

Robustness of neural networks and adversarial attacks

BELIEF 2022

7th International Conference on Belief Functions, October 27, 2022

Stéphane Canu, INSA Rouen, LITIS

Outline

1. Introduction

2. AI in cars

3. Adversarial attacks and robustness

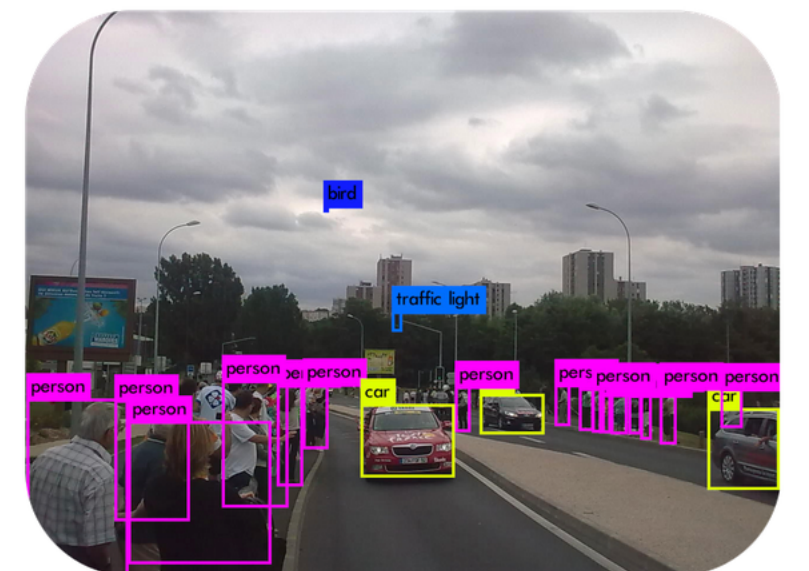
4. Exact attacks



PROJECT

Recent progresses in machine learning in general and especially in deep learning make it possible to include this technology in more and more autonomous vehicles. However, before this possible future becomes reality and our roads are made safer with algorithms replacing human drivers, it is necessary to know how to prove the quality of the decisions made.

This project aims at strengthening local research dynamics about safety issues associated with the use of artificial intelligence in mobility. To achieve this goal, the project will endeavor to formalize the problem, to propose algorithms to solve it and to demonstrate its feasibility



<https://chaire-raimo.github.io/>

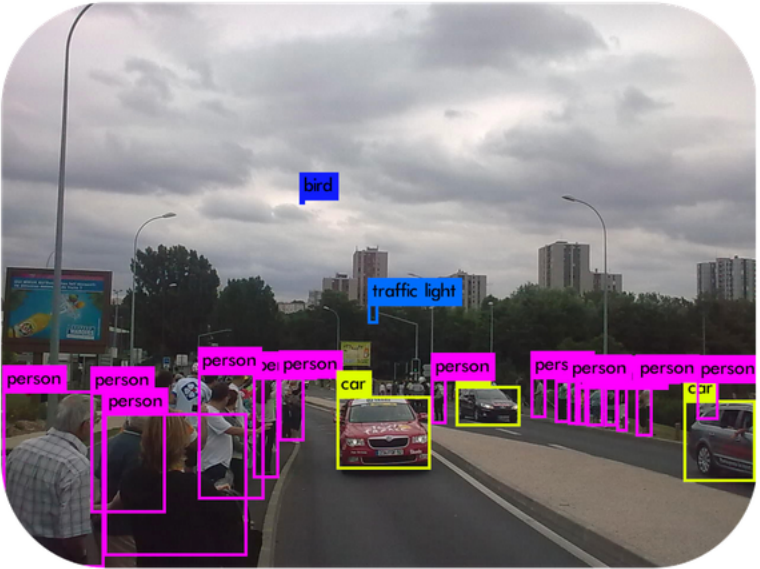
Raimo's team



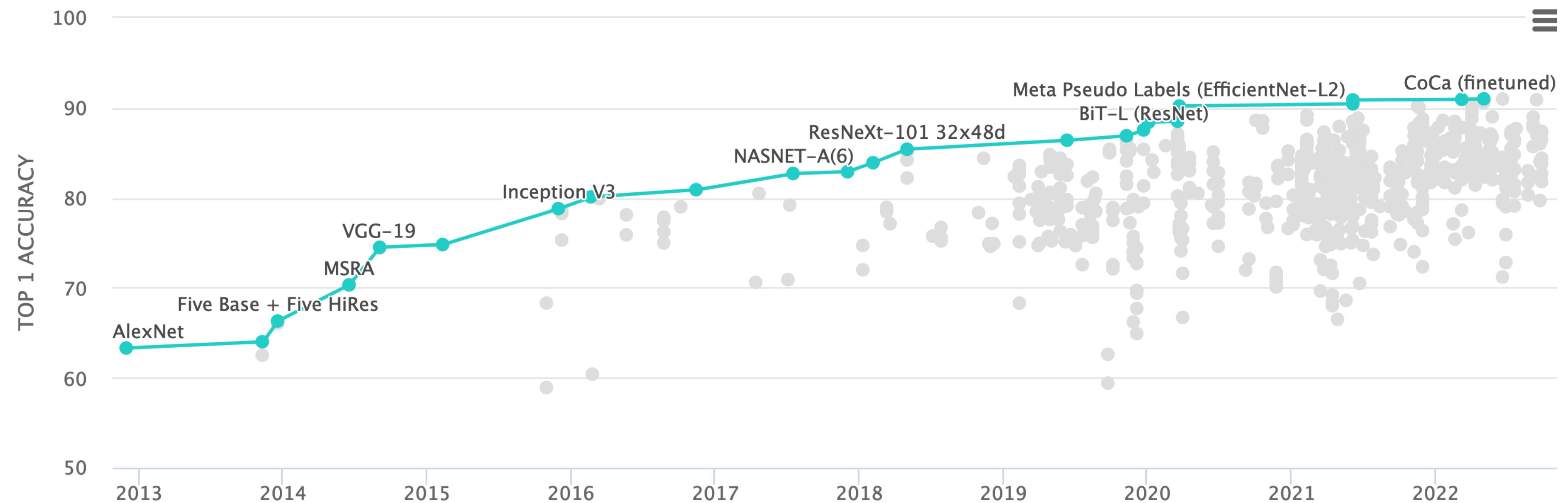
PROJECT

Recent progresses in machine learning in general and especially in deep learning make it possible to include this technology in more and more autonomous vehicles. However, before this possible future becomes reality and our roads are made safer with algorithms replacing human drivers, it is necessary to know how to prove the quality of the decisions made.

This project aims at strengthening local research dynamics about safety issues associated with the use of artificial intelligence in mobility. To achieve this goal, the project will endeavor to formalize the problem, to propose algorithms to solve it and to demonstrate its feasibility



ImageNet progress time line



2011: 50% - 2012 : 66 %

2022: 91 %

Outline

1. Introduction

2. AI in cars

3. Adversarial attacks and robustness

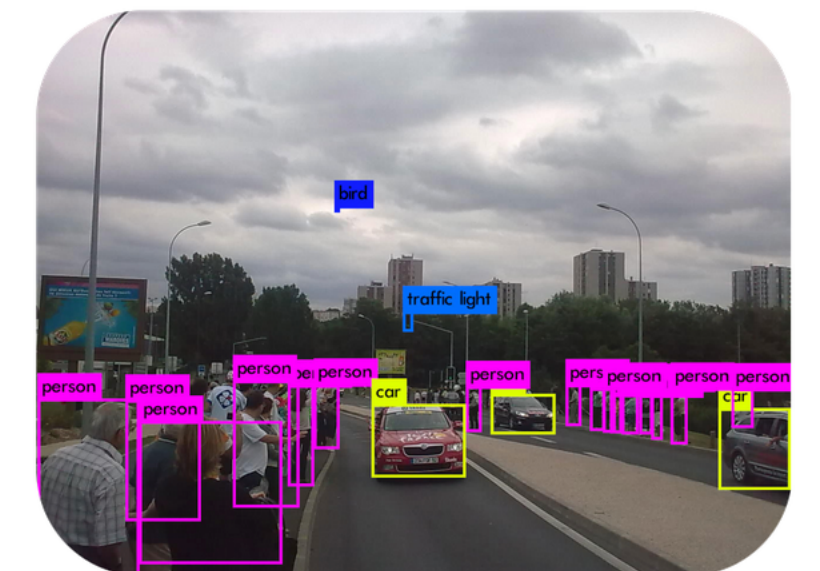
4. Exact attacks



PROJECT

Recent progresses in machine learning in general and especially in deep learning make it possible to include this technology in more and more autonomous vehicles. However, before this possible future becomes reality and our roads are made safer with algorithms replacing human drivers, it is necessary to know how to prove the quality of the decisions made.

This project aims at strengthening local research dynamics about safety issues associated with the use of artificial intelligence in mobility. To achieve this goal, the project will endeavor to formalize the problem, to propose algorithms to solve it and to demonstrate its feasibility



<https://chaire-raimo.github.io/>

Autonomous vehicles: when?

According to Elon Musk: planned for 2014, 2015, 2016, ..., 2020, 2021, 2022, autonomous cars are now expected in 2023



One day Elon Musk will be right...

Elon Musk removing his hands from the wheel with Autopilot engaged during an interview (Bloomberg, 2014).
numerama.com/vroom/972975-tesla-prevues-en-2014-2015-2016-2018-2019-2020-2021-et-2022-les-voitures-autonomes-sont-maintenant-promises-pour-2023.html

ADAS AI components

(e.g. OpenPilot)

Diving agent

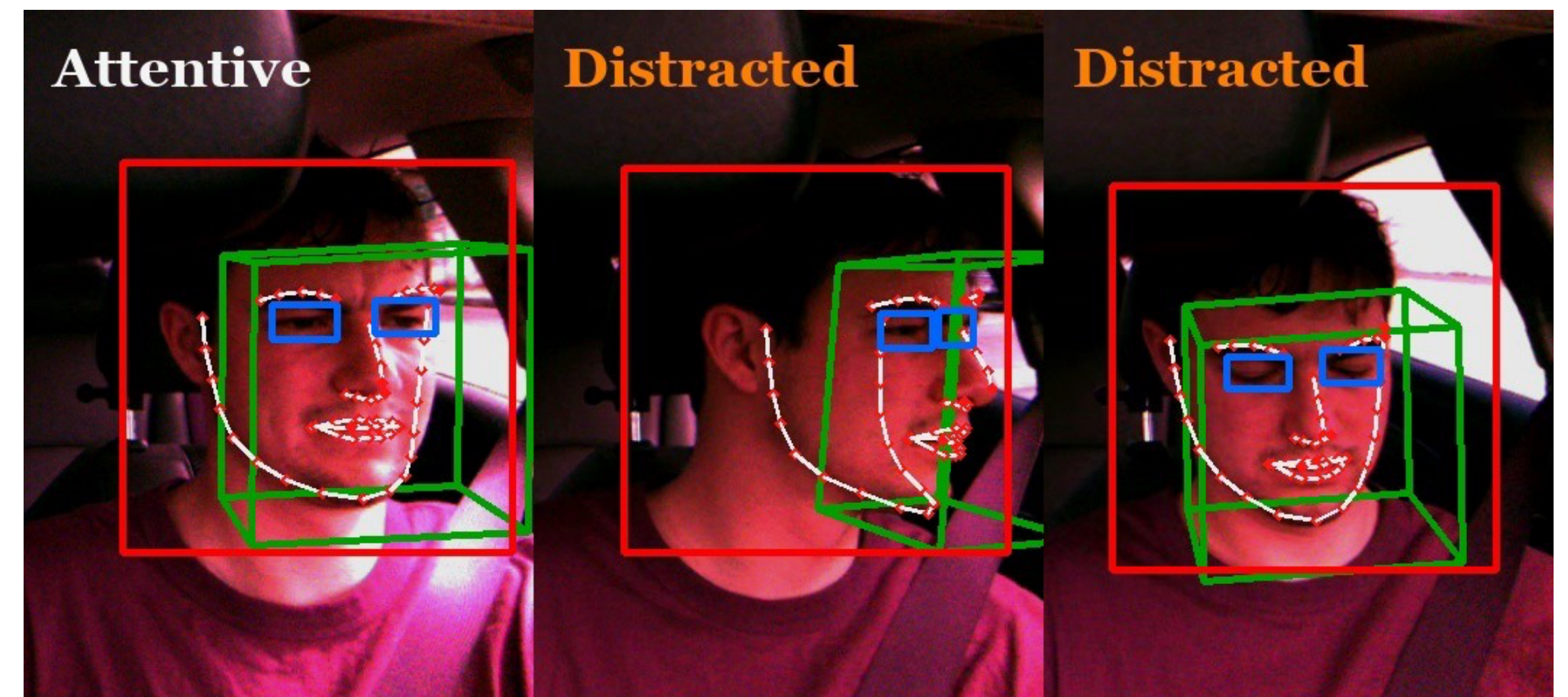
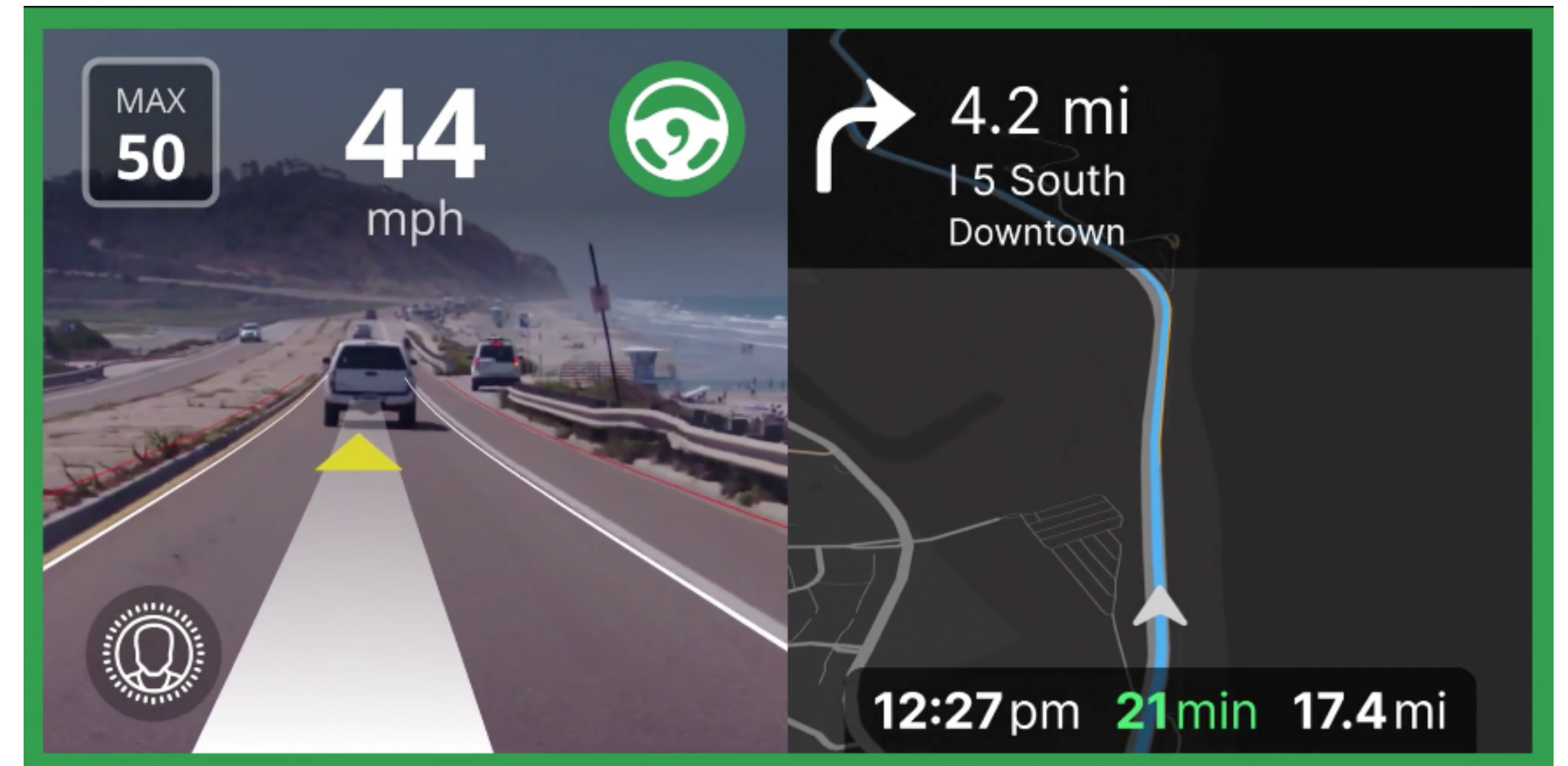
- Automated lane-centering
- Adaptive cruise control
- Assisted lane change
-

Driver monitoring system

- Safety concerns

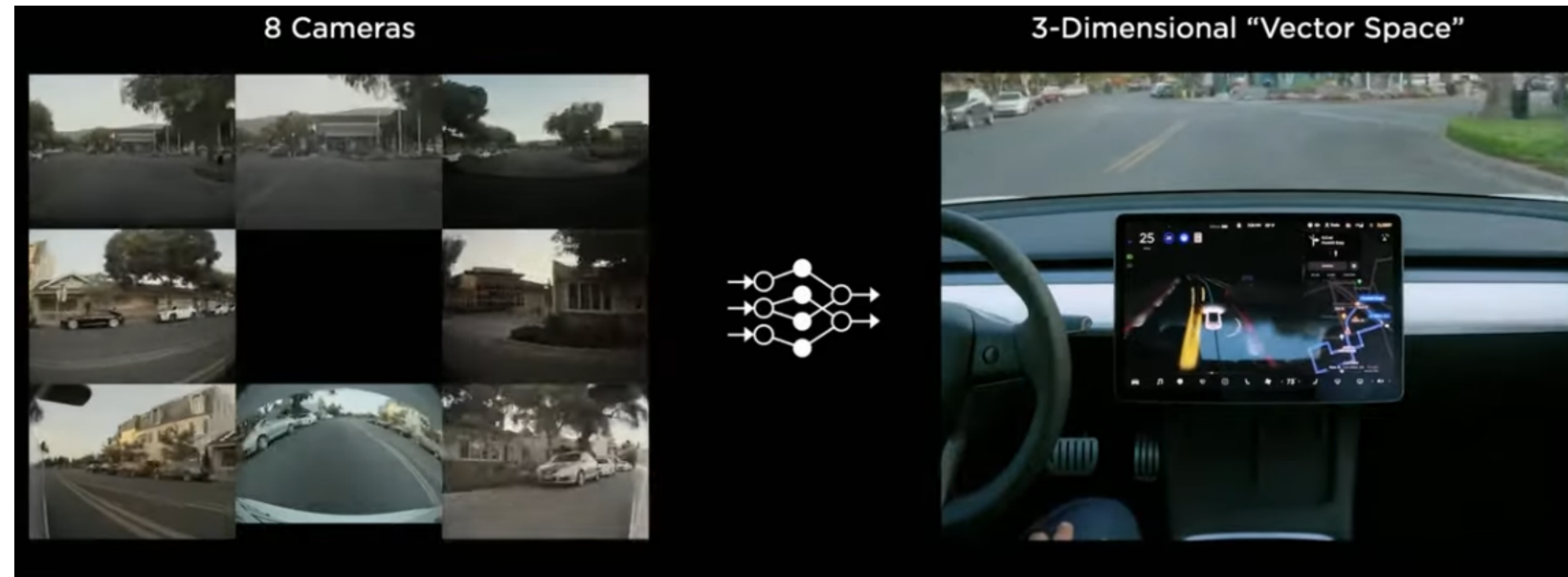
Anti Collision System

- Safety concerns



Tesla's perception module

1. Feature extractor: backbone
2. Multi scale feature fusion
3. Multi camera fusion
4. Time filtering
5. Multi task decision module
 - item location (cars, pedestrian...)
 - traffic signs (Stop sign, traffic light...)
 - lane prediction
 - ...



Tesla's perception module

- | | |
|---|---|
| 1. Feature extractor: backbone | ResNet50 (2020), RegNet (2021)
(from a CVPR 2020 Facebook paper) |
| 2. Multi scale feature fusion | EfficientDet
(from a 2019 Google paper) |
| 3. Multi camera fusion | Transformers
(from a 2020 Facebook paper) |
| 4. Time filtering | LSTM (recurrent neural network)
NIPS (1996) |
| 5. Multi task decision module | Hydranet
CMU & Google (2018) |
| ○ item location (cars, pedestrian...) | |
| ○ traffic signs (Stop sign, traffic light...) | |
| ○ lane prediction | |
| ○ ... | |

This perception module contains: 48 networks, 1,000 outputs tensors, 70,000 GPU h to train
performs 40 prediction per second

Outline

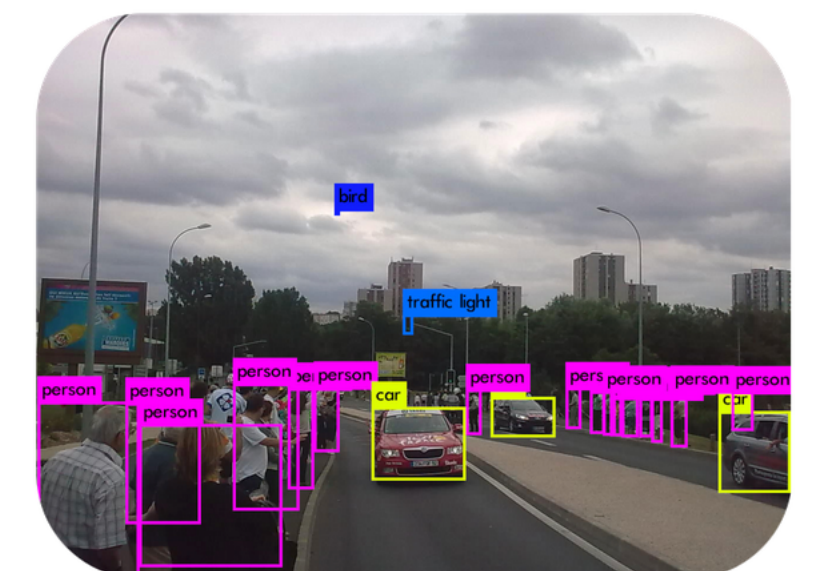
1. Introduction
2. AI in cars
3. **Adversarial attacks and robustness**
4. Exact attacks



PROJECT

Recent progresses in machine learning in general and especially in deep learning make it possible to include this technology in more and more autonomous vehicles. However, before this possible future becomes reality and our roads are made safer with algorithms replacing human drivers, it is necessary to know how to prove the quality of the decisions made.

This project aims at strengthening local research dynamics about safety issues associated with the use of artificial intelligence in mobility. To achieve this goal, the project will endeavor to formalize the problem, to propose algorithms to solve it and to demonstrate its feasibility



<https://chaire-raimo.github.io/>

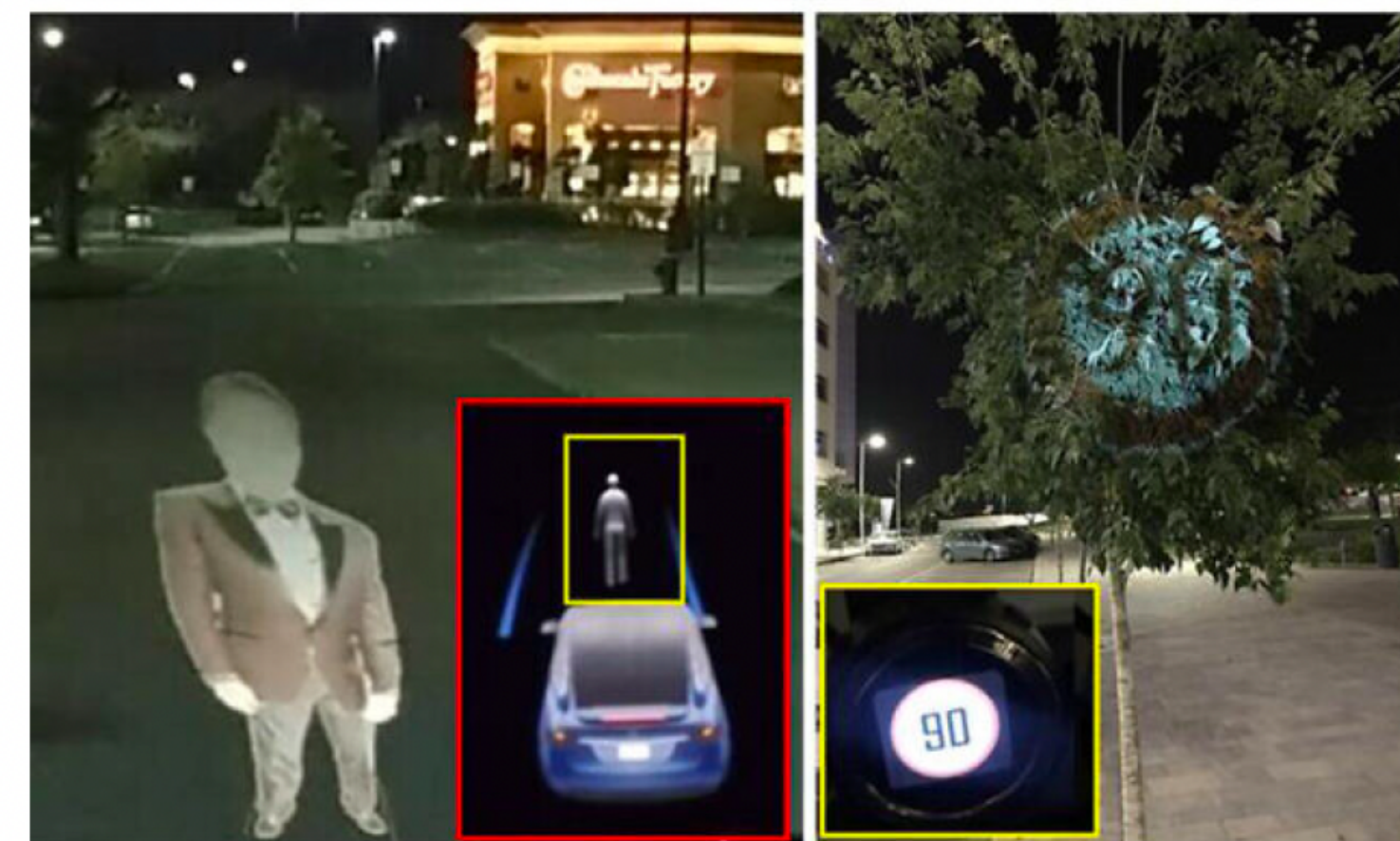
Attacks



Eykholt et al, Robust Physical-World Attacks on Deep Learning Visual Classification, CVPR 2018



Zhang et al., CAMOU: Learning Physical Vehicle Camouflages to Adversarially Attack Detectors in the Wild, ICLR 2019



<https://www.mcafee.com/blogs/other-blogs/mcafee-labs/model-hacking-ad-as-to-pave-safer-roads-for-autonomous-vehicles/>

Nassi et al., Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks, 2020

Qayyum, et al., Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial ML, IEEE Communications, 2019

Intriguing properties of neural networks, Szegedy ICLR 2014



Figure 5: Adversarial examples generated for AlexNet [9].(Left) is a correctly predicted sample, (center) difference between correct image, and image predicted incorrectly magnified by 10x (values shifted by 128 and clamped), (right) adversarial example. All images in the right column are predicted to be an “ostrich, *Struthio camelus*”. Average distortion based on 64 examples is 0.006508. Please refer to <http://goo.gl/huaGPb> for full resolution images. The examples are strictly randomly chosen. There is not any postselection involved.



Adversarial examples

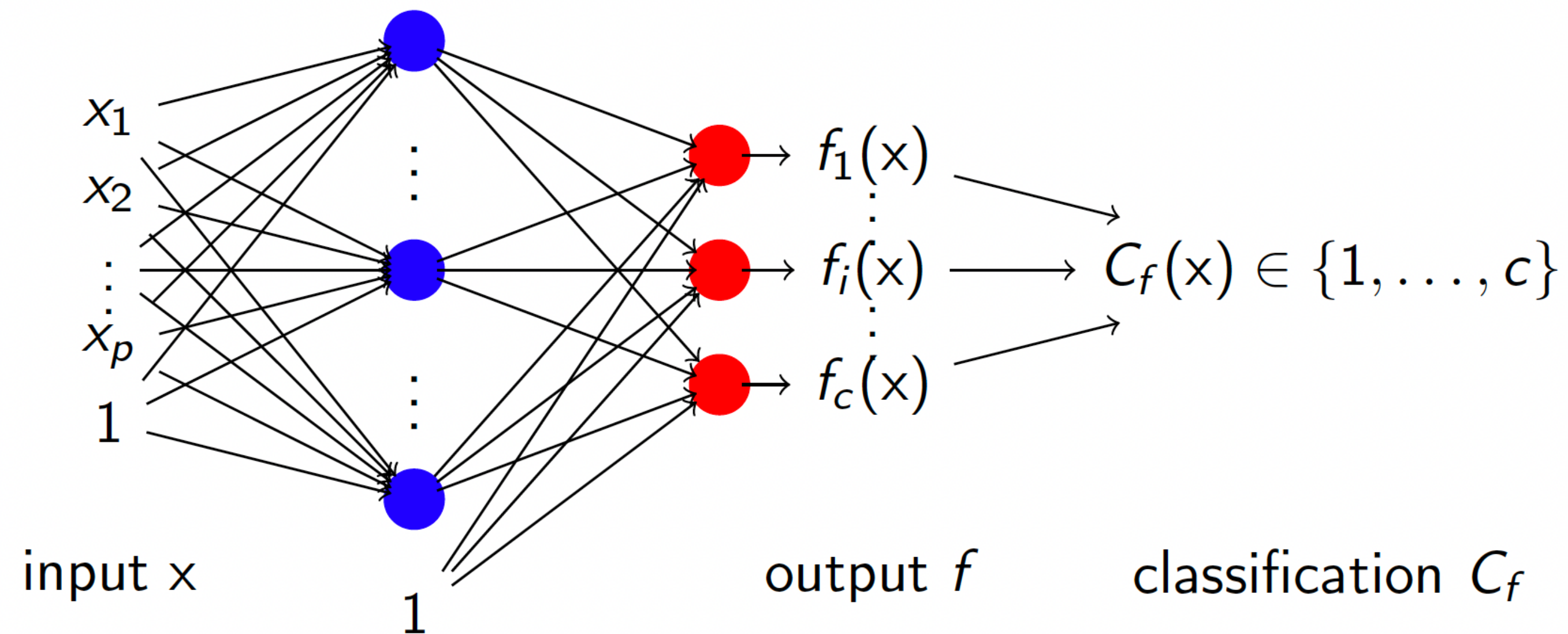
Classification model

A classification model (e.g. Neural Network) with c output nodes

$$\begin{aligned} f : \mathcal{X} \subseteq \mathbb{R}^p &\longrightarrow \mathbb{R}^c \\ x &\longmapsto f(x) \end{aligned}$$

The associated classification (or decision function)

$$C_f(x) = \operatorname{argmax}_{k=1,\dots,c} f_k(x)$$

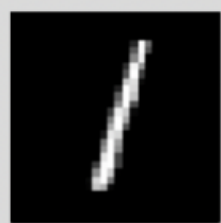
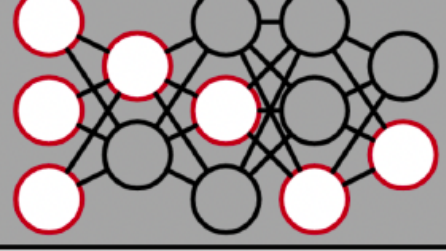


Adversarial examples

Definition (Generic adversarial)

$a_{f,x}$ is an adversarial example of f at x if $a_{f,x}$ is a valid input close to x and

$$C_f(x) \neq C_f(a_{f,x}) \quad \text{that is} \quad c^\star = \operatorname{argmax}_{k=1,\dots,c} f_k(x) \neq \operatorname{argmax}_{k=1,\dots,c} f_k(a_{f,x})$$

		Input	Model Activations	Output
x	Legitimate			1
$a_{f,x}$	Adversarial			4

from Papernot et al., 2016

Definition (Specific (or targeted) adversarial)

a_{f,x,t_a} is a specific adversarial example of f at x for the adversarial targeted class t_a if a_{f,x,t_a} is a valid input close to x and

$$f_{c^\star}(a_{f,x,t_a}) + \alpha \leq f_{t_a}(a_{f,x,t_a}) \quad \text{with} \quad c^\star = \operatorname{argmax}_{k=1,\dots,c} f_k(x)$$

Adversarial noise

Definition (adversarial noise (or perturbation or distortion))

A vector $\Delta_{f,x}$ is an adversarial noise of f at x if

$$a_{f,x} = x + \Delta_{f,x}$$

is an adversarial example for f at x

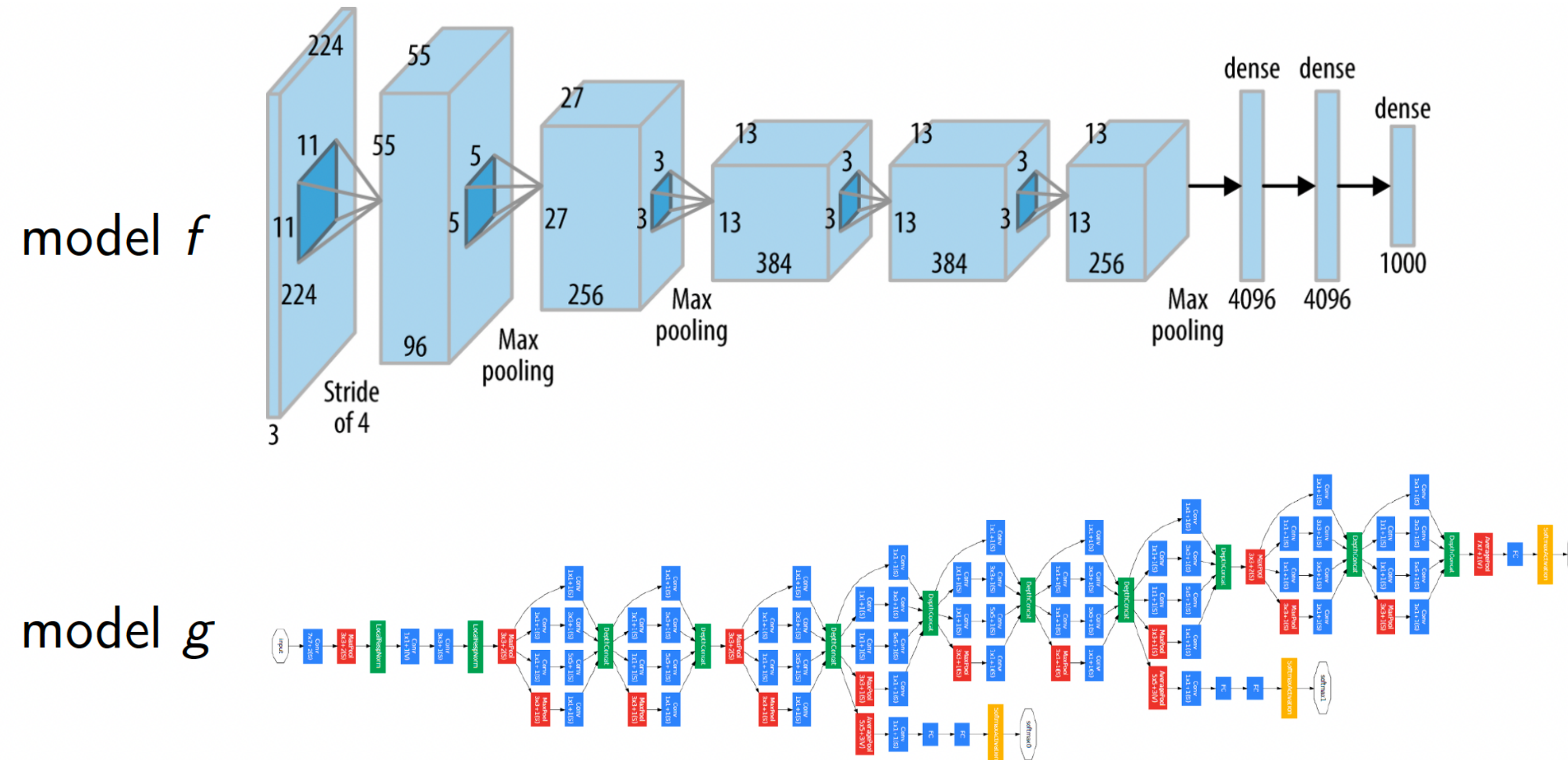
Given $a_{f,x}$ the associated adversarial noise is $\Delta_{f,x} = x - a_{f,x}$

Definition (**Universal** adversarial perturbation)

A perturbation Δ_f is a universal of f if, **for any** $x \in \mathcal{X}$, $a_f = x + \Delta_f$ is a generic adversarial example for f at x , that is

$$IP(C_f(x) \neq C_f(x + \Delta_f)) \quad \text{large}$$

Adversarial transfer



Definition (Adversarial transfer)

An adversarial example $a_{f,x}$ of classification model f at input x adversarially transfers on model g if

$$C_g(x) \neq C_g(a_{f,x})$$

3 components to define adversarial examples

- a **valid** example $a \in \mathcal{X}$ (feasible solution)
- adversarial close to x : $D(x, a)$ a **dissimilarity measure** (a distance)
- $\operatorname{argmax}_{k=1, \dots, c} f_k(x) \neq \operatorname{argmax}_{k=1, \dots, c} f_k(a)$: **adversarial loss** L ,

$$\begin{aligned} L : \quad \mathbb{R}^c \times \mathbb{R}^c &\longrightarrow \mathbb{R} \\ s, t &\longmapsto L(s, t) \end{aligned}$$

- ▶ **training class**: $c^\star = \operatorname{argmax}_{k=1, \dots, c} f_k(x)$ with training pair

$$(x, c^\star) \Rightarrow \max_a L(f(a), \text{ohe}(c^\star))$$

- ▶ **targeted class**: $t_a \neq \operatorname{argmax}_{k=1, \dots, c} f_k(x) \Rightarrow \min_a L(f(a), \text{ohe}(t_a))$

Generating adversarial examples in one step

Evasion Attacks against ML at Test Time Biggio, et al., ECML 2013

$$\begin{cases} \min_{a \in \mathcal{X}} & f_{c^*}(a) \\ \text{subject to} & \|x - a\| \leq \delta. \end{cases} \quad (1)$$

One step projected gradient descent (ρ large enough)

$$a_{f,x} = \text{Proj}_{\mathcal{A}_x}(x - \rho \nabla_x f_i(x)) \quad \text{with} \quad \mathcal{A}_x = \{a \in \mathcal{X} \mid \|x - a\| \leq \delta\}$$

Fast Gradient Sign Method (FGSM), (I. Goodfellow et al, ICLR 2015)

The problem, given (x, t)

$$\begin{cases} \max_{a \in \mathcal{X}} & J(f(a), t) & \text{training loss} \\ \text{subject to} & \|x - a\| \leq \delta \end{cases}$$

Fast Gradient Sign Method (FGSM) ($\rho = \frac{1}{4}, .1$ or $.007$)

$$a = x + \rho \text{sign}\left(\nabla_x J(f(x), t)\right)$$

Adversarial Attack as an optimization problem

Carlini & Wagner (CW), 2017

Specific attack: Given x and $t_a \neq c^*$
$$\begin{cases} \min_{a \in \mathcal{X}} & D(x, a) \\ \text{subject to} & C_f(a) = t_a \end{cases}$$

Define an objective function L such that $C_f(a) = t_a$ iff $L(f(a), t_a) \leq 0$

$$\begin{cases} \min_{a \in \mathcal{X}} & D(x, a) \\ \text{subject to} & L(f(a), t_a) \leq 0 \end{cases} \qquad \min_{a \in \mathcal{X}} D(x, a) + \lambda L(f(a), t_a)$$

- stochastic gradient descent solver (SGD is slow, use GPU)
- compare 3 $D(x, a) = \|x - a\|_p^p$, ℓ_2 , ℓ_0 and ℓ_∞ attacks
- compare 7 objective function L

and the winner is the Carlini & Wagner ℓ_2 attack

Euclidean distance ℓ_2 and the hinge loss (with confidence α)

$$L(f(a), t_a) = \max[\alpha - (f_{t_a}(a) - \max_{k \neq t_a} f_k(a)), 0]$$

Multi-step (iterative) approach

Iterative FGSM, (PGD) (Kurakin et al, ICLR 2017)

The problem, given (x, c^*)
$$\begin{cases} \max_{a \in \mathcal{X}} & J(f(a), c^*) & \text{training loss} \\ \text{subject to} & \|x - a\| \leq \delta \end{cases}$$

The i-FGSM (PGD) proposed solution: build a sequence with (small) ρ_i

$$\begin{cases} a_0 = x \\ a_{i+1} = \text{Proj}_{\mathcal{A}_x} \left(a_i + \rho_i \text{sign} \left(\nabla_x J(f(a_i), c^*) \right) \right) \end{cases}$$

- ρ chosen to change the value of each pixel only by 1 on each step
- due to the non concavity, it only converges towards local maxima
- i-FGSM is equivalent to (the ℓ_∞ version of) Projected Gradient Descent (PGD), Madry et al., ICLR 2018 (sign?)

Multi-step (iterative) approach ...with hyperparameters to tune

Iterative FGSM, (PGD) (Kurakin et al, ICLR 2017)

The problem, given (x, c^*)
$$\begin{cases} \max_{a \in \mathcal{X}} & J(f(a), c^*) & \text{training loss} \\ \text{subject to} & \|x - a\| \leq \delta \end{cases}$$

The i-FGSM (PGD) proposed solution: build a sequence with (small) ρ_i

$$\begin{cases} a_0 = x \\ a_{i+1} = \text{Proj}_{\mathcal{A}_x} \left(a_i + \rho_i \text{sign} \left(\nabla_x J(f(a_i), c^*) \right) \right) \end{cases}$$

Stepsize? **When to stop?**

- ρ chosen to change the value of each pixel only by 1 on each step
- due to the non concavity, it only converges towards local maxima
- i-FGSM is equivalent to (the ℓ_∞ version of) Projected Gradient Descent (PGD), Madry et al., ICLR 2018 (sign?)

Parameter-free Attacks

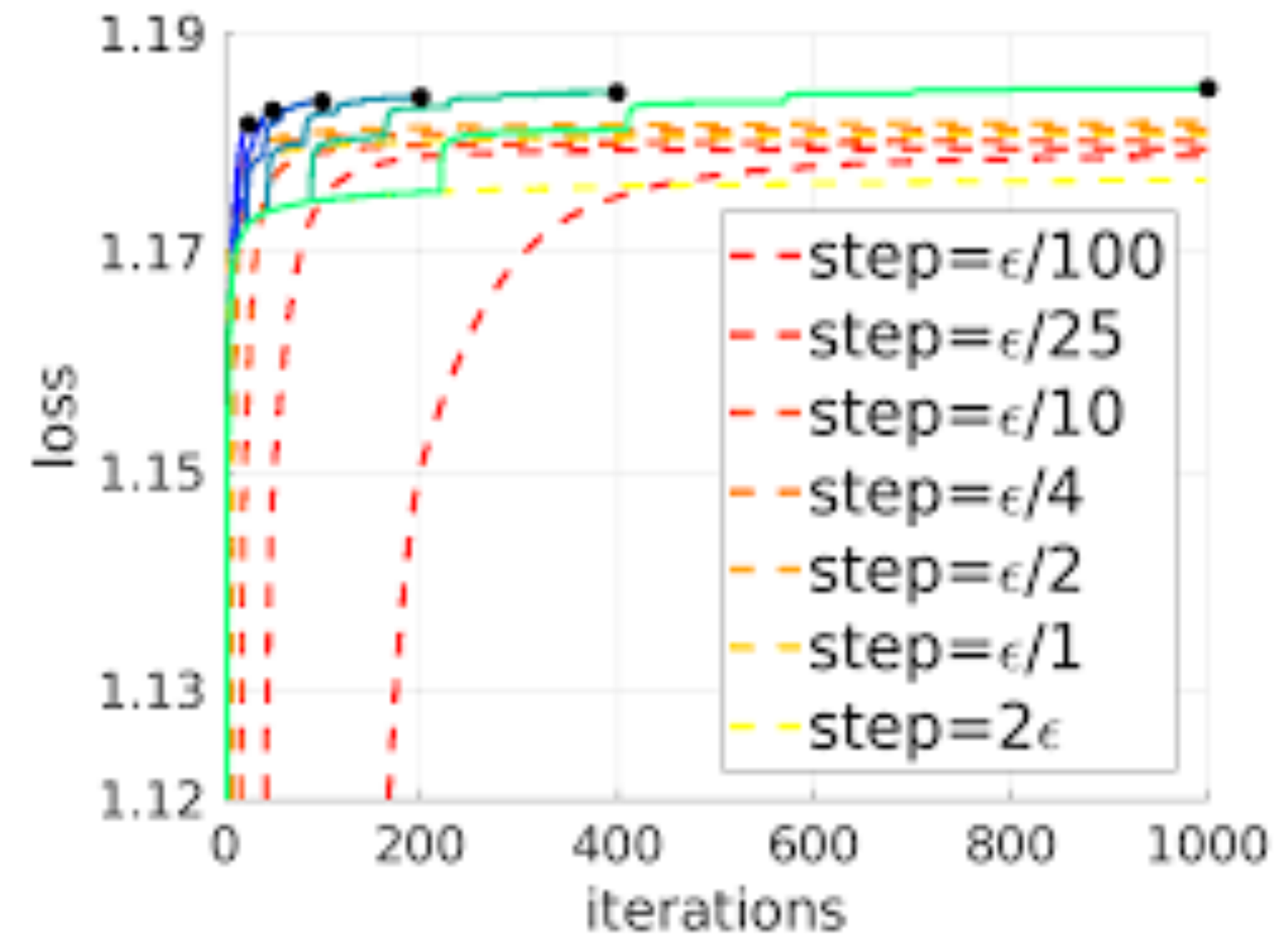
Croce & Hein, ICML 2020

Auto-Projected Gradient Descent (APGD)

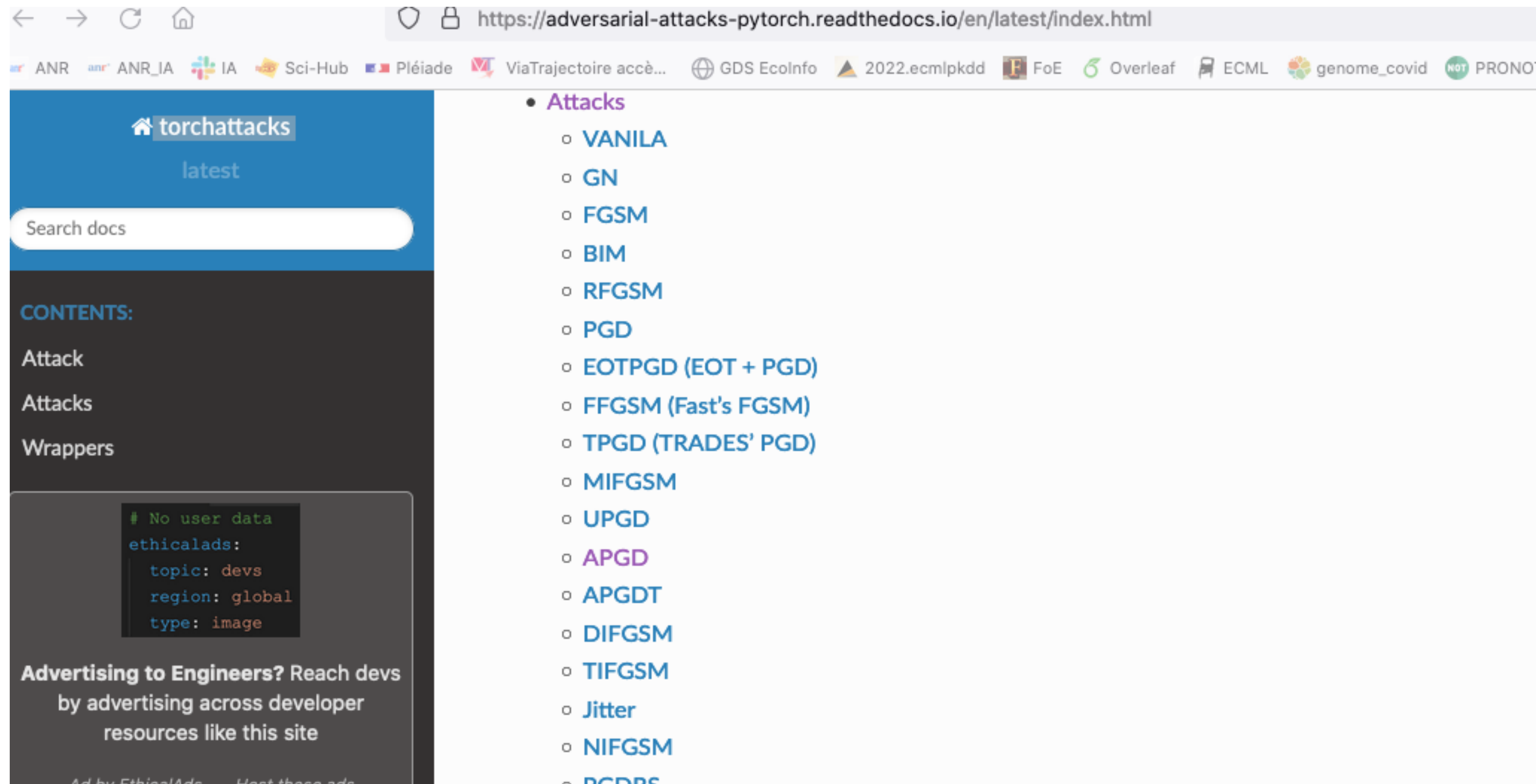
- automatic tuning of the hyperparameters
- Inspired from AutoML techniques
 - exploration
 - halving

AutoAttack

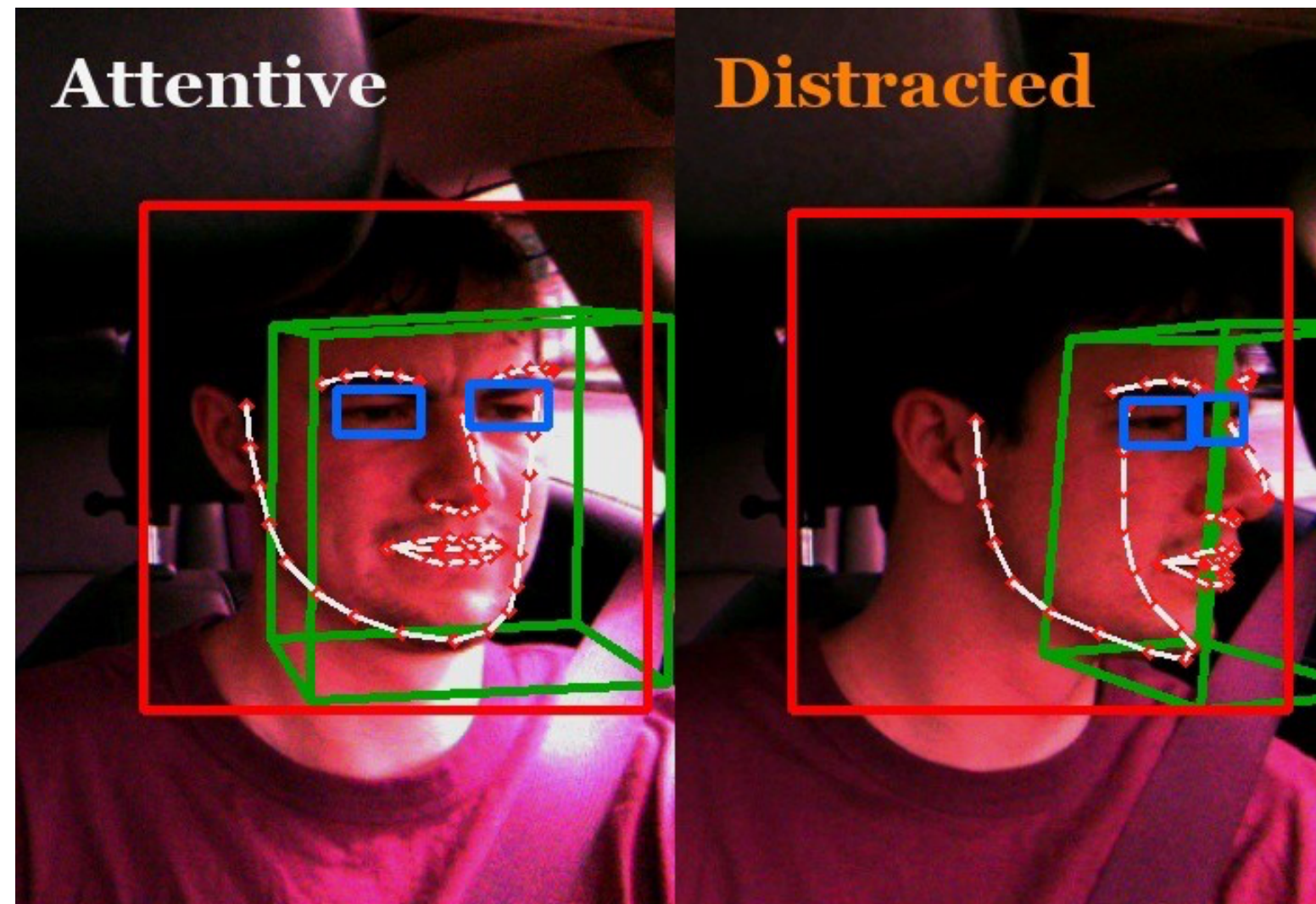
Combine 5 different Attack algorithms



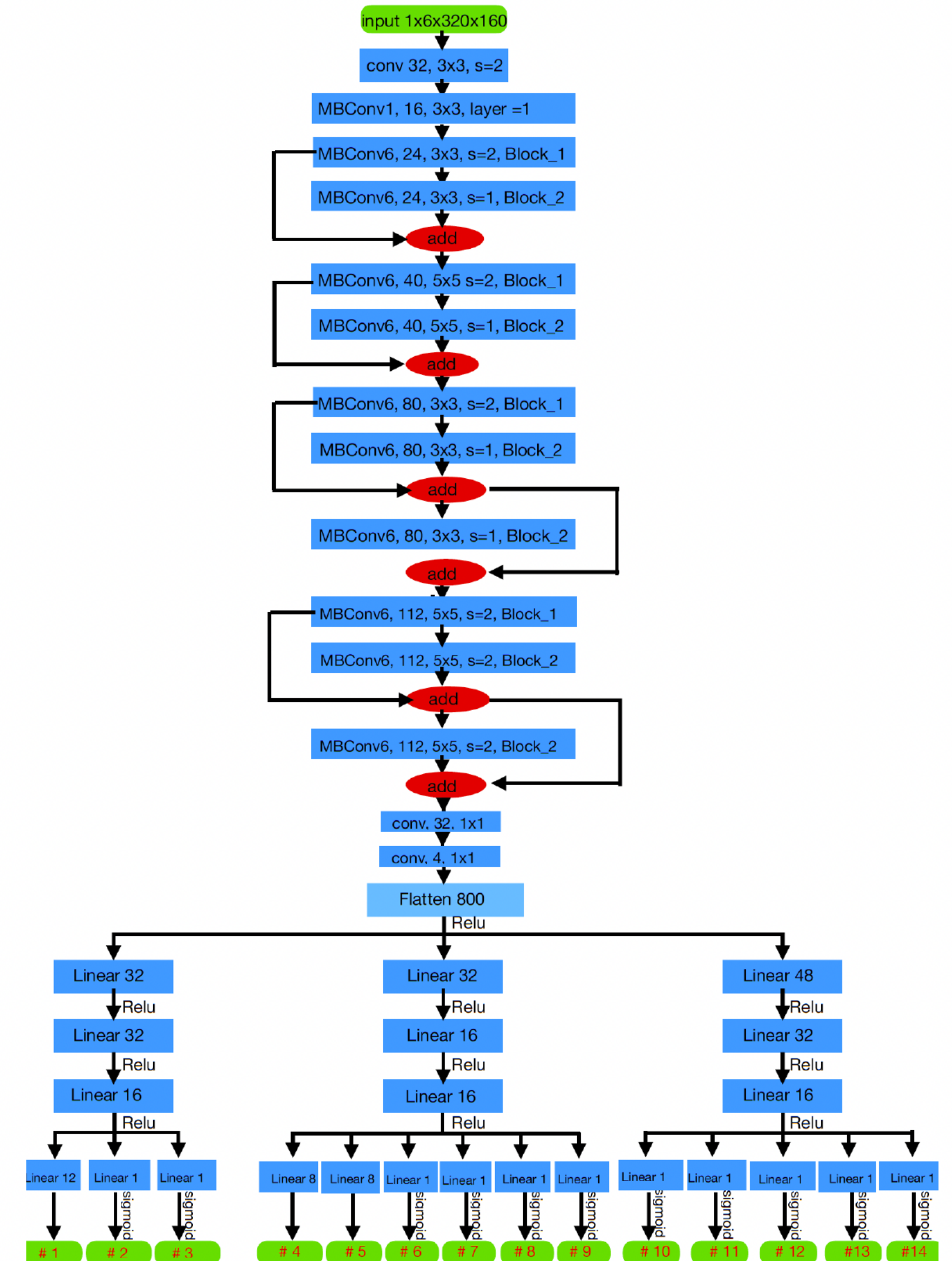
Torchattacks implementation



Driver monitoring model under attack!



- Input: YUV 420 (6 channels)
 - ▶ EfficientNet b0 architecture
 - ▶ Tan et. al. (Google), ICML 2019
- Output: 45-features (03/22)
 - ▶ Face position (12 values)
 - ▶ Eyes positions (8 values)
 - ▶ sunglasses
 - ▶ visible face probability
 - ▶ blinking
 - ▶ ...
- Training data: fine tuning
 - ▶ pytorch inside
 - ▶ Qualcomm Snapdragon 845



Datasets: Pandora & Driving Monitoring Dataset



Figure 2: Example of images from Pandora Dataset

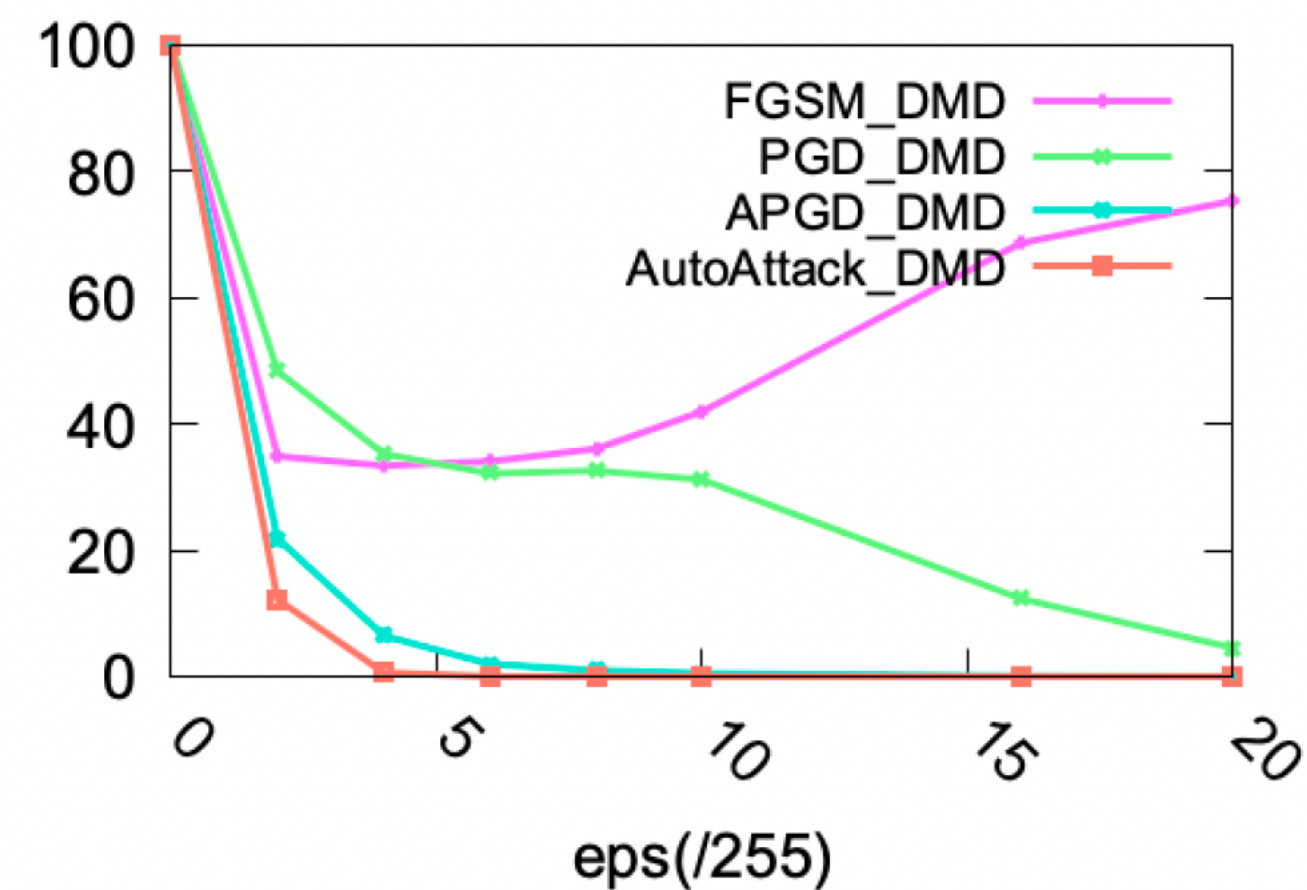
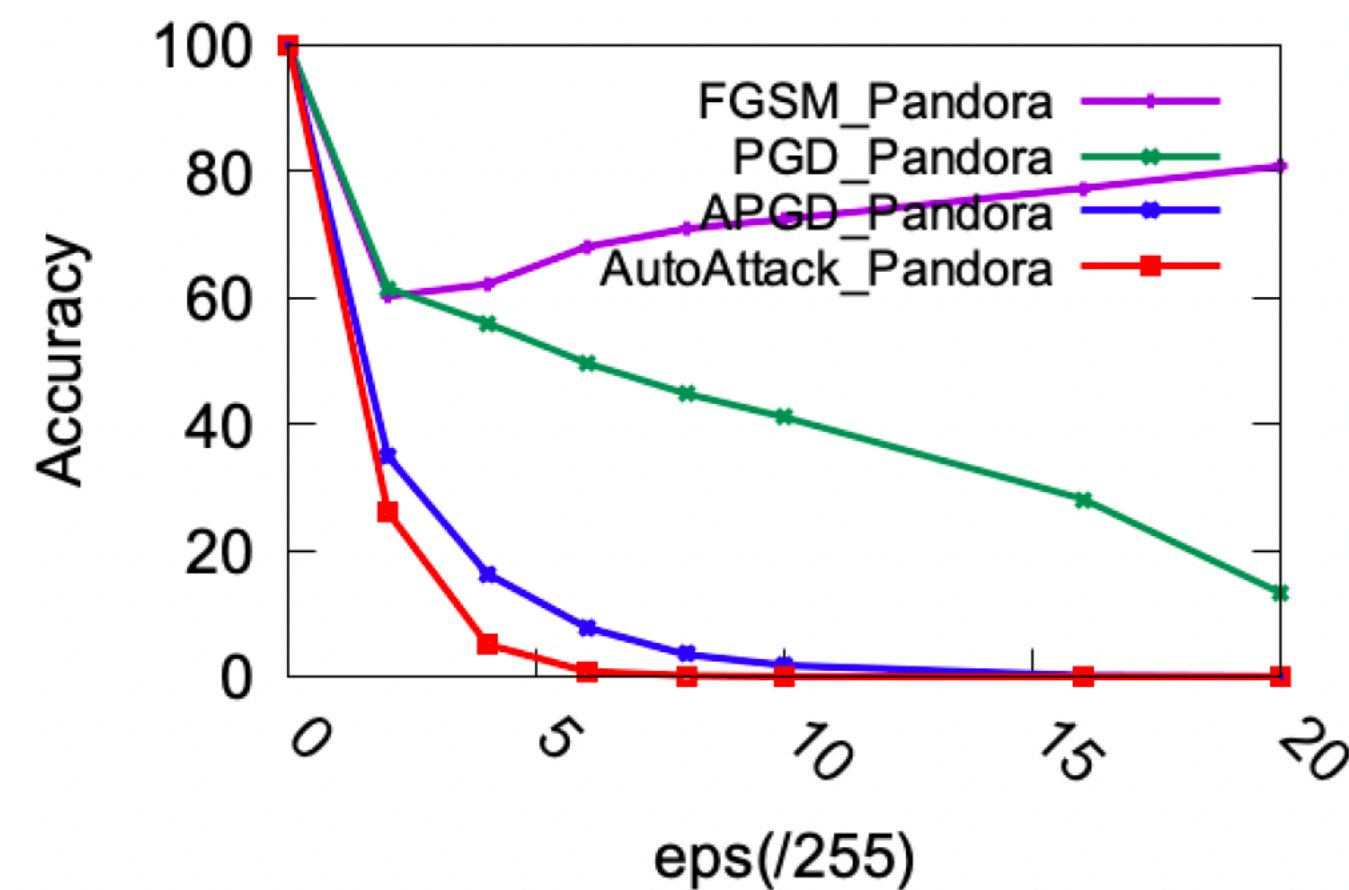
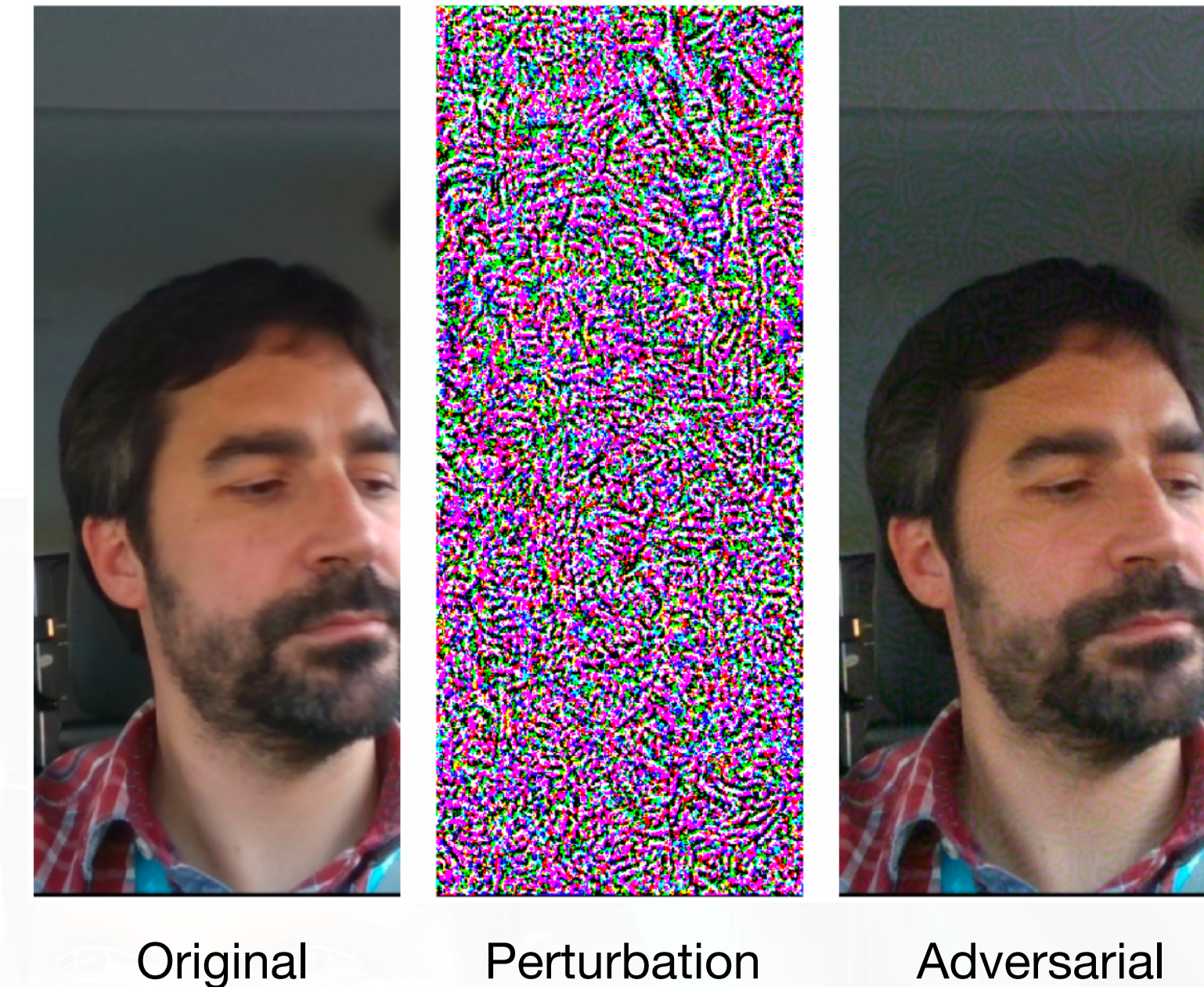


Figure 3: Example of images extracted from the DMD Dataset.

Distracted correctly detected: 10495 in Pandora and 12615 in DMD.

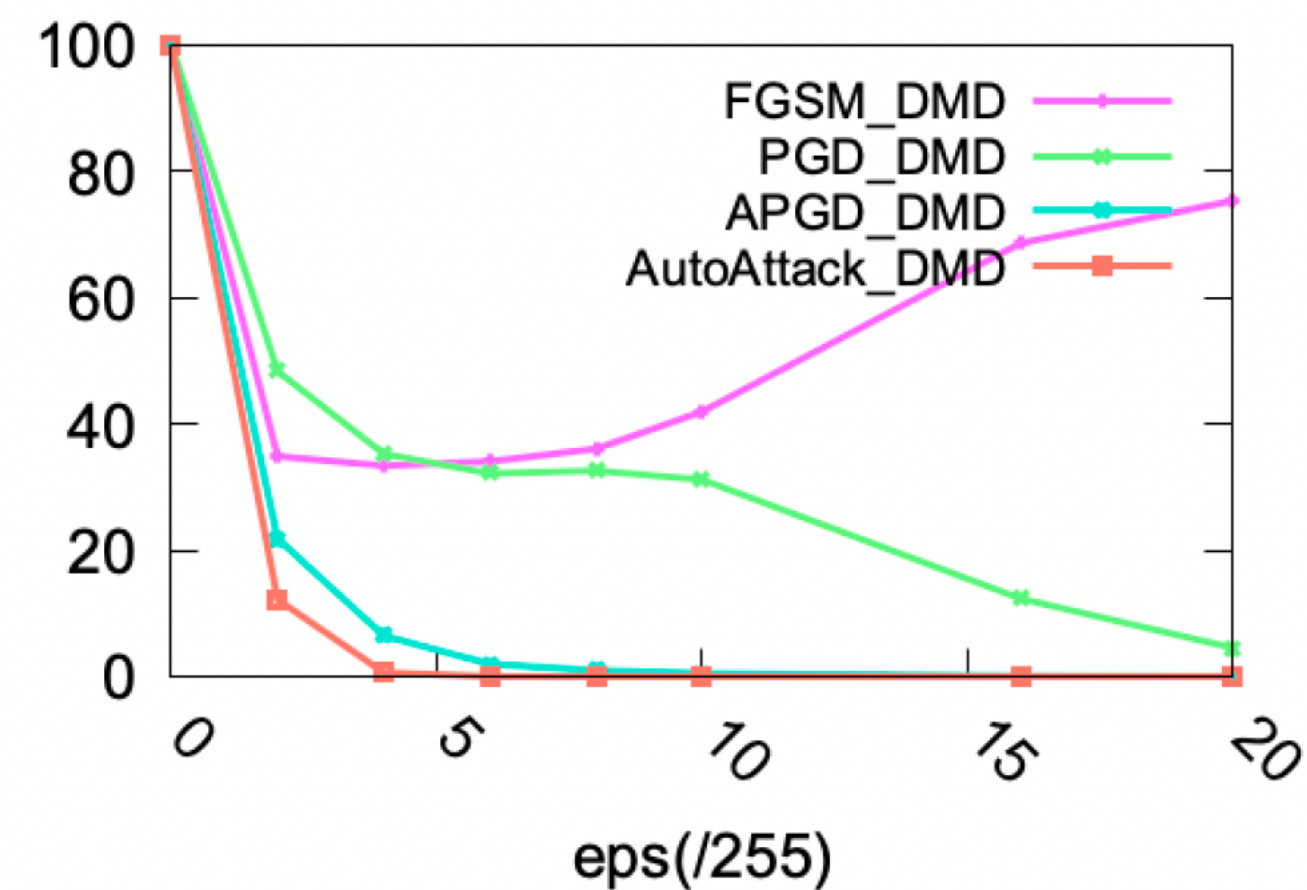
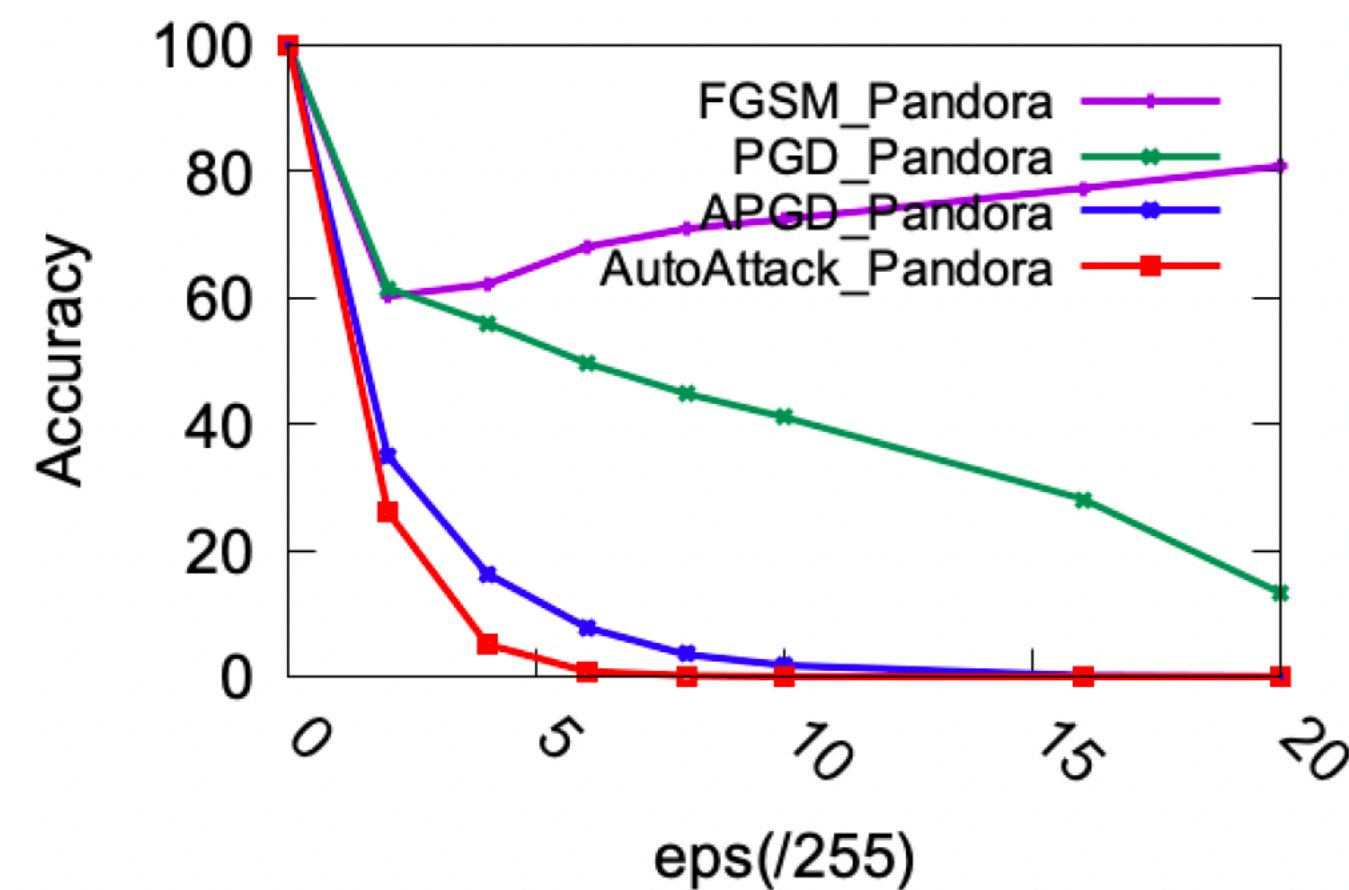
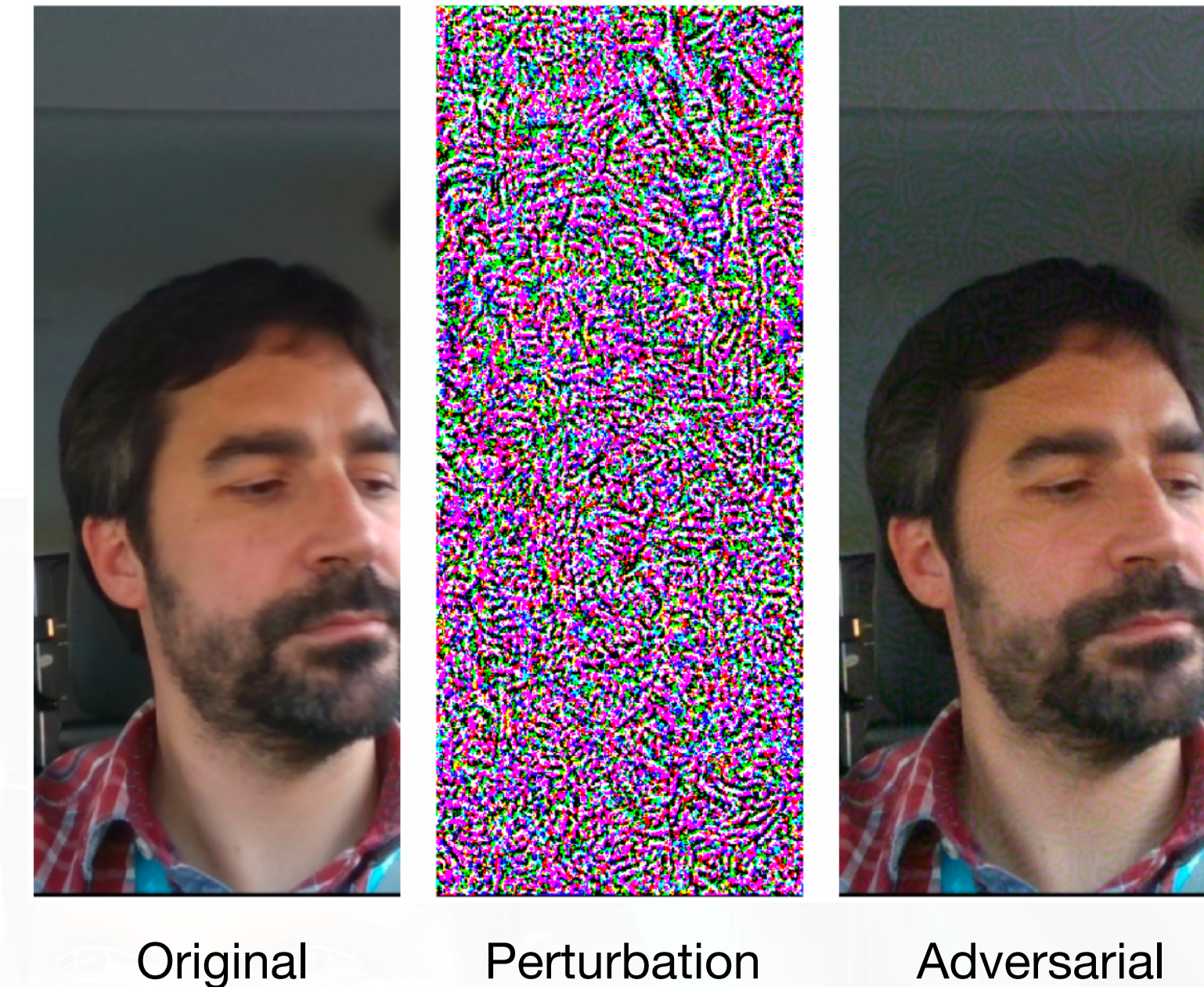
Attack performance

- Accuracy on original data: 100%
- Attack settings:
 - torchattacks
- Accuracy on adversarial data:



Attack performance

- Accuracy on original data: 100%
- Attack settings:
 - torchattacks
- Accuracy on adversarial data:



100 % Distracted



100 % Attentive

Track the progress in adversarial robustness



Leaderboard: CIFAR-10, $\ell_\infty = 8/255$, untargeted attack

Show entries

Search:

Rank	Method	Standard accuracy	AutoAttack robust accuracy	Best known robust accuracy	AA eval. potentially unreliable	Extra data	Architecture	Venue
1	Fixing Data Augmentation to Improve Adversarial Robustness <i>66.56% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)</i>	92.23%	66.58%	66.56%	×	☑	WideResNet-70-16	arXiv, Mar 2021
2	Improving Robustness using Generated Data <i>It uses additional 100M synthetic images in training. 66.10% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)</i>	88.74%	66.11%	66.10%	×	×	WideResNet-70-16	NeurIPS 2021
3	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples <i>65.87% robust accuracy is due to the original evaluation</i>	91.10%	65.88%	65.87%	×	☑	WideResNet-70-16	arXiv, Oct 2020

Outline

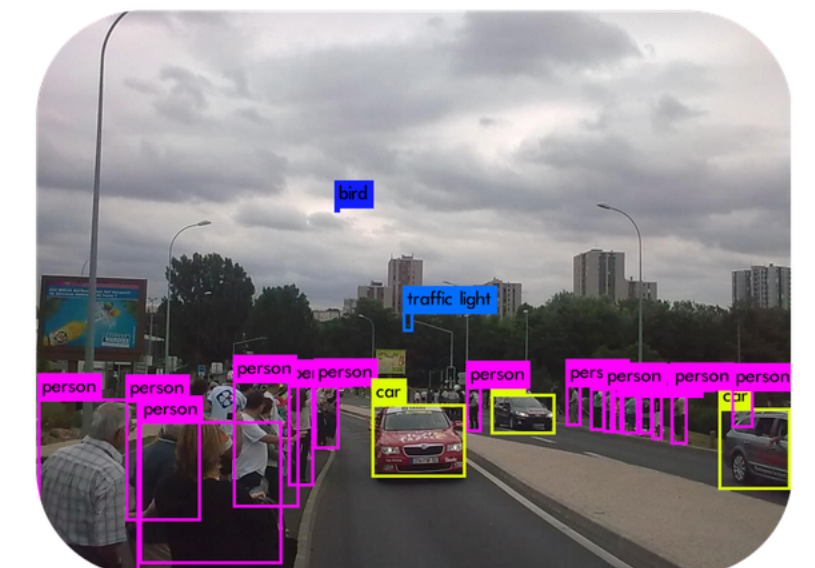
1. Introduction
2. AI in cars
3. Adversarial attacks and robustness
4. Exact attacks



PROJECT

Recent progresses in machine learning in general and especially in deep learning make it possible to include this technology in more and more autonomous vehicles. However, before this possible future becomes reality and our roads are made safer with algorithms replacing human drivers, it is necessary to know how to prove the quality of the decisions made.

This project aims at strengthening local research dynamics about safety issues associated with the use of artificial intelligence in mobility. To achieve this goal, the project will endeavor to formalize the problem, to propose algorithms to solve it and to demonstrate its feasibility



<https://chaire-raimo.github.io/>

3 formal ways to search for adversarial examples

- 1 Minimizing the Adversarial Distortion (Bunel et al., NeurIPS 2018)

$$\begin{cases} \min_{a \in \mathcal{X}} D(x, a) & = \|x - a\| \\ \text{subject to } L(f(x), f(a)) \geq \alpha & = \max_{k \neq c^*} f_k(a) > f_{c^*}(a) + \alpha \end{cases} \quad (2)$$

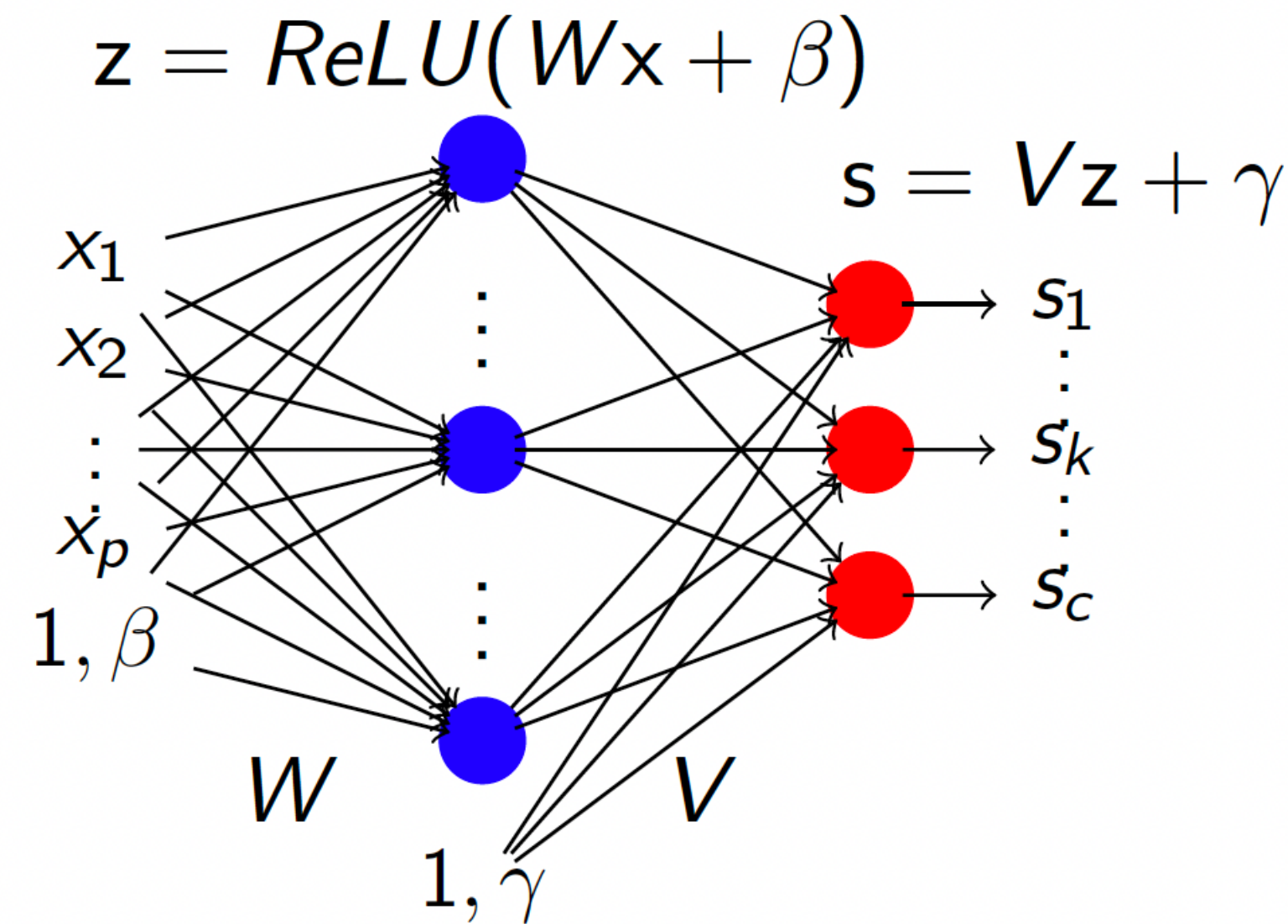
- 2 Maximizing the adversarial loss (Wong & Kolter, ICML 2018)

$$\begin{cases} \max_{a \in \mathcal{X}} L(f(x), f(a)) & = f_{t_a}(a) - f_{c^*}(a) \\ \text{subject to } D(x, a) \leq \delta & = \|x - a\| \leq \delta \end{cases} \quad (3)$$

- 3 Robustness as a verification problem (Katz et al, CAV, 2017)
A classifier f is robust to perturbations on x if and only if:

$$\begin{aligned} & \forall a \in \mathcal{A}_x, (s = f(a)) \implies \mathcal{P}(s) \\ \mathcal{A}_x = \{a \in \mathcal{X} \mid D(x, a) \leq \delta\} \quad & \mathcal{P}(s) = \max_{k \neq c^*} s_k < s_{c^*} \end{aligned} \quad (4)$$

The particular case of a one hidden layer MLP



The Neural Network function f with c output nodes

$$\begin{aligned} z &= \text{ReLU}(Wx + \beta) \\ f(x) &= Vz + \gamma \end{aligned} \quad \begin{aligned} h &= Wx + \beta, \\ z &= \max(h, 0) \\ f(x) = s &= Vz + \gamma \end{aligned}$$

The associated classification (or decision function)

$$C_f(x) = \underset{k=1, \dots, c}{\operatorname{argmax}} s_k$$

Formal verification as an optimization problem

1 Minimizing the Adversarial Distortion

$$\left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\| \\ \text{subject to} & \max_{k \neq c^*} f_k(a) > f_{c^*}(a) \end{array} \right. \quad \left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\|^2 \\ \text{subject to} & \begin{array}{l} h = Wa + \beta \\ z = \max(h, 0) \\ s = Vz + \gamma \\ \max_{k \neq c^*} s_k > s_{c^*} \end{array} \end{array} \right.$$

Formal verification as an optimization problem

1 Minimizing the Adversarial Distortion

$$\left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\| \\ \text{subject to} & \max_{k \neq c^*} f_k(a) > f_{c^*}(a) \end{array} \right. \quad \left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\|^2 \\ \text{subject to} & h = Wa + \beta \\ & z = \max(h, 0) \\ & s = Vz + \gamma \\ & \max_{k \neq c^*} s_k > s_{c^*} \end{array} \right.$$

The ReLUplex

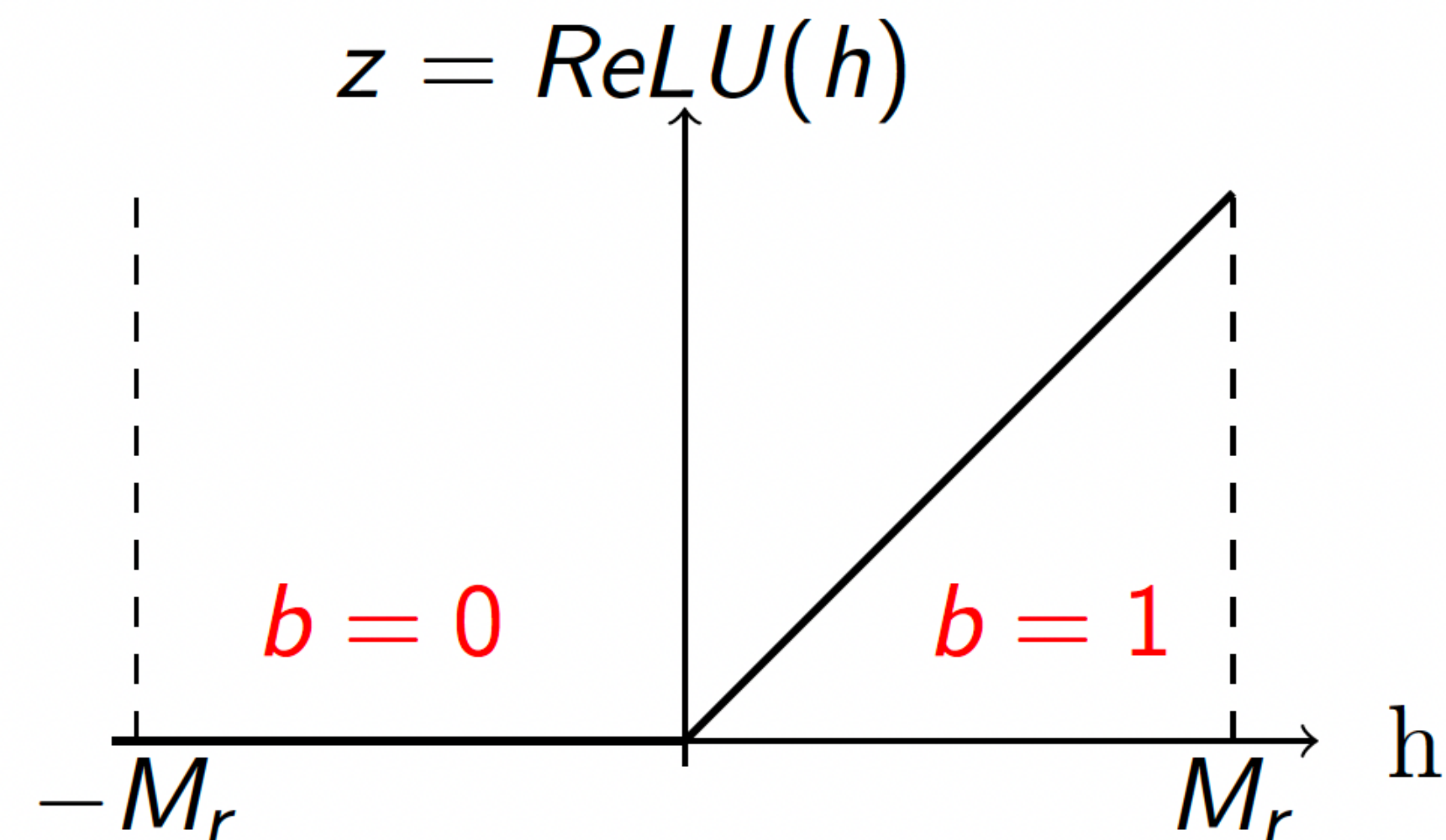
Lomuscio, Maganti, Katz et al, 2017

the ReLU can be formulated as a set of linear constraints

Given $M_r \geq \|h\|_\infty$ and **binary variables** $b \in \{0, 1\}^e$

$$z = \max(h, 0) \quad \Leftrightarrow \quad \begin{array}{ll} z_i \geq 0, & i = 1, \dots, e \\ z_i \leq M_r b_i, & i = 1, \dots, e \\ z_i \leq h_i + M_r(1 - b_i), & i = 1, \dots, e \\ z_i \geq h_i, & i = 1, \dots, e \end{array}$$

$$\begin{array}{l} b_i = 0 \Leftrightarrow z_i = 0 \\ b_i = 1 \Leftrightarrow z_i = h_i \geq 0 \end{array}$$



Exact search for adversarial examples as a MIP

Thanks to the ReLUPlex,

$$\left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\|_p^p \\ \text{subject to} & h = Wa + \beta \\ & z = \max(h, 0) \\ & s = Vz + \gamma \\ & \max_{i \neq i^*} s_i > s_{i^*} \end{array} \right\} \left\{ \begin{array}{ll} \min_{\substack{a \in [0,1]^p, \\ b \in \{0,1\}^e}} & \|x - a\|_p^p \\ \text{subject to} & h = Wa + \beta \\ & z_i \geq 0 & i = 1, \dots, e \\ & z_i \leq M_r b_i & i = 1, \dots, e \\ & z_i \leq h_i + M_r(1 - b_i) & i = 1, \dots, e \\ & z_i \geq h_i & i = 1, \dots, e \\ & s = Vz + \gamma \\ & \max_{i \neq i^*} s_i > s_{i^*} \end{array} \right.$$

$$\|x - a\|_\infty, \|x - a\|_1 \quad \text{MILP}$$

$$\|x - a\|_2^2 \quad \text{MIQP}$$

$$\|x - a\|_0 \quad \text{MILP with more binary variables}$$

→ max, convolution, pooling can also be linearized

Mixed integer linear program (MILP)

- linear cost
- linear constraints
- **integer** and continuous variables

Definition (mixed integer linear program – MILP (canonical form))

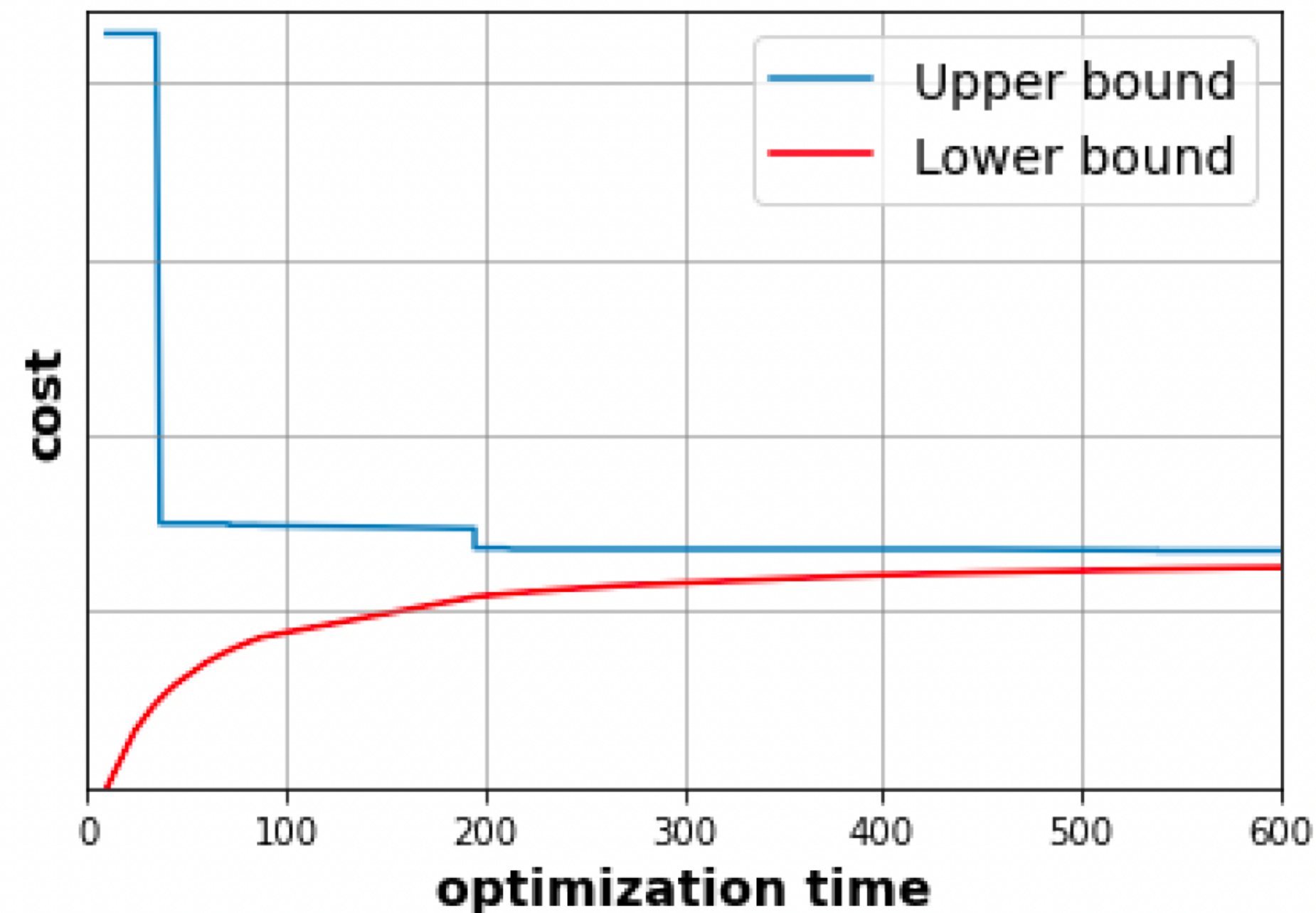
$$\left\{ \begin{array}{ll} \min_{a \in \mathbb{R}^p, \mathbf{b} \in \mathbb{N}^q} & J(a, b) = w^t a + d^t b \quad \longleftarrow \text{linear} \\ \text{s.t.} & Aw + Bz \leq c \quad \longleftarrow \text{linear} \\ & w \geq 0, \end{array} \right.$$

for some given $w \in \mathbb{R}^p$, $c \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times p}$, $B \in \mathbb{R}^{m \times q}$ and $d \in \mathbb{R}^q$.

- A **mixed binary linear program** is a MILP with $\mathbf{b} \in \{0, 1\}^q$ binary.
- When its domain is not empty and bounded, a MILP admits a unique global minimum.

MIP, Upper bound & Lower bound

$$\|x - a_{lb}\|_p^p \leq \|x - a_{x,f}^*\|_p^p \leq \|x - a_{ub}\|_p^p$$



- Optimality:
 - ▶ it may be "easy" to find the optimal solution...
 - ▶ ...and very hard to prove it
- Computational efficiency: how to manage your time budget?
 - ▶ initialization
 - ▶ acceleration through stronger relaxation

MIP acceleration using asymmetric bounds (x5)

$$\begin{array}{|c|} \hline 1.1 \\ 2.8 \\ -0.2 \\ 0.9 \\ -2.2 \\ \hline \end{array} = \begin{array}{|c|} \hline 1.1 \\ 2.8 \\ 0 \\ 0.9 \\ 0 \\ \hline \end{array} - \begin{array}{|c|} \hline 0 \\ 0 \\ 0.2 \\ 0 \\ 2.2 \\ \hline \end{array}$$

$$\mathbf{w} = \mathbf{w}_+ - \mathbf{w}_-$$

$$\ell \leq a \leq u \ \& \ h = \mathbf{w}^\top a + \beta \Rightarrow \underbrace{\mathbf{w}_+^\top \ell - \mathbf{w}_-^\top u + \beta}_{\ell'} \leq h \leq \underbrace{\mathbf{w}_+^\top u - \mathbf{w}_-^\top \ell + \beta}_{u'}$$

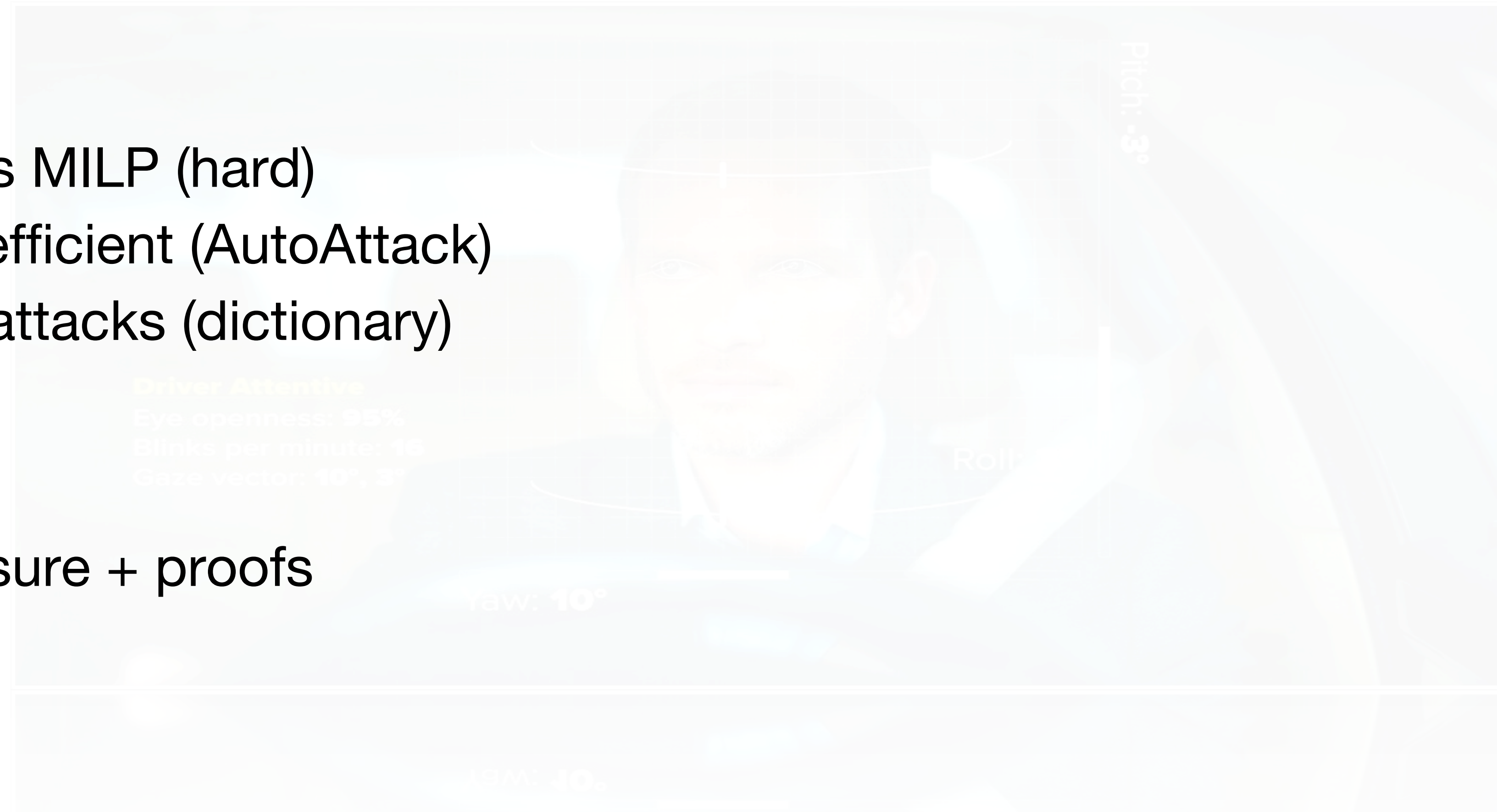
Pre computing binary variables: if $0 \leq \ell'_i$ then $b_i = 1$
 if $u'_i \leq 0$ then $b_i = 0$

Non symmetric bound (ReLU)

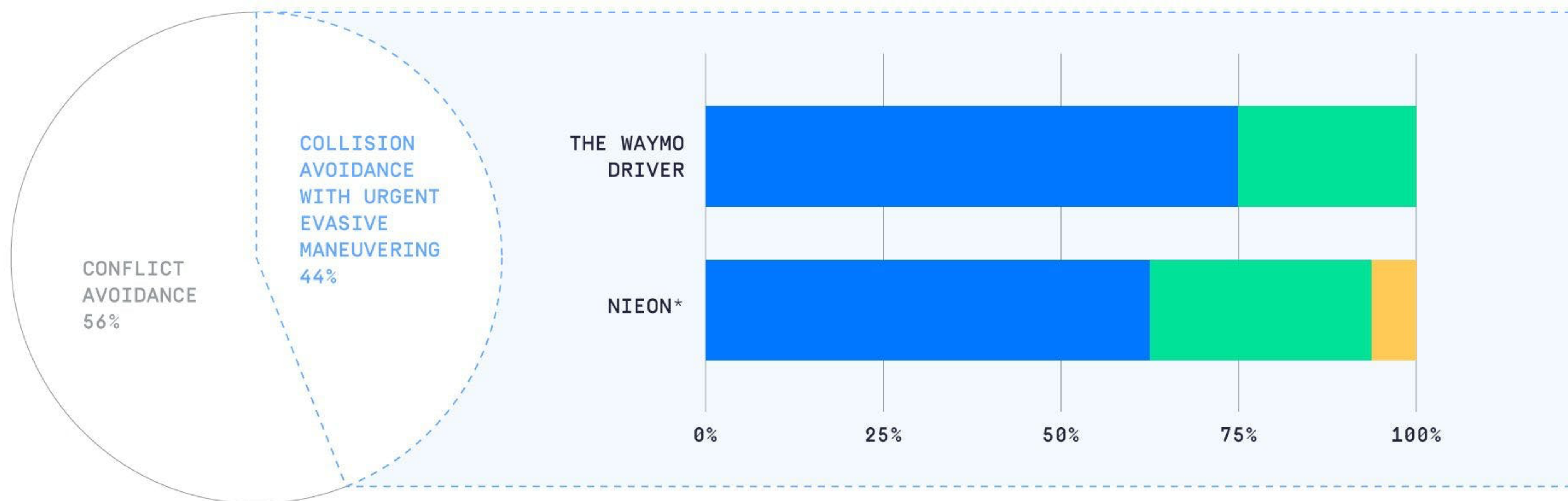
$$\begin{array}{ll} z_i \geq 0, & i = 1, \dots, e \\ z_i \leq u' b_i, & i = 1, \dots, e \\ z_i \leq h_i - \ell' (1 - b_i), & i = 1, \dots, e \\ z_i \geq h_i, & i = 1, \dots, e \end{array}$$

Conclusion

- Deep networks can be (and will be) attacked
- Adversarial attack
 - Exact solution is MILP (hard)
 - Relaxation are efficient (AutoAttack)
 - Semi universal attacks (dictionary)
- Safety guarantees
 - Reliability measure + proofs
 - Robust training



The Waymo Driver's collision avoidance performance in simulated tests

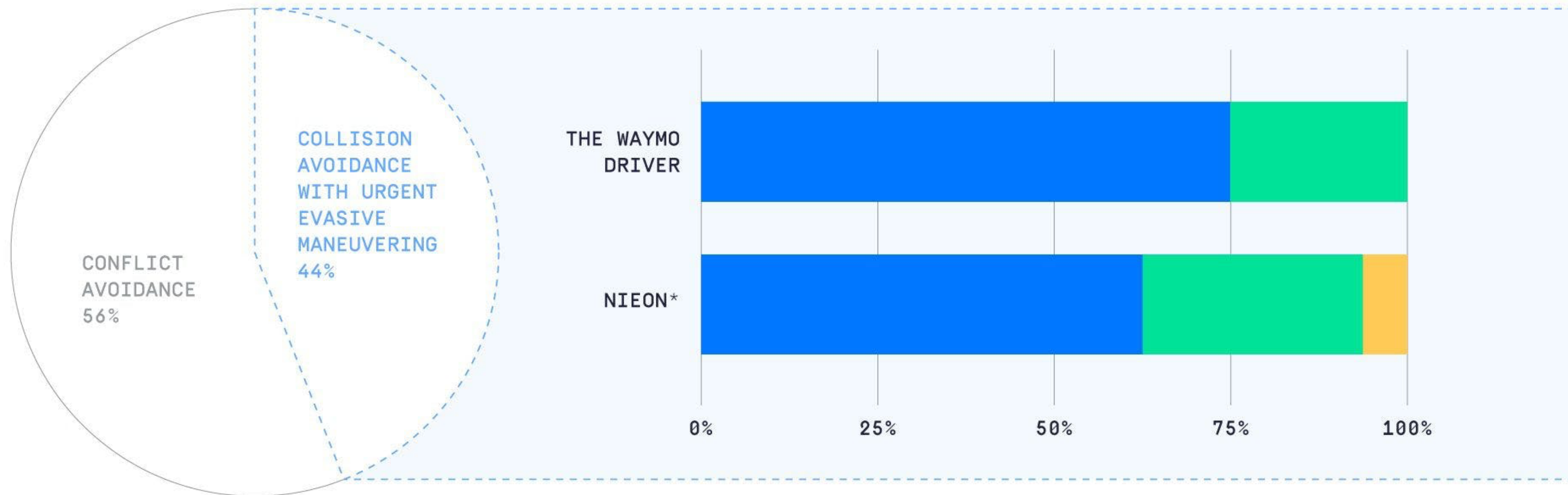


POTENTIALLY AVOIDABLE FATAL CRASHES
AMONG HUMAN DRIVERS IN CHANDLER, AZ
[2008 - 2017]

*NON-IMPAIRED, WITH EYES ALWAYS ON THE CONFLICT
HUMAN DRIVER THAT DOESN'T EXIST IN THE HUMAN POPULATION

AVOIDED CRASH ■
MITIGATED CRASH ■
CRASH NOT MITIGATED ■

The Waymo Driver's collision avoidance performance in simulated tests

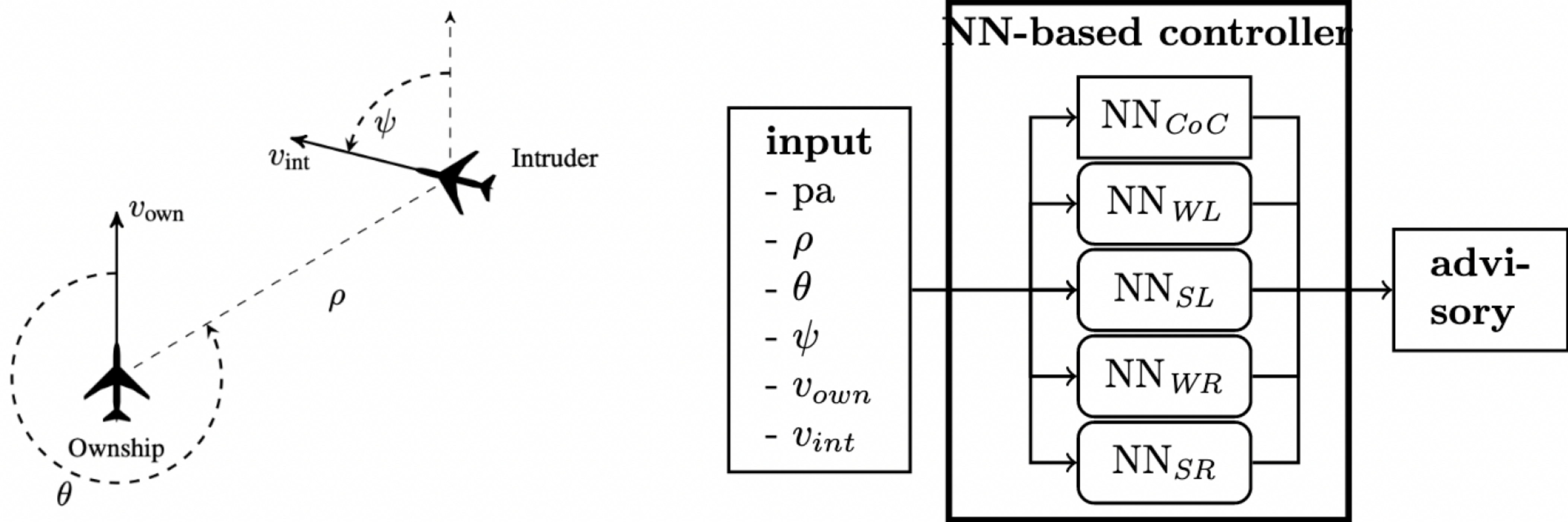


POTENTIALLY AVOIDABLE FATAL CRASHES
AMONG HUMAN DRIVERS IN CHANDLER, AZ
[2008 - 2017]

*NON-IMPAIRED, WITH EYES ALWAYS ON THE CONFLICT
HUMAN DRIVER THAT DOESN'T EXIST IN THE HUMAN POPULATION

AVOIDED CRASH ■
MITIGATED CRASH ■
CRASH NOT MITIGATED ■

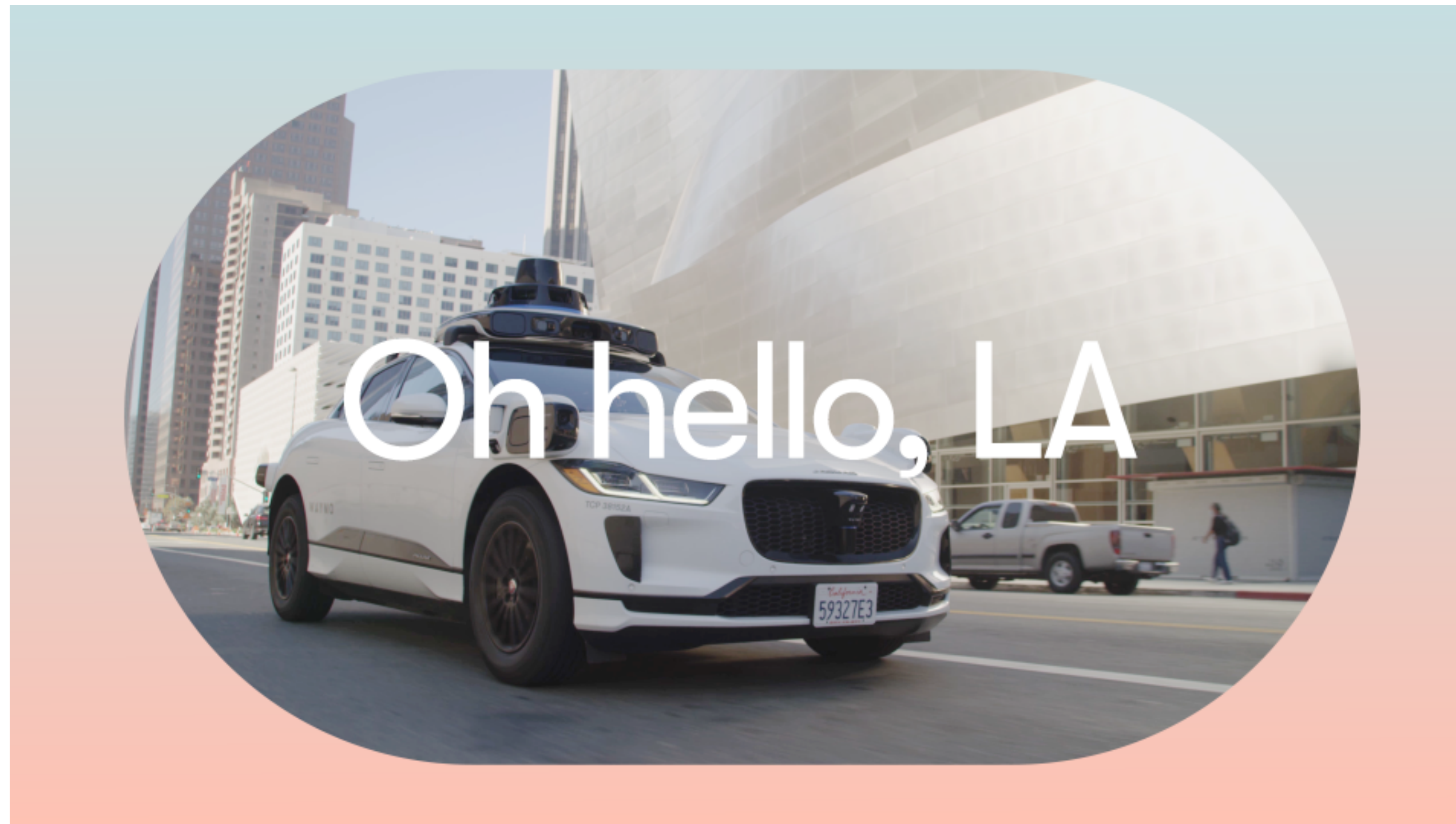
Next-Generation Airborne Collision Avoidance System



COC (Clear Of Conflict) SR (Strong Right) SL (Strong Left) WR (Weak Right) WL (Weak Left)

Waymo autonomous taxi

- In Phoenix metropolitan area (fully autonomous service)
- San Francisco (Tester program)
- Los Angeles is next



ADAS ratings

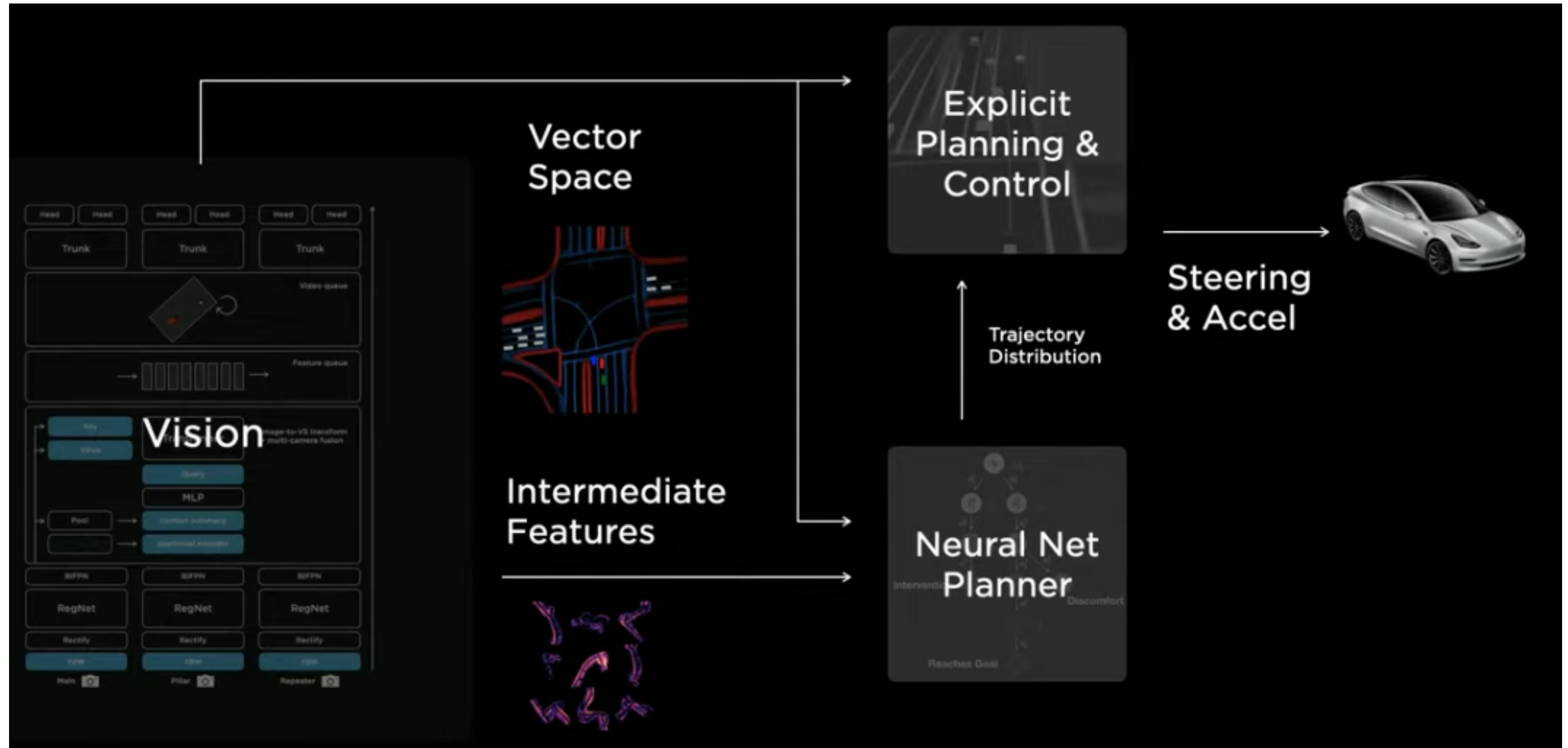
US Consumer Reports' for Advanced Driver Assistance Systems (2020)

Some players:

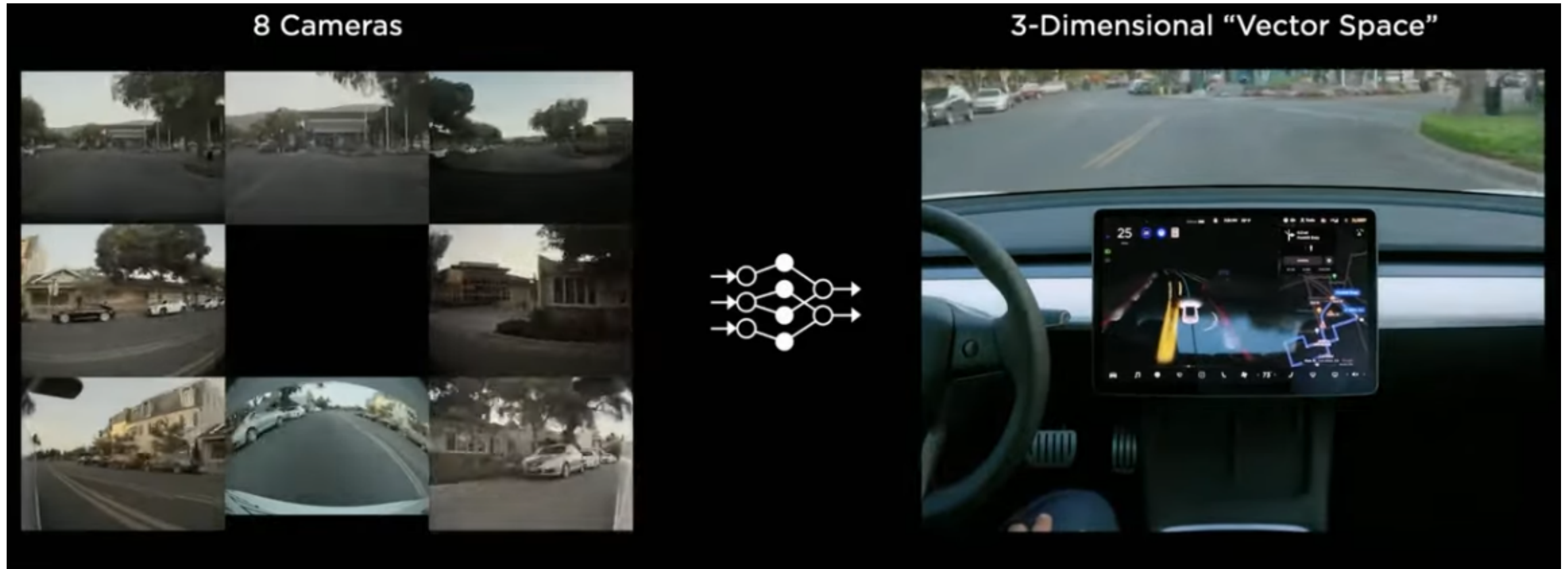
- OpenPilot (open source 50 k)
- Super Cruise (Cadillac 110 k)
- AutoPilot (Tesla, 2M)
- Mobil Eye (54 M)

SYSTEM NAME	SCORE	CAPAB. & PERF.	KEEPING DRIVER ENGAGED	EASE OF USE	CLEAR WHEN SAFE TO USE	UNRE-SPONSIVE DRIVER
Comma Two Open Pilot	78	8	9	8	6	8
Cadillac Super Cruise	69	8	7	3	8	9
Tesla Autopilot	57	9	3	7	2	6
Ford/Lincoln Co-Pilot 360	52	8	4	3	4	5
Audi Driver Assistance Plus	48	8	3	3	2	6
Mercedes-Benz Driver Assistance	46	6	4	4	2	5
Subaru Eyesight	46	7	4	3	4	5
Hyundai Smart Sense, Kia Drive Wise	46	5	4	5	4	4
BMW Active Driving Assistance Pro	44	7	3	3	2	6
Porsche Active Safe	41	4	3	6	2	5
Volvo Pilot Assist	41	6	3	3	2	5
Toyota/Lexus Safety Sense 2.0	40	5	4	2	4	5
Honda/Acura Sensing	40	6	4	2	4	4
Nissan/Infiniti ProPILOT Assist	40	5	3	3	4	7
Volkswagen Driver Assistance	39	4	3	6	2	5
Land Rover Driver Assist	38	4	3	6	2	4
Buick/Chevy Driver Confidence	36	3	3	5	2	6
Mazda i-ACTIVSENSE	27	3	2	5	2	1

Tesla's driving agent



Tesla's perception module



input: 8 cameras

output: 640 x 460 3D map of the surroundings

An infinity of driving situations ...

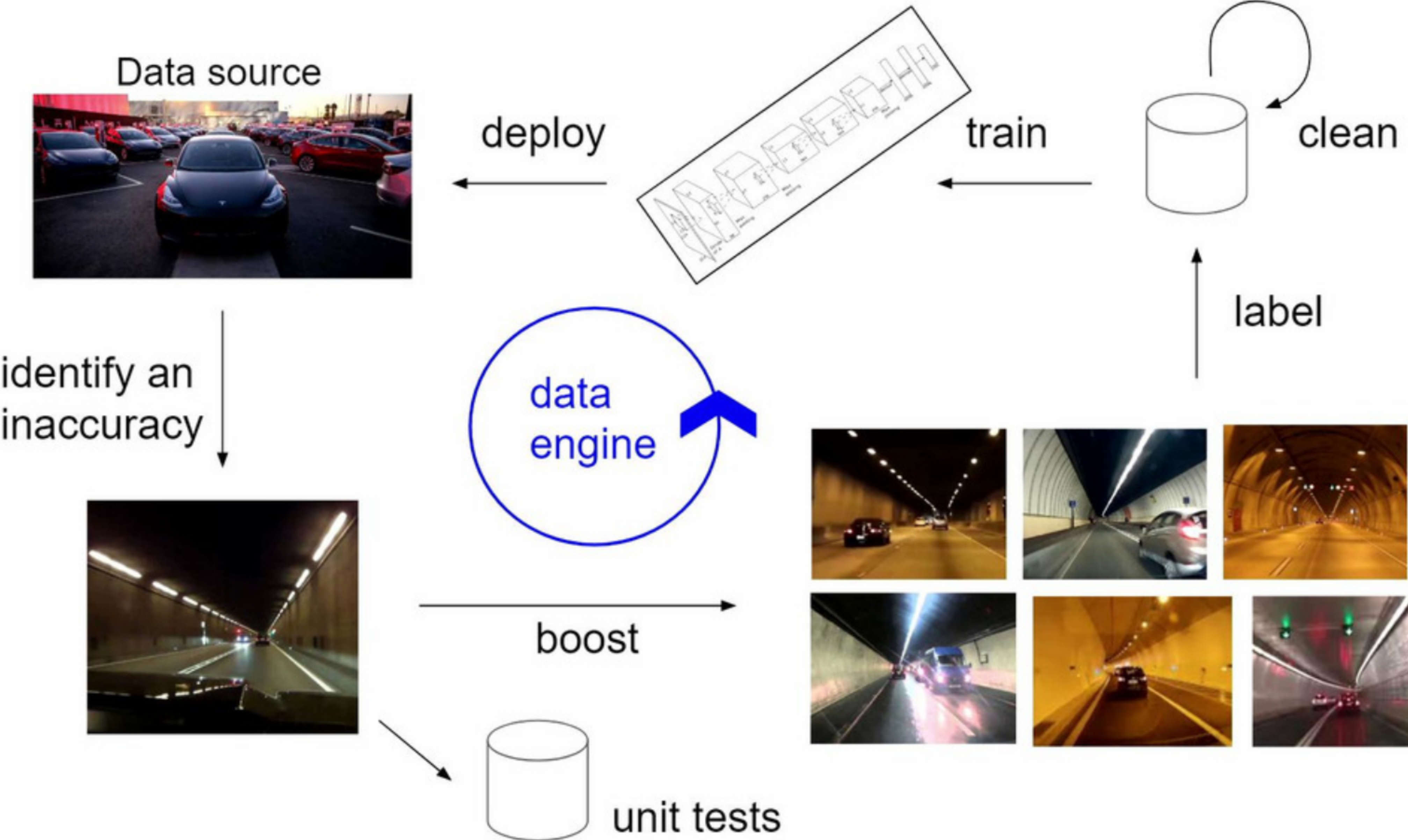


Taiwan, june 2020,



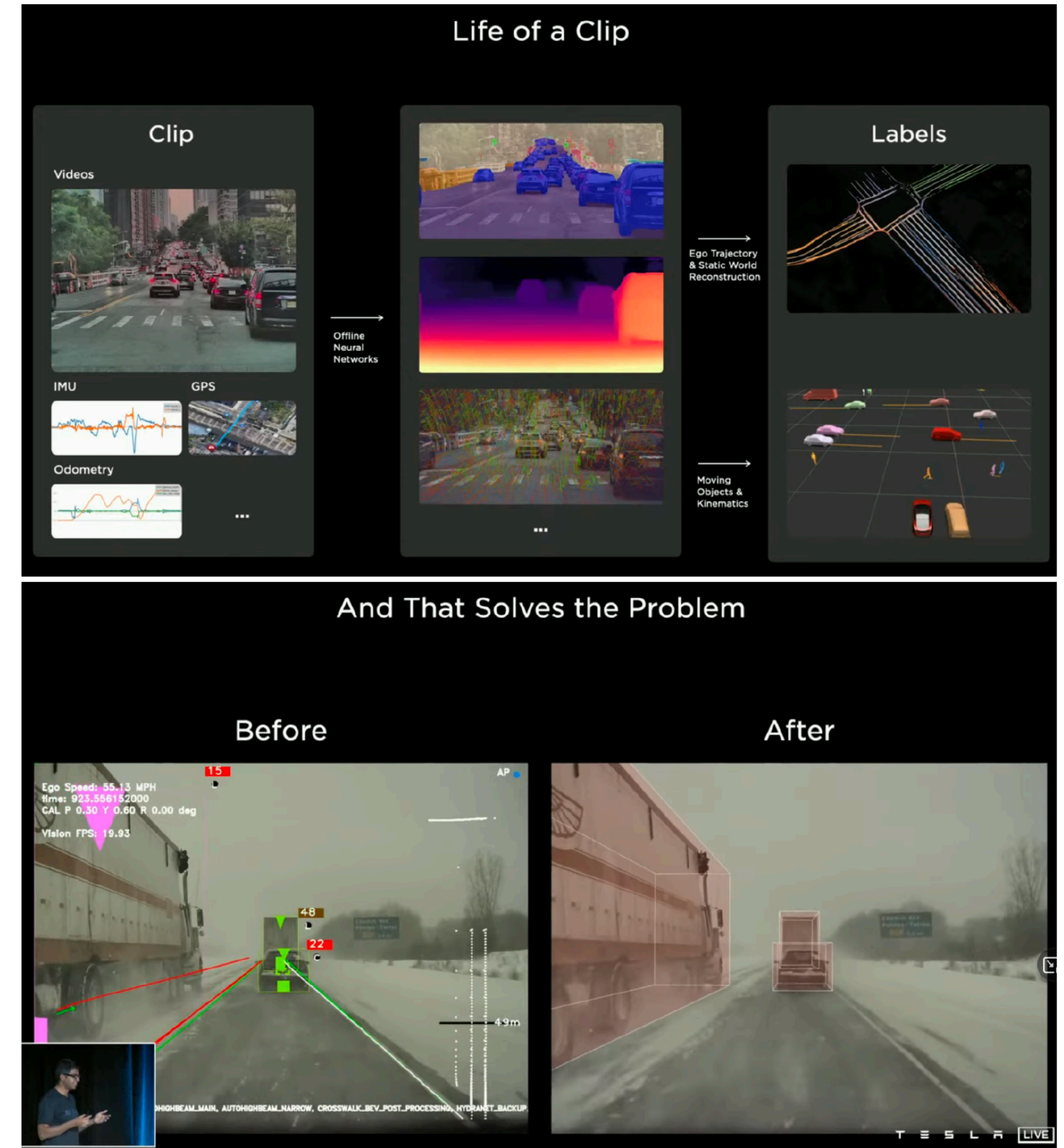
Andrej Karpathy - AI for Full-Self Driving at Tesla, Scaled ML, feb 2020,

Data engine at Tesla



Data production at Tesla

- Gathering process
 - ▶ 221 triggering situations
- manual labelling (1000 person)
 - ▶ 2d -> 3d
 - ▶ reconstruction labelling
- auto labelling
 - ▶ use specifically trained networks
 - ▶ human to clean
- simulation
 - ▶ rare event
 - ▶ sensor robustness
 - ▶ adversarial examples



Research as a challenge

WAYMO

Open

Dataset

The field of machine learning is changing rapidly. Waymo is in a unique position to contribute to the research community with some of the largest and most diverse autonomous driving datasets ever released.

Check out the newly released motion dataset in our Waymo Open Dataset and [2021 Challenges](#)!

Access Waymo
Open Dataset



A promotional image for the Waymo Open Dataset. It features a circular frame containing a street scene with a white Waymo self-driving car in the foreground. The scene is overlaid with a grid of blue dots, representing the dataset's sensor data. The background of the slide is dark grey.

Volvo data collection vehicle

