SoK: On the Semantic AI Security in Autonomous Driving

Junjie Shen, Ningfei Wang, Ziwen Wan, Yunpeng Luo, Takami Sato, Zhisheng Hu†, Xinyang Zhang‡, Shengjian Guo‡, Zhenyu Zhong‡, Kang Li‡, Ziming Zhao§, Chunming Qiao§, Qi Alfred Chen

{junjies1, ningfei.wang, ziwenw8, yunpel3, takamis, alfchen}@uci.edu, †zhu@zoox.com
‡{xinyangzhang, sjguo, edwardzhong, kangli01}@baidu.com, §{zimingzh, qiao}@buffalo.edu
UC Irvine, †ZOOX, ‡Baidu Security, §University at Buffalo

***Abstract*—Autonomous Driving (AD) systems rely on AI components to make safety and correct driving decisions. Unfortunately, today's AI algorithms are known to be generally vulnerable to adversarial attacks. However, for such AI component-level vulnerabilities to be semantically impactful at the system level, it needs to address non-trivial semantic gaps both (1) from the system-level attack input spaces to those at AI component level, and (2) from AI component-level attack impacts to those at the system level. In this paper, we define such research space as *semantic AI security* as opposed to generic AI security. Over the past 5 years, increasingly more research works are performed to tackle such semantic AI security challenges in AD context, which has started to show an exponential growth trend. However, to the best of our knowledge, so far there is no comprehensive systematization of this emerging research space.**

In this paper, we thus perform the first systematization of knowledge of such growing semantic AD AI security research space. In total, we collect and analyze 53 such papers, and systematically taxonomize them based on research aspects critical for the security field such as the attack/defense targeted AI component, attack/defense goal, attack vector, attack knowledge, defense deployability, defense robustness, and evaluation methodologies. We summarize 6 most substantial scientific gaps observed based on quantitative comparisons both vertically among existing AD AI security works and horizontally with security works from closely-related domains. With these, we are able to provide insights and potential future directions not only at the design level, but also at the research goal, methodology, and community levels. To address the most critical scientific methodology-level gap, we take the initiative to develop an open-source, uniform, and extensible system-driven evaluation platform, named *PASS*, for the semantic AD AI security research community. We also use our implemented platform prototype to showcase the capabilities and benefits of such a platform using representative semantic AD AI attacks.

I. Introduction

Autonomous Driving (AD) vehicles are now a reality in our daily life, where a wide variety of commercial and private AD vehicles are already driving on the road. For example, commercial AD services, such as self-driving taxis [1, 2], buses [3, 4], trucks [5, 6], delivery vehicles [7] are already publicly available, not to mention the millions of Tesla cars [8] that are equipped with Autopilot [9]. To achieve driving autonomy in complex and dynamic driving environments, AD systems are designed with a collection of AI components to handle the core decision-making process such as perception, localization, prediction, and planning, which essentially forms the "brain" of the AD vehicle. However, this makes these AI components highly security-critical as errors in them can cause various road hazards and even fatal consequences [10, 11].

Unfortunately, today's AI algorithms, especially deep learning, are known to be generally vulnerable to adversarial attacks [12, 13]. However, since these AI algorithms are only components of the entire AD system, it is widely recognized that such generic AI component-level vulnerabilities do not necessarily lead to system-level vulnerabilities [14–17]. This is mainly due to the large **semantic gaps**: (1) from the system-level attack input spaces (e.g., adding stickers [18], laser shooting [19]) to those at the AI component level (e.g., image pixel changes [12, 13]), or *system-to-AI semantic gap*, which needs to overcome fundamental design challenges to map successful attacks at the AI input space back to the problem space, generally called the *inverse-feature mapping problem* [15]; and (2) from AI component-level attack impacts (e.g., misdetected road objects) to those at the system level (e.g., vehicle collisions), or *AI-to-system semantic gap*, which is also quite non-trivial, e.g., when the misdetected object is at a far distance for automatic emergency braking [14, 16], or the misdetection can be tolerated by subsequent AI modules like object tracking [17]. Thus, for an AI security work to be semantically meaningful at the system level, it must explicitly or implicitly address these 2 general semantic gaps. In this paper, we call such research space **semantic AI security** (as opposed to generic AI security), following the semantic adversarial deep learning concept by Seshia et al. [14].

Over the past 5 years, increasingly more research works are performed to tackle the aforementioned semantic AI security challenges in AD context, which started to show an exponential growth trend since 2019 (Fig. 1). However, to the best of our knowledge, so far there is no comprehensive systematization of this emerging research space. There are surveys related to AD security, but they either did not focus on AD AI components (e.g., on sensor/hardware [20], in-vehicle network [21]), or touched upon AD AI components but did not focus on the works that addressed the semantic AI security challenges above [22, 23]. Since (1) the latter ones are much more semantically and thus practically meaningful for AD systems, (2) now a substantial amount of them have appeared (over 50 as in Fig. 1), and (3) such attacks in AD context have especially high safety-criticality since AD vehicles are heavy, fast-moving, and operate in public spaces, we believe now is a good time to summarize the current status, trends, as well as scientific gaps, insights, and future research directions.

In this paper, we perform the first systematization of knowledge (SoK) of the growing semantic AD AI security research space. In total, we collect and analyze 53 such papers, with

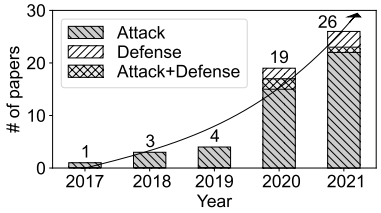

Figure 1. Number and trends of semantic AD AI security papers (collected with a focus on top-tier venues (§II-C)).

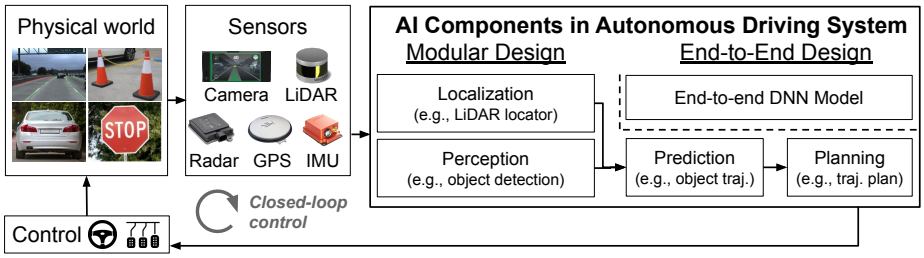

Figure 2. Overview of AD system designs and the roles of AD AI components.

a focus on those published in (commonly-recognized) top-tier venues across security, computer vision, machine learning, AI, and robotics areas in the past 5 years since the first one appeared in 2017 (§II-C). Next, we taxonomize them based on research aspects critical for the security field, including the targeted AI component, attack/defense goal, attack vector, attack knowledge, defense deployability, defense robustness, as well as evaluation methodologies (§III). For each research aspect, we emphasize the observed domain/problem-specific design choices, and summarize their current status and trends.

Based on the systematization, we summarize 6 most substantial scientific gaps (§IV) observed based on quantitative comparisons both vertically among existing AD AI security works and horizontally with security works from closely-related domains (e.g., drone [24]). With these, we are able to provide insights and potential future directions not only at the design level (e.g., under-explored attack goals and vectors), but also at the research goal and methodology levels (e.g., the general lack of system-level evaluations), as well as at the community level (e.g., the substantial lacking of open-sourcing specifically for the works in the security community).

Among all these scientific gaps, the one on the general lack of system-level evaluation is especially critical as it may lead to meaningless attack/defense progress at the system level due to the AI-to-system semantic gap (§IV-A). To effectively fill this gap, it is highly desired to have a community-level effort to collectively build a common system-level evaluation infrastructure, since (1) the engineering efforts for building such infrastructure share common design/implementation patterns; and (2) in AD context, the system-level evaluation results are only comparable (and thus scientifically-meaningful) if the same evaluation scenario and metric calculation are used.

In this paper, we thus take the initiative to address this critical scientific methodology-level gap by developing a uniform and extensible system-driven evaluation platform, named *PASS* (Platform for Autonomous driving Safety and Security), for the semantic AD AI security research community (§V). We choose a simulation-centric hybrid design leveraging both simulation and real vehicles to balance the trade-offs among fidelity, affordability, safety, flexibility, efficiency, and reproducibility. The platform will be fully open-sourced (will be available on our project website [25]) so that researchers can collectively develop new interfaces to fit future needs, and also contribute attack/defense implementations to form a semantic AD AI security benchmark, which can improve comparability, reproducibility, and also encourage open-sourcing. We have implemented a prototype, and use it to showcase an example usage that performs system-level evaluation of the most popular AD AI attack category, camera-based STOP sign detection [18, 26], using 45 different combinations of system-level scenario setups (i.e., speeds, weather, and lighting). We find that the AI component-level and AD system-level results are quite different and often contradict each other in common driving scenarios, which further demonstrates the necessity and benefits of such system-level evaluation infrastructure building effort. Demos are available on our project website **https://sites.google.com/view/cav-sec/pass** [25].

In summary, this work makes the following contributions:

- We perform the first SoK of the growing semantic AD AI security research space. In total, we collect and analyze 53 such papers, and systematically taxonomize them based on research aspects critical for the security field, including the targeted AI component, attack/defense goal, attack vector, attack knowledge, defense deployability, defense robustness, and evaluation methodologies.
- We summarize 6 most substantial scientific gaps observed based on quantitative comparisons both vertically among existing AD AI security works and horizontally with security works from closely-related domains. With these, we are able to provide insights and potential future directions not only at the design level, but also at the research goal, methodology, and community levels.
- To address the most critical scientific methodology-level gap, we take the initiative to develop an open-source, uniform, and extensible system-driven evaluation platform *PASS* for the semantic AD AI security research community. We also use our implemented prototype to showcase the capabilities and benefits of such a platform using representative AD AI attacks.

II. Background and Systematization Scope

A. A Primer on AD Systems and the AI Components

1) AD Levels and Deployment Status: The Society of Automotive Engineers (SAE) defines 6 AD levels – Level 0 (L0) to 5 (L5) [27], where as the level increasing, the driving is more automated. In particular, L0 refers to no automation. However, the vehicle may provide warning features such as Forward Collision Warning and Lane Departure Warning. L1 refers to driver assistance and is the minimum level of AD. In L1 vehicles, the AD system is in charge of *either* steering *or* throttling/braking. Examples of L1 features are Automated Lane Centering and Adaptive Cruise Control. L2 means partial automation, where the AD system controls *both* steering *and* throttling/braking. Although L1 and L2 can at least partially drive the vehicle, the driver must actively monitor and be ready

to take over at any circumstances. When the autonomy level goes beyond L3, the driver does not need to be attentive when the AD system is operating in its Operational Design Domains (ODDs). However, in L3, the driver is required to take over when the AD system requests to do so. None of L4 and L5 vehicles require a driver seat. The difference is that L4 AD systems can only operate in limited ODDs whereas L5 can handle all possible driving scenarios. Currently, L2 AD is already widely available and targets consumer vehicles (e.g., Tesla Autopilot [9], Cadillac Super Cruise [28], OpenPilot [29]). L4 AD is under rapid deployment targeting transportation services such as self-driving taxis [1,2], buses [3,4], trucks [5,6], and delivery robots [7], with some of them already entered the commercial operation stage that charges customers [30,31].

2) AI Components in AD Systems: The AD systems generally have multiple AI components, including perception, localization, prediction, and planning as the major categories, as shown in Fig. 2. *Perception* refers to perceiving the surrounding environments and extracting the semantic information for driving, such as road object (e.g. pedestrian, vehicles, obstacles) detection, object tracking, segmentation, lane/traffic light detection. Perception module usually takes diverse sensor data as the inputs, including camera, LiDAR, RADAR, etc. *Localization* refers to finding the position of the AD vehicle relative to a reference frame in an environment [32]. *Prediction* aims to estimate the future status (position, heading, velocity, acceleration, etc.) of surrounding objects. *Planning* aims to make trajectory-level driving decisions (e.g., cruising, stopping, lane changing, etc.) that are safe and correct (e.g., conforming to traffic rules). Besides such modular design, people also explored *end-to-end DNN models* for AD. Currently, since the modular design is easier to debug, interpret, and hard-code safety rules/measures, it is predominately adopted in industry-grade AD systems, while the end-to-end designs are only for demonstration purposes [33]. For more details, we refer the readers to recent surveys [32–35].

B. Adversarial (AI) Attacks and Defenses

Recent works find that AI models are generally vulnerable to *adversarial attacks* [12,36] Some works further explored such attacks in the physical world [18,19,37,38]. With that, some defenses [39,40] are proposed recently to defend against such attacks. Such AI attacks/defenses are directly related to AD systems due to the high reliance on AI components (§II-A2). However, as described in §I, generic AI component-level vulnerabilities do not necessarily lead to system-level vulnerabilities due to the general system-to-AI and AI-to-system semantic gaps. In this paper, we thus focus on the works that address *semantic AI security* in the AD context.

C. Systematization Scope

In our systematization, we consider within-scope the attacks that aim to address the *semantic AI security* challenges (defined in §I) and the defenses that are designed specifically for addressing the AI component-level vulnerabilities revealed in such attack works. Thus, we consider out-of-scope the works that assume digital space perturbations without justifying the feasibility in the AD context (i.e., without addressing the system-to-AI semantic gap) [41–44] and the ones that focus only on AI algorithms that are not used in representative AD system designs today (§II-A2), e.g., general image classification [45–47]. Curious readers can refer to general adversarial attack/defense SoK [48] and surveys [49,50].

When collecting the papers, we mainly focus on the ones published in commonly-recognized top-tier venues [51] 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.

D. Related Work

Before this paper, a few AD security-related surveys have been published [20–23], but none of them focus on the emerging semantic AD AI research space (i.e., our scope defined in §II-C). For example, Kim et al. [20] and Ren et al. [21] focus on AD-related sensor/hardware security and in-vehicle network security, instead of AD AI components. Qayyum et al. [23] and Deng et al. [22] touched upon the security of AI components in AD, but did not focus on the works that addressed the semantic AI security challenges (e.g., most of the included works are on generic AI and sensor security without studying impacts on AD AI behaviors). In fact, Deng et al. [22] considers the semantic AD AI security as a future direction. This is mainly because both of them focus on works published in 2019 and before. However, the majority (85%) and the exponential growth of semantic AD AI security works are after 2019 (Fig. 1). This leads to much more complete, diverse, and quantifiable observations of the current status, trends, and scientific gaps of this emerging research space in this paper, not to mention that we also take the initiative to address one of the most critical gaps.

In the general CPS (Cyber-Physical System) area, there are also SoKs on the security of technologies related to AD AI, for example on drones [24] that share similar controller designs, on Automatic Speech Recognition and Speaker Identification (ASR/SI) [52] that are also AI-enabled CPS, and on sensor technology [53] that provide the main inputs to AD AI. In comparison, the SoKs on drone and ASR/SI security focused on domain-specific attack vectors (e.g., ground control station channel and voice signals), which thus lead to vastly different set of systematized knowledge (§III) due to the CPS domain differences. The SoK on sensor security is complementary to us as it does not directly consider AI component security but sensor attack is one major attack vector to AD AI (§III-B2).

III. Systematization of Knowledge

In this section, we taxonomize the semantic AD AI security works published in the past 5 years since the first one appeared in 2017. In total, 53 papers fall into our scope (§II-C) across security, CV, ML, AI, and robotics research areas; 48 discovered new attacks (Table I) and 8 developed effective new defense solutions (Table II). Fig. 1 shows the number of papers in each year. The earliest paper [54] dates back to 2017 when adversarial attacks (§II-B) began to be applied

to many real-world application scenarios including AD. Since then, AD AI security has gained much attention as reflected in the exponential growth trend of paper numbers over the years.

Similar to many other fields, AD AI security research also has gone through enlightening periods where early beliefs are later overturned. For example, the 2017 paper [54] questions the severity of adversarial attacks in AD vehicles and claims that "no need to worry about adversarial examples in object detection in autonomous vehicles". Soon after that, two papers in 2018 propose successful attacks against camera object detection with physical-world attack demonstrations. As the community exponentially grew after 2019 and now at over 50 papers in this research space, we think now might be a good time to systematize the existing research efforts.

A. (Attack/Defense) Targeted AI Components

Status and trends. The targeted AI components in the existing works are summarized in Tables I and II. As shown in the tables and Fig. 6 in Appendix, most (>86%) of the existing works target perception, while localization, chassis, and end-to-end driving are all less or equal to 6.2%. Among the perception works, the two most popular ones are camera (60.0%) and LiDAR (21.5%) perception. More detailed summary of their designs, applications in AD systems, and vulnerabilities are in Appendix A. Currently, *none* of the existing works study the downstream AI components such as prediction and planning, which will be discussed more in §IV-D.

B. Systematization of Semantic AD AI Attacks

Table I summarizes the semantic AD AI attacks. We taxonomize them based on 3 research aspects critical for the security field: Attack goal, attack vector, and attacker's knowledge.

1) Attack Goals: We categorize the attack goals in the existing works based on the general security properties such as *integrity*, *confidentiality*, and *availability* [96]:

Integrity (of AI components). Integrity in AD context can be viewed as the integrity of the AI component outputs (i.e., whether they are changed by the attacker), which can directly impact the correctness of the AI driving behaviors. From this view, its violations manifest as the following AD-specific attack goals considered in existing works:

- *Safety hazards.* Safety is the top priority in AD design [97], and it is not only for the AD vehicle and its passengers, but also for other road users (e.g., other vehicles, pedestrians). Many existing attacks aim for safety hazards. For example, attacks on object detection and segmentation can cause safety hazards if a front vehicle is undetected [56, 71, 73]; attacks on object tracking can potentially lead to collisions if the front vehicle trajectory is incorrectly tracked [17, 75]; lane detection, localization, and end-to-end driving model attacks [78, 79, 92–94] can cause lane departure and thus potential collisions.
- *Traffic rule violations.* Since AD systems by design are required to follow traffic rules, violations can lead to financial penalties for individuals and reputational losses for AD companies. Existing attacks aim to hide the STOP sign to cause STOP sign violations [18, 26, 37]. Traffic light detection attack can cause the AD system to recognize a red light as green light, which may lead to red light violations [80]. In addition, the attacks on lane detection, localization, and end-to-end driving models can also cause vehicle lateral deviations, which can violate the lane line boundaries [78, 79, 92–94].
- *Mobility degradation.* A key benefit that AD technologies can bring is improved access to mobility-as-a-service (MaaS) [98]. A few works aim to degrade the mobility of AD vehicles by manipulating the AI component outputs. In those works, they fool either the object detection to recognize a static blocking obstacle [19, 81, 84] or the traffic light detection to recognize a permanent red light [80], which can cause the AD vehicle to be stuck on the road for a prolonged time. This not only delays the AD vehicle's trip to destination, but also may block the traffic and cause congestion.

Confidentiality (Privacy). Confidentiality is related to the sensitive information from or collected by the AD vehicle. This includes not only the vehicle identification information (e.g., the VIN from Chassis [59]), but also the privacy-sensitive location data [91] as it can reveal the passenger privacy.

Availability (of AI components). Availability in the general cybersecurity area means the systems, services, and information under viewed are accessible to users when they need them [96]. For an AD AI component, its "users" can be considered as all the downstream AD components and vehicle subsystems that are counting on its timely and reliable outputs to function correctly. Thus, the availability of an AD AI component can be defined as its capability to provide timely and reliable outputs.

Following this definition, example attacks on AD AI availability can be attacks causing delays or failures in the outputting function of a given AI component, e.g., by interrupting its input/output messaging channels, causing software crashes/hangs in it, or by causing the vehicle system to fall back to human operators (so stop the outputting function of the whole AD AI stack).

Status and trends. Vast majority (96.3%) of existing works focus on integrity, while only 3.7% are on confidentiality and so far none of them are on availability. More discussion on this are in §IV-E.

2) Attack Vectors: Existing attacks on AD systems leverage a diverse set of attack vectors and we broadly categorize them in two categories: *physical-layer* and *cyber-layer*. Physical-layer attack vectors refer to those tampering the sensor inputs to the AI components via physical means. We further decompose physical-layer attack vectors into *physical-world attack* and *sensor attack* vectors, where the former modifies the physical-world driving environment and the latter leverages unique sensor properties to inject erroneous measurements. Cyber-layer attack vector refers to those that require internal access to the AD system, its computation platform, or even the system development environment.

Physical-world attack vectors:

- *Object texture* refers to changing the surface texture (e.g., color) of 2D or 3D objects. It is often used by adversarial attacks to embed the malicious sensing inputs. In attack deployment, this is often fabricated as patches [18, 26, 78],

Targeted AI component		Paper	Year	Field	Attack goal: Integrity	Attack goal: Confidentiality	Attack goal: Availability	Attack vector: Physical-layer: Phys. world: Object texture	Attack vector: Physical-layer: Phys. world: Object shape	Attack vector: Physical-layer: Phys. world: Object position	Attack vector: Physical-layer: Sensor attack: GPS spoofing	Attack vector: Physical-layer: Sensor attack: LiDAR spoofing	Attack vector: Physical-layer: Sensor attack: Radar spoofing	Attack vector: Physical-layer: Sensor attack: Laser/IR light	Attack vector: Physical-layer: Sensor attack: Acoustic signal	Attack vector: Physical-layer: Sensor attack: Translucent patch	Attack vector: Cyber layer: ML backdoor	Attack vector: Cyber layer: Malware & s/w compromise	Attacker's knowledge	Eval. level: Component-level	Eval. level: System-level	Open source
Camera perception	Object detection	Lu et al. [54]	'17	V	✓			✓											○	✓		
		Eykholt et al. [18]	'18	S	✓			✓											○	✓		
		Chen et al. [37]	'18	M	✓			✓											○	✓		✓
		Zhao et al. [26]	'19	S	✓			✓											○	✓		
		Xiao et al. [55]	'19	V	✓			✓	✓										○	✓		
		Zhang et al. [56]	'19	M	✓			✓											◐	✓		✓
		Nassi et al. [57]	'20	S	✓			✓											○	✓	✓	
		Man et al. [58]	'20	S	✓									✓					○	✓		✓
		Hong et al. [59]	'20	S	✓													✓	○		✓	✓
		Huang et al. [60]	'20	V	✓			✓											○	✓		✓
		Wu et al. [61]	'20	V	✓			✓											○	✓		
		Xu et al. [62]	'20	V	✓			✓											○	✓		
		Hu et al. [63]	'20	V	✓			✓											◐	✓		
		Hamdi et al. [64]	'20	M	✓				✓										◐	✓		
		Ji et al. [65]	'21	S	✓										✓				◐	✓		
		Lovisotto et al. [66]	'21	S	✓			✓											●	✓		✓
		Wang et al. [67]	'21	S	✓									✓					◐	✓		
		Köhler et al. [68]	'21	S	✓									✓					●	✓		
		Wang et al. [69]	'21	S	✓			✓											●	✓		✓
		Zolfi et al. [70]	'21	V	✓											✓			○	✓		
		Wang et al. [71]	'21	V	✓			✓											◐	✓		✓
		Zhu et al. [72]	'21	M	✓									✓					○	✓		
	Semantic segmentation	Nakka et al. [73]	'20	V	✓			✓											○	✓		
		Nesti et al. [74]	'22	V	✓			✓											○	✓		
	Object tracking	Jha et al. [75]	'20	S	✓													✓	○		✓	
		Jia et al. [17]	'20	M	✓			✓											○	✓		✓
		Ding et al. [76]	'21	M	✓			✓											○	✓		
		Chen et al. [77]	'21	M	✓			✓											○	✓		
	Lane detection	Sato et al. [78]	'21	S	✓			✓											○	✓	✓	✓
		Jing et al. [79]	'21	S	✓			✓											◐	✓	✓	
	Traffic light detection	Wang et al. [67]	'21	S	✓									✓					○	✓		
		Tang et al. [80]	'21	S	✓						✓								○		✓	
LiDAR perception	Object detection	Cao et al. [19]	'19	S	✓							✓							○	✓	✓	
		Sun et al. [81]	'20	S	✓							✓							●	✓		
		Hong et al. [59]	'20	S	✓													✓	○		✓	
		Tu et al. [82]	'20	V	✓				✓										○	✓		
		Zhu et al. [83]	'21	S	✓					✓									◐	✓		
		Yang et al. [84]	'21	S	✓				✓										◐	✓	✓	
		Hau et al. [85]	'21	S	✓							✓							○	✓		
		Li et al. [86]	'21	V	✓						✓								○	✓		✓
		Zhu et al. [87]	'21	O	✓					✓									◐	✓		
	Semantic segmentation	Tsai et al. [88]	'20	M	✓				✓										○	✓		✓
		Zhu et al. [87]	'21	O	✓					✓									◐	✓		
RADAR perception	Obj. detection	Sun et al. [89]	'21	S	✓								✓						◐	✓	✓	
MSF perception		Cao et al. [38]	'21	S	✓				✓										○	✓	✓	✓
		Tu et al. [90]	'21	O	✓				✓										○	✓		
LiDAR localization		Luo et al. [91]	'20	S		✓												✓	○		✓	
MSF localization		Shen et al. [92]	'20	S	✓						✓								○	✓	✓	
Camera localization		Wang et al. [67]	'21	S	✓									✓					○	✓		
Chassis		Hong et al. [59]	'20	S		✓												✓	○		✓	
End-to-end driving		Liu et al. [93]	'18	S	✓			✓									✓		○	✓	✓	✓
		Kong et al. [94]	'20	V	✓			✓											○	✓		✓
		Hamdi et al. [64]	'20	M	✓				✓										◐	✓		
		Boloor et al. [95]	'20	O	✓			✓											●	✓	✓	✓

Field: S = Security, V = Computer Vision, M = ML/AI, O = Others, e.g., Robotics, arXiv;
Attacker's knowledge: ○ = white-box, ◐ = gray-box, ● = black-box

Table I. Overview of existing semantic AD AI attacks in our SoK scope (§II-C). (s/w = software)

posters [18, 37, 54], and camouflages [56, 60, 63], or displayed by projectors [66] and digital billboards [57]. Existing attacks have applied this attack vector on various objects such as STOP signs [18, 26, 37, 54], road surfaces [78,79], vehicles [56,60,63,71], clothes [61,62], physical billboards [94]. To disguise as benign looking, a few attacks also constrain the texture perturbations to improve the attack stealthiness [60,78,79].

- *Object shape* refers to perturbing the shape of 3D objects such as vehicles [55,88], traffic cones [38], rocks [38] or irregularly-shaped road objects [82, 84, 90]. Some were demonstrated in physical world via 3D printing [19,84].
- *Object position* refers to placing physical objects at specific locations in the environment. Prior work [83] applies this attack vector by controlling a drone to hold a board at a particular location in the air to fool the LiDAR object detector to misdetect the front vehicle.

Sensor attack vectors:

- *LiDAR spoofing* refers to injecting additional points in the LiDAR point cloud via laser shooting. Prior works carefully craft the injected point locations to fool the LiDAR object detection [19,81,85].
- *RADAR spoofing* refers to injecting malicious signals to RADAR inputs to cause it to resolve fake objects at specific distances and angles. Prior work [89] demonstrates RADAR spoofing capability in physical world and shows that it can cause AD system to recognize fake obstacles.
- *GPS spoofing* refers to sending fake satellite signals to the GPS receiver, causing it to resolve positions that are specified by the attacker. Prior works leverage GPS spoofing to attack MSF localization [92], LiDAR object detection [86], and traffic light detection [80].
- *Laser/IR light* refers to injecting/projecting laser or light directly to the sensor rather than the environment. Prior works use such attack vector to project malicious light spots in the camera image such that it can misguide the camera localization [67] or object detection [67,72]. Moreover, some prior works also use it to cause camera effects such as lens flare [58] and rolling shutter effect [68] in order to fool the object detection.
- *Acoustic signal* has been shown to disrupt or control the outputs of Inertial Measurement Units (IMUs) [99,100]. Prior work [65] used this attack vector to attack the camera stabilization system, which has built-in IMUs, to manipulate the camera object detection results.
- *Translucent patch* refer to sticking translucent films with color spots on camera lens. It can cause misdetect road objects such as vehicles and STOP signs [70].

Cyber-layer attack vectors:

- *ML backdoor* is an attack that tampers the training data or training algorithm to allow the model to produce desired outputs when specific triggers are present. Prior work [93] leverages this to attack the end-to-end driving model by presenting the trigger on a roadside billboard.
- *Malware & software compromise* is generic cyber-layer attack vectors assumed in prior works to eavesdrop or modify sensor data [75], or execute malicious programs alongside the AI components [59, 91]. Particularly, a domain-specific instance is the Robot Operating System (ROS) node compromises [59], which can modify inter-component messages within ROS-based AD systems, e.g., Apollo v3.0 and earlier [101] and Autoware [102].

Status and trends. As shown in Table I, existing works predominantly adopt physical-layer attack vectors, with 63.0% and 27.8% using physical-world and sensor attack vectors, respectively. Among the physical-world ones, object texture is the most popular (half of the all attacks). This is likely because such attack vectors are direct physical-world adaptations of the digital-space perturbations in general adversarial attacks. In contrast, only 6 (11.1%) attacks leverage cyber-layer attack vectors. More discussions on this are in §IV-C.

3) Attacker's Knowledge: We follow the general definitions of attacker's knowledge by Abdullah et al. [52]:

White-box. This setting assumes that the attacker has complete knowledge of the AD system, including the design and implementation of the targeted AI components, corresponding sensor parameters, etc. Among existing attacks, such white-box setting is the most commonly-adopted (Table I).

Gray-box. This setting assumes that part of the information required by white-box attacks is unavailable. Prior gray-box attacks all assume the lack of knowledge of AI model internals such as weights. However, some works still require confidence scores in the model outputs [56,63–65,71,79,83,84], and some require detailed sensor parameters [65, 67, 89].

Black-box. This is the most restrictive setting where the attacker cannot access any of the internals in the AD vehicle. Prior works that belong to this category are either transfer-based attacks [66, 69], which generate attack inputs based on local white-box models, or the ones that do not require any model-level knowledge in attack generation [68, 81].

Status and trends. As shown in Table I, existing works commonly assume the white-box settings in their attack designs (66.7%). However, in recent years, we start to see increasing research efforts in the more challenging but also more practical attack settings, i.e., gray-box (24.1%) and black-box (9.3%), mostly published in recent two years (except one). This greatly benefited the practicality and realism of this research space, and also shows that the community practices have evolved significantly from earlier years [22, 23].

C. Systematization of AD AI Defenses

Table II summarizes the defenses included in our SoK. We taxonomize them from 4 research aspects critical for the security field: Defense methods, defense goals, defense deployability, and robustness to adaptive attacks.

1) Defense Methods: In general, the defense methods in existing works can be categorized into two categories:

Consistency checking. Consistency checking is a general attack detection technique that cross-checks the attacked information either with other (ideally independent) measurement sources, or with invariant properties of itself. For example, prior works use the detected objects from stereo cameras [104] and object trajectory from prediction models [107] to cross-check LiDAR object detection results. Li et al. [103] and Nassi et al. [57] created detection models to determine whether the current camera object detection results are consistent with the

Target AI Component		Paper	Year	Field	Defense method	Defense goal	Deployability					Robust to adaptive attack	Eval. level		Open source
							Neg. timing overhead	Neg. resource overhead	No model training	No additional dataset	No h/w modification		Component-level	System-level	
Camera perception	Object detection	Nassi et al. [57]	'20	S	Driving context & physics consistency checking	D	✓				✓	✓	✓		✓
		Li et al. [103]	'20	V	Driving context consistency checking	D	?				✓	?	✓		✓
		Liu et al. [104]	'21	S	Sensor fusion based consistency checking	D						✓	✓		
		Chen et al. [105]	'21	V	Adversarial training	M	✓	✓		✓	✓	?	✓		
	Obj. tracking	Jia et al. [106]	'20	V	Predict & remove perturbation from inputs	M	?				✓	?	✓		✓
Camera perception & localization		Wang et al. [67]	'21	S	Consistency checking on reflected lights	D	?				✓	?	✓		
LiDAR perception	Obj. detection	Sun et al. [81]	'20	S	Consistency checking on point cloud free space	D	✓	✓	✓	✓	✓	✓	✓		
		Sun et al. [81]	'20	S	Augment inputs with obj. conf. from front view	M	?			✓	✓	✓	✓		
		You et al. [107]	'21	S	Trajectory consistency checking	D	✓		✓	✓	✓	?	✓		

Field: S = Security, V = Computer Vision, M = ML/AI, O = Others, e.g., Robotics, arXiv; Defense goal: D = detection, M = mitigation,
? = not available in the paper and we cannot conclude from the design, conf. = confidence, h/w = hardware

Table II. Summary of existing semantic AD AI defense solutions in our SoK scope (§II-C).

driving context. Some works also leverage physical invariant properties of the sensor source such as light reflection [57,67] and LiDAR occlusion patterns [81] to detect AI attacks.

Adversarial robustness improvement. Several defenses try to improve the robustness of the AI component against attacks. For example, Chen et al. [105] applied adversarial training [13] to make the camera object detection model more robust. Jia et al. [106] improved the model robustness by predicting and removing potential adversarial perturbations from the model inputs. Sun et al. [81] augment the point cloud with point-wise confidence scores from a front view segmentation model to improve the robustness of LiDAR object detection.

Status and trends. As shown in Table II, existing defenses share common general defense strategies, with 66.7% (6/9) based on consistency checking and 33.3% (3/9) based on adversarial robustness improvement. We discuss a few possible new directions later in §IV-B.

2) Defense Goals: The defense methodologies in the above section can be naturally mapped to two defense goals: (1) **Detection:** All consistency checking based defenses are designed to detect the attack attempts; and (2) **Mitigation:** Improving the adversarial robustness of models can generally reduce the attack success rate and raise the bar of the attackers. However, it is not designed to detect the attack, and also cannot fundamentally eliminate/prevent AI vulnerabilities.

Status and trends. Among the existing defenses, attack detection and mitigation are the main focus; so far, none of the works aims for other goals such as attack prevention. More discussions on this are in §IV-B.

3) Defense Deployability: In this section, we list the defense design properties that are highly desired for the deployment in practical AD settings:

Negligible timing overhead. Timing overhead is one of the most important factors that may limit the deployability of defense for real-time systems such as AD systems. Among the existing defenses, 4 have negligible or no timing overhead by design or shown in evaluation [57,81,105,107], 1 fails to catch up with the camera and LiDAR frame rate in evaluation [104]. For the remaining ones, we cannot conclude their timeliness from their papers (e.g., no timing overhead evaluation).

Negligible resource overhead. In practice, the hardware that runs the AD system may have limited computation resources (e.g., limited GPUs) due to budget concerns. Among the defenses, except the ones that do not require *extra* ML models [81,89,105], all others will impose additional demand on the computation resources in order to execute the models. This may prevent them from deploying onto vehicles that have already fully utilized the hardware resources.

No model training. The requirement of model training imposes extra deployment burdens. Among the defenses that require extra ML models, only one [107] uses public pre-trained models without fine-tuning.

No additional dataset. Closely related to the previous property, some defenses not only require model training, but also need additional datasets during training process. This will limit the deployability due to the efforts in dataset preparation.

No hardware modification. This property means whether the defense requires changes to the hardware on AD vehicle or adding new hardware (e.g., additional sensors). Among the defenses, Liu et al. [104] require stereo cameras, which may not be available on some AD vehicles.

Status and trends. As shown in Table II, almost all existing defenses (7/8) have the awareness for at least one of these deployability design aspects, especially for that regarding hardware modification (7/8) and timing overhead (4/8). However, the awareness on the need for no model training, negligible resource overhead, and no additional dataset are currently lacking (2/8, 2/8, and 3/8 respectively).

4) Robustness to Adaptive Attacks: Adaptive attacks are designed to circumvent a particular defense. They often assume complete knowledge of the defense internals, including the design and implementation, and are designed to challenge the fundamental assumptions of the defense. Adaptive attack evaluation has become *strongly-advocated* in recent adversarial AI defenses, and prior work has also proposed guidelines on how to properly design adaptive attacks [108]. However, among the AD AI defenses, we find that *only 3 papers* conduct some forms of adaptive attack evaluation [57,81,104]. Specifically, Nassi et al. [57] applied existing adversarial attacks on their defense model and find that it is challenging to circumvent

the attack detection; Liu et al. [104] evaluated simultaneous attacks against their consistency checking design between LiDAR and camera object detection, and find that the detection is still effective. Sun et al. [81] designed adaptive attacks based on the defense assumptions (the LiDAR occlusion patterns), and show that they fail to break the two defenses.

Status and trends. Despite being strongly-advocated in general adversarial AI defense research [108], currently not many (3/8) of the existing defenses in AD AI security conduct evaluation against adaptive attacks. However, with the increasing adoption of such practices in the general adversarial AI domain, this situation on the semantic AD AI security domain should also see improvements.

D. Systematization of Evaluation Methodology

At the evaluation methodology level, besides the expected differences due to problem formulations (e.g., attack targets and goals), we notice an interesting general disparity in the choices of the *evaluation levels*, which is relatively unique for semantic AI security as opposed to generic AI security. Specifically, since here the attack targets are AI components but the ultimate goals are to achieve system-level attack effect (e.g., crashes), the evaluation methodology can be designed at both AI component and AD system levels:

Component-level evaluation is to evaluate attack/defense impacts only at the targeted AI component level (e.g., object detection) without involving (1) other necessary components in the full-stack AD systems (e.g., object tracking, prediction, planning); and (2) the closed-loop control [109]. The evaluation metrics are thus component-specific, e.g., detection rate [18, 26], location deviation [92], steering error [94], etc.

System-level evaluation refers to evaluating the attack/defense impacts at the vehicle driving behavior level considering full-stack AD system pipelines and closed-loop control. The evaluation setups used to achieve this can be generally classified into two categories: (1) *Real vehicle-based*, where a real vehicle is under partial (e.g., steering only [79]) or full (i.e., steering, throttle, and braking [57, 89]) closed-loop control of the AD system to perform certain driving maneuvers with the presence of attacks, which are typically deployed in the physical world as well; and (2) *Simulation-based*, where the critical elements in the closed-loop control (e.g., sensing/actuation hardware and the physical world) are fully or partially simulated, using either existing simulation software [38, 78, 80, 92, 93] or custom-built modeling [92].

Status and trends. As shown in Table I and II, the majority (90.5%) of the surveyed works performed component-level evaluation, likely due to the ease of experiment efforts (no need to set up real vehicle or simulation), while only 25.4% (16/63) adopted some forms of system-level evaluation. More discussions are in §IV-A.

IV. Scientific Gaps and Future Directions

Based on the systematization of existing works on AD AI security, we summarize a list of scientific gaps that we observed and also discuss possible solution directions. To avoid subjective opinions and bias, such observations are drawn from quantitative comparisons *vertically* among the choices made in existing AD AI security works along with different design angles in §III and/or *horizontally* with security works in closely-related CPS (Cyber-Physical Systems) domains such as drone and Automatic Speech Recognition and Speaker Identification (ASR/SI) based on recent SoKs [24, 52].

A. Evaluation: General Lack of System-Level Evaluation

Scientific Gap 1: *It is widely recognized that in AD system, AI component-level errors do not necessarily lead to system-level effect (e.g., vehicle collisions). However, system-level evaluation is generally lacking in existing works.*

As identified in §III-D, currently it is not a common practice for AD AI security works to perform system-level evaluation: overall only 25.4% of existing works perform that, and such a number is especially low (*7.4%*) for the most extensively-studied AI component, camera object detection. In fact, the vast majority (74.6%) of these works *only* performed component-level evaluation without making any efforts to experimentally understand the system-level impact of their attack/defense designs. Admittedly, for CPS systems such system-level evaluation is generally more difficult due to the involvement of physical components. However, we find that for existing security research on related CPS domains such as drone and ASR/SI, actually *almost all* attack works perform system-level evaluations (100% for drone, 94% for ASR/SI based on the SoKs [24, 52]). This shows that the current AD AI security research community is particularly lagging behind in such common evaluation practice in CPS security research.

In the CPS verification 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* [14, 16]. 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 [14, 16], or such errors can be effectively tolerated by downstream AI modules such as object tracking [17]. 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. [17], for camera object detection-only AI attacks, a component-level success rate of up to 98% can still be not enough to affect object tracking results. Thus, we believe that for semantic AD AI security research, the current general lack of system-level evaluation is a critical *scientific methodology-level gap* that should be addressed as soon as possible.

Future directions. Specific to the AD problem domain, there are various technical challenges to effectively fill this gap at the research community level. Among the possible options (§III-D), real vehicle-based system-level evaluation is generally unaffordable for most academic research groups (e.g., $250K per vehicle [110]), not to mention the need for easily-accessible testing facilities, the high time and engineering efforts needed to set up the testing environment, and also

high safety risks (especially since most AD AI attacks aim at causing safety damages). Simulation-based evaluation is much more affordable, accessible, and safe, but researchers still need to spend substantial engineering efforts to instrument existing AD simulation setup for the security-specific research needs, e.g., modifying the simulated physical environment rendering [38, 78] or sensing channels [92] for launching attacks. These might explain why system-level evaluation is not commonly adopted for AD AI security research today.

To address such impasse, it is highly desired if there can be a community-level effort to collectively build a common system-level evaluation infrastructure, since (1) the engineering efforts spent in instrumenting either a real AD vehicle or AD simulation for security research evaluation share common design/implementation patterns (e.g., common attack entry points as in §III-B2); 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 [111] 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, later in §V we take the initiative to lay the groundwork to foster such a community-level effort.

B. Research Goal: General Lack of Defense Solutions

Scientific Gap 2: *Compared to AD AI security attacks, their effective defense solutions are substantially lacking today, especially those on attack prevention.*

As shown in Tables I and II, among existing semantic AD AI security works, the ratio of discovered attacks (85.7%) is much higher than effective defense solutions (14.3%). Furthermore, all existing defenses focus on attack detection and mitigation, and *none* studied *prevention*, which is a more ideal form of defense. In comparison, such ratios are much more balanced in the drone security field [24], where 51% are on attacks and 49% are on defenses. Also, among the defenses, *33%* are for prevention. While attack discovery is the necessary first step for security research into an emerging area, it is imperative to follow up with effective defense solutions, especially those on attack prevention, to close the loop of the discovered attacks and actually benefit the society.

Future directions. As shown in Tables I and II, many of the AD AI components with discovered attacks actually do not have any effective defense solutions yet (e.g., lane detection, MSF perception), which are thus concrete defense research directions. Note that there indeed exist generic AI defenses that are directly applicable (e.g., input transformation [39]), but it is already found that such generic defenses are generally ineffective against the CPS domain AI attacks, e.g., for both AD [38, 78] and ASR/SI [112].

Besides considering the currently-undefended AI components, there are also possibly-applicable defense strategies that have not been explored yet, for example certified robustness [40]. While unexplored (§III-C1), such a defense is actually highly desired in the AD AI security domain since it can provide strong theoretical guarantees for AI security in such a safety-critical application domain. The challenge is that today's certified robustness designs only focus on small 2D digital-space perturbations, and thus their extensions to today's popular AD AI attack vectors (e.g., physical-world or sensor attacks) are still open research problems. This requires addressing at least the system-to-AI semantic gap, not to mention potential deployability challenges as such methods generally have high overhead [113] and thus might be hard to meet the strong real-time requirement in AD context (§III-C3).

C. Attack Vector: Cyber-Layer Attack Vectors Under-Explored

Scientific Gap 3: *Existing works predominately focus on physical-layer attack vectors, which leaves the cyber-layer ones substantially under-explored.*

As shown in Table I, there are only 11.1% (6/54) AD AI attack works that leverage cyber-layer attack vectors in their attack designs, assuming ML backdoors [93], malware [75], remote exploitation [91], and compromised ROS nodes [59], respectively. Among these four, only the last one is relatively domain-specific to AD. In related CPS domains such as drones and ASR/SI, such ratio is much higher (>50%): 53.8% for drone attack works [24] and 52.9% for ASR/SI ones [52]. One observation we have is that many of these works in related CPS domains exploit domain-specific networking channels such as ground station communication channel [114] and First-Person View (FPV) channel [115–117] in drones, audio files, and telephone networks [118] in ASR/SI systems. However, *no existing works in AD AI security* studied such aspects, which can thus be one direction to fill this research gap.

Future directions. The general design goal of AD AI stack is indeed mostly towards achieving autonomy only using onboard sensing; however, this does not mean that there are no networking channels that can severely impact end-to-end driving. For example, at least the following two are domain-specific, widely-adopted in real-world AD vehicles, and highly security and safety critical:

High Definition (HD) Map update channel. For L4 vehicles, the accuracy of HD Map is critical for driving safety as it is the fundamental pillar for almost all AD modules such as localization, prediction, routing, and planning. Real world dynamic factors (e.g., road constructions), make it imperative to keep the HD Map frequently updated, which is typically over-the-air. For example, when a Waymo AD vehicle detects a road construction, it updates the HD Map and shares the map update with the operations center and the rest of the fleet in real time [119].

Remote AD operator control channel. Another channel that is also relevant to high-level AD is the control channel for remote operators. Unlike lower-level AD vehicles, L4 and above do not have safety drivers onboard. However, after an AD vehicle reaches a fail-safe state, a remote operator usually take over the control (e.g., happened to a Waymo vehicle stuck on the road [120]), which is directly safety-critical if hijacked.

D. Attack Target: Downstream AI Component Under-Explored

Scientific Gap 4: *There is a substantial lack of works focusing on downstream AI components (e.g., prediction,*

planning), which are equally (if not more) important compared to upstream ones w.r.t system-level attack effect.

Among the vast majority (50/54) of attack works that target the more practical modular AD system designs (§II-A), so far *none* of them targeted downstream AI components such as prediction and planning; they predominately focus on the upstream ones such as perception and localization. This is understandable since under the most popularly-considered physical-layer attack model (§III-B2), upstream ones can be much more directly influenced by attack inputs (e.g., via physical-world or sensor attacks), while to affect the inputs of downstream ones such as predication, one has to manipulate the upstream component outputs (e.g., object detection) first. However, these downstream ones are actually at least equally (if not arguably more) important than the upstream ones. For example, errors in the obstacle trajectory prediction or path planning will actually more directly affect driving decisions and thus lead to system-level effects. Thus, it is highly desired for future AD AI security research to fill this gap.

Future directions. To study downstream AI component security, a general challenge is how to study their component-specific security properties with sufficient isolation with upstream component vulnerabilities, so that we can isolate the causes and gain more security design insights specific to the targeted downstream component. Here we discuss several possible solution directions:

Physical-layer attacks by road object manipulation. The prediction component in AD systems predicts obstacle trajectory based on its detected physical properties (e.g., obstacle type, dimension, position, orientation, speed). Therefore, assuming upstream components such as AD perception is functioning correctly, the attacker can directly control a road object in the physical world (e.g., an attack vehicle on the road), in order to control the inputs of prediction without relying on vulnerabilities in upstream components. However, this requires the attack design to ensure that the triggers of the prediction vulnerability are *semantically-realizable*, e.g., the attacker-controlled obstacle needs to have a consistent type, no breaks in the historical trajectory, moving at a reasonable speed, etc.

Physical-layer attacks by localization manipulation. The planning component takes the prediction and localization outputs as inputs to calculate the optimal driving path in the near future. Therefore, the change in the localization directly affects the decision-making in the planning. To attack the planning, one can leverage localization-related attack vectors (e.g., GPS spoofing) to control planning inputs. However, this only works for AD systems that purely rely on such input (e.g., GPS) for localization; if MSF-based localization is used, it is still hard to isolate planning-specific vulnerabilities from MSF algorithm vulnerabilities (e.g., [92]).

Cyber-layer attacks. Perhaps the most direct way to manipulate the inputs to the downstream components is via cyber-layer attacks. For example, a compromised ROS node [59] in the AD system can directly send malicious messages to prediction or planning. Unlike the road object manipulation method, this does not require the inputs to be semantically-realizable. This thus forms another motivation to explore more on cyber-layer attacks (§IV-C).

E. Attack Goal: Goals Other Than Integrity Under-Explored

Scientific Gap 5: *Existing attacks predominantly focus on safety/traffic rule violations or mobility degradation (can be viewed as integrity in AD domain), while other important security properties such as confidentiality/privacy, availability, and authenticity are under-explored.*

As shown in §III-B1, almost all (96.3%) existing attacks target the AD-specific manifestation of integrity (i.e., modify the driving correctness). However, the study of other important security properties such as confidentiality and availability [96] is substantially lacking: only 2 (3.7%) touch upon confidentiality and none of them study availability. In comparison, the attack goals in drone security are more balanced: 57.9% (11/19) on integrity/safety, 31.6% (6/19) on privacy/confidentiality, and 88.9% (8/19) on availability. Although integrity is highly important in AD context as vehicles are heavy and fast-moving, these other properties are also equally important from the general security field perspective.

Future directions. To foster future research into these less-explored security properties, here we discuss a few possible new research angles. For confidentiality/privacy, existing efforts only consider AD system-internal information leakage (e.g., location), but one can also consider potential private information extraction from AD system-*external* information such as from sensor inputs. In the drone security area, one main privacy attack scenario is along this direction, e.g., people-spying using the drone camera [121,122]. For availability, since it is closely related to the communication channels among AI components, more extensive exploration on the cyber-layer attacks (§IV-C) may make such attacks feasible. So far, no existing works consider authenticity in AD context, but in today's AD deployment and commercialization, authenticity can manifest as the authentication of the safety drivers, mutual authentication between the passenger and the AD vehicle for robotaxi, and mutual authentication between the consumers and the delivery vehicle for AD goods delivery, which are within the broader scope of AD AI stack.

F. Community: Substantial Lack of Open Sourcing

Scientific Gap 6: *The open-sourcing status of AD AI security works from the security community is much lacking compared to related communities/domains (e.g., CV, ML/AI, ASR/SI), which harms the reproducibility and comparability for this emerging area in the security community.*

For each work in Table I and Table II, we searched for their code or open implementation in the paper and also on the Internet. Overall, there are less than *20.6% (7/34)* papers from security conferences that release source code. The situation is much worse if we narrow it down to the sensor attack works, where only *1 (8.3%) out of 12* works released its attack code. In comparison, the papers published in CV and ML/AI conferences have over 50% open-source percentages. Interestingly, also in the CPS domain, 50% ASR/SI papers in the security conferences release their code, which is roughly the same as in CV and ML/AI conferences. This indicates that it is not that security researchers are not willing to share

code, but more likely related to the *diversity* of AD AI security papers in the security conferences. In fact, security conference papers adopt a much more diverse set of attack vectors than any other conferences as shown in Table I. For example, among the 15 works that use sensor attack vectors, only 2 are from CV conferences and 1 is from ML/AI conferences. For those papers, it is indeed more difficult to share the code or implementation since hardware designs are usually involved. In comparison, the attack vectors in ASR/SI are much more uniform; they predominantly leverage malicious sound waves, which is convenient to modify, and can be evaluated digitally.

Future directions. There is no doubt that we should encourage more open-sourcing efforts from the security community such that future works can benefit from it. However, the challenge is in which form the hardware implementations should be shared such that such sharing can be more directly useful for the community. Here we discuss two possible solution directions based on our observations from prior works:

Open-source hardware implementation references. Since the reproduction cost and effort for hardware implementations are generally higher than software-only ones, it is actually more desired for researchers to release as many details about their hardware design as possible. This often means the circuit diagram, printed circuit board layout, bill of materials, and detailed experiment procedures. One example is the LiDAR spoofing work by Shin et al. [123], where they clearly list such details on a website and thus other groups were able to reproduce and build upon their designs [19].

Open-source attack modeling code. Another interesting trend in recent sensor attack works is that they often model the attack capability in the digital space for large-scale evaluation. For example, Man et al. [58] modeled the camera lens flare effect caused by attacker's light beams in digital images, Ji et al. [65] modeled the image blurry effect caused by adversarial acoustic signals on the camera stabilization system; Cao et al. [19] models the LiDAR spoofing capability as the number of points that can be injected to the point cloud; Shen et al. [92] models GPS spoofing as arbitrary GPS position that can be controlled by the attacker. Since these modelings are either validated in the physical-world experiments or backed up by the literature, sharing such code will greatly ease the reproduction in future works.

Our initiated community-level evaluation infrastructure development effort in §V may also help fill this gap by encouraging open-sourcing practices (§V-A).

V. *PASS*: Uniform and Extensible System-Driven Evaluation Platform for AD AI Security

Among all the identified scientific gaps, the one on the general lack of system-level evaluation (§IV-A) is especially critical as proper evaluation methodology is crucial for valid scientific progress. In this section, we take the initiative to address this critical scientific methodology-level gap by developing an open-source, uniform, and extensible system-driven evaluation platform, named *PASS* (Platform for Autonomous driving Safety and Security).

Method	FD	AF	SF	FX	EF	RP	Papers
Real vehicle-based	●	○	○	○	○	○	[57, 79, 89]
Simulation-based	◐	●	●	●	●	●	[38, 75, 78, 80, 84, 91–93]

FD = fidelity, AF = affordability, SF = safety, FX = flexibility, EF = efficiency, RP = reproducibility; ○ = low, ◐ = medium, ● = high

Table III. System-level evaluation methods used in existing works. Such complementariness motivates us to choose a *simulation-centric hybrid design* for our evaluation platform.

A. Design Goals and Choices

As described in §IV-A, to effectively fill this gap at the research community level, a common system-level evaluation infrastructure is highly desired. To foster this, we take the initiative to build a *system-driven evaluation platform* that unifies the common system-level instrumentations needed for AD AI security evaluation based on our SoK (§III), and abstracts out the customized attack/defense implementation and experimentation needs as platform APIs. Such a platform thus directly provides the aforementioned evaluation infrastructure as it allows semantic AD AI security works to directly plug in their attack/defense designs to obtain system-level evaluation results, such that (1) without the need to spend time and efforts to build the lower-level evaluation infrastructure; and (2) generating results in the same evaluation environment and thus directly comparable to prior works. Note that although some existing works developed simulation-based system-level testing methods for AI-based AD [16, 124–126], their designs are for safety not security (no threat models), and thus cannot readily support the evaluation of different AD AI attack/defense methods.

This platform will be fully open-sourced so that researchers can collectively develop new APIs to fit future attack/defense evaluation needs, and also contribute attack and defense implementations to form a semantic AD AI security benchmark, which can improve comparability, reproducibility, and also encourage open-sourcing (currently lacking in the security community as in §IV-F). We also plan to provide remote evaluation services for research groups that do not have the hardware resources required to run this platform, which will be maintained by the project team.

Simulation-centric hybrid design. To build such a community-level evaluation infrastructure, a fundamental design challenge is the trade-off between real vehicle-based and simulation-based evaluation methodology, which is shown in Table III. Real vehicle-based is more fidel as it has the vehicle, sensors, and physical environment all in the evaluation loop, but simulation is better in all other important research evaluation aspects, ranging from cost, free of safety issues, high flexibility in scenario customization, convenience of attack deployment, much faster evaluation iterations, to high reproducibility. Considering their strengths and weakness, we choose a *simulation-centric hybrid design*, in which we mainly design for simulation-based evaluation infrastructure and only use real vehicles for simulation fidelity improvement (e.g., digital twin [127] and sensor model calibration [127–131]) and exceptional cases where it is easy to set up the physical scenario and maintain safety.

We choose the design to be simulation-centric because we

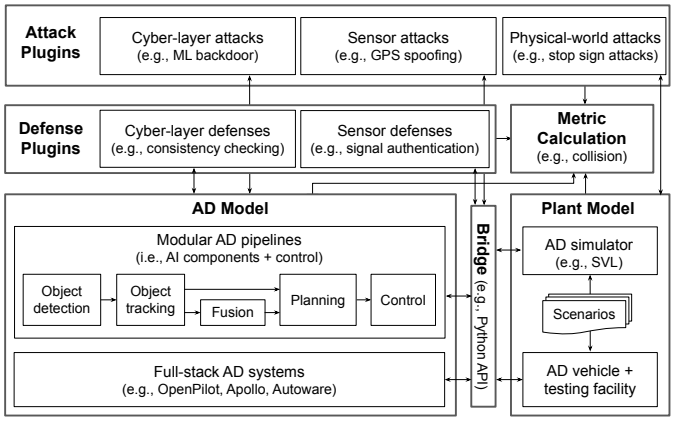

Figure 3: Design of our system-driven evaluation platform *PASS* for the semantic AD AI security community.

Figure 4: AD vehicles for the system-driven evaluation platform: (a) A real-vehicle sized chassis with L4 AD sensors and closed-loop control; (b) A L4 AD vehicle built upon Lincoln MKZ. Both are equipped with L4 AD-grade sensors such as LiDARs, cameras, RADARs, GPS, and IMU. Detailed specifications are on our project website **https://sites.google.com/view/cav-sec/pass** [25].

find that the fidelity drawback of simulation-based approach is tolerable since (1) our results show that for today's representative AD AI attacks, the attack results and characteristics are highly correlated in today's industry-grade simulator and physical world, which indicates sufficient fidelity at least for such AD AI security evaluation purpose (detailed in §V); and (2) the simulation fidelity technology is still evolving as this is also the need for the entire AD industry [132,133], for example recently there are various new advances in both industry [127] and academia [127–131]); and (3) after all the simulation environment can be considered as a different environmental domain, and practical attacks/defenses should be capable of such domain adaptations. Such a design is also more beneficial from the community perspective as it makes the setup of such a platform more affordable for more research groups.

B. PASS Design

Our system-driven evaluation platform *PASS* is guided by two general design rationales: (1) *uniform*, aiming at unifying the common system-level evaluation needs at attack/defense implementation, scenario setup, and evaluation metric levels observed in our SoK (§III); and (2) *extensible*, aiming at making the platform capable of conveniently supporting new future attacks/defenses, AD system designs, and evaluation setups. Specifically, for attacks/defenses on a certain driving task, our platform defines a list of driving scenarios and metrics, which help unify the experimental settings and effectiveness measurements. To enable extensibility for future research, the platform provides a set of interfaces to simplify the deployment of new attacks/defenses. Based on our SoK, we design the attack/defense interfaces to support the common categories of attack vectors (§III-B2) and defense strategies (§III-C1). In addition, the platform provides a modular AD system pipeline with pluggable AI components, which makes the platform easily extensible to different AD system designs.

Fig. 3 shows the *PASS* design, consists of 6 main modules:

1) AD model: The AD model hosts the end-to-end AD system with the targeted AI components under testing. Specifically, the platform provides two AD model choices: modular AD pipeline and full-stack AD system (e.g., OpenPilot [29], Apollo [101], and Autoware [102]). The modular AD pipeline is provided as a flexible option to enable AD systems with replaceable AI components and different system structures. To configure the modular AD pipeline, the user only needs to modify a human-readable configuration file to specify the desired AI components to be included.

2) Plant model: This is the testing vehicle and physical driving environment. As described in §V-A, our platform is mainly built upon AD simulation (e.g., industry-grade ones such as SVL [134]). To configure the simulation environment, we define a set of scenarios to describe the AD vehicle initial position, equipped sensors, testing road, and road objects (e.g., vehicles and pedestrians). The scenario descriptions are also provided as human-readable files for easy modification. Benefited from the general AD model structure, the evaluation can also be performed on a real AD vehicle (e.g., L4-capable vehicles as shown in Fig. 4) under the driving scenarios provided by the testing facility.

3) Bridge: Bridge is the communication channel between the AD and plant models for sensor data reading and AD vehicle actuation. For better extensibility, it supports function hooking for modifying the communication data at runtime.

4) Attack/Defense plugins: These two host the available attacks and defenses, which are built upon the attack/defense interfaces provided by the platform. To ease the deployment of future attacks/defenses, these interfaces are designed to support common attack vectors (§III-B2) and defense strategies (§III-C1). Specifically, 3 general types of interfaces are defined based on the SoK:

- *Physical-world attack interface* enables dynamically loading 2D/3D objects to the simulation environment at arbitrary locations. It covers many physical-world attacks leveraging object texture, shape, and position.
- *Sensor attack/defense interface* enables registering customized sensor data operations, which model the sensor attack or defense behaviors, in the bridge.
- *Cyber-layer attack/defense interfaces* support adding/replacing components in the AD system and serve as versatile interfaces to enable cyber-layer attacks/defenses. For example, ML backdoor attacks can be implemented by replacing an existing ML model with a trojan one. Consistency-checking based defense can be implemented by adding component with consistency checking logic.

Figure 5: Reproduced semantic AD AI attacks on STOP sign hiding: (a) RP2 [18] in simulator, (b) SS [37] in simulator, and (c) SS [37] in physical world.

5) Metric calculation: This is in charge of collecting measurements from all other modules in the platform and calculating the scenario-dependent evaluation metrics. Based on our SoK of attack goals (§III-B1), we include system-level metrics such as safety (e.g., collision rate [75,78]), traffic rule violation (e.g., lane departure rate [78,92]), trip delay, etc. For comprehensiveness we also include component-level metrics (e.g., frame-wise attack success rate [18,26,37]).

Implementation. We have implemented several variations for each module in the platform. The AD model supports modular AD pipelines for STOP sign attack evaluation (Appendix C) and full-stack AD systems (Apollo [101] and Autoware [102]). Our plant model includes an industry-grade AD simulator [134] and real L4-capable AD vehicles (Fig. 4). For bridge, we reuse the ones (Python APIs, CyberRT, and ROS) provided by the AD simulator. Specifically, we modified the AD simulator, bridge, and modular AD pipeline to enable the 3 general attack/defense interfaces mentioned above. For metrics, we implemented collision and STOP sign-related violation checkings. Note that we *do not intend to cover all existing attacks/defenses, and also believe we should not*; our goal (§V-A) is to initiate a community-level effort to *collaboratively* build a common system-level evaluation infrastructure, which is the only way to make such community-level infrastructure support sustainable and up-to-date. We will fully open-source the platform (will be available on our project website [25]) and welcome community contributions of new attack/defense interfaces and implementations.

C. Case Study: System-Level Evaluation on Stop Sign Attacks

Evaluation methodology. In this section, we use *PASS* to perform system-level evaluations for the existing STOP sign attacks to showcase the platform capabilities and benefits. We choose STOP sign attacks because they are the most studied AD AI attack type in the literature (§5) and *none* of them have conducted system-level evaluations (§III-D).

Evaluated attacks. We evaluate the most safety-critical STOP sign AI attack goal: STOP sign *hiding*, which generally leverage malicious object textures to make a STOP sign undetected (§III-B2). We reproduced 3 representative designs: *ShapeShifter (SS)* [37], *Robust Physical Perturbations (RP2)* [18], and *Seeing Isn't Believing (SIB)* [26]. Fig. 5 (a) and (b) show the reproduced adversarial STOP signs. Details are in Appendix B.

AD model configurations. We configure the modular AD pipeline (§V-B) in the platform as a general AD system that includes camera object detection, object tracking, fusion (optional), planning, and control. Specifically, we configure 3 variations of the AD pipeline based on the availability of map and fusion (detailed configurations are in Appendix C).

Driving scenarios. We select a straight urban road with a STOP sign at an intersection shown in Fig. 5 (a) and (b). We evaluate under 5 common local-road driving speeds (10–30 mph with 5-mph step), 3 lighting conditions (sunrise, noon, sunset), and 3 weather conditions (sunny, cloudy, rainy). During evaluation, the malicious STOP sign images are deployed in the simulator via physical-world attack interface (§V-B).

Evaluation metrics. To evaluate system-level attack effectiveness, we select traffic rule related metrics for STOP sign scenarios: *stopping rate* refers to the percentage of successfully stopping (but may already violate the stop line), *violation rate* is the percentage of violating the stop line, *violation distance* is the distance over the stop line if violated. For comparison, we also calculate component-level metrics used in prior works [18,26]: *frame-wise attack success rate*, f_{succ}, is the percentage of frames that are misdetected in different STOP sign distance ranges, *the best attack success rate in n consecutive frames*, $f_{\text{succ}}^{\max(n)}$, is proposed by Zhao et al. [26] as a better metric to measure whether enough success has been accumulated. We use $n = 50$ (instead of 100 [26]) since most of our simulation scenarios complete within 100 frames due to the low driving speeds (i.e., short braking distances).

Result highlights. Our evaluation results show that at the component level, all 3 attacks have very high effectiveness: SS achieves $f_{\text{succ}} > 90\%$ in 10m and $f_{\text{succ}}^{\max(50)}$ close to *100%* in all scenarios; RP2 and SIB both achieve $f_{\text{succ}}^{\max(50)} = 100\%$ when the speed is 10 mph. Both the aggregated attack results and those at specific distance/angle ranges are all very similar to the reported component-level attack effectiveness in the original papers (Appendix B). However, similar to prior security/robustness studies for other AD AI components [14, 17], we find that such high attack success is *far from achieving the system-level attack goals*: RP2 and SIB are not able to cause STOP sign violations at all across all driving scenario combinations; SS can only succeed when the speed is very low (10mph) while failing for all other speeds (15-30mph), which are more common speed ranges for STOP sign roads. More details are in Table IV in Appendix.

Such low system-level attack effectiveness is mainly due to the tracking component in the modular AD pipeline, which were set up using representative parameters recommended by Zhu et al. [135]. Although the attacks are quite effective at close distances, the STOP sign track created before is still alive, so that the AD system is always aware of the STOP sign and keeps committing the stop decisions. For SS with low speed (10 mph), the attack can hide the detection in more consecutive frames, and thus the STOP sign track can be eventually deleted. This thus again validates the non-triviality of the AI-to-system semantic gap, which further demonstrates the necessity of system-level evaluations for semantic AD AI security (§IV-A). Meanwhile, here we are able to experimentally quantify whether and how much the system-level attack effectiveness for a given AD AI attack depends on the driving scenario (i.e., driving speed), which is only made possible with our simulation-centric design that can more flexibly support different driving scenarios, AD designs,

and system-level metrics.

D. Simulation Fidelity Evaluation

Although the simulation fidelity drawback is tolerable since (1) the simulation fidelity is evolving and can be improved by our real vehicle setup and (2) practical attacks/defenses should be capable of domain adaptations (§V-A), we still hope that today's simulation fidelity is readily usable for AD AI security evaluation purposes. In this section, we thus take the STOP sign attack as an example to concretely understand this.

Experimental setup. We use the SS attack, and calculate the similarity between the STOP sign detection results in the physical world and the simulation environment in our platform. The physical-world setup is shown in Fig. 5 (c). Correspondingly, we use a simulation environment with similar road geometry. More details are in Appendix D.

Results. Among 20 physical-world driving traces, we find that an average correlation coefficient r of 0.75 (lowest: 0.62, highest: 0.80), and all correlations are statistically significant ($p < 0.05$). For Pearson's correlation, $r > 0.5$ is considered strongly correlated [136]. This shows that at least for such representative AD AI attack type, today's simulation fidelity is sufficient at least for our AD AI security evaluation purpose.

E. Educational Usage of PASS

Beyond research usage, *PASS* can also be used for educational purposes. We recently organize a Capture The Flag (CTF) event focusing on AD AI security (AutoDriving CTF [137], co-located with DEF CON 29). Specifically, we leverage an earlier version of *PASS* to create simulation-based CTF challenges on GPS/LiDAR spoofing, lane detection attacks, etc. In total, over 100 teams across the globe participated in the CTF. Since *PASS* was provided as a resource for the teams, after the event, we asked the 5 winning teams for feedback on the platform (IRB exempted). Particularly, all 5 teams consider the platform as helpful for solving the challenges. However, due to relatively high resource requirement (e.g., GPUs), 2 teams suggested providing the platform as an online service. This is also the reason why providing remote evaluation services is part of our design goals in §V-A.

VI. Conclusion

In this paper, we perform the first systematization of knowledge of the growing semantic AD AI security research space. We collect and analyze 53 such papers, and systematically taxonomize them based on research aspects critical for the security field. We summarize 6 most substantial scientific gaps based on both quantitative vertically and horizontal comparisons, and provide insights and potential future directions not only at design level, but also at research goal, methodology, and community levels. To address the most critical scientific methodology-level gap, we take the initiative to develop an open-source, uniform, and extensible system-driven evaluation platform *PASS* for the community. We hope that the systematization of knowledge, identified scientific gaps, insights, future direction, and our community-level evaluation infrastructure building efforts can foster more extensive, realistic, and democratized future research into this critical research space.

References

[1] "Baidu To Operate 3,000 Driverless Apollo Go Robotaxis in 30 Cities in 3 Years." https://www.torquenews.com/1/baidu-operate-3000-driverless-apollo-go-robotaxies-30-cities-3-years.

[2] "Waymo has launched its commercial self-driving service in Phoenix - and it's called 'Waymo One'." www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12.

[3] "Baidu Has Been Working on Autonomous Vehicles. It Got the Green Light for Commercial Self-Driving Bus Service in China." https://www.barrons.com/articles/baidu-gets-green-light-for-commercial-self-driving-bus-service-in-china-51618390800.

[4] "Britain's first self-driving shuttle bus hits the streets, but scares passengers away." https://thenextweb.com/news/uk-first-self-driving-shuttle-bus-scares-passengers-away.

[5] "UPS joins race for future of delivery services by investing in self-driving trucks." https://abcnews.go.com/Business/ups-joins-race-future-delivery-services-investing-driving/story?id=65014414.

[6] "Aurora Expands Autonomous Trucking Tests in Texas." https://www.ttnews.com/articles/aurora-expands-autonomous-trucking-tests-texas.

[7] "Nuro can now operate and charge for autonomous delivery services in California." https://tcrn.ch/3mL70PI.

[8] "Tesla Sold 2 Million Electric Cars: First Automaker To Reach Milestone." insideevs.com/news/542197/tesla-sold-2000000-electric-cars/.

[9] "Tesla Autopilot." https://www.tesla.com/autopilot.

[10] "Wall Street Journal: Tesla's Driver-Assistance Autopilot Draws Safety Scrutiny." https://www.wsj.com/articles/teslas-driver-assistance-autopilot-draws-safety-scrutiny-11582672837?mod=article_inline.

[11] "Uber Self-Driving Car Crash: What Really Happened." https://www.forbes.com/sites/meriameberboucha/2018/05/28/uber-self-driving-car-crash-what-really-happened.

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

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

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense," *Computers & Security*, p. 102150, 2021.

[21] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The Security of Autonomous Driving: Threats, Defenses, and Future Directions," *IEEE*, 2019.

[22] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q.-L. Han, "Deep Learning-Based Autonomous Driving Systems: A Survey of Attacks and Defenses," *IEEE Transactions on Industrial Informatics*, 2021.

[23] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and the Way Forward," *IEEE COMST*, 2020.

[24] B. Nassi, R. Bitton, R. Masuoka, A. Shabtai, and Y. Elovici, "SoK: Security and Privacy in the Age of Commercial Drones," in *IEEE S&P*, pp. 73–90, IEEE, 2021.

[25] "Project Website for the SoK of Semantic AD AI Security." https://sites.google.com/view/cav-sec/pass.

[26] 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.

[27] SAE On-Road Automated Vehicle Standards Committee and others, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," *SAE International*, 2018.

[28] "Cadillac Super Cruise." https://www.cadillac.com/world-of-cadillac/innovation/super-cruise.

[29] "Comma AI openpilot." https://github.com/commaai/openpilot.

[30] "Baidu rolls out China's first paid, driverless taxi service." https://techwireasia.com/2021/05/baidu-rolls-out-chinas-first-paid-driverless-taxi-service/.

[31] "Waymo's driverless taxi service can now be accessed on Google Maps." https://techcrunch.com/2021/06/03/waymos-driverless-taxi-service-can-now-be-accessed-on-google-maps/.

[32] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications," *IOT-J*, vol. 5, no. 2, pp. 829–846, 2018.

[33] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE access*, vol. 8, pp. 58443–58469, 2020.

[34] 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.

[35] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A Survey of End-to-End Driving: Architectures and Training Methods," *TNNLS*, 2020.

[36] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *IEEE S&P*, 2017.

[37] 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*, pp. 52–68, Springer, 2018.

[38] 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.

[39] W. Xu, D. Evans, and Y. Qi, "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," in *NDSS*, 2018.

[40] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *IEEE S&P*, 2019.

[41] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated White-box Testing of Deep Learning Systems," in *SOSP*, pp. 1–18, 2017.

[42] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars," in *ICSE*, 2018.

[43] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun, "DLFuzz: Differential Fuzzing Testing of Deep Learning Systems," in *ESEC/FSE*, 2018.

[44] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial Examples for Semantic Segmentation and Object Detection," in *ICCV*, 2017.

[45] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," *arXiv preprint arXiv:1611.01236*, 2016.

[46] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting Adversarial Attacks With Momentum," in *CVPR*, pp. 9185–9193, 2018.

[47] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models," in *AISec*, 2017.

[48] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "Sok: Security and Privacy in Machine Learning," in *EuroS&P*, IEEE, 2018.

[49] N. Akhtar and A. Mian, "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey," *IEEE Access*, 2018.

[50] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial Attacks and Defences: A Survey," *arXiv preprint arXiv:1810.00069*, 2018.

[51] "CSRankings: Computer Science Rankings." http://csrankings.org/.

[52] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "SoK: The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems," in *IEEE S&P*, pp. 730–747, IEEE, 2021.

[53] C. Yan, H. Shin, C. Bolton, W. Xu, Y. Kim, and K. Fu, "SoK: A Minimalist Approach to Formalizing Analog Sensor Security," in *IEEE S&P*.

[54] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles," in *CVPR Workshop of Negative Results in Computer Vision*, 2017.

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

[56] 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.

[57] 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.

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

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

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

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

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

[71] 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.

[72] 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.

[73] K. K. Nakka and M. Salzmann, "Indirect Local Attacks for Context-Aware Semantic Segmentation Networks," in *ECCV*, 2020.

[74] F. Nesti, G. Rossolini, S. Nair, A. Biondi, and G. Buttazzo, "Evaluating the Robustness of Semantic Segmentation for Autonomous Driving against Real-World Adversarial Patch Attacks," in *WACV*, 2022.

[75] 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.

[76] L. Ding, Y. Wang, K. Yuan, M. Jiang, P. Wang, H. Huang, and Z. J. Wang, "Towards Universal Physical Attacks on Single Object Tracking," in *AAAI*, 2021.

[77] X. Chen, C. Fu, F. Zheng, Y. Zhao, H. Li, P. Luo, and G.-J. Qi, "A Unified Multi-Scenario Attacking Network for Visual Object Tracking," in *AAAI*, 2021.

[78] 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.

[79] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, and S. Wu, "Too Good to Be Safe: Tricking Lane Detection in Autonomous Driving with Crafted Perturbations," in *USENIX Security*, 2021.

[80] 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.

[81] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures," in *USENIX Security*, 2020.

[82] J. Tu, M. Ren, S. Manivasagam, M. Liang, B. Yang, R. Du, F. Cheng, and R. Urtasun, "Physically Realizable Adversarial Examples for LiDAR Object Detection," in *CVPR*, 2020.

[83] Y. Zhu, C. Miao, T. Zheng, F. Hajiaghajani, L. Su, and C. Qiao, "Can We Use Arbitrary Objects to Attack LiDAR Perception in Autonomous Driving?," in *ACM CCS*, 2021.

[84] K. Yang, T. Tsai, H. Yu, M. Panoff, T.-Y. Ho, and Y. Jin, "Robust Roadside Physical Adversarial Attack Against Deep Learning in Lidar Perception Modules," in *Asia CCS*, 2021.

[85] Z. Hau, K. T. Co, S. Demetriou, and E. C. Lupu, "Object Removal Attacks on LiDAR-based 3D Object Detectors," 2021.

[86] Y. Li, C. Wen, F. Juefei-Xu, and C. Feng, "Fooling LiDAR Perception via Adversarial Trajectory Perturbation," 2021.

[87] Y. Zhu, C. Miao, F. Hajiaghajani, M. Huai, L. Su, and C. Qiao, "Adversarial Attacks against LiDAR Semantic Segmentation in Autonomous Driving," in *SenSys*, 2021.

[88] T. Tsai, K. Yang, T.-Y. Ho, and Y. Jin, "Robust Adversarial Objects against Deep Learning Models," in *AAAI*, 2020.

[89] Z. Sun, S. Balakrishnan, L. Su, A. Bhuyan, P. Wang, and C. Qiao, "Who Is in Control? Practical Physical Layer Attack and Defense for mmWave-Based Sensing in Autonomous Vehicles," *IEEE Transactions on Information Forensics and Security*, 2021.

[90] J. Tu, H. Li, X. Yan, M. Ren, Y. Chen, M. Liang, E. Bitar, E. Yumer, and R. Urtasun, "Exploring Adversarial Robustness of Multi-Sensor Perception Systems in Self Driving," *arXiv:2101.06784*, 2021.

[91] M. Luo, A. C. Myers, and G. E. Suh, "Stealthy Tracking of Autonomous Vehicles with Cache Side Channels," in *USENIX Security*, 2020.

[92] 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.

[93] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning Attack on Neural Networks," in *NDSS*, 2018.

[94] Z. Kong, J. Guo, A. Li, and C. Liu, "PhysGAN: Generating Physical-World-Resilient Adversarial Examples for Autonomous Driving," in *CVPR*, 2020.

[95] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Attacking Vision-based Perception in End-to-End Autonomous Driving Models," *JSA*, 2020.

[96] W. Stallings, L. Brown, M. D. Bauer, and M. Howard, *Computer Security: Principles and Practice*, vol. 2. Pearson Upper Saddle River, 2012.

[97] Aptiv, Audi, Baidu, BMW, Continental, Daimler, Fiat Chrysler Automobiles, HERE, Infineon, Intel and Volkswagen, "Safety First for Automated Driving." https://www.daimler.com/documents/innovation/other/safety-first-for-automated-driving.pdf, 2019.

[98] M. Bertoncello and D. Wee, "Ten ways autonomous driving could redefine the automotive world," *McKinsey & Company*, vol. 6, 2015.

[99] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors," in *USENIX Security*, pp. 881–896, 2015.

[100] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks," in *EuroS&P*, pp. 3–18, IEEE, 2017.

[101] "Baidu Apollo." https://github.com/ApolloAuto/apollo.

[102] "Autoware-AI." https://github.com/Autoware-AI/autoware.ai.

[103] S. Li, S. Zhu, S. Paul, A. Roy-Chowdhury, C. Song, S. Krishnamurthy, A. Swami, and K. S. Chan, "Connecting the Dots: Detecting Adversarial Perturbations Using Context Inconsistency," in *ECCV*, 2020.

[104] J. Liu and J. Park, ""Seeing is not Always Believing": Detecting Perception Error Attacks Against Autonomous Vehicles," *TDSC*, 2021.

[105] P.-C. Chen, B.-H. Kung, and J.-C. Chen, "Class-Aware Robust Adversarial Training for Object Detection," in *CVPR*, 2021.

[106] S. Jia, C. Ma, Y. Song, and X. Yang, "Robust Tracking against Adversarial Attacks," in *ECCV*, 2020.

[107] C. You, Z. Hau, and S. Demetriou, "Temporal Consistency Checks to Detect LiDAR Spoofing Attacks on Autonomous Vehicle Perception," in *MAISP*, 2021.

[108] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On Adaptive Attacks to Adversarial Example Defenses," *arXiv:2002.08347*, 2020.

[109] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback Control of Dynamic Systems*. 2002.

[110] "What it really costs to turn a car into a self-driving vehicle." https://qz.com/924212/.

[111] "A Policy on Geometric Design of Highways and Streets, 7th Edition," *AASHTO*, 2018.

[112] S. Hussain, P. Neekhara, S. Dubnov, J. McAuley, and F. Koushanfar, "WaveGuard: Understanding and Mitigating Audio Adversarial Examples," in *USENIX Security*, 2021.

[113] C. Xiang and P. Mittal, "DetectorGuard: Provably Securing Object Detectors against Localized Patch Hiding Attacks," in *ACM CCS*, 2021.

[114] A. Luo, "Drones Hijacking," *DEF CON, Paris, France*, 2016.

[115] N. Rodday, "Hacking a Professional Drone," *Black Hat Asia*, 2016.

[116] T. Multerer, A. Ganis, U. Prechtel, E. Miralles, A. Meusling, J. Mietzner, M. Vossiek, M. Loghi, and V. Ziegler, "Low-Cost Jamming System Against Small Drones Using a 3D MIMO Radar based Tracking," in *EURAD*, IEEE, 2017.

[117] E. Deligne, "ARDrone Corruption," *Journal in Computer Virology*, vol. 8, no. 1, pp. 15–27, 2012.

[118] H. Abdullah, M. S. Rahman, W. Garcia, K. Warren, A. S. Yadav, T. Shrimpton, and P. Traynor, "Hear" No Evil", See" Kenansville"*: Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification systems," in *IEEE S&P*, IEEE, 2021.

[119] "The Waymo Driver Handbook: How our highly-detailed maps help unlock new locations for autonomous driving." https://blog.waymo.com/2020/09/the-waymo-driver-handbook-mapping.html.

[120] "Waymo Self-Driving Car Gets Stuck by Cones, Drives Away From Assistance." https://www.vice.com/en/article/y3dv55/waymo-self-driving-car-gets-stuck-by-cones-drives-away-from-assistance.

[121] Y. Wang, H. Xia, Y. Yao, and Y. Huang, "Flying Eyes and Hidden Controllers: A Qualitative Study of People's Privacy Perceptions of Civilian Drones in The US.," *Proc. Priv. Enhancing Technol.*, 2016.

[122] B. Nassi, R. Ben-Netanel, A. Shamir, and Y. Elovici, "Drones' Cryptanalysis-Smashing Cryptography with a Flicker," in *IEEE S&P*, 2019.

[123] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and Dazzle: Adversarial Optical Channel Exploits Against Lidars for Automotive Applications," in *CHES*, pp. 445–467, Springer, 2017.

[124] M. O'Kelly, H. Abbas, and R. Mangharam, "Computer-Aided Design for Safe Autonomous Vehicles," in *RWS*, 2017.

[125] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components," in *IV*, 2018.

[126] T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, H. Ravanbakhsh, M. Vazquez-Chanlatte, and S. A. Seshia, "VerifAI: A Toolkit for the Formal Design and Analysis of Artificial Intelligence-based Systems," in *CAV*, 2019.

[127] "SVL Digital Twin." https://www.svlsimulator.com/docs/digital-twin/gomentum-dtl/.

[128] 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.

[129] K. Nakashima and R. Kurazume, "Learning to Drop Points for LiDAR Scan Synthesis," in *IROS*, IEEE, 2021.

[130] R. Weston, O. P. Jones, and I. Posner, "There and Back Again: Learning to Simulate Radar Data for Real-World Applications," in *ICRA*, pp. 12809–12816, IEEE, 2021.

[131] 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*, pp. 7230–7240, 2021.

[132] "GM Safety Report 2018." https://www.gm.com/content/dam/company/docs/us/en/gmcom/gmsafetyreport.pdf.

[133] "Waymo Safety Report 2021." https://storage.googleapis.com/waymo-uploads/files/documents/safety/2021-08-waymo-safety-report.pdf.

[134] LG, "SVL Simulator: An Autonomous Vehicle Simulator." https://github.com/lgsvl/simulator.

[135] 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*, pp. 366–382, 2018.

[136] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 2013.

[137] "AutoDriving CTF." https://autodrivingctf.org/.

[138] D. Frossard and R. Urtasun, "End-to-end Learning of Multi-sensor 3D Tracking by Detection," in *ICRA*, IEEE, 2018.

[139] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *ECCV*, pp. 641–656, 2018.

[140] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," in *CVPR*, 2017.

[141] P. Biber and W. Straßer, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," in *IROS*, IEEE, 2003.

[142] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[143] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and Precise Vehicle Localization based on Multi-Sensor Fusion in Diverse City Scenes," in *ICRA*, pp. 4670–4677, IEEE, 2018.
[144] "ShapeShifter." https://github.com/shangtse/robust-physical-attack.
[145] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.
[146] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *CVPR*, pp. 7263–7271, 2017.
[147] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing Robust Adversarial Examples," in *ICML*, 2018.
[148] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a Crime: Real and Stealthy Attacks on state-of-the-art Face Recognition," in *ACM CCS*, 2016.
[149] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple Object Tracking: A Literature Review," *AI*, 2020.
[150] "Tesla Autopilot Traffic Light and Stop Sign Control." https://www.tesla.com/ownersmanual/modely/en_eu/GUID-A701F7DC-875C-4491-BC84-605A77EA152C.html.
[151] United States Department of Transportation - Federal Highway Administration, "Manual on Uniform Traffic Control Devices for Streets and Highways - Chapter 2B. Regulatory Signs." https://mutcd.fhwa.dot.gov/htm/2003r1/part2/part2b1.htm.
[152] "Baidu Apollo Estimates Object Distances based on Pinhole Camera Model." https://github.com/ApolloAuto/apollo/blob/v6.0.0/modules/perception/camera/lib/obstacle/transformer/multicue/obj_mapper.cc#L64.
[153] Police Radar Information Center, "Vehicle Acceleration and Braking Parameters." https://copradar.com/chapts/references/acceleration.html.
[154] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing," in *2007 American Control Conference*, pp. 2296–2301, IEEE, 2007.
[155] K. H. Ang, G. Chong, and Y. Li, "PID Control System Analysis, Design, and Technology," *CST*, 2005.

APPENDIX

A. Targeted AI Components

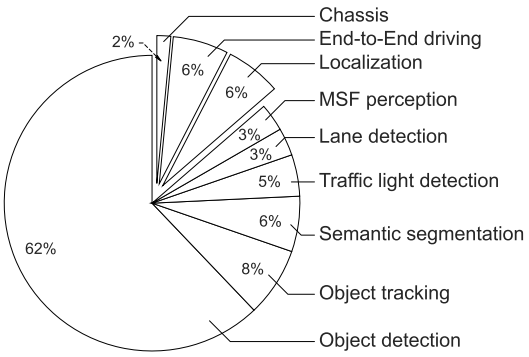

Figure 6: Distribution of (attack/defense) targeted AI components in semantic AD AI security papers.

1) Perception:

Road object detection identifies obstacles (e.g., bounding box), lane lines, and traffic lights from sensor data. The ones targeted by existing semantic AD AI security works includes:

Object detection and *segmentation* are commonly used to detect vehicles, pedestrians, cyclists, traffic objects (e.g., traffic cones), etc., in AD systems. Existing attacks on these aim to cause adversarial effects including *hiding* (i.e., making objects disappear), *creation* (i.e., detecting non-existing objects), and *alteration* (i.e., changing the types of objects).

Lane detection can detect lane line shapes and relative position in lane and is widely used in L2 AD systems for Automated Lane Centering. Prior works [78, 79] show lane detection models are vulnerable to physical perturbations on road, which can lead AD vehicle to drive out of lane line.

Traffic light detection identifies the traffic light and detects the current signal color. Prior works [66, 67, 80] leverages its dependency on localization or light projection to induce wrong detection, no detection, or color change.

MSF perception is commonly used in L4 AD systems [101] to fuse detection results from different perception sources (e.g., LiDAR and camera) for higher accuracy and robustness [138–140]. Prior works found that MSF perception is vulnerable to adversarial 3D objects [38, 90]. Such attacks can hide the adversarial objects or cause other obstacles being misdetected.

Object tracking builds the trajectories of one or more detected objects in a sensor frame sequence to tolerant the occasional false positions and false negatives. However, prior works [17, 76, 77] demonstrate successful adversarial attacks that can move a roadside vehicle into the driving lane or move a front vehicle out of the road [17, 75].

2) Localization:

LiDAR/Camera localizations are commonly used in AD systems for localizing the vehicle. They operate by finding the best matched location of the live LiDAR point cloud or camera image in a map [141, 142]. Prior works discover that such localization algorithms may lead the sensitive location data [91] or can be disrupted by IR lights [67].

MSF localization is predominantly used in L4 AD systems to achieve robust and accurate localization by fusing location sources such as GPS, LiDAR, and IMU [143]. Prior work [92] finds that Kalman Filter based MSF design is vulnerable to strategic attacks leveraging a single attack source, such as GPS spoofing, to inject large deviations in the localization results.

3) Chassis: The Chassis component acts as the information hub of the vehicle, which periodically broadcast privacy-sensitive diagnosis information, including Vehicle Identification Number (VIN), to other components. Prior work [59] demonstrates using a compromised Robot Operating System (ROS) node to intercept the Chassis message to steal the VIN.

4) End-to-End Driving: End-to-End driving [35] is a distinctive AD system design paradigm from the more common modular designs used in industry-grade AD systems [101, 102]. Due to DNN-based design, end-to-end driving is vulnerable to adversarial attacks, which can cause driving model to predict incorrect control commands [64, 94]. Prior work [93] show that trojan attack enables triggering specific driving behaviors (e.g., turn right) using road signs with specific patterns.

B. STOP Sign Attack Reproduction

ShapeShifter (SS) [37] reproduction and results. We directly use the official open-source code [144] to reproduce SS on Faster R-CNN [145]. With that, we can ensure that the attack generated by the source code should have the same effect as the original work. We validated in simulation using the same distance/angle settings as the paper. Results show that our reproduction can achieve 80% (12/15) attack success rate while the original paper reports 73% (11/15). In addition, the successful attack distances/angles of the reproduced attack align well with the paper with the only exception that our reproduction is successful at 25 feet / 0° but the paper reports

failure. Such results also indicate a sufficient simulation fidelity, which is consistent with our fidelity evaluation in §V-D.

Robust Physical Perturbation (RP2) [18] reproduction and results. Same as the paper, we select YOLOv2 [146] as the targeted object detection model. Specifically, we implement the loss function including adversarial loss with Expectation over Transformation [147], total variation loss, l_p loss, and non-printability score loss [148] as used in RP2 design [18]. We compare the attack effectiveness in simulation to the reported results in the paper in the same distance settings (0–30 feet in outdoor environment). Our results show that we can achieve an attack success rate of 66.4% in all camera frames, which is similar to the 63.5% reported in the paper.

Seeing Isn't Believing (SIB) [26] reproduction/modeling. Since SIB is not open-sourced, we tried our best to reproduce it but were still unable to achieve the same attack effectiveness shown in their paper [26]. Nevertheless, in our simulation platform evaluation (§V-C), we apply SIB using a modeling-based approach, i.e., by sampling STOP sign detection failures using *exactly the same* frame-wise attack success rate at different STOP sign distances/angles reported in the paper. Specifically, the attack success rates in the paper are from 32–94% for distances from 5–25 m on YOLOv3 [26].

C. Example AD and Plant Model Setups in PASS

AD model setup. We illustrate the detailed AD model setup for the STOP sign attack evaluation (§V-C) as an example to demonstrate the usage of *PASS*. Specifically, we adopt the typical modular AD pipeline design including object detection, object tracking, fusion, planning, and control.

Object detection. We use the same models and detection thresholds as originally used in the STOP sign attacks: SS on Faster R-CNN (threshold: 0.3) [37], RP2 on YOLOv2 (threshold: 0.4) [18], and SIB on YOLOv3 (threshold: 0.4) [26].

Object tracking. We adopt a general Kalman Filter based multi-object tracker [149] which tracks the STOP sign locations and sizes. Next, we convert the detected STOP sign location from 2D image coordinates to real-world coordinates. Here, we adopt 2 variants: (1) HD Map based (denoted *Map*). Similar to the design in Tesla Autopilot [150], we use the GPS position to query the HD Map for the distance to the front intersection and use it as an approximation for the STOP sign distance. (2) Pinhole camera based (denoted *Pinhole*). We apply the pinhole camera model to estimate the STOP sign distance based on the heuristics on standard STOP sign sizes [151], which is a similar design to Apollo [152]. For track management, a new STOP sign track is created when the STOP sign is detected for 0.2 $\times$ FPS = 4 (frames per second) frames, and an existing track will be deleted after misdetection for 2 $\times$ FPS = 40 consecutive frames (parameters recommended by Zhu et al. [135]).

Fusion. The fusion component is optional and thus can be disabled in the pipeline. If enabled, it can fuse object detection results from multiple data sources. In addition, if the HD Map contains detailed static road objects (e.g., traffic signs) it can serve as a data source to fuse with object detection model outputs. Since the output format of all fusion sources are consistent (i.e., objects), we extend the object tracker to accept detections from multiple sources to facilitate the fusion.

Planning. We adopt a lane following planner where the future trajectory locates at the center of current driving lane [101]. The planner by default maintains a desired speed, but reduces it if *a STOP sign presents* or *there exists a front obstacle with a close distance or slowing down*. Taking the STOP sign as an example, we first calculate a *braking distance* based on the current driving speed and the safe vehicle deceleration ($-3.4m/s^2$ [153]). If a STOP sign is currently tracked and its distance to the vehicle reaches the braking distance, the planner starts to decelerate. If the STOP sign is no longer tracked and current speed is smaller than the desired driving speed (i.e., the speed that the vehicle intends to maintain if no STOP sign), the planner accelerates the vehicle.

Control. The control module will execute the planning decisions, and we use the classic Stanley controller [154] for lateral control (i.e., steering) and a Proportional-Integral-Derivative (PID) controller [155] for longitudinal control (i.e., throttling and braking). Consistent with existing L2 AD systems such as OpenPilot [29], We set the frequency of detection, tracking, fusion, and planning as 20 Hz, and control as 100 Hz.

In our evaluation (§V-C), We adopt 3 AD pipeline variants based on the availability of HD Map and fusion: *Map*, *Pinhole*, and *Fusion*. Specifically, *Map* and *Pinhole* vary in the STOP distance estimation, and *Fusion* refers to fusing STOP sign detection with the one from the HD Map.

Plant model setups. *PASS* supports two options for the plant model. For simulation-based evaluation, we adopt the SVL [134], which is an industry-grade AD simulator. For real-world experiments, we can use the two L4-capable AD vehicles in our project team as shown in Fig. 4, where they are equipped with AD-grade sensors including multiple short- and long-range cameras, LiDARs (16-line, 32-line, and 64-line), mmWave RADAR, duel-antennas GPS with RTK, high-precision IMU, etc. As described in §V-A, the platform is simulation-centric; it is mainly for simulation-based evaluation and only use the real vehicles for simulation fidelity improvements and physical world evaluation in exceptional cases.

D. Setup for Simulation Fidelity Evaluation

In the physical-world experiments, we print the adversarial STOP sign and overlay it on a portable STOP sign as shown in Fig. 5 (c). We have obtained the permit from our institute's transportation department to reserve a dead-end single-lane road for controlled experiments. Correspondingly, we use a simulation environment with similar road geometry. In both settings, the vehicle drives at 10 mph from 300 feet until passes the STOP sign. In the physical-world setting, we use an iPhone 11 mounted on the vehicle windshield to record the driving traces. We collect a total of 20 driving traces with different STOP sign placements in the physical world. Since simulation results are generally quite stable, we only collect one trace in the simulation. We then run object detection on each camera frame to obtain the STOP sign detection confidences in the driving traces. To ensure the traces are synchronized and comparable, we trim and interpolate confidences in the physical-world traces based to a 20 Hz frequency same as in the simulation. We then calculate the Pearson's correlation between the simulation trace and *each* physical-world trace.

Attack	AD Pip	Spd. (mph)	L	W	Component-level metrics				System-level metrics		
					f_{succ}			$f_{\text{succ}}^{\max(50)}$	Stop	Vio.	Vio.
					<10m	<20m	<30m		rate	rate	dist.
SS [37]	Map	10	N	S	91.3%	78.8%	67.4%	100.0%	**0.0%**	**100.0%**	∞
		15			99.8%	82.2%	60.4%	99.8%	100.0%	0.0%	0 m
		20			100.0%	87.8%	68.6%	100.0%	100.0%	0.0%	0 m
		25			100.0%	88.0%	69.2%	100.0%	100.0%	0.0%	0 m
		30			100.0%	88.7%	71.2%	98.0%	100.0%	0.0%	0 m
		25	N	C	100.0%	88.8%	69.8%	100.0%	100.0%	0.0%	0 m
				R	99.3%	86.7%	68.3%	99.0%	100.0%	0.0%	0 m
			SR	S	100.0%	83.9%	66.0%	100.0%	100.0%	0.0%	0 m
			SUS		100.0%	88.0%	69.2%	100.0%	100.0%	0.0%	0 m
	Pinhole	10	N	S	91.3%	78.8%	67.5%	100.0%	**0.0%**	**100.0%**	∞
		15			100.0%	82.5%	60.6%	100.0%	100.0%	0.0%	0 m
		20			100.0%	87.8%	68.6%	100.0%	100.0%	0.0%	0 m
		25			100.0%	88.5%	69.5%	100.0%	100.0%	0.0%	0 m
		30			100.0%	85.4%	68.6%	97.1%	100.0%	0.0%	0 m
	Fusion	10	N	S	100.0%	86.1%	73.8%	100.0%	100.0%	0.0%	0 m
		15			100.0%	82.3%	60.4%	100.0%	100.0%	0.0%	0 m
		20			100.0%	87.8%	68.6%	100.0%	100.0%	0.0%	0 m
		25			100.0%	89.8%	70.6%	99.8%	100.0%	0.0%	0 m
		30			100.0%	88.7%	71.2%	97.5%	100.0%	0.0%	0 m
RP2 [18]	Map	10	N	S	78.0%	51.8%	46.3%	100.0%	100.0%	0.0%	0 m
		15			73.4%	46.9%	46.0%	84.5%	100.0%	0.0%	0 m
		20			76.1%	54.8%	52.2%	88.0%	100.0%	0.0%	0 m
		25			61.6%	42.1%	43.6%	54.5%	100.0%	0.0%	0 m
		30			70.0%	49.8%	45.8%	59.6%	100.0%	0.0%	0 m
		25	N	C	56.7%	38.6%	36.8%	50.0%	100.0%	0.0%	0 m
				R	66.6%	49.4%	54.2%	65.5%	100.0%	0.0%	0 m
			SR	S	59.5%	44.5%	54.5%	71.6%	100.0%	0.0%	0 m
			SUS		63.8%	48.3%	54.6%	69.0%	100.0%	0.0%	0 m
	Pinhole	10	N	S	78.0%	51.8%	46.3%	100.0%	100.0%	0.0%	0 m
		15			73.6%	47.0%	46.4%	85.1%	100.0%	0.0%	0 m
		20			77.7%	56.8%	53.9%	91.4%	100.0%	0.0%	0 m
		25			60.9%	42.4%	44.0%	54.9%	100.0%	0.0%	0 m
		30			74.5%	53.8%	49.1%	64.3%	100.0%	0.0%	0 m
	Fusion	10	N	S	78.0%	51.8%	46.3%	100.0%	100.0%	0.0%	0 m
		15			73.4%	46.9%	46.0%	84.5%	100.0%	0.0%	0 m
		20			78.0%	56.1%	53.3%	90.2%	100.0%	0.0%	0 m
		25			66.3%	46.2%	47.0%	59.8%	100.0%	0.0%	0 m
		30			76.2%	55.1%	49.6%	65.9%	100.0%	0.0%	0 m
SIB [26]	Map	10	N	S	74.7%	57.4%	46.7%	100.0%	100.0%	0.0%	0 m
		15			45.3%	28.9%	21.2%	47.1%	100.0%	0.0%	0 m
		20			73.7%	59.8%	50.3%	100.0%	100.0%	0.0%	0 m
		25			76.7%	66.2%	58.9%	100.0%	100.0%	0.0%	0 m
		30			83.4%	74.1%	67.8%	100.0%	100.0%	0.0%	0 m
	Fusion	10	N	S	74.7%	57.4%	46.7%	100.0%	100.0%	0.0%	0 m
		15			45.3%	28.9%	21.2%	47.1%	100.0%	0.0%	0 m
		20			73.7%	59.8%	50.3%	100.0%	100.0%	0.0%	0 m
		25			76.7%	66.2%	59.0%	100.0%	100.0%	0.0%	0 m
		30			80.6%	70.4%	63.8%	97.5%	100.0%	0.0%	0 m

SR, N, SUS: sunrise, noon, sunset (6 am, 12 pm, 18 pm), C: cloudy (SVL cloudiness = 0.5), R: rainy (SVL rain = 0.5);
AD Pip: AD pipeline variants, Spd.: speed, L: lighting, W: weather, vio.: violation, dist.: distance;
$f_{\text{succ}}^{\max(50)}$: best attack success rate in 50 consecutive frames;
f_{succ}: frame-wise attack success rate.

Table IV. Component- and system-level evaluation results of the 3 stop sign AI attacks on our evaluation platform. The results are averaged over 10 runs of each configuration. Detailed AD pipeline setup can be found in Appendix C. We do not apply the lighting and weather variations for SIB since the object detection outputs are directly modeled and thus will not be influenced by them (Appendix B).