Yale

Explainability of Neural Networks (XAI)

CPSC680: Trustworthy Deep Learning

Rex Ying

Readings

- Readings are updated on the website (syllabus page)
- **Lecture 4 readings**:
  - https://arxiv.org/abs/1703.01365

  (Integrated Gradients)

Content

- Introduction to Explainability
- Explainability Settings
- Explainable Models
- Gradient-based Methods
- Perturbation Methods
- (Next lecture: Methods using Surrogate Models)

Content

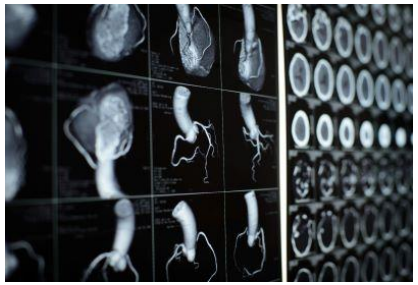
- Introduction to Explainability
- Explainability Settings
- Explainable Models
- Gradient-based Methods
- Perturbation Methods

Explainability

- The **black-box** nature of deep learning makes it a **major challenge** to:
  - Understand what is learned by the ML model
  - Extract insights of the underlying data we are trying to model
- Explainable Artificial Intelligence (XAI) is an umbrella term for any research trying to solve the **black-box problem for AI**
- Why is it useful?
  - Enable users to **understand the decision-making** of the model
  - **Gain trust from human users** of the deep learning system
- Simple-to-read guide: 2004.14545.pdf (arxiv.org)

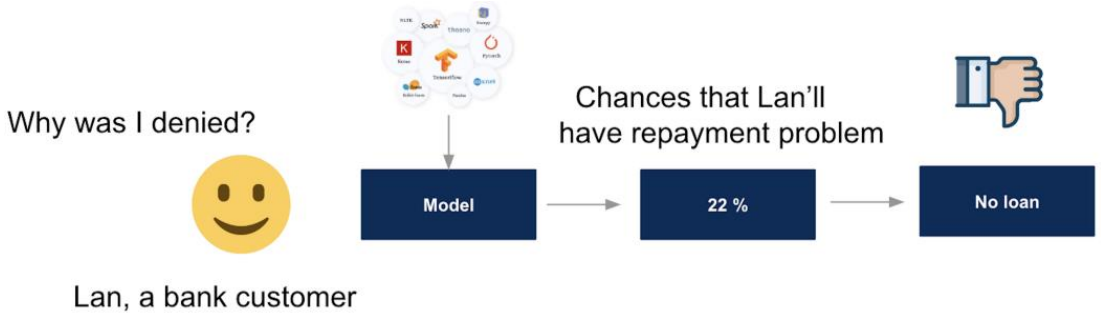
Goal of Explainability

- Model's behavior might be different from the underlying phenomenon
- **Explaining ground truth phenomenon**



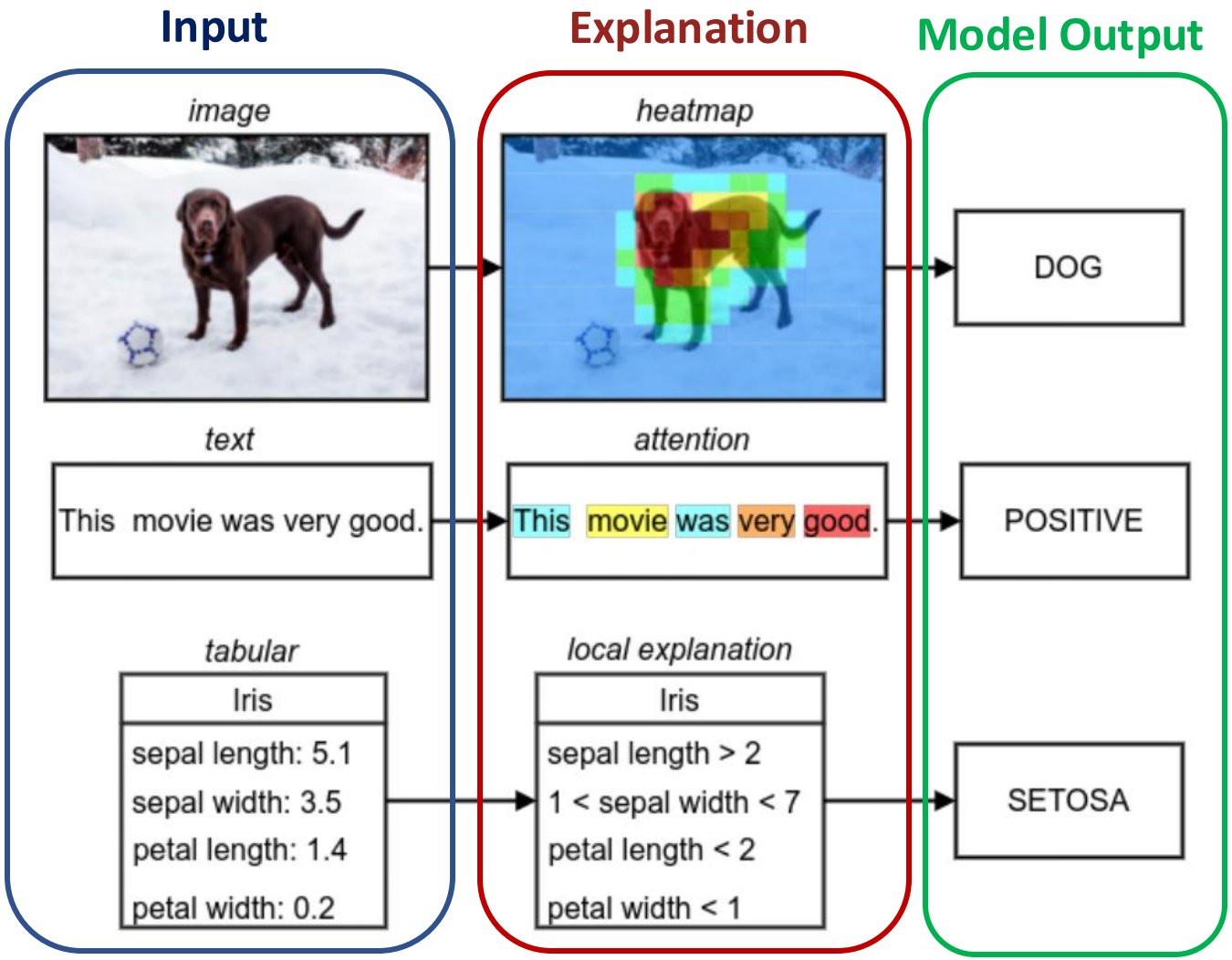
What are the characteristics of certain diseases in terms of imaging?

- **Explaining model predictions**



Why does the model recommend no loan for Person X?

Forms of Explanation



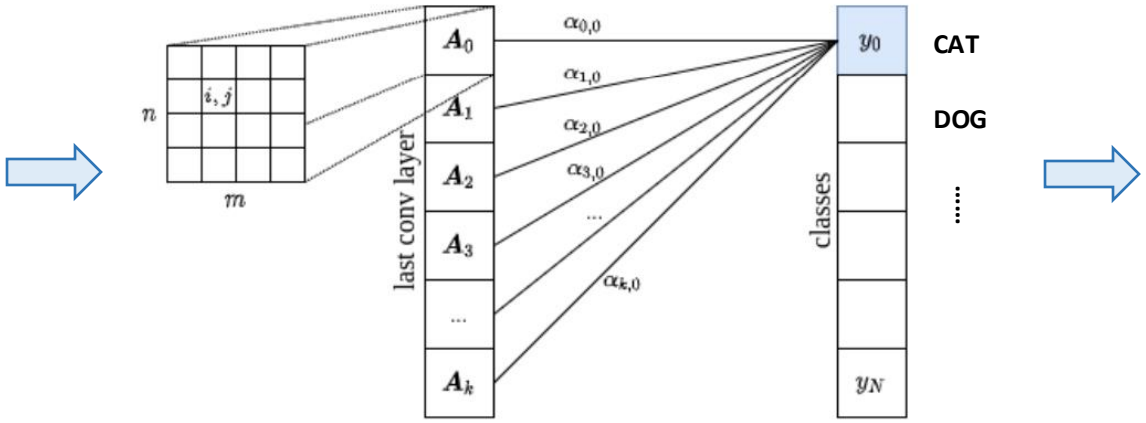
Example: Computer Vision

Explanation in Computer Vision:

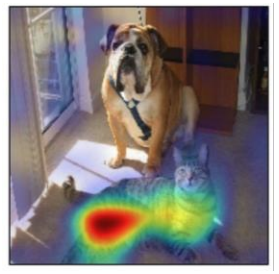
A particular region of the image **displays the predicted class of objects** (cat / dog in this example)

Importance scores on pixels

original graph

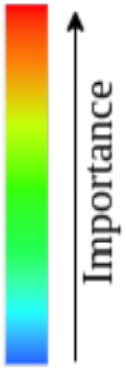


computation process of **CNN** and the prediction



explanation of "cat"

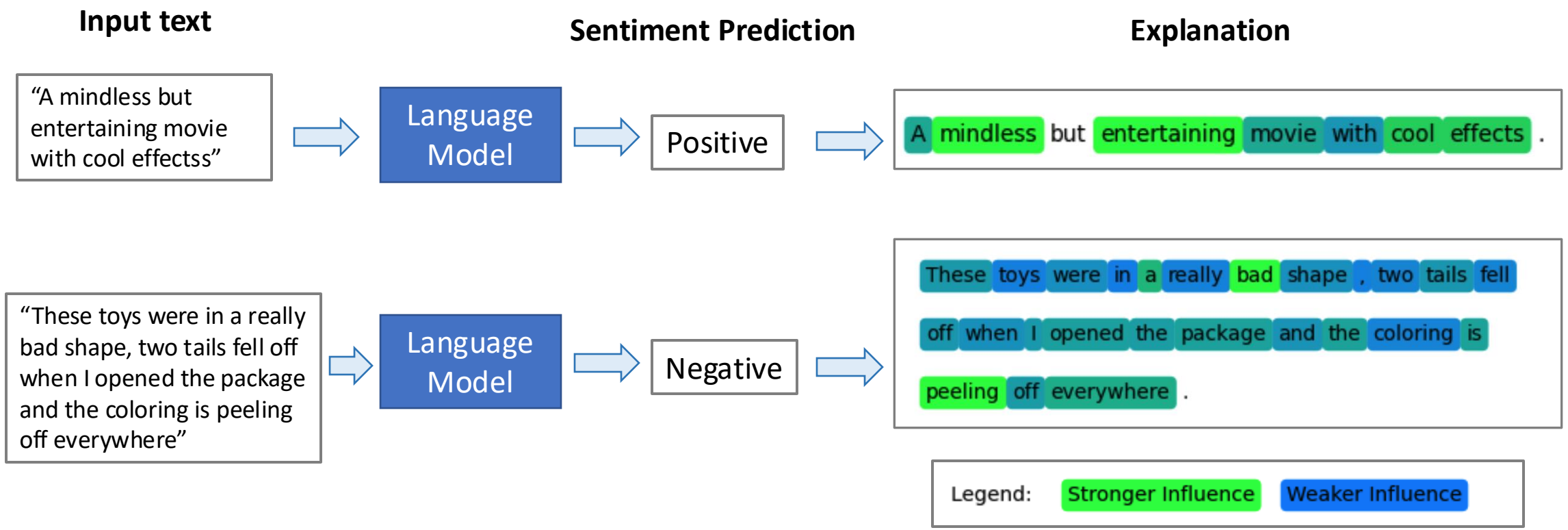
explanation of "dog"



Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization."

Example: Natural Language Processing

Explanation in Natural Language Processing: important tokens that lead to the prediction



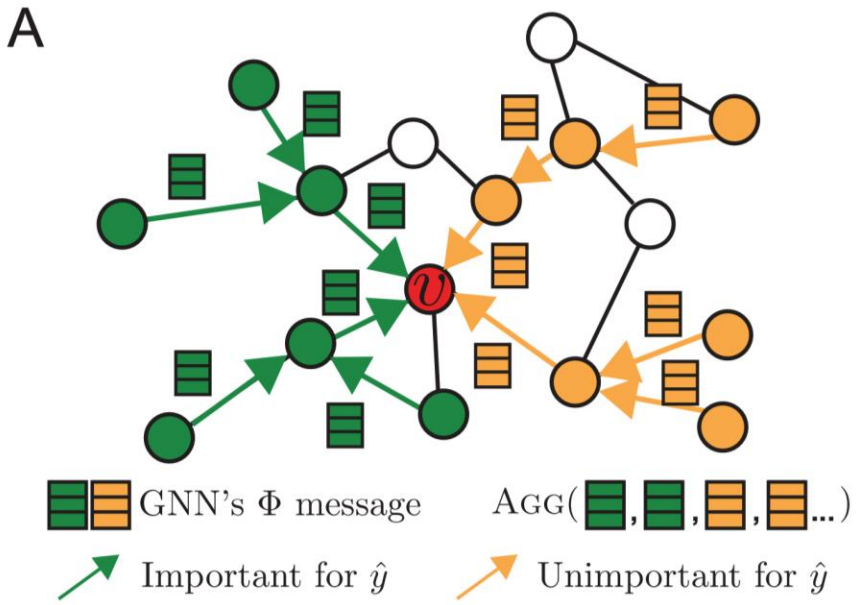
Dunn, Andrew, Diana Inkpen, and Răzvan Andonie. "Context-Sensitive Visualization of Deep Learning Natural Language Processing Models."

Example: Graph Learning

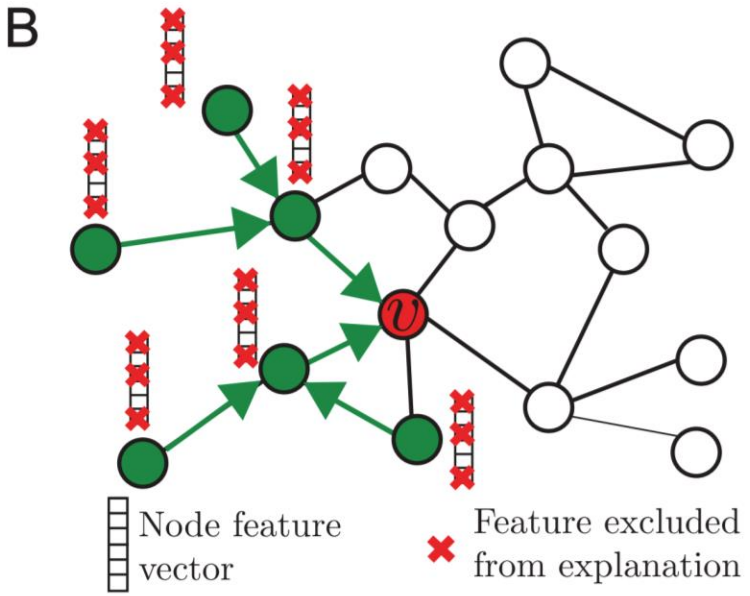
Explanation in Graph Learning: an important **subgraph structure** and a small **subset of node features** that play a crucial role in GNNs prediction

Explanations for prediction at **node v**

A: Import subgraph structure



B: important subset of features



Ying, et al. "Gnnexplainer: Generating explanations for graph neural networks."

Reasons for Explainability

Why do we need Explainability?

- **Trust:** Explainability is a prerequisite for humans to **trust and accept** the model's prediction.
- **Causality:** Explainability (e.g. attribute importance) conveys **causality** to the system's target prediction: attribute X causes the data to be Y
- **Transferability:** The model needs to convey an understanding of decision-making for humans before it can be **safely deployed to unseen data**.
- **Fair and Ethical Decision Making:** Knowing the reasons for a certain decision is a societal need, in order to perceive if the prediction **conforms to ethical standards.**

Content

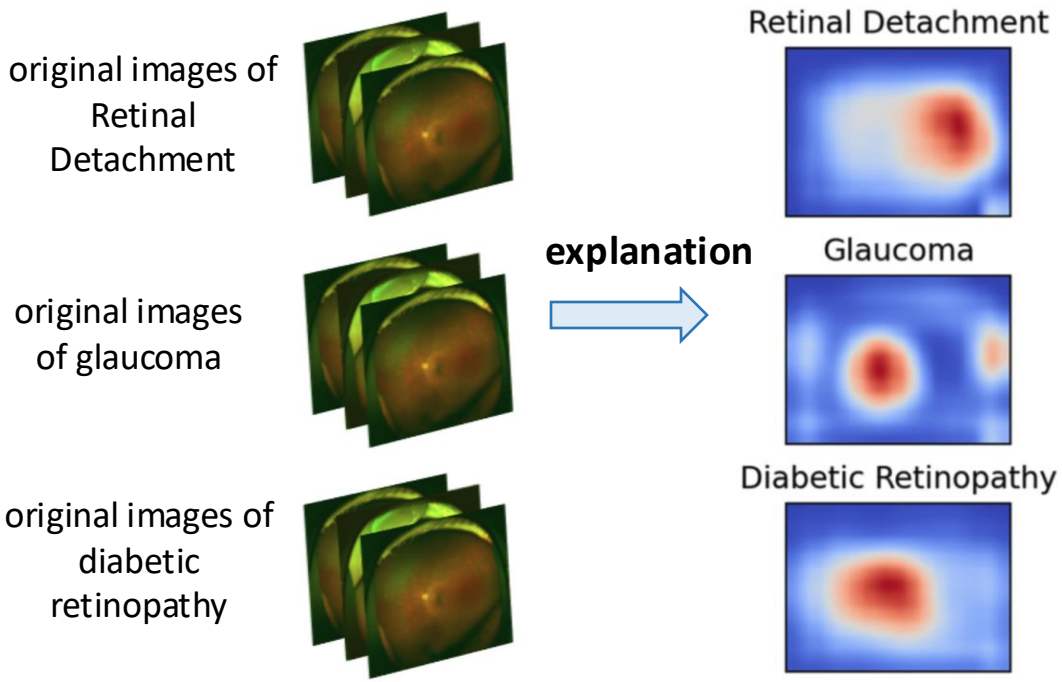
- Introduction to Explainability
- Explainability Settings
- Explainable Models
- Gradient-based Methods
- Perturbation Methods

Explainability Settings (1)

By target:

- **Instance-level:** a **local** explanation for a single input x and the prediction $\hat{y}$
  - identify the important components of individual instances

- **Model-level:** a **global** explanation for a specific dataset D or classes of D
  - provide **high-level insights** into the model's decision-making behaviors



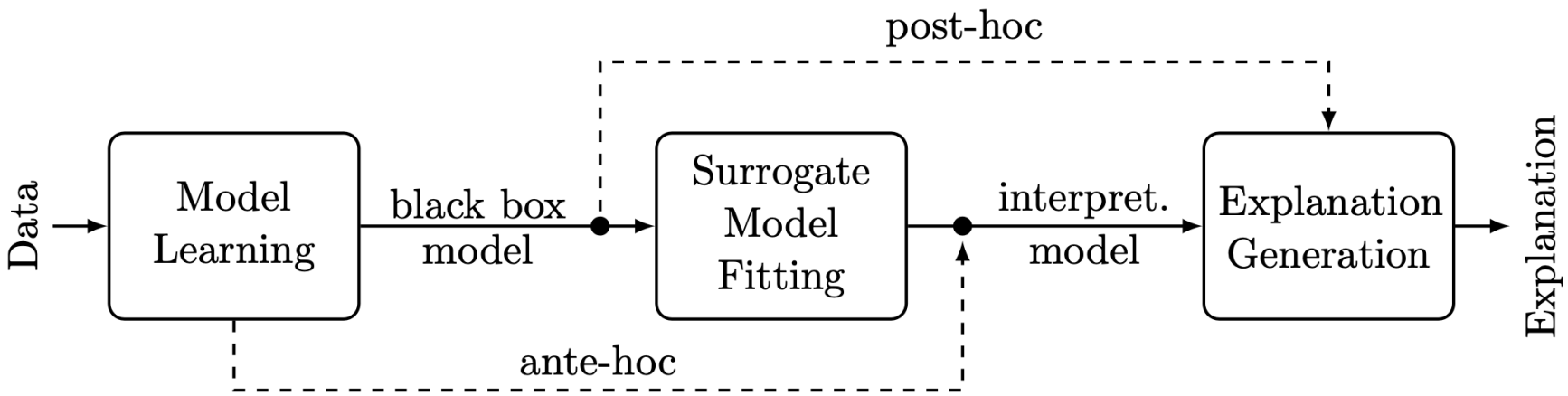
Example: model-level explanations for each class

Engelmann, Justin, Amos Storkey, and Miguel O. Bernabeu. "Global explainability in aligned image modalities."

Explainability Settings (2)

By stages:

post-hoc: Explainability is created after model creation **(for black-box models)**



ante-hoc: Explainability is **built-in** from the beginning of the model creation **(for intrinsically interpretable / white-box models)**

By applicability of the method:

model-specific: the machanism for generating explanation is **model-dependent** and works only for a specific model.

model-agnostic: the machanism for generating explnation is **applicable** for many or even all model classes

Content

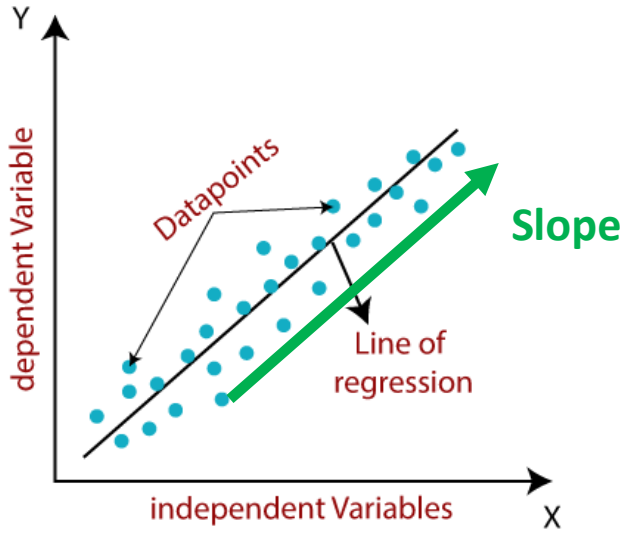
- Introduction to Explainability
- Explainability Settings
- Explainable Models
- Gradient-based Methods
- Perturbation Methods

Explainable Models: Linear regression

- **Linear regression**
  - **Slope is explainable** (how much does one variable affects a prediction)
  - $y = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots$

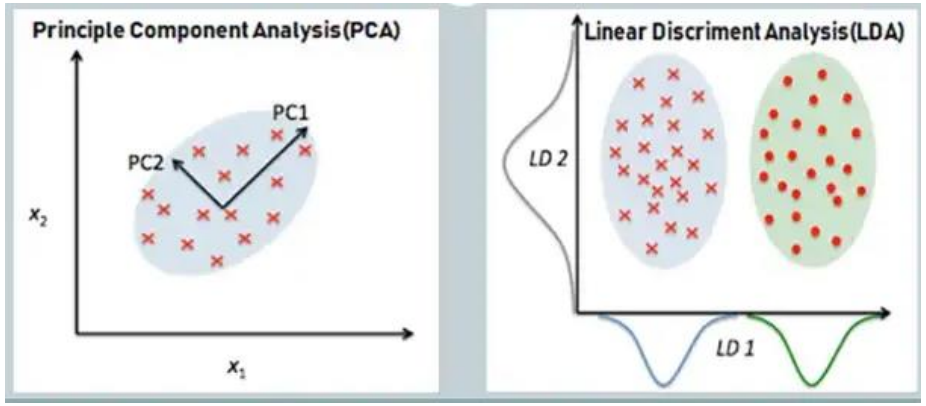
prediction weights features

- Each feature has an associated **weights**, indicating its **importance**
  - "A change of Δx amount to feature x_1 will result in increase of prediction by Δy

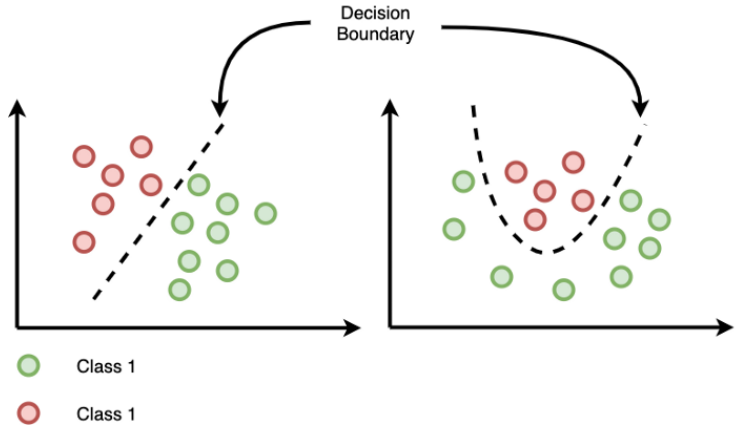


Explainable Models: Dimension Reduction

- **Dimension reduction**
  - Dimension reduction allows us to visualize the training data distribution



Source

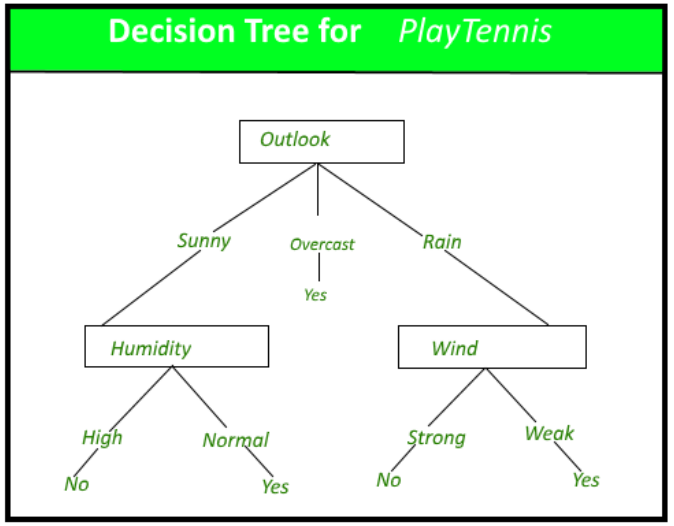


Source

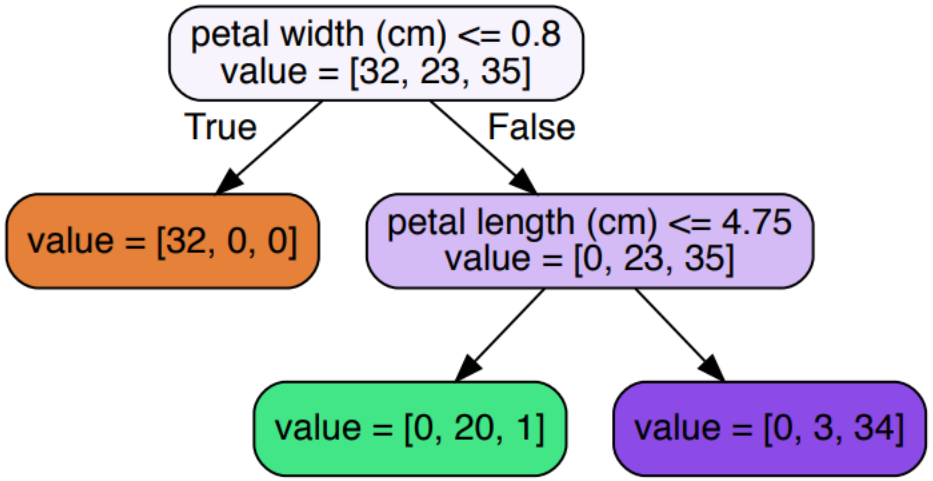
- Decision boundary can be visualized and understood
  - Instances at the boundary characterizes how different classes are different

Explainable Models: Decision Tree

- **Decision trees** are very explainable!
- On every node of the decision tree, we understand a criteria for prediction
- We can perform statistics for each decision node
  - E.g. if the condition of the node is met, **80% of the instances will be classified as being positive**



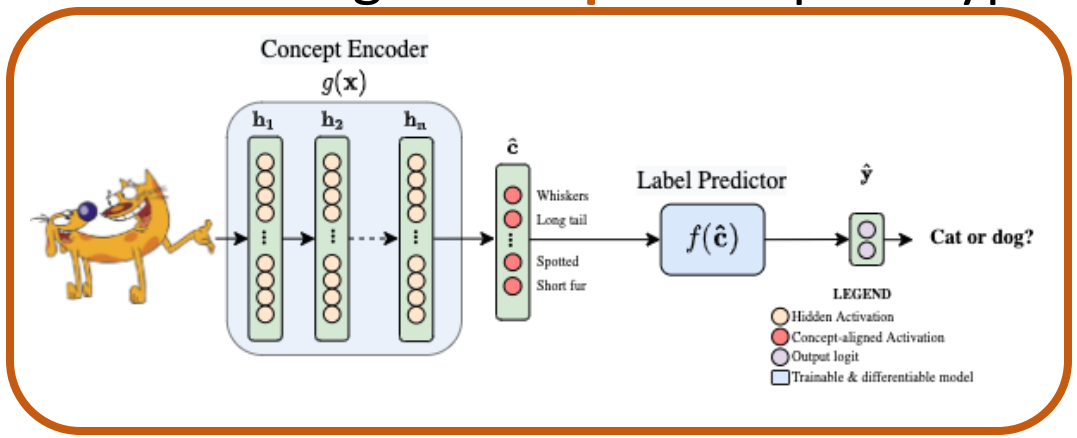
Source

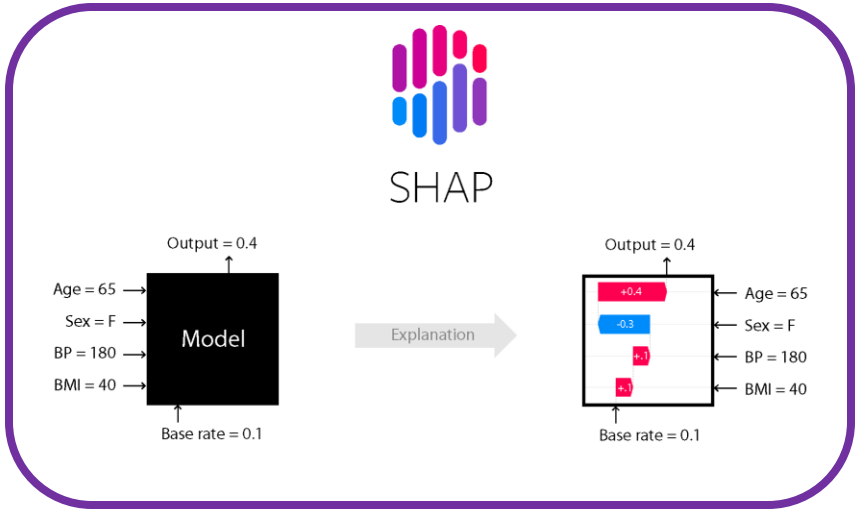


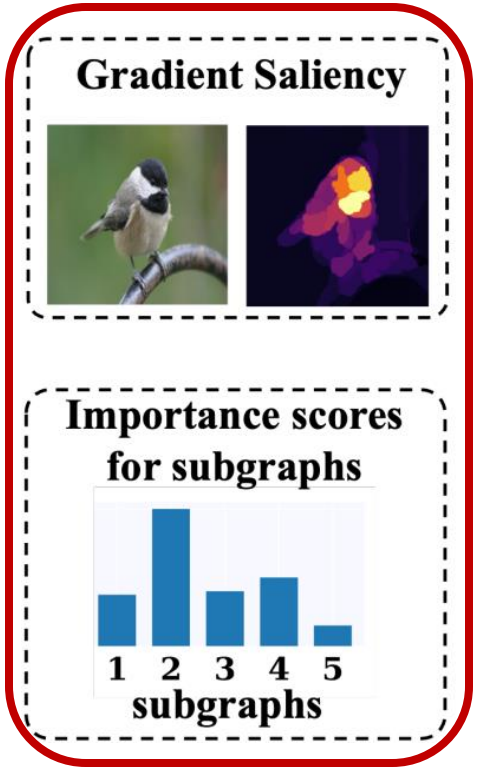
Source

Explainable Characteristics

- What makes model explainable?
  - **Importance** values (for pixels, features, words, nodes in graphs …)
  - **Attributions**: straightforward relationships between prediction and input features
  - Encourage **concepts** and prototypes





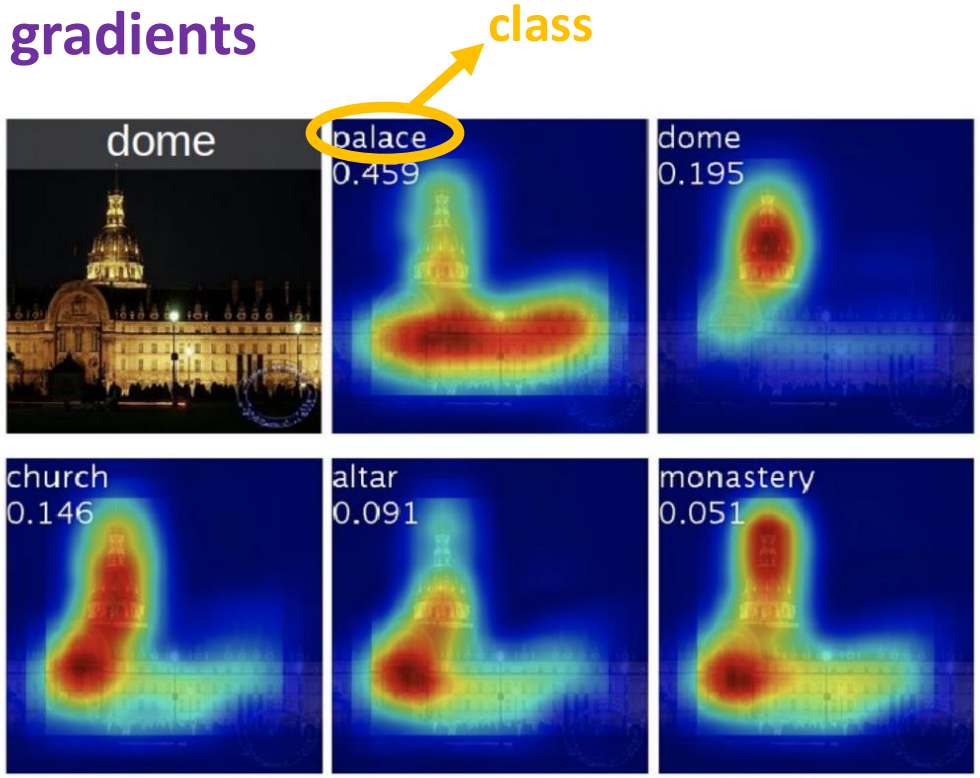


Content

- Introduction to Explainability
- Explainability Settings
- Explainable Models
- Gradient-based Methods
- Perturbation Methods

Gradient-based Explanation

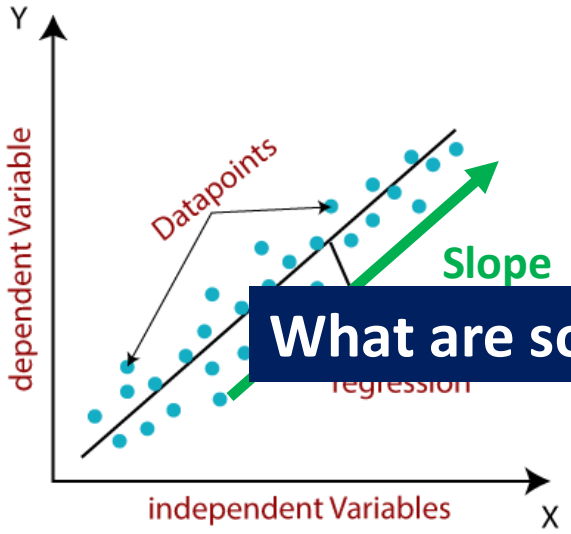
- **Gradient-based methods** identify the **saliency of input features** based on gradient signals passed from output to **input features**
- **Intuition: important features tend to have high gradients**
- **We typically care about magnitude**



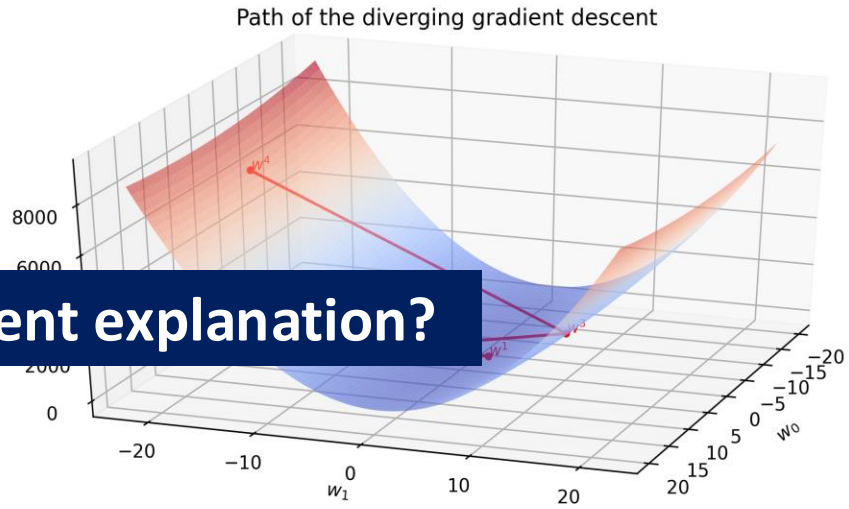
Saliency Map (in the form of a heatmap) highlights the discriminative regions, revealing model's decision-making logic

Zhou, Bolei, et al. "Learning deep features for discriminative localization."

Why Gradients?



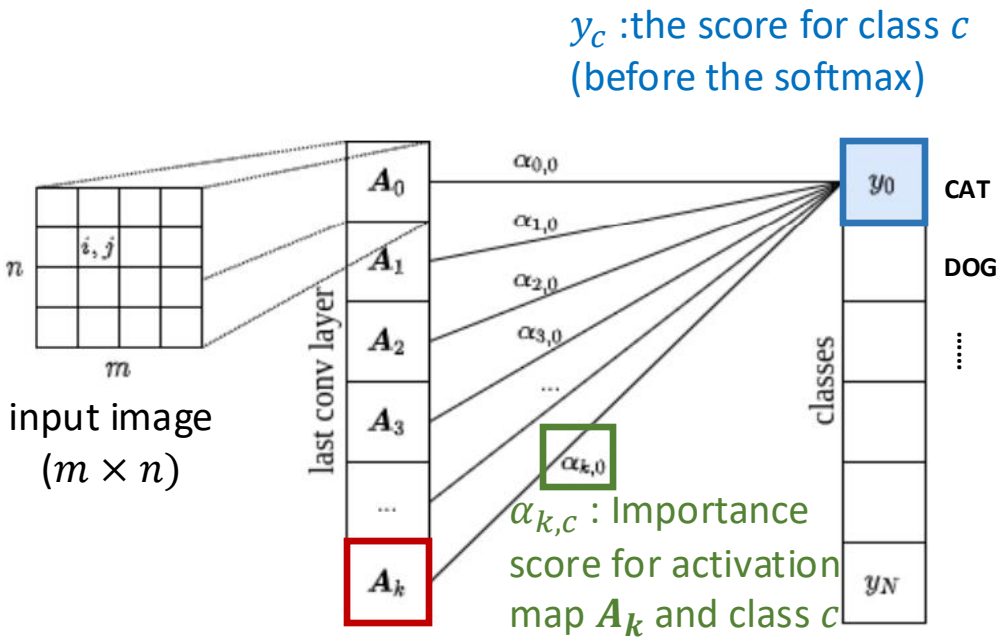
What are some problems with Gradient explanation?



- **The optimization landscape** of deep networks is very complex and a global scope is no longer possible
- Gradient is a **local approximation** of the slope
- Each **dimension** of the gradient vector can indicate how much the prediction is impacted by the input

Grad-CAM (1)

- **Gradient-weighted Class Activation Map (Grad-CAM):**



Architecture of CNN

gradients of the output w.r.t. the last convolutional layer

Importance score $\alpha_{k,c}$ for activation map A_k and class c:

$$\alpha_{k,c} = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\partial y_c}{\partial A_{k,i,j}}$$

$A_{k,i,j}$: value at (i, j) in the $m \times n$ feature map A_k

Saliency map for class c:

$$map_c = \text{ReLU}(\sum_k \alpha_{k,c} A_k)$$

map_c has **the same dimension as the input image**

Why ReLU?

Grad-CAM (2)

Example: class 0 is "cat"

Important score $\alpha_{k,0} = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\partial y_0}{\partial A_{k,i,j}}$

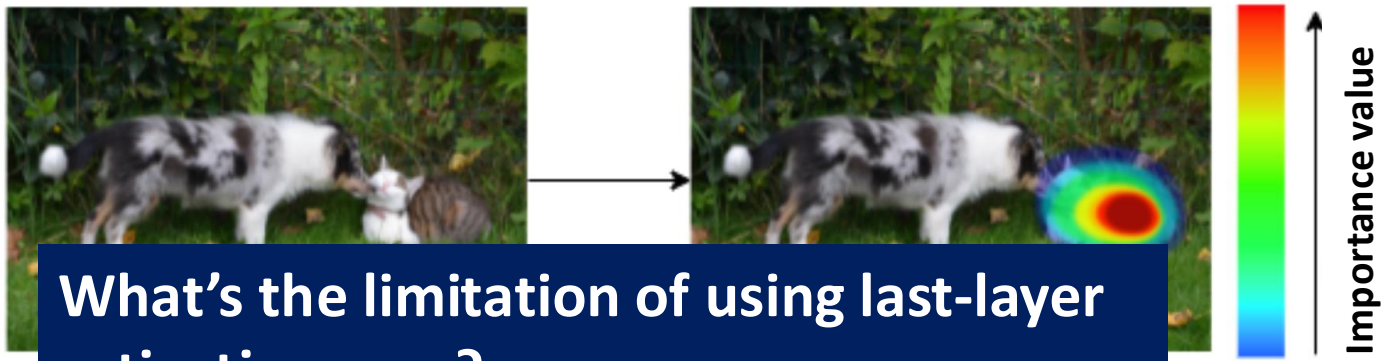
Saliency map of "cat" :
(of the same size as the original image)

$$map_0 = \text{ReLU}(\alpha_{0,0} A_0 + \alpha_{1,0} A_1 + \ldots + \alpha_{k,0} A_k)$$

A_k: feature activation map with size $m \times n$

corresponds to a **pixel** on the original image

Visualization of the **Saliency map of "cat" :**



What's the limitation of using last-layer activation map?

Grad-CAM: Evaluation

- **Localization Evaluation:**
  - Given an image, first obtain class **predictions from the network**
  - Generate Grad-CAM **maps** for **each of the predicted classes**
  - **Binarize** with **threshold of 15% of max intensity**

Localize the flying kites (despite their small sizes)

Localize the man and the pizza (ignoring the woman)

Visual Explanations highlight image regions that are important for producing the captions

Grad-CAM:Comparison to DenseCap

Localizations of a global caption generated by Grad-CAM:

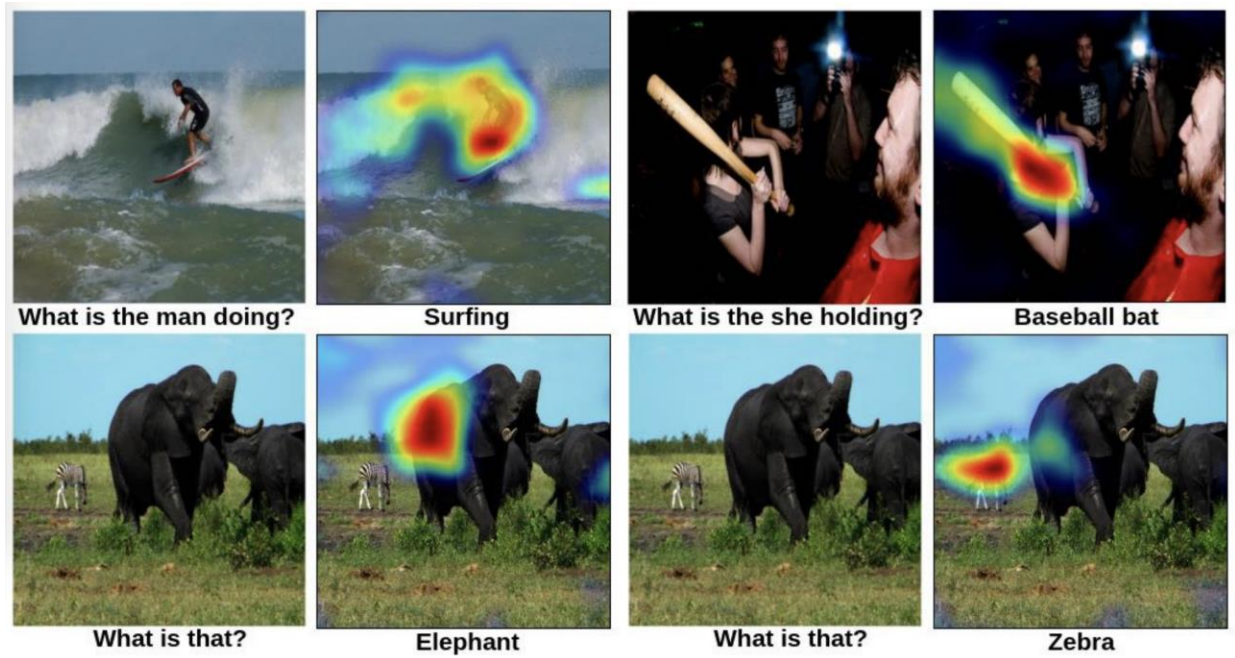
DenseCap: jointly localizes and generates captions for salient regions in each image.

Localization of DenseCap: bounding boxes for regions of interest

Localization of Grad-CAM: more **fine-grained** details **with importance values**

Grad-CAM: VQA Evaluation

- **Visual Question Answering (VQA):** VQA pipelines consist of **a CNN to process images** and **a language model for questions**. The model will predict the answer to the question.
- Grad-CAM: **visualize salient regions** of the image that **explain the answer**



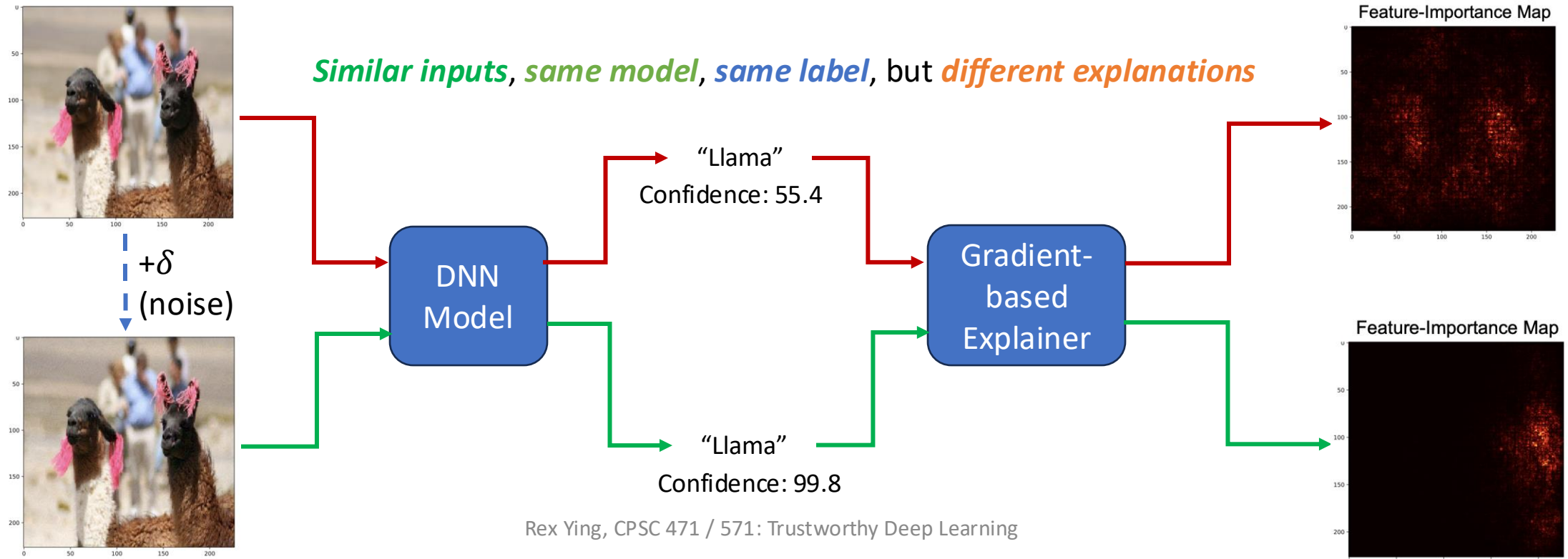
Grad-CAM: Comparison to Human Attention

- Use the **rank correlation** to compare the **Grad-CAM visualizations** and **Human attention maps** over visual question answering pairs
- **Correlation: 0.136**
- statistically higher than chance or random attention maps (**zero correlation**)

Human Attention:
Das et. al collected human attention maps for a subset of VQA dataset.
These maps have **high intensity** where **humans looked** in the image in order to answer a visual question.

Sensitivity of Gradients

- Saliency maps using a vanilla gradient are sensitive to small perturbations in the input instance.
  - Adding a small perturbation may change the interpretation significantly, even though the prediction is unchanged.



Integrated Gradients (1)

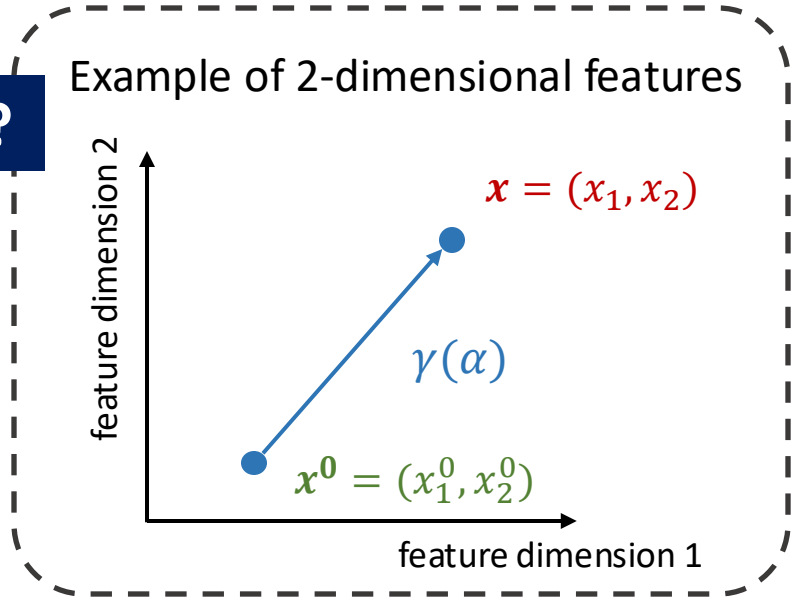
- **Integrated Gradients (IG):** given a **reference (baseline)** input $\boldsymbol{x^0}$, the integrated gradient of the model f w.r.t. the i-th feature x_i for an input $\boldsymbol{x}$:

$$IG_i(\boldsymbol{x}) = \left(x_i - x_i^0\right)\int_0^1 \frac{\partial f\left(\boldsymbol{x^0} + \alpha(\boldsymbol{x} - \boldsymbol{x^0})\right)}{\partial x_i} d\alpha$$

  - $\boldsymbol{x}$: input; $\boldsymbol{x^0}$: baseline input; x_i: i-th feature of $\boldsymbol{x}$; x_i^0: i-th feature of $\boldsymbol{x^0}$
  - **Integral path:** $\gamma(\alpha) = \boldsymbol{x^0} + \alpha \times (\boldsymbol{x} - \boldsymbol{x^0}), \alpha \in [0,1]$

- The reference (baseline) input: **Comparison to saliency?**
  - a black image or a zero embedding vector

IG_i: sensitivity of f to changes in the i-th feature from x^0 to x along $\gamma(\alpha)$ in direction i
Higher IG_i $\Leftrightarrow$ Higher importance of the i-th feature



Integrated Gradients (2)

- IG can be **approximated** by a Riemann summation of the integral

$$IG_i(\boldsymbol{x}) \approx \left(x_i - x_i^0\right) \frac{1}{M} \sum_{k=1}^{M} \frac{\partial f\left(\boldsymbol{x}^0 + \frac{k}{M}\left(\boldsymbol{x} - \boldsymbol{x}^0\right)\right)}{\partial x_i}$$

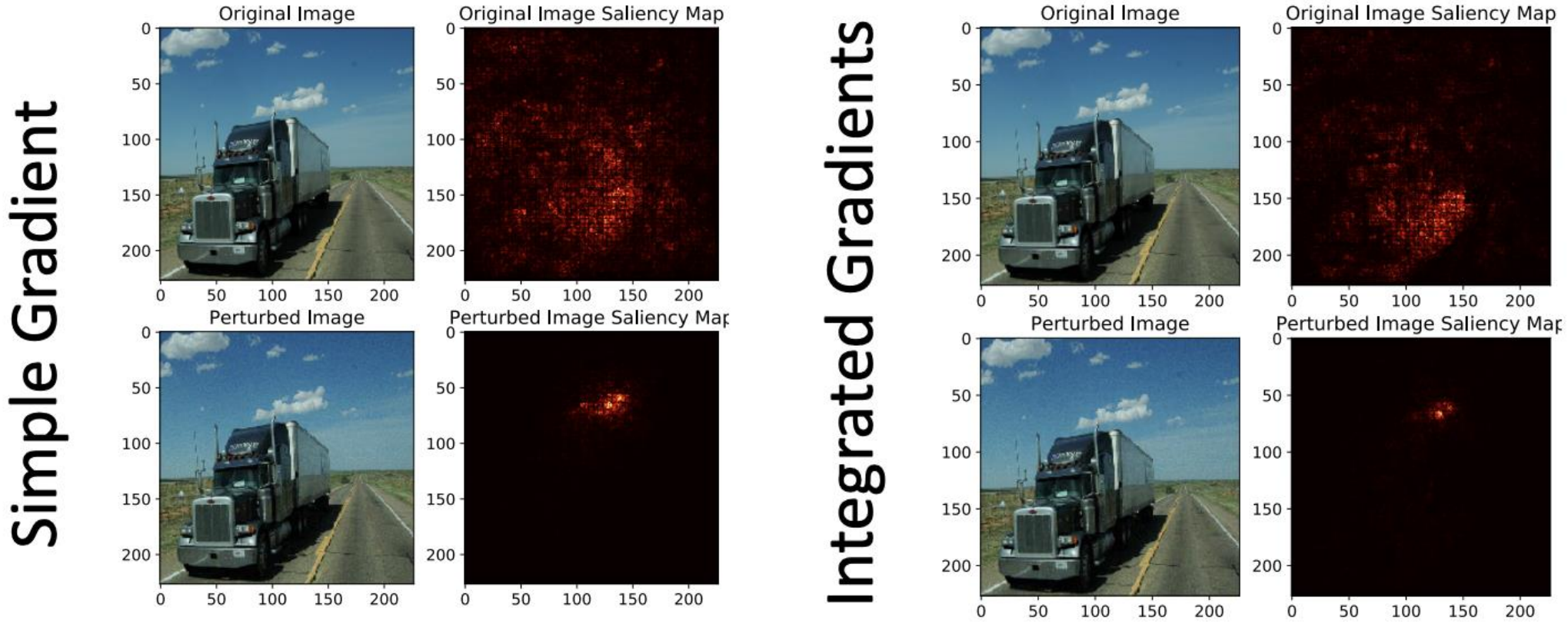
  - M is the number of steps in the Riemann approximation of this integral (recommended M: 20 to 300 steps)

Observation: Integrated Gradients can better reflect distinctive features of the input image

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks."

Sensitivity of Integrated Gradients

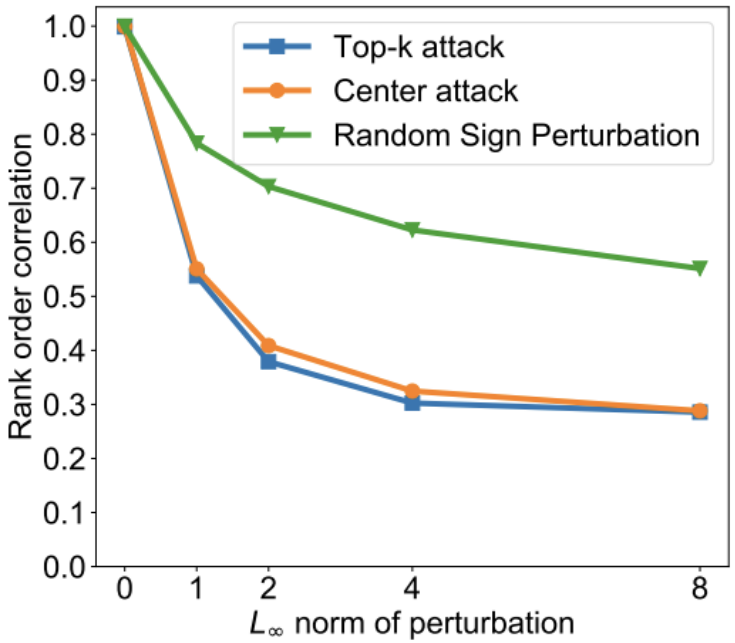
- Integrated Gradients is still vulnerable to adversarial noise.



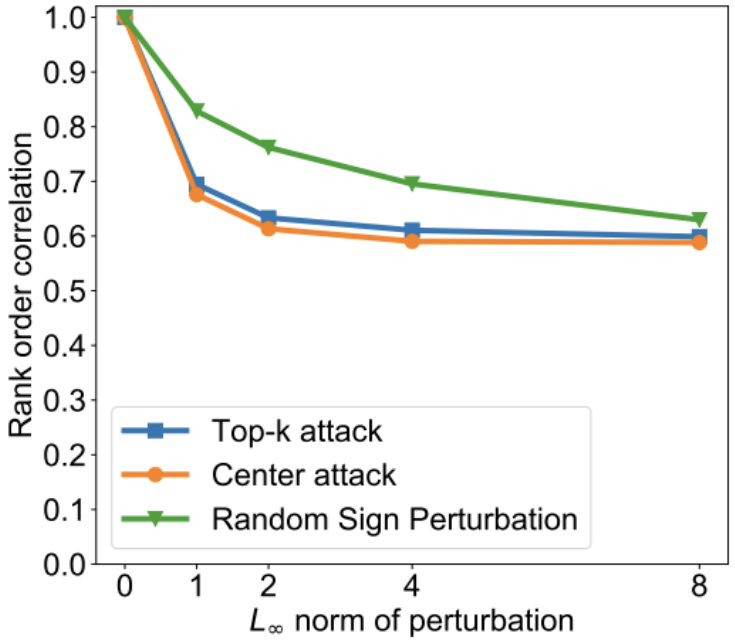
Sensitivity of Integrated Gradients

- Integrated Gradients is still vulnerable to adversarial noise.
- However, statistically Integrated Gradients is **more robust** than the Vanilla Gradient-based method

Simple Gradients



Integrated Gradients



- **Top-K attack**: optimize the noise to minimize the importance of top-k most important features
- **Center attack**: optimize the noise to maximize the spatial displacement of the saliency map
- **Random sign**: randomly add a random nosie $\pm\epsilon$

The y-axis is the correlation of the Saliency map at x_0 and at perturbed input $x_0 + \delta$

source

Comparison of IG and Grad-CAM

- IG is more **flexible** in the setting than Grad-CAM
  - Adjustable parameters: baseline input, number of steps M, etc.
- IG satisfies the **"sensitivity" atom** by introducing a baseline input
  - **"sensitivity" atom:** if input $\boldsymbol{x}$ differs from $\boldsymbol{x}'$ along feature x_i only, and the prediction $f(\boldsymbol{x}) \neq f(\boldsymbol{x}')$**.** Then x_i should have a **non-zero importance score**.
  - Grad-CAM might generate the same saliency maps for $\boldsymbol{x}$ and $\boldsymbol{x}'$
- IG is **more robust** to noise / small perturbations on the input
  - The saliency maps of Grad-CAM might change drastically due to small variation of input (e.g., for input $\boldsymbol{x}$ and perturbed input $\boldsymbol{x}'$, the prediction $f(\boldsymbol{x}) = f(\boldsymbol{x}')$, the saliency maps differ greatly)
- IG has **larger computation complexity**.

Content

- Introduction to Explainability
- Explainability Settings
- Explainable Models
- Gradient-based Methods
- Perturbation Methods

- **When do we need black-box explainability?**
- **How should we tackle black-box explainability?**

Perturbation-based Explanation

- **Perturbation methods**
  - **Post-hoc, model-agnostic** explanation for **black-box models**
  - Use perturbation (altering or removing the input features) to identify features that can greatly influence predictions
  - **Intuition:** the model's performance **decreases dramatically** when the model does not have access to **the most relevant information**.
  - The performance drop can be used to create a **sensitivity heatmap** to visualize the **importance of each portion**

Discussion: How are these techniques related to adversarial attacks?

Principle of Perturbation-based Explanation

original performance | **drop:** 0.93-0.56=0.37 | **drop:** 0.93-0.79=0.14 | **drop:** 0.93-0.91=0.02

"Hummingbird" = 0.93 | "Hummingbird" = 0.56 | "Hummingbird" = 0.79 | "Hummingbird" = 0.91

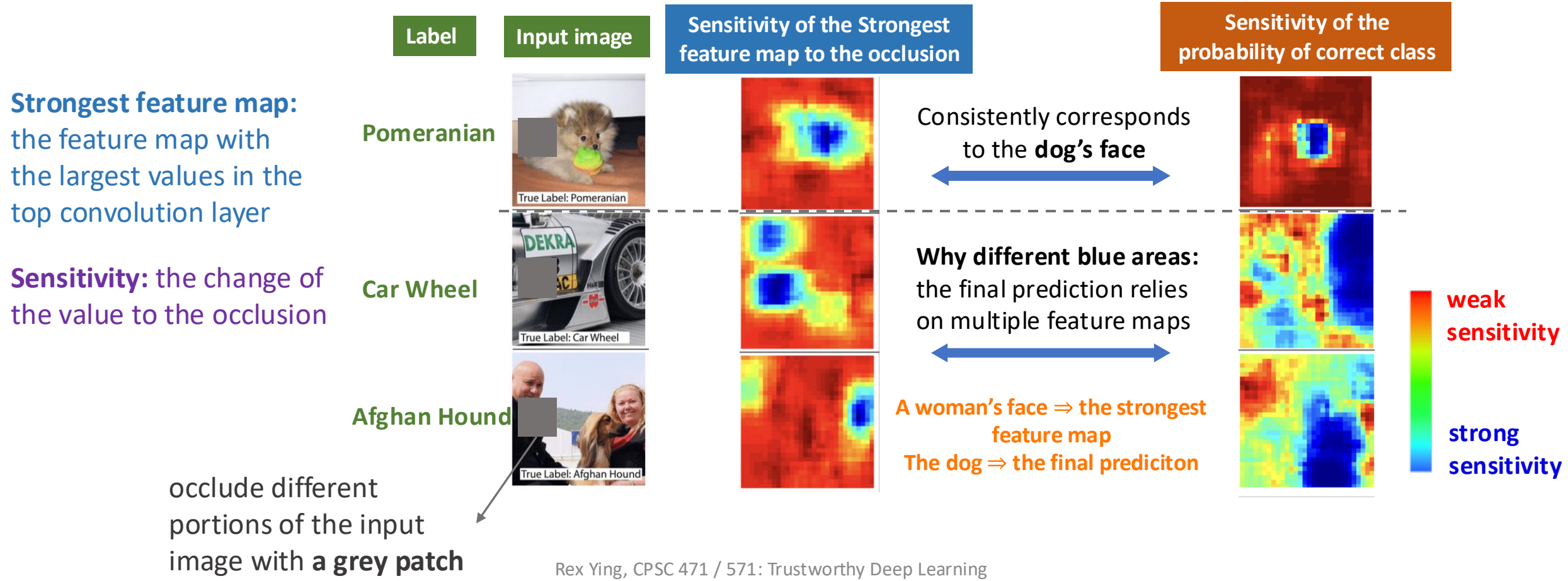
Highest performance drop indicates the **highest importance** of this portion

small performance drop indicates a **lower importance**

Use **grey patch** to occlude a portion of input instance

Occlusion Sensitivity

- **Occlusion Sensitivity:** measure the **sensitivity** of the model's output to **occlusion** in different regions by a small grey patch



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks."

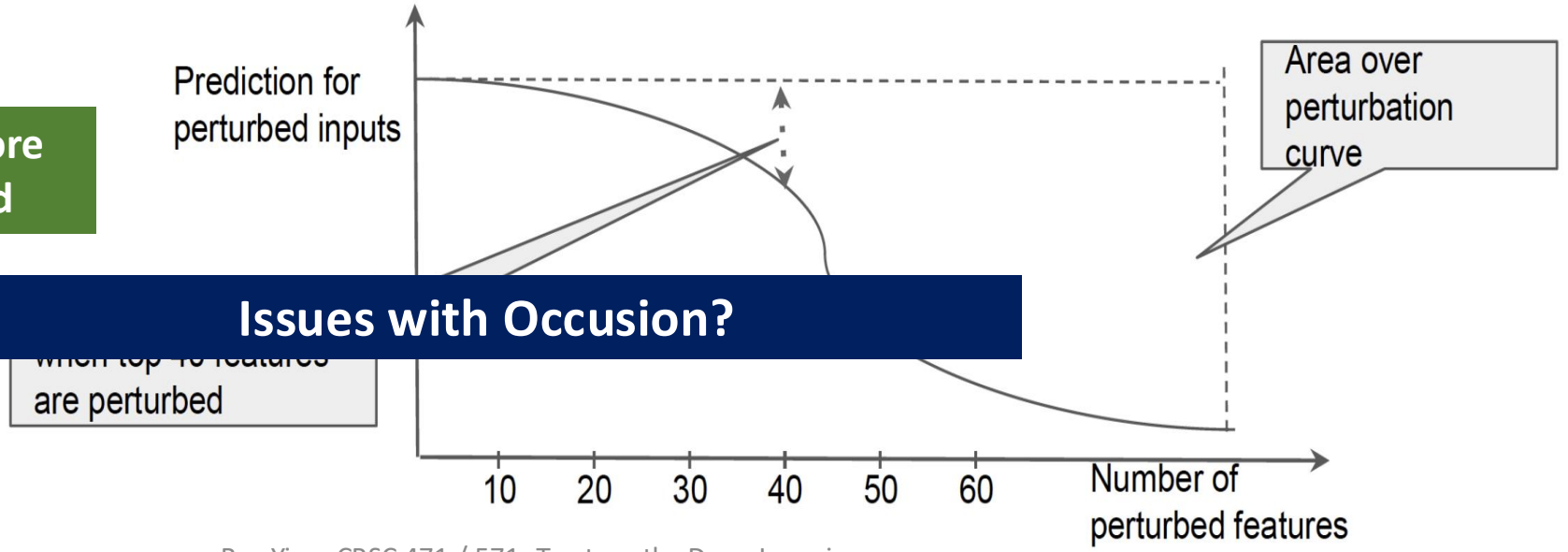
Meaningful Perturbation and Evaluation

- Other types of **perturbation** [1]:
  - **Blur:** blurring the region area
  - **Constant:** replacing with a constant value
  - **Noise:** adding noise to the region

[1] Fong, Ruth C., and Andrea Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation."

- **Area over Perturbation Curve (AOPC):** to evaluate the perturbation methods

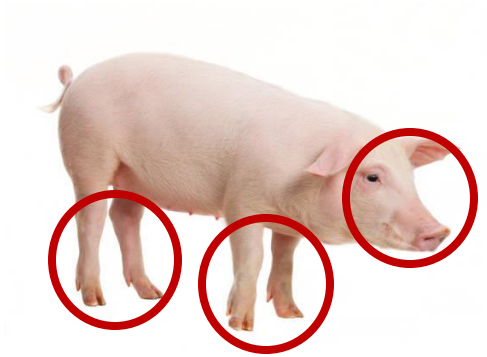
A higher AOPC indicates a more efficient perturbation method



Issues with Occusion?

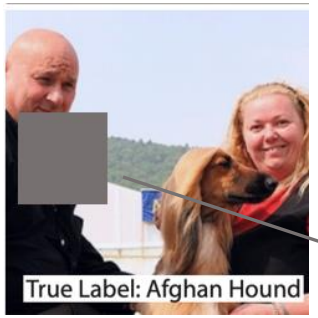
Issues with Occlusion and Perturbation

- Efficiency: occlusion at token / pixel level is very time-consuming
  - Too many forward propagations are needed
- Correlation is not modeled well
  - *E.g.* Presence of Part A "or" part B results in prediction of a class



Removing one of the patch may not have much effect on the logits

- The shape of the occluding patch is pre-defined



The size of the **grey patch** is always the same – **same granularity**

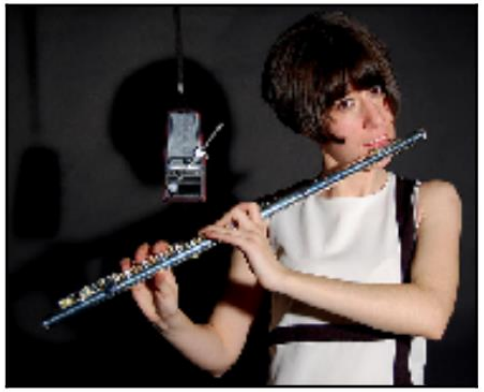
Explanation as Masks

- The learnable mask consists of values between 0 and 1
- $\text{argmin}_M \, f(\phi(\boldsymbol{x})) + g(M)$

  Masked prediction **Regularization**

Original input

flute: 0.9973

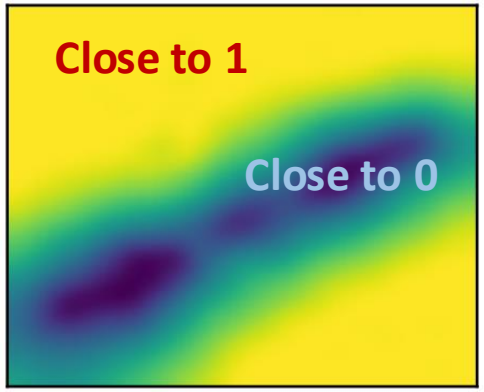


Decrease in model confidence after masking

flute: 0.0007

Explanation of "flute" class in this instance

Learned Mask



Explanation as Masks

- The learnable mask consists of values between 0 and 1
- $\text{argmin}_M \, f\big(\phi(\boldsymbol{x})\big) + g(M), \quad 0 \leq M \leq 1$

  Masked prediction Regularization

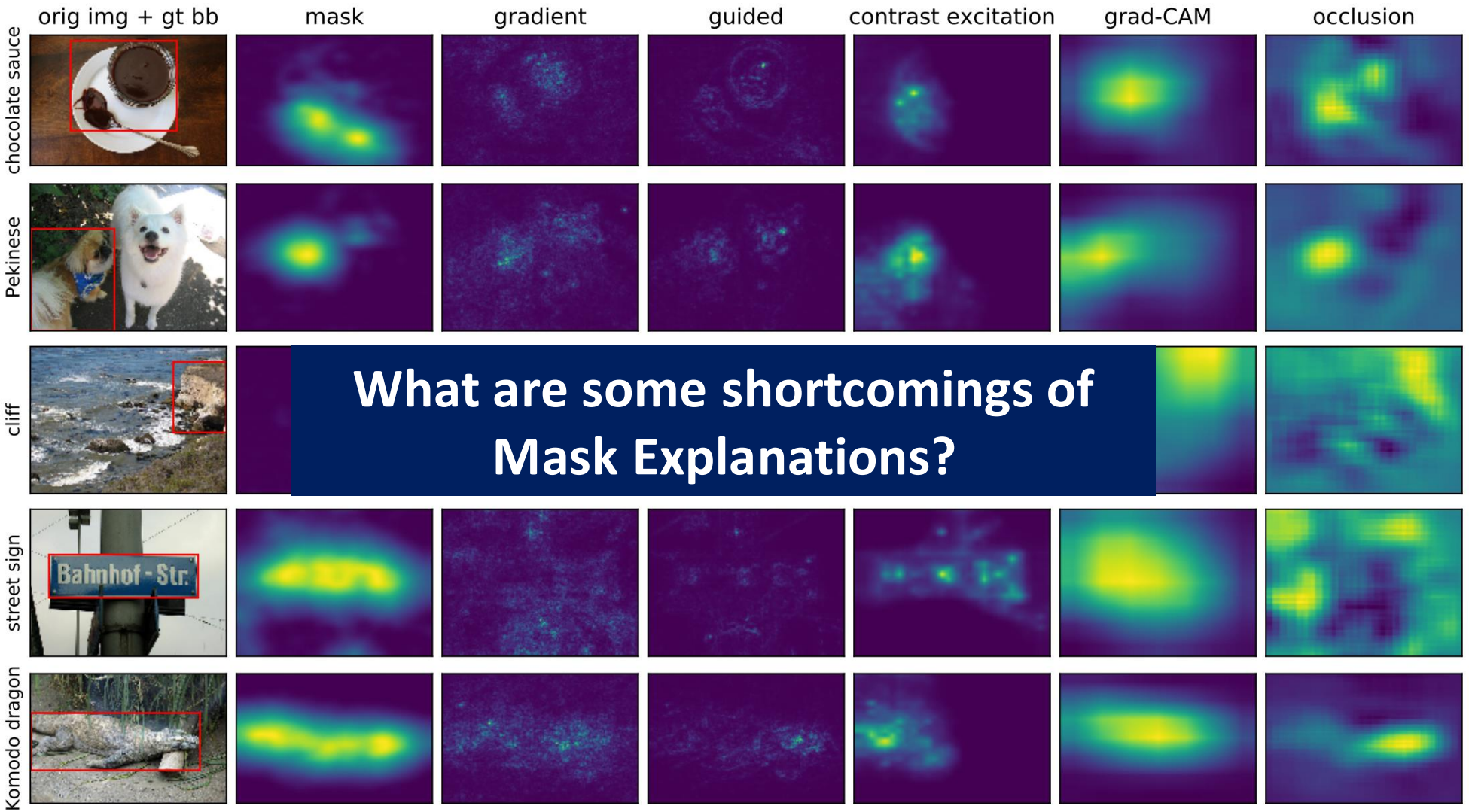
Mask $\boldsymbol{M}$ is randomly initialized

- Replace the masked portion of the input
  - $\phi(\boldsymbol{x}) = \boldsymbol{x} \odot M + \boldsymbol{x}^0 \odot (\mathbf{1} - M)$
- Control the desirable properties of the mask
  - $g(M) = \lambda_1 \|1 - M\| + \sum_u \nabla M(u)$

What's the rationale behind $g(M)$?

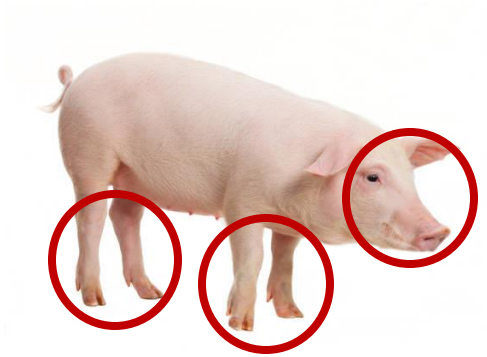
Image gradient (difference between adjacent pixels)

Comparison with Other Methods



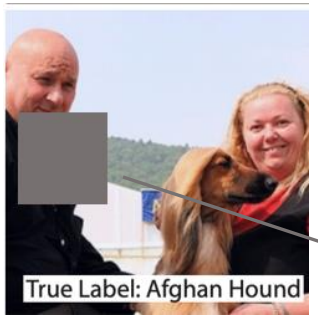
Does Masking Solve These Challenges?

- **Efficiency**: occlusion at token / pixel level is very time-consuming
  - Too many forward propagations are needed
- **Correlation** is not modeled well
  - *E.g.* Presence of Part A "or" part B results in prediction of a class



Removing one of the patch may not have much effect on the logits

- The **shape** of the occluding patch is pre-defined



The size of the **grey patch** is always the same – **same granularity**

What are more issues of masking-based methods?

Problems with Mask Explanations

- **Efficiency**
  - Also slow when the input is large (too many pixels, tokens, nodes etc.)
  - Requires **optimization** for generating explanation for every instance
- **Stability**
  - Explanation can vary across different runs depending on **random seed** of the optimization
  - Mask can get stuck in **local optimum** instead of global optimum
- **Robustness**
  - The process is analogous to the problem of finding **adversarial examples**
  - Explanations might not provide the true insight!

Summary

- The goal of XAI is to enable users to **understand the decision-making** of the model and **gain the trust** of human users of the deep learning system.
- The explanation for a DL system can be categorized as being **model-level or instance-level**; **ante-hoc or post-hoc**; **model-specific or model-agnostic**.
- **Explainable Models** include decision trees, linear models, etc.
- **Gradient-based Explanation**: Saliency, Grad-CAM, Integral gradient
- The change of prediction to **perturbation** over individual regions reveals the importance of the specific region
  - Occlusion and mask-based optimization
- All methods introduced today are instance-level explainability methods