



Tutorial on Deep Learning Interpretation: A Data Perspective

Zhou Yang, Fan Yang, Ninghao Liu, Xia (Ben) Hu and Fang Jin





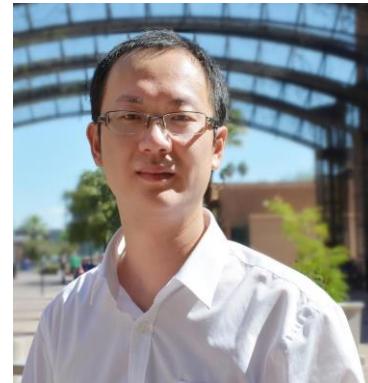
Zhou Yang
GWU



Fan Yang
Rice



Ninghao Liu
UGA



Hu Ben Xia
Rice



Fang Jin
GWU

<https://sites.google.com/gwmail.gwu.edu/tutorial-proposal-cikm-2022/home>

Roadmap

- Part 1: Introduction
- Part 2: Image-based Model Interpretation
- Part 3: Graph-based Model Interpretation
- Part 4: Text-based Model Interpretation
- Part 5: Deep Reinforcement Learning Interpretation
- Part 6: Hands-on Examples

Part 1: Introduction

Outline

1. Introduction to Interpretable Machine Learning
2. Interpretable Deep Learning
3. Evaluation of Interpretation

Outline

1. Introduction to Interpretable Machine Learning
2. Interpretable Deep Learning
3. Evaluation of Interpretation

Machine Learning is Everywhere

Playing Go



Medical Diagnosis



Scene Understanding



Voice Recognition

Machine Learning is Everywhere



What have been learned inside the models?



Interpretable Machine Learning



Safety of AI Models

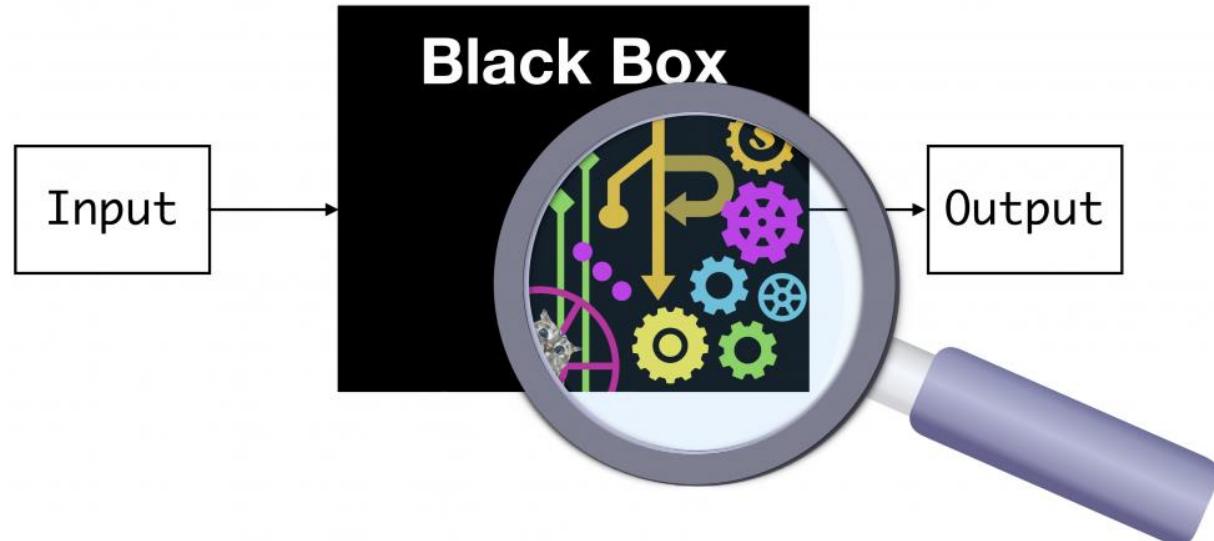


Trust of AI Decision

Policy and Regularization



What is Interpretable Machine Learning

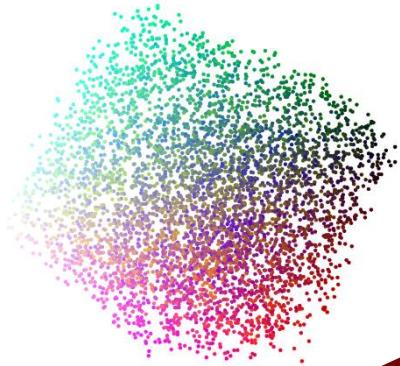


Interpretable Machine Learning is the ability to explain or to present the behavior of a black-box ML model in understandable terms to a human

Bang, Seojin, et al. "Explaining a black-box by using a deep variational information bottleneck approach." AAAI, 2021.

Overview

“ Which part of data are most responsible for a specific prediction ”



Data

Image / Text / Tabular

“ What are the model looking for when making the predictions? ”

Task

Classification / Ranking / Outliers



“ Why a specific instance has been classified into a certain category? ”

Developer / Expert / End-User



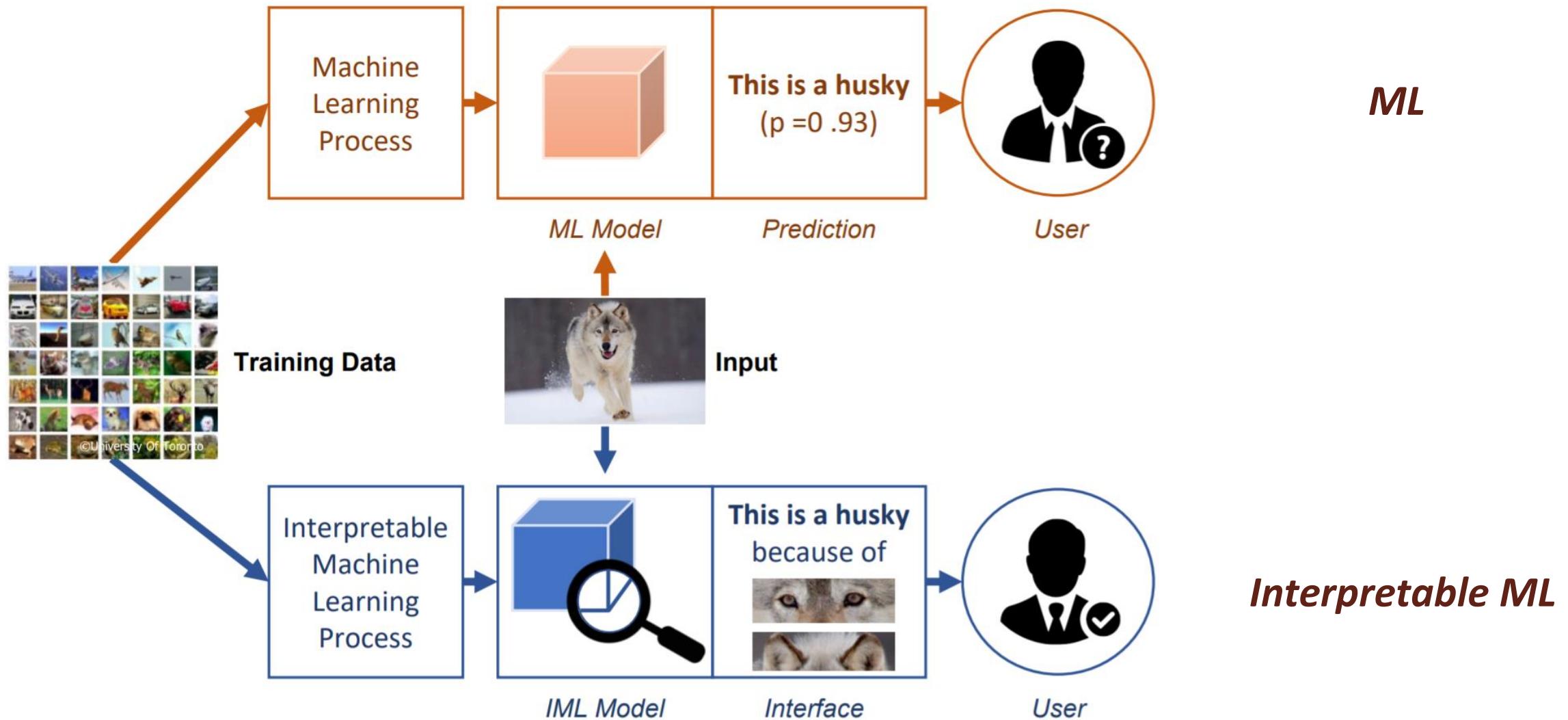
Stake Holder



Model

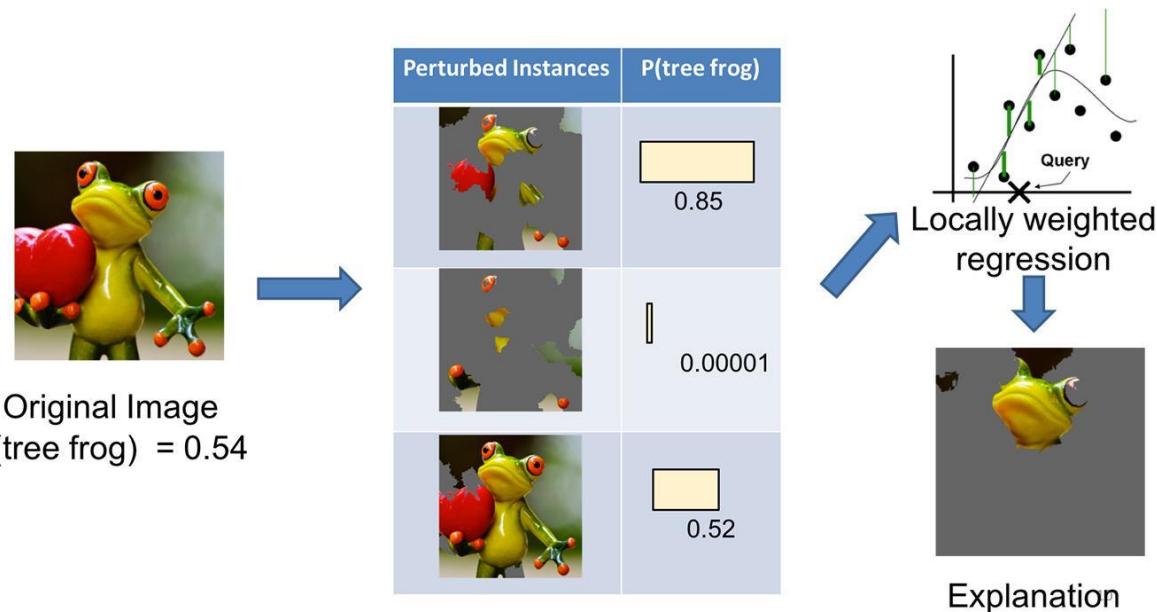
CNN / CF / OD

Pipeline

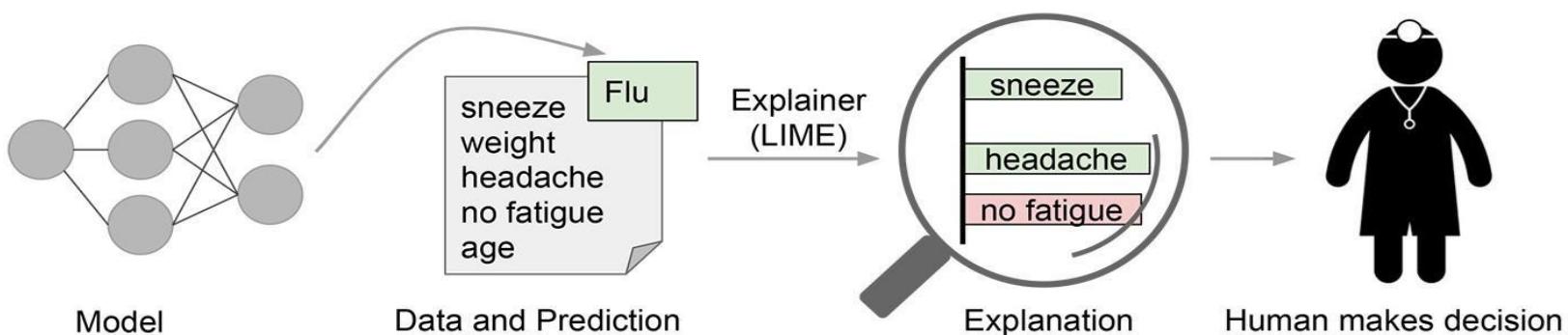


Examples

1 Image Classification



2 Medical Diagnosis

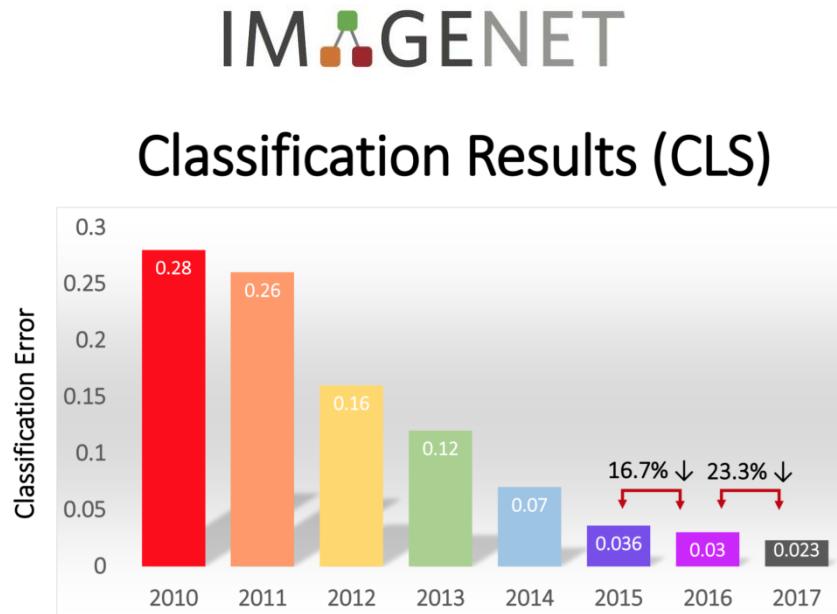


Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." KDD, 2016.

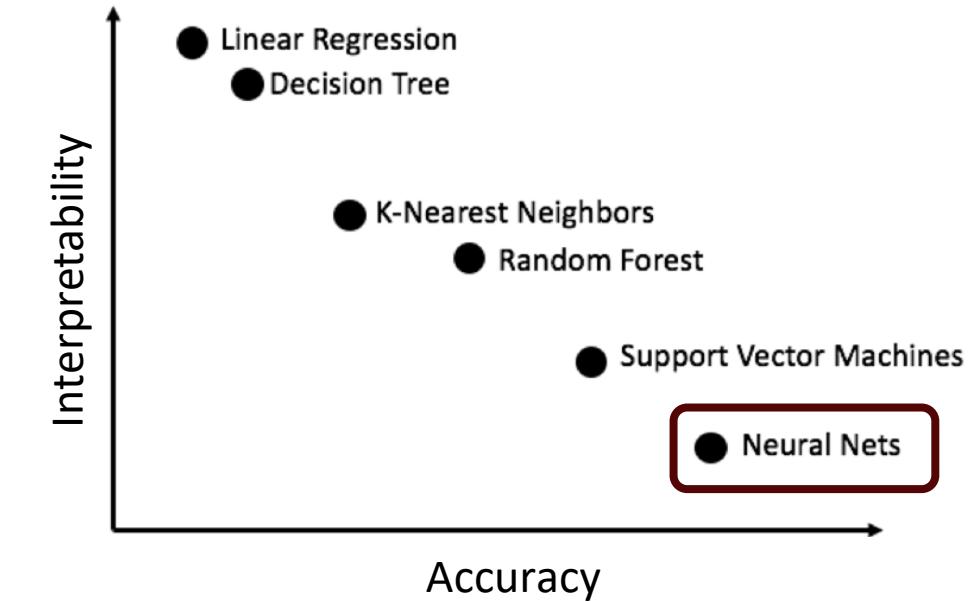
Outline

1. Introduction to Interpretable Machine Learning
2. Interpretable Deep Learning
3. Evaluation of Interpretation

Interpretable Deep Learning



DNNs make lots of *progresses*

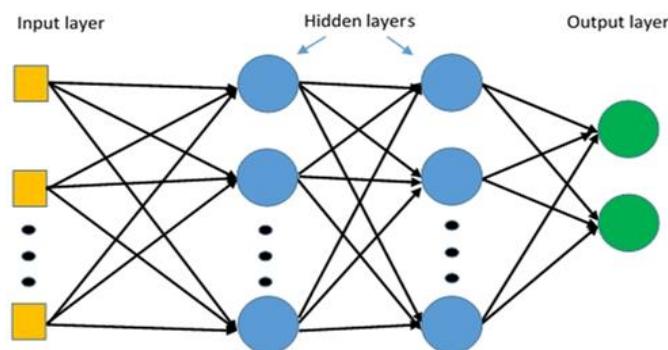


DNNs are regarded as *black boxes*

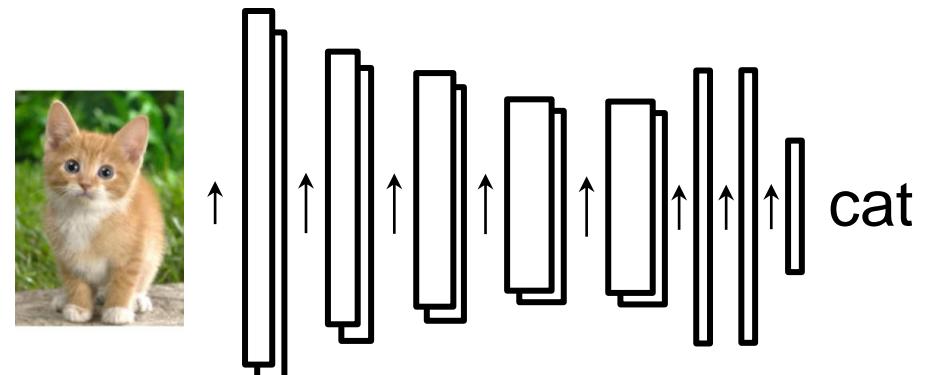


Definition - Interpretability of DNNs

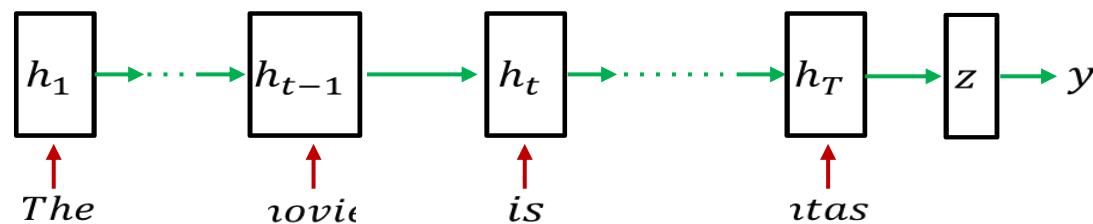
“Interpretability of DNNs enable us to explain the behavior of a black-box DNN model in understandable terms to humans”[1]



Multilayer Perceptron (MLP)



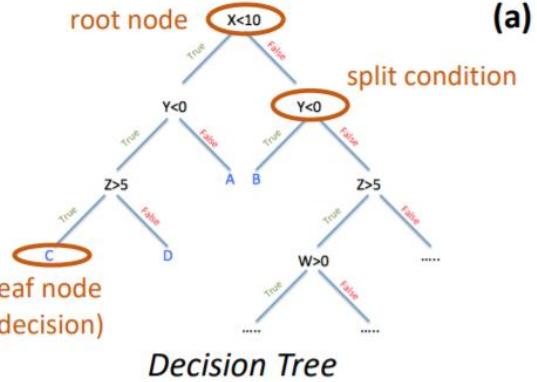
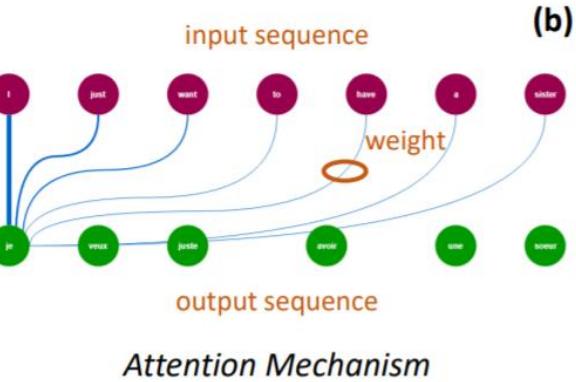
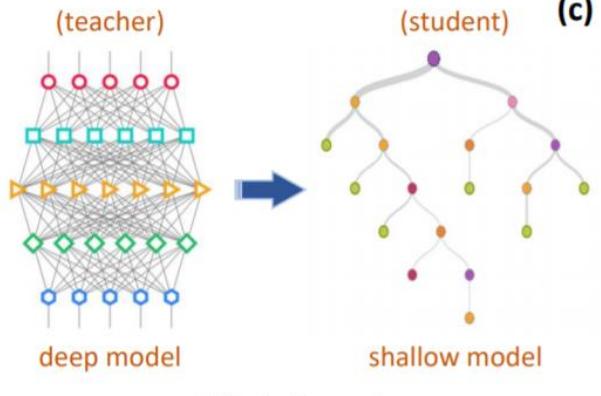
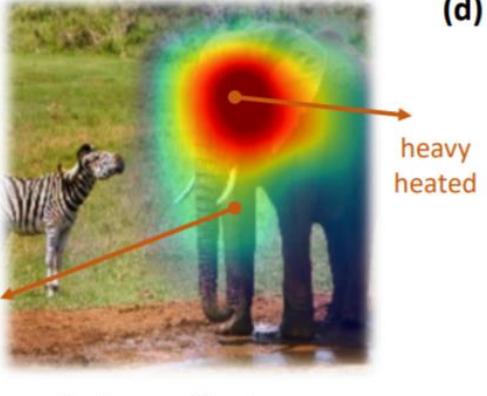
Convolutional Neural Networks (CNN)



Recurrent Neural Networks (RNN)

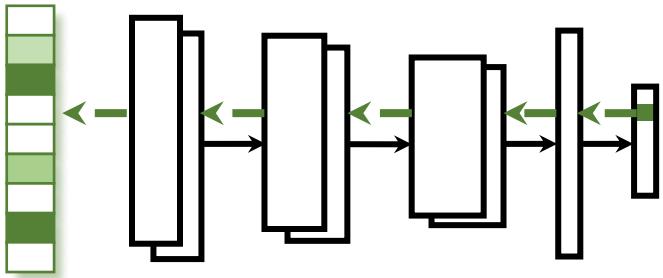
Bang, Seojin, et al. "Explaining a black-box by using a deep variational information bottleneck approach." AAAI, 2021.

Categorization

		Interpretation Scope	
		Global	Local
Intrinsic	 <p>(a) Decision Tree</p>	 <p>(b) Attention Mechanism</p>	
Posthoc	 <p>(c) Mimic Learning</p>	 <p>(d) Instance Heatmap</p>	

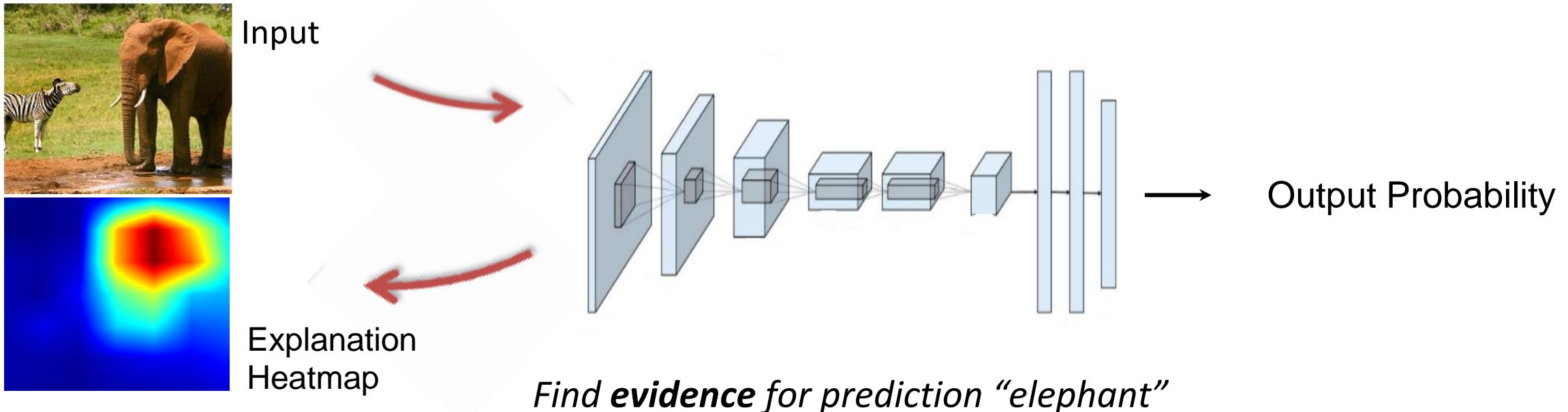
- ✓ Intrinsic - Global
 - *decision tree*
 - *rule base*
- ✓ Intrinsic - Local
 - *Attentional model*
- ✓ Posthoc - Global
 - *Mimic learning*
- ✓ Posthoc - Local
 - *Heatmap*
 - *Influential sample*

Post-hoc Local Explanation



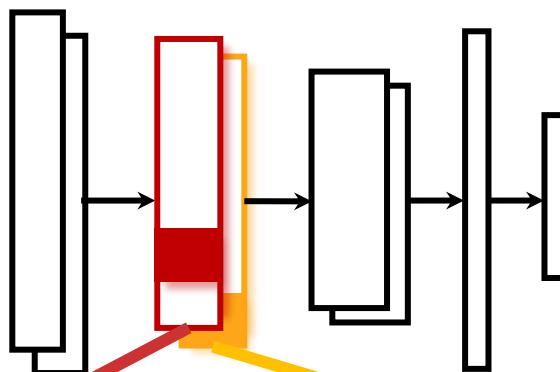
Post-hoc Interpretation

- Given an *input instance*
- A *pre-trained DNN*
- Contribution score for each feature in *input*



Post-hoc Global Explanation

Give a global understanding about what knowledge has been captured by a DNN model

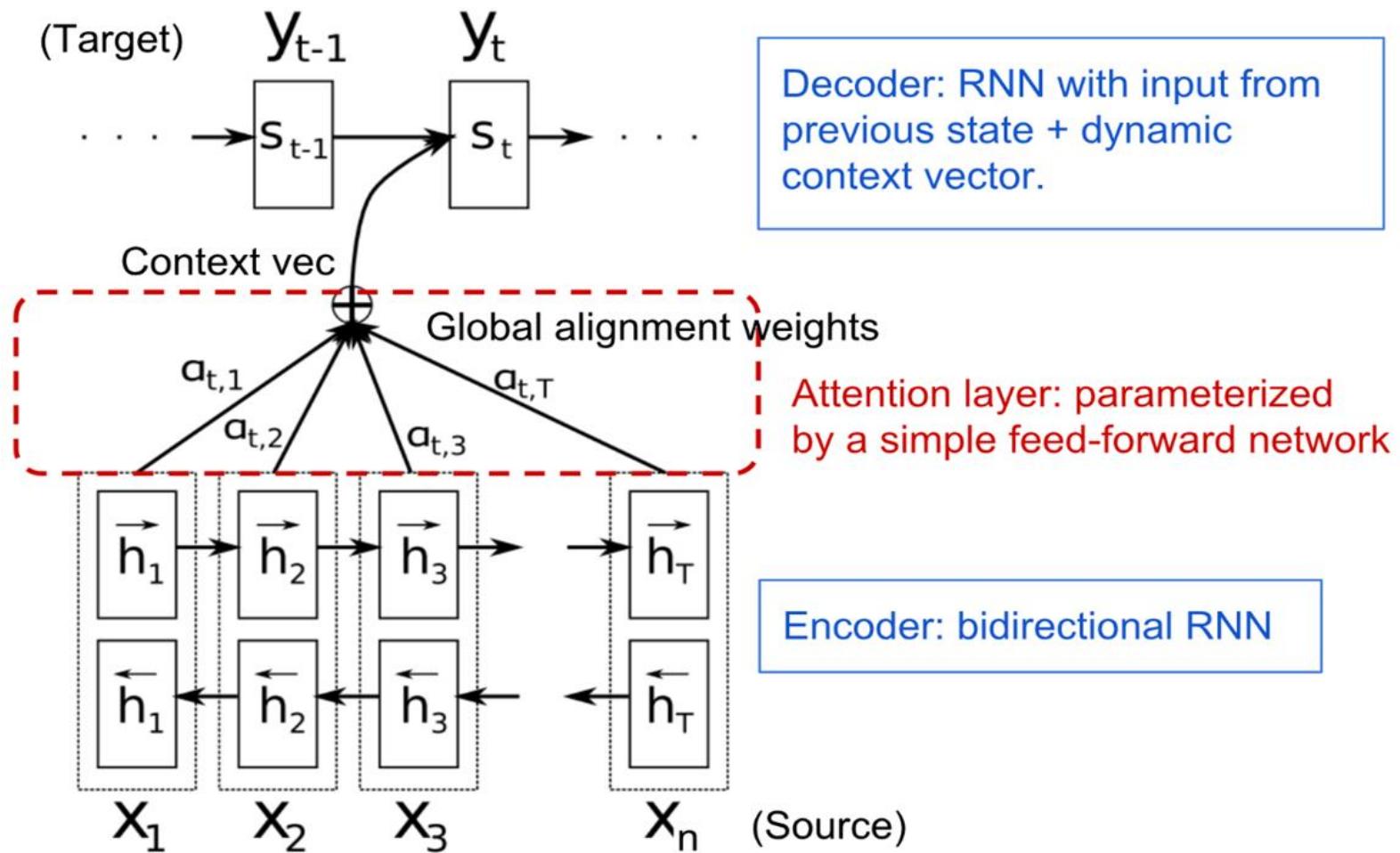


Activation Maximization

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} \mathbf{f}_l(\mathbf{x}) - \mathcal{R}(\mathbf{x})$$



Intrinsic Attentional Model



Intrinsic Interpretable Model (Local)

Design justifiable model architectures that can explain why a specific decision is made

Interpretation heatmap

by ent423 ,ent261 correspondent updated 9:49 pm et ,thu march 19, 2015 (ent261) a ent114 was killed in a parachute accident in ent45 ,ent85 ,near ent312 ,a ent119 official told ent261 on wednesday .he was identified thursday as special warfare operator 3rd class ent23 ,29 ,of ent187 ,ent265 .`` ent23 distinguished himself consistently throughout his career .he was the epitome of the quiet professional in all facets of his life ,and he leaves an inspiring legacy of natural tenacity and focused

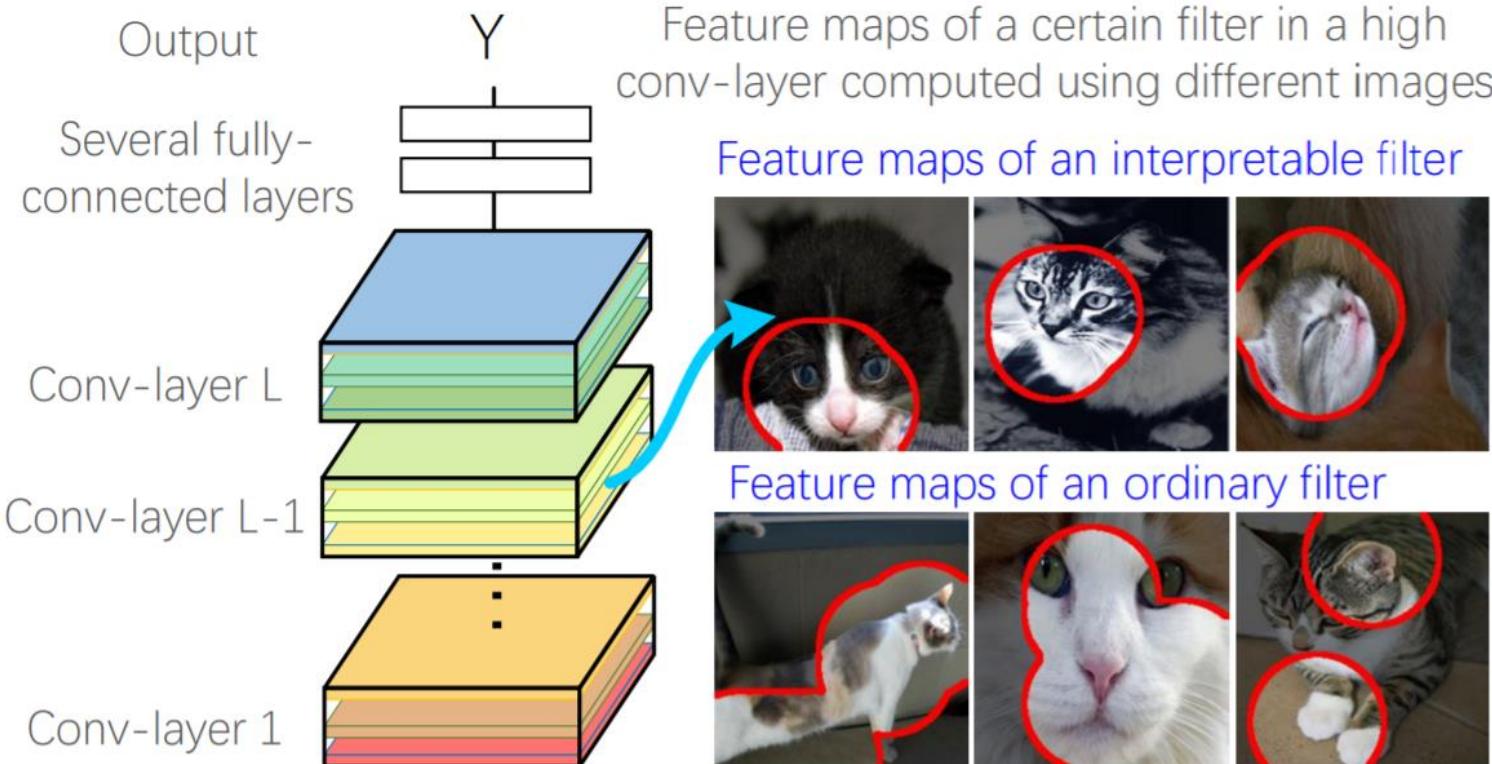
...

Interpretation Visualization

- Contribution score for each feature in input
- Deeper color indicates higher contribution

Intrinsic Interpretable Model (Global)

Globally interpretable models that offer a certain extent of working transparency



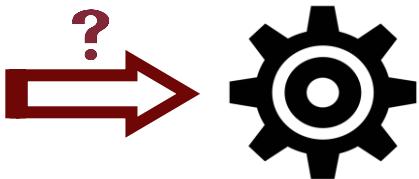
In interpretable CNN, each filter in high-layers represents a specific object part



Outline

1. Introduction to Interpretable Machine Learning
2. Interpretable Deep Learning
3. Evaluation of Interpretation

Evaluation Perspectives

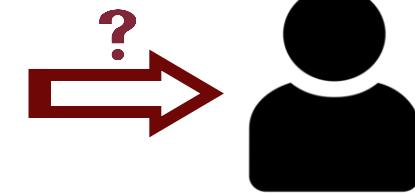


Are the generated explanations
faithful to the original model?

Fidelity



Ensure the explanations can
faithfully reflect the model



Are the generated explanations
friendly to the human users?

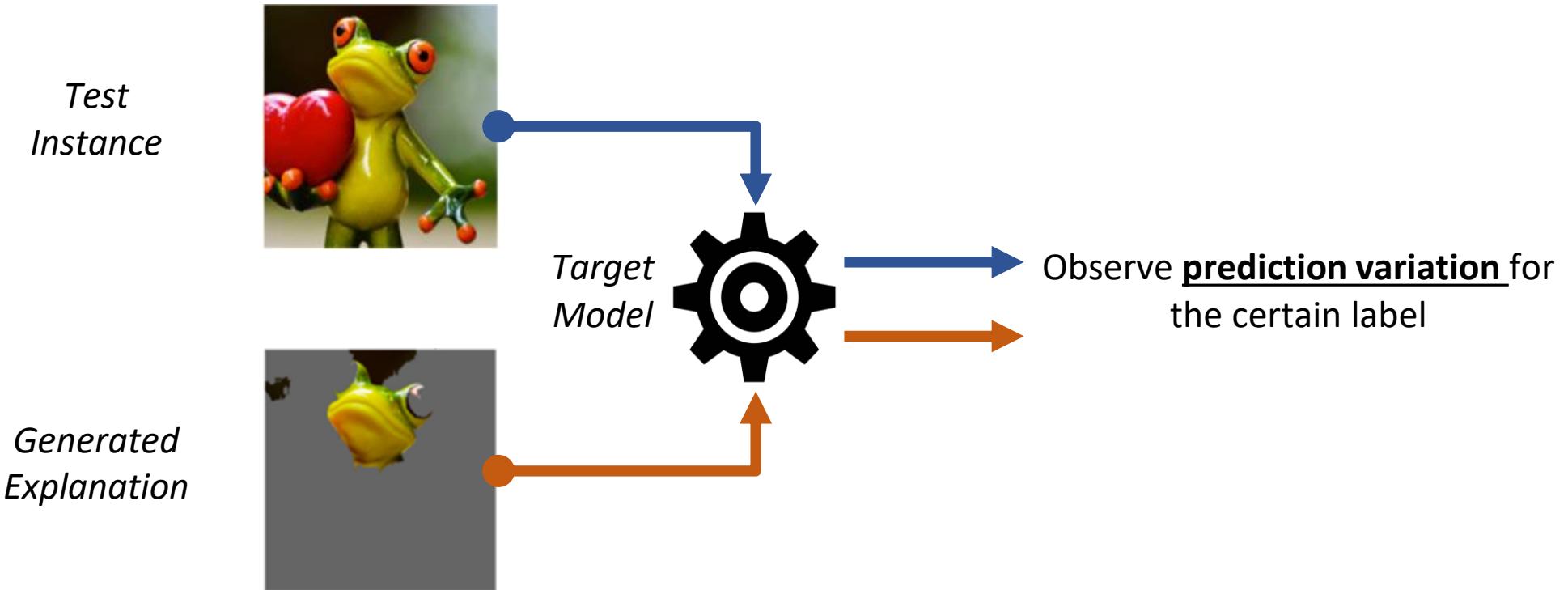
Persuasibility



Ensure the explanations can be
easily comprehended by humans

Philosophy of Fidelity Evaluation

Ablation Analysis



If the generated explanation is **faithful** to the target model, the **prediction variation** should be **small**.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." KDD, 2016.

Fidelity Evaluation Cases

Image Feature

flute: 0.9973

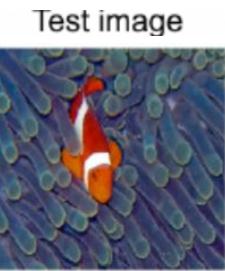


flute: 0.0007



Fong, Ruth C., et al. "Interpretable explanations of black boxes by meaningful perturbation." ICCV, 2017.

Training Data



RBF SVM



Inception

Koh, Pang Wei, et al. "Understanding black-box predictions via influence functions." ICML, 2017.

Text Feature

Positive (99.74%)

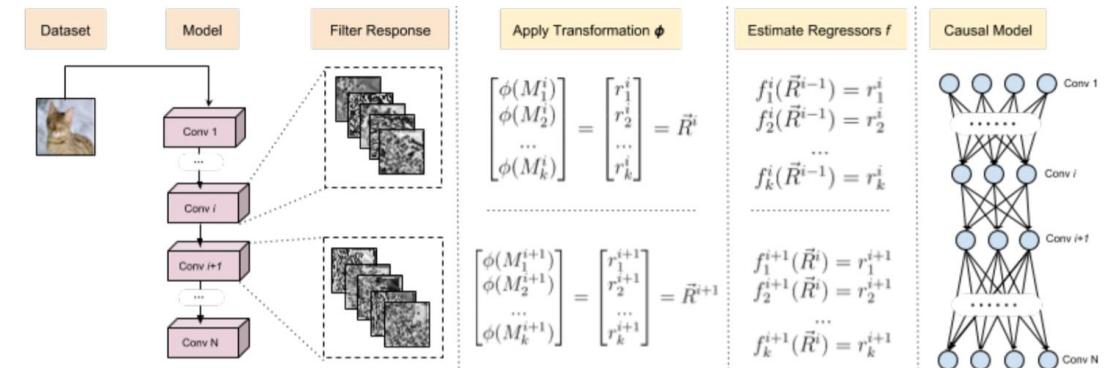
Occasionally melodramatic, it's also extremely effective.

Negative (99.00%)

Occasionally melodramatic, it's also terribly effective.

Du, Mengnan, et al. "On attribution of recurrent neural network predictions via additive decomposition." The WebConf, 2019.

Model Component

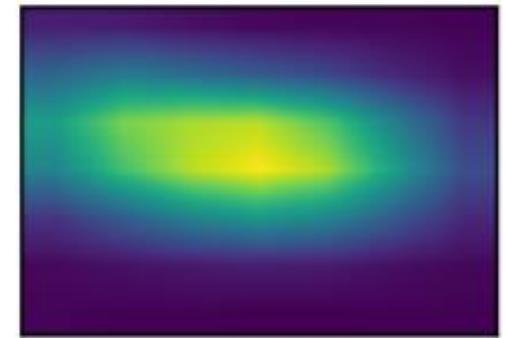
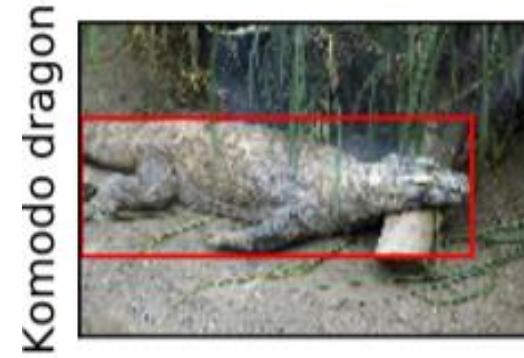
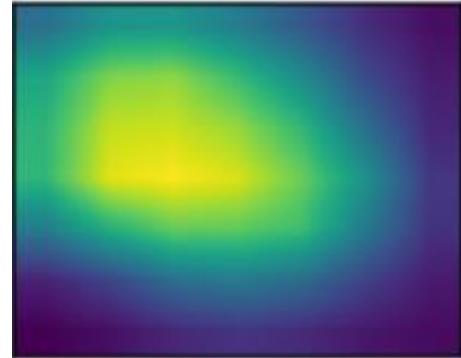


Narendra, Tanmayee, et al. "Explaining deep learning models using causal inference." arXiv, 2018.

Persuasibility with Image Bounding

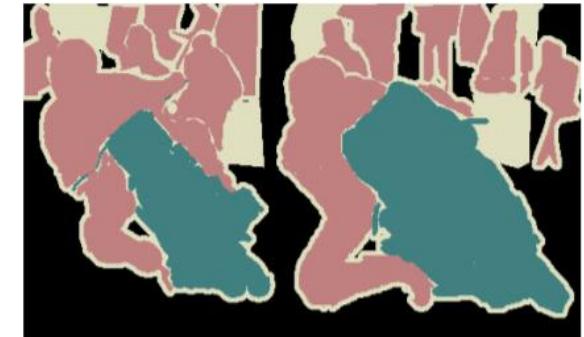
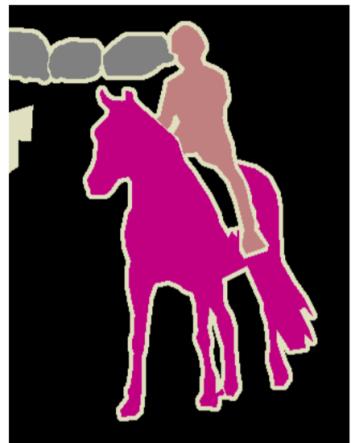
Evaluation with Bounding Box

Fong, Ruth C., et al. *Interpretable explanations of black boxes by meaningful perturbation.*” ICCV, 2017.



Evaluation with Semantic Segmentation

Long, Jonathan, et al. *“Fully convolutional networks for semantic segmentation.” CVPR, 2015.*



Persuasibility with Text Rationale

Evaluation with Text Annotation

Task: movie review

Label: negative

The movie is so badly put together that even the most casual viewer may notice the miserable pacing and stray plot threads.

Task: beer appearance

Label: positive

A beautiful beer, coal black with a thin brown head. Extremely powerful flavors, but everything is muted by the intense alcohol . the alcohol is so strong.

Persuasibility with User Study

Evaluation with Human-Computer Interaction (HCI)

The alien's preferences:

lazy or nervous → nodding
nodding and wearing glasses → clumsy
bubbly or clumsy → brave
faithful and cold or brave and passive → candy or dairy and fruit
sleepy or patient and obedient → spices and grains or dairy
brave and sleepy or patient or laughing → dairy and fruit or grains
crying or sleepy and faithful → grains and spices or fruit

Observations: patient, wearing glasses, lazy

Recommendation: milk, guava

Ingredients:

- Vegetables: okra, carrots, spinach
- Spices: turmeric, thyme, cinnamon
- Dairy: milk, butter, yogurt
- Fruit: mango, strawberry, guava
- Candy: chocolate, taffy, caramel
- Grains: bagel, rice, pasta



Mental Model ?

User Satisfaction ?

User Trust ?

Is the alien happy with the recommended meal?

- Yes
 No

Submit Answer

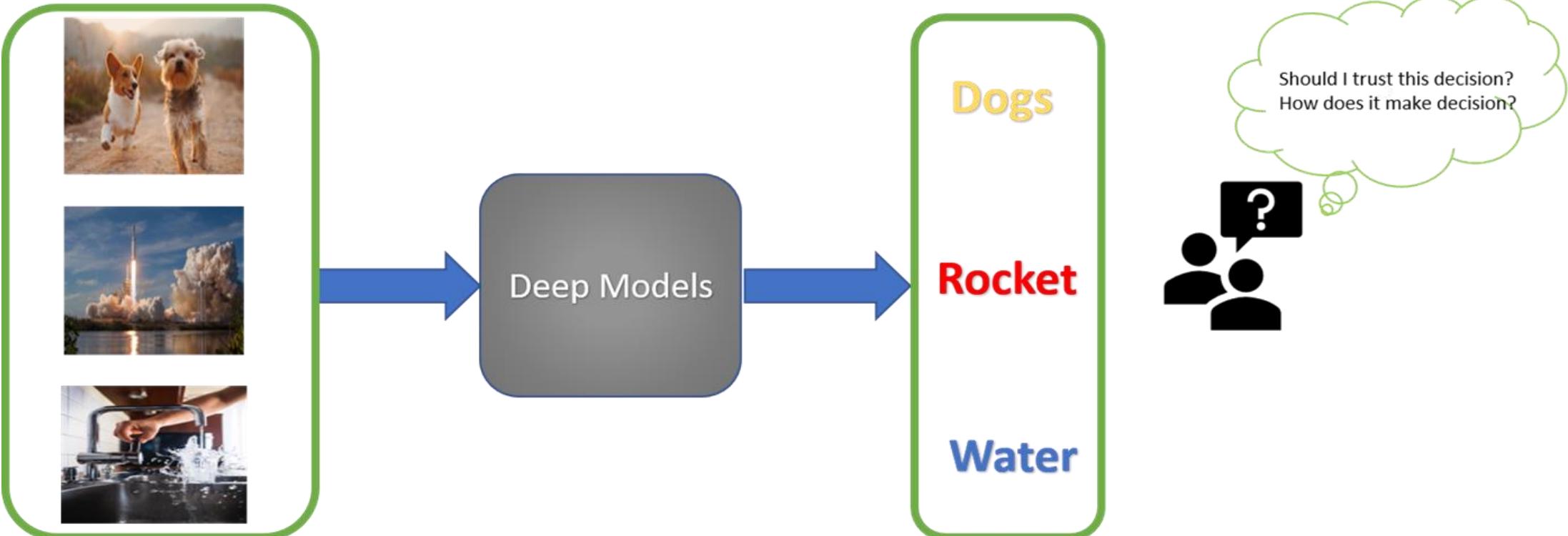
Lage, Isaac, et al. "An evaluation of the human-interpretability of explanation." arXiv preprint arXiv:1902.00006 (2019).

Part 2: Image-based Model Interpretation

Outline

1. Background: why we need image-based interpretation
2. Taxonomy of Interpretation:
 - *Model-specific vs model-agnostic*
 - *Global vs Local*
3. Saliency based methods
 - *Overview*
 - *Gradient, Integrated Gradient, SmoothGrad, CAM, Grad-CAM*
 - *Lime*
 - *Shap*
 - *LEG*
 - *Medical Applications*

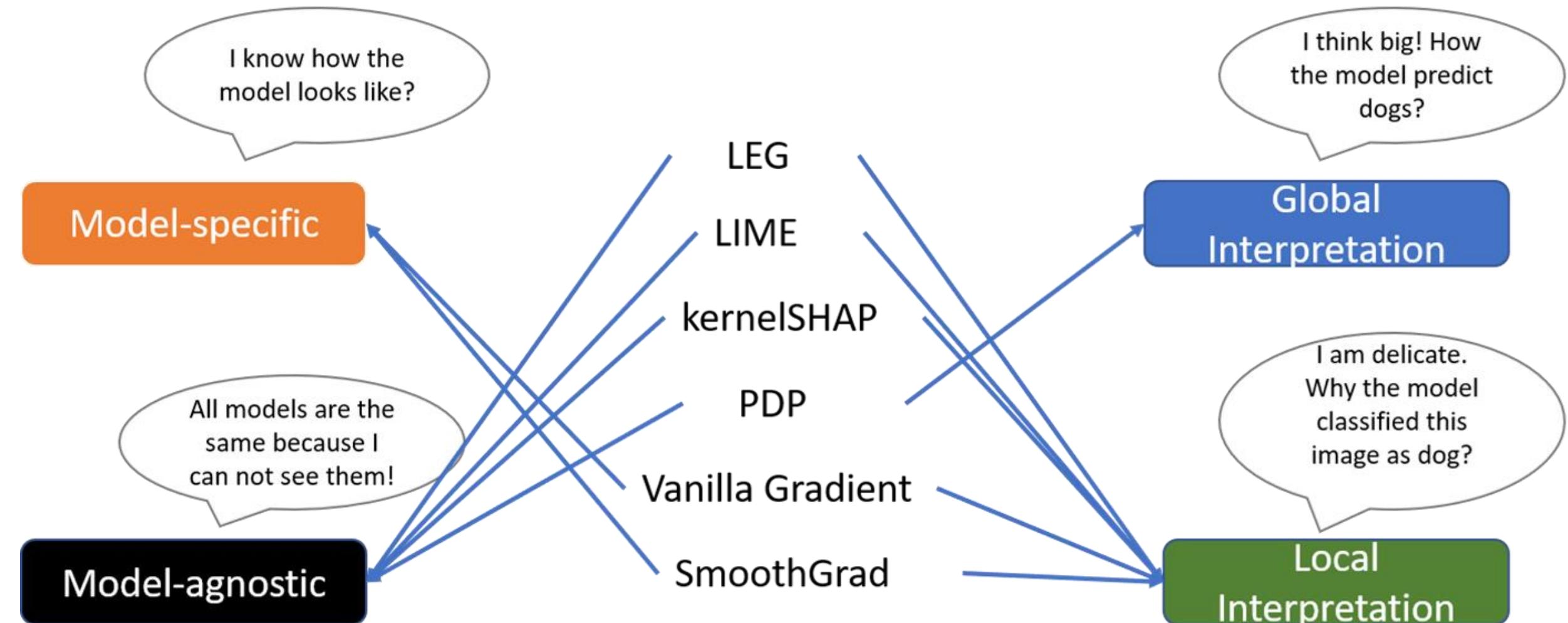
Why we need interpretation?



Outline

1. Background: why we need image-based interpretation
2. Taxonomy of Interpretation:
 - *Model-specific vs model-agnostic*
 - *Global vs Local*
3. Saliency based methods
 - *Overview*
 - *Gradient, Integrated Gradient, SmoothGrad, CAM, Grad-CAM*
 - *Lime*
 - *Shap*
 - *LEG*
 - *Medical Applications*

Categories of Interpretation:



Model-specific vs model-agnostic Interpretation

Model-specific interpretation

- Model-specific interpretation tools are limited to specific model classes. Tools that only work for the interpretation of e.g. neural networks are model-specific.

- Example:

Gradient methods. For a specific classification model S with input x , the classification result is:

$$\text{class}(x) = \operatorname{argmax}_{c \in C} S_c(x)$$

Then, we can use the derivative of $S(x)$ with respect to x to interpret this specific model S :

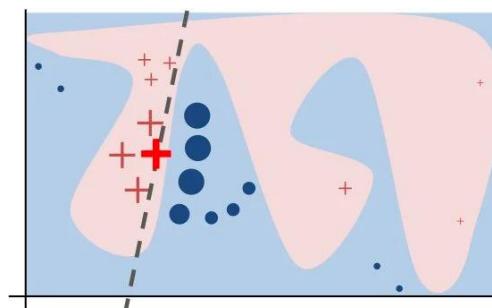
$$M_c(x) = \partial S_c(x) / \partial x$$

Model-specific vs model-agnostic Interpretation

Model-agnostic Interpretation

- Model-agnostic tools can be used on any machine learning model and are applied after the model has been trained.
- Example:
LIME. For any machine learning model denoted by f , we can locally approximate it with a simple, interpretable model g :

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \quad \mathcal{L}(f, g, \pi_x) + \Omega(g)$$



Global vs Local Interpretation

Global interpretation

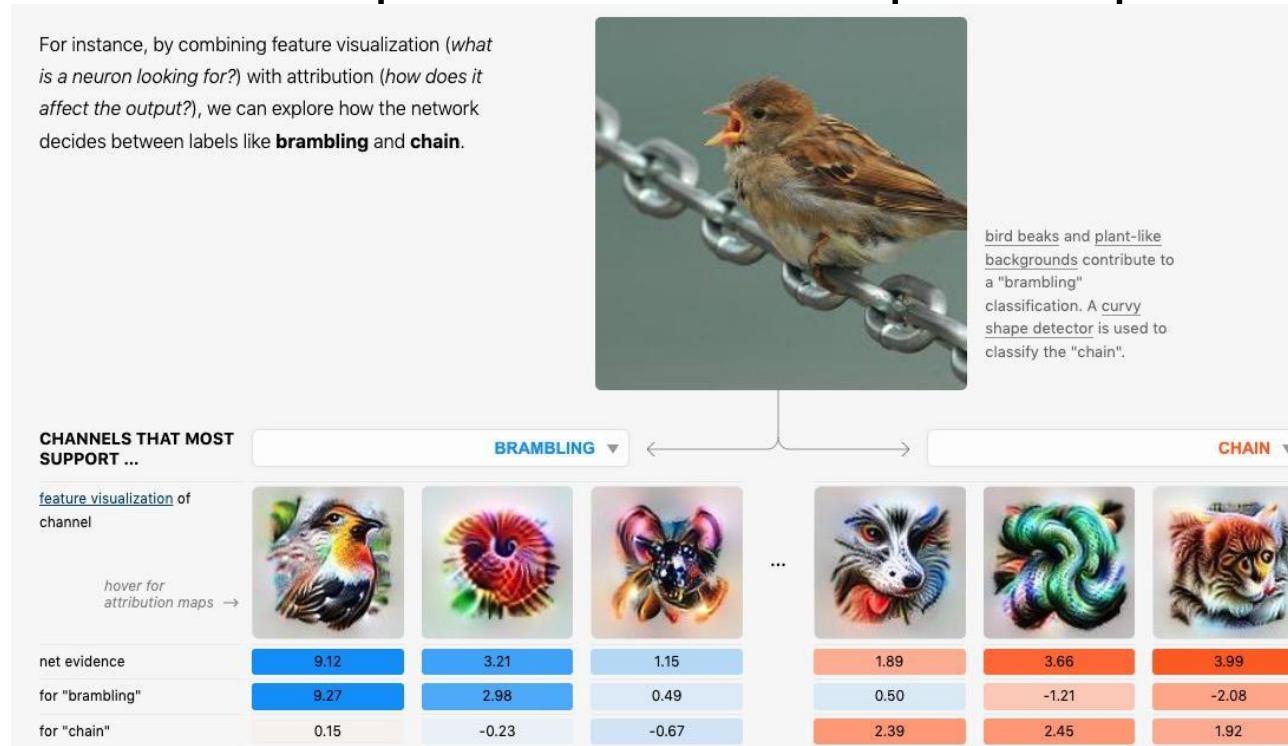
- Global interpretation explains the entire model behavior: for a given black-box model $f(x)$, we can find a simple and interpretable function $g(x)$, such that $g(x) \approx f(x)$.
- Example 1:
If we want to explain a random forest (black-box) model, we can build a linear regression model and obtain the coefficients of each predictor.

Global vs Local Interpretation

Global interpretation

- Example 2:

Checking the utilization of parameters to interpret deep learning CNN models

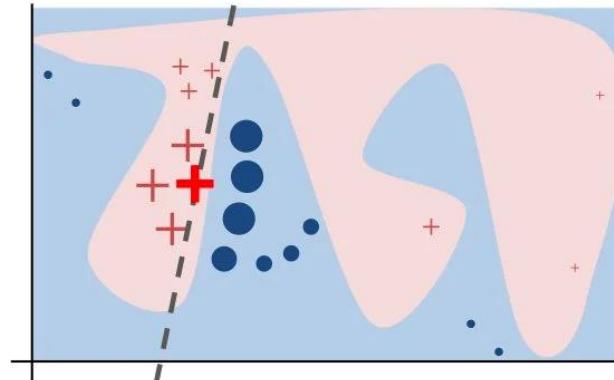


The Building Blocks of Interpretability. (<https://distill.pub/2018/building-blocks/>)

Global vs Local Interpretation

Local interpretation

- Local interpretation explains an individual prediction and the effect of a specific feature value on the prediction.
- Example: Local Interpretable Model-Agnostic Explanations (LIME).
Lime trains a surrogate model by generating a new dataset from the data point of interest:



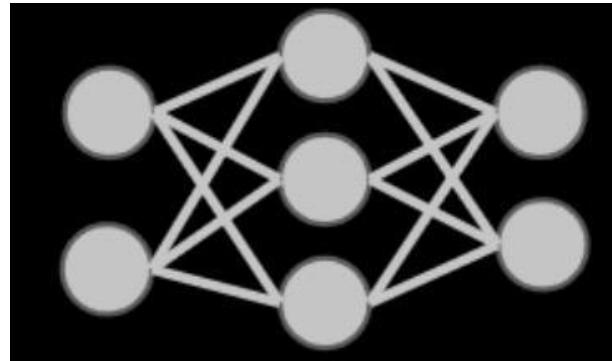
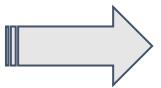
Outline

1. Background: why we need image-based interpretation
2. Taxonomy of Interpretation:
 - *Model-specific vs model-agnostic*
 - *Global vs Local*
3. Saliency based methods
 - *Overview*
 - *Gradient, Integrated Gradient, SmoothGrad, CAM, Grad-CAM*
 - *Lime*
 - *Shap*
 - *LEG*
 - *Medical Applications*

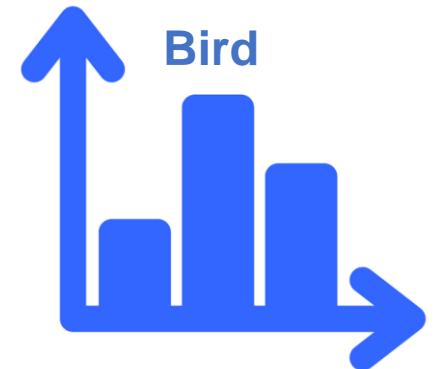
Saliency map overview



Bird



Black box model

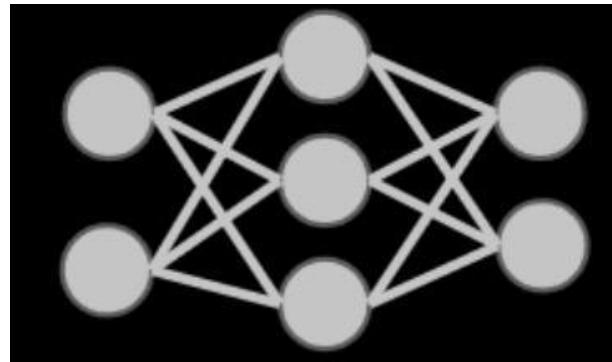
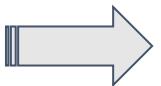


Prediction

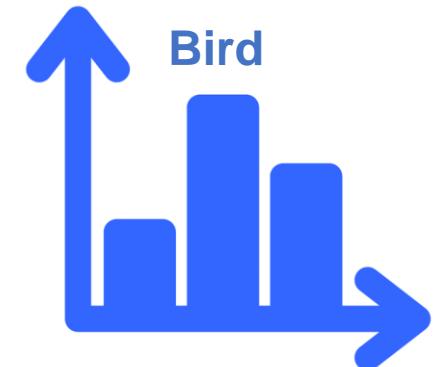
Saliency map overview



Bird



Black box model



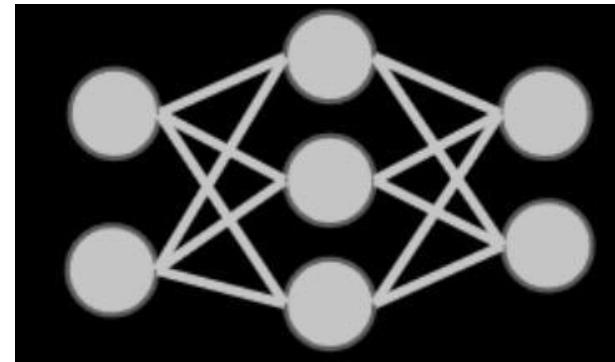
Prediction

A light blue speech bubble with a wavy bottom edge, containing the text "Why bird?".

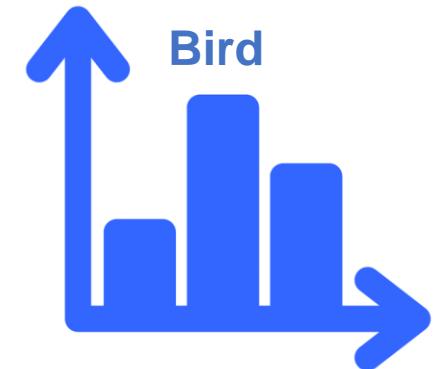
Saliency map overview



Bird

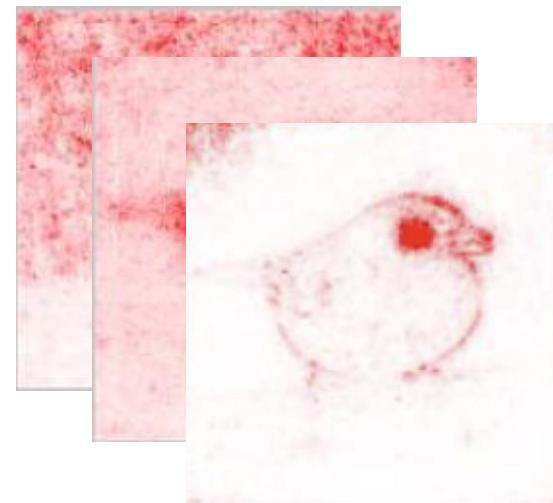


Black box model

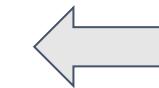


Prediction

- Heatmap based visualization
- Need differentiable model in most cases
- Normally involve gradient

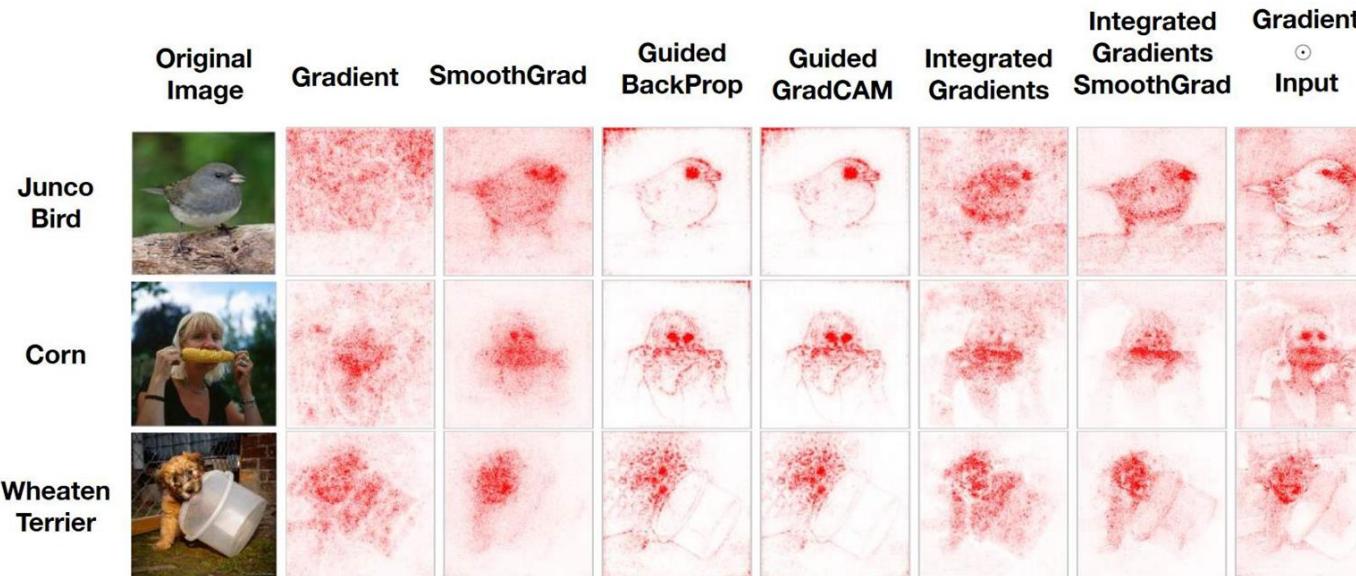


Saliency maps



A light blue speech bubble with a wavy bottom edge. Inside the bubble, the word "Why bird?" is written in black text.

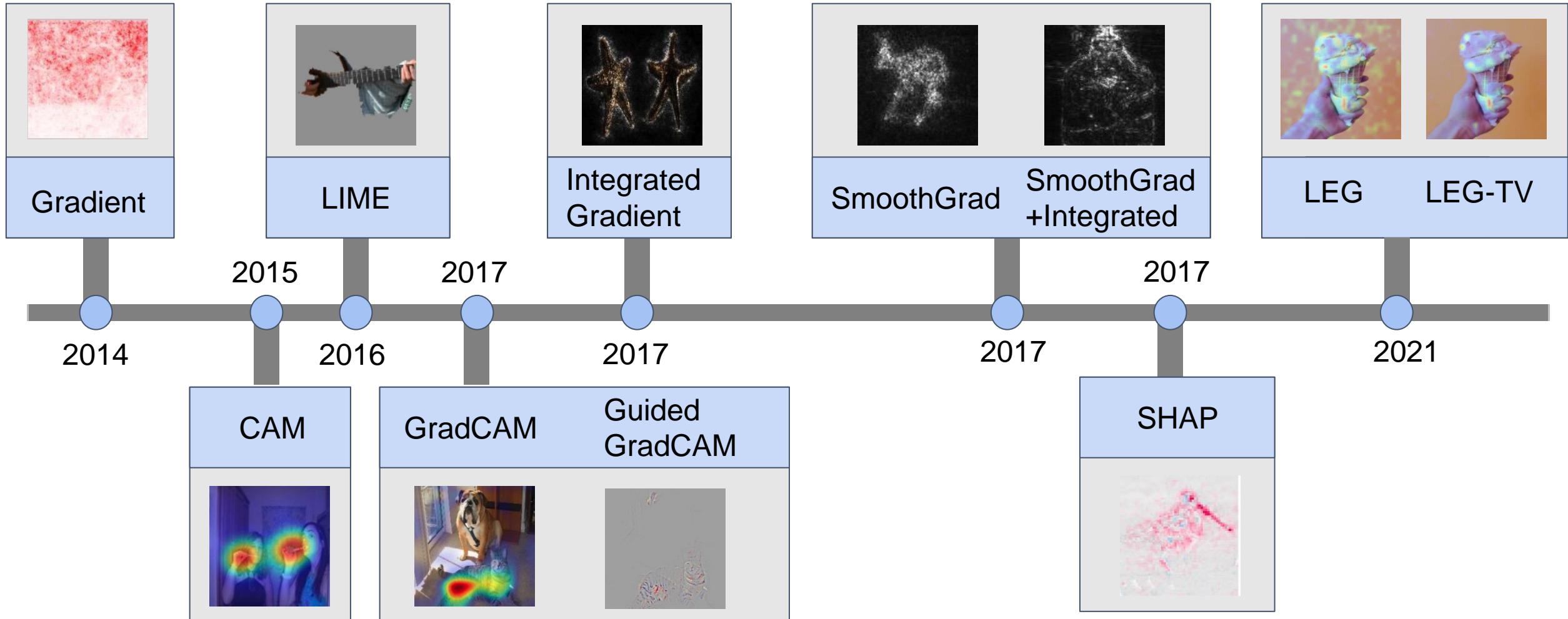
Saliency Example - Gradients



- Saliency refers to unique features (pixels, resolution) of the image in the context of visual processing.
- These unique features depict the visually alluring locations in an image.
- Saliency maps process images to differentiate visual features in images.

Adebayo, et al. "Sanity checks for saliency maps." *Advances in neural information processing systems*. 2018.

Saliency map overview



Outline

1. Background: why we need image-based interpretation
2. Taxonomy of Interpretation:
 - *Model-specific vs model-agnostic*
 - *Global vs Local*
3. Saliency based methods
 - *Overview*
 - *Gradient, Integrated Gradient, SmoothGrad, CAM, Grad-CAM*
 - *Lime*
 - *Shap*
 - *LEG*
 - *Medical Applications*

Gradients as sensitivity maps

Consider a system that classifies an image into one class from a set C.

$$\text{class}(x) = \operatorname{argmax}_{c \in C} S_c(x)$$

Locating “important” pixels by the derivative of $S(x)$ with respect to x:

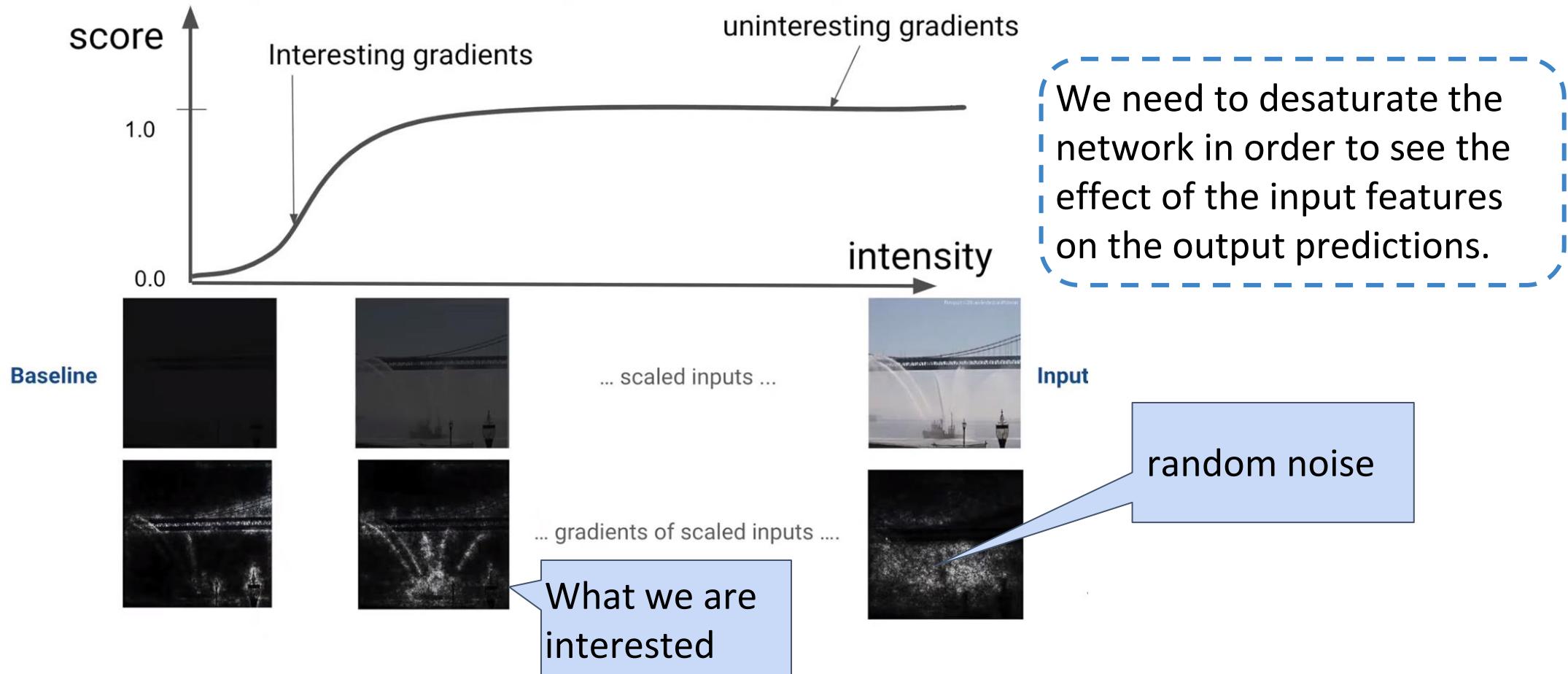


$$M_c(x) = \partial S_c(x) / \partial x$$

Adebayo, et al. "Sanity checks for saliency maps." *Advances in neural information processing systems*. 2018.

Integrated Gradient

Integrated Gradients combines the implementation invariance of gradients with the sensitivity



Integrated Gradient

Original image



Top label and score

Top label: reflex camera

Score: 0.993755

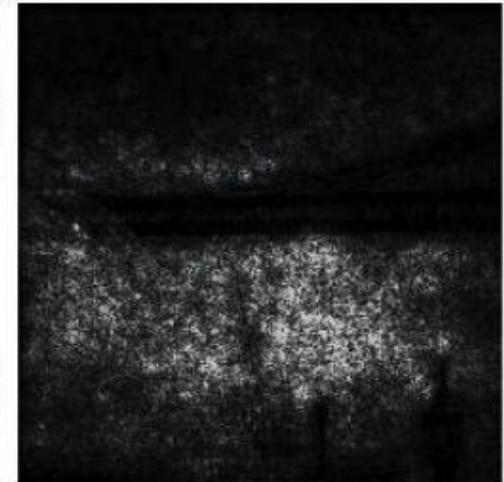
Top label: fireboat

Score: 0.999961

Integrated gradients

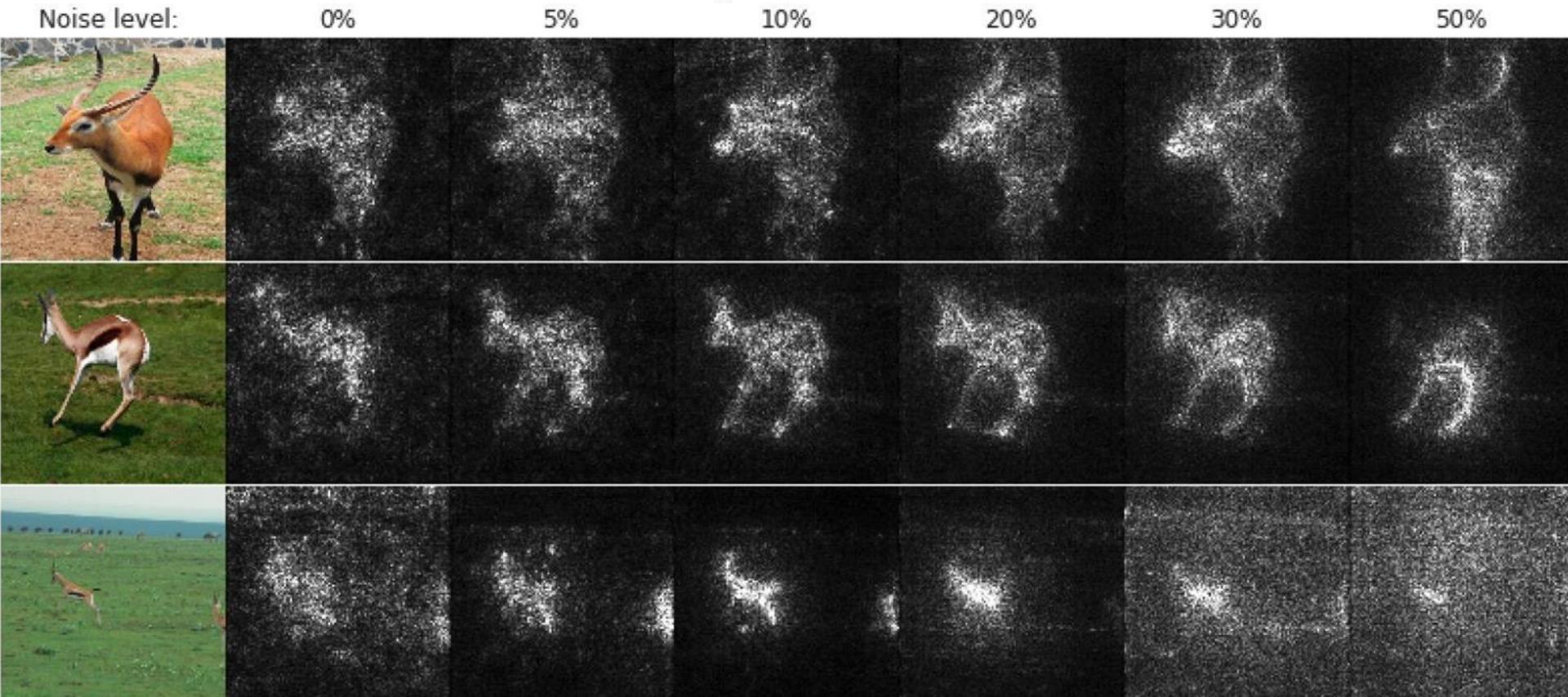


Gradients at image



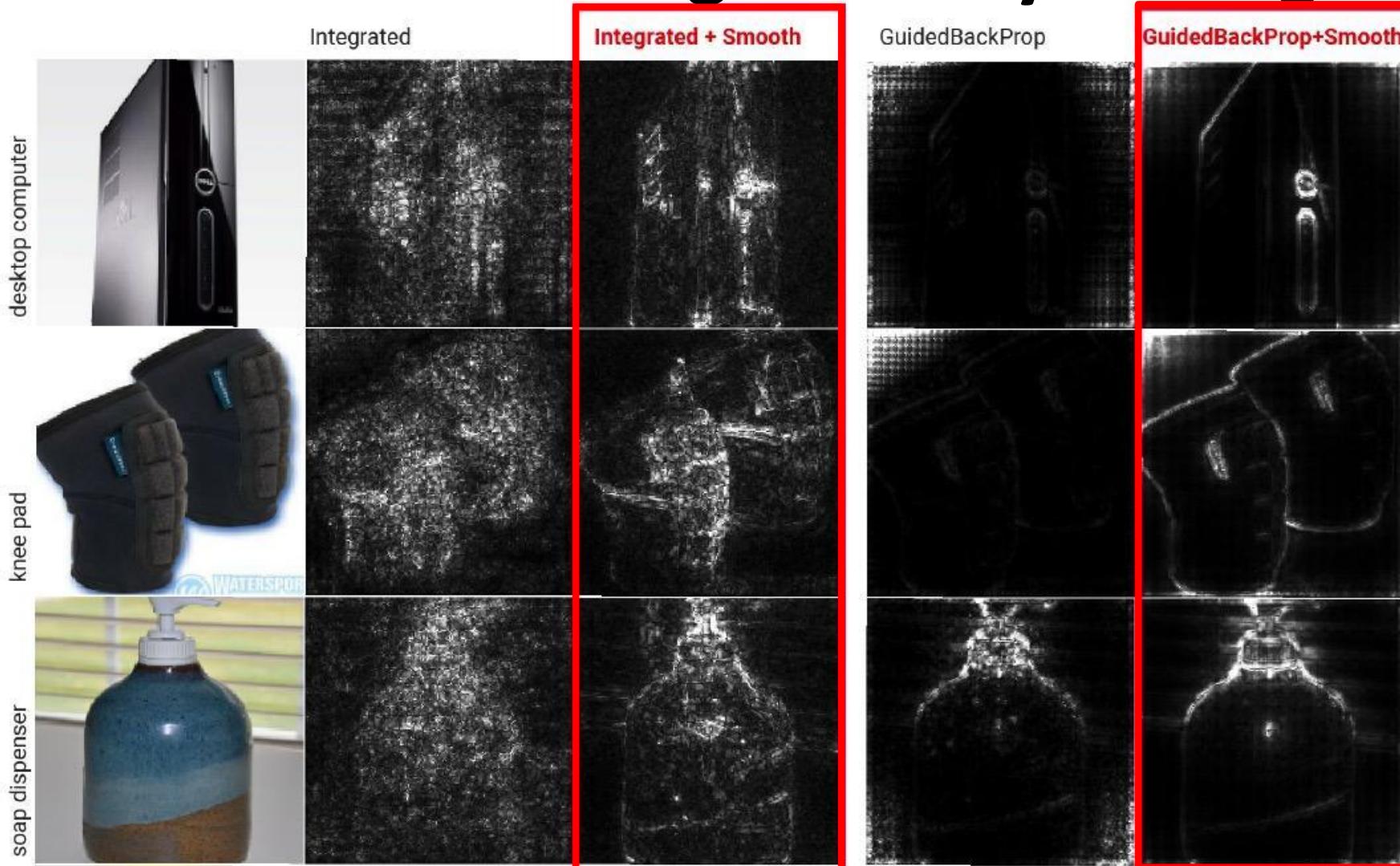
SmoothGrad: removing noise by adding noise

$$\hat{M}_c(x) = \frac{1}{n} \sum_1^n M_c(x + \mathcal{N}(0, \sigma^2))$$



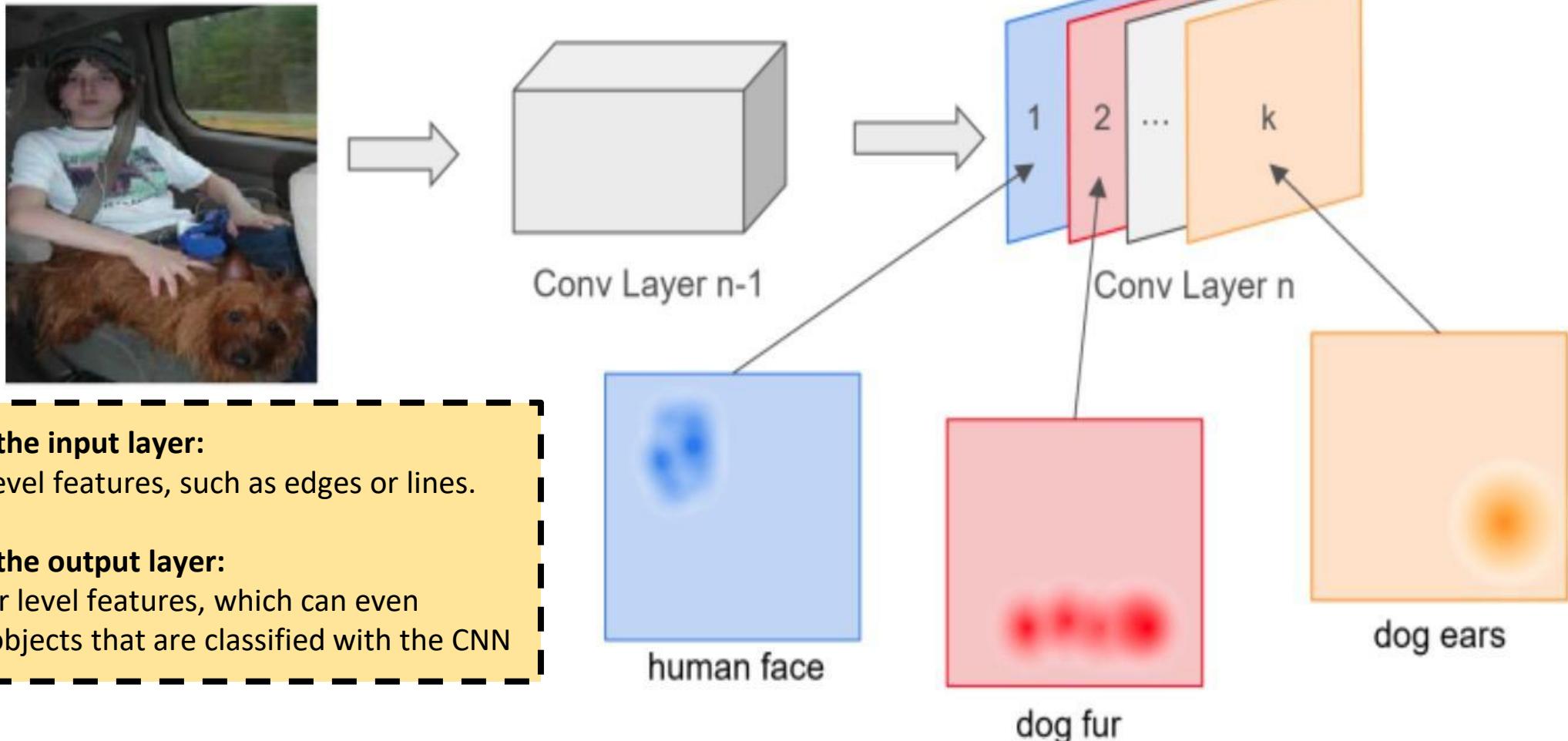
Smilkov, et al. "SmoothGrad: removing noise by adding noise." arXiv preprint arXiv:1706.03825. 2017.

SmoothGrad: removing noise by adding noise

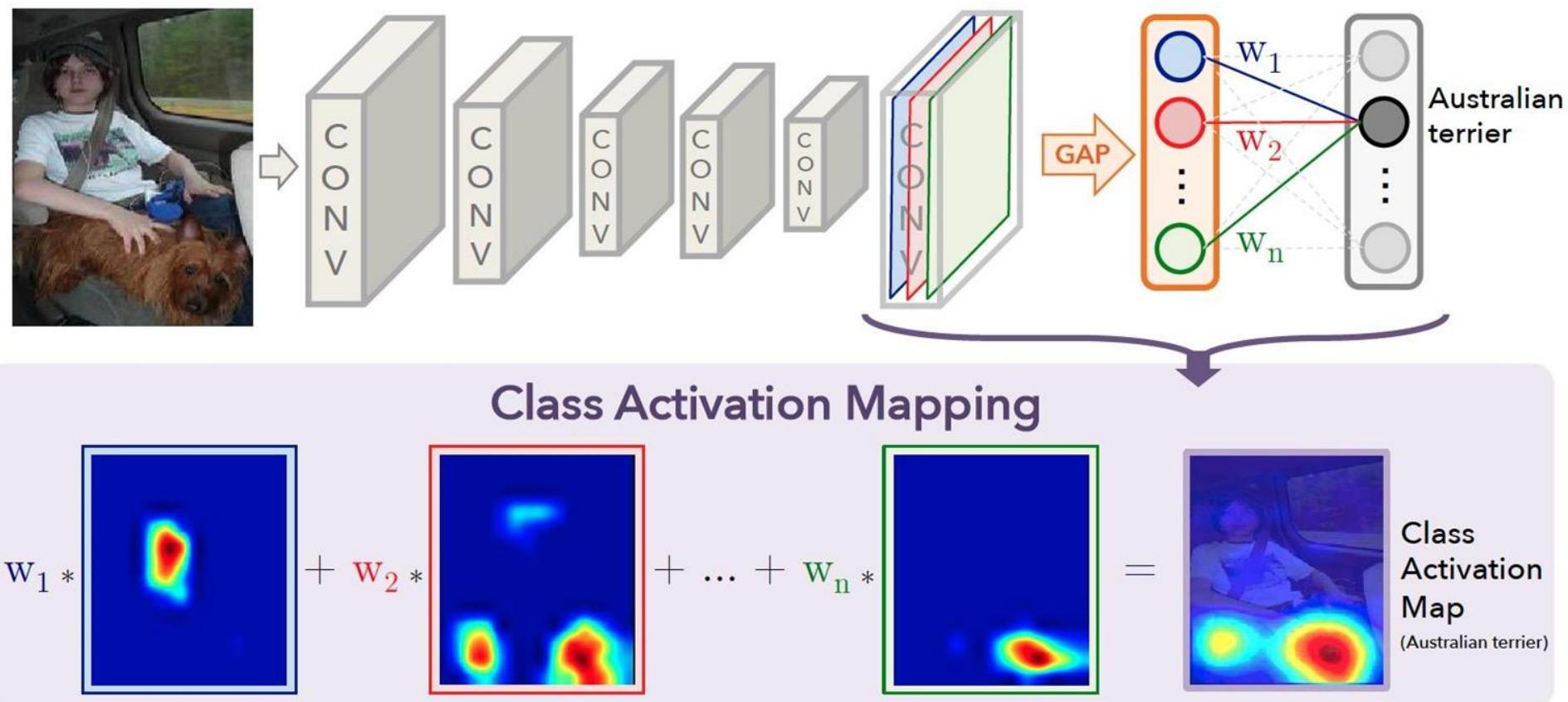


Using *SmoothGrad* in addition to existing gradient-based methods

Class Activation Mapping (CAM)



Class Activation Mapping (CAM)



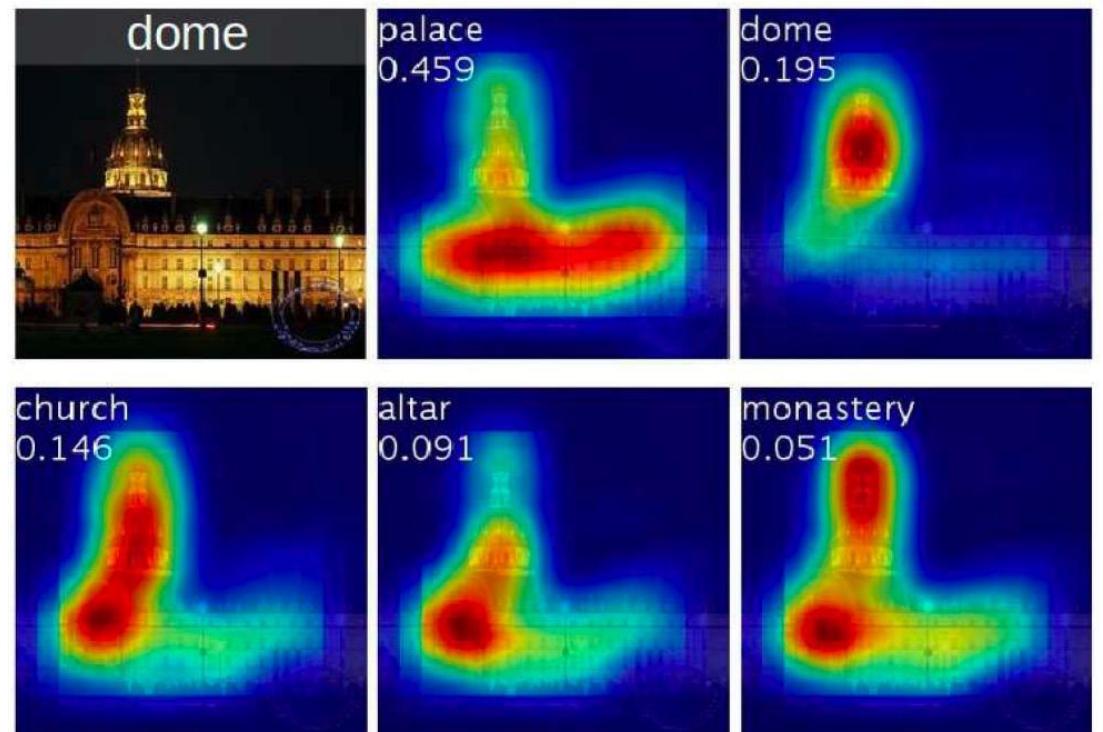
The predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

Class Activation Mapping (CAM)

CAMs for **one** object class



CAMs generated from
the **top 5** predicted categories



Gradient-weighted Class Activation Mapping (Grad-CAM)

Why Grad-CAM?

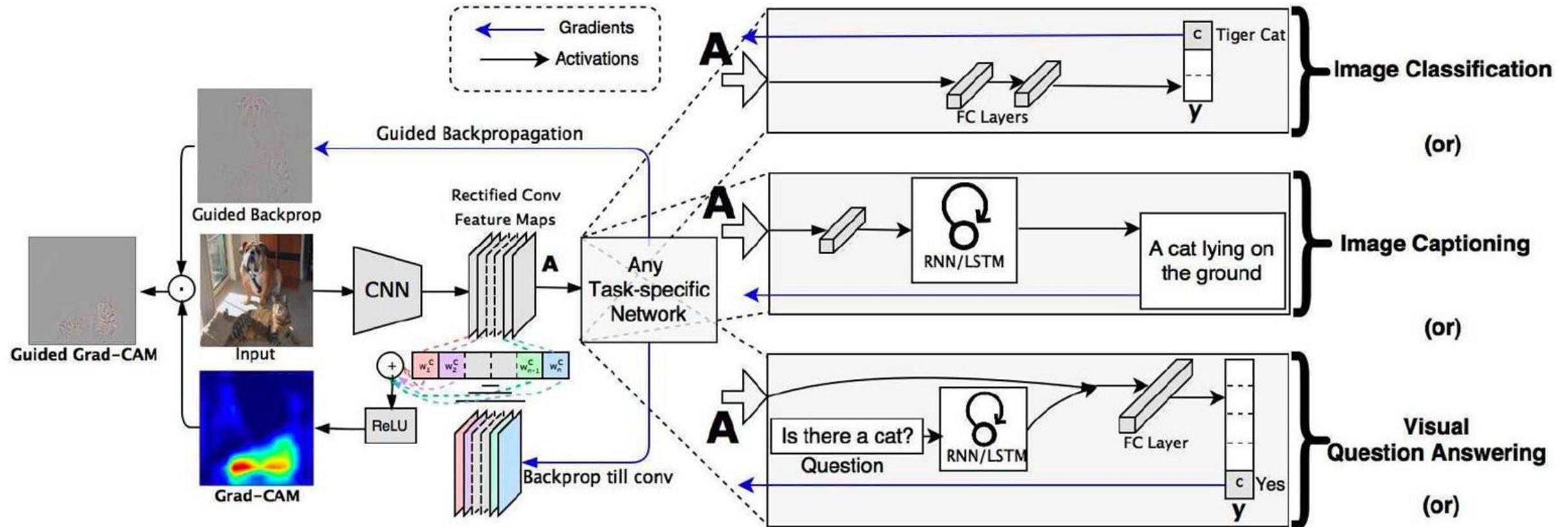
CAM can only be used by a restricted class of image classification CNNs which do not contain fully-connected layers.

Grad-CAM as a *generalization* to CAM

(For a fully-convolutional architecture, Grad-CAM reduces to CAM)

- ❑ Grad-CAM is applicable to a wide variety of CNN model-families:
 - ❑ CNNs with fully-connected layers (e.g. VGG),
 - ❑ CNNs used for structured outputs (e.g. captioning),
 - ❑ CNNs used in tasks with multi-modal inputs (e.g. visual question answering)

Gradient-weighted Class Activation Mapping (Grad-CAM)



Guided Grad-CAM:

Combine Grad-CAM with existing fine-grained visualizations to create a high-resolution class-discriminative visualization.

(Fuse Guided Backpropagation and Grad-CAM visualizations via pointwise multiplication)

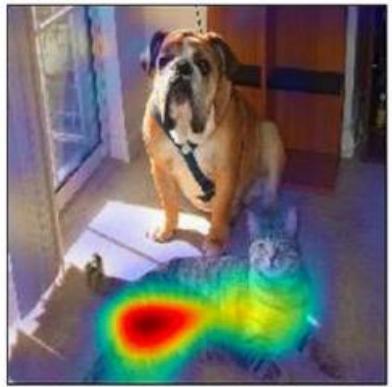
Gradient-weighted Class Activation Mapping (Grad-CAM)



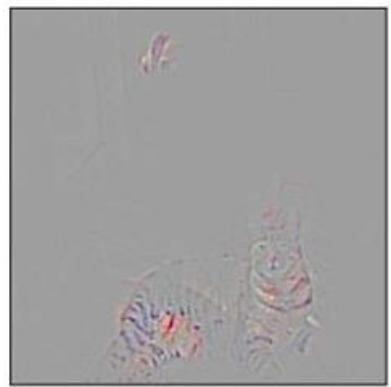
(a) Original Image



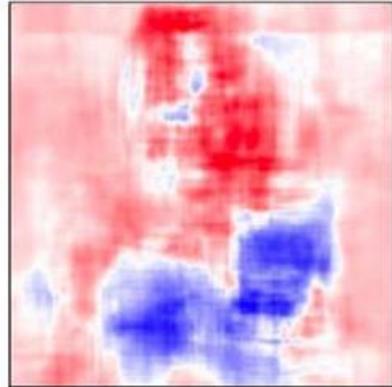
(b) Guided Backprop ‘Cat’



(c) Grad-CAM ‘Cat’



(d) Guided Grad-CAM ‘Cat’



(e) Occlusion map for ‘Cat’



(f) ResNet Grad-CAM ‘Cat’



(g) Original Image



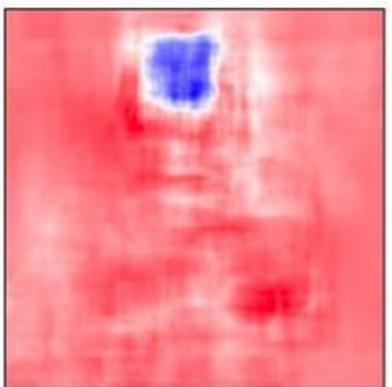
(h) Guided Backprop ‘Dog’



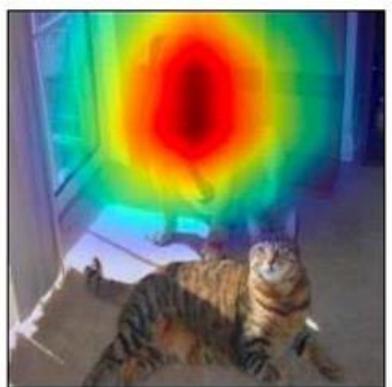
(i) Grad-CAM ‘Dog’



(j) Guided Grad-CAM ‘Dog’



(k) Occlusion map for ‘Dog’



(l) ResNet Grad-CAM ‘Dog’

Gradient-weighted Class Activation Mapping (Grad-CAM)

On other tasks:

- (I). Image captioning explanations
- (II). Visualizing QA model



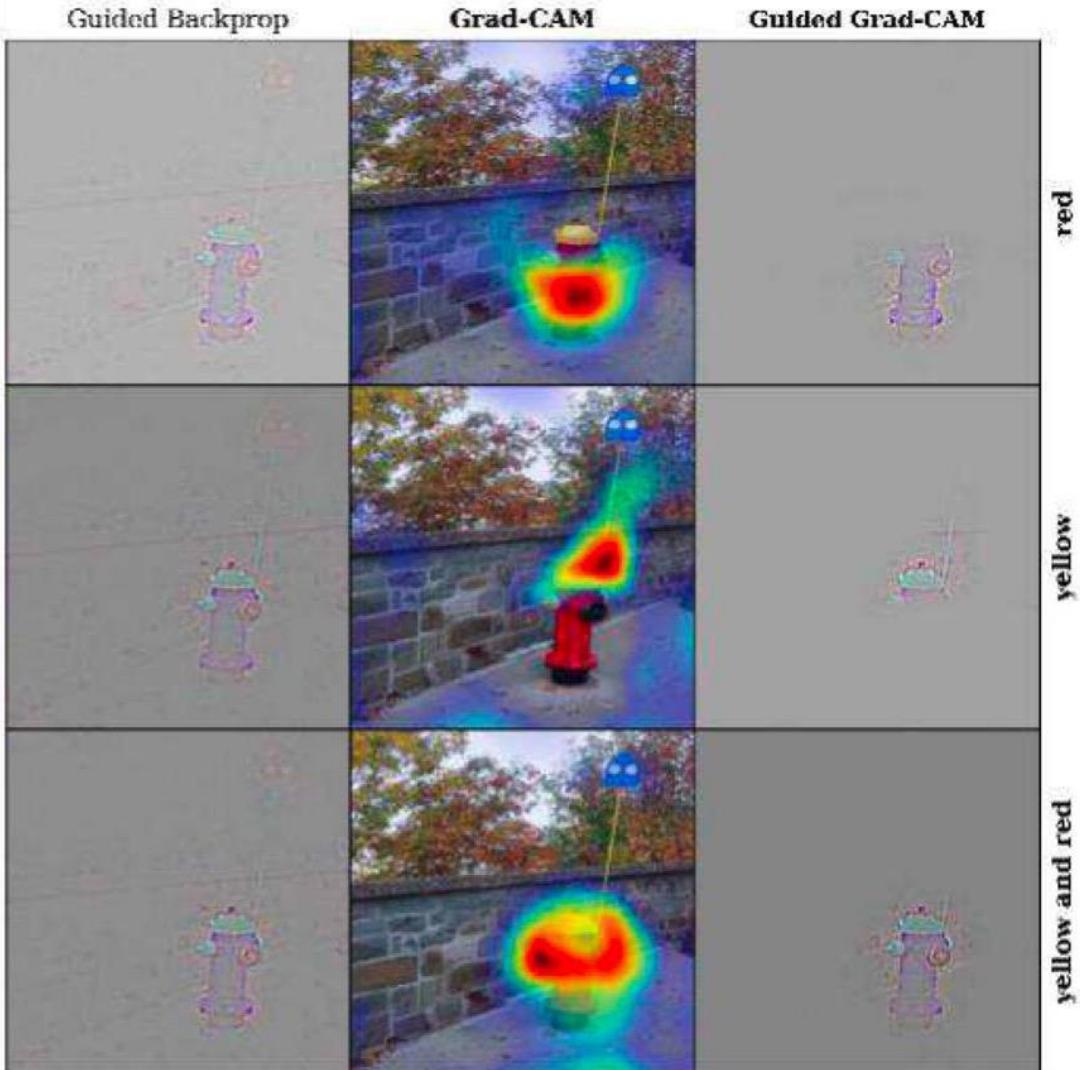
Gradient-weighted Class Activation Mapping (Grad-CAM)

On other tasks:

- (I). Image captioning explanations
- (II). Visualizing QA model



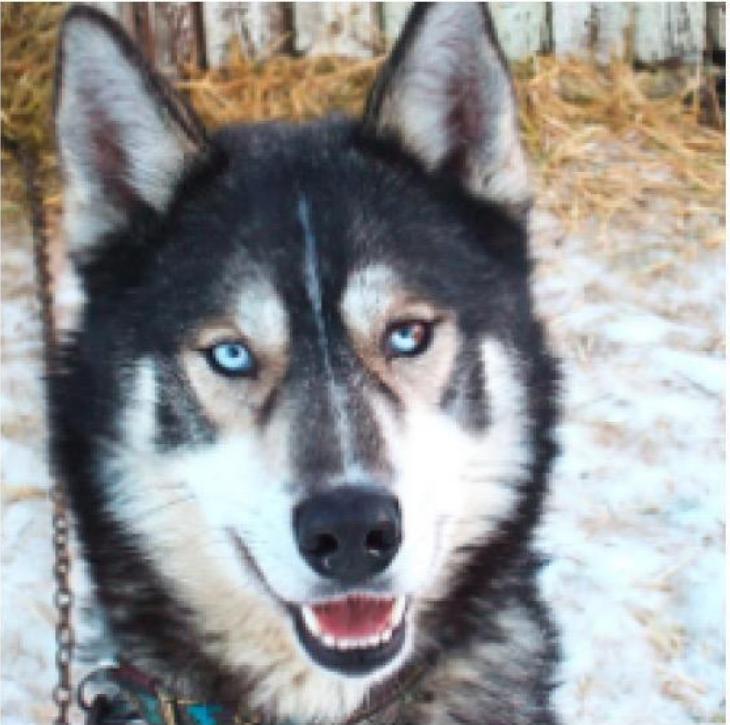
What color is the firehydrant?



Outline

1. Background: why we need image-based interpretation
2. Taxonomy of Interpretation:
 - *Model-specific vs model-agnostic*
 - *Global vs Local*
3. Saliency based methods
 - *Overview*
 - *Gradient, Integrated Gradient, SmoothGrad, CAM, Grad-CAM*
 - *Lime*
 - *Shap*
 - *LEG*
 - *Medical Applications*

Local Interpretable Model-agnostic Explanations (LIME)



(a) Husky classified as wolf



(b) Explanation

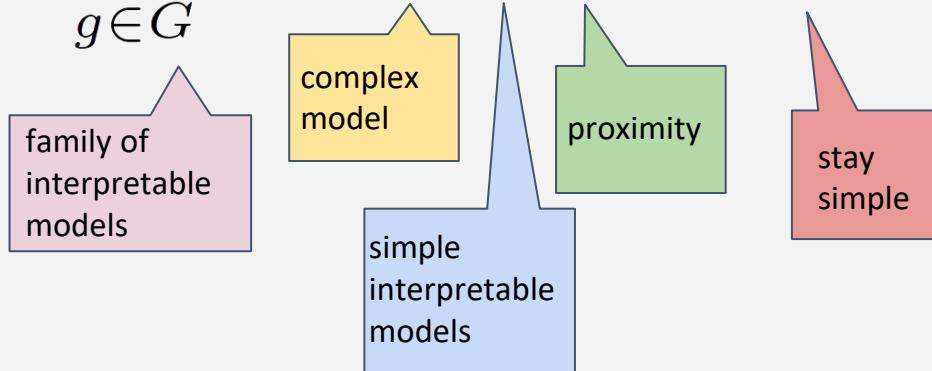
Why Should I Trust You?

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." KDD, 2016.

Local Interpretable Model-agnostic Explanations (LIME)

The explanation produced by LIME is obtained by the following:

$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$



Sparse Linear Explanations:

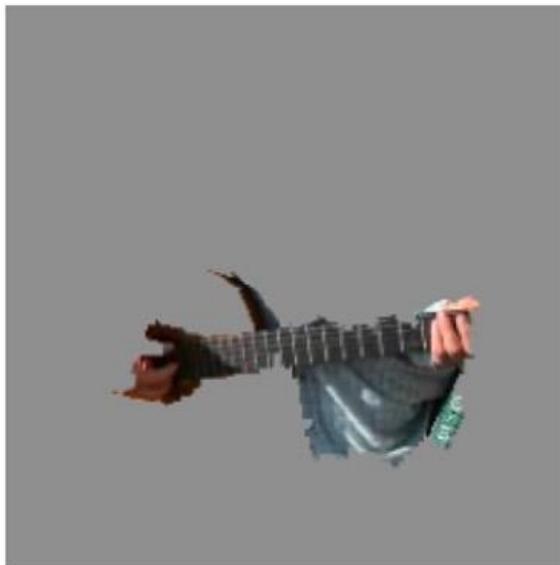
$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$$

Local Interpretable Model-agnostic Explanations (LIME)

Explaining an image classification prediction made by Google's Inception neural network.
The top 3 classes predicted are “Electric Guitar” ($p = 0:32$), “Acoustic guitar” ($p = 0:24$) and
“Labrador” ($p = 0:21$)



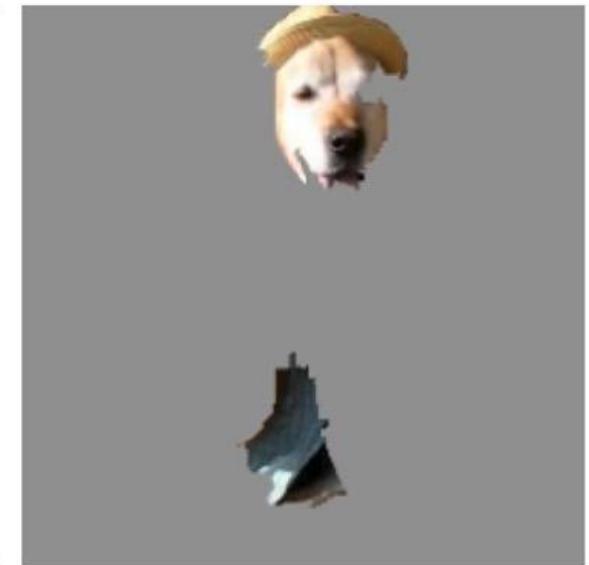
(a) Original Image



(b) Explaining *Electric guitar*



(c) Explaining *Acoustic guitar*



(d) Explaining *Labrador*

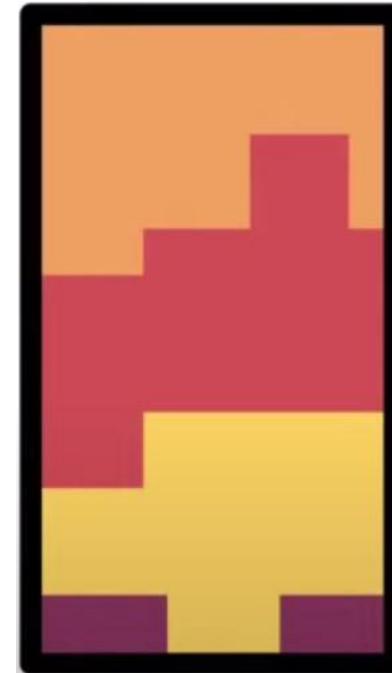
SHapley Additive exPlanations (SHAP)

Question:

If we have a coalition C that collaborates to produce a value V, how much did each individual member contribute to that final value?

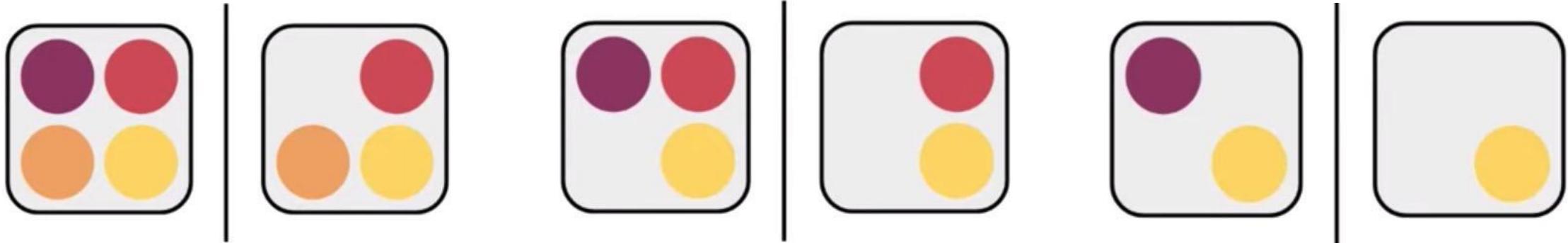


C



V

SHapley Additive exPlanations (SHAP)



contains
member 1

remove
member 1

contains
member 1

remove
member 1

contains
member 1

remove
member 1

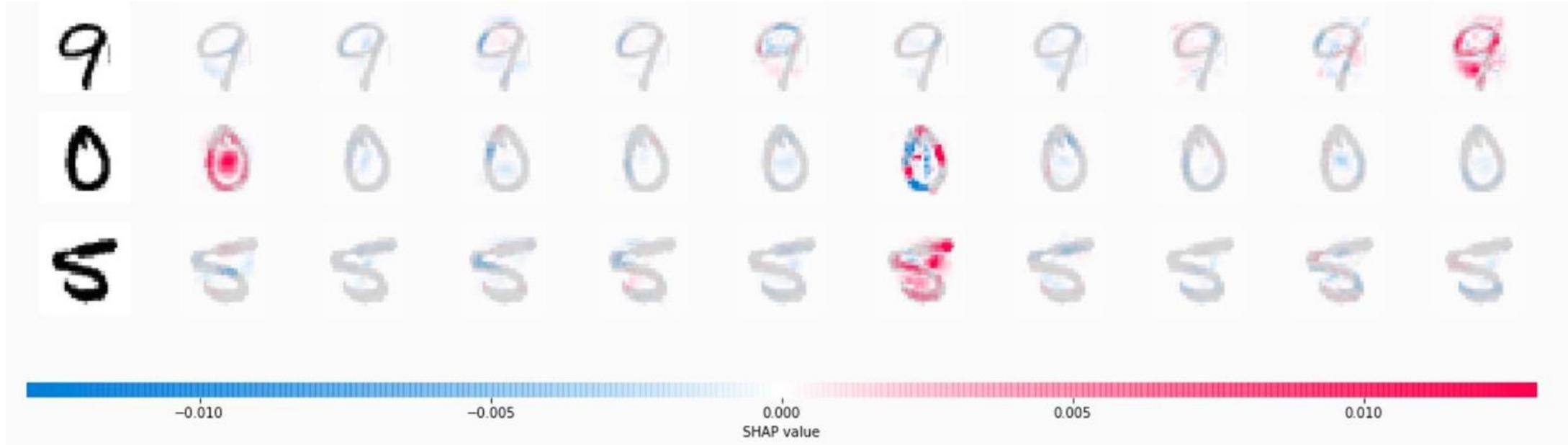
.....

Enumerate all the such coalition pairs, calculate the marginal contribution.

Then, *Shapley value*:

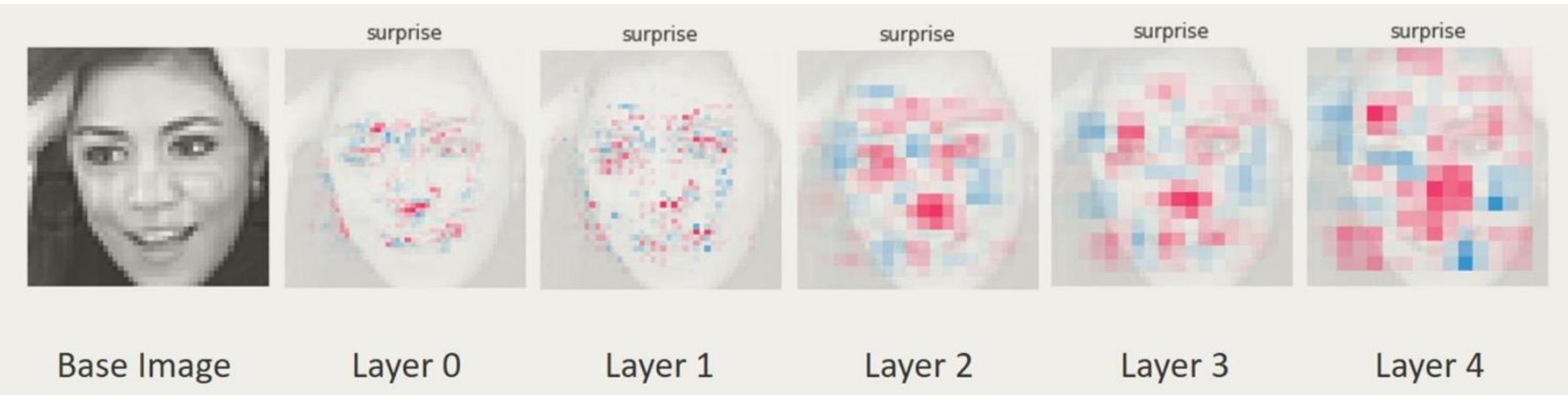
the average amount of contribution that a particular member makes to the coalition value

SHapley Additive exPlanations (SHAP)



The plot above shows the explanations for each class. Note that the explanations are ordered for the classes 0-9 going left to right along the rows.

SHapley Additive exPlanations (SHAP)



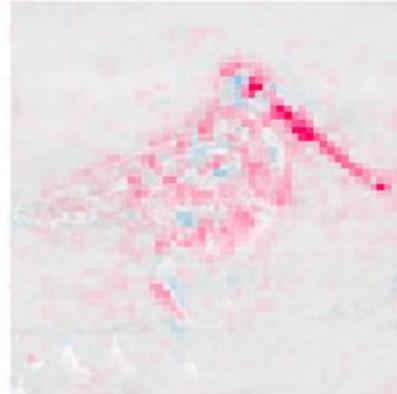
SHAP explains this image fascinatingly. Early layers focus on face features whereas the following layers mention areas in the face. Pixels pushing the prediction higher are shown in red whereas lower are shown in blue.

how SHAP can keep you from black box ai (<https://sefiks.com/2019/07/01/how-shap-can-keep-you-from-black-box-ai/>)

SHapley Additive exPlanations (SHAP)



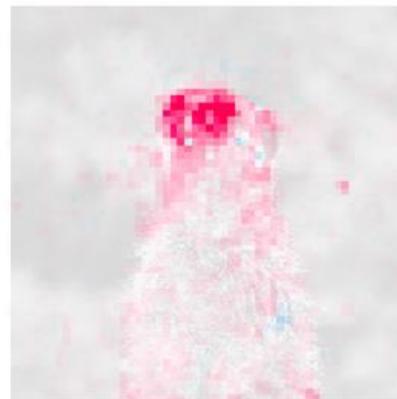
dowitcher



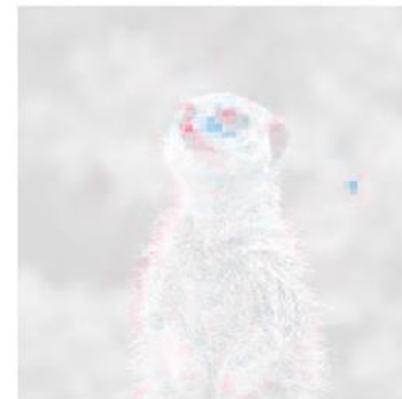
red-backed_sandpiper



meerkat



mongoose

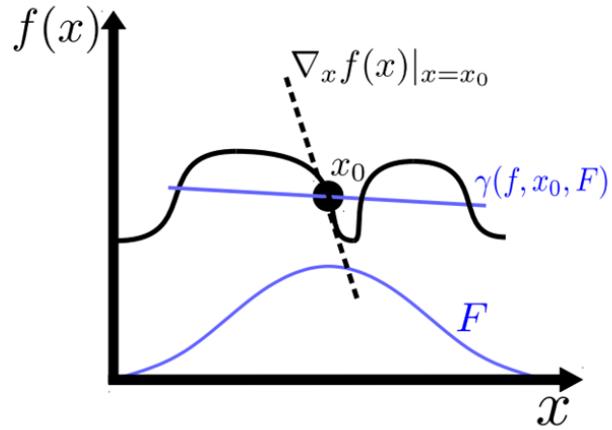


Outline

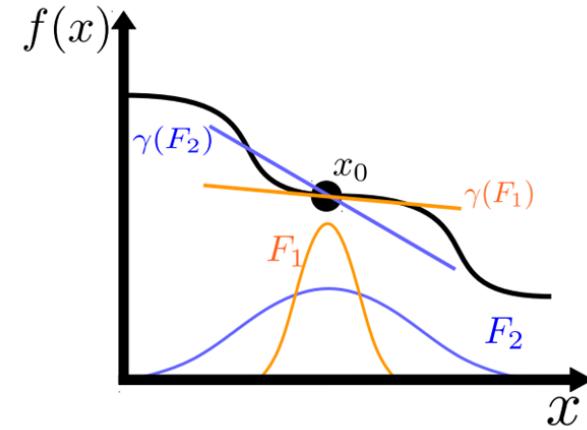
1. Background: why we need image-based interpretation
2. Taxonomy of Interpretation:
 - *Model-specific vs model-agnostic*
 - *Global vs Local*
3. Saliency based methods
 - *Overview*
 - *Gradient, Integrated Gradient, SmoothGrad, CAM, Grad-CAM*
 - *Lime*
 - *Shap*
 - ***LEG***
 - *Medical Applications*

Linearly Estimated Gradient (LEG)

LEG seeks to find a local linear approximation of $f(x)$ in a neighborhood around x_0 ; choice of the distribution, F , determines the size of the neighborhood.



(a) Gradient vs. LEG



(b) Effect of F on LEG

- * local gradient may be noisy and unstable
- * We can change the distribution to get local or global explanation

LEG-TV

- **Hypothesis:**
 - For interpretation of image classifiers, one expects that the saliency scores are located at a certain region, i.e., a contiguous body or a union of such bodies.
- A smart procedure would make use of this assumption.

- We propose the following procedure for estimating LEG. It can be obtained by solving a linear program.

$$\min_g \|Dg\|_1$$

subject to $\left\| \frac{1}{n} \sum_i f(\tilde{x}_i) \tilde{x}_i - \Sigma g \right\|_\infty \leq \lambda.$



Linearly Estimated Gradient (LEG)



- LEG estimates for ImageNet images classified by VGG-19.
- LEG-TV, compared to LEG, provides a more human readable estimate of local saliency.
- Both approaches select pixels that are critical for the label, such as nose and ear of golden retriever, bottom of cone and scoop of ice-cream.

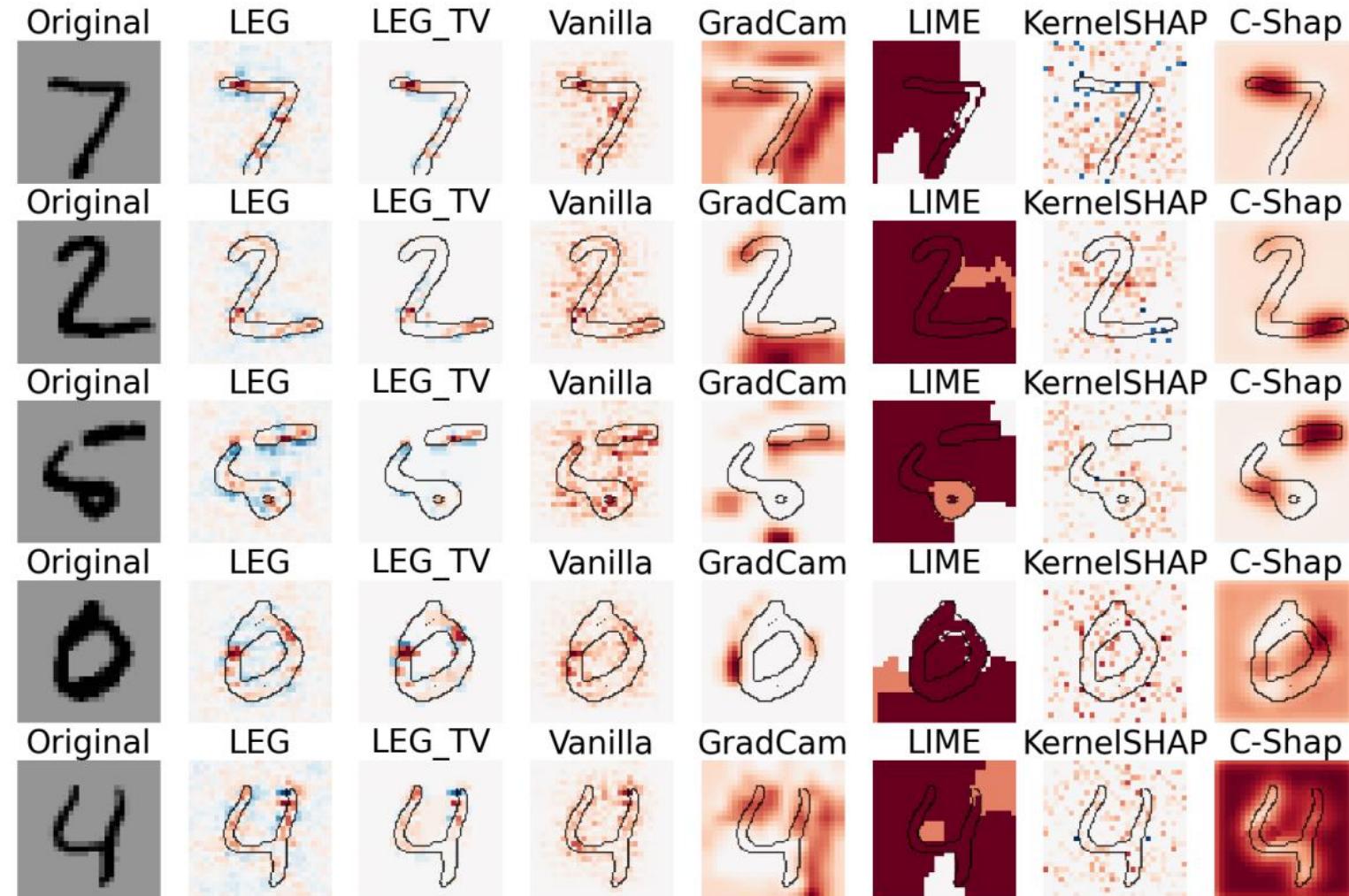


(a) Origin

(b) LEG

(c) LEG-TV

Examples of Explanations on MNIST



Linearly Estimated Gradient (LEG)

Experiment - Sensitive analysis

Faithful to the local gradient

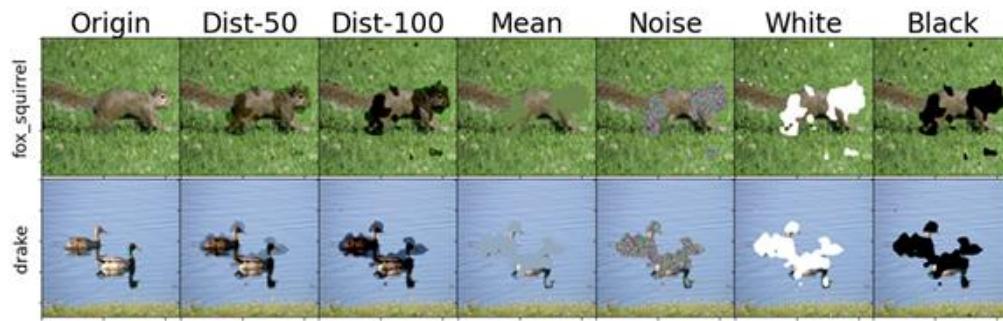


Figure 5: Examples of LEG-TV estimates shown by different masking techniques with 10% masked.

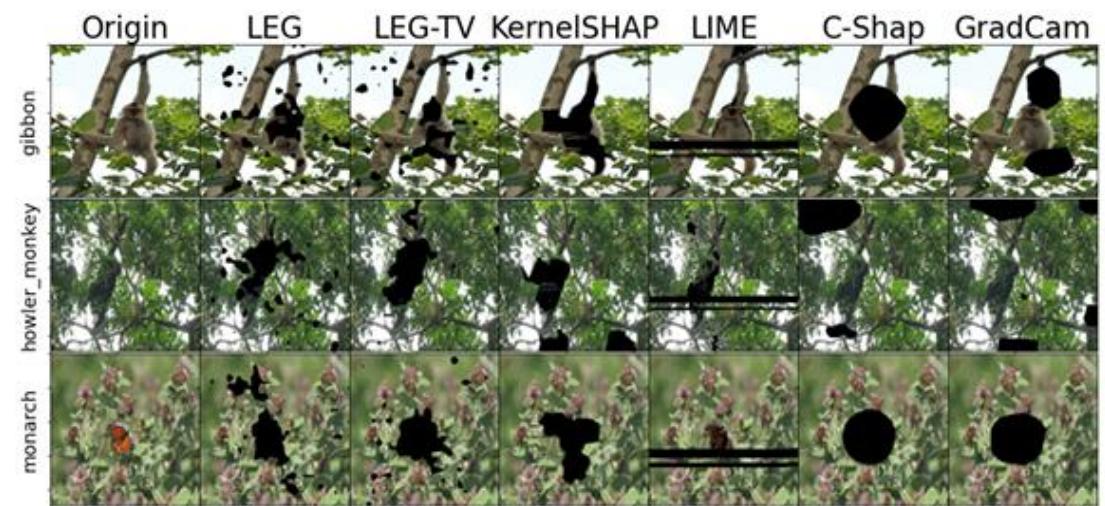
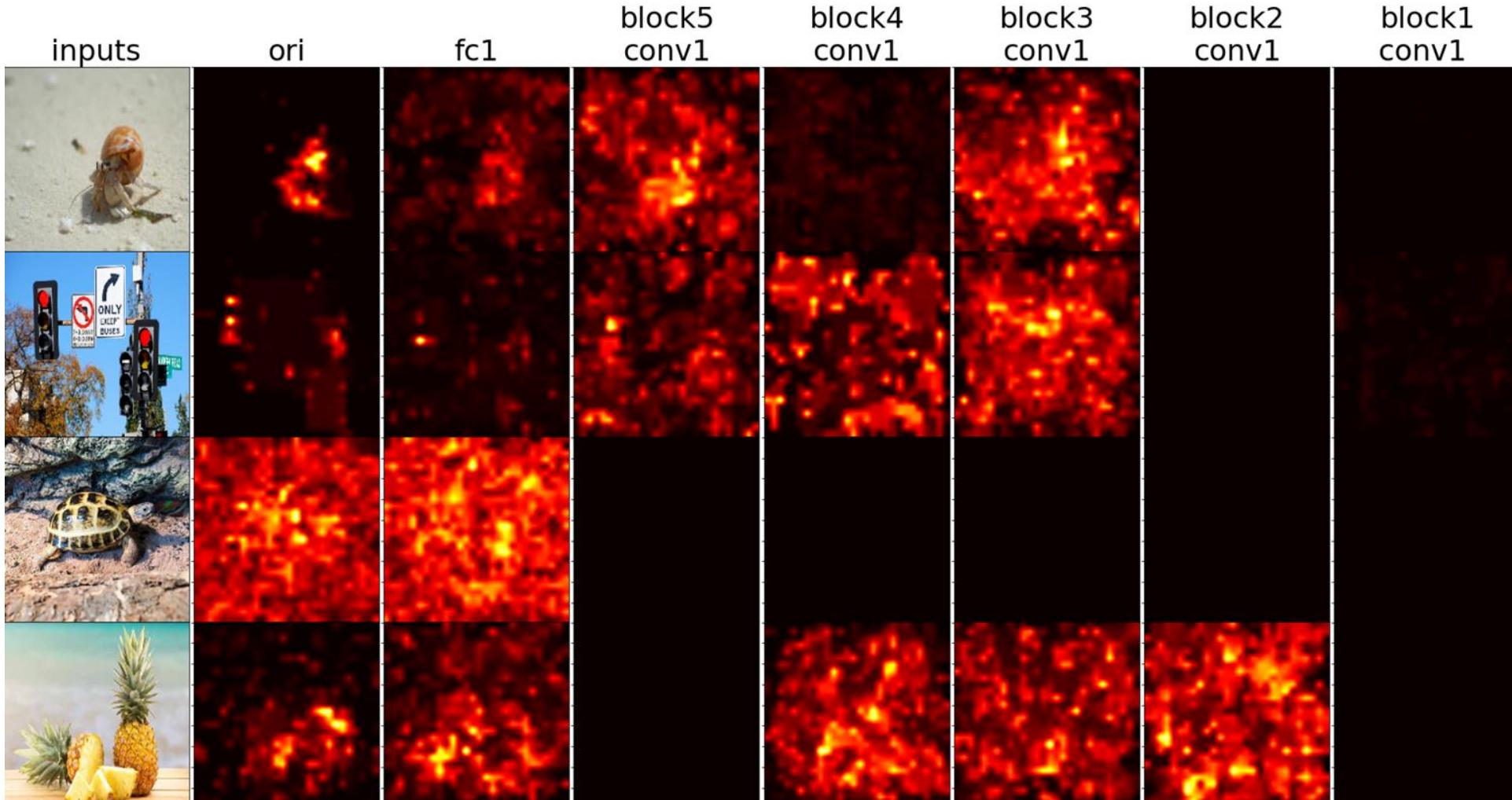


Figure 6: Examples of 10% images masked for all methods.

Linearly Estimated Gradient (LEG)

Experiment - sanity check

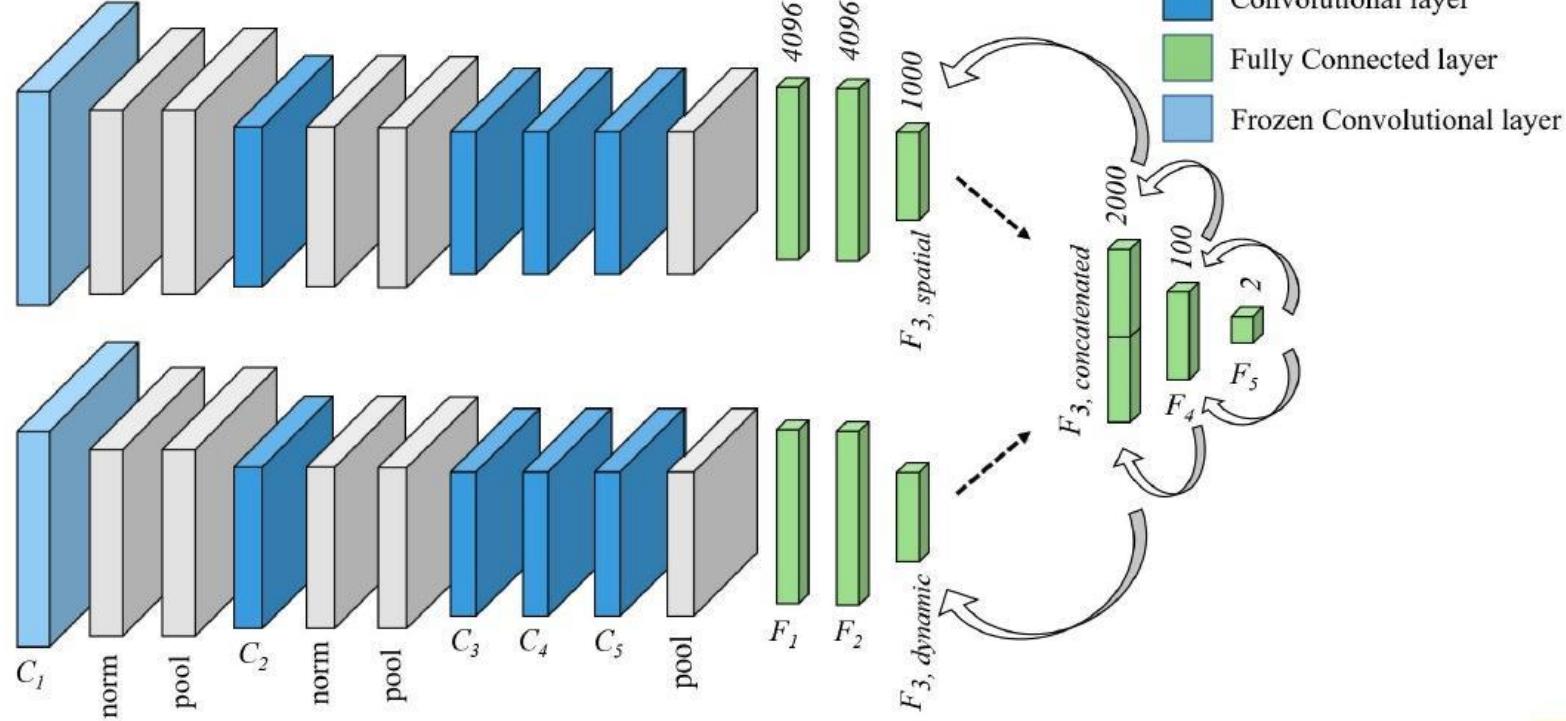


Outline

1. Background: why we need image-based interpretation
2. Taxonomy of Interpretation:
 - *Model-specific vs model-agnostic*
 - *Global vs Local*
3. Saliency based methods
 - *Overview*
 - *Gradient, Integrated Gradient, SmoothGrad, CAM, Grad-CAM*
 - *Lime*
 - *Shap*
 - *LEG*
 - *Medical Applications*

Medical application: XAI on breast MRI

spatial input



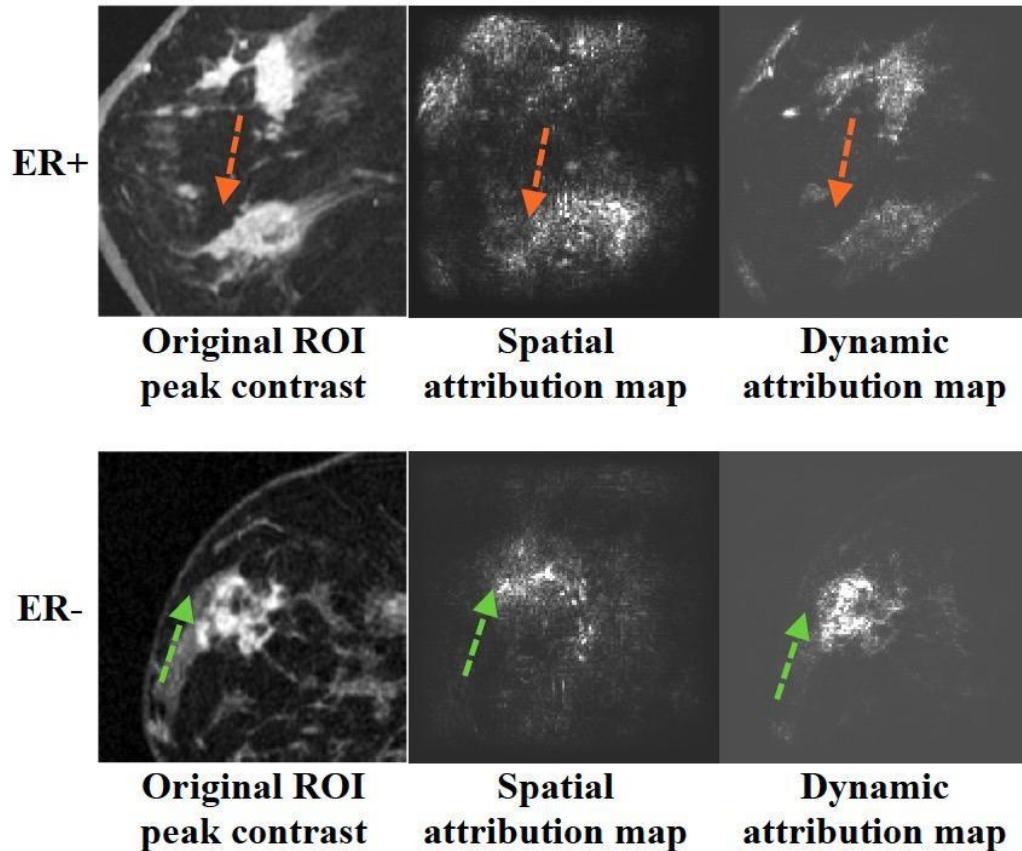
Explainable AI for medical imaging: deep-learning CNN ensemble for classification of estrogen receptor status from breast MRI

Dual domain CNN structure for breast cancer subtype classification, with identical structures adapted from AlexNet

Papanastasopoulos, et al. "Explainable AI for medical imaging: deep-learning CNN ensemble for classification of estrogen receptor status from breast MRI." SPIE Medical Imaging, 2020

Medical application: XAI on breast MRI

Method: backpropagation-based integrated gradients + SmoothGrad



In the spatial domain:

Model learned from the fatty tissue surrounding the tumor in the spatial domain more frequently.

In the dynamic domain:

Model distinguishes the tumoral tissue from fatty and dense tissue more effectively.

Can Existing Algorithms Fulfill Clinical Requirements?

- The existing XAI methods are typically not designed for **clinical purposes**. :(

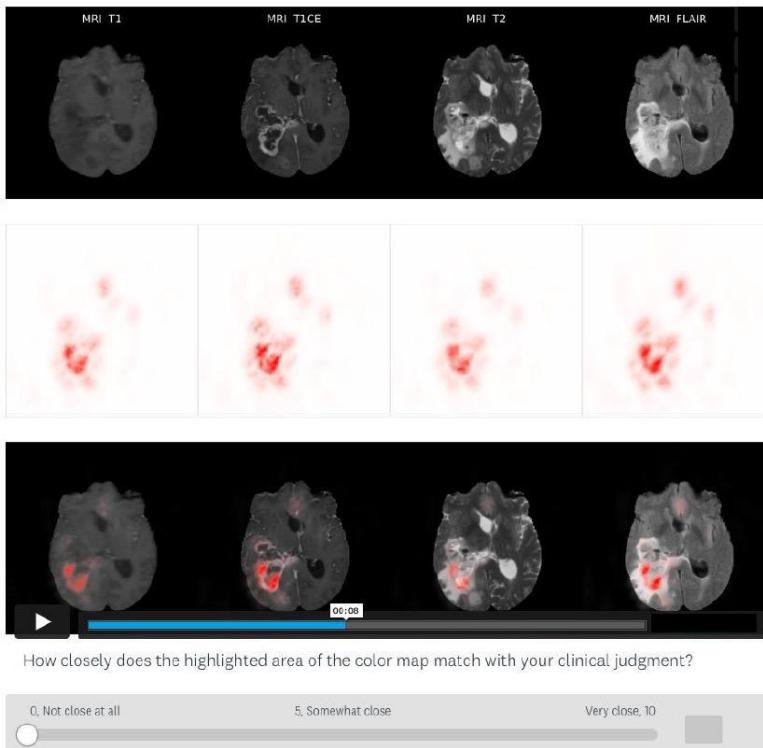
Weina Jin, et al. conduct extensive experiments to evaluate on 16 existing XAI algorithms using the “*computational metric modality specific feature importance*” (MSFI).

- Brain imaging data:
the publicly available BraTS 2020 dataset and a BraTS-based synthetic dataset.
- Limitation and future research:
The existing XAI algorithm relies on the accuracy and robustness of the prediction model.
Improving the prediction model is the basis for improving the effectiveness of XAI.

Can Existing Algorithms Fulfill Clinical Requirements?

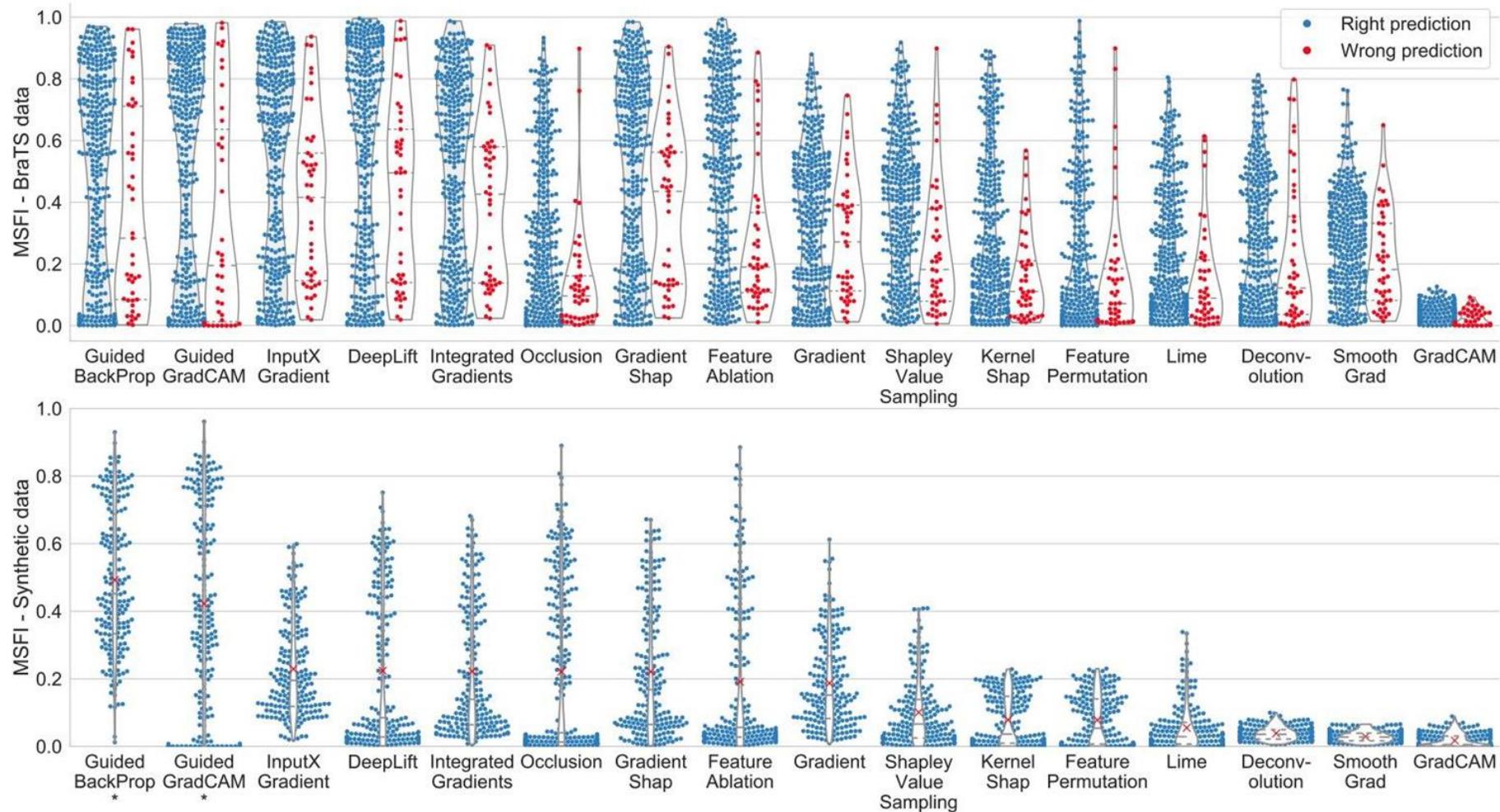
Column: MRI modality.

Row: MRI, heatmap, and heatmap overlaid on MRI.



	MSFI (BraTS)	Stat. Sig.	MSFI (Synthetic)	MI	diffAUC	FP	IoU	Doctors' Rating	Speed (second)
Guided BackProp	0.48±0.33	NS	0.49±0.21	0.80±0.27	0.06±0.08	0.07±0.11	0.02±0.02	0.6±0.1	1.7±1.1
Guided GradCAM	0.50±0.36	**	0.42±0.29	0.81±0.26	0.07±0.08	0.06±0.11	0.02±0.02	0.1±0.0	2.2±1.4
DeepLift	0.54±0.34	*	0.22±0.23	0.53±0.45	0.05±0.02	0.07±0.12	0.05±0.04	0.6±0.2	3.8±2.0
InputXGradient	0.51±0.32	*	0.23±0.14	0.87±0.16	0.04±0.02	0.07±0.11	0.05±0.04	0.1±0.0	1.7±1.1
Integrated Gradients	0.48±0.31	*	0.22±0.19	0.73±0.39	0.05±0.02	0.07±0.10	0.05±0.04	0.5±0.0	62±29
Gradient Shap	0.48±0.31	*	0.22±0.19	0.53±0.40	0.05±0.02	0.07±0.10	0.05±0.04	0.5±0.0	6.8±3.0
Feature Ablation	0.48±0.30	***	0.19±0.23	0.27±0.44	-0.02±0.08	0.07±0.10	0.03±0.04	0.4±0.4	74±23
Gradient	0.34±0.23	NS	0.19±0.13	0.47±0.16	0.07±0.13	0.05±0.07	0.02±0.01	0.6±0.6	1.8±1.1
Occlusion	0.28±0.26	***	0.22±0.25	0.60±0.33	0.04±0.03	0.03±0.07	0.02±0.02	0.6±0.2	989±835
Shapley Value Sampling	0.38±0.24	***	0.10±0.10	0.47±0.65	-0.04±0.13	0.07±0.09	0.03±0.04	0.2±0.1	2018±654
Kernel Shap	0.28±0.25	**	0.08±0.08	NaN	-0.05±0.09	0.05±0.07	0.03±0.04	0.1±0.0	194±100
Feature Permutation	0.23±0.26	NS	0.08±0.07	NaN	-0.05±0.07	0.04±0.07	0.02±0.04	0.1±0.0	14±2.2
Deconvolution	0.26±0.23	NS	0.04±0.02	0.73±0.39	0.05±0.08	0.04±0.07	0.02±0.01	0.4±0.4	1.8±1.0
Smooth Grad	0.27±0.17	*	0.03±0.02	0.67±0.00	0.19±0.16	0.04±0.06	0.02±0.01	0.7±0.1	12±6
Lime	0.24±0.21	**	0.05±0.07	0.53±0.58	-0.03±0.11	0.04±0.06	0.03±0.04	0.1±0.0	341±181
GradCAM	0.04±0.03	***	0.02±0.02	NaN	0.07±0.09	0.01±0.01	0.01±0.01	0.0±0.0	0.6±0.3

Can Existing Algorithms Fulfill Clinical Requirements?



The distributions of scores of existing XAI methods show poor performance of XAI on the two medical datasets

Summarization

- How to engage with domain experts, human in the loop of providing effective interpretation?
- Computing efficiency needs further improvement for the perturbation-based methods
- How to define comprehensive evaluation metrics without ground truth?
- Limitations for medical image interpretation

Part 3: Graph-based Model Interpretation

Outline

1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

Ninghao Liu, Qizhang Feng, and Xia Hu. "Interpretability in Graph Neural Networks." In *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer. 2022.

Outline

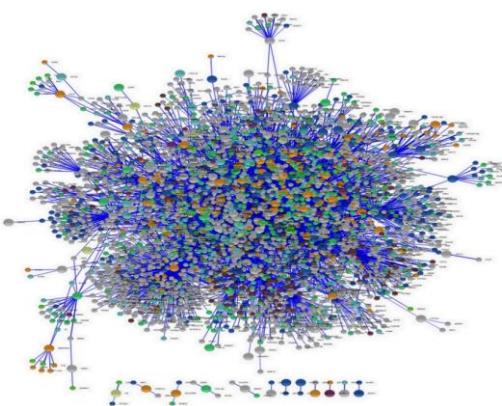
1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

Graph Data is Everywhere!

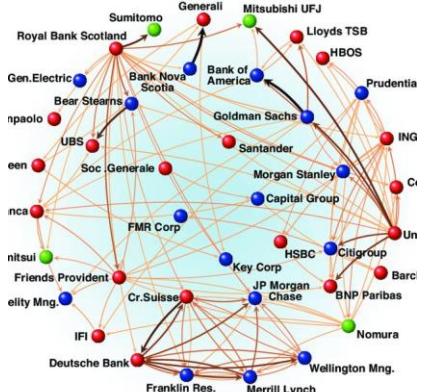
Social Networks



Biology Networks



Finance Networks



Internet of Things



Recommender Systems



Transportation Networks

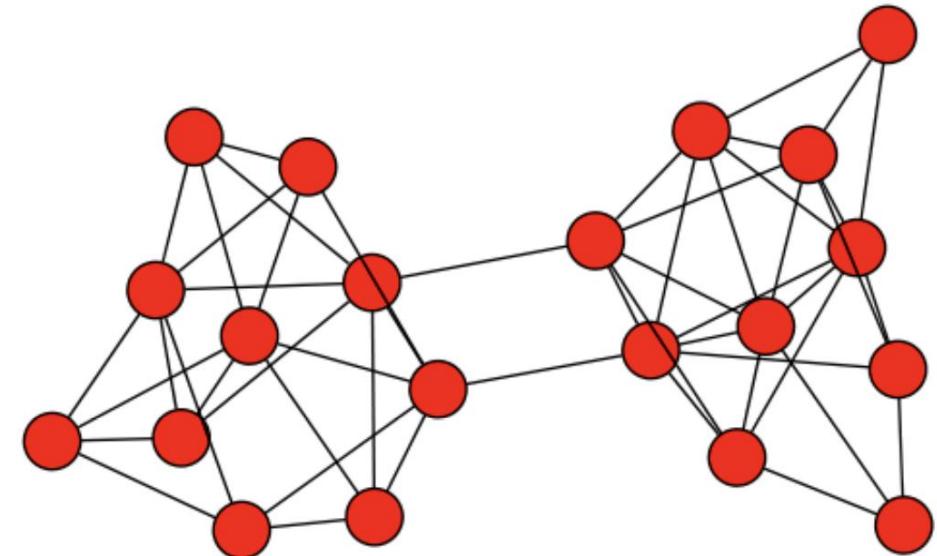


Networks (Graphs)

A general description of data and their relations.

A **homogeneous graph** is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

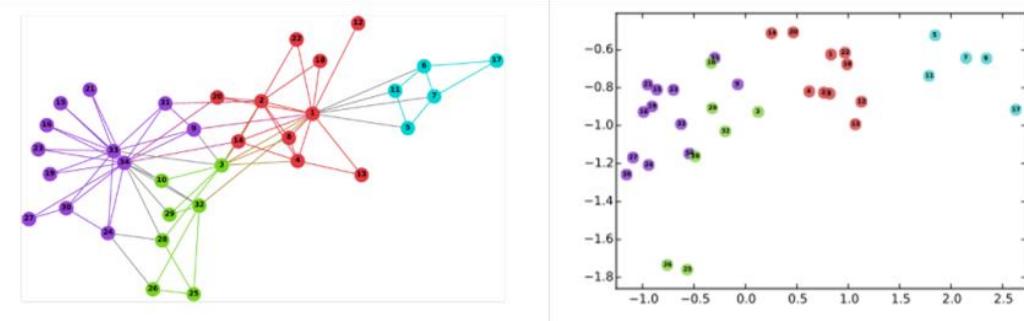
- \mathcal{V} is the node set, \mathcal{E} is the edge set.
- $A \in \mathbb{R}^{n \times m}$: adjacency matrix.
- $X \in \mathbb{R}^{n \times m}$: node feature matrix.
- $n = |\mathcal{V}|$, m is the feature dimension.



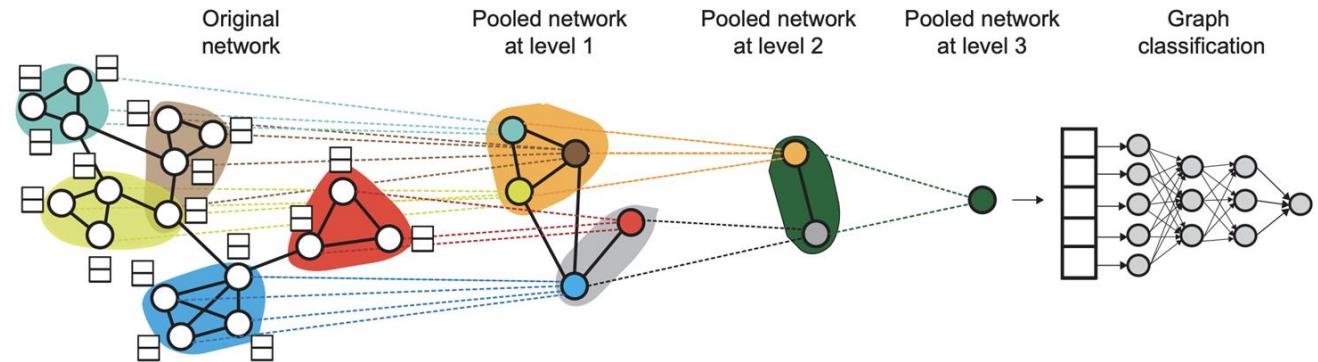
There are more complex graphs, where nodes and relations are of multiple types.

Background: Graph Representation Learning

- Learning representations of **nodes** (i.e., node embeddings).



- Learning representations of **graphs** (i.e., graph embeddings).



- Learning representations of **edges** or **node features**, etc.

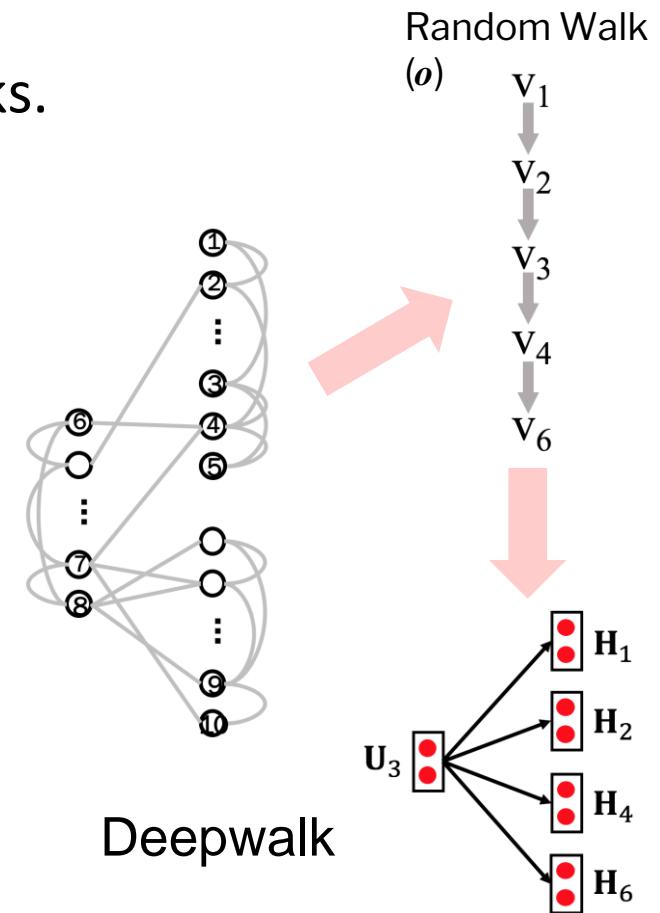
Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." KDD. 2014.

Background: Graph Representation Learning

- Preserve **attributes** and underlying **structure** of networks.
 - Pair-wise proximity
 - Higher-order proximity
 - Community structures
 - Feature similarity
 - Global structure (e.g., centrality)
 -
- **Similar nodes/graphs are mapped closer.**
- Example: DeepWalk, node2vec, LINE.

maximize
↓

$$p(v_j|v_i) = \frac{\exp(\langle \mathbf{H}_j, \mathbf{U}_i \rangle)}{\sum_v \exp(\langle \mathbf{H}_v, \mathbf{U}_i \rangle)}$$



Tang et al. LINE: Large-scale Information Network Embedding. WWW 2015.
Grover et al. node2vec: Scalable feature learning for networks. KDD. 2016.

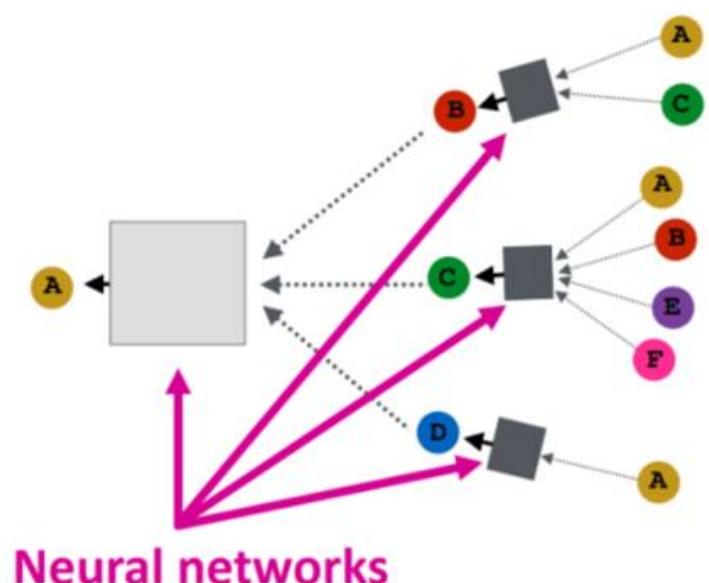
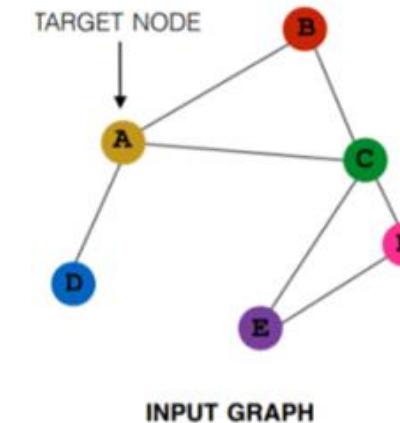
Background: Graphs Neural Networks

Graph Neural Networks (GNNs)

Iteratively **aggregate information** from neighbors towards the target node:

$$H_i^{l+1} = \sigma\left(\sum_{j \in \mathcal{V}_i \cup \{i\}} \frac{1}{c_{i,j}} H_j^l W^l\right),$$

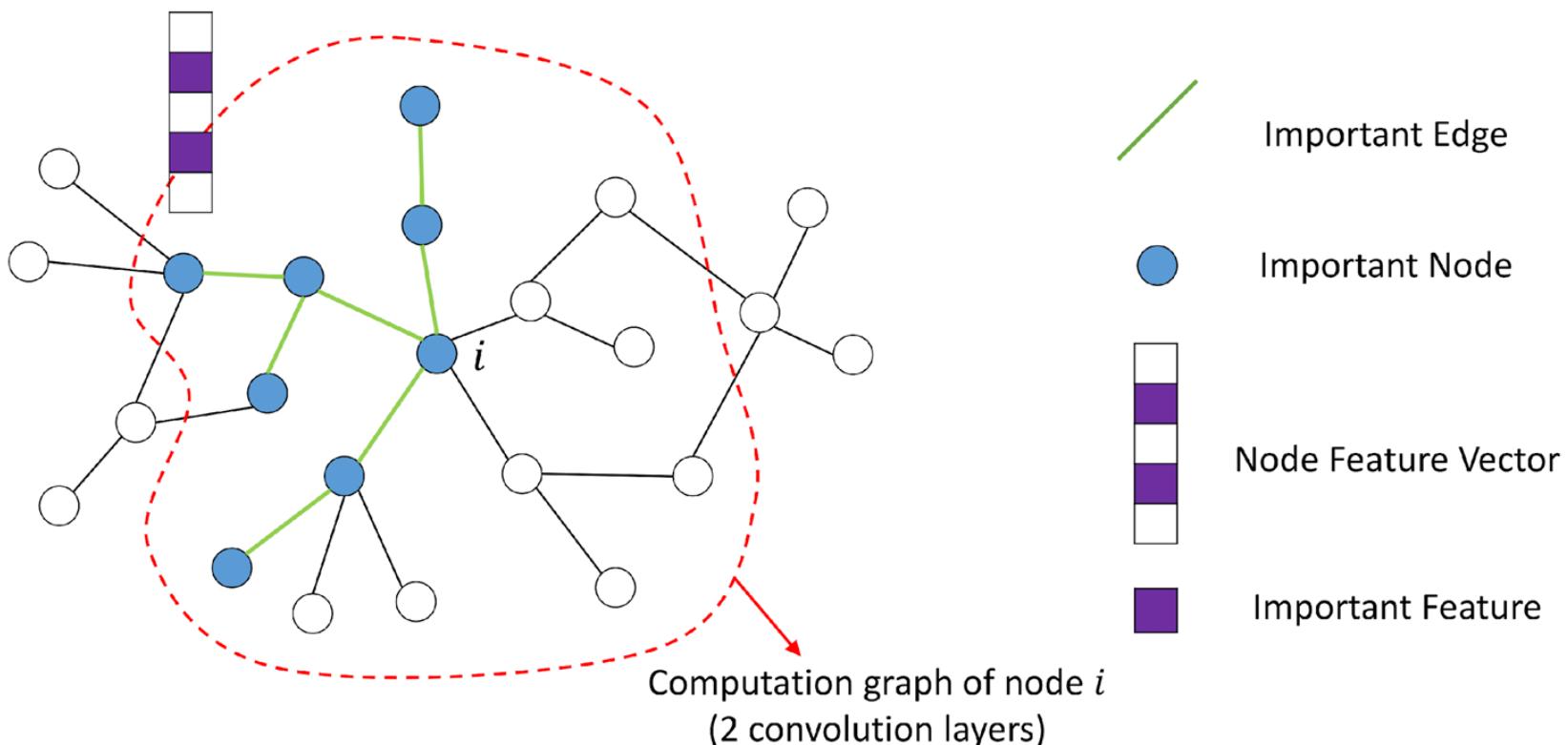
- H_i^l is the embedding of node i at layer l
- W^l is the trainable parameters at layer l
- \mathcal{V}_i is the neighbors of node i
- $\frac{1}{c_{i,j}} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, where $\tilde{A} = A + I$, and \tilde{D} is the diagonal degree matrix of \tilde{A} .



Welling, Max, and Thomas N. Kipf. "Semi-supervised classification with graph convolutional networks." ICLR. 2017.

Interpretation in Graph Modelling

Interpretation tries to identify what are the **important features**, **important nodes**, **important edges**, and **important subgraphs** that contribute to the prediction.



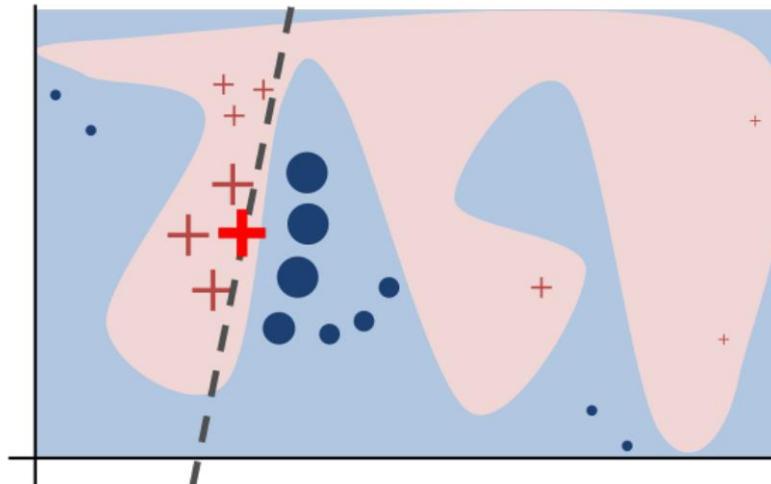
Outline

1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

Approximation-based Explanation

Use a **simple and interpretable** model to fit the target model's decision, so the explanation can be extracted from the simple model.

- **White-box** Approximation utilize information (e.g., **gradients, neuron activations**) inside the model.
- **Black-box** Approximation does NOT utilize information inside the model.

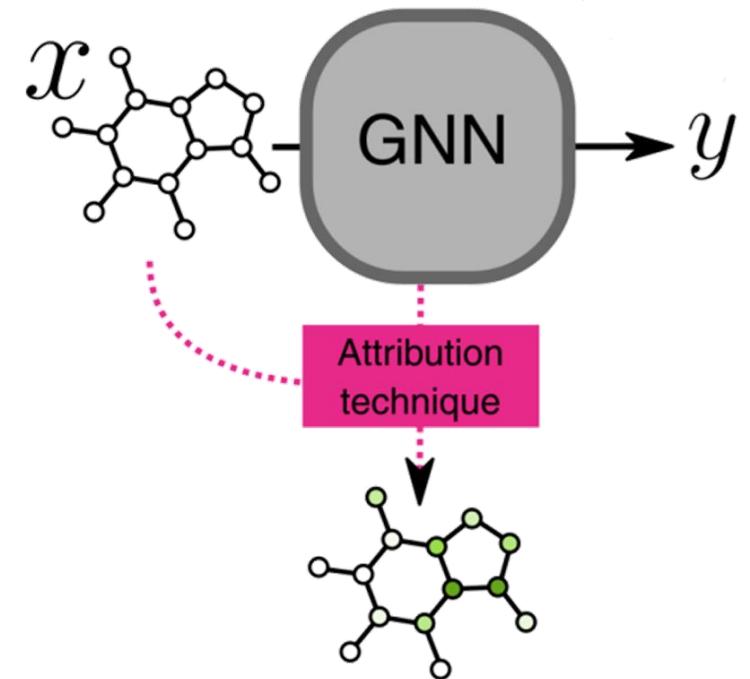


Ribeiro et al. " Why should i trust you?" Explaining the predictions of any classifier." KDD. 2016.

Sensitivity Analysis (SA)

- White-box approximation.
- Let \mathbf{x} denote the feature vector of a node of interest.
- Let $f(\mathcal{G})$ denote the target prediction for \mathcal{G} .
- Sensitivity score:

$$\mathcal{S}(\mathbf{x}) = \|\nabla_{\mathbf{x}} f(\mathcal{G})\|^2,$$



where the local gradient of the prediction with respect to the input node features is used to represent **node importance**.

- **Edge importance** obtained by averaging end nodes' importance.

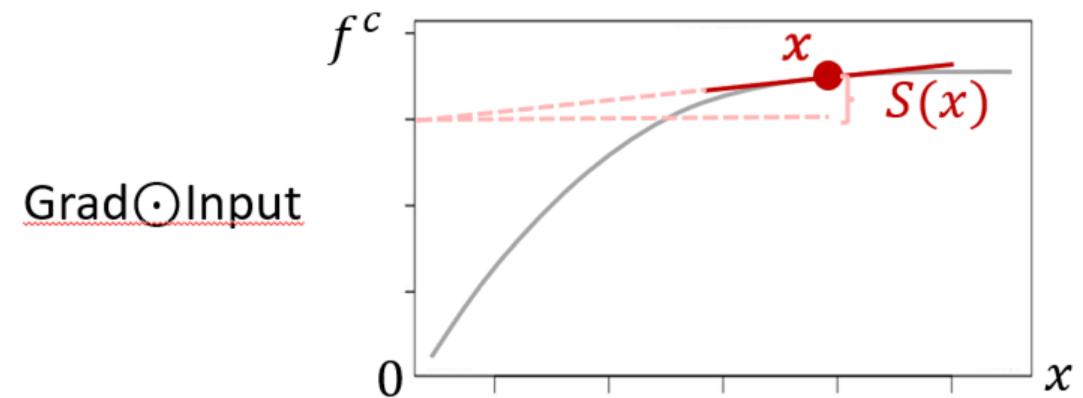
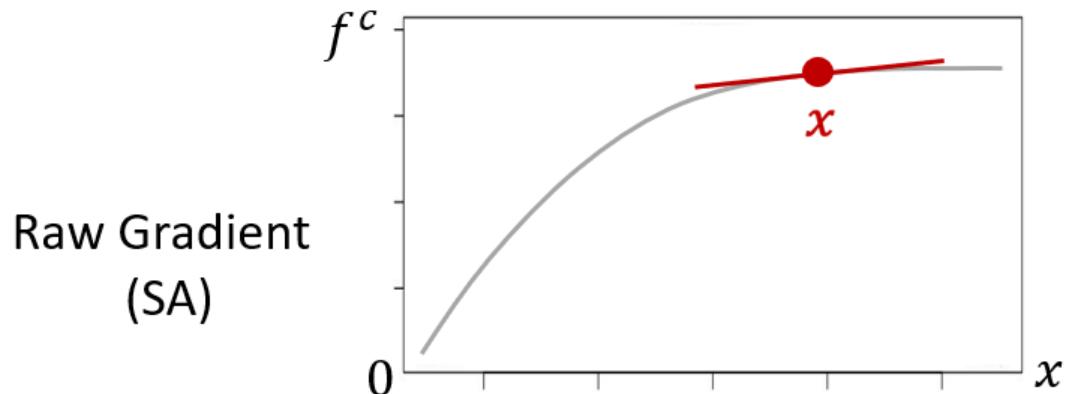
Grad \odot Input

- Extend feature sensitivity to **feature contribution**

$$\mathcal{S}(\mathbf{x}) = \nabla_{\mathbf{x}}^{\top} f(\mathcal{G}) \odot \mathbf{x},$$

where \odot denotes the element-wise product of the input features and the gradients.

- Issue: saturation.



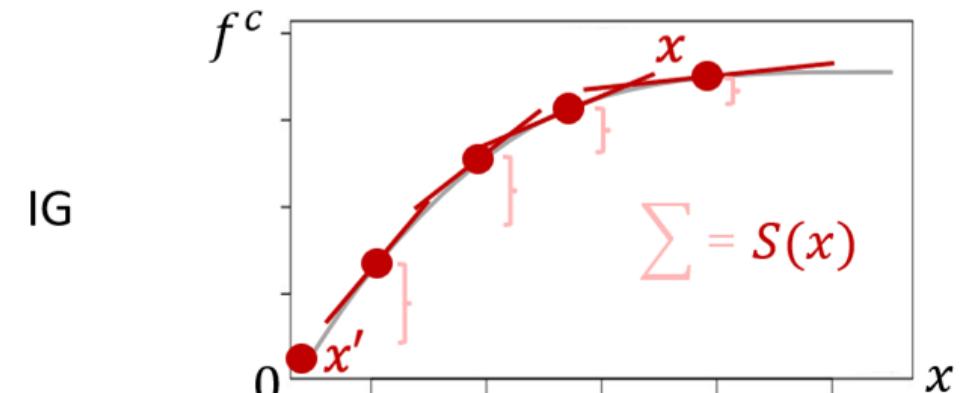
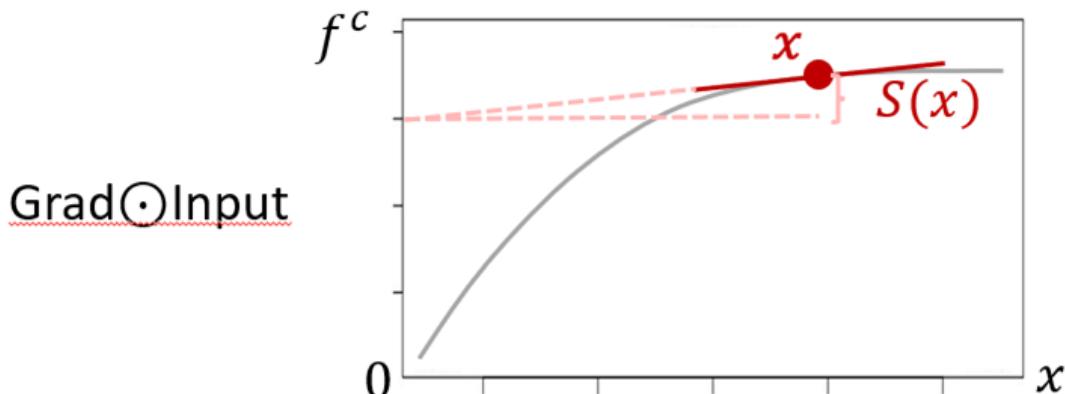
Integrated Gradients (IG)

- Aggregates feature contribution along a designed path in the input space.
- The path starts from a chosen baseline point \mathcal{G}' and ends at the target input \mathcal{G} :

$$S(\mathbf{x}) = (\mathbf{x} - \mathbf{x}') \int_{\alpha=0}^1 \nabla_{\mathbf{x}} f(\mathcal{G}' + \alpha(\mathcal{G} - \mathcal{G}')) d\alpha,$$

where \mathbf{x}' is the feature vector of the baseline input \mathcal{G}' .

- Grad ⊙ Input can be seen as a special case of IG:
 - The path has only one hop; \mathbf{x}' is chosen as all-zero

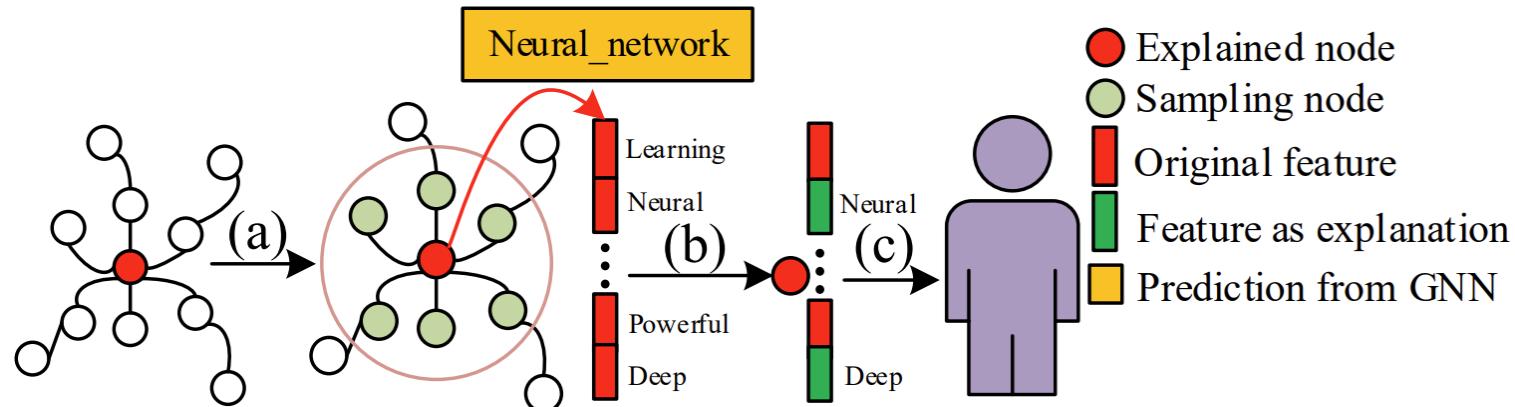


GraphLime

- Black-box approximation.
- Focused on finding important features (and nodes).
- Given the target node v_t , its neighborhood space is defined as:

$$\mathcal{V}_t = \{v \mid \text{distance}(v_t, v) \leq k, v \in \mathcal{V}\}$$

We could then collect a set of instances $\{(x_i, y_i)\}$, where x_i and y_i are the feature vector and prediction of $v_i \in \mathcal{V}_t$.



Huang, Qiang, et al. "Graphlime: Local interpretable model explanations for graph neural networks." TKDE. 2022.

GraphLime

- Employs HSIC Lasso (Hilbert-Schmidt independence criterion Lasso) to measure the relation between features and predictions of the nodes:

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{2} \left| \left| \bar{\mathbf{L}} - \sum_{z=1}^d \beta_z \bar{\mathbf{K}}^{(z)} \right| \right|_F^2 + \rho \|\beta\|_1,$$

s.t. $\beta_1, \dots, \beta_d \geq 0,$

where $\bar{\mathbf{L}} = \frac{\mathbf{HLH}}{\|\mathbf{HLH}\|_F}$ and $\bar{\mathbf{K}}^{(z)} = \frac{\mathbf{HK}^{(z)}\mathbf{H}}{\|\mathbf{HK}^{(z)}\mathbf{H}\|_F}$ are the normalized centered Gram matrixes, $\mathbf{H} = \mathbf{I}_k - \frac{1}{k} \mathbf{1}_k \mathbf{1}_k^T$ is the centering matrix, $\mathbf{L}_{i,j} = L(y_i, y_j)$ and $[\mathbf{K}^{(z)}]_{ij} = K(x_i^{(z)}, x_j^{(z)})$ are the kernels for the output and the z -th dimensional input.

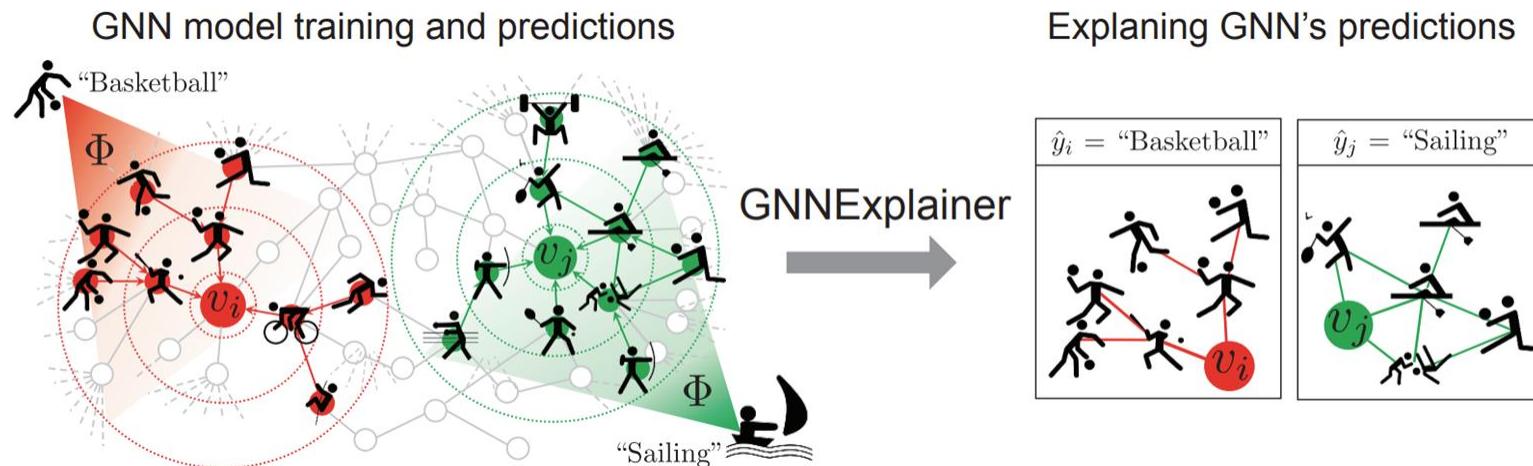
- β_z is the importance of the z -th feature.

Outline

1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - **Perturbation Methods**
 - *Decomposition Methods*
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

GNNExplainer

- **Intuition:**
 - Masking out the **important parts** will have a **significant impact** on the output
 - Masking out the **unimportant parts** will lead to a **negligible impact**.
- Given model prediction on a node v_t , GNNExplainer finds a compact subgraph \mathcal{G}_s from the original graph around v_t that is most crucial for its prediction.



Ying, Zhitao, et al. "Gnnexplainer: Generating explanations for graph neural networks." NeurIPS. 2019.

GNNExplainer

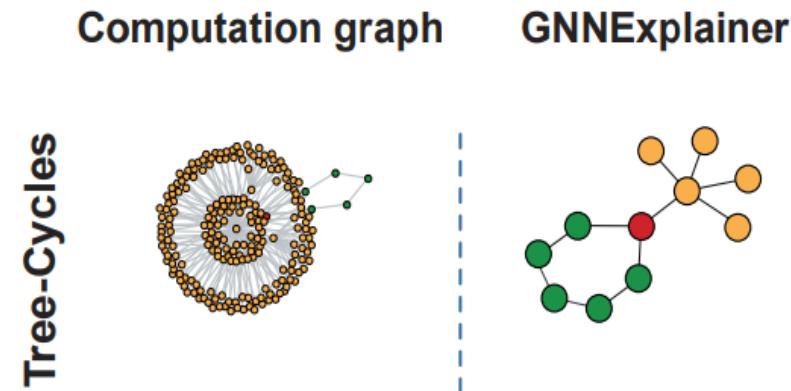
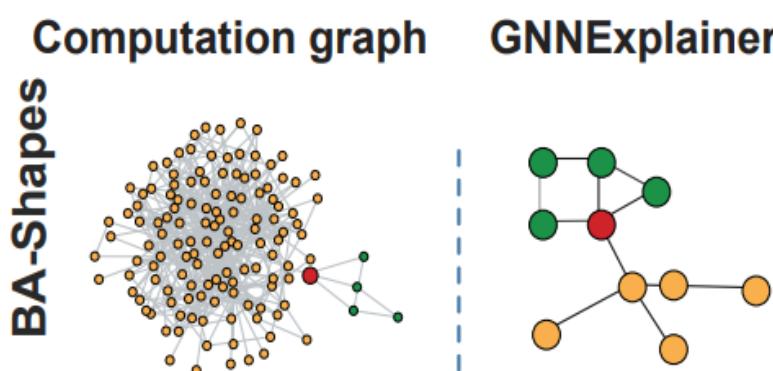
Idea: Choose a **subgraph** \mathcal{G}_S which maximizes the mutual information (MI) between the predictions of the original graph \mathcal{G} and the subgraph \mathcal{G}_S

$$\max_{\mathcal{G}_S} MI(Y, (\mathcal{G}_S, X_S)) = H(Y) - H(Y \mid \mathcal{G} = \mathcal{G}_S, X = X_S)$$

where X_S is the node features of the subgraph \mathcal{G}_S , Y is the predicted label distribution, and its entropy $H(Y)$ is a constant.

How to extract the subgraph?

- Apply a trainable soft mask M on adjacency matrix on the adjacency matrix of \mathcal{G} .

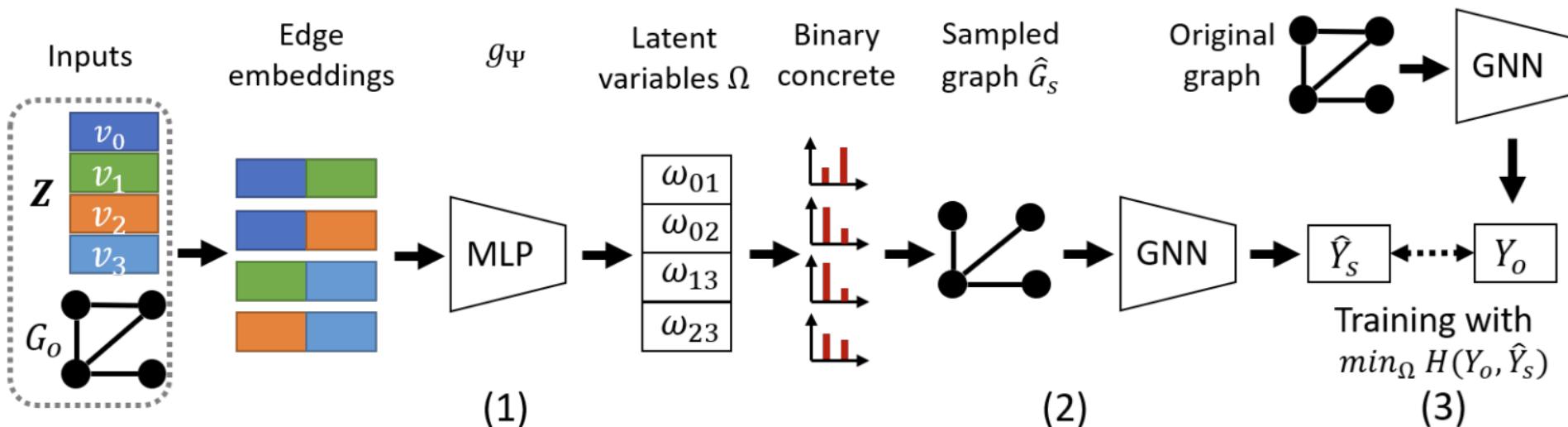


PGExplainer

PGExplainer learns a **mask function** applied on **edges** to explain the predictions. It uses a deep neural network to generate edge mask values:

$$M_{i,j} = \text{MLP}_\Psi ([\mathbf{z}_i; \mathbf{z}_j]),$$

where Ψ denotes the trainable parameters of the MLP, \mathbf{z}_i and \mathbf{z}_j are the feature embeddings of the node i and j .



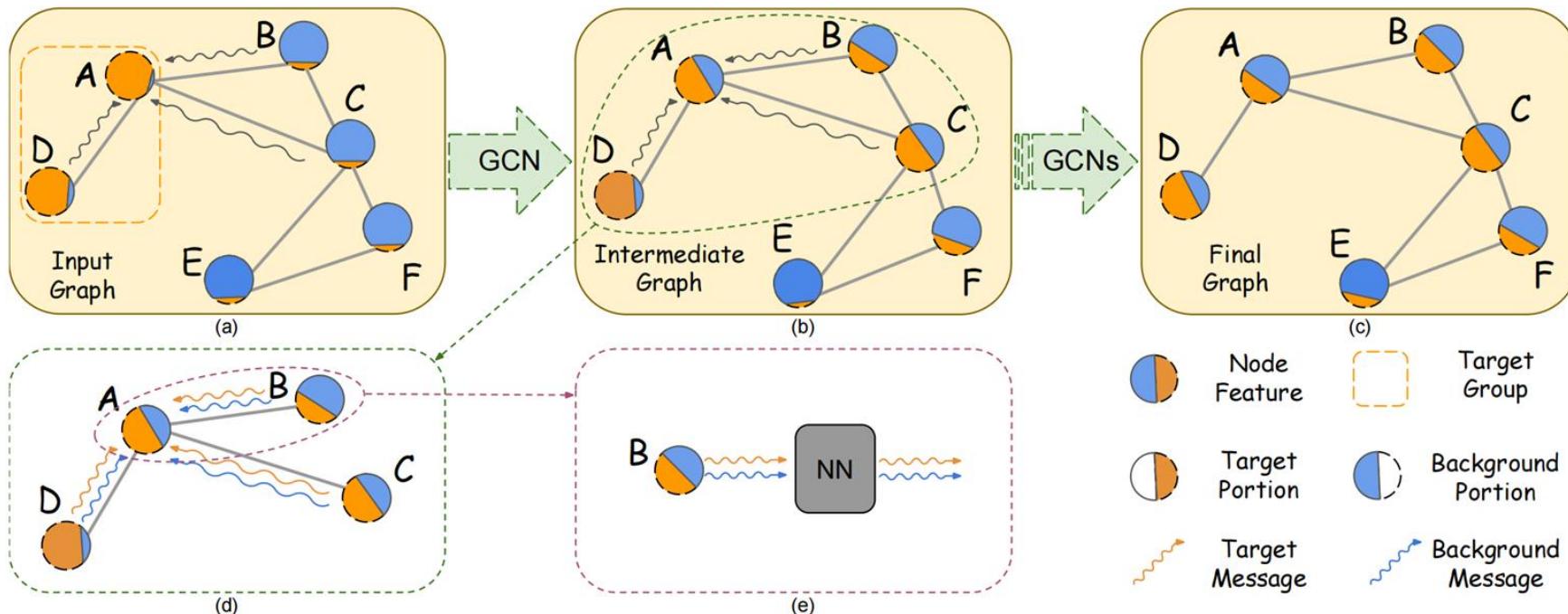
Luo, Dongsheng, et al. "Parameterized explainer for graph neural network." NeurIPS. 2020.

Outline

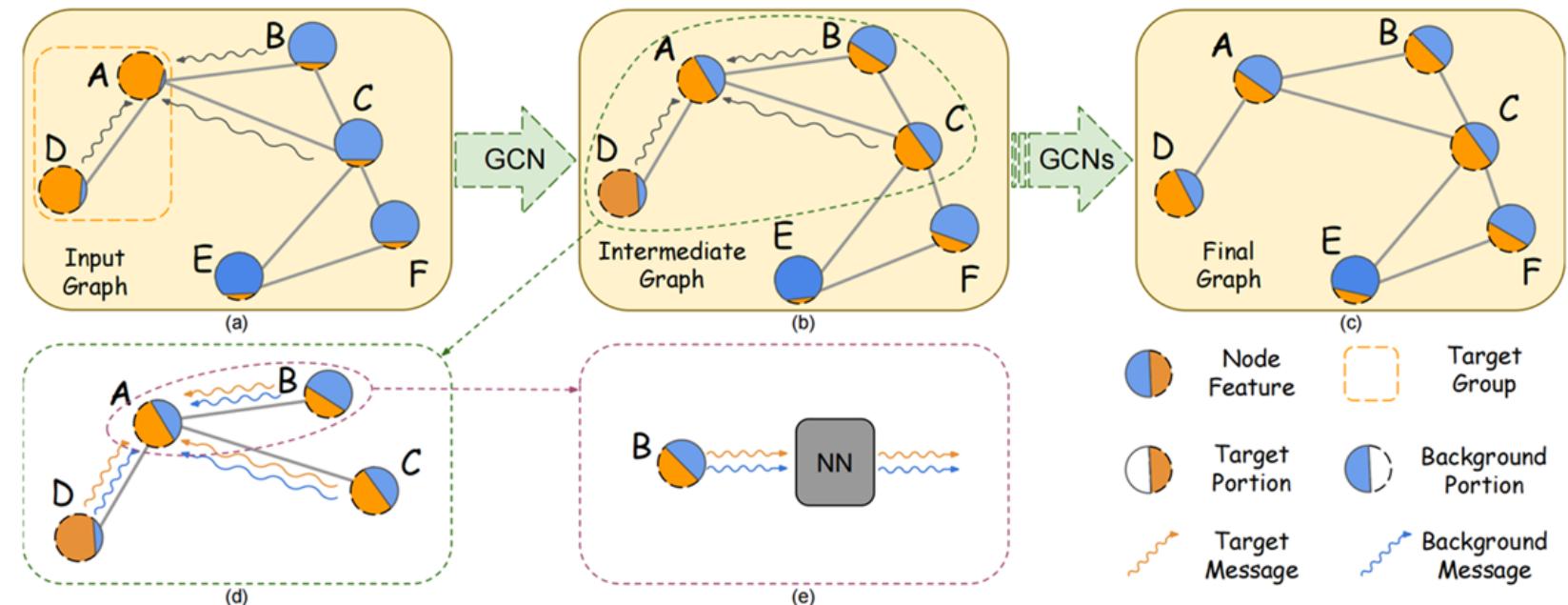
1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - ***Decomposition Methods***
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

DEGREE

- White-box explanation: Closely examine each GNN layer.
- The information flow in GNN's message propagation is decomposable.
 - **Information flow = Target flow + Background flow.**



DEGREE



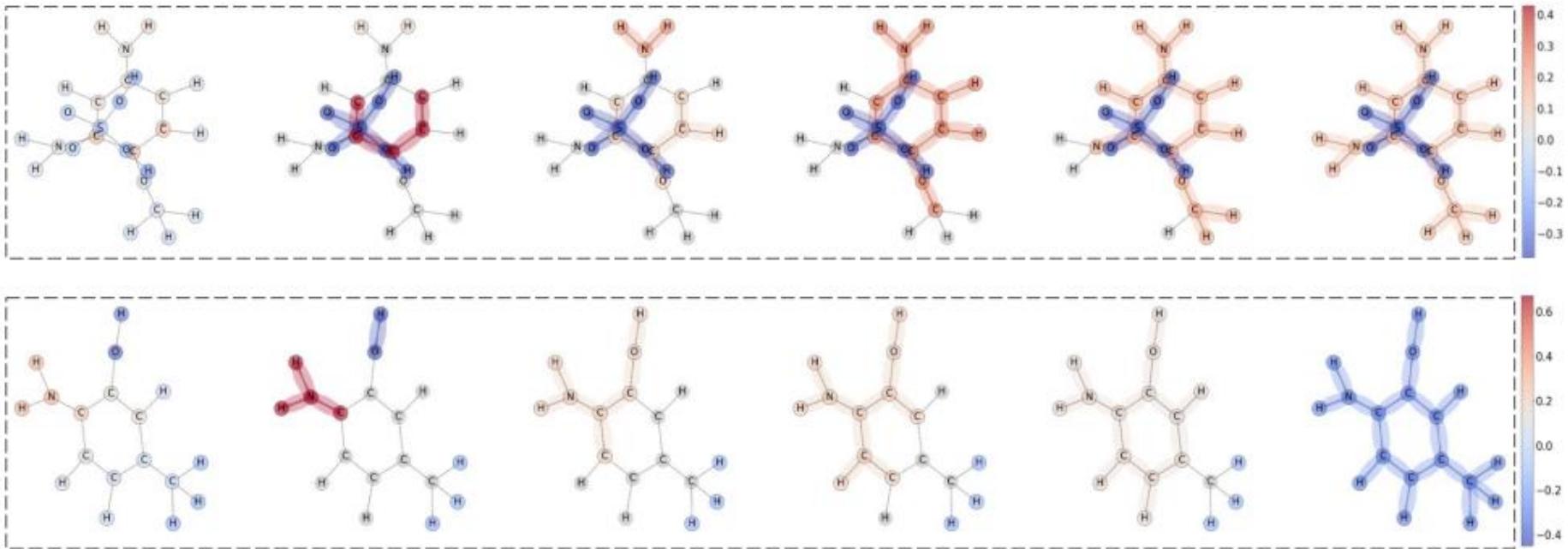
The corresponding decomposition to a **GCN layer** can be designed as:

$$\begin{aligned}\gamma[t] &= \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X}^\gamma[t] \mathbf{W}, \quad \beta[t] = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X}^\beta[t] \mathbf{W}, \\ \mathbf{X}^\gamma[t+1] &= \gamma[t] + \mathbf{b} \cdot \frac{|\gamma[t]|}{|\gamma[t]| + |\beta[t]|}, \quad \mathbf{X}^\beta[t+1] = \beta[t] + \mathbf{b} \cdot \frac{|\beta[t]|}{|\gamma[t]| + |\beta[t]|},\end{aligned}$$

where $\mathbf{X}^\gamma[t]$ and $\mathbf{X}^\beta[t]$ are the target and background portions of the input $\mathbf{X}[t]$.

Decomposition schemes could also be designed for other layers: Fully Connected Layer, MaxPooling, ReLU, and Softmax.

DEGREE



Find important nodes -> Find important subgraphs with greedy search.

Advantage: Interpretation fidelity.

Limitation: Each layer requires a decomposition schema to be designed.

Outline

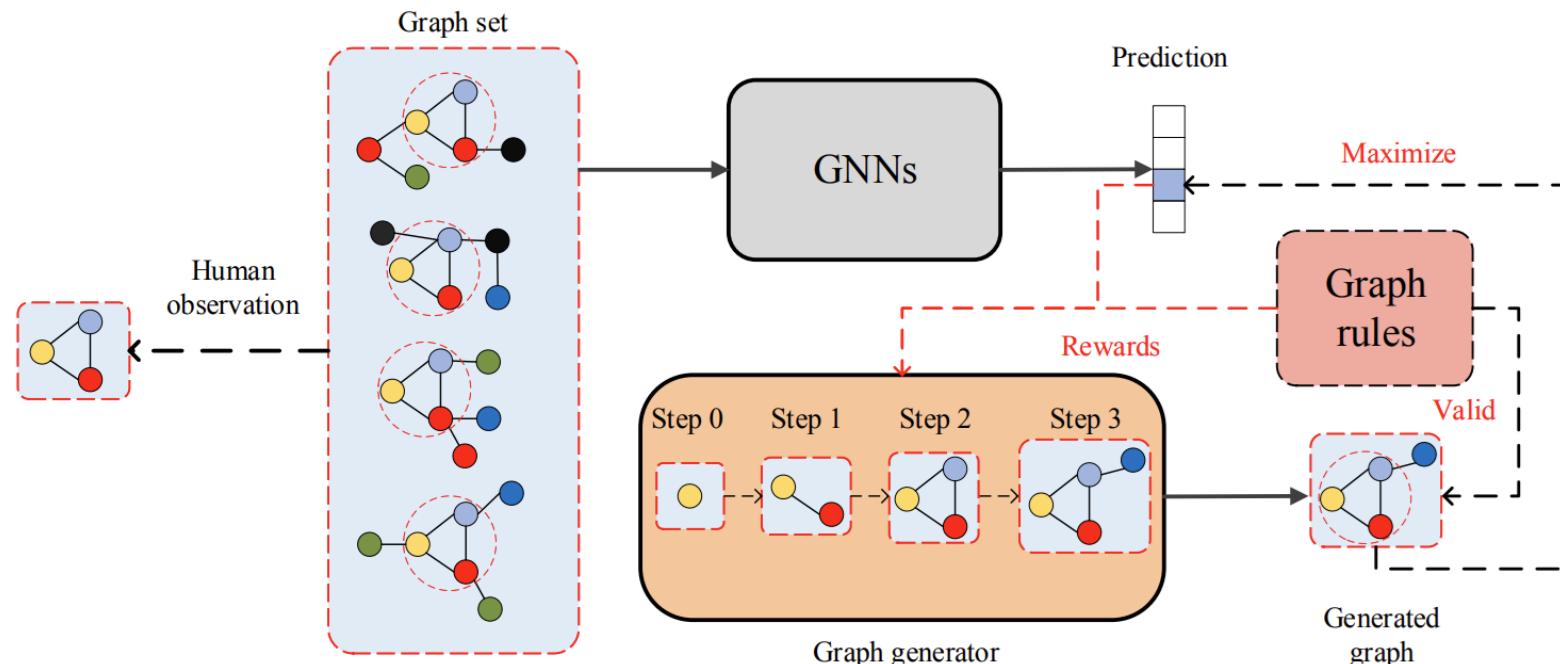
1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - ***Generative Methods***
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

XGNN

XGNN obtains explanation by generating a graph that maximizes the prediction of the target GNN f . The graph generation is defined as a reinforcement learning task:

$$\mathcal{G}^* = \operatorname{argmax}_{\mathcal{G}} P(f(G) = c_i),$$

where c_i is a chosen class to be explained for, and \mathcal{G}^* is the optimal graph we need.



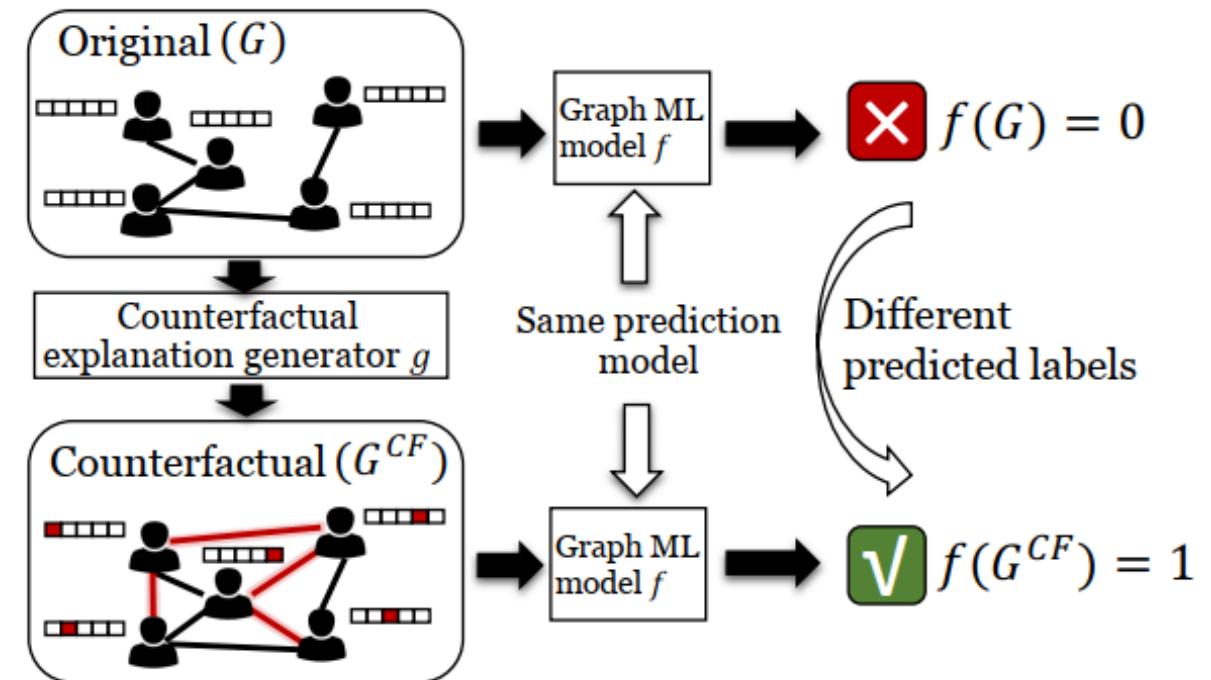
Yuan, Hao, et al. "Xgnn: Towards model-level explanations of graph neural networks." KDD. 2020.

Generative Counterfactual Explanations

- **Counterfactual Explanation (CFE):** How should the input G be slightly perturbed to new features G' to obtain a different predicted label (often a desired label)?

Contributions:

- Work for graph data (i.e., discrete and disorganized).
- Train a CFE generator, with the autoencoder architecture, to be generalizable to **unseen graphs**.
- Incorporate **causality** to generate more realistic counterfactuals.



Outline

1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - *Generative Methods*
 - ***Evaluation: Datasets & Metrics***
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

Synthetic Datasets

Dataset	BA-Shapes	Community	Tree-Cycles	Tree-Grid	BA-2motifs
Base					
Motifs					
Features	None	$\mathcal{N}(\mu_l, \sigma_l)$	None	None	None
Example					

Question: Does the model really use the ground-truth motifs?

Real-World Datasets

Dataset	Task	Domain	Node	Edge	Scale
MUTAG ^[1]	Graph Classification	Chemistry	Atoms	Chemical Bonds	4.3k
Delaney Solubility ^[2]	Graph Regression	Chemistry	Atoms	Chemical Bonds	1.1k
REDDIT-BINARY ^[3]	Graph Classification	Social Community	Users	User Interactions	2.0k
Bitcoin-Alpha/OTC ^[4]	Node Classification	Finance Risk Control	Users	User Ratings	3.8k
MNIST SuperPixel-Graph ^[5]	Graph Classification	Computer Vision	Centroids of Superpixels	Adjacency of Superpixels	70.0k

[1] Debnath, Asim Kumar, et al. "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity." *Journal of medicinal chemistry* 34.2 (1991): 786-797.

[2] Delaney, John S. "ESOL: estimating aqueous solubility directly from molecular structure." *Journal of chemical information and computer sciences* 44.3 (2004): 1000-1005.

[3] Yanardag, Pinar, and S. V. N. Vishwanathan. "Deep graph kernels." *KDD*. 2015.

[4] Kumar, Srijan, et al. "Edge weight prediction in weighted signed networks." *ICDM*. 2016.

[5] Dwivedi, Vijay Prakash, et al. "Benchmarking graph neural networks." *arXiv*. 2020.

Measuring the Quality of Explanations.

Fidelity:

$$fidelity = \frac{1}{N} \sum_{i=1}^N \left(f^{y_i}(\mathcal{G}_i) - f^{y_i}(\mathcal{G}_i \setminus \mathcal{G}'_i) \right),$$

where \mathcal{G}_i is the i -th graph, \mathcal{G}'_i is the explanation for it, and $\mathcal{G}_i \setminus \mathcal{G}'_i$ represents the perturbed i -th graph in which the identified explanation is removed.

Contrastivity:

$$Contrastivity = \frac{d_H(\hat{m}_0, \hat{m}_1)}{\hat{m}_0 \vee \hat{m}_1},$$

where d_H is the Hamming distance, and $\hat{m}_0 \vee \hat{m}_1$ are binarized heat-maps for positive and negative classes.

Sparsity:

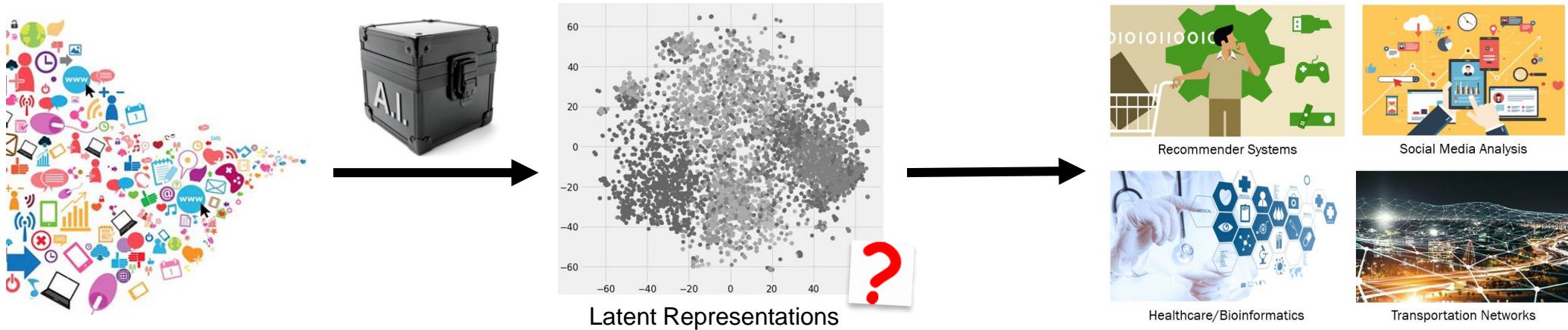
$$Sparsity = 1 - \frac{\hat{m}_0 \vee \hat{m}_1}{|V|},$$

Stability is the performance gap of the target model before and after adding noise to the explanation.

Outline

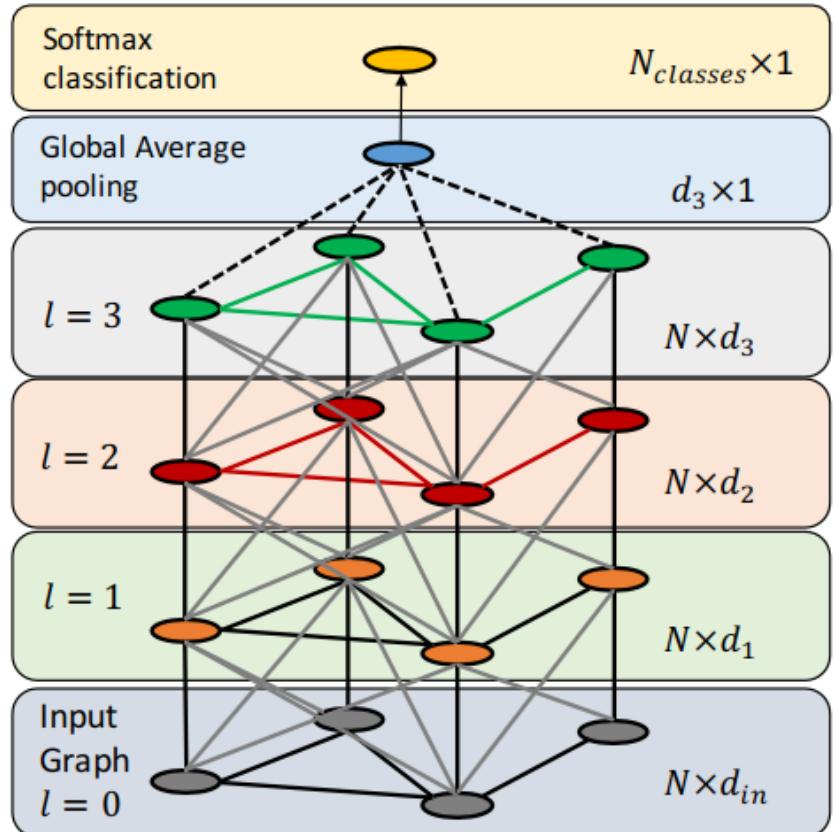
1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

Opacity of Latent Representations

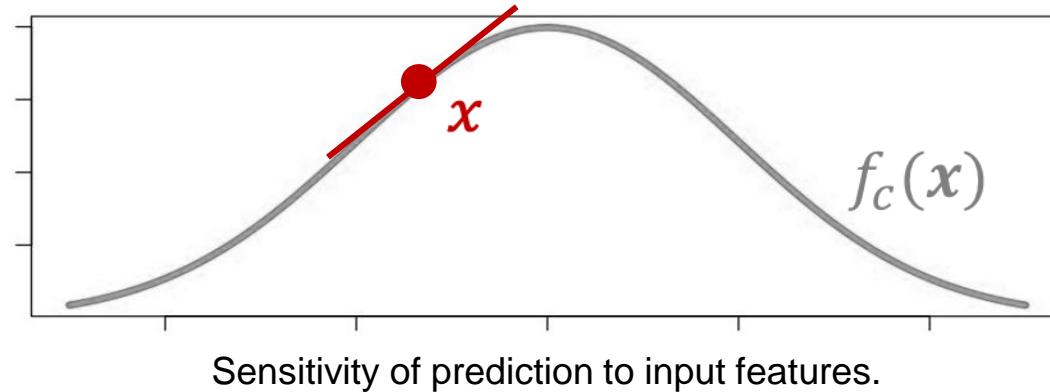


- **Prevalence of representation (embedding) learning:**
 - “All you need is embedding”.
 - Effective data representation benefits downstream applications.
- **Representation space opacity:**
 - The **meanings** of latent dimensions are not obtainable.
 - Given a representation vector, **why** is it mapped there?
 - Traditional interpretation methods cannot be applied here.

Opacity of Latent Representations



$$f_c(\mathbf{x}) \approx f_c(\mathbf{x}_0) + \boxed{\nabla_{\mathbf{x}} f_c(\mathbf{x}_0)^T \cdot (\mathbf{x} - \mathbf{x}_0)}$$

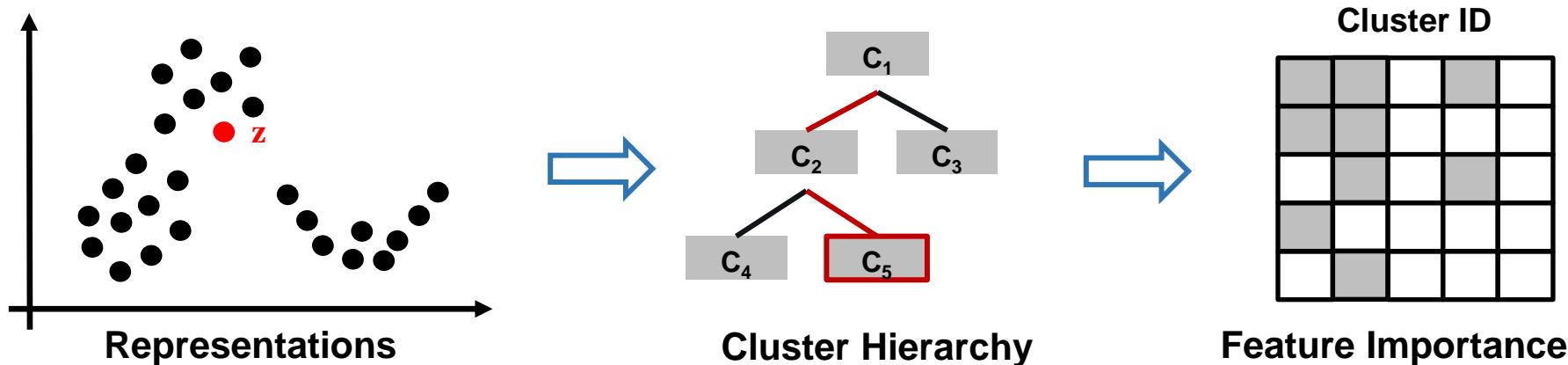


Challenge: $f_c(\mathbf{x})$ is not available in unsupervised representation learning.

Outline

1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

Interpretation by a Taxonomy



Input:

- $Z \in \mathbb{R}^{N \times D}$: Representation vectors.
- $X \in \mathbb{R}^{N \times M}$: Human understandable attributes that describe instances (e.g., product attributes in recommender systems).

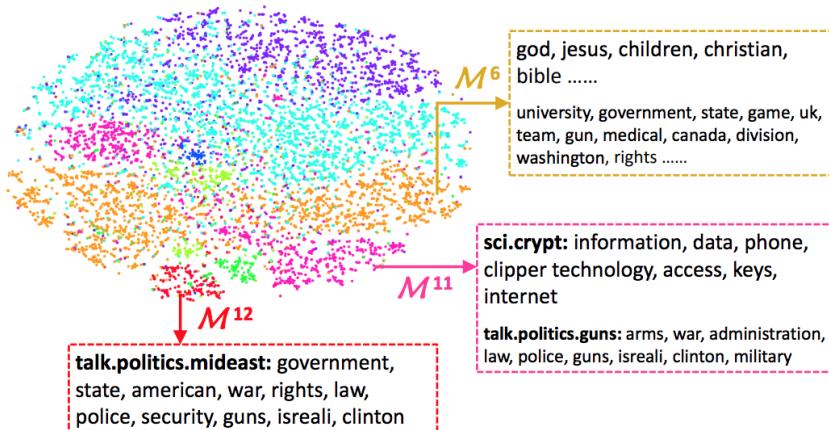
Output – A taxonomy (global interpretation):

- Cluster structures in latent space.
- Attribute importance scores of each cluster.

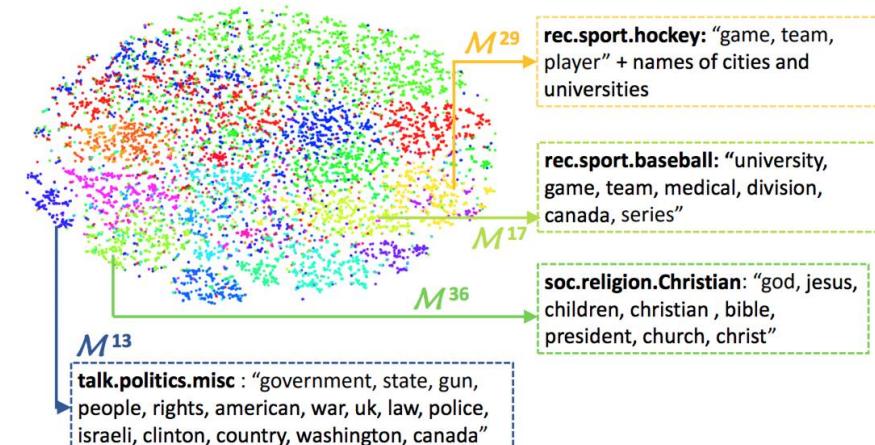
Liu, N., Huang, X., Li, J & Hu, X. On Interpretation of Network Embedding via Taxonomy Induction. KDD. 2018.

Experiment: A Case Study

- **Dataset:** 20NG.
- **Preprocessing:** Documents -> graph data. Each doc is a node.
- **Visualization** of representations and interpretation results:
 - Different cluster granularities (7 vs 20) in two figures.
 - The key words show the topic of each cluster.
 - $M^6 = M^{13} \cup M^{17} \cup M^{29} \cup M^{36}$.



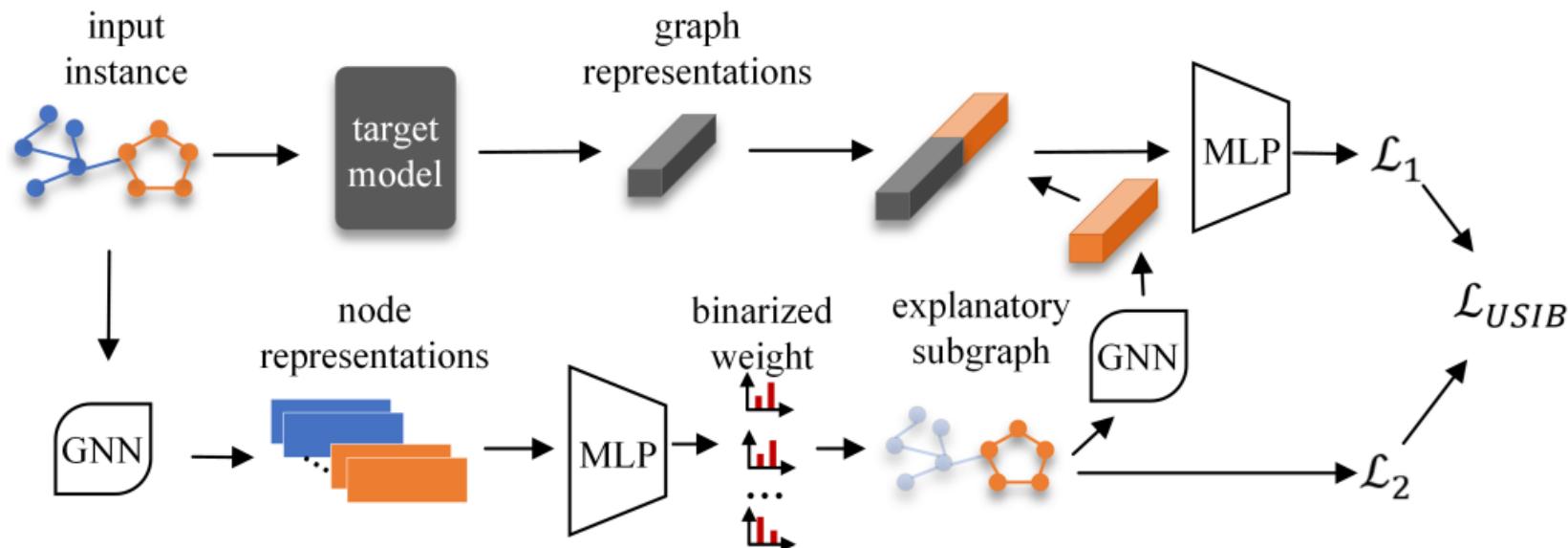
The interpretation with 7 leaf clusters.



The interpretation with 20 leaf clusters.

Learnable Interpreter

- Explain **unsupervised graph-level representations** learned by **GNNs**.
- Based on the **Information-Bottleneck** principle, to find the most informative yet compressed subgraph.
- Learn an explanation module (e.g., MLP) that outputs **edge weights**.



Qinghua Zeng, et al. "Towards Explanation for Unsupervised Graph-Level Representation Learning." arXiv preprint arXiv:2205.09934 (2022).

Outline

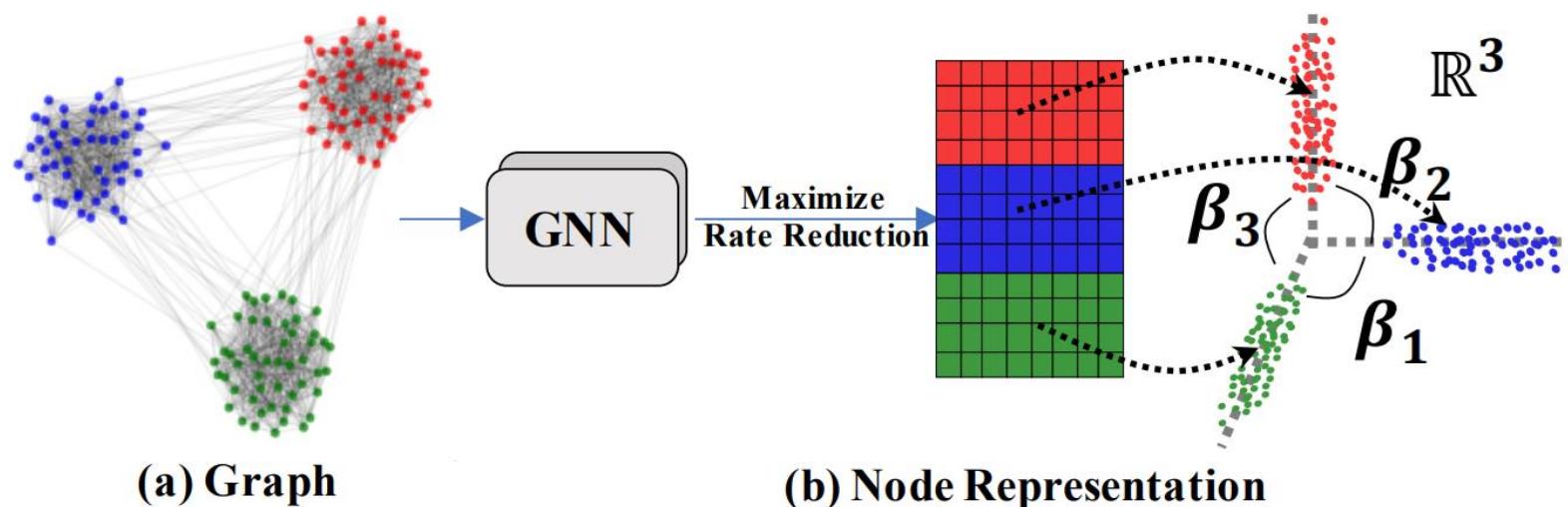
1. Background: Graph representation learning, Graph Neural Networks
2. Interpretability for Supervised Graph Models
 - *Approximation Methods*
 - *Perturbation Methods*
 - *Decomposition Methods*
 - *Generative Methods*
 - *Evaluation: Datasets & Metrics*
3. Interpretability for Unsupervised Graph Models
 - *Post-Hoc Interpretation*
 - *Intrinsic Interpretability in Graph Modeling*

Control Embedding Distribution

Traditional Methods: Model the **local similarity** of connected nodes.

Additional desirable properties for node representations:

- The whole representation should be **diverse**.
- The representation within groups should be similar but span their own subspaces.

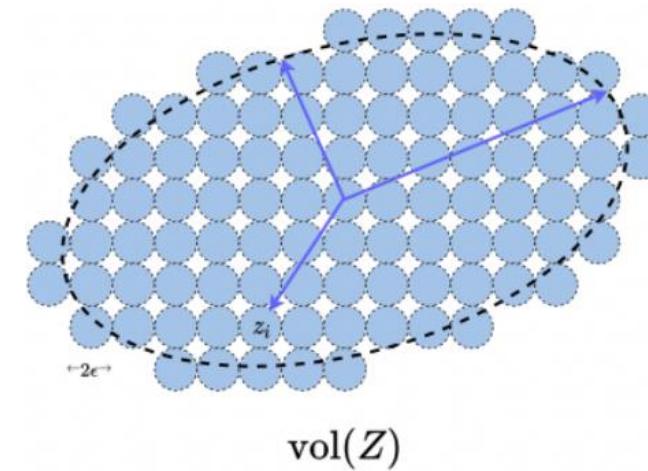


Coding Rate² (Ma et al., 2007)

Suppose we have a set of node representations $\mathbf{W} = (w_1, w_2, \dots, w_m)$, then the number of bits needed to encode the data \mathbf{W} is¹:

$$R(\mathbf{W}) \doteq \frac{1}{2} \log_2 \det\left(\mathbf{I} + \frac{n}{m\epsilon^2} \mathbf{W}\mathbf{W}^\top\right).$$

**$R(\mathbf{W})$ is an intrinsic measure
for the volume of \mathbf{W} .**



1: ϵ is the error allowable for encoding every vector w_i in \mathbf{W} .

2: This slide is largely based on Yi Ma slides at https://book-wright-ma.github.io/Lecture-Slides/Lecture_21_22.pdf

Geometric Graph Representation Learning (G²R)

Graph neural networks map the graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ to node representations \mathbf{Z} ,

$$\mathbf{A} \in \mathbb{R}^{N \times N}, \mathbf{X} \in \mathbb{R}^{D \times N} \xrightarrow{\text{GNN}(\mathbf{A}, \mathbf{X} | \theta)} \mathbf{Z} \in \mathbb{R}^{d \times N}.$$

Maximize the following objective function:

$$\begin{aligned} & \Delta R_{\mathcal{G}}(\mathbf{Z}, \mathbf{A}, \epsilon) \\ &= R_{\mathcal{G}}(\mathbf{Z}, \epsilon) - R_{\mathcal{G}}^c(\mathbf{Z}, \epsilon | \mathcal{A}) \\ &\doteq \frac{1}{2} \log \det \left(\mathbf{I} + \frac{d}{N\epsilon^2} \mathbf{Z} \mathbf{Z}^\top \right) - \frac{1}{d} \sum_{i=1}^N \frac{\text{tr}(\mathbf{A}_i)}{2N} \cdot \log \det \left(\mathbf{I} + \frac{d}{\text{tr}(\mathbf{A}_i)\epsilon^2} \mathbf{Z} \mathbf{A}_i \mathbf{Z}^\top \right) \end{aligned}$$

Larger $R_{\mathcal{G}}$ → more bits in representation → **diverse** representations.

Smaller $R_{\mathcal{G}}^c$ → less bits in representation → **similar representations within groups**.

Will Representation Learned by G²R (nearly) Orthogonal?

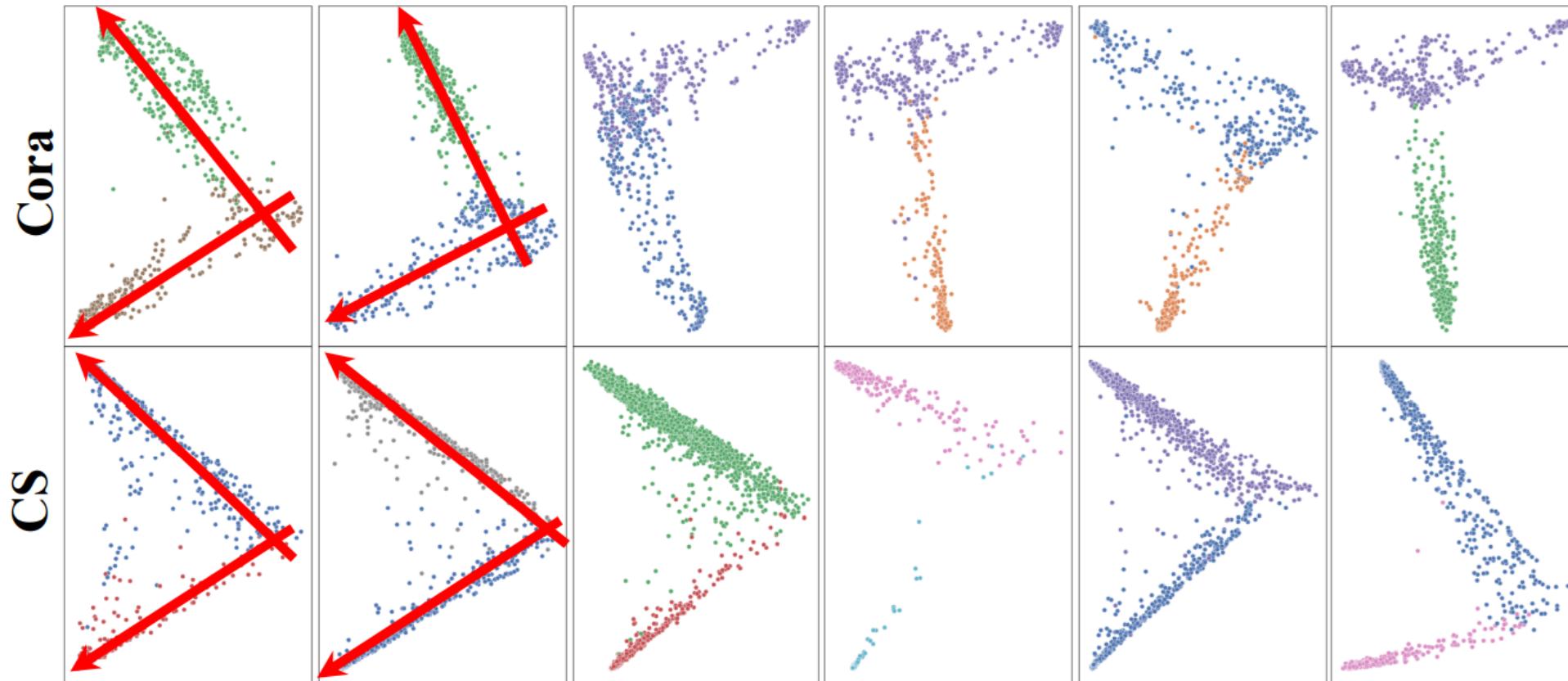


Figure: PCA visualization of learned representations.

The representations of nodes in different classes learned by G²R are nearly orthogonal to each other.

G²R Performance

Table: Performance comparison to unsupervised methods.

Statistic		Cora		CiteSeer		PubMed		CoraFull	CS	Physics	Computers	Photo
Metric	Feature	Public	Random	Public	Random	Public	Random	Random	Random	Random	Random	Random
Feature	X	58.90	60.19	58.69	61.70	69.96	73.90	40.06	88.14	87.49	67.48	59.52
PCA	X	57.91	59.90	58.31	60.00	69.74	74.00	38.46	88.59	87.66	72.65	57.45
SVD	X	58.57	60.21	58.10	60.80	69.89	73.79	38.64	88.55	87.98	68.17	60.98
isomap	X	40.19	44.60	18.20	18.90	62.41	63.90	4.21	73.68	82.84	72.66	44.00
LLE	X	29.34	36.70	18.26	21.80	52.82	54.00	5.70	72.23	81.35	45.29	35.37
DeepWalk	A	74.03	73.76	48.04	51.80	68.72	71.28	51.65	83.25	88.08	86.47	76.58
Node2vec	A	73.64	72.54	46.95	49.37	70.17	68.70	50.35	82.12	86.77	85.15	75.67
DeepWalk+F	X, A	77.36	77.62	64.30	66.96	69.65	71.84	54.63	83.34	88.15	<u>86.49</u>	65.97
Node2vec+F	X, A	75.44	76.84	63.22	66.75	70.6	69.12	54.00	82.20	86.86	85.15	65.01
GAE	X, A	73.68	74.30	58.21	59.69	76.16	<u>80.08</u>	42.54	88.88	91.01	37.72	48.72
VGAE	X, A	77.44	76.42	59.53	60.37	78.00	77.75	53.69	88.66	90.33	49.09	48.33
DGI	X, A	81.26	82.11	<u>69.50</u>	70.15	77.70	79.06	53.89	<u>91.22</u>	<u>92.12</u>	79.62	70.65
GRACE	X, A	80.46	80.36	68.72	68.04	<u>80.67</u>	OOM	53.95	90.04	OOM	81.94	70.38
GraphCL	X, A	<u>81.89</u>	81.12	68.40	69.67	OOM	81.41	OOM	OOM	OOM	79.90	OOM
GMI	X, A	80.28	<u>81.20</u>	65.99	<u>70.50</u>	OOM	OOM	OOM	OOM	OOM	52.36	OOM
G ² R(ours)	X, A	82.58	83.32	71.2	70.66	81.69	81.69	59.70	92.64	94.93	82.24	90.68

The G²R design even benefits downstream application performance.

Control the Meaning of Embeddings



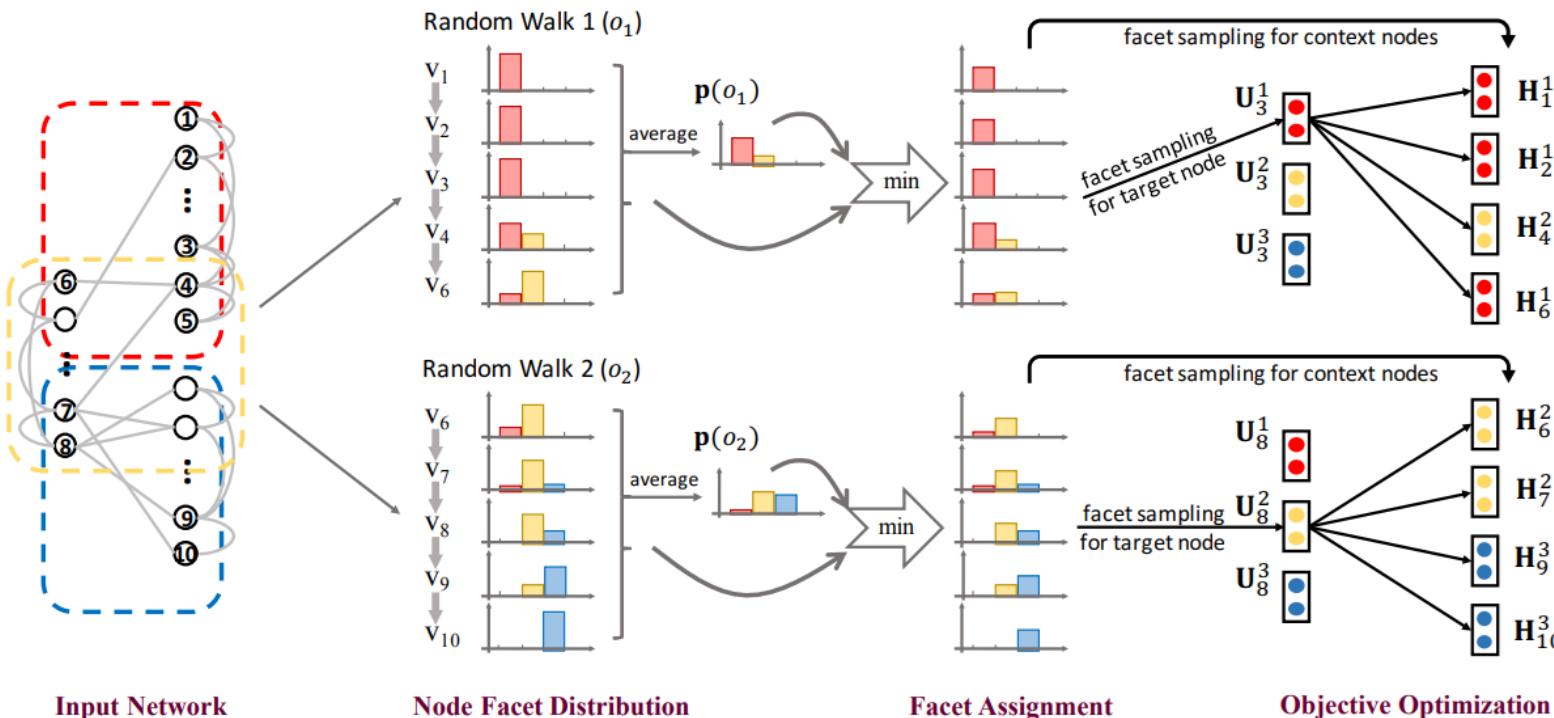
- Is a single vector enough?
- Each node embedding has multiple segments (i.e., facets).
- The meaning of each segment is known.
- Benefit downstream applications such as social mining and recommender systems.

Liu, N., Tan, Q., Li, Y., Yang, H., Zhou, J. & Hu, X.. *Is a Single Vector Enough? Exploring Node Polysemy for Network Embedding*. KDD. 2019.

Park, Chanyoung, Carl Yang, Qi Zhu, Donghyun Kim, Hwanjo Yu, and Jiawei Han. "Unsupervised differentiable multi-aspect network embedding." KDD. 2020.

Polysemous Node Embedding

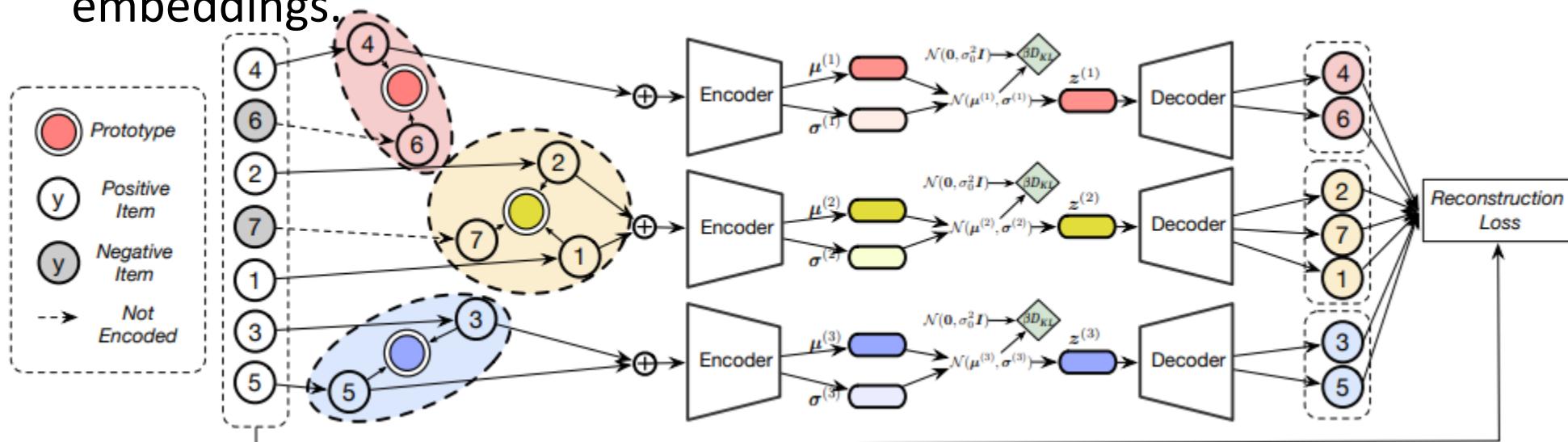
- Each node has **multiple embedding segments** (facets).
- Motivated by word “**polysemy**” in natural language.
- Given the context, only update the embedding segments of activated facets.



Liu, N., Tan, Q., Li, Y., Yang, H., Zhou, J. & Hu, X.. Is a Single Vector Enough? Exploring Node Polysemy for Network Embedding. KDD. 2019.

Disentangled Node Embedding

- Scenario: Recommender Systems.
- Encourage embedding independence for items in different groups.
- An end-to-end learning framework.
 - Each group is associated with a “prototype”, which is trained along with embeddings.



Part 4: Text-based Model Interpretation

Outline

1. Background: text-based model Interpretation
2. Interpretation for text-based models
 - *Feature importance*
 - *Surrogate model*
 - *Example-driven explanations*
3. Application
4. Summarization

Background

Why focus on these three types of methods?

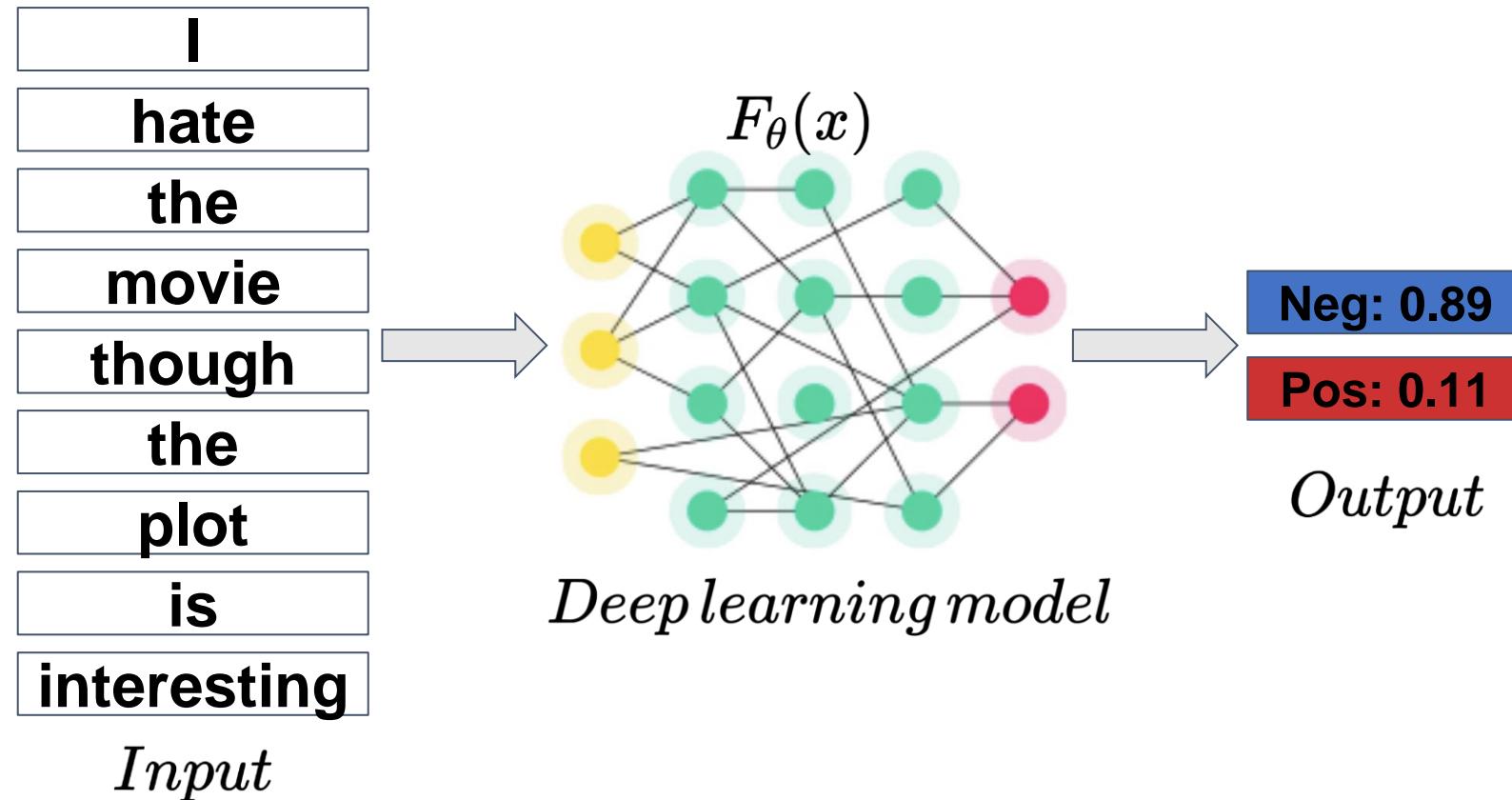
- Model-agnostic
- Fast, easy-to-compute
- Faithful to underlying model

Benefits: Can answer critical questions like:

- Why did my model fail on this particular input?
- What is the impact of this particular training point?

Background

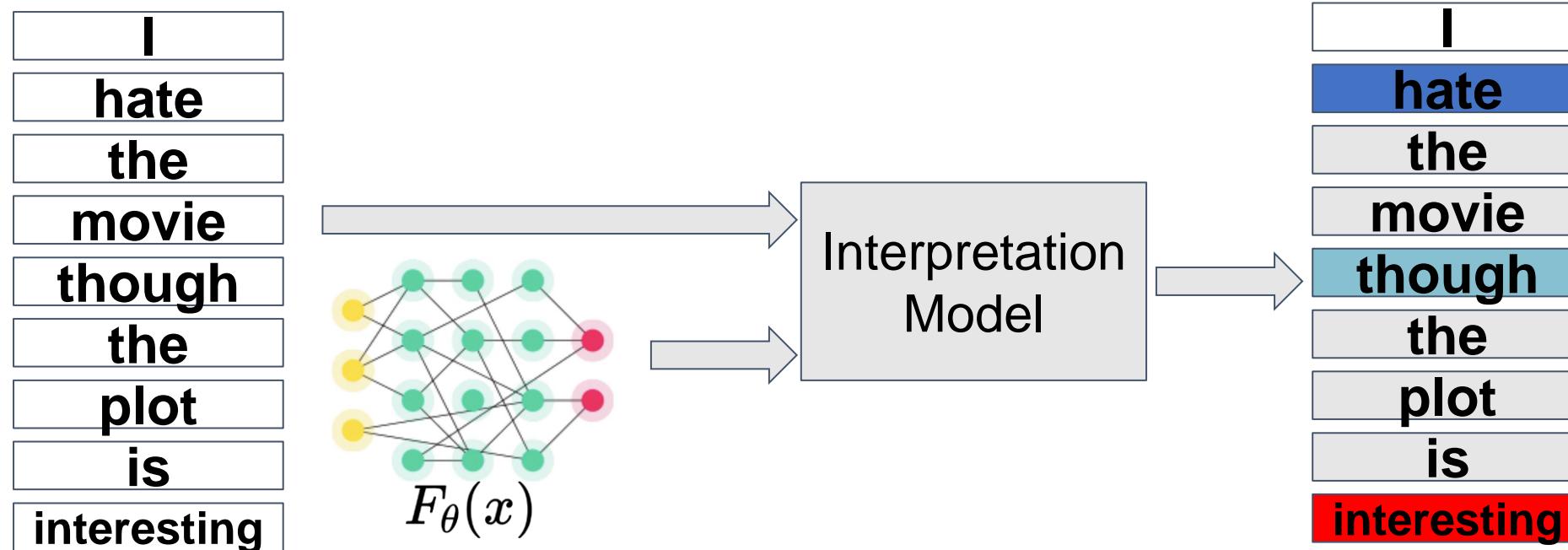
What is text-based interpretation?



Background

What is text-based interpretation?

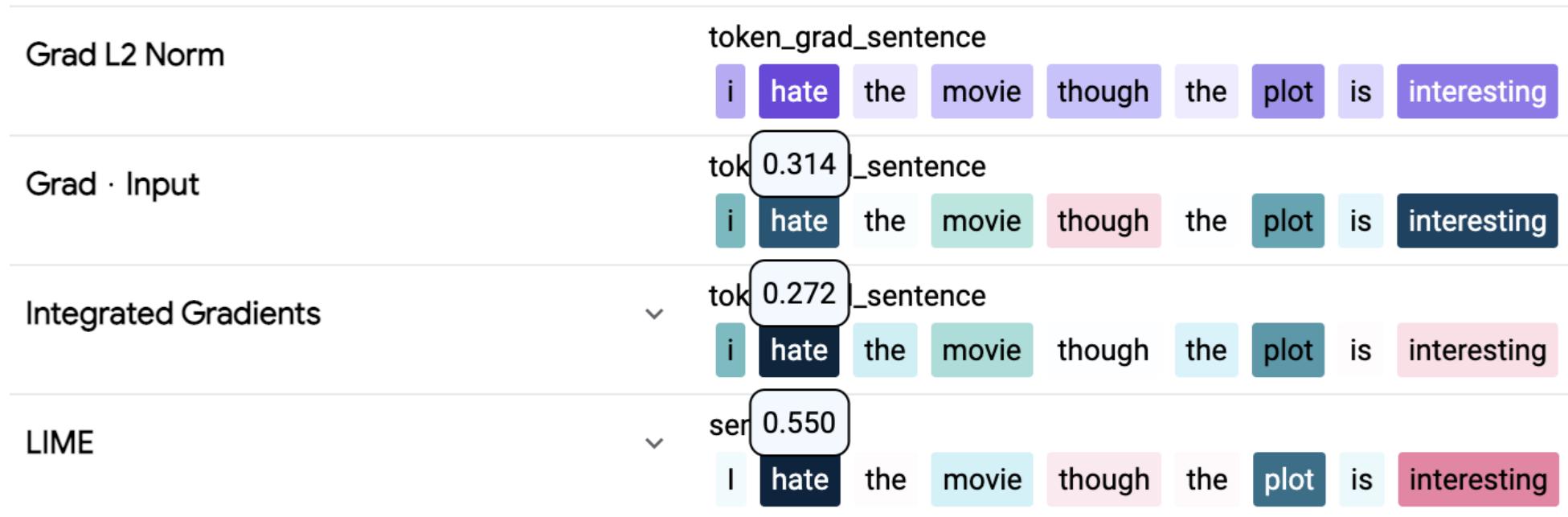
- Answer questions like:
 - Which words cause the DNN model classifying it as negative?



Background

What is text-based interpretation?

- Can help us better understand the target model.



Background

What is text-based interpretation?

- Can help us better understand the target model.

Simple Gradients Visualization

See saliency map interpretations generated by [visualizing the gradient](#).

Saliency Map:

[CLS] The [MASK] rushed to the emergency room to see her patient . [SEP]

Mask 1 Predictions:

47.1% **nurse**

16.4% **woman**

10.0% **doctor**

3.4% **mother**

3.0% **girl**

Outline

1. Background: text-based model Interpretation
2. Interpretation for text-based models
 - *Feature importance*
 - *Surrogate model*
 - *Example-driven explanations*
3. Application
4. Summarization

Feature Importance

LIME:

- Look at model's predictions for a bunch of nearby inputs.
- Closer points are more important than further points.
- Fit a linear model. Its weights are the feature importances.

The movie is ~~mediocre~~, maybe even bad.

The movie is ~~mediocre~~, maybe even bad.

The movie is ~~mediocre~~, maybe even ~~bad~~.

The movie is mediocre, ~~maybe even ~~bad~~~~.

The movie is ~~mediocre~~, maybe even ~~bad~~.

The movie is mediocre, maybe even bad.

Feature Importance

Leave-one-out:

- Simplest method is leave-one-out:
 - define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed

SQuAD

Context: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.

Question:

- (0.90, 0.89) Where did the Broncos practice for the Super Bowl ?
- (0.92, 0.88) Where did the practice for the Super Bowl ?
- (0.91, 0.88) Where did practice for the Super Bowl ?
- (0.92, 0.89) Where did practice the Super Bowl ?
- (0.94, 0.90) Where did practice the Super ?
- (0.93, 0.90) Where did practice Super ?
- (0.40, 0.50) did practice Super ?

SQuAD

Context: QuickBooks sponsored a “Small Business Big Game” contest, in which Death Wish Coffee had a 30-second commercial aired free of charge courtesy of QuickBooks. Death Wish Coffee beat out nine other contenders from across the United States for the free advertisement.

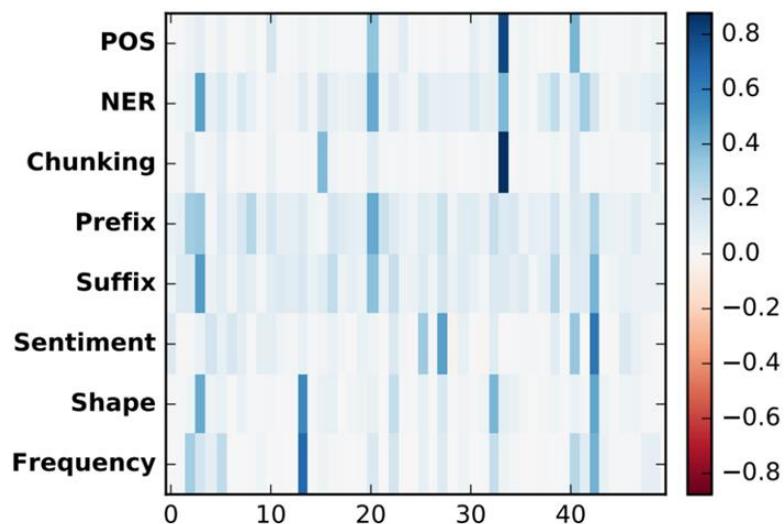
Question:

- What company won free advertisement due to QuickBooks contest ?
- What company won free advertisement due to QuickBooks ?
- What company won free advertisement due to ?
- What company won free due to ?
- What won free due to ?
- What won due to ?
- What won due to to
- What won due
- What won
- What

Feature Importance

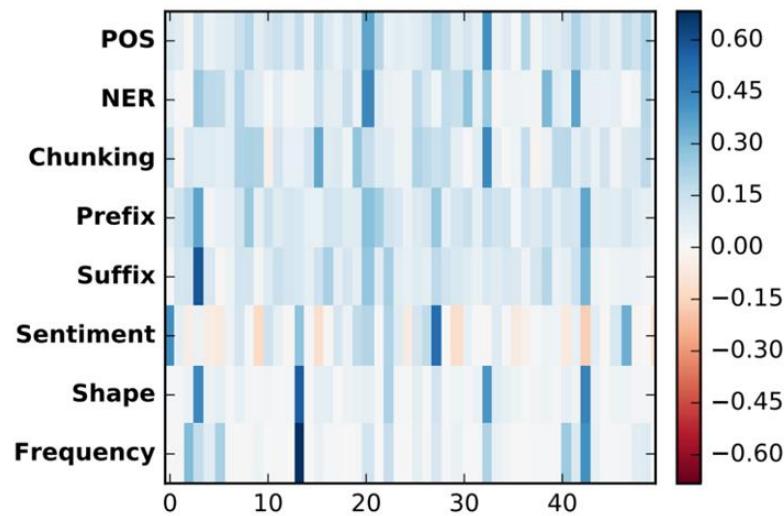
Leave-one-out: dimension importance

- define importance as drop in prediction confidence when a feature (e.g., token, phrase) is removed.
- Importance score for dimension d is: $I(d) = \frac{1}{|E|} \sum_{e \in E} \frac{S(e, c) - S(e, c, \neg d)}{S(e, c)}$



(a) Word2vec, no dropout.

$$I(d) = \frac{1}{|E|} \sum_{e \in E} \frac{S(e, c) - S(e, c, \neg d)}{S(e, c)}$$



(b) Word2vec, with dropout.

Heatmap of word vector dimension importance I(d)

Feature Importance

SHAP:

- Relies on Shapley values defined in Game Theory.
- It is a local model-agnostic interpretation method.

0th instance:

	base value				f(x)
-2.036220	-0.187151	1.661918	3.510987	5.360056	6.721336125

well , and i was sold out) was overcome so that it can toy with our emotions . it that i was rel

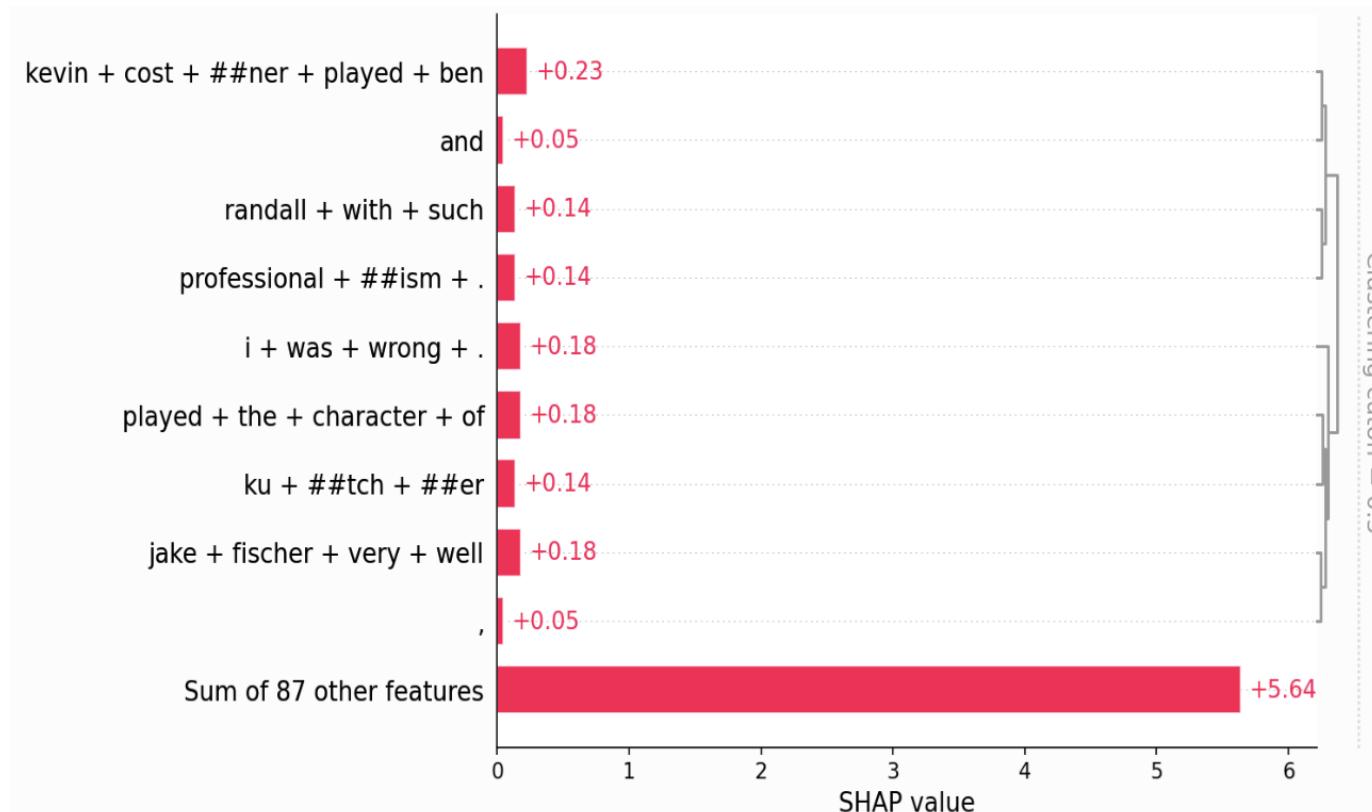
i went and saw this movie last night after being coaxed to by a few friends of mine . i ' ll admit that i was reluctant to see it because from what i knew of ashton kutcher he was only able to do comedy . i was wrong . kutcher played the character of jake fischer very well , and kevin costner played ben randall with such professionalism . the sign of a good movie is that it can toy with our emotions . this one did exactly that . the entire theater (which was sold out) was overcome by laughter during the

Lundberg, Scott M., et al. "A unified approach to interpreting model predictions." *Advances in neural information processing systems*, 2017.

Feature Importance

SHAP:

- Relies on Shapley values defined in Game Theory.
- is a local diagnostic



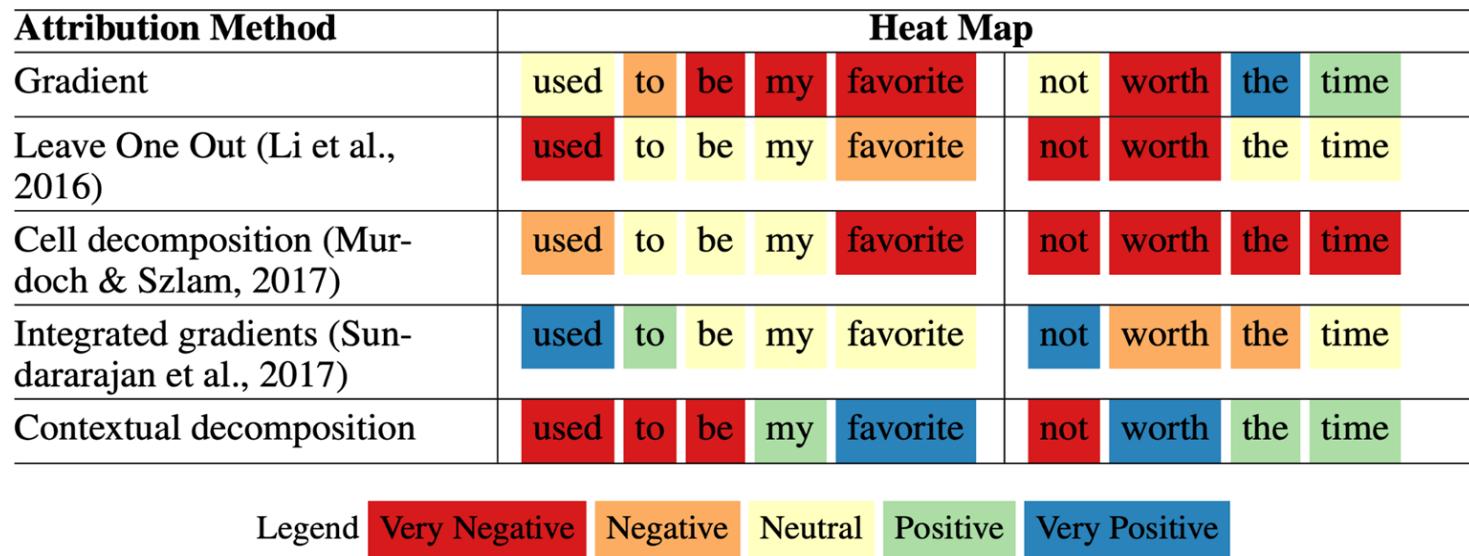
Lundberg, Scott M., et al. "A unified approach to interpreting model predictions." *Advances in neural information processing systems*, 2017.

Feature Importance

Contextual Decomposition - word contribution

Given a sentence, it provides a decomposition of the output of a trained LSTM model as a sum of two contributions.

1. Resulting solely from the given phrase
2. Involving at least in part, elements outside of the phrase



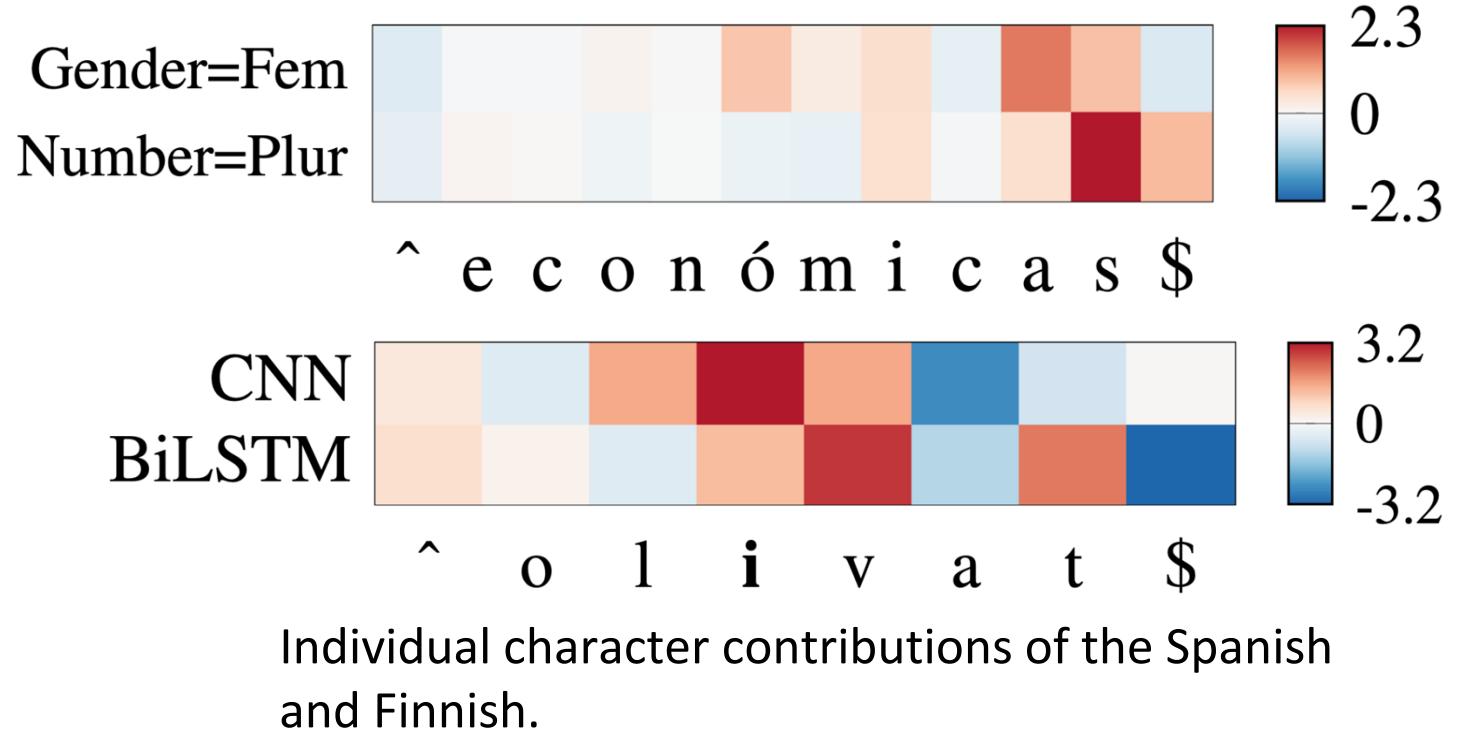
Heat maps for portion of yelp review with different attribution techniques. Only CD captures that "favorite" is positive.

Feature Importance

Contextual Decomposition - character contribution

The output of network is decomposed into two parts:

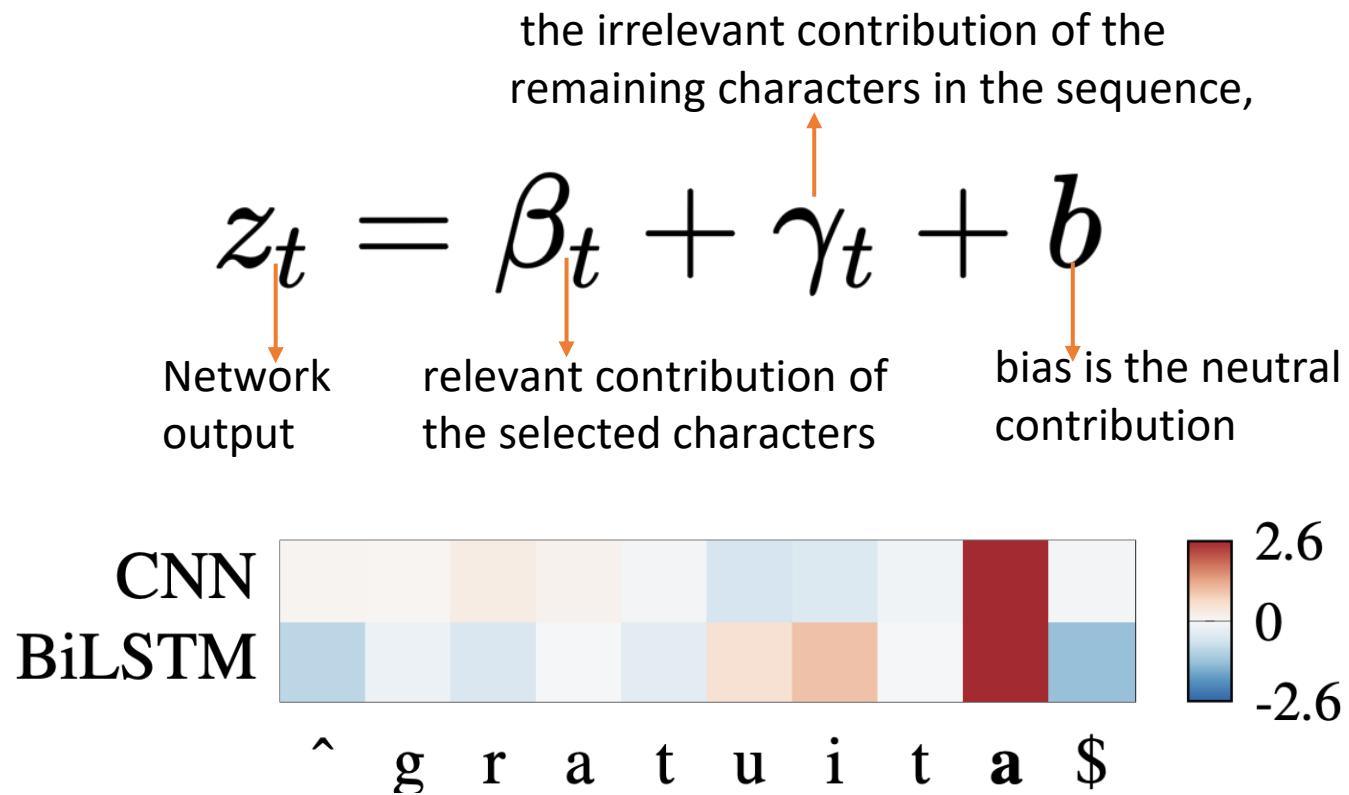
- Relevant contribution
- Irrelevant contribution



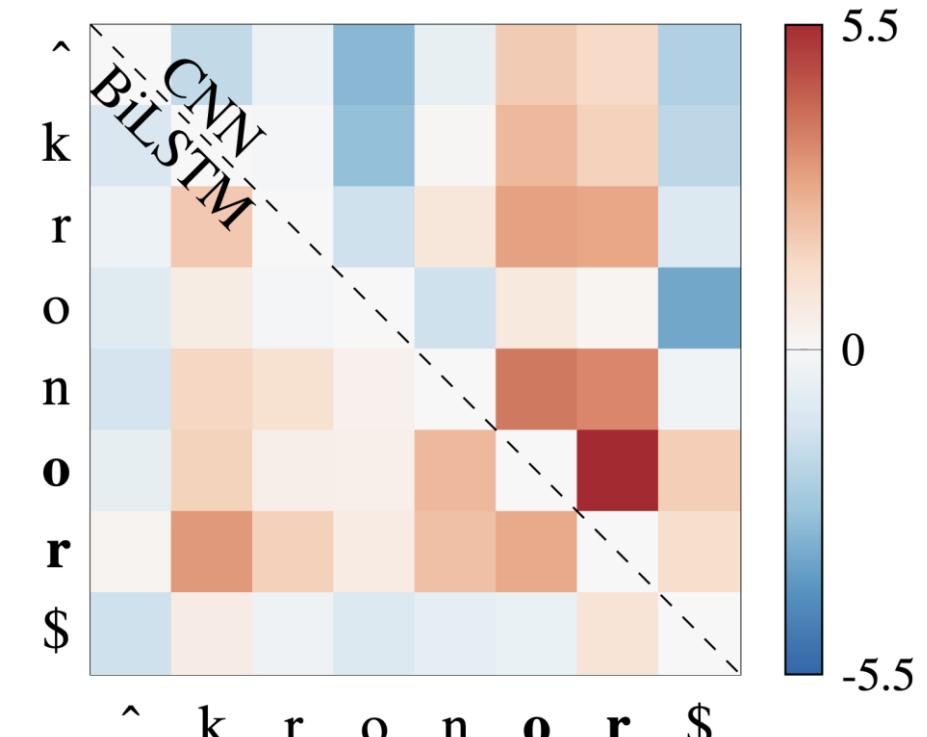
Godin, Frédéric, et al. "Explaining Character-Aware Neural Networks for Word-Level Prediction: Do They Discover Linguistic Rules?." arXiv ,2018.

Feature Importance

Contextual Decomposition - character contribution



Example of Spanish. Word (adjective): gratuita (free), target: Gender=Fem.



Character-level contributions for predicting a particular class.

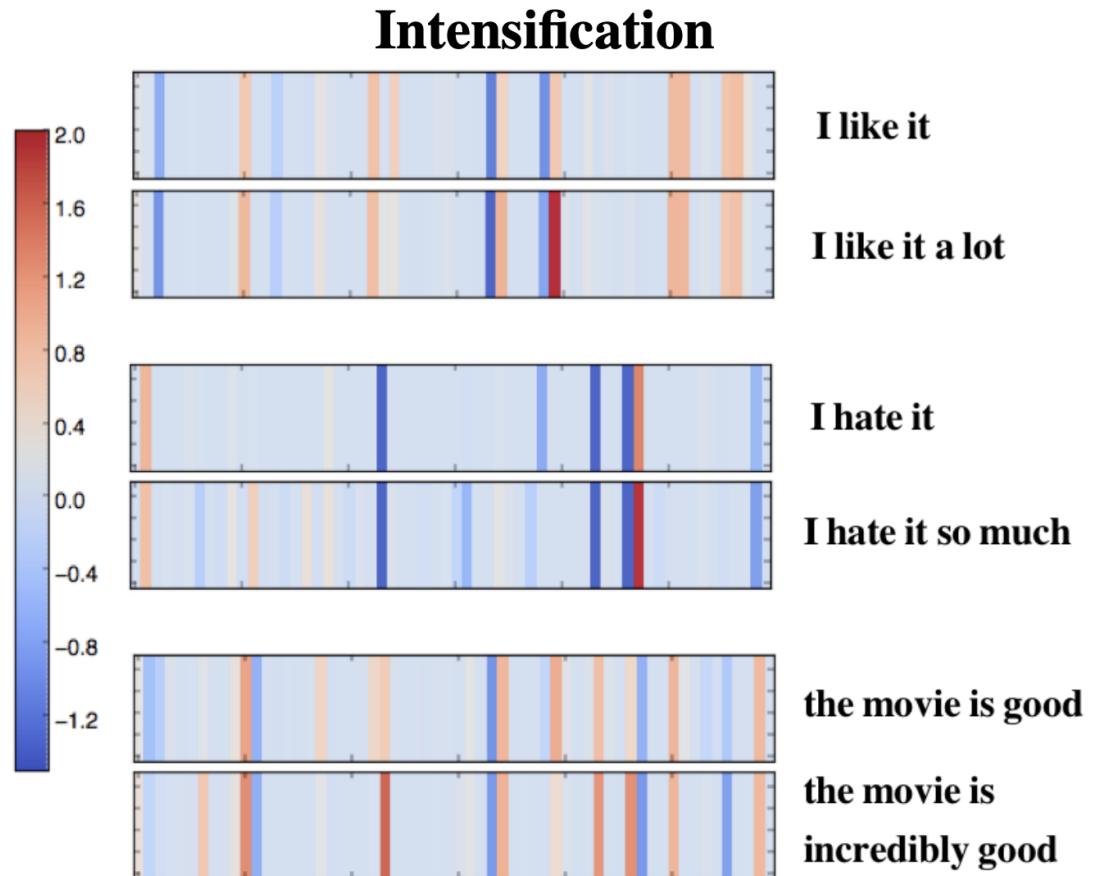
Feature Importance

First-order saliency:

$$S_c(e) \approx w(e)^T e + b$$

$$w(e) = \frac{\partial(S_c)}{\partial e} \Big|_e$$

$$S(e) \approx |w(e)|$$



Visualizing intensification. Each vertical bar shows the value of one dimension in the final sentence/phrase representation after compositions.

Outline

1. Background: text-based model Interpretation

2. Interpretation for text-based models

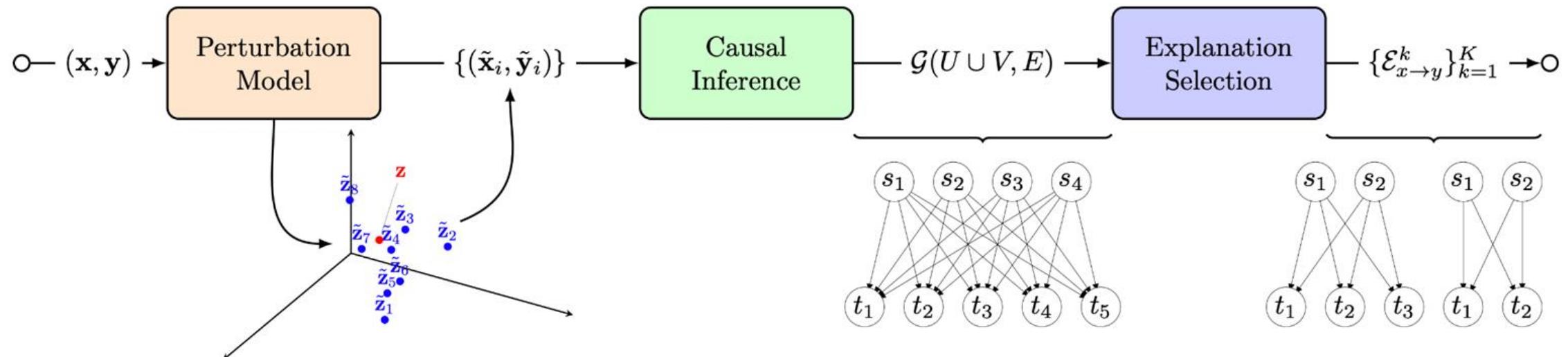
- *Feature importance*
- *Surrogate model*
- *Example-driven explanations*

3. Application

4. Summarization

Surrogate model

Causal framework



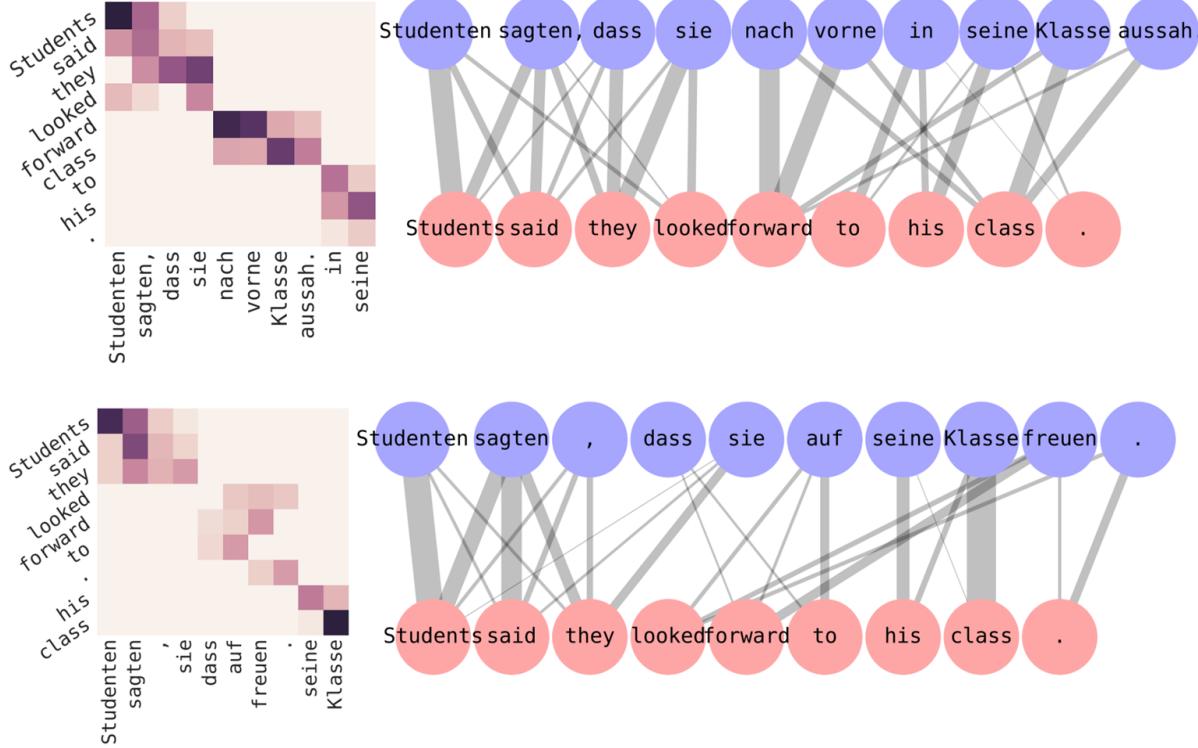
3 steps:

1. Generate perturbed versions of inputs
2. Use the perturbed inputs to estimate a causal graph model
3. Generate explanations(Subgraphs)

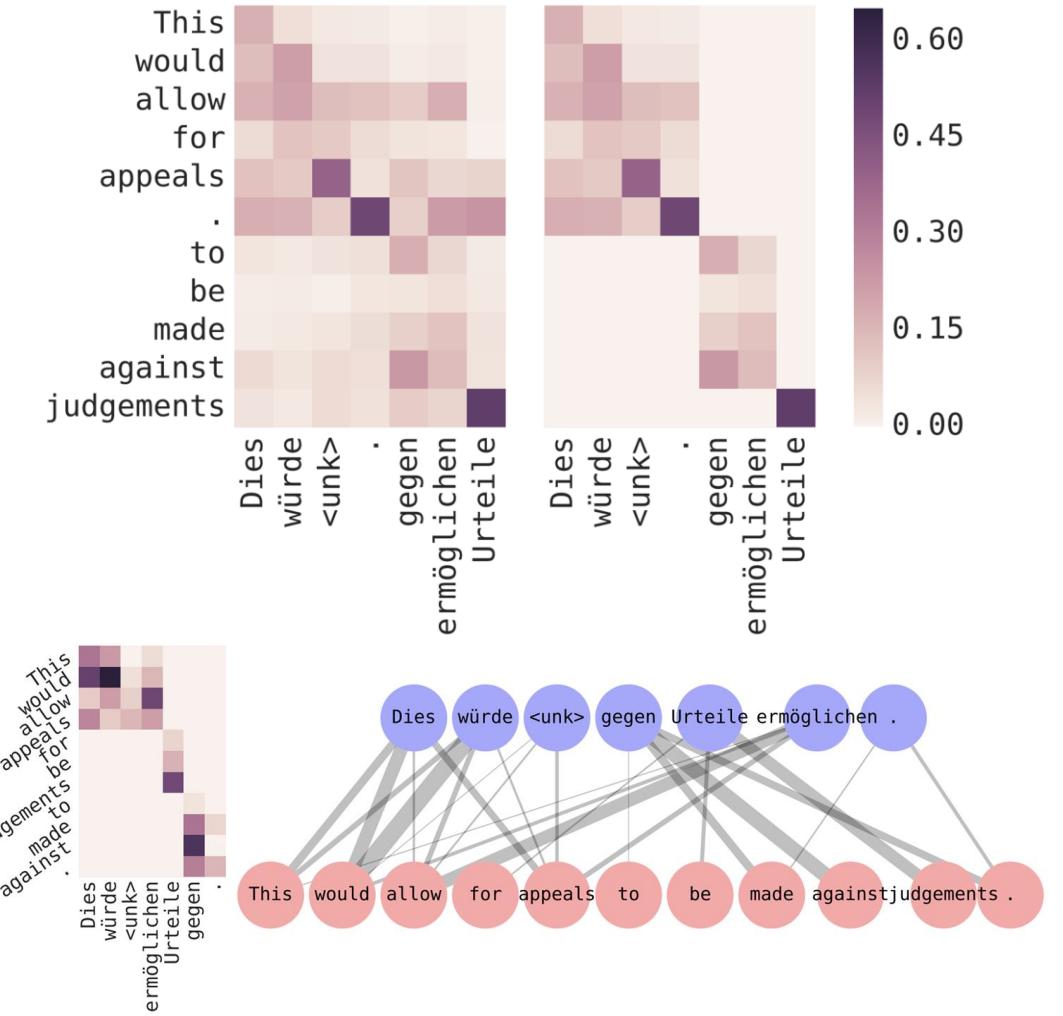
Alvarez-Melis, David, et al. "A causal framework for explaining the predictions of black-box sequence-to-sequence models." arXiv, 2017.

Surrogate model

Causal framework



Explanations for the predictions of three Black-Box translators: Azure (top), NMT (middle) and human (bottom).



Top: Original and clustered attention matrix of the NMT system for a given translation. Bottom: Dependency estimates and explanation graph generated by SOCRAT with $S = 100$

Alvarez-Melis, David, et al. "A causal framework for explaining the predictions of black-box sequence-to-sequence models." arXiv, 2017.

Example-driven

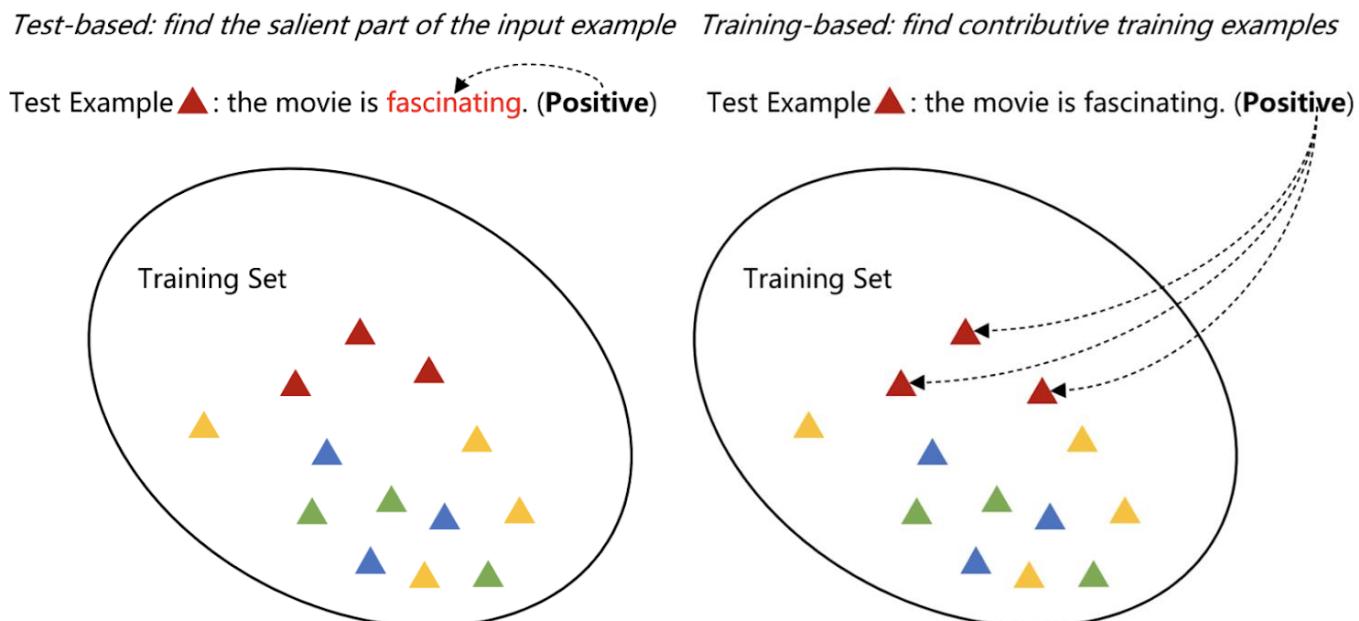
Find influential examples:

$$\theta = \arg \min_{\theta} \frac{1}{n} \sum_{z_i} L(z_i; \theta)$$
$$z_i = (\mathbf{x}_i, y_i)$$

The importance of z_i :

- is measured by the change of θ when z_i is removed from the training set, which is:

$$\theta_{-z} - \theta$$



Outline

1. Background: text-based model Interpretation

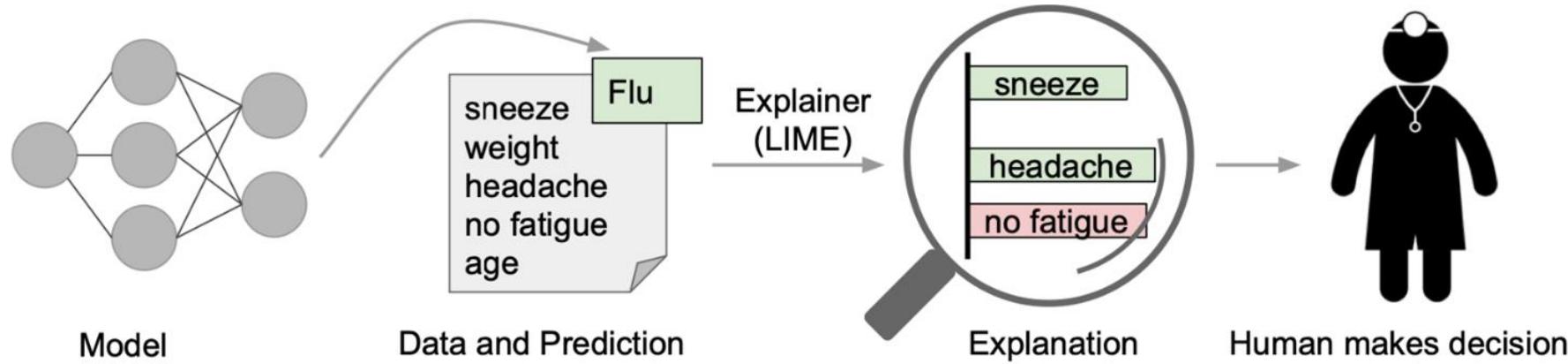
2. Interpretation for text-based models

- *Feature importance*
- *Surrogate model*
- *Example-driven explanations*

3. *Application*

4. *Summarization*

Medical Application: LIME for Patient Diagnosis



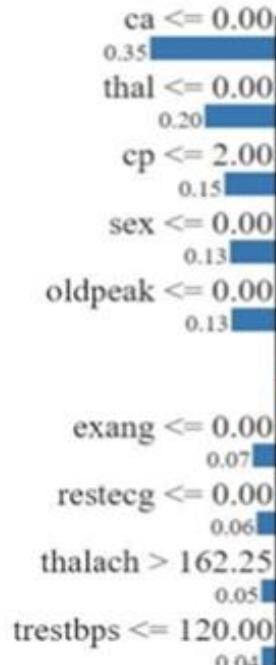
- A model predicts that a patient has the flu.
- LIME highlights the symptoms in the patient's history that led to the prediction. Sneeze and headache are portrayed as contributing to the “flu” prediction, while “no fatigue” is evidence against it.
- With these results, a doctor can make an informed decision about whether to trust the model’s prediction.

Ribeiro, Marco Tulio, et al. "Why should i trust you?" Explaining the predictions of any classifier." SIGKDD, 2016.

Medical Application: Heart Disease dataset with LIME and SHAP

Explainable AI meets Healthcare: A Study on Heart Disease Dataset

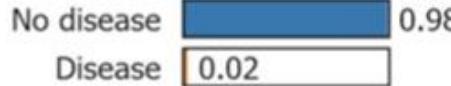
No disease



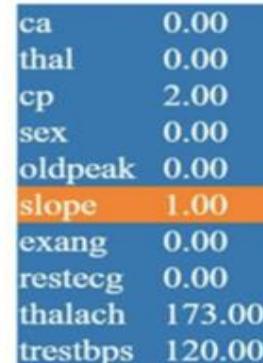
Disease

Predicted: [0.9808606 0.0191394]

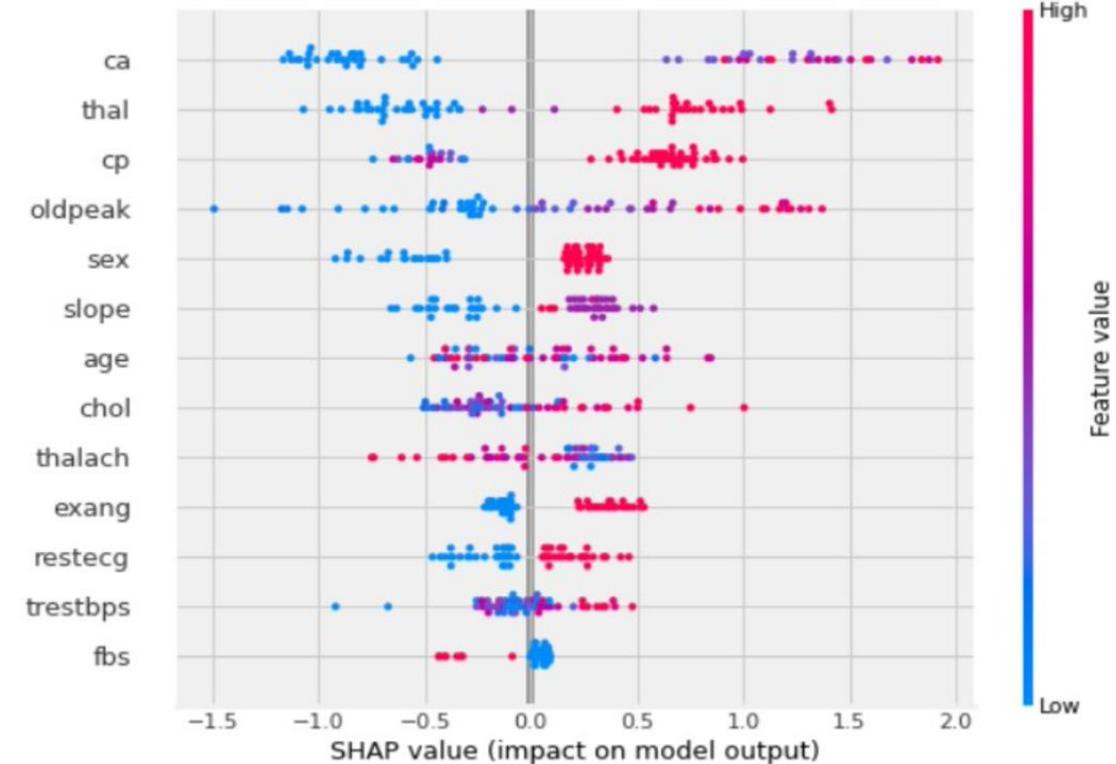
Prediction probabilities



Feature Value



Explanations generated by LIME



Explanations generated by SHAP summary plot

Summarization

Text-based explanation has its uniqueness:

- Cannot directly apply what have been built for image interpretations.
- Words are discrete
- Need to define the neighbor of a word
- Reliable metrics are needed

Part 5: Deep Reinforcement Learning Interpretation

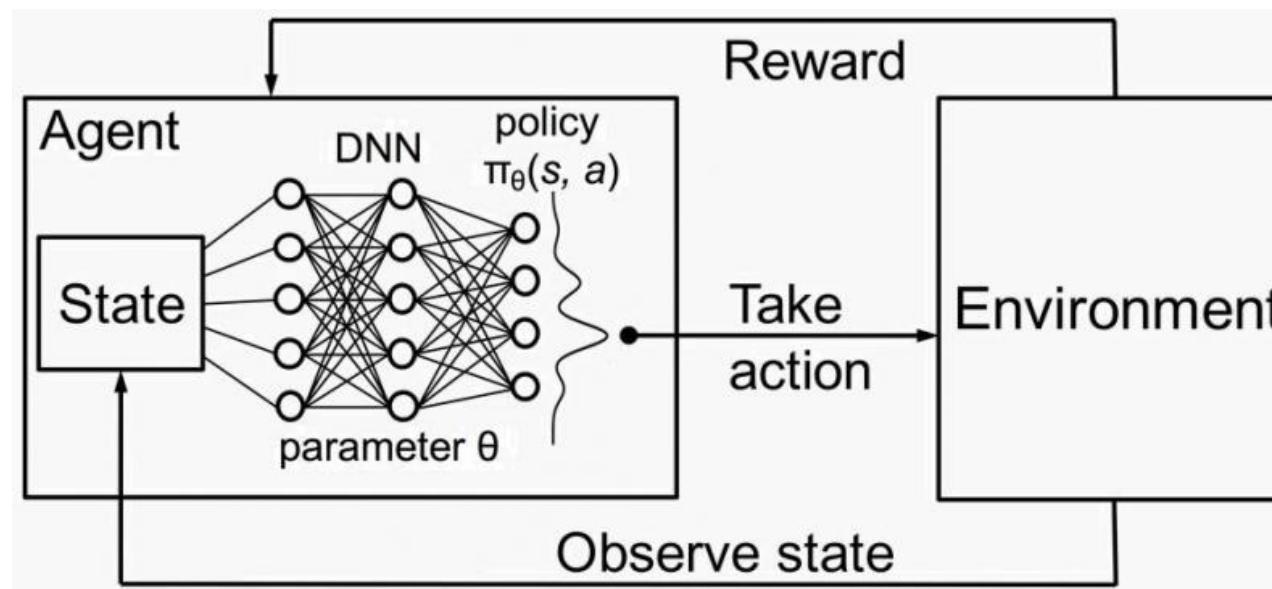
Outline

1. Background: Deep Reinforcement learning interpretation
2. Models for Reinforcement learning Interpretation
 - *Decision tree-based interpretation*
 - *Saliency-based interpretation*
 - *Causal model*
3. *Application*
4. *Summarization*

Background

Characteristics of deep reinforcement learning interpretation

- The underlying models are neural networks
- Agents have complex strategies
- Agents interact with each others



Outline

1. Background: Deep Reinforcement learning interpretation
2. Models for Reinforcement learning Interpretation
 - *Decision tree-based interpretation*
 - *Saliency-based interpretation*
 - *Causal model*
3. *Application*
4. *Summarization*

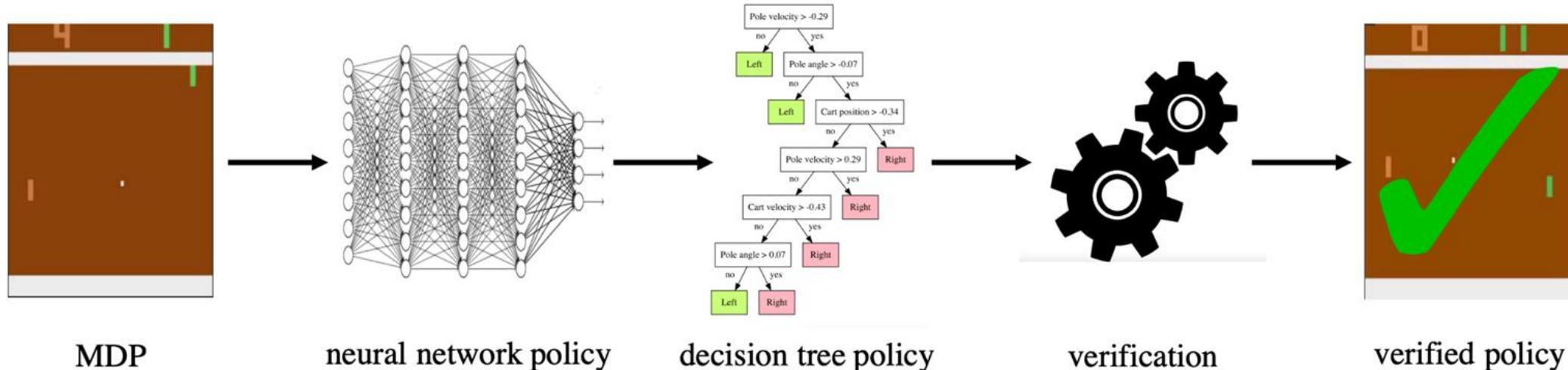
Decision Tree based

Why use decision tree-based interpretations?

- They are nonparametric (can represent very complex policies), and
- They are highly structured, making them easy to verify

Idea:

- Use imitation learning to extract policy
- Supervised learning is used to train a decision tree policy



Liu, Guiliang, et al. "Learning Tree Interpretation from Object Representation for Deep Reinforcement Learning." *Advances in Neural Information Processing Systems*, 2021.

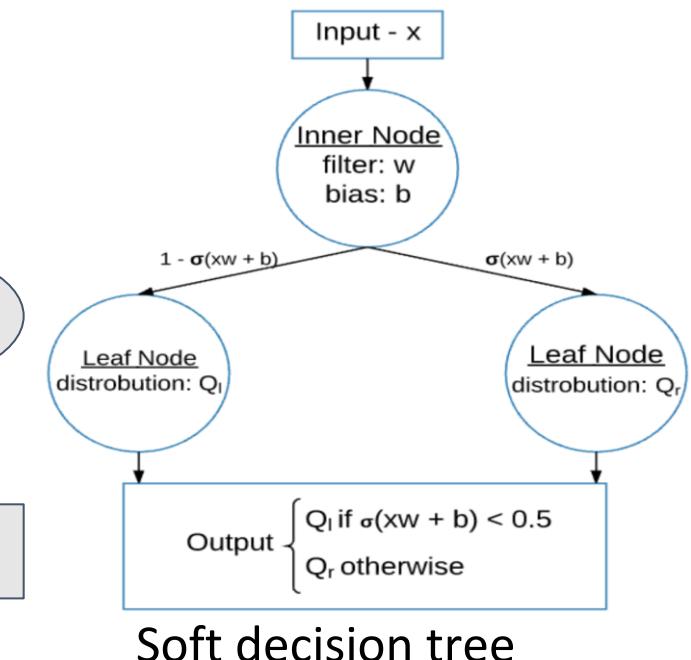
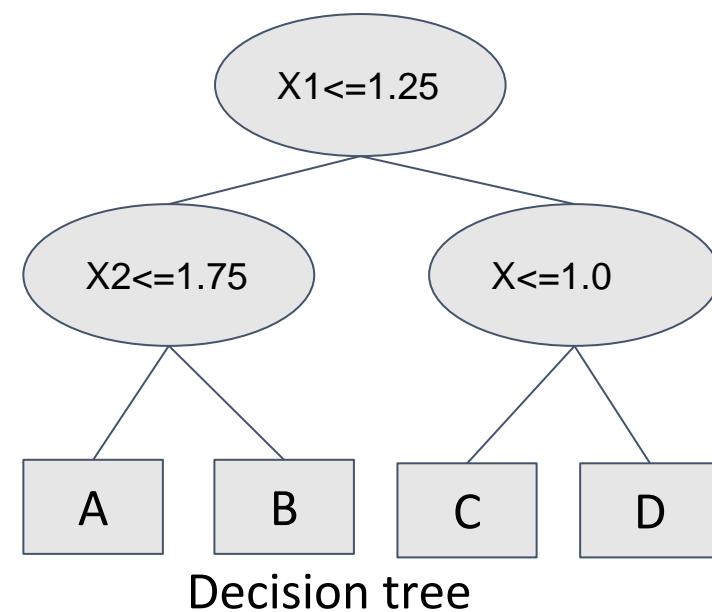
Decision Tree based

Soft Decision Tree:

- Traditional decision tree branches nodes by fixed thresholds.
- Soft decision tree branches nodes by a probability.

$$\text{Right: } p_i(x) = \sigma(\beta(xw_i + b_i))$$

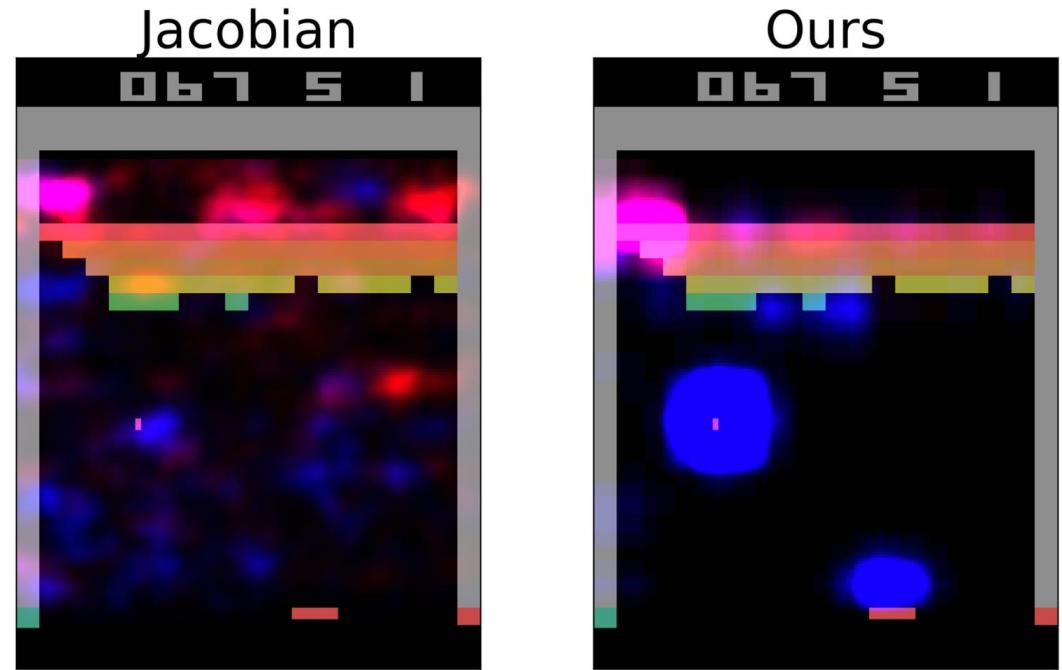
$$\text{Left: } 1 - p_i(x)$$



Barros, Rodrigo Coelho, et al. "A survey of evolutionary algorithms for decision-tree induction." *IEEE Transactions on Systems, Man, and Cybernetics*, 2011
Frosst, Nicholas, and Geoffrey Hinton. "Distilling a neural network into a soft decision tree." *arXiv*, 2017.

Saliency Map on A3C

- Highlight parts of input that are important for agent to take a specific action.
- Perturbative-based saliency methods.
- Saliency measures how a model's output changes when some of the input information is removed.



Red indicates saliency for the critic; blue is saliency for the actor.

Saliency Map on A3C

- Perturbation of image centered at image coordinate (i, j) is:

$$\Phi(I_t, i, j) = I_t \odot (1 - M(i, j)) + A(I_t, \sigma_A) \odot M(i, j)$$

Perturbed Image

Processed Frame

Mask

Blurred Frame

- Saliency metric for image location (i, j) at time t is:

$$S_{V^\pi}(t, i, j) = \frac{1}{2} \|V^\pi(I_{1:t}) - V^\pi(I'_{1:t})\|^2.$$

Value estimate

Saliency Map on A3C

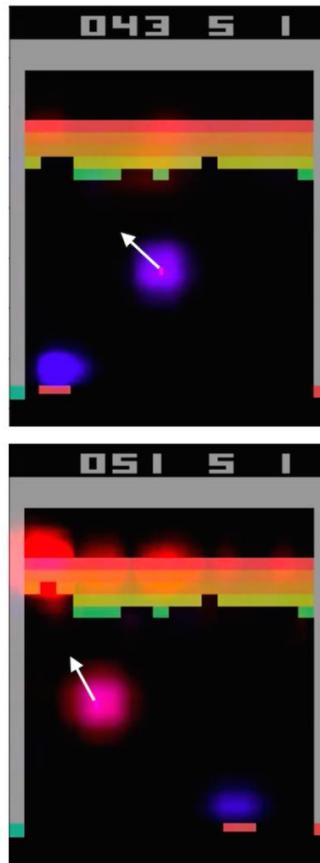
- Visualizing strong Atari 2600 policies using an actor-critic network.
- The actor's saliency map is blue and the critic's saliency map is red.
- White arrows denote motion of the ball.



(a) Pong: kill shot



(b) SpaceInvaders: aiming

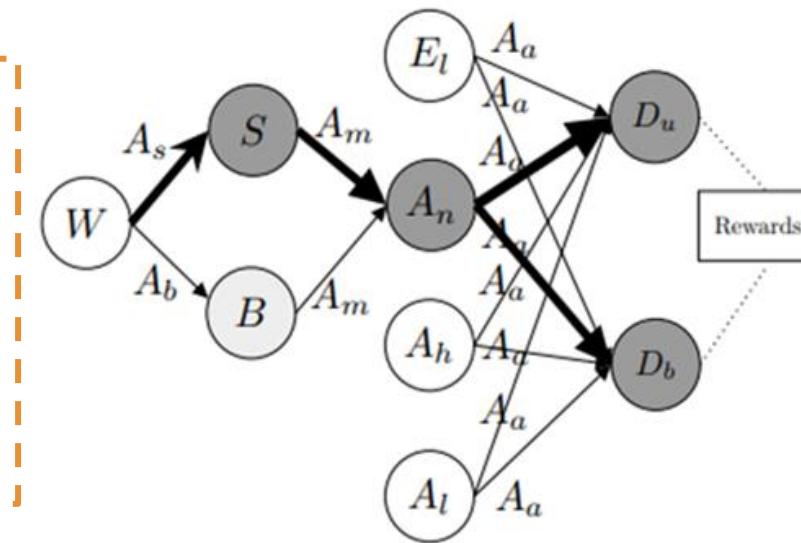


(c) Breakout: tunneling

Causal Model

Action Influence Model:

- Use causal models for explanations.
 - Generate explanations for *why* and *why not* question.



Action influence graph of a Starcraft II agent.

State variables:

- W - Worker number
- S - Supply depot number
- B - barracks number
- E - enemy location
- A_n - Ally unit number
- A_h - Ally unit health
- A_l - Ally unit location
- D_u - Destroyed units
- D_b - Destroyed buildings

Actions:

- A_s - build supply depot
- A_b - build barracks
- A_m - train offensive unit
- A_a - attack

Causal Model

Example:

Actual Instantiation:

$$m = (W = 12, S = 1, B = 2, A_n = 22, D_u = 10, D_b = 7)$$

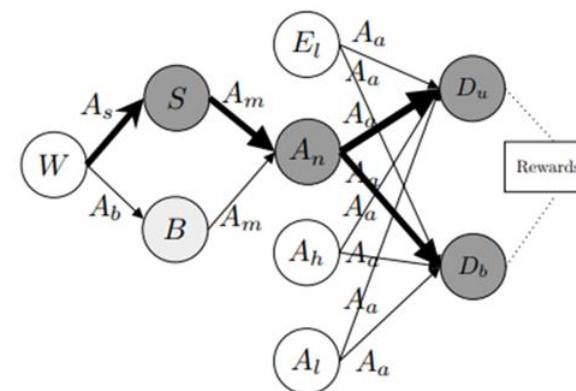
Counterfactual Instantiation:

$$m' = (W = 12, S = 3, B = 2, A_n = 22, D_u = 10, D_b = 7)$$

- Contrast [S = 1] with [S = 3] to obtain the explanations.

Question Explanation:

- Why not build_barracks (A_b) ?
- Because it is more desirable to do action build_supply_depot (A_s) to have more Supply Depots (S) as the goal is to have more Destroyed Units (D_u) and Destroyed buildings (D_b)



State variables:
W - Worker number
S - Supply depot number
B - Barracks number
E - enemy location
 A_n - Ally unit number
 A_h - Ally unit health
 A_l - Ally unit location
 D_u - Destroyed units
 D_b - Destroyed buildings

Actions:
 A_s - build supply depot
 A_b - build barracks
 A_m - train offensive unit
 A_a - attack

Figure 1: Action influence graph of a Starcraft II agent

Outline

1. Background: Deep Reinforcement learning interpretation
2. Models for Reinforcement learning Interpretation
 - *Decision tree-based interpretation*
 - *Saliency-based interpretation*
 - *Causal model*
3. *Application*
4. *Summarization*

Medical Application

Extension from previous work:

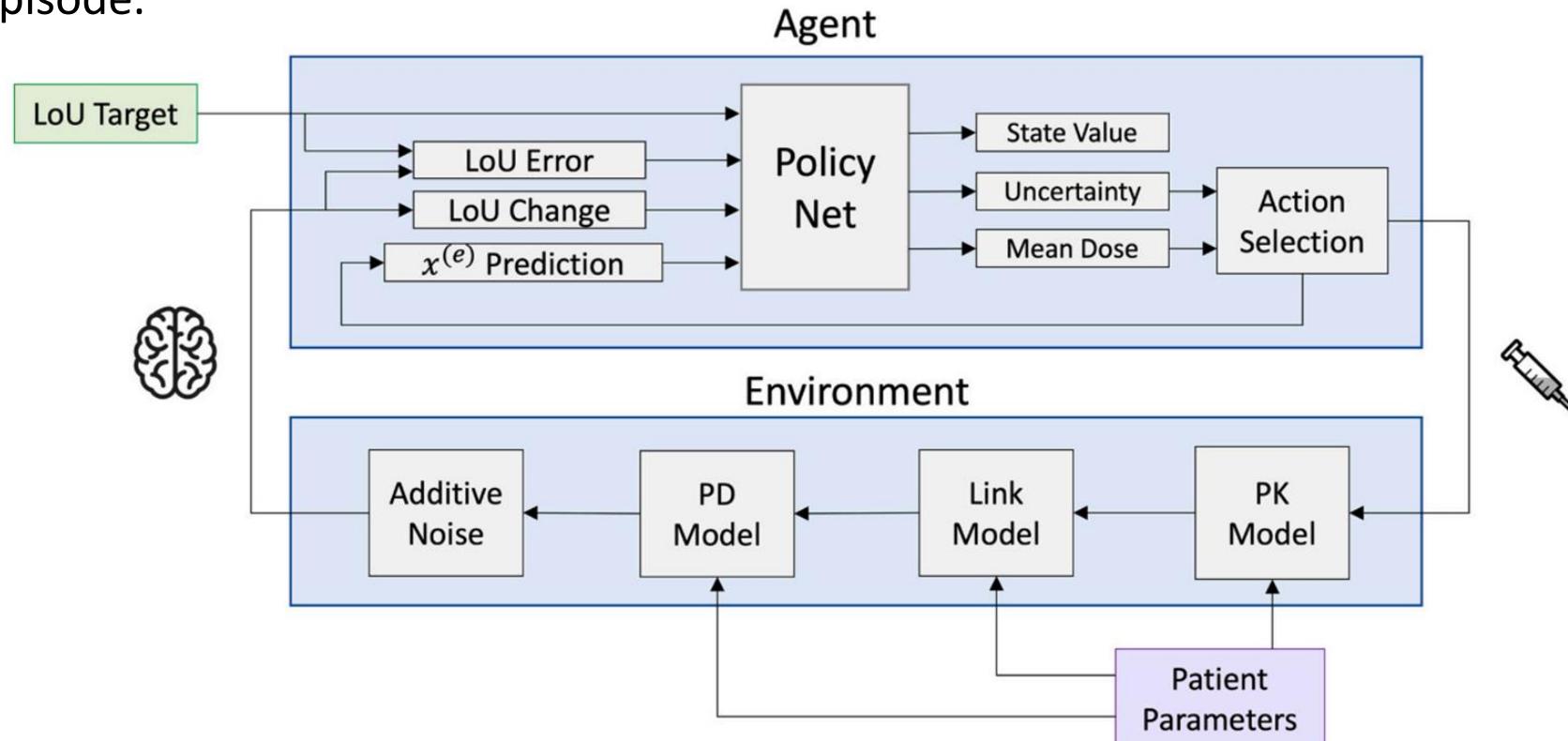
- Utilize an actor-critic RL to enable the agent to administer continuous valued doses.
- Training is performed using multiple reward functions
- Using SHAP values to provide insight into which observations guide the agent's decision making.

Schamberg, Gabriel, et al. "Continuous action deep reinforcement learning for propofol dosing during general anesthesia." *Artificial Intelligence in Medicine*, 2022.

Medical Application

Automate propofol dosing with DRL:

- Interactions between agent and environment during episode simulation.
- The state value is only used in the training update step and is not used to select an action during an episode.



Schamberg, Gabriel, et al. "Continuous action deep reinforcement learning for propofol dosing during general anesthesia." *Artificial Intelligence in Medicine*, 2022.

Medical Application

Reward function:

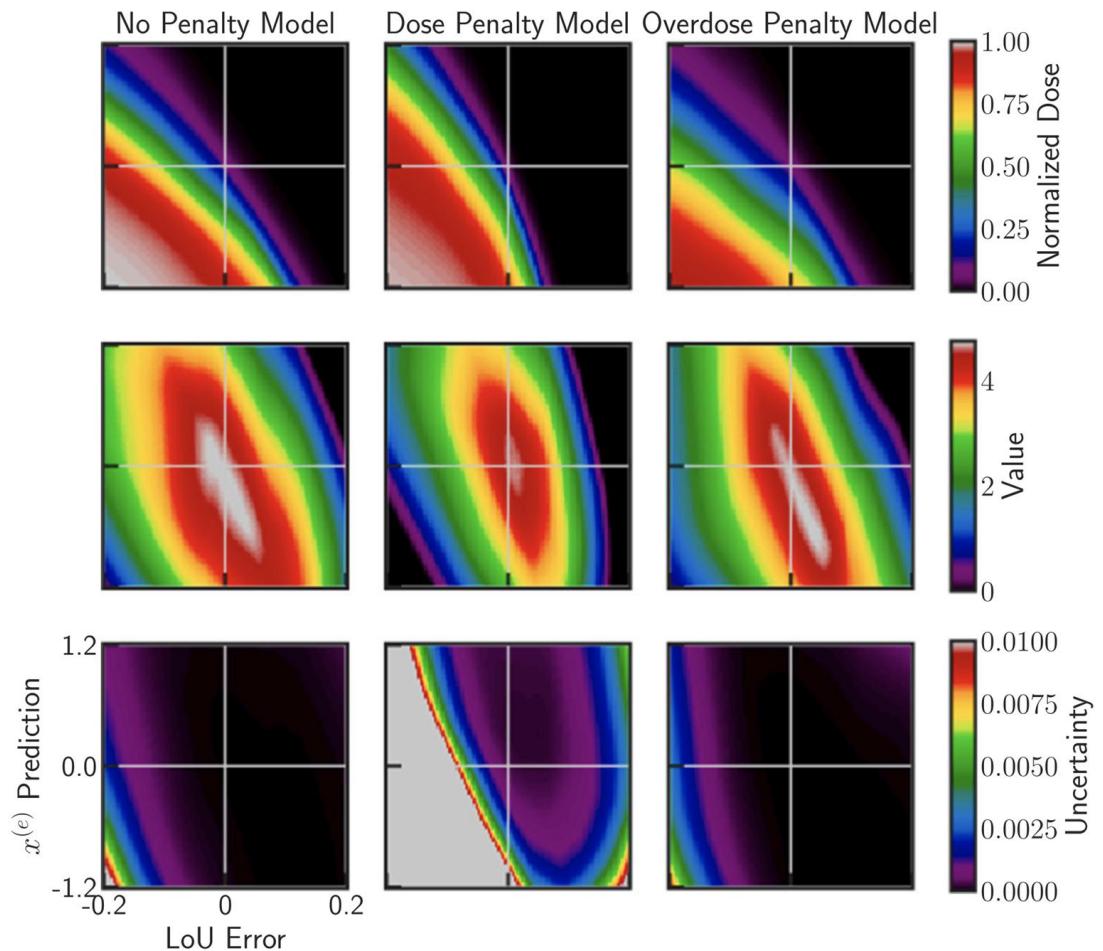
$$r_k \triangleq r(\mathbf{o}_k, a_k; \rho_1, \rho_2) = 0.5 - |o_{k+1}^{(1)}| - \rho_1 a_k - \rho_2 \max\{o_{k+1}^{(1)}, 0\}$$

- The “**dose penalty**” is controlled by ρ_1 .
- The “**overdose penalty**” is controlled by ρ_2 .
- Add 0.5 to the reward to shift the unpenalized reward range to (-0.5, 0.5).

Medical Application

Results:

- Each column represents the network outputs for one of the three trained models.
- Each row represents one of the three network outputs.



Schamberg, Gabriel, et al. "Continuous action deep reinforcement learning for propofol dosing during general anesthesia." Artificial Intelligence in Medicine, 2022.

Summarization

- There are limited number efforts in DRL interpretation.
- Cannot directly apply what have been built for image interpretations.
- Need special efforts to interpret complex strategies from agents.
- No uniformly applicable framework.
- Reliable metrics are needed.

Part 6: Hands-on Examples

Outline

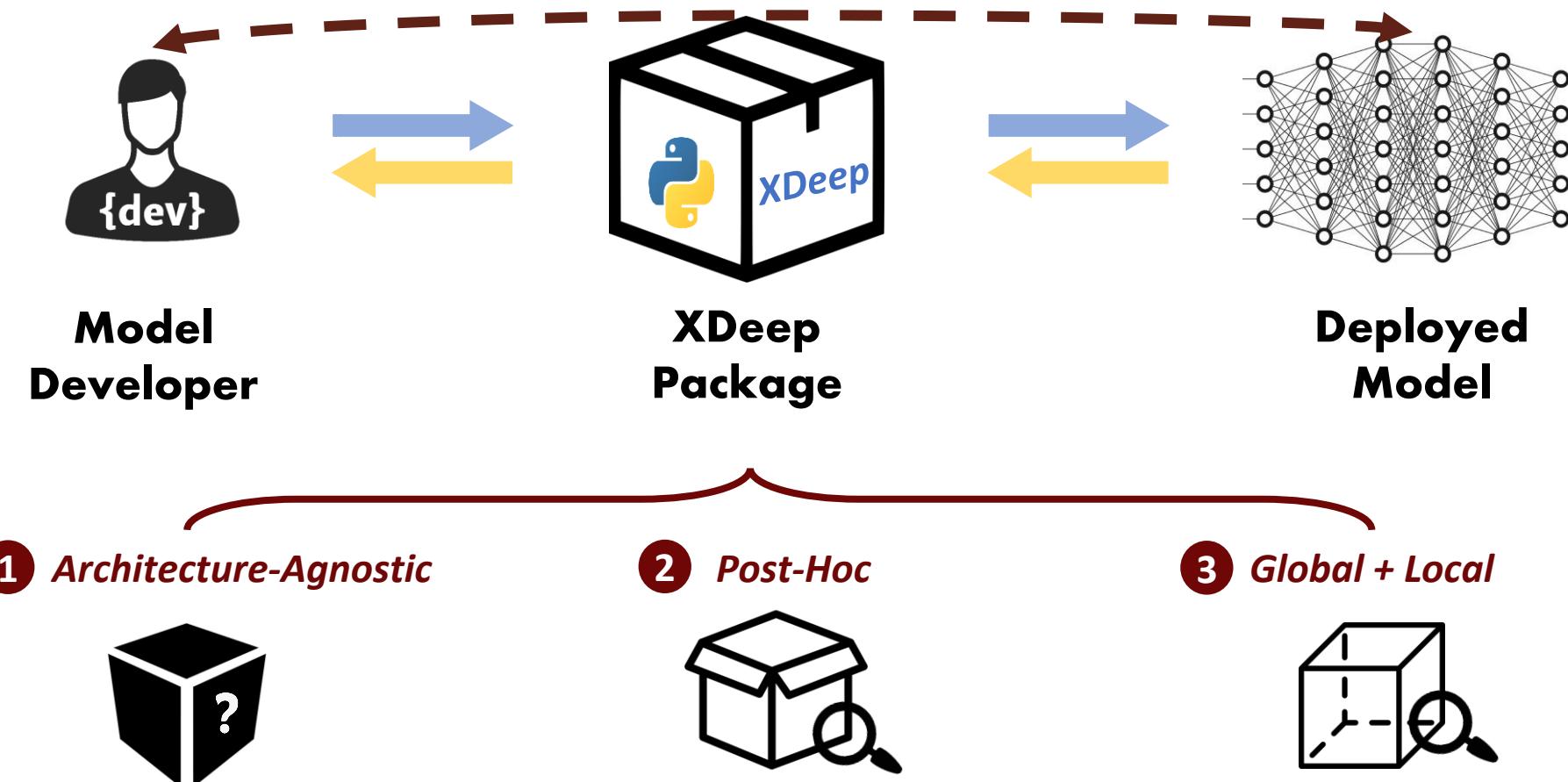
1. Case Studies on Classification Settings with XDeep
 - Text / Tabular / Image
2. Case Studies on Other Tasks with Captum
 - Regression / Question Answering / Segmentation
3. Other Useful Toolboxes for Interpretable Machine Learning
 - AIX360 / InterpretML / Alibi EXPLAIN / OmniXAI / OpenXAI

Outline

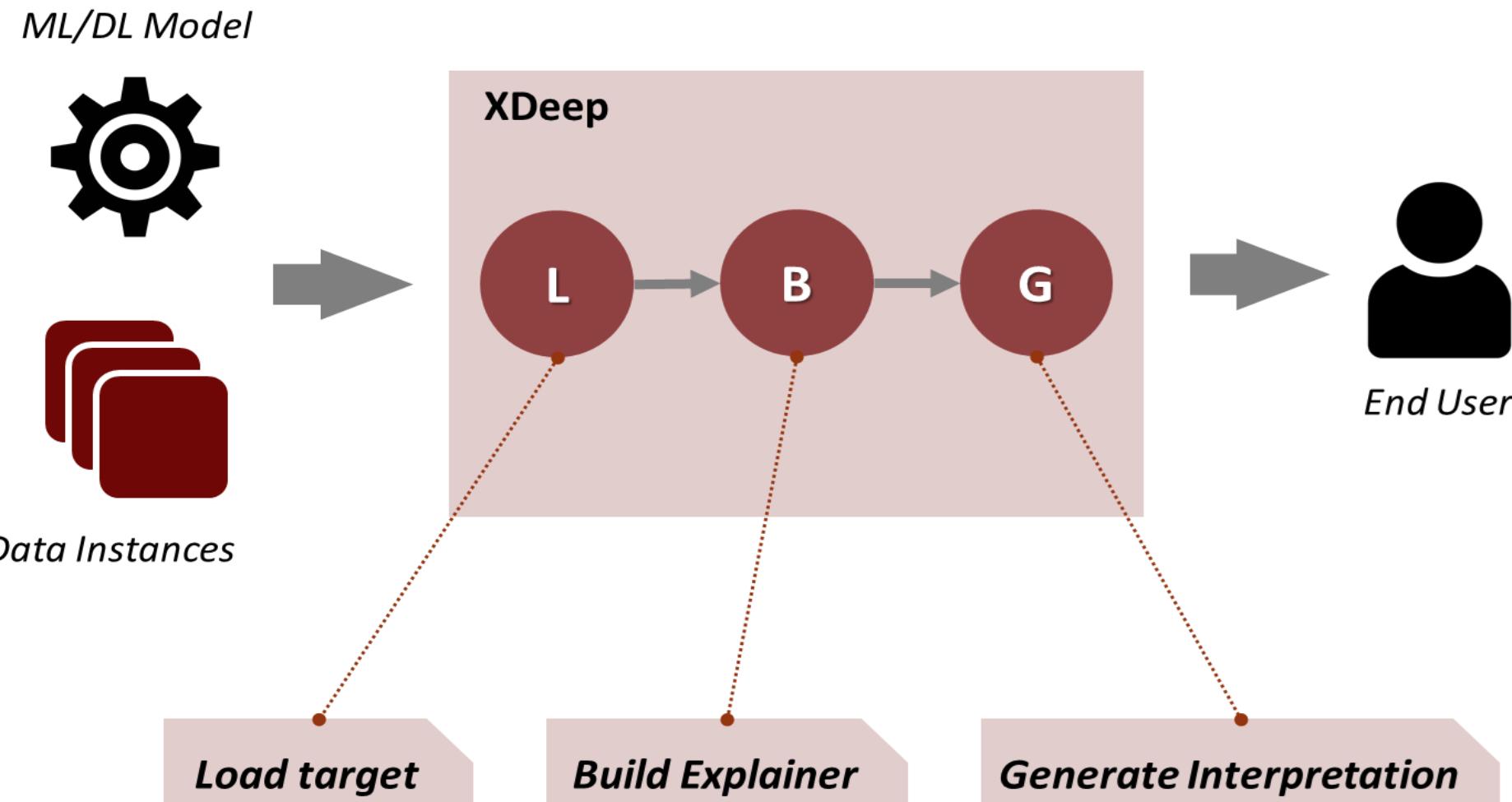
1. Case Studies on Classification Settings with XDeep
 - Text / Tabular / Image
2. Case Studies on Other Tasks with Captum
 - Regression / Question Answering / Segmentation
3. Other Useful Toolboxes for Interpretable Machine Learning
 - AIX360 / InterpretML / Alibi EXPLAIN / OmniXAI / OpenXAI

XDeep Python Library

Gap between Human Developers and Deployed Models



Use XDeep for Interpretation



Example 1 – LIME on Text

Task: Sentiment Classification with Movie Review Dataset

"I love this movie.
I've seen it many times
and it's still awesome."



"This movie is bad.
I don't like it at all.
It's terrible."



Colab Notebook Link:



[Open in Colab](#)

(No additional steps needed for running)

Example 2 – LIME on Table

Task: Income Prediction with Adult Dataset



Colab Notebook Link:

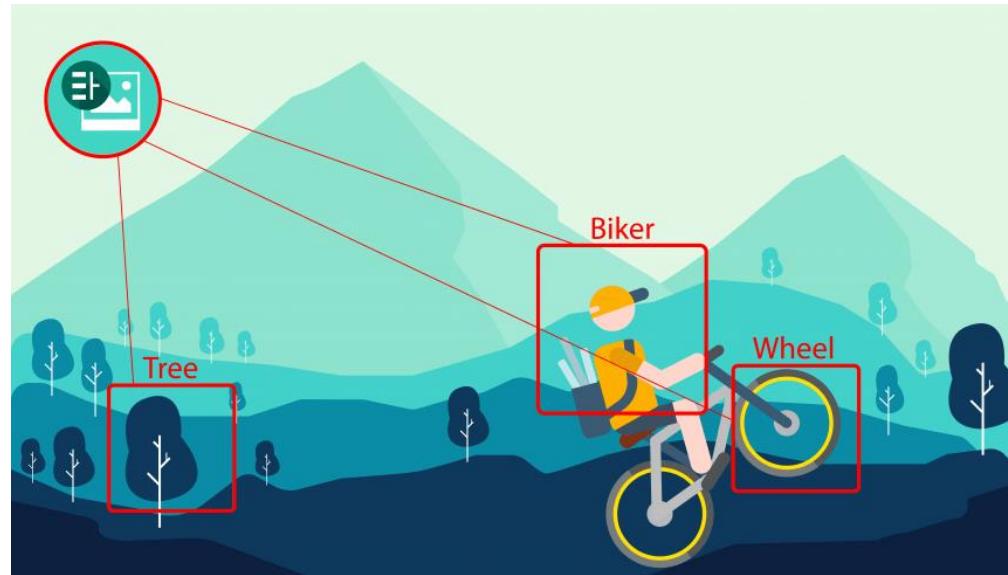


[Open in Colab](#)

(No additional steps needed for running)

Example 3 – Grad-CAM on Image

Task: Visual Recognition on LSVRC-12 Dataset



Colab Notebook Link:

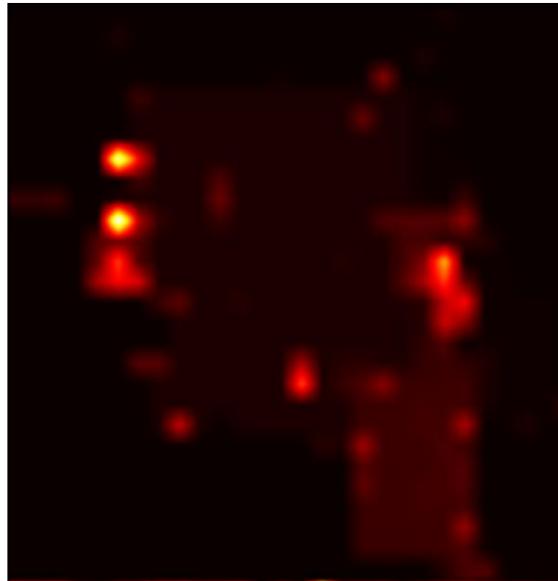


Open in Colab

(No additional steps needed for running)

Example 4 – LEG on Image

Task: Visual Recognition on ImageNet Dataset



Colab Notebook Link:



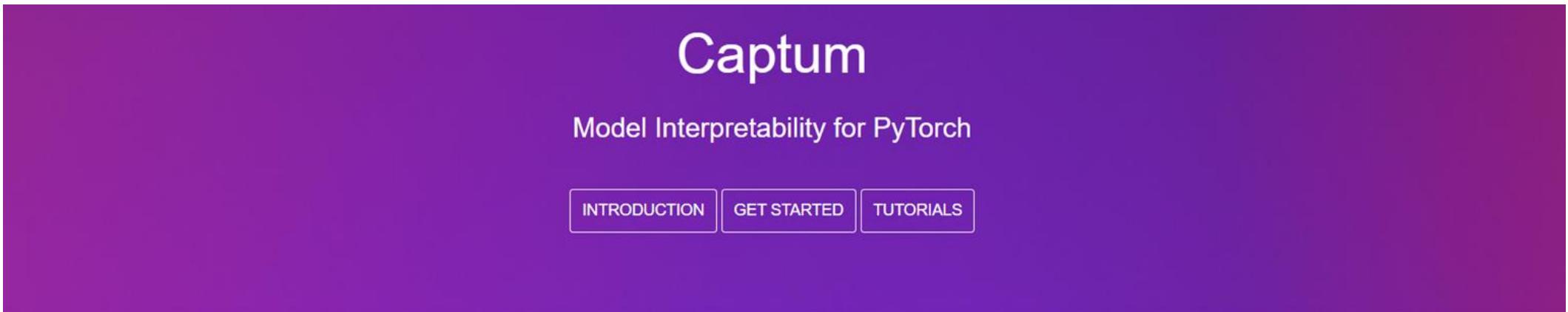
[Open in Colab](#)

(No additional steps needed for running)

Outline

1. Case Studies on Classification Settings with XDeep
 - Text / Tabular / Image
2. Case Studies on Other Tasks with Captum
 - Regression / Question Answering / Segmentation
3. Other Useful Toolboxes for Interpretable Machine Learning
 - AIX360 / InterpretML / Alibi EXPLAIN / OmniXAI / OpenXAI

Captum Library for PyTorch



KEY FEATURES



Multi-Modal

Supports interpretability of models across modalities including vision, text, and more.



Built on PyTorch

Supports most types of PyTorch models and can be used with minimal modification to the original neural network.

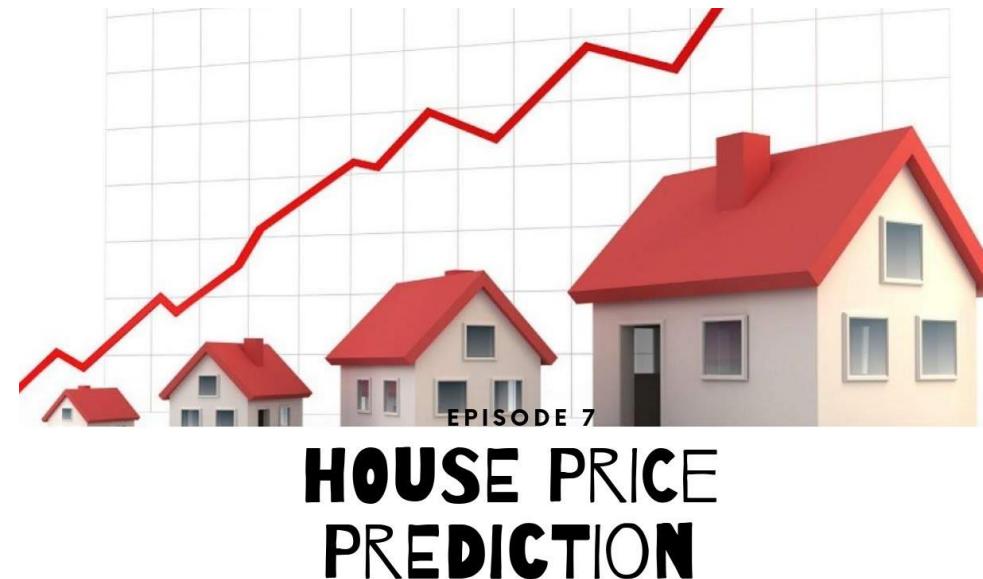


Extensible

Open source, generic library for interpretability research. Easily implement and benchmark new algorithms.

Example 5 – Interpret Regression

Task: Housing Price Prediction with Boston Dataset



Colab Notebook Link:

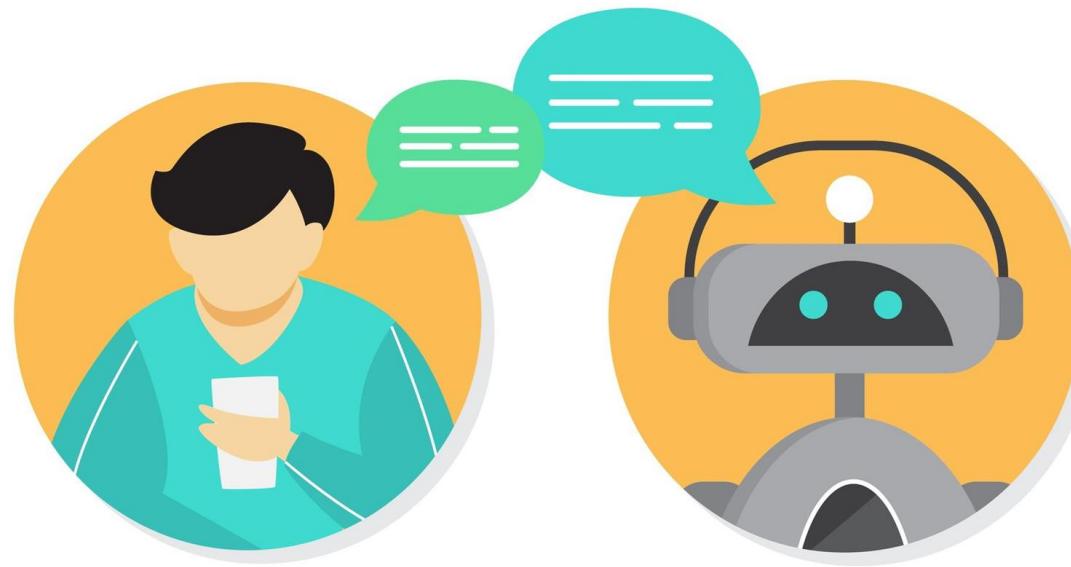


[Open in Colab](#)

(No additional steps needed for running)

Example 6 – Interpret QA BERT

Task: Question Answering with SQuAD Dataset



Colab Notebook Link:

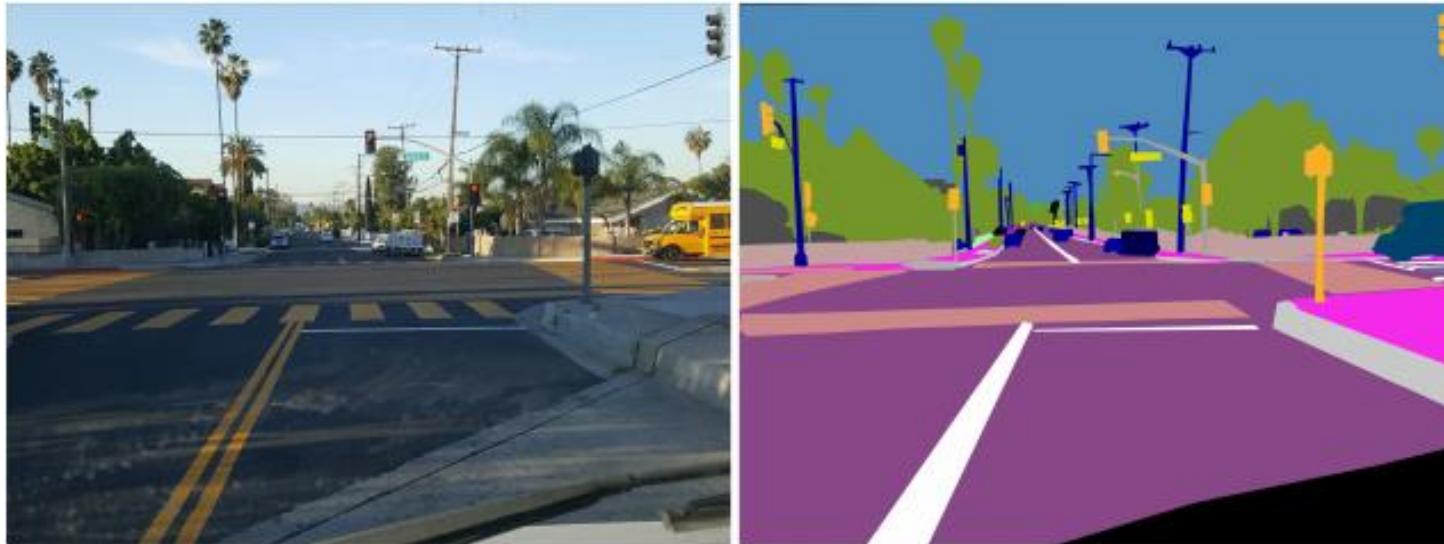


[Open in Colab](#)

(No additional steps needed for running)

Example 7 – Interpret Segmentation

Task: Image Segmentation with COCO Dataset



Colab Notebook Link:  [Open in Colab](#)

(No additional steps needed for running)

Outline

1. Case Studies on Classification Settings with XDeep
 - Text / Tabular / Image
2. Case Studies on Other Tasks with Captum
 - Regression / Question Answering / Segmentation
3. Other Useful Toolboxes for Interpretable Machine Learning
 - AIX360 / InterpretML / Alibi EXPLAIN / OmniXAI / OpenXAI

AI Explainability 360 (AIX 360) - IBM

IBM Research Trusted AI

Home Demo Resources Events Videos Community

AI Explainability 360

This extensible open source toolkit can help you comprehend how machine learning models predict labels by various means throughout the AI application lifecycle. We invite you to use it and improve it.



[API Docs ↗](#) [Get Code ↗](#)

Not sure what to do first? Start here!

Read More Learn more about explainability concepts, terminology, and tools before you begin. →	Try a Web Demo Step through the process of explaining models to consumers with different personas in an interactive web demo that shows a sample of capabilities available in this toolkit. →	Watch Videos Watch videos to learn more about AI Explainability 360 toolkit. →	Read a Paper Read a paper describing how we designed AI Explainability 360 toolkit. →	Use Tutorials Step through a set of in-depth examples that introduce developers to code that explains data and models in different industry and application domains. →	Ask a Question Join our AI Explainability 360 Slack Channel to ask questions, make comments, and tell stories about how you use the toolkit. →
---	--	---	--	---	---

<https://aix360.mybluemix.net/>

InterpretML – Microsoft Research

 InterpretML

Documentation

Contribute

GitHub

Understand Models. Build Responsibly.

A toolkit to help understand models and enable responsible machine learning

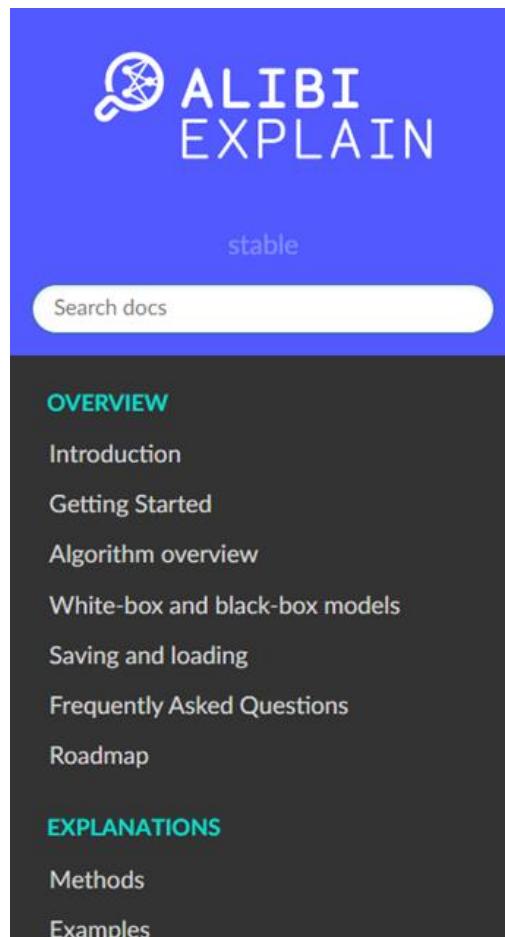
Get Started

Learn More



<https://interpret.ml/>

ALIBI EXPLAIN – Seldon IO



The screenshot shows the Alibi Explain documentation website. At the top left is the Alibi Explain logo with a magnifying glass icon. To its right, the word "stable" is displayed. Below the logo is a search bar labeled "Search docs". The main navigation menu is organized into two sections: "OVERVIEW" and "EXPLANATIONS". The "OVERVIEW" section contains links to "Introduction", "Getting Started", "Algorithm overview", "White-box and black-box models", "Saving and loading", "Frequently Asked Questions", and "Roadmap". The "EXPLANATIONS" section contains links to "Methods" and "Examples".

» Alibi Explain

[Edit on GitHub](#)



Alibi Explain

Alibi Explain is an open source Python library aimed at machine learning model inspection and interpretation. The focus of the library is to provide high-quality implementations of black-box, white-box, local and global explanation methods for classification and regression models.

Overview

- [Introduction](#)
- [Getting Started](#)
- [Algorithm overview](#)

<https://docs.seldon.io/projects/alibi/en/stable/#>

OpenXAI – Harvard



What is OpenXAI?

OpenXAI is a general-purpose lightweight library that provides a comprehensive list of functions to systematically evaluate the reliability of post hoc explanation methods. The library provides implementations and easy-to-use APIs for various state-of-the-art explanation methods and evaluation metrics. It is also flexible enough to accommodate new datasets (both synthetic and real-world), explanation methods, and evaluation metrics.

OpenXAI is an open-source framework for evaluating and benchmarking post hoc explanation methods.



Easy to Code

OpenXAI library is minimally dependent on external packages and can benchmark explanation methods with just 10 lines of code.



Easy to Evaluate

OpenXAI integrates a wide range of evaluation metrics, including faithfulness, stability, and fairness metrics.

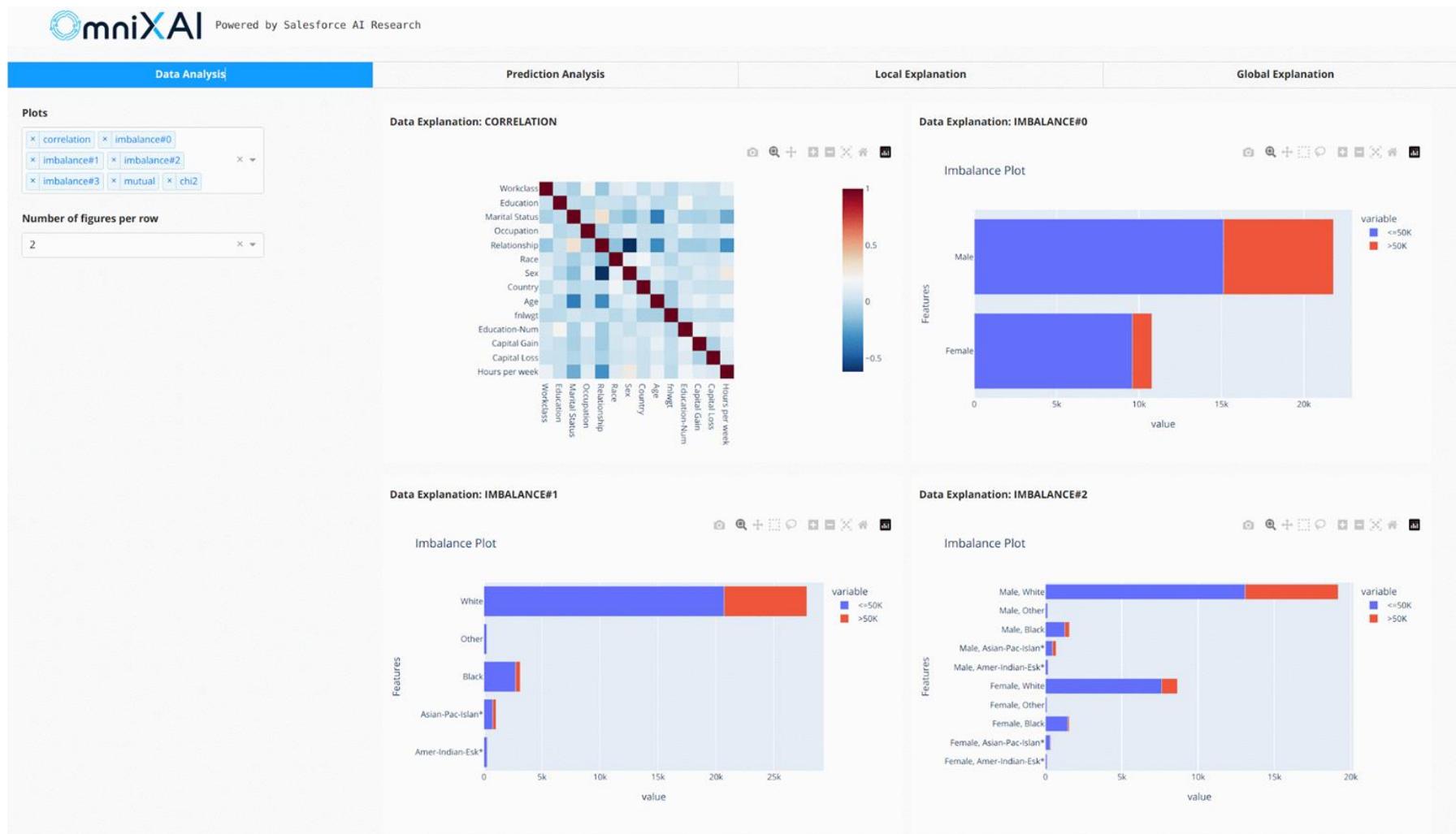


Easy to Benchmark

OpenXAI provides an intuitive abstract template with dataloaders, trained models, and XAI-ready datasets to easily and reliably benchmark explanation methods.

<https://open-xai.github.io/>

OmniXAI – Salesforce



<https://github.com/salesforce/OmniXAI>

Overall Comparison

Data Type	Method	OmniXAI	InterpretML	AIX360	Eli5	Captum	Alibi
Tabular	LIME	✓	✓	✓		✓	
	SHAP	✓	✓	✓		✓	✓
	PDP	✓	✓				
	ALE	✓					✓
	Sensitivity	✓	✓				
	Integrated gradient	✓				✓	✓
	Counterfactual	✓					✓
	Linear models	✓	✓	✓	✓		✓
	Tree models	✓	✓	✓	✓		✓
	L2X	✓					
Image	LIME	✓				✓	
	SHAP	✓				✓	
	Integrated gradient	✓				✓	✓
	Grad-CAM	✓			✓	✓	
	CEM	✓		✓			✓
	Counterfactual	✓					✓
Text	L2X	✓					
	Feature visualization	✓					
	LIME	✓			✓	✓	
	SHAP	✓				✓	
	Integrated gradient	✓				✓	✓
Timeseries	Counterfactual	✓					
	L2X	✓					
	SHAP	✓					
	Counterfactual	✓					

Yang, Wenzhuo, Hung Le, Silvio Savarese, and Steven CH Hoi.
"OmniXAI: A Library for Explainable AI." arXiv preprint arXiv:2206.01612 (2022).