

# Poster: Security of Deep Learning based Lane Keeping Assistance System under Physical-World Adversarial Attack

Takami Sato\*, Junjie Shen\*, Ningfei Wang, Yunhan Jack Jia<sup>†</sup>, Xue Lin<sup>‡</sup>, and Qi Alfred Chen  
University of California, Irvine; <sup>†</sup>ByteDance AI Lab; <sup>‡</sup>Northeastern University

**Abstract**—Lane-Keeping Assistance System (LKAS) is convenient and widely available today, but also extremely security and safety critical. In this work, we design and implement the first systematic approach to attack real-world DNN-based LKASes. We evaluate our approach on a state-of-the-art LKAS and our preliminary results show that our attack can successfully cause it to drive off lane boundaries within as short as 1.3 seconds.

## I. INTRODUCTION

Lane-Keeping Assistance System (LKAS) is an Level-2 driving automation technology that automatically steers a vehicle to keep it within the current traffic lane. Due to its high convenience for human drivers, today it is widely available in a variety of vehicle models such as Honda Civic, Toyota Prius, Nissan Cima, Volvo XC90, Mercedes-Benz C-class, Audi A4, and Tesla Model S. While convenient, such function is extremely security and safety critical: When LKAS starts to make wrong steering decisions, an average driver reaction time of 2.3 seconds may not be enough to prevent the vehicle from colliding into vehicles in adjacent lanes or in opposite directions, or driving off road to hit road curbs or fall down the highway cliff. Even with collision avoidance systems, it cannot prevent the vehicle from hitting the curb, falling down the highway cliff, or being hit by other vehicles that fail to yield. Thus, it is urgent and highly necessary to understand the security property of LKAS.

To achieve lane keeping, the most critical step in an LKAS is lane detection, which by default uses camera due to the nature of lane lines. So far, Deep Neural Network (DNN) based detection achieve the state-of-the-art accuracy and is adopted in the most performant LKASes today such as Tesla Autopilot and OpenPilot [1]. Thus, the end-to-end security of the latest LKAS technology highly depends on the security of such DNN models. While recent works show that DNN models are vulnerable to carefully crafted input perturbations, their methods cannot be directly applied to attack DNN-based LKASes due to 3 unique challenges. First, prior methods are mostly designed for classification or object detection, and none of their attack formulations can be directly applied for lane detection. Second, to affect the camera input of an LKAS, the perturbations need to be realizable in the physical world and can normally appear on traffic lane regions. Moreover, such perturbations must not affect the original human-perceived lane information for stealthiness. Prior works have explored such threats for traffic signs [2], but not for traffic lanes.

Third, to cause end-to-end impact to an LKAS, the attack needs to affect a sufficient number of consecutive camera

frames, and most importantly, the attacks on later frames are dependent on those on earlier frames. For example, if the attack successfully deviates the detected lane to the right in a frame, the LKAS will control the vehicle heading accordingly, which causes the following frames to capture road areas more to the right and thus directly affect their attack generation. To the best of our knowledge, no prior work considers attacking a sequence of image frames with such strong inter-dependencies.

The only prior effort that successfully attacked an LKAS is from Tencent [3], where they fooled the Tesla DNN-based LKAS to follow fake lane lines created by a line of big white dots on road regions originally without lane lines. However, it is neither attacking the scenarios where an LKAS is designed for, i.e., roads with lane lines, nor generating the perturbations systematically by addressing all the three challenges above.

To fill this critical research gap, in this work we design and implement the first systematic approach to attack real-world DNN-based LKASes. To practically introduce perturbations, we identify road patches as the threat model, which is specific to lane detection models and can normally appear in the physical world. For stealthiness, we restrict the perturbations to be within lane lines, and the color space to be on the gray scale to pretend to be a benign but dirty road patch. We then formulate the malicious road patch generation as an optimization problem, and design a multi-frame path bending objective function specifically for the lane detection task. To address the challenge from the inter-dependencies among attacks on consecutive camera frames, we design a novel car motion model based input generation process and a gradient aggregation technique. We evaluate our approach on a state-of-the-art LKAS, OpenPilot, and our preliminary results show that our attack can successfully deviate an LKAS to drive off the lane boundaries within as short as 1.3 seconds, which is far shorter than 2.3 seconds, the average driver reaction time.

## II. THREAT MODEL AND PROBLEM FORMULATION

**Threat model.** We assume that the attacker can possess the same LKAS as the one in the victim vehicles and has the full knowledge of the LKAS via reverse engineering. Before attacking, the attacker can also collect camera frames on the target road by driving her own vehicle with the LKAS.

**Realizable and stealthy physical-world perturbation design.** We identify malicious road patches as the attack vector, since they are realizable in the physical world and can normally appear around traffic lanes. For stealthiness, we restrict these road patches to not cover the original lane lines and their color to be on the gray scale to pretend to be benign but dirty.

---

\*The first two authors contributed equally.

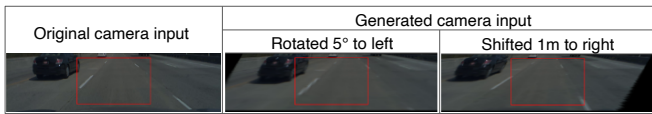


Fig. 1. Car motion model based camera input generation from the original camera input. The red rectangle denotes the model input area.

**Attack goal.** By attacking the LKAS, we aim to cause the victim car to have a lateral deviation large enough to drive out of the current lane boundaries within the common driver reaction time, which thus fundamentally breaks the design goal of LKAS and can cause severe safety consequences. Assuming the victim vehicle locates at the lane center before the attack, the required deviation is 0.745 meters on the highway in the US and the average driver reaction time is 2.3 seconds.

### III. ATTACK METHODOLOGY

With the problem formulation above, we design the following novel techniques to address the challenges in §I.

**Car motion model based input generation.** To consider the inter-dependencies among attacks on consecutive camera frames, we need to dynamically update camera inputs according to the driving trajectory changes during the patch generation. To address this, we use a bicycle model to simulate the changes to car trajectory, which is then used to update camera inputs by applying perspective transformations to the original non-attacked camera inputs. Fig. 1 shows an example of this generation process. On the bird’s eye view (BEV), we apply a car position shift and heading angle change from the original car trajectory and then project the BEV image back to the camera perspective. Although it causes some distortion and partial missing area, the model input area, which locates at the center, is still complete and usable.

**Multi-frame path bending objective function.** To generate the malicious road patch, we adopt an optimization-based method, which has shown both high efficiency and effectiveness in previous works. Since the lateral controller of the LKAS is not differentiable, we introduce a surrogate objective function to deviate the car as much as possible. The lateral controller calculates a *desired driving path* based on the detected lane lines, and numerically solves a steering angle plan to enforce this path. The desired driving path is typically represented by a polynomial function. Assuming the car strictly follows the path, the derivatives of the path are essentially the wheel angles it needs to apply. Thus, we formulate our objective function:

$$f(X_1, \dots, X_T, s_0) = \sum_{t=1}^T \sum_{d \in D} \nabla p_t(d; \{X_j | j \leq t\}, s_0) + \lambda \|\Omega_t(X_t)\|_2^2 \quad (1)$$

, where  $p_t(x)$  is the desired driving path in the  $t$ -th frame,  $X_t$  is the  $t$ -th generated camera inputs including the malicious road patch,  $s_0$  is the initial state and  $D$  is the set of the points where the controller makes steering angle decisions,  $\lambda$  is a weight of the L2 regularization, and  $\Omega$  is a function that extracts the patch area in a camera input.

**Gradient aggregation.** Based on the objective function, we obtain the gradients of each camera input. However, the gradient descent is not directly applicable to update the malicious road patch since the patch sizes and portions are different in each camera input. To address this, we transform

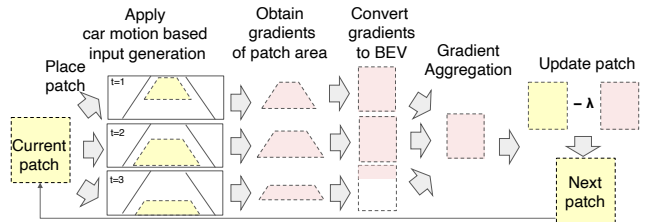


Fig. 2. Overview of the optimization pipeline of the malicious road patch.

TABLE I. ATTACK EFFECTIVENESS WHEN THE DEVIATION GOAL IS 0.745 METERS (DRIVING OFF LANE BOUNDARIES ON THE HIGHWAY).

Scenario	Avg. Speed	Attack Time	Patch Size (W × L)
comma2k19-1	126 km/h (78 mph)	0.9 s	3.6 m × 36 m
comma2k19-2	105 km/h (65 mph)	1.0 s	3.6 m × 36 m
LGSVL-1	72 km/h (45 mph)	1.3 s	3.6 m × 36 m

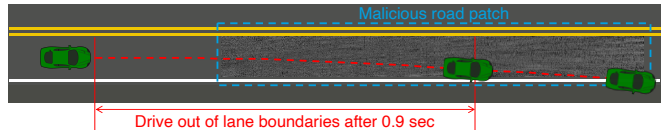


Fig. 3. Malicious road patch and car trajectory for comma2k19-1.

all camera inputs to BEV to align gradients to the same scale and take a weighted average as shown in Fig. 2.

### IV. EARLY RESULTS

We evaluate our method on a state-of-the-art open-source LKAS, OpenPilot [1], which is reported to have similar performance as Tesla Autopilot and GM Super Cruise, and better than all other manufacturers. We evaluate our method on 3 scenarios and the results are summarized in Table I. The comma2k19-1 and comma2k19-2 are real-world highway scenarios selected from the comma2k19 dataset. The LGSVL-1 is a simulated highway scenario created by LGSVL, an industry-grade photo-realistic Autonomous Driving simulator. As shown, our attack succeeds to cause the victim vehicle to drive out of the highway lane boundaries (over 0.745 meters deviations) within 1.3 seconds, which is much smaller than the average driver reaction time (2.3 seconds). Fig. 3 shows an example malicious road patch generated by our method.



### V. CONCLUSION AND FUTURE PLANS

In this work, we design and implement the first systematic approach to attack real-world DNN-based LKASes. We evaluate our approach on a state-of-the-art LKAS and our preliminary results show that our attack can successfully cause it to drive off lane boundaries within as short as 1.3 seconds. In the future, we plan to (1) perform more comprehensive evaluation including more diverse scenarios, different car types, other DNN lane detection models, (2) demonstrate the attack in real-world experiments, and (3) design effective defenses.

### REFERENCES

- [1] “OpenPilot: Open Source Driving Agent,” <https://github.com/commaai/openpilot>, 2018.
- [2] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, “Robust Physical-World Attacks on Deep Learning Visual Classification,” in *CVPR*, 2018.
- [3] “Experimental Security Research of Tesla Autopilot,” [https://keenlab.tencent.com/en/whitepapers/Experimental\\_Security\\_Research\\_of\\_Tesla\\_Autopilot.pdf](https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf), 2019.

# Security of Deep Learning based Lane Keeping Assistance System under Physical-World Adversarial Attack

PRESENTERS  
 Takami Sato  Junjie Shen   
 takamis@uci.edu junjies1@uci.edu

- RESEARCH PROBLEM & MOTIVATION**
- Lane-Keeping Assistance System (LKAS)**
    - Automatically steers vehicle to keep it in lane
    - Level-2 driving automation
    - Widely available in a variety of vehicle models
      - Honda Civic, Toyota Prius, Nissan Cima, Volvo XC90, Audi A4, and Tesla Model S
  - When LKAS fails to keep in lane:
    - Avg. driver reaction time 2.3 s not enough for**
      - Drive off-road to hit road curbs or fall down highway cliff
      - Crash into other cars or be crashed into
  - Our work:**
    - First systematic approach to LKAS
    - Target most performant design today: DNN-based LKAS

- THREAT MODEL**
- Attacker possesses the same LKAS as victim
  - Has full knowledge of the LKAS
  - Can collect road images before attack

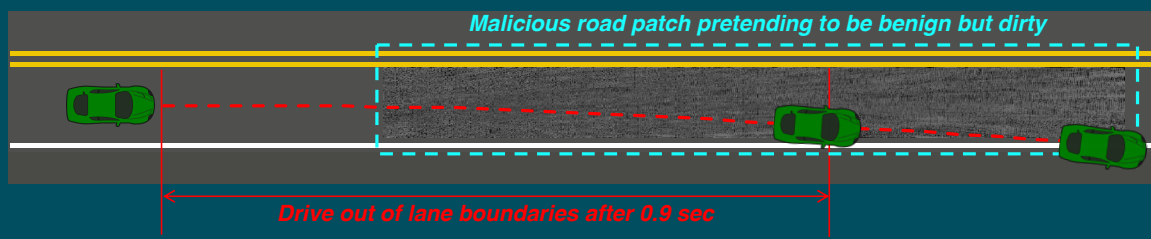
- ATTACK GOAL**
- Drive out of current lane boundary within 2.3 seconds (avg. driver reaction time)
  - The required deviation is 0.745 m on highway

- DESIGN CHALLENGES**
- Most prior attack targets obj. detection, **not LKAS**
  - Need to be **realizable** & **stealthy** in physical world
  - Attack to consecutive frames are inter-dependent

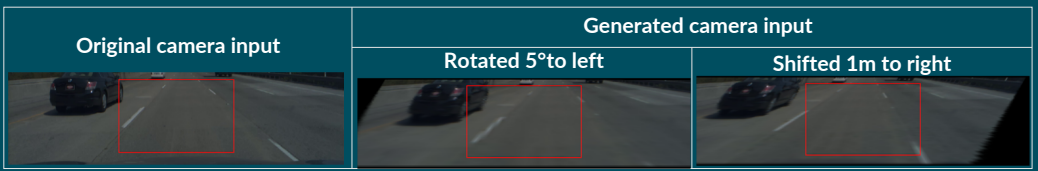
**REALIZABLE & STEALTHY PHYSICAL-WORLD PERTURBATION DESIGN**

- New attack vector: malicious dirty road patch**
  - Realizable in the physical world
  - Can normally appear around traffic lane
  - Not cover the original lane lines
  - Only use gray-scale color to pretend to be benign but dirty

# Our attack can cause vehicles running state-of-the-art LKAS to drive off highway lane after 0.9 sec.



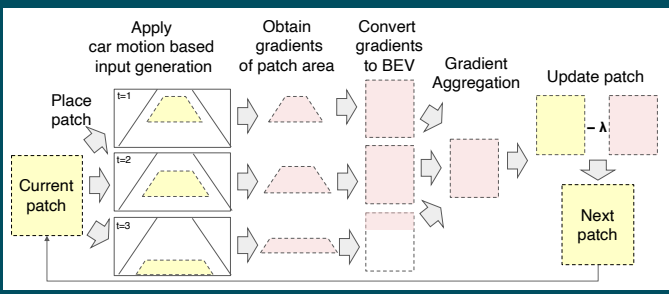
## Car Motion Model based Input Generation



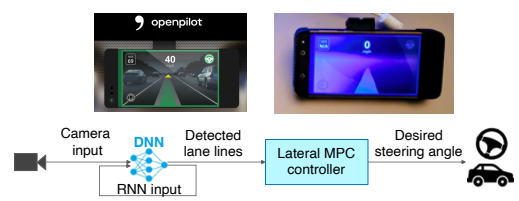
## Multi-frame Path Bending Objective Function

$$f(X_1, \dots, X_T, s_0) = \sum_{t=1}^T \sum_{d \in D} \nabla p_t(d; \{X_j | j \leq t\}, s_0) + \lambda \|\Omega_t(X_t)\|_2^2$$

## Gradient Aggregation



- ATTACK METHODOLOGY**
- Car Motion Model based Input Generation**
    - Simulate inter-dependency by bicycle model
  - Multi-frame Path Bending Objective Function**
    - Design attack as optimization problem considering inter-dependency
  - Gradient Aggregation**
    - Update malicious road path while keeping inter-dependency, realizability, and stealthiness
- EVALUATION SETUP**
- SCENARIOS**
    - Comma2k19 dataset: real-world highway driving
    - LGSVL: industry-grade driving simulator
  - TARGET LKAS: OpenPilot**
    - Open source, on par w/ Tesla & GM Super Cruise



- EARLY RESULTS**
- Success criteria: car deviates 0.745 m within 2.3 s

Eval. Scenario	Avg. Speed	Attack Success Time	Patch Size
comma2k19-1	126 km/h (78 mph)	0.9 s	3.6m × 36m
comma2k19-2	105 km/h (65 mph)	1.0 s	3.6m × 36m
LGSVL-1	72 km/h (45 mph)	1.3s	3.6m × 36m

- FUTURE PLANS**
- More comprehensive evaluation
  - Real-world experiment
  - Design effective defenses
- Takami Sato\*, Junjie Shen\*, Ningfei Wang, Yunhan Jack Jia, Xue Lin, Qi Alfred Chen**  
 \*Contributed equally

**UC Irvine**  
 ByteDance 字节跳动 Northeastern University

Take a picture to download the poster abstract