$PASS$: A System-Driven Evaluation Platform for Autonomous Driving Safety and Security

Zhisheng Hu Baidu Security
Junjie Shen UC Irvine
Shengjian Guo Baidu Security
Xinyang Zhang Baidu Security
Zhenyu Zhong Baidu Security
Qi Alfred Chen UC Irvine
Kang Li Baidu Security

***Abstract*—Safety and security play critical roles for the success of Autonomous Driving (AD) systems. Since AD systems heavily rely on AI components, the safety and security research of such components has also received great attention in recent years. While it is widely recognized that AI component-level (mis)behavior does not necessarily lead to AD system-level impacts, most of existing work still only adopts component-level evaluation. To fill such critical scientific methodology-level gap from component-level to real system-level impact, a system-driven evaluation platform jointly constructed by the community could be the solution. In this paper, we present $PASS$ (Platform for Auto-driving Safety and Security), a system-driven evaluation prototype based on simulation. By sharing our platform building concept and preliminary efforts, we hope to call on the community to build a uniform and extensible platform to make AI safety and security work sufficiently meaningful at the system level.**

I. Introduction

Safety and security play vital roles in the fact that Autonomous Driving (AD) vehicles become a reality in our daily lives. Since AD systems are usually designed with a collection of AI models to handle the core decision-making process such as perception, localization, prediction, and planning, safety and security research in such components' contexts receives exponentially increasing attention especially in recent years. The research works reveal that today's AI models are generally vulnerable to adversarial attacks [15, 38].

Since AI models are only components of the entire AD system, it is also widely recognized that AI component-level (mis)behavior does not necessarily lead to AD system-level effect [11, 22, 33, 35], e.g., when the misdetected object is at a far distance for automatic emergency braking [11, 35], or the misdetection can be tolerated by subsequent AI modules like object tracking [22]. However, we find that at this point system-level evaluation is generally lacking in existing AD AI security/safety works: We performed a survey of existing such works that aimed at creating system-level impacts on AD systems in most recent 5 years, but found that the vast majority only adopted component-level evaluation (e.g., analyzing model accuracy without involving any interactions/integration with other AI components in AD systems). The gap from component-level to system-level effect may lead to meaningless attack/defense progress of AD safety and security research as pointed out by prior works [11, 22, 35]. To bridge the gap, a common system-driven evaluation infrastructure built jointly by the community could be an effective and sustainable solution direction.

In this paper, we thus propose $PASS$ (Platform for Auto-driving Safety and Security), a simulation-based system-driven evaluation prototype, to bridge the gap. $PASS$ serves as an easy-to-use and fair system-driven evaluation platform to various AI safety and security works in the AD context. $PASS$ is built in a modular design in representative AD systems, standardized attack/defense implementation interfaces, system-level evaluation scenarios and metrics. With the modular design, we expect existing works to be more easily reproduced, and new attacks/defenses, AD systems, and scenarios can be collectively developed by researchers to fit future needs.

We have implemented a preliminary prototype of $PASS$, and share in this paper our previous attempts on using it for system-driven safety and security evaluation in AD context (in the form of AD Capture The Flag (CTF) competitions). We will also provide a list of future direction to improve the evaluation platform. We will fully open-source the current prototype and welcome community contributions of new attack/defense interfaces and implementations. In summary, this work makes the following contributions: (1). To address the critical gap between component-level and system-level evaluations, we take the initiative to develop an open-source, uniform, and extensible system-driven evaluation platform for the AD AI safety and security research community. (2). We have implemented a preliminary prototype of $PASS$ and used it to organize two international AD CTF competitions in recent two years to raise the awareness of AD safety and security in the research community. Over 100 teams across the globe have participated in the competitions.

II. Background and Design Rationale

A. Background

1) AI Components in AD Systems: A typical industry-grade AD system relies on multiple AI components (e.g., localization, perception, and planning shown in Fig. 1) to make logical, safe, and correct driving decisions.

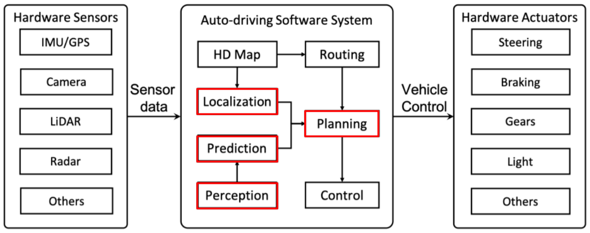


Fig. 1. Typical AI components in an AD system (boxes with red border).

Workshop on Automotive and Autonomous Vehicle Security (AutoSec) 2022
24 April 2022, San Diego, CA, USA
ISBN 1-891562-75-4
https://dx.doi.org/10.14722/autosec.2022.23018
www.ndss-symposium.org

Localization refers to integrating multiple data from the hardware sensors such as camera, LiDAR, radar, etc. to locate the position of the AD vehicle in an environment. *Perception* also takes multiple sensor data to finish the tasks of perceiving the surrounding environment such as object detection and tracking, traffic sign and signal classification, and lane line detection. *Prediction* is responsible for estimating the future status of surrounding objects, including their position, orientation and speed. *Planning* generates an overall route navigation decisions such as cruising, lane changing, stopping without violating any traffic rule. Then, the *control* component transmits the specific driving decision to the hardware executor, so that the vehicle will follow the planned route. We suggest readers to refer to recent surveys [32, 46] for more details.

2) Recent AD AI safety/security works and the scientific gap regarding system-level evaluation: As AI components are playing an increasingly important role in AD systems, relevant safety and security research works have also received attention. Recent works have shown that even in the physical world, AI components are often vulnerable to *physical adversarial perturbations* [5–7, 12, 24]. However, since the targeted AI components are only a subset of the entire AD system, it is widely recognized that such AI component-level misbehavior does not necessarily lead to AD system-level effect [11, 22, 33, 35]. However, we found that system-level evaluation is generally lacking in existing works. Specifically, one of our ongoing works analyzes recent AI security and safety works [36] aimed at creating system-level impact on AD systems in recent 5 years published in commonly-recognized top-tier venues [2] in closely-related fields to AD AI (i.e., security, Computer Vision (CV), Machine Learning (ML), AI, and robotics), as well as a few well-known works published in arXiv and other venues based on our best knowledge. Particularly, for the top-tier venues, we *exhaustively* search over the paper lists from 2017 to 2021 to find the ones that fall into our scope above. Due to the page limit, we only show the preliminary results in the most representative subset of them that attack camera object detection component, which is the most essential and extensively-studied AI component in AD systems, in Table I.

In particular, among these 23 camera object detection attack works, the vast majority (86.96%) *only adopt component-level evaluation* (e.g., analyzing model accuracy without involving any interactions/integration with other AI components in AD systems). However, in the CPS area, it is actually already widely-recognized that for CPS with AI components, *AI component-level errors do not necessarily lead to system-level effects* [11, 35]. For AD systems, this is especially true due to the high end-to-end system-level complexity and closed-loop control dynamics, which can explicitly or implicitly create fault-tolerant effects for AI component-level errors. In fact, various such counterexamples have already been discovered in AD system context, e.g., when the object detection model error is at a far distance for automatic emergency braking systems [11, 35], or such errors can be effectively tolerated by downstream AI modules such as object tracking [22]. This means that even with high attack success rates shown at the AI component level, it is actually possible that such an attack *cannot cause any meaningful effect to the AD vehicle driving behavior*. For example, as concretely estimated by Jia et al. [22], for camera object detection-only AI attacks (e.g., those in Table I), a component-level success rate of up to 98% can still be not enough to affect object tracking results. Thus, we believe that such current general lack of system-level evaluation is a critical *scientific methodology-level gap* that should be addressed as soon as possible.

Paper	Targeted Component	Eval. level	
		Component-level	System-level
Lu et al. [27]	object detection	✓	
Eykholt et al. [12]	object detection	✓	
Chen et al. [7]	object detection	✓	
Zhao et al. [48]	object detection	✓	
Xiao et al. [44]	object detection	✓	
Zhang et al. [47]	object detection	✓	
Nassi et al. [30]	object detection	✓	✓
Man et al. [28]	object detection	✓	
Hong et al. [17]	object detection		✓
Huang et al. [19]	object detection	✓	
Wu et al. [43]	object detection	✓	
Xu et al. [45]	object detection	✓	
Hu et al. [18]	object detection	✓	
Hamdi et al. [16]	object detection	✓	
Ji et al. [21]	object detection	✓	
Lovisotto et al. [26]	object detection	✓	
Köhler et al. [23]	object detection	✓	
Wang et al. [40]	object detection	✓	
Zolfi et al. [51]	object detection	✓	
Wang et al. [41]	object detection	✓	
Zhu et al. [50]	object detection	✓	
Wang et al. [42]	Traffic light detection	✓	
Tang et al. [39]	Traffic light detection		✓

TABLE I. EVALUATION METHODOLOGIES OF EXISTING AD CAMERA OBJECT DETECTION ATTACK WORKS.

B. Design rationale of system-driven evaluation infrastructure

To enable system-level evaluation, prior works adopt two types of methodologies: *real vehicle-based* [30] and *simulation-based* [5, 17, 34, 39] evaluations. However, a fundamental design trade-off exists between these two evaluation methodologies. Specifically, real vehicle-based is *more fidel* since the vehicle, sensors, and physical environment are all in the evaluation loop. However, it requires costly full AD systems (e.g., $250K per vehicle [9]) and testing tracks, which are generally unaffordable for most academic research groups. On the other hand, the simulation-based methodology is better in all other important research evaluation aspects, ranging from a much lower cost, free of safety issues, high scenario flexibility, convenience of attack deployment, much faster evaluations, to high reproducibility. The main concern is the evaluation fidelity. However, the simulation fidelity technology is still evolving as this is also the need for the entire AD industry [10], for example recently there are various new advances in both industry [4] and academia [8, 29]); Considering the benefits, we thus adopt a simulation-based design for the system-driven evaluation infrastructure.

Call for community-level effort. For such system-driven evaluation infrastructure, it is highly desired if its development can be a community-level effort, since (1) the engineering efforts spent in instrumenting AD simulation for security research evaluation share common design/implementation patterns (e.g., common attack entry points); and (2) in AD context, the system-level attack effect can be highly influenced by driving scenario setups (e.g., large braking distance differences in highway and local roads [3] and thus the system-level evaluation results are only comparable (and thus scientifically-meaningful) if the same evaluation scenario and metric calculation are used. Considering the criticality of such a methodology-level scientific gap, in this work we thus take the initiative to lay the groundwork to foster such a community-level effort, which is detailed in the next section.

III. $PASS$: A PLATFORM TO BRIDGE THE GAP

In this section, we present $PASS$, our initial efforts to bridge the aforementioned scientific gap regarding the current general lack of system-level evaluation (§II-A2), by designing a simulation-based system-driven evaluation platform.

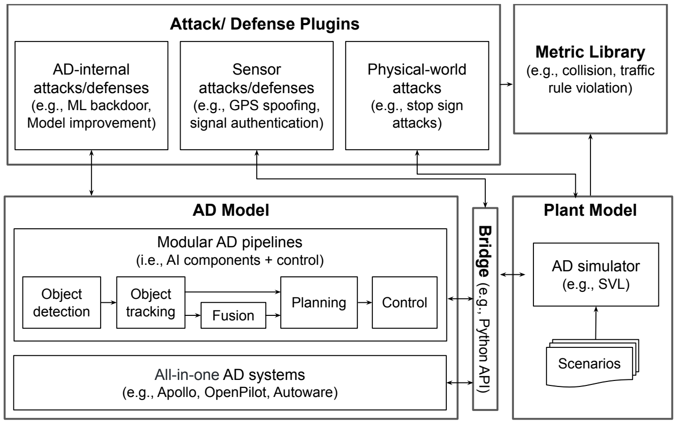

Attack/ Defense Plugins
AD-internal attacks/defenses (e.g., ML backdoor, Model improvement)
Sensor attacks/defenses (e.g., GPS spoofing, signal authentication)
Physical-world attacks (e.g., stop sign attacks)
Metric Library (e.g., collision, traffic rule violation)
AD Model
Modular AD pipelines (i.e., AI components + control)
Object detection
Object tracking
Fusion
Planning
Control
All-in-one AD systems (e.g., Apollo, OpenPilot, Autoware)
Bridge (e.g., Python API)
Plant Model
AD simulator (e.g., SVL)
Scenarios

Fig. 2. Design of $PASS$ (Platform for Auto-driving Safety and Security).

A. Design goals

The main goal of $PASS$ is building a uniform and extensible system-driven evaluation platform for various AI safety and security works in the AD context. To achieve uniformity, evaluation scenarios and metrics need to be unified so that the evaluation results of different works can be intuitively visible and comparable. Attack/defense implementations, evaluation setup and AD design should be standardized and modular so that the existing works can be easily reproduced and new attacks/defenses, AD system designs, and evaluation scenarios can be collectively developed by researchers to fit future needs.

B. $PASS$ overview

As show in Fig. 2, guided by the design goals, we build a simulation-based evaluation platform prototype with five main modules: **AD model**, **plant model**, **bridge**, **attack/defense plugins** and **metric library**.

1) AD model: $PASS$ provides a modular AD system pipeline similar to industry-grade AD systems, including AI components commonly targeted by recent attacks/defenses. Each AI component is designed to be replaceable for future needs. $PASS$ also provides some all-in-one AD systems such as Apollo Opensource [14], OpenPilot [31] and Autoware [13].

2) Plant model: The plant model consists of vehicle kinematics and physical driving environment. $PASS$ chooses a high fidelity industry-grade simulator SVL [25] to provide the plant model. Compared to real vehicles and testing tracks, simulation-based plant model has great advantages in affordability, efficiency, and safety (§II-B). The plant model also defines a list of driving scenarios to describe the evaluation setup including AD vehicle's initial position, equipped sensors, drivable area, and surrounding environmental dynamics (e.g., vehicles, pedestrians, traffic signals). The driving scenarios are formalized as human-readable configuration files for easy modification and contribution.

3) Bridge: The bridge serves as a communication channel between the AD and plant models, allowing sensor data to be read and the AD vehicle to be actuated. It supports function hooking for modifying communication data at runtime for better extensibility.

4) Attack/defense plugins: The attack/defense implementation is abstracted as three types of plugins in $PASS$. The plugins allow researchers to deploy their attacks and defenses directly in the platform without worrying the low-level implementation. In particular, each plugin is designed as a Python API that takes different kinds of attack/defense as input. For example, physical-world attack plugin can load adversarial patches (e.g., famous stop sign attacks *ShapeShifter* [7], *Robust Physical Perturbations* [12], and *Seeing Isn't Believing* [48]) to the simulation world at arbitrary locations; AD-internal attack/defense plugin can replace simple AI components inside the AD systems; And sensor attack/defense plugin can modify/check sensor information on the bridge.

5) Metric library: Metric library is in charge of collecting measurements from all other modules in the platform and calculating the scenario-dependent evaluation metrics. With the measurements from the plant model, metric library can quantify the impact of safety&security works at system-level (e.g., collision rate [20, 34]), traffic rule violation (e.g., lane departure rate [34, 37]), trip delay, etc. We also include component-level metrics (e.g., frame-wise attack success rate [7, 12, 48]) for comprehensiveness.

IV. EXPERIENCES FROM PRELIMINARY EFFORTS

We have implemented a preliminary prototype of $PASS$; in this section, we share our previous attempts on using it for system-level attack/defense evaluation in the AD context. We also provide a list of future actions to improve the platform.

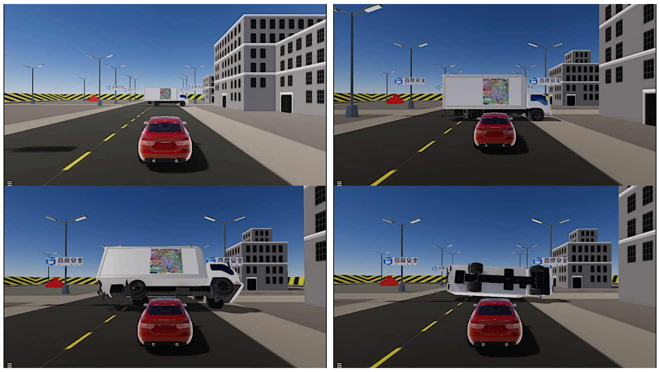
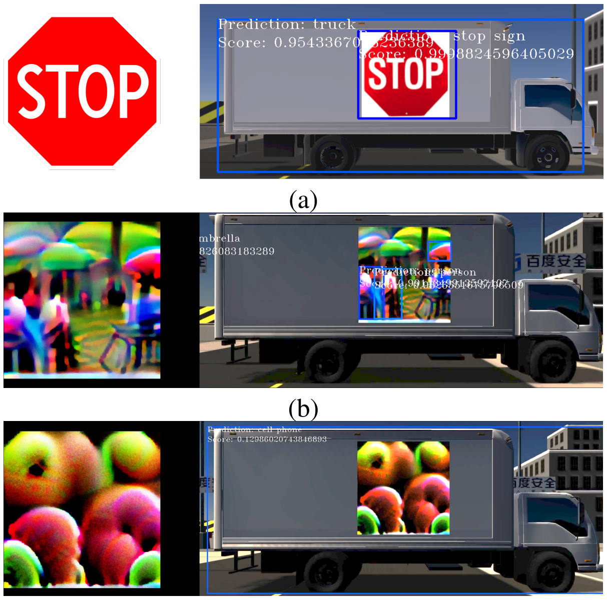


Fig. 3. Demonstration of the "Invisible Truck" challenge

A. Our preliminary efforts

We used the preliminary prototype to organize two AD security Capture The Flag (CTF) competitions [1]: AutoDriving CTF @ BCTF and AutoDriving CTF @ DEF CON 29 in the year 2020 and 2021, respectively. Unlike the AD safety and security works in many academic research, which mainly use component-specific metrics (e.g., AI model detection rate) to demonstrate the effect, our AutoDriving CTF challenges are designed to study how the attacks/defenses can affect the AD vehicles with respect to the end-to-end system-level behaviors (e.g., whether AD vehicles collide with obstacles or not).

Leveraging these CTF challenges, we visually demonstrate that successful attacks at the component-level might not deliver a successful result at the system-level. Fig. 3 shows the "invisible truck" challenge. Players are asked to generate adversarial patches, which will be attached on the side panel of a truck. The player's goal is to make the truck undetectable by the object detection component and consequently make the AD vehicle crash into the truck. The AD vehicle is equipped with both an object detection component and a tracking component [49]. The latter one tracks detected objects in a frame sequence to tolerant the occasional false positions and false negatives. Clearly, players need to overcome a system with both components in order to score in such challenge. Causing a single frame misclassified by the object detection is not enough, instead, the players need to maintain mis-detection in a long sequence of continuous frames.



(a)

(b)

(c)

Fig. 4. Some examples of failed and successful attacks in the "invisible truck" challenge submitted by participating teams in AutoDriving CTF @ BCTF.

Fig. 4 presents a few sample submissions from various teams in AutoDriving CTF @ BCTF. The attacks that cannot achieve component-level success such as the naive drawing shown in Fig. 4-(a) cannot make the AD vehicle crash into the truck. While some adversarial patch that can achieve relatively high component-level performance (e.g., the patch shown in Fig. 4-(b) has over 60% success rate), it cannot fool the tracking component and thus fails to pass the challenge. Only few attacks (e.g., the patch shown in Fig. 4-(c)) can deceive continuous frames and eventually lead to collisions. For more details of the challenge and team submissions, you can find the video demo at https://shorturl.at/nvhrz.

Apart from the aforementioned camouflage patch challenge, we also presented other challenges related to AD safety and security such as sensor spoofing/hijacking; multi-sensor fusion manipulations; robust control, etc. We hope that through these CTF challenges, the community can intuitively understand the need for system-level evaluation. With the simulation of the physical world interactions, it also encourages the research teams to overcome physical world attack challenges such as color distortions and viewing angle variations.

B. Towards a better evaluation platform

Based on the above experience, we summarize a few follow-up direction (and welcome community feedback) for building a better system-driven evaluation infrastructure:

(1.) More AD system designs. The most important and attractive feature of the future infrastructure should include a variety of AD systems, especially those used in real/-commercial AD vehicles. In the current version of $PASS$, we implement 3 variations of modular AD pipelines based on the availability of map information and fusion methods. Understanding the difficulties of reproducing commercial AD pipelines or contributing a new one, we hope to work with the community to propose data communication standards between different components within the modular AD pipelines so that different AD system designs can be easily plugged into the infrastructure. **(2). More attack/defense interfaces**. To cover the emerging research works in AD safety and security, more types of attack/defense interfaces are needed. In the current version of $PASS$, we implement 3 general attack/defense interfaces including physical-world attack interface (e.g., loading adversarial patches to the simulation world at arbitrary locations), sensor attack/defense interface (e.g., sensor data modification), and AD-internal attack/defense interfaces (e.g., simple AI model replacement). We hope to work with the community to propose more types of attack/defense interfaces in the future. **(3). More evaluation scenarios**. Although the simulation-based evaluation platform greatly accelerates the generation of driving scenarios compared with real road tests, these scenarios cannot cover all unexpected ones in the real world. The current version of $PASS$ covers 45 stop sign scenarios and almost 1,000 traffic light scenarios across different environment setting such as road type, speed limit, weather, and lighting. We hope to work with the community to standardize and parameterize the driving scenarios so that everyone can quickly expand the scene data set.

We hope $PASS$ can initiate a community-level effort to collaboratively build a common system-driven evaluation infrastructure for AD safety and security. We will open-source the platform completely and welcome community contributions of new attack/defense interfaces, new AD systems, and new system-level evaluation scenarios (e.g., driving scenarios).

V. Conclusion

In this paper, we present our initial efforts to develop an open-source, uniform, and extensible system-driven evaluation platform for the community. By sharing our evaluation infrastructure building efforts we hope to call on the community to foster more extensive, realistic, and democratized future research into the critical research space of AD AI security.

Acknowledgment

This work was supported in part by NSF CNS-1932464, NSF CNS-1929771 and NSF CNS-2145493.

References

[1] “Autonomous driving CTF contest,” https://autodrivingctf.org/.

[2] “CSRankings: Computer Science Rankings,” http://csrankings.org/.

[3] “A Policy on Geometric Design of Highways and Streets, 7th Edition,” *AASHTO*, 2018.

[4] “Applied Intuition,” https://www.appliedintuition.com/.

[5] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, “Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks,” in *IEEE S&P*. IEEE, 2021.

[6] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, “Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving,” in *ACM CCS*, 2019.

[7] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, “ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector,” in *ECML PKDD*. Springer, 2018, pp. 52–68.

[8] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, “GeoSim: Realistic Video Simulation via Geometry-Aware Composition for Self-Driving,” in *CVPR*, 2021, pp. 7230–7240.

[9] “What it really costs to turn a car into a self-driving vehicle,” https://qz.com/924212/.

[10] “GM Safety Report 2018,” https://www.gm.com/content/dam/company/docs/us/en/gmcom/gmsafetyreport.pdf.

[11] T. Dreossi, A. Donzé, and S. A. Seshia, “Compositional Falsification of Cyber-Physical Systems with Machine Learning Components,” *Journal of Automated Reasoning*, vol. 63, no. 4, pp. 1031–1053, 2019.

[12] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, T. Kohno, and D. Song, “Physical Adversarial Examples for Object Detectors,” in *WOOT*, 2018.

[13] “Autoware-AI,” https://github.com/Autoware-AI/autoware.ai.

[14] “Baidu Apollo,” https://github.com/ApolloAuto/apollo.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” *arXiv preprint arXiv:1412.6572*, 2014.

[16] A. Hamdi, M. Müller, and B. Ghanem, “SADA: Semantic Adversarial Diagnostic Attacks for Autonomous Applications,” in *AAAI*, 2020.

[17] D. K. Hong, J. Kloosterman, Y. Jin, Y. Cao, Q. A. Chen, S. Mahlke, and Z. M. Mao, “AVGuardian: Detecting and Mitigating Publish-Subscribe Overprivilege for Autonomous Vehicle Systems,” in *EuroS&P*, 2020.

[18] S. Hu, Y. Zhang, S. Laha, A. Sharma, and H. Foroosh, “CCA: Exploring the Possibility of Contextual Camouflage Attack on Object Detection,” in *ICPR*. IEEE, 2021.

[19] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu, “Universal Physical Camouflage Attacks on Object Detectors,” in *CVPR*, 2020.

[20] S. Jha, S. Cui, S. Banerjee, J. Cyriac, T. Tsai, Z. Kalbarczyk, and R. K. Iyer, “ML-driven Malware that Targets AV Safety,” in *DSN*. IEEE, 2020.

[21] X. Ji, Y. Cheng, Y. Zhang, K. Wang, C. Yan, W. Xu, and K. Fu, “Poltergeist: Acoustic Adversarial Machine Learning against Cameras and Computer Vision,” in *IEEE S&P*, 2021.

[22] Y. Jia, Y. Lu, J. Shen, Q. A. Chen, H. Chen, Z. Zhong, and T. Wei, “Fooling Detection Alone is Not Enough: Adversarial Attack against Multiple Object Tracking,” in *ICLR*, 2020.

[23] S. Köhler, G. Lovisotto, S. Birnbach, R. Baker, and I. Martinovic, “They See Me Rollin’: Inherent Vulnerability of the Rolling Shutter in CMOS Image Sensors,” in *ACSAC*, 2021.

[24] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” *arXiv e-prints*, p. arXiv:1607.02533, Jul 2016.

[25] LG, “SVL Simulator: An Autonomous Vehicle Simulator,” https://github.com/lgsvl/simulator.

[26] G. Lovisotto, H. Turner, I. Sluganovic, M. Strohmeier, and I. Martinovic, “SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations,” in *USENIX Security*, 2021.

[27] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, “No Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles,” *arXiv preprint arXiv:1707.03501*, 2017.

[28] Y. Man, M. Li, and R. Gerdes, “GhostImage: Remote Perception Attacks against Camera-based Image Classification Systems,” in *RAID*, 2020.

[29] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma, and R. Urtasun, “LiDARsim: Realistic LiDAR Simulation by Leveraging the Real World,” in *CVPR*, 2020.

[30] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici, “Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks,” in *ACM CCS*, 2020.

[31] “Comma AI openpilot,” https://github.com/commaai/openpilot.

[32] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles,” *IV*, vol. 1, no. 1, pp. 33–55, 2016.

[33] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, “Intriguing Properties of Adversarial ML Attacks in the Problem Space,” in *IEEE S&P*. IEEE, 2020, pp. 1332–1349.

[34] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, “Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack,” in *USENIX Security*, 2021.

[35] S. A. Seshia, S. Jha, and T. Dreossi, “Semantic Adversarial Deep Learning,” *IEEE Design & Test*, vol. 37, no. 2, pp. 8–18, 2020.

[36] J. Shen, N. Wang, Z. Wan, Y. Luo, T. Sato, Z. Hu, X. Zhang, S. Guo, Z. Zhong, K. Li, Z. Zhao, C. Qiao, and Q. A. Chen, “SoK: On the Semantic AI Security in Autonomous Driving,” *arXiv preprint arXiv:2203.05314*, 2022.

[37] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, “Drift with Devil: Security of Multi-Sensor Fusion based Localization in High-Level Autonomous Driving under GPS Spoofing,” in *USENIX Security*, 2020.

[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing Properties of Neural Networks,” in *ICLR*, 2014.

[39] K. Tang, J. S. Shen, and Q. A. Chen, “Fooling Perception via Location: A Case of Region-of-Interest Attacks on Traffic Light Detection in Autonomous Driving,” in *NDSS Workshop AutoSec*, 2021.

[40] D. Wang, C. Li, S. Wen, Q.-L. Han, S. Nepal, X. Zhang, and Y. Xiang, “Daedalus: Breaking Nonmaximum Suppression in Object Detection via Adversarial Examples,” *IEEE Transactions on Cybernetics*, 2021.

[41] J. Wang, A. Liu, Z. Yin, S. Liu, S. Tang, and X. Liu, “Dual Attention Suppression Attack: Generate Adversarial Camouflage in Physical World,” in *CVPR*, 2021.

[42] W. Wang, Y. Yao, X. Liu, X. Li, P. Hao, and T. Zhu, “I Can See the Light: Attacks on Autonomous Vehicles Using Invisible Lights,” in *ACM CCS*, 2021.

[43] Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, “Making an Invisibility Cloak: Real World Adversarial Attacks on Object Detectors,” in *ECCV*, 2020.

[44] C. Xiao, D. Yang, B. Li, J. Deng, and M. Liu, “MeshAdv: Adversarial Meshes for Visual Recognition,” in *CVPR*, 2019.

[45] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, “Adversarial T-shirt! Evading Person Detectors in A Physical World,” in *ECCV*, 2020.

[46] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A Survey of Autonomous Driving: Common Practices and Emerging Technologies,” *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[47] Y. Zhang, P. H. Foroosh, and B. Gong, “CAMOU: Learning A Vehicle Camouflage For Physical Adversarial Attack On Object Detections In The Wild,” *ICLR*, 2019.

[48] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, “Seeing isn’t Believing: Towards More Robust Adversarial Attack Against Real World Object Detectors,” in *ACM CCS*, 2019.

[49] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, “Online Multi-Object Tracking with Dual Matching Attention Networks,” in *ECCV*, 2018, pp. 366–382.

[50] X. Zhu, X. Li, J. Li, Z. Wang, and X. Hu, “Fooling thermal infrared pedestrian detectors in real world using small bulbs,” in *AAAI*, 2021.

[51] A. Zolfi, M. Kravchik, Y. Elovici, and A. Shabtai, “The Translucent Patch: A Physical and Universal Attack on Object Detectors,” in *CVPR*, 2021.