I ILLINOIS

CSL | Coordinated Science Lab

COLLEGE OF ENGINEERING

Saurabh Jha, **Subho S. Banerjee**, James Cyriac, Zbigniew T. Kalbarczyk and Ravishankar K. Iyer

Computer Science, Electrical and Computer Engineering

AVFI: Fault Injection for Autonomous Vehicles

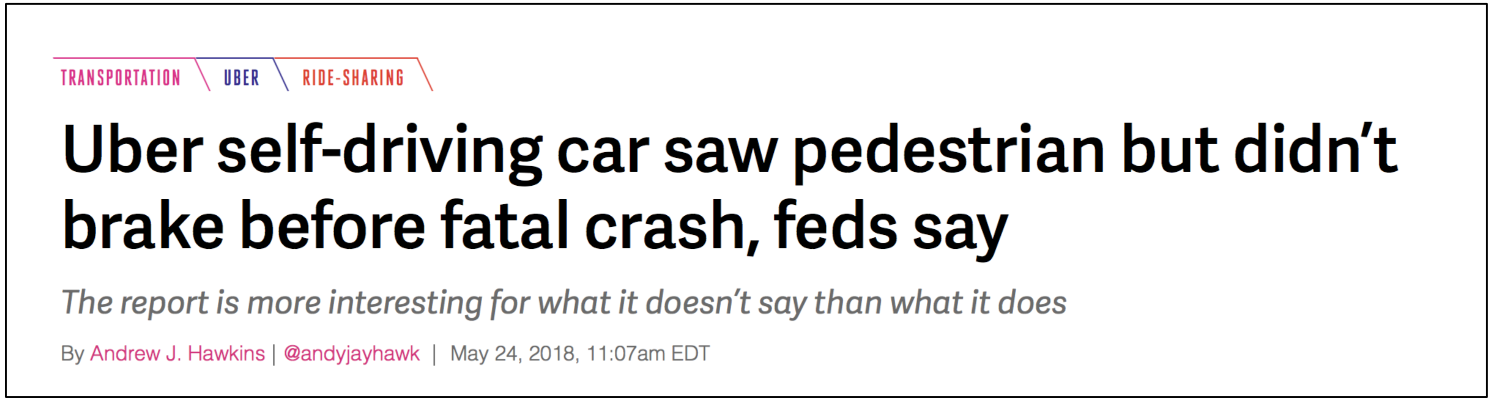
csl.illinois.edu

Fault Injection to Measure Resilience of AVs

- Recent media attention on Tesla/Waymo/Uber AVs

- Resilience and Safety characteristics vary across computing kernels and computing systems

- **Research Gap: Methods to Assess End-to-End Resilience of AVs not available**



TRANSPORTATION \ UBER \ RIDE-SHARING \

Uber self-driving car saw pedestrian but didn't brake before fatal crash, feds say

The report is more interesting for what it doesn't say than what it does

By Andrew J. Hawkins | @andyjayhawk | May 24, 2018, 11:07am EDT

Safety and Reliability Issues **[Banerjee et al., DSN 2018]**

- **Data and Machine Learning:** 64% of reports were problems in the machine learning system (perception, control)

- **Compute system-related**: 30% or more due to failures in computing stack (e.g., watchdogs, networks)

- **Human in the loop:** Human in the loop systems (driver + other cars), have to anticipate the other actors on the roads

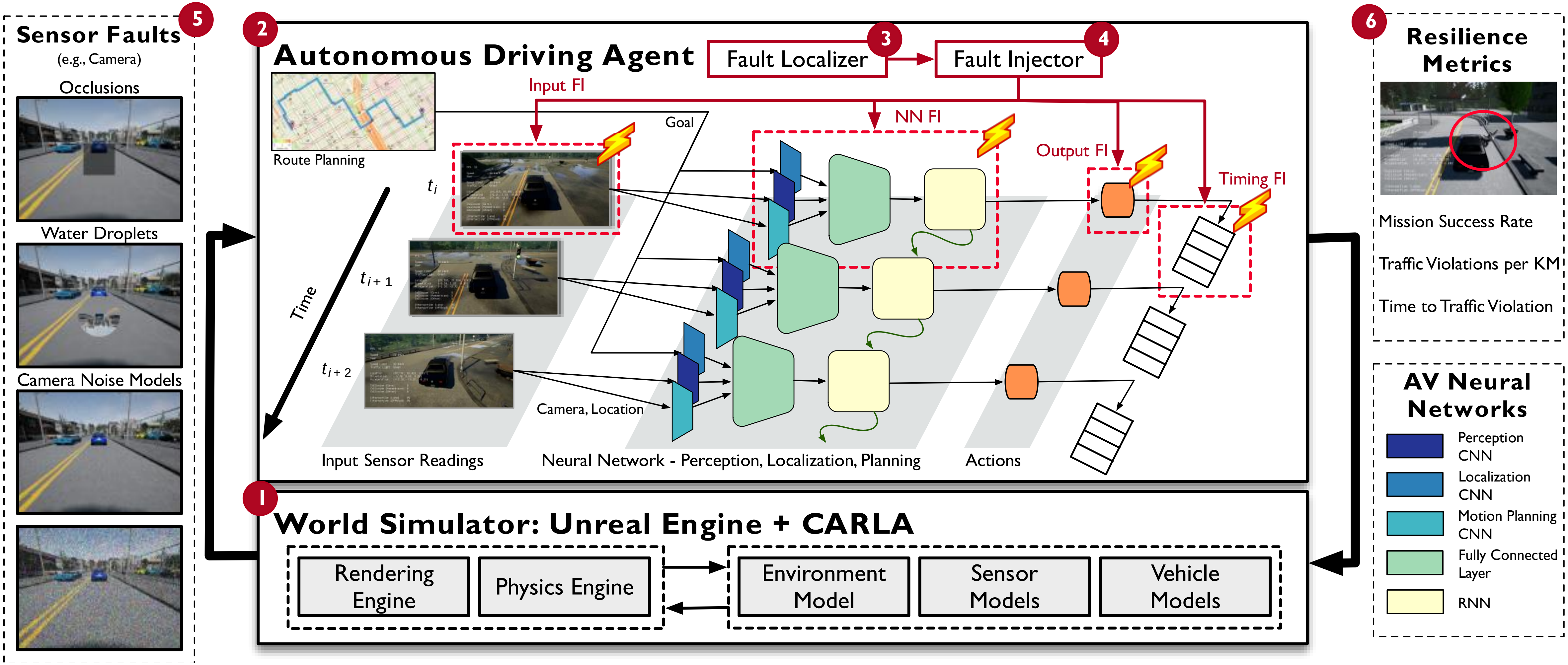
Challenges

- Heterogeneity of system components makes this a challenging problem
    - Complex integration of Sensors, ML, Actuators, Mechanical Components
    - Significant heterogeneity in AV systems: Bayesian Learning, DNNs…

- Interplay between uncertainty at system level: **HW/SW faults** & **algorithmic faults** (ML prediction errors)
    - Unknown Inputs and Inaccuracies in ML predictions
    - Data faults vs Hardware faults

- No robust resilience metrics: Understanding propagation and masking to evaluate safety violations
    - Masking of faults and errors at hardware, software and traffic-levels

AVFI Design



5 Sensor Faults
(e.g., Camera)

Occlusions

Water Droplets

Camera Noise Models

2 Autonomous Driving Agent

3 Fault Localizer

4 Fault Injector

Input FI

NN FI

Output FI

Timing FI

Route Planning

Goal

t_i

t_{i+1}

t_{i+2}

Time

Camera, Location

Input Sensor Readings

Neural Network - Perception, Localization, Planning

Actions

1 World Simulator: Unreal Engine + CARLA

Rendering Engine

Physics Engine

Environment Model

Sensor Models

Vehicle Models

6 Resilience Metrics

Mission Success Rate

Traffic Violations per KM

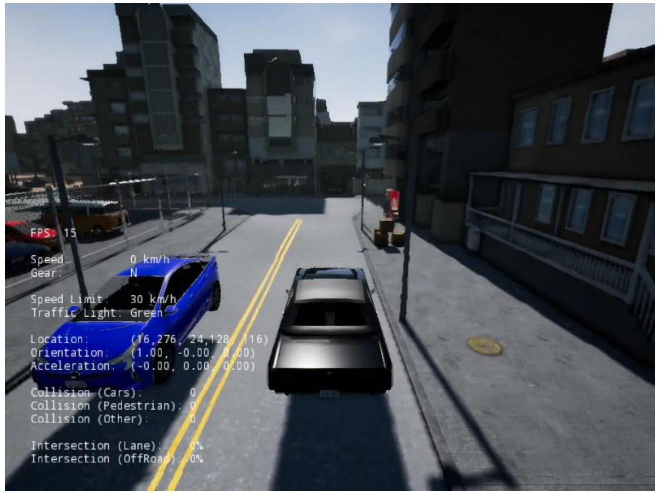
Time to Traffic Violation

AV Neural Networks

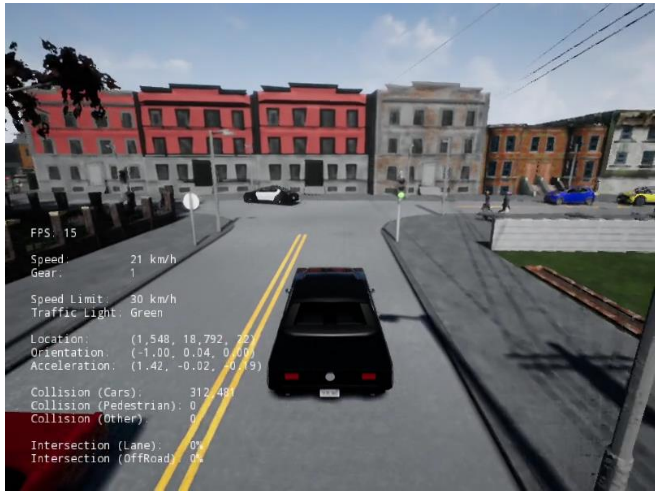
- Perception CNN
- Localization CNN
- Motion Planning CNN
- Fully Connected Layer
- RNN

[1] Dosovitskiy, Alexey, et al. "CARLA: An open urban driving simulator." *arXiv preprint arXiv:1711.03938* (2017)

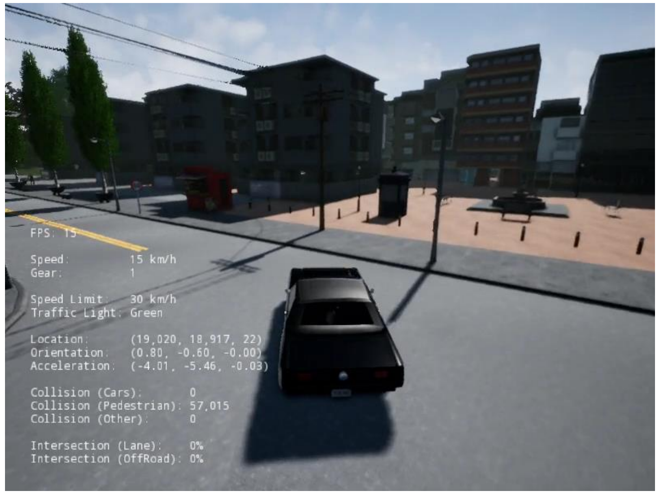
Example Injections



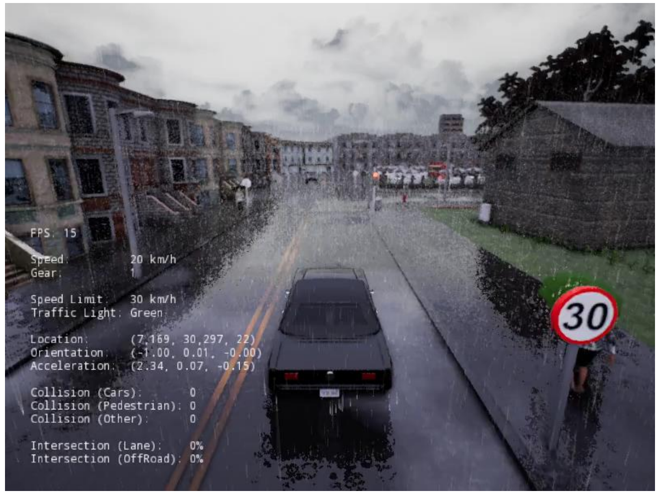
```
FPS: 15

Speed:            0 km/h
Gear:             N

Speed Limit:      30 km/h
Traffic Light:    Green

Location:         (16,276, 24,128, 116)
Orientation:      (1.00, -0.00, 0.00)
Acceleration:     (-0.00, 0.00, 0.00)

Collision (Cars):        0
Collision (Pedestrian):  0
Collision (Other):       0

Intersection (Lane):     0%
Intersection (OffRoad):  0%
```



```
FPS: 15

Speed:            21 km/h
Gear:             1

Speed Limit:      30 km/h
Traffic Light:    Green

Location:         (1,548, 18,792, 22)
Orientation:      (-1.00, 0.04, 0.00)
Acceleration:     (1.42, -0.02, -0.19)

Collision (Cars):        312,481
Collision (Pedestrian):  0
Collision (Other):       0

Intersection (Lane):     0%
Intersection (OffRoad):  0%
```



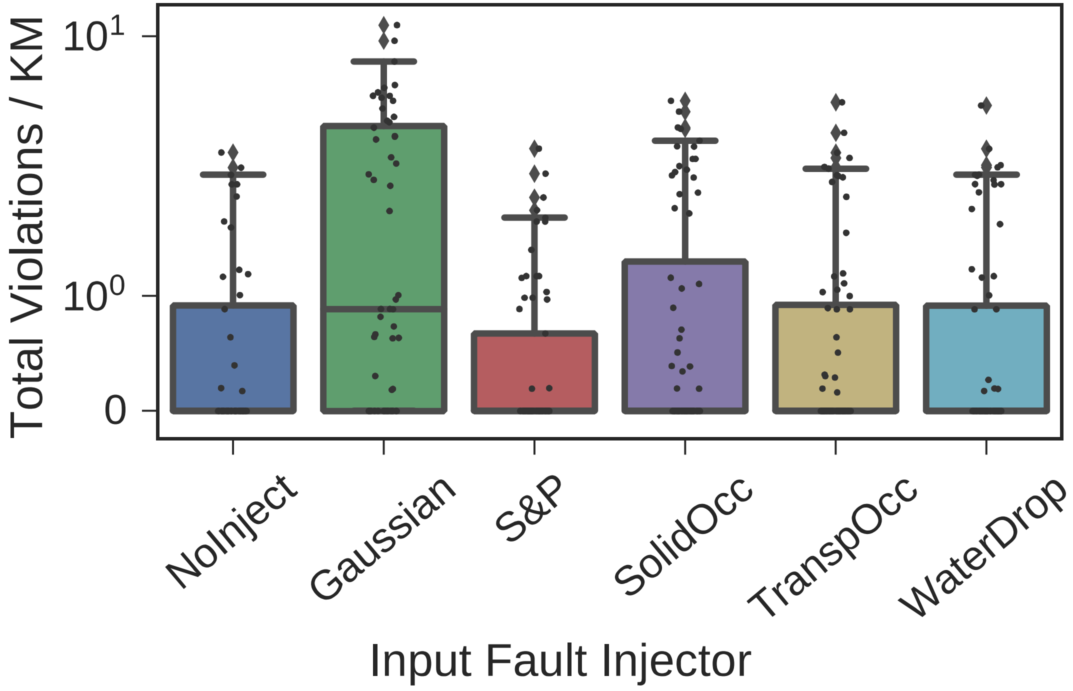
```
FPS: 15

Speed:            15 km/h
Gear:             1

Speed Limit:      30 km/h
Traffic Light:    Green

Location:         (19,020, 18,917, 22)
Orientation:      (0.80, -0.60, -0.00)
Acceleration:     (-4.01, -5.46, -0.03)

Collision (Cars):        0
Collision (Pedestrian):  57,015
Collision (Other):       0

Intersection (Lane):     0%
Intersection (OffRoad):  0%
```



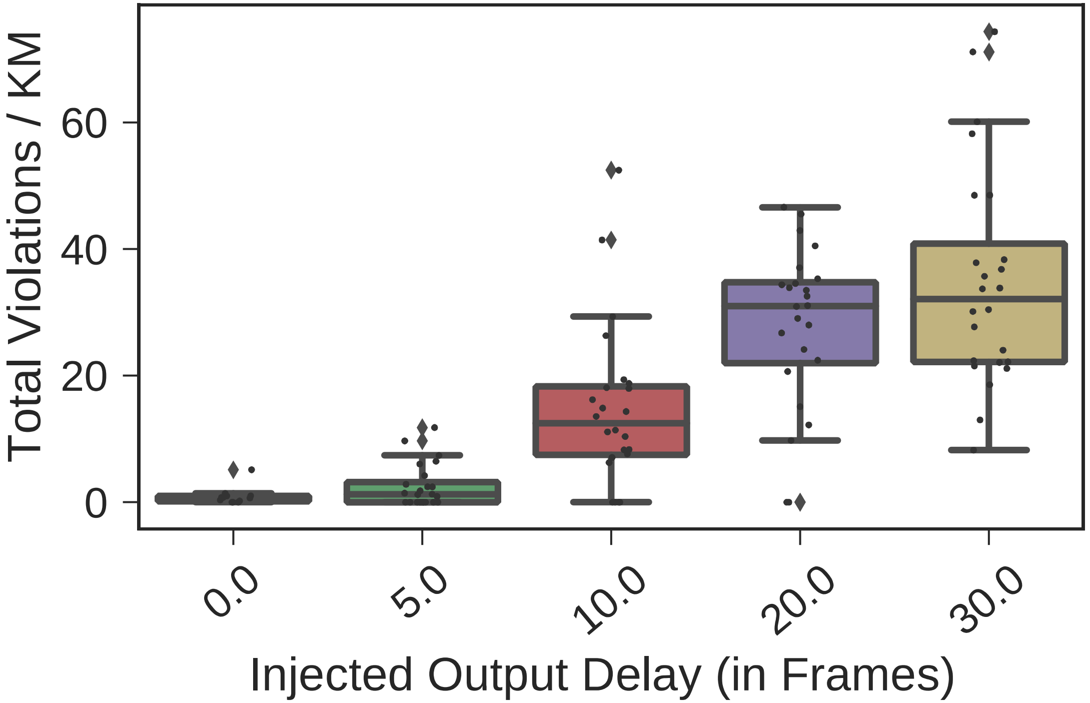
```
FPS: 15

Speed:            20 km/h
Gear:             1

Speed Limit:      30 km/h
Traffic Light:    Green

Location:         (7,169, 30,297, 22)
Orientation:      (-1.00, 0.01, -0.00)
Acceleration:     (2.34, 0.07, -0.15)

Collision (Cars):        0
Collision (Pedestrian):  0
Collision (Other):       0

Intersection (Lane):     0%
Intersection (OffRoad):  0%
```

Fault Injection Results

Input Sensor Fault Injection



Delay Injection



- **Sensor models**: GPS, LIDAR, RADAR, SONAR
- **Network failure** – Clock synchronization, Route Planning

Looking Forward

- Need for End-to-End resilience safety assessment
  - Holistic view of at system stack
  - Need to focus beyond DNNs
  - Traffic resilience needs to be accounted
- Fault injection is challenging: Time – Coverage trade off
- Improve system resilience by targeting most vulnerable kernels and system units

Questions?

Code: Simulator + Injector
Simulator – https://github.com/carla-simulator/carla
Injector – https://gitlab.engr.illinois.edu/DEPEND/av-imitation-learning-fault-injection