Towards Robust Perception for Autonomous Vehicles

Dauwels Lab
@ DauwelsL
www.dauwels.com

1 Oct 2019

Justin Dauwels

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Machine Learning in the Media…

The New York Times

A Machine May Not Take Your Job, but One Could Become Your Boss

THE WALL STREET JOURNAL.

THE FUTURE OF EVERYTHING

White-Collar Robots Are Coming for Jobs

The Guardian

Robots will destroy our jobs – and we're not ready for it

The Washington Post

Technology

As Walmart turns to robots, it's the human workers who feel like machines

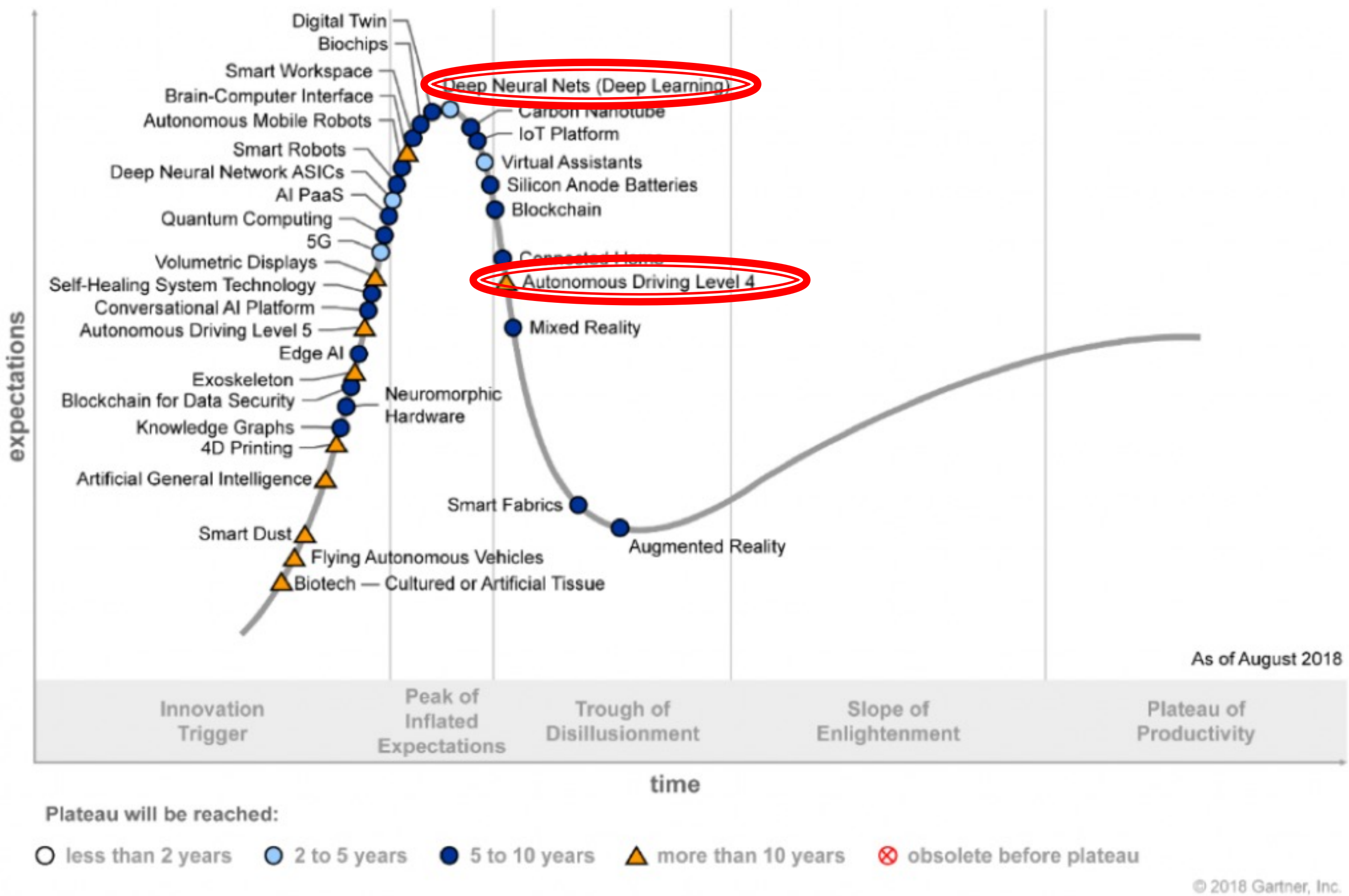
New York Times: https://www.nytimes.com/2019/06/23/technology/artificial-intelligence-ai-workplace.html

The Guardian: https://www.theguardian.com/technology/2017/jan/11/robots-jobs-employees-artificial-intelligence

The Wall Street Journal: https://www.wsj.com/articles/white-collar-robots-are-coming-for-jobs-11548939601

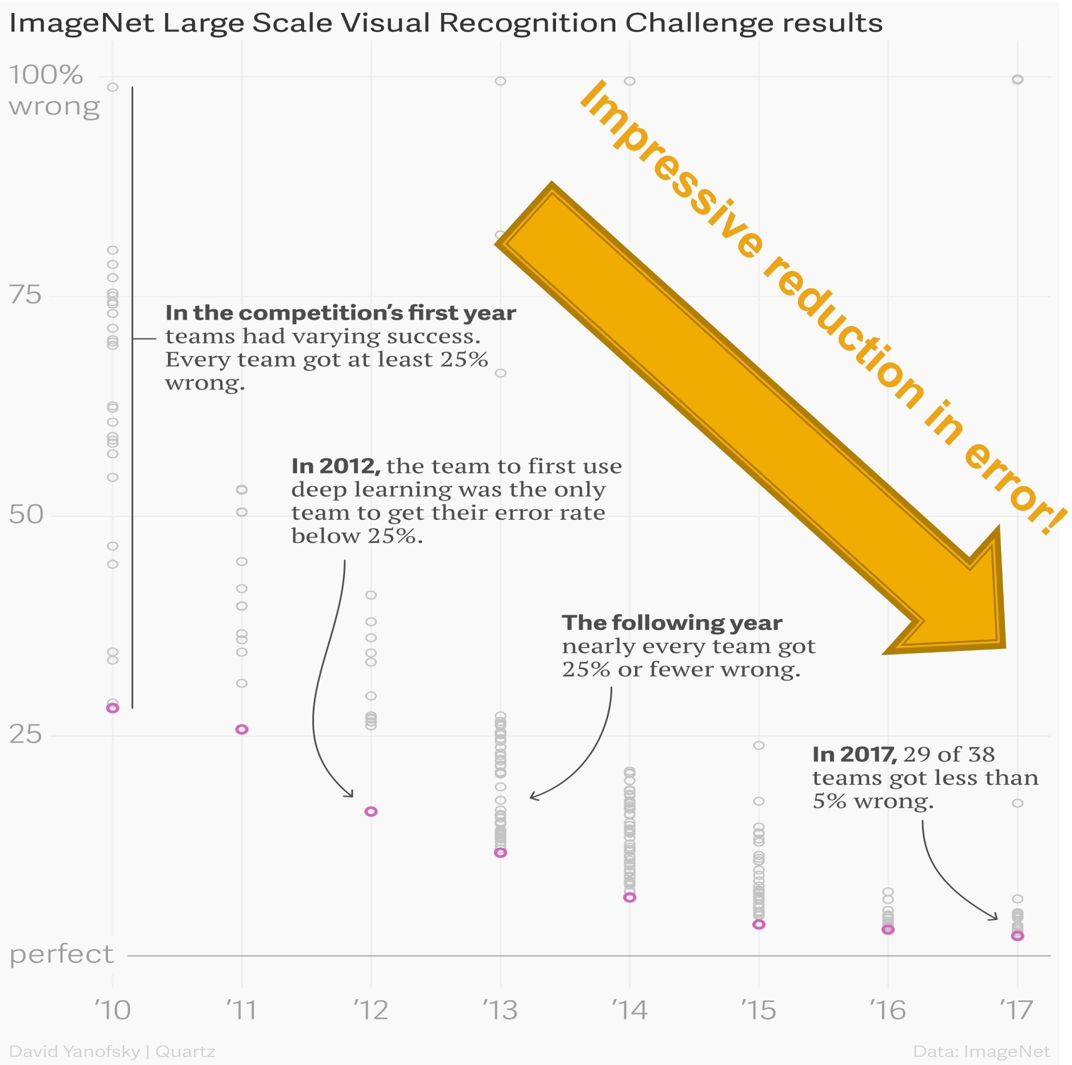
The Washington Post:https://www.washingtonpost.com/technology/2019/06/06/walmart-turns-robots-its-human-workers-who-feel-like-machines/?noredirect=on&utm_term=.ffce4a05d2f6

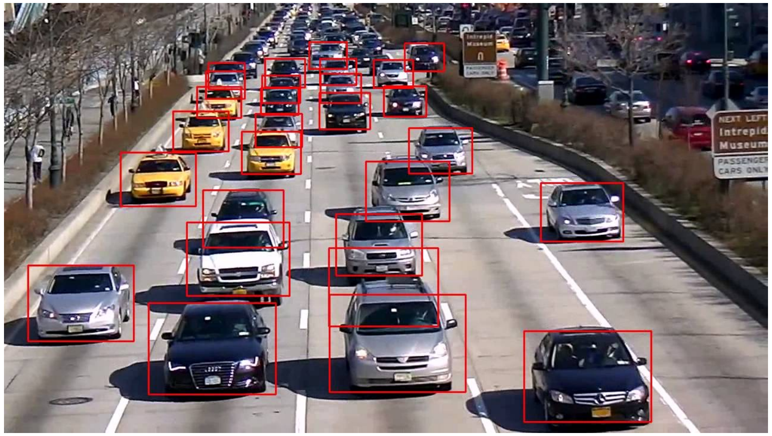
Gartner Hype Cycle

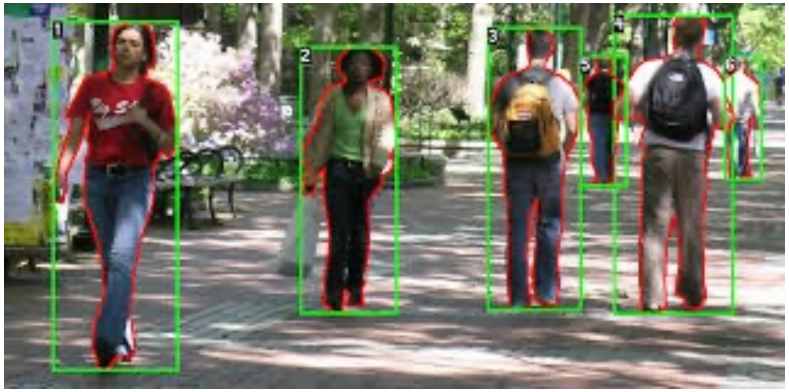


Reference: https://www.criticallink.com/2018/11/gartners-latest-emerging-technologies-hype-cycle/

Success Story: Computer Vision







ImageNet Benchmark: https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/

Overview

- Development of New ML Methods
  - When does deep learning fail?
  - How can we fix it?
- What does that mean for AVs?
  - Development of models of perception errors
  - Simulation of AVs with realistic perception
- Conclusions and Future Work

Overview

- **Development of New ML Methods**
  - When does deep learning fail?
  - How can we fix it?
- What does that mean for AVs?
  - Development of models of perception errors
  - Simulation of AVs with realistic perception
- Conclusions and Future Work

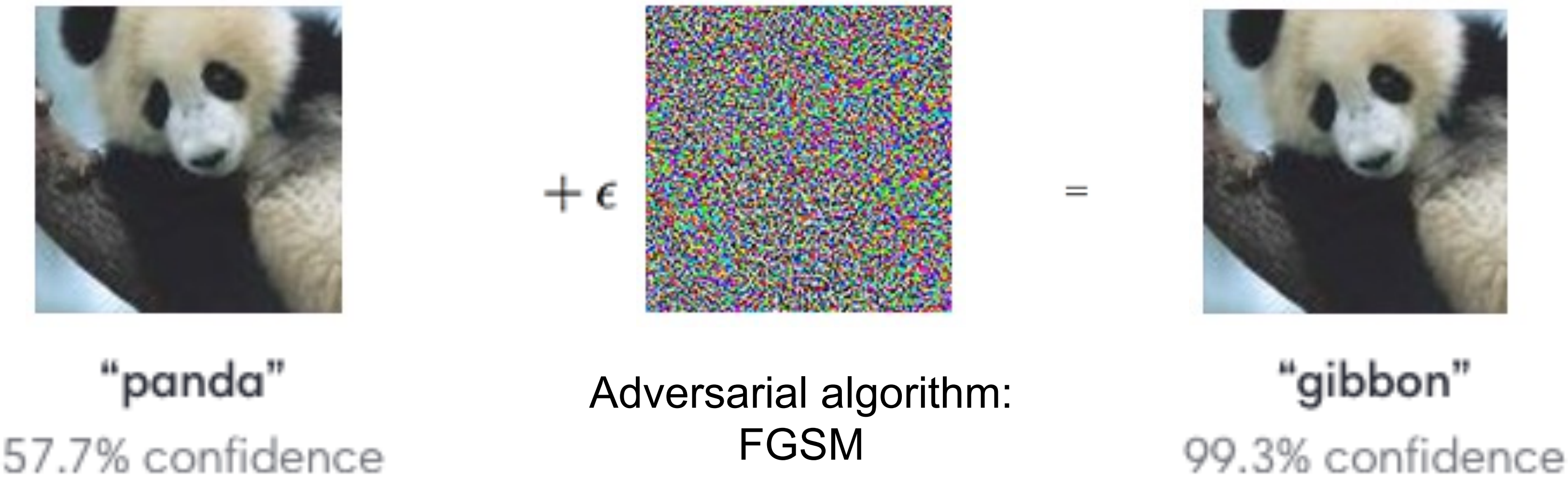
What do you see?

This is what GoogleNet sees…

Going deeper with convolutions, Christian Szegedy, 2014, https://arxiv.org/abs/1409.4842
Intriguing properties of neural networks, Christian Szegedy, 2013, https://arxiv.org/abs/1312.6199

Adversarial Attack

- Classification Results (GoogLeNet)



Adversarial algorithm:
FGSM

Observation:
- Deep Learning is **fragile**.
- There is **gap** between human perception and deep learning.

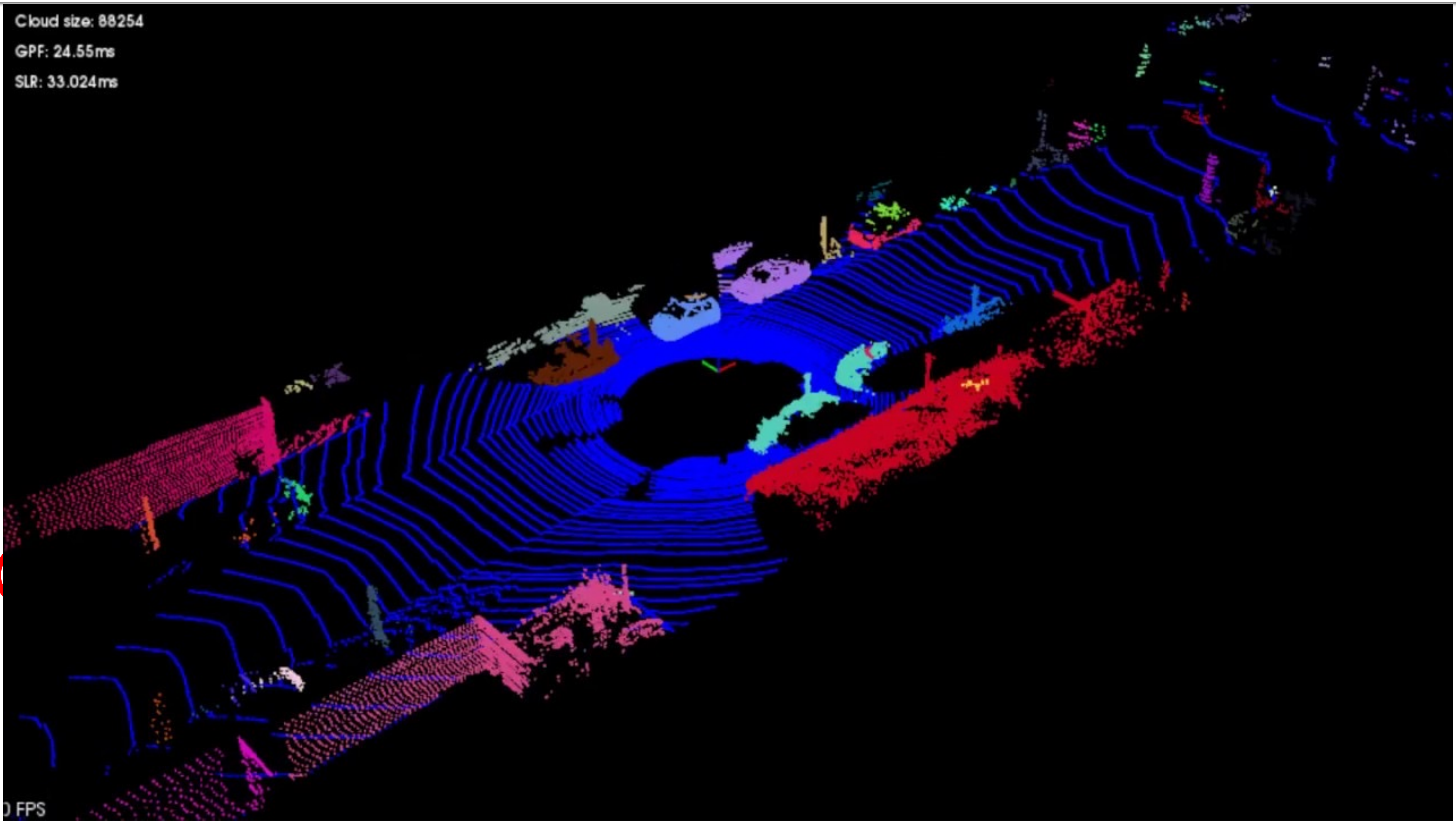
Going deeper with convolutions, Christian Szegedy, 2014, https://arxiv.org/abs/1409.4842
Intriguing properties of neural networks, Christian Szegedy, 2013, https://arxiv.org/abs/1312.6199

Also in real-life situations!

- **Raindrop** was wrongly detected as a **car** by Fast R-CNN

Fast R-CNN, Ross Girshick, 2015, https://arxiv.org/abs/1504.08083

Driverless Car Perception System

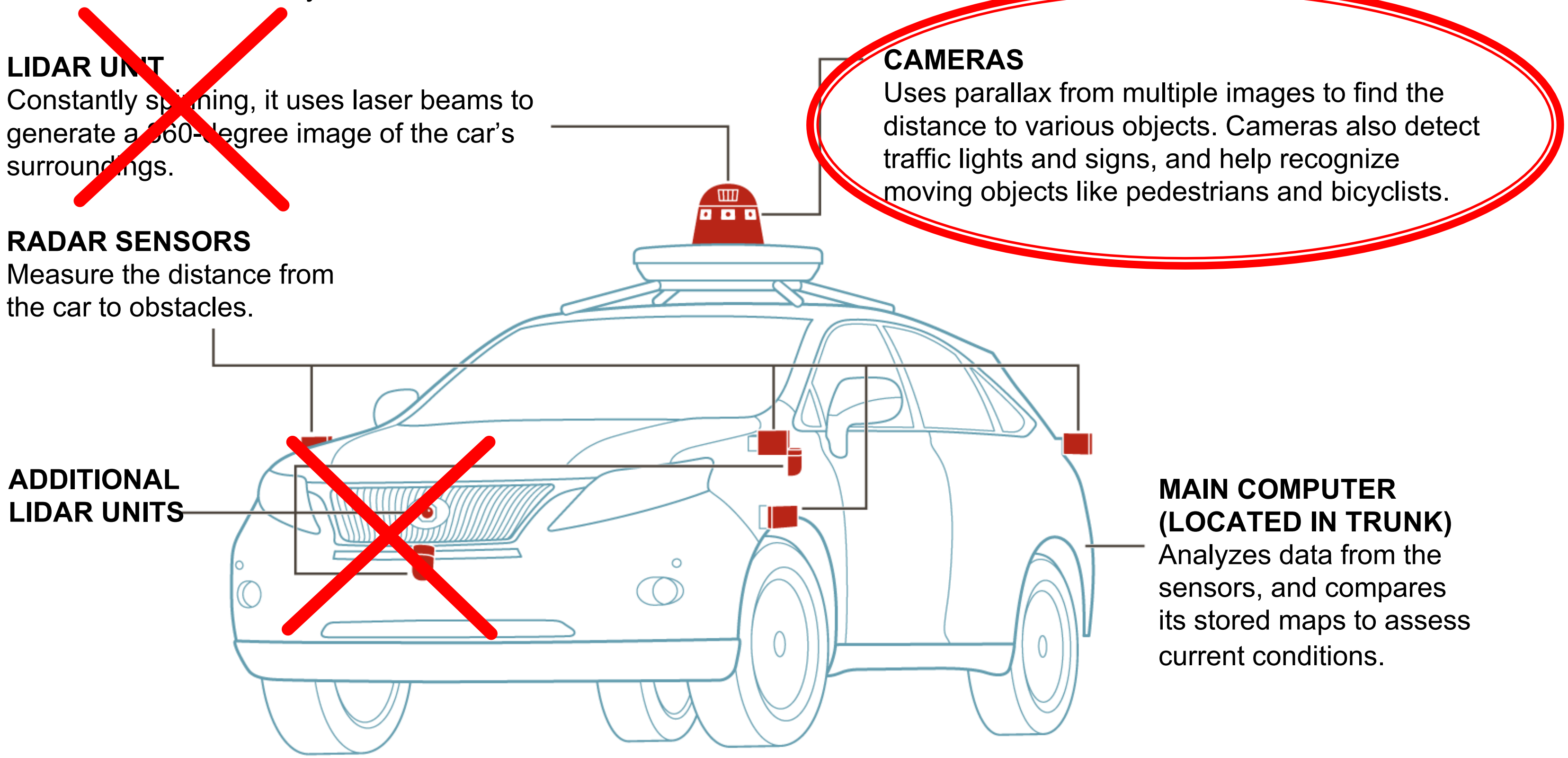


LIDAR = **Li**ght **D**etection **a**nd **R**anging

Driverless Car Perception System

What the Car Sees

The car's sensors gather data on nearby objects, like their size and rate of speed. It categorizes the objects — as cyclists, pedestrians or other cars and objects — based on how they are likely to behave.



Perception via camera only can be vulnerable

THE DAILY NEWSLETTER
Sign up to our daily email newsletter

NewScientist

News Technology Space Physics Health Environment Mind Video | Tours Events Jobs

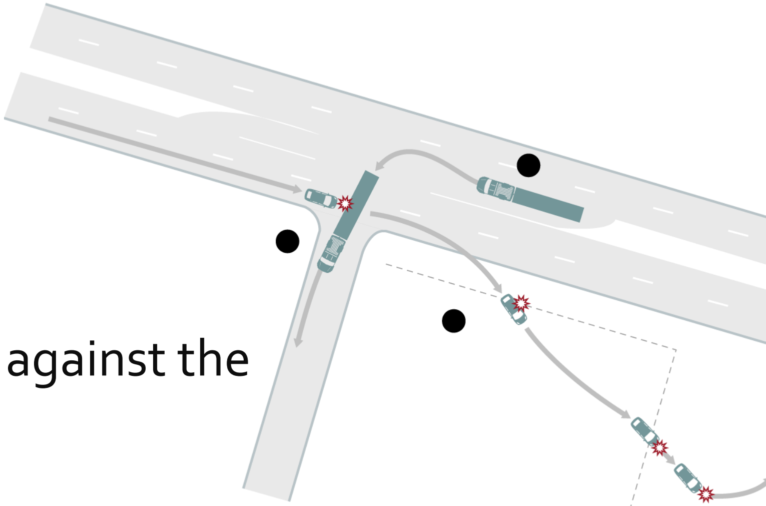
Tesla driver dies in first fatal autonomous car crash in US

TECHNOLOGY 1 July 2016

Source: https://www.newscientist.com/article/2095740-tesla-driver-dies-in-first-fatal-autonomous-car-crash-in-us/

Perception via camera only can be vulnerable

- The First **Driver Casualty** in Self-Driving Car Accident
  - 7th May 2016, Florida US
  - Tesla Model S, Autopilot mode
  - Driver passed away
  - The system didn't distinguish the white truck against the brightly lit sky, and failed to apply brakes.



→ **Could have been prevented if the system had LIDAR sensors**

Source: https://www.newscientist.com/article/2095740-tesla-driver-dies-in-first-fatal-autonomous-car-crash-in-us/

Also *rotations* confuse the detectors

- We applied *Inception* to rotated images of car

Rethinking the Inception Architecture for Computer Vision, Christian Szegedy, 2015, https://arxiv.org/abs/1512.00567

Also *rotations* confuse the detectors (2)

- Real-life scenario

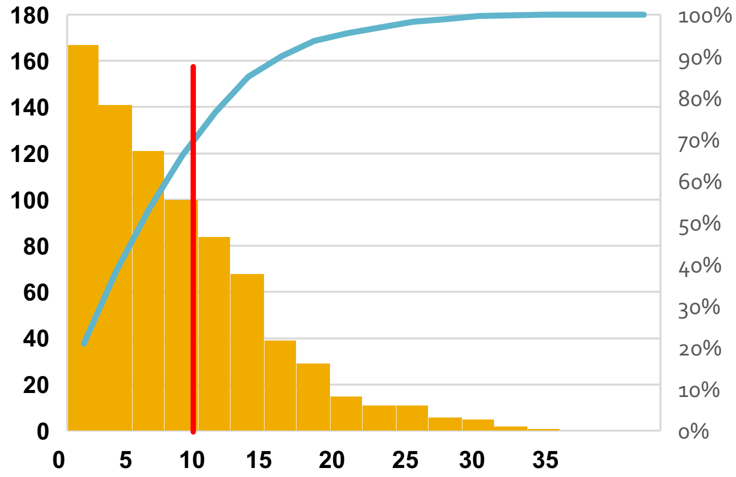
Image source: https://metro.co.uk/2008/07/07/car-puddle-soakings-new-youth-craze-252530/

Why are the detectors confused by rotations?

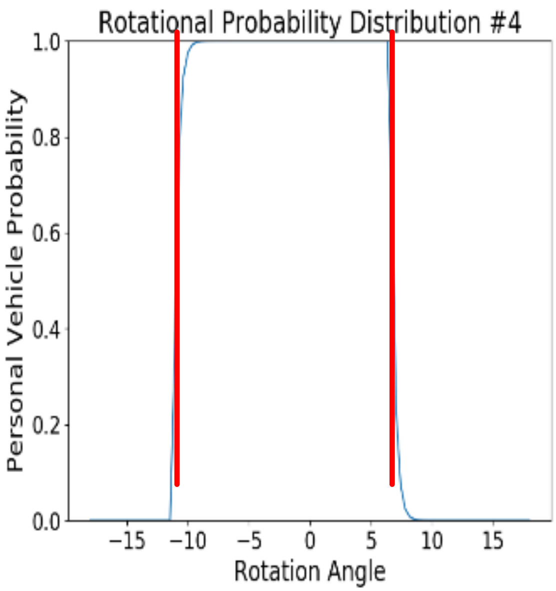
- *Inception* is trained on **ImageNet** dataset
- **73%** of the images of cars in ImageNet has rotation angle < 10º

→ **The detector *Inception* has never "seen" rotated cars before…**

Rotation angle



Histogram of rotation angles



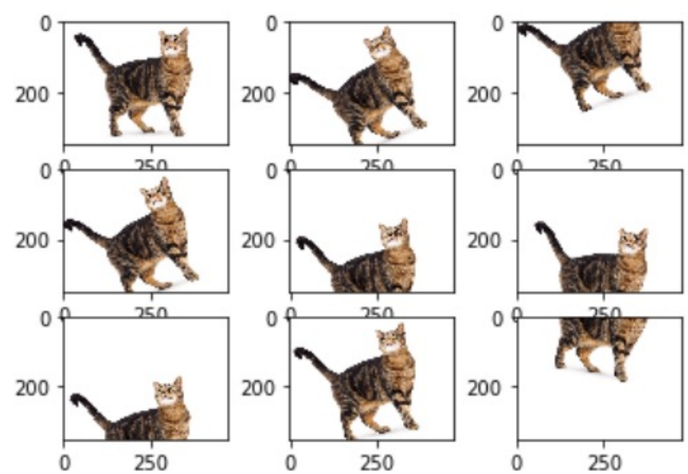
Detection rate of *Inception*

Image Source: http://image-net.org/synset?wnid=n03770679 **Rethinking the Inception Architecture for Computer Vision, Christian Szegedy, 2015,** https://arxiv.org/abs/1512.00567

How To Make the Detector More Robust?

TRADITIONAL APPROACH: Augment the Training Data

- Cropping, rotation, flipping input images



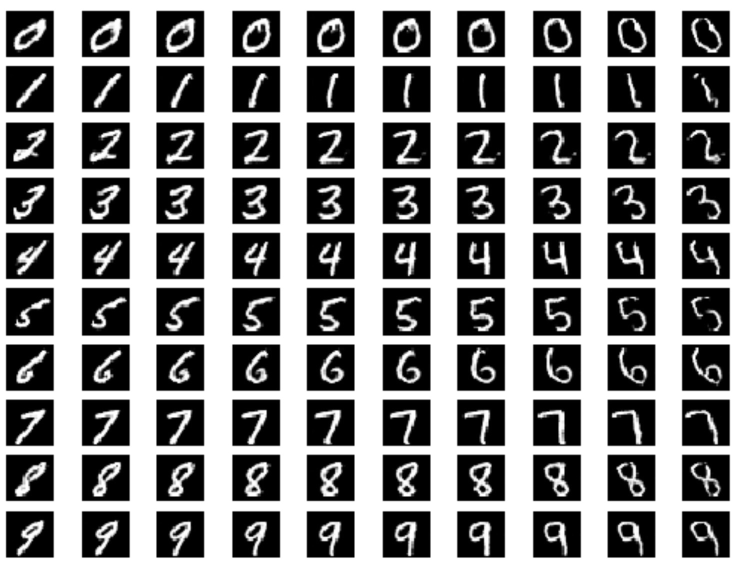
- Cumbersome!
- The algorithm does not become "smarter"

Image source: https://stackoverflow.com/questions/44819922/data-augmentation-is-shifting-needed

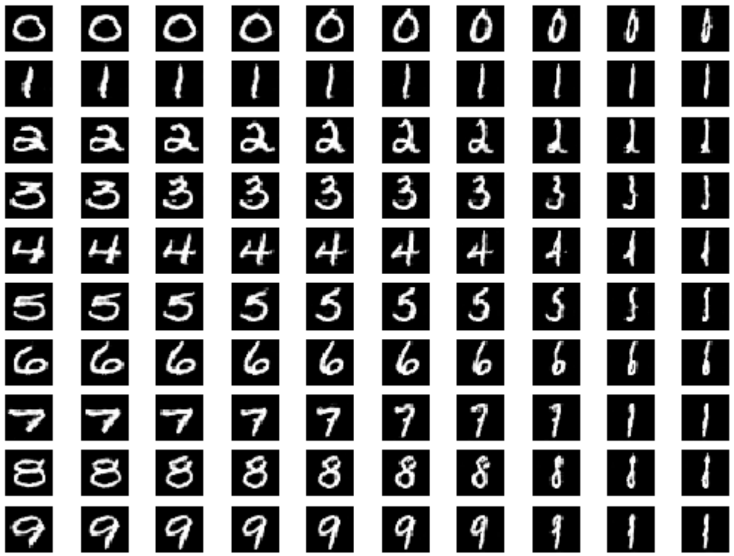
How To Make the Detector More Robust?

PROPOSED APPROACH: Disentangled Representation

Incorporate **physical variables** into the detector (e.g., rotation angle)



Samples from a model with latent vectors encoding **rotation**



Samples from a model with latent vectors encoding **horizontal scaling**

Affine Matrix is Encoded into the Detector

1. Skew Matrix

$$M = \begin{bmatrix} 1 & m \\ n & 1 \end{bmatrix}$$

3. Rotation Matrix

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2. Zoom Matrix

$$M = \begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}$$

4. Affine Matrix

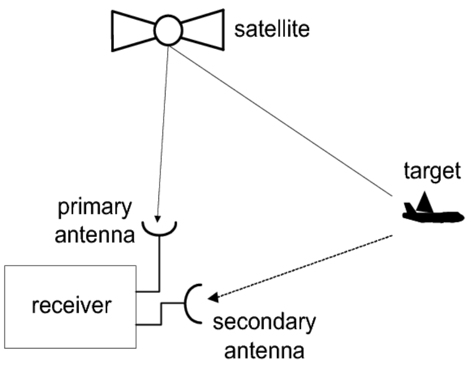
$$M = \begin{bmatrix} 1 & m \\ n & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}$$

Two different fields

- Statistical Signal Processing

$$\hat{\theta} = \arg\max_{\theta} p(Data; \theta)$$

  - θ are parameters (e.g., physical variables)
  - p is known model (e.g., pulse design, environment)



- Machine Learning (Deep Learning)

$$\hat{w} = \arg\max_{w} p(Data; w)$$

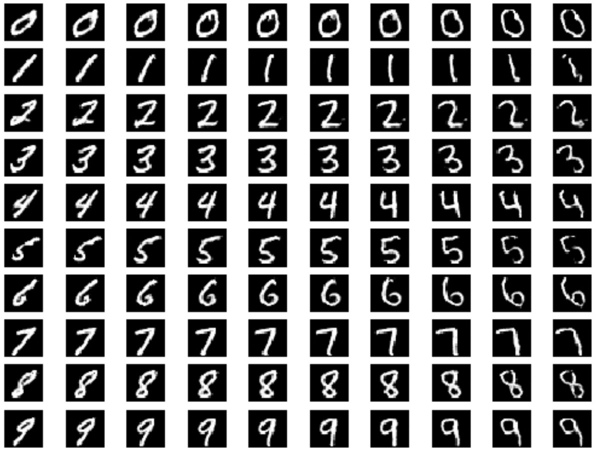
  - w are "weights" (millions or more!)
  - p is "flexible" function (e.g., neural network)

Let's take the best of both worlds!

- Interpretable Machine Learning

$$\hat{\theta}, \hat{w} = \arg\max_{\theta, w} p(Data; \theta, w)$$

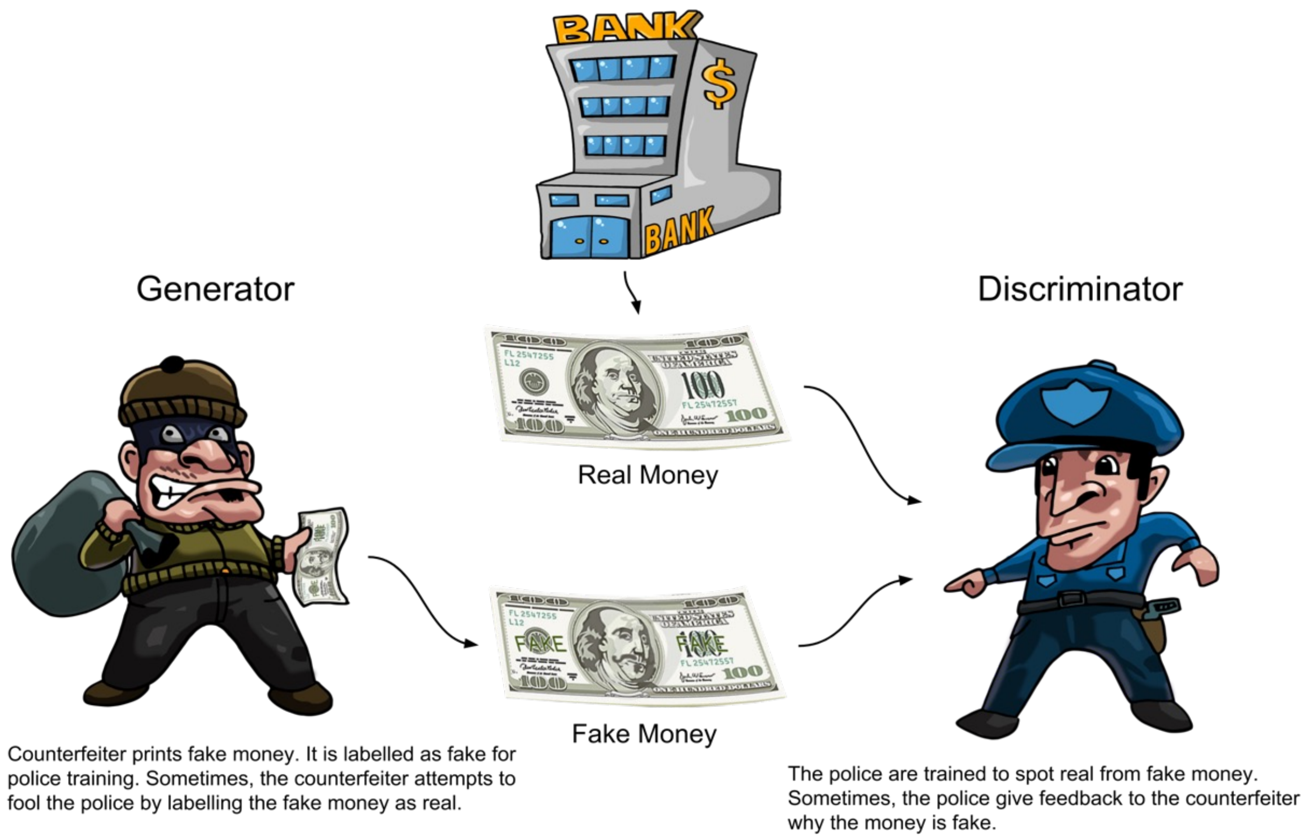
  - θ are parameters (e.g., physical variables)
  - w are "weights" (millions or more!)
  - *p* is model with "flexible" and "known" parts



Model *p* is able to generate **rotated** digits

Generative Adversarial Network (GAN)

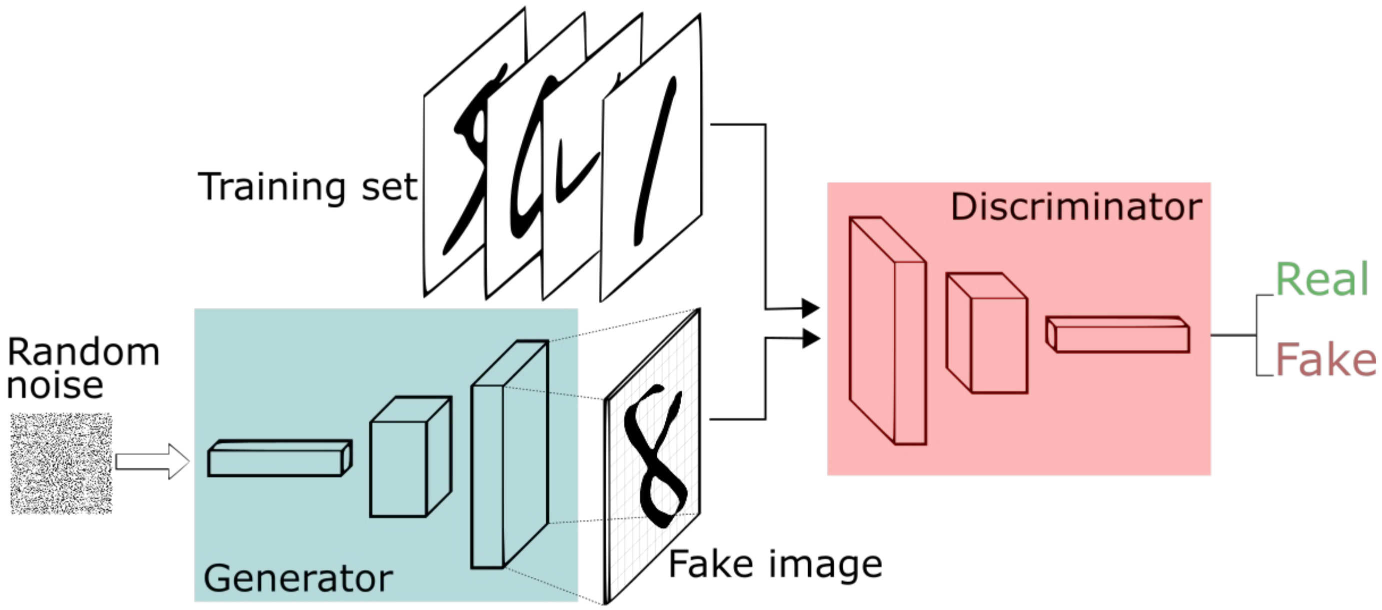
- GAN Principle: Minimax/adversarial game: counterfeiter & police



GAN working principle illustration

Generative Adversarial Networks, Ian J. Goodfellow, 2014, https://arxiv.org/abs/1406.2661

Standard GAN

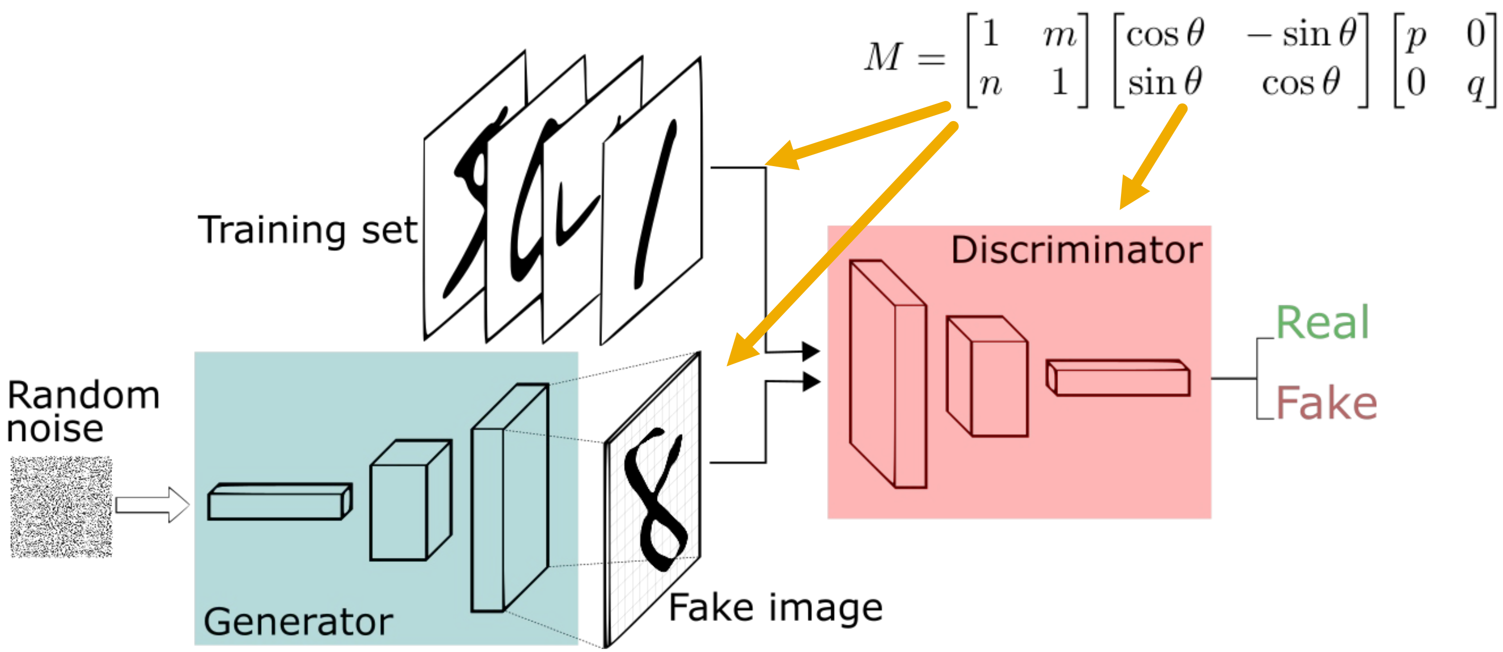


D should be 1 if “real”, 0 otherwise

$$\min_G \max_D V(D,G) := \mathbb{E}_{x \sim P_{\text{data}}(x)} \log[D(x)] + \mathbb{E}_{z \sim P_z(z)} \log[1 - D(G(z))].$$

Generative Adversarial Networks, Ian J. Goodfellow, 2014, https://arxiv.org/abs/1406.2661

Affine Disentangled GAN (ADIS-GAN)

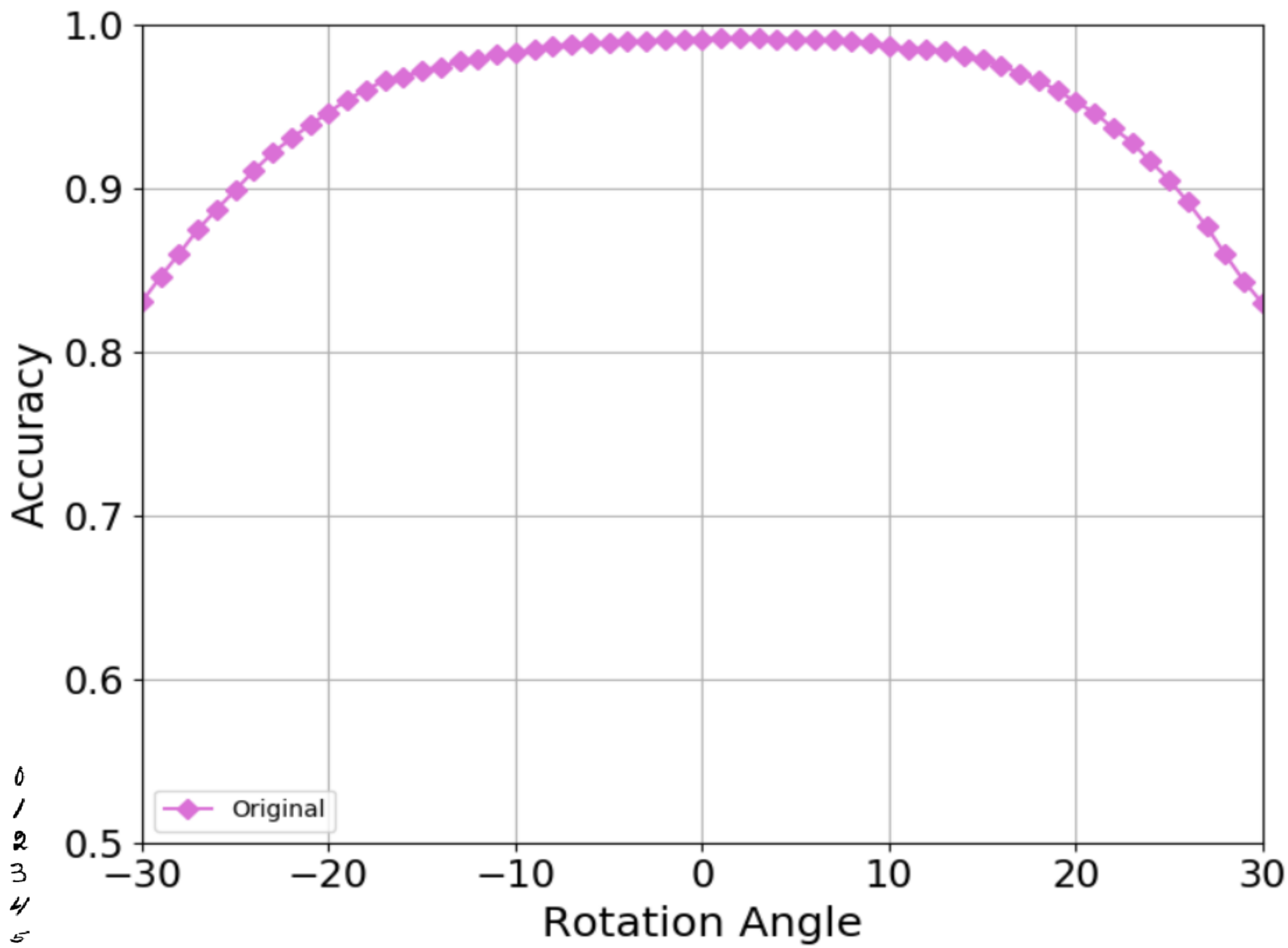


By introducing the affine matrix concept into the generation process, the generator "understands" the *data distribution (digits)* and *affine parameters* in an integrated way

$$\min_G \max_D V(D,G) := \mathbb{E}_{x\sim P_{\text{data}}(x)} \log[D(x)] + \mathbb{E}_{z\sim P_z(z)} \log[1 - D(G(z))] - \lambda L(M)$$

Generative Adversarial Networks, Ian J. Goodfellow, 2014, https://arxiv.org/abs/1406.2661

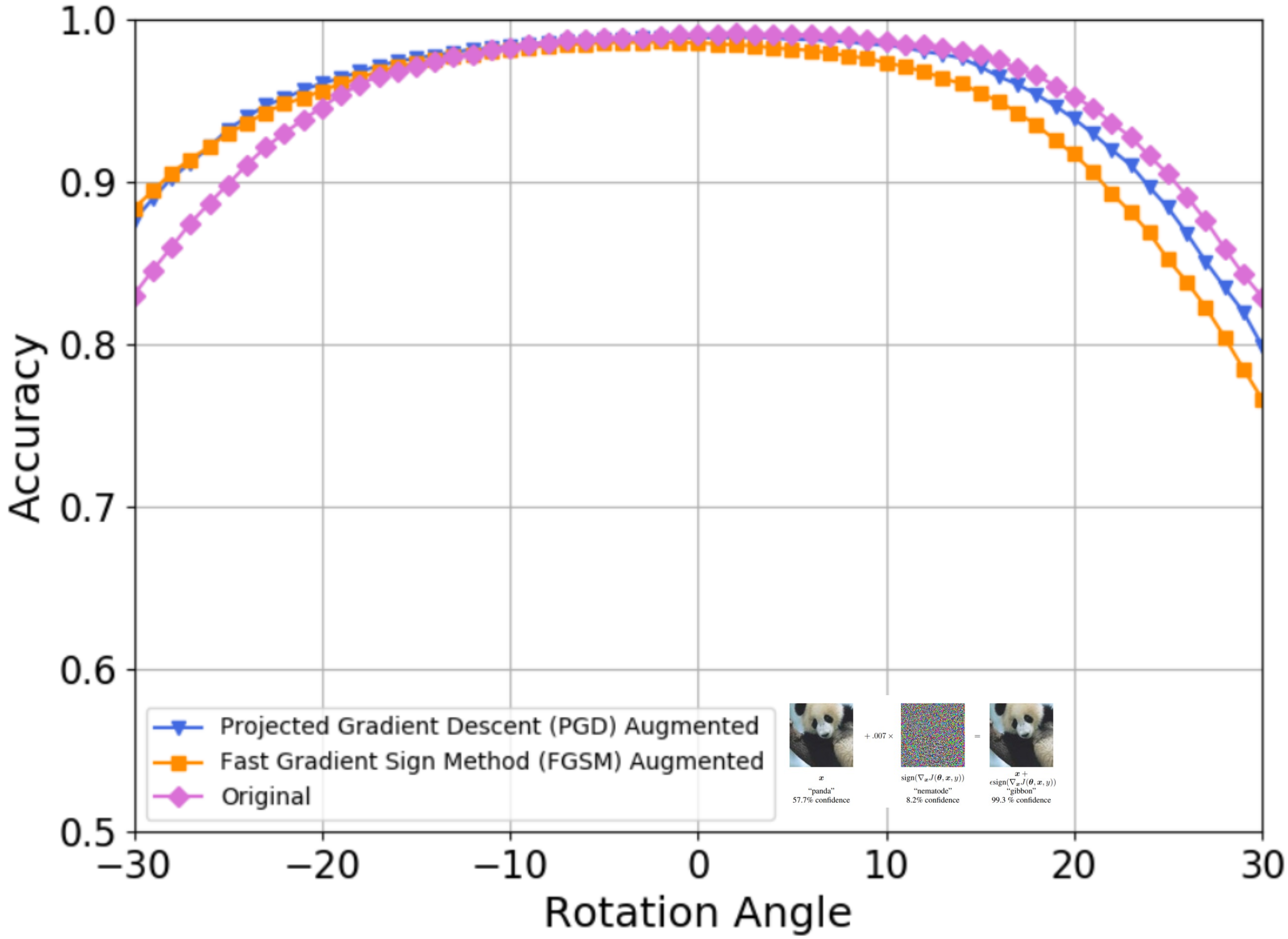
Detection performance for different rotation angles (MNIST dataset)



MNIST

CNN object detector is sensitive to rotations

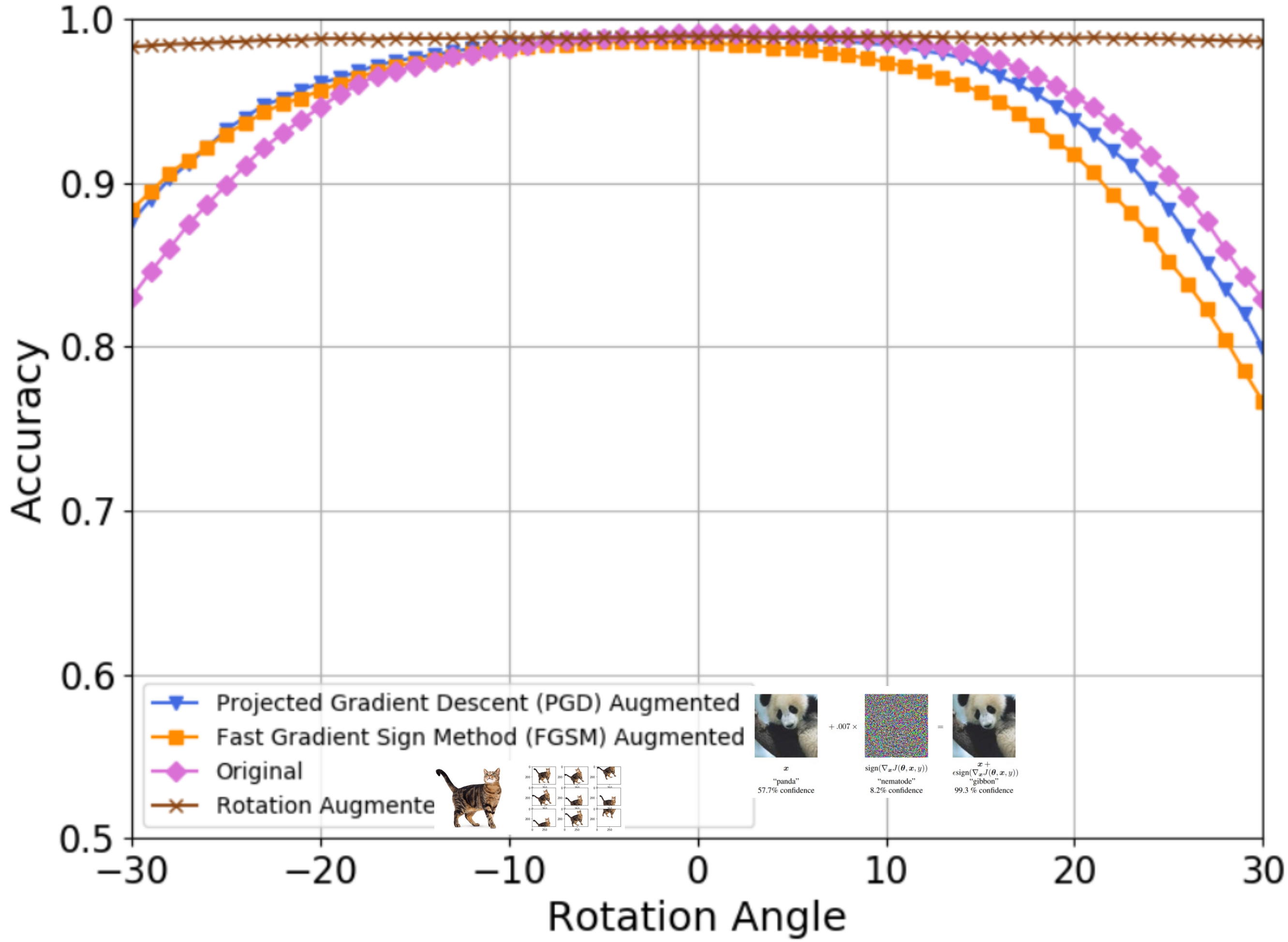
Detection performance for different rotation angles (MNIST dataset)



Training CNN object detector on datasets with adversarial examples does not help

Intriguing properties of neural networks, Christian Szegedy, 2013, https://arxiv.org/abs/1312.6199
Towards Deep Learning Models Resistant to Adversarial Attacks, Aleksander Madry, 2017, https://arxiv.org/abs/1706.06083

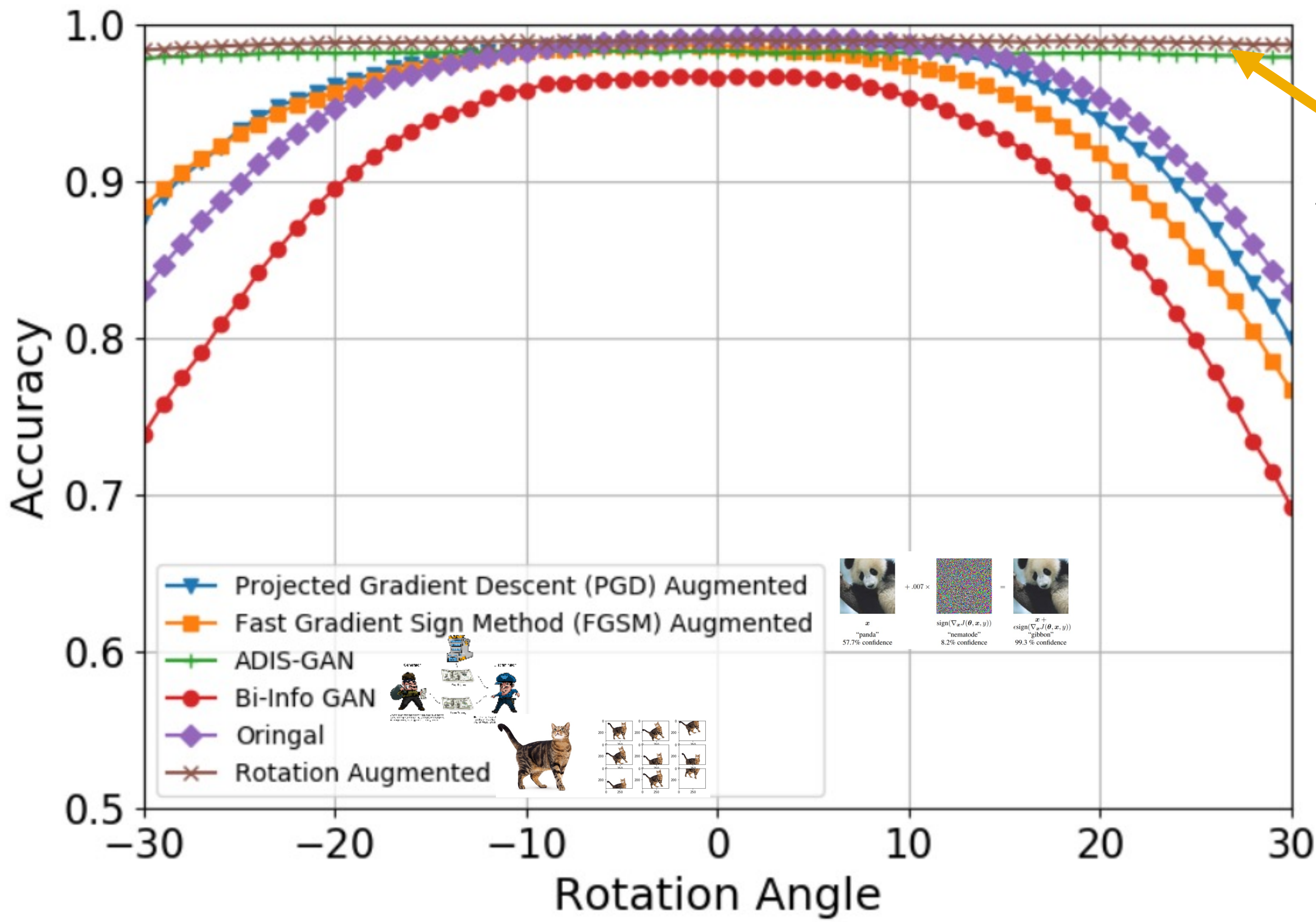
Detection performance for different rotation angles (MNIST dataset)



Training CNN object detector on datasets with rotated examples does help!

Intriguing properties of neural networks, Christian Szegedy, 2013, https://arxiv.org/abs/1312.6199
Towards Deep Learning Models Resistant to Adversarial Attacks, Aleksander Madry, 2017, https://arxiv.org/abs/1706.06083

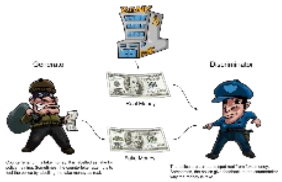
Detection performance for different rotation angles (MNIST dataset)

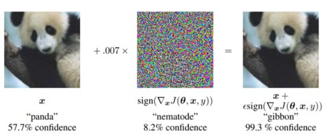


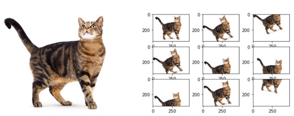
ADIS-GAN is robust!

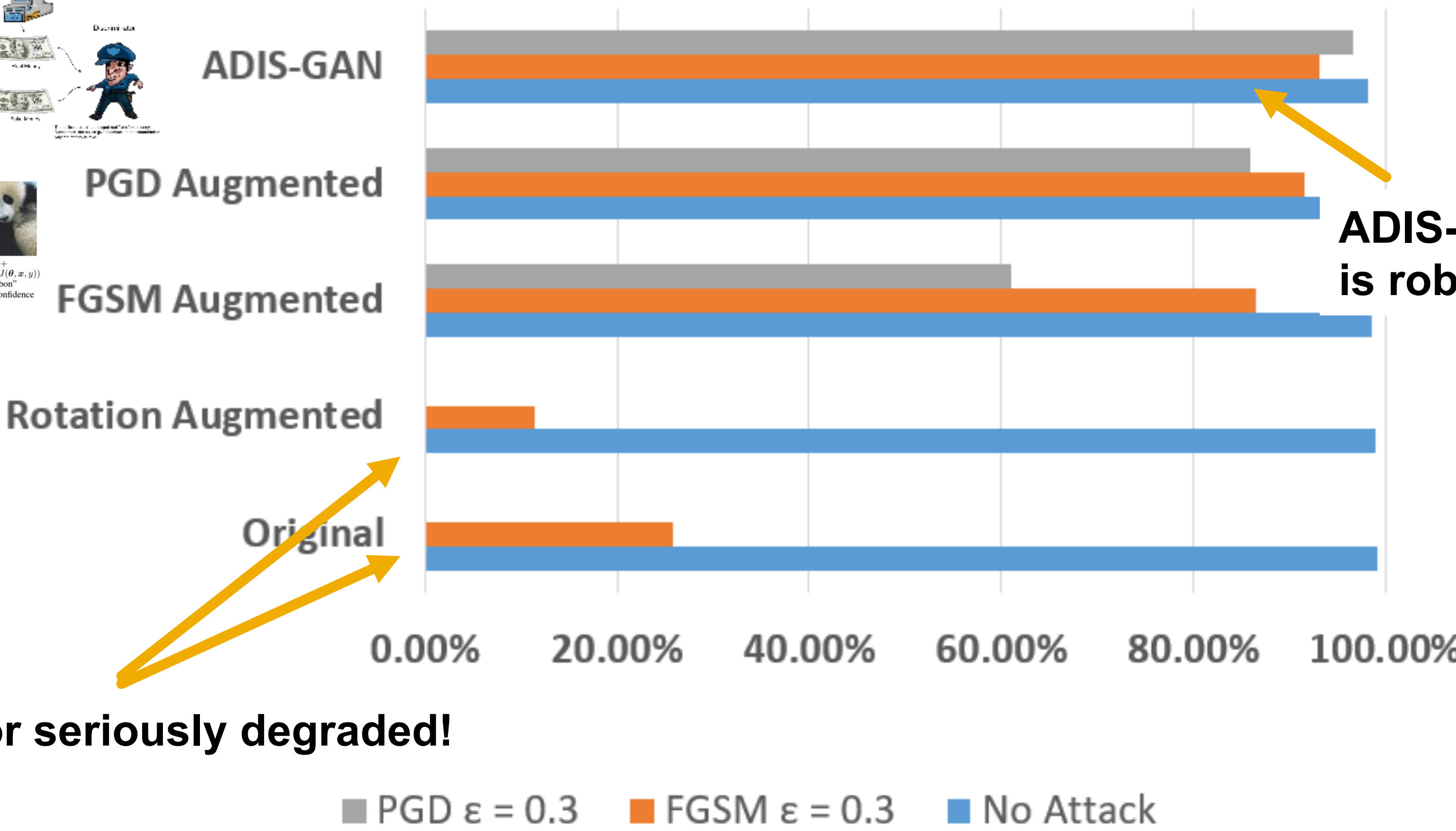
Intriguing properties of neural networks, Christian Szegedy, 2013, https://arxiv.org/abs/1312.6199
Towards Deep Learning Models Resistant to Adversarial Attacks, Aleksander Madry, 2017, https://arxiv.org/abs/1706.06083

Robustness Against Adversarial Attacks (MNIST dataset)









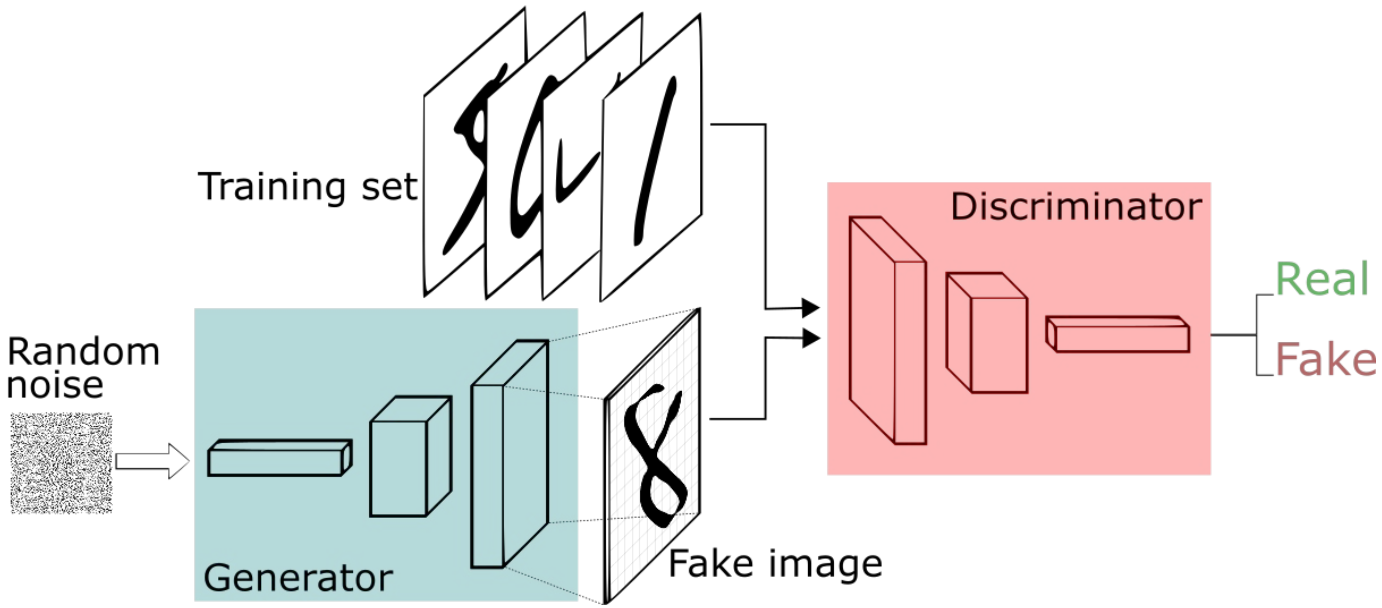
ADIS-GAN is robust!

CNN detector seriously degraded!

ADIS-GAN is robust against rotations and adversarial attacks!

Intriguing properties of neural networks, Christian Szegedy, 2013, https://arxiv.org/abs/1312.6199
Towards Deep Learning Models Resistant to Adversarial Attacks, Aleksander Madry, 2017, https://arxiv.org/abs/1706.06083

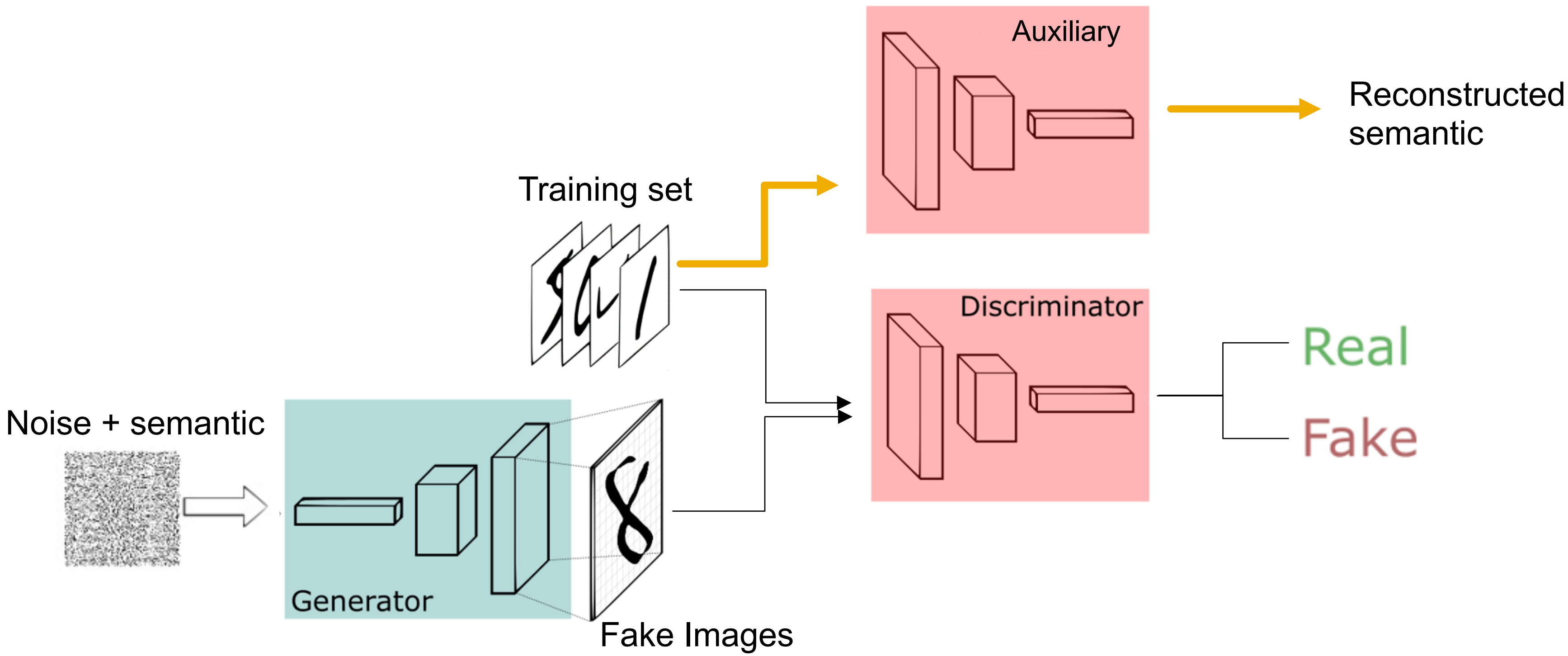
Standard GAN



$$\min_G \max_D V(D,G) := \mathbb{E}_{x\sim P_{\text{data}}(x)} \log[D(x)] + \mathbb{E}_{z\sim P_z(z)} \log[1 - D(G(z))].$$

Generative Adversarial Networks, Ian J. Goodfellow, 2014, https://arxiv.org/abs/1406.2661

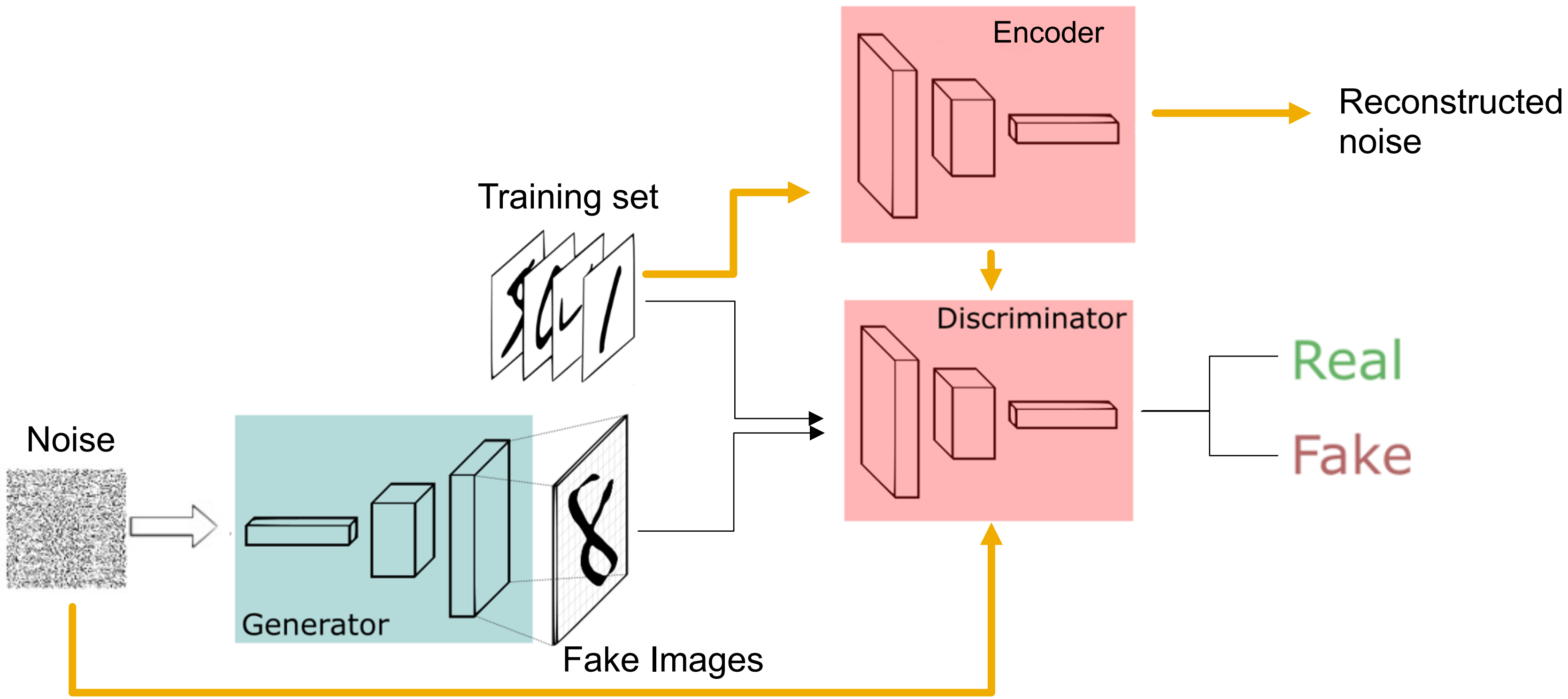
InfoGAN



$$\min_{G,Q} \max_{D} V(D,G,Q) := \mathbb{E}_{x\sim P_{\text{data}}(x)} \log[D(x)] + \mathbb{E}_{z\sim P_z(z),c\sim P_c(c)} \log[1 - D(G(c,z))] - \lambda L(G,Q).$$

InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, Xi Chen, 2016 : https://arxiv.org/abs/1606.03657

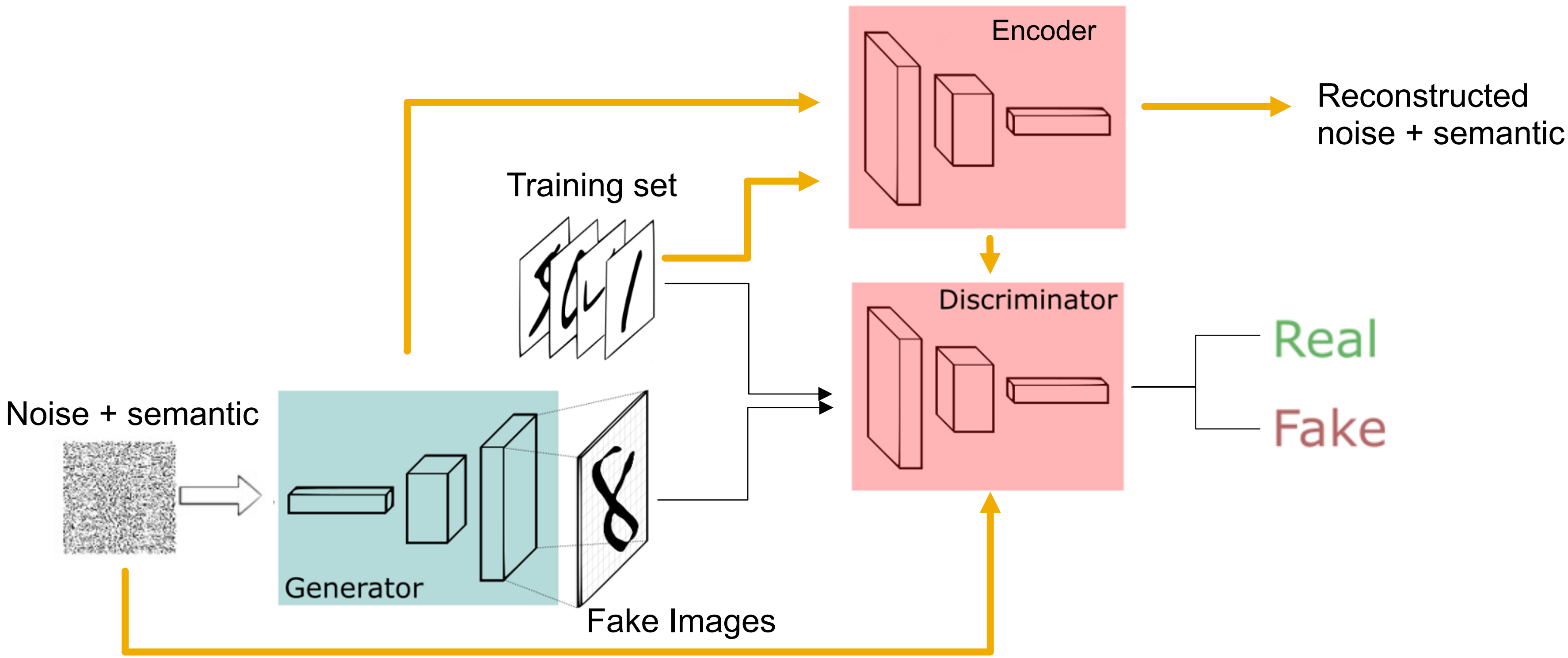
Bi-directional GAN



$$\min_{G,E} \max_{D} V(D,G,E) := \mathbb{E}_{x\sim P_{\text{data}}(x)} \log[D(x, E(x))] + \mathbb{E}_{z\sim P_z(z)} \log[1 - D(z, G(z))].$$

Adversarial Feature Learning, Jeff Donahue, 2016: https://arxiv.org/abs/1605.09782

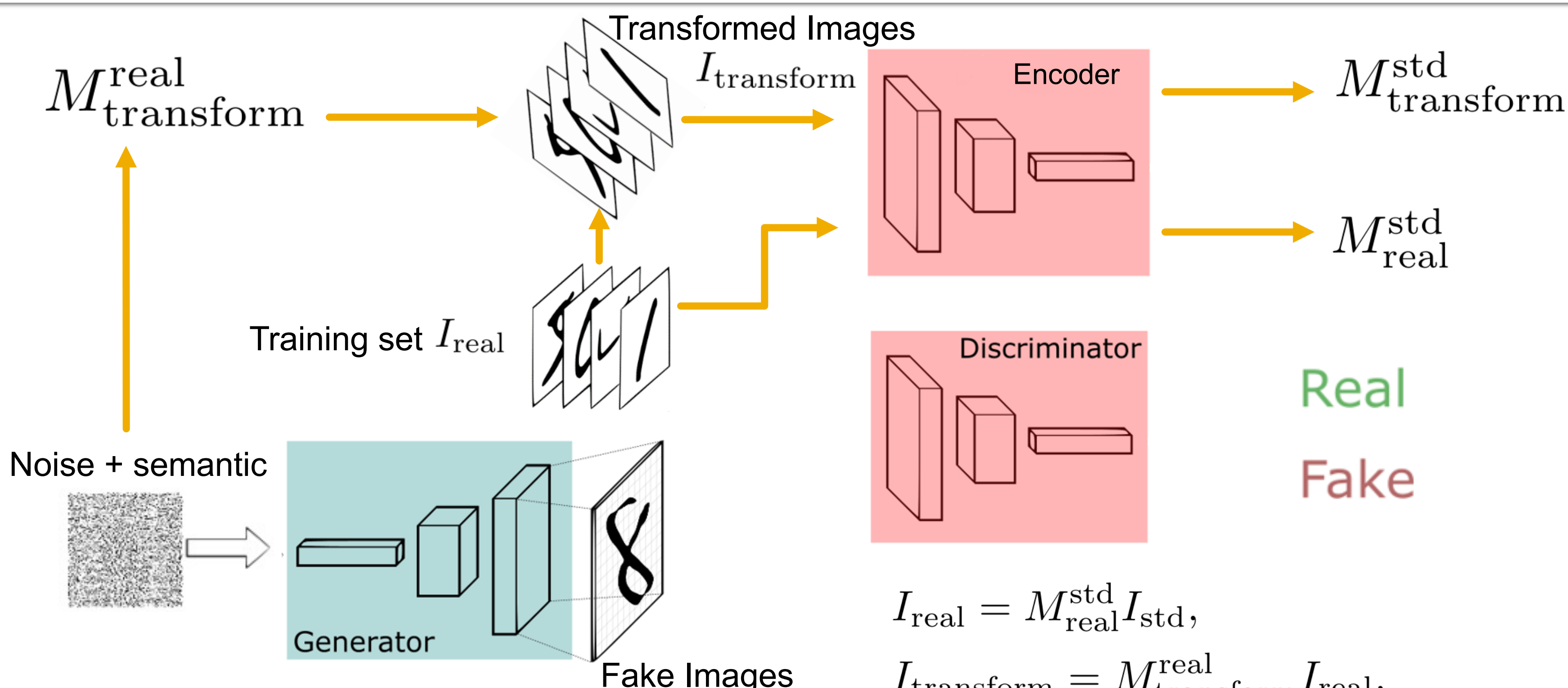
Bi-directional InfoGAN



$$\min_{G,E} \max_{D} V(D,G,E) := \mathbb{E}_{x \sim P_{\text{data}}(x)} \log[D(x, E(x))] + \mathbb{E}_{z \sim P_z(z), c \sim P_c(c)} \log[1 - D((c,z), G(c,z))] - \lambda L(G,E).$$

Inferencing Based on Unsupervised Learning of Disentangled Representations, Tobias Hinz, 2018: https://arxiv.org/abs/1803.02627

Affine Disentangled GAN (ADIS-GAN)



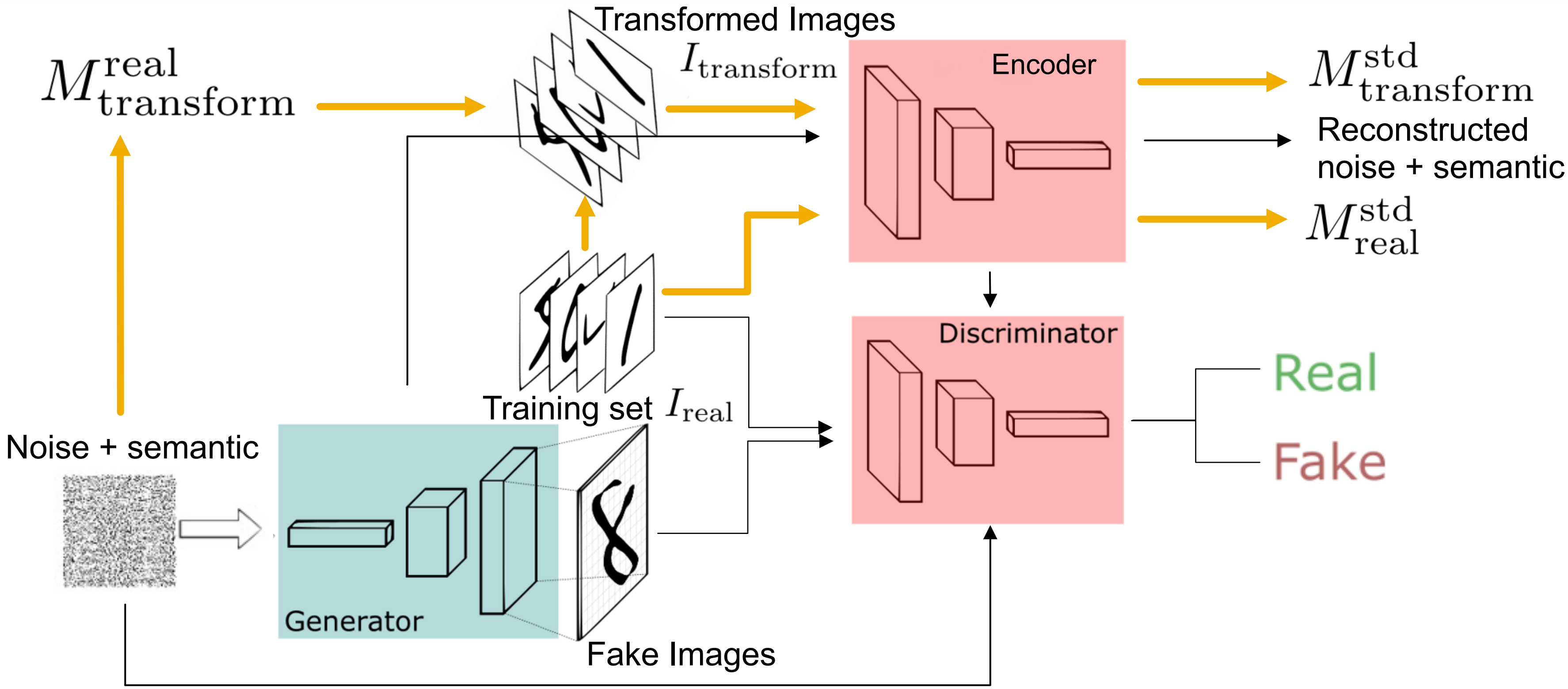
$$M = \begin{bmatrix} 1 & m \\ n & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}$$

$$I_{real} = M^{std}_{real} I_{std},$$
$$I_{transform} = M^{real}_{transform} I_{real},$$
$$I_{transfrom} = M^{std}_{transform} I_{std},$$
$$M^{real}_{transform} = M^{std}_{transform} (M^{std}_{real})^{-1}.$$

By introducing the affine matrix concept into the encoding process, the model "understands" the *data distribution (digits)* and *affine parameters* in an integrated way

Affine Disentangled GAN 2019, https://arxiv.org/abs/1907.05274

Affine Disentangled GAN (ADIS-GAN)



By introducing the affine matrix concept into the encoding process, the model "understands" the *data distribution (digits)* and *affine parameters* in an integrated way

$$\min_{G,E}\max_{D} V(D,G,E) := \mathbb{E}_{x\sim P_{\text{data}}(x)} \log[D(x,E(x))] + \mathbb{E}_{z\sim P_z(z),c\sim P_c(c)} \log[1 - D((c,z),G(c,z))]$$

$$- \lambda L(G,E) - \beta L(M) \qquad M^{real}_{transform} = M^{std}_{transform}(M^{std}_{real})^{-1}$$

Affine Disentangled GAN 2019, https://arxiv.org/abs/1907.05274

Conclusions and Future Work

- Summary
  - Standard deep learning systems **sensitive** to adversarial attacks and transformations.
  - To make deep learning more robust, we need to let those systems understand the **physical world**.
  - We encode physical properties of objects by means of **hidden variables**, and let the model infer what physical transformations have taken place in a given scene.

- Next steps
  - The language of **graphical models** provides a solid foundation to **encode information** and to develop **efficient inference and learning methods** (e.g., variational inference).
  - Powerful pathway towards **more robust** deep learning systems.

Overview

- Development of New ML Methods
  - When does deep learning fail?
  - How can we fix it?
- **What does that mean for AVs?**
  - Development of models of perception errors
  - Simulation of AVs with realistic perception
- Conclusions and Future Work

Overview

- Introduction
- Motivation
- Modeling of perception error: our approach
- Experimental study using simulations
- Conclusion and future work

CETRAN

Centre of Excellence for Testing and Research of Autonomous Vehicles at NTU

Centre of Excellence	Test Centre	Standards Development	Public Engagement

Milestone 2 Testing Framework

Document Review → Virtual Test → Circuit Test → Road Test

Process Management

System Safety

Scenario Testing

Performance evaluation

Scenario Demonstration

Performance measurement

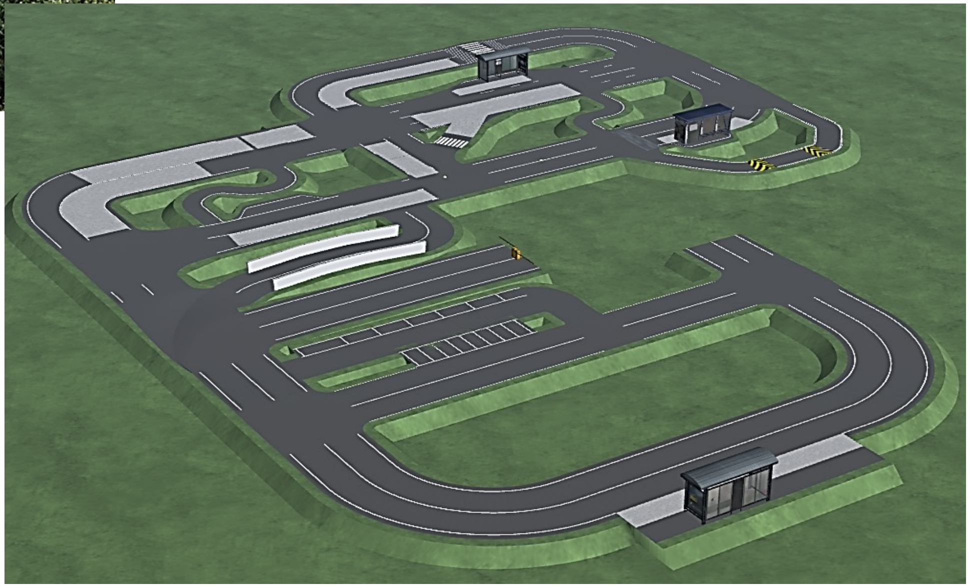
Evaluation in deployment environment

CETRAN

CETRAN Test Track

CETRAN Test Track & Digital Twin

CETRAN AV Test Circuit



CETRAN AV Test Circuit model in IPG CarMaker

CETRAN Test Track & Digital Twin

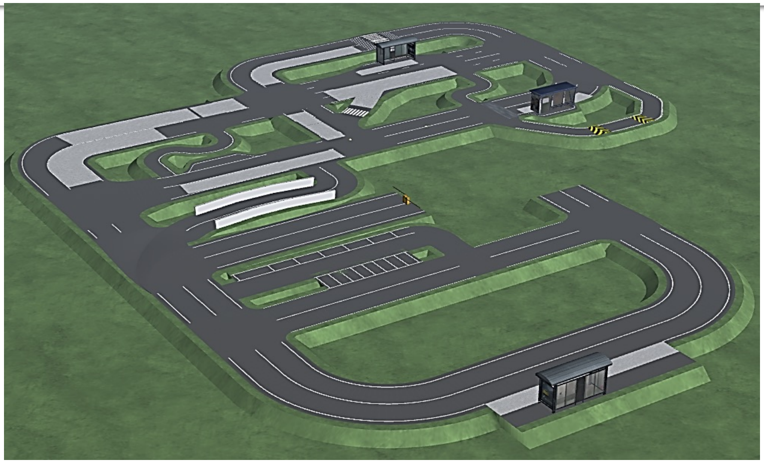
CETRAN AV Test Circuit

CETRAN AV Test Circuit model in VectorZero RoadRunner

Virtual testing

- Challenge:
  - Increasing complexity of systems
  - Increasing complexity of road test setup
  - Unlimited number of different traffic situations

- Solution:
  - Virtual driving test as a complement of proving ground and field tests
  - Identification of highly critical scenarios via simulation



- Benefits:
  - Reproducibility, variations and safety
  - Coverage of exotic test cases
  - Lower cost

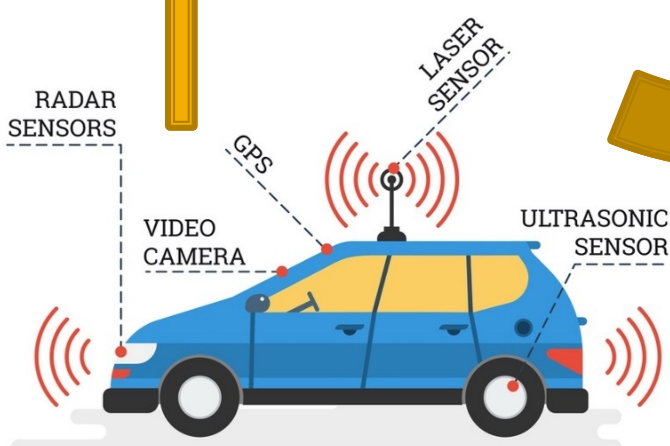
- Crucial for success:
  - Acceptance of virtual driving tests

Virtual testing: Sensors and their imperfections

- high reflection due to unwanted targets such as metallic guard rails on a curve road

- noise affects depth estimations
- incorporate wrong reflections
- contain missing information

- lighting conditions i.e. sunny, cloudy, dark etc. have an adverse impact on them.
- create visibility problems

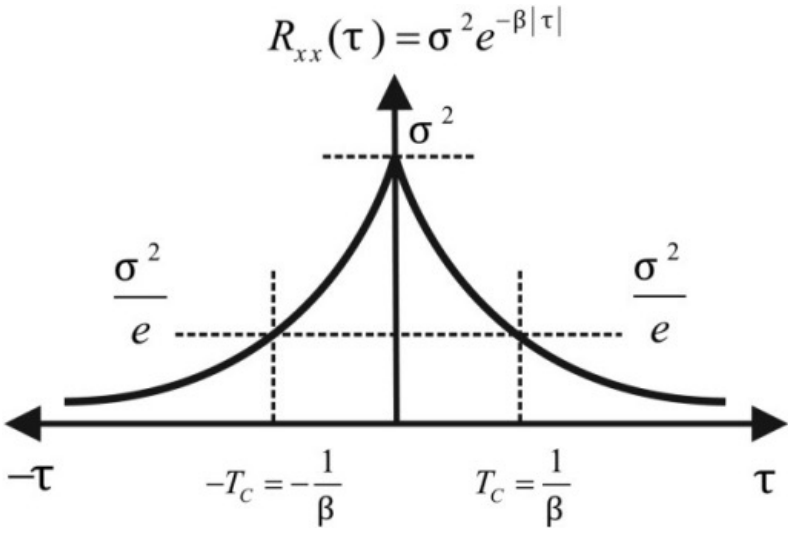


Source: https://healthimpactnews.com/2018/will-driverless-cars-cause-dna-damage-and-cancer/

Existing Sensor Error Model

Some of the current robotics simulators (e.g. GAZEBO, V-REP) incorporate a simplified first order Gauss Markov model as sensor error model.

$$\dot{X} = {}^{-1}/_{T_c} * X + W$$



Where X is zero mean random process, T_c is correlation time, and W is white noise.

Source: https://openi.nlm.nih.gov/detailedresult.php?img=PMC3812568_sensors-13-09549f4&req=4

Shah, Shital, et al. "Airsim: High-fidelity visual and physical simulation for autonomous vehicles." Field and Service Robotics. Springer 2018.
Collins, Thomas, Nadeesha Oliver Ranasinghe, and Wei-Min Shen. "ReMod3D: A high-performance simulator for autonomous, selfreconfigurable robots." Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013

Overview

- Introduction
- Motivation
- Modeling of perception error: our approach
- Experimental study using simulations
- Conclusion and future work

Motivation

A typical AV simulation

- assume *perfect* sensors
  or even *perfect* perception
- No physical models

+

Perception Error Model (PEM)

- describing the actual performance of the sensing and perception (S&P)
- in the *decision-making* context

A representative AV simulation

- represents the uncertainty of the real world,
  under different sensory conditions and edge cases
- meaningful virtual alternative to physical tests

Overview

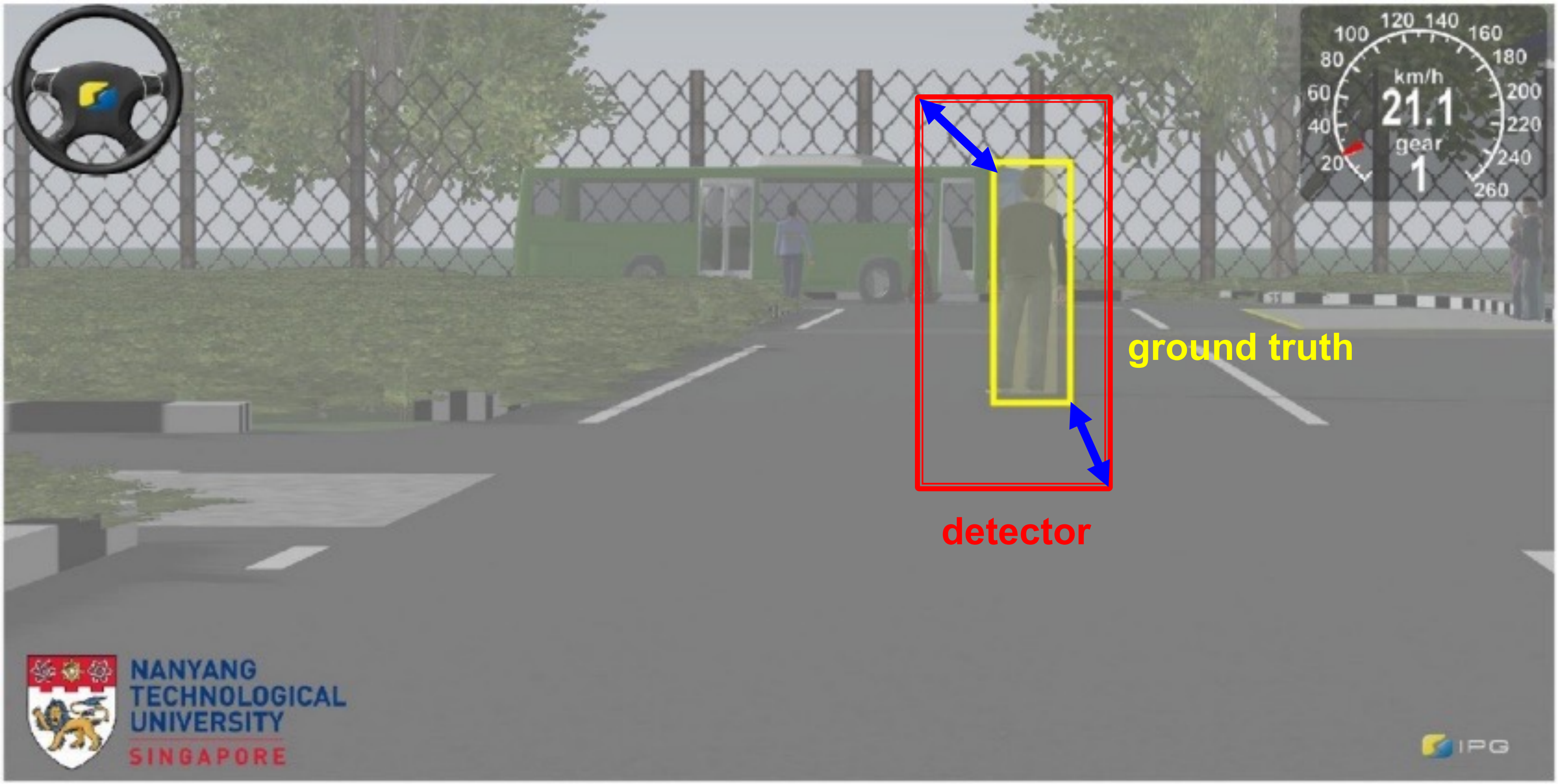
- Introduction
- Motivation
- **Modeling of perception errors: our approach**
- Experimental study using simulations
- Conclusion and future work

Modeling of Perception Errors

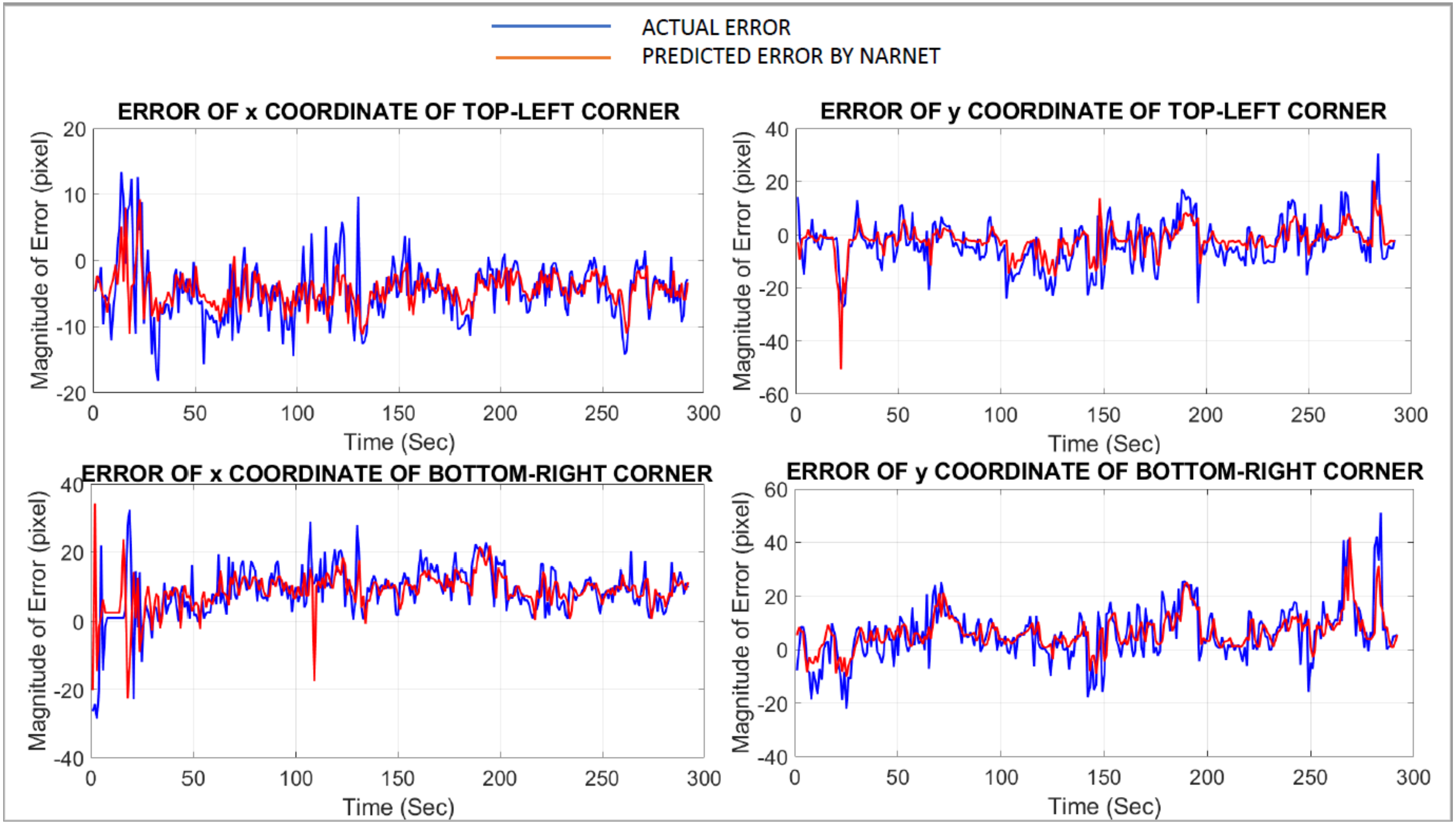
- Phased approach towards modeling perception errors
- **Phase A:**
  A camera-based PEM to represent specific error in perceived object info
  (position and 2D dimensions)
- **Phase B:**
  A comprehensive *sensor-agnostic* PEM to represent more detailed errors in perceived object info
  (label, position, 3D dimensions, orientation, visibility, latency)

PEM $\Rightarrow$ Perception Error Model

Phase A: Illustration of 2D bounding boxes

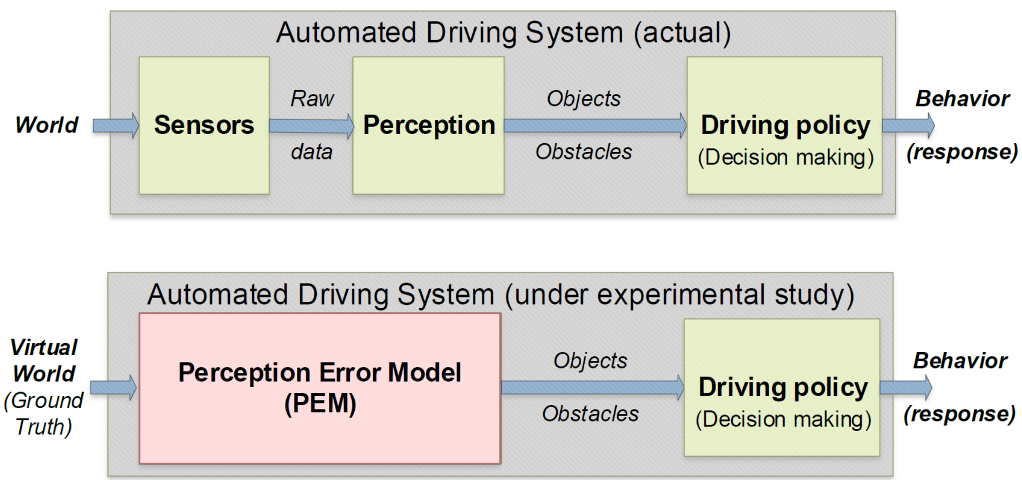


Phase A: Modeling of 2D Bounding Box Offset: Nonlinear Auto-Regressive Model



Phase B:
A comprehensive sensor-agnostic PEM

- We model perception errors using a *sensor-agnostic* model that can represent different sensors typically used in an AV sensing and perception (S&P) subsystem
- Helps to study impact of perception error on decision making (aka AV behavior)



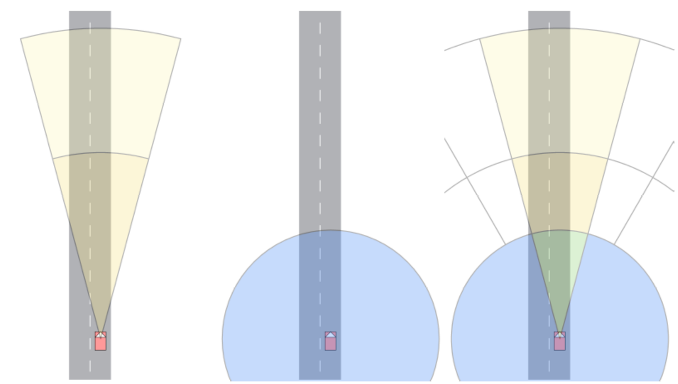


Figure 3: Example of a zone-based partitioning of the S&P errors. (a) shows a simple camera FoV, divided into 2 zones based on range. (b) represents a LiDAR. In (c), we can see that if the sensor setup include multiple sensors, the overlap can be easily identified and thus creating an additional zone where objects can be detected by both sensors and hence, the error will depend on how the signals are fused.

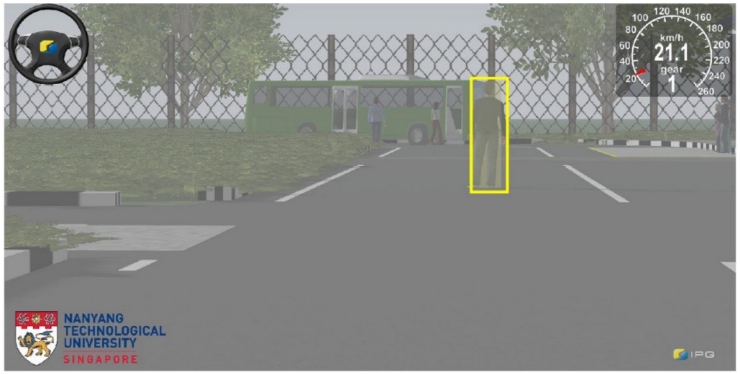
Overview

- Introduction
- Motivation
- Modeling of perception errors: our approach
- Experimental study using simulations
- Conclusion and future work

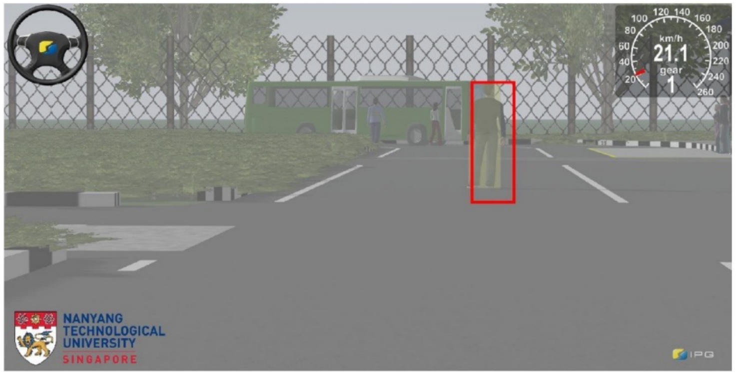
Experimental Study using simulations

- For each phase, we perform experimental study using different set of tools
- Phase A:
  Co-simulation with IPG CarMaker and Autoware
  - Objective: model errors in perceived object position and 2D bounding box using only a camera sensor
- Phase B:
  Co-simulation with LGSVL simulator and Baidu Apollo
  - Objective: a more comprehensive *sensor-agnostic* error model that can help in analyzing the impact of perception errors on decision making (AV behavior)

Phase A: Simulation with PEM
NAR model for 2D Bounding Boxes

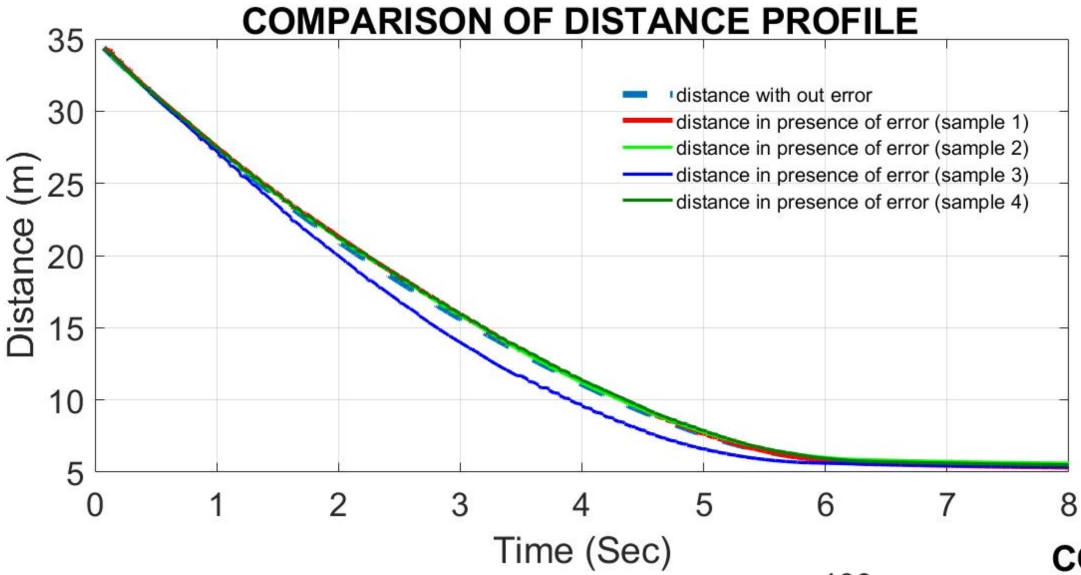


Bounding Box without error



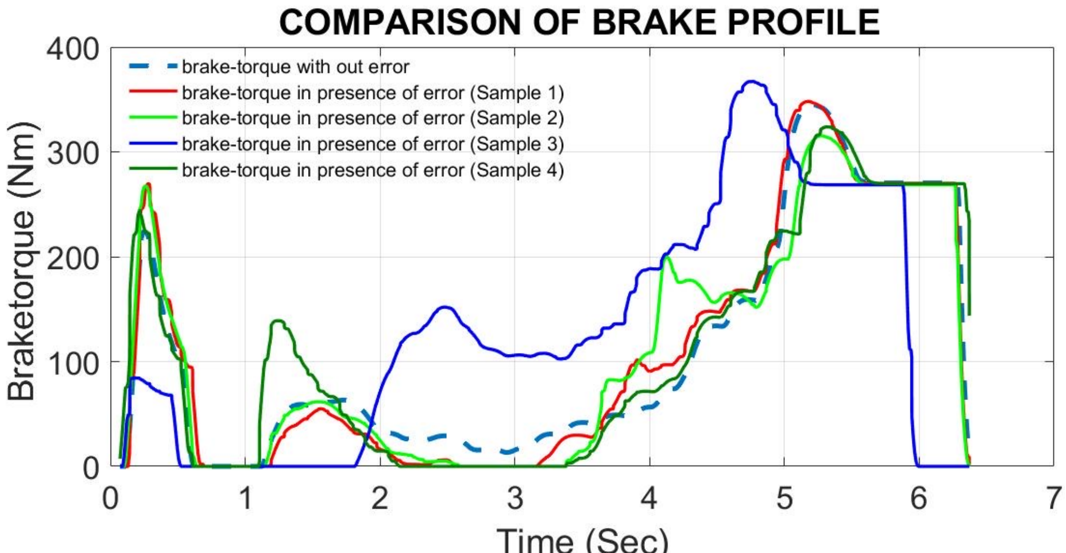
Bounding Box with error

Phase A: Study the impact on AV behavior (with NAR model for 2D Bounding boxes)



Distance profile of vehicle in presence of errors (PEM)

Brake profile of vehicle in presence of errors (PEM)



Phase A: Study the impact on AV behavior (with NAR model for 2D Bounding boxes)

- Due to the **random errors in the bounding box's** position of the detected object, the **perceived distance** of the object from the vehicle is **noisy**, and subsequently, the **brake-torque** of the vehicle becomes non-optimal.

- This may lead to **erroneous behaviour of the vehicle**, with potential dramatic consequences.

- In a **more complex traffic environment**, involving multiple pedestrians and vehicles, we can expect a significant effect of the error in distance profile.

Phase B: co-simulation platform

- For Phase B experimentation, we use a virtual vehicle driven by Apollo (ADS) in a virtual environment modeled in LGSVL simulator
- We replace the Perception module of Apollo with a virtual sensor (implementing PEM), that communicates over CyberRT
  - This virtual sensor uses ground truth, injects error as per the applicable PEM, and publishes the erroneous object list over CyberRT, to be read by the behavior subsystem
- We implement a (python-based) automated test framework to execute different test cases (scenarios with parameter variants) under different PEMs

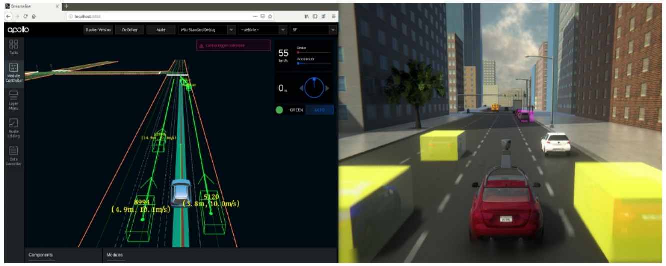
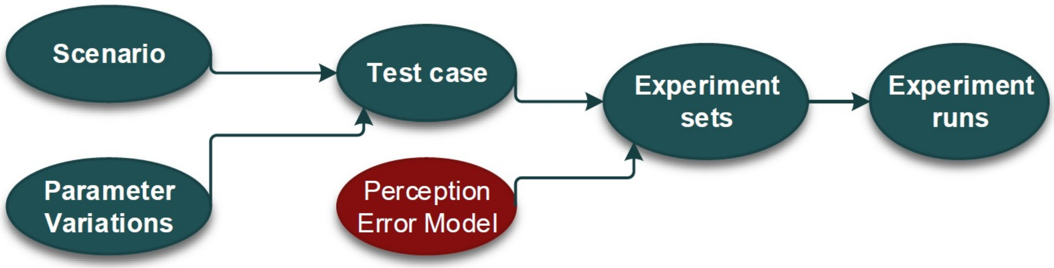
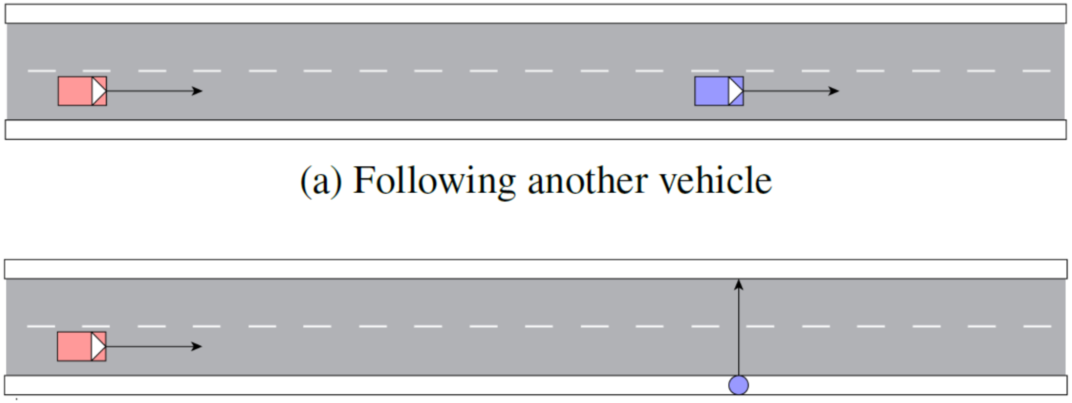


Figure 4: Screenshot of the co-simulation in a generic urban driving situation. Bounding boxes (yellow/purple) of PEM-based objects OM rendered by LGSVL (right) are consistent with what Apollo sees (left), even undetected objects. Sample videos from experiments are available for reference under supplementary material.

Phase B: Scenario-based tests and experimental study





(a) Following another vehicle

(b) Pedestrian on an urban road

Figure 6: Illustration of 2 scenarios used in our experiments.

Phase B: co-simulation platform

Phase B: Perception Error and AV behavior under different PEMs

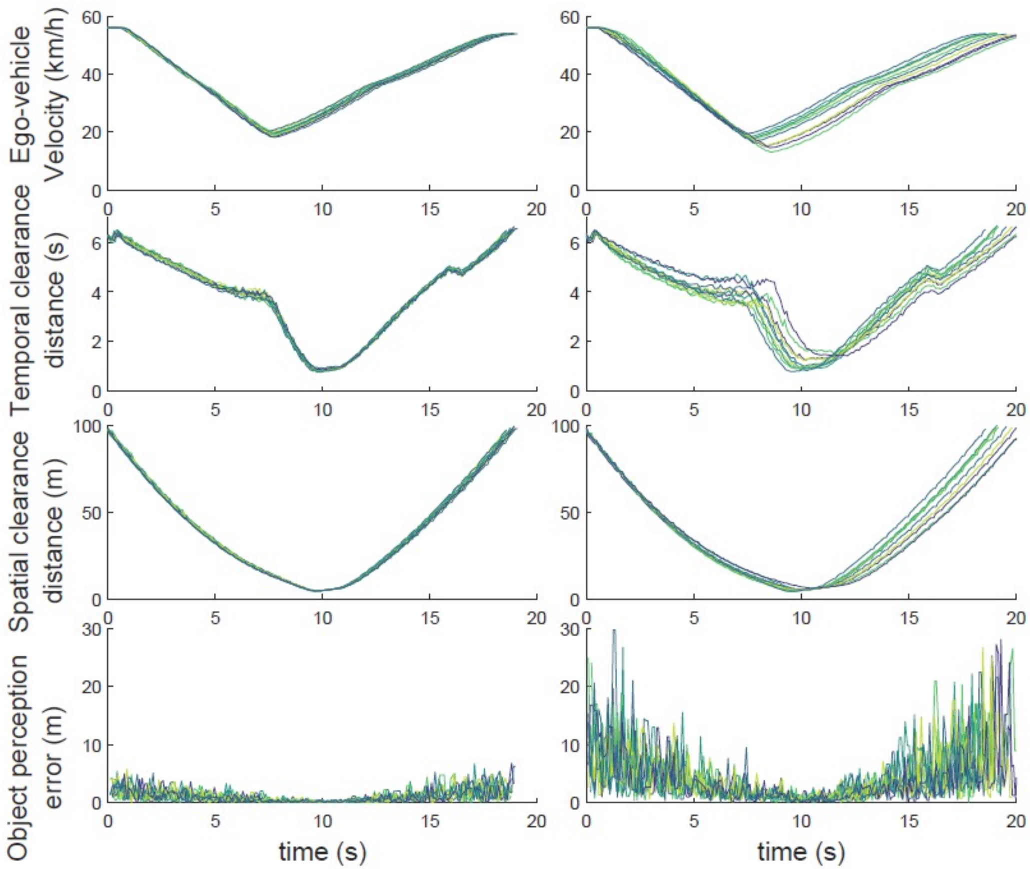
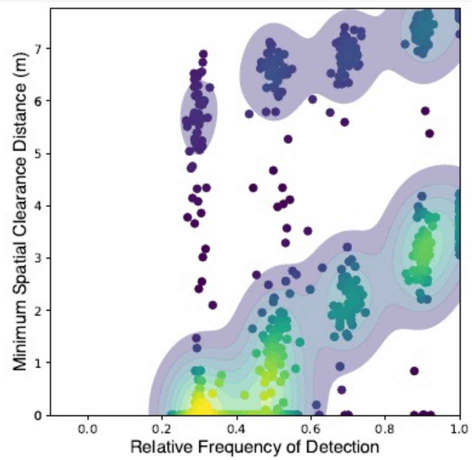


Figure 7: Illustration of ego-vehicle behavior and perception error for pedestrian jaywalking test case with two error models: DSAS (left) and DLAL (right).

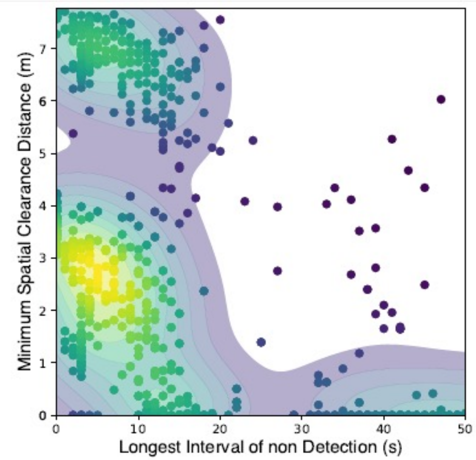
Phase B: Relationship between safety evaluation metrics under different PEMs

Min Spatial Clearance vs. Relative Freq of Detection



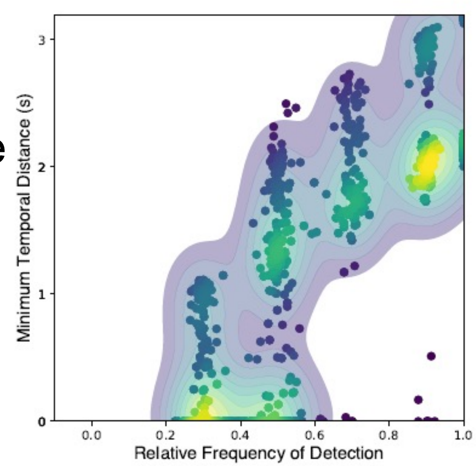
(a) Following another vehicle

Min Spatial Clearance vs. **Longest Interval of non-Detection**



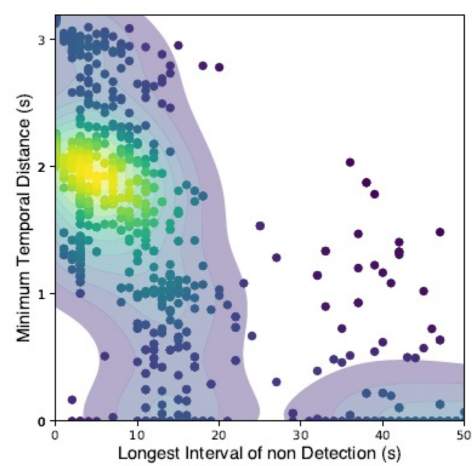
(b) Following another vehicle

Min Temporal Distance vs. Relative Freq of Detection



(d) Following another vehicle

Min Temporal Distance vs. **Longest Interval of non-Detection**



(e) Following another vehicle

Overview

- Introduction
- Motivation
- Modeling of perception error: our approach
- Experimental study using simulations
- Conclusion and future work

Conclusion

- We describe a phased approach to test and study AV sensing & perception (S&P) errors in a virtual environment, under simulated urban scenarios
  - Special focus on how S&P errors can impact decision making (AV behavior)
  - Two different co-simulation environments (Carmaker-Autoware, LGSVL-Apollo)
- We study different PEMs: both hand-crafted and learnt from real-life data
- We highlight weaknesses of current metrics used to evaluate perception
- We currently focus on the S&P function of detecting other road users
- **Future goals:**
  - Extend PEM to model errors in other S&P functions (e.g. detection of lane markings, road signs, free space)
  - Compare simulation results vs. field measurements (e.g. from test circuit)

Thanks!

- Research Team

- Funding support

NATIONAL RESEARCH FOUNDATION PRIME MINISTER'S (SINGAPORE

SMART Singapore-MIT Alliance for Research and Technology

Ministry of Education SINGAPORE

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

BMW

TÜV SÜD

ST Engineering

SCHAEFFLER

Towards Robust Perception for Autonomous Vehicles

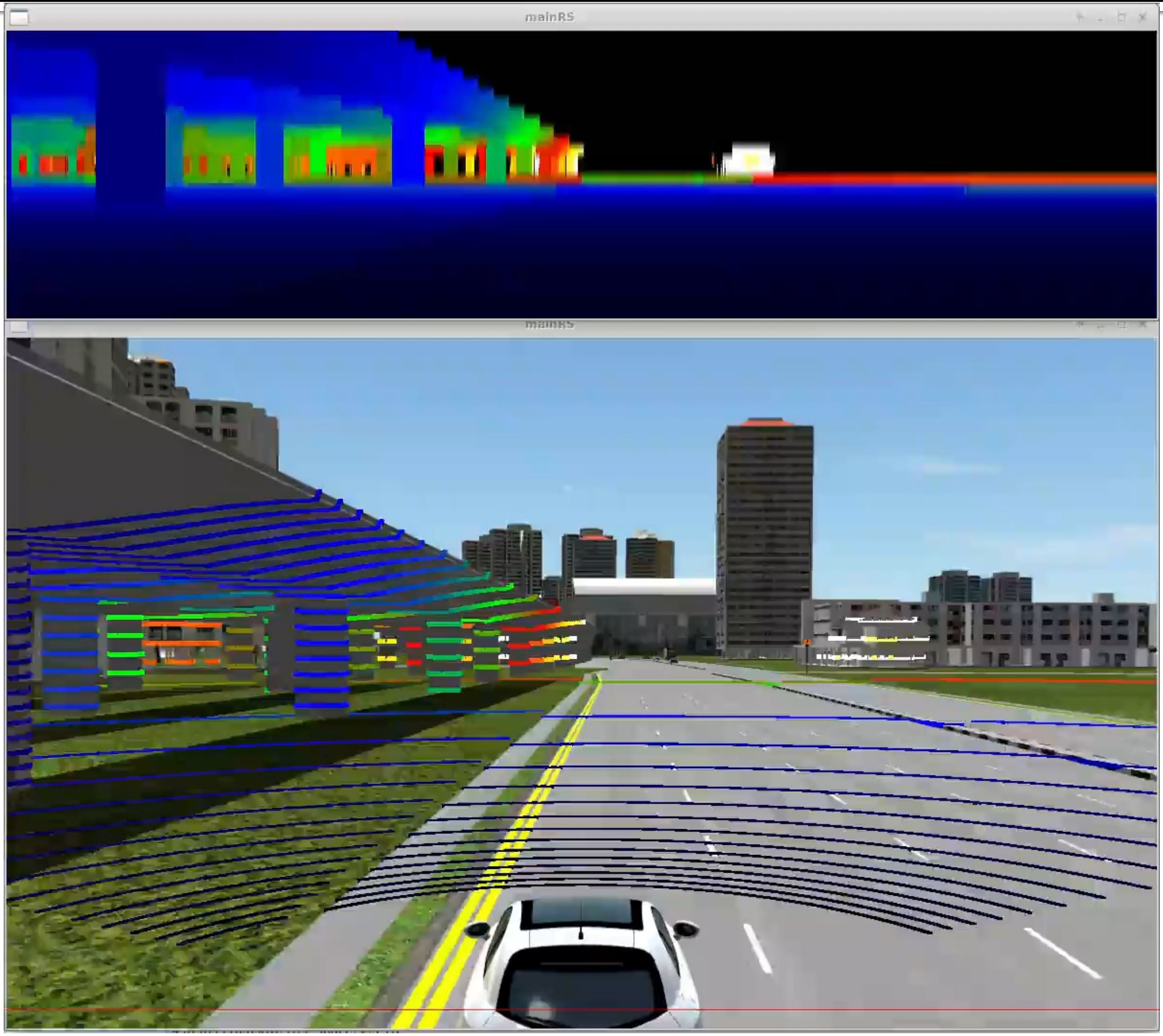
Dauwels Lab
@ DauwelsL
www.dauwels.com

1 Oct 2019

Justin Dauwels

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Ongoing Work:
Simulation in Singapore context



Modeling of Bounding Box Offset

- To choose the best model from a set of possible models, we can take the help of the most common model selection method is the **Bayesian Information Criteria (BIC).**

- The model with smallest BIC is favored when selecting models.
  - $BIC = \ -2\ln(\hat{L}) + K\ln(n)$, for **linear ARMA**

  where $\hat{L}$ = likelihood function of the model, K = number of estimated parameters in the model and n = sample size (observation)
  - $BIC = \ n\ln(SSE/n) + p\ln(n)$, for **NAR**

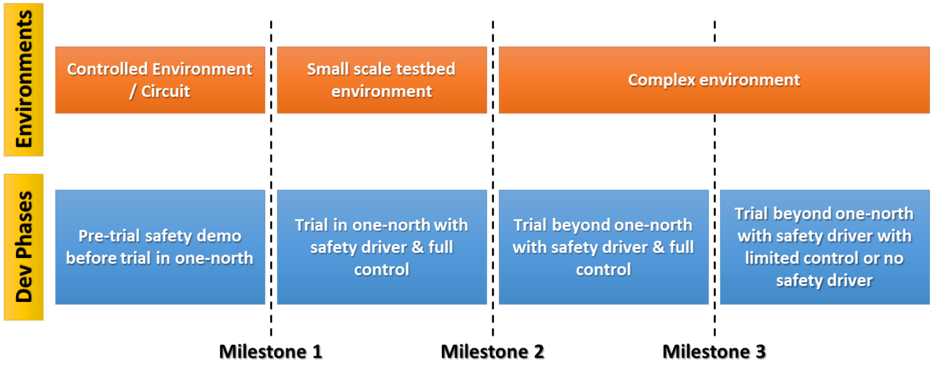
  where SSE = sum squared error, n = number of training samples and p = number of parameters (weights and biases).

Model selection by BIC

BIC VALUES COMPARISON

	Coordinates	linear ARMA	NARNET
Dataset 1	x_{tl}	1752.1	811.1
	y_{tl}	2033.8	1148.7
	x_{br}	1919.1	1006.1
	y_{br}	2106.6	1178.1
Dataset 2	x_{tl}	3134.3	1904.6
	y_{tl}	3028.5	1841.5
	x_{br}	3569.2	2164.3
	y_{br}	3082.8	1973.9

Milestones for AV Trials

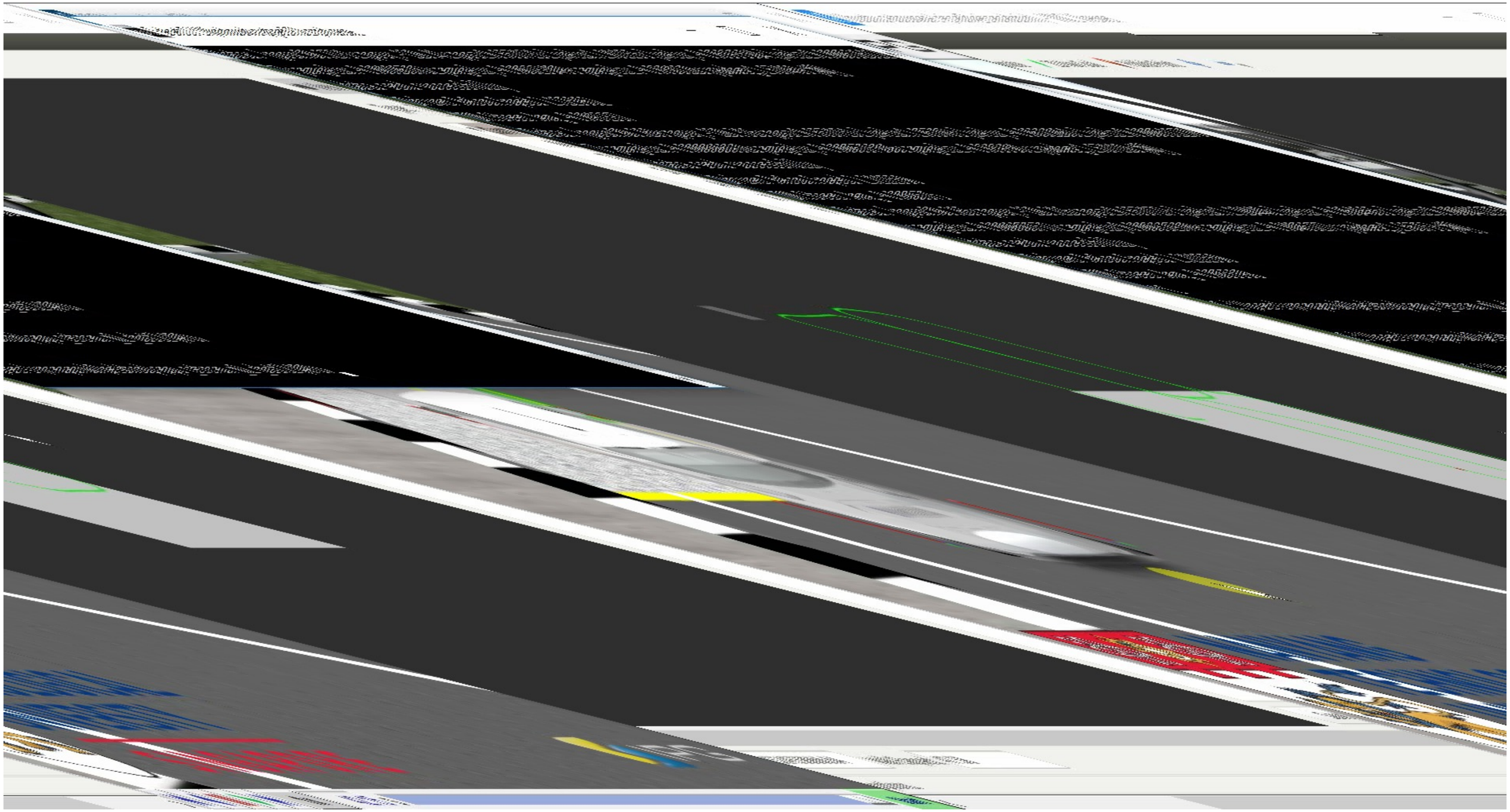


- **Milestone 1 (Established)**
  - Ability to safely conduct testing of autonomous vehicles with safety driver in a small scale testbed.
- **Milestone 2 (Rollout Phase)**
  - Ability to safely conduct testing of autonomous vehicles with safety driver in a complex environment with minimal interaction by the safety driver. Requirements to be supported by Technical Reference.
- **Milestone 3 (Proposed)**
  - Ability to safely conduct testing of autonomous vehicles without or with a remote safety driver in a complex environment. This implies high technical maturity. Requirements to be supported by a Singapore Standard.

Dataset

- To develop the error model, ***KITTI object tracking evaluation 2012*** has been utilized.
  - Videos in the dataset are taken in different conditions, e.g., sunny road with shadows, urban road with traffic, highways etc.
  - The 2D bounding boxes for each images have been labeled manually for ‘car’& ‘pedestrian’ classes.
  - The resolution of the images is **1238 x 374 pixels** and the frame rate is **10 fps**.

Sample Simulations (No Error)



Motivation

Perception error model

+

Many AV simulators assume **perfect sensors**
(no physical models)

Models uncertainty under various environmental conditions, i.e., bright sun, snow, rain etc.

A high-fidelity AV simulation must incorporate **perception errors** that capture the imperfections in the perception system.

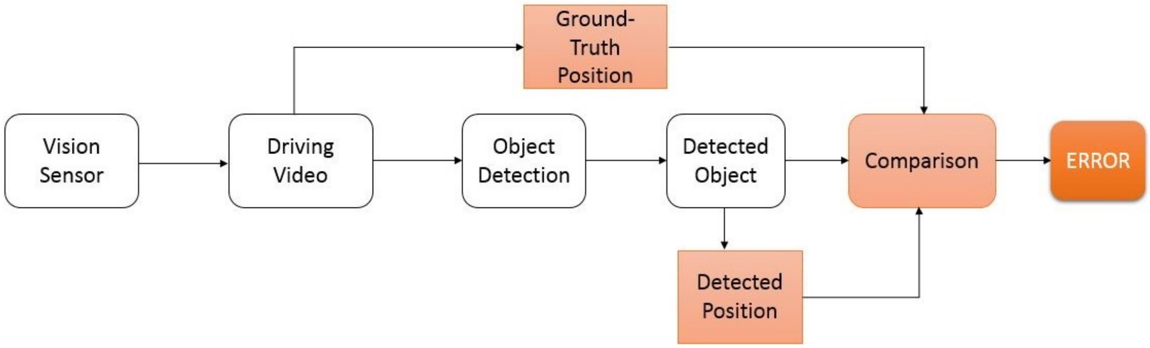
Phase A:
A camera-based PEM

- We model perception errors associated with the **camera sensor**
- *Faster RCNN** has been chosen as state-of-the-art object detector
- We model the error in the bounding boxes (top-left and bottom-right corner)

$$E_{tl} = [x_{tl}, y_{tl}]_d - [x_{tl}, y_{tl}]_{gt},$$

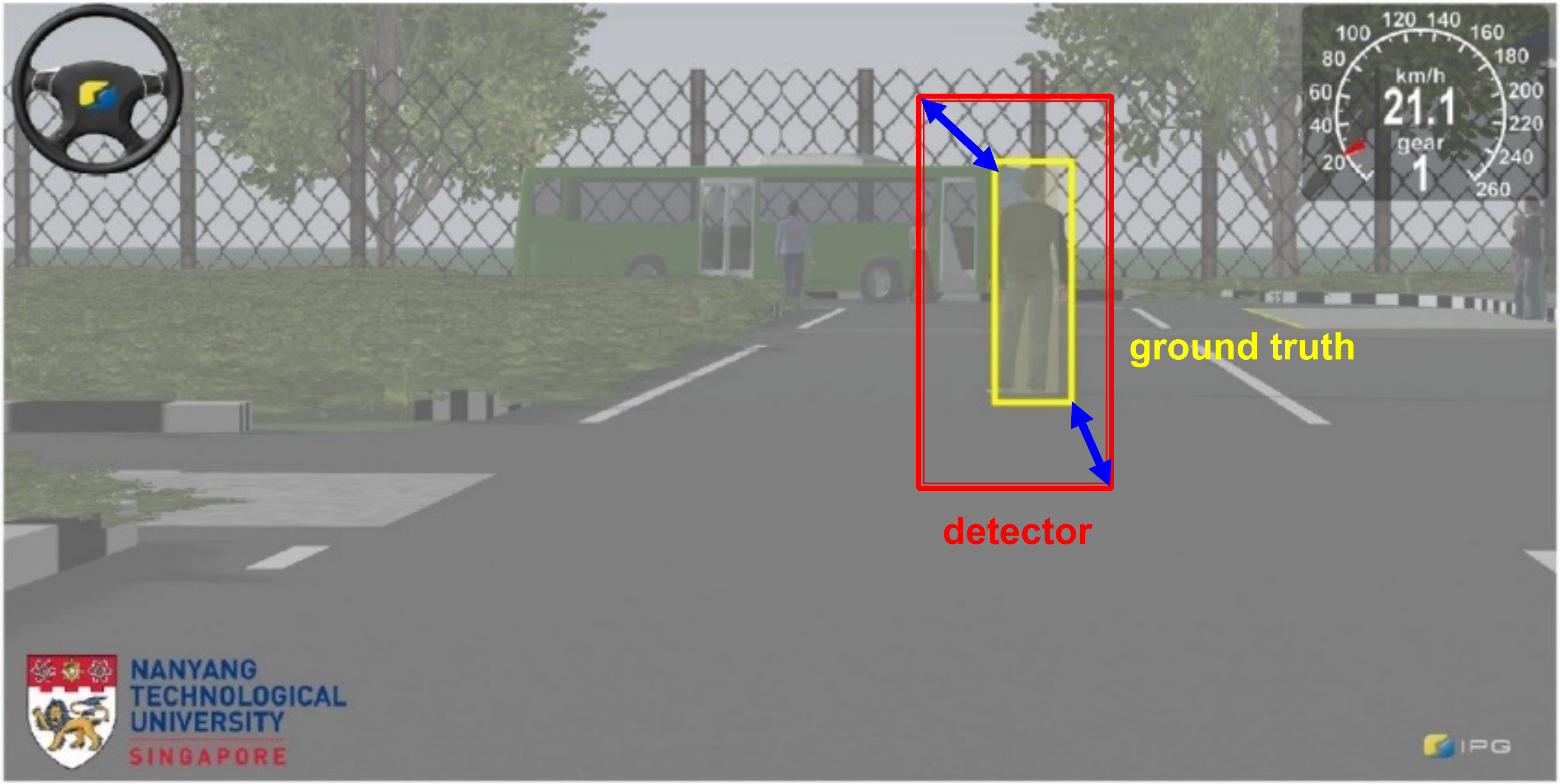
$$E_{br} = [x_{br}, y_{br}]_d - [x_{br}, y_{br}]_{gt},$$

where E_{tl} and E_{br} are errors in pixel coordinates of topleft and bottom-right corner respectively. *d* and *gt* stand for detected and ground-truth bounding boxes resp.

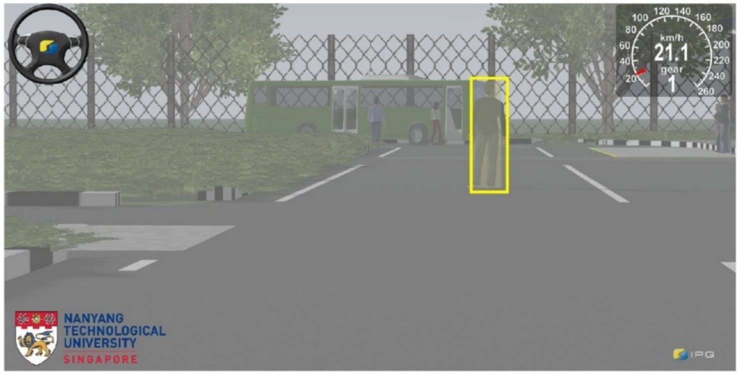


*Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in Neural Information Processing Systems (NIPS). 2015

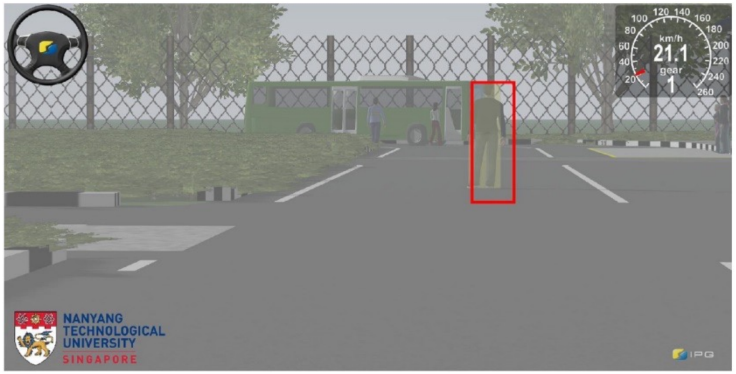
Phase A: Illustration of 2D bounding boxes



Phase A: NAR model for 2D bounding boxes



Bounding Box without error



Bounding Box with error

Phase A: Modeling 2D bounding box offset

- **Linear ARMA(p,q) model**

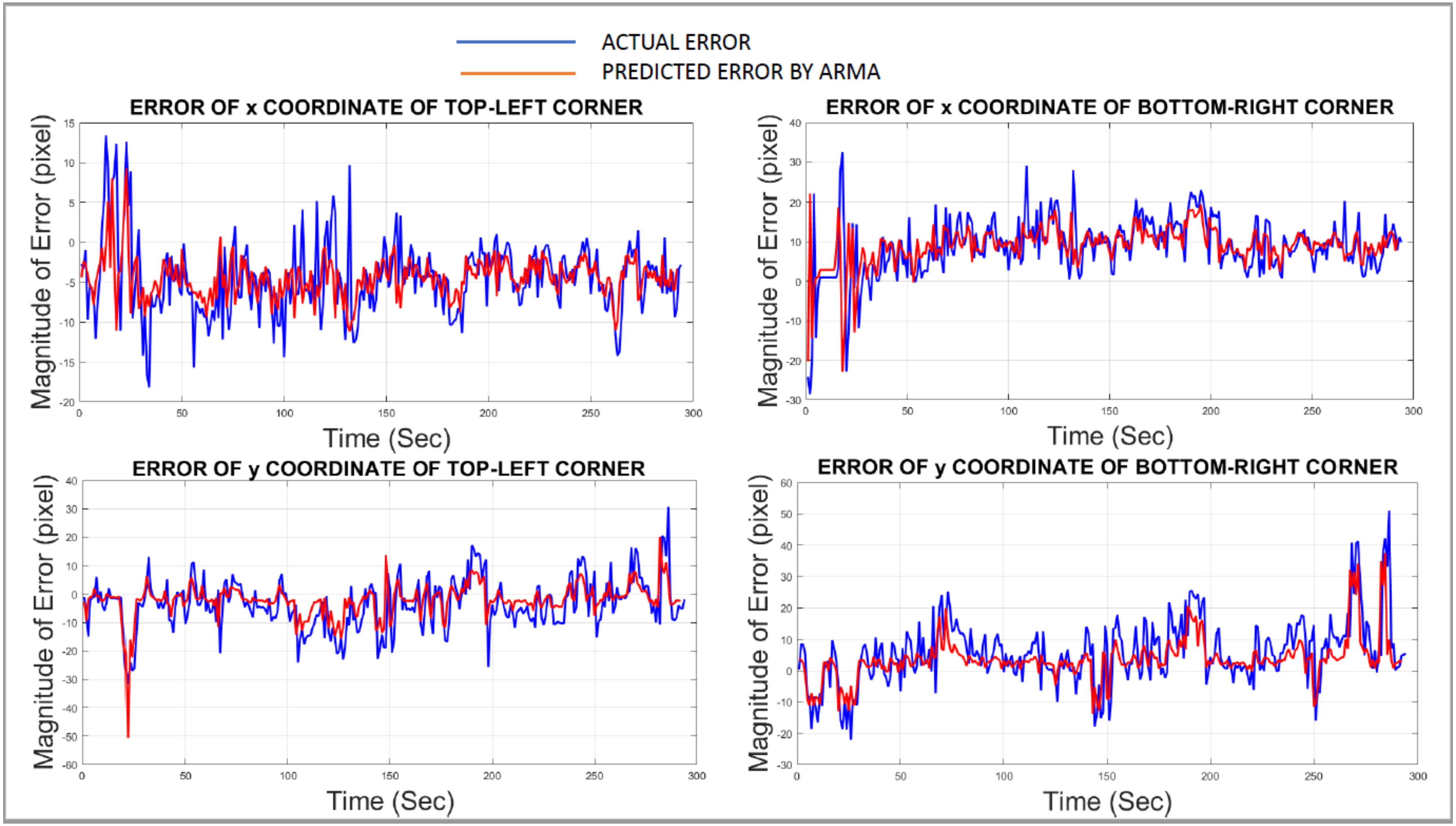
$$X_t = C + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}$$

- **Nonlinear Auto-Regressive (NAR) Model**

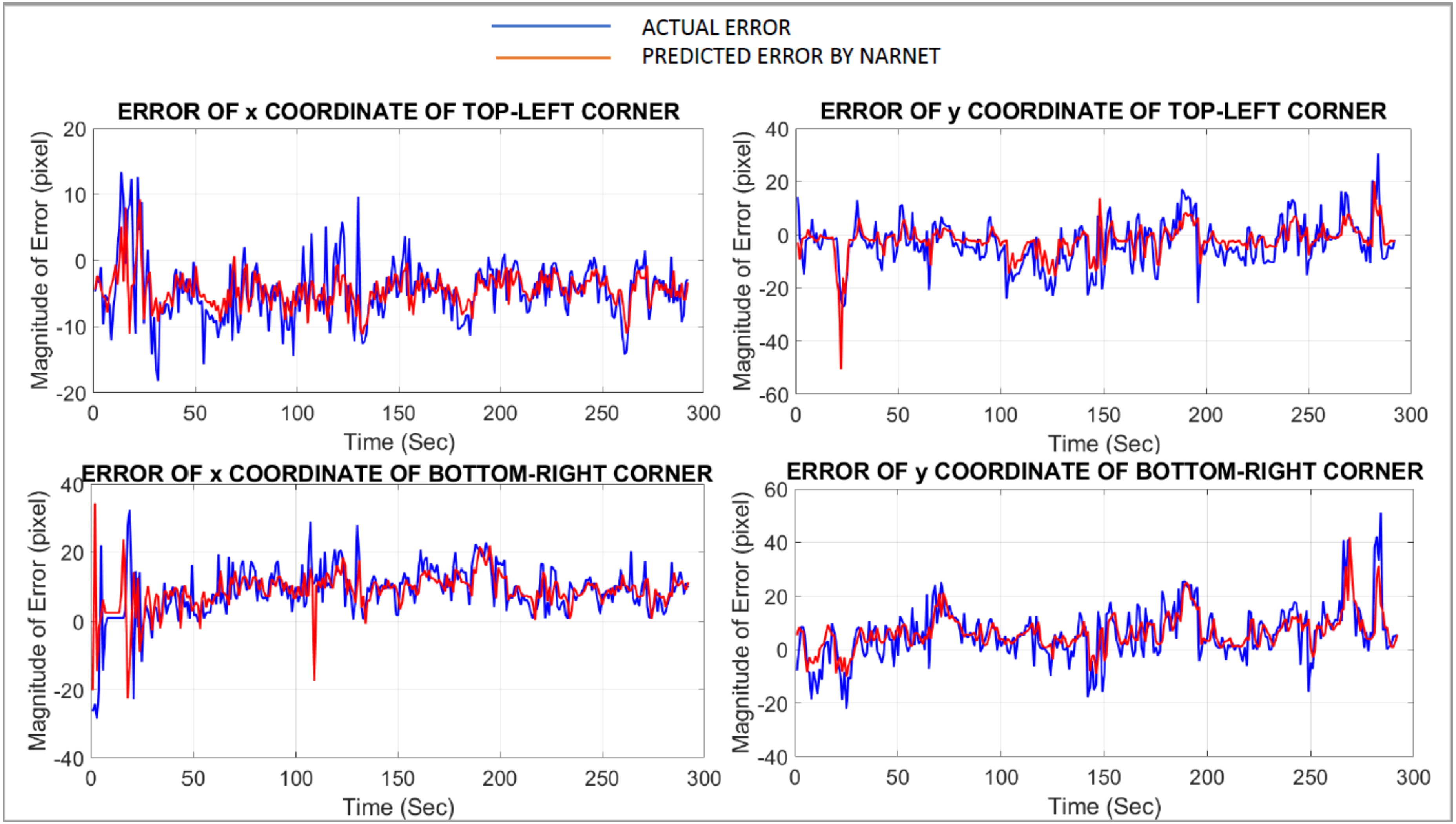
$$y_t = G(y_{t-1}, y_{t-2}, y_{t-3}, \ldots) + \varepsilon_t$$

  - ***G*** **= Artificial neural network**
  - **1 hidden layer with 10, 15, 20 and 25 neurons.**

Phase A: Modeling of 2D Bounding Box Offset: Linear ARMA

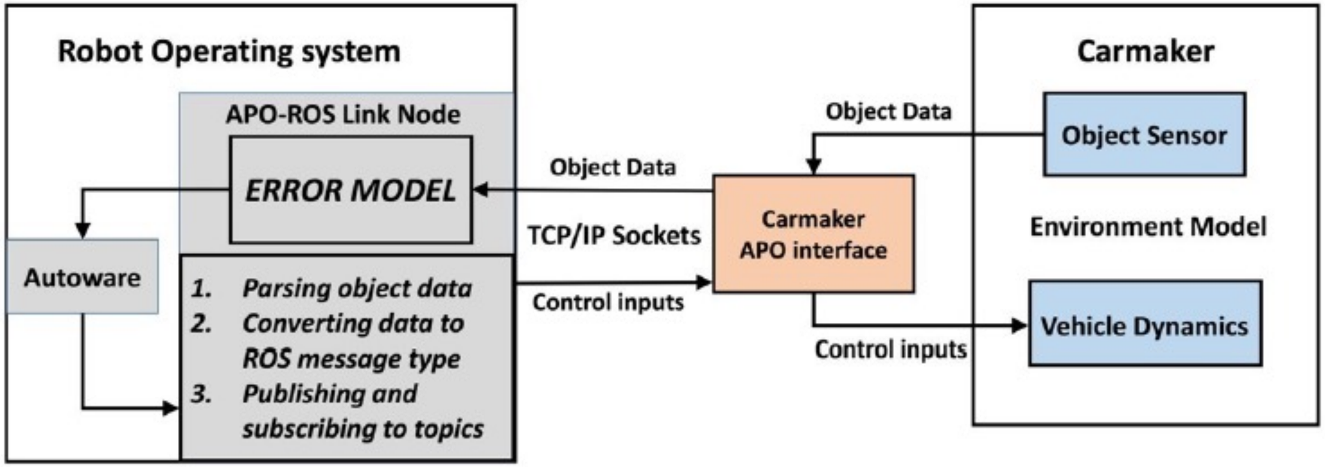


Phase A: Modeling of 2D Bounding Box Offset: Nonlinear Auto-Regressive Model



Phase A: co-simulation platform

- *IPG Carmaker* is used for modelling the vehicle and the environment
- Ego vehicle data and sensor object list extracted using APO (using TCP/IP-based communication stack from IPG)
- Data from Carmaker is passed on to a third party ADS (*Autoware*)
- ADS generates control signals (intents) that are sent to CarMaker, which performs virtual actuation considering vehicle dynamics



Phase A: Simulation Example (No error)