

Announcements

- HW 4 due Wednesday
- Project Milestone due April 23

Lecture 23: Recommender Systems

CIS 4190/5190

Spring 2025

Recommender Systems

- **Media recommendations:** Netflix, Youtube, etc.
- **News feed:** Google News, Facebook, Twitter, Reddit, etc.
- **Search ads:** Google, Bing, etc.
- **Products:** Amazon, ebay, Walmart, etc.
- **Dating:** okcupid, eharmony, coffee-meets-bagel, etc.

Recommender Systems

- **Account for:**
 - 75% of movies watched on Netflix [1]
 - 60% of YouTube video clicks [2]
 - 35% of Amazon sales [3]

[1] McKinsey & Company (Oct 2013): <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> [Note: non-authoritative source; estimates only]

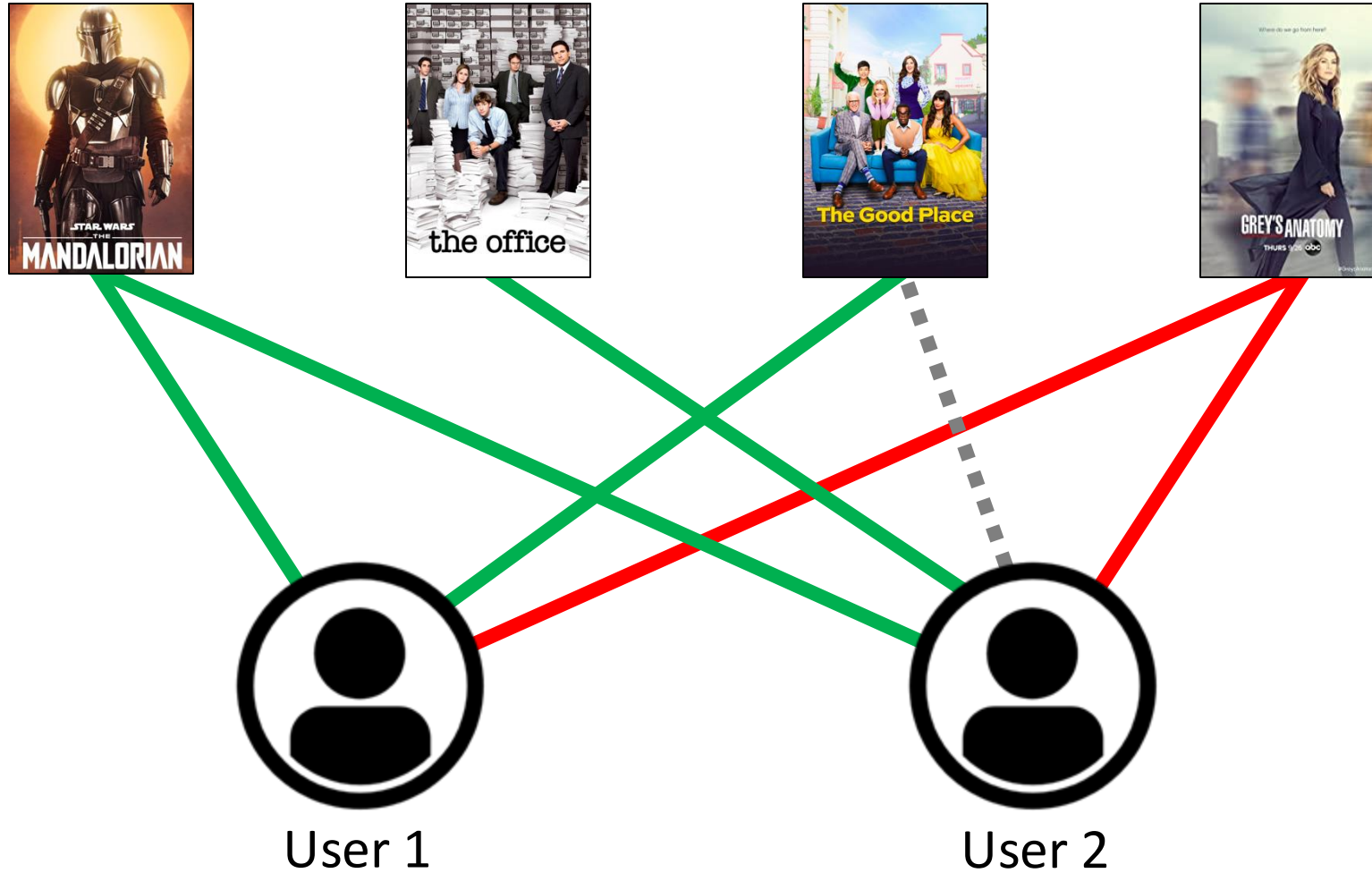
[2] J. Davidson, et al. (2010). The YouTube video recommendation system. Proc. of the 4th ACM Conference on Recommender systems (RecSys). doi.org/10.1145/1864708.1864770

[3] M. Rosenfeld, et al. (2019). Disintermediating your friends: How online dating in the United States displaces other ways of meeting. Proc. National Academy of Sciences 116(36).

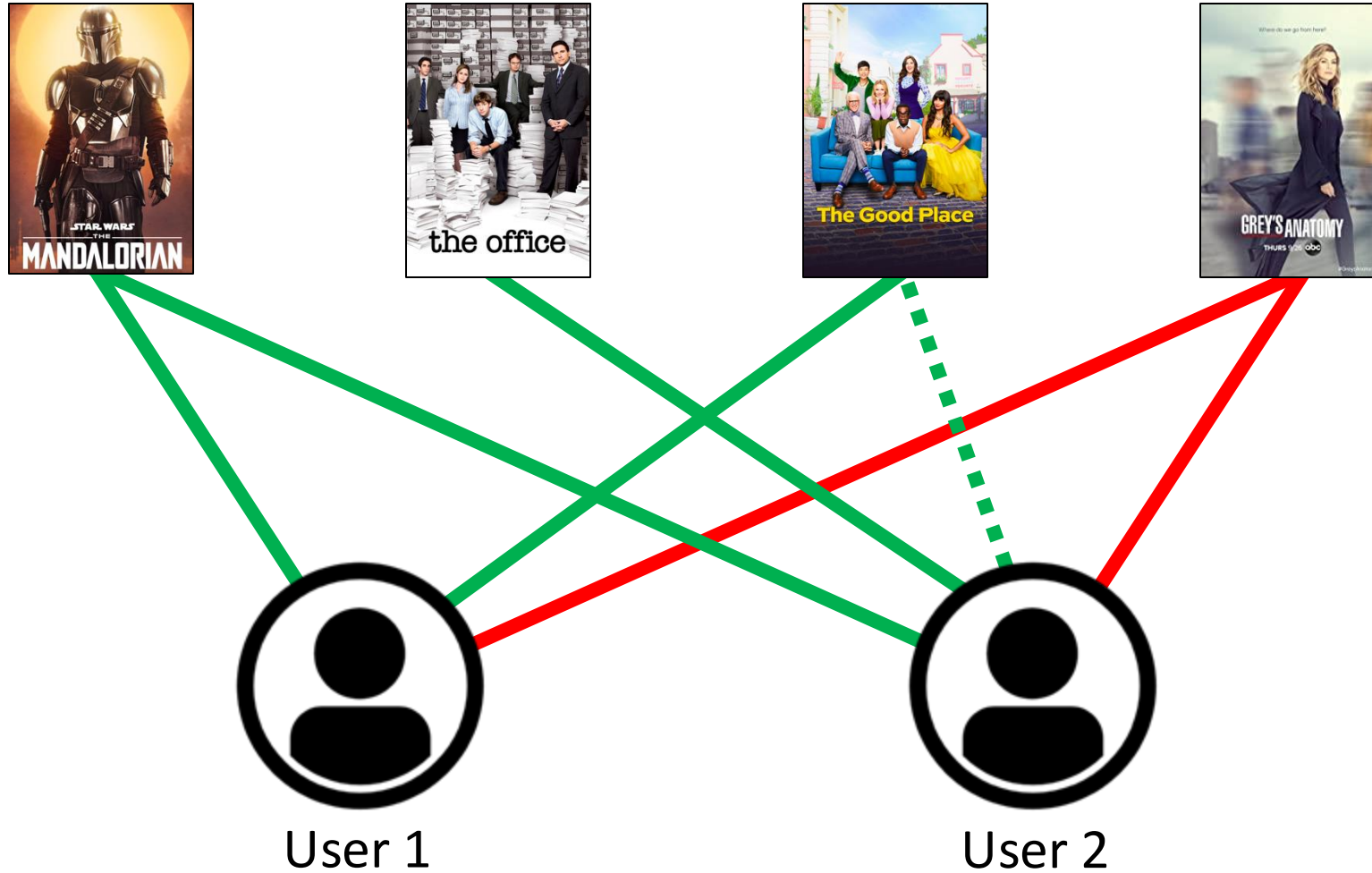
Popularity-Based Recommendation

- Just recommend whatever is currently popular
- Simple and effective, always try as a baseline
- Can be combined with more sophisticated techniques

Collaborative Filtering



Collaborative Filtering



Collaborative Filtering

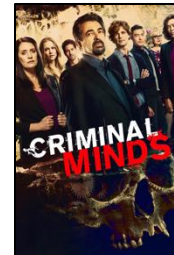
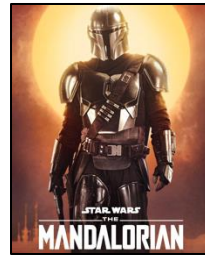
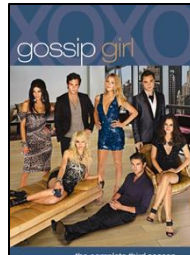
- **Given:**








- Matrix $X_{i,k} = \begin{cases} \text{rating}_{i,k} & \text{if user } i \text{ rated product } k \\ \text{N/A} & \text{otherwise} \end{cases}$
- Assume fixed set of n users and m products
- **Not given any information about the products!**

- **Problem:** Predict what $X_{i,k}$ would be if it is observed
 - Not quite supervised or unsupervised learning!

Collaborative Filtering


Missing entries!








	Gossip Girl	The Office	The Mandalorian	Criminal Minds	The Good Place	Grey's Anatomy	...
 Grace	4	5	4	1	5	3	...
 Eric	1	4	5	1	5	3	...
 Haren	5	5	5	1	3	4	...
 Sai	1	2	5	4	3	5	...
 Siyan	3	1	1	3	4	5	...
 Nikhil	2	3	4	2	2	2	...
 Felix	1	1	1	5	2	2	...

Collaborative Filtering

Missing entries!



	Gossip Girl	The Office	The Mandalorian	Criminal Minds	The Good Place	Grey's Anatomy	...
 Grace		5		1	5		...
 Eric		4	5		5	3	...
 Haren	5		5		3	4	...
 Sai		2					...
 Siyan	3	1		3		5	...
 Nikhil				2	2		...
 Felix	1		1		2		...

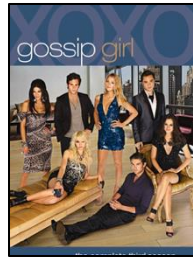
General Strategy








- **Step 1:** Construct user-item ratings
- **Step 2:** Identify similar users
- **Step 3:** Predict unknown ratings

Step 1: Constructing User-Item Ratings

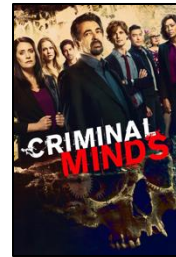
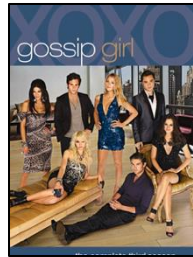
- Can use explicit ratings (e.g., Netflix)
- Can be implicitly inferred from user activity
 - User stops watching after 15 minutes
 - User repeatedly clicks on a video
- Feedback can vary in strength
 - **Weak:** User views a video
 - **Strong:** User writes a positive comment








Step 2: Identifying Similar Users



	Gossip Girl	The Office	The Mandalorian	Criminal Minds	The Good Place	Grey's Anatomy	...
 Grace		5		1	5		...
 Eric		4	5		5	3	...
 Haren	5		5		3	4	...
 Sai		2					...
 Siyan	3	1		3		5	...
 Nikhil				2	2		...
 Felix	1		1		2		...

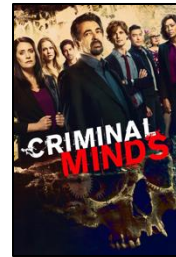
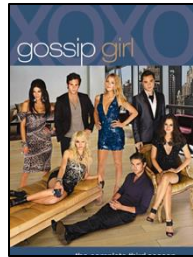
Step 2: Identifying Similar Users










	Gossip Girl	The Office	The Mandalorian	Criminal Minds	The Good Place	Grey's Anatomy	...
 Grace		5		1	5		...
 Eric		4	5		5	3	...
 Haren	5		5		3	4	...
 Sai		2					...
 Siyan	3	1		3		5	...
 Nikhil				2	2		...
 Felix	1		1		2		...

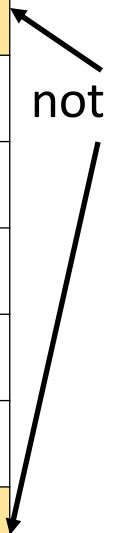
similar

Step 2: Identifying Similar Users



	Gossip Girl	The Office	The Mandalorian	Criminal Minds	The Good Place	Grey's Anatomy	...
 Grace		5		1	5		...
 Eric		4	5		5	3	...
 Haren	5		5		3	4	...
 Sai		2					...
 Siyan	3	1		3		5	...
 Nikhil				2	2		...
 Felix	1		1		2		...

not similar



Step 2: Identifying Similar Users

- **How to measure similarity?**

- Distance $d(X_i, X_j)$, where X_i is vector of ratings for user i

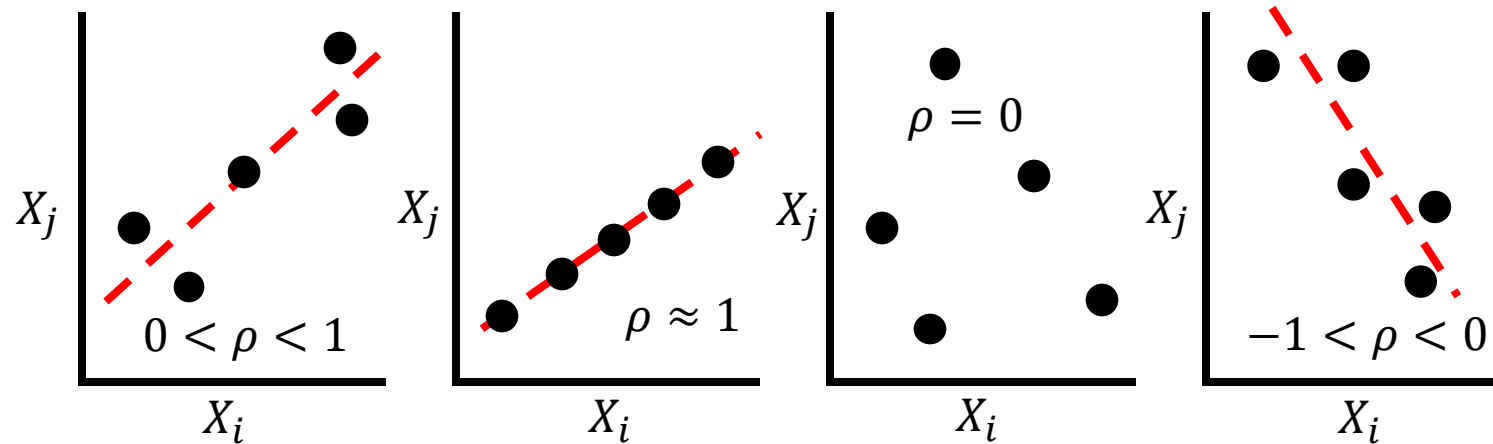
- **Strategy 1:** Euclidean distance $d(X_i, X_j) = \|X_i - X_j\|_2$

- Ignore entries where either X_i or X_j is N/A
- **Shortcoming:** Some users might give higher ratings everywhere!

- Similar issues with other distance metrics such as cosine similarity

Step 2: Identifying Similar Users

- **Strategy 2:** Pearson correlation: $\rho = \frac{\sum_{k=1}^m (X_{i,k} - \bar{X}_i)(X_{j,k} - \bar{X}_j)}{\sqrt{\sum_{k=1}^m (X_{i,k} - \bar{X}_i)^2 \sum_{k=1}^m (X_{j,k} - \bar{X}_j)^2}}$
 - Here, $\bar{X}_i = \frac{1}{m} \sum_{k=1}^m X_{i,k}$
 - Normalization by variance deals with differences in individual rating scales



Step 3: Predict Unknown Ratings

- **Weighted averaging strategy**

- Compute weights $w_{i,j} = g(d(X_i, X_j))$ based on the distances
- Normalize the weights to obtain $\bar{w}_{i,j} = \frac{w_{i,j}}{\sum_{j=1}^n w_{i,j}}$
- For user i rating item k , predict

$$X_{i,k} = \bar{X}_i + \sum_{j=1}^n \bar{w}_{i,j} \cdot (X_{j,k} - \bar{X}_j)$$

Step 3: Predict Unknown Ratings

- **Variations**

- Instead of weights, choose a neighborhood (e.g., threshold based on similarity, top-k based on similarity, or use k-means clustering)
- Instead of subtracting the mean, normalize by standard deviation

Matrix Factorization

- **Model family:** Consider parameterization

$$X_{i,k} \approx U_i^\top V_k$$

- Both $U_i \in \mathbb{R}^d$ and $V_k \in \mathbb{R}^d$ are parameters
- U_i represents “features” for user i
- V_k represents “features” for product k

Matrix Factorization

- **Loss function:**

$$L(U, V; X) = \sum_{i=1}^n \sum_{k=1}^m 1(X_{i,k} \neq \text{N/A}) \cdot (X_{i,k} - U_i^\top V_k)^2$$

- **Optimizer:**

- Can be minimized using gradient descent
- **“Alternating” least squares:** Hold U fixed, then optimizing V is linear regression (and vice versa), so alternate between the two

Collaborative Filtering

- **Pros**

- No domain knowledge needed, only user behavior
- Captures that users may have diverse preferences

- **Cons**

- Suffers when data is sparse
- Does not consider item content, so cannot generalize to new items
- Does not consider user features, so cannot generalize to new users

Content-Based Approaches

- **Step 1:** Manually construct feature vector U_i for item
- **Step 2:** Manually construct feature vector V_k for user
- **Step 3:** Train a model using supervised learning to predict the user's rating for the given item:

$$X_{i,j} \approx f_{\beta}(U_i, V_k)$$

Content-Based Approaches

- **Pros**

- Incorporates external sources of knowledge on items/users to generalize
- More explainable since recommendations are based on handcrafted features

- **Cons**

- Requires domain knowledge and feature engineering
- Narrow recommendations

Hybrid Approaches

- **Combine collaborative filtering with content-based approaches**
 - Ensemble different predictions
 - Concatenate collaborative filtering features with handcrafted features
- **Deep-learning based approaches**
 - Can be used with both approaches (or a combination)
 - Active area of research

Other Considerations

- **Challenges measuring utility**
 - Ratings can be misleading
 - Fake reviews/ratings are commonplace
- **Time-varying preferences**
 - User preferences change, item popularities change
 - Can upweight recent data (e.g., exponentially weighted moving average)
- **Evaluation**
 - **Offline:** Split users into train/test, and evaluate model on test users
 - **Online:** Split users into train/test, and run separate algorithms for each

What About New Users?

- Called the “cold start” problem
- **Feature-based approach**
 - Just featurize the user!
- **Collaborative filtering**
 - Need to collect ratings from the user!
 - Use multi-armed bandits

Lecture 24: Robustness

CIS 4190/5190

Spring 2025

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

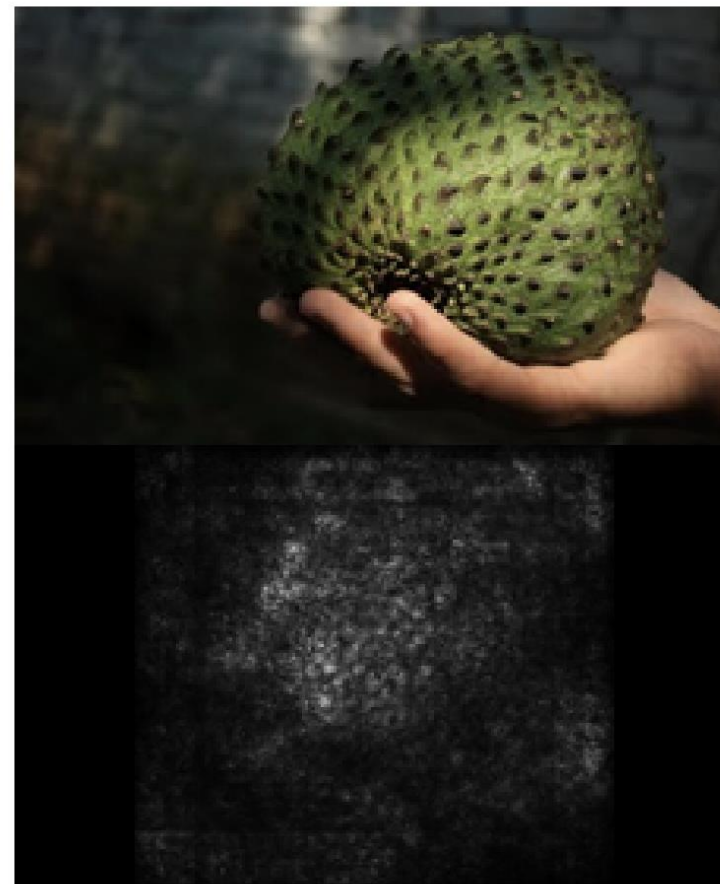
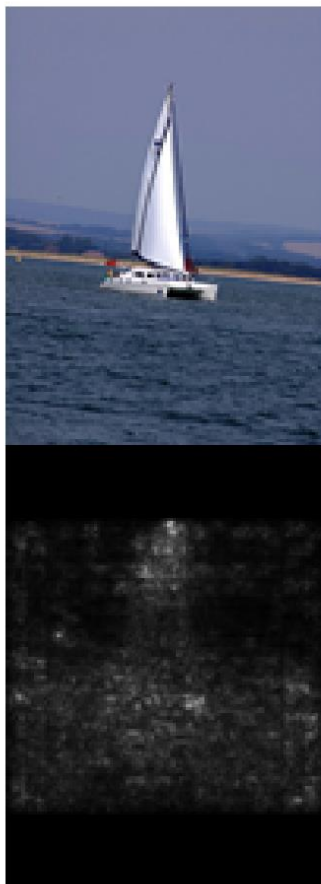
- Consider the gradient of the loss with respect to the input:

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

- **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



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

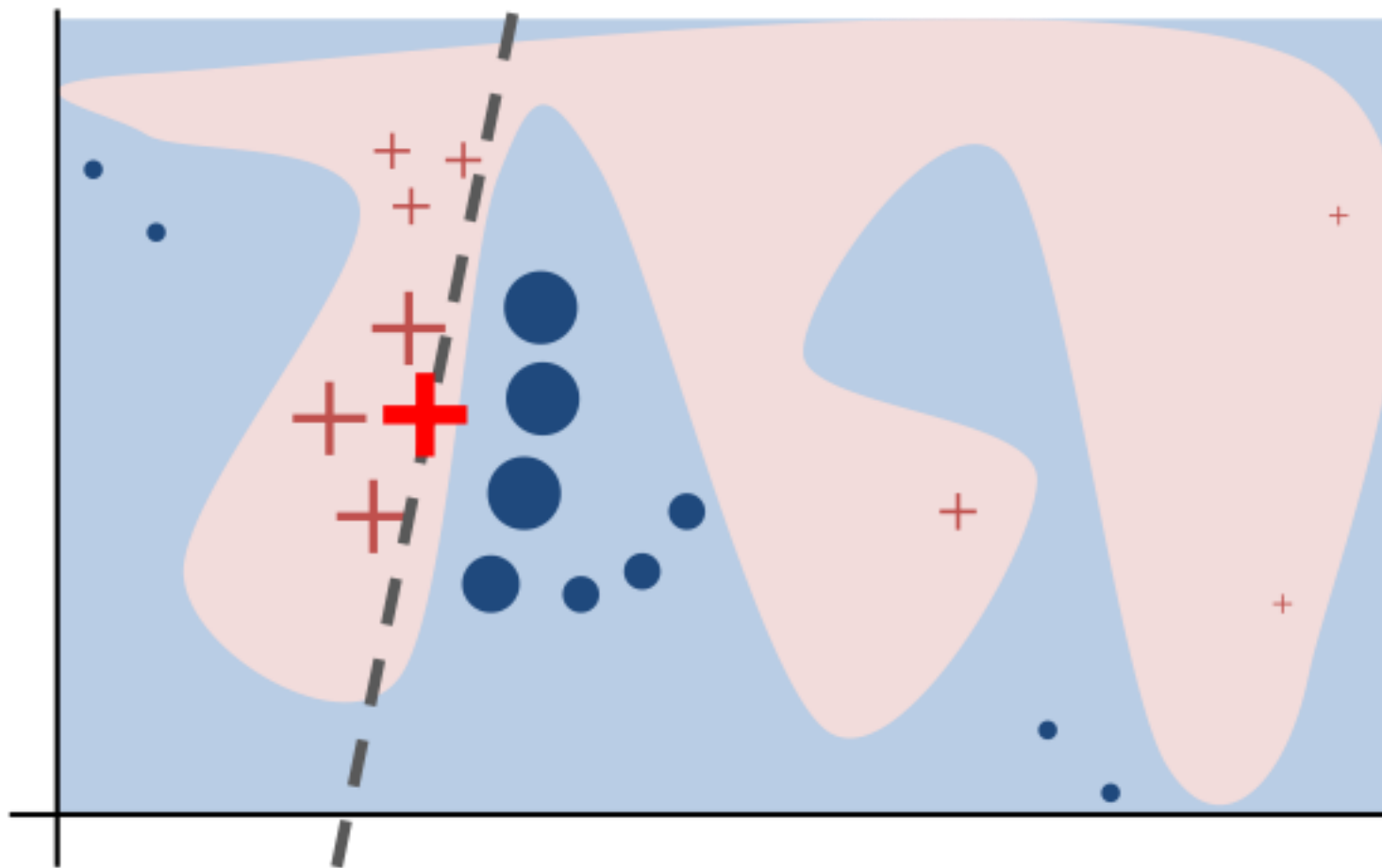
Local Explanations

- Construct dataset

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

- Here, $\epsilon \sim N(0, \sigma^2)$ is i.i.d. Gaussian noise
- Fit a linear model to this dataset 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*

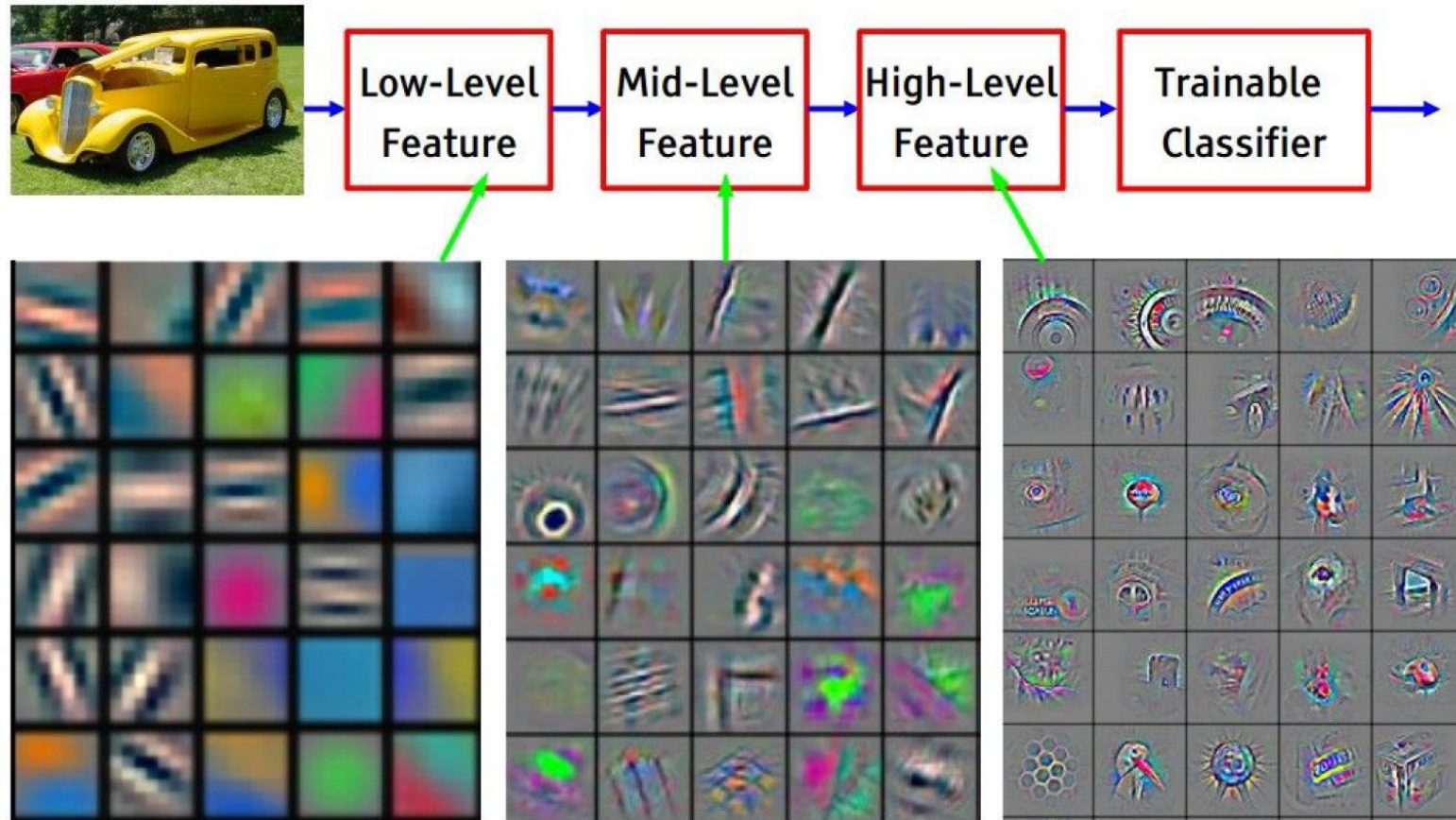
(d) Explaining *Labrador*

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

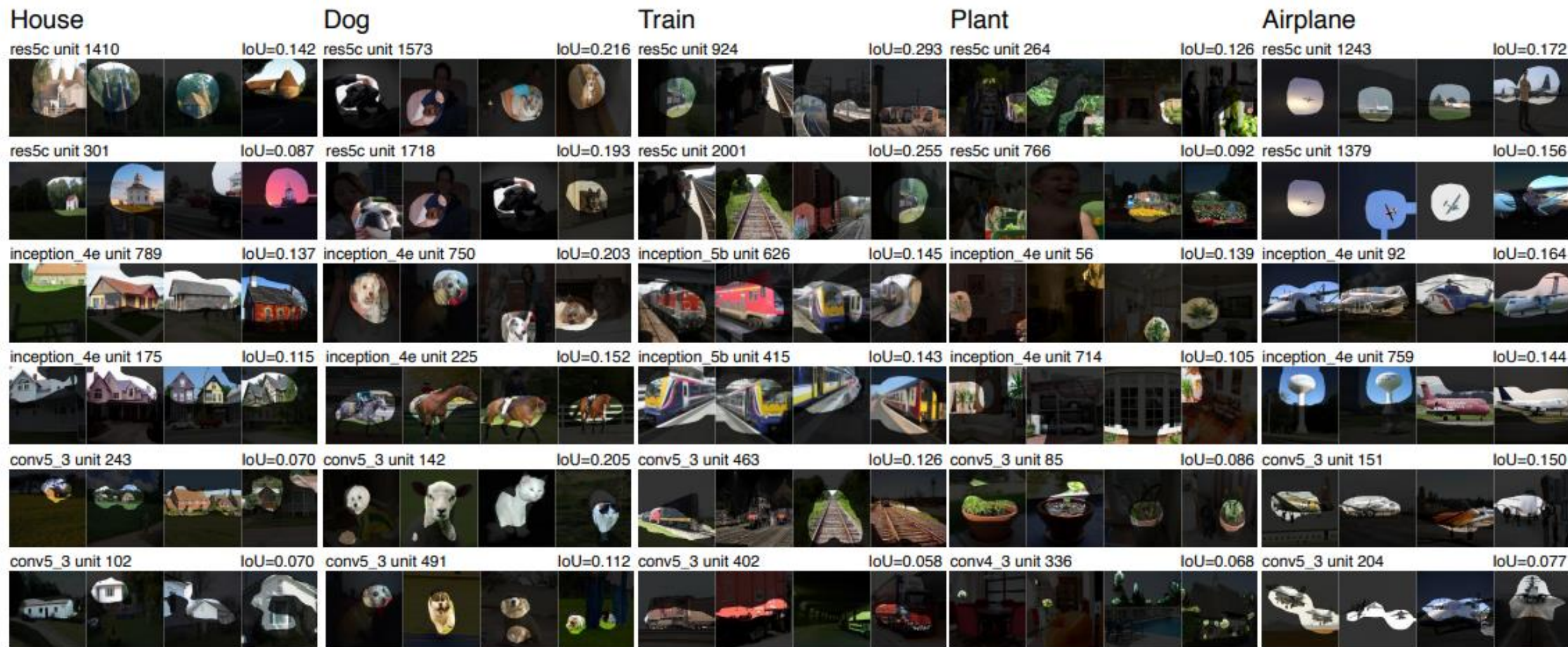
- **Neuron visualization:** Look at $\nabla_x g_\beta(x)$ for an intermediate layer g_β
- **Network dissection:** Look at groups of pixels corresponding to objects

Neuron Visualization



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

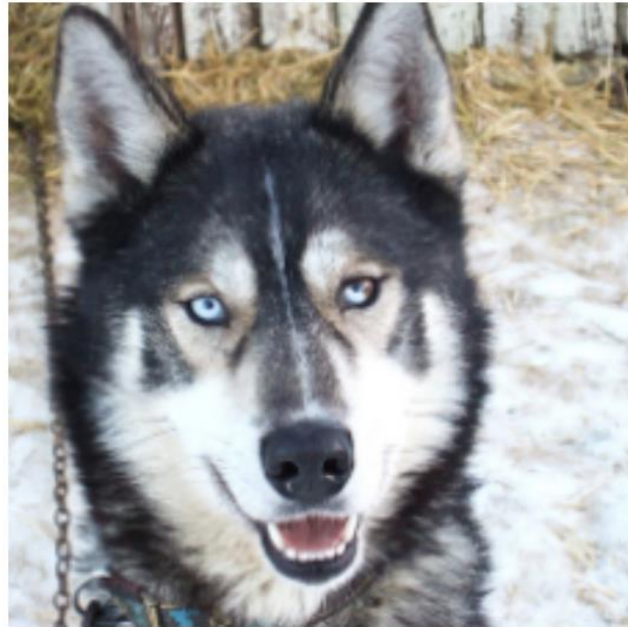
Neural Network Dissection



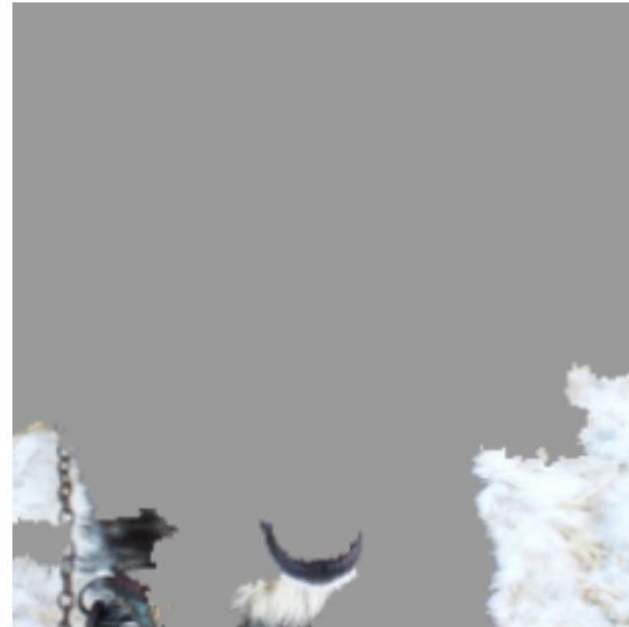
Why Are Explanations Useful?

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

An Interesting Local Explanation



(a) Husky classified as wolf



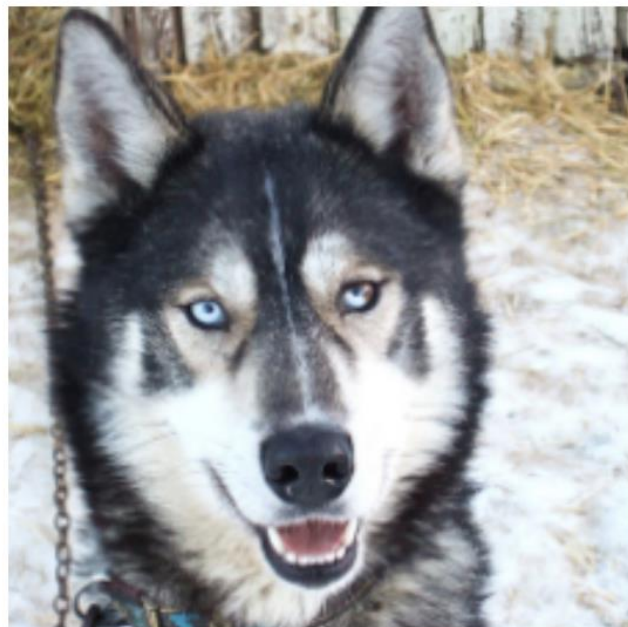
(b) Explanation

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

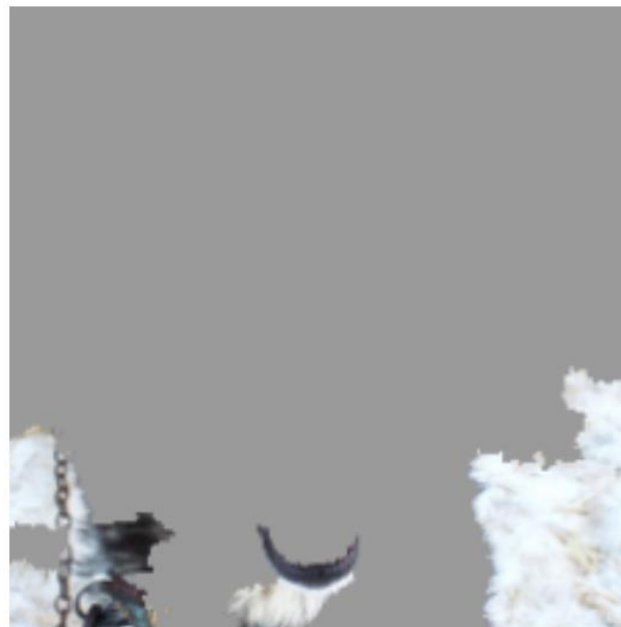
Correlated Inputs/Features

- 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.

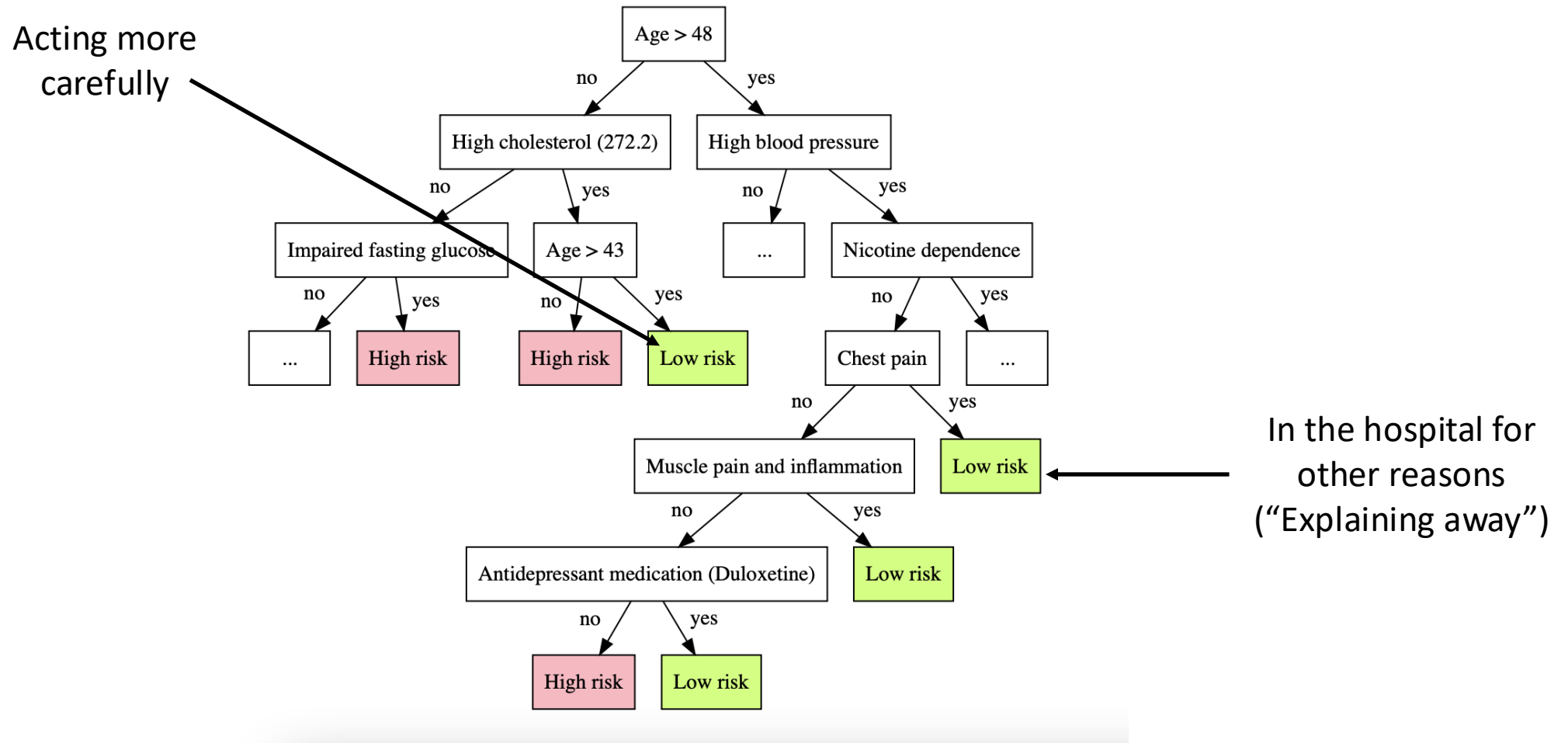
Problematic Correlations

- 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

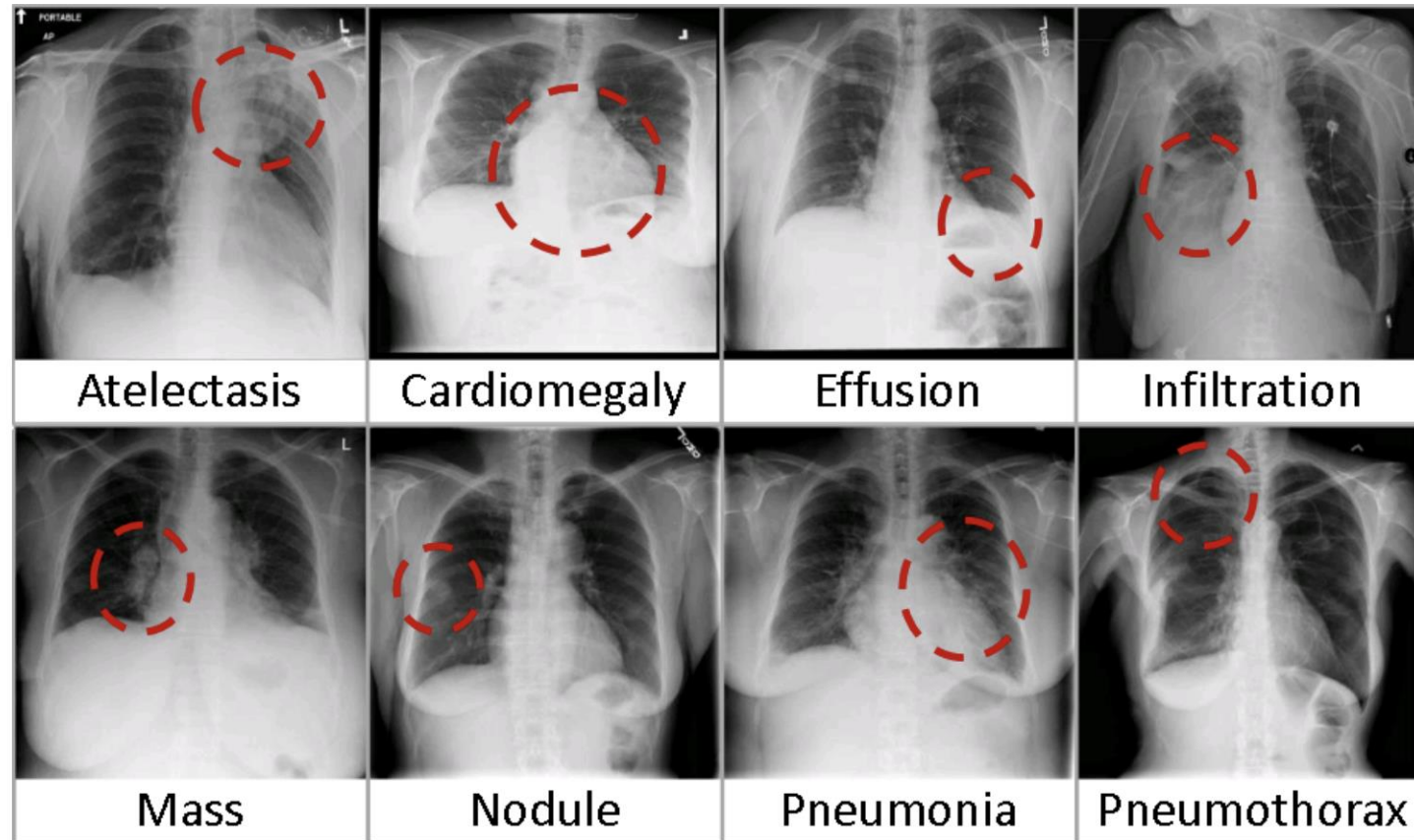


Figure 1. *Eight common thoracic diseases observed in chest X-rays that validate a challenging task of fully-automated diagnosis.*

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**

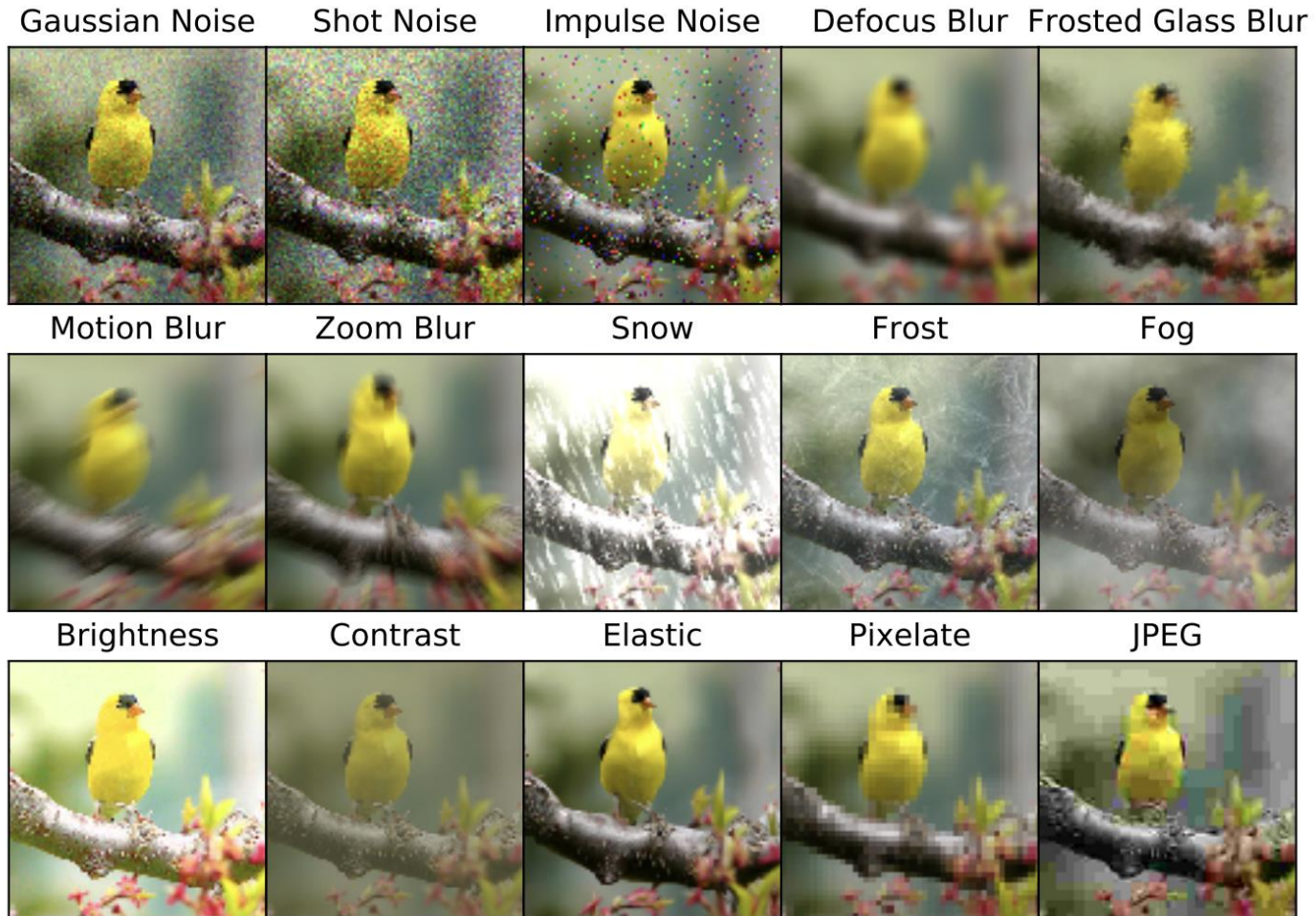
Robustness to Distribution Shift

- 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



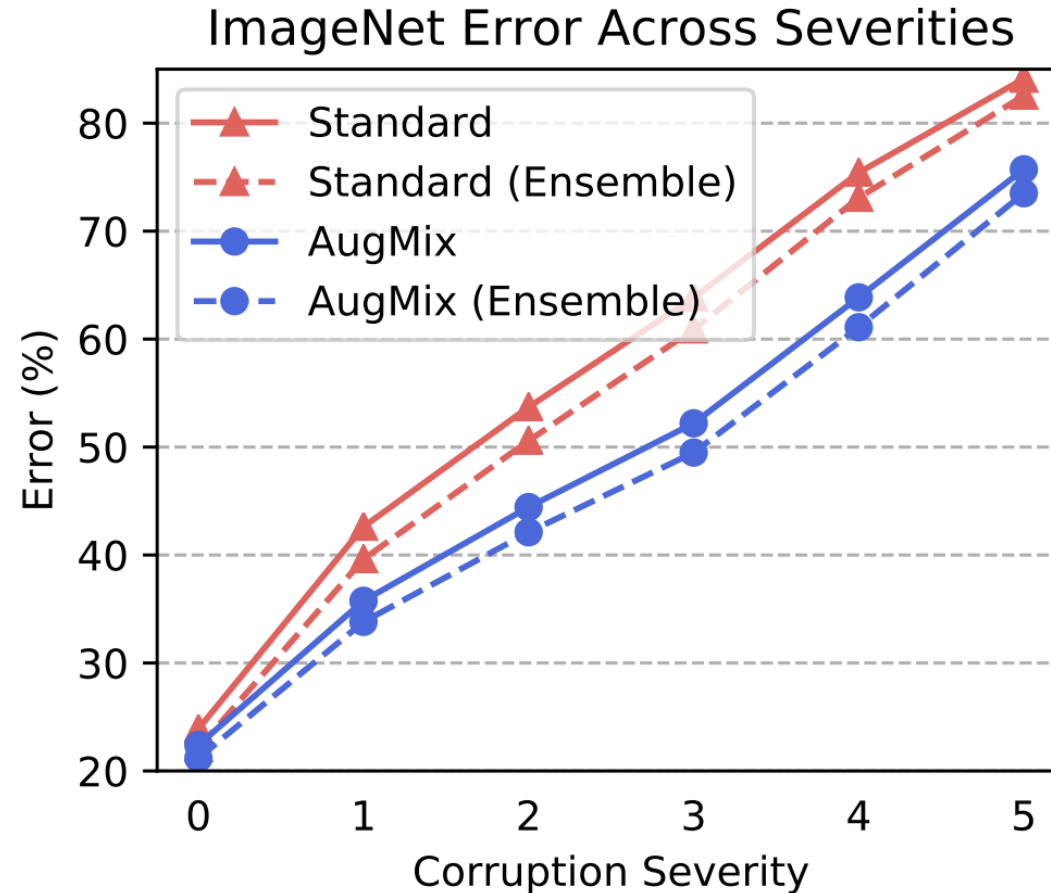
Example: Synthetic Perturbations

- **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

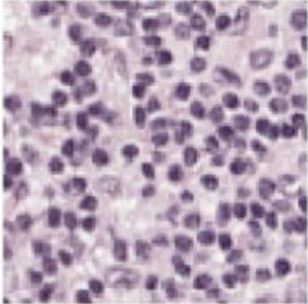
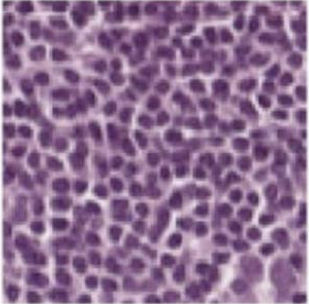
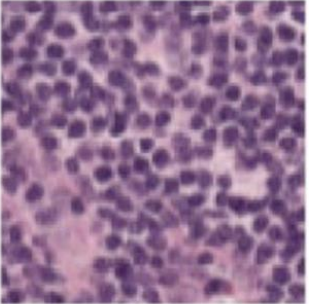
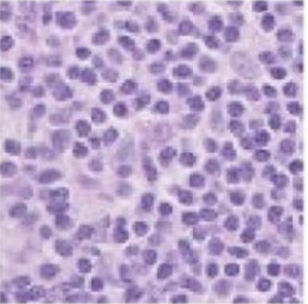
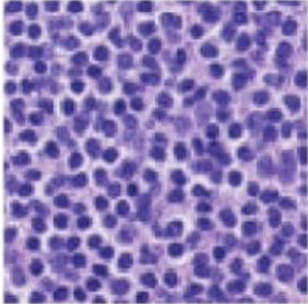
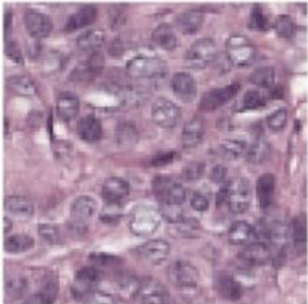
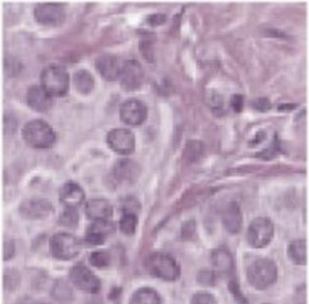
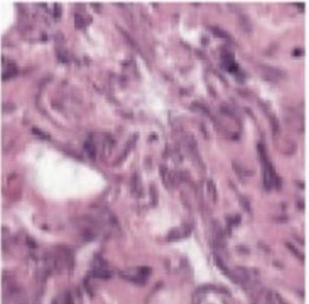
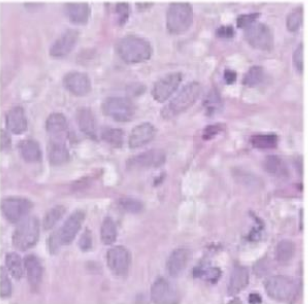
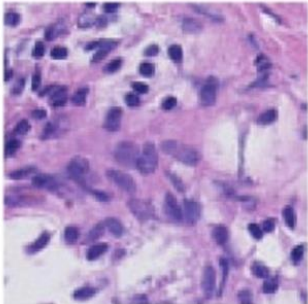
Example: Real Perturbations

Train			Test (OOD)
$d = \text{Location 1}$	$d = \text{Location 2}$	$d = \text{Location 245}$	$d = \text{Location 246}$
			
Vulturine Guineafowl	African Bush Elephant
			
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”?

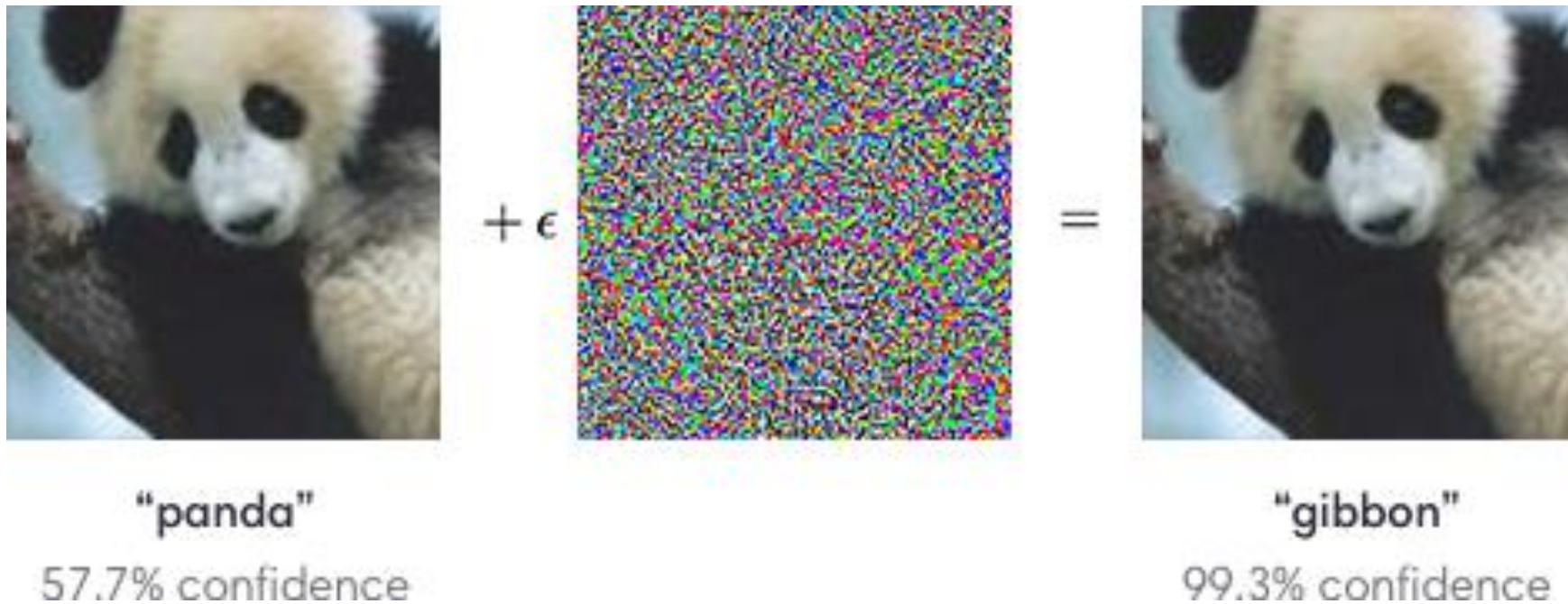
Robustness to Adversarial Perturbations

- (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



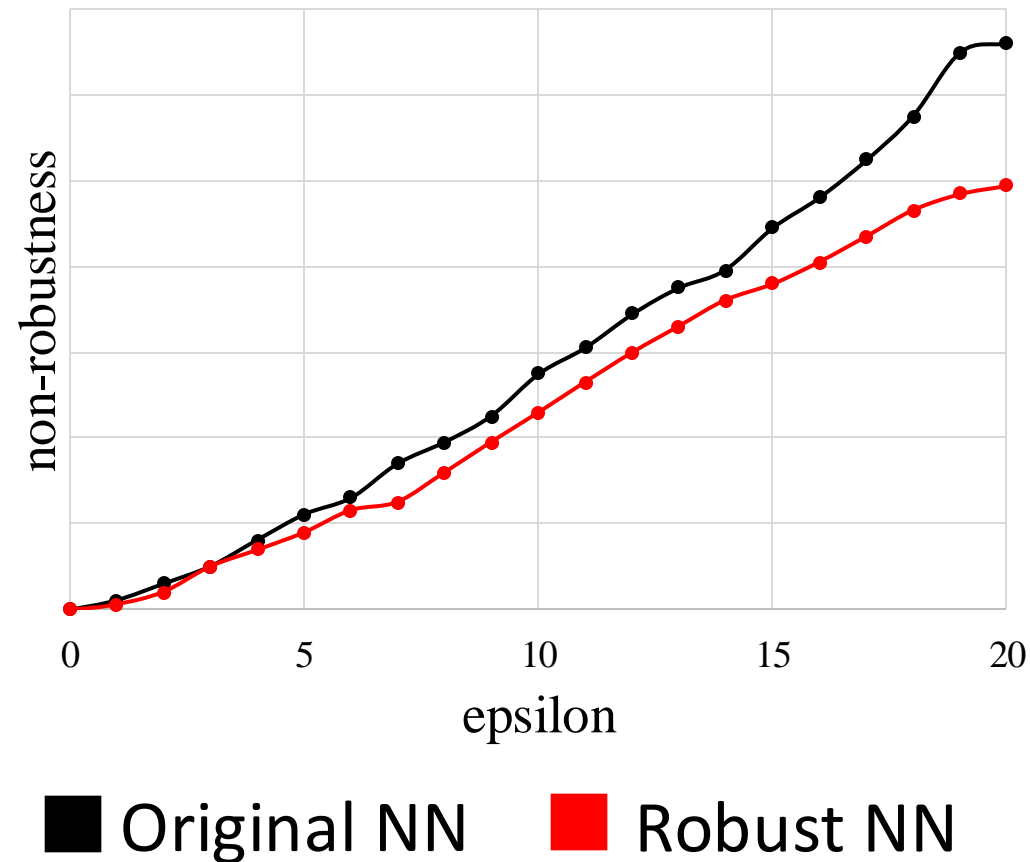
Szegedy et al., Intriguing Properties of Neural Networks, 2014

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?

Adversarial Robustness



Improving Robustness?

- **Problem**

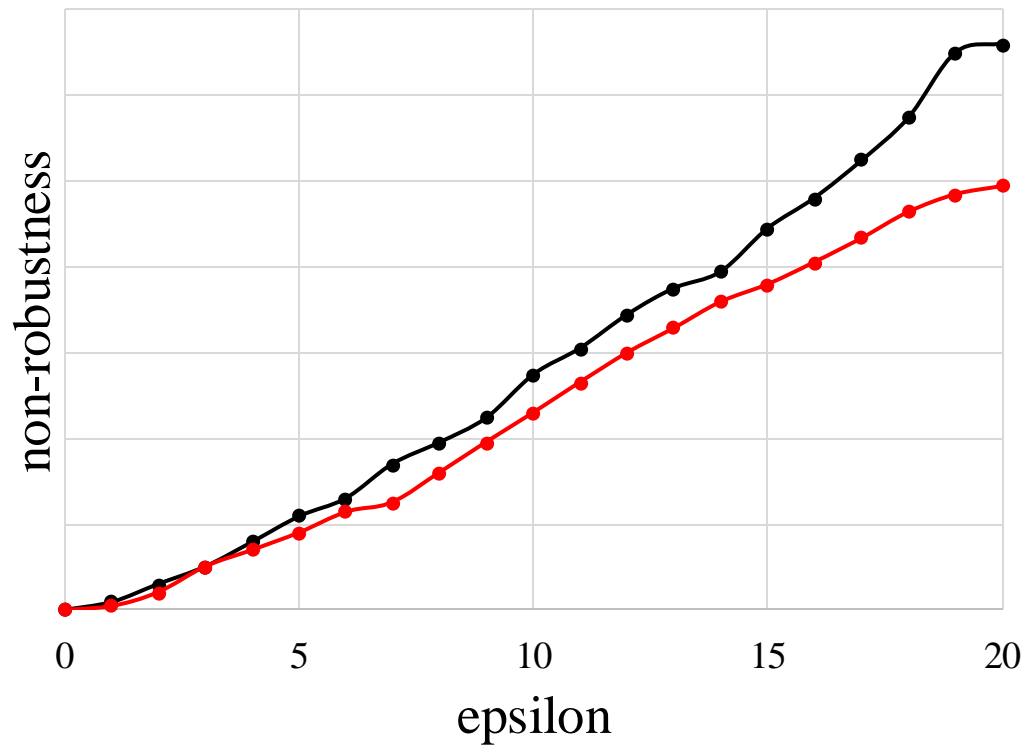
- Only robust to the current adversary
- What if the adversary changes? **Distribution shift!**

- **Example**

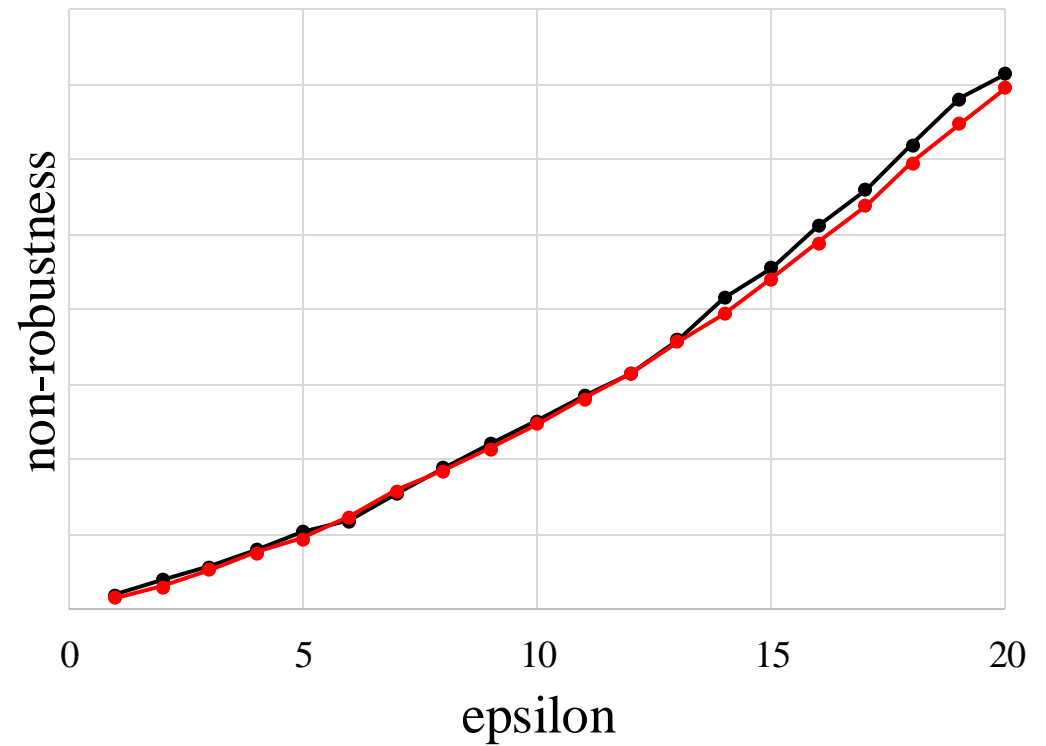
- Adversarial training using one adversary
- Test against a more powerful adversary

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?

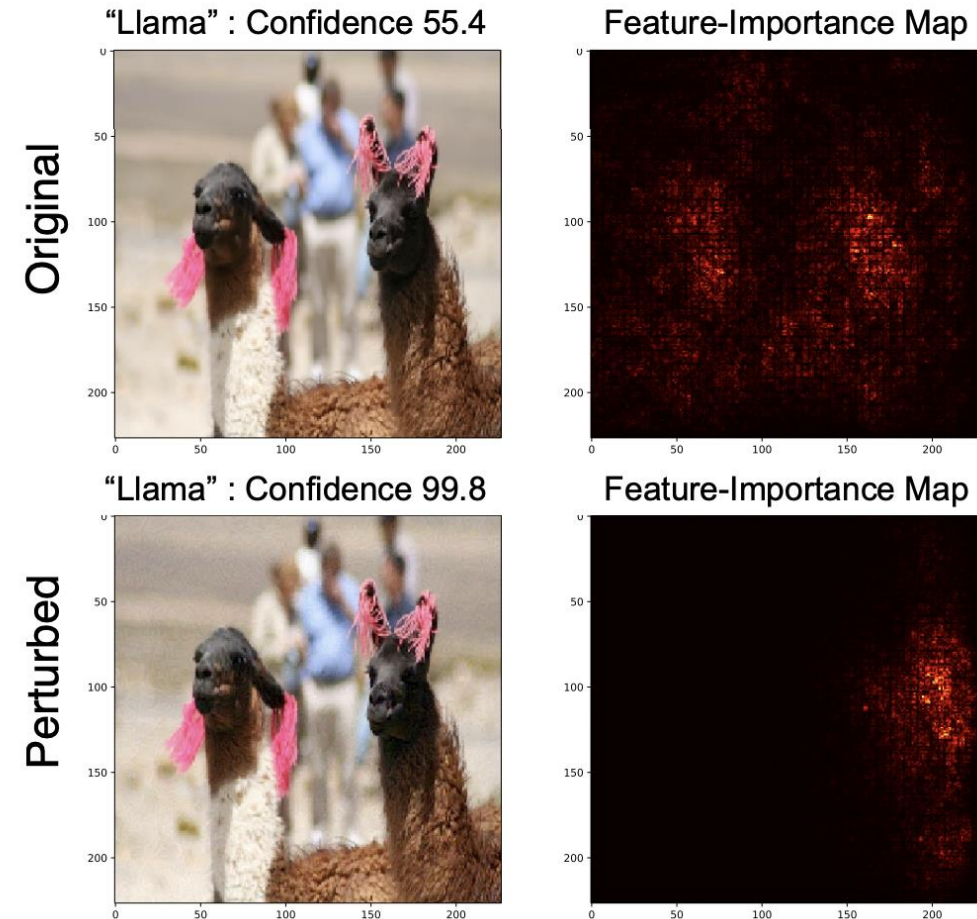
- **Idea:** Check if explanation makes sense
- **Question:** Are explanations of neural networks robust?

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

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

Fragility of Explanations

Simple Gradient



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**

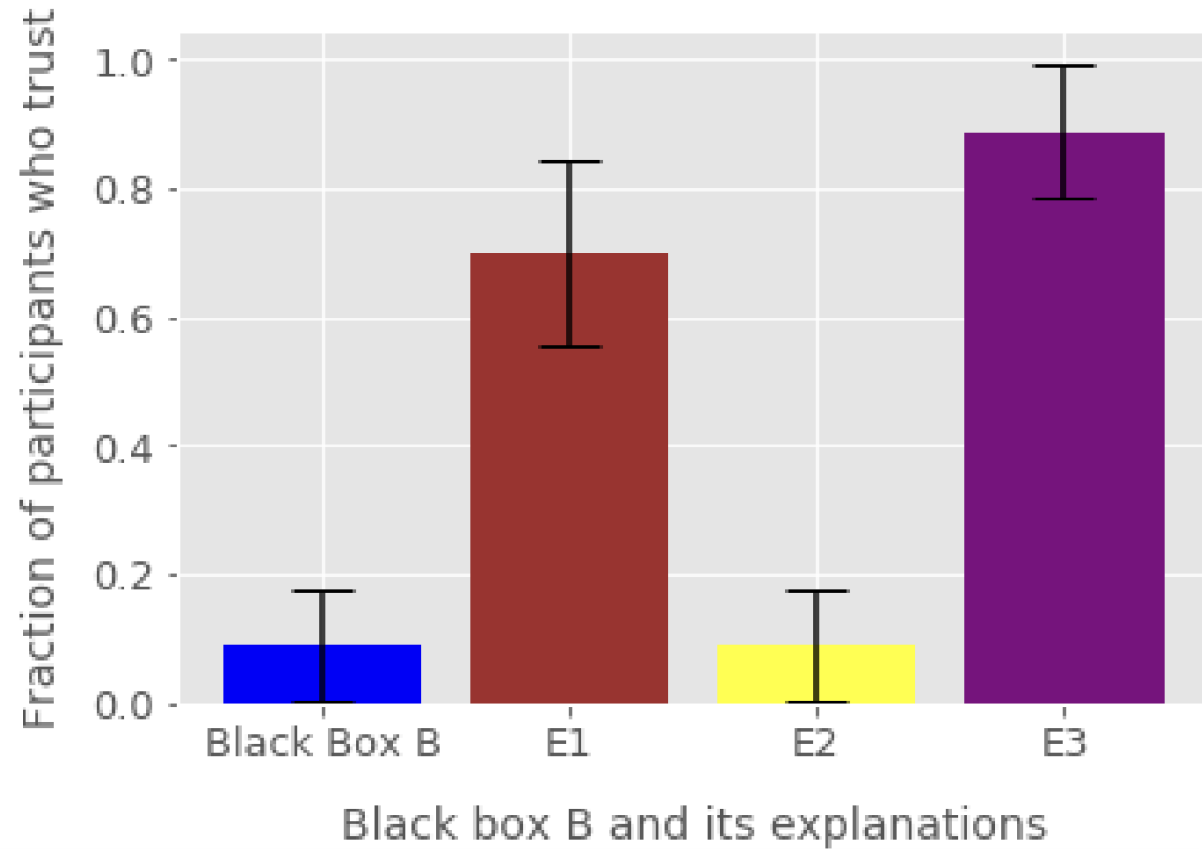
If **Age** \geq 50 and **Has-Kids** = Yes and **Prior-FTA** = No, then **Not Risky**

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!