

Announcements

- HW 6 due **Wednesday, April 19 at 8pm**

Lecture 24: Robustness

CIS 4190/5190

Spring 2022

Agenda

- **Interpretability & Explainability**
- **Robustness to distribution shift**
- **Robustness to adversarial attacks**

Interpretability & Explainability

- **Interpretability:** How does the model make predictions?
 - Useful for debugging issues with the model
 - Not feasible for deep neural networks
- **Explainability:** How did the model make a specific prediction?
 - “Local” interpretation that can still be very useful for debugging

Input Gradients

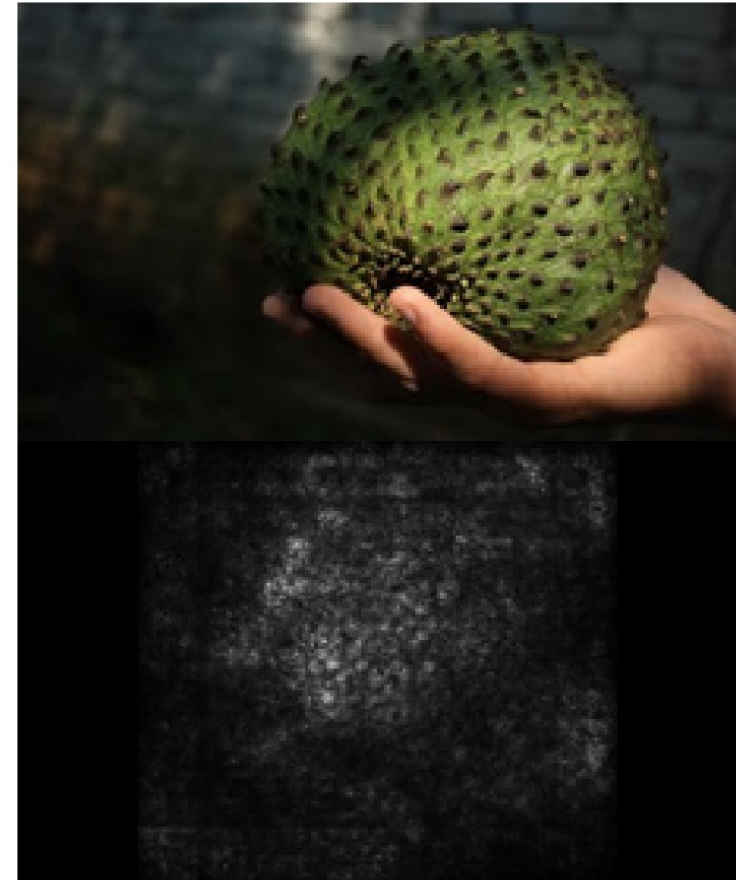
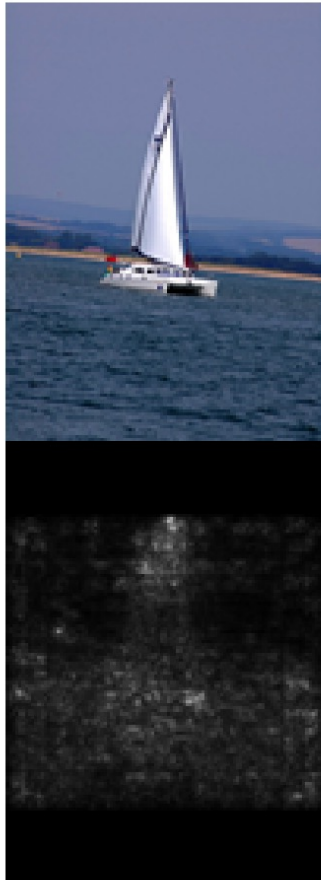
- Consider the gradient of the loss with respect to the input:

$$s = \nabla_x \tilde{L}(f_\beta(x), y)$$

- **Intuition**

- The gradient $s_{i,j}$ captures the effect of perturbing input $x_{i,j}$ on the loss when assuming the true label is y
- Larger gradients \rightarrow more “important” feature
- **Note:** y does not need to be the true label!

Saliency Maps



Simonyan et al., Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. 2013

Lots of Modifications

- **Guided backpropagation:** Zero out negative signals in backward pass
- **Integrated gradients:** Average over range of gradients
- **Local explanations:** Use sampling + fit model instead of gradient

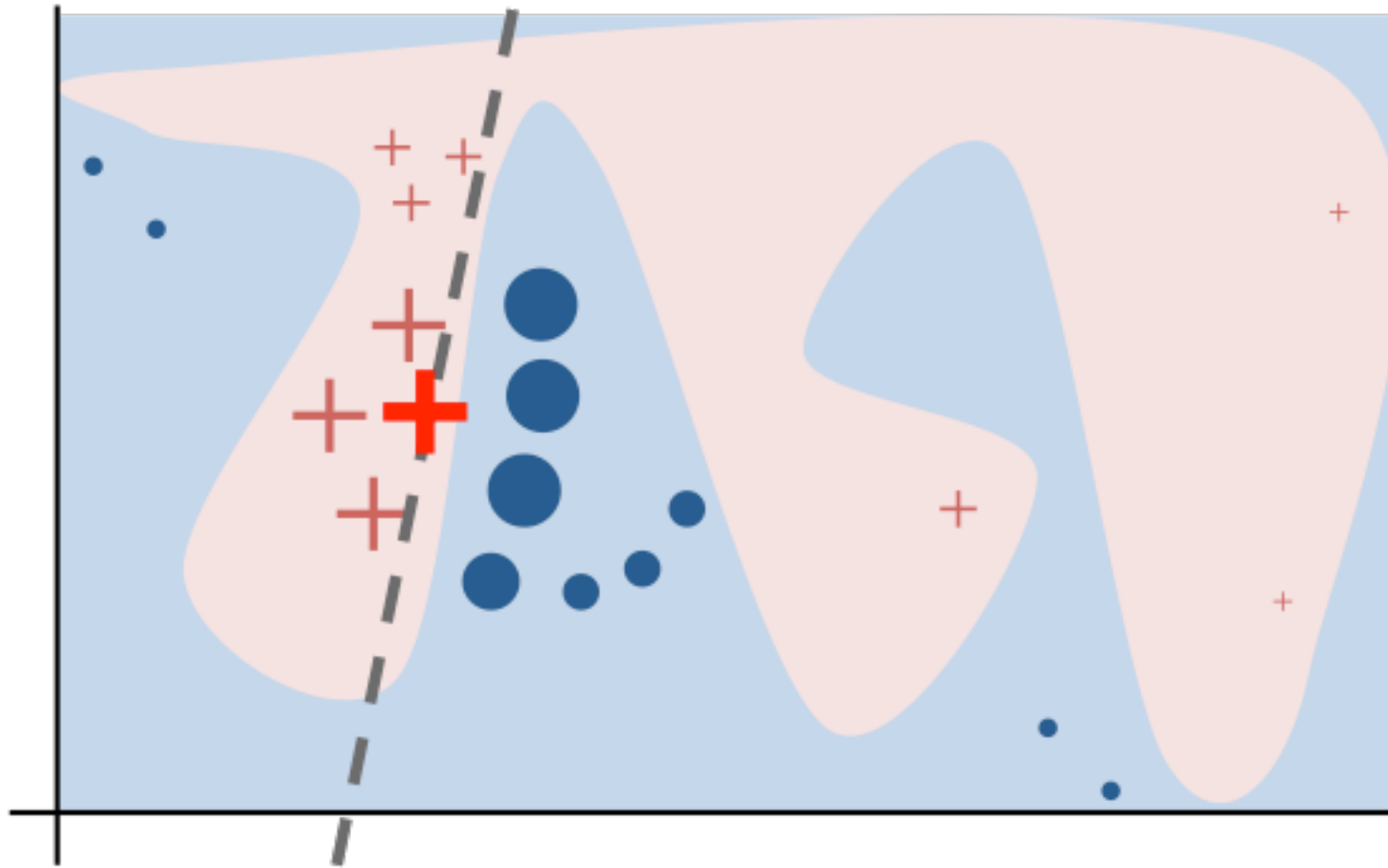
Local Explanations

- Construct dataset

$$\mathbf{Z} = \left(\mathbf{x} + \boldsymbol{\epsilon}, f_{\beta}(\mathbf{x} + \boldsymbol{\epsilon}) \right)$$

- Here, $\boldsymbol{\epsilon} \sim N(0, \sigma^2)$ is i.i.d. Gaussian noise
- Fit a linear model to this dataset \mathbf{Z}
- “Smoothed” saliency maps (recover saliency maps as $\sigma \rightarrow 0$)

Local Explanations



Ribeiro et al., "Why Should I Trust You? Explaining the Predictions of Any Classifier", 2016

Local Explanations



(a) Original Image

(b) Explaining *Electric guitar*

(c) Explaining *Acoustic guitar*

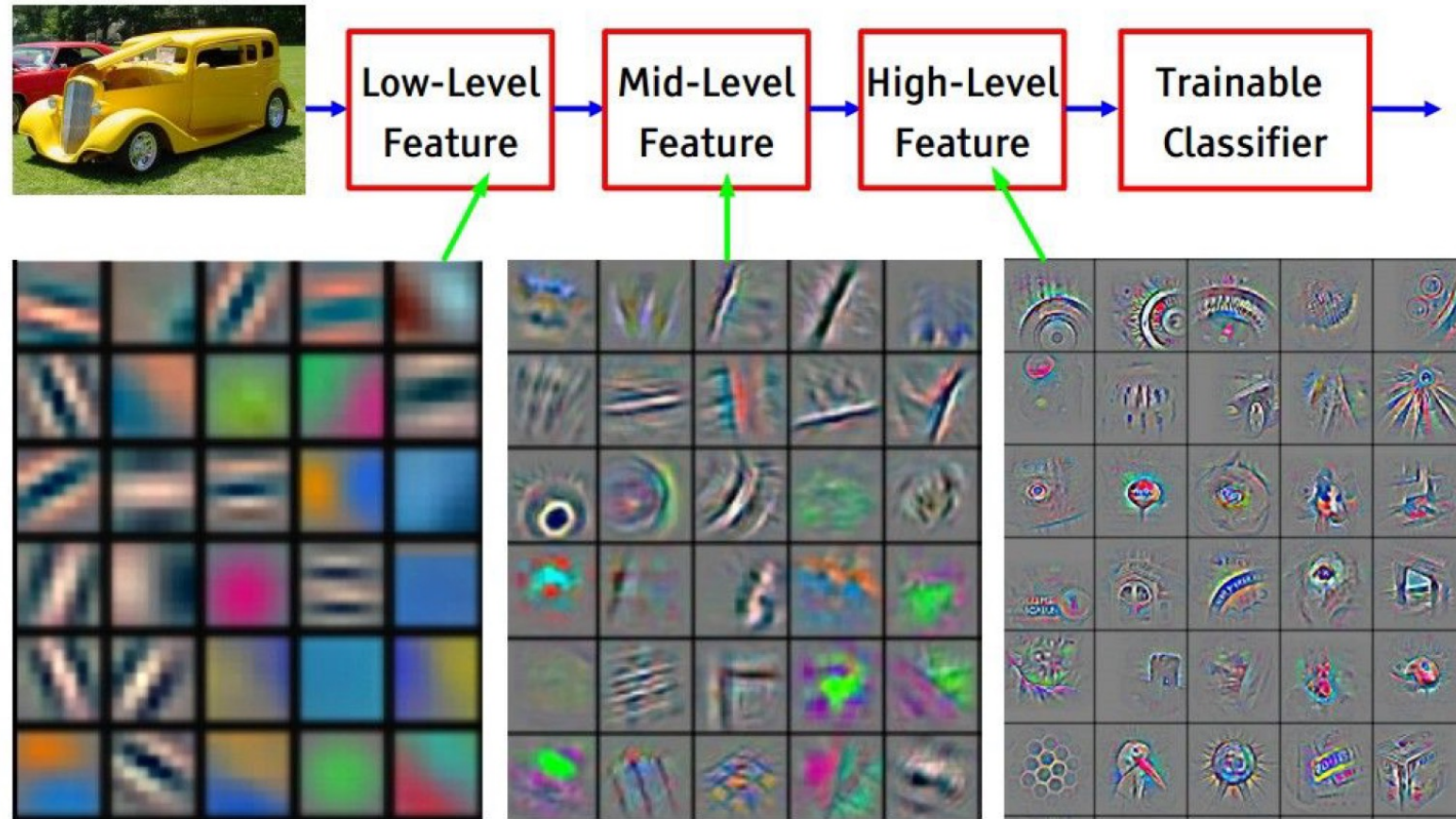
(d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google’s Inception network, highlighting positive pixels. The top 3 classes predicted are “Electric Guitar” ($p = 0.32$), “Acoustic guitar” ($p = 0.24$) and “Labrador” ($p = 0.21$)

Neuron Visualization

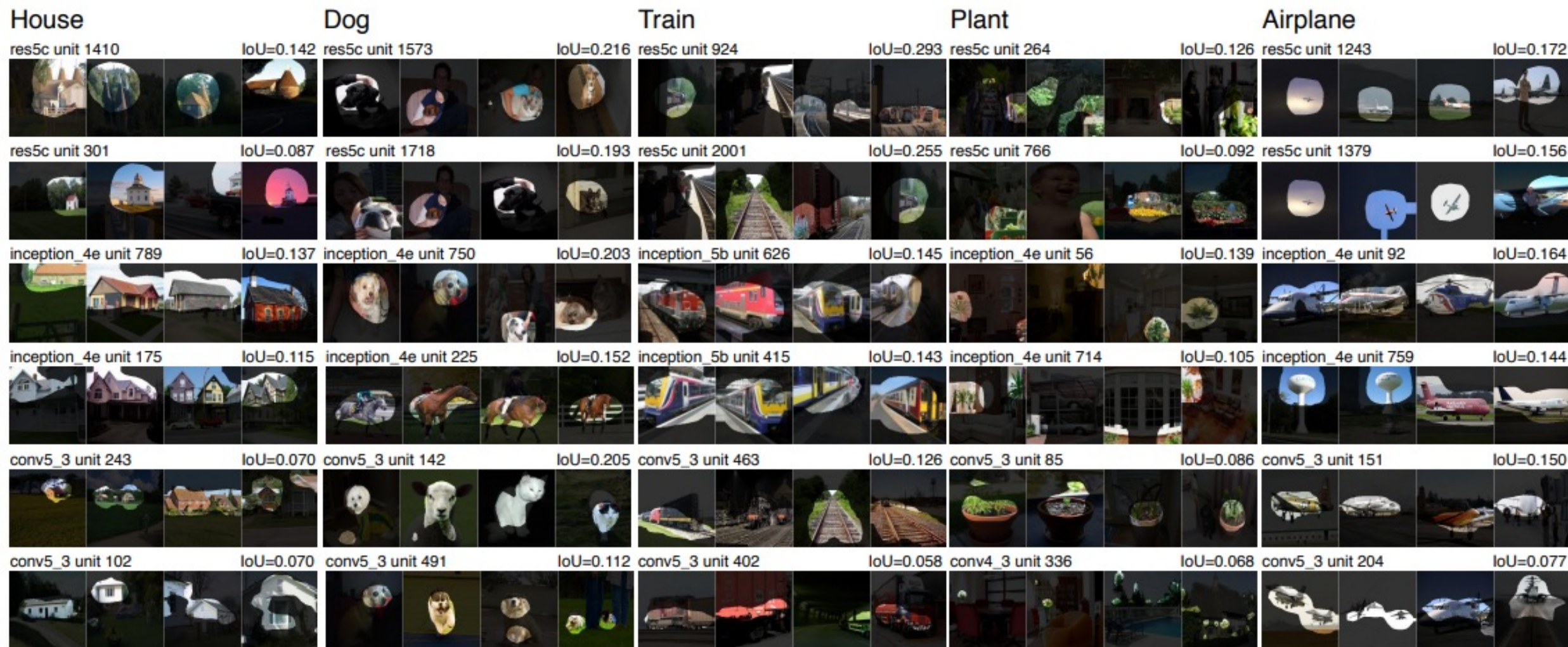
- **Neuron visualization:** Look at $\nabla_x g_\beta(x)$ for an intermediate layer g_β
- **Network dissection:** Look at groups of pixels corresponding to objects

Neuron Visualization



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

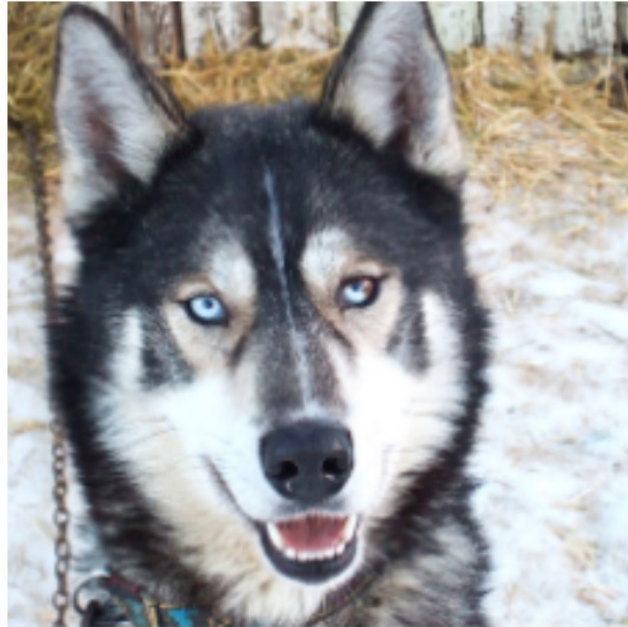
Neural Network Dissection



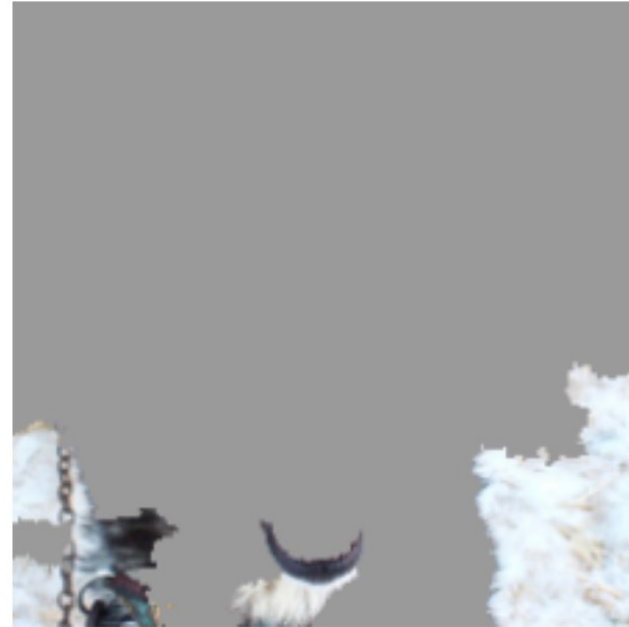
Why Are Explanations Useful?

- Models do not always use the information we expect them to!

An Interesting Local Explanation



(a) Husky classified as wolf



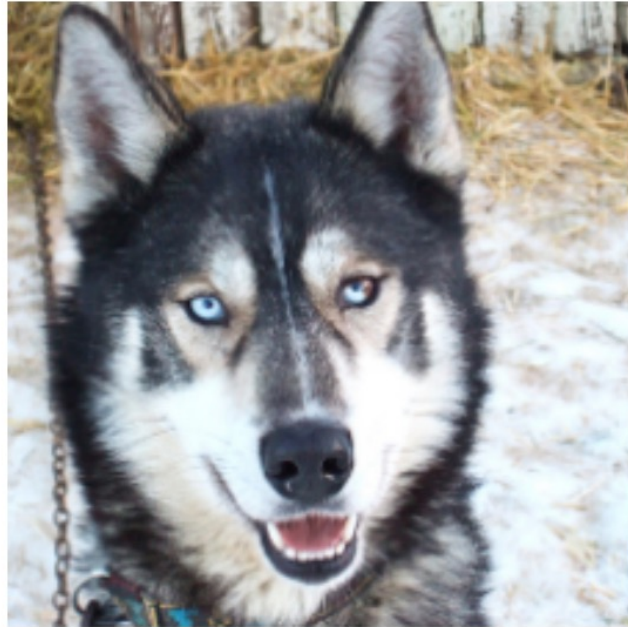
(b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the “Husky vs Wolf” task.

Correlated Inputs/Features

- Suppose two features x_1 and x_2 are highly correlated
- Which one should the model use to predict the label y ?
 - Doesn't make a difference!

Correlated Inputs/Features



(a) Husky classified as wolf



(b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

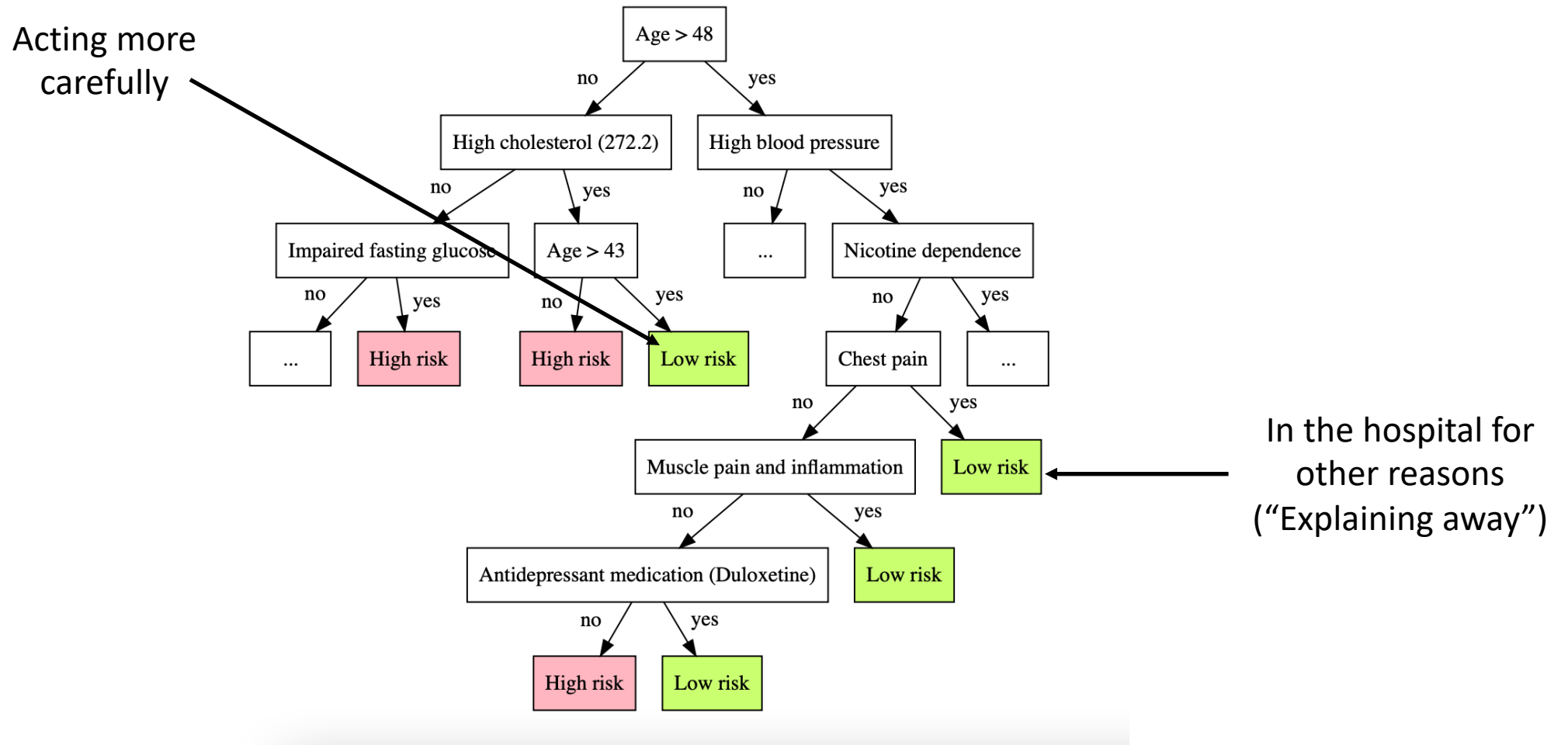
Problematic Correlations

- In practice, unexpected features can be correlated with the output
- **Example**
 - Model predicts “has asthma” → “lower pneumonia risk”
 - Why?
- **Explanation**
 - A patient who has asthma is more careful and receives better medical care
 - **Patients with asthma have better outcomes for pneumonia!**
 - **Does not mean we should label asthma patients as lower risk!**

Example: Diabetes prediction

- **Input:** ~400 patient features (e.g., lab tests, current medications, etc.)
- **Label:** Does the patient have diabetes?
- Train a decision tree to solve this problem

Example: Diabetes prediction



Example: Chest X-Rays

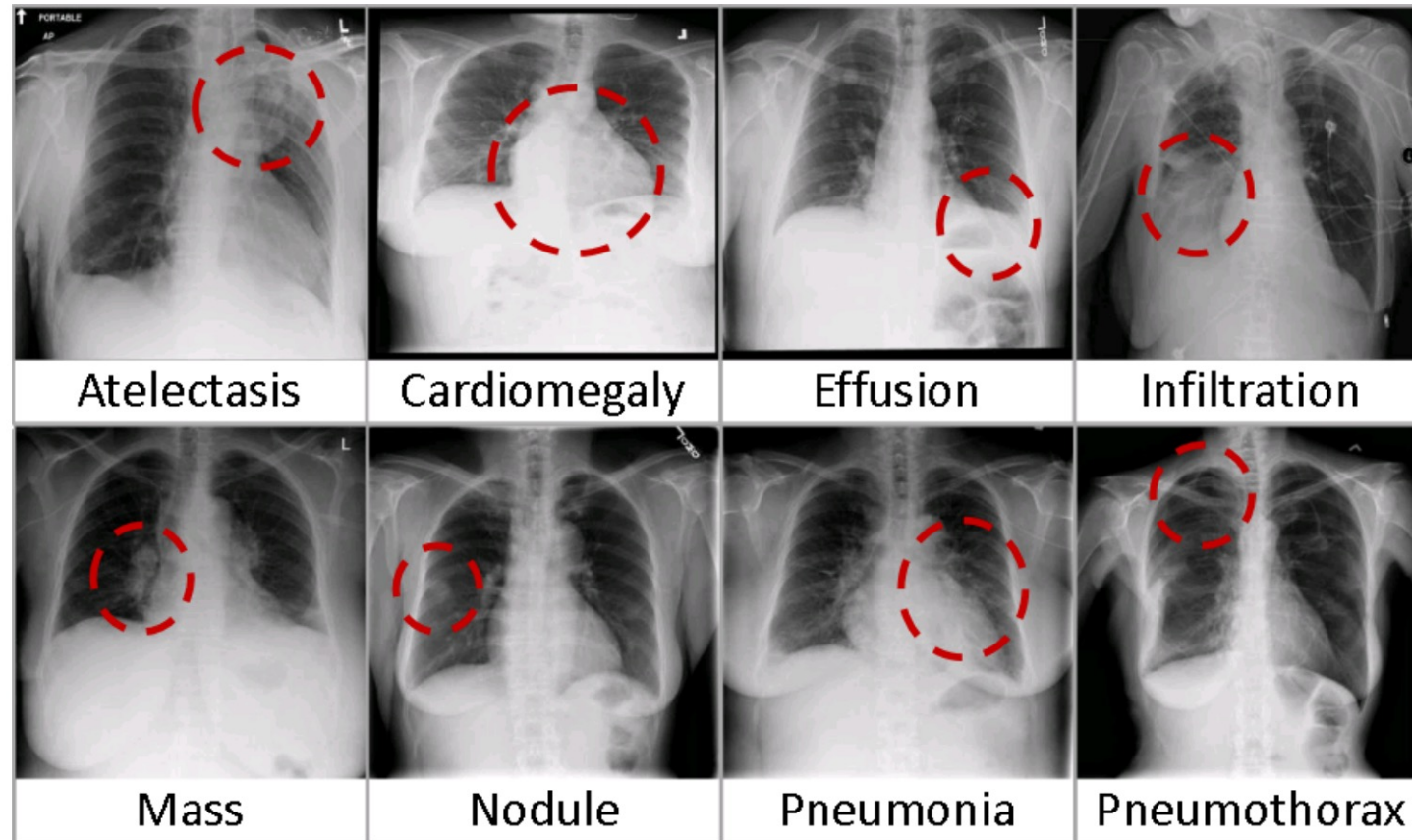


Figure 1. *Eight common thoracic diseases observed in chest X-rays that validate a challenging task of fully-automated diagnosis.*

Example: Chest X-Rays

- **Task:** Diagnose pneumothorax from chest x-ray
- **Problem:** Some of the patients were already treated!
 - Treatment is visible in chest x-ray
 - Deep neural network is predicting who was already treated!

Potential Solutions

- **No general solutions (yet)**
- **Good practices**
 - Be very careful with data processing/cleaning
 - Use existing interpretability techniques to better understand model
 - Work closely with domain experts to examine potential data/model issues

Agenda

- **Interpretability & Explainability**
- **Robustness to distribution shift**
- **Robustness to adversarial attacks**

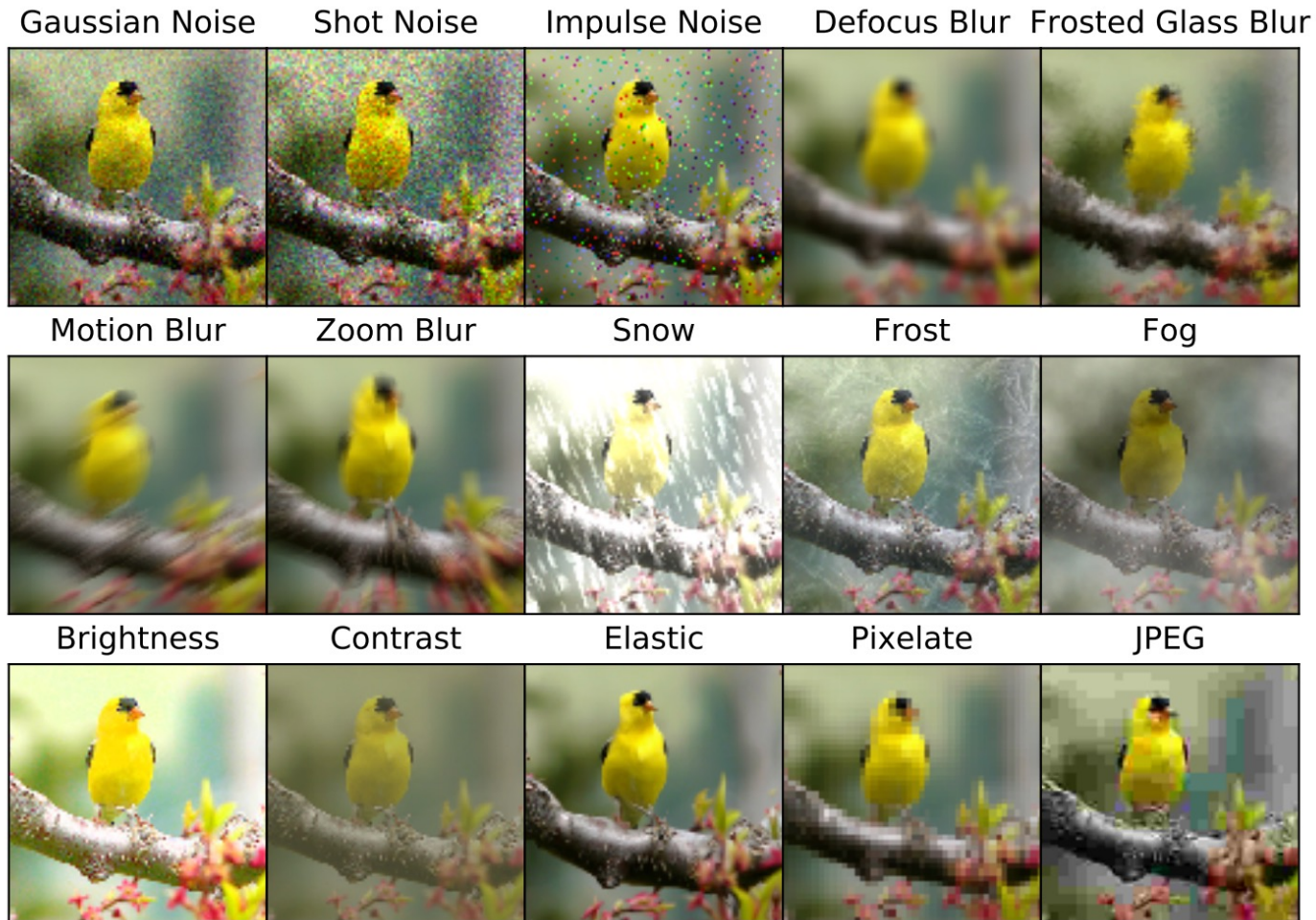
Robustness to Distribution Shift

- Neural networks generalize well **on distribution**
- **Ideal scenario**
 - Test set and training set are i.i.d. from the same distribution
 - **Equivalently:** Test set is obtained by shuffling entire dataset and then splitting
- **Often fails in practice! “Distribution shift”**

Robustness to Distribution Shift

- **Images/computer vision**
 - Added noise, color shifts, lighting changes, different resolution, etc.
- **Audio/speech-to-text**
 - Noisy background, changes in recording device, etc.
- **Natural language processing**
 - Substitute synonyms, add unrelated text, etc.

Example: Synthetic Perturbations



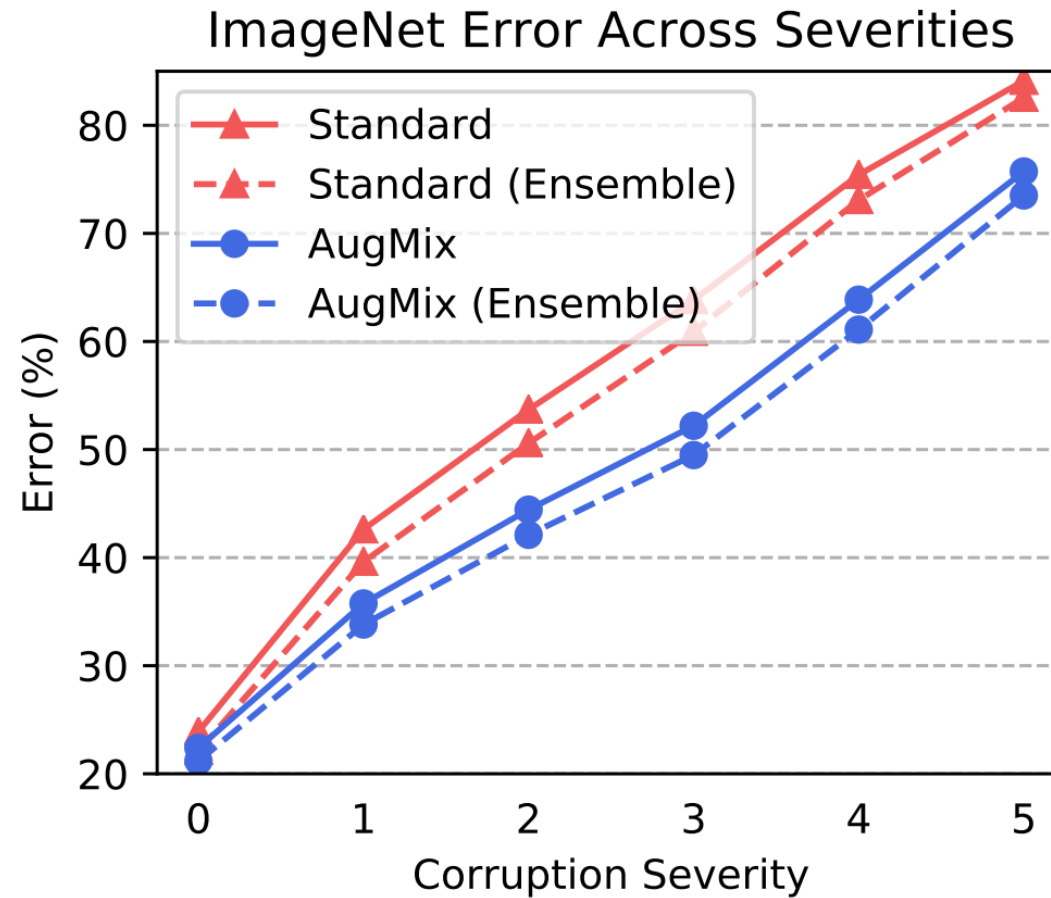
Example: Synthetic Perturbations

- **Question:** Why should the model be robust?
- **Answer:** Humans are robust!

Example: Synthetic Perturbations

- **Significantly reduces performance**
 - 20% error rate → 80% error rate
- **Data augmentation can help (but not 100% solution)**

Example: Synthetic Perturbations



Example: Natural Language Processing

Article: Super Bowl 50











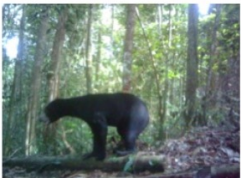
Paragraph: *“Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”*

Question: *“What is the name of the quarterback who was 38 in Super Bowl XXXIII?”*






Original Prediction: John Elway

Prediction under adversary: Jeff Dean

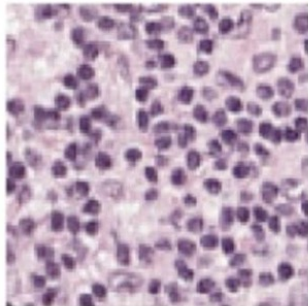
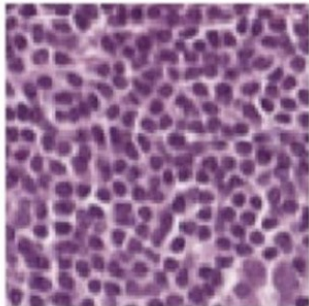
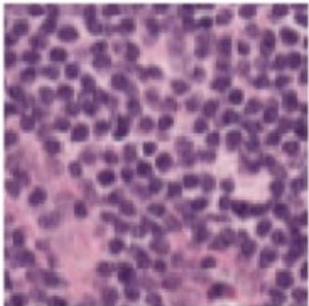
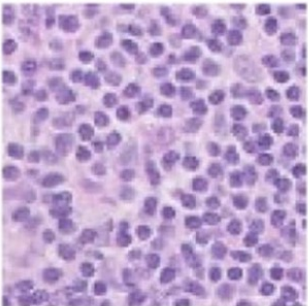
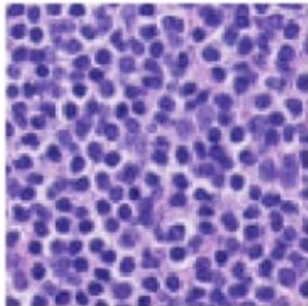
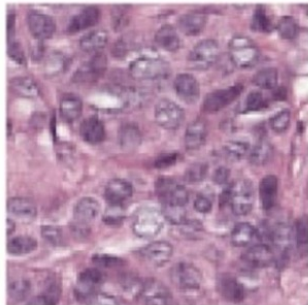
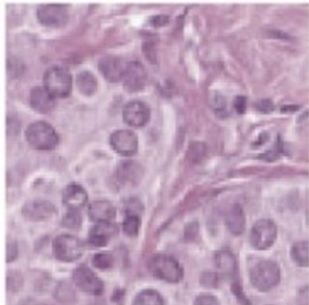
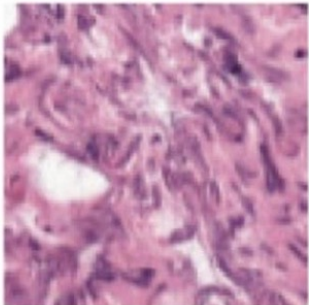
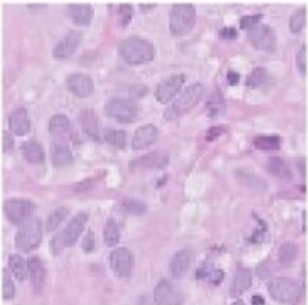
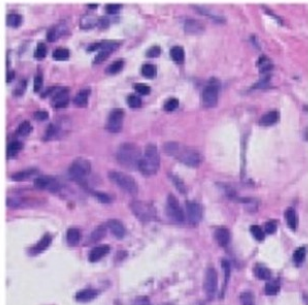
Example: Real Perturbations

Train			Test (OOD)
$d = \text{Location 1}$	$d = \text{Location 2}$	$d = \text{Location 245}$	$d = \text{Location 246}$
			
Vulturine Guinea fowl	African Bush Elephant	...	Wild Horse
			
Cow	Cow	Southern Pig-Tailed Macaque	Great Curassow
Test (ID)			
$d = \text{Location 1}$	$d = \text{Location 2}$	$d = \text{Location 245}$	
			
Giraffe	Impala	Sun Bear	

Example: Real Perturbations

	Train			Test	
Satellite Image (x)					
Year / Region (d)	2002 / Americas	2009 / Africa	2012 / Europe	2016 / Americas	2017 / Africa
Building / Land Type (y)	shopping mall	multi-unit residential	road bridge	recreational facility	educational institution

Example: Real Perturbations

Train			Val (OOD)	Test (OOD)	
	d = Hospital 1	d = Hospital 2	d = Hospital 3	d = Hospital 4	d = Hospital 5
y = Normal					
y = Tumor					

Potential Solutions

- **No general strategy (yet)**
- **Good practices**
 - Train on as large & diverse of a dataset as possible
 - Use data augmentation when possible
 - If available, finetune on location-specific dataset (transfer learning)

Agenda

- **Interpretability & Explainability**
- **Robustness to distribution shift**
- **Robustness to adversarial attacks**

Robustness to Adversarial Attacks

- **Example:**

- Want to reject email attachment if it contains malicious code
- Use machine learning to predict if code is malicious

- **What can go wrong?**

- Attacker perturbs code (e.g., add random lines of dead code) until it is labeled benign by the machine learning model!
- **Strong form of robustness is needed**

Example: Function Name Prediction

- **Task:** Given a function (e.g., as a string), predict its name
- **Attack:** Add a random line of irrelevant code

Example: Function Name Prediction

```
void f1(int[] array){  
    boolean swapped = true;  
    for (int i = 0;  
        i < array.length && swapped; i++){  
        swapped = false;  
        for (int j = 0;  
            j < array.length-1-i; j++) {  
            if (array[j] > array[j+1]) {  
                int temp = array[j];  
                array[j] = array[j+1];  
                array[j+1] = temp;  
                swapped = true;  
            }  
        }  
    }  
}
```

Prediction: **sort** (98.54%)

```
void f3(int[] array){  
    boolean swapped = true;  
    for (int i = 0;  
        i < array.length && swapped; i++){  
        swapped = false;  
        for (int j = 0;  
            j < array.length-1-i; j++) {  
            if (array[j] > array[j+1]) {  
                int temp = array[j];  
                array[j] = array[j+1];  
                array[j+1] = temp;  
                swapped = true;  
            }  
        }  
    } int upperhexdigits;  
}
```

Prediction: **escape** (100%)

Robustness to Adversarial Perturbations

- **Task:**

- Photo ID verification
- Goal is to check whether uploaded photo matches a photo ID

- **Attack:**

- User perturbs their image to match the photo in the ID
- Challenge for machine learning in online identity verification!



(Valid photo ID from Papesh 2018)

Robustness to Adversarial Perturbations

- **Robustness:** Similar images \Rightarrow same label
- **Goal:** Robust to **any** small perturbation in **some family**
 - **Note:** Very far from solving this problem
- **Key question:** What is “some family”?

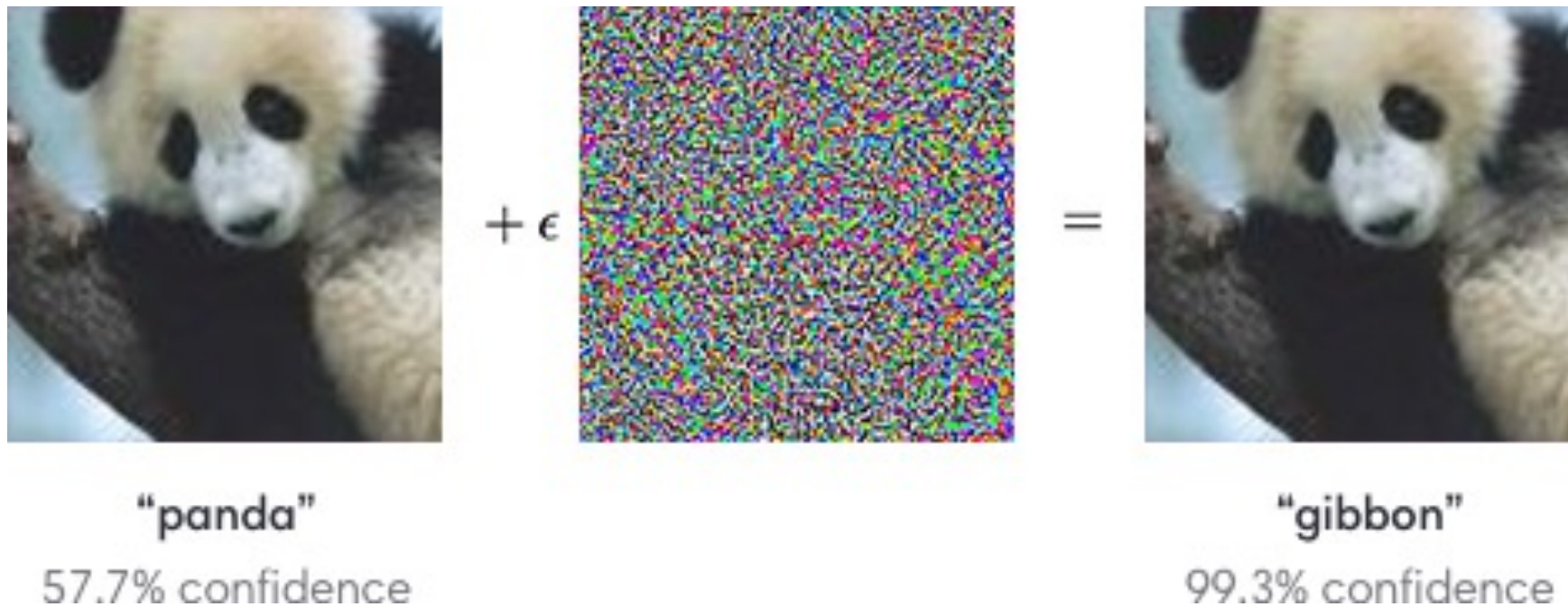
Robustness to Adversarial Perturbations

- (Very limited) example for images:

$$\|x - x'\|_{\infty} \leq \epsilon \Rightarrow \text{same label}$$

- **Question:** Why should the model be robust to these perturbations?
 - Should not change the label
 - Humans are robust!

Robustness to Adversarial Perturbations

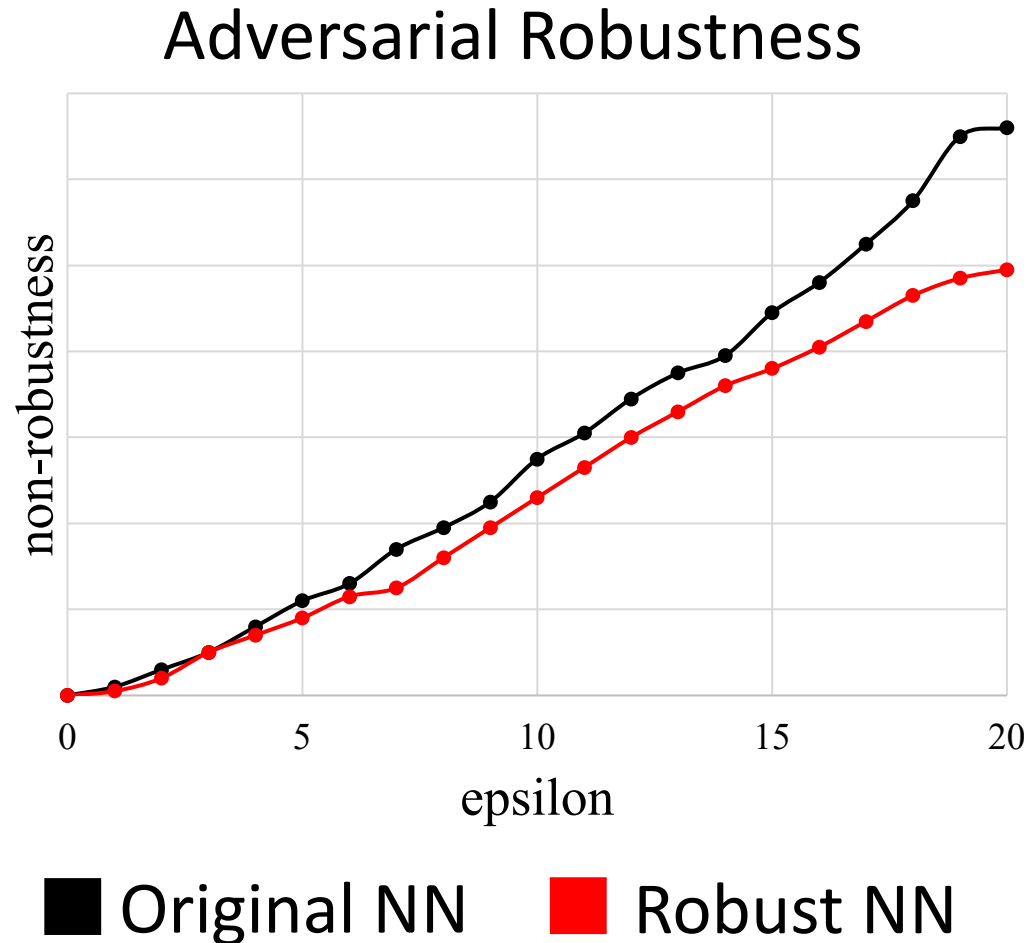


Szegedy et al., Intriguing Properties of Neural Networks, 2014

Robustness to Adversarial Perturbations

- **Strategy for improving adversarial robustness**
 - Data augmentation!
 - **Adversarial training:** Use adversary to generate new examples for training
- Does it work?

Improving Robustness?



Improving Robustness?

- **Problem**

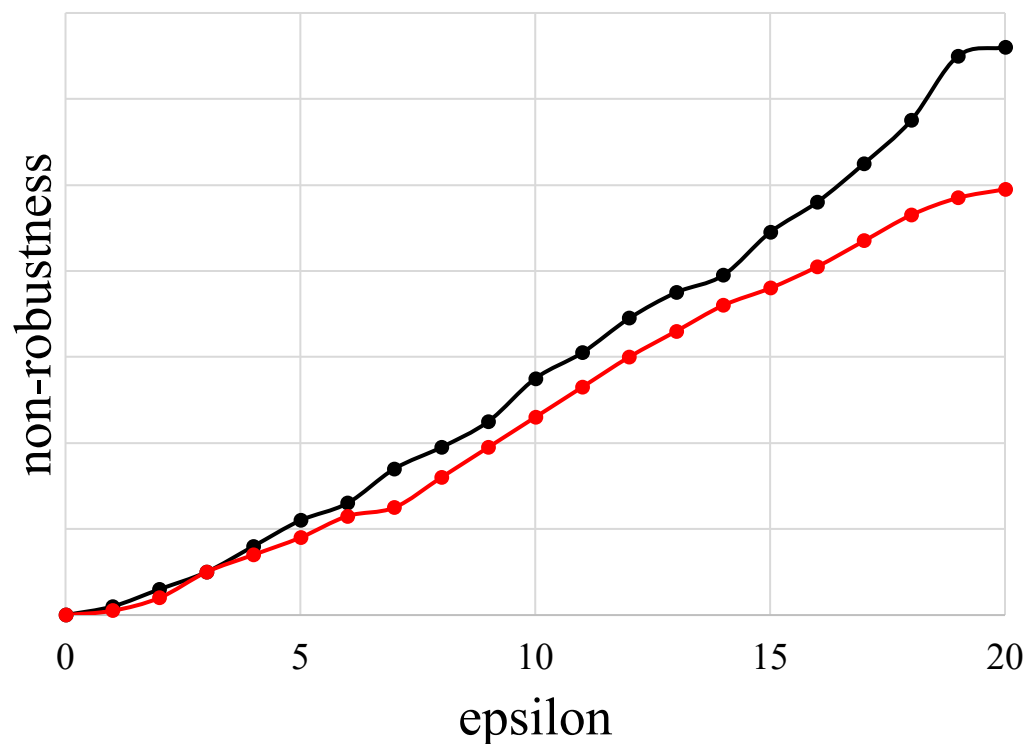
- Only robust to the current adversary
- What if the adversary changes? **Distribution shift!**

- **Example**

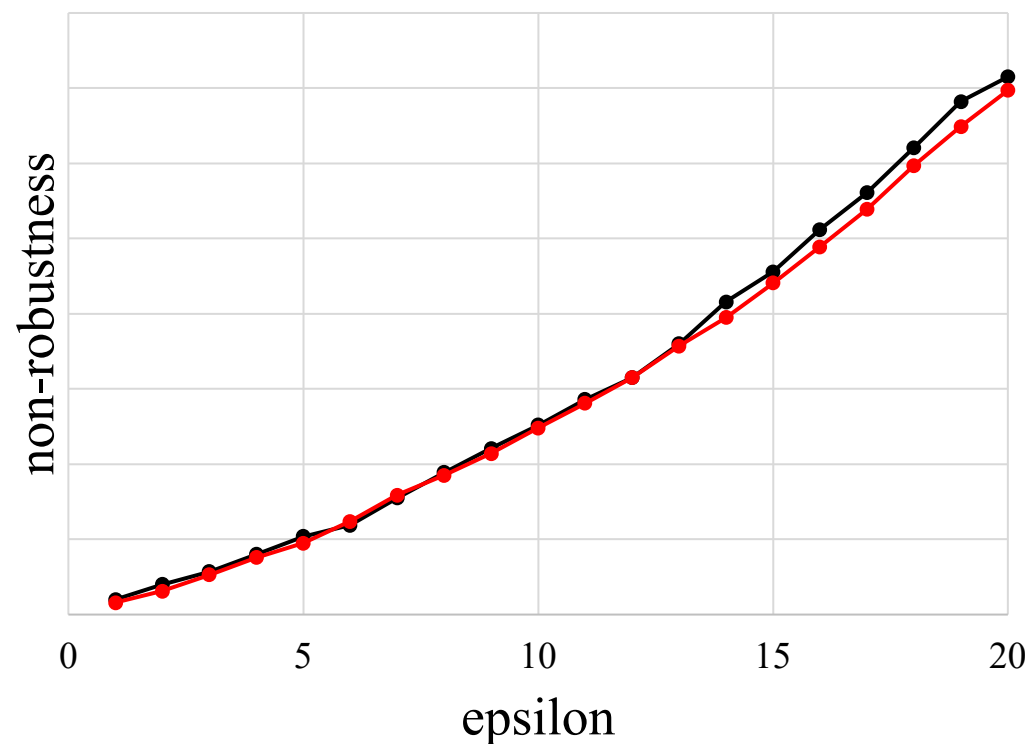
- Adversarial training using one adversary
- Test against a more powerful adversary

Improving Robustness?

Algorithm's Own Metric



Our Metric



■ Original NN ■ Robust NN

Potential Solutions

- **No general strategy (yet)**
- **Good practices**
 - Use the strongest adversary you can design
 - Use variety of different adversaries

Can Uncertainty Help?

- **Recall:** Most neural networks predict an uncertainty

$$p_{\beta}(y | x)$$

- **Idea:** Can we use uncertainty to detect adversarial attacks?
- **Answer: No!**
 - Adversarial examples can have very high confidence
 - Probabilities can be overconfident even for normal test examples!

Potential Solutions

- **General solutions for non-adversarial setting:** Calibrated prediction
- **Intuition:** Among examples where neural network predicts it is correct with probability p , it is correct for a fraction $\approx p$
- **Algorithms:** Temperature scaling, isotonic regression, etc.

Potential Solutions

- **No general solutions for adversarial setting**
- **Good practices**
 - Don't blindly trust predicted probabilities!

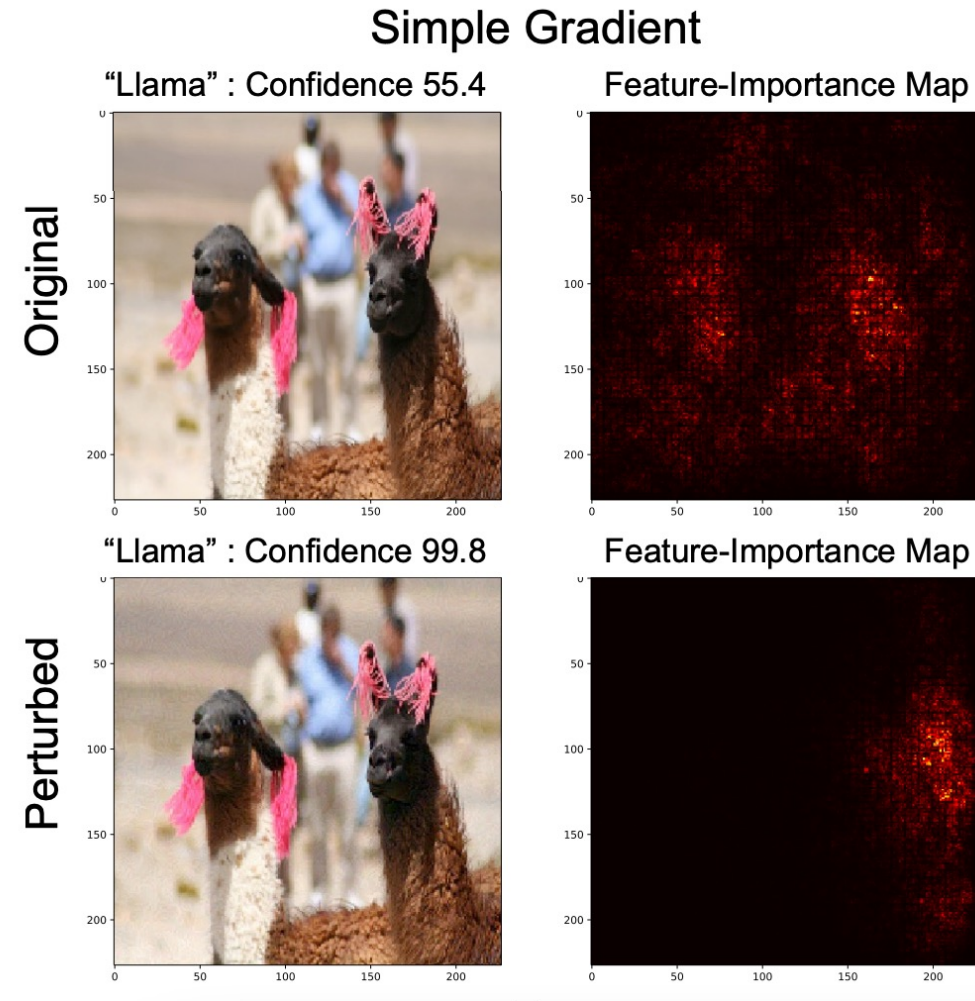
Can Explanations Help?

- **Idea:** Check if explanation makes sense
- **Question:** Are explanations of neural networks robust?

$$\text{Explain}(x + \epsilon) \approx \text{Explain}(x)$$

- **Answer: No!**
- Not even robust to distribution shift

Fragility of Explanations



Fragility of Explanations

- Not just a problem for neural networks!

If **Race** \neq African American:

If **Prior-Felony** = Yes and **Crime-Status** = Active, then **Risky**
If **Prior-Convictions** = 0, then **Not Risky**

If **Race** = African American:

If **Pays-rent** = No and **Gender** = Male, then **Risky**
If **Lives-with-Partner** = No and **College** = No, then **Risky**
If **Age** \geq 35 and **Has-Kids** = Yes, then **Not Risky**
If **Wages** \geq 70K, then **Not Risky**

Default: **Not Risky**

If **Current-Offense** = Felony:

If **Prior-FTA** = Yes and **Prior-Arrests** \geq 1, then **Risky**
If **Crime-Status** = Active and **Owns-House** = No and **Has-Kids** = No, then **Risky**
If **Prior-Convictions** = 0 and **College** = Yes and **Owns-House** = Yes, then **Not Risky**

If **Current-Offense** = Misdemeanor and **Prior-Arrests** $>$ 1:

If **Prior-Jail-Incarcerations** = Yes, then **Risky**
If **Has-Kids** = Yes and **Married** = Yes and **Owns-House** = Yes, then **Not Risky**
If **Lives-with-Partner** = Yes and **College** = Yes and **Pays-Rent** = Yes, then **Not Risky**

If **Current-Offense** = Misdemeanor and **Prior-Arrests** \leq 1:

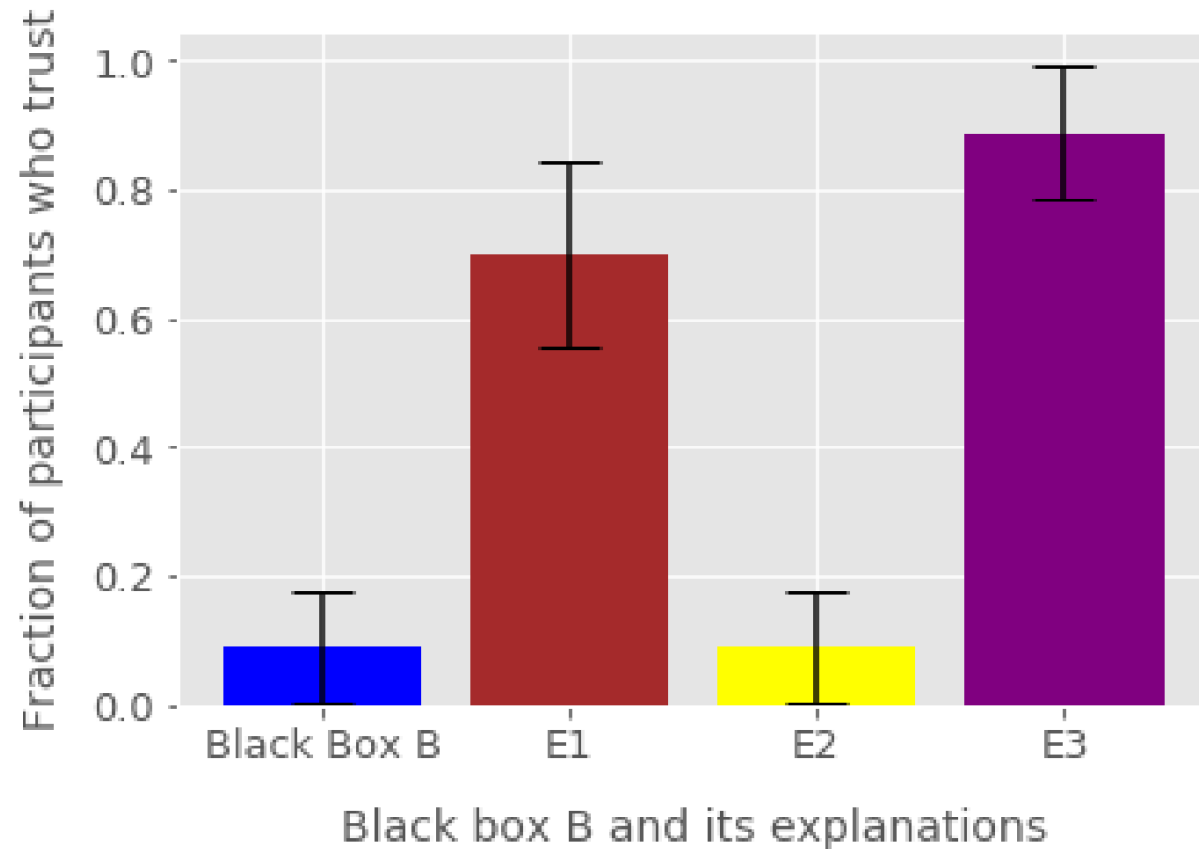
If **Has-Kids** = No and **Owns-House** = No and **Prior-Jail-Incarcerations** = Yes, then **Risky**
If **Age** \geq 50 and **Has-Kids** = Yes and **Prior-FTA** = No, then **Not Risky**

Default: **Not Risky**

Misleading Explanations

- Can construct explanations to mislead users into trusting a model
- **Strategy**
 - Design a set of features that users believe are trustworthy
 - Generate an explanation that highlights these features as important
- Users believe the model is using trustworthy features even if it is not

Misleading Explanations



E1 & E3 are misleading explanations

Potential Solutions

- **No general strategy (yet)**
- **Good practices**
 - Be careful when interpreting explanations!

Conclusion

- Robustness and interpretability remain key challenges for neural networks (and machine learning more broadly)
- **Good practices**
 - Use variety of techniques to try and understand what models are doing (interpretation, extensive testing on different examples, etc.)
 - Be careful when training models!
 - **Monitor performance of models running in production**
- Lots of ongoing research!