ADAPTIVE ADVERSARIAL VIDEOS ON ROADSIDE BILLBOARDS: DYNAMICALLY MODIFYING TRAJECTORIES OF AUTONOMOUS VEHICLES

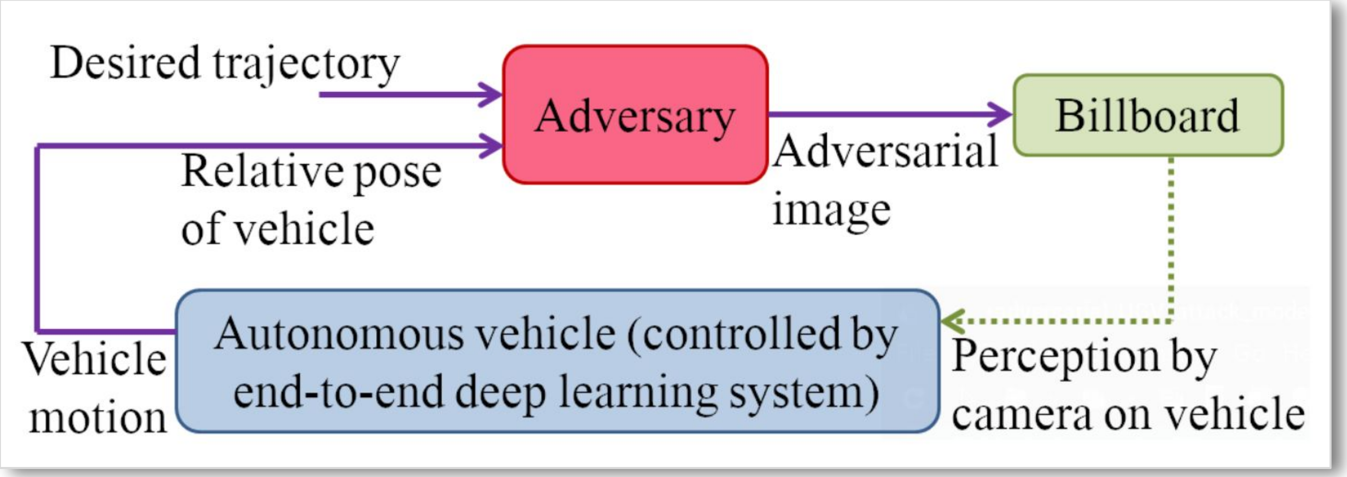
Written by: Naman Patel, et al.

Presented by: Dev Shah

The Aim of this paper

- ➢ Deep neural networks (DNNs) are utilized in self-driving vehicles and robotics
- ➢ These systems are susceptible to adversarial attacks causing targeted misclassification.

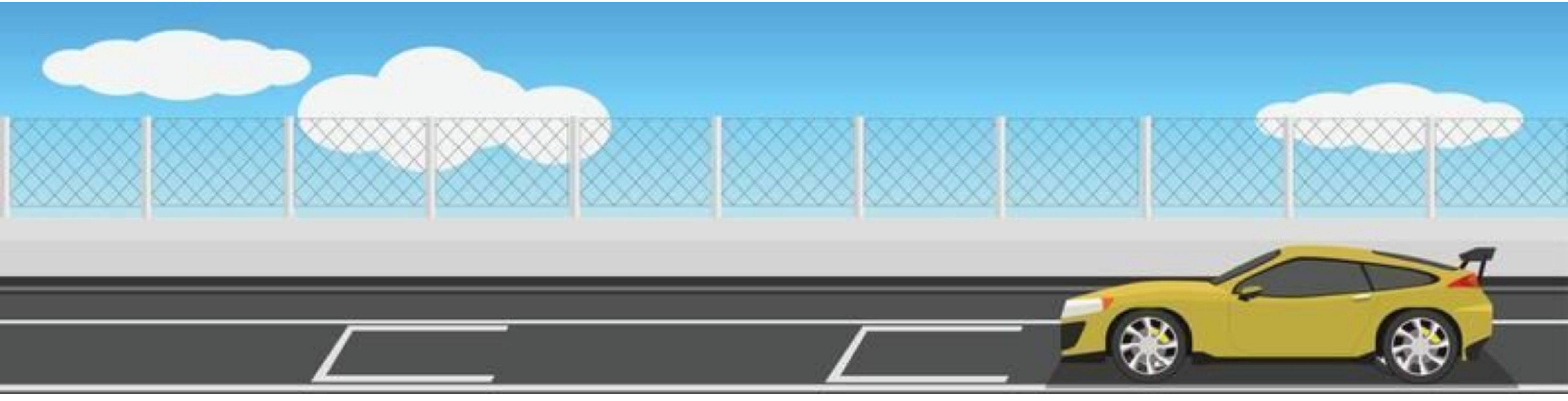
The authors investigate the susceptibility of these systems by displaying videos on billboards to upcoming vehicles causing the DNN controller to generate steering commands in order to change lanes thus, causing accidents.



Desired trajectory
Adversary
Billboard
Adversarial image
Relative pose of vehicle
Autonomous vehicle (controlled by end-to-end deep learning system)
Perception by camera on vehicle
Vehicle motion

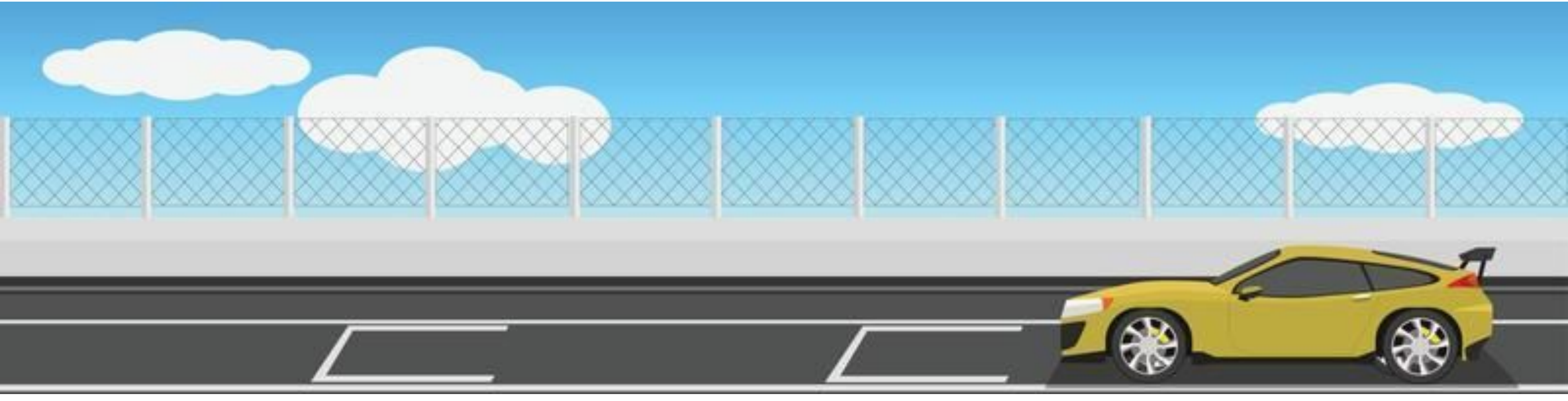
Conditional Imitation Learning

- It takes inputs as current image, destination and suggested action
- It generates commands to operate steering and speed of the vehicle



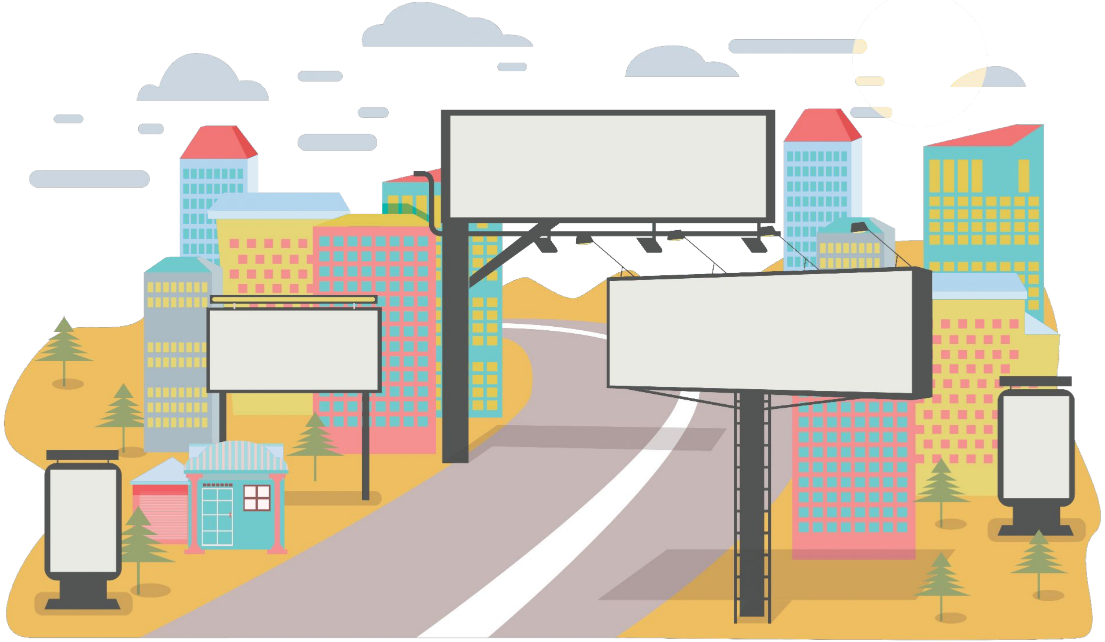
CARLA (CAR Learning to Act)

- ➢ Unreal Engine 4 based simulator
- ➢ Environment - Minitown, adjustable weather, buildings, vegetation, traffic signals, infrastructure
- ➢ Controlling agent - Car’s camera (customisable features)
- ➢ Feasible to test under various weather conditions



Contributions of the paper

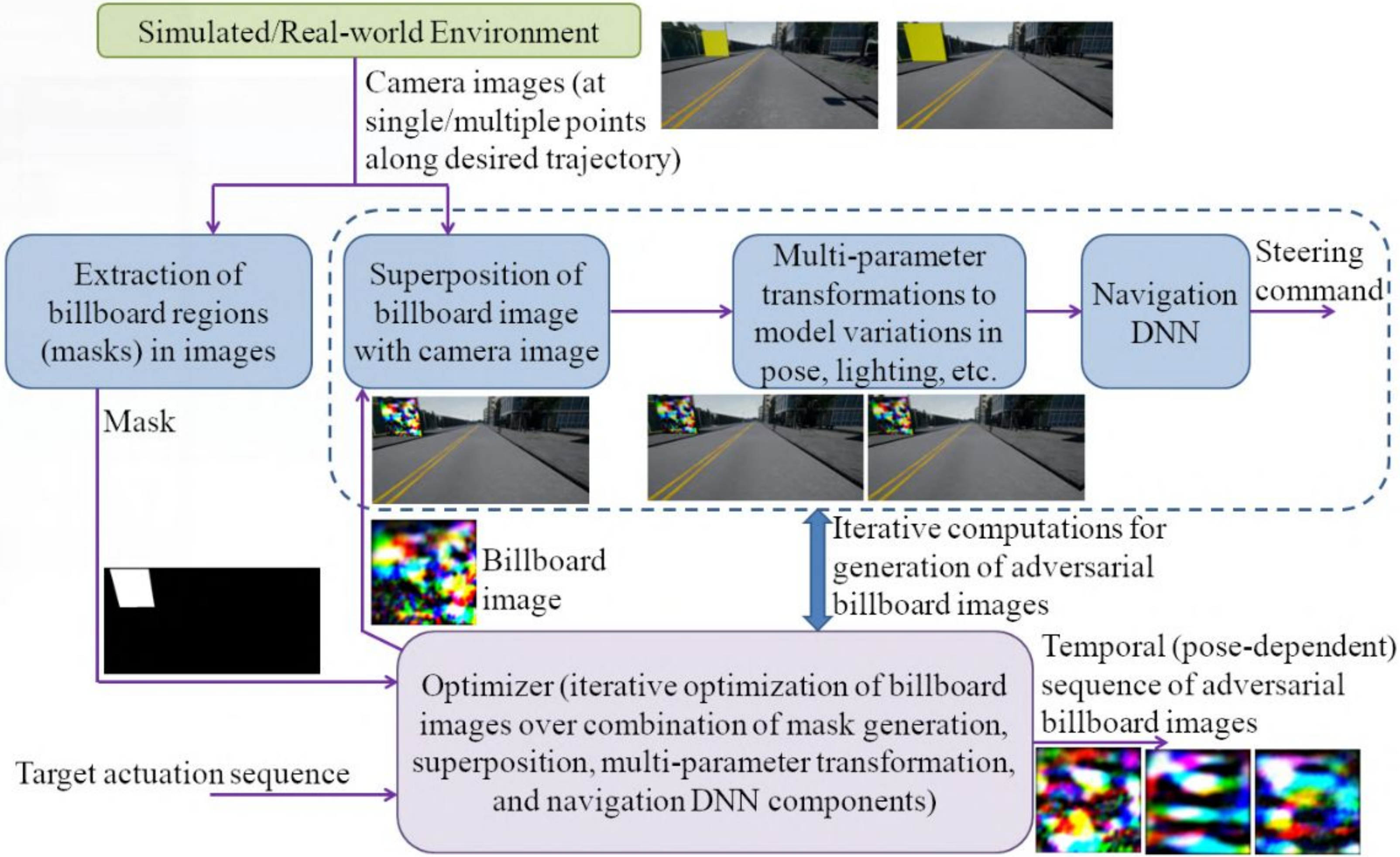
- Framework
- Dynamic
- Adaptive
- Iterative
- Robust



Creating the attack

$$L(M) = \sum_i \mathbb{E}_{T\sim\mathcal{T}}\{|N(T(f_{p_i}(C_i, M))) - s_{des,i}|\} + R(M)$$

- C - Image of billboard observed from vehicle’s camera
- M - Matrix representing image to be displayed on the billboard
- p - Relative pose of the vehicle
- $f_p(C, M)$- pose-dependant function representing image formed by superimposing M on C
- $\mathcal{T}$- Series of transformations of M to mitigate the effect of variabilities
- N - Vehicle’s navigation DNN mapping (camera image to steering angle)
- s_{des} - Desired adversarial steering angle
- E{.} - Expectation computed over distribution $\mathcal{T}$
- R(M) - Regularizar that smoothes M (removes pixelation to increase the rate of attack)
- L(M) - Loss function



Simulated/Real-world Environment

Camera images (at single/multiple points along desired trajectory)

Extraction of billboard regions (masks) in images

Superposition of billboard image with camera image

Multi-parameter transformations to model variations in pose, lighting, etc.

Navigation DNN

Steering command

Mask

Billboard image

Iterative computations for generation of adversarial billboard images

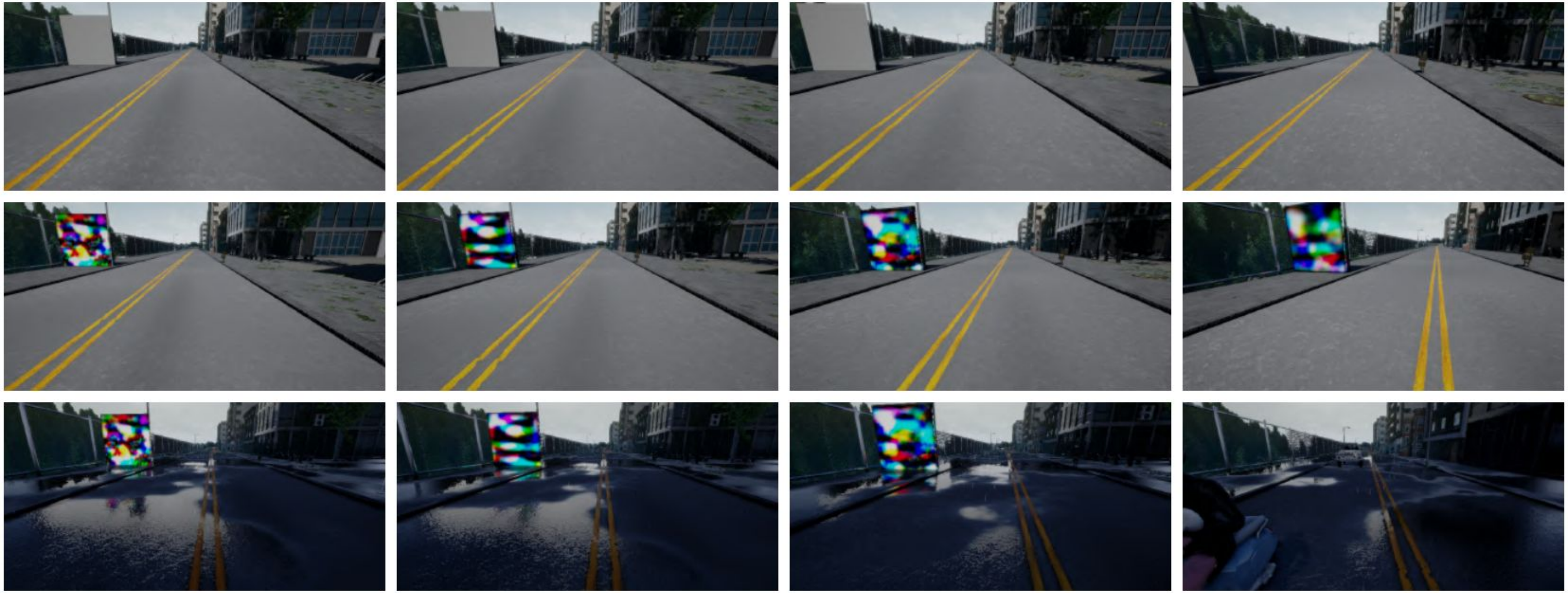
Target actuation sequence

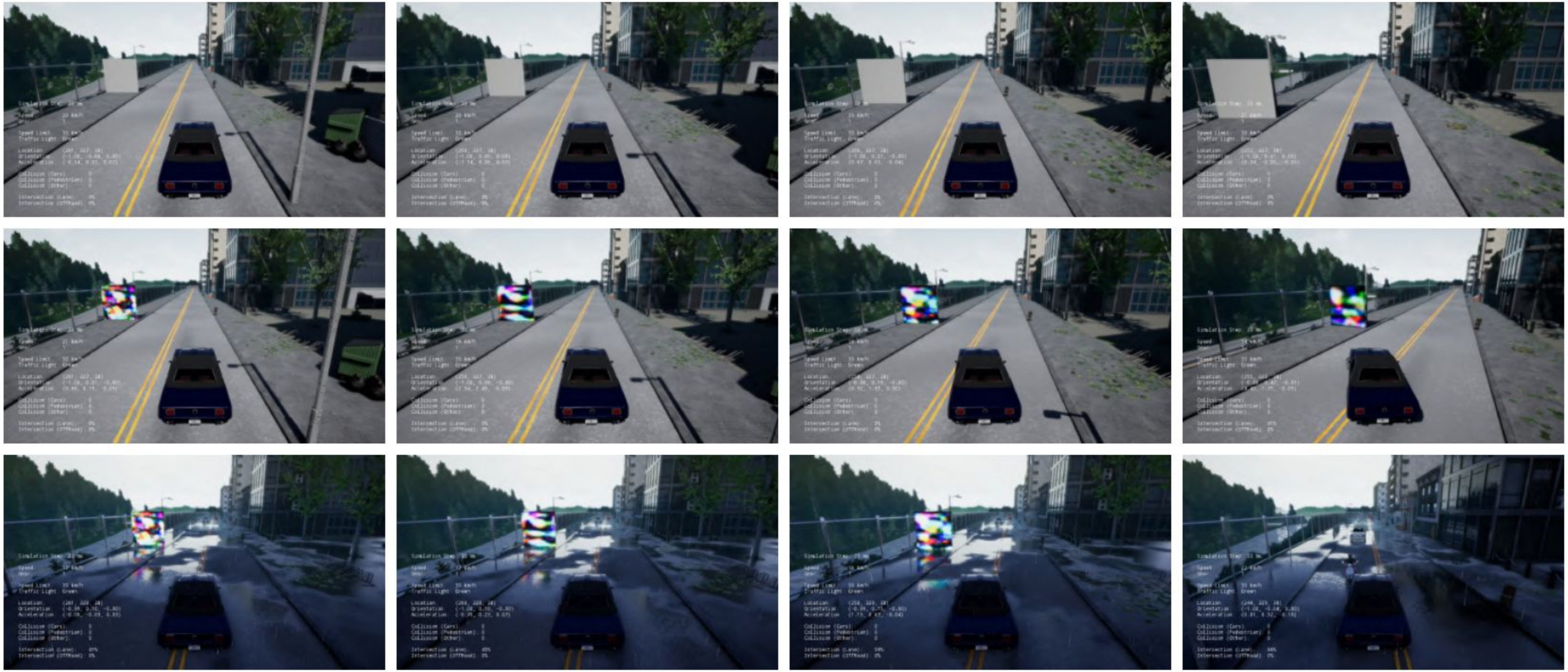
Optimizer (iterative optimization of billboard images over combination of mask generation, superposition, multi-parameter transformation, and navigation DNN components)

Temporal (pose-dependent) sequence of adversarial billboard images

Attacking Scenarios Results

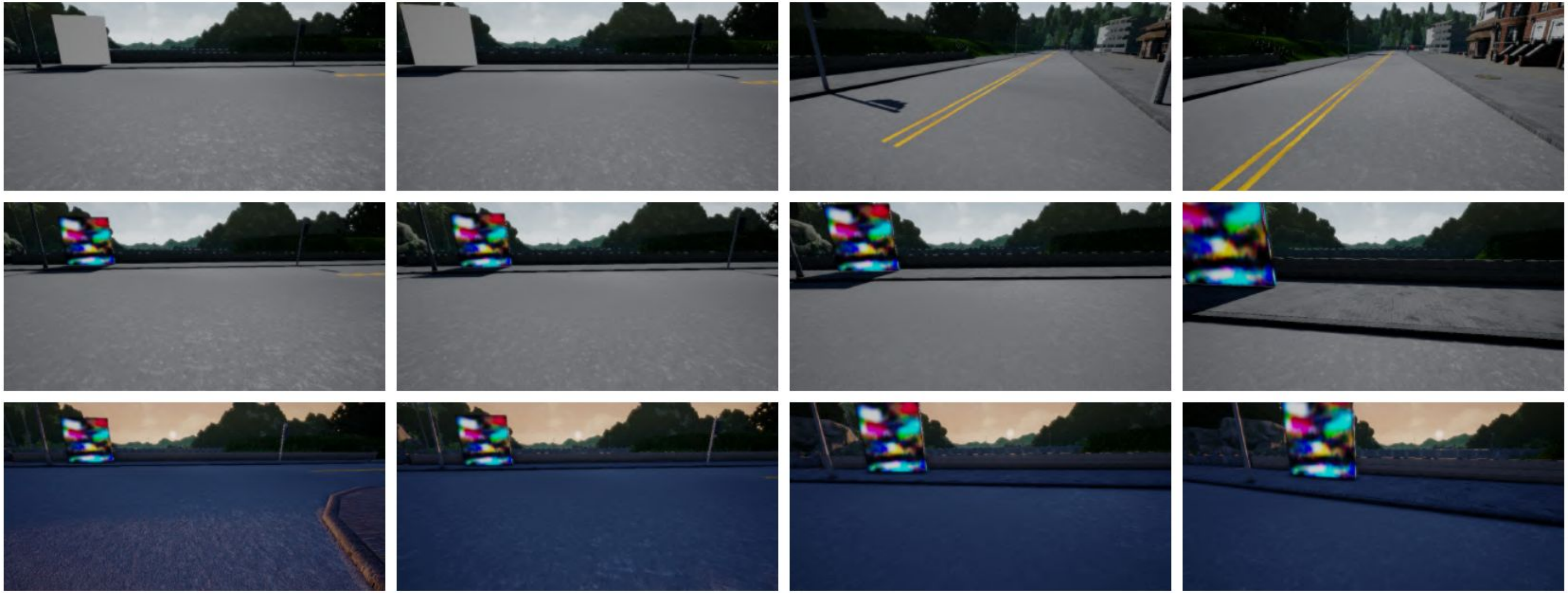
1. Attacking a car moving in straight lane to turn left.

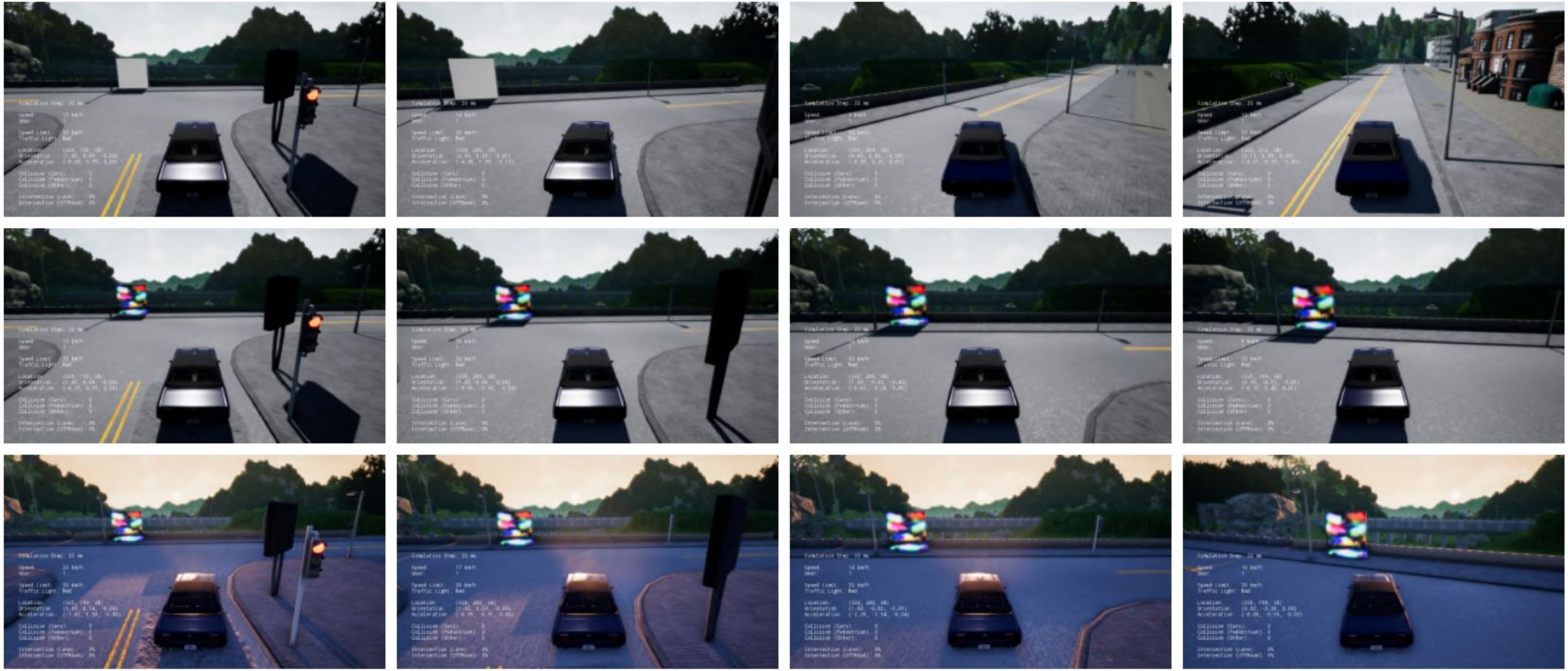




Attacking Scenarios Results

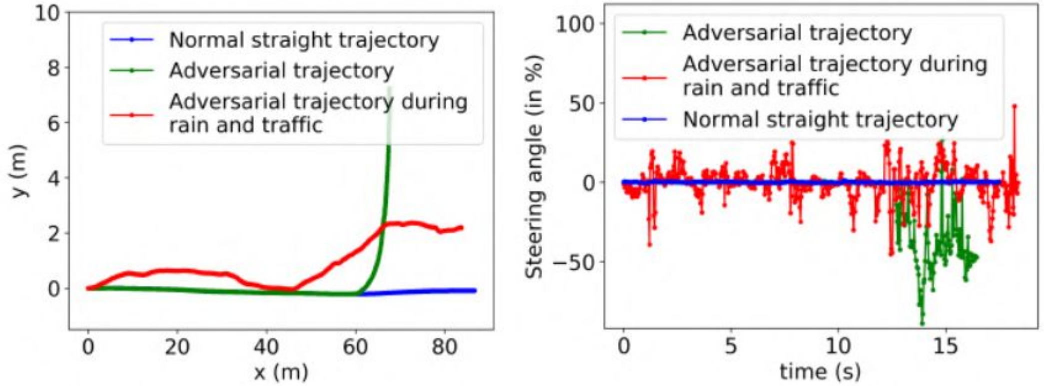
2. Attacking a car making a right turn to move straight.



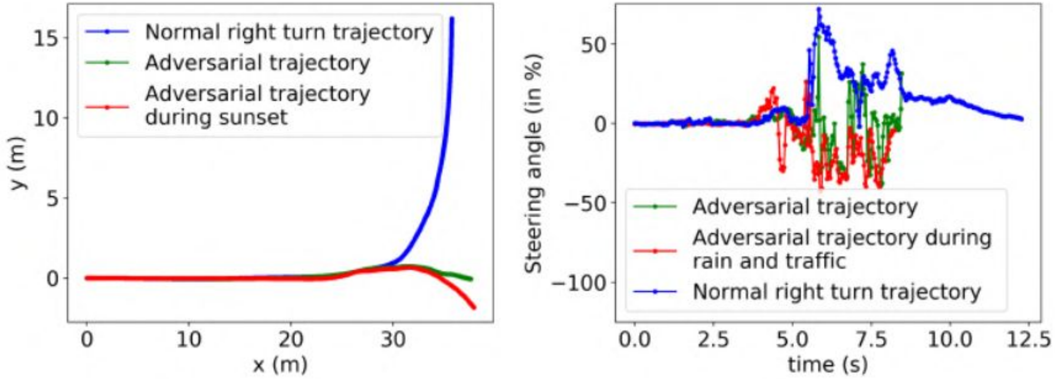


Resulting trajectories

1.

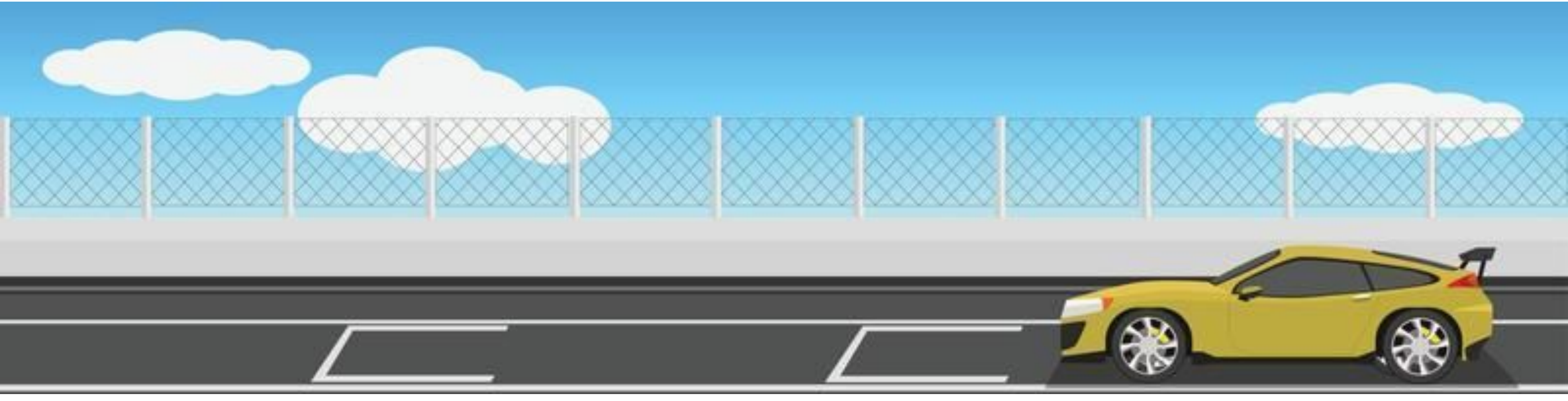


2.



Discussion

1. Improvements in the DNN?
2. How effective this attack is if tested in the real world? What parameters would hinder it?



THANK YOU!