Special Topics in Mechano-Informatics II

Interpretable and Adversarial Machine Learning

Qibin Zhao

Tensor Learning Team
RIKEN AIP
https://qibinzhao.github.io

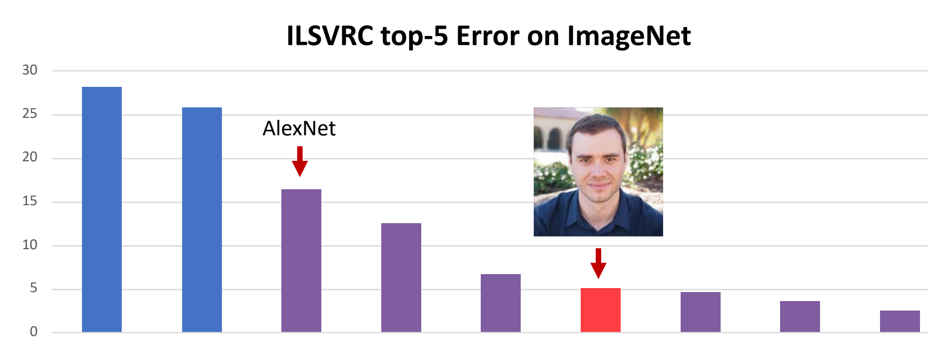
Machine Learning: The Success Story

Image classification

Reinforcement Learning

Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

Machine translation



ML Achieves Superhuman Performance

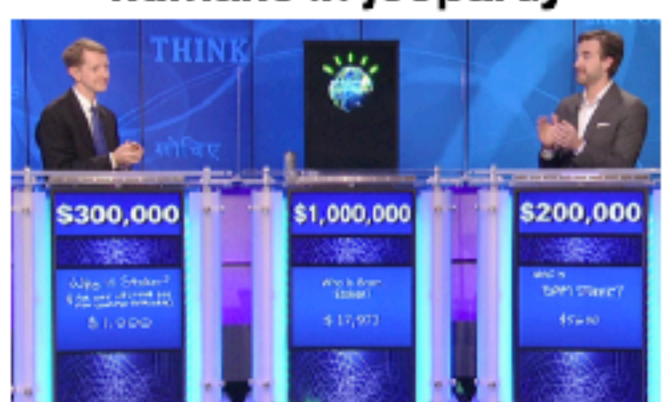
AlphaGo beats Go human champ

Computer out-plays humans in "doom"

Deep Net outperforms humans in image classification

Autonomous search-and-rescue drones outperform humans

IBM's Watson destroys humans in jeopardy



DeepStack beats professional poker players

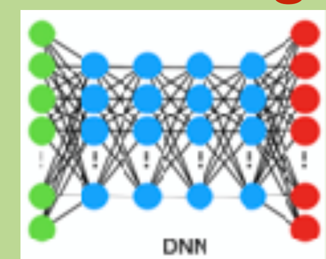
Deep Net beats human at recognizing traffic signs

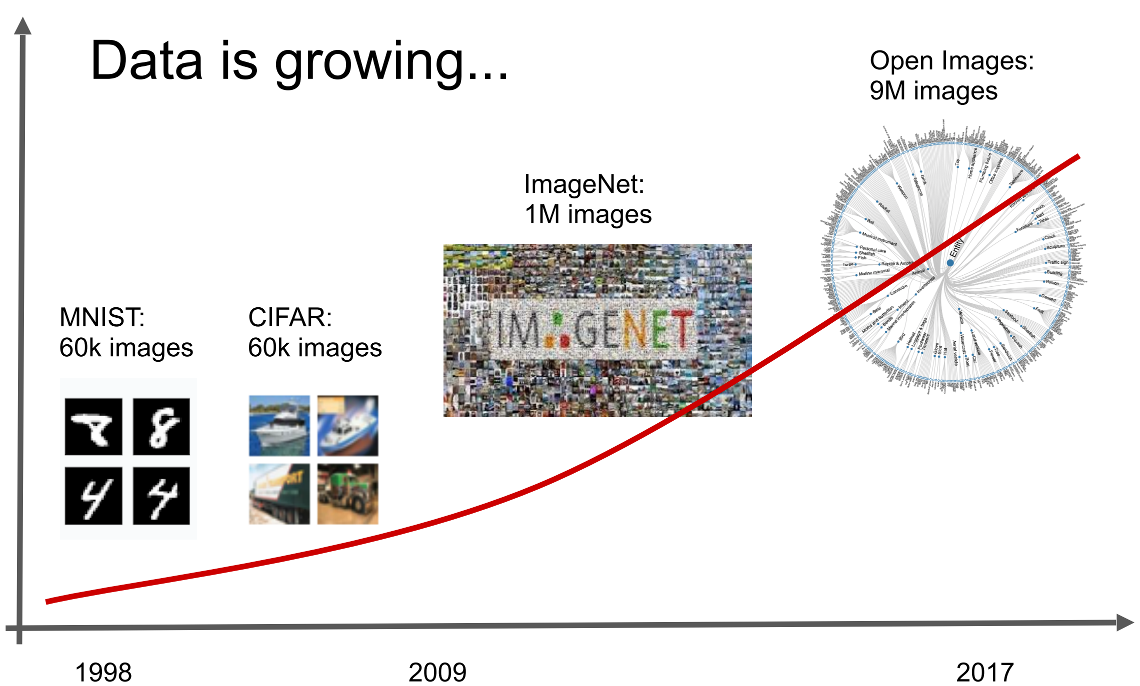
Evolution of ML

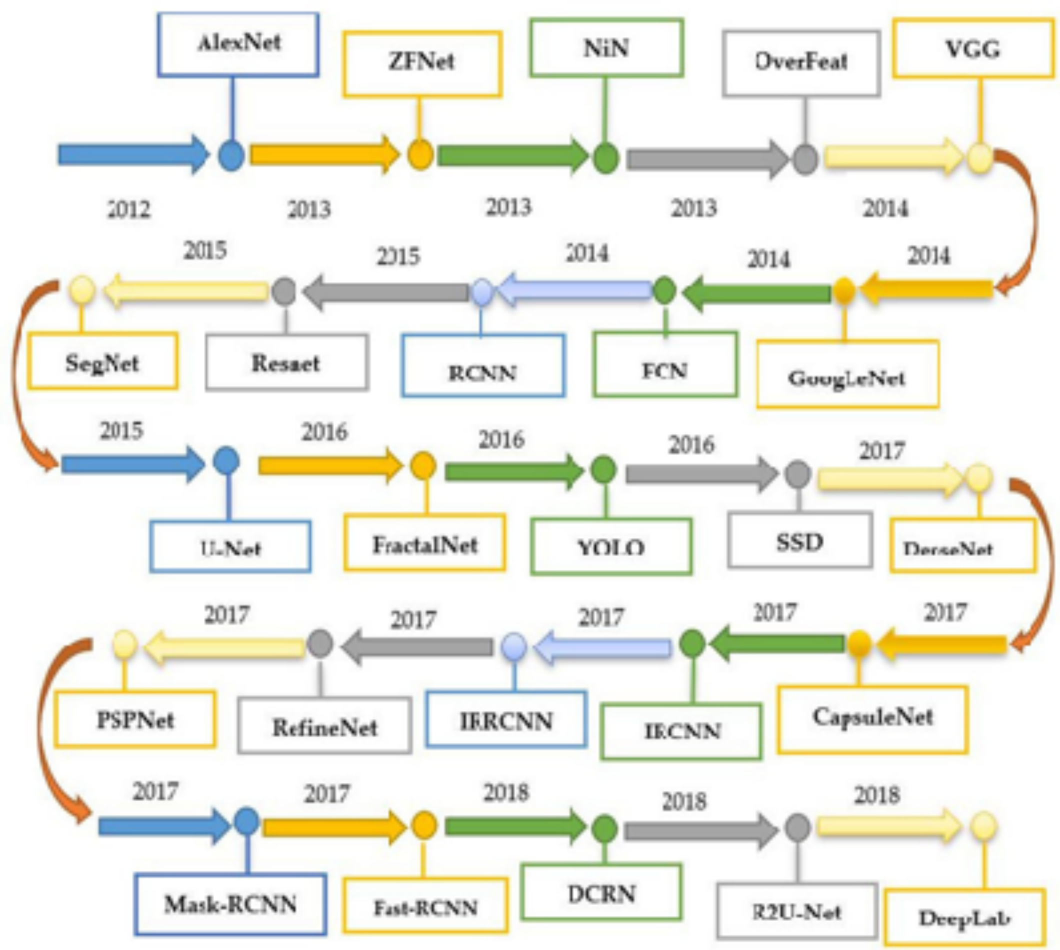
Big Data

Computational Resources

Machine Learning







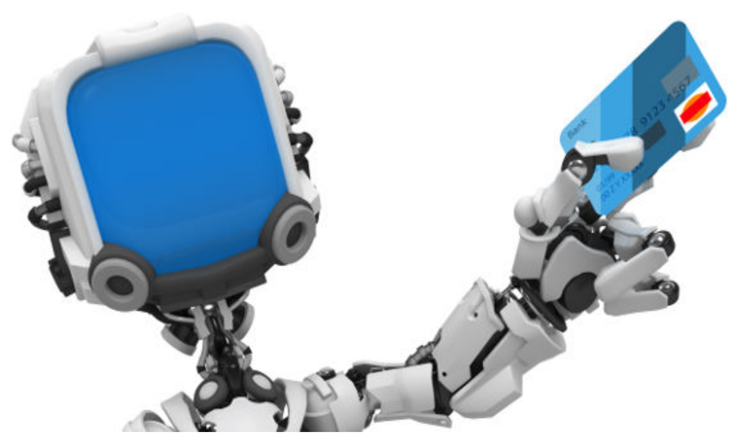
ML in Physical World

Autonomous Driving

Healthcare

Smart City

Malware Classification



Fraud Detection

Biometrics Recognition

Consequences in Real-world Applications

The FBI Has Access to Over 640 Million Photos of Us Through Its Facial Recognition Database

By Neema Singh Guliani, ACLU Senior Legislative Counsel
JUNE 7, 2019 | 3:15 PM

TAGS: Face Recognition Technology, Surveillance Technologies, Privacy & Technology

Andrew J. Hawkins @andyjayhawk

Follow

In 2016, a Tesla driver using Autopilot crashed into the side of a truck and was killed. It happened again three months ago, but this time with a completely new version of Autopilot. What's the heck is going on?? theverge.com/2019/5/17/1862 …

1:14 PM - 17 May 2019

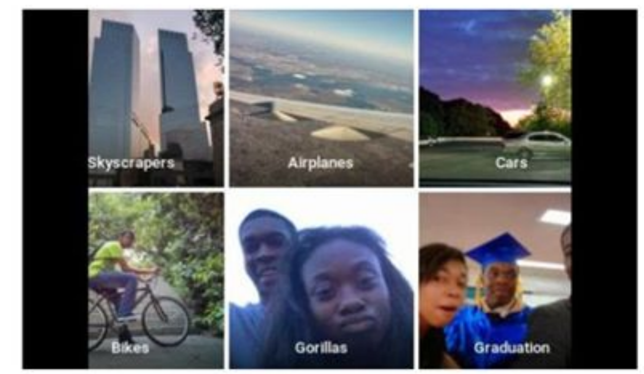
NEWS

Home | Video | World | US & Canada | UK | Business | Tech | Science | Magazine

Technology

Google apologises for Photos app's racist blunder

1 July 2015 | Technology



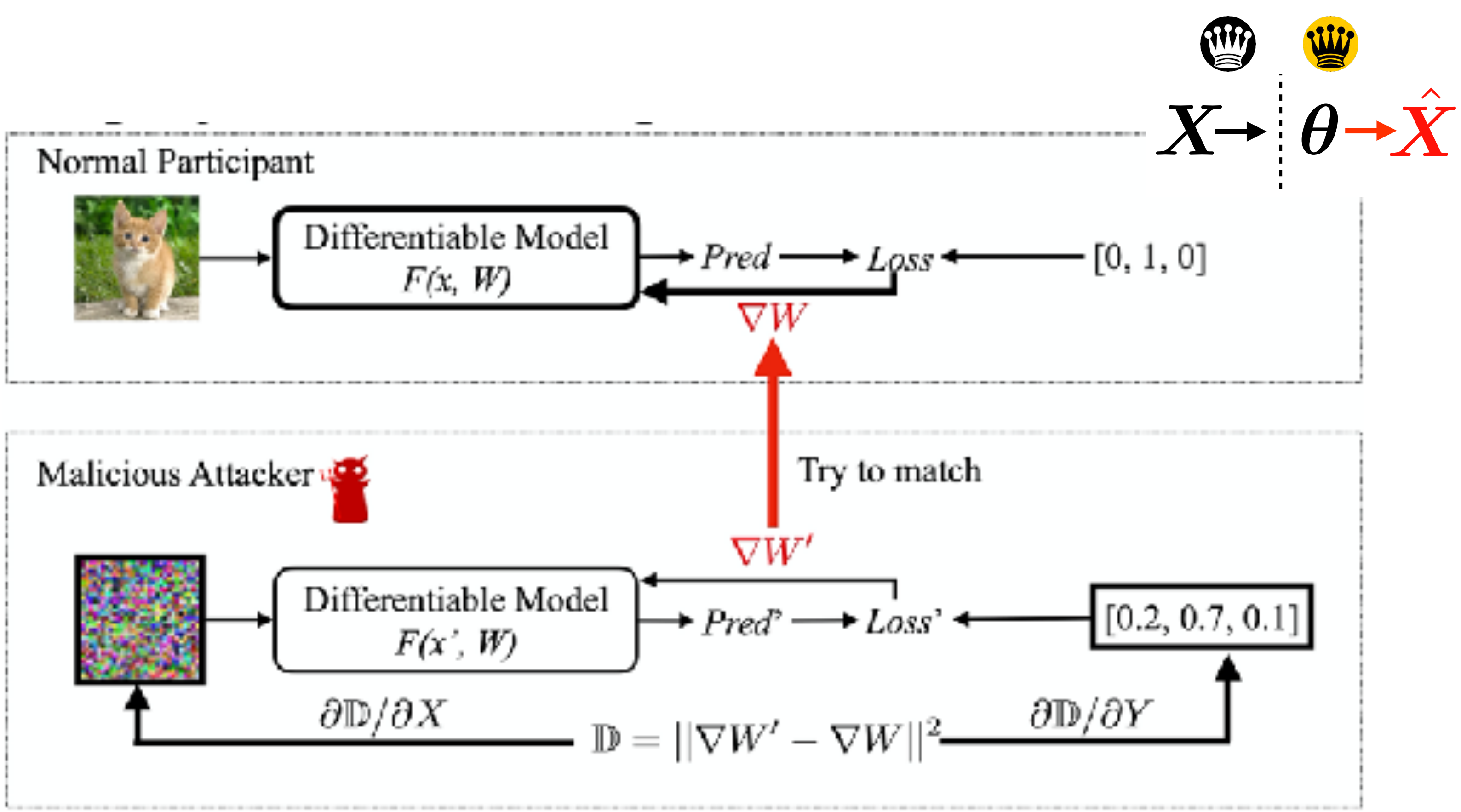
Robust Physical-World Attacks on Machine Learning Models

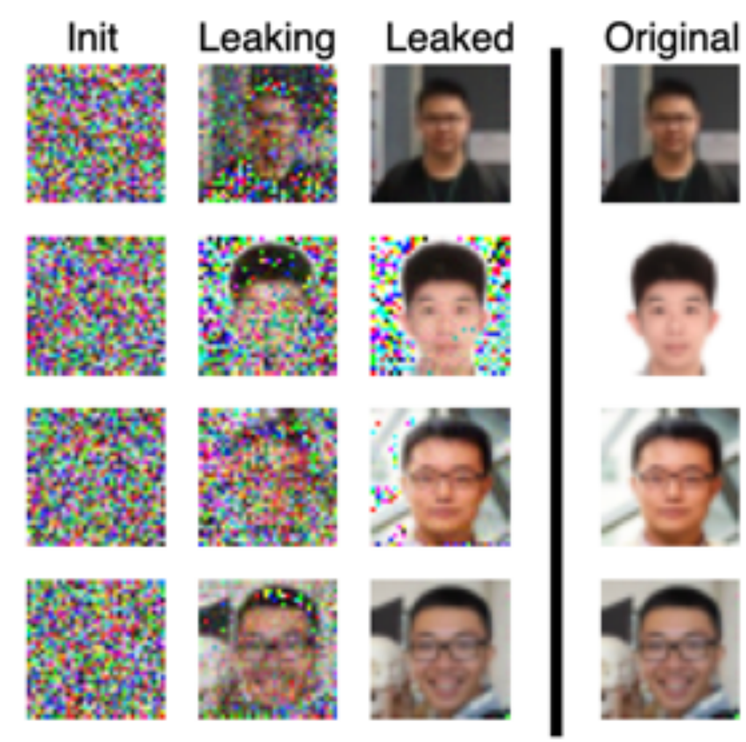
Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, Dawn Song

(Submitted on 27 Jul 2017 (v1), last revised 30 Jul 2017 (this version, v2))

Privacy: Deep Leakage from Gradients

- Federated learning: model is moving while private training data never leaves local device
- However, training data can be leaked by publicly shared gradients





(Ligeng Zhu et al., Deep Leakage from Gradients. NeurIPS 2019)

Building Trust between Human and AI

Training models off of sensitive data needs privacy preserving safeguards

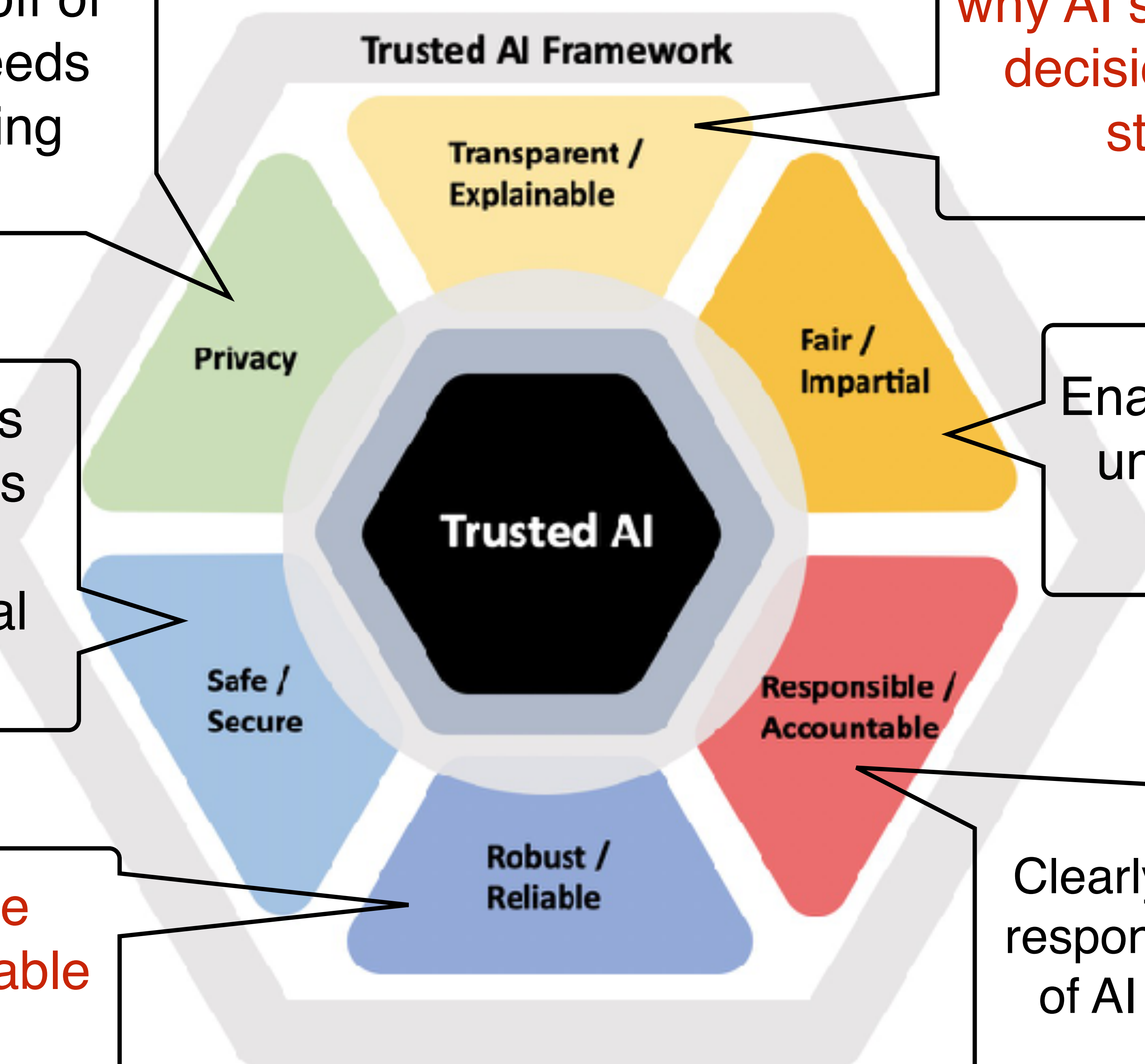
Explanations on how and why AI systems make the decisions for different stakeholders

Protect AI systems from potential risks that may cause physical and digital harm

Enable equitable and unbiased decision making

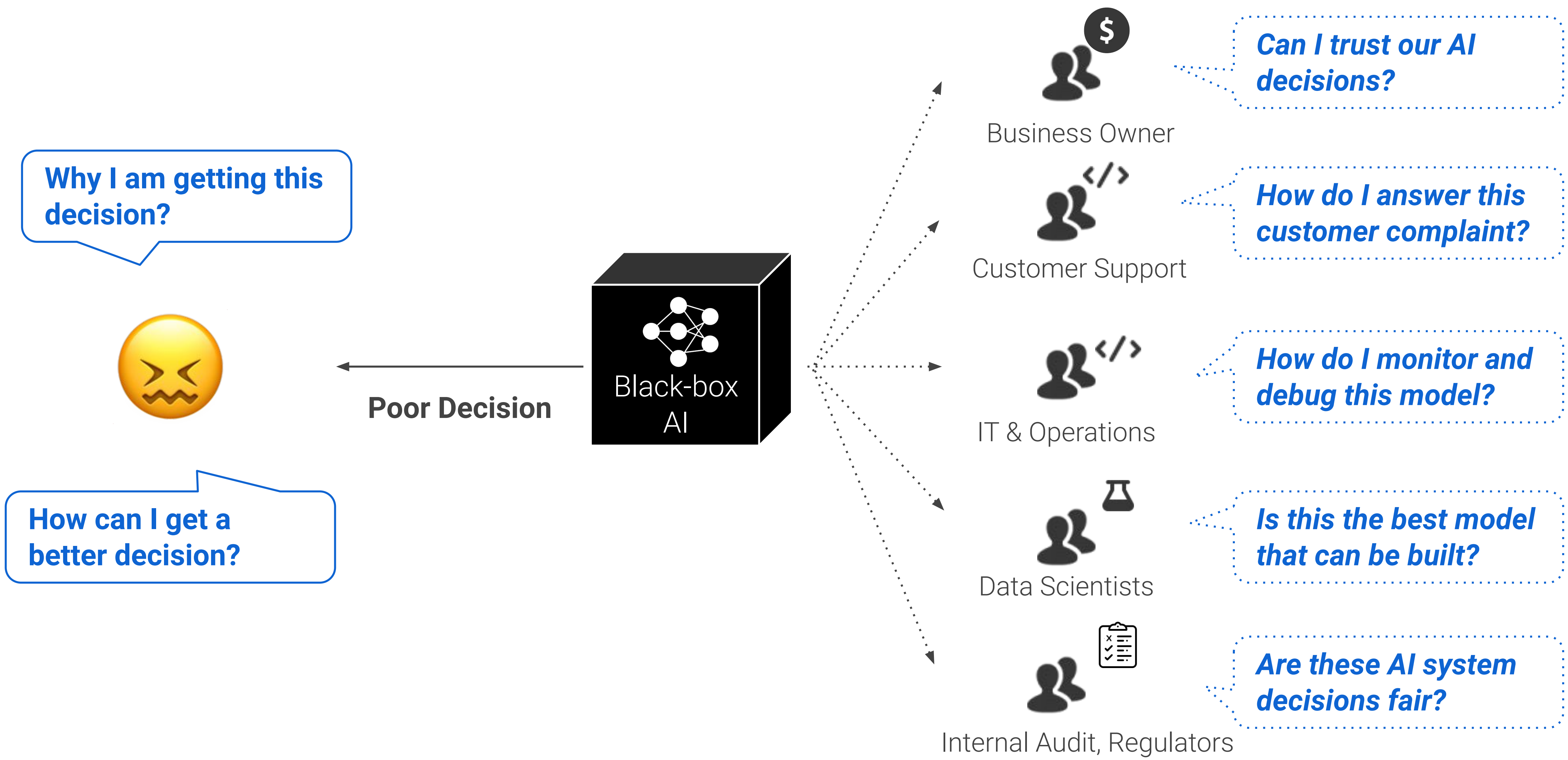
Ability to produce consistent and reliable outputs

Clearly determine who is responsible for the output of AI system decisions

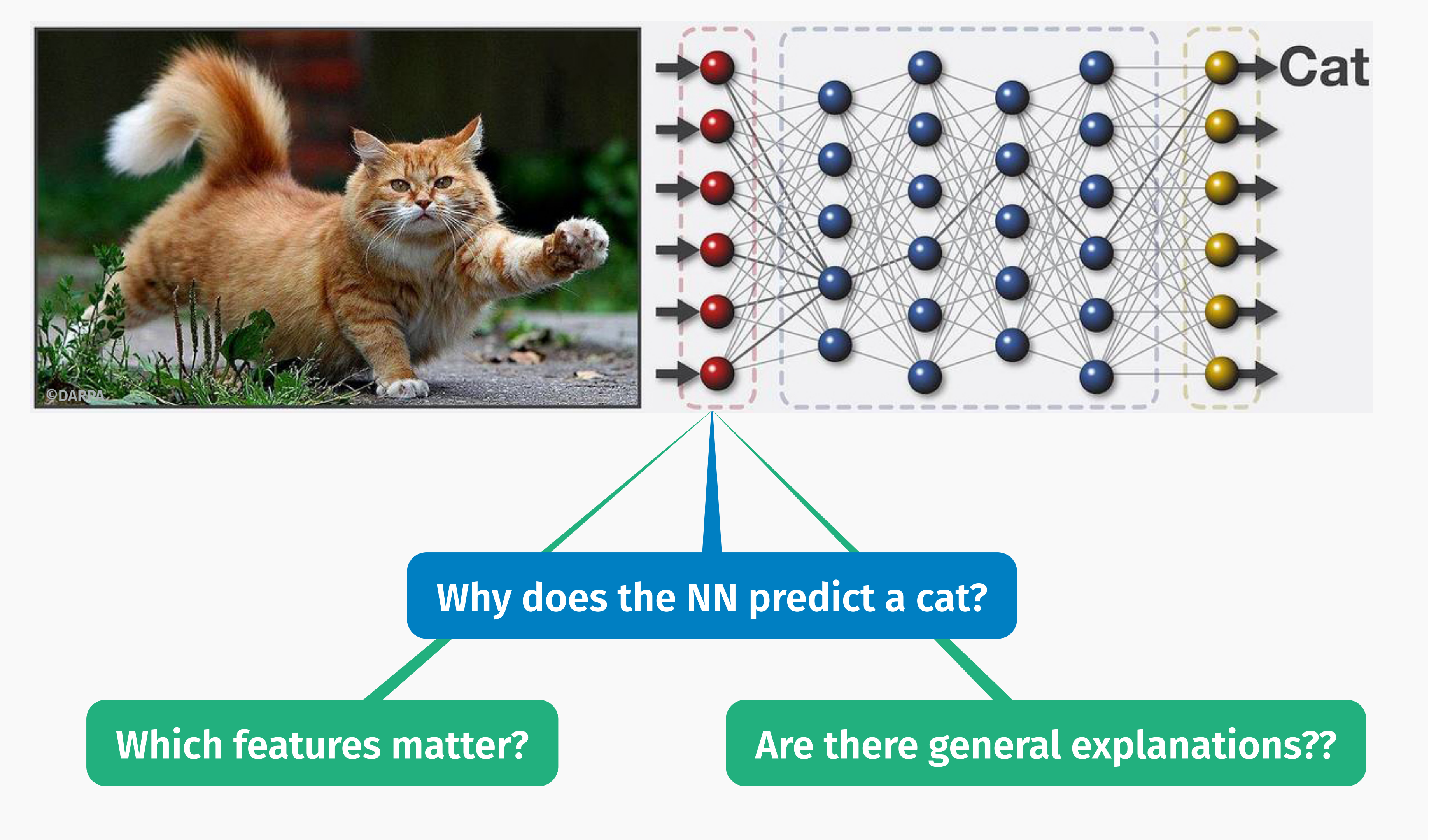


Interpretable/Explainable Machine Learning

Black-box AI Creates Confusions



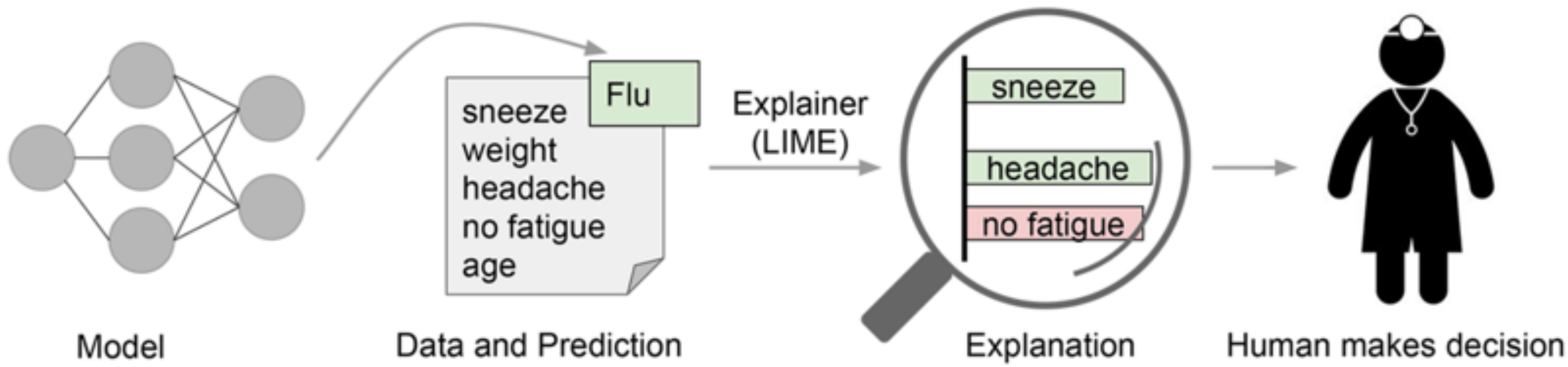
Black-box Model



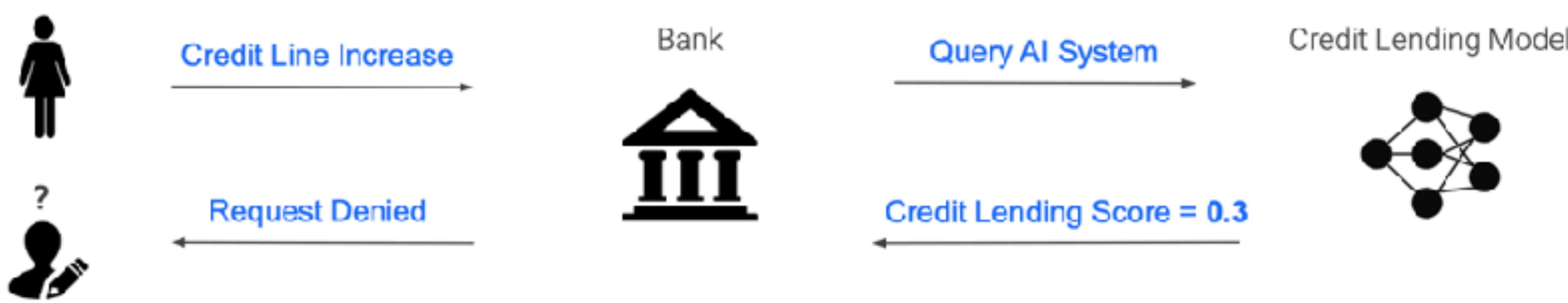
- Internals are unknown to observer
- Internals are known but uninterpretable

Explanations in ML world

Medical Diagnosis



Credit Evaluation



Why? Why not?
How?

Fair lending laws [ECOA, FCRA] require credit decisions to be explainable

What is Explainable AI

Black Box AI

Data → Black-Box AI → AI product → Decision, Recommendation

Confusion with Today's AI Black Box

- Why did you do that?
- Why did you not do that?
- When do you succeed or fail?
- How do I correct an error?

Explainable AI

Data → *Explainable* AI → *Explainable* AI Product → Decision / Explanation

Feedback

Clear & Transparent Predictions

- I understand why
- I understand why not
- I know why you succeed or fail
- I understand, so I trust you

Significance: Strong impacts

Manipulability: Controllable effects

Complexity: Gaining insights

Low Interpretability ⟷ High Interpretability

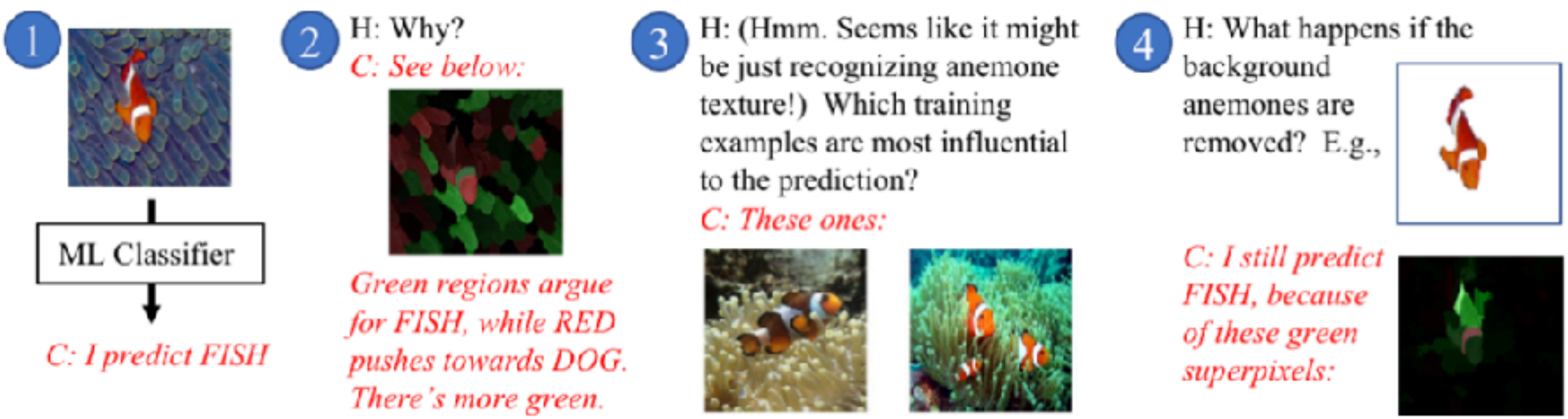
What is Interpretable/Explainable ML

There is no mathematical definition of interpretability. Two proposed definitions in the literature are:

- *Interpretability is the degree to which a human can understand the cause of a decision. — Tim Miller*

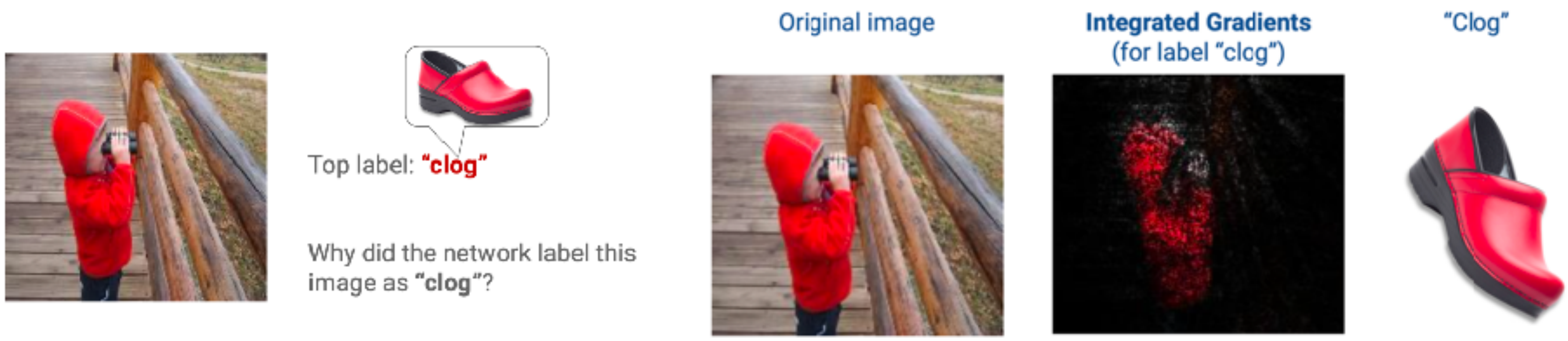
- *Interpretability is the degree to which a human can consistently predict the model's result. — Been Kim*

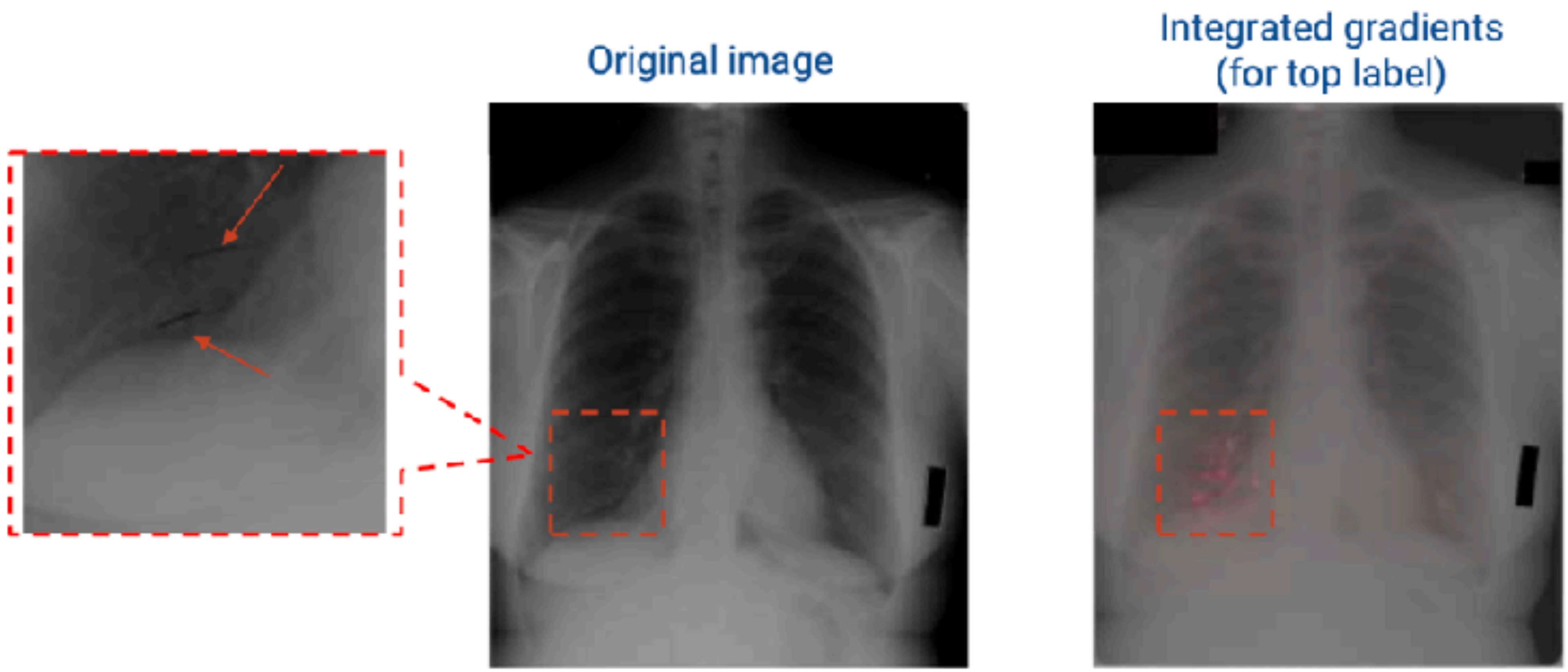
Why Explainability?
Generating Explanation for the End-User



Weld, D., et al, The challenge of crafting intelligible intelligence, Communications of ACM (2018)

Why Explainability?
Debug (Mis)-Prediction



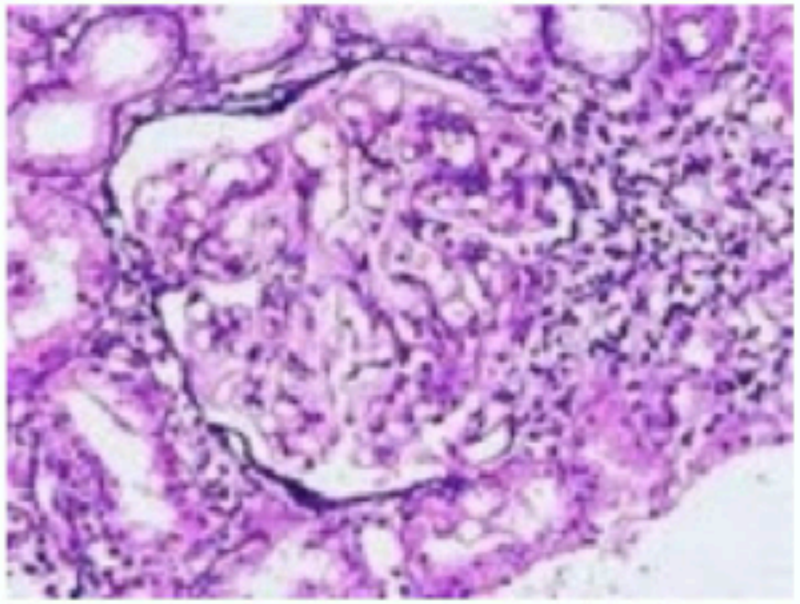


Why Explainability: Verify the ML Model/System

Wrong decisions can be costly and dangerous

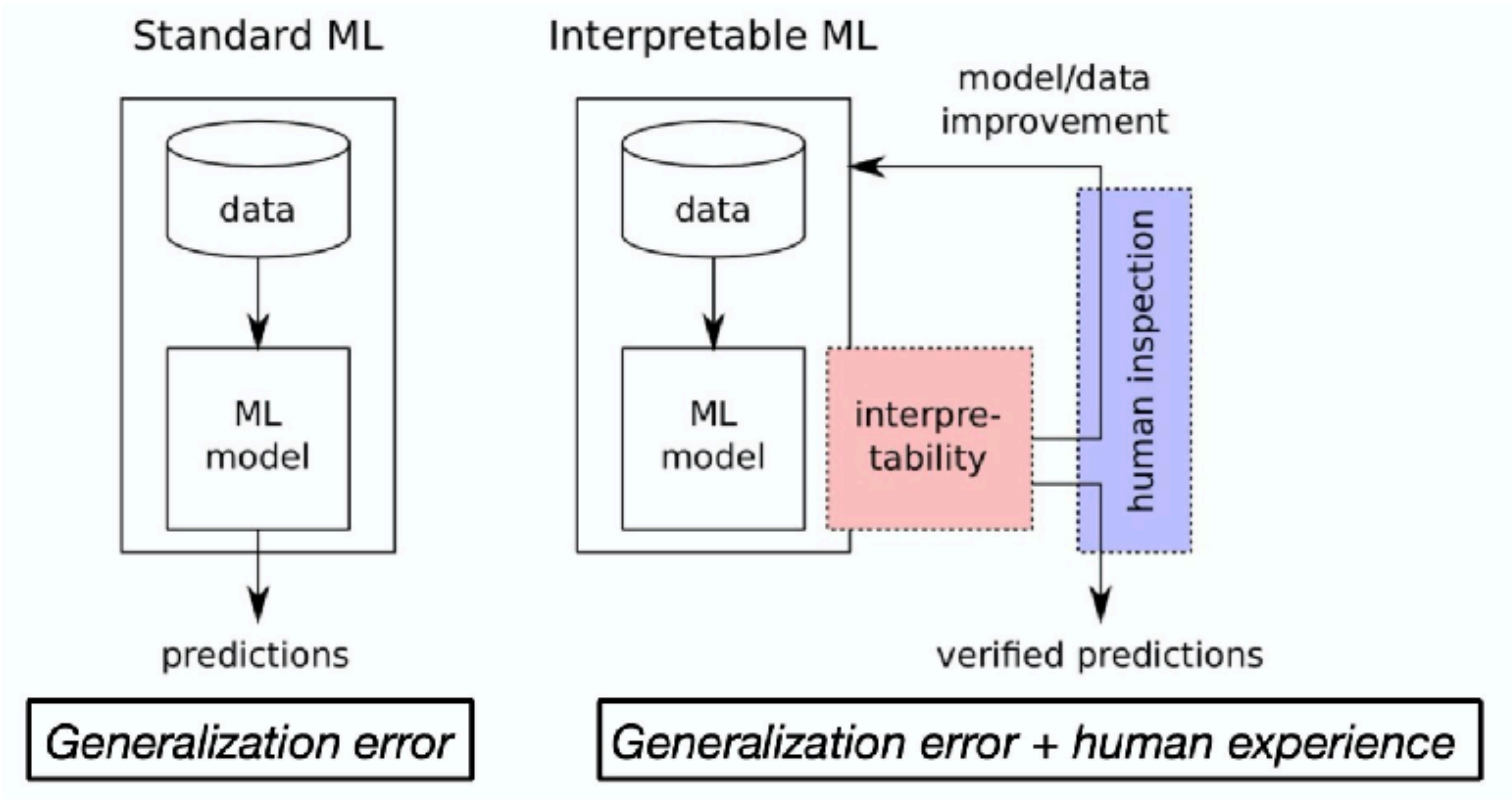
"Autonomous car crashes, because it wrongly recognizes …"

"AI medical diagnosis system misclassifies patient's disease …"



Credit: Samek, Binder, Tutorial on Interpretable ML, MICCAI'18

Why Explainability? Improve ML Model

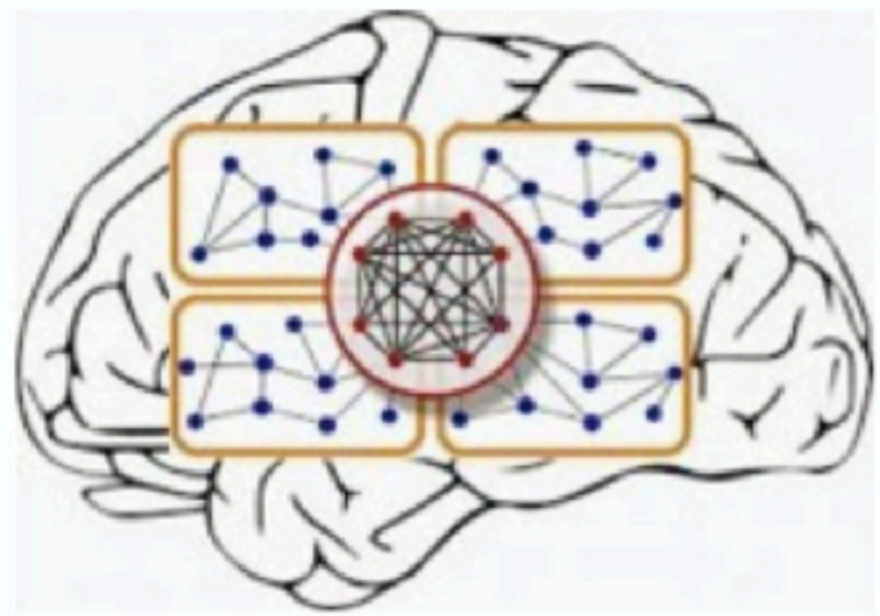


Credit: Samek, Binder, Tutorial on Interpretable ML, MICCAI'18

Why Explainability: Learn New Insights

"It's not a human move. I've never seen a human play this move." (Fan Hui)

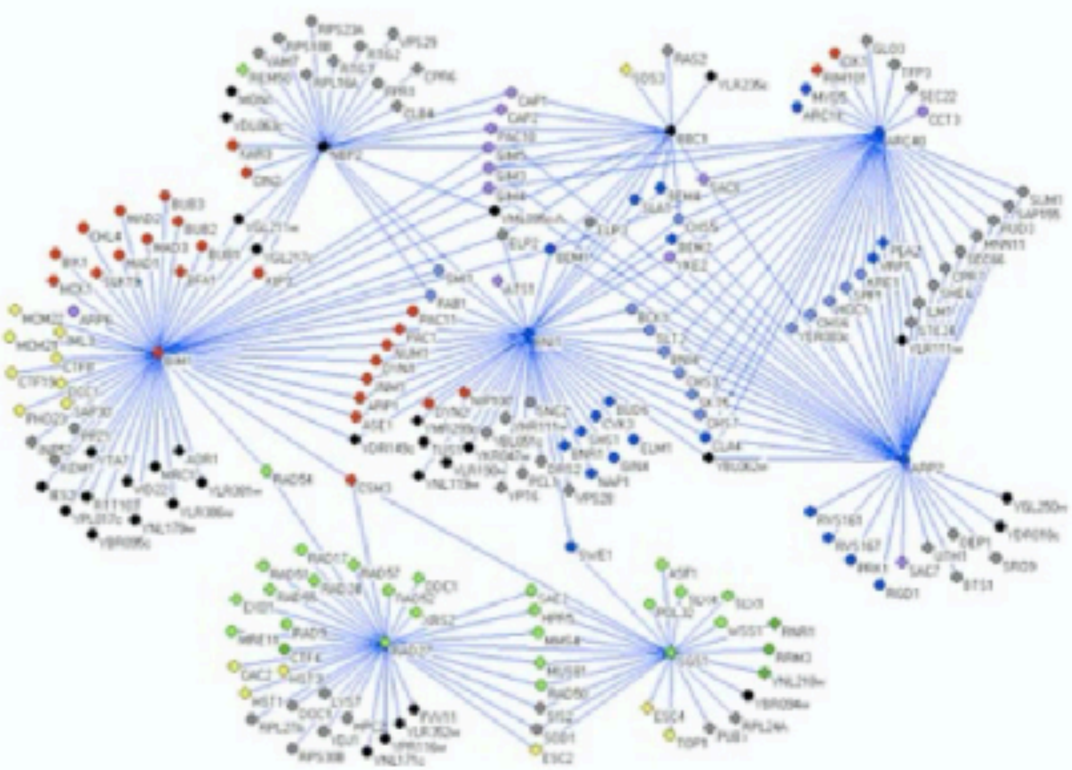
Old promise:
"Learn about the human brain."

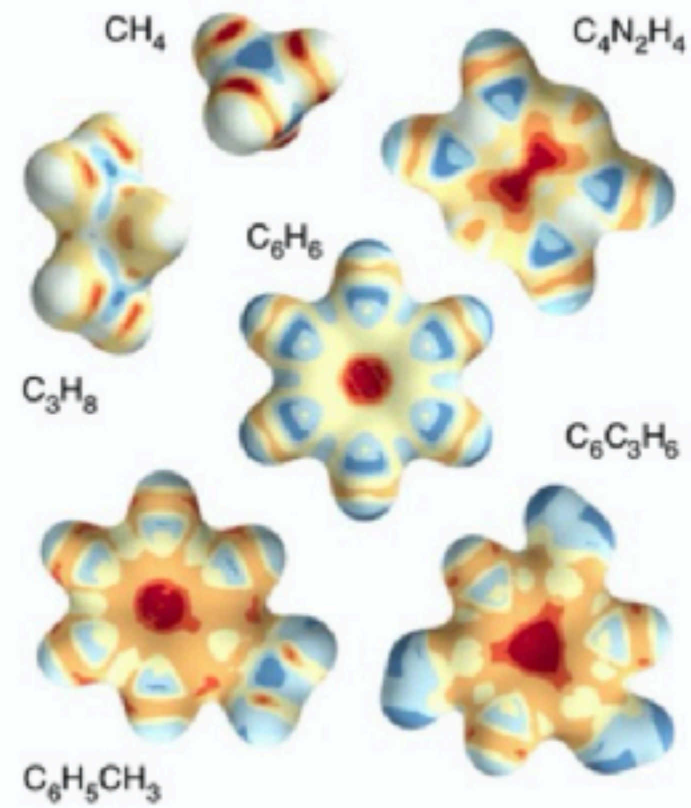


Credit: Samek, Binder, Tutorial on Interpretable ML, MICCAI'18

Why Explainability: Learn Insights in the Sciences

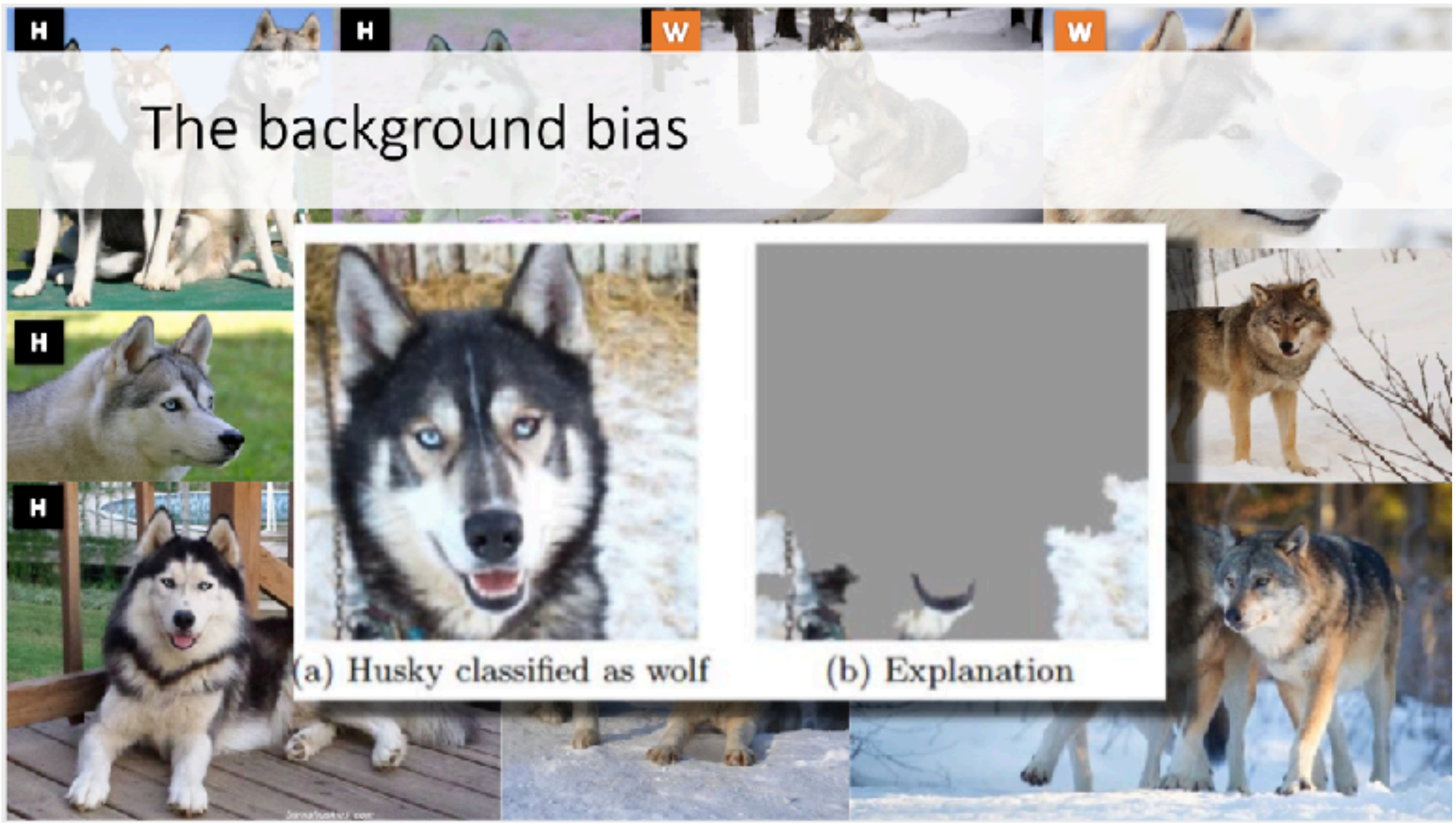
Learn about the physical / biological / chemical mechanisms.
(e.g. find genes linked to cancer, identify binding sites …)





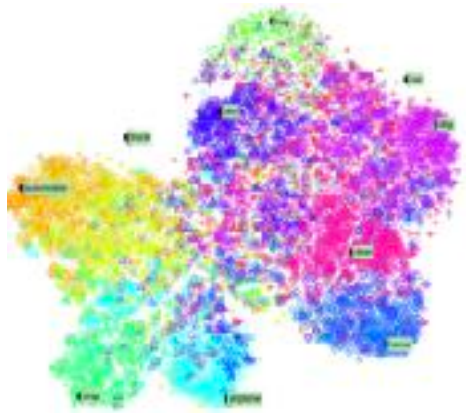
Credit: Samek, Binder, Tutorial on Interpretable ML, MICCAI'18

Why Interpretability: Find Bias and Fairness



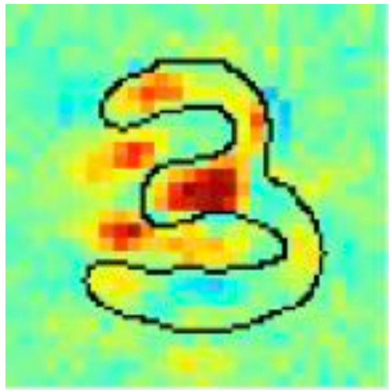
What kind of Interpretation?

- Data: Which dimensions of the data are most relevant for the task?



- Model: What concept does a particular neural encode?

- Prediction: Explain why a certain instance has been classified as a certain class



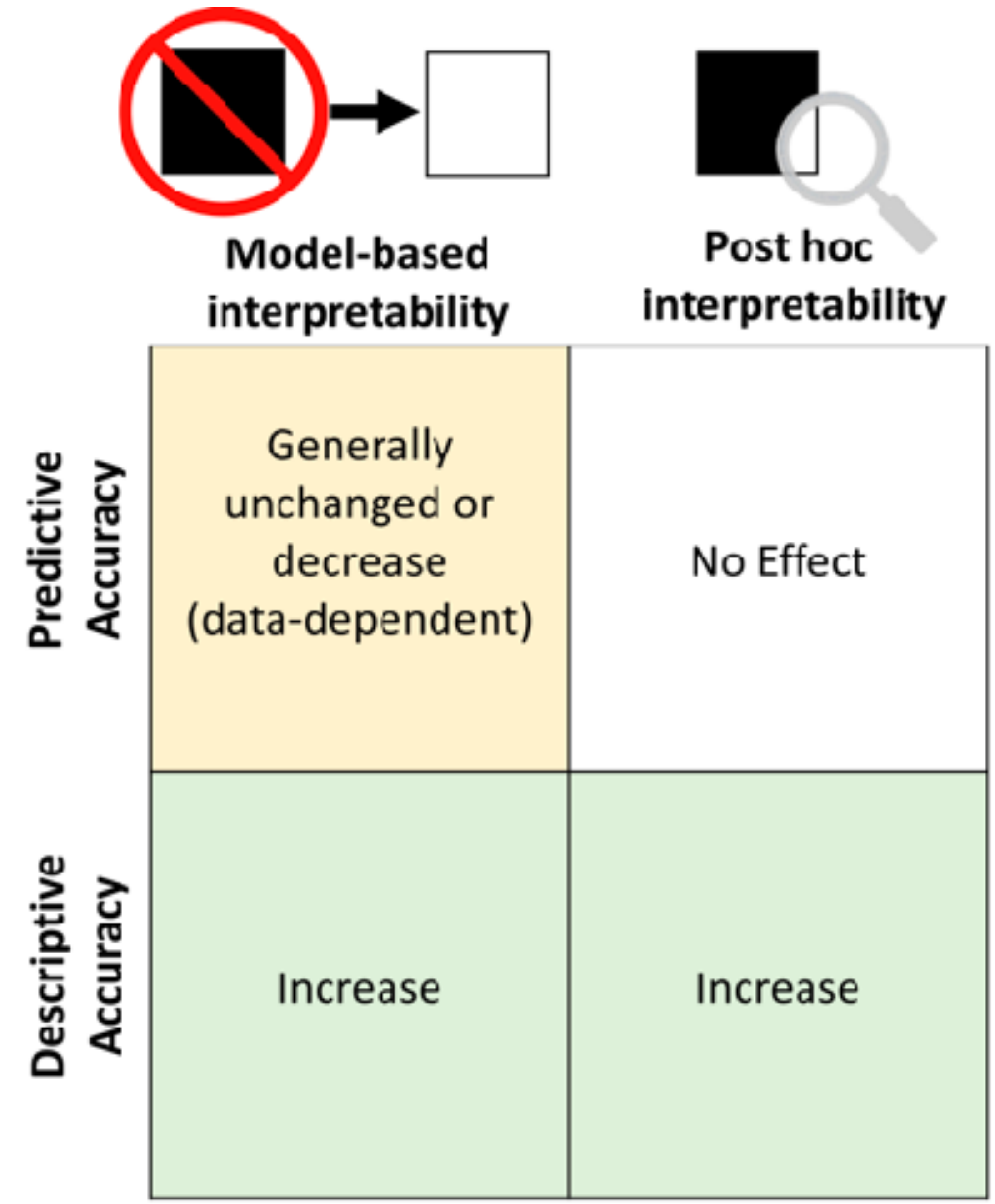
Model-based vs. Post Hoc Interpretability

Model-based

- Simpler model to fit the data
- Lower predictive accuracy but higher descriptive accuracy

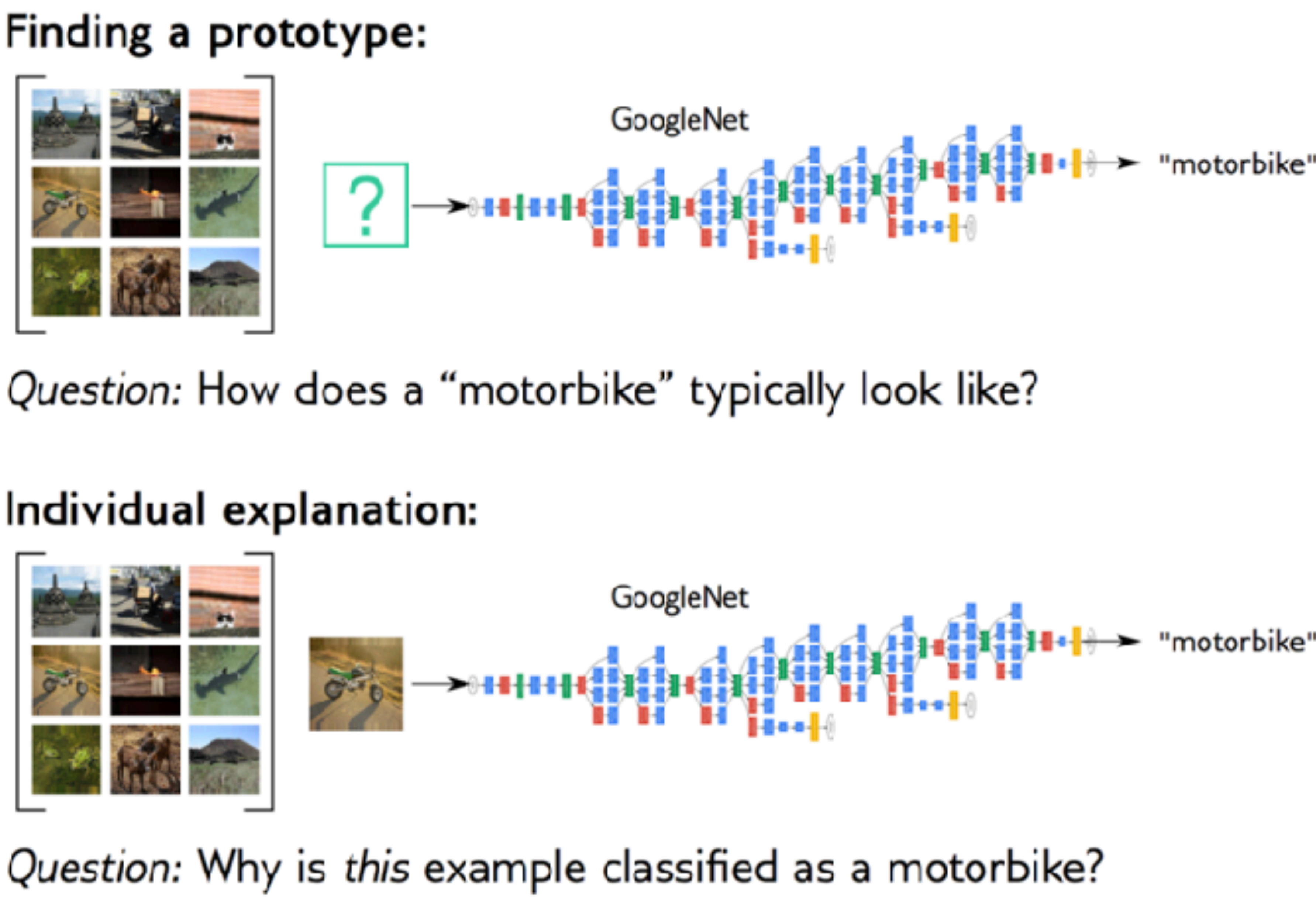
Post hoc

- Analyze or visualize information of a trained model
- Unchanged predictive accuracy



Definitions, methods, and applications in interpretable machine learning (Murdoch et al. PNAS 2019)

Global vs. Local Explanations



Global vs. Local Interpretation

Global interpretation

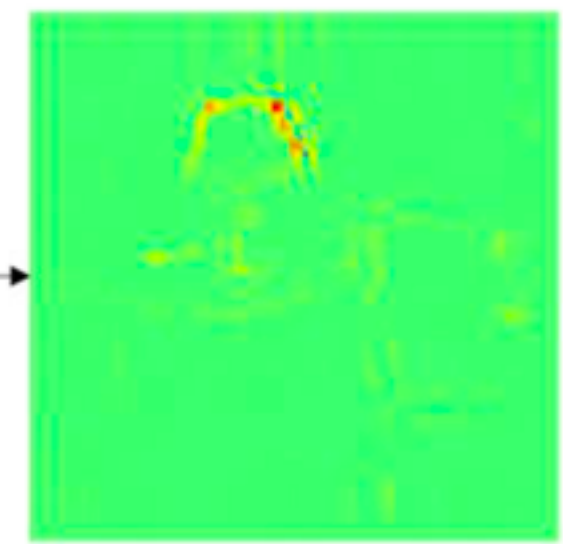
- Understanding how a lamp typically looks like

model's prototypical lamp

Local interpretation

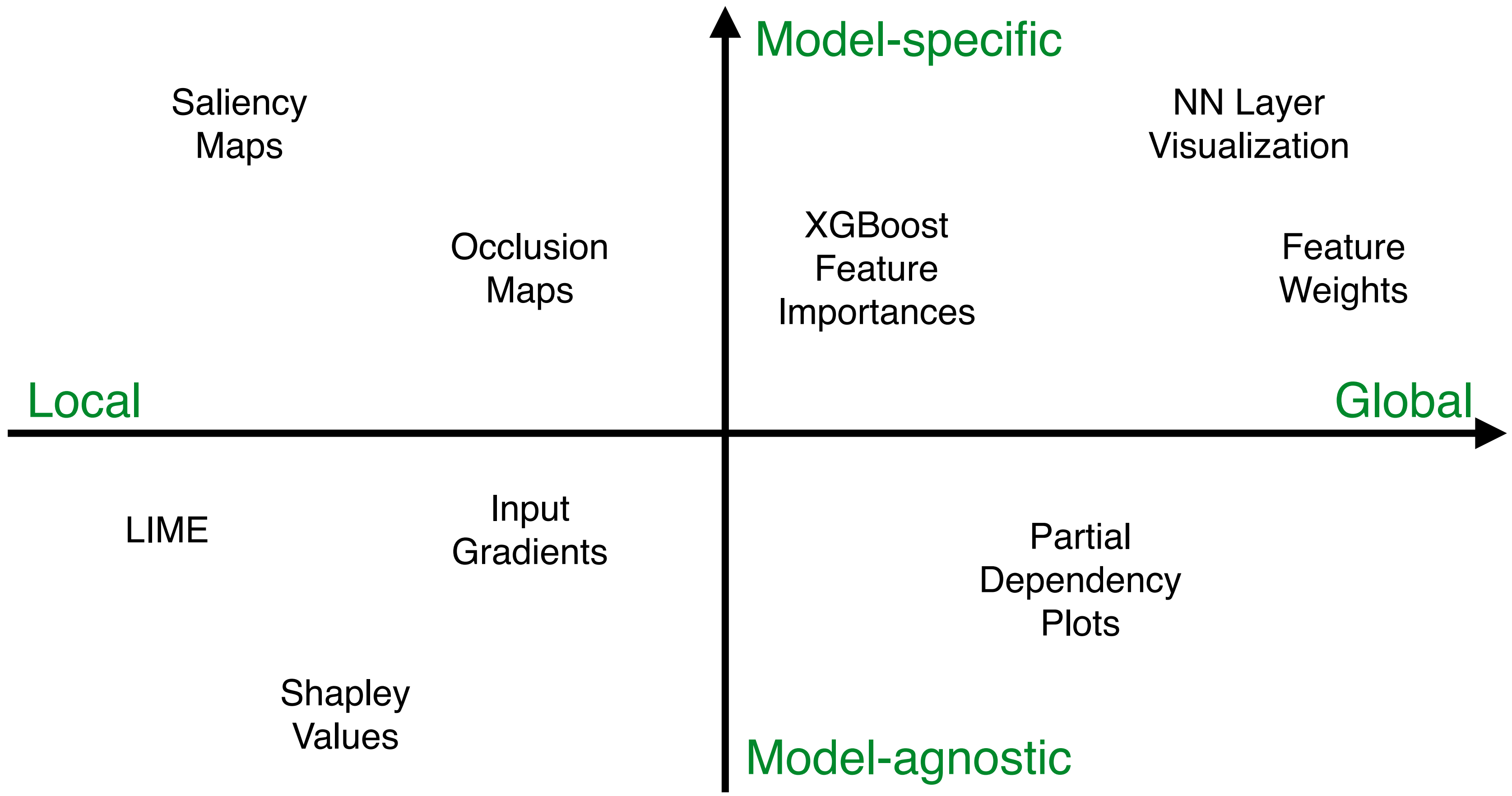
- Understanding why this image contain a lamp

some image of a lamp



why it is classified as a lamp

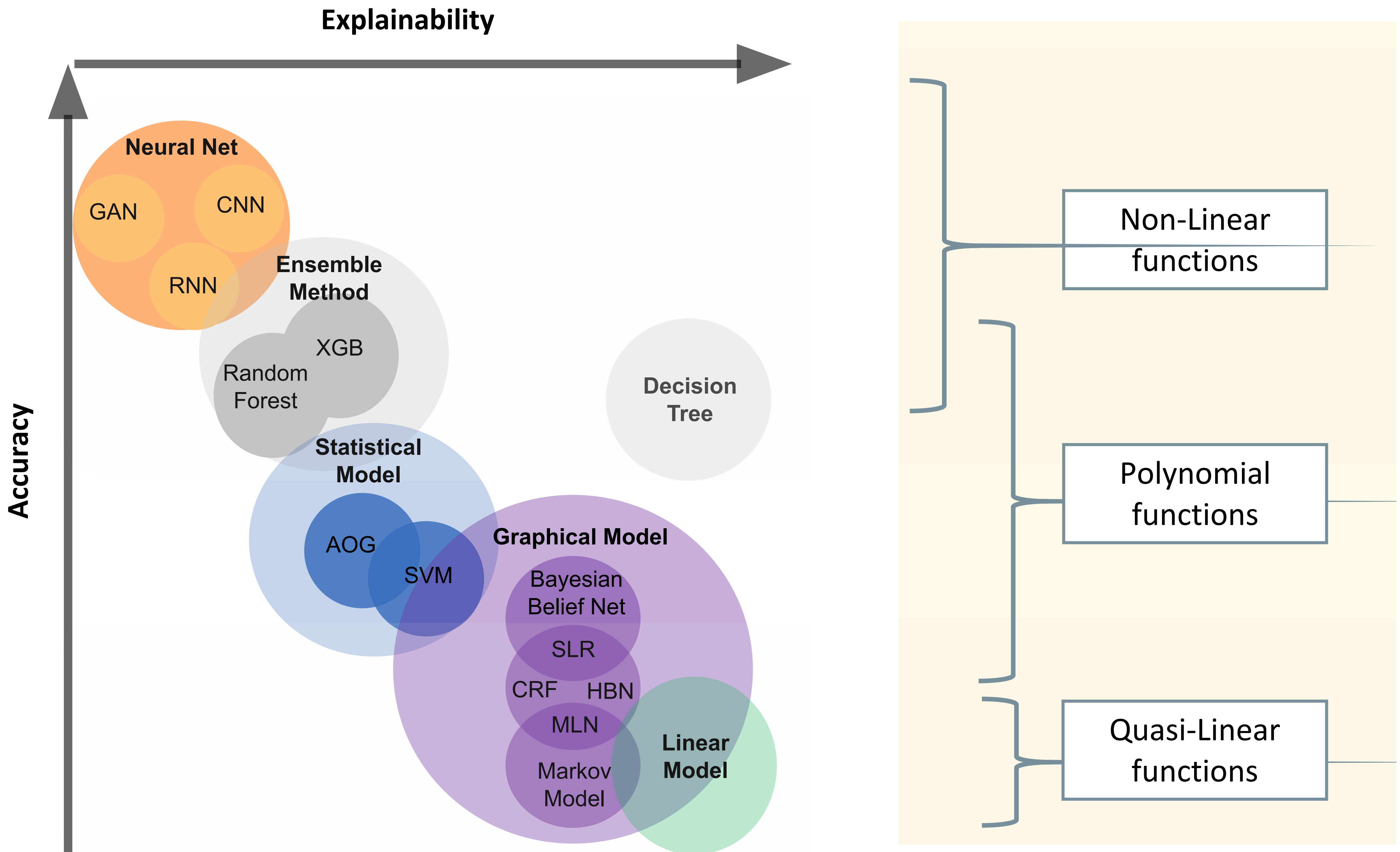
Taxonomy of Interpretability Methods



- Local: interpretation for specific instance
- Global: interpretation for model output

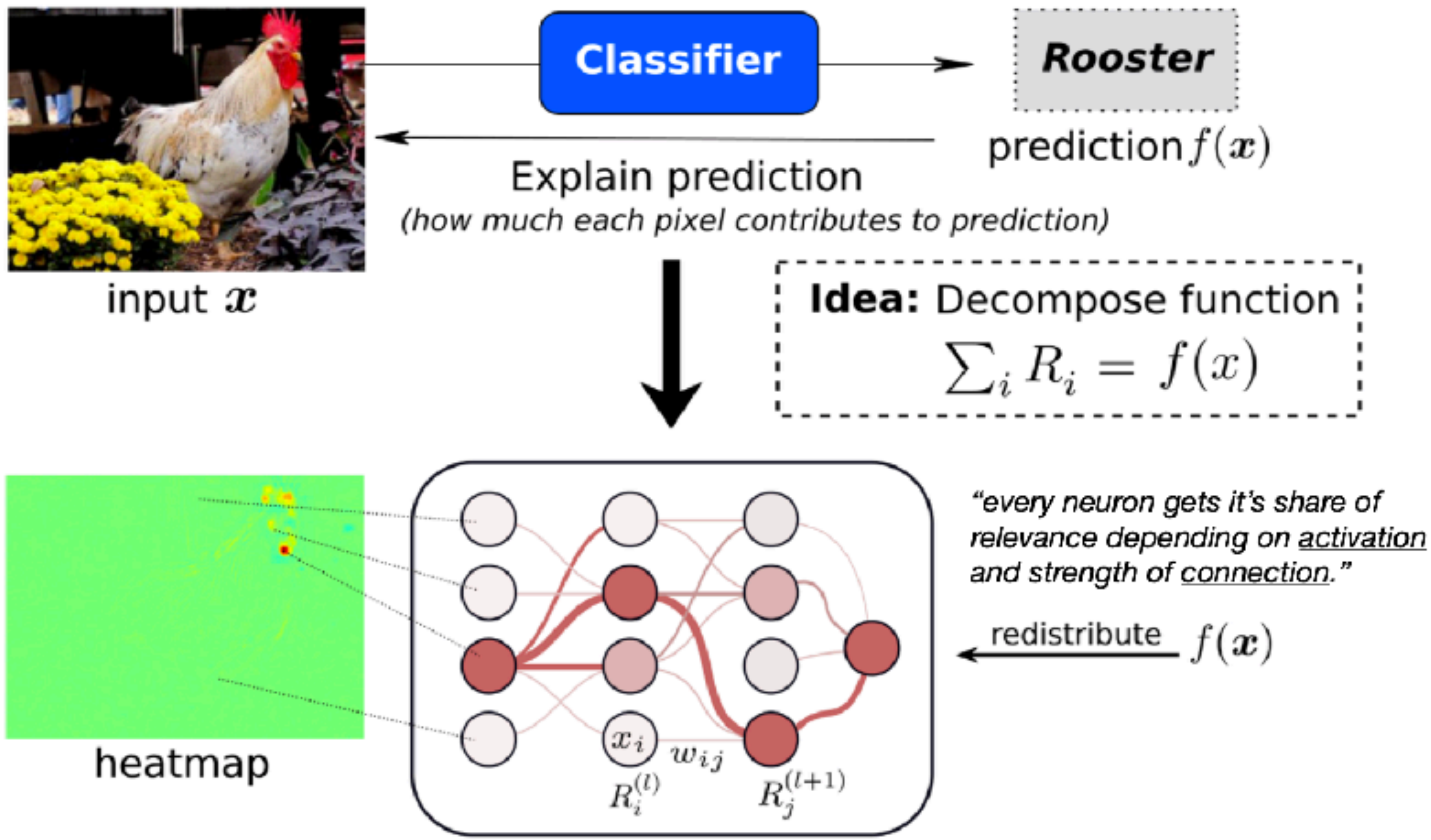
- Model-specific: only for specific model class, access to model internals
- Model-agnostic: for any models, post hoc, analyzing input and output without access to model internals

Accuracy vs. Explainability



Explaining Decision

Layer-wise Relevance Propagation (LRP)
(Bach et al. 2015)



Sensitivity Analysis

Consider a function f, a data point $\boldsymbol{x} = (x_1, \ldots, x_d)$, and the prediction

$$f(x_1, \ldots, x_d).$$

Sensitivity analysis measures the local variation of the function along each input dimension

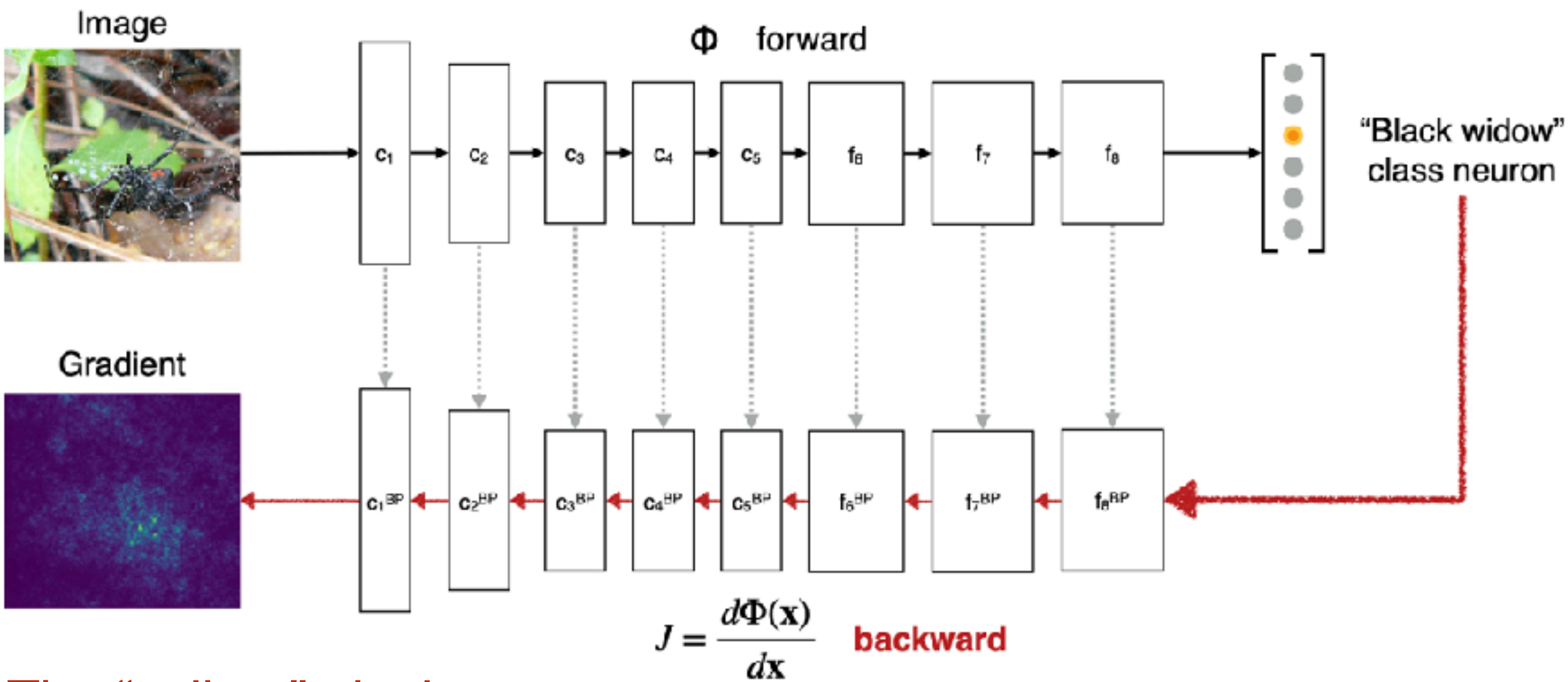
$$R_i = \left(\frac{\partial f}{\partial x_i} \Big|_{x=\boldsymbol{x}} \right)^2$$

Remarks:

- Easy to implement (we only need access to the gradient of the decision function).
- But does it really explain the prediction?

Saliency via Backpropagation

Sensitivity analysis of target neuron w.r.t. input pixels



The "salient" pixels usually light up

Deep inside convolutional networks, Simonyan, Vedaldi, Zisserman, ICLR, 2014

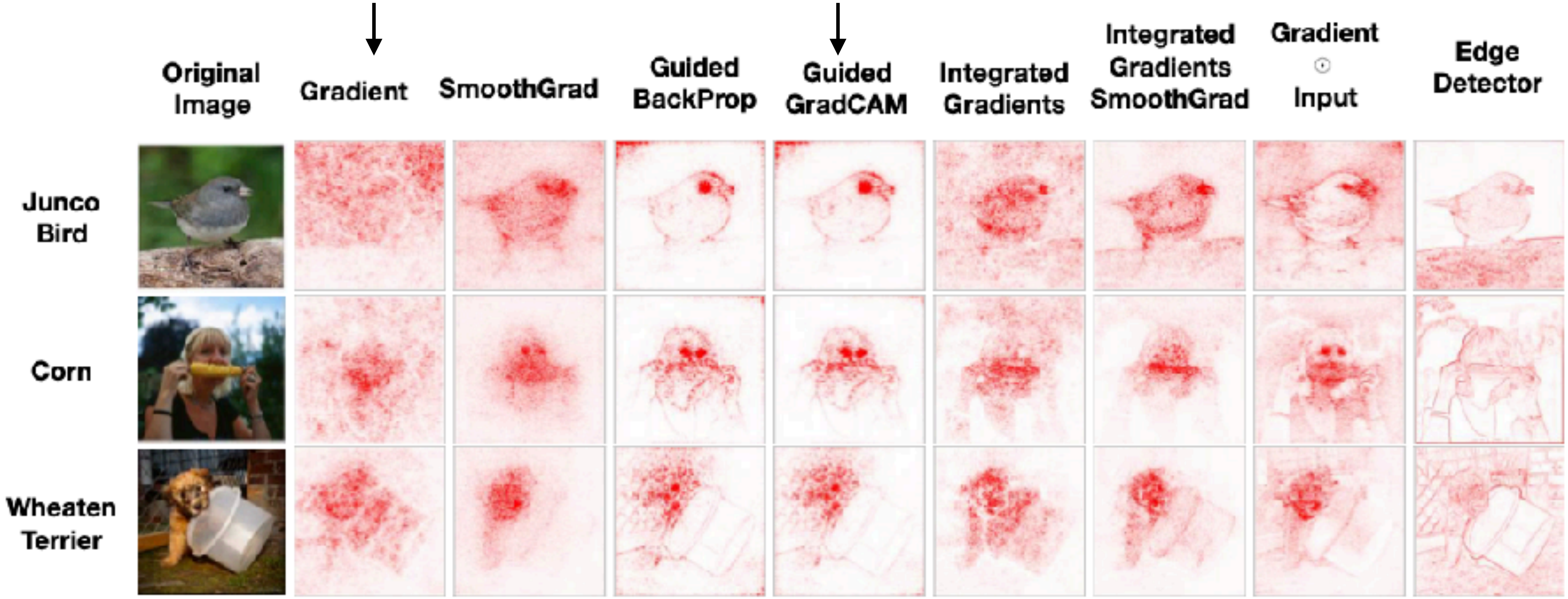
Saliency Map

Saliency maps provide a visual representation of the input sensitivity of an output class

$$\left.\frac{\partial S_c(x)}{\partial x}\right|_{x_0}$$

$$\mathrm{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

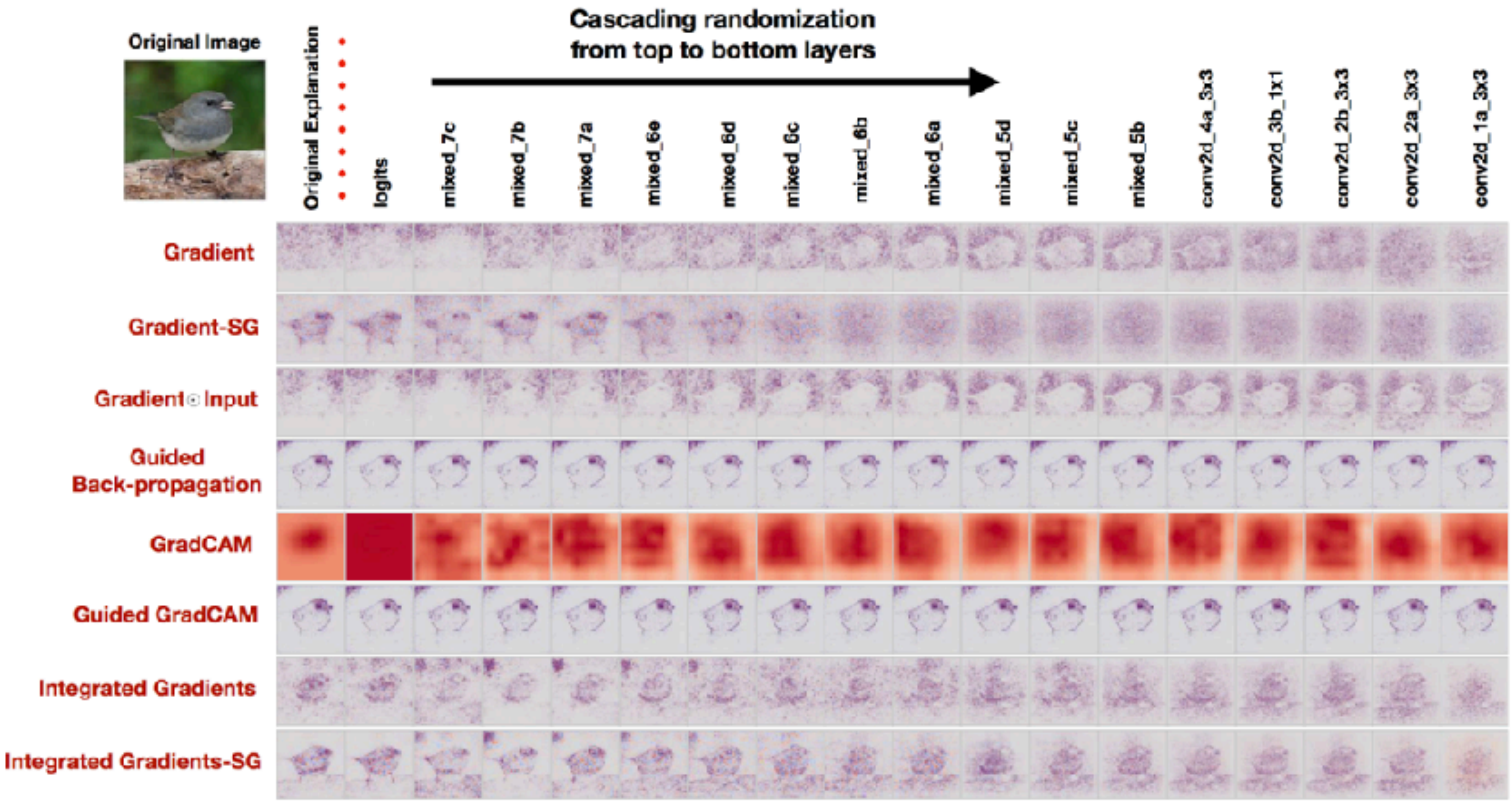
$$\alpha_k^c = \frac{1}{Z}\sum_{i,j}\frac{\partial y^c}{\partial A_{ij}^k}$$



Sanity Checks for Saliency Maps (Adebayo et al., NeurIPS 2018)
Deep Inside Convolutional Networks (Simonyan et al., ICLR 2014)

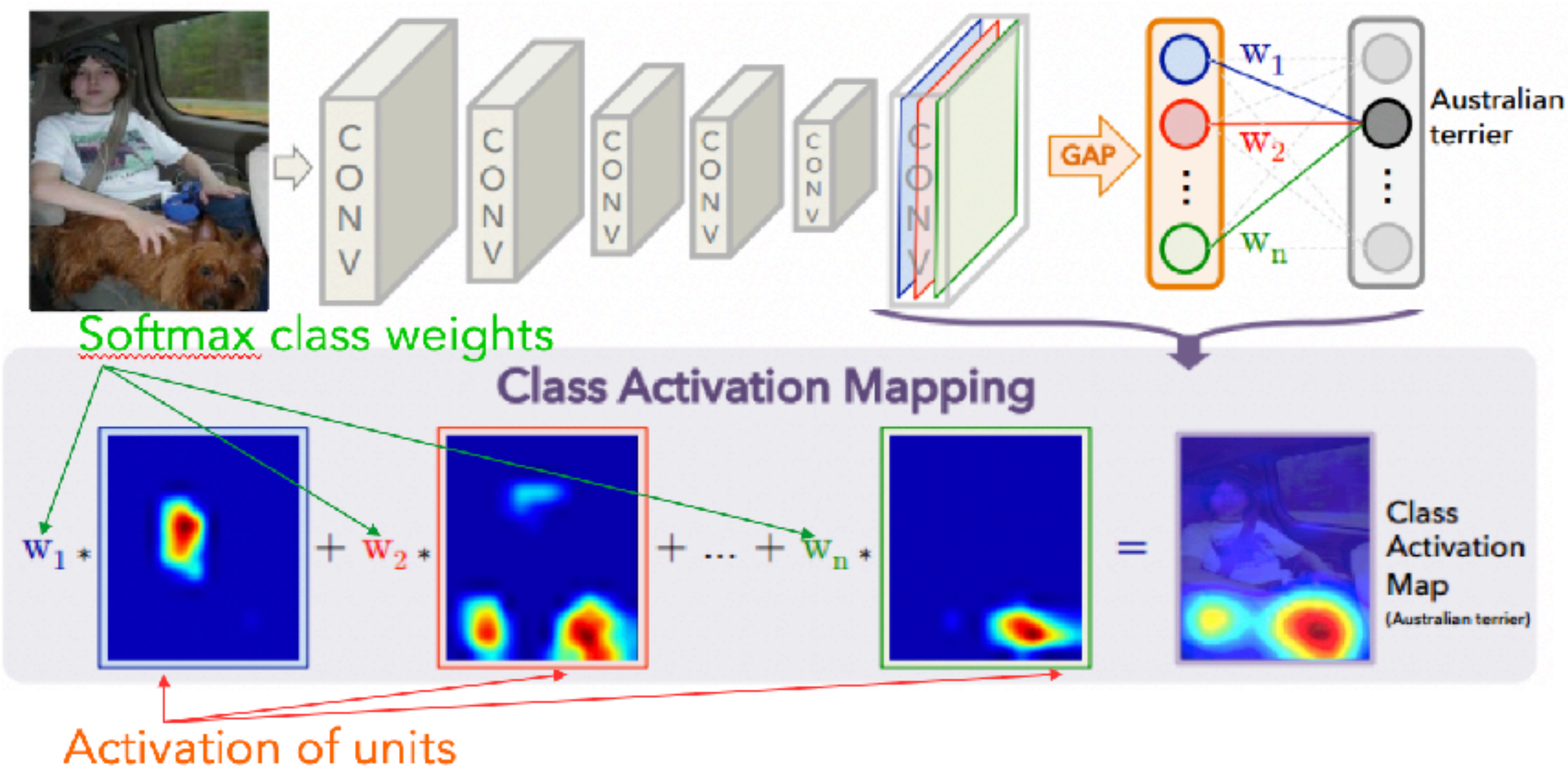
Sanity Check-1

- When randomizing weight, model gives random prediction
- Does saliency map change?

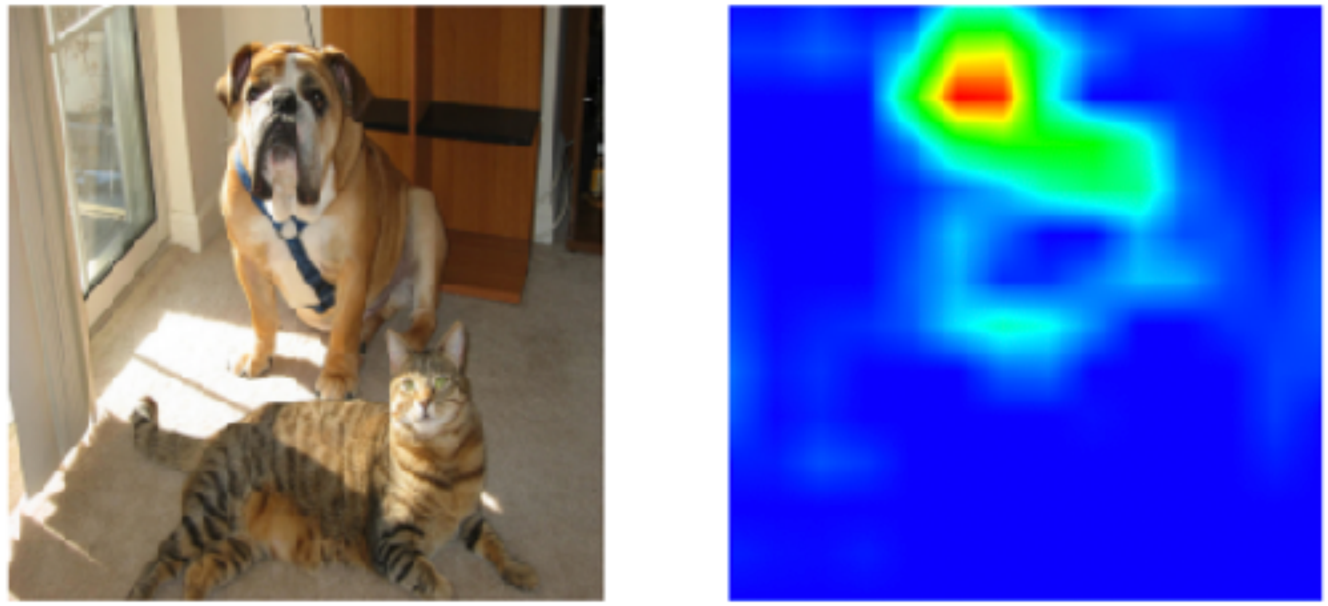


Sanity Checks for Saliency Maps (Adebayo et al. NeurIPS 2018)

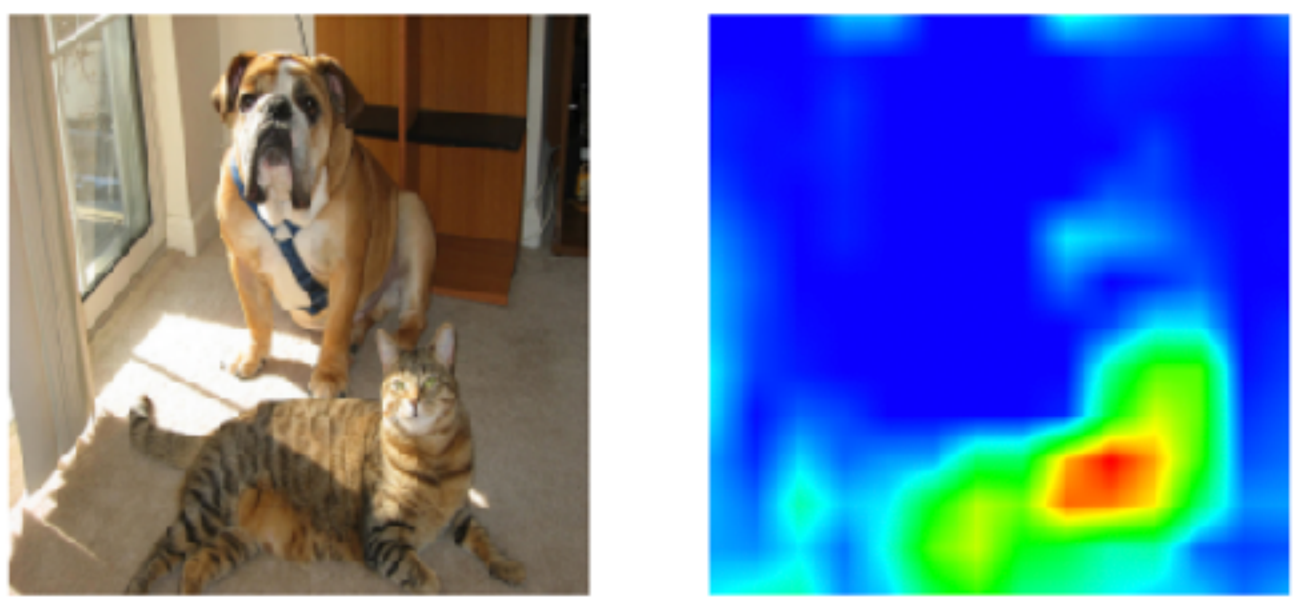
Class Activation Maps (CAM)



Grad-CAM

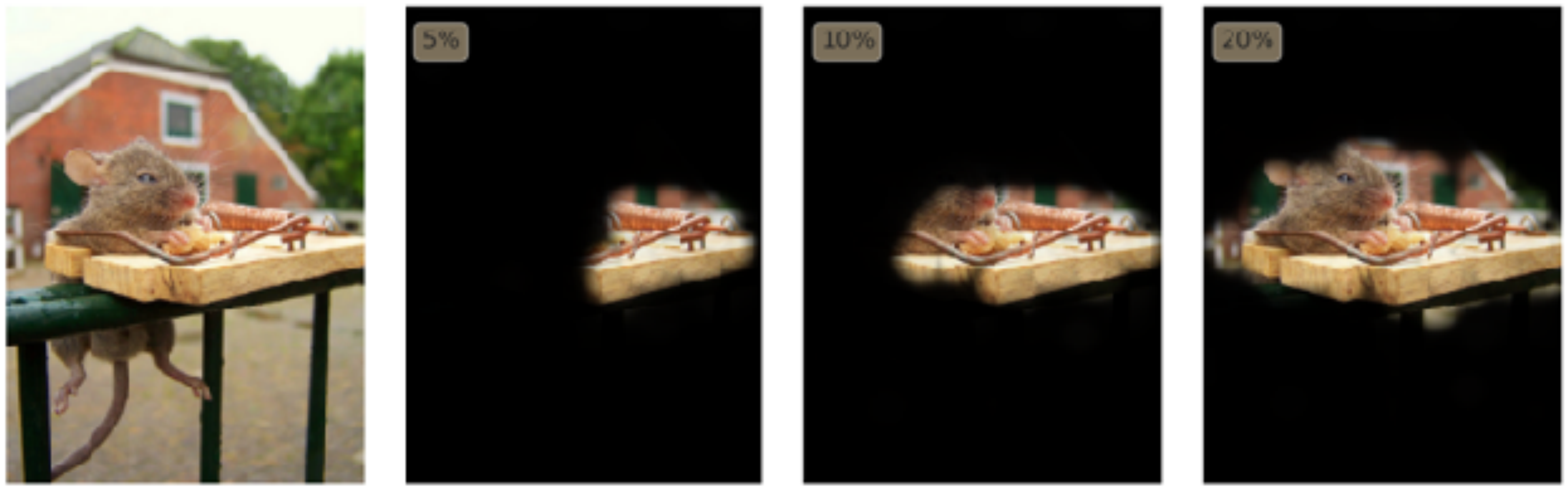


What animal is in this picture? Dog



What animal is in this picture? Cat

Extremal Perturbations

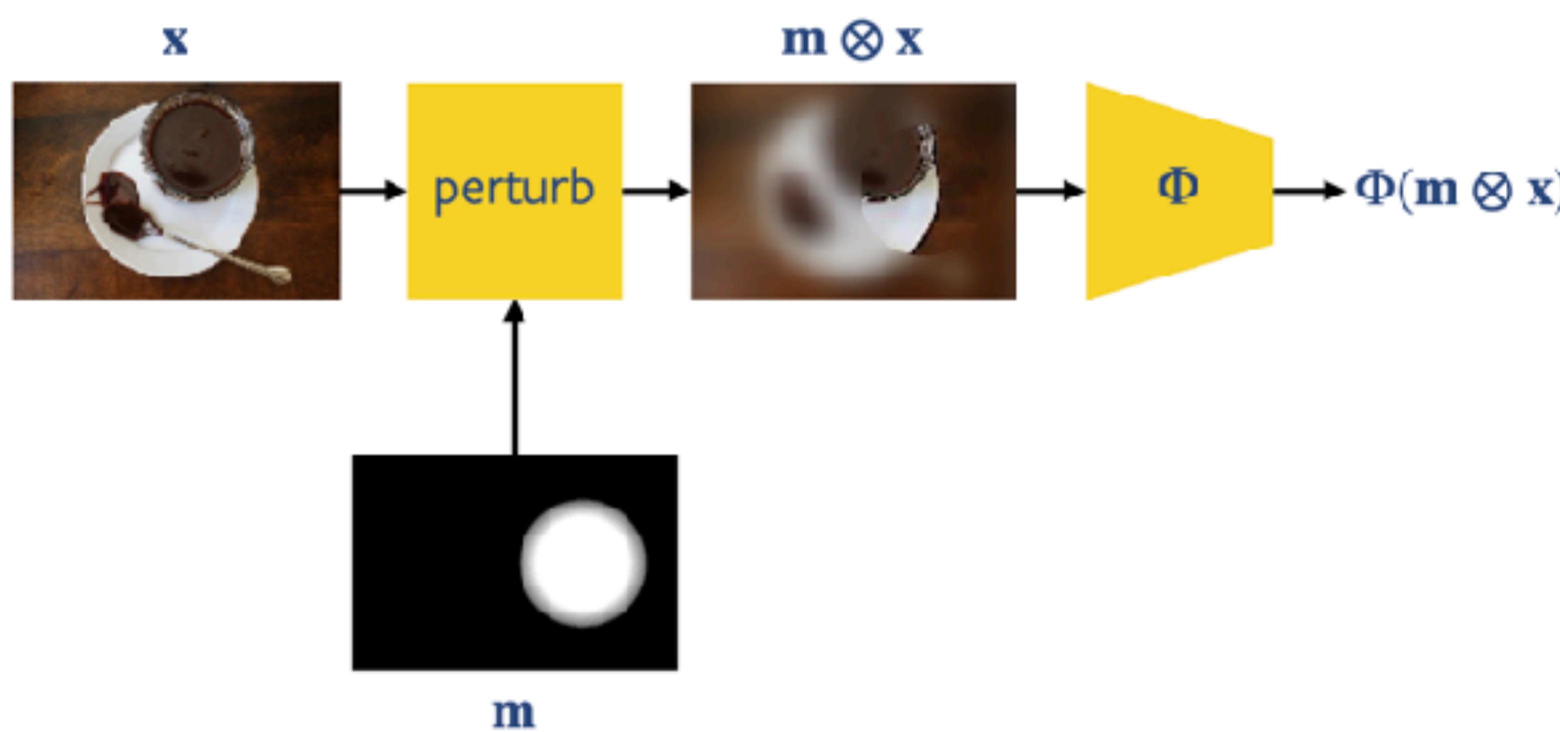


Learn a **fixed-sized** mask **m** to perturb input **x** that maximally **preserves** the network's output

A mask is optimized to maximally excite the network:

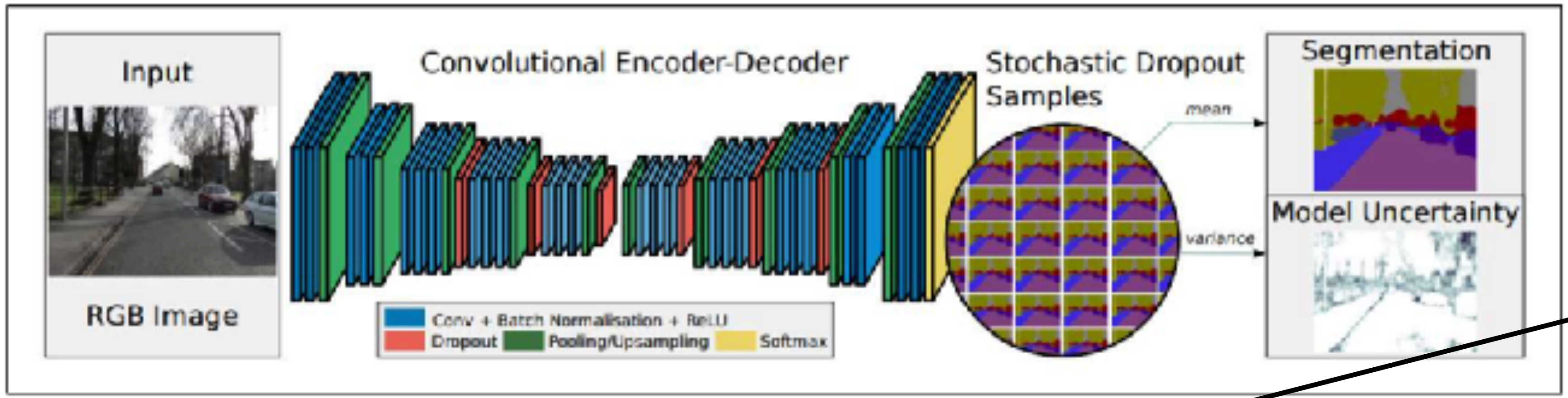
$$\operatorname*{argmax}_{\mathbf{m}} \Phi(\mathbf{m} \otimes \mathbf{x})$$

subject to $\text{area}(\mathbf{m}) = a$

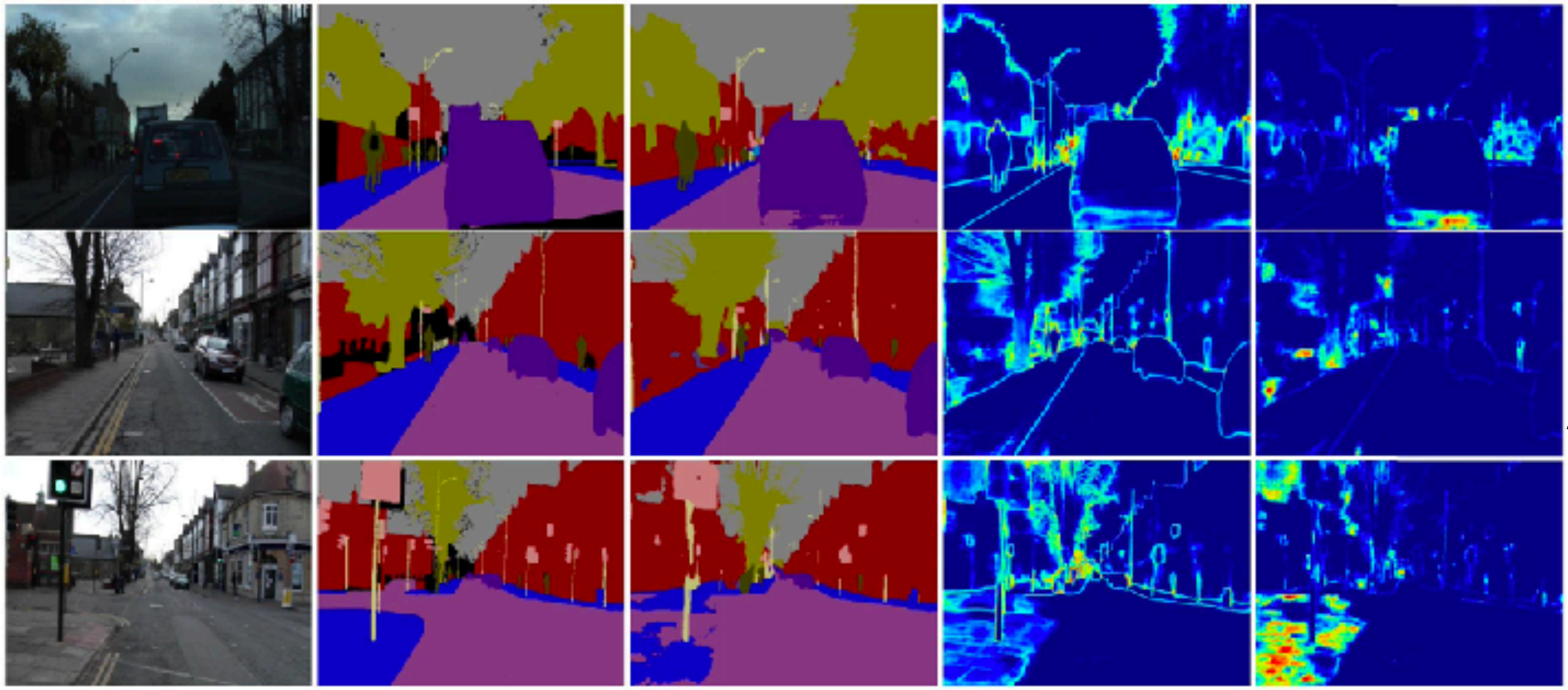


[Fong et al., ICCV 2019]

Uncertainty Map



Sensing uncertainty



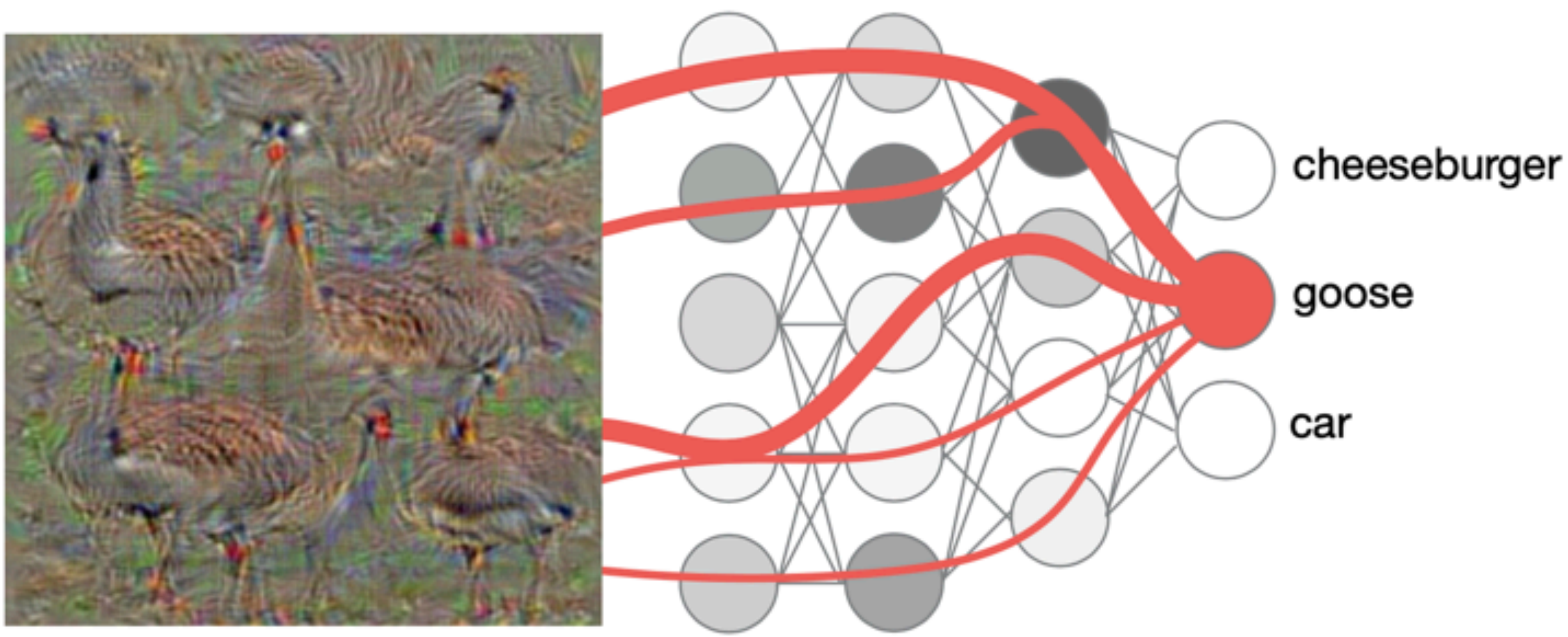
Modeling uncertainty

(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?
(Kendall et al. NeurIPS 2017)

Interpreting Model

- Find prototypical example of a category
- Find pattern maximizing activity of a neuron



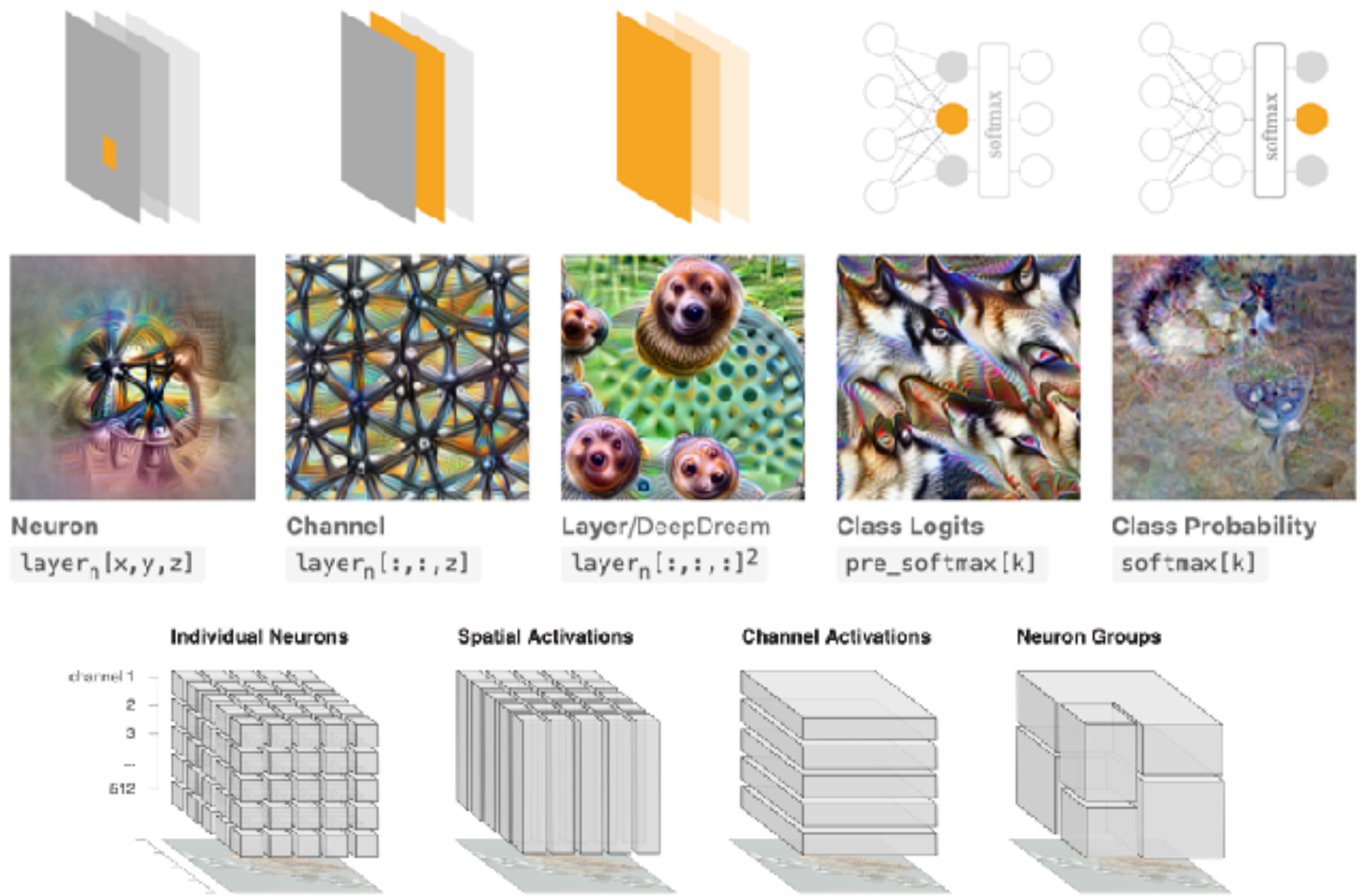
simple regularizer
(Simonyan et al. 2013)

$$\max_{x \in \mathcal{X}} p_\theta(\omega_c \,|\, x) + \lambda \Omega(x)$$

Activation Maximization

Visualize the exemplar of class (output layer) or representation (hidden layer) by optimization w.r.t. input

$$\max_x h^l_{i,j,c}(x) \qquad\qquad \max_x S_c(x) - \lambda R(x)$$

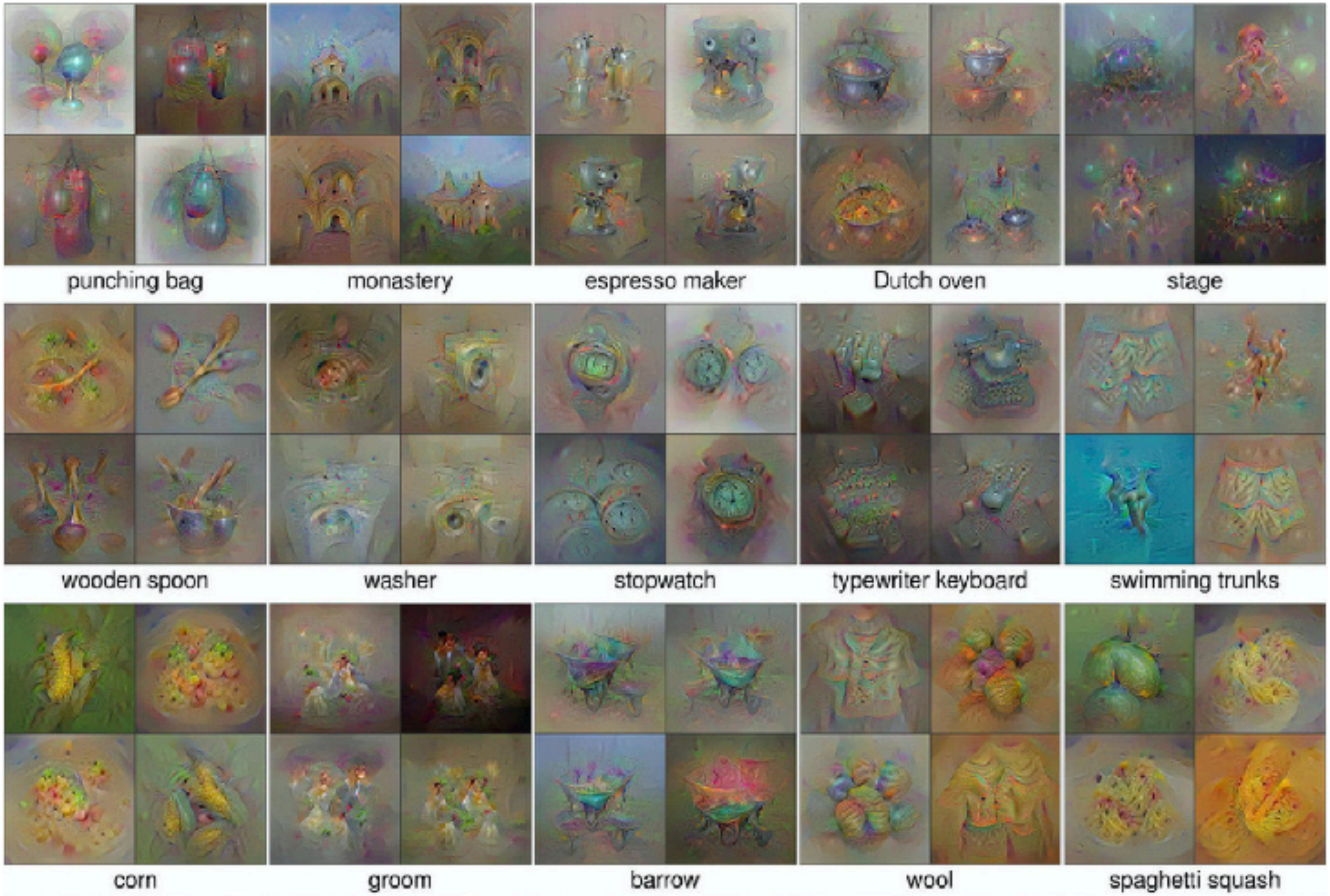


https://distill.pub/2017/feature-visualization/; https://distill.pub/2018/building-blocks/

Multifaceted Feature Visualization

Class maximization w.r.t. inputs

$$\max_x S_c(x) - \lambda R(x)$$



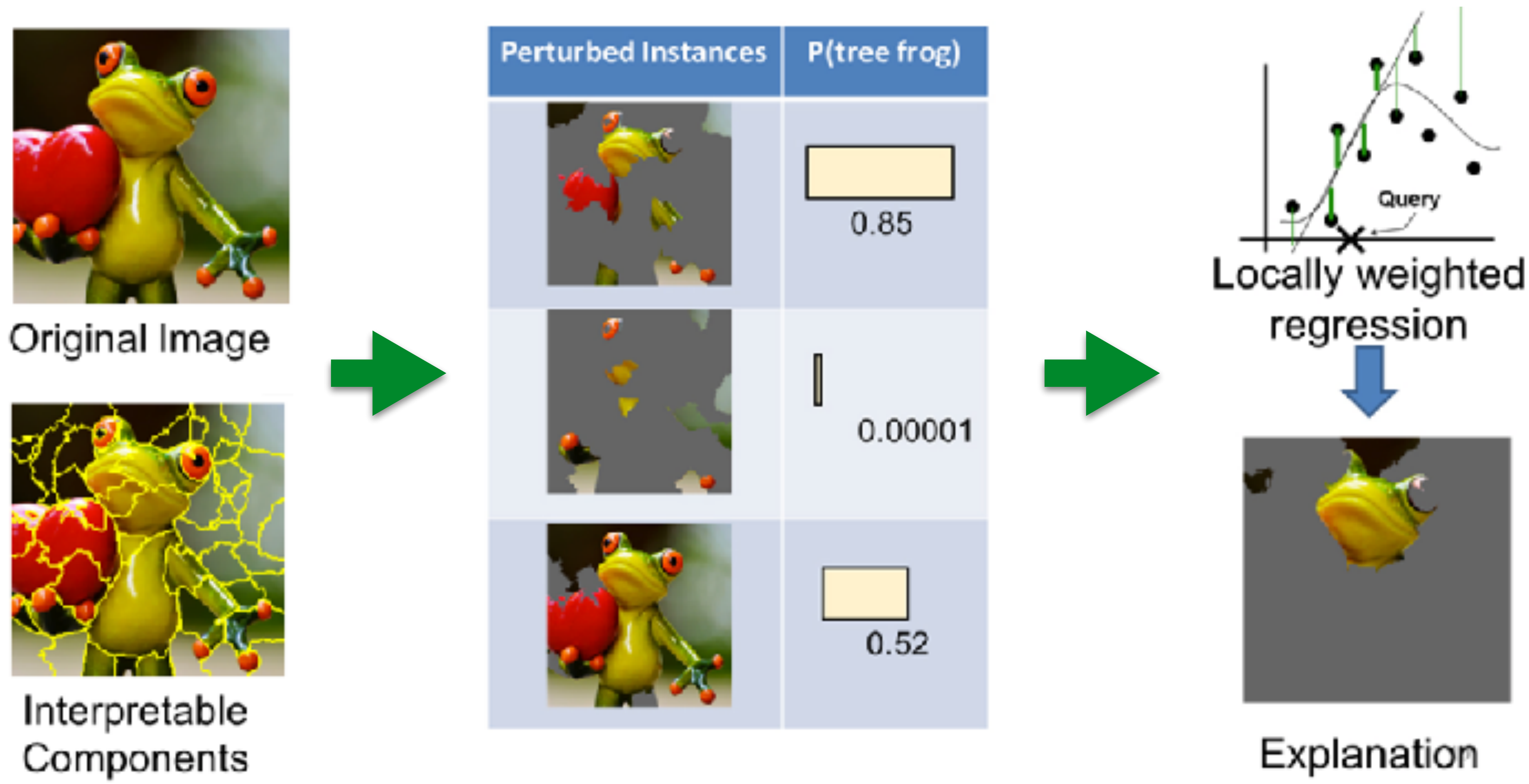
Multifaceted Feature Visualization: (Nguyen et al. ICML 2016 Best Paper Award)

Activation Maximization

https://distill.pub/2017/feature-visualization/；https://distill.pub/2018/building-blocks/

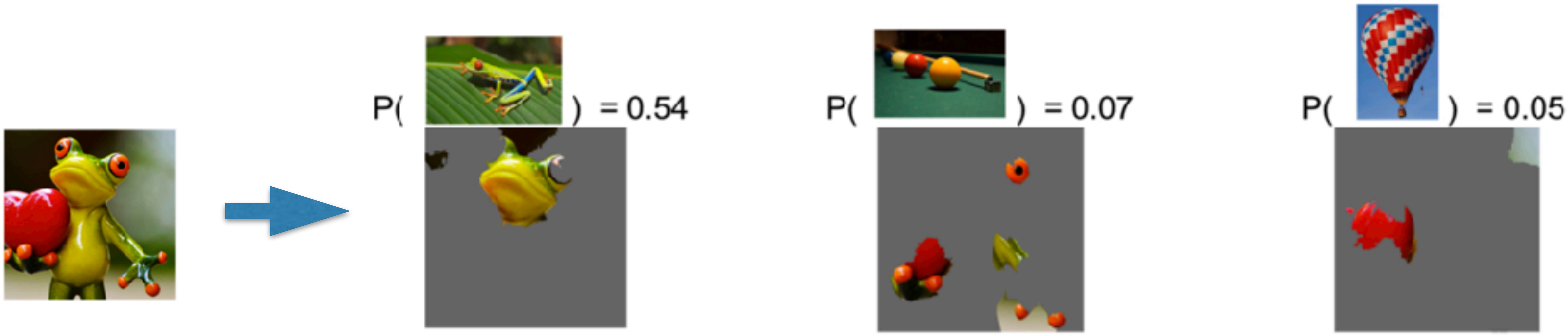
LIME (Local Interpretable Model-Agnostic Explanations)

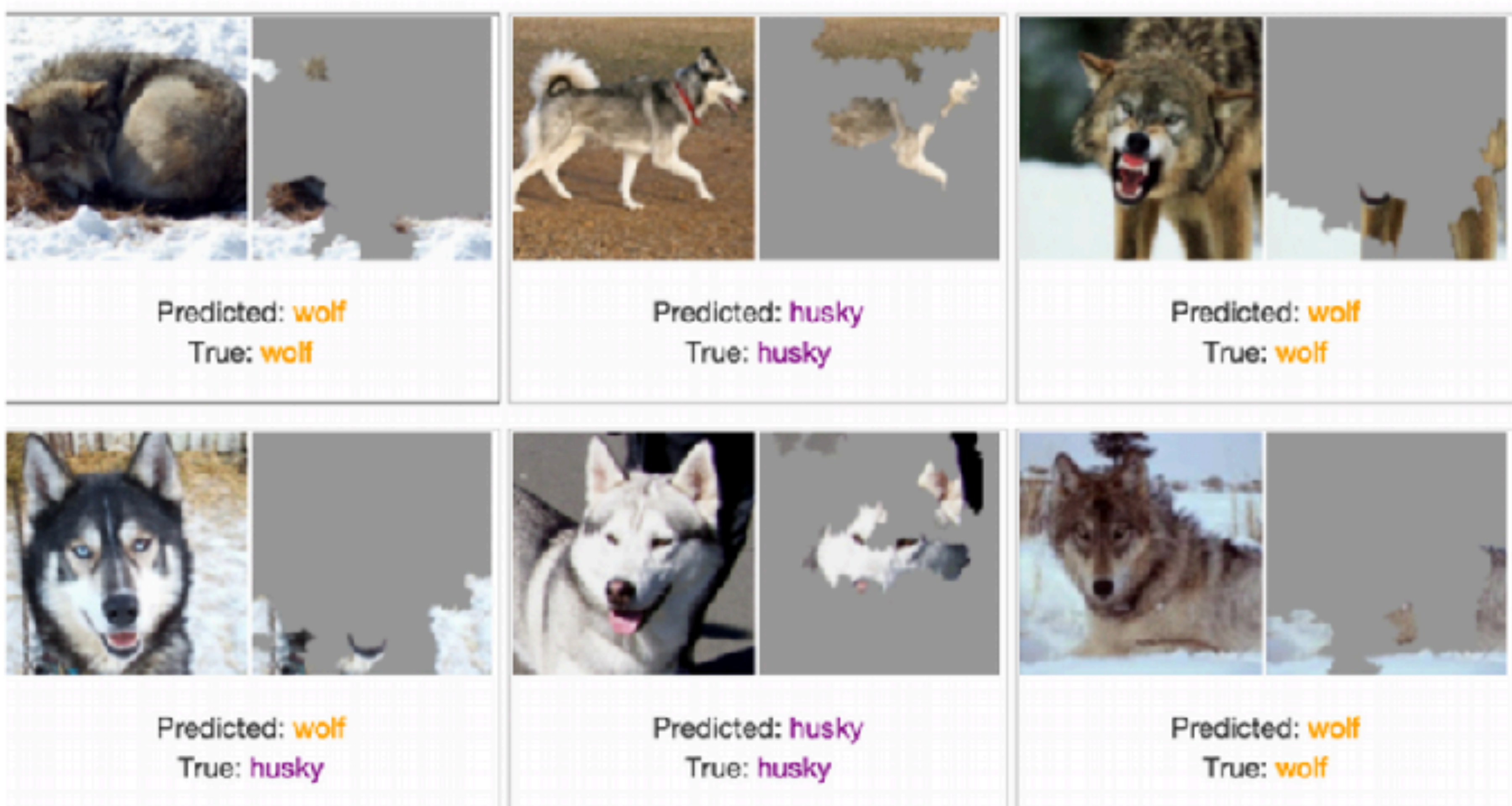
- Surrogate models are trained to approximate the predictions of the underlying black box model (model-agnostic approach)
- Explain the decision by evidence of interpretable region



“Why Should I Trust You?” Explaining the Predictions of Any Classifier (Ribeiro et al. KDD 2016)
Model-Agnostic Interpretability of Machine Learning (Ribeiro et al. AAAI 2018)

LIME: More Examples





Influence Functions

- Influence of model's prediction by training points
- Identify the training points "responsible" for a given prediction
- How predictions change if removing a training point z?

$$\mathcal{I}(z, z_{\text{test}}) = -\nabla_\theta \mathcal{L}(z_{\text{test}}, \theta)^T H_\theta^{-1} \nabla_\theta \mathcal{L}(z, \theta)$$

Hessian $$H_\theta = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 \mathcal{L}(z_i, \theta)$$

- How predictions change if a training point z is modified?

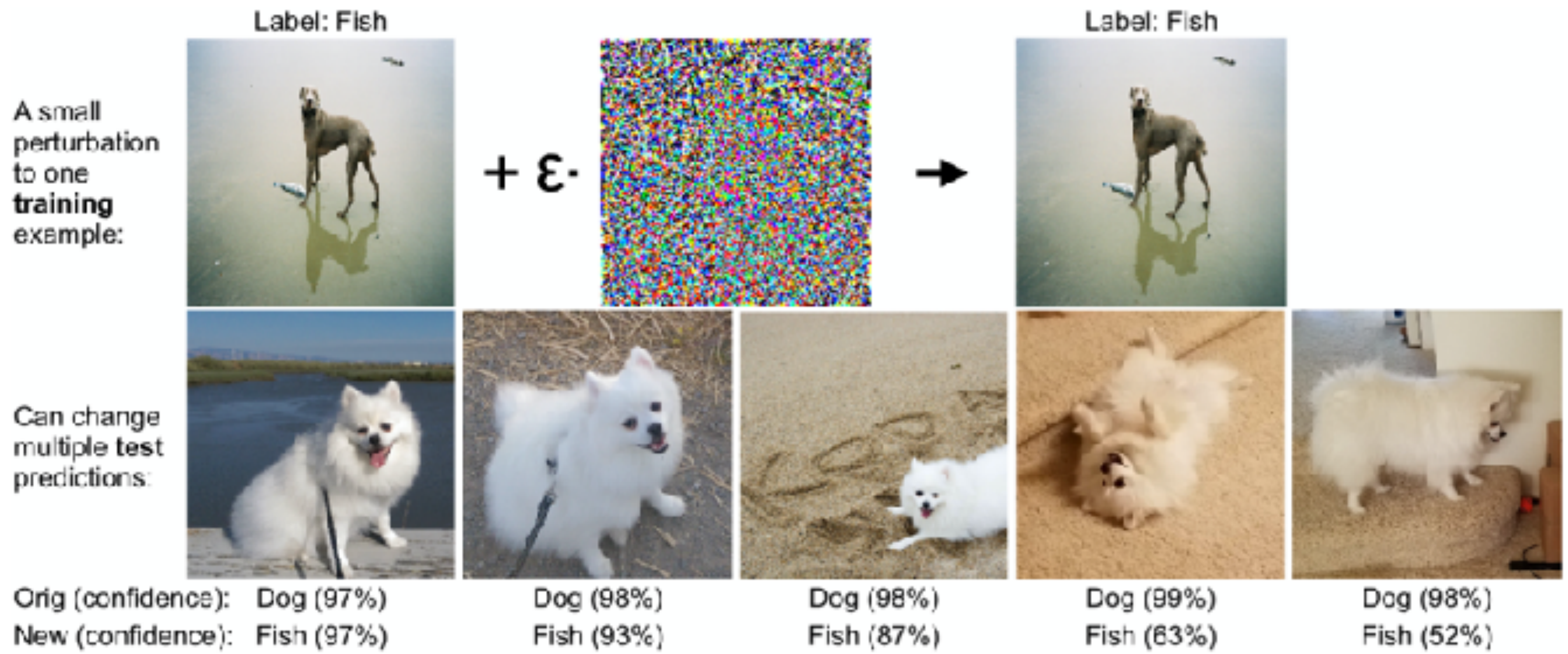
$$\mathcal{I}(z, z_{\text{test}}) = -\nabla_\theta \mathcal{L}(z_{\text{test}}, \theta)^T H_\theta^{-1} \nabla_x \nabla_\theta \mathcal{L}(z, \theta)$$

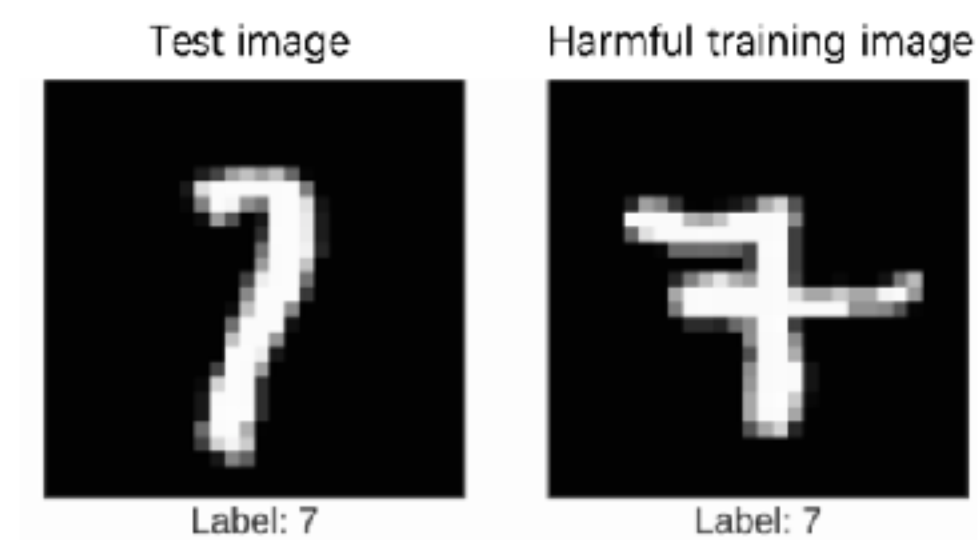
$$z \mapsto z + \delta, \quad \nabla_\delta \mathcal{L}(z_{\text{test}}, \theta') = \mathcal{L}(z, z_{\text{test}})^T \delta$$

- Poising attack

Understanding Black-box Predictions via Influence Functions (Koh and Liang, ICML 2017)

Influence Functions

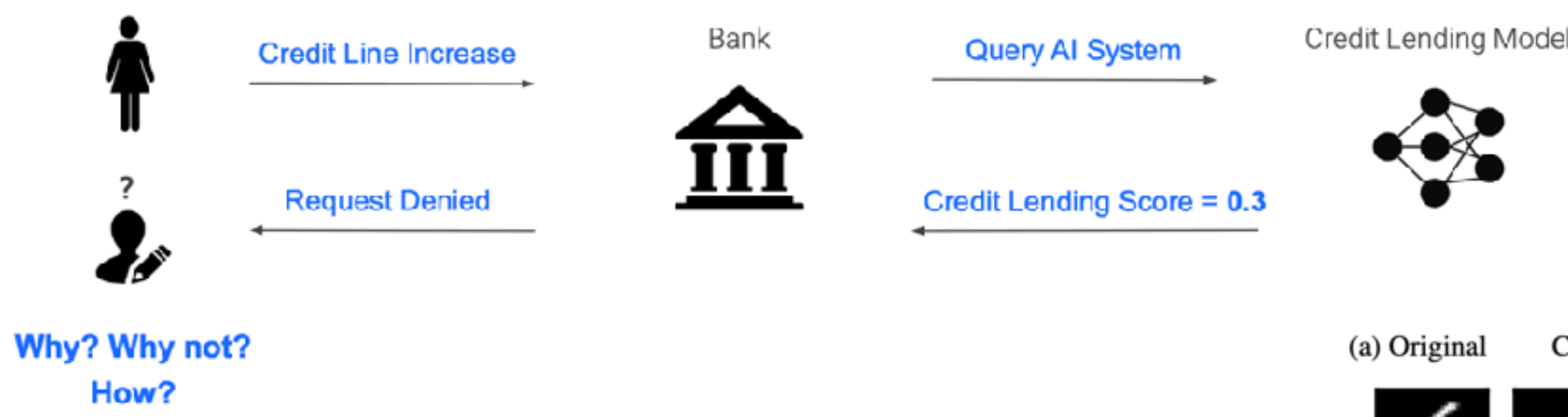


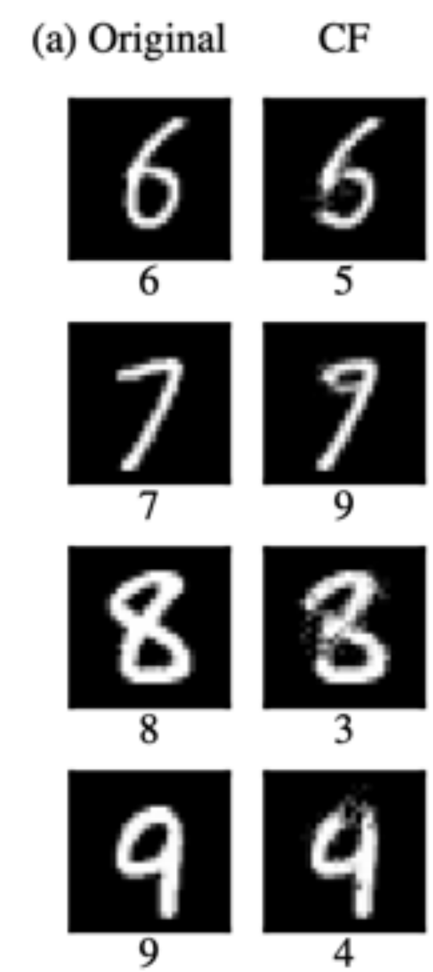


Understanding Black-box Predictions via Influence Functions (Koh and Liang, ICML 2017)

Counterfactual Explanations

Credit Evaluation





- What do I need to change for the bank to approve my loan?
- Which symptoms would lead to a different medical diagnosis?

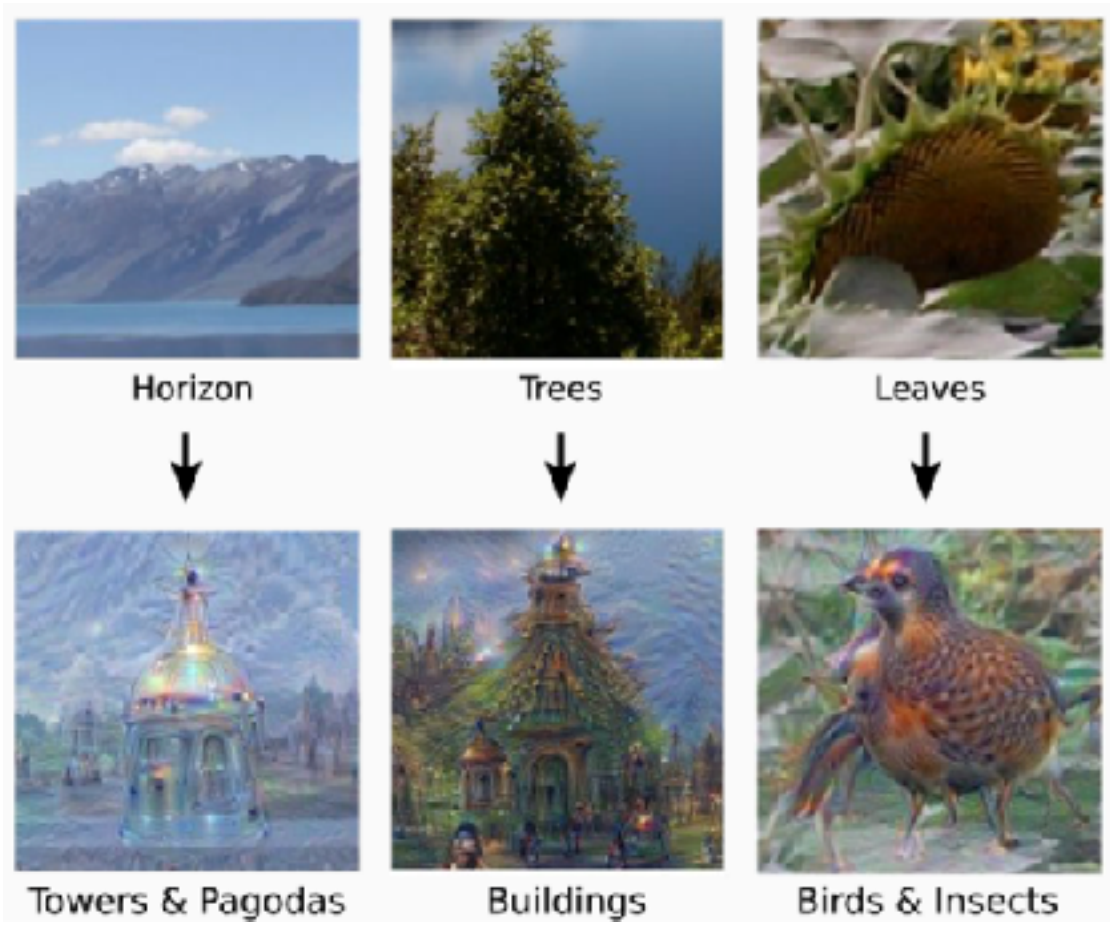
$$\min_{x'} \max_{\lambda} \lambda (f_\theta(x') - y')^2 + d(x_0, x')$$

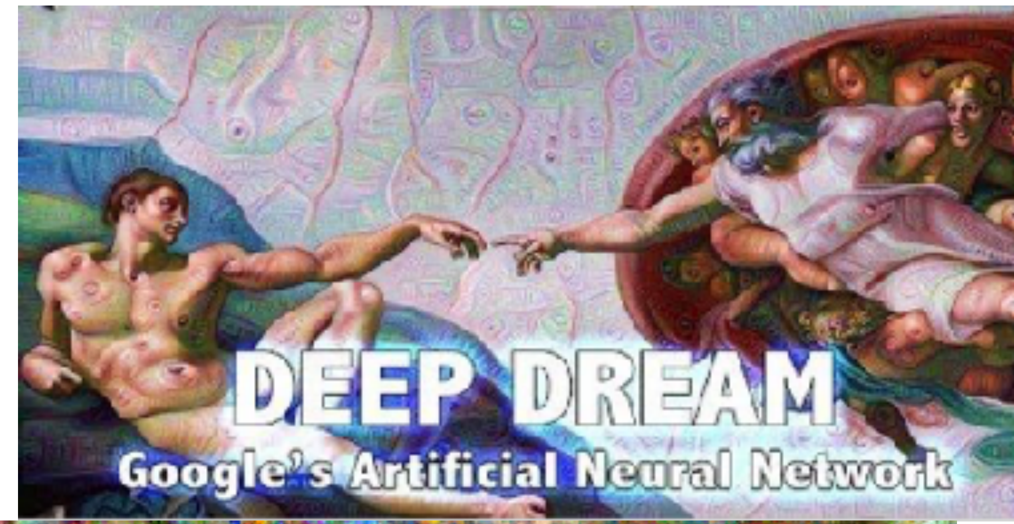
$$d(x_0, x') = \|x_0 - x'\|_1$$

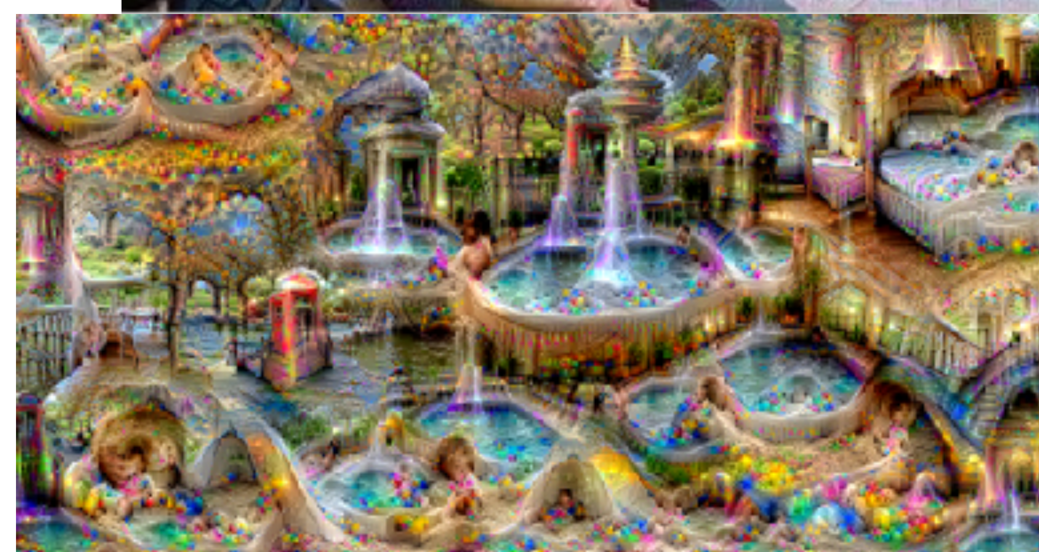
- Adversarial example with sparsity of perturbations

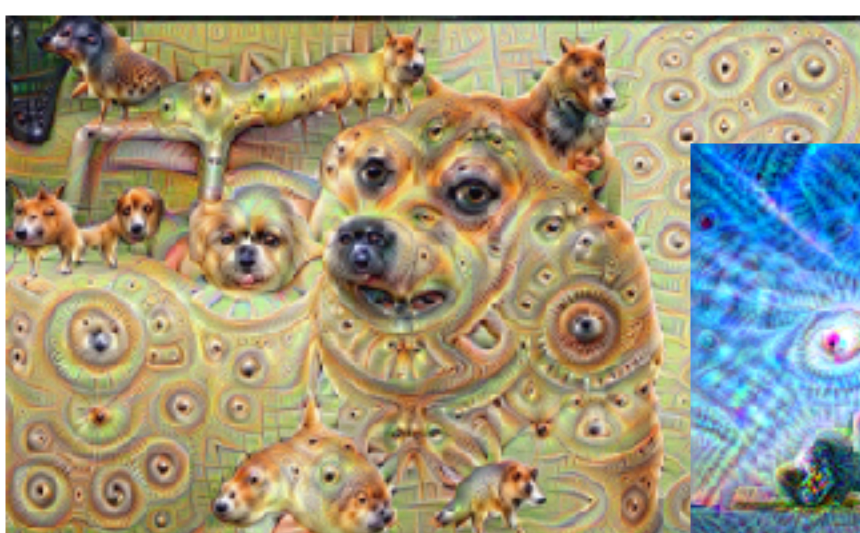
Counterfactual explanations without opening the black box (Wachter et al. 2017)

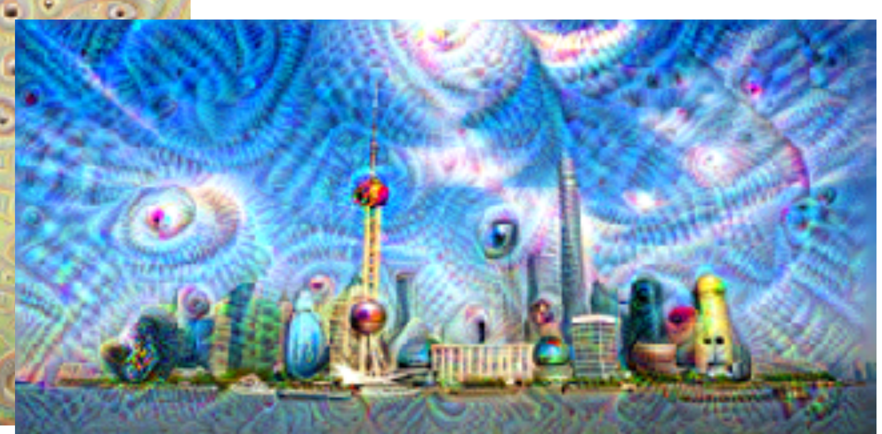
Is Google's DeepDream Art?



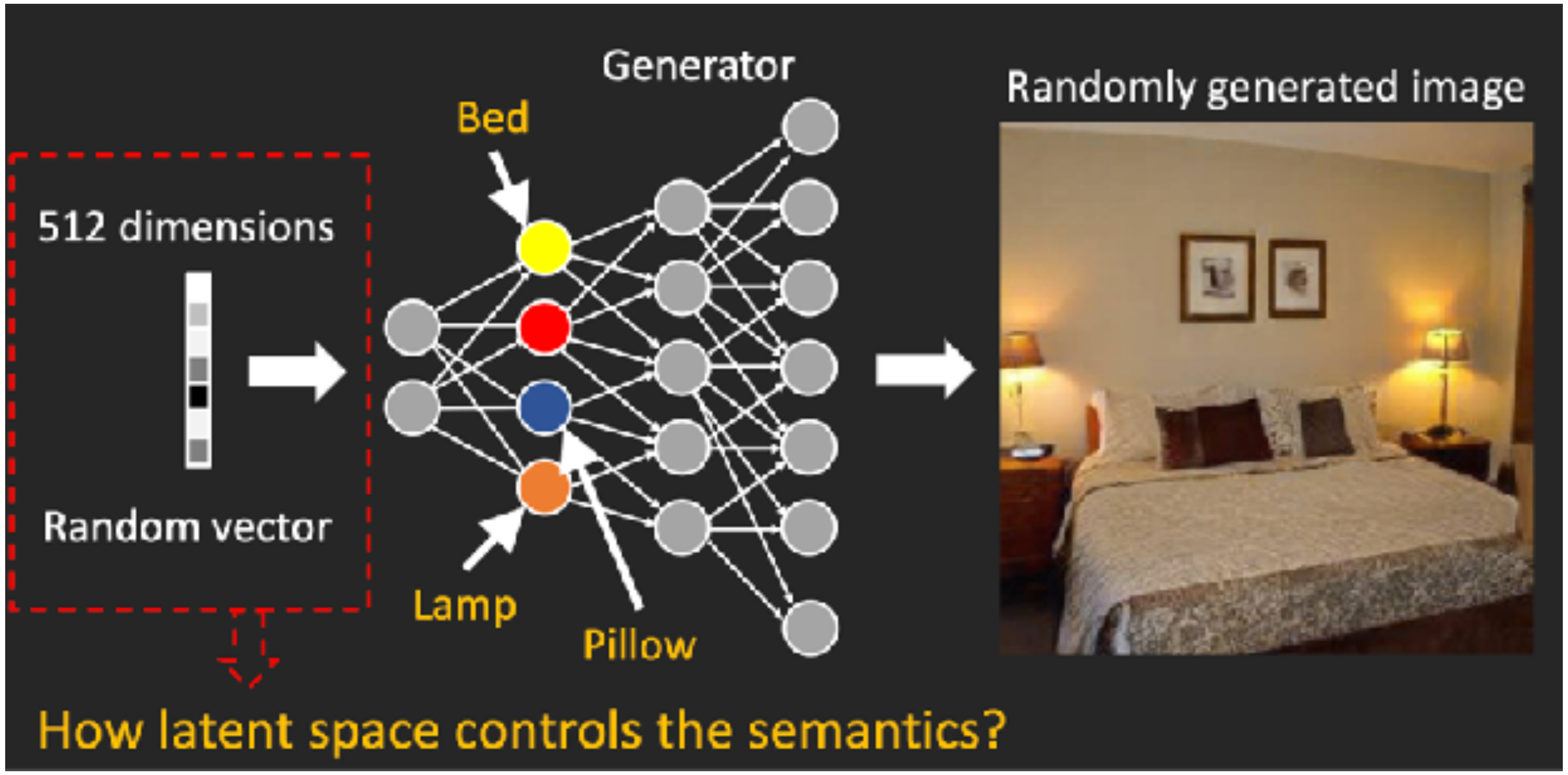


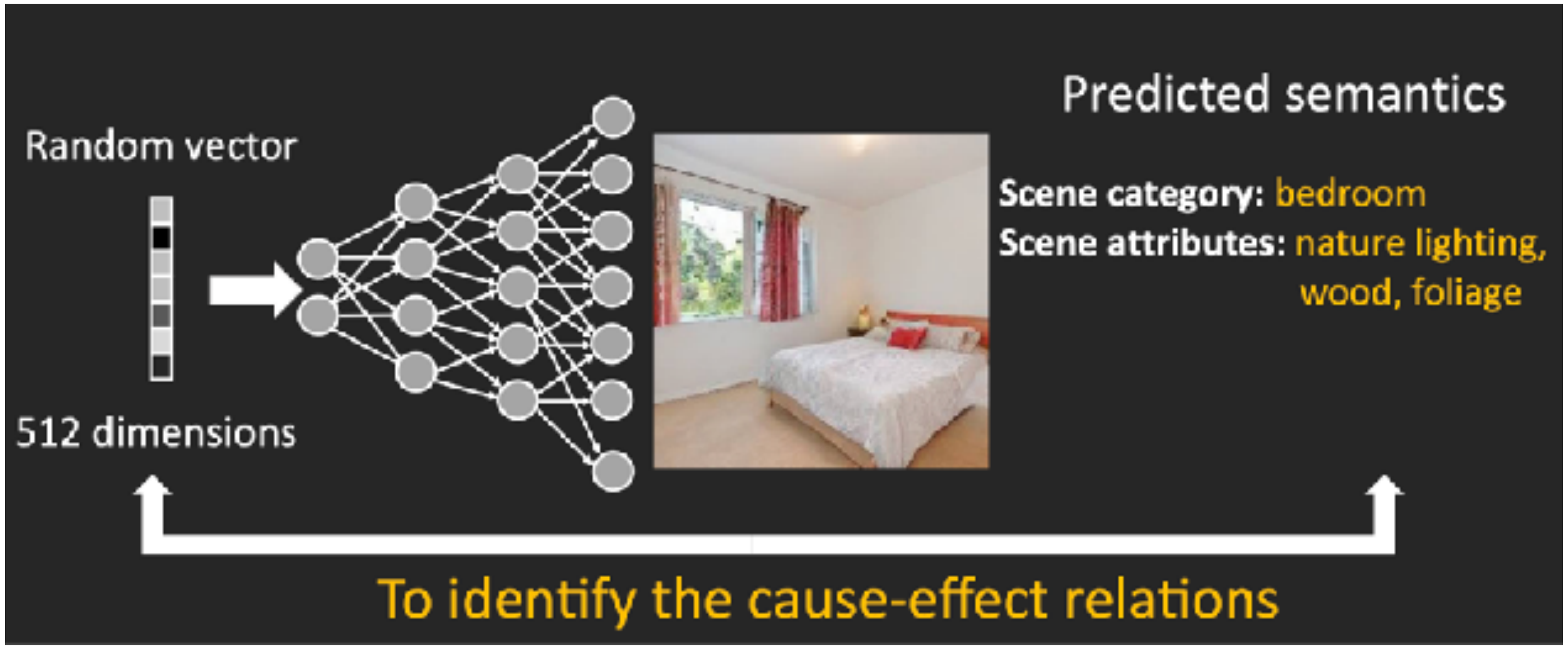






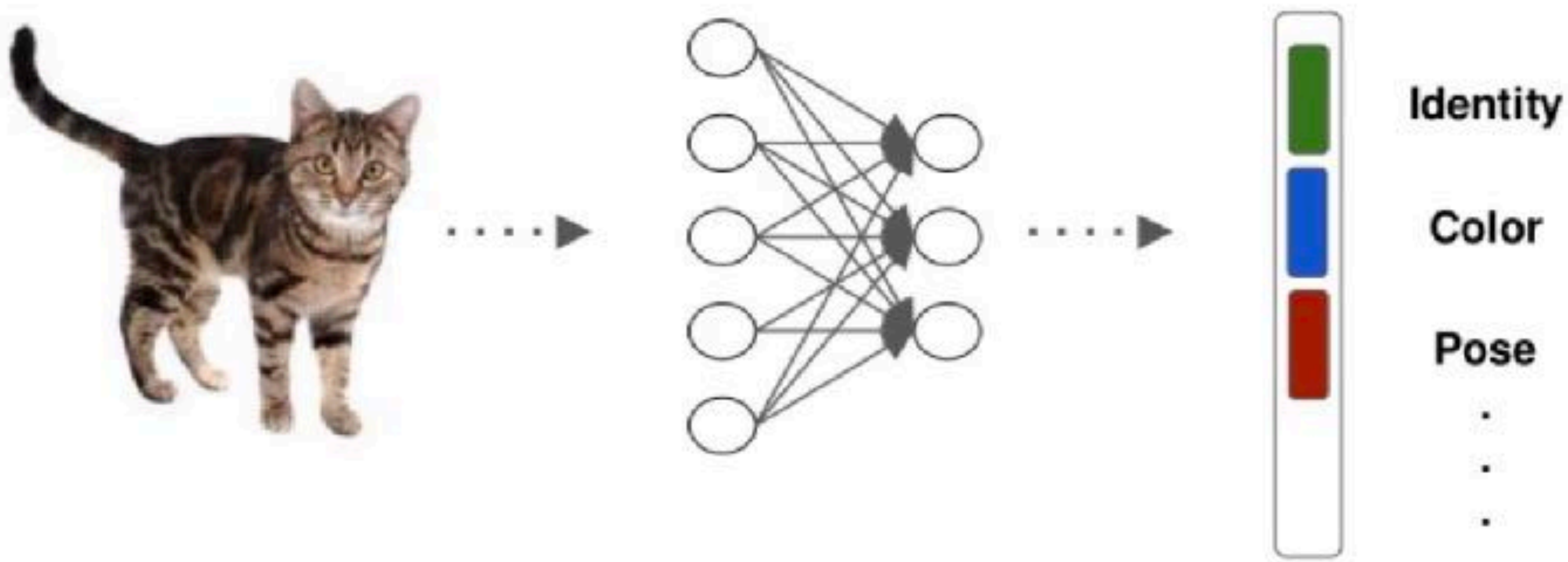
Deep Generative Representation





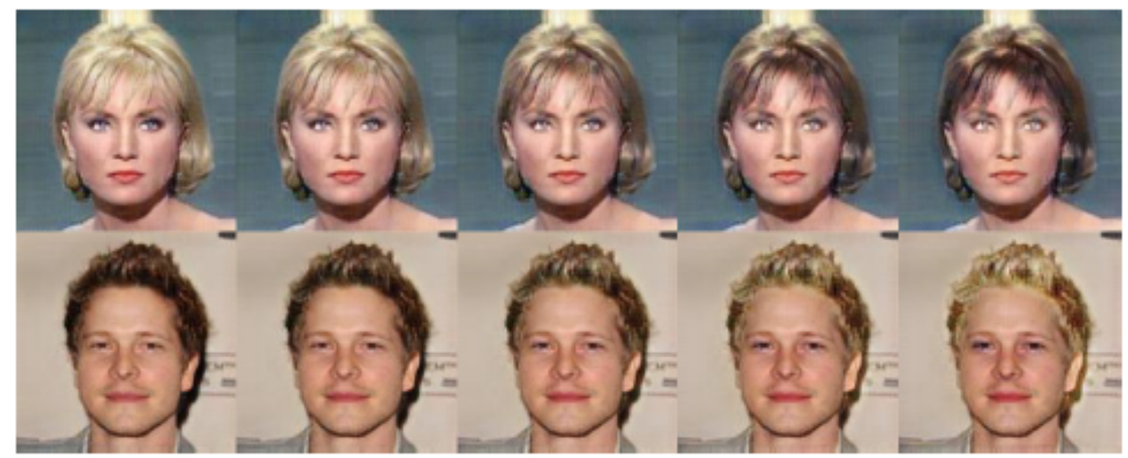
Disentangled Representations

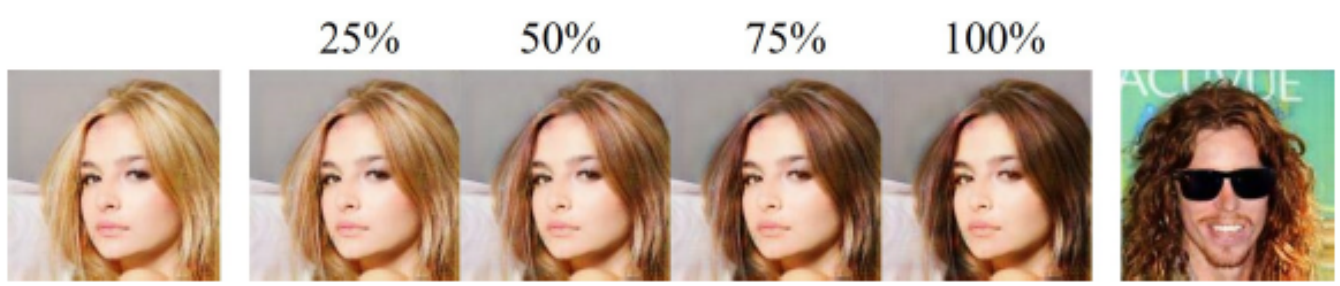
- Factorize distribution over the latent variables
  - Single change in factor should lead to single change representations



Application: Image Translation

- Image resynthesis by manipulating latent factors

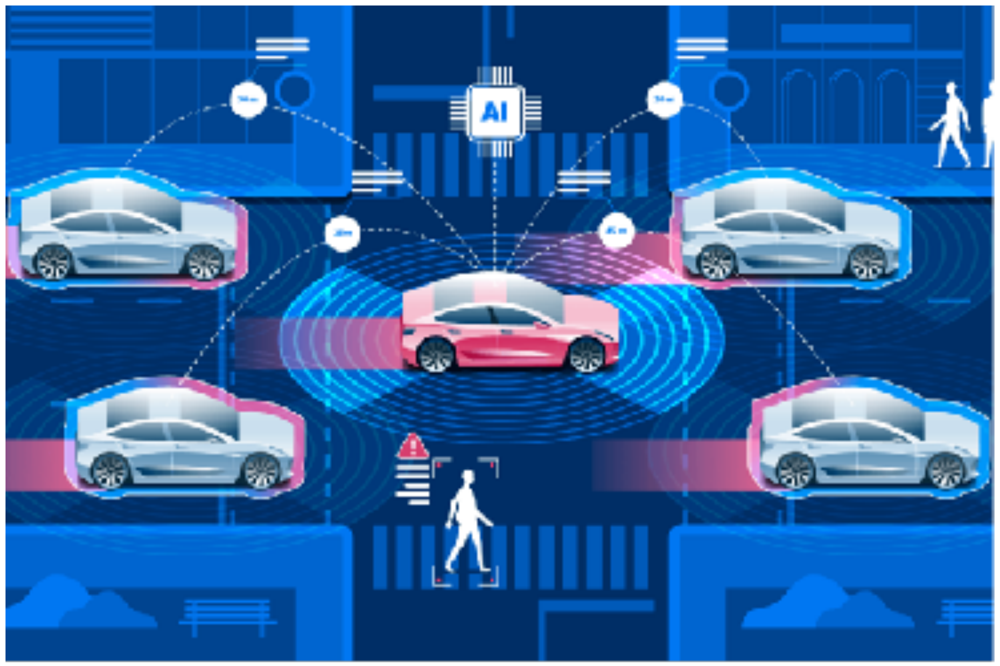




Multi-Attribute Transfer via Disentangled Representation (Zhang et al., AAAI 2019)

Adversarial Machine Learning (Reliability and Robustness)

Extreme Reliability and Safety



Autonomous vehicles

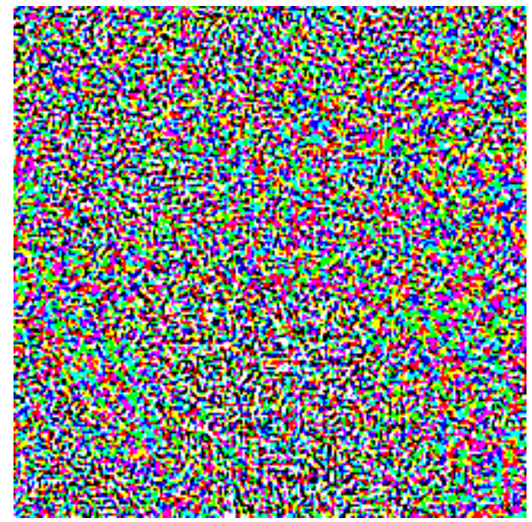
Air traffic control

Medical diagnosis

Surgery robots

Problem: DNNs are Brittle

$+ .007 \times$



$=$

$\boldsymbol{x}$

"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

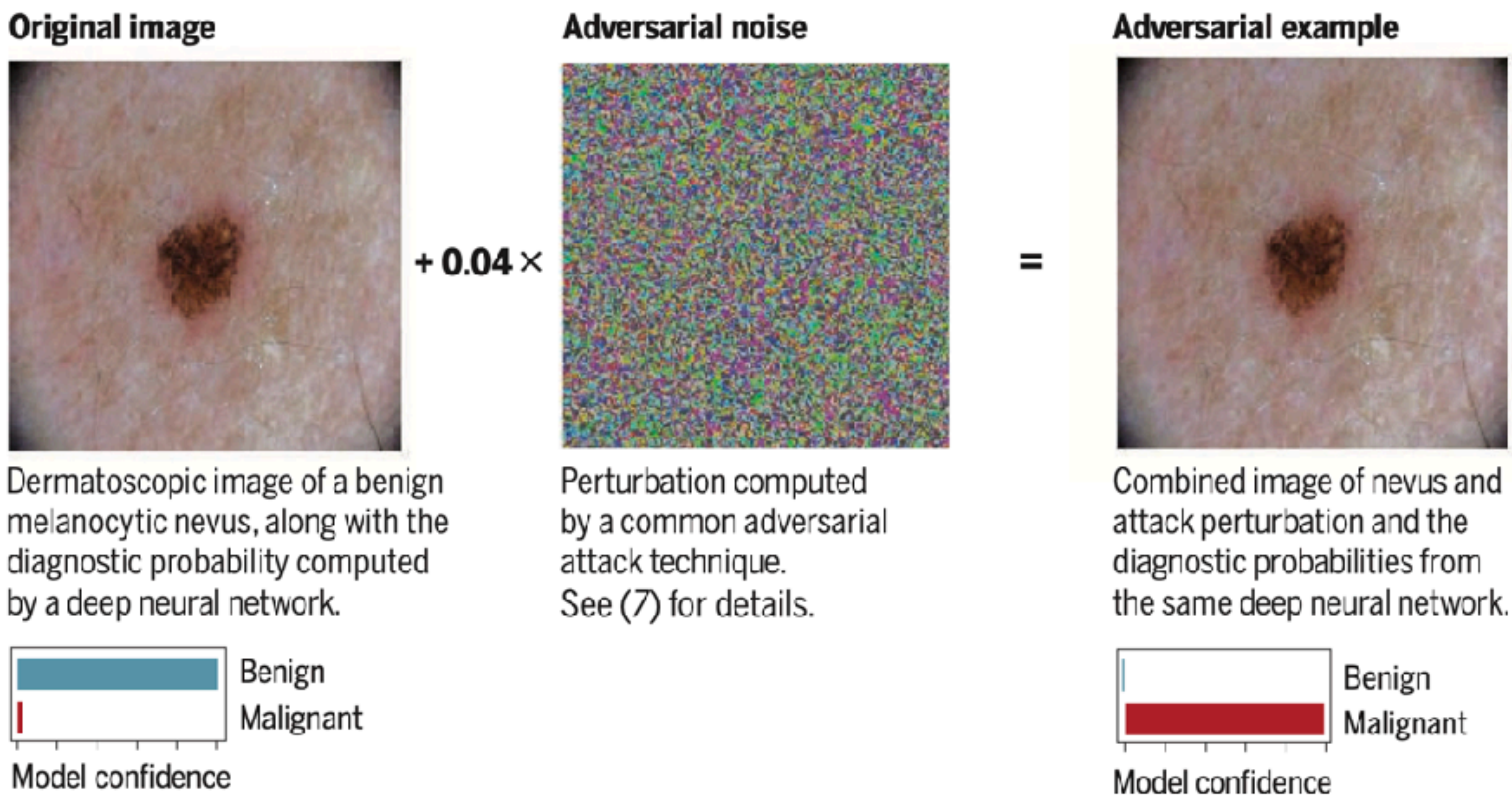
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Inconsistent perception between human and ML

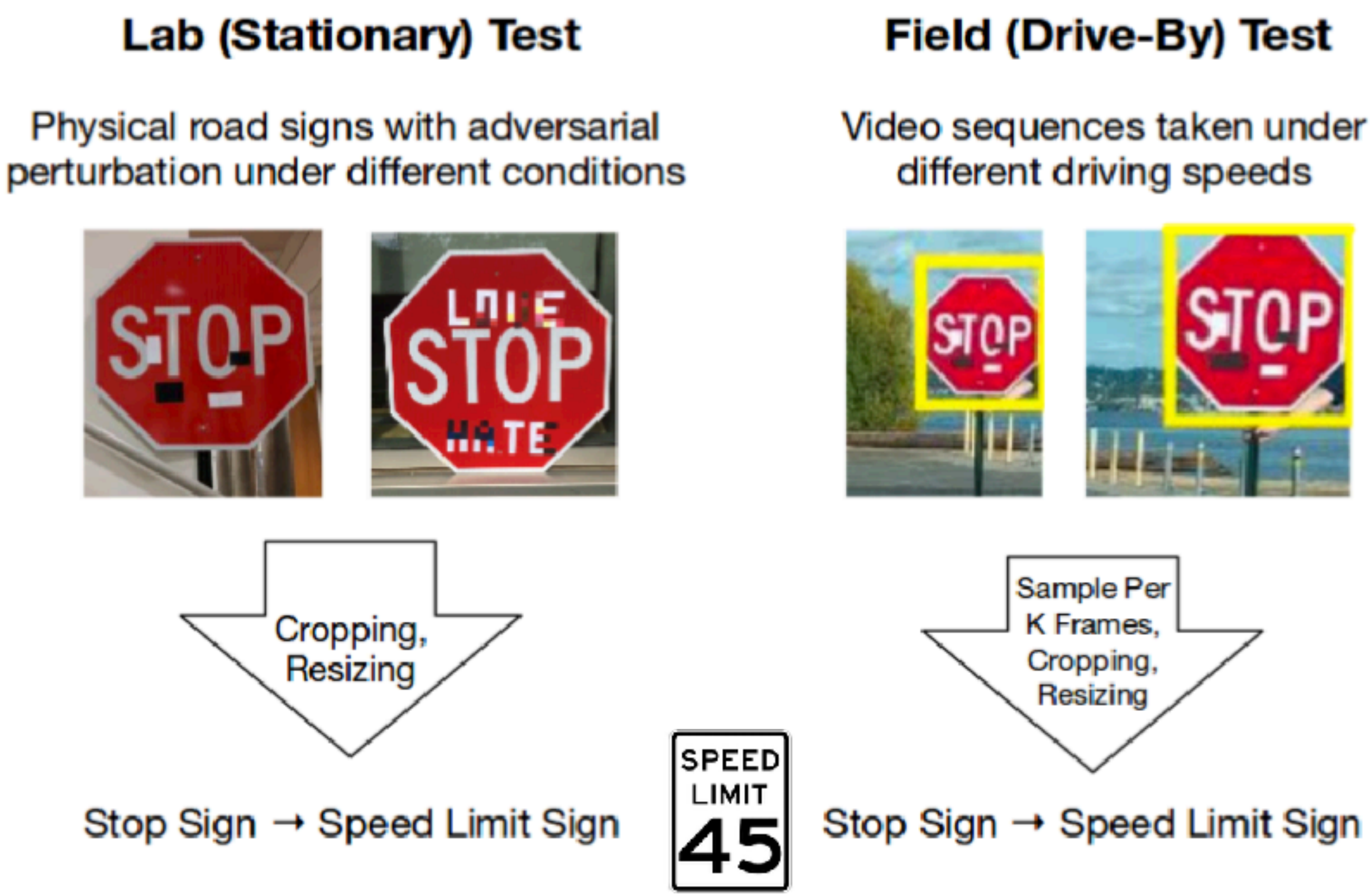
(Goodfellow et al., ICLR 2015)

Reliability: Medical Diagnosis



(Finlayson et al. Science 2019)
Adversarial attacks on medical machine learning

Robust Physical-World Attacks

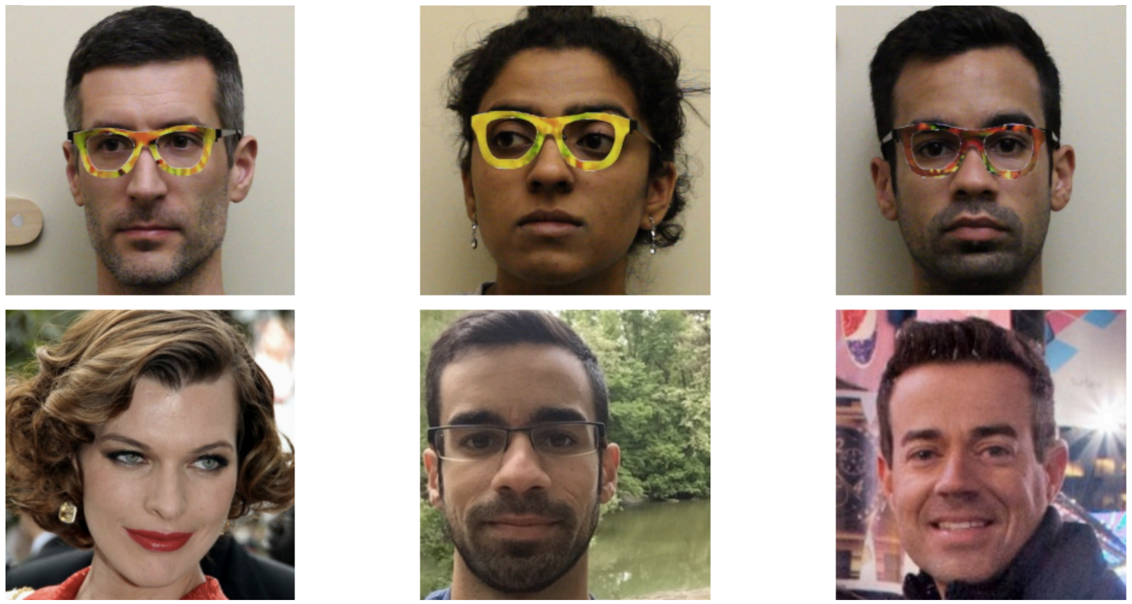


(Eykholt et al., Robust physical-world attacks on deep learning visual classification, CVPR 2018)

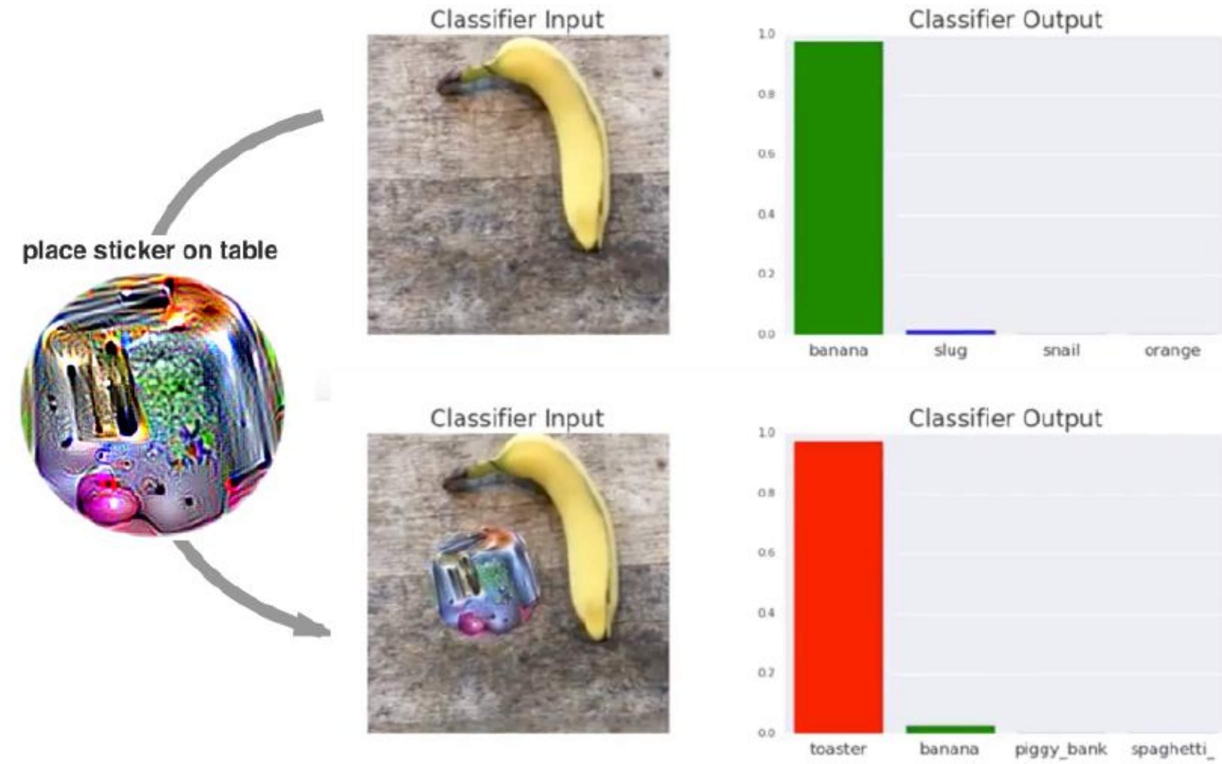
More Examples

[Sharif Bhagavatula Bauer Reiter 2016]:
Glasses that fool face recognition

Input Image
Recognized

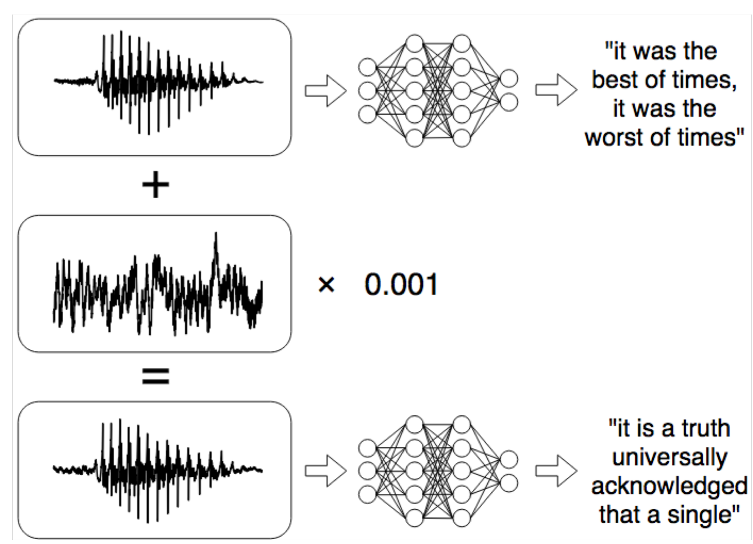


- Adversarial patch



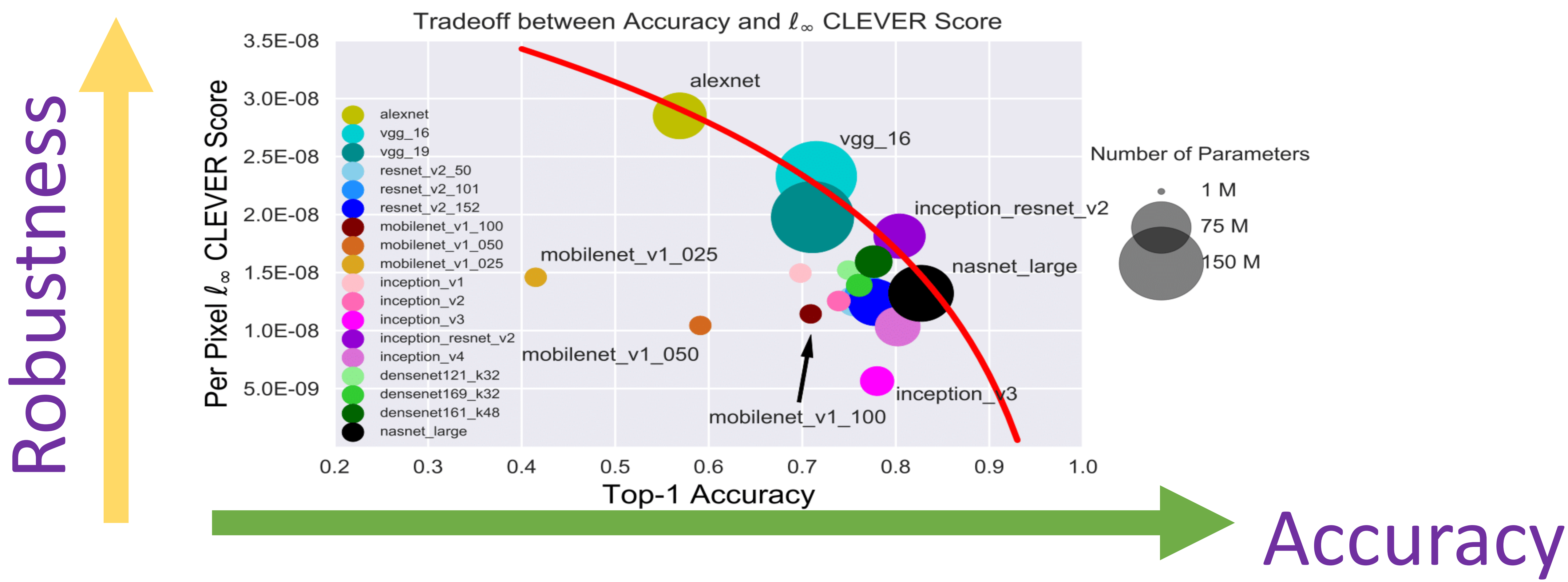
(Brown et al., 2017)

[Simen Thys et al., 2019]
Fooling automated surveillance cameras: adversarial patches to attack person detection



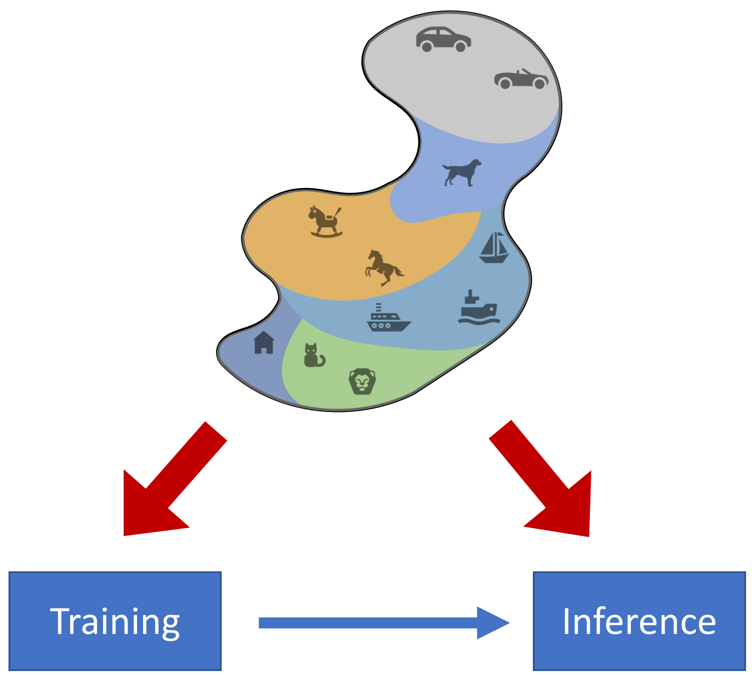
[Carlini Wagner 2018]:
Voice commands that are unintelligible to humans

Accuracy vs. Adversarial Robustness

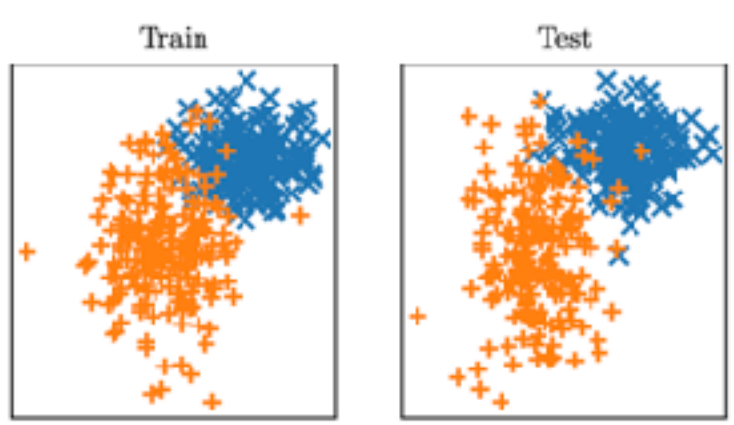


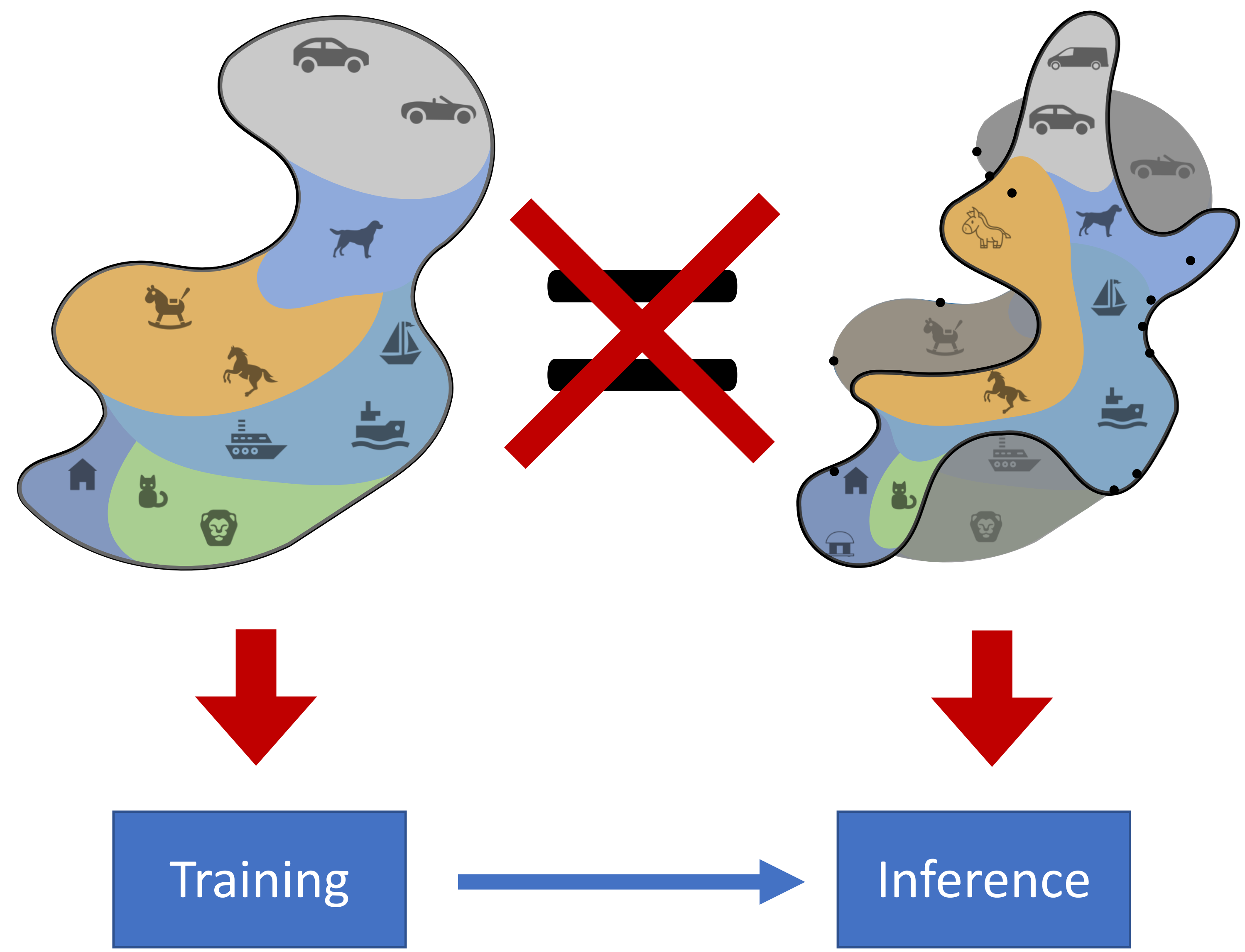
(D. Su et al., Is Robustness the Cost of Accuracy? - A Comprehensive Study on the Robustness of 18 Deep Image Classification Models, ECCV 2018)

Limitation of ML Framework



All training and testing data examples drawn independently from same distribution



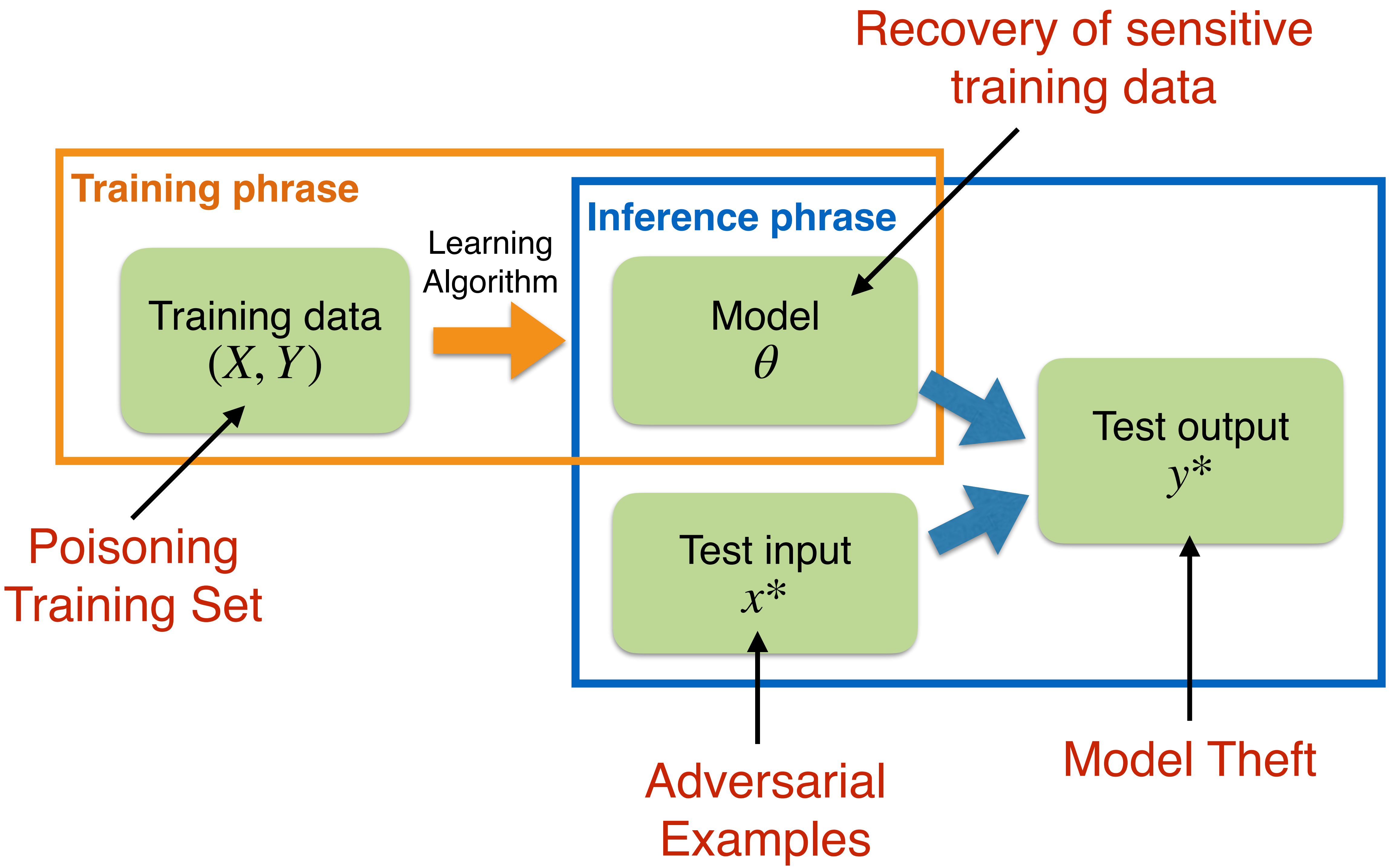


Real-world application

Implication of Adversarial Examples

- ML has high score of accuracy but not sufficiently intelligent
- Distinct principles between human perception and ML
- Risky for safety critical applications
- Limitations of current ML methods
- Trust between human and AI

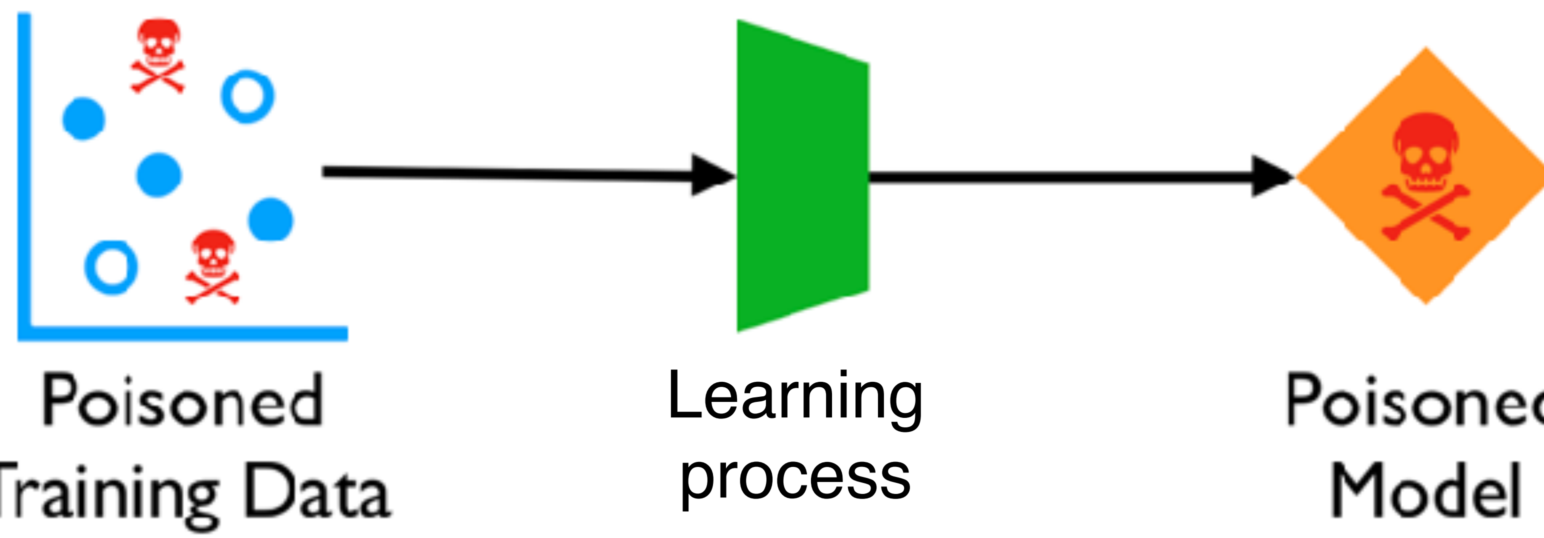
Attacks on ML Pipeline

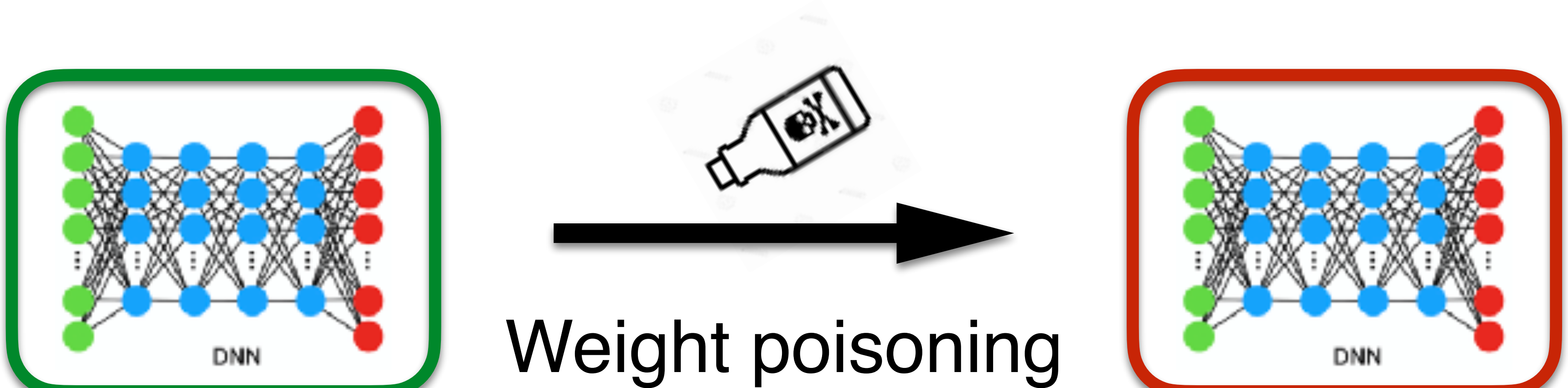


Poisoning Attack

- By poisoning training data, the model will be compromised
- Planting backdoors in training data, such that the data with backdoors will be misclassified

Data poisoning

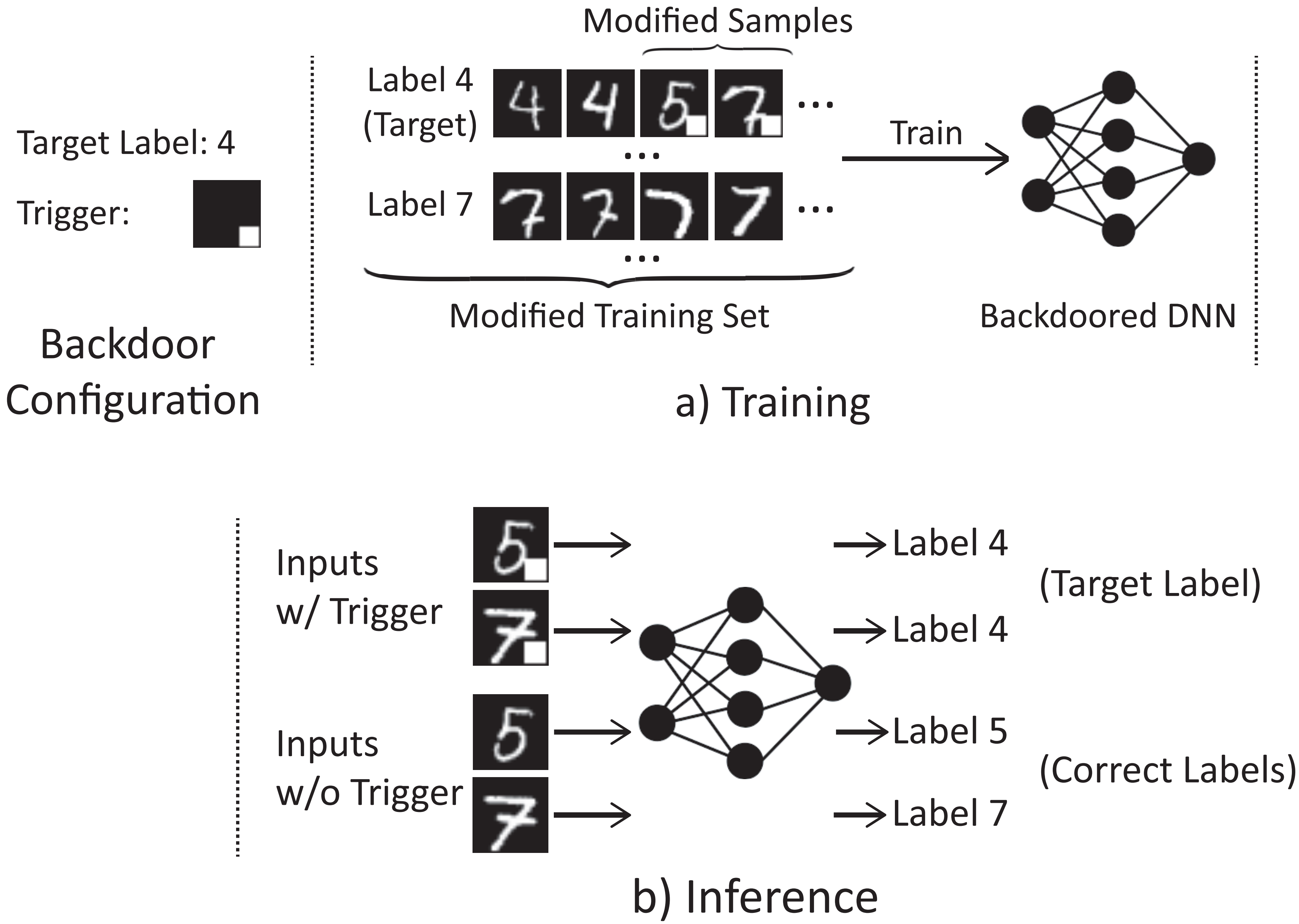




Clean Model

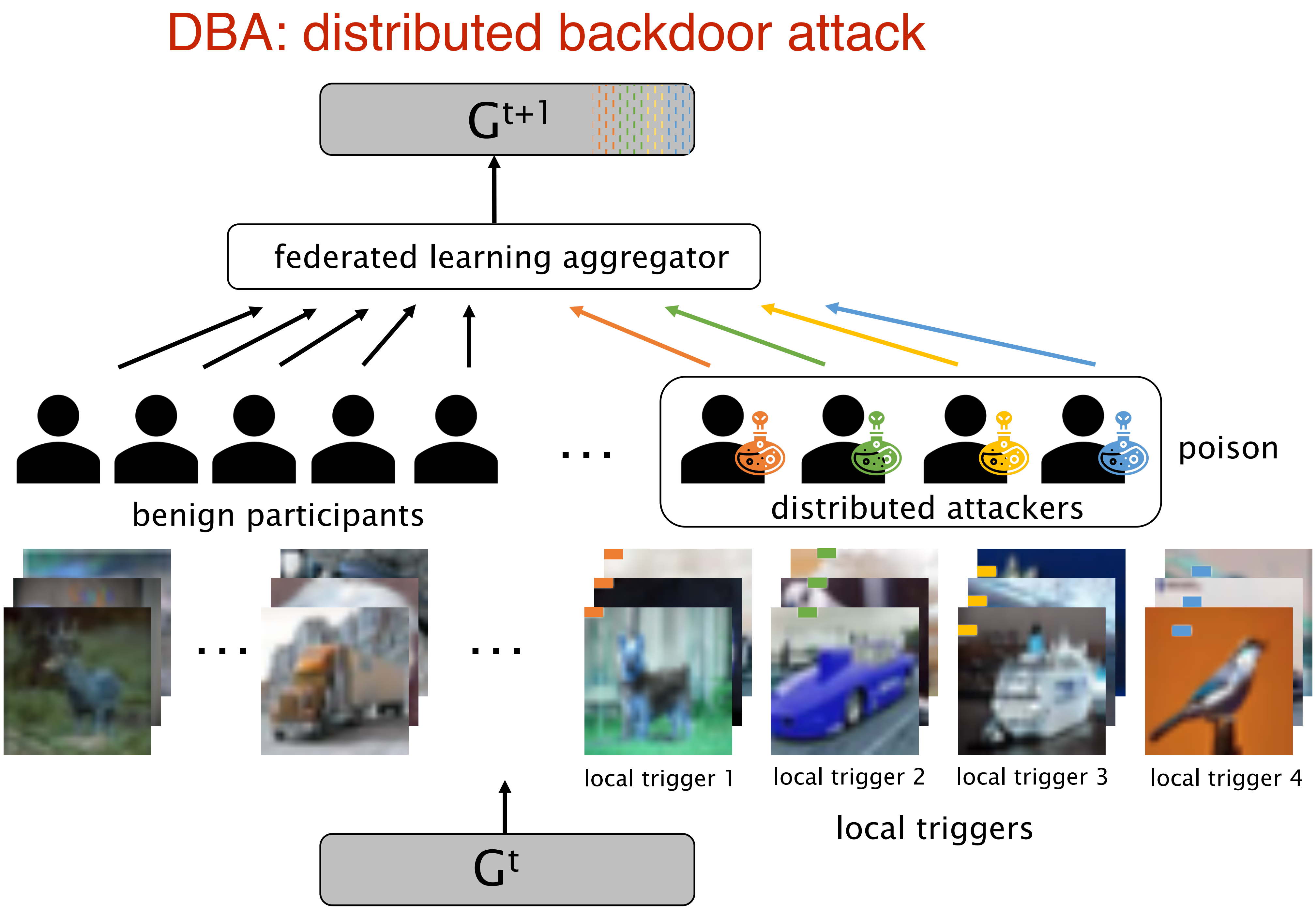
Poisoned Model

Trojan Attack



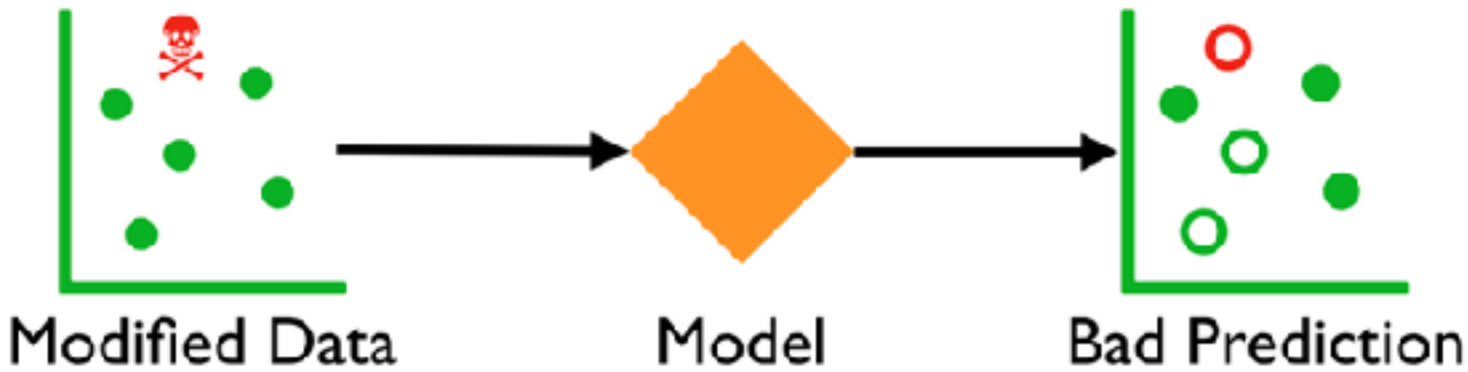
(Bolun Wang et al., Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. IEEE Security and Privacy, 2019)

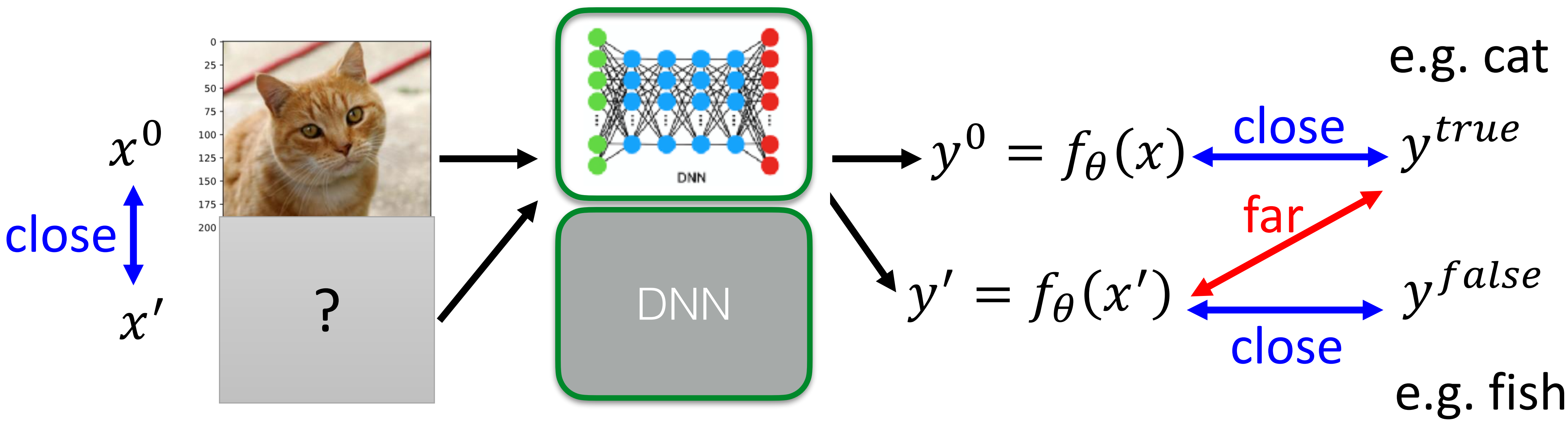
Backdoor Attack against Federated Learning



(Chulin Xie, et al., DBA: Distributed Backdoor Attacks against Federated Learning. ICLR 2020)

Evasion Attack: Adversarial Examples





Target vs. non-targeted attack

White-box vs. black-box attack

Intriguing Properties of NN (1)

Natural basis vector w.r.t.
i-th hidden unit

$$x' = \arg\max_{x \in \mathcal{I}} \langle \phi(x), e_i \rangle$$

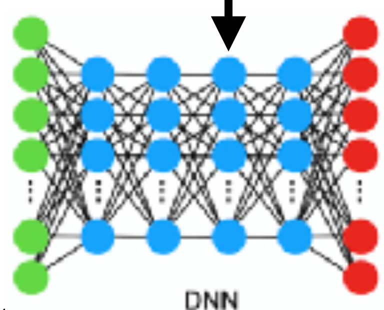
Input images

hidden layer activations

$$x' = \arg\max_{x \in \mathcal{I}} \langle \phi(x), v \rangle$$

Random vector

hidden layer activations




(a) Unit sensitive to white flowers.

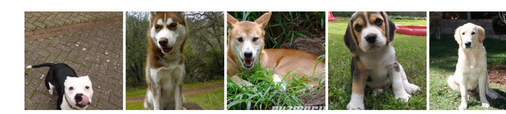

(b) Unit sensitive to postures.


(c) Unit senstive to round, spiky flowers.

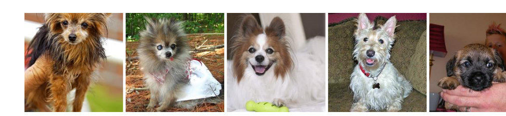

(d) Unit senstive to round green or yellow objects.

Basis activation has specific semantic property


(a) Direction sensitive to white, spread flowers.


(b) Direction sensitive to white dogs.


(c) Direction sensitive to spread shapes.


(d) Direction sensitive to dogs with brown heads.

Random activations also has specific semantic property

Uninterpretable and counter-intuitive properties of DNN

- No distinction between individual high level units and random activations

(Szegedy et al. Intriguing properties of neural networks, ICLR 2014)

Intriguing Properties of NN (2)

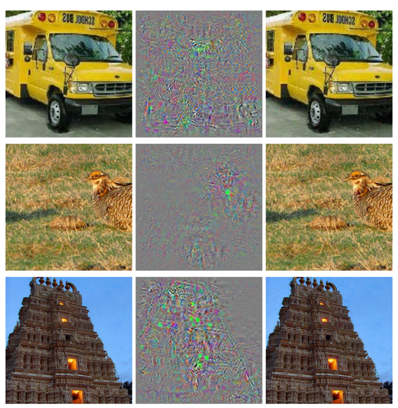
$$\text{Minimize } c|r| + \text{loss}_f(\boxed{x+r}, l) \text{ subject to } x + r \in [0,1]^m$$

Optimization of Perturbation

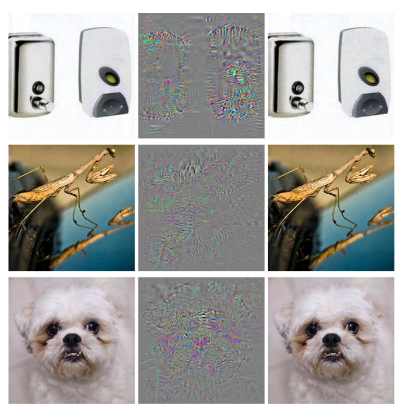
Adversarial Example

Wrong Label

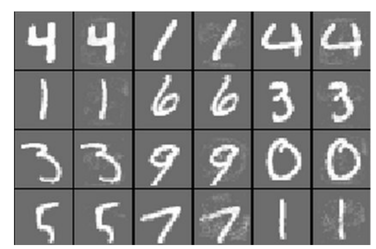
Ostrich, struthio, camelus



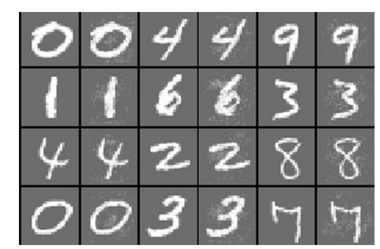
(a)



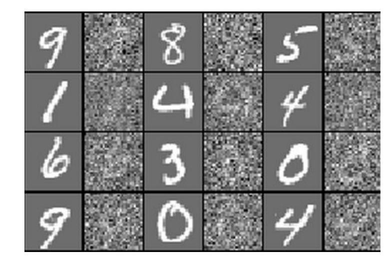
(b)



(a) Even columns: adversarial examples for a linear (FC) classifier (stddev=0.06)



(b) Even columns: adversarial examples for a 200-200-10 sigmoid network (stddev=0.063)



(c) Randomly distorted samples by Gaussian noise with stddev=1. Accuracy: 51%.

Uninterpretable and counter-intuitive properties of DNN

- Hardly perceptible perturbation can cause misclassification of network
- These distorted images or adversarial examples generalize fairly well even to different models trained by different dataset

(Szegedy et al. Intriguing properties of neural networks, ICLR 2014)

Why Do Adversarial Examples Happen?

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$$

Adversarial Example — Input Data — Perturbation

Linear layer

$$\boldsymbol{w}^T\tilde{\boldsymbol{x}} = \boldsymbol{w}^T\boldsymbol{x} + \boxed{\boldsymbol{w}^T\boldsymbol{\eta}}$$

$$\boxed{\max_{\eta} \boldsymbol{w}^T\eta \quad \|\boldsymbol{\eta}\|_\infty < \epsilon} \xrightarrow{\eta = \operatorname{sign}(\boldsymbol{w})} \boxed{\boldsymbol{w}^T\boldsymbol{\eta} \propto \epsilon m n}$$

Average magnitude of $\boldsymbol{w}$ — Dimensionality of input

- Early explanations for adversarial examples is highly nonlinearity and overfitting of NN (is it wrong?)
- Adversarial samples are caused by high-dimensionality of input and models being too linear rather than too nonlinear
- Linear models lack the capacity to resist adversarial perturbation
- Generalization of adversarial examples across different models can be explained as the perturbations being highly aligned with the weight vectors of model

(Goodfellow et al. Explaining and Harnessing Adversarial Examples, ICLR 2015)

FGSM: Fast Gradient Sign Method

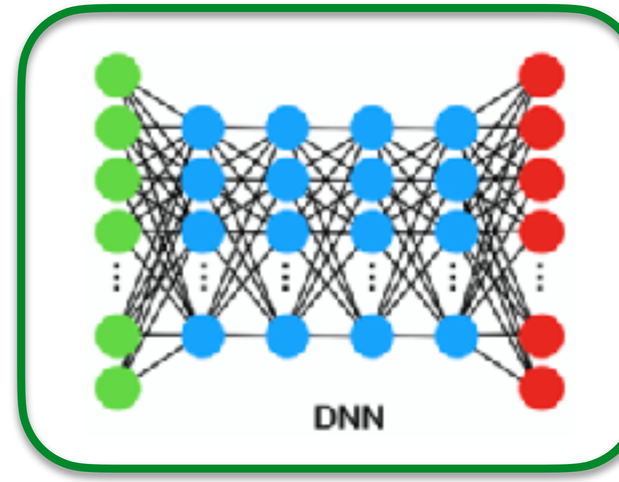
Adversarial Examples

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$$

Adversarial Example

Input Data

Perturbation



Model parameter θ

Adversarial Attack

Perturbation

Model Parameters

Input Data

Label

$$\eta = \epsilon \, \mathrm{sign}(\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}, y))$$

Gradient of loss function w.r.t. input

(Goodfellow et al. Explaining and Harnessing Adversarial Examples, ICLR 2015)

Objective of Adversarial Training

Adversarial Training

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)))$$

Loss for training data

Regularizer for Robustness

Adversarial Example

Perturbation

- Adversarial examples are continually updated given current model
- The larger model capacity is required to reduce error on adversarial examples
- Adversarially trained model shows great robustness to adversarial examples
- The weight of model are more localized and interpretable
- Adversarial training = Active learning

(Goodfellow et al. Explaining and Harnessing Adversarial Examples, ICLR 2015)

Optimization for Adversarial Attack

Standard training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \mathcal{L}(\theta, x, y)$$

Model Parameter Loss Input Label

Goal:

$$\min_{\delta} \|\delta\|_p \quad \text{s.t. } f_\theta(x+\delta) \neq f_\theta(x)$$

Perturbation Model Input Model parameter

Optimization:

$$\max_{\delta} \mathcal{L}(\theta, \boxed{x+\delta,}\, y) \quad \text{s.t. } \boxed{\|\delta\|_p \leq \epsilon}$$

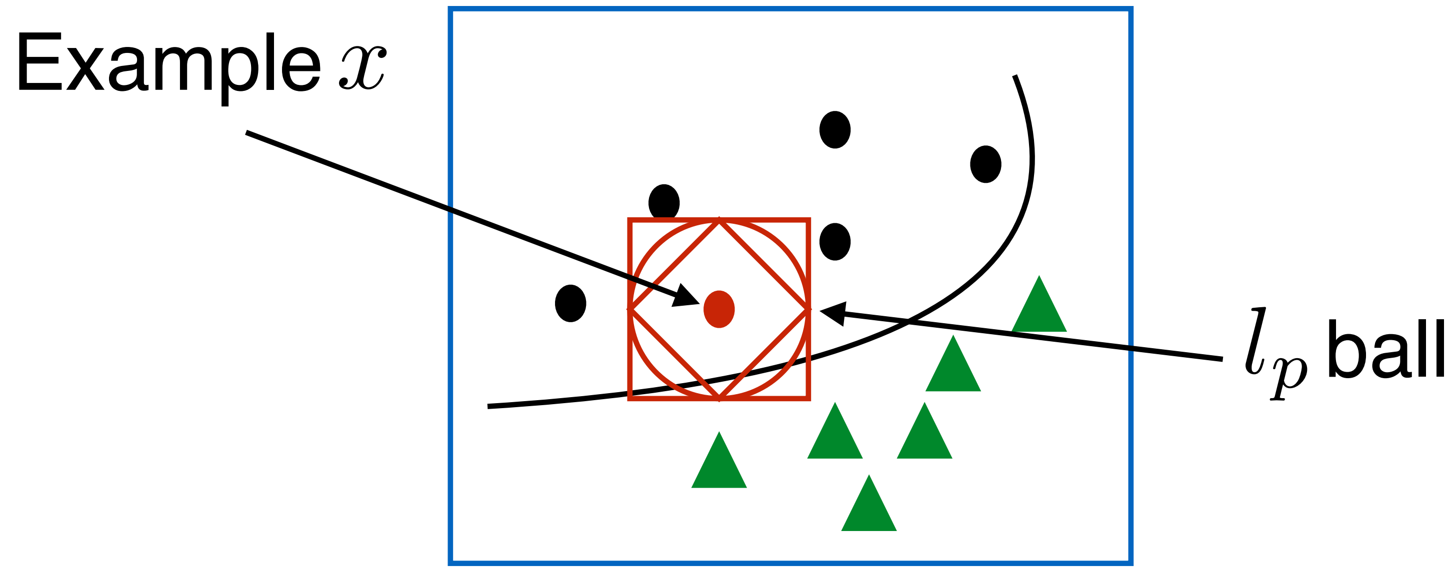
Loss Adversarial Example True Label Keep Inperceptible

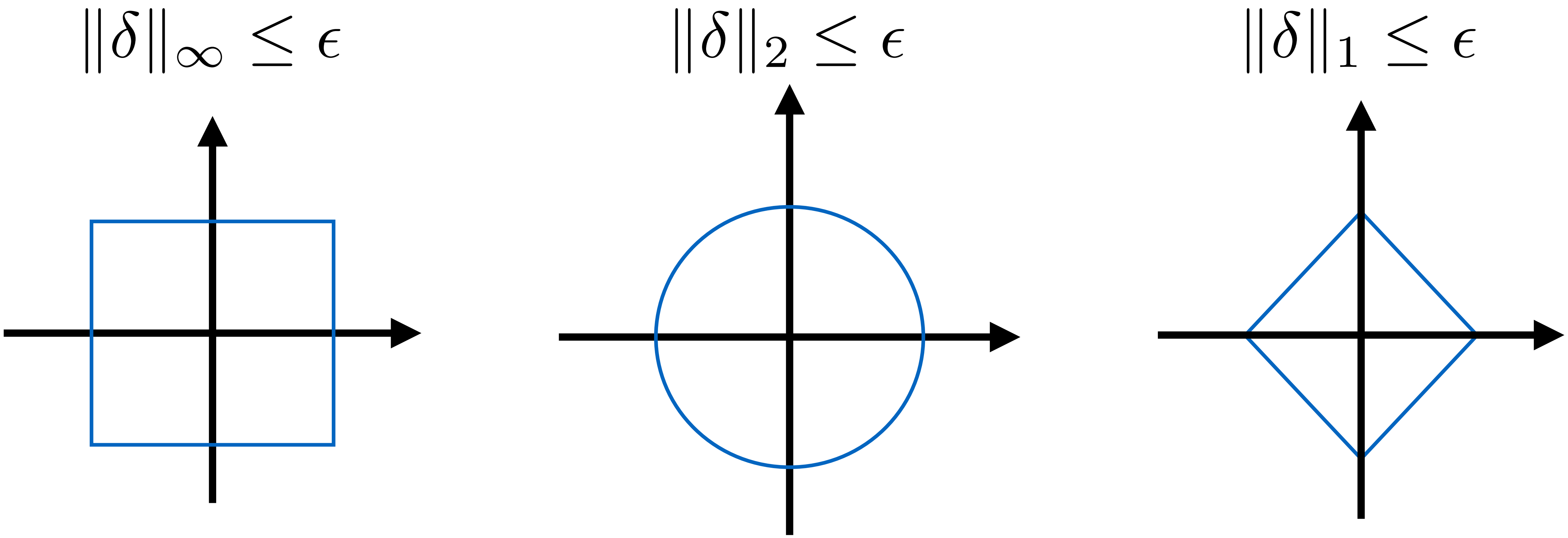
Gradient:

$$\nabla_\theta \mathcal{L}(\theta, x, y) \qquad \nabla_\delta \mathcal{L}(\theta, x+\delta, y)$$

Geometry of l_p-Norm

$\|\delta\|_p \leq \epsilon$





Target Attacks

Goal: $$\min_{\delta} \|\delta\|_p \quad \text{s.t. } f_\theta(x+\delta) = y'$$

DNN model

Target Label

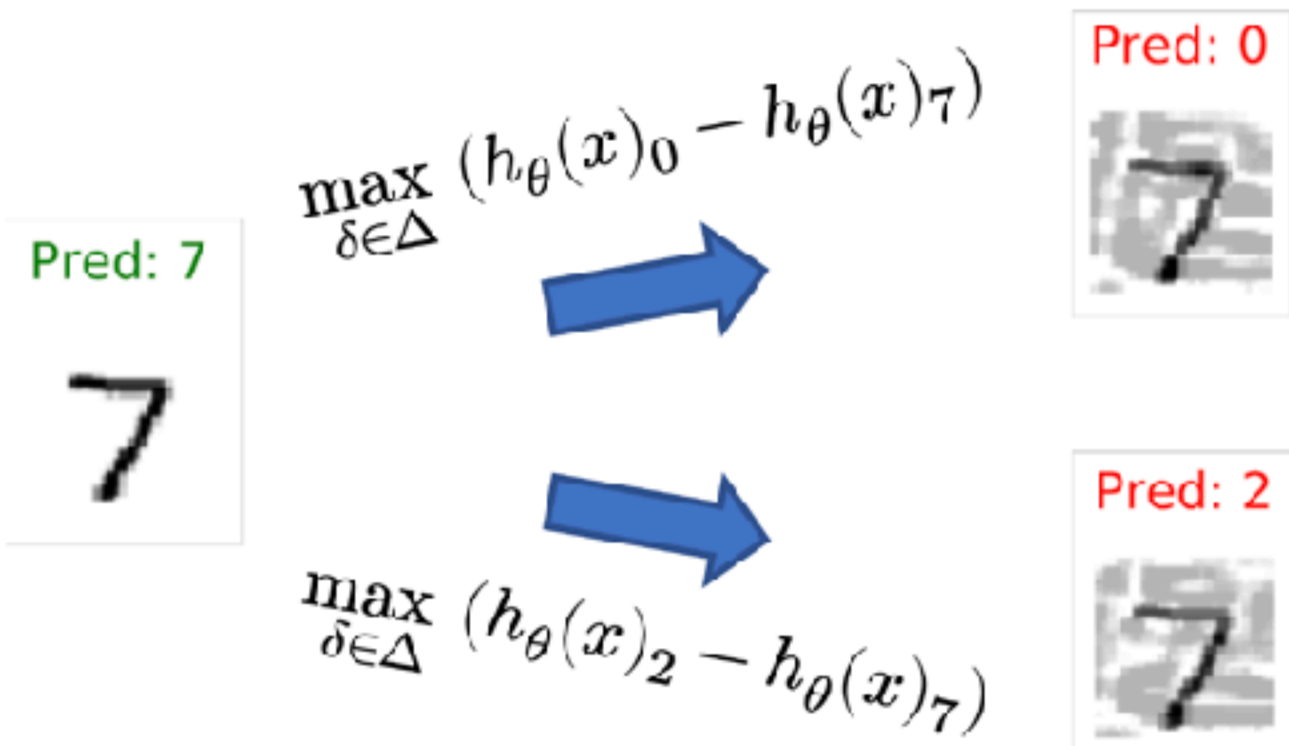
Optimization problem:

$$\max_{\delta}\{\boxed{\mathcal{L}(\theta, x+\delta, y)} - \boxed{\mathcal{L}(\theta, x+\delta, y')}\} \quad \text{s.t. } \|\delta\|_p \leq \epsilon$$

Loss w.r.t. true label

Loss w.r.t. target label

Targeted Attacks: Example



Note: A targeted attack can succeed in “fooling” the classifier, but change to a different label than target

White-box Attacks

Fast approaches

- Fast gradient sign $\delta = \epsilon \, \mathrm{sgn}(\nabla_x \mathcal{L}(\theta, x, y))$
- Fast gradient $\delta = \epsilon \left(\frac{\nabla_x \mathcal{L}(\theta, x, y)}{\|\nabla_x \mathcal{L}(\theta, x, y)\|_2} \right)$

Iterative approach

$$\max_{\delta} \mathcal{L}(\theta, x + \delta, y) - \lambda \|\delta\|_p$$

Target specific optimization

$$\min_{\delta} \mathcal{L}(\theta, x + \delta, y') + \lambda \|\delta\|_p$$

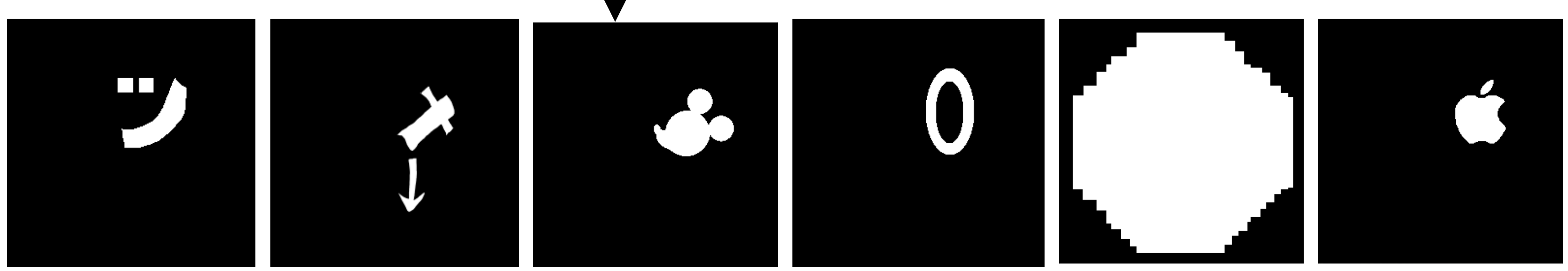
Need to know model f_θ

Adversarial Examples with Spatial Constraints

$$\min_{\delta} \sum_{i=1}^{n} \mathcal{L}(\theta, x_i + \delta, y') + \lambda \|\delta\|_p$$

$$\left\{ \right.$$$$\left. \right\}$$

$$\min_{\delta} \sum_{i=1}^{n} \mathcal{L}(\theta, x_i + M_x \cdot \delta, y') + \lambda \| M_x \cdot \delta \|_p$$



Subtle Poster

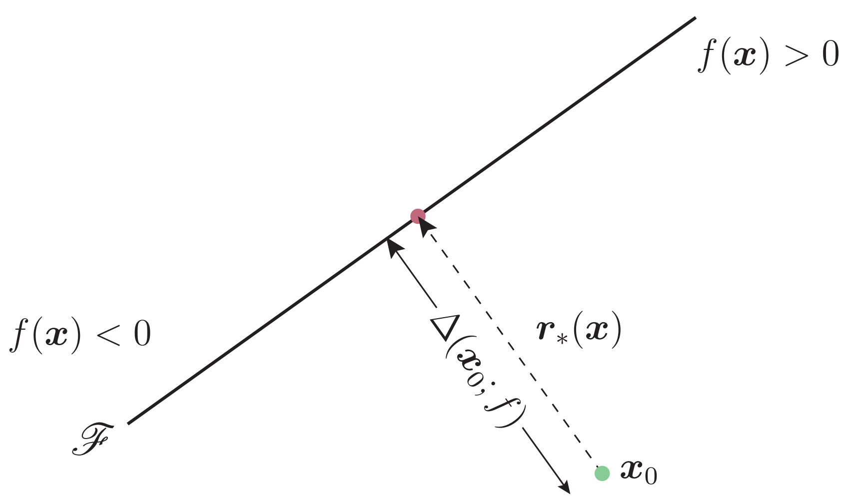
Camouflage Sticker

Mimic vandalism

"Hide in the human psyche"

DeepFool

$$\arg\min_{\boldsymbol{r}_i} \|\boldsymbol{r}_i\|_2 \text{ subject to } f(\boldsymbol{x}_i) + \nabla f(\boldsymbol{x}_i)^T \boldsymbol{r}_i = 0.$$



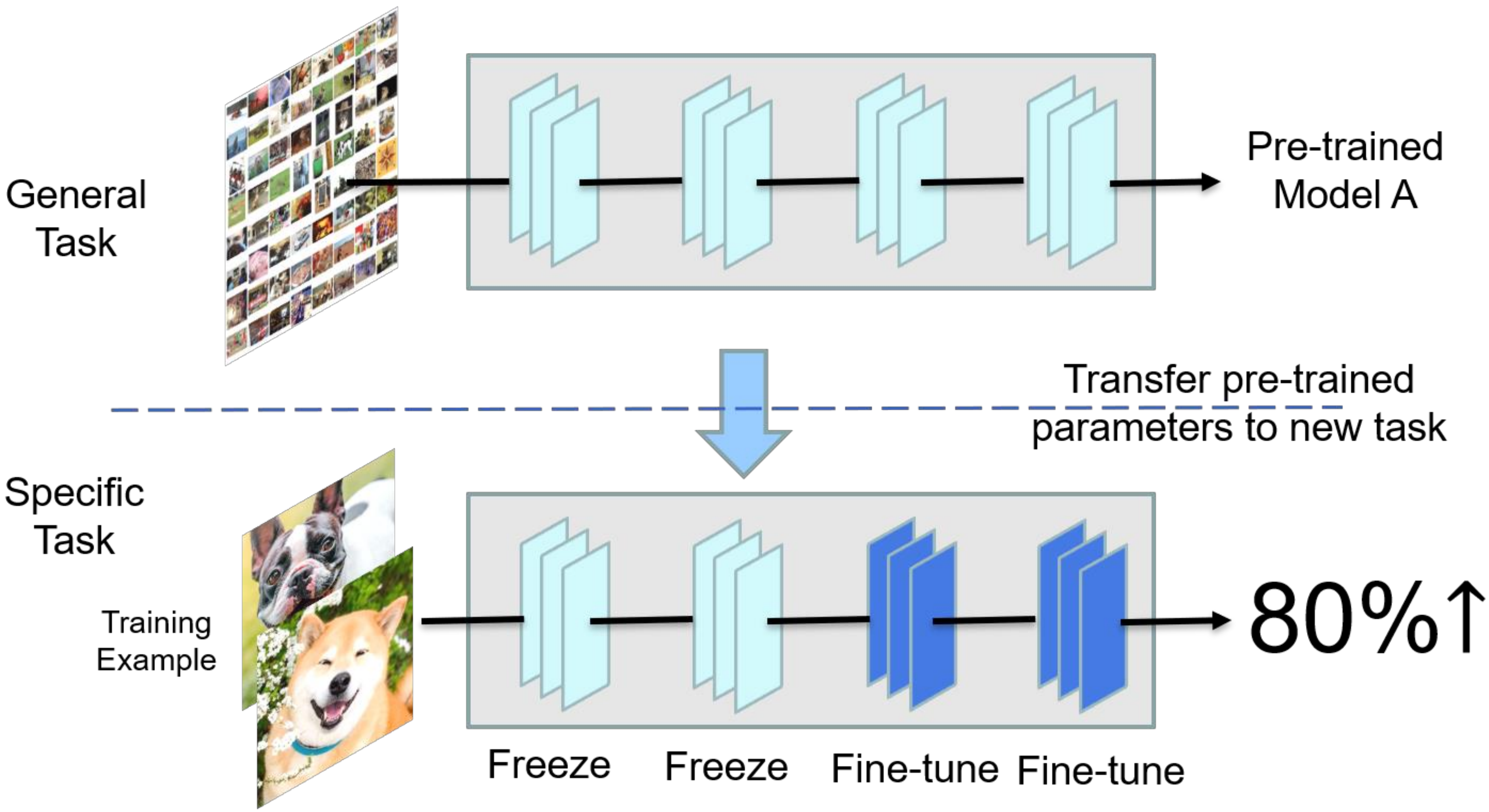
Algorithm 1 DeepFool for binary classifiers

input: Image $\boldsymbol{x}$, classifier f.
output: Perturbation $\hat{\boldsymbol{r}}$.
Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
while $\text{sign}(f(\boldsymbol{x}_i)) = \text{sign}(f(\boldsymbol{x}_0))$ **do**
    $\boldsymbol{r}_i \leftarrow -\frac{f(\boldsymbol{x}_i)}{\|\nabla f(\boldsymbol{x}_i)\|_2^2} \nabla f(\boldsymbol{x}_i)$,
    $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$,
    $i \leftarrow i + 1$.
end while
return $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$.

- Iterative optimization of perturbations for linear classifiers

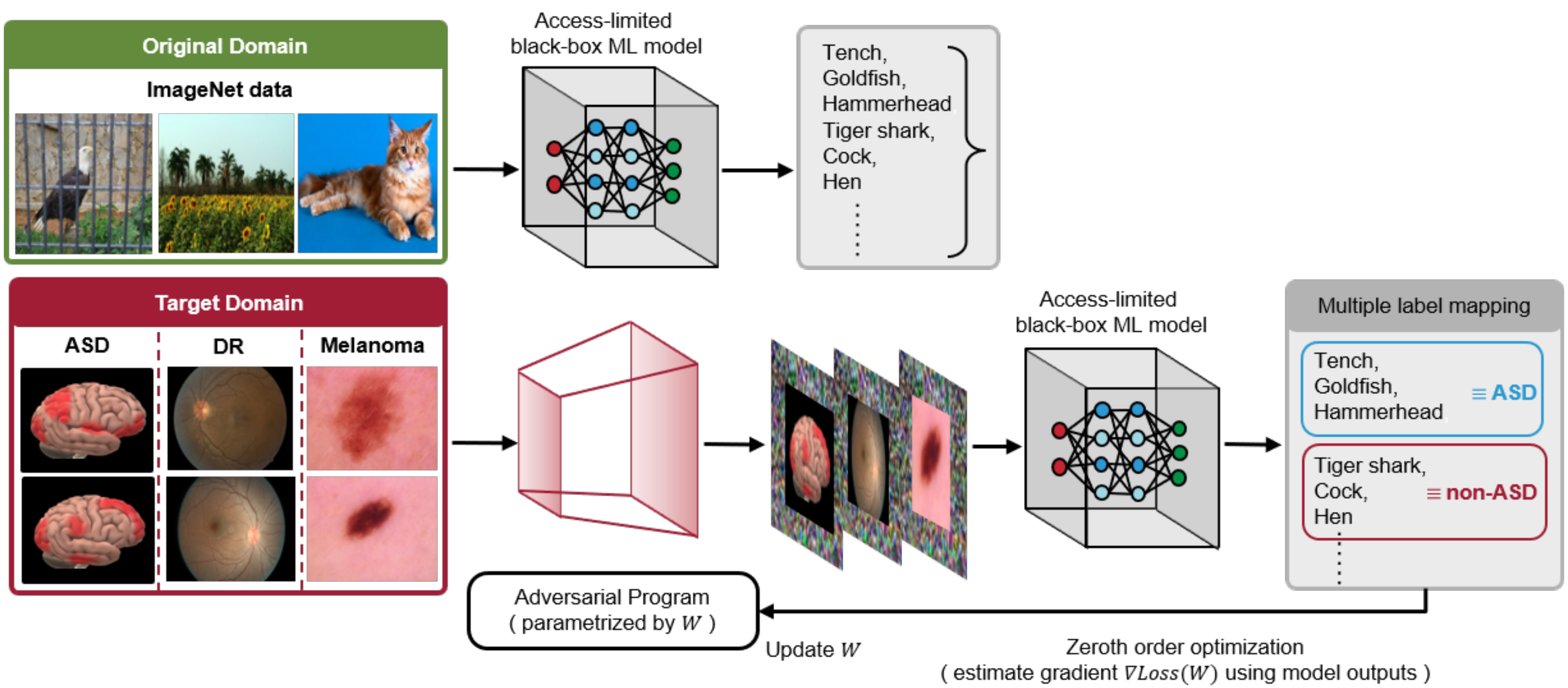
(Moosavi-Dezfooli et al., DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks, CVPR 2016)

Application to Transfer Learning



Black-box Adversarial Reprogramming (BAR)

- Transfer learning: from finetuning to black-box setting
- Cross domain and data limited transfer learning

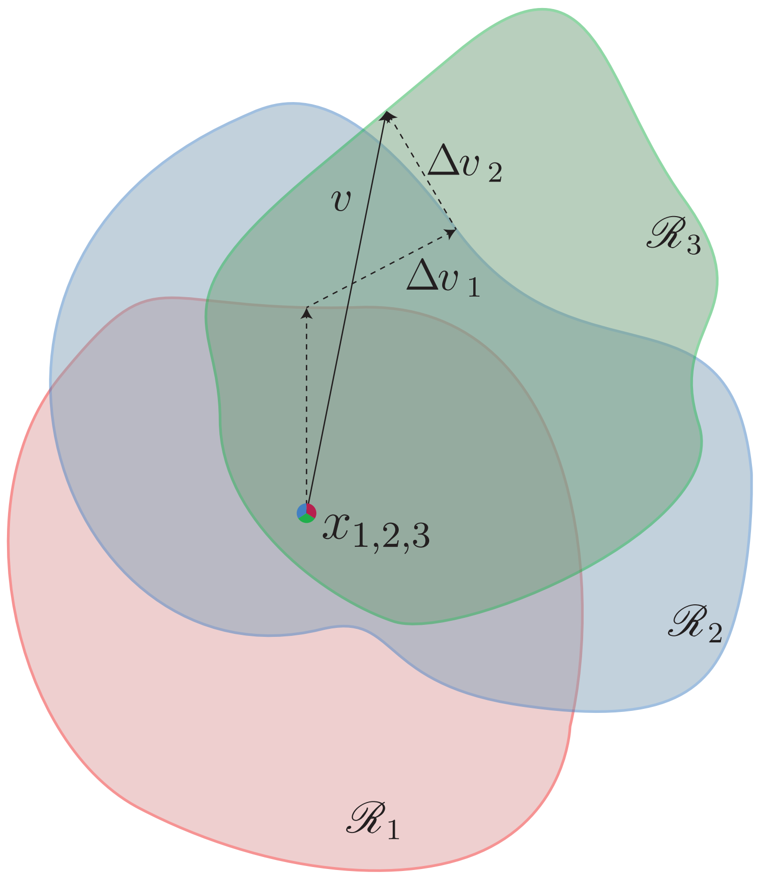


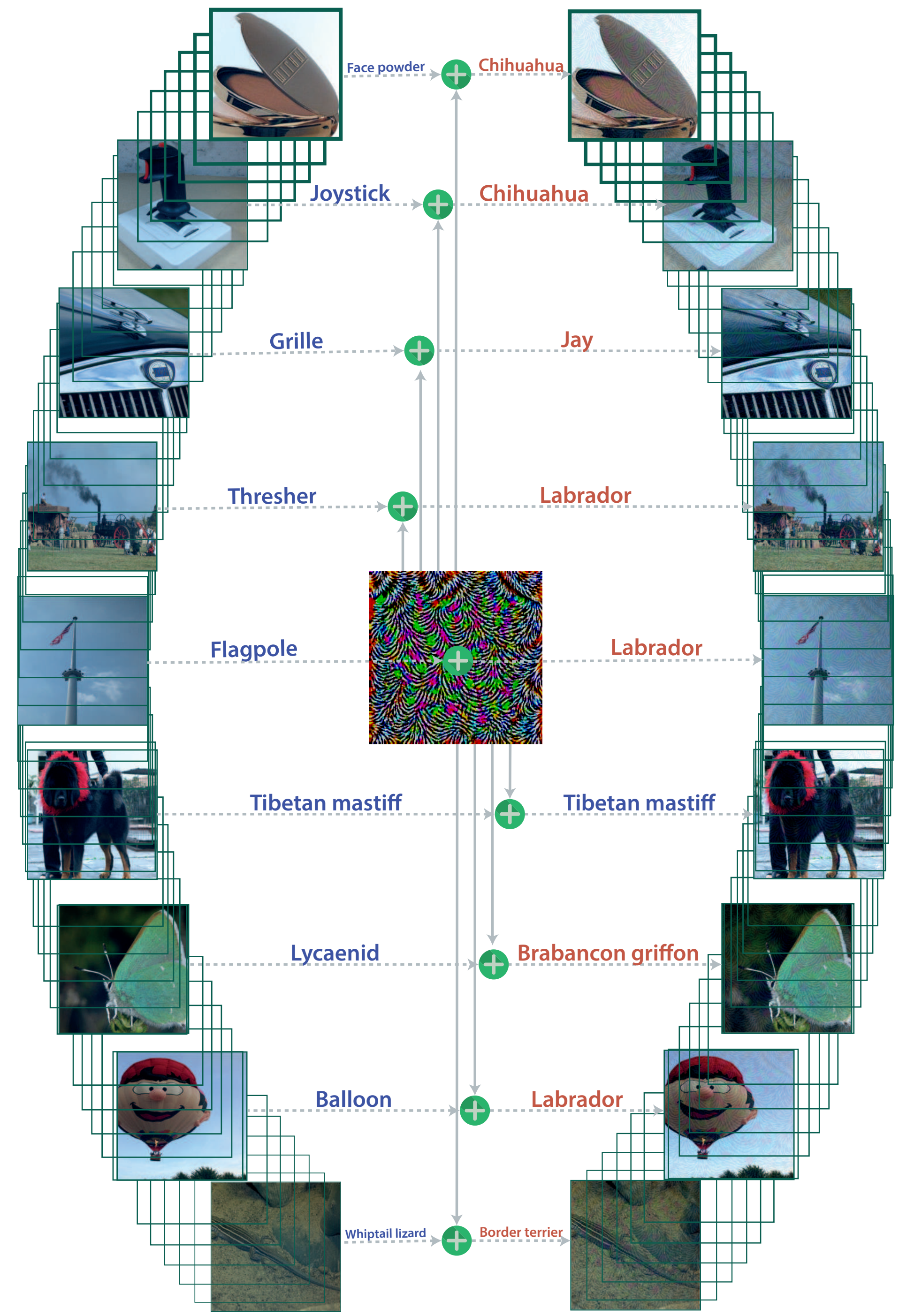
(Y. Tsai et al., Transfer Learning without Knowing: Reprogramming Black-box Machine Learning Models with Scarce Data and Limited Resources, ICML 2020)

Universal Adversarial Perturbations

Universal perturbation to

- Data sample
- Models
- Input transformations
- Ensemble methods





(Moosavi-Dezfooli et al., Universal Adversarial Perturbations, CVPR 2017)

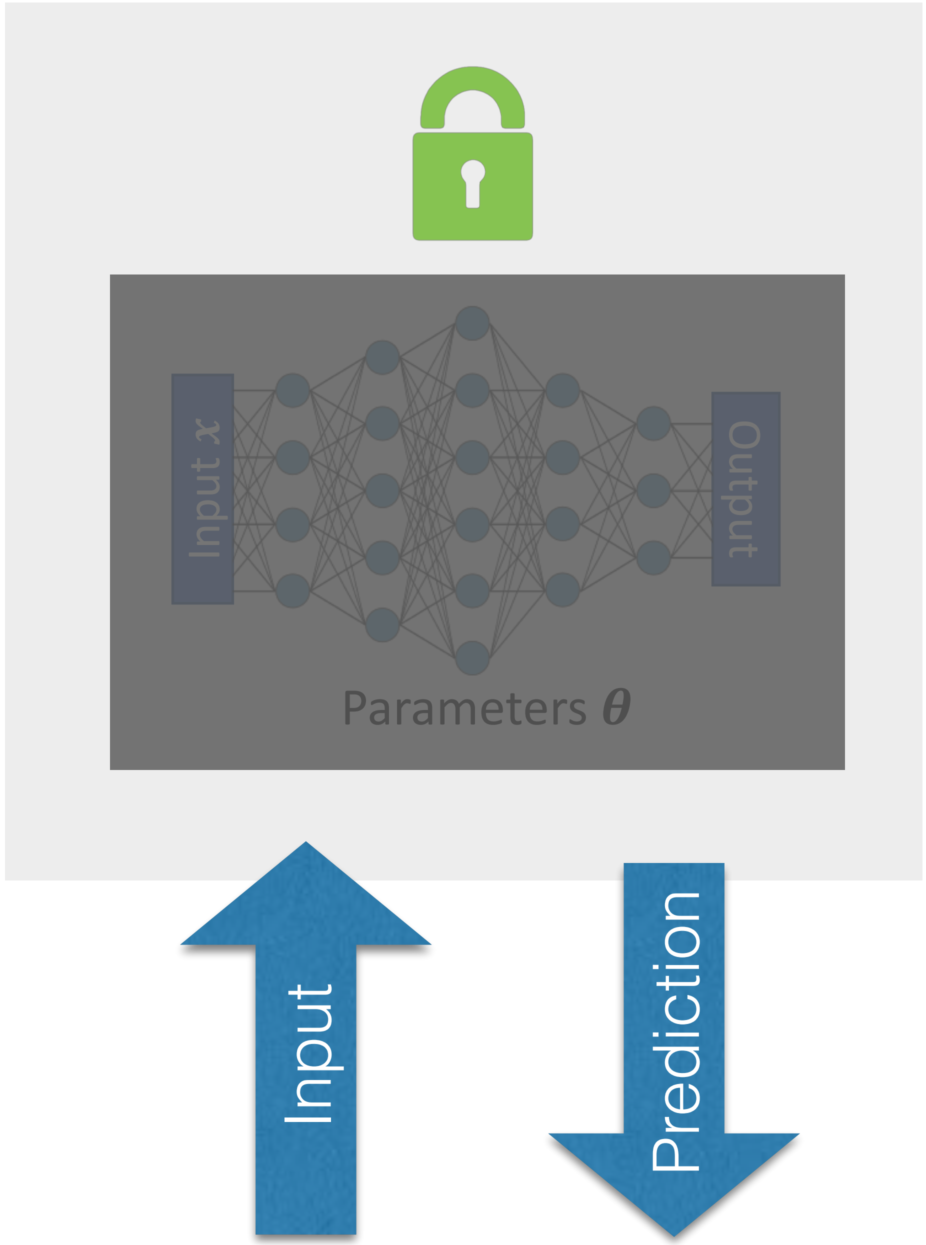
Black-box Attacks

Zero-query attack

- Random perturbation
- Difference of means
- Transferability based attack

Query based attack

- Finite difference gradient estimation
- Query reduced gradient estimation

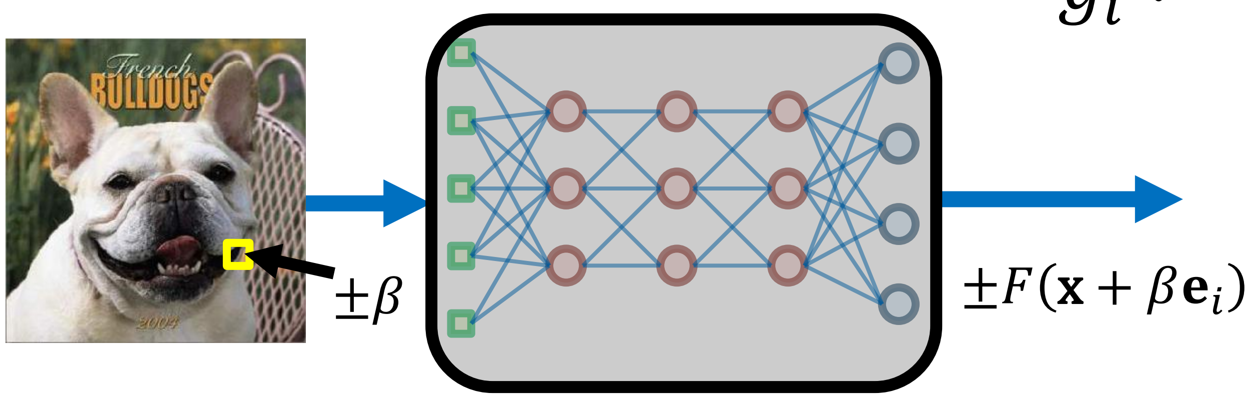


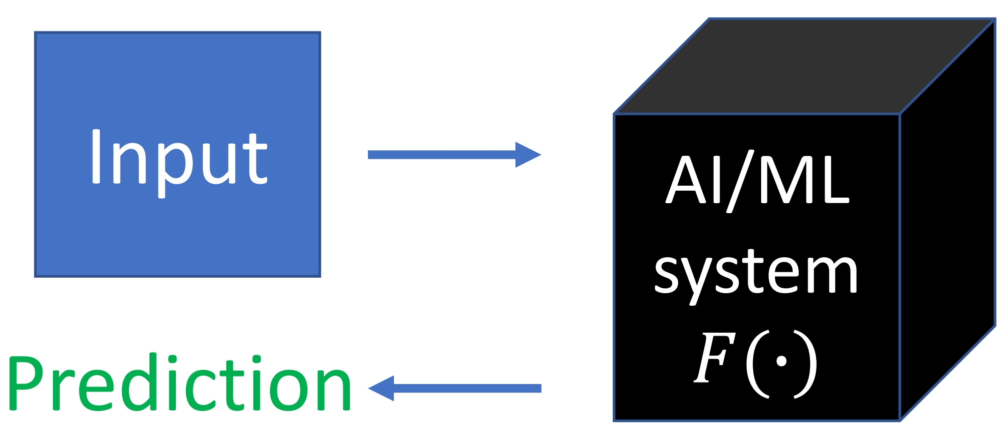
Zero knowledge about model and training data

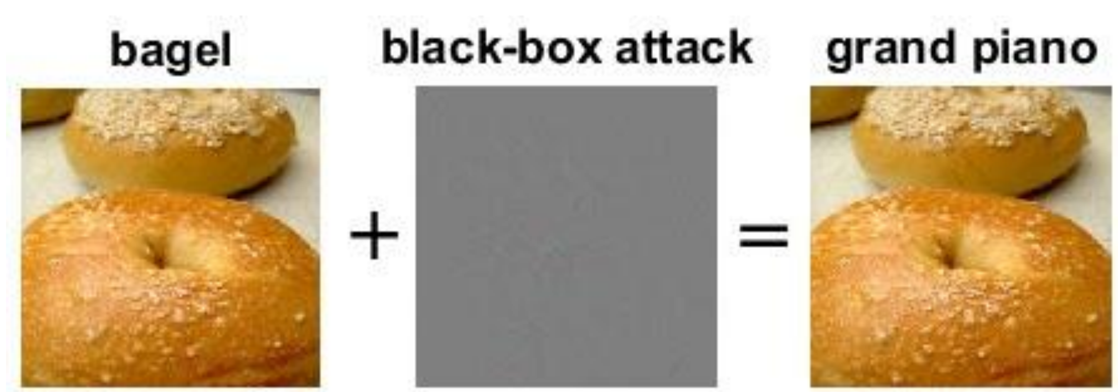
Key Technique

- Black-box system is also vulnerable to adversarial attack
- Gradient estimation from system outputs instead of back-prop

$$g_i := \frac{\partial f(\mathbf{x})}{\partial x_i} \approx \frac{f(\mathbf{x} + \beta \mathbf{e}_i) - f(\mathbf{x} - \beta \mathbf{e}_i)}{2\beta}$$





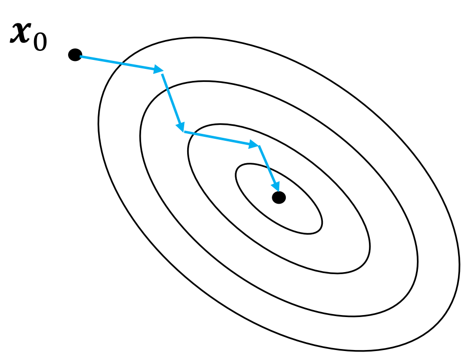


Zero-Order Optimization

- Estimate gradient using function value coordinate by coordinate (Chen et al., 2017)

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + he_i) - f(x - he_i)}{2h}$$

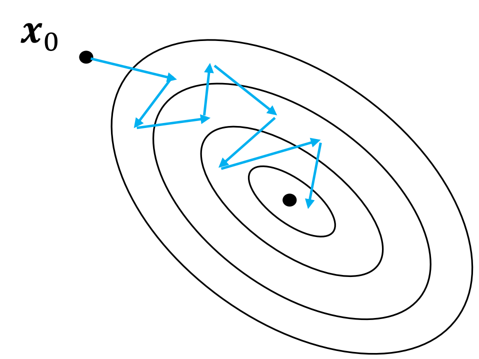
SGD (first order)



Convergence rate $E[\|\nabla F(\boldsymbol{x}_T)\|_2^2] = O(1/\sqrt{T})$

T is # of iterations

ZO-SGD



Convergence rate $E[\|\nabla F(\boldsymbol{x}_T)\|_2^2] = O(\sqrt{d}/\sqrt{T})$
[Duchi, et al., T-IT'15]

d is # of variables

Question: Better gradient estimate & ZO method with better convergence rate?

(S. Ghadimi & G. Lan, Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming, SIAM J. Optim. 2013)

Query Based Attack

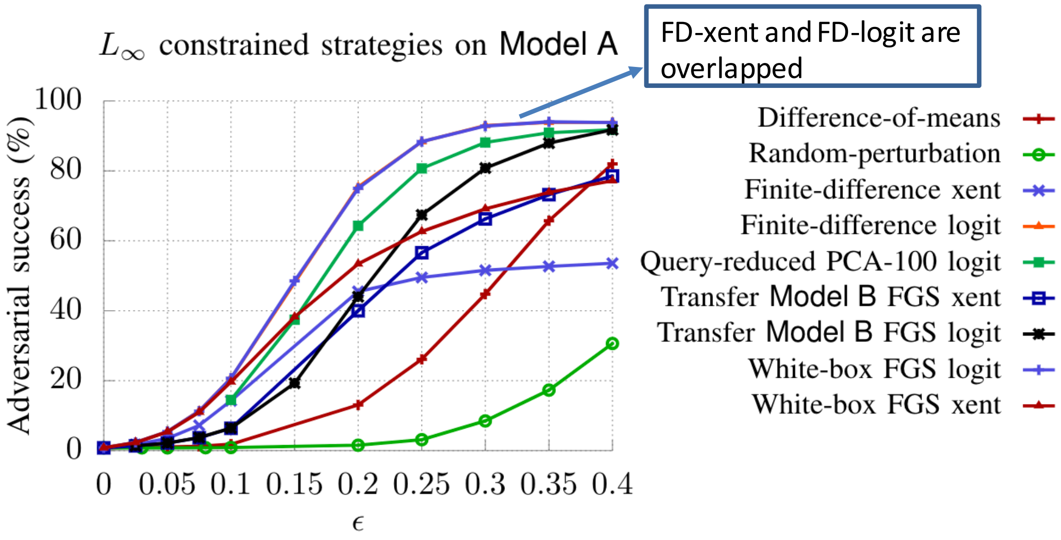
- Finite difference gradient estimation

$$\mathrm{FD}_{\mathbf{x}}(g(\mathbf{x}),\delta) = \begin{bmatrix} \frac{g(\mathbf{x}+\delta \mathbf{e}_1)-g(\mathbf{x}-\delta \mathbf{e}_1)}{2\delta} \\ \vdots \\ \frac{g(\mathbf{x}+\delta \mathbf{e}_d)-g(\mathbf{x}-\delta \mathbf{e}_d)}{2\delta} \end{bmatrix}$$

- An example of approximate FGSM with finite difference

$$x_{adv} = \mathbf{x} + \epsilon \cdot \mathrm{sign}\left(\mathrm{FD}_{\mathbf{x}}(\ell_f(\mathbf{x}, y), \delta)\right)$$

- Similar attack success rate with white-box attack

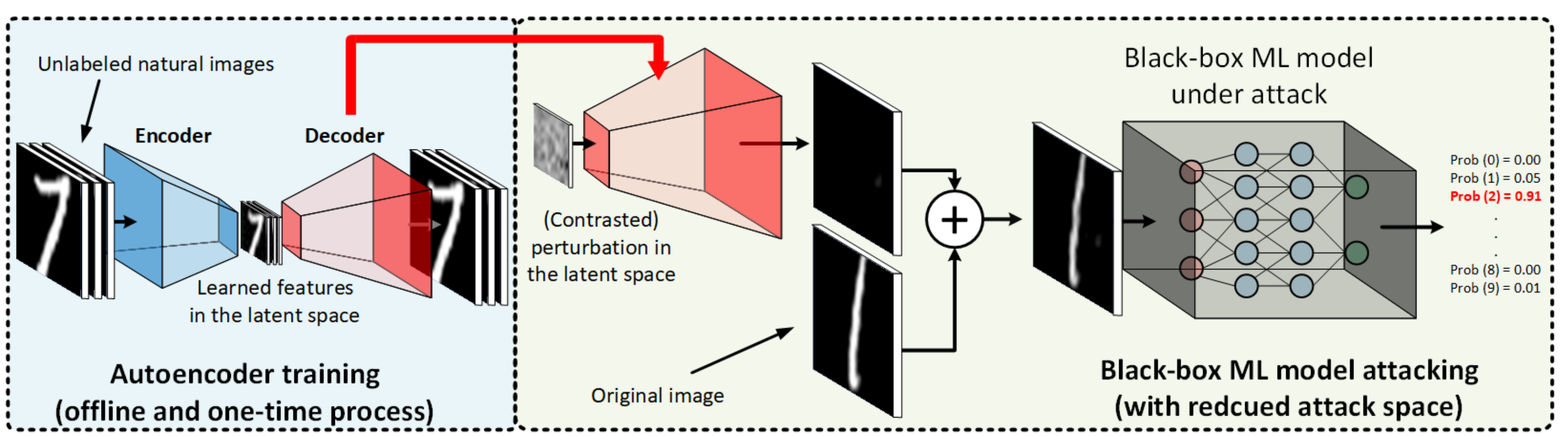


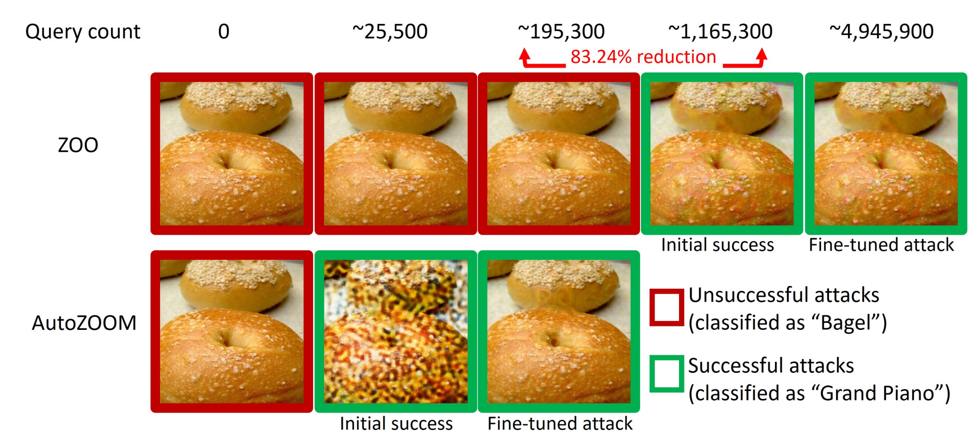
AutoZOOM

- Scaled random full gradient estimation for efficient query

$$i)\quad \mathbf{g} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = b \cdot \frac{f(\mathbf{x} + \beta \mathbf{u}) - f(\mathbf{x})}{\beta} \cdot \mathbf{u},\ \mathbf{u} \text{ is a unit-lenght vector} \qquad ii)\quad \bar{\mathbf{g}} = \frac{1}{q} \sum_{j=1}^{q} \mathbf{g}_j$$

- Autoencoder for dimensional reduction of perturbations





(Chun-Chen Tu et al., AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks, AAAI-19)

Summary of Attack Methods

Poisoning Attack

Adversarial Backdoor Embedding (Tan and Shokri, 2019)

Backdoor Attack (Gu, et. al., 2017)

Poisoning Attack on Support Vector Machines (SVM) (Biggio et al., 2013)

Clean Label Feature Collision Attack (Shafahi, Huang et. al., 2018)

White-Box

Auto-PGD (Croce and Hein, 2020)
Wasserstein Attack (Wong et al., 2020)
Targeted Universal Adversarial Perturbations (Hirano and Takemoto, 2019)
Projected Gradient Descent (PGD) (Madry et al., 2017)
Elastic Net (Chen et al., 2017)
Universal Perturbation (Moosavi-Dezfooli et al. 2016)
Feature Adversaries (Sabour et al. 2016)
DeepFool [Moosavi-Dezfooli et al., CVPR 2016]
L-BFGS [Szegedy et al. ICLR 2014]
FGSM [Goodfellow et al. ICLR 2015]

Evasion Attack

ZO-SVRG [Liu et. al. NeurIPS 2018]
ZO-NES [Ilyas et. al. ICML 2018]
AutoZoom [Chen et al. AAAI 2019]
ZO-signSGD [Liu et. al. ICLR 2019]
ZO-Natural Gradient Descent [Zhao et. al. AAAI 2019]
ZO-ADMM [Zhao et. al. ICCL 2019]
ZO-ADAM [Chen et. al. NeurIPS 2019]
ZO hard-label attack [Cheng et. al. ICLR 2019]
Sign-OPT [Cheng et. al. ICLR 2020]
Square Attack (Andriushchenko et al., 2020)

Black-Box

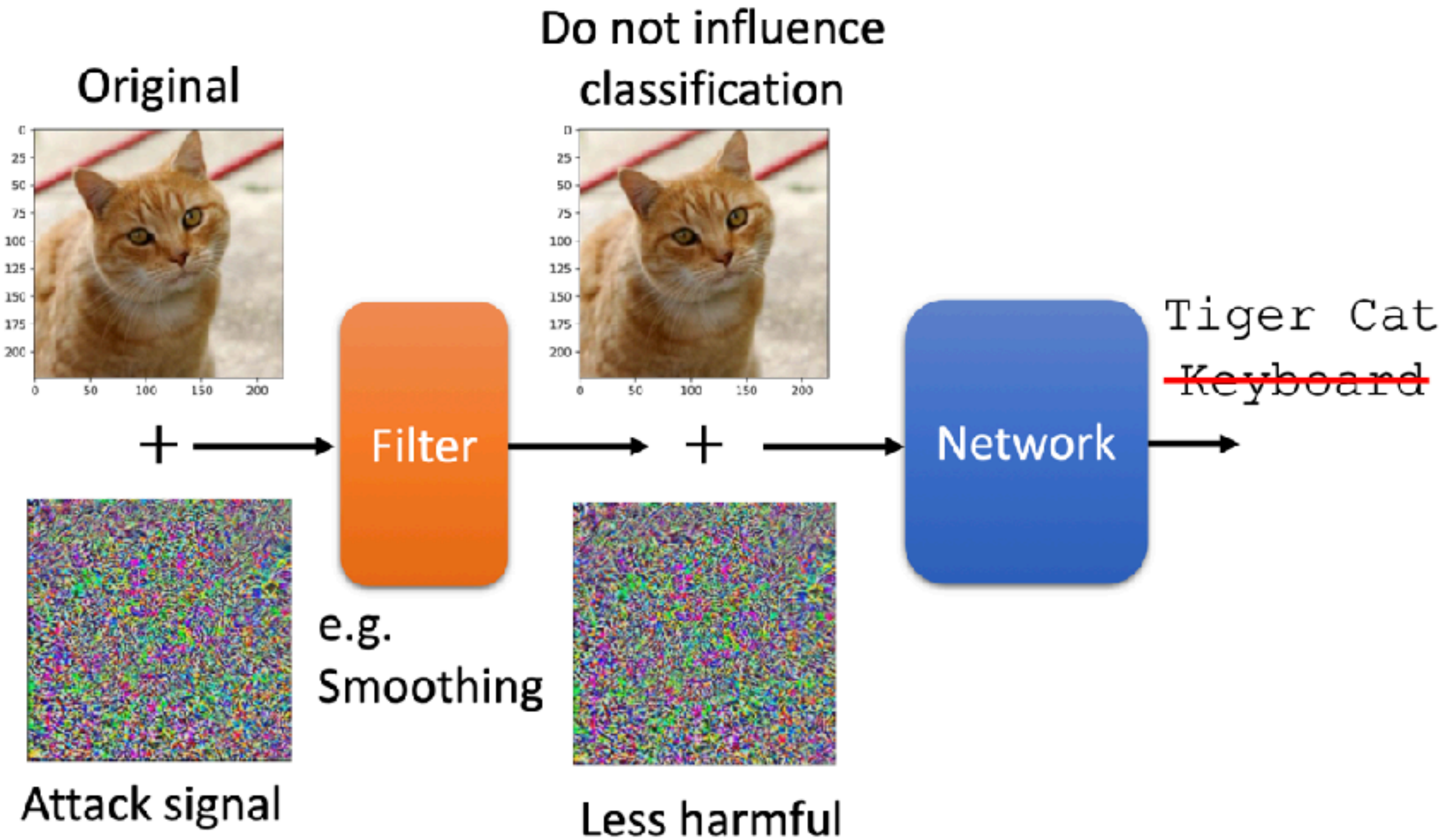
Software of Attacks

- https://github.com/bethgelab/foolbox
- https://github.com/IBM/adversarial-robustness-toolbox
- https://github.com/tensorflow/cleverhans
- https://github.com/Trusted-AI/adversarial-robustness-toolbox/wiki/ART-Attacks

Adversarial Defense

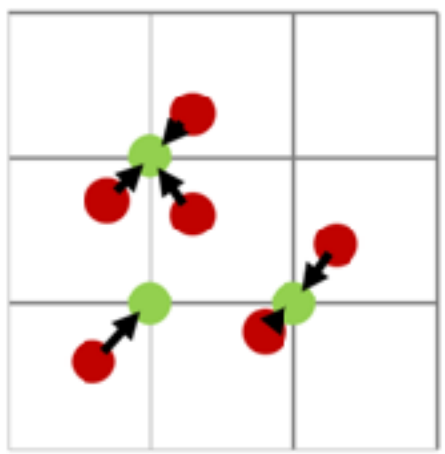
- Cannot be defensed by weight regularization, dropout and model ensemble
- Two types
  - Passive defense: Find adversarial examples without modifying the model, special case of Anomaly Detection
  - Proactive defense: Training a model that is robust to adversarial examples

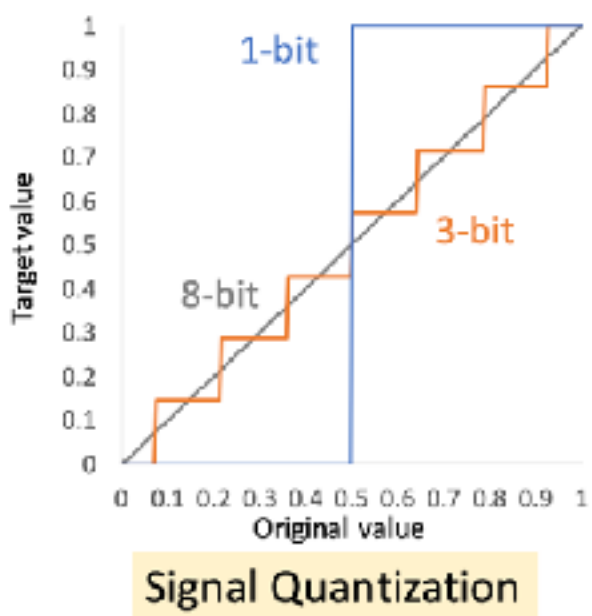
Passive Defense

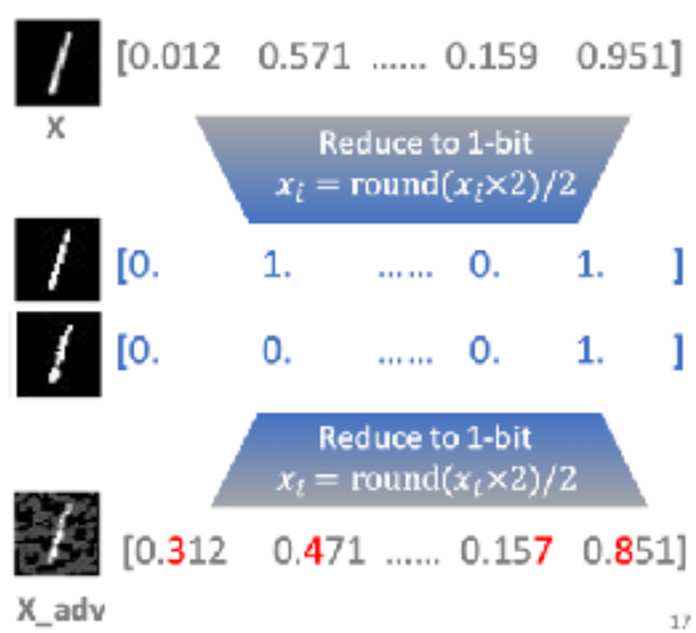


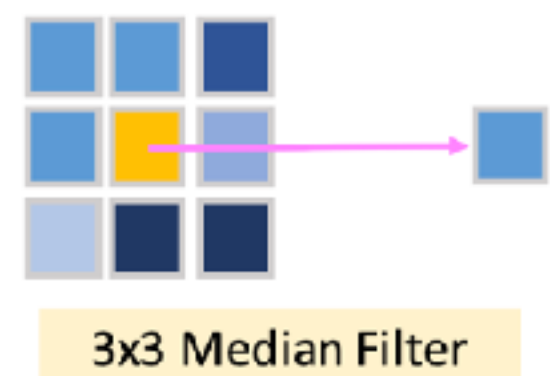
Feature Squeezing

- Goal: Detect adversarial examples
- Feature Squeezer: coalesces similar samples into a single one





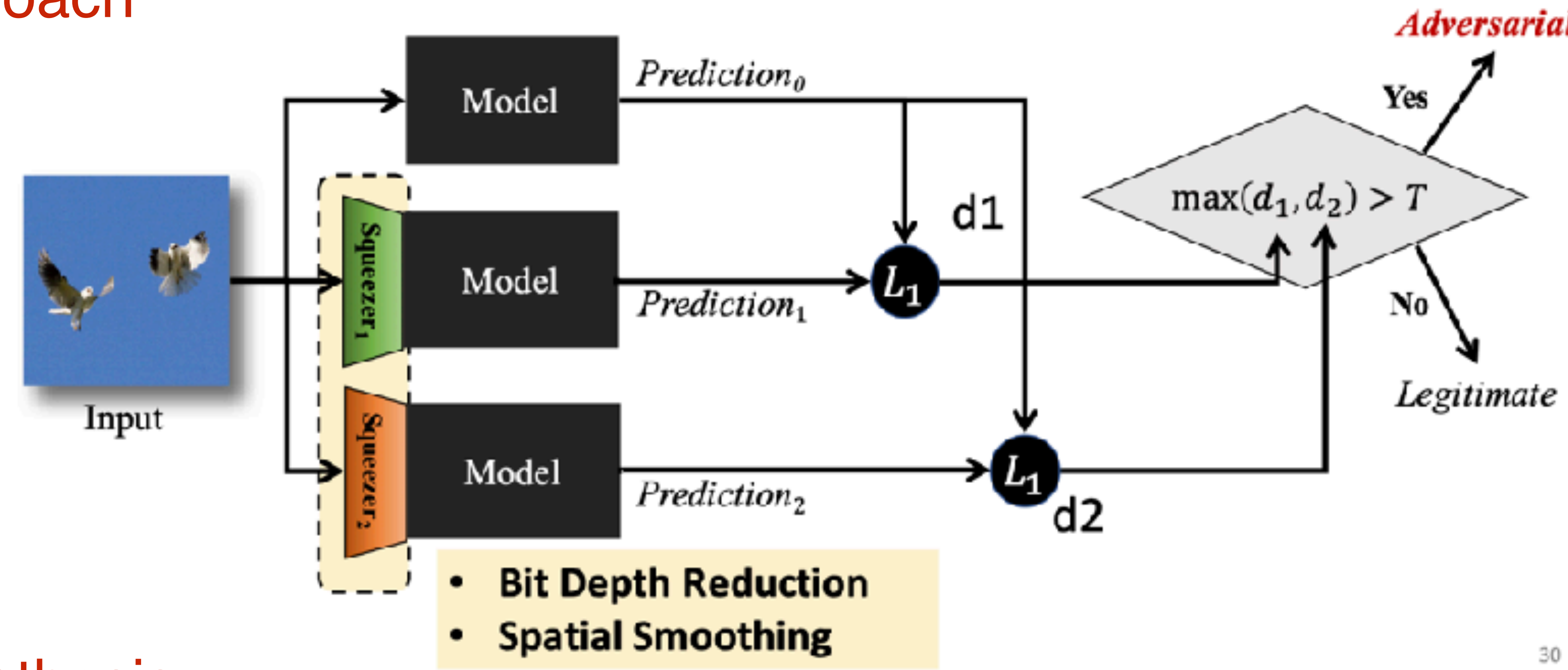




(Xu et al. NDSS 2018)
Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks

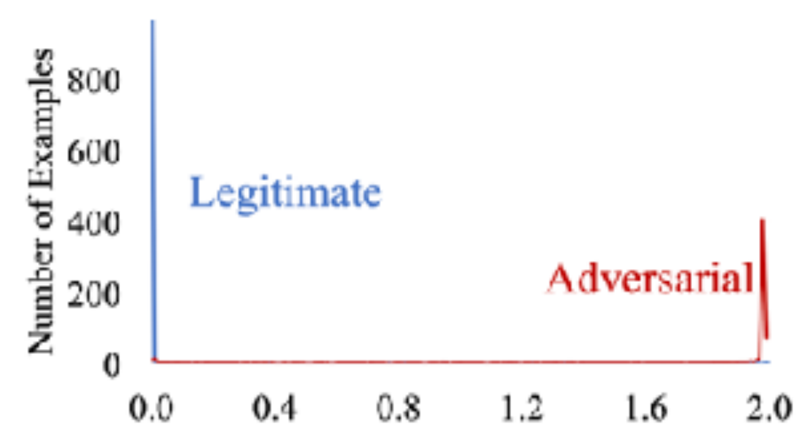
Feature Squeezing

- Approach



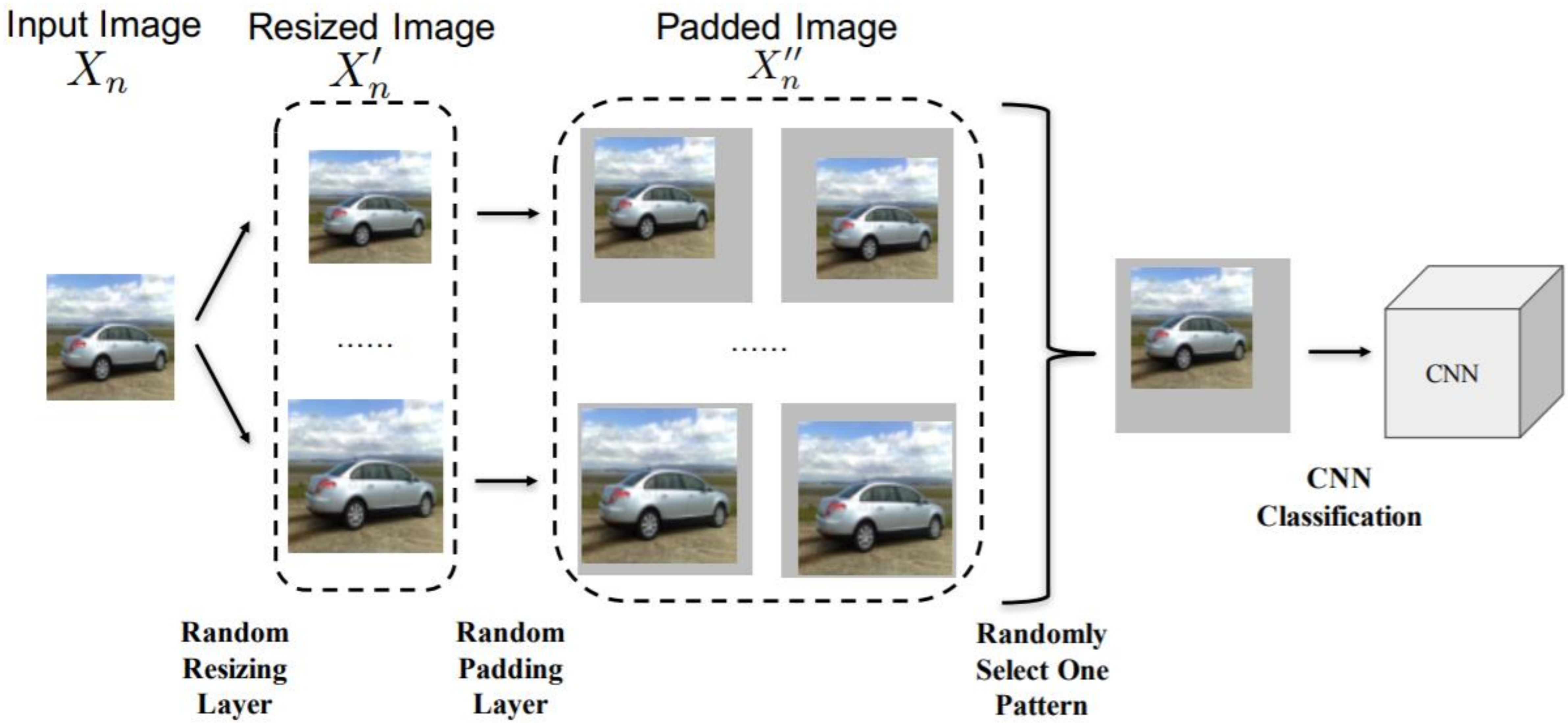
- Hypothesis
  - Feature squeezing barely change legitimate input
  - Destruct adversarial perturbations

Dataset	Squeezer	Adversarial Examples (FGSM, BIM, CW_∞, Deep Fool, CW_2, CW_0, JSMA)	Legitimate Images
MNIST	None	13.0%	99.43%
	1-bit Depth	62.7%	99.33%
ImageNet	None	2.78%	69.70%
	4-bit Depth	52.11%	68.00%



Passive Defense

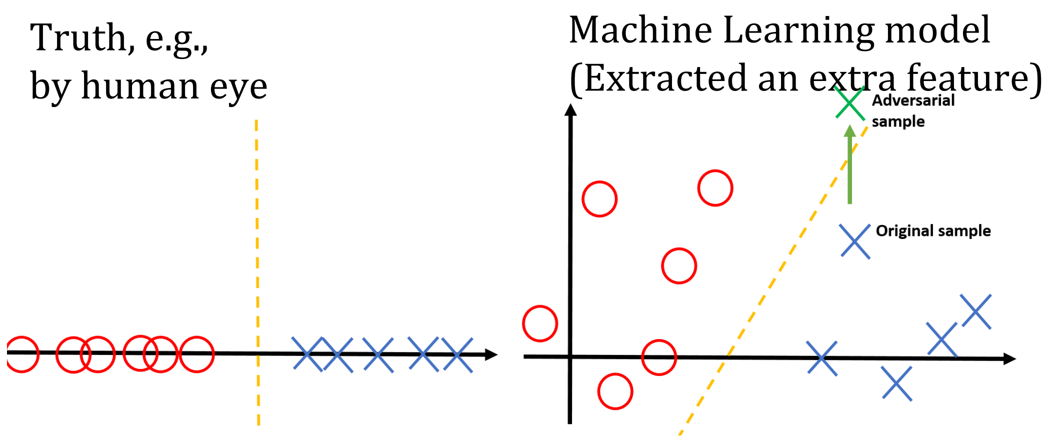
- Randomization



https://arxiv.org/abs/1711.01991

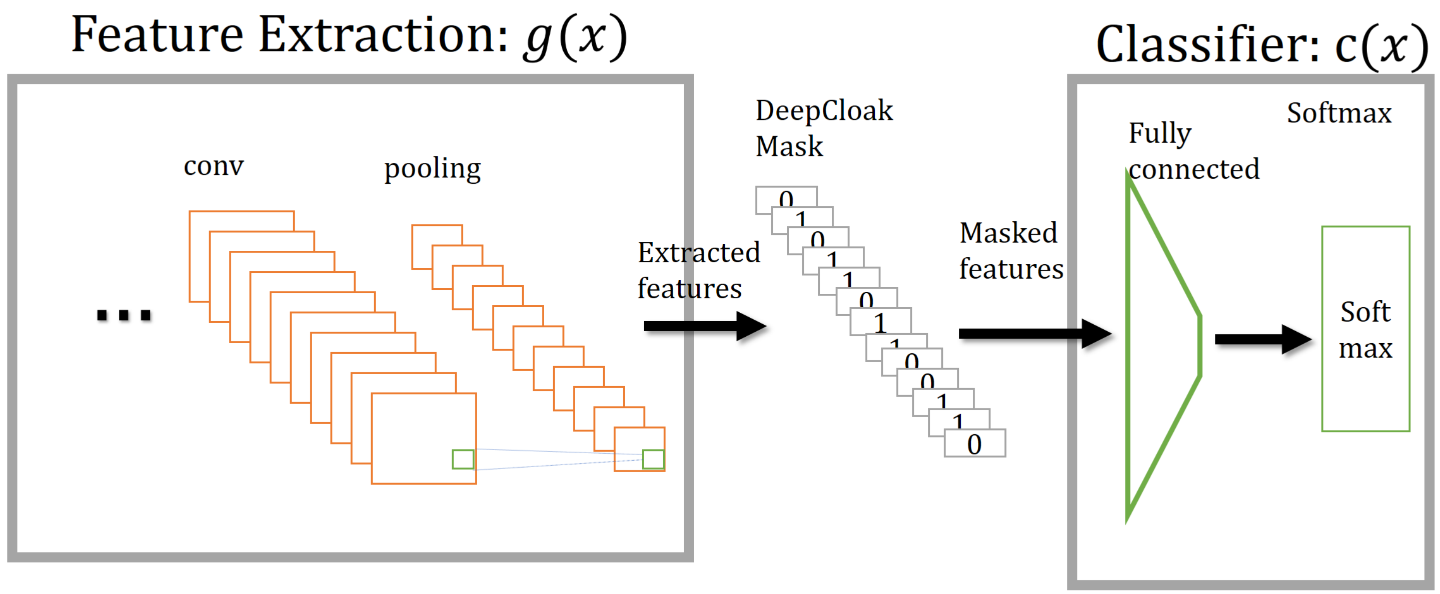
DeepCloak: Masking DNN

- Motivation: Unnecessary features in DNNs make model vulnerable



- Idea: Insert a mask layer in DNN model to remove unnecessary features

$$F(x)=g(c(x))$$



(Gao et al. DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples, ICLR 2017 workshop)

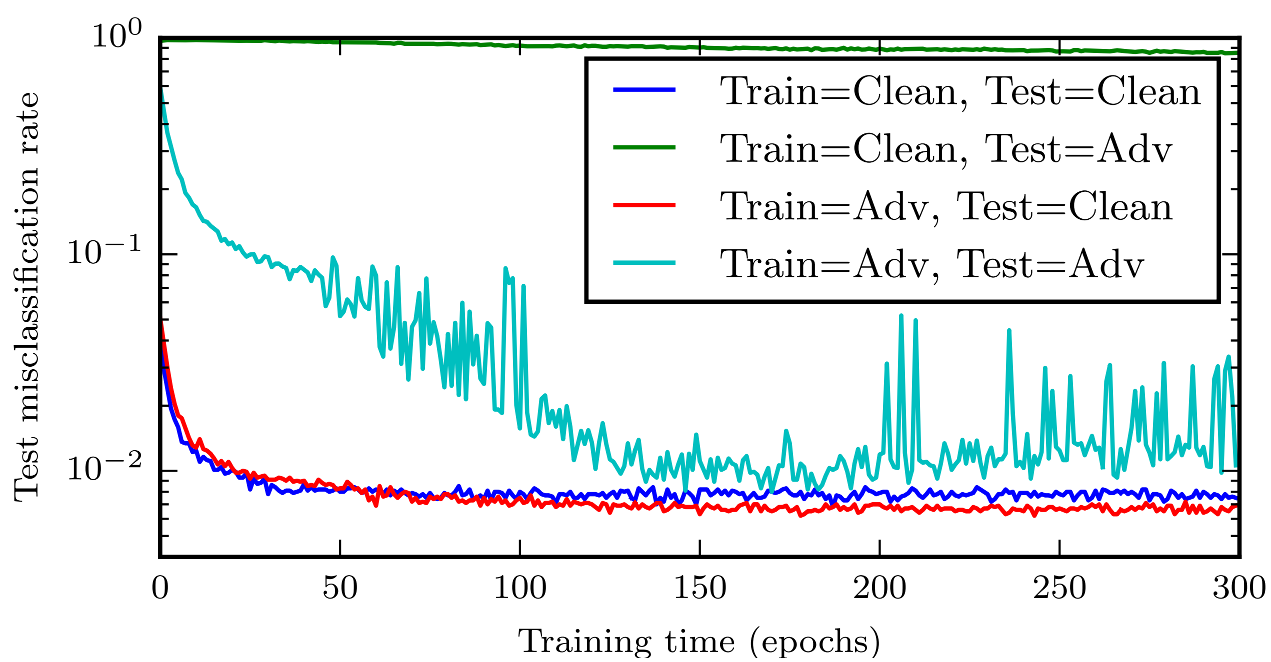
Proactive Defense: Adversarial Training

1. Choose a set of perturbations: e.g., noise of small ℓ_∞ norm: 

2. For each example  , find an adversarial example:  + 

3. Train the model on  + 

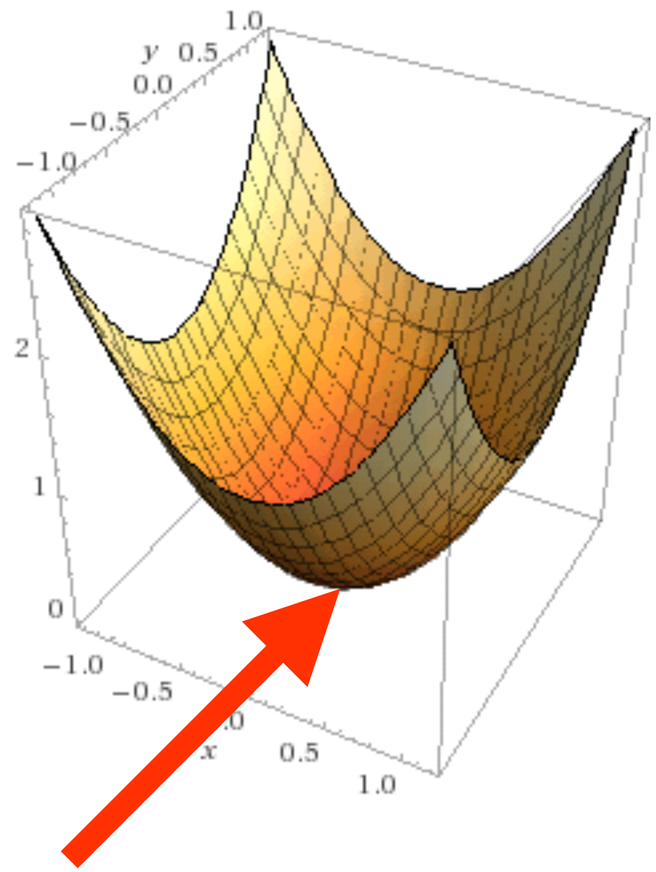
4. Repeat until convergence



Szegedy et al., 2014
Madry et al., 2017

Adversarial Machine Learning

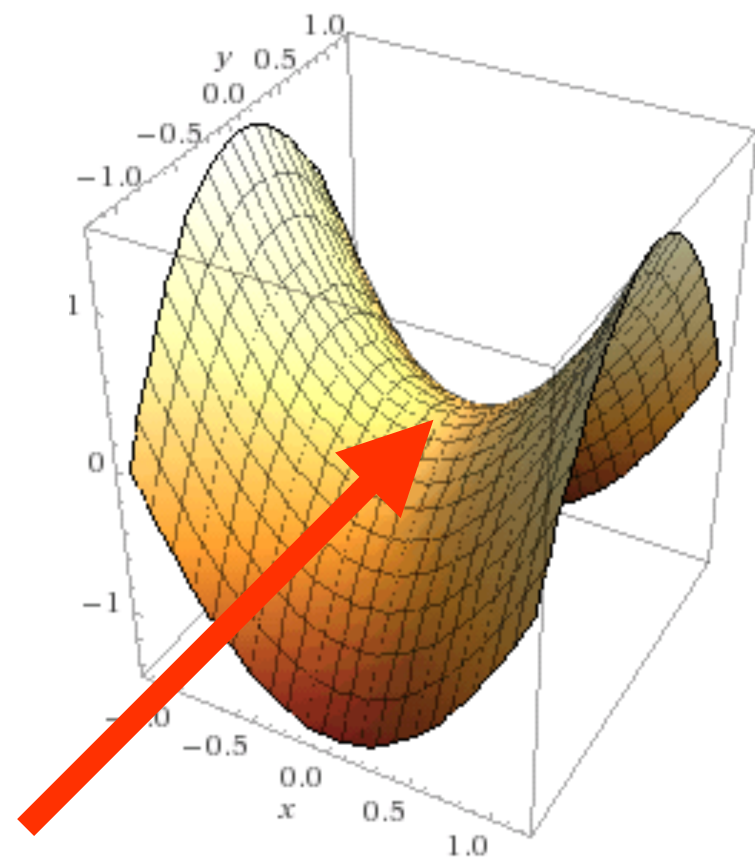
Traditional ML:
optimization



Minimum

One player,
one cost

Adversarial ML:
game theory



Equilibrium

More than one player,
more than one cost

Standard vs. Adversarial Training

- Standard training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \mathcal{L}(\theta, x, y)$$

Model Parameter

Loss

Input

Label

- Adversarial examples

$$\max_{\delta} \mathcal{L}(\theta, \boxed{x+\delta,}\, y) \quad \text{s.t.} \quad \boxed{\|\delta\|_p \leq \epsilon}$$

Loss

Adversarial Example

True Label

Keep Inperceptible

- Adversarial training as a minimax problem

$$\boxed{\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}(\theta, x, y) + \boxed{\max_{\delta} \mathcal{L}(\theta, x+\delta, y)} \right]}$$

Optimize Defense

Optimize Attack

Adversarial Training

- Adversarial training as a minimax problem

$$\boxed{\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}(\theta, x, y) + \boxed{\max_{\delta} \mathcal{L}(\theta, x+\delta, y)} \right]} \quad \text{s.t. } \|\delta\|_p \le \epsilon$$

Optimize Defense

Optimize Attack

- Be simplified as

$$\boxed{\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\boxed{\max_{\delta} \mathcal{L}(\theta, x+\delta, y)} \right]} \quad \text{s.t. } \|\delta\|_p \le \epsilon$$

Outer Minimization

Inner Maximization

Active Learning or Data Augmentation or Regularization

Adversarial Training

Inner maximization

$$\max_{\delta} \mathcal{L}(\theta, x + \delta, y) \quad \text{s.t.} \quad \|\delta\|_p \leq \epsilon$$

- Local search (lower bound on objective)
- Combinatorial optimization (exactly solve objective)
- Convex relaxation (upper bound on objective)

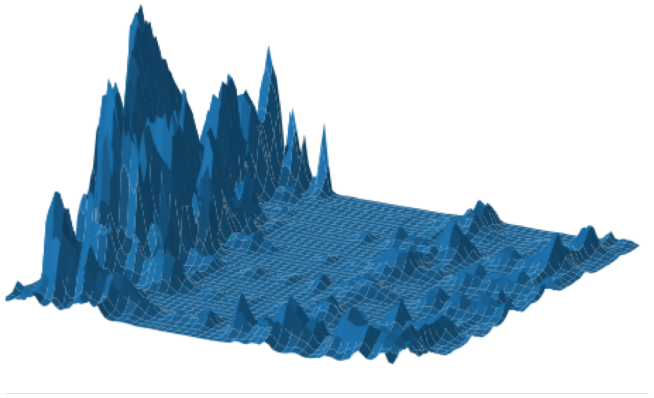
Outer maximization

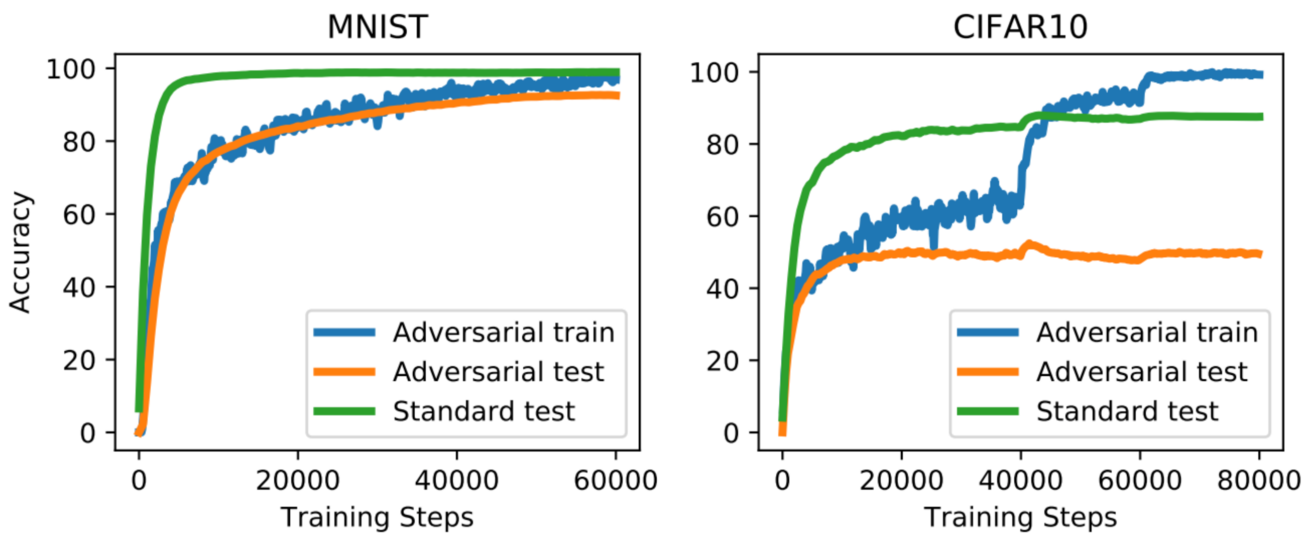
$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \, \mathcal{L}(\theta, x + \delta', y)$$

- Adversarial training
- Provably rousting training

Adversarial Robustness is Not Free

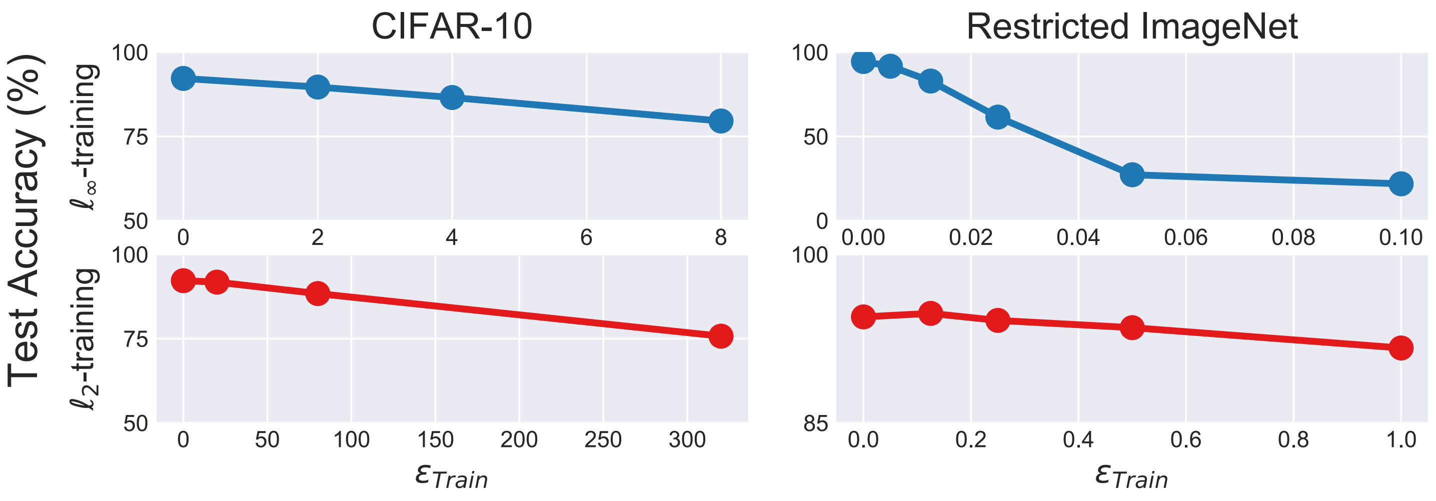
- Optimization during training more difficult and models need to be larger
- More training data might be required





(Schmidt et al., Adversarially Robust Generalization Requires More Data, NeurIPS 2018)

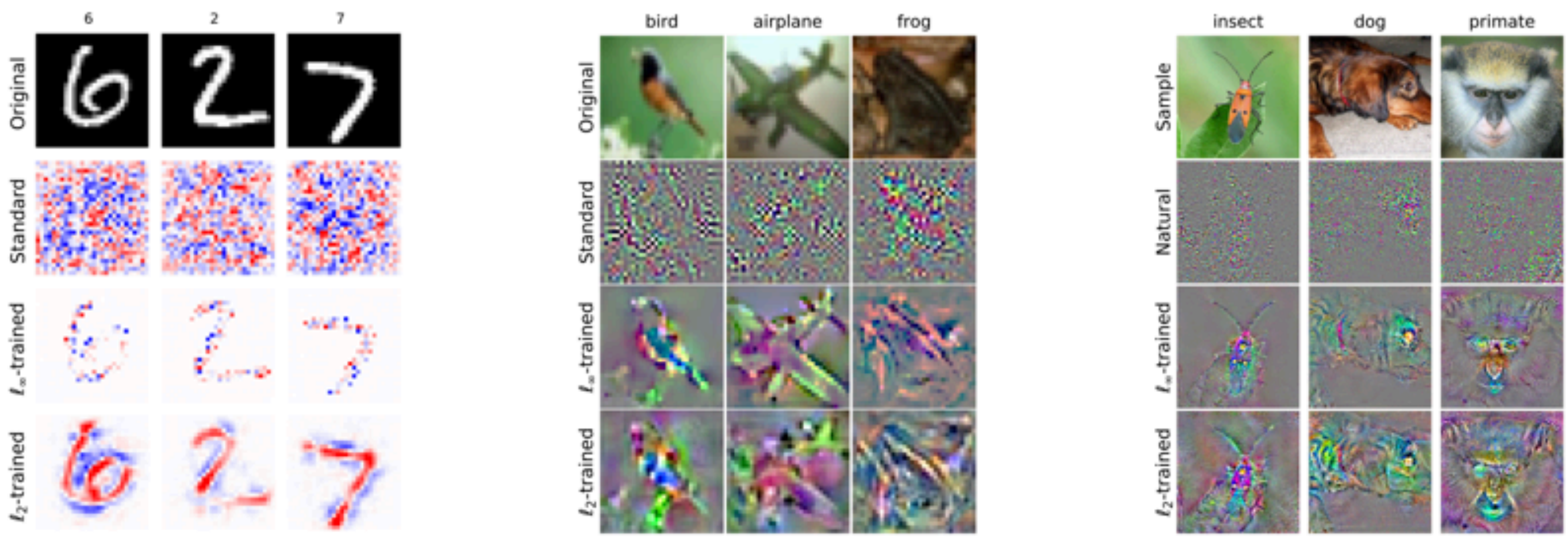
- Might need to lose on “standard” measures of performance



(Tsipras et al. 2018)

But There Are (Unexpected) Benefits

- The representation learned by robust model is more interpretable
- Align better to human perception

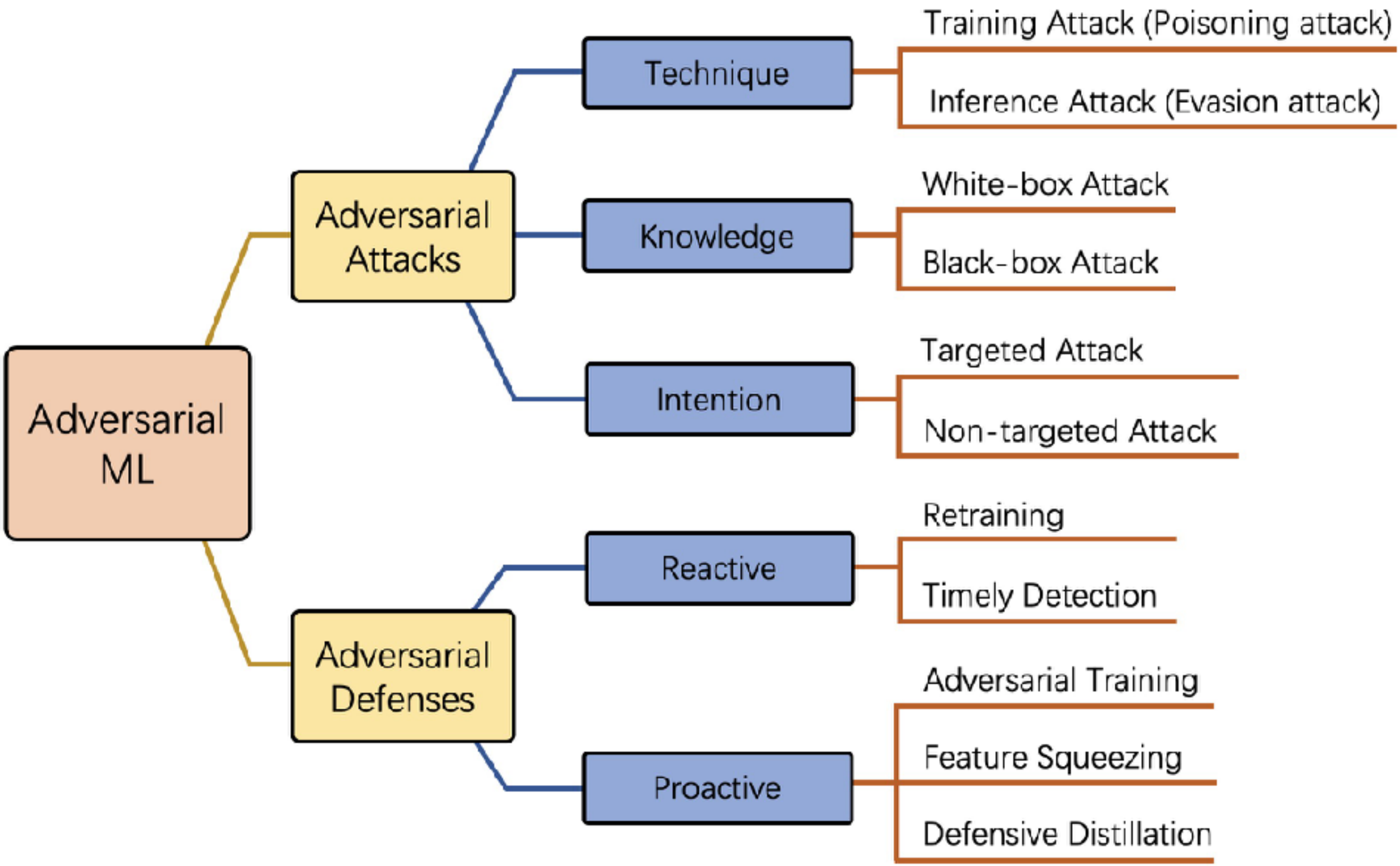


(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

Loss gradient w.r.t. input

(Tsipras et al. Robustness may be at odds with accuracy, NeurIPS 2018)

Taxonomy of Adversarial ML



How to Evaluate Adversarial Robustness?

Game-based approach

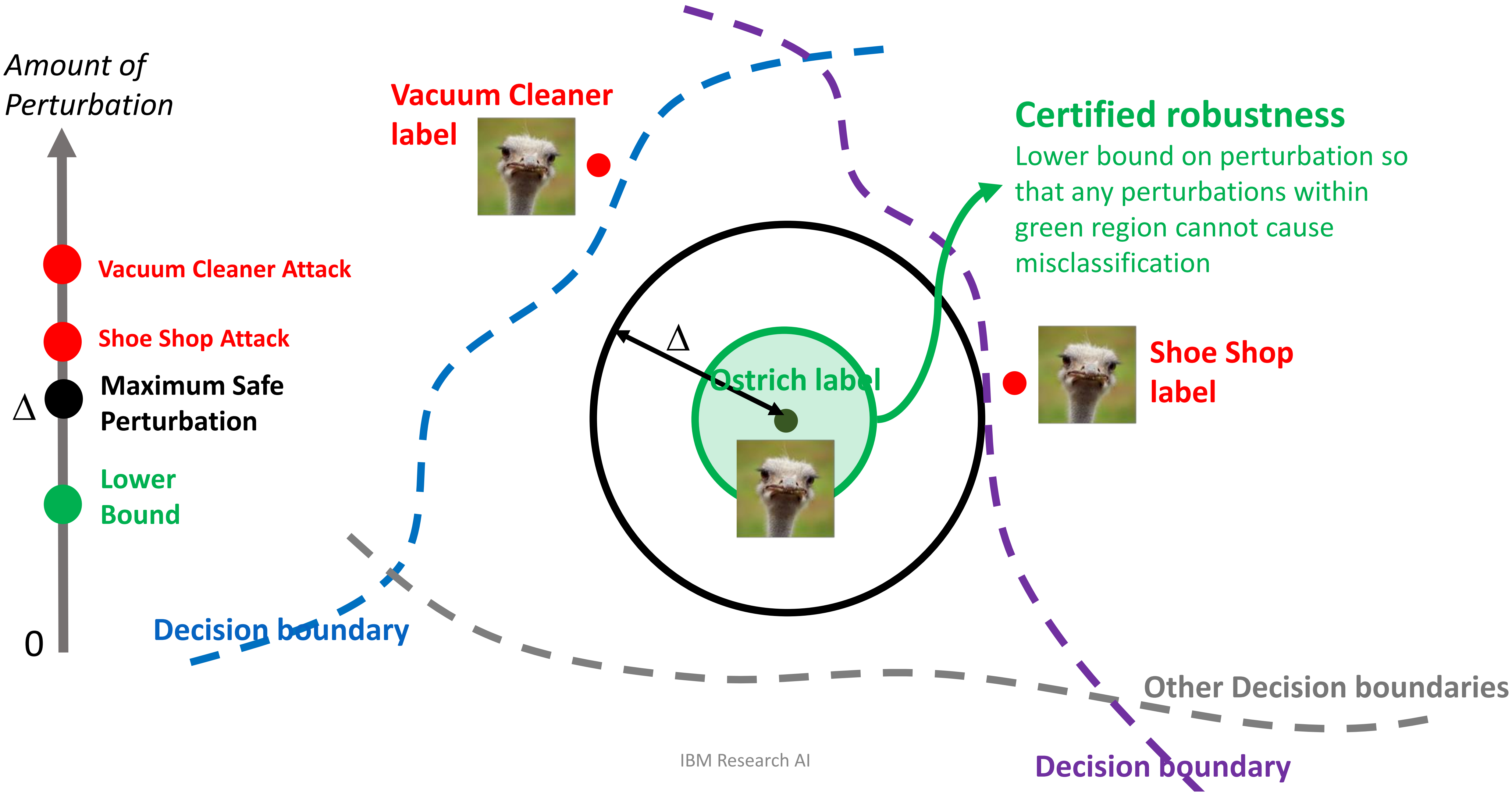
- Specify a set of players (attacks and defenses)
- Benchmark the performance against each attacker-defender pair
- No guarantee on unseen threats and future attacks

Verification-based approach

- Attack-independent: does not use attacks for evaluation
- Can provide robustness certificate: e.g., no attacks can alter the decision of the ML model if the attack strength is limited
- Optimal verification is computationally impractical for large DNN

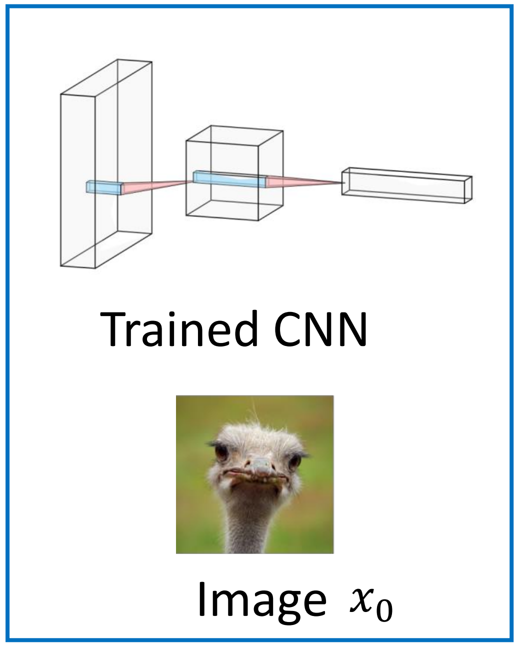
Zhang et al., Efficient Neural Network Robustness Certification with General Activation Functions, NIPS 2018

Verification: Lower Bounds on Robustness



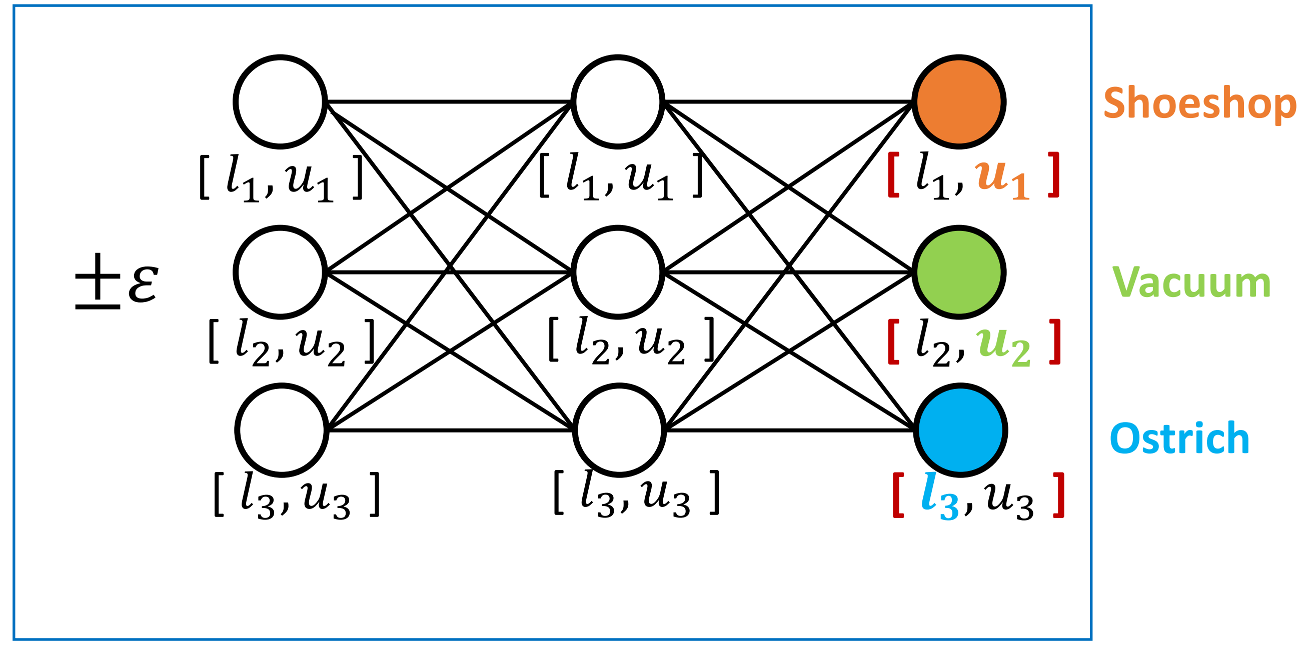
IBM Research AI

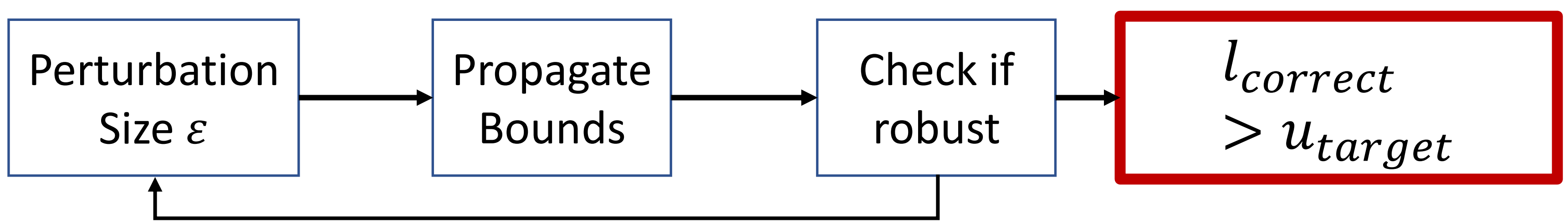
Efficient Certified Bound with Activation Bounds



Input

$\|x - x_0\| \le \varepsilon$



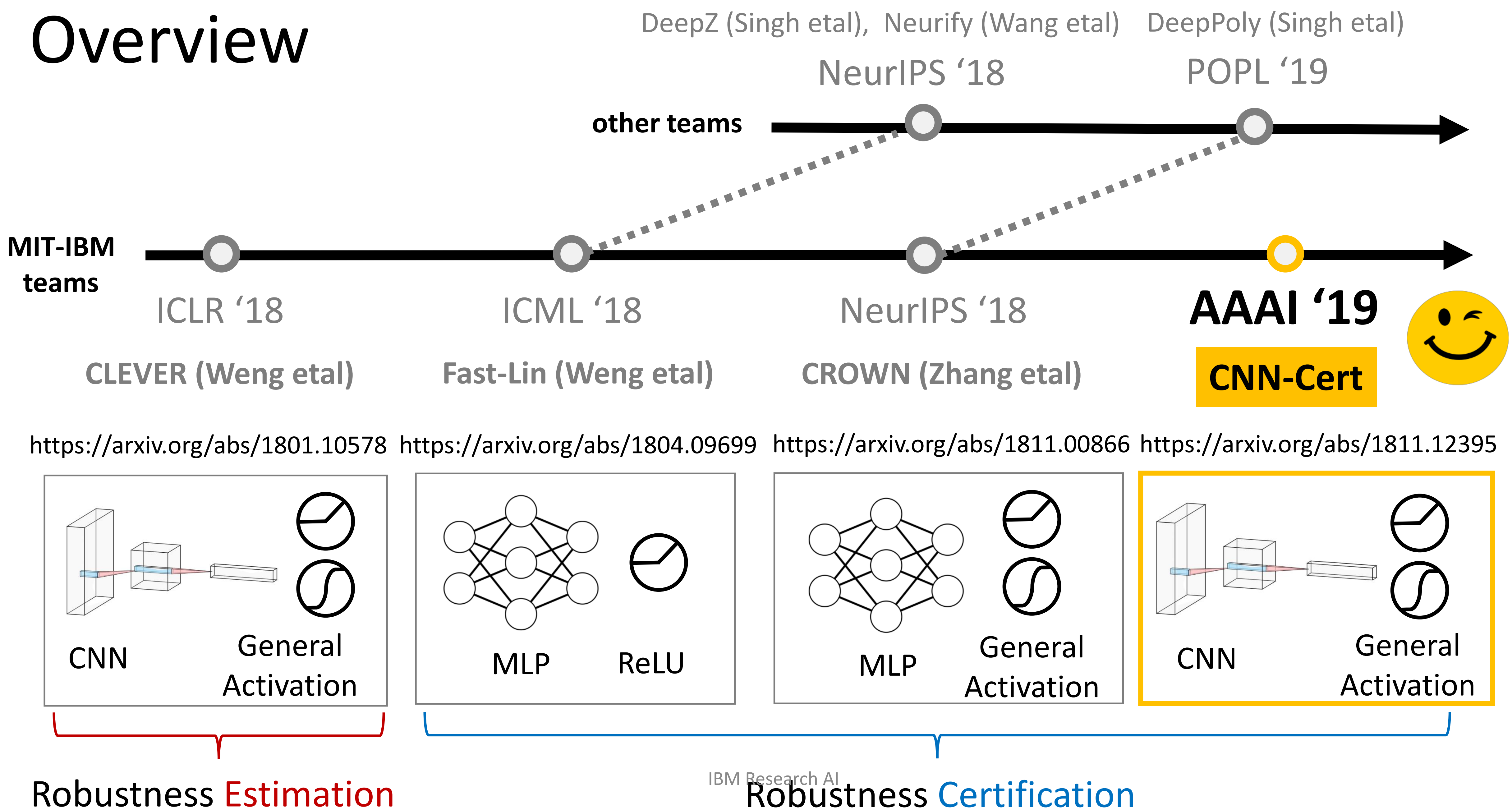


$l_{ostrich} > u_{vacuum}$

Robustness Certificate: Given a data input and a model, the top-1 prediction of the perturbed input will not be altered if the perturbation (e.g. L_p norm ball) is smaller than $\varepsilon_{certified}$

Timeline of Robustness Certification

Overview



Challenges

- How to improve the state-of-the-art adversarial training methods
- Adversarial training is effective, but not scalable and efficient
- Tradeoff between accuracy and robustness
- Understand the nature of vulnerability of DNNs
- How to evaluate and certificate model robustness
- Robustness to adaptive adversary, i.e. attack-agnostic defense
- Need for human-like machine perception and understanding