# Physical Adversarial Examples against End-to-End Autonomous Driving Models

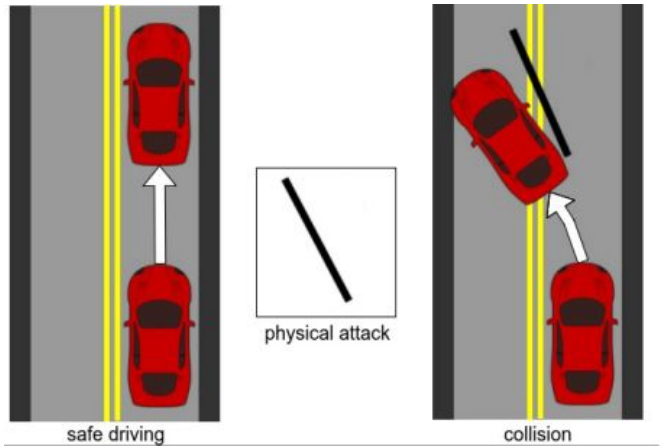

- Saurav Kumar
- Mithun Bhardwaj
- Kumar Gaurav

# Problem Introduction

- Manipulation of physical environment that leads to physical impact.

- Attack examples:
  - Removal of old lanes.
  - Shadows of electric poles, trees, overbirges etc. with different shadow intensity and width.

- These attacks may misguide the learned model to be perceived as lanes.




**Discarded by humans but tricks the neural network**

# Related work

Numerous work has been done on the topics such as

- Imitation Learning and Reinforcement learning  to take  images as inputs and outputs the controls of the agent in a closed loop system.
- Various physical adversarial attack have recently been studied and its impact computed.
- Framework for Certifying Robustness of  Neural Networks
- Improving the defence of the neural network against the adversarial attacks.

# Paper’s approach & Work

- Uses CARLA Imitation Learning Neural Network,which  inputs frames and outputs the steering angle.
- Goal is to iteratively project a line with estimated positions and orientation to maximize (deviate right)/ minimize(deviate  left) the summation of steering angle for each episode
- Uses bayesian optimizer to find the specific position and orientation of the attack ,such that the agent deviates maximum from the original trajectory.

$$\text{Collide Right}: \max_{l,\delta} \sum_{\tau=0}^{\Delta} f_{sa}(y_{f_l+\tau}(x_l+\delta), h_{f_l+\tau})$$

$$\text{Collide Left}: \min_{l,\delta} \sum_{\tau=0}^{\Delta} f_{sa}(y_{f_l+\tau}(x_l+\delta), h_{f_l+\tau})$$

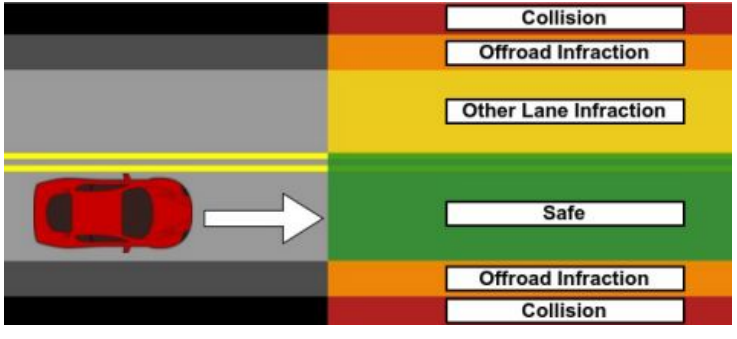
$$\text{subject to}: \quad l \in L, \quad \delta \in S.$$

# Our Approach & Work

- Reproduced the paper's results, using CARLA simulator and found the position and orientation which maximizes the deviation of the agent form the original trajectory.
- Fix the position and orientation from the above step and optimize over the **shadow intensity** and **width** of the projected line such that imitation network is tricked
- Goal is to find the boundary in colour and width space , where the network is tricked to perceive the a shadow as lane.
- Optimized over the summation of infraction rather than the summation of the steering angle to find the boundary.

$$\max_{c,w} \sum_{\tau=0}^{\Delta} f_{infraction}\ (y_{f_{c,w}+\tau}(x_{c,w}+\delta), h_{f_{c,w}+\tau})$$

$$subject\ to: \quad c \in [0,255], \qquad w \in [2,30), \qquad \delta \in S$$

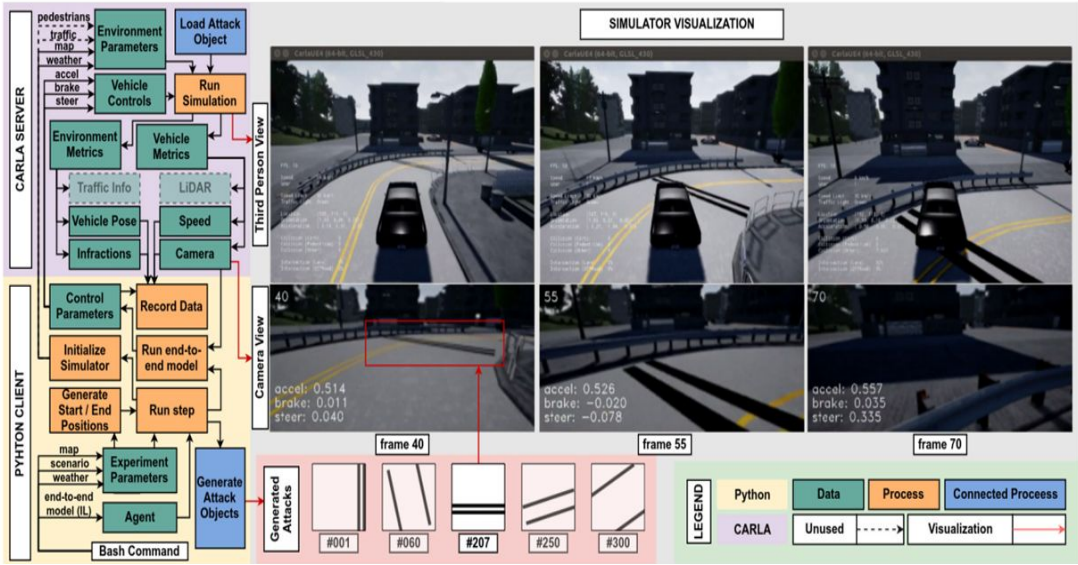


# Implementation

Objective :

$$\max_{c,w} \sum_{\tau=0}^{\Delta} f_{infraction}\ (y_{f_{c,w}+\tau}(x_{c,w}+\delta), h_{f_{c,w}+\tau})$$

$$subject\ to: \quad c \in [0,255], \qquad w \in [2,30), \qquad \delta \in S$$




Python Client :

- Generate the attacks
- Load the environment with attacks
- Run IL Neural Network and simulate.
- Record Data

Carla Server :

- Run Simulation
- Load the environment parameters & vehicle controls from the python client

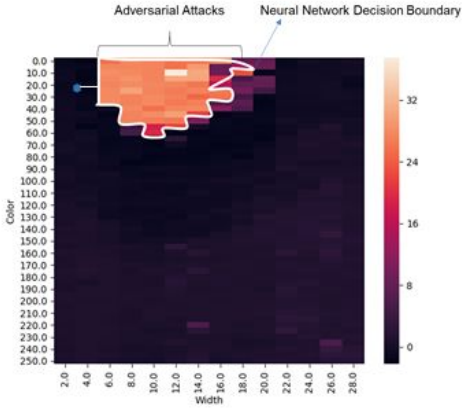
# Simulation

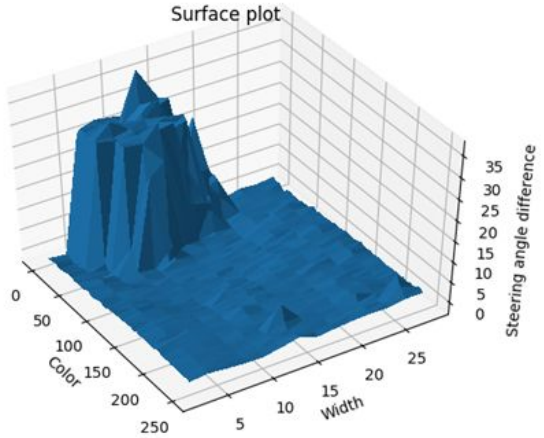

Shadow not perceived as lane


Shadow perceived as lane

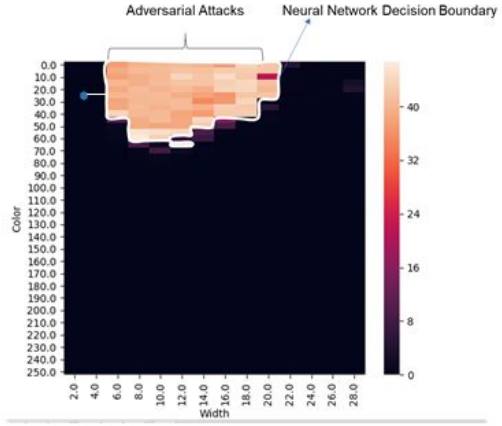
# Decision Boundary and Perturbation

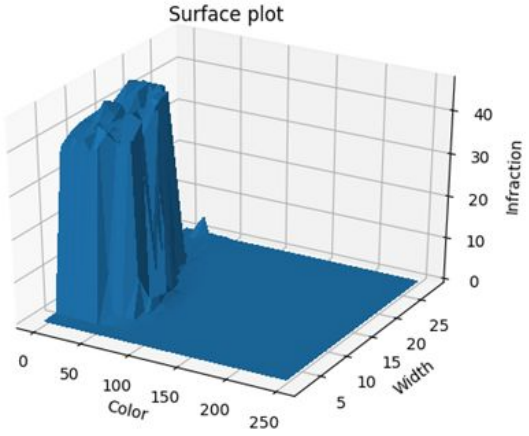
Steering Angle summation




Infraction summation



# Steering Angle Prediction

- **Why?**
  - Having the knowledge about the amount of deviation that might occur because of the current state is important to be able to defend it
- **Regression** - Learn the mapping from the characteristics of the adversary (color and width) to the amount of deviation from ideal behavior.
- **Challenges:**
  - Data distribution is not easy to learn -> Most of the values are zeros
  - The few instances where we have non zero values, the range of values is very high (0.0027 to 36 or higher $\approx 10^4$ ).

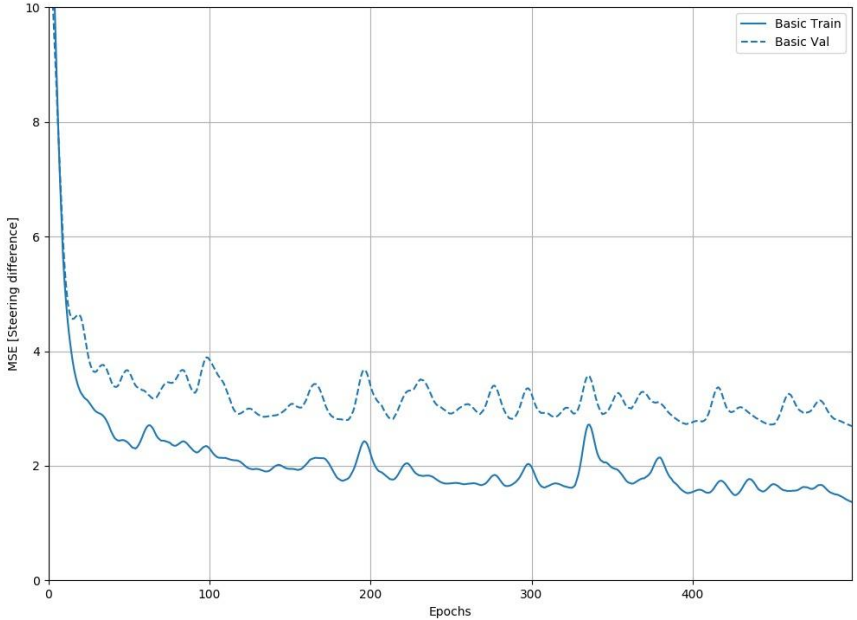
How does it make the training difficult?

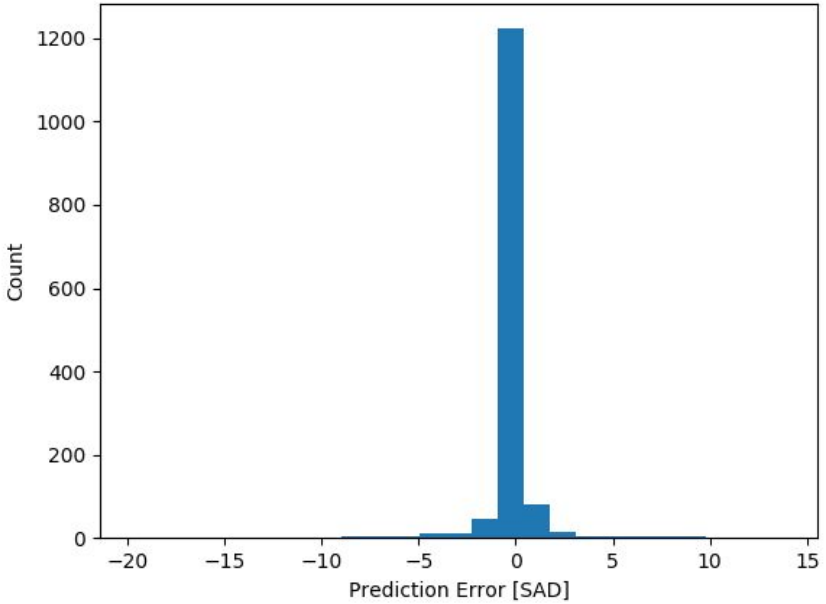
- When most of the values are zeros, the test-train and training-validation data split will not be balanced making the interpretation of the results difficult.

Design decisions to overcome this:

1. Increase the size of the validation set so that the distribution is more representative of the training set, or
2. Sample uniformly from the training set. This is hard to do in most cases because we don’t know if the training set represents the complete distribution of the data that will be encountered. This is easier to implement on our data since we know the bounds on the input

# Results




- 1428 samples in our test set (Data collection is in process)
- 95.7% of the predictions are within ± 0.5

# Infraction Classification

- **Why?**
  - It's hard to put a threshold on steering angle prediction to classify for adverse behavior but infraction makes it easier.
- **Classification** - 0 - Normal behavior, 1 - adverse behavior (infraction > 0)
- **Classifier -** SVM with polynomial kernel of degree 4
- **Results** -

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.93 | 0.96 | 1061 |
| 1 | 0.82 | 0.96 | 0.89 | 367 |
| accuracy | | | 0.94 | 1428 |
| macro avg | 0.90 | 0.95 | 0.92 | 1428 |
| weighted avg | 0.94 | 0.94 | 0.94 | 1428 |

- Emphasis on recall of class 1 (Of all the infractions, how many were identified)

# Future Work

- Predict the steering angle and infraction using images
- Retrain the network to defend the adversarial attacks.
- Check the vulnerability of other end to end self driving models for adversarial attack compared to the IL model in CARLA.
- Implement CNN-Cert to certify measure of robustness on the current network for colour intensity and width value.

# References

[1] 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 models,” 2017.

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: ´ An open urban driving simulator,” in CoRL, 2017.

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: ´ An open urban driving simulator,” in CoRL, 2017.

[4] Chawin Sitawarin, Arjun Nitin Bhagoji,Mung Chiang, Prateek Mittal and Arsalan Mosenia, “DARTS: Deceiving Autonomous Cars with Toxic Signs” , 2018

[5] Epic Games Inc., “What is unreal engine?,” 2019.