles in

CARLA. In *2022 IEEE 25th international conference on intelligent transportation systems (ITSC)*, pages 2620–2627. IEEE Press, 2022. Place: Macau, China Number of pages: 8.

- [18] Lucian M. Sasu and Sorin M. Grigorescu. Cyberdet: Real-time adversarial attacks detection for autonomous robots and self-driving cars. In *2025 IEEE Intelligent Vehicles Symposium (IV)*, pages 1935–1941, 2025.
- [19] Thomas Seel, Manon Kok, and Ryan S. McGinnis. Inertial sensors—applications and challenges in a nutshell. *Sensors*, 20(21), 2020.
- [20] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Proceedings of the 6th conference on robot learning*, pages 726–737. PMLR, 2023.
- [21] Andrea Stocco, Brian Pulfer, and Paolo Tonella. Model vs system level testing of autonomous driving systems: a replication and extension study. *Empirical Software Engineering*, 28(3):73, 2023.
- [22] Lu Wang, Tianyuan Zhang, Yikai Han, Muyang Fang, Ting Jin, and Jiaqi Kang. Attack end-to-end autonomous driving through module-wise noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops*, pages 8349–8352, 2024.
- [23] Han Wu, Syed Yunas, Sareh Rowlands, Wenjie Ruan, and Johan Wahlström. Adversarial driving: Attacking end-to-end autonomous driving. In *2023 IEEE intelligent vehicles symposium (IV)*, pages 1–7, 2023.