Safety and Robustness of Deep Learning

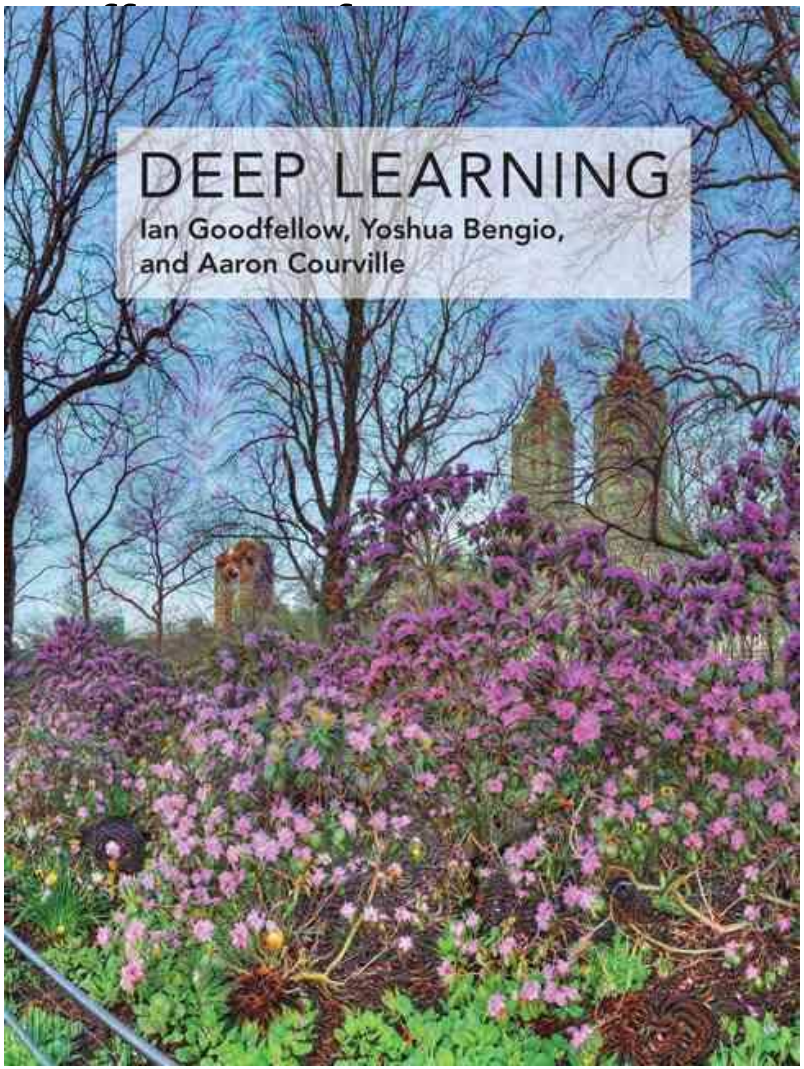
Prof. Marta Kwiatkowska

Department of Computer Science
University of Oxford

The unstoppable rise of deep learning

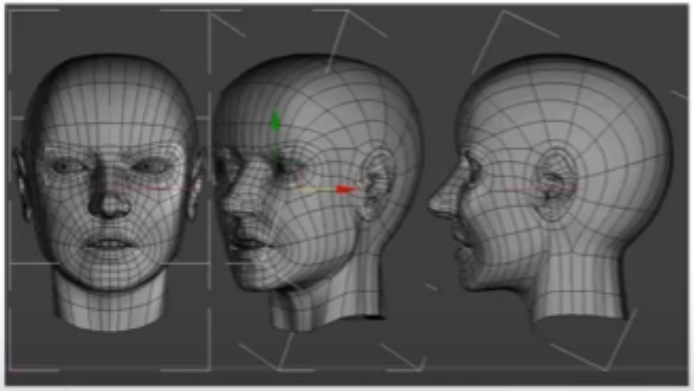
- Neural networks timeline
  - 1940s First proposed
  - 1998 Convolutional nets
  - 2006 Deep nets trained
  - 2011 Rectifier units
  - 2015 Vision breakthrough
  - 2016 Win at Go
  - 2019 Turing Award

- Enabled by
  - Big data
  - Flexible, easy to build models
  - Availability of GPUs



Much interest from tech companies,

DeepFace
Closing the Gap to Human-Level Performance in Face Verification



Yaniv Taigman
Ming Yang
Marc'Aurelio Ranzato
Lior Wolf
- 2014

97.35% accuracy
Trained on the largest facial dataset – 4M facial images belonging to more than 4,000 identities.

Google Translate—here shown on a mobile phone—will use deep learning to improve its translations between texts.

...healthcare,

nature International weekly journal of science

Home | News | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Authors

Archive > Volume 542 > Issue 7639 > Letters > Article > Article metrics > News

Article metrics for:

Dermatologist-level classification of skin cancer with deep neural networks

Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau & Sebastian Thrun

Nature **542**, 115–118 (02 February 2017) | doi:10.1038/nature21056

Last updated: 24 July 2017 10:10:28 EDT

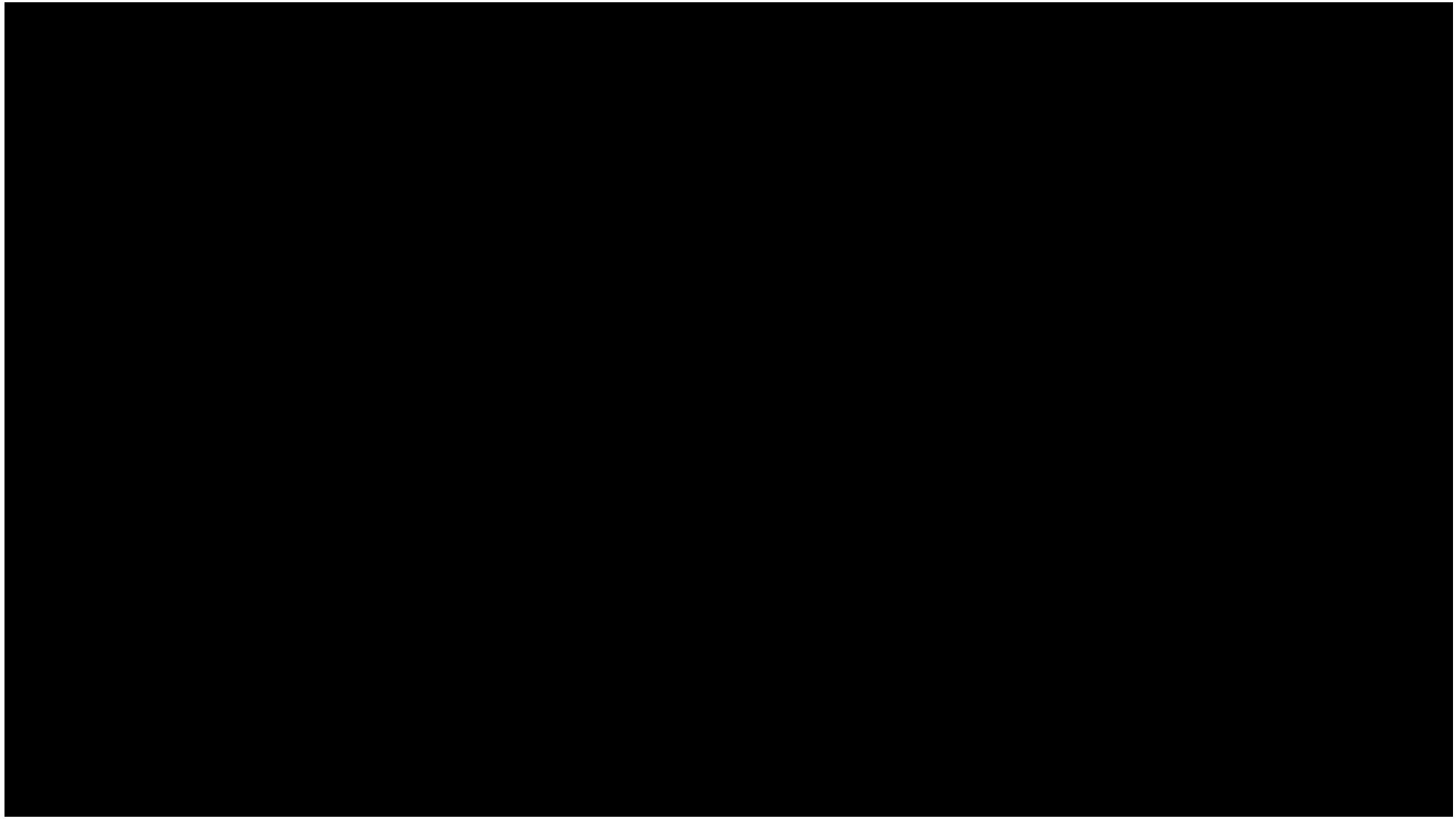
The Stanford University team said the findings were "incredibly exciting" and would now be tested in clinics.

Eventually, they believe using AI could revolutionise healthcare by turning anyone's smartphone into a cancer scanner.

Cancer Research UK said it could become a useful tool for doctors.

The AI was repurposed from software developed by Google that had learned to spot the difference **between images of cats and dogs**.

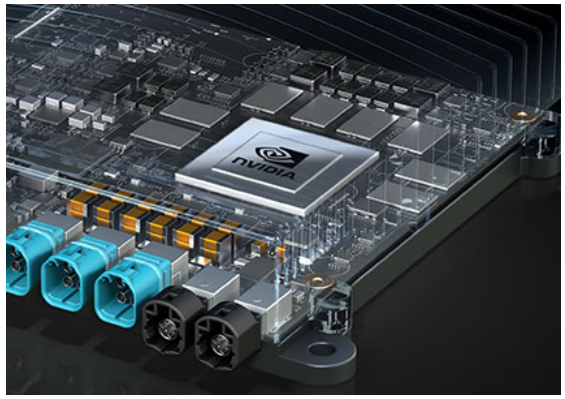
...and automotive industry



https://www.youtube.com/watch?v=mCmO_5ZxdvE

What you have seen

- PilotNet by NVIDIA (regression problem)
  - end-to-end controller for self-driving cars
  - neural network
  - lane keeping and changing
  - trained on data from human driven cars
  - runs on DRIVE PX 2

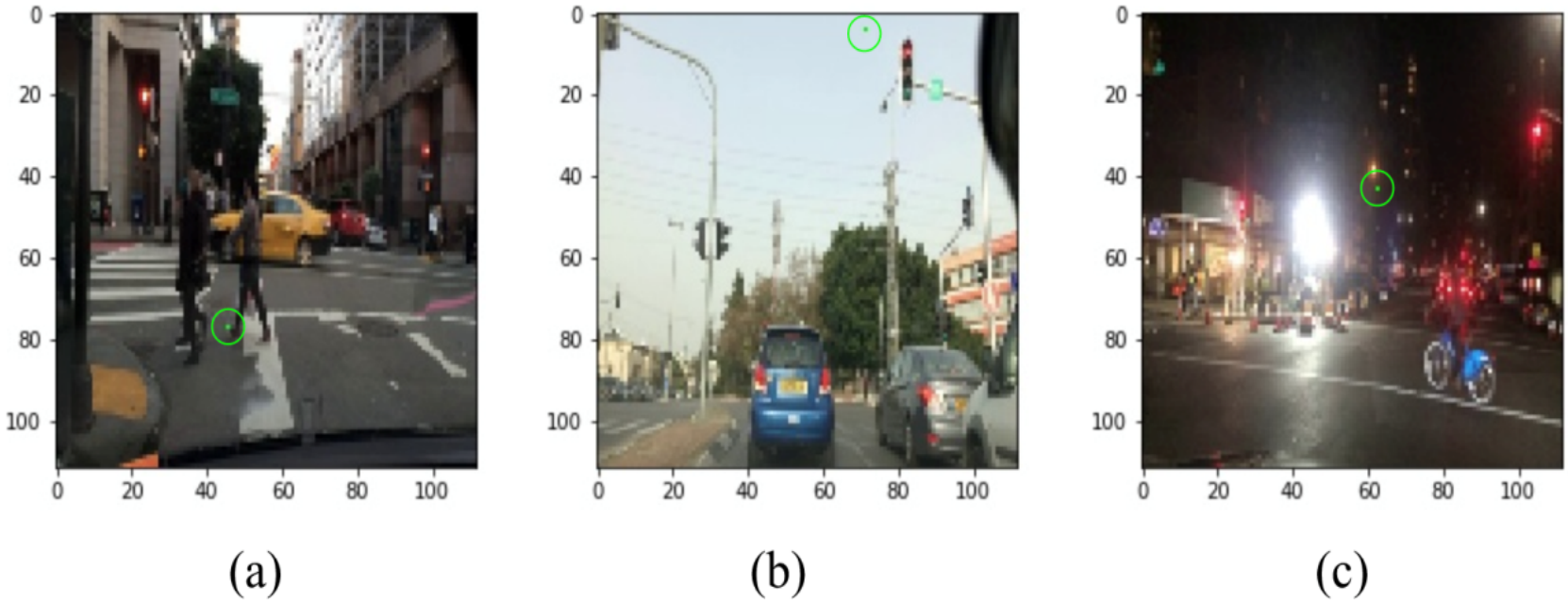


- Traffic sign recognition (classification problem)
  - conventional object recognition
  - neural network solutions already planned…

- BUT
  - neural networks don't come with rigorous guarantees!

PilotNet https://arxiv.org/abs/1604.07316

Should we worry about safety?



Red light classified as green with (a) 68%, (b) 95%, (c) 78% confidence after one pixel change.

– TACAS 2018, https://arxiv.org/abs/1710.07859

Can we verify that such behaviour cannot occur?

Unwelcome news recently…

Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam

Leer en español

By DAISUKE WAKABAYASHI MARCH 19, 2018

Tesla Says Crashed Vehicle Had Been on Autopilot Before Fatal Accident

By GREGORY SCHMIDT MARCH 31, 2018

RELATED COVERAGE

Tesla Looked Like the Fu
Ask if It Has One. MARC

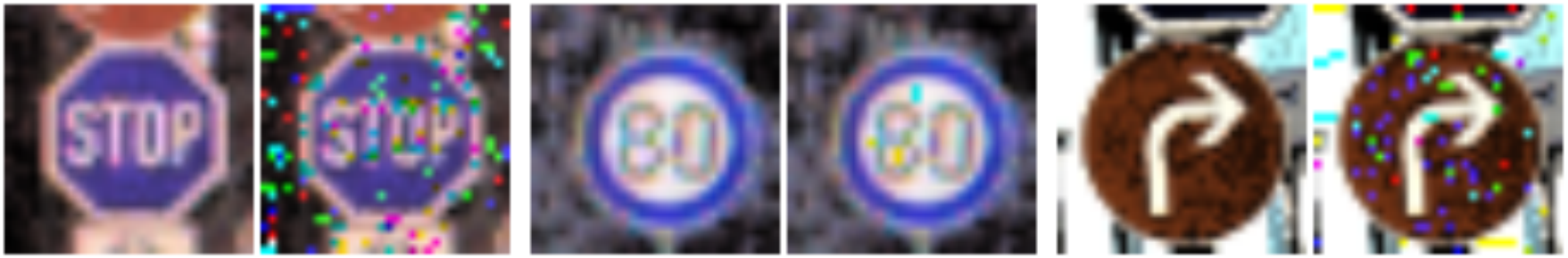
Fatal Tesla Crash Raises New Questions About Autopilot System

U.S. Safety Agency Criticizes Tesla Crash Data Release

How can this happen if we have 99.9% accuracy?

https://www.youtube.com/watch?v=B2pDFjIvrIU

German traffic sign benchmark…



stop	30m speed limit	80m speed limit	30m speed limit	go right	go straight
Confidence	0.999964		0.99		

NB traffic sign detection feature is being introduced in cars

Real traffic signs in Alaska!

Need to consider physical attacks, not only digital…

Physical attacks in the real world

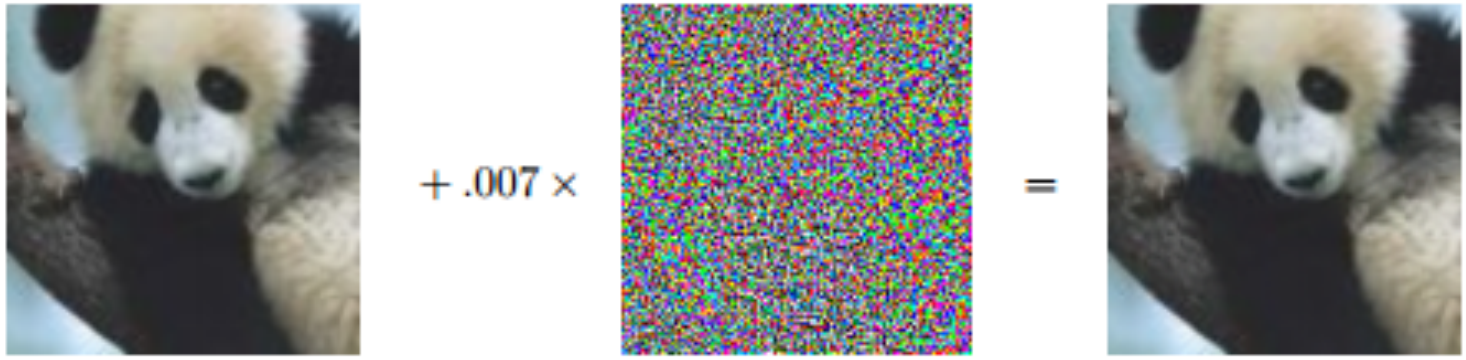
- Resilience testing of Tesla 2016

- (misread as 85mph, sudden acceleration)

Credits: McAfee

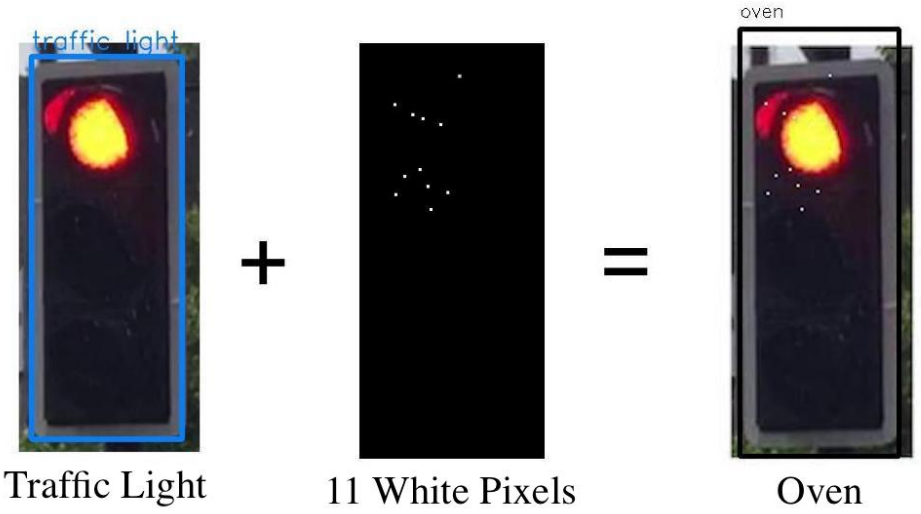
https://www.mcafee.com/blogs/other-blogs/mcafee-labs/model-hacking-adas-to-pave-safer-roads-for-autonomous-vehicles/

Deep neural networks can be fooled!



- They are unstable wrt adversarial perturbations
  - often imperceptible changes to the image [Szegedy et al 2014, Biggio et al 2013 …]
  - sometimes artificial white noise
  - practical attacks, potential security risk
  - transferable between different architectures

Deep neural networks can be fooled!



- Also image segmentation networks, pose recognition, speech recognition, sentiment analysis, …
  - well chosen modifications (manipulations) of pixels or signals

Risk and robustness

- Conventional learning theory
  - empirical risk minimisation [Vapnik 1991]
- Substantial growth in techniques to evaluate robustness
  - variety of robustness measures, different from risk
  - e.g. minimal expected distance to misclassification
- Methods based on optimisation or stochastic search
  - gradient sign method [Szegedy et al 2014]
  - optimisation, tool DeepFool [Moosavi-Desfooli et al 2016]
  - constraint-based, approximate [Bastani et al 2016]
  - adversarial training with cleverhans [Papernot et al 2016]
  - universal adversarial example [Moosavi-Desfooli et al 2017]

This lecture

- Motivation
- Safety and robustness
  - Adversarial examples
  - Maximum safe radius
  - Feature extraction
  - Feature robustness
- Safety verification
  - Reduction to finite search
  - Game-based approximate verification
  - Algorithms for bounding MSR
  - Experimental results
- Current directions

Course materials

- Lab session webpage (includes slides)

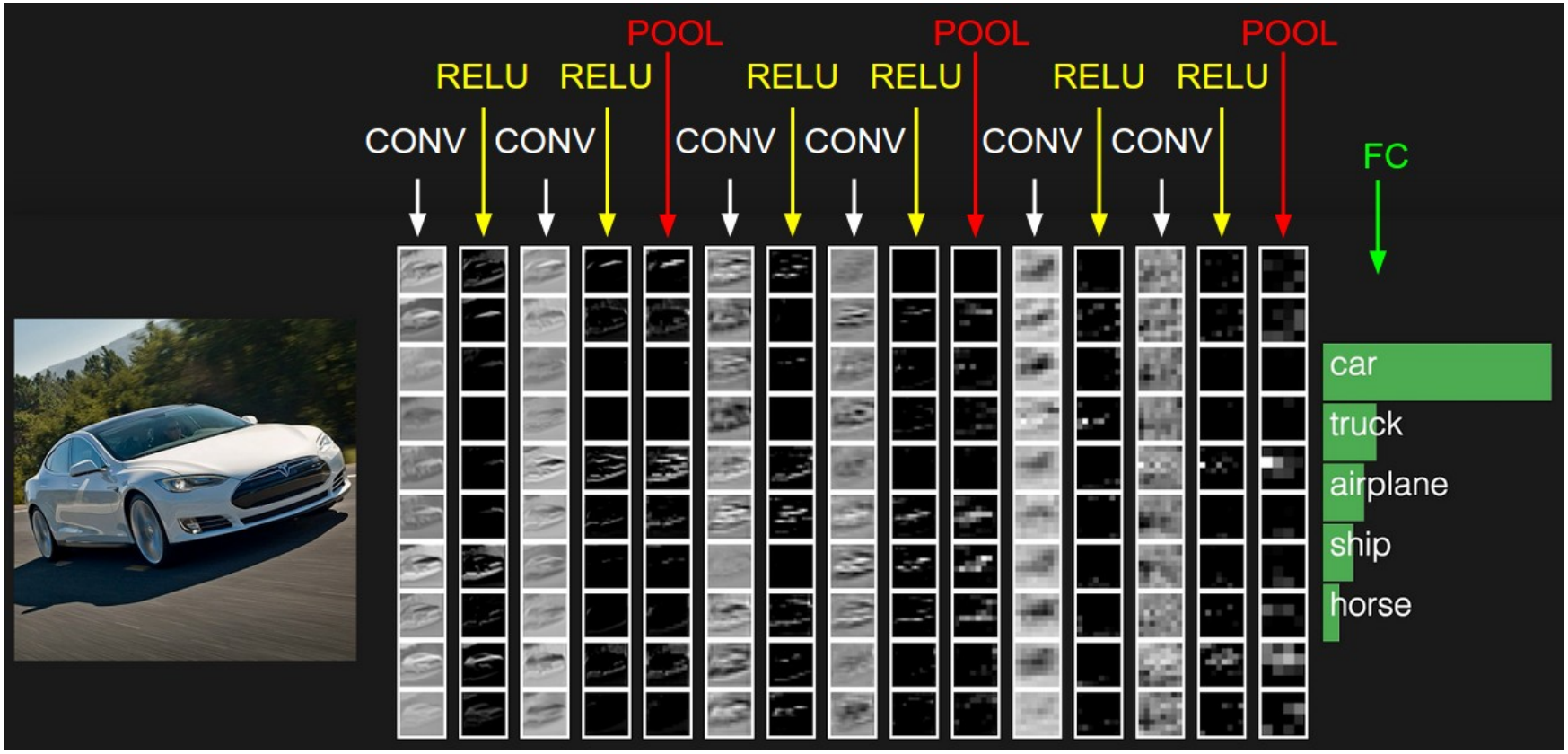
  https://github.com/minwu-cs/DeepGame/blob/master/Oxford-AIMS-CDT-SystemsVerification.md

- Main reference
  - TCS 2019, https://arxiv.org/abs/1807.03571
- Further references for automated verification
  - search/SMT: CAV 2017, https://arxiv.org/abs/1610.06940
  - game: TACAS 2018, https://arxiv.org/abs/1710.07859
  - global optim: IJCAI 2018, https://arxiv.org/abs/1805.02242
  - videos: CVPR 2020, https://arxiv.org/abs/1907.00098
- and for probabilistic safety
  - Bayesian GP: AAAI 2019, https://arxiv.org/abs/1809.06452
    AISTATS 2020, https://arxiv.org/abs/1905.11876
  - Bayesian NN: IJCAI 2019, https://arxiv.org/abs/1903.01980
    ICRA 2020, https://arxiv.org/abs/1909.09884

Course schedule

- Monday Marta
  - Safety and robustness
  - Lecture am, Lab pm (Min)
- Tuesday Daniel (note lectures starts 11:05)
  - GANs
  - Lecture am, Lab pm (Isaac)
- Wednesday Daniel
  - Explainability
  - Lecture am, Lecture/demo pm (Alessandro)
- Thursday Alessandro
  - Lecture/demo am, Lab (Isaac+Min catch up)
- Friday free
- Assessment: Labs

Deep feed-forward neural network



Convolutional multi-layer network

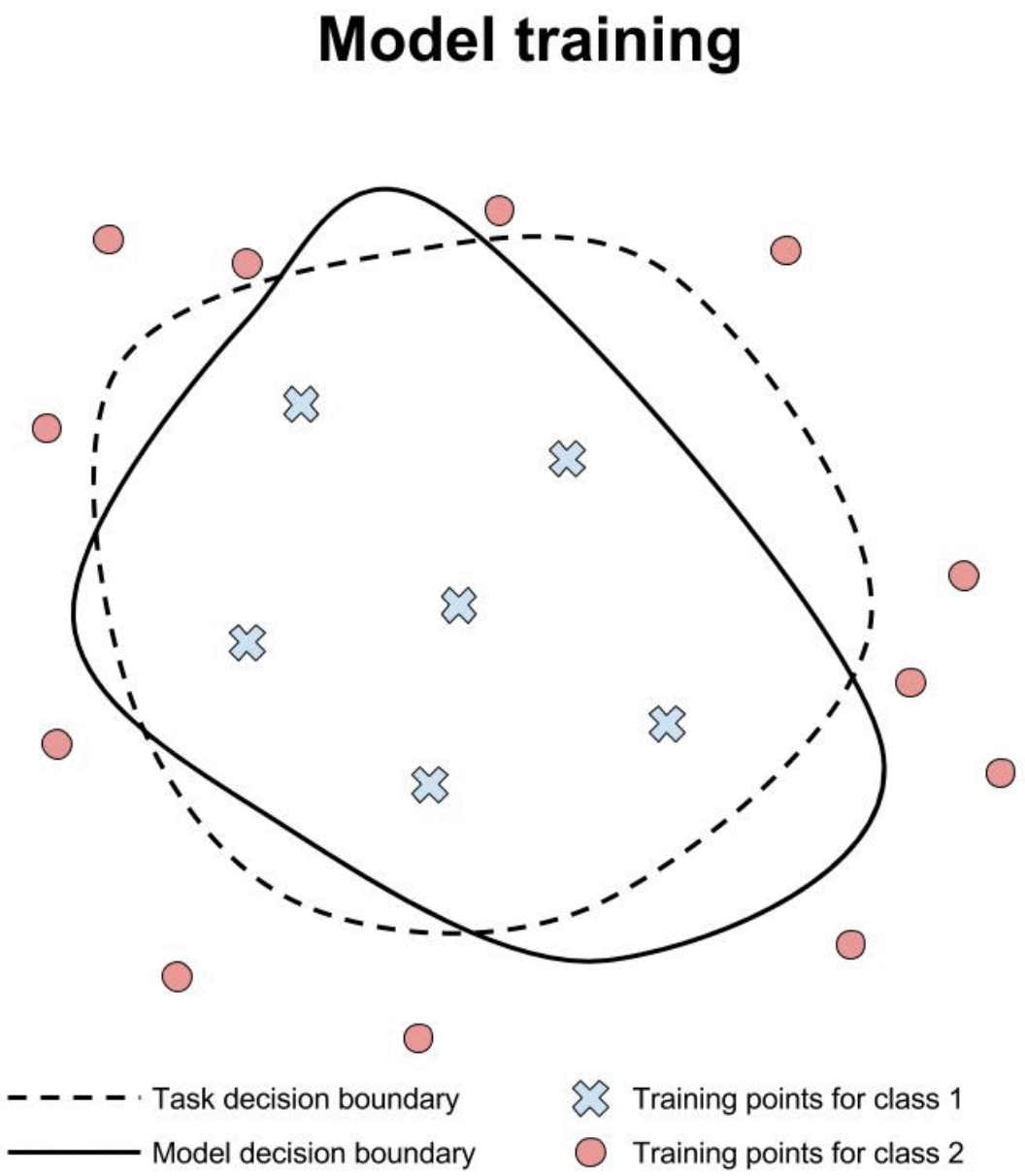
http://cs231n.github.io/convolutional-networks/#conv

Problem setting

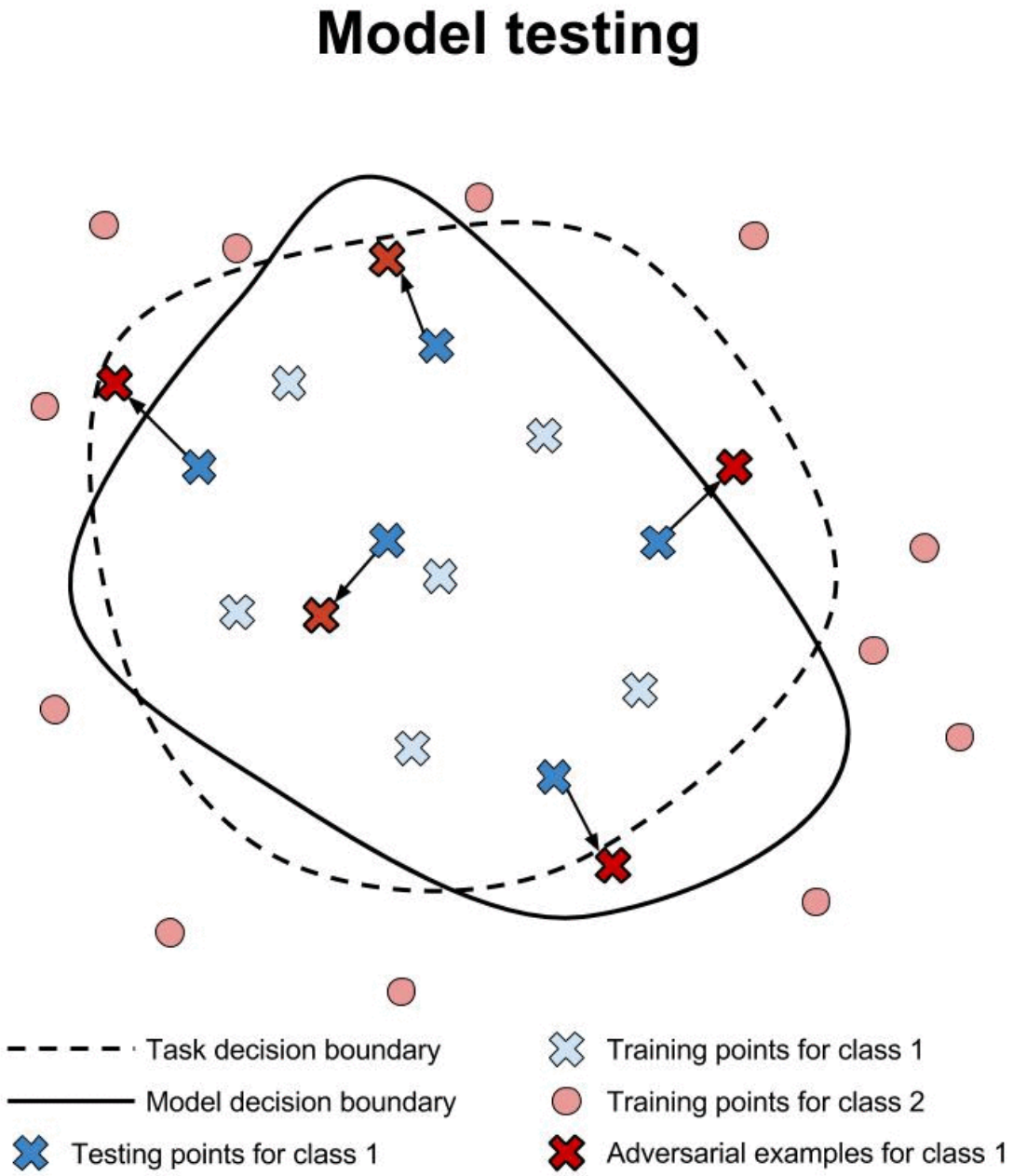
- Assume
  - vector spaces D, D_{L1}, …, D_{Ln}, one for each layer
  - D normalised to lie in $[0,1]^n$
  - $f : D_{L0} \rightarrow \{c_1,\ldots c_k\}$ classifier modelling human perception ability
- The network $N : D_{L0} \rightarrow C=\{c_1,\ldots c_k\}$ approximates f from M training examples $\{(x_i,c_i)\}_{i=1..M}$
  - built from activation functions ϕ_0, ϕ_1, …, ϕ_n, one for each layer
  - for point (image) $\alpha \in D$, its activation in layer k is

    $$a_{x,k} = \phi_k(\phi_{k-1}(\ldots\phi_1(x)))$$
  - where $\phi_k(x) = \sigma(xW_k+b_k)$ and $\sigma(x) = \max(x,0)$
  - W_k learnable weights, b_k bias, σ ReLU
- Notation
  - $N_c(\alpha)$ – confidence that α belongs to class c
  - $N(\alpha) = \arg\max_{c\in C} N_c(\alpha)$

Training vs testing



Training vs testing



Robustness

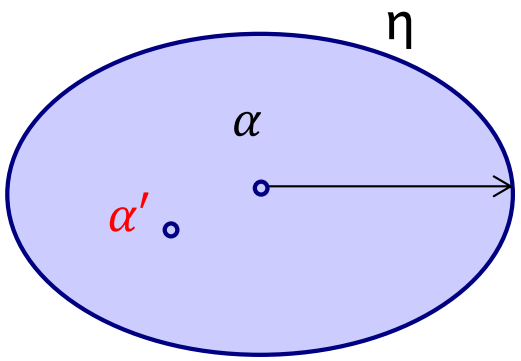
- Regularisation such as dropout improves smoothness
- Common smoothness assumption
  - each point $\alpha \in D$ in the input layer has a region η around it such that all points in η classify the same as α
- Pointwise (local) robustness [Szegedy et al 2014]
  - N is not robust at point α if $\exists\ \alpha' \in \eta$ such that $N(\alpha) \neq N(\alpha')$
- Robustness (network property)
  - smallest perturbation weighted by input distribution
  - reduced to non-convex optimisation problem

Distance metrics and neighbourhood

- Assume D normalised to lie in $[0,1]^n$
  - For $\alpha \in D$, $\alpha[i]$ is a dimension (pixel)
- Work with L_p norm $\sqrt[p]{\sum_{i=1}^{n} |\alpha[i]|^p}$, e.g. L_0 , L_1, L_2, L_∞
  - i.e. $\| \alpha - \alpha' \|_1 = \sum_{i=1}^{n} |\alpha[i] - \alpha'[i]|$
- Define ε–ball (neighbourhood)
  - $\eta(\alpha, L_p, \varepsilon) = \{\alpha' \mid \| \alpha - \alpha' \|_p \leq \varepsilon\}$
- Define $N : D \rightarrow \{c_1, \ldots c_k\}$ Lipschitz continuous wrt L_p iff there exists constant h_c for each class c s.t for all α, α' we have
  - $| N_c(\alpha) - N_c(\alpha') | \leq h_c \| \alpha - \alpha' \|_p$
- All networks with bounded inputs are Lipschitz

Safety of classification decisions

- Safety assurance process is complex
- Here focus on safety at a point as part of such a process
  - same as pointwise robustness…

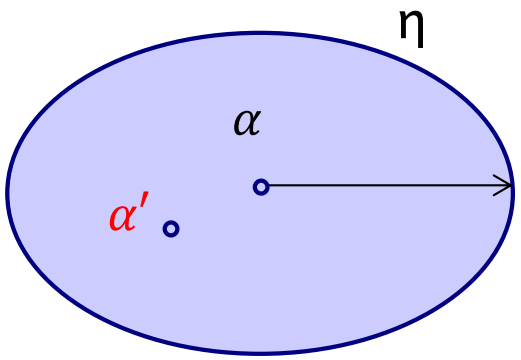


- Assume given
  - trained network N : D → $\{c_1,\ldots c_k\}$
  - diameter for support region η
  - norm, e.g. L^2, L^∞

- Define safety as invariance of classification decision
  - i.e. $\nexists \alpha' \in \eta$ such that $N(\alpha) \neq N(\alpha')$
- Also wrt family of safe manipulations
  - e.g. scratches, weather conditions, camera angle, etc

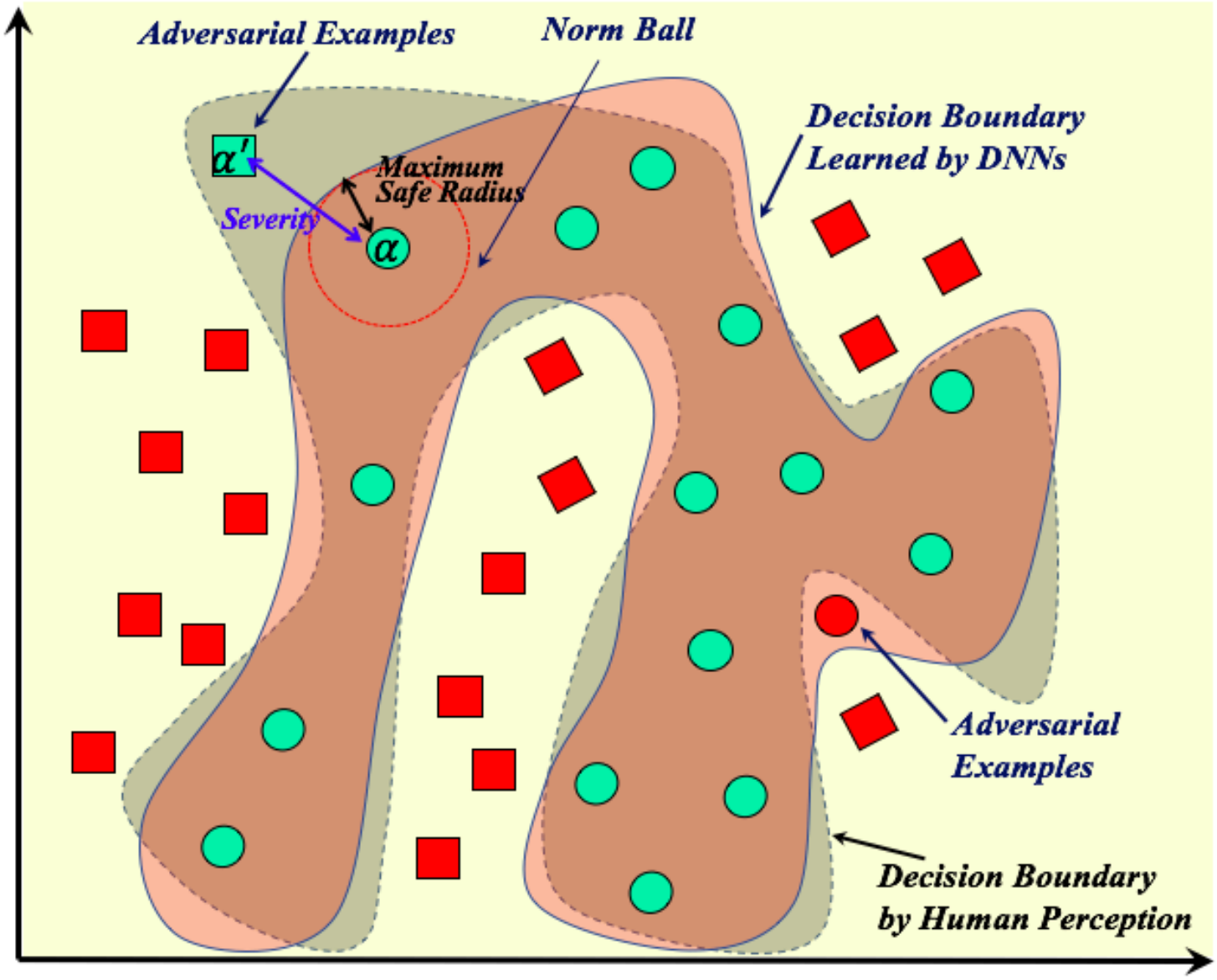
Adversarial examples

- Fix distance metric L_p
- Assume given
  - input α
  - support region η



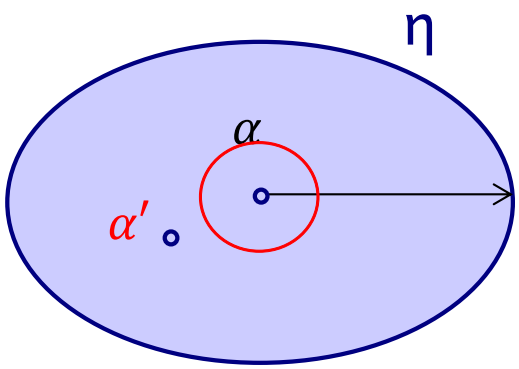
- Define untargeted adversarial examples at a point α
  - $\text{adv}(\alpha, \varepsilon) = \{\alpha' \in \eta(\alpha, L_p, \varepsilon) \mid N(\alpha) \neq N(\alpha')\}$
- Targeted adversarial examples for specific class $c = N(\alpha')$
- Can also consider wrt class of manipulations

Adversarial examples



Maximum safe radius

- Define maximum safe radius (MSR)
  - $\text{MSR}(\alpha) = \inf \{\varepsilon > 0 \mid \text{adv}(\alpha, \varepsilon) \neq \emptyset\}$
- i.e. the minimum distance from α to adversarial example



- Intuitively,
  - If $\text{adv}(\alpha, \varepsilon_0) \neq \emptyset$ then $\text{MSR}(\alpha) \leq \varepsilon_0$ so finding an adversarial example gives an upper bound on MSR
  - Conversely, if $\text{adv}(\alpha, \varepsilon) = \emptyset$ for all $0 \leq \varepsilon \leq \varepsilon_0$ then $\varepsilon_0 \leq \text{MSR}(\alpha)$ so ruling out adversarial examples gives a lower bound on MSR
- But $\eta(\alpha, L_p, \varepsilon)$ is infinite!

Safety verification

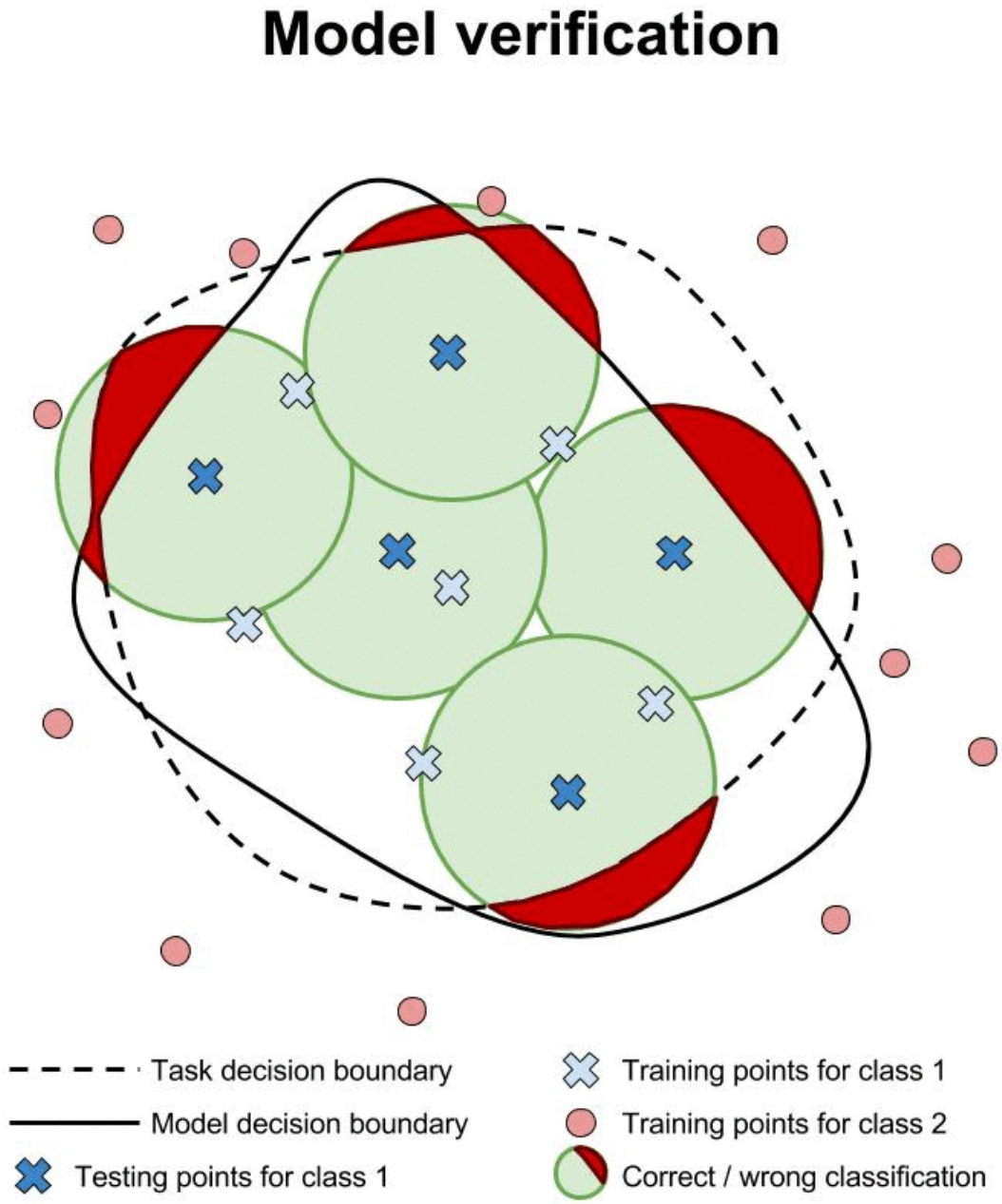
- Take as a specification set of manipulations and region η
  - work with pointwise robustness as a safety criterion
  - focus on safety wrt a set of manipulations
  - exhaustively search the region for misclassifications

- Challenges
  - high dimensionality, nonlinearity, infinite region, huge scale

- Automated verification (= ruling out adversarial examples)
  - need to ensure finiteness of search
  - guarantee of decision safety if adversarial example not found

- Falsification (= searching for adversarial examples)
  - good for attacks, no safety guarantees

Training vs testing vs verification



Searching for adversarial examples…

- Input space for most neural networks is high dimensional and non-linear
- Where do we start?
- How can we apply structure to the problem?

- Image of a tree has 4,000 x 2,000 x 3 dimensions = 24,000,000 dimensions
- We would like to find a very ‘small’ change to these dimensions

Feature-based exploration

- Searching by trying every combination of pixel values is intractable
- We can 'reduce' the dimensionality of an image by reducing it only to its salient features

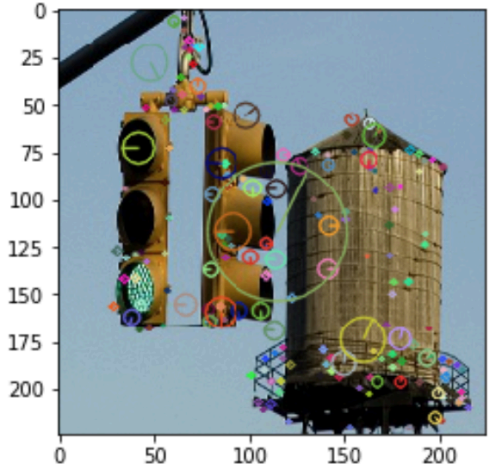
$\Lambda(\alpha)$ – Set of features given an image

λ_r – Response strength of the feature (roughly how 'important' it is)

λ_x – X coordinate of a keypoint

λ_y – Y coordinate of a keypoint

λ_s – Radius of a keypoint

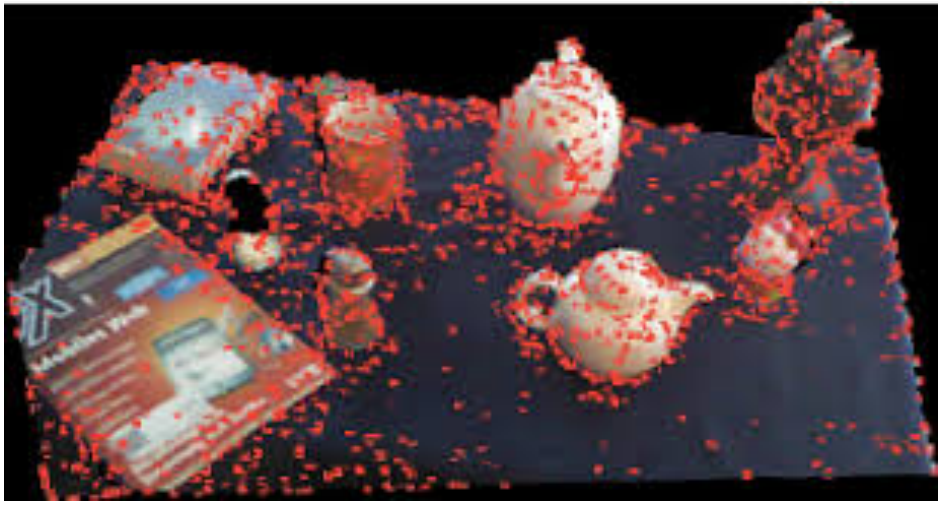


Feature extraction algorithms (SIFT)

- (1) Scale space extrema detection

We blur the image in order to detect extrema of different sizes

- (2) Keypoint localization and description

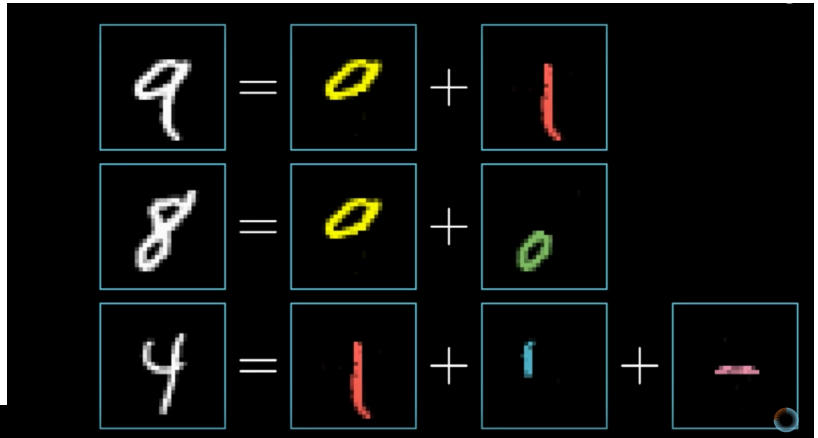


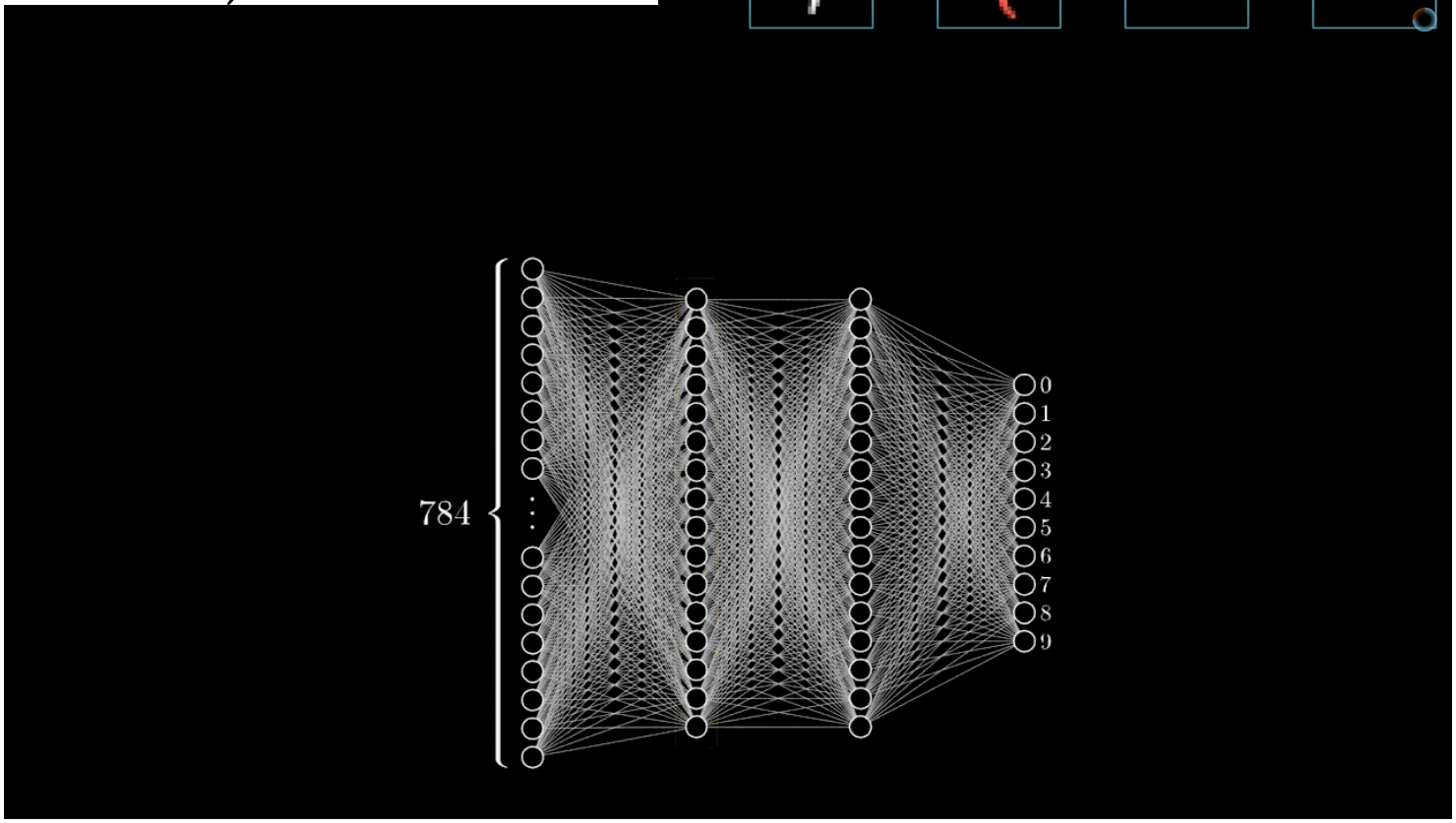
Localization looks at the gradients from the scale space to describe each keypoint

SIFT is invariant to scale, rotation and translation

Intuition for feature-based exploration

- Known fact: neural networks are executing feature extraction under the hood…
- (3blue1brown animation by Grant Sanderson)

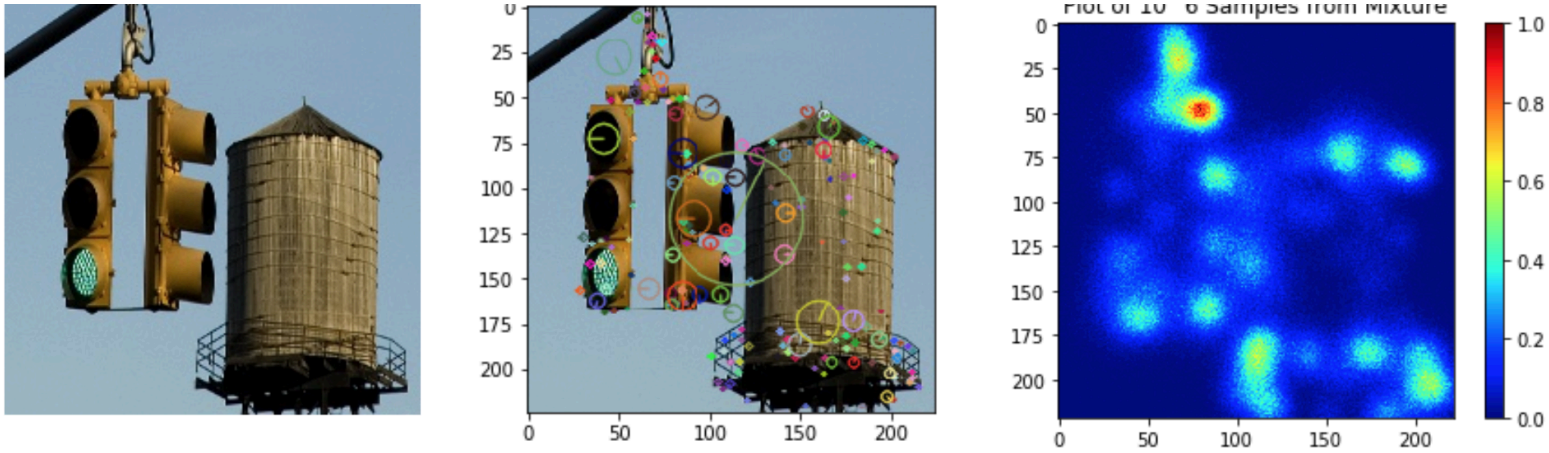




Feature-based representation

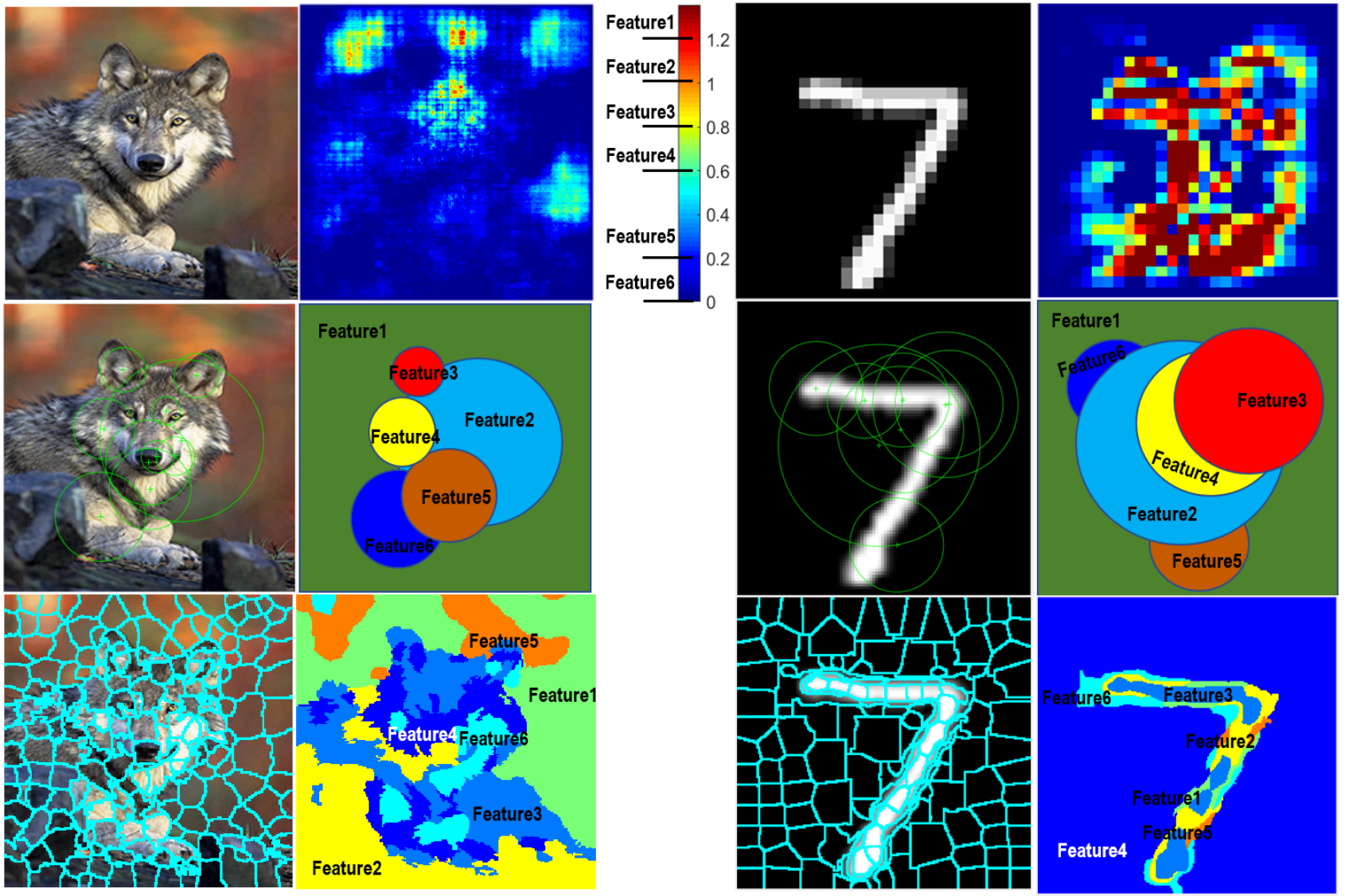
- Employ the SIFT algorithm to extract features
- Reduce dimensionality by focusing on salient features
- Use a Gaussian mixture model in order to assign each pixel a probability based on its perceived saliency

$$\mathcal{G}_{i,x} = \frac{1}{\sqrt{2\pi\lambda_{i,s}^2}} exp\left(\frac{-(p_x - \lambda_{i,x})^2}{2\lambda_{i,s}^2}\right) \quad \mathcal{G}_{i,y} = \frac{1}{\sqrt{2\pi\lambda_{i,s}^2}} exp\left(\frac{-(p_y - \lambda_{i,y})^2}{2\lambda_{i,s}^2}\right)$$



TACAS 2018, https://arxiv.org/abs/1710.07859

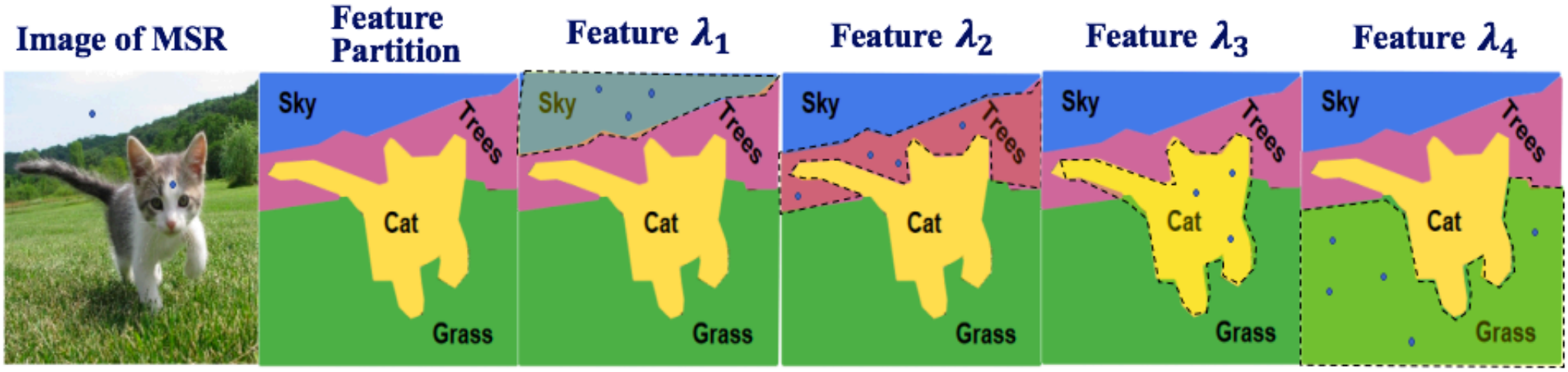
Examples of feature extraction methods



From top: saliency map, SIFT, K-means clustering

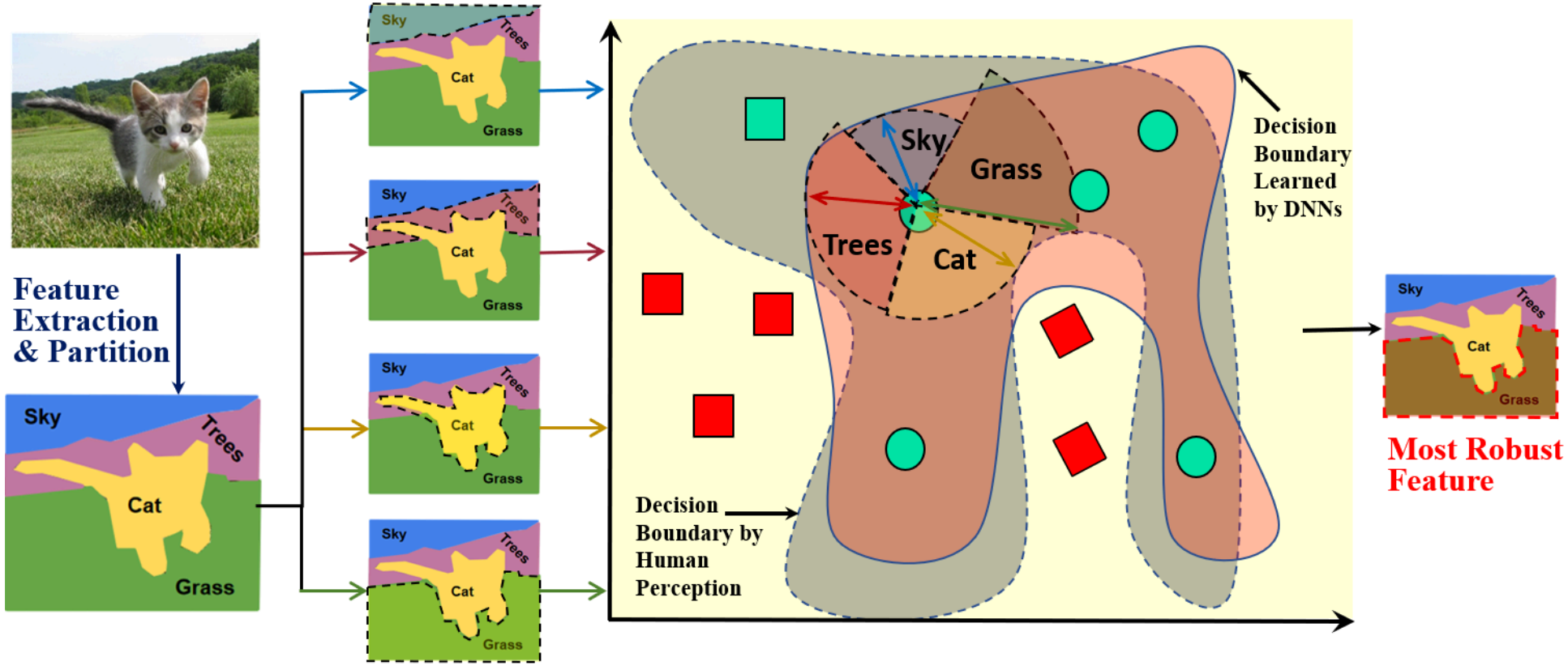
Feature partitioning

- Assume disjoint features



- Reduction in dimensionality

Feature robustness problem (FR)



Identifies the feature that is most robust to perturbations

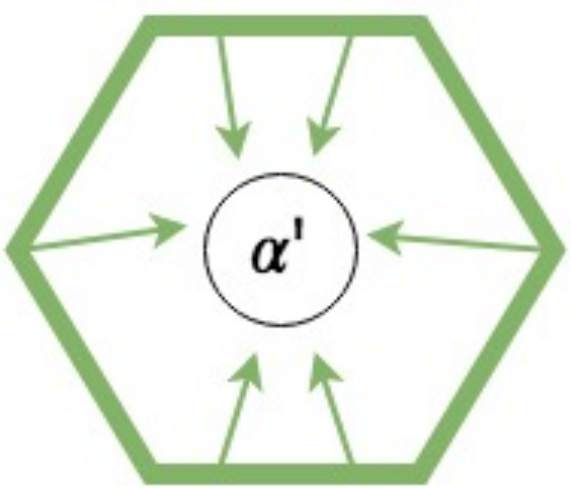
(see main reference for more info)

Lecture part 2

- Motivation
- Safety and pointwise robustness
  - Adversarial examples
  - Maximum safe radius
  - Feature extraction
  - Feature robustness
- Safety verification
  - Reduction to finite search
  - Game-based approximate verification
  - Algorithms for bounding MSR
  - Experimental results
- Current research directions

Lipschitz networks

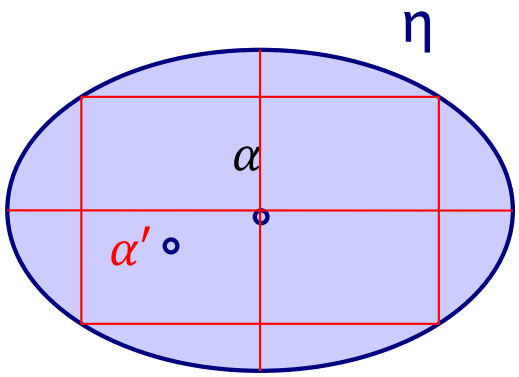
- Lipschitz continuity limits the rate of change of output
- For Lipschitz networks, there exists a diameter such that every image within it shares the classification of a given input



- Use this fact to provide safety guarantees – suffices to inspect the corners of the region

Safety verification

- Recall that $\eta(\alpha, L_p, \varepsilon)$ is infinite in continuous spaces, and thus cannot be exhaustively searched
- Idea: reduce the computation of MSR to finite optimisation
  - finite 'grid' of points in the input space
  - rely on Lipschitz constant to bound the output range within cells
  - then suffices to just check the grid points

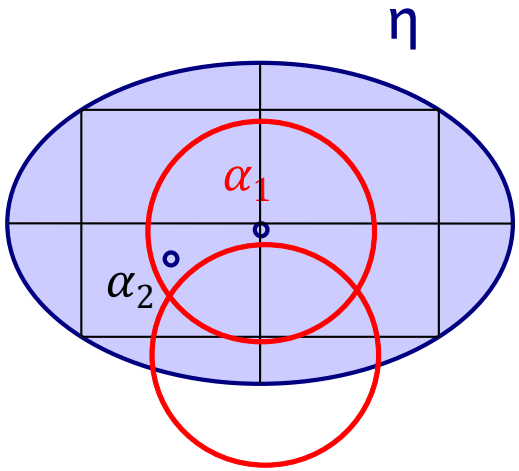


- Since estimating Lipschitz constant difficult,
  - assume existence of a (not necessarily tight) Lipschitz constant
  - determine grid step size in terms of manipulation magnitude $\tau \in (0,1]$, noting that this is proportional to 'confidence gap' $\text{conf}(\alpha', \text{N}(\alpha'))$, where

$$\text{conf}(\alpha, c) = \min\{c' \in C, c \neq c'\} \left(\text{N}_c(\alpha) - \text{N}_{c'}(\alpha)\right)$$

Safety verification

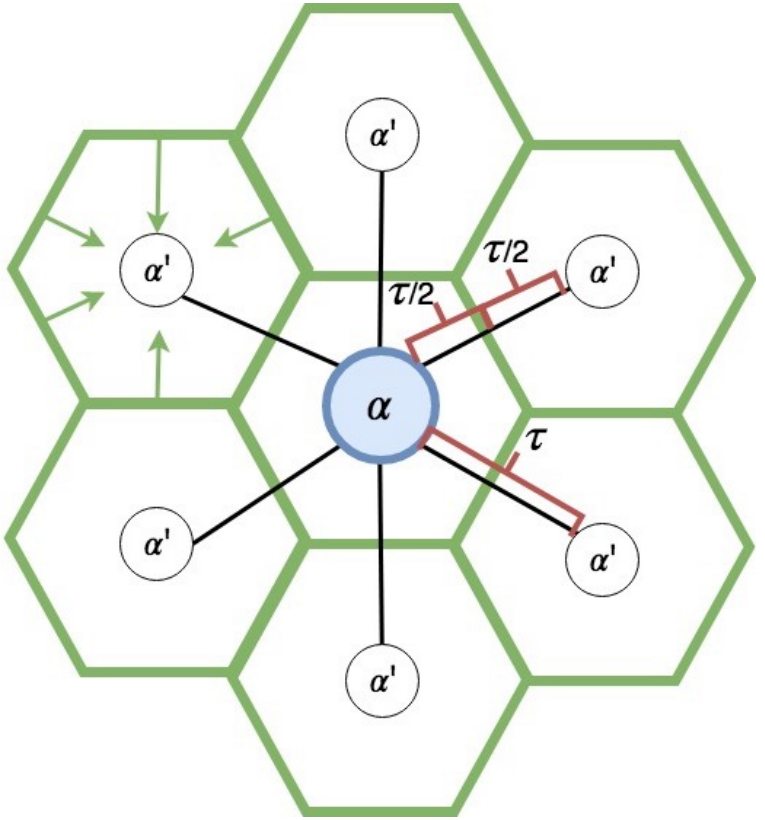
- Define point $\alpha_1 \in \eta(\alpha, L_p, \varepsilon)$ a misclassification aggregator for radius δ if for any $\alpha_2 \in \eta(\alpha_1, L_p, \delta)$ we have $N(\alpha_2) \neq N(\alpha)$ implies $N(\alpha_1) \neq N(\alpha)$
  - i.e. classification guarantee within radius δ
- Define grid of points centred on α with step $\tau \in (0,1]$ (points near α that can be reached by τ pixel changes)



- Cover the region with balls with radius $\frac{1}{2}\tau$ centred at each point
- Choose τ sufficiently small (details omitted) so that all points in $\eta(\alpha, L_p, \varepsilon)$ are misclassification aggregators (inversely proportional to Lipschitz constant and proportional to 'confidence gap')

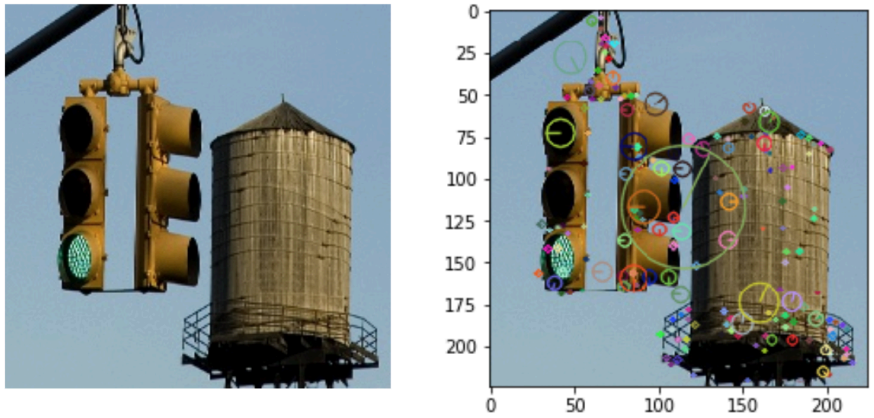
Safety guarantees

- Cover the region with a 'grid' of diameter $\frac{1}{2}\tau$ (half of manipulation size τ)
- Reduces MSR computation to finite grid search
- Finding an adversarial example gives upper bound
- If an adversarial example not found then we obtain a lower bound
- Maximum error for MSR is $\frac{1}{2}\tau$
- Use Monte Carlo Tree Search (MCTS) for upper bounds, and A* search for lower bounds



Game-based search

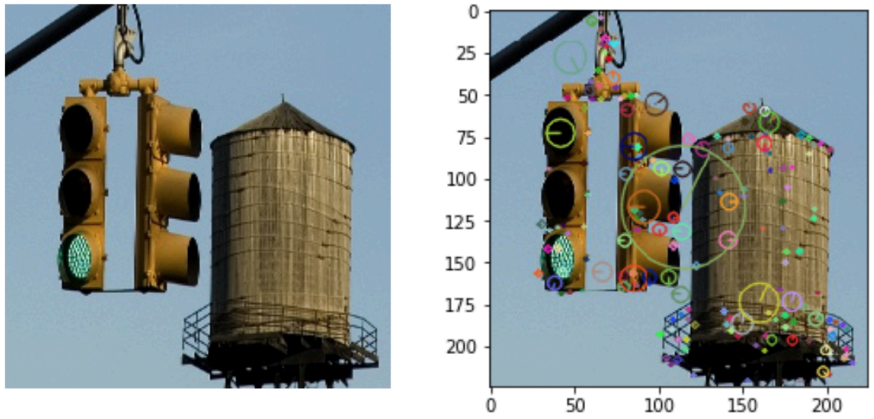
- Goal is finding adv. example, reward inverse of distance (severity)
- Player 1 selects the feature to manipulate



- Each feature represents a possible move for player 1
- Player 2 then selects the pixels in the feature to manipulate by $+/-\tau$
- Method black/grey box, can approximate the maximum safe radius for a given input

Game-based search

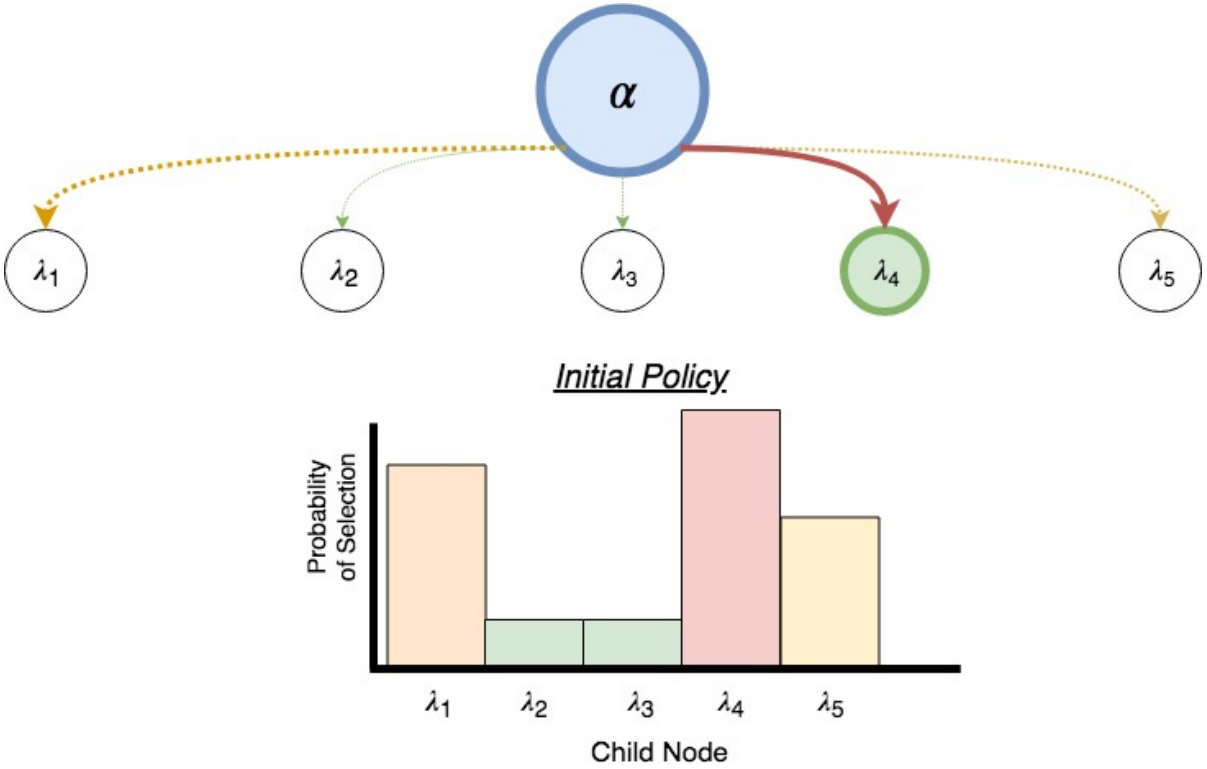
- Initial state is input image α
- Player 2 objective is to minimize the distance to adv. example, and Player 1 is cooperative or competitive
- Optimal strategies yield MSR, giving provable guarantees



- Terminates when adversarial example found, or perturbation outside $\eta(\alpha, L_p, \varepsilon)$, or time budget exceeded
- Employ search to explore the (finite) game tree

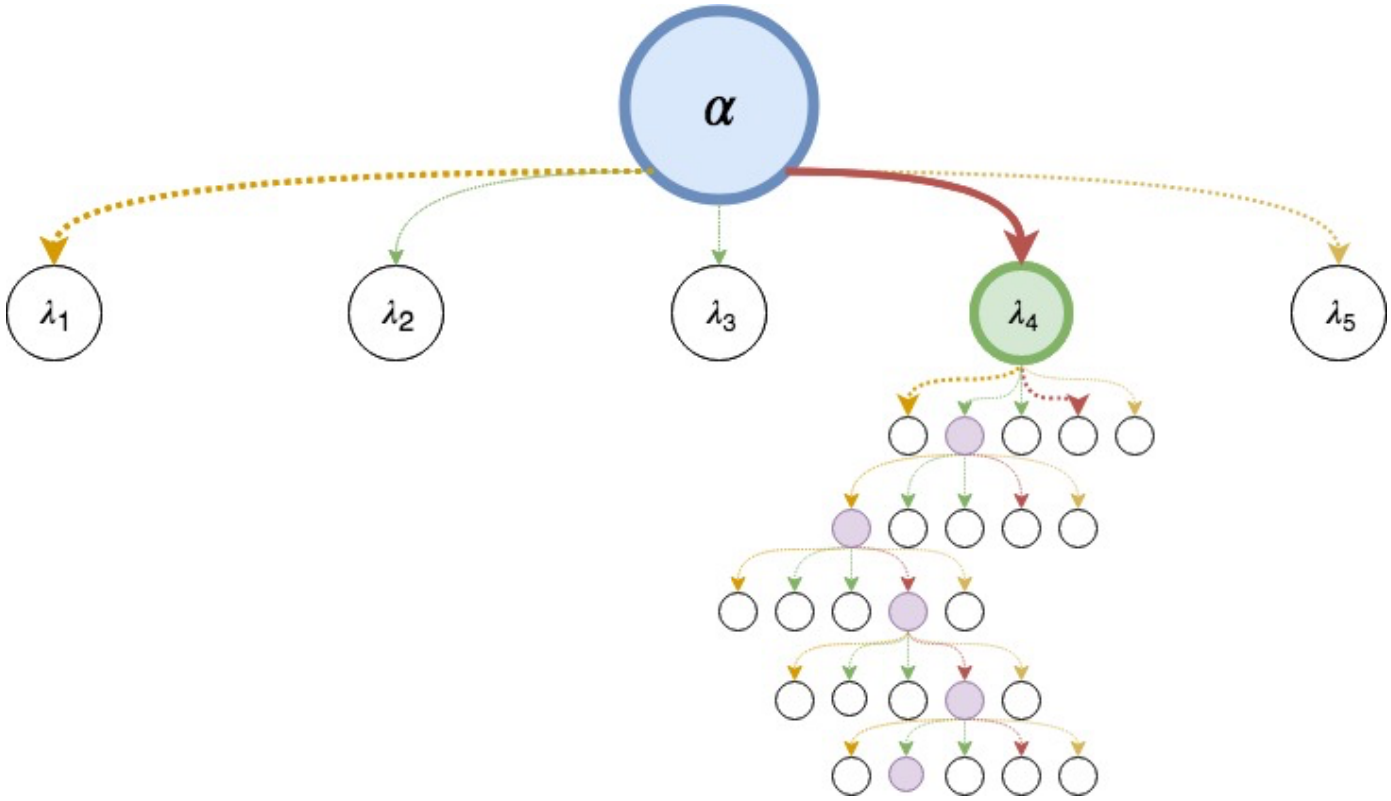
MCTS: selection/expansion

- The root of the tree represents the original image, and each child represents a potential manipulated image
- First, select a manipulation based on each player's strategy
- If the child has never been selected from previously then we "expand" the tree to select a new leaf.



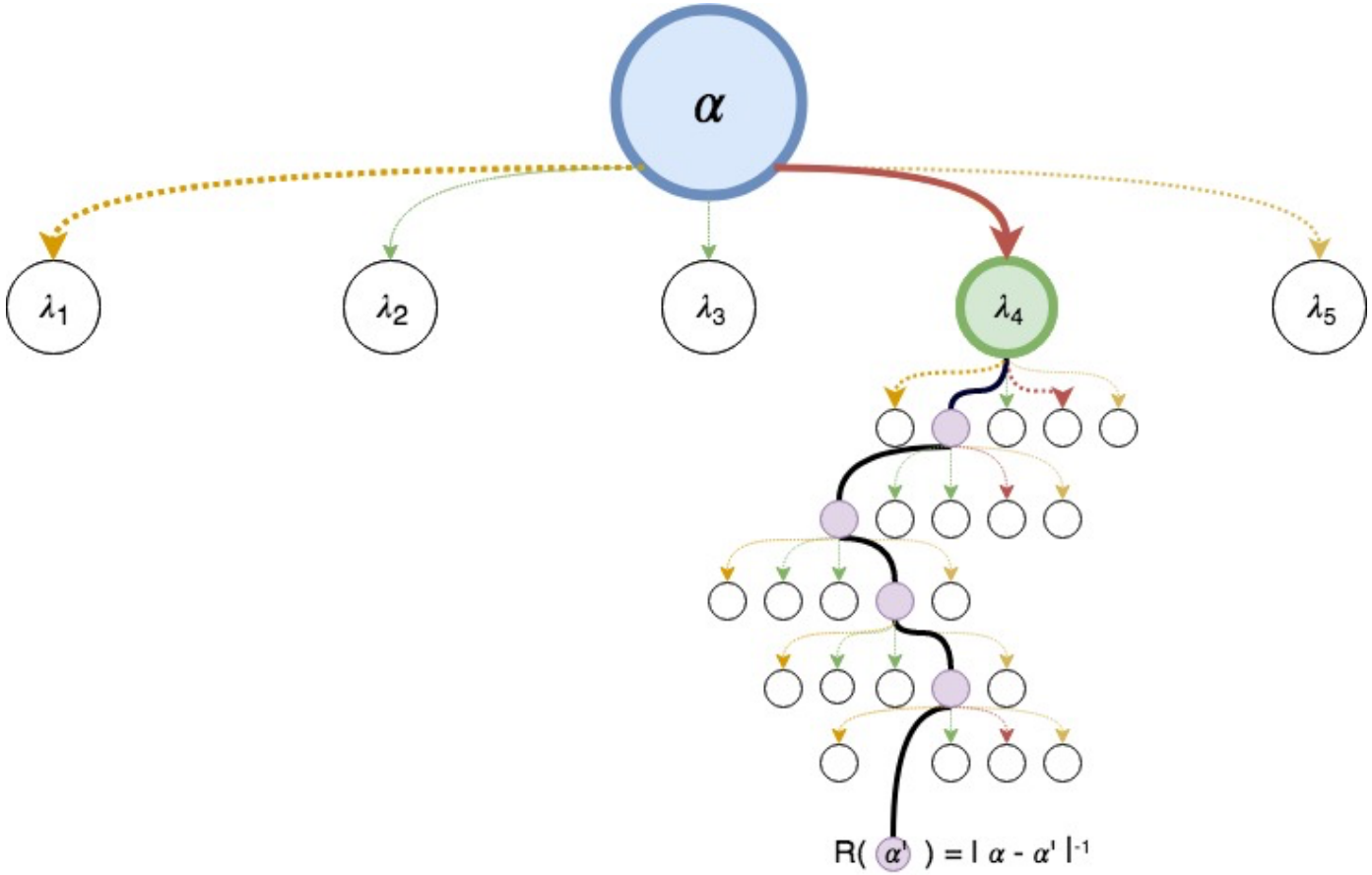
MCTS: simulation

- After a new child has been added to the tree, we approximate the reward of visiting this child by continuously searching the tree until we have either timed out or hit an adversarial example
- These nodes are not recorded as a part of the partial tree

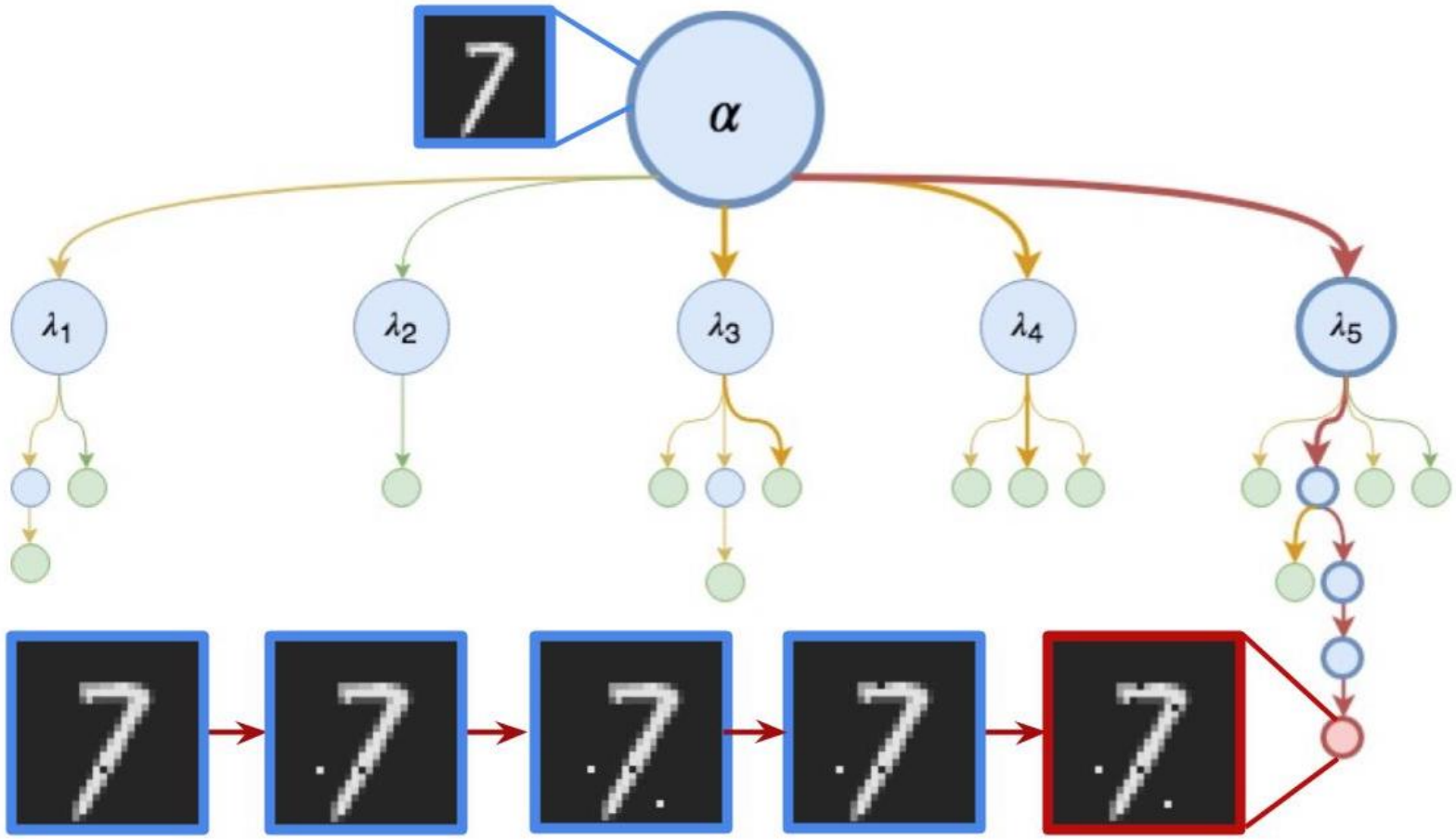


MCTS: backpropagation

- After we have terminated the tree, we calculate the reward, and backpropagate that reward up the tree to update our exploration policy (update each player's strategies)

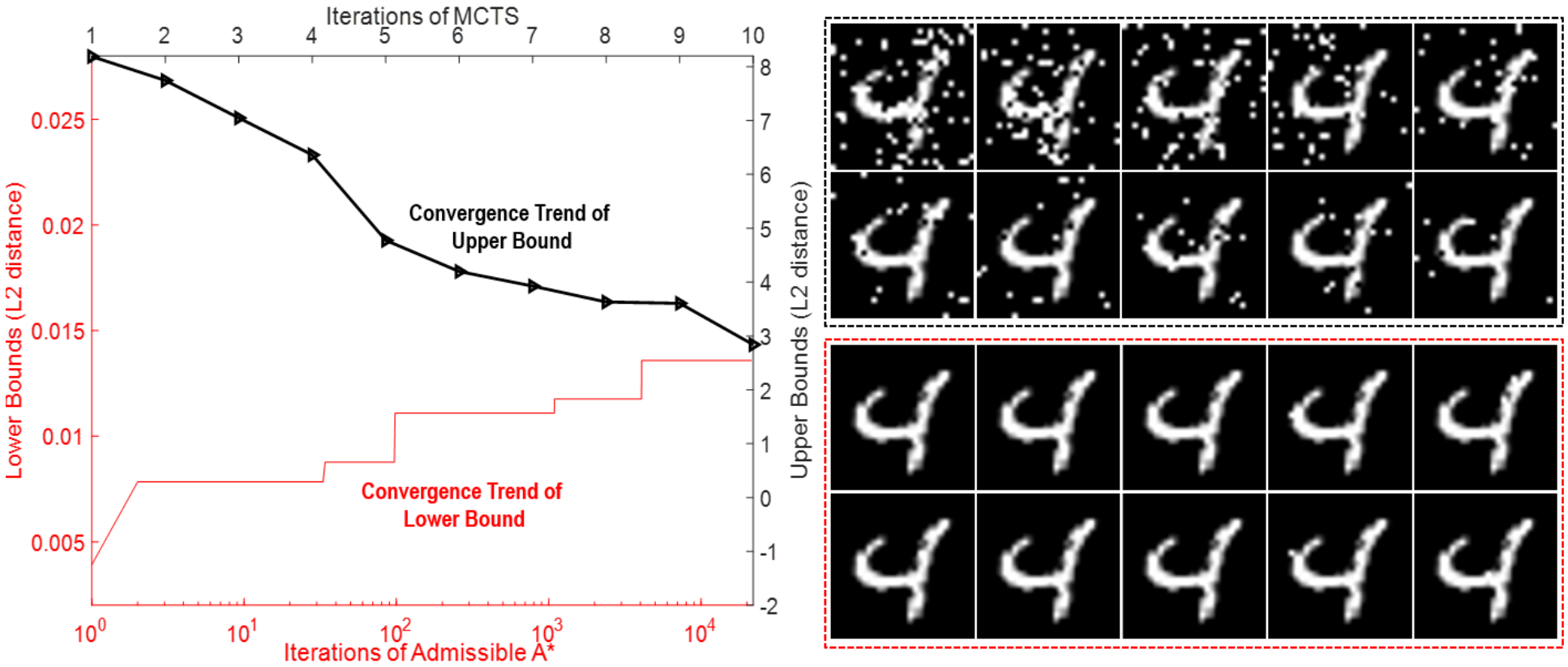


Tree expands until example is found

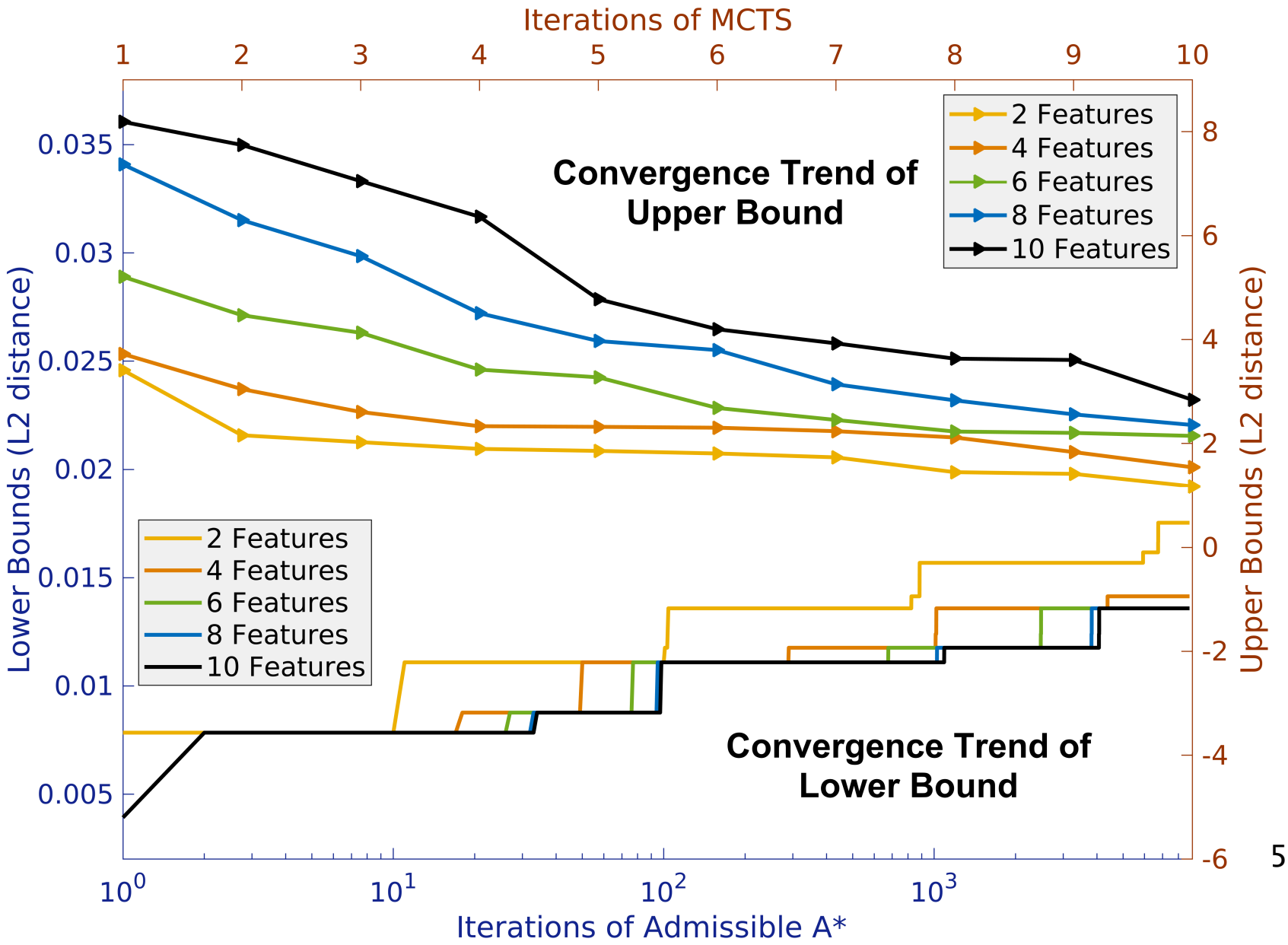


Now also lower bounds (MNIST)

- Convergence of lower and upper bounds on maximum safe radius

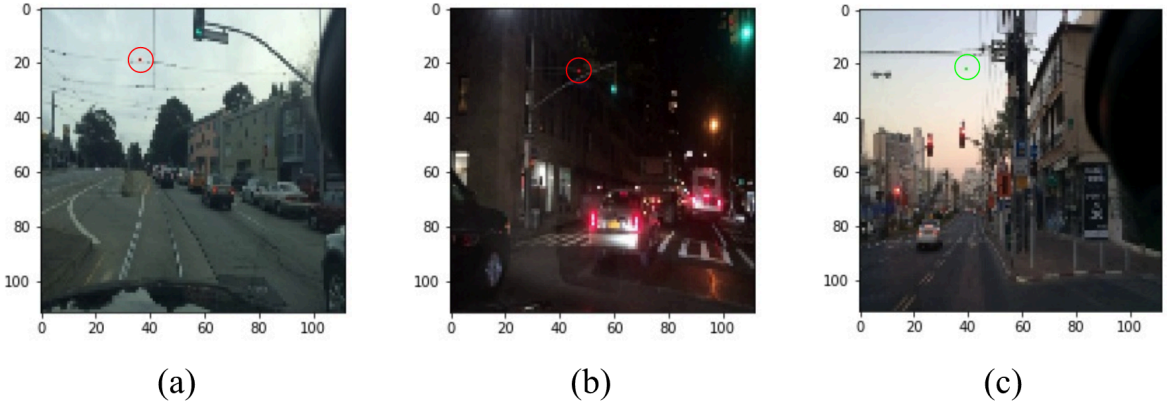


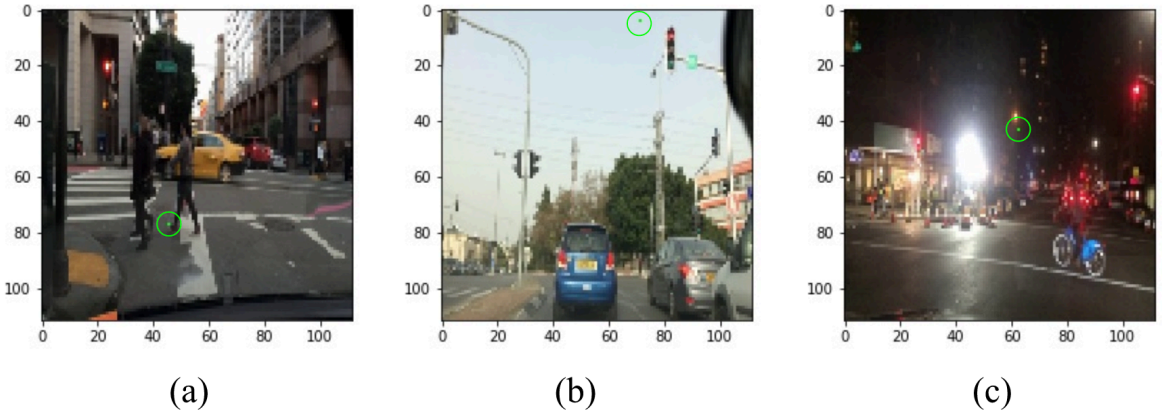
Convergence trend for FR



Evaluating safety-critical scenarios: Nexar

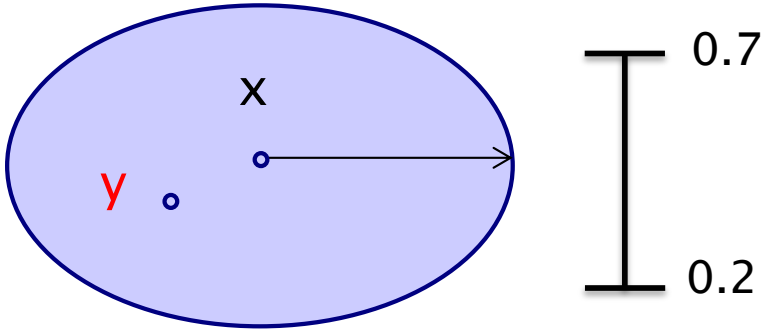
- Using our Game-based Monte Carlo Tree Search method we were able to reduce the accuracy of the network form 95% to 0%
- On average, each input took less than a second to manipulate (.304 seconds)
- On average each image was vulnerable to 3 pixel changes





Alternative approach: reachability analysis

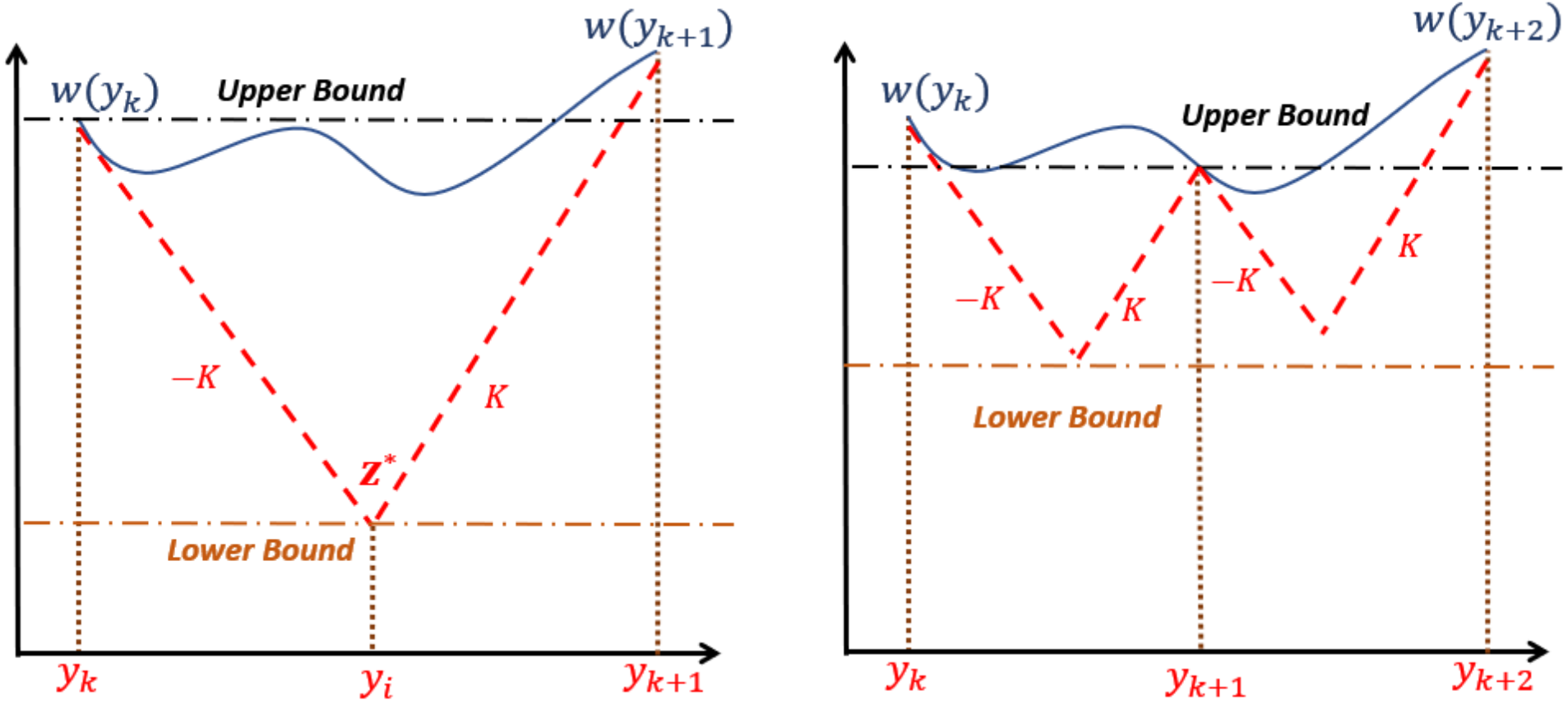
- Rather than search the discretized region, can we compute the reachable values?
- Under assumption of Lipschitz continuity
  - for $x \in \eta$, compute maximum/minimum value of $f(\eta)$
  - using global optimisation
  - anytime fashion
- Gives provable guarantees
  - best/worst case confidence values
  - pointwise confidence diameter
  - can average over input distribution



- Method NP-complete
  - wrt the number of input dimensions, not number of neurons

- IJCAI 2018, https://arxiv.org/abs/1805.02242

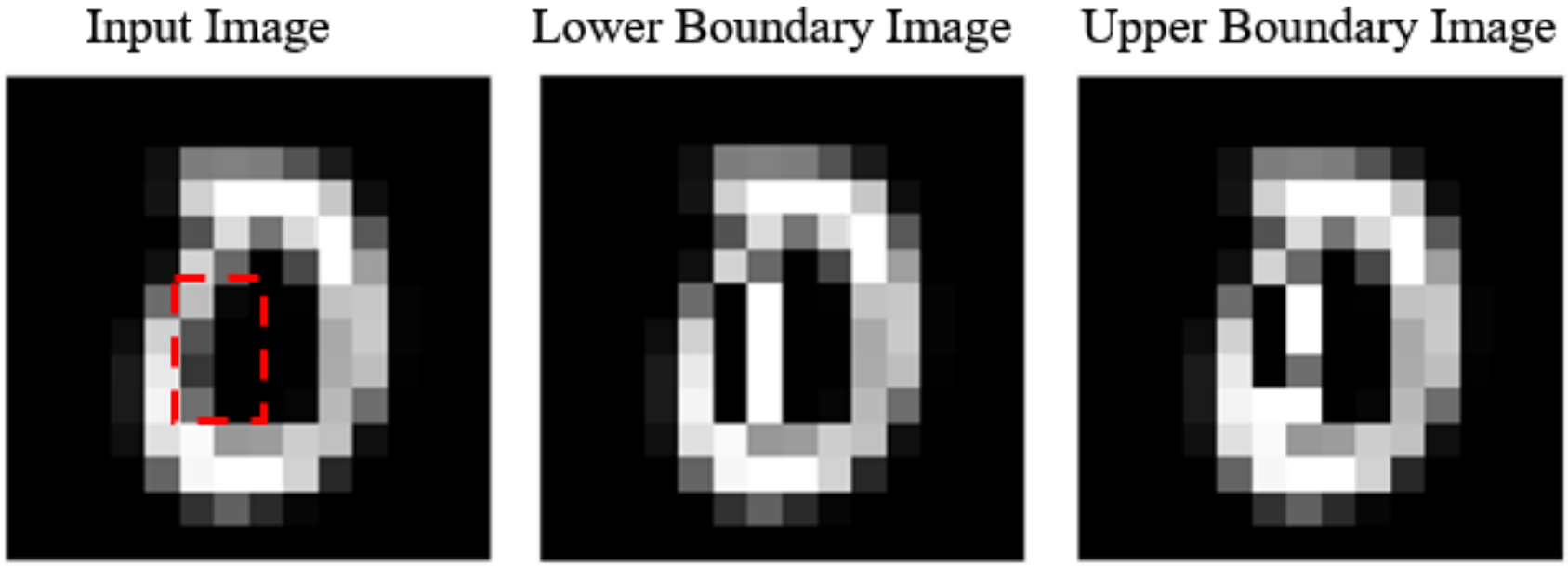
Global optimization: main idea



- Adaptive nested optimization, asymptotic convergence
  - construct a series of lower and upper bounds
- K – Lipschitz constant

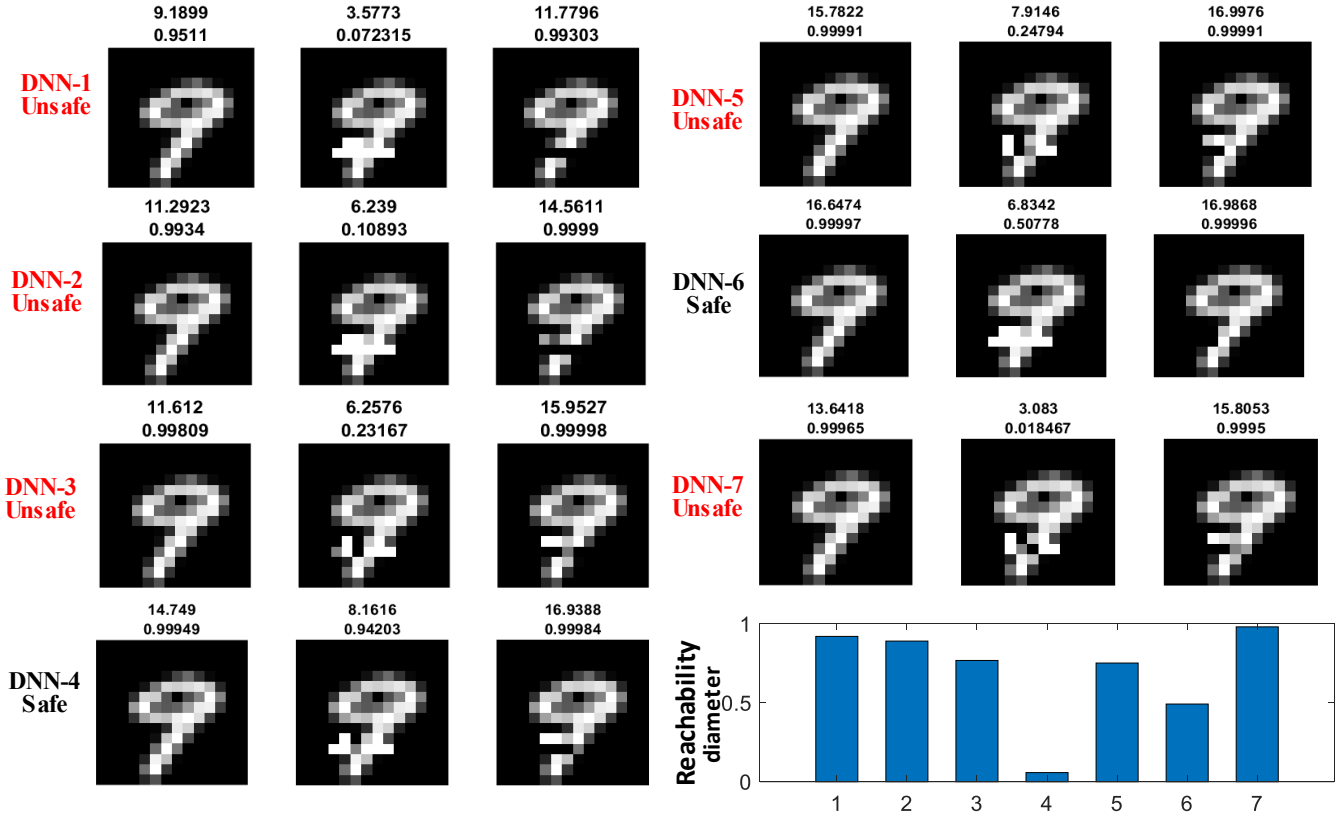
MNIST example

- Take an image and select a feature within it



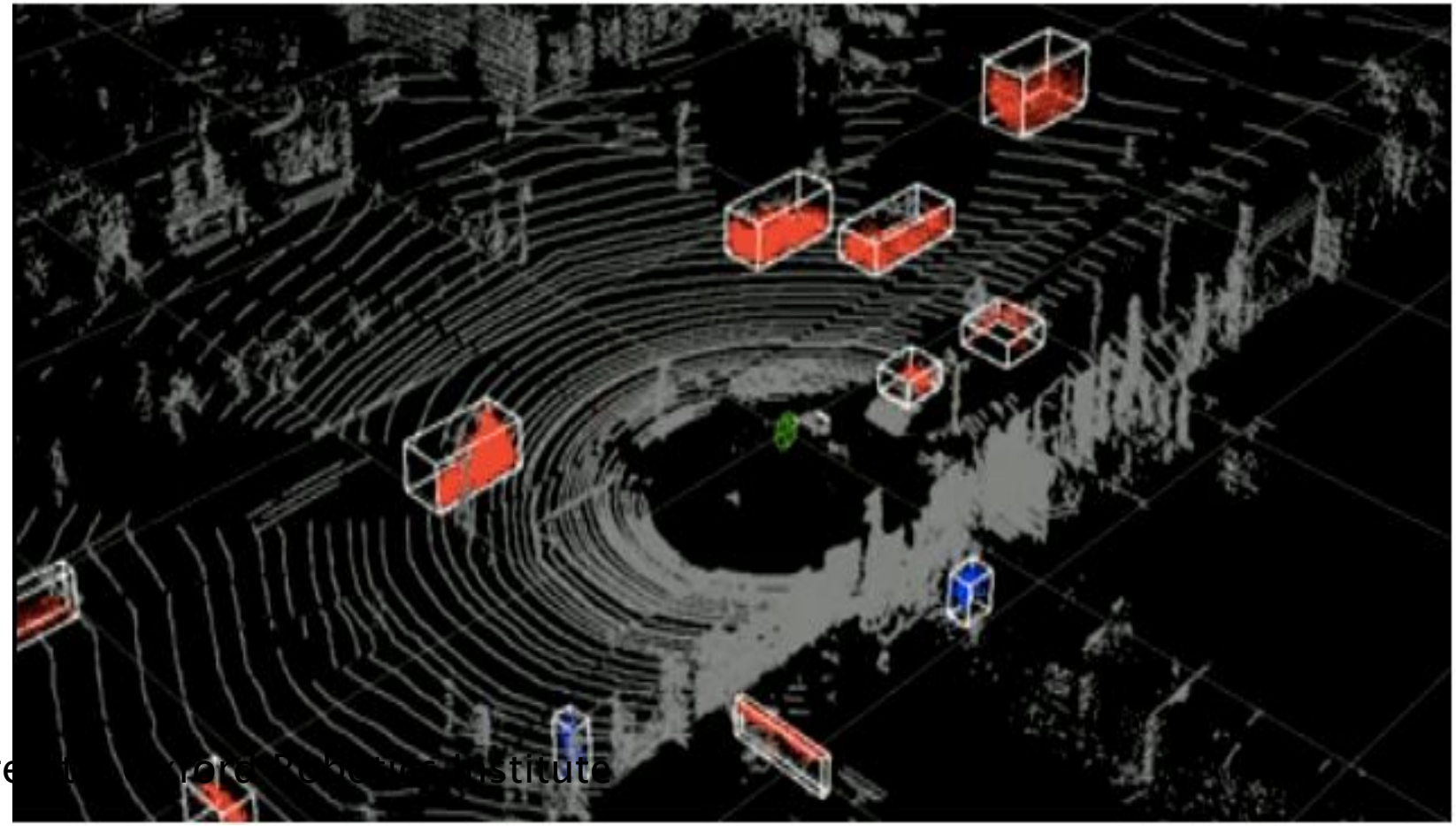
- Safety verification for the feature
  - manipulating the feature can only reduce confidence to 74.36%

MNIST network comparison



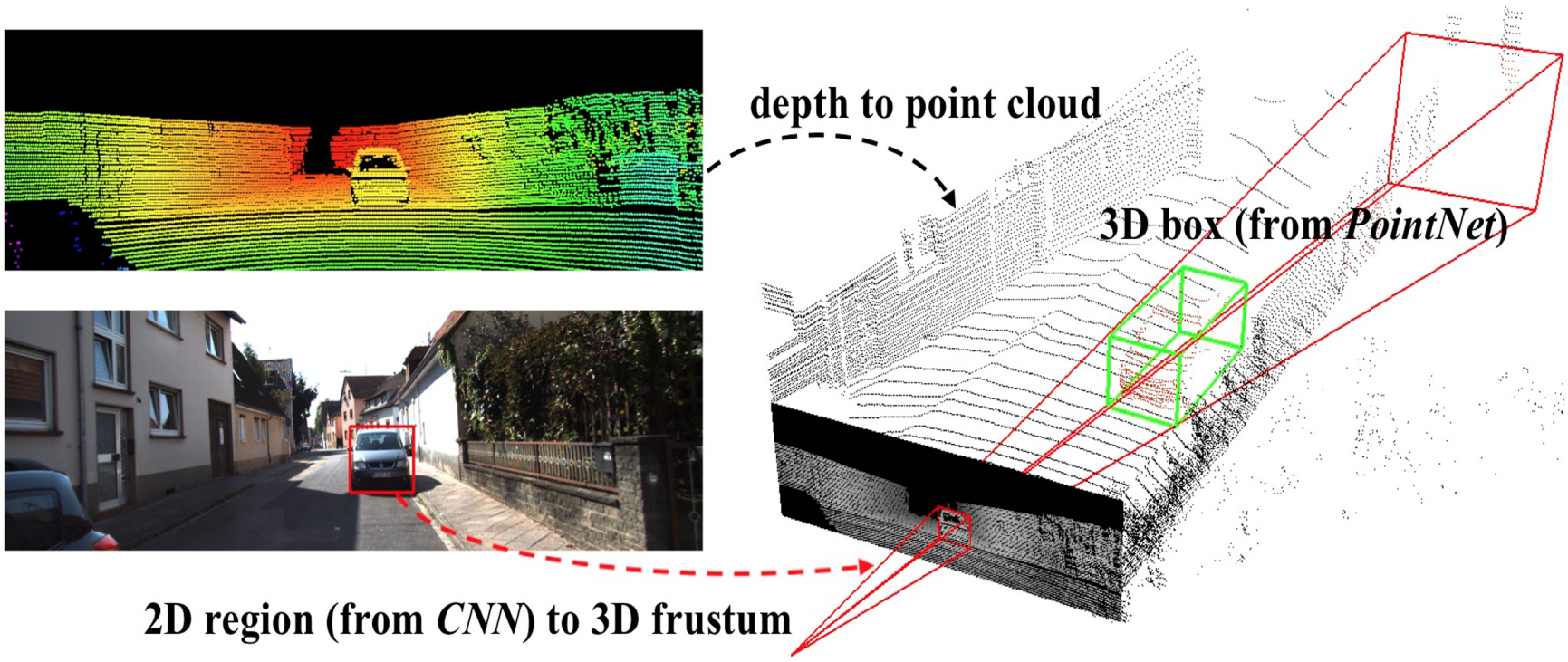
- Showing pointwise confidence diameter
- Can obtain global robustness evaluation by averaging wrt the test data distribution

Recent progress: 3D deep learning

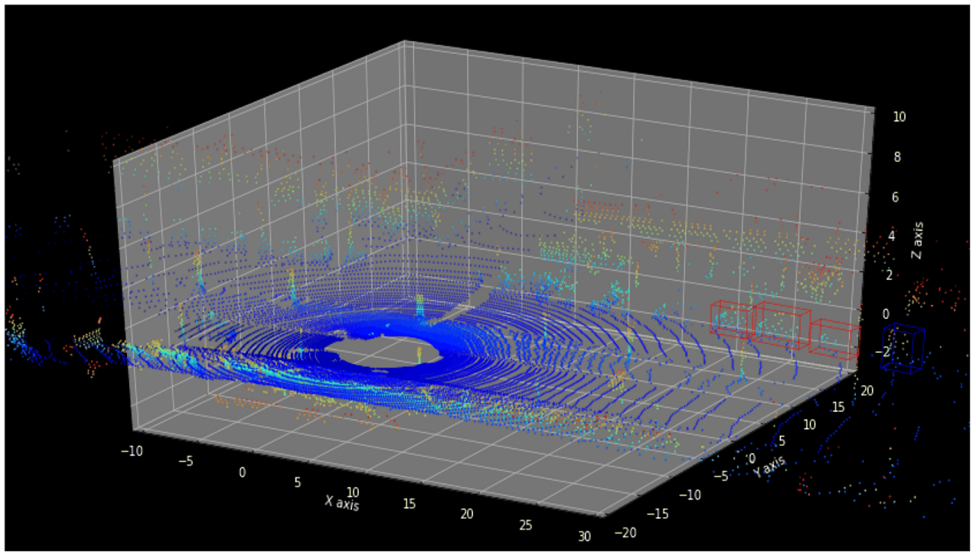


Cre[illegible]rd Robotics Institute

3D deep learning



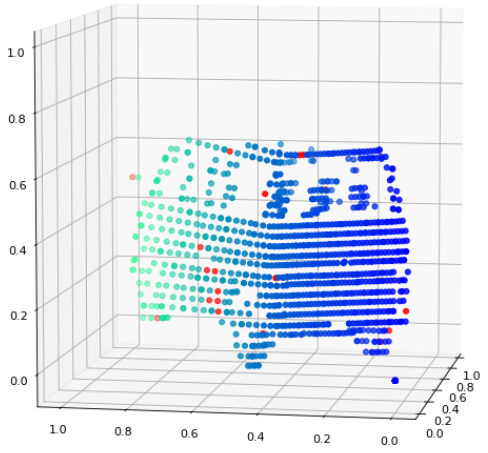
What is LiDAR?

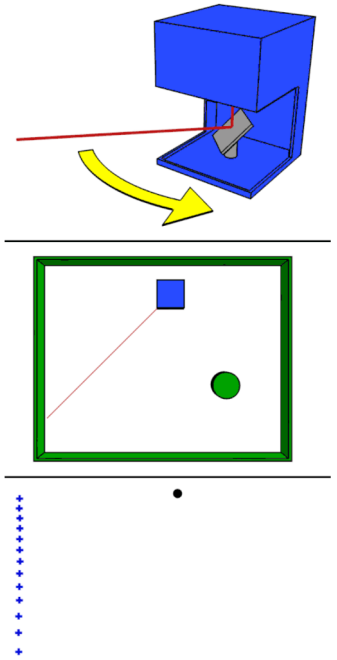


LiDAR stands for 'Light Detection And Ranging.' Differences in laser return times and wavelengths can be used to make digital 3D representations of the environment.

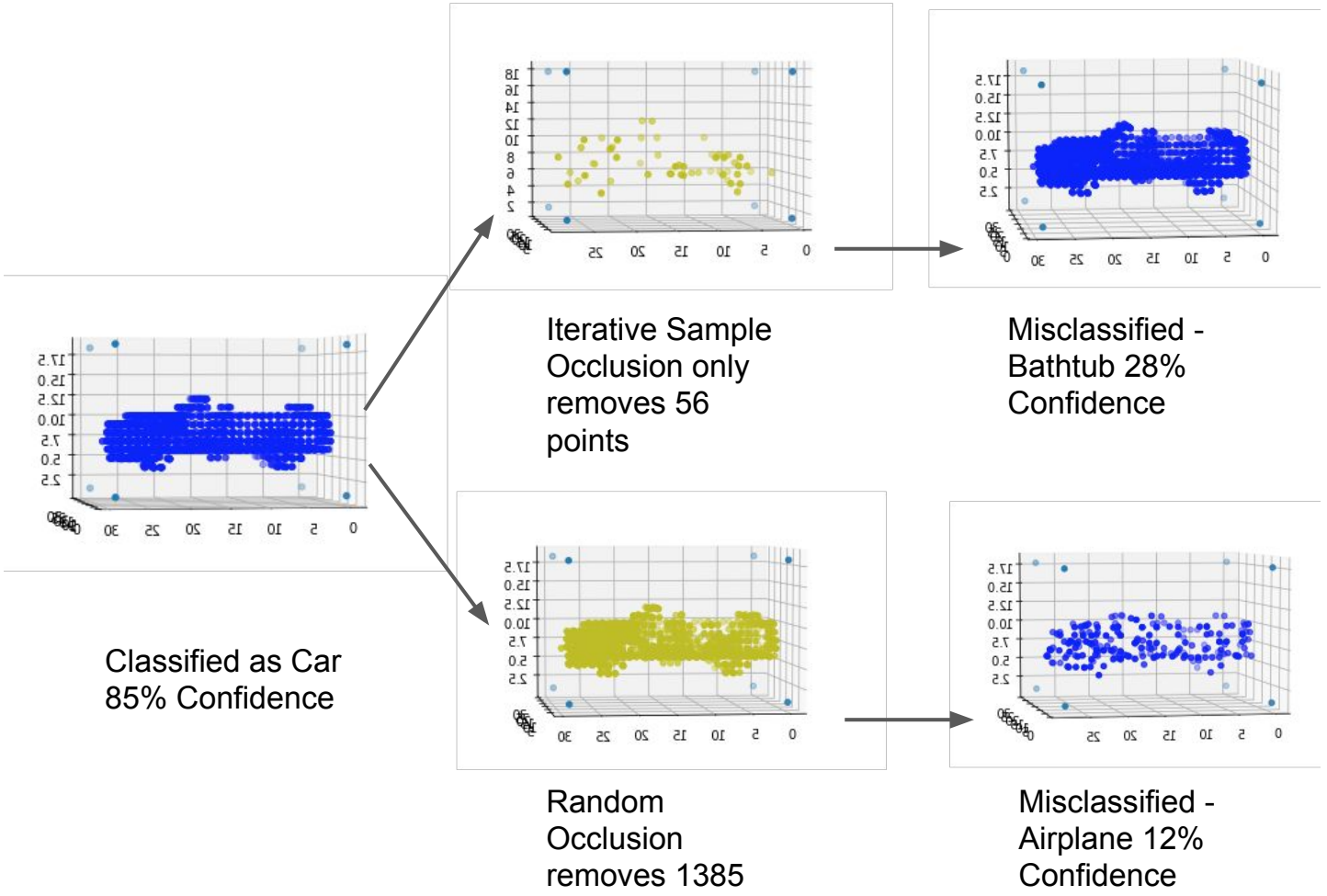
LiDAR and inherent error in point clouds

- Point ordering matters
- Partial occlusion of contiguous points
- Dark black could affect the reliability of sensor
- Misoriented sensors
- Need sub-second decision making





Attacks on Lidar



Classified as Car
85% Confidence

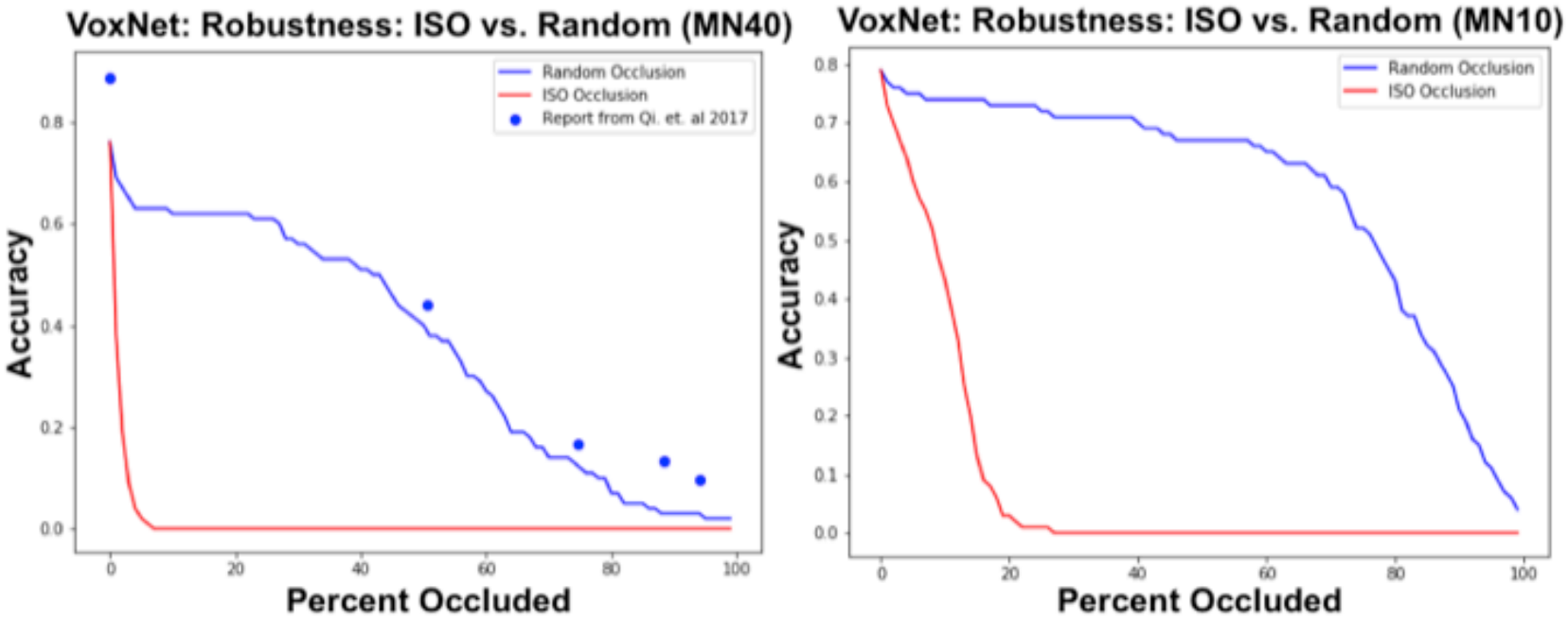
Iterative Sample Occlusion only removes 56 points

Misclassified - Bathtub 28% Confidence

Random Occlusion removes 1385

Misclassified - Airplane 12% Confidence

Attack success



…reduce accuracy to 0% after occlusion of 6.5% of the occupied input space, targeting the critical set

CVPR 2019, https://arxiv.org/abs/1904.00923

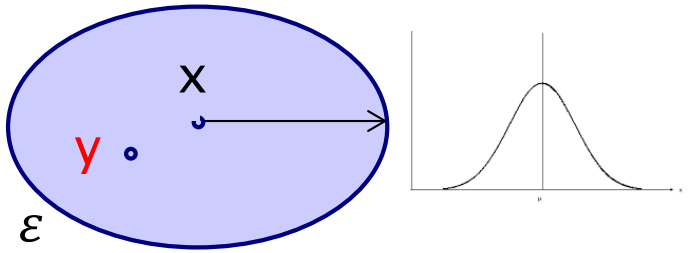
Probabilistic guarantees

- Requiring that no adversarial examples exist too strict!
- Need to probabilistic guarantees: probability that local perturbations result in predictions that are close to original
- Taking account of the learning process
- Bayesian neural networks have prior on weights
  - account for noise, uncertainty, etc
  - return an uncertainty measure along with the output
- Need to compute posterior probability
  - often intractable
  - can we do better?

Statistical robustness guarantees

- Work with Bayesian neural networks
- Define safety with prob $1-\varepsilon$

$Prob(\exists y \in \eta \text{ s.t. } f(x) \neq f(y) \mid D) \leq \varepsilon$

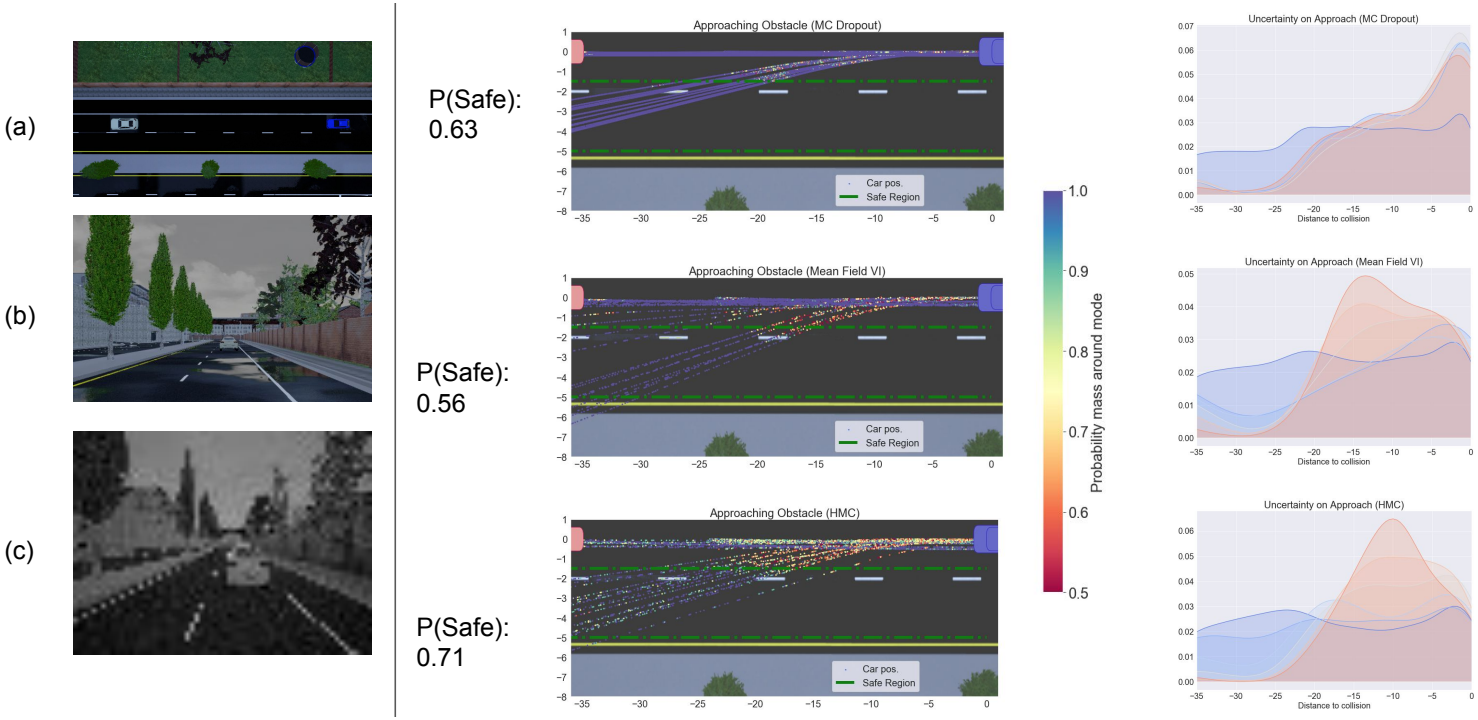


- i.e. conditioned on training data D
- Method: sample the weights, then employ statistical model checking (Massart bounds, sequential test)
  - compare robustness and accuracy trade offs for different inference methods

IJCAI 2019, https://arxiv.org/abs/1903.01980

Uncertainty quantification

- Safety verification for Bayesian neural network autonomous driving controllers



ICRA 2020, https://arxiv.org/abs/1909.09884

But more progress needed…

'I hate them': Locals reportedly are frustrated with Alphabet's self-driving cars

- Alphabet's self-driving cars are said to be annoying their neighbors in Arizona, where Waymo has been testing its vehicles for the last year.
- More than a dozen locals told The Information they they hated the cars, which often struggle to cross a T-intersection near the company's office.
- The anecdotes highlight how challenging it is for self-driving cars, which are programmed to drive conservatively, to handle certain situations.

Published 3:04 PM ET Tue, 28 Aug 2018 | Updated 12:53 PM ET Wed, 29 Aug 2018

CNBC

Source: Waymo

Self-driving cars should be allowed to mount pavements and break speed limit in emergencies

share

28

A Tesla Model S

Summary

- Deep learning should be more critically evaluated when put into practice in safety- and security-critical situations
- Adversarial examples help in understanding the robustness of DNN decision boundaries
- Overviewed methods for safety verification of deep neural networks
  - search-based and feature-guided exploration, with guarantees
  - reachability computation for Lipschitz continuous networks
  - probabilistic guarantees in a Bayesian framework
- Projects
  - See AIMS projects, happy to discuss related topics

Verification for ML challenging

- What's different about machine learning?
  - programming by pattern matching, not logic
  - black box, lacks interpretability
  - corner cases are unseen examples, not missed conditions
  - data quality and coverage crucial
  - accuracy can be misleading

- Why is ML difficult to verify?
  - algorithmic foundations of ML not well understood
  - training obscure, not clear how to choose the training method
  - dependence on choice of loss functions and optimisation
  - need to debug data, not programs…

- Need synthesis, not just verification…

Acknowledgements

- My group and collaborators in this work
- Project funding
  - ERC Advanced Grant VERIWARE
  - EPSRC Mobile Autonomy Programme Grant

- See also
  - PRISM www.prismmodelchecker.org

- New ERC Advanced Grant FUN2MODEL

  "From FUNction-based TO MOdel-based automated probabilistic reasoning for DEep Learning"
- Postdoctoral and PhD positions