LiDAR Segmentation-based Adversarial Attacks on Autonomous Vehicles

Blake Johnson

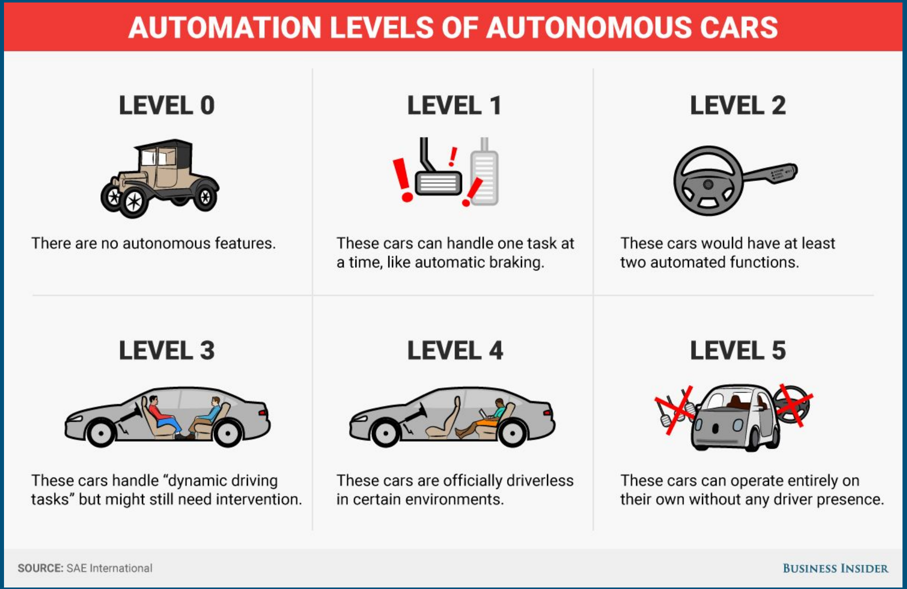
Outline

- **Intro**
  - AVs
  - LiDAR
- Background
  - Neural Networks
  - Adversarial Examples
  - LiDAR Data Processing
- Attack Scenarios
- Adversarial Location Generation
  - Attack Framework
  - Loss Variables
  - Total Loss Function
  - White-box Attack
- Attack Execution
  - Setup
  - Results
    - SemanticKITTI Dataset
    - Real-World
- Conclusion

Introduction

Autonomous Vehicles (AVs)

- Utilize numerous sensors to drive (cameras, sonar, GPS, etc.)
- Various levels
- Rising popularity
- Utilize LiDAR (Light Detection and Ranging) for 3D perception of environment

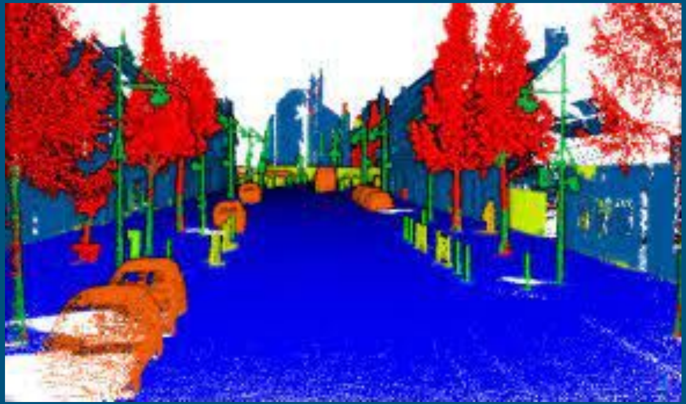


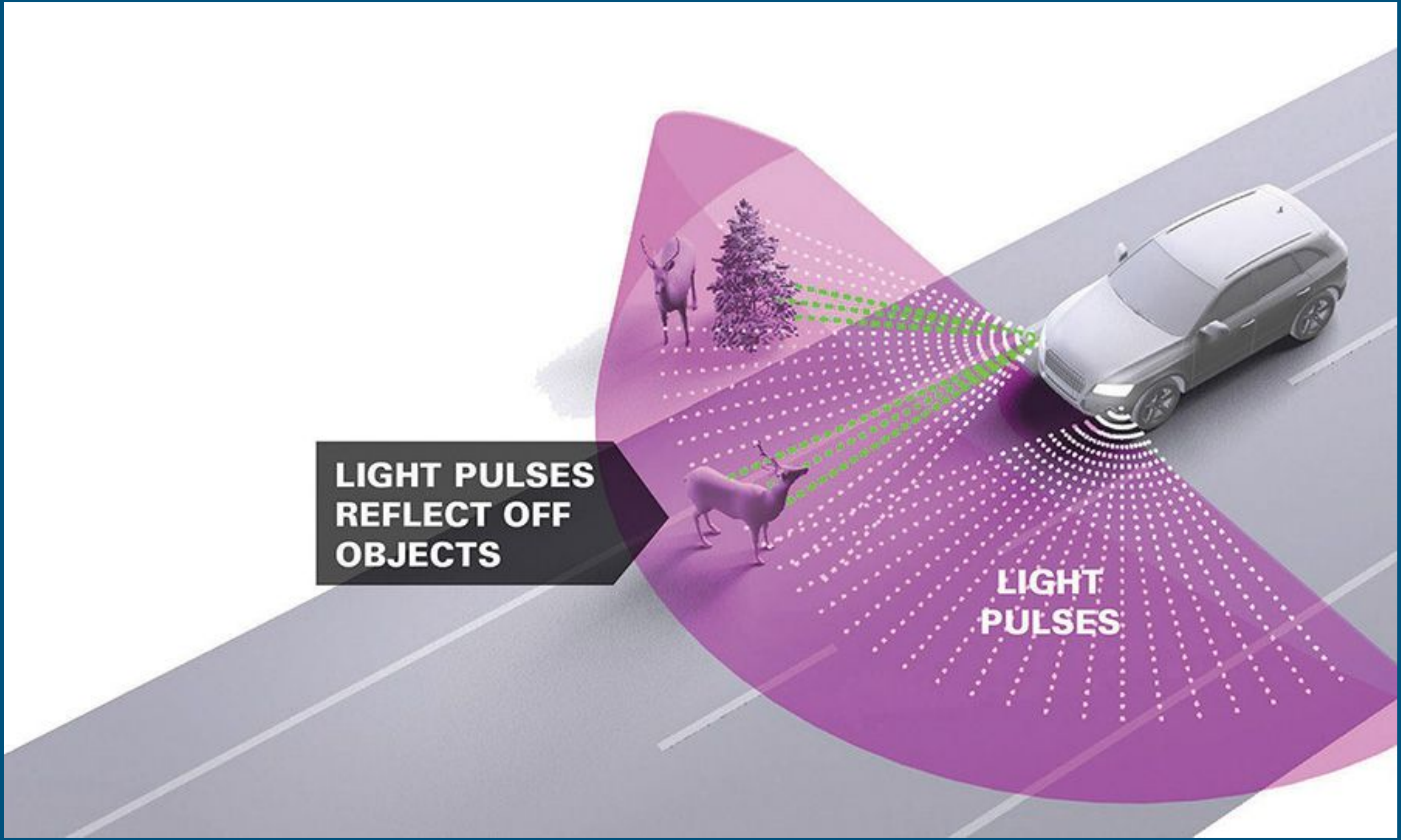
Waymo Driverless Taxi

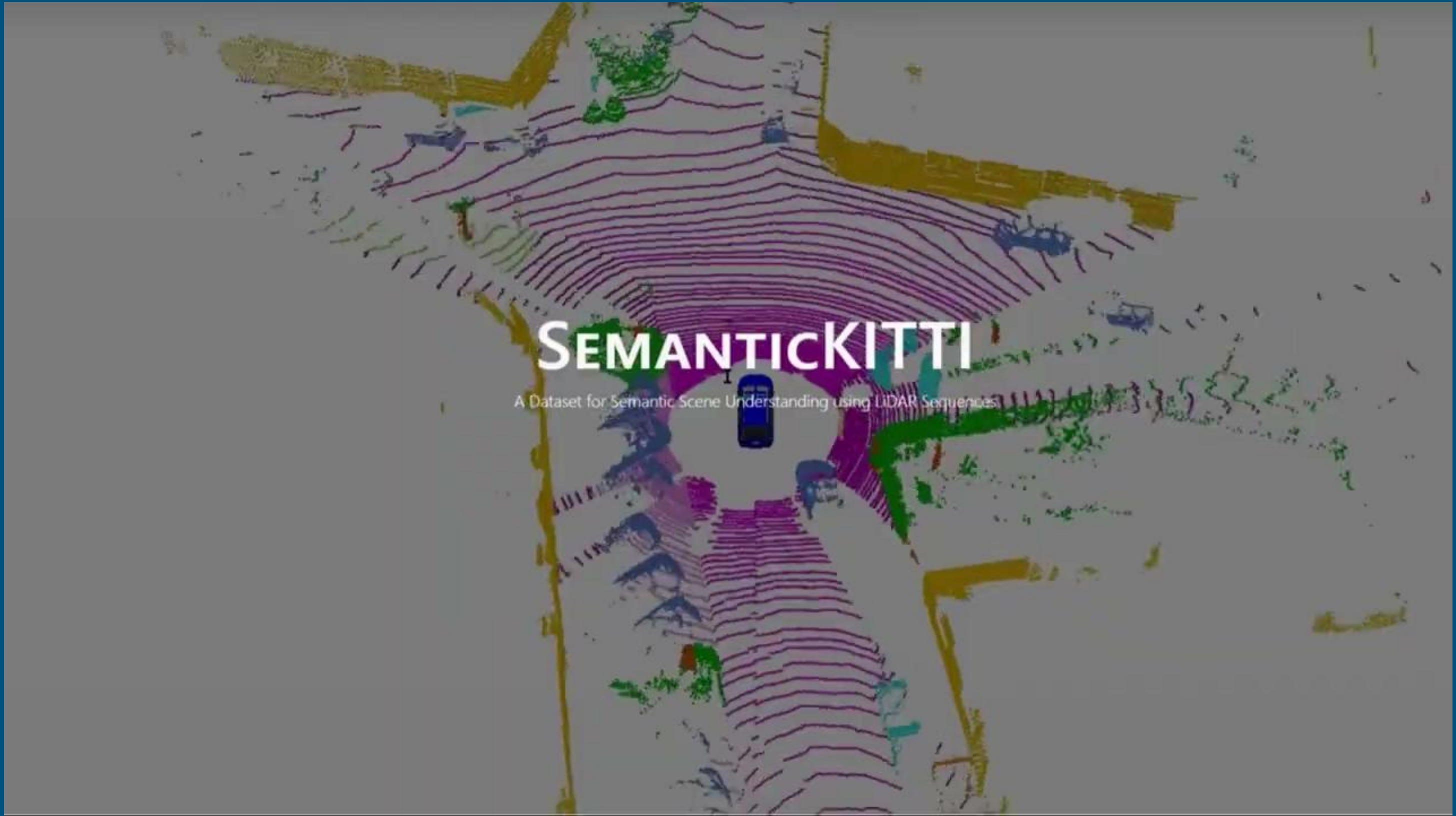
Honda’s Level 3 Self-Driving car

LiDAR (Light Detection and Ranging)

- LiDAR fires lasers into surroundings to measure distance from potential objects
- Generates 3D point cloud through firing lasers at various angles
- Segmentation step of LiDAR separates point cloud into regions
  - Regions are labeled with classes (grass, vehicle, road, etc.) by neural network
- Vulnerable to adversarial attacks




LIGHT PULSES
REFLECT OFF
OBJECTS
LIGHT
PULSES


SEMANTICKITTI
A Dataset for Semantic Scene Understanding using LiDAR Sequences

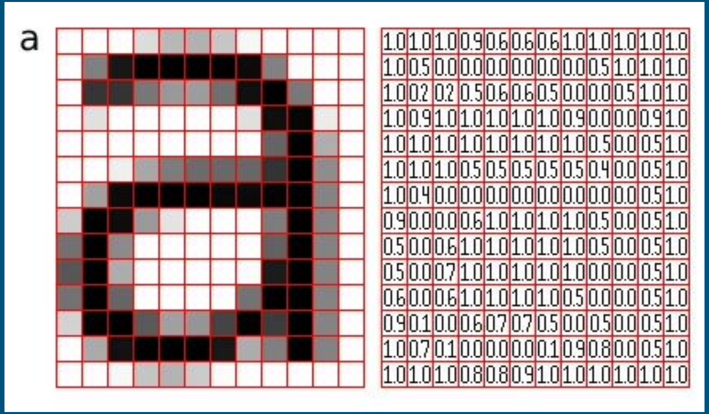
Outline

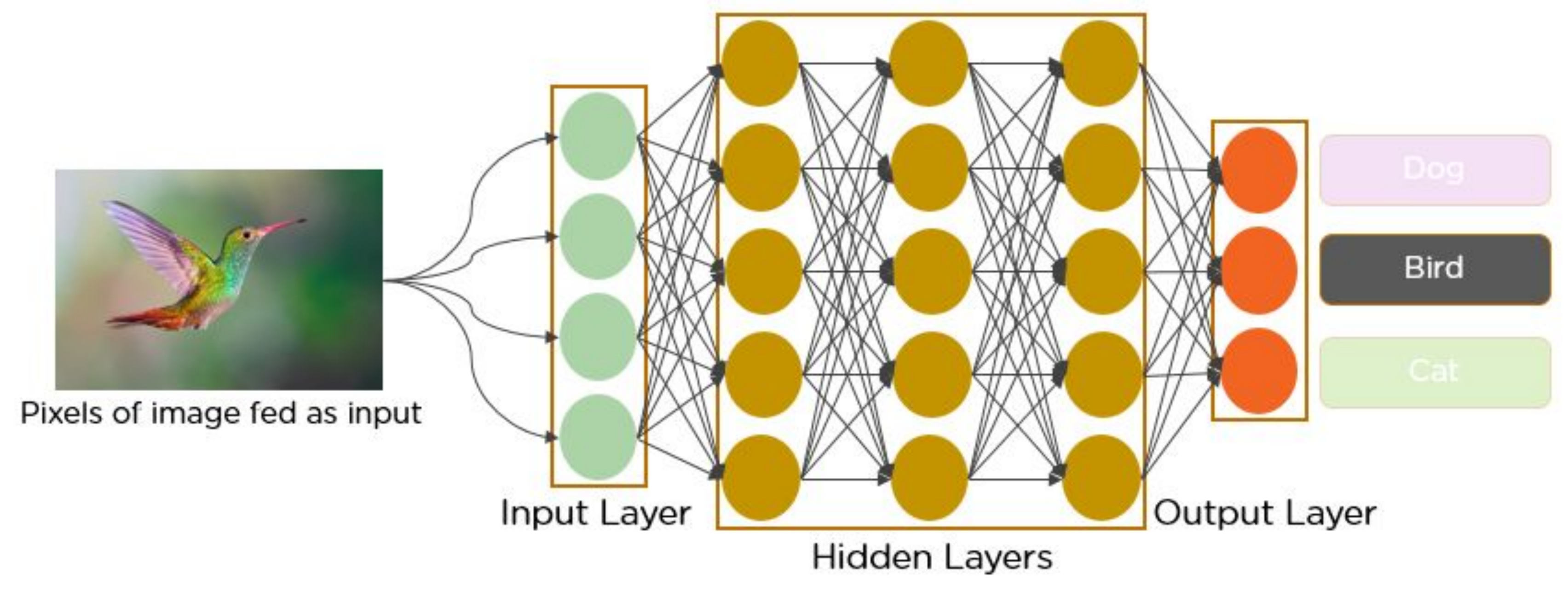
- Intro
  - AVs
  - LiDAR
- **Background**
  - Neural Networks
  - Adversarial Examples
  - LiDAR Data Processing
- Attack Scenarios
- Adversarial Location Generation
  - Attack Framework
  - Loss Variables
  - Total Loss Function
  - White-box Attack
- Attack Execution
  - Setup
  - Results
    - SemanticKITTI Dataset
    - Real-World
- Conclusion

Background

Neural Networks-Framework

- Consist of up to millions of interconnected nodes
- Organized into layers with data flowing one-way
- A node’s incoming connections issued weight values
- Data value flowing through node is multiplied by weight
- Product is compared to threshold value
- Sent to outgoing connections or stopped
- Convolutional Neural Network (CNN)



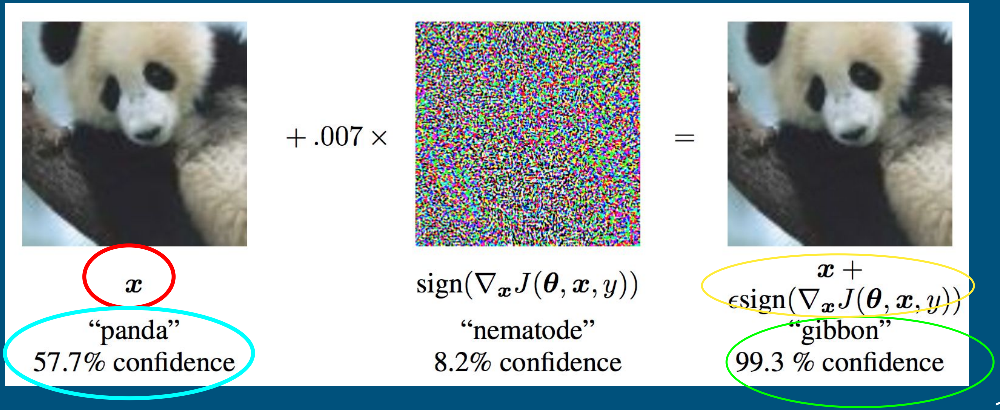

Pixels of image fed as input
Input Layer
Hidden Layers
Output Layer
Dog
Bird
Cat

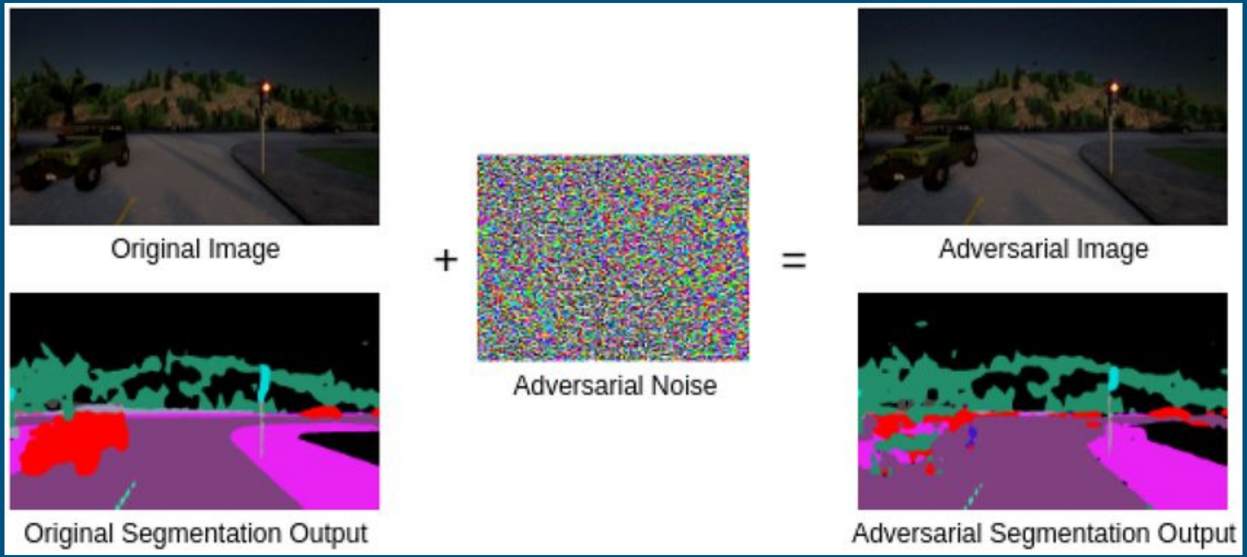
Neural Networks-Training

- Weight and threshold values are randomized
- Input data is fed into net
- Data is multiplied and transformed while flowing through layers
- Weight and threshold values repeatedly adjusted
- Complete when input data with specific labels consistently produces similar outputs

Adversarial Examples

- Maliciously created inputs
- Indistinguishable to human eye
- Intention of fooling machine learning models
- Goal is to result in misclassification of given input
- M(x’)≠y or M(x’)=y’

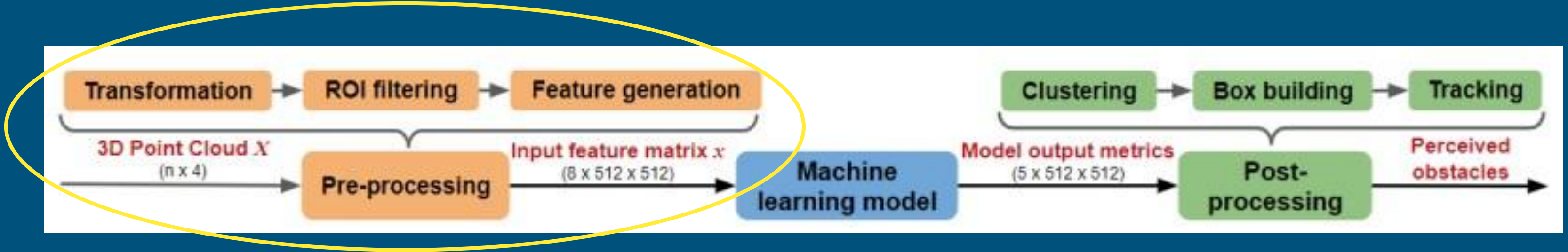



Original Image
+
Adversarial Noise
=
Adversarial Image
Original Segmentation Output
Adversarial Segmentation Output

LiDAR Data Processing

Pre-Processing

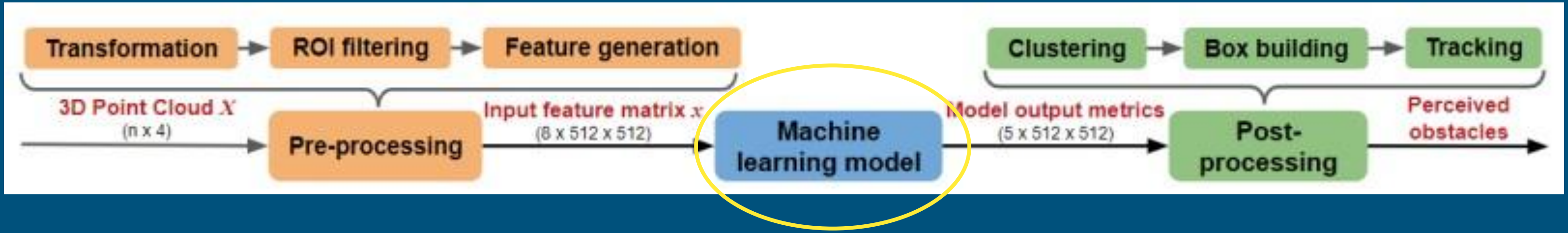
- Raw data points transformed into coordinate system
- ROI filters out irrelevant data points
- Filtered point cloud is mapped to 512 x 512 cells
- Eight features are created for each cell
- This generates feature matrix (8 x 512 x 512)



DNN-based Segmentation

- Feature matrix is used as input for convolutional neural network (CNN)
- CNN produce output of five metrics (5 x 512 x 512)

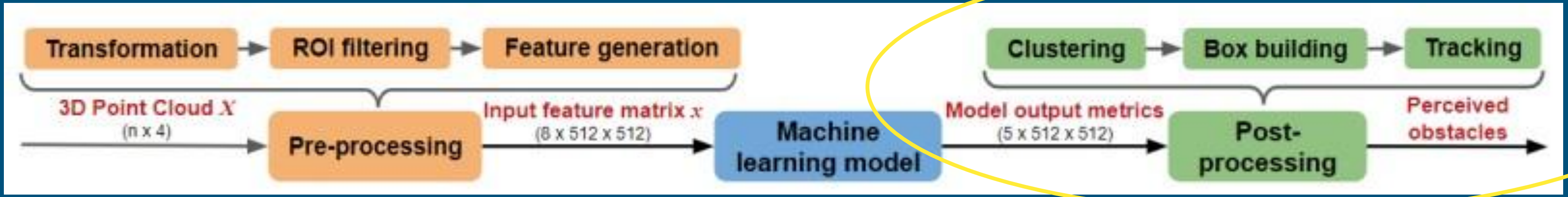
Metrics	Description
Center offset	Offset to the predicted center of the cluster the cell belongs to.
Objectness	The probability of a cell belonging to an obstacle.
Positiveness	The confidence score of the detection.
Object height	The predicted object height.
Class probability	The probability of the cell being a part of a vehicle, pedestrian, etc.



Post-Processing

Objectness	The probability of a cell belonging to an obstacle.
Positiveness	The confidence score of the detection.

- Connected graph is created from output metrics for object cluster candidates
- Candidates filtered by average positiveness
- Bounding box constructed from object cluster candidate’s dimensions
- Individual frames of processed results are connected to generate tracked objects



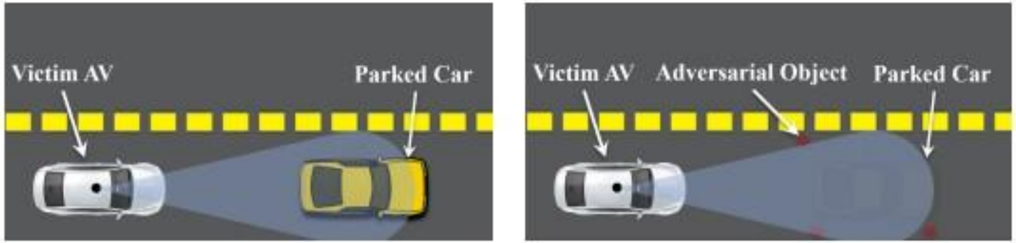
Outline

- Intro
  - AVs
  - LiDAR
- Background
  - Neural Networks
  - Adversarial Examples
  - LiDAR Data Processing
- **Attack Scenarios**
- Adversarial Location Generation
  - Attack Framework
  - Loss Variables
  - Total Loss Function
  - White-box Attack
- Attack Execution
  - Setup
  - Results
    - SemanticKITTI Dataset
    - Real-World
- Conclusion

Attack Scenarios

Vehicle Hiding Attack

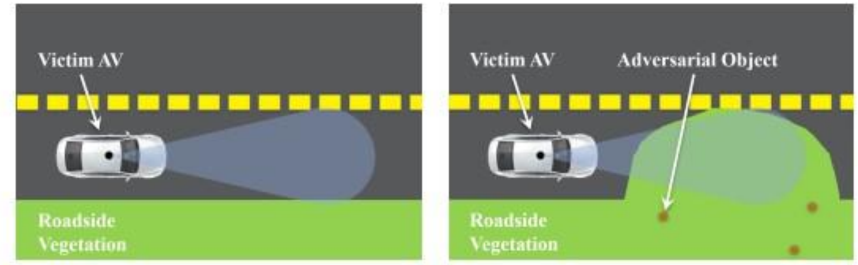
- Driving environment consists of car parked in place
- Adversarial objects added to make car disappear from LiDAR perception system of victim AV
- Effects



(a) Before attack (b) After attack

Road Surface Changing Attack

- Driving environment consists of an open road
- Adversarial objects added to make LiDAR perception system of victim AV perceive road as some undrivable surface
- Effects

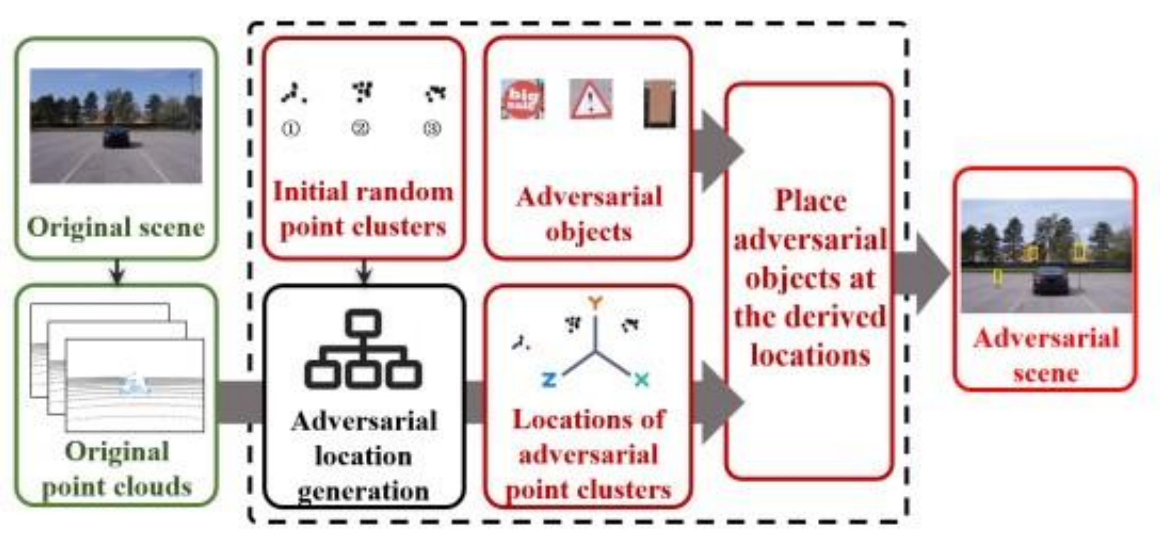


(a) Before attack (b) After attack

Outline

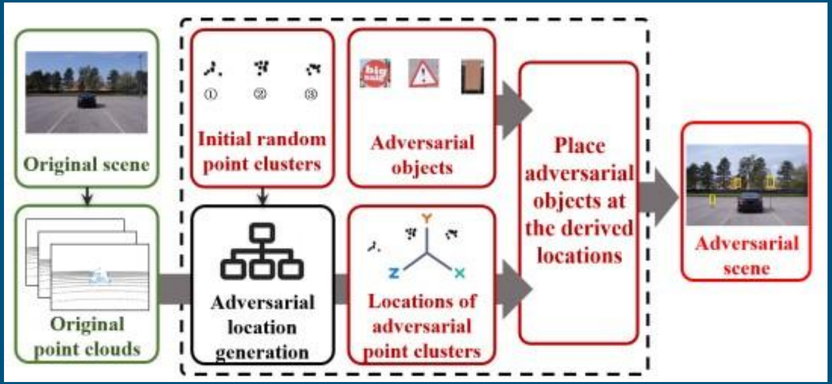
- Intro
  - AVs
  - LiDAR
- Background
  - Neural Networks
  - Adversarial Examples
  - LiDAR Data Processing
- Attack Scenarios
- **Adversarial Location Generation**
  - Attack Framework
  - Loss Variables
  - Total Loss Function
  - White-box Attack
- Attack Execution
  - Setup
  - Results
    - SemanticKITTI Dataset
    - Real-World
- Conclusion

Adversarial Location Generation

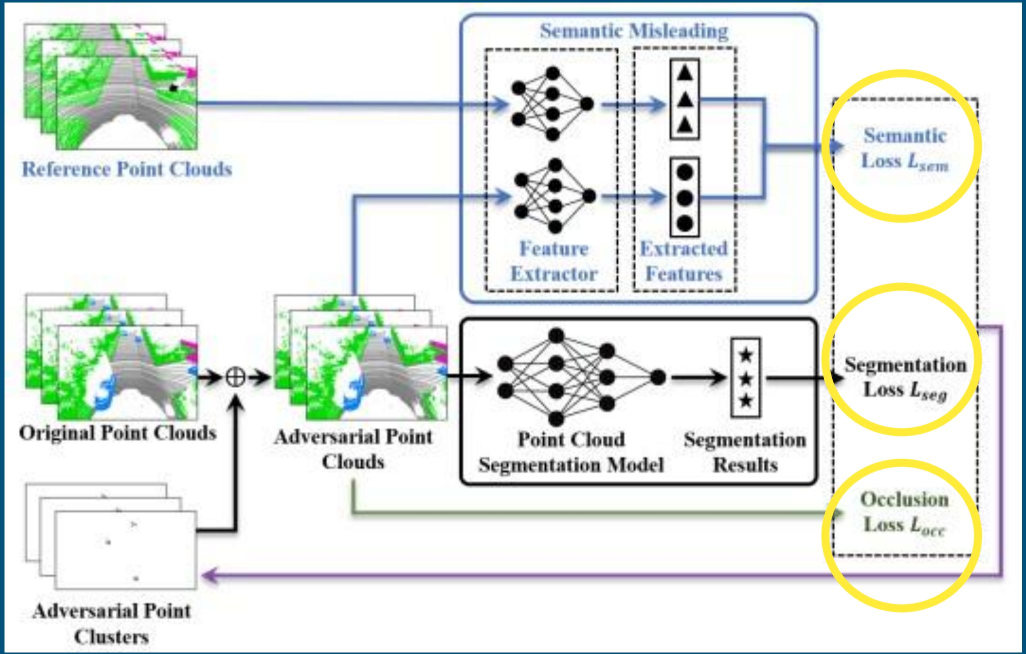

Original scene
Original point clouds
Initial random point clusters
①
②
③
Adversarial location generation
Adversarial objects
Locations of adversarial point clusters
Y
Z
X
Place adversarial objects at the derived locations
Adversarial scene

Attack Framework

- Find optimized locations for adversarial objects
  - Mimic victim AV driving patterns to collect 3D point cloud data
  - Initialize adversarial objects as random point clusters
  - Add random clusters to original point cloud
  - Optimize cluster center location through loss function
- Place adversarial objects at these locations

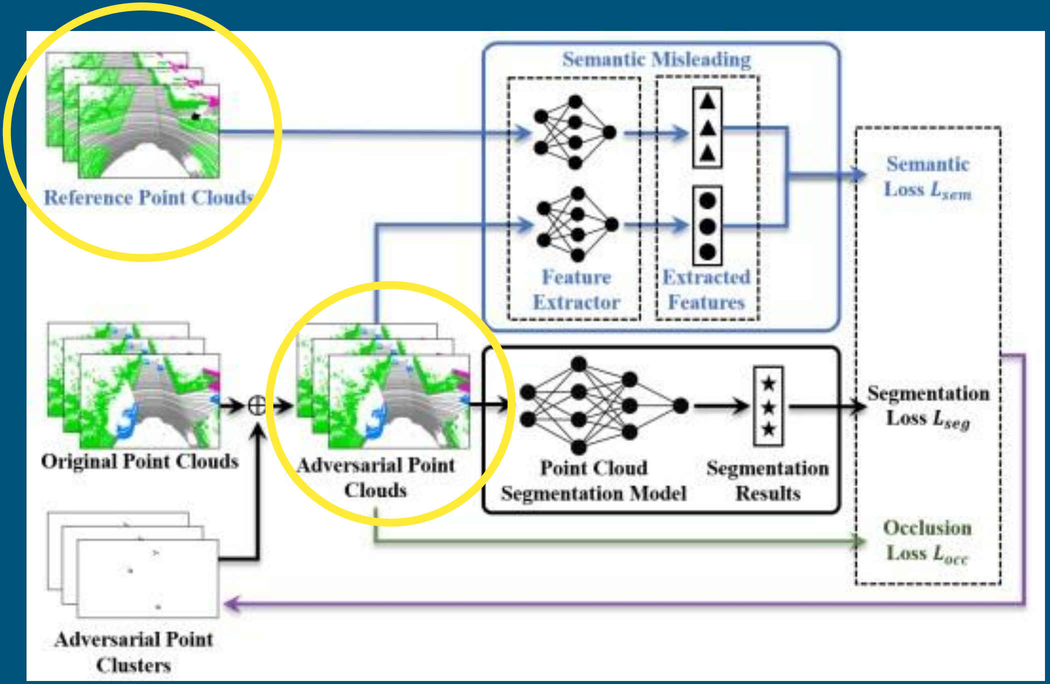


Variables of Loss


Semantic Misleading
Reference Point Clouds
Feature Extractor
Extracted Features
Semantic Loss L_{sem}
Original Point Clouds
Adversarial Point Clouds
Point Cloud Segmentation Model
Segmentation Results
Segmentation Loss L_{seg}
Occlusion Loss L_{occ}
Adversarial Point Clusters

Semantic Loss

- Associated measurement of semantic misleading method
- Goal of semantic misleading is to make semantic features of reference point clouds and adversarial point clouds similar
- Global features (large-scale structures)
- Feature extractor used to extract semantic features of point clouds
- Semantic loss is measurement of this similarity

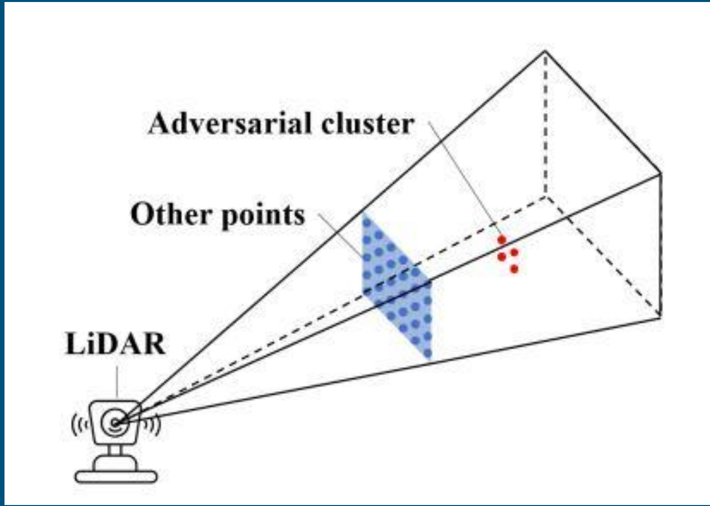

Reference Point Clouds
Semantic Misleading
Feature Extractor
Extracted Features
Semantic Loss L_{sem}
Original Point Clouds
Adversarial Point Clouds
Point Cloud Segmentation Model
Segmentation Results
Segmentation Loss L_{seg}
Occlusion Loss L_{occ}
Adversarial Point Clusters

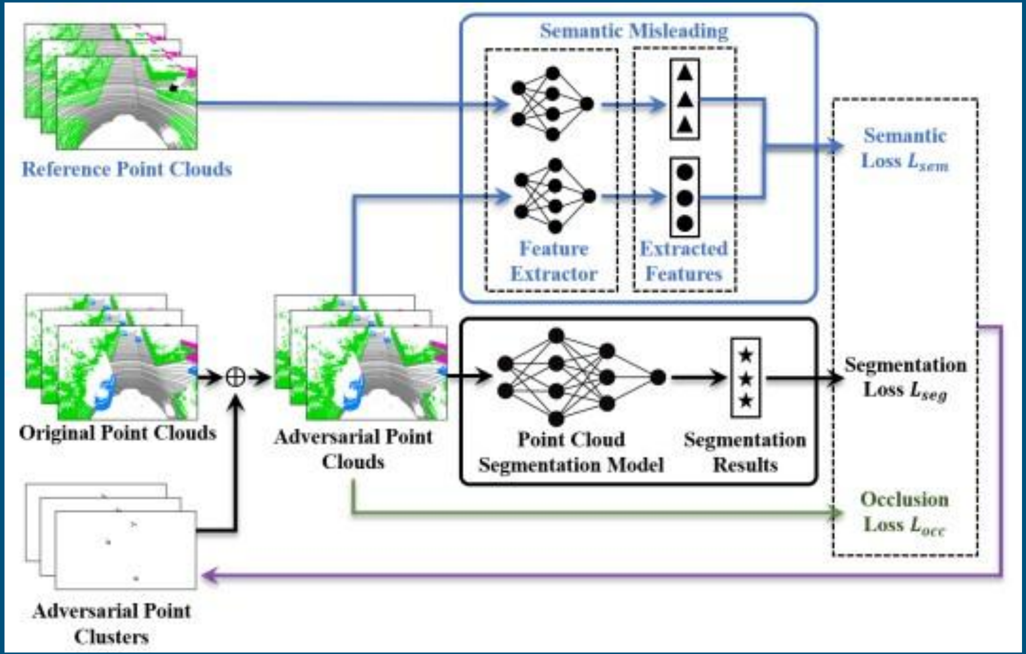
Segmentation Loss

- Measurement of distance between target label and predicted (actual) label
- Target points not misclassified -> positive loss value
- Smaller confidence means larger positive values
- Sum of these values

Occlusion Loss

- Unique from the other two loss variables
- Created to prevent adversarial clusters from being obstructed by other real-world objects
- Value is zero if not blocked
- High loss values for blocked clusters
- Sum of these values




Semantic Misleading
Reference Point Clouds
Feature Extractor
Extracted Features
Semantic Loss L_{sem}
Original Point Clouds
Adversarial Point Clouds
Point Cloud Segmentation Model
Segmentation Results
Segmentation Loss L_{seg}
Occlusion Loss L_{occ}
Adversarial Point Clusters

Total Loss Function

- $L_t = L_{seg} + \alpha L_{sem} + \beta L_{occ}$
- Alpha and beta are predefined hyper-parameters
- Seek to minimize
- L_t' is the gradient of the loss function
- Indicates how small perturbations change the loss
- Minimizing this allows for finding of optimal values tolerable to perturbations
- Resistant to location errors

White-Box Attack

- We know the semantic segmentation model used in the victim AV’s LiDAR perception system
- Locations of adversarial objects need to be reasonable
- This is done through bounding boxes
- Locations of constrained adversarial clusters can be derived through optimization algorithm:
- Adam Optimizer is used to find optimized value of (pk1, pk2, pk3)

$$\min_{\{O_k^a\}_{k=1}^K} \quad L^* = L_t + \eta L_t'$$

$$\begin{aligned} \text{s.t.} \quad & \{x_{k1}^a\}_{k=1}^K \in [A_{min}, A_{max}], \\ & \{x_{k2}^a\}_{k=1}^K \in [B_{min}, B_{max}], \\ & \{x_{k3}^a\}_{k=1}^K \in [C_{min}, C_{max}], \end{aligned}$$

$$x_{k1}^a = \frac{(A_{max} - A_{min})}{2} \cdot tanh(p_{k1}) + \frac{(A_{max} + A_{min})}{2}$$

$$x_{k2}^a = \frac{(B_{max} - B_{min})}{2} \cdot tanh(p_{k2}) + \frac{(B_{max} + B_{min})}{2}$$

$$x_{k3}^a = \frac{(C_{max} - C_{min})}{2} \cdot tanh(p_{k3}) + \frac{(C_{max} + C_{min})}{2}$$

Outline

- Intro
  - AVs
  - LiDAR
- Background
  - Neural Networks
  - Adversarial Examples
  - LiDAR Data Processing
- Attack Scenarios
- Adversarial Location Generation
  - Attack Framework
  - Loss Variables
  - Total Loss Function
  - White-box Attack
- **Attack Execution**
  - Setup
  - Results
    - SemanticKITTI Dataset
    - Real-World
- Conclusion

Attack Execution

Setup

- Attack is detailed in *Adversarial Attacks against LiDAR Semantic Segmentation in Autonomous Driving*
- Use 5 different point cloud segmentation models on public dataset SemanticKITTI attack : PointNet, PointNet++, PointASNL , Cylinder 3D, SqueezeSeg
- Data collected through LiDAR-mounted (Ouster OS1-64) vehicle for real-world attack
- Collected on two campus roads and three parking lots
- Data manually labeled

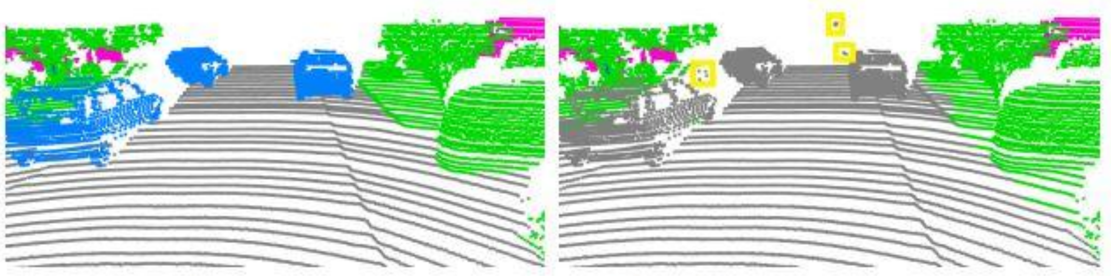
Results

SemanticKITTI

- 20 random scenes containing 5 consecutive point cloud frames
- Adversarial point clusters added to scenes
- Alpha, beta, and eta hyper-parameters set to 0.1, 1, 0.1
- Adam Optimizer set to 0.1
- After locations are derived, adversarial point clusters are replaced 100 times and results recorded
- Average is found for attack success rate

Models	Vehicle Hiding	Road Surface Changing
PointNet	82%	78%
SqueezeSeg	77%	66%
Cylinder3D	72%	63%
PointNet++	69%	60%
PointASNL	62%	58%

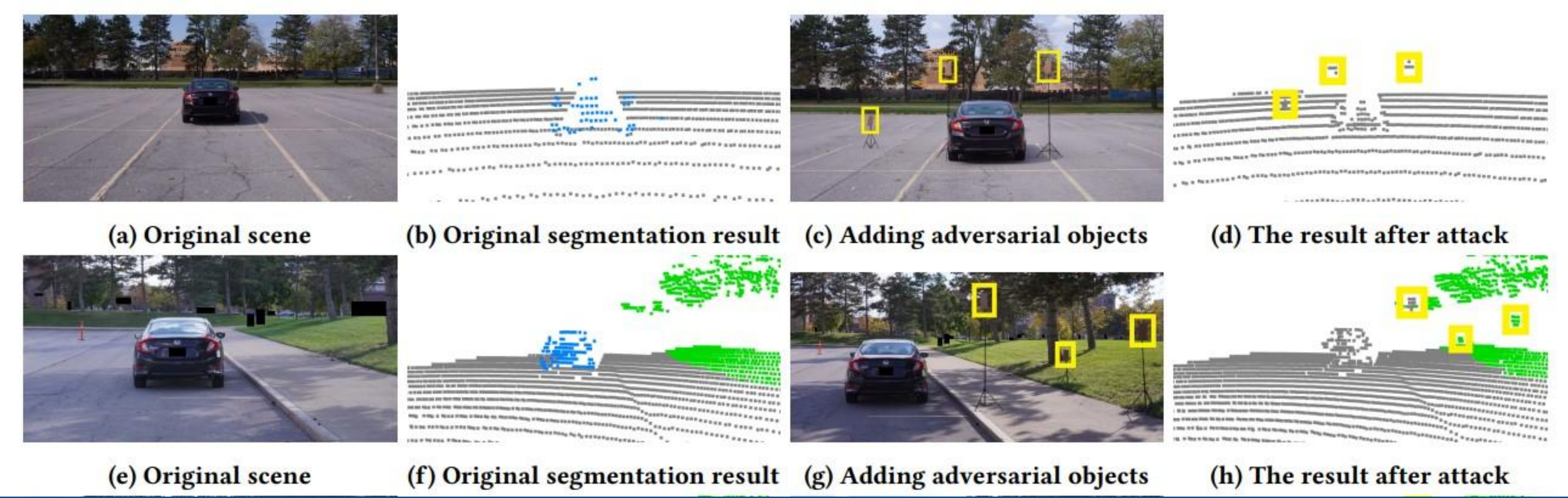
Table 1: Success rates of attacks using SemanticKITTI data on different segmentation models [7]



(a) Original segmentation result **(b) The result after attack**

Real-World Attacks

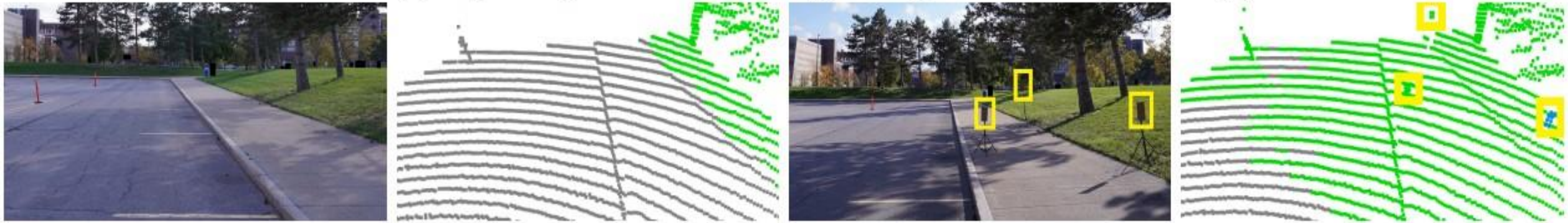
Vehicle Hiding Attack



(a) Original scene (b) Original segmentation result (c) Adding adversarial objects (d) The result after attack

(e) Original scene (f) Original segmentation result (g) Adding adversarial objects (h) The result after attack

Road Surface Changing Attack



(i) Original scene (j) Original segmentation result (k) Adding adversarial objects (l) The result after attack

Outline

- Intro
  - AVs
  - LiDAR
- Background
  - Neural Networks
  - Adversarial Examples
  - LiDAR Data Processing
- Attack Scenarios
- Adversarial Location Generation
  - Attack Framework
  - Loss Variables
  - Total Loss Function
  - White-box Attack
- Attack Execution
  - Setup
  - Results
    - SemanticKITTI Dataset
    - Real-World
- **Conclusion**

Conclusion

Conclusion

- High success rates in adversarial attacks
- Vulnerability of LiDAR

Questions?

References

- https://dl.acm.org/doi/pdf/10.1145/3485730.3485935
- https://dl.acm.org/doi/pdf/10.1145/3319535.3339815
- https://www.aljazeera.com/wp-content/uploads/2022/12/2022-12-13-Waymo-Test-3.jpg?resize=1800%2C1676
- https://japan-forward.com/wp-content/uploads/2021/03/HONDA-AUTONOMOUS_LEGEND-009-scaled.jpg
- https://s3-prod.autonews.com/s3fs-public/OEM06_310269985_AR_-1_XBTYTHYOMPQZ.jpg
- https://miro.medium.com/v2/resize:fit:640/format:webp/1*QJtTdtPhikY3KywV44L0Pw.png
- https://editor.analyticsvidhya.com/uploads/25366Convolutional_Neural_Network_to_identify_the_image_of_a_bird.png
- https://pytorch.org/tutorials/_images/fgsm_panda_image.png
- https://www.researchgate.net/publication/358114298/figure/fig2/AS:11431281099289035@1669249094251/The-impact-of-adversarial-attacks-in-an-application-related-to-AVs-such-as-segmentation.ppm