Xplique

👋 A Neural Networks Explainability Toolbox

```
pip install xplique
```

Thomas FEL*, Lucas HERVIER*, David VIGOUROUX,
Antonin POCHE, Thibault BOISSIN, Philippe DEJEAN,
Benjamin DEPORTE, David PETITEAU, Justin PLAKOO
and all the **DEEL** *team*

ANITI ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE · Université Fédérale Toulouse Midi-Pyrénées · DEEL DEpendable & Explainable Learning · BROWN · SNCF · IVADO · SAINT EXUPÉRY · CRIAQ

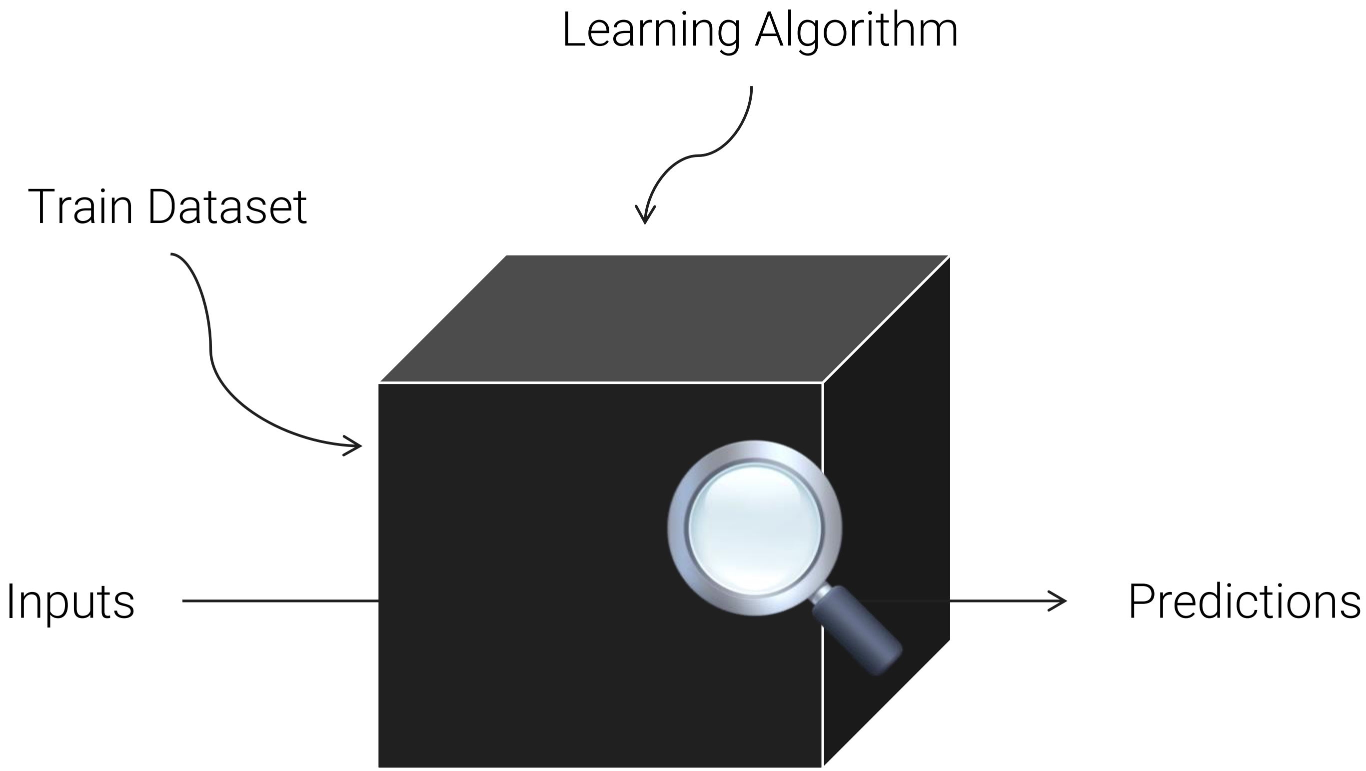
Thomas FEL

PhD Student, ANITI & DEEL Team. SNCF Research & Innovation.
Working on Explainable AI
under the supervision of the Prof. Thomas Serre (Brown University, ANITI)

Summary

1. A short introduction to XAI
2. Attribution Methods + Metrics module
3. Feature Visualization module
4. Concepts module

1. Introduction
The Black-box problem



1. Introduction

A Conceptual challenge

"An Explanation is a set of statements [...] which clarifies the cause, the context and consequences of those facts. [...] The component of an explanation can be implicit and interwoven with one another"

Jess DRAKE, LOGIC

- An explanation provides information
- An explanation depends on domain knowledge
- An explanation helps to complete the knowledge of the domain

Lombrozo (2006), Hempel & Oppenheim (1948), Bechtel and Abrahamsen (2005), Chater and Oaksford (2006), Glennan (2002), Keil (2006)

1. Introduction

Why do we need Explanations ?

Build *trust* in the model prediction

Elucidate important aspects of learned models

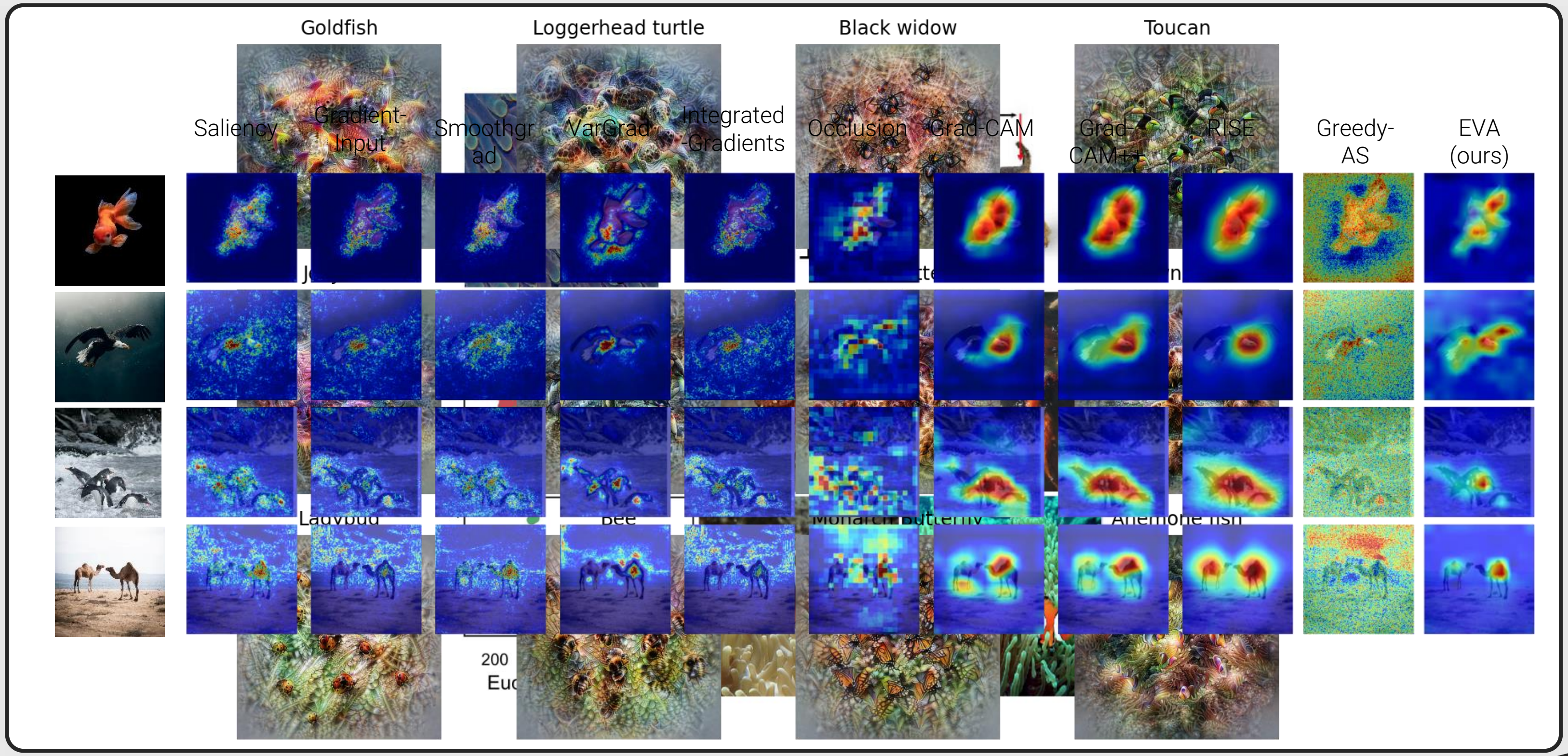
Help satisfy regulatory requirements and Certification process

Reveal bias or other unintended effects learned by a model

Detect and prevent failure cases

Debug & Train better model

1. Introduction
A Technical challenge



Model
- Feature Viz,
- Concept Activation Vector
- Explanation 'by design'
- ...

Predictions
- Feature Attribution
- Feature Inversion
- ...

Data
- Nearest Neighbourhood
- Influence Function
- ...

1. Introduction

Taxonomy

(soon) Influenciae
(Agustin Martin Picard)

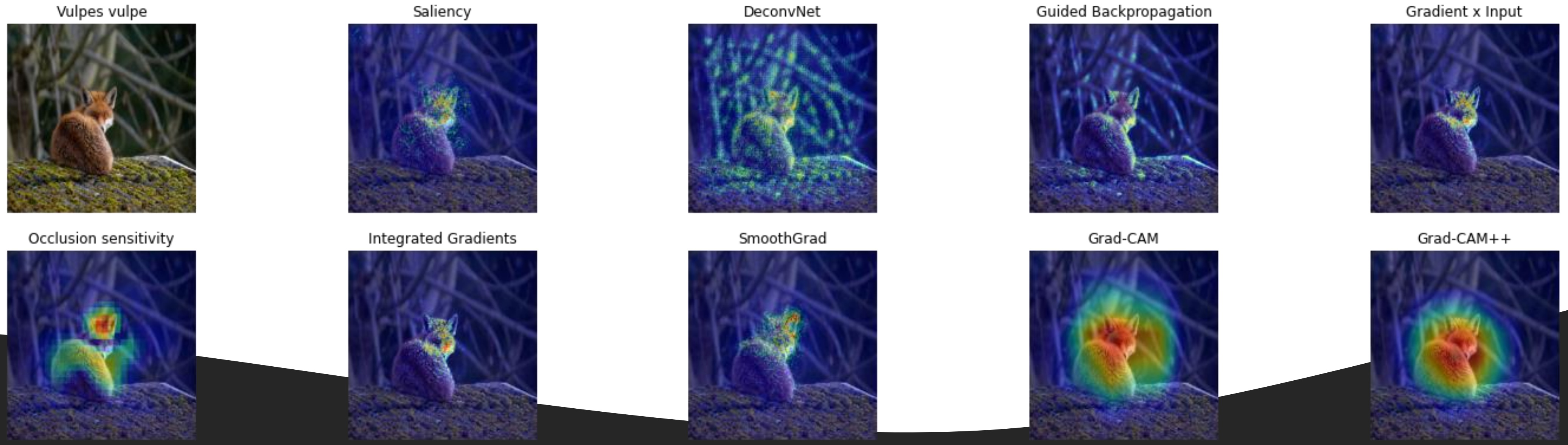
	Ex Ante (A priori)	Ex Post (Post hoc)
Local	Prototype Explanation Network SENN	**Attributions Methods** **Feature Inversion** Influence Functions
	Explanation 'by design'	**Surrogate Model**
Global	Concept bottleneck model	K-NN on latent space **Feature Viz** **Concept Activation Vectors**

2. Feature Attributions

Explaining specific predictions

Consider the following general supervised learning setting: input space $\mathcal{X} \subseteq \mathbb{R}^d$, an output space $\mathcal{Y} \subseteq \mathbb{R}$, and a black-box predictor $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}$, which for some test input $\mathbf{x}$ predicts the output $\mathbf{f}(\mathbf{x})$.

We then define a feature attribution explanation as a function $\Phi : \mathcal{F} \times \mathbb{R}^d \to \mathbb{R}^d$, that given a black-box predictor $\mathbf{f}$, and a test point $\mathbf{x}$, provides importance scores $\Phi(\mathbf{f}, \mathbf{x})$.



2. Feature Attributions

Explaining specific predictions

Saliency Maps Symonyan & al (2013)[1]

$$\Phi = \nabla f(x) \implies \phi_i = \frac{\partial f(x)}{\partial x_i}$$

In an infinitesimal neighborhood (often not feasible), what are the features that most impact the output score ?

Fast, really noisy, not really meaningfull (except for Lipschitz networks?)

Free Parameters

SmoothGrad Smilkov & al (2017)[2]

$$\Phi = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I\sigma)}[\nabla f(x+\epsilon)]$$

$$\Phi = \frac{1}{N}\sum_{i=0}^{N} \nabla f(x+\epsilon)$$

As the name suggests, smoothes out the gradient by averaging the effect of small perturbations within a neighborhood around each pixels.

N~80 to have good results on 224x224 images. Slower than Saliency. Lot of variants (VarGrad, Squaregrad...)

Parameters: N, epsilon

[1] Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
[2] SmoothGrad: removing noise by adding noise

2. Feature Attributions

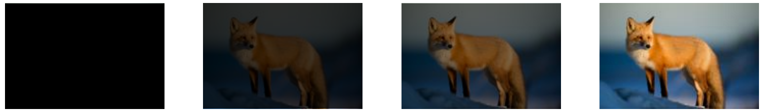
Explaining specific predictions

Integrated Gradients Sundarajan & al (2017)[1]

$$\Phi = (x - x_0) \int_0^1 \frac{\partial f(x_0 + \alpha(x - x_0))}{\partial x} d\alpha$$

$$\Phi = (x - x_0) \frac{1}{N} \sum_{i=0}^{N} \frac{\partial f(x_0 + \frac{i}{N}(x - x_0))}{\partial x}$$

Averages the gradient values along the path from a baseline state to the current value. The baseline state is often set to zero.



N~80, Axiom Grounded, lots of tricks to leverage Integral approximation. What is a good baseline (x0) ?

Parameters: N, baseline

Occlusion Ancona & al (2017)[2]

$$\phi_i = f(x) - f(x_{[x_i = x_0]})$$

Sweeps a patch that occludes pixels over the image, and uses the variations of the model prediction to identify critical areas.

hard to tune, black-box, need a baseline. Always good to have.

Parameters: patch size, patch stride, baseline

[1] Axiomatic Attribution for Deep Networks

[2] Towards better understanding of gradient-based attribution methods for Deep Neural Networks

2. Feature Attributions

Explaining specific predictions

Rise Petsiuk & al (2018)[1]

$$\phi_i = \mathbb{E}[f(x \odot m) | m = 1]$$

Probes the model with randomly masked versions of the input image.

N~8000, Hard to compute. Baseline sensitive. Parameter sensitive.
Random binary masks (i.i.d) are upsampled from low resolution
Parameters : N, Low Res Grid, M=1 Probability

Sobol Attribution Method

Thomas Fel*, Rémi Cadène*, Mathieu Chalvidal, Matthieu Cord, David Vigouroux, Thomas Serre, NeurIPS (2021)[2]

$$\phi_i = \frac{\mathbb{E}_{\boldsymbol{M}_{\sim i}}(\mathrm{Var}_{\boldsymbol{M}_i}(\boldsymbol{f} \circ \zeta(\boldsymbol{x}, \boldsymbol{M}) | \boldsymbol{M}_{\sim i}))}{\mathrm{Var}(\boldsymbol{f} \circ \zeta(\boldsymbol{x}, \boldsymbol{M}))}$$

Estimates the variance of the model output (Sobol' indices) w.r.t a perturbation function using Quasi-Monte Carlo sampling.
N~4000. No Baseline.
Parameters : Grid size, Perturbation function (e.g inpainting, blurring...)

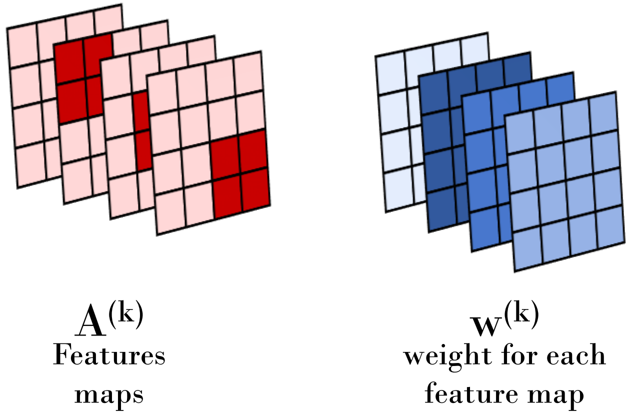
[1] RISE: Randomized Input Sampling for Explanation of Black-box Models
[2] Look at the Variance! Efficient Black-box Explanations with Sobol-based Sensitivity Analysis

2. Feature Attributions

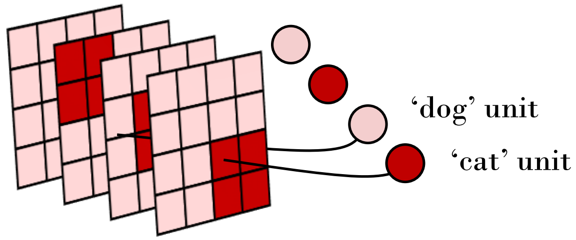
Explaining specific predictions

CAM Zhou & al (2016)[1] **&**
Grad-CAM Selvaraju & al (2017)[2]

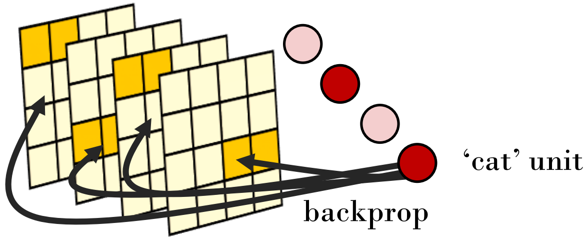
$$\Phi = ReLU\left(\sum_{k=0}^{K} w^{(k)} A^{(k)}\right)$$



For CAM (Conv + Global Average Pooling, one unit per class), the weight is 1 only for the feature map of the class else 0.



For Grad-CAM (any ConvNet), the weight is the avg of the gradients of each feature maps.



$$w^{(k)} = \frac{1}{Z} \sum_i \sum_j \frac{\partial f(x)}{\partial A_{ij}^{(k)}}$$

[1] Learning Deep Features for Discriminative Localization
[2] Visual Explanations from Deep Networks via Gradient-based Localization

2. Feature Attributions

Explaining specific predictions

Grad-CAM Selvaraju & al (2017)

$$w^{(k)} = \frac{1}{Z} \sum_i \sum_j \frac{\partial f(x)}{\partial A_{ij}^{(k)}}$$

$$\Phi = ReLU\left(\sum_{k=0}^{K} w^{(k)} A^{(k)}\right)$$

Uses the gradients flowing back into the last convolution layer to generate a weight. This weight is used for the feature map. Aggregates all the weighted feature maps and removes the negative values. The results are then extrapolated (gives smooth results)

Quick to compute (1 forward, ~1/2 backward) and give good results. Lots of variants (Ablation-CAM, Grad-CAM++, Score-CAM, Shapley-CAM...).

Parameters : a conv layer

~white-box state of the art.

2. Feature Attributions

‘Cool, but how to use that with Xplique ?’

```
pip install xplique
```

ANITI DEEL BROWN SNCF SAINT EXUPÉRY IVADO CRIAQ

```python
import xplique
from xplique.attributions import (Saliency,
                                  SmoothGrad,
                                  GradCAM)
```

ANITI ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE DEEL DEpendable & Explainable Learning BROWN SNCF SAINT EXUPÉRY IVADO CRIAQ

```python
import xplique
from xplique.attributions import (Saliency,
                                  SmoothGrad,
                                  GradCAM)
```

ANITI DEEL BROWN SNCF SAINT EXUPÉRY IVADO CRIAQ

2. Feature Attributions

```
import numpy as np
from xplique.attributions import Saliency, GradCAM, SmoothGrad

model = tf.keras.Model(...)

# initialize your explainers
explainers = [
  Saliency(model, batch_size=50),
  GradCAM(model, conv_layer='mixed3d'),
  SmoothGrad(model, nb_samples=50)
]

inputs = np.random.rand(10, 224, 224, 3)
labels = np.random.rand(10, 1000)

# explainer share the same api once initialized
explanations = [
  explainer(inputs, labels) for explainer in explainers
]
```

Multiple explainers

2. Feature Attributions

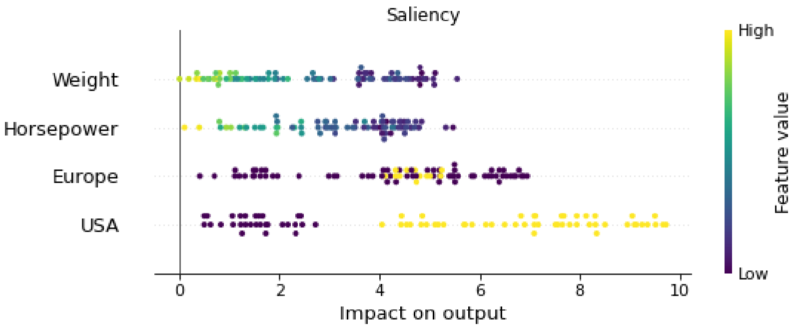
Explaining specific predictions

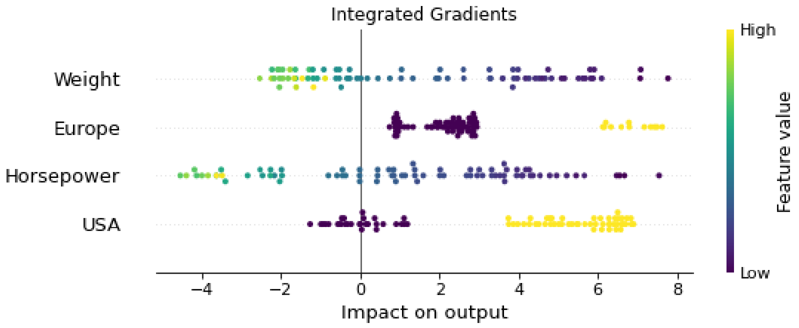
Demo

2. Feature Attributions

Other data format!

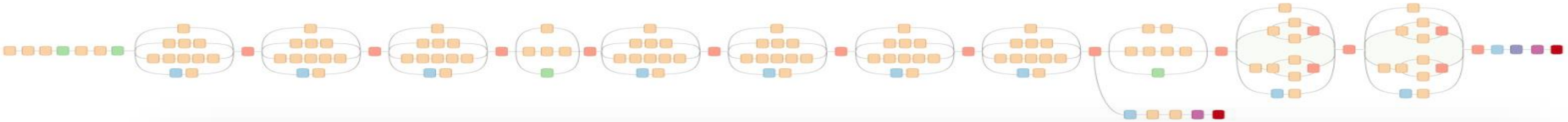
Bounding Box
(notebook coming soon!)

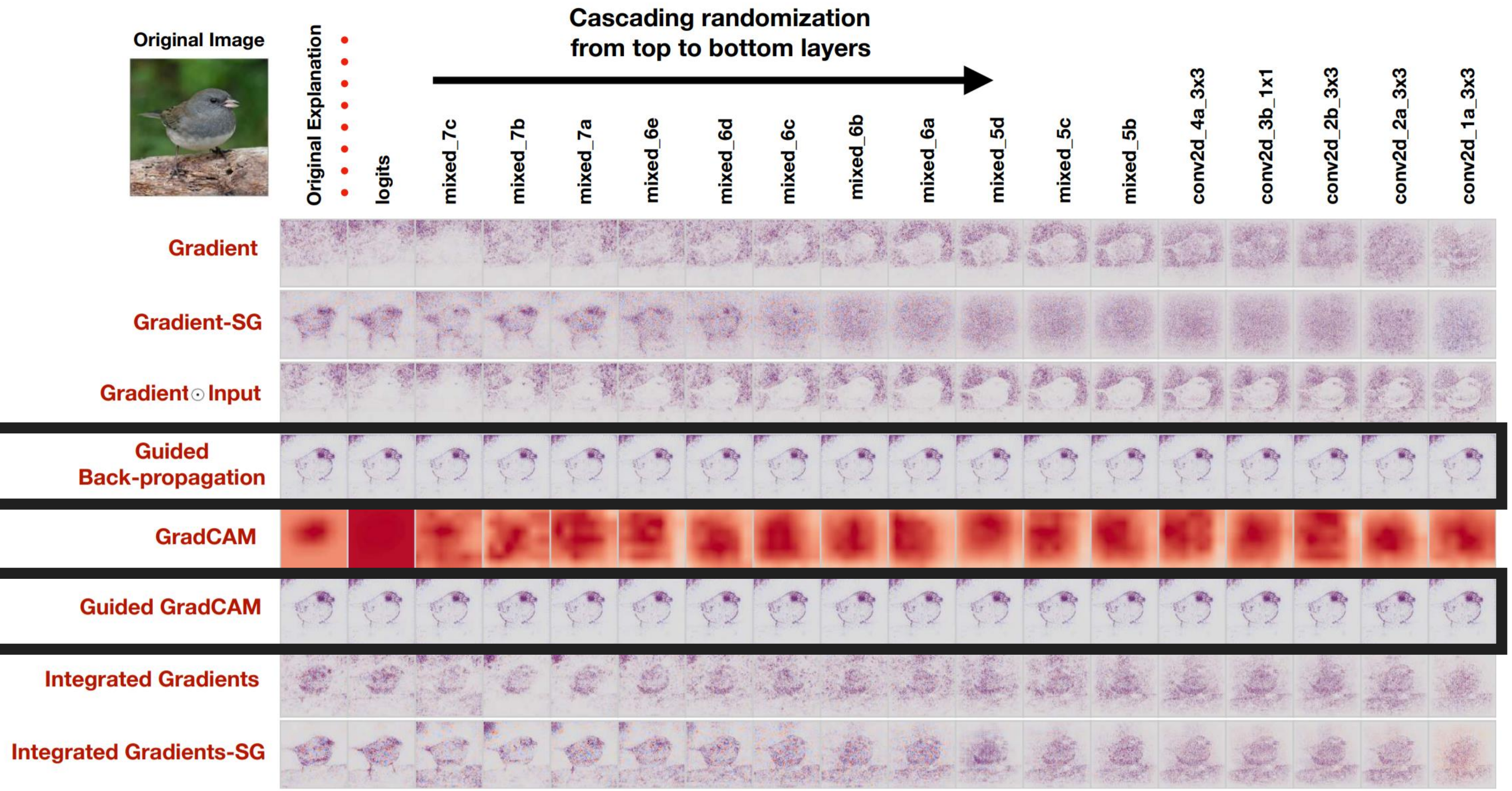




Tabular data

& Even time series !




Cascading randomization
from top to bottom layers
Original Image
Original Explanation
logits
mixed_7c
mixed_7b
mixed_7a
mixed_6e
mixed_6d
mixed_6c
mixed_6b
mixed_6a
mixed_5d
mixed_5c
mixed_5b
conv2d_4a_3x3
conv2d_3b_1x1
conv2d_2b_3x3
conv2d_2a_3x3
conv2d_1a_3x3
Gradient
Gradient-SG
Gradient ⊙ Input
Guided
Back-propagation
GradCAM
Guided GradCAM
Integrated Gradients
Integrated Gradients-SG

Confirmation bias.

Just because it makes sense to humans doesn't mean it reflects the evidence for prediction.

2. Feature Attributions

Attribution methods can be manipulated

Fairwashing Explanations with Off-Manifold Detergent

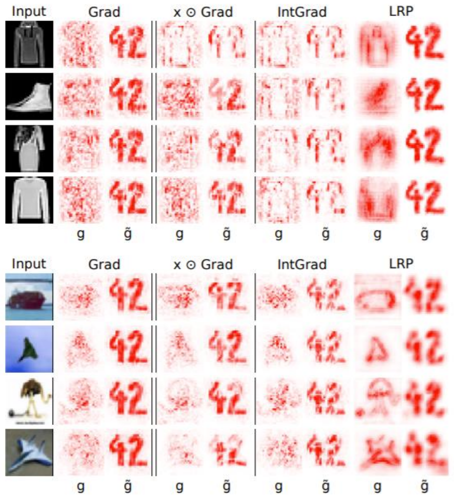
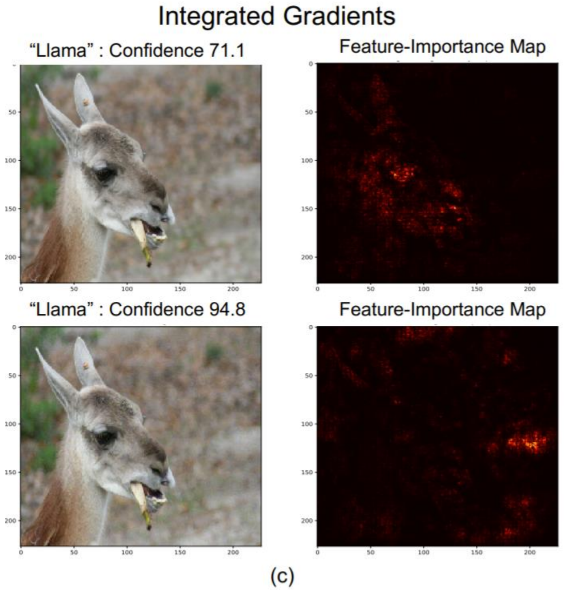


Figure 2. Example explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right). Images from the test sets of FashionMNIST (**top**) and CIFAR10 (**bottom**).

Interpretation of Neural Networks is Fragile



Interpretable Deep Learning under Fire

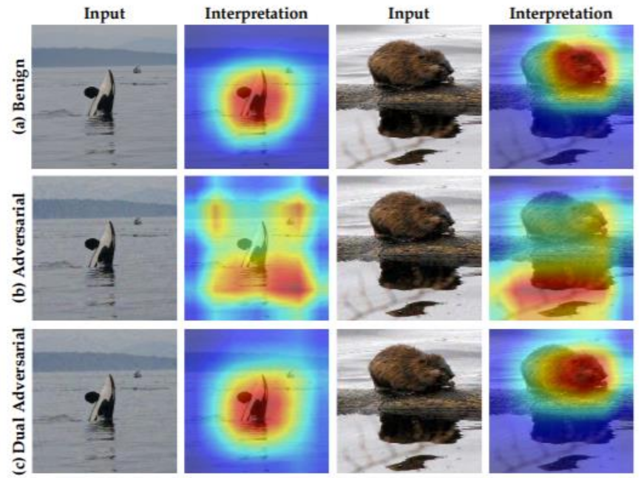
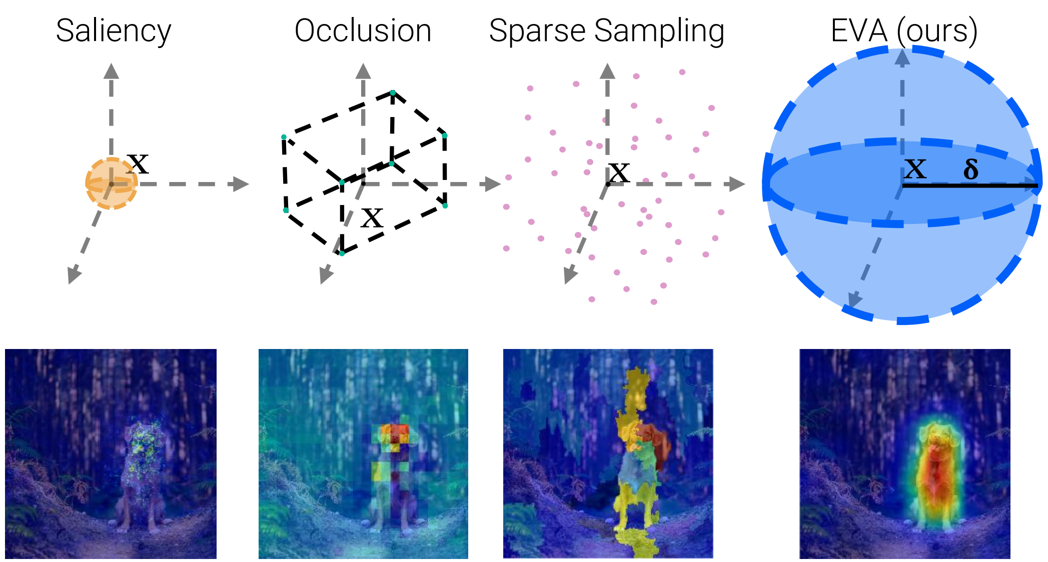
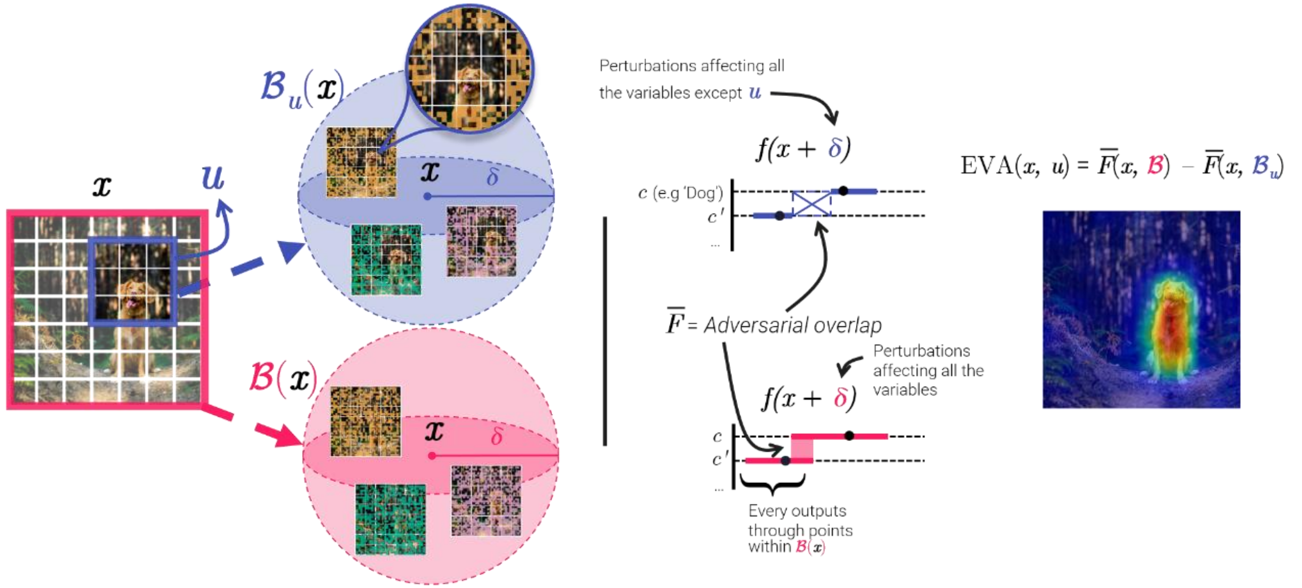


Figure 1: Sample (a) benign, (b) regular adversarial, and (c) dual adversarial inputs and interpretations on ResNet [22] (classifier) and CAM [64] (interpreter).

2. Feature Attributions

Formal method for robust & efficient explainability





(1) $\mathcal{B}(x)$ are the points around x with perturbation δ. $\mathcal{B}_u(x)$ is a subset of $\mathcal{B}(x)$ where the variables u – the dog head – are not perturbed

(2) Without perturbating u, we can't reach any adversarial, thus u is important

(3) Attribution after computing importance on each variables

ANITI Université Fédérale Toulouse Midi-Pyrénées DEEL DEpendable & Explainable Learning BROWN

Don't Lie to Me! Robust and Efficient Explainability with Verified Perturbation Analysis

Thomas FEL*, Mélanie DUCOFFE*, David VIGOUROUX, Rémi CADENE, Mikael CAPELLE, Claire NICODÈME, Thomas SERRE, *Pre-print under review*

2. Feature Attributions

Fidelity metric

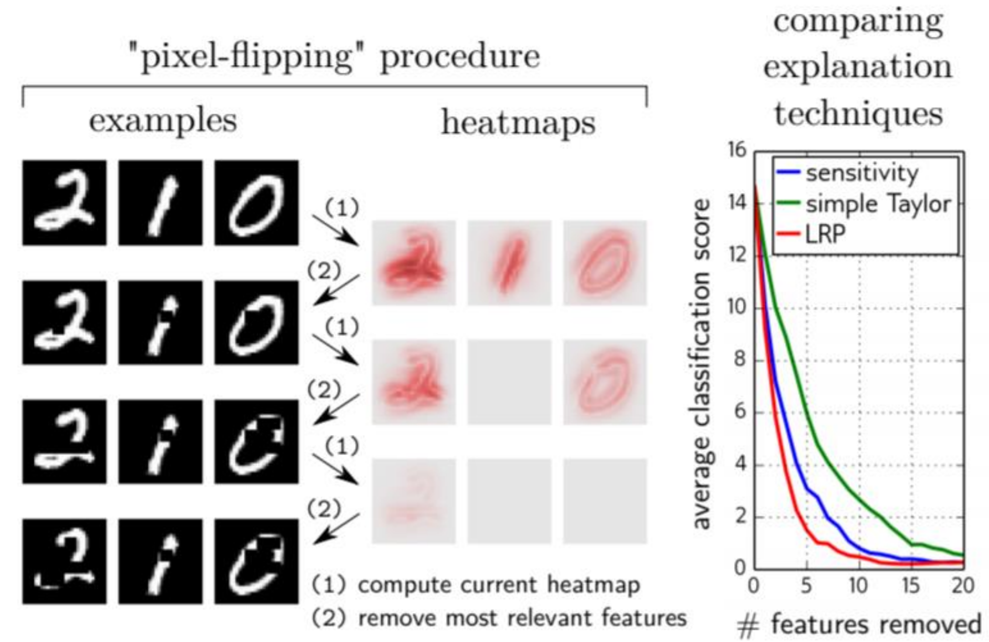
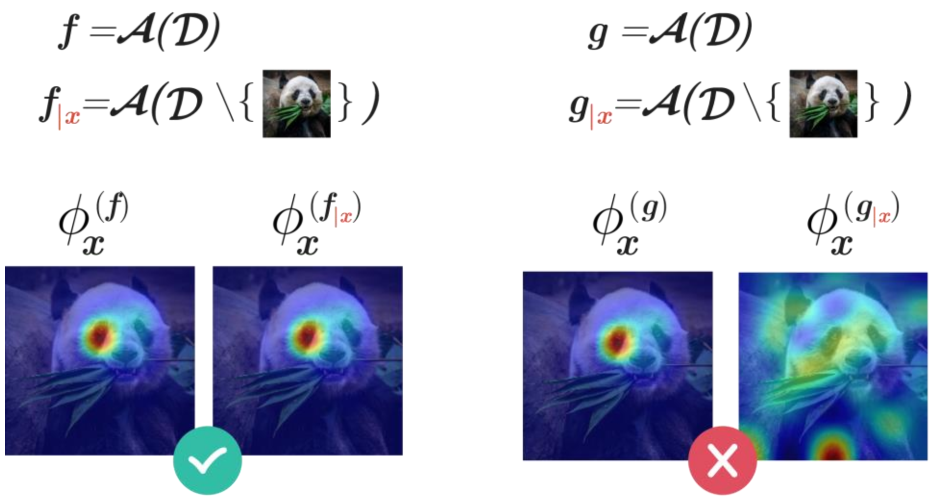


Figure 8: Illustration of the "pixel-flipping" procedure. At each step, the heatmap is used to determine which region to remove (by setting it to black), and the classification score is recorded.

Evaluating the visualization of what a Deep Neural Network has learned, Samek & al, 2015.

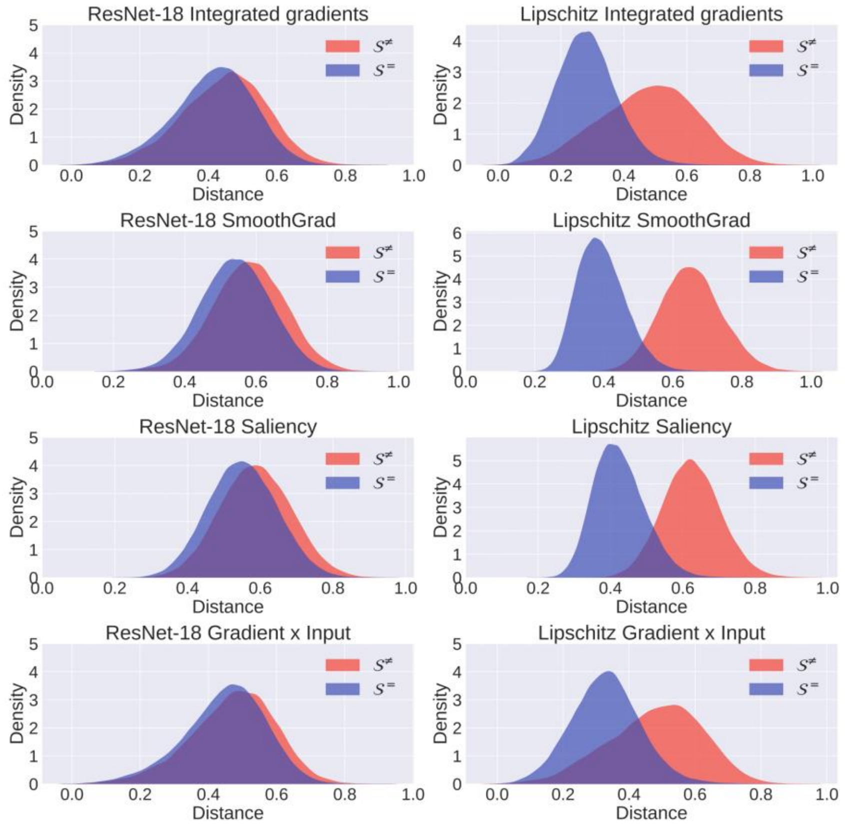
2. Feature Attributions

Consistency & Representativity metrics



f has Algorithmically Stable Explanations

g has Algorithmically Unstable Explanations



1-Lipschitz model gives 'better' explanations!

How Good is your Explanation? Algorithmic Stability Measures to Assess the Quality of Explanations for Deep Neural Networks
Thomas FEL, David VIGOUROUX, Rémi CADENE, Thomas SERRE, *WACV (2022)*

2. Feature Attributions

Metrics

```
import xplique
from xplique.attributions import GradCAM
from xplique.metrics import Deletion


model = tf.keras.Model(...)

inputs = np.random.rand(10, 224, 224, 3) # images
labels = np.random.rand(10, 1000) # one-hot

explainer = GradCAM(model, batch_size=32)
explanations = explainer(inputs, labels)
```

2. Feature Attributions

Fidelity metric

Demo

2. Feature Attributions

Attribution Method	Type of Model	Source	Tabular Data	Images	Time-Series	Tutorial
Deconvolution	TF	Paper	✓	✓	WIP	Open in Colab
Grad-CAM	TF	Paper		✓	WIP	Open in Colab
Grad-CAM++	TF	Paper		✓	WIP	Open in Colab
Gradient Input	TF	Paper	✓	✓	WIP	Open in Colab
Guided Backprop	TF	Paper	✓	✓	WIP	Open in Colab
Integrated Gradients	TF	Paper	✓	✓	WIP	Open in Colab
Kernel SHAP	Callable*	Paper	✓	✓	WIP	Open in Colab
Lime	Callable*	Paper	✓	✓	WIP	Open in Colab
Occlusion	Callable*	Paper	✓	✓	WIP	Open in Colab
Rise	Callable*	Paper	WIP	✓	WIP	Open in Colab
Saliency	TF	Paper	✓	✓	WIP	Open in Colab
SmoothGrad	TF	Paper	✓	✓	WIP	Open in Colab
SquareGrad	TF	Paper	✓	✓	WIP	Open in Colab
VarGrad	TF	Paper	✓	✓	WIP	Open in Colab

Attribution Metrics	Type of Model	Property	Source
MuFidelity	TF	Fidelity	Paper
Deletion	TF	Fidelity	Paper
Insertion	TF	Fidelity	Paper
Average Stability	TF	Stability	Paper
MeGe	TF	Representativity	Paper
ReCo	TF	Consistency	Paper
(WIP) e-robustness			

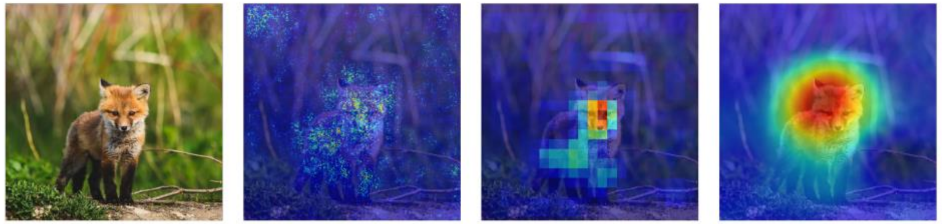
2. Feature Attributions

Fidelity doesn't mean Usefulness

What I Cannot Predict, I Do Not Understand:
A Human-Centered Evaluation Framework for Explainability Methods

Thomas Fel[1,3,4] * Julien Colin[1,3] * Rémi Cadène[1,2] † Thomas Serre[1,3]
[1]Carney Institute for Brain Science, Brown University, USA [2]Sorbonne Université, CNRS, France
[3]Artificial and Natural Intelligence Toulouse Institute, Université de Toulouse, France
[4] Innovation & Research Division, SNCF
{thomas_fel, julien_colin, remi_cadene}@brown.edu

Which explanation is the most useful to humans?



Prediction: **Red Fox** Saliency (0.92) Occlusion (0.89) Grad-CAM (0.89)

4.2. Faithfulness metric as a proxy for Usefulness?

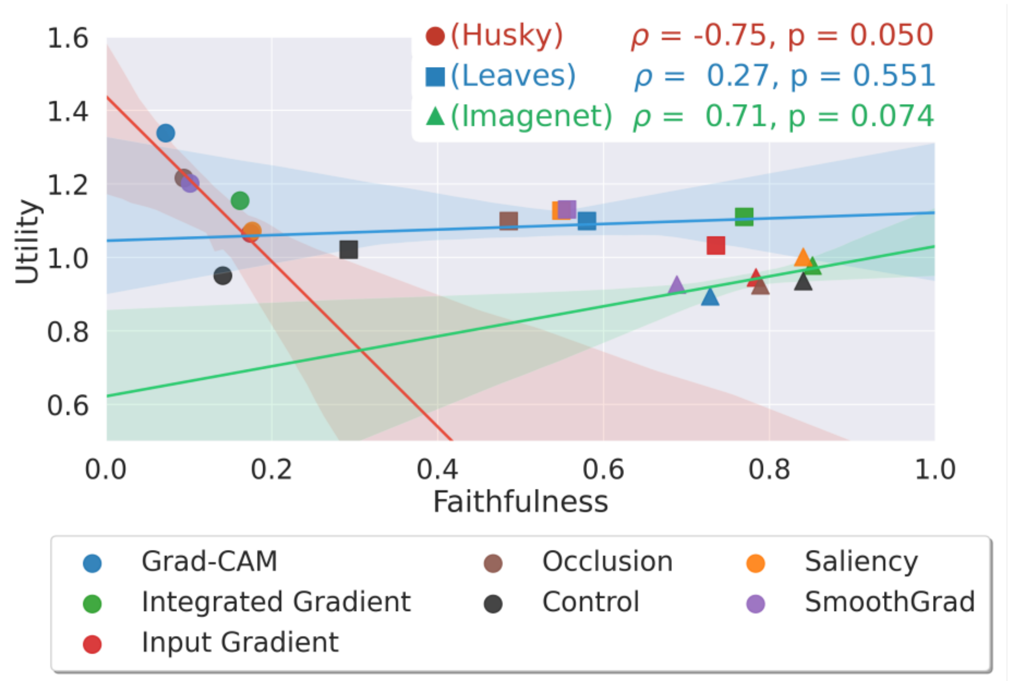


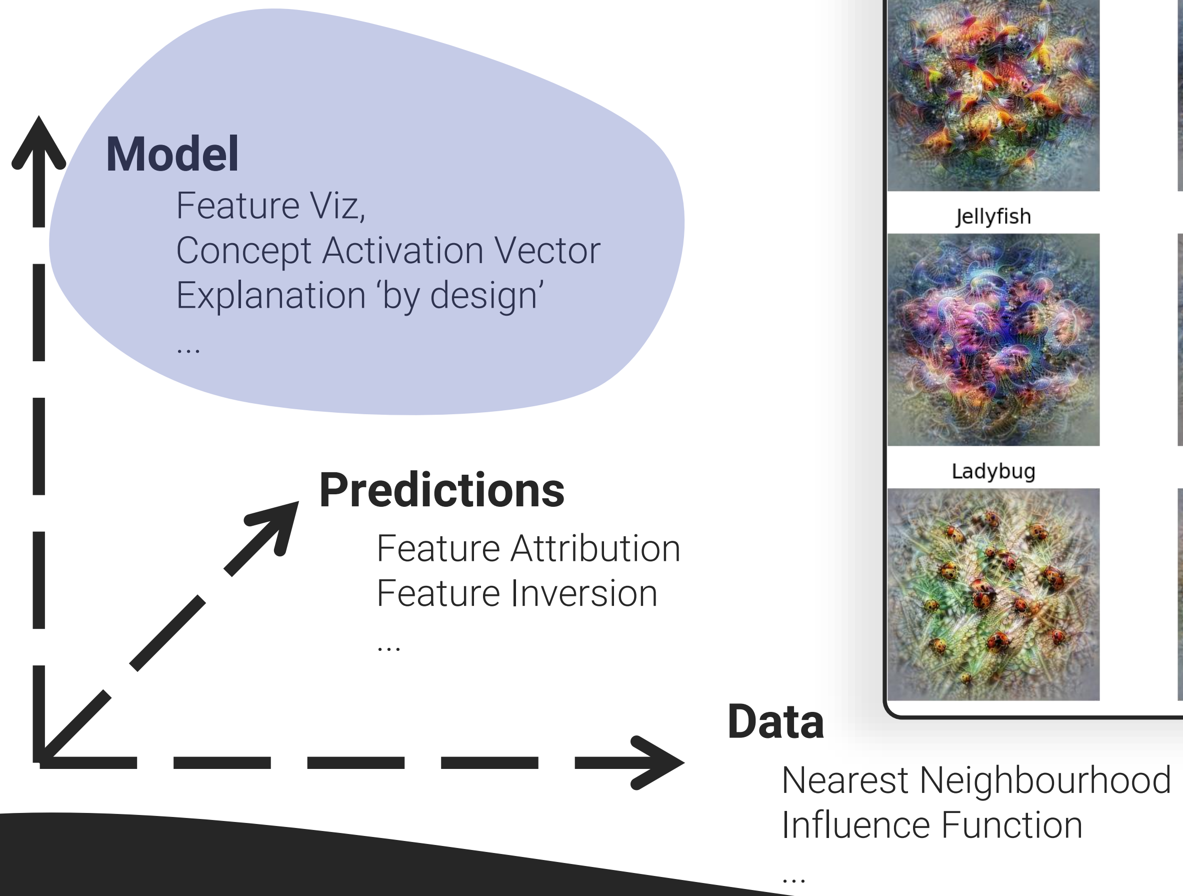
Figure 6. ***Utility* vs Faithfulness correlation.** The utility scores on the two datasets Husky vs. Wolf (point marker) and Leaves (square marker) are plotted showing a poor or anti-correlation between the two measures. Concerning the ImageNet dataset (triangle marker), the *Utility* scores are insignificant since none of the methods improves the baseline.

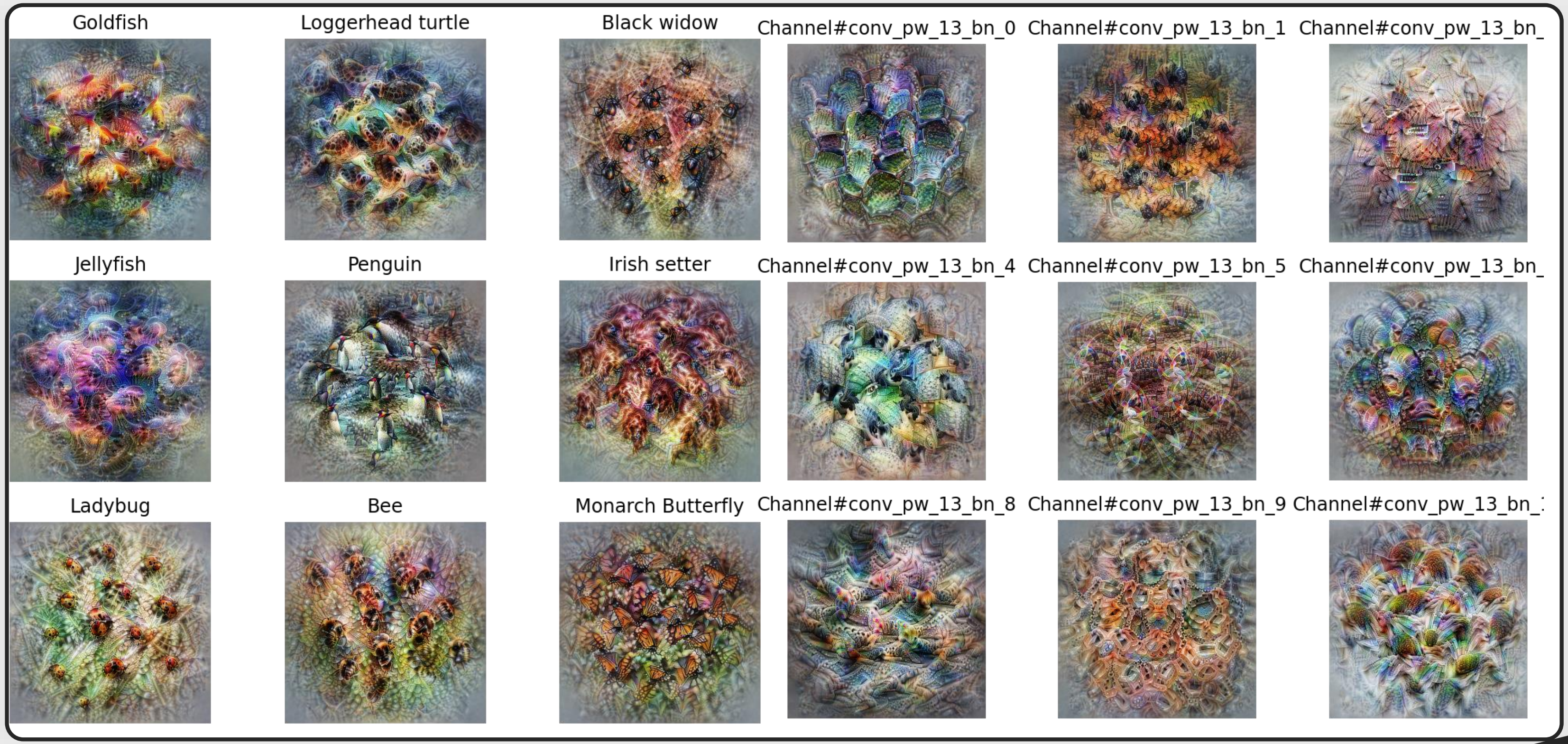
2. Feature Attributions

Tips & Tricks

- Always use multiple methods
- Sobol, Rise, Grad-Cam and Smoothgrad work in more cases
- Clip percentile before using `imshow(.)`
- Beware of confirmation bias
- The *quality* of the explanation does not depend only on the method! Robust models (e.g. 1-Lipschitz model) seem to give better explanations

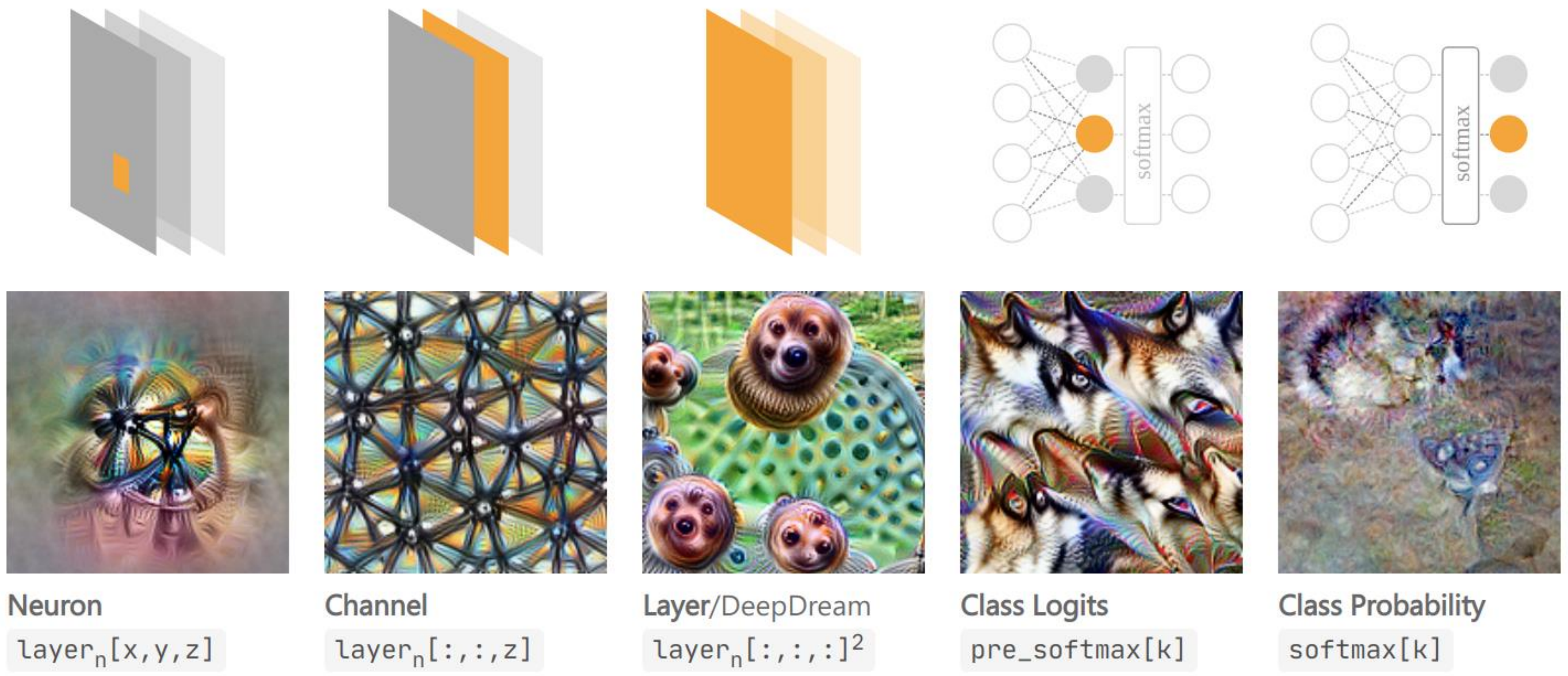
3. Feature Visualization



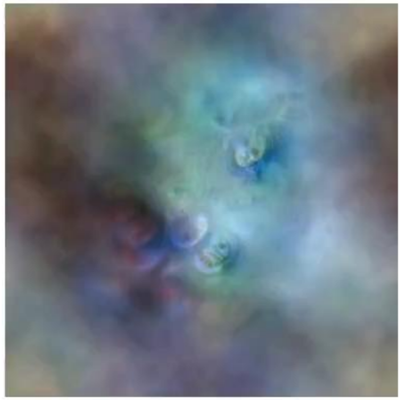


3. Feature Visualization

Overview

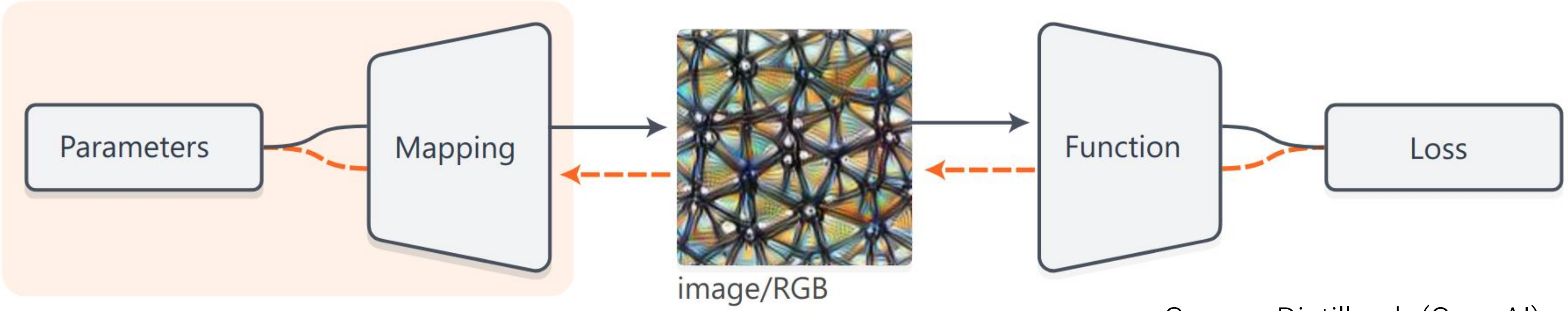


Source: Distill.pub (OpenAI)



3. Feature Visualization
Parameterization

FIGURE 1: As long as an image parameterization is differentiable, we can backpropagate (⟵ – –) through it.



$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{C}^d \;\; s.t \;\; \Omega(\mathbf{u})<\lambda_1}{arg\ max} \; \underset{\tau\in\mathcal{T}}{\mathbb{E}}(\mathbf{f}(\tau \circ \mathcal{F}^{-1}(\mathbf{u}))$$

$$\mathbf{x}^* = \mathcal{F}^{-1}(\mathbf{u}^*)$$

Source: Distill.pub (OpenAI)

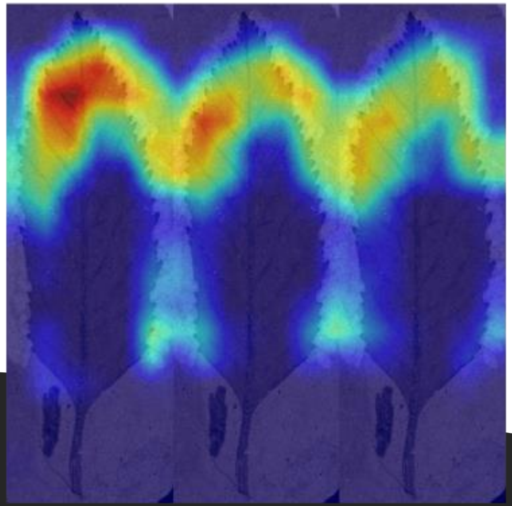
3. Feature Visualization

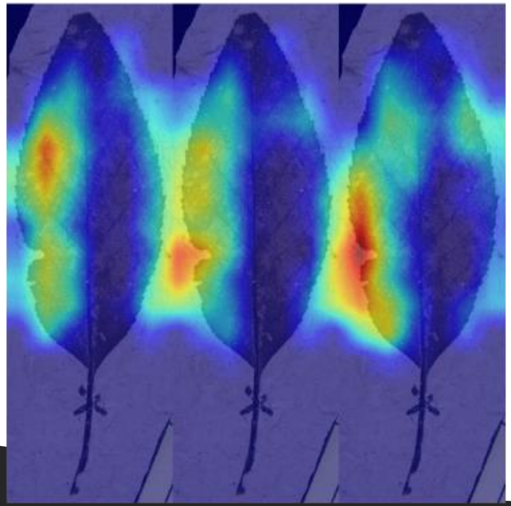
7.19 - Betulaceae

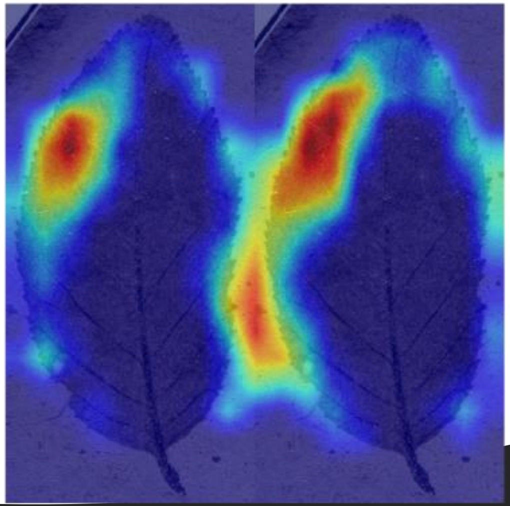
5.56 - Betulaceae

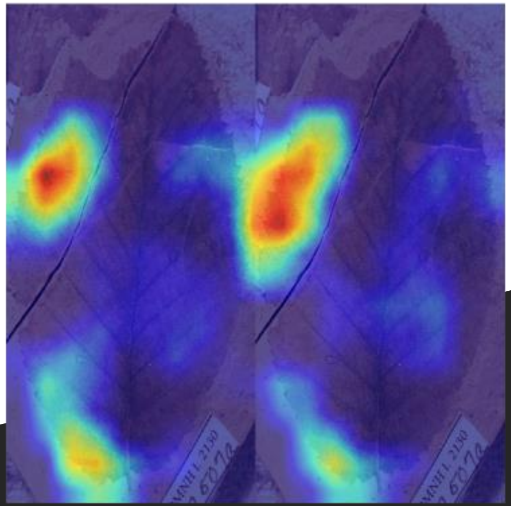
5.31 - Betulaceae

5.20 - Betulaceae

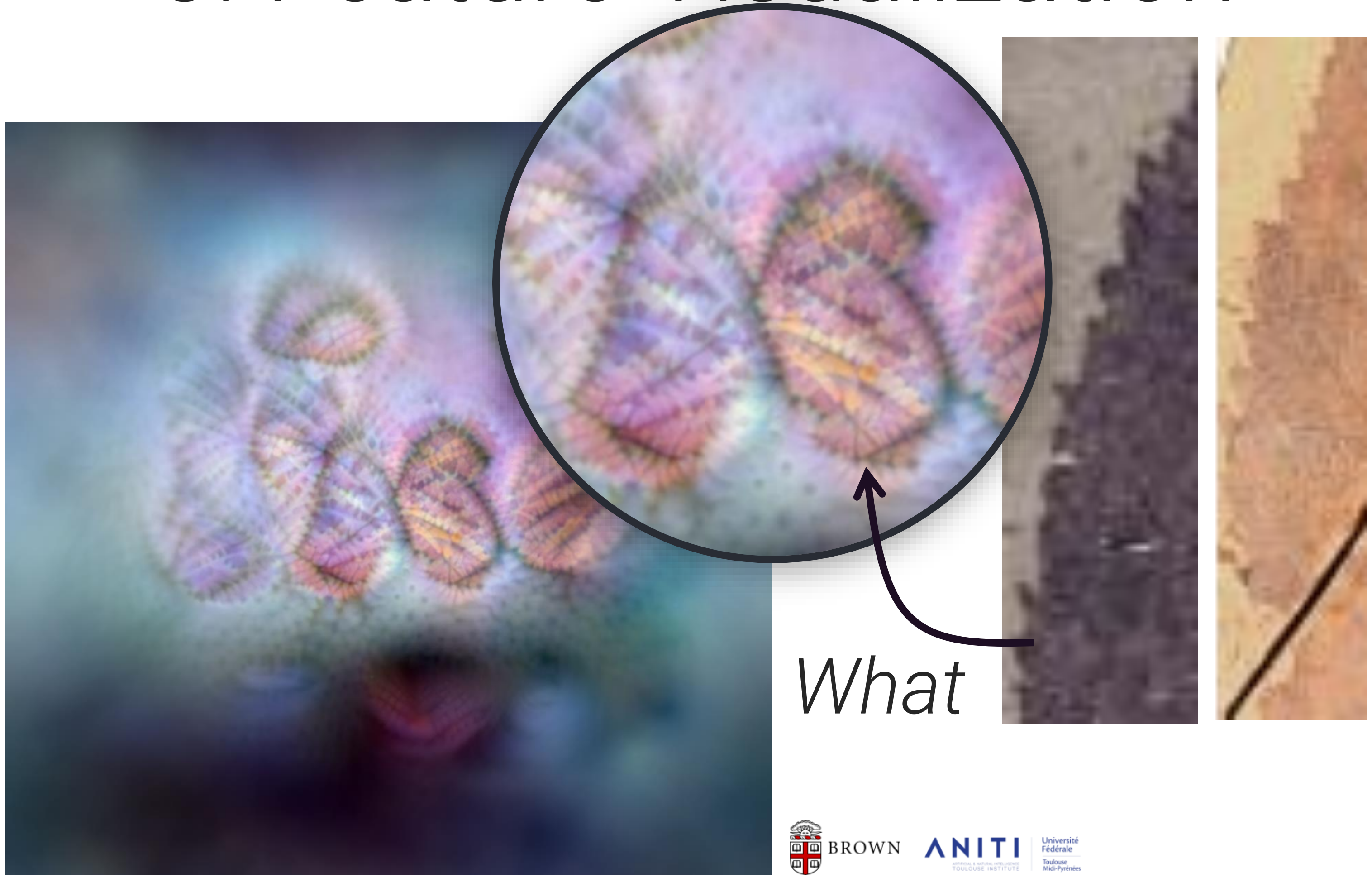


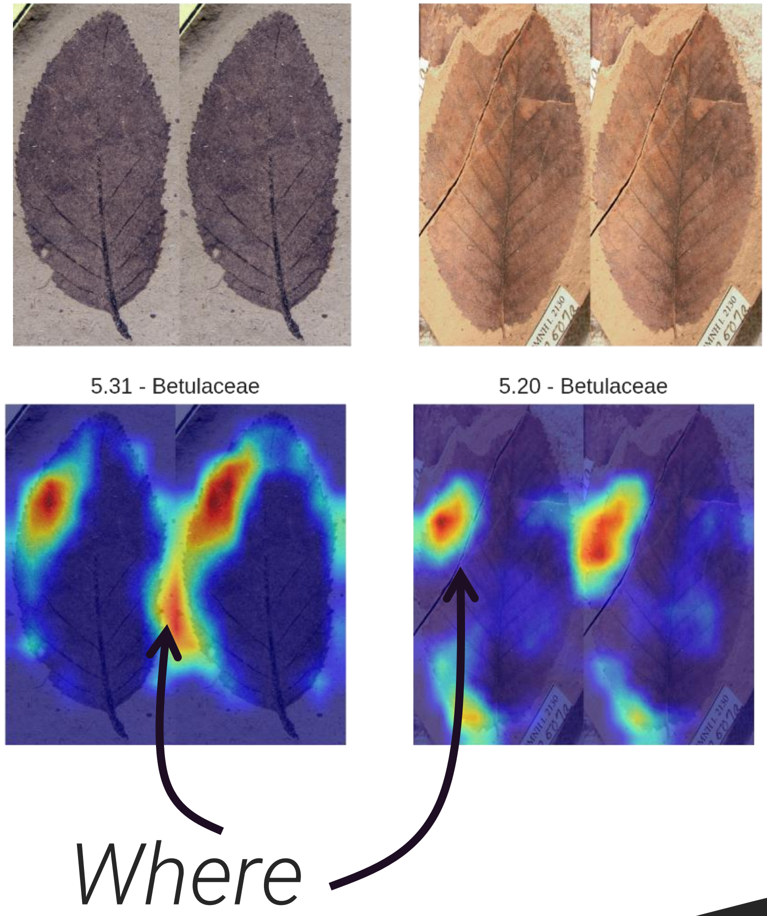






3. Feature Visualization





Paleo AI Project, Serre LAB
Ivan Felipe Rodriguez*, Thomas FEL*, Thomas Serre, Peter Wilf

3. Feature Visualization

Explaining logits

Demo

4. Testing with Concept Activation Vectors

Beyond feature Attribution

4. Testing with Concept Activation Vectors

TCAV

Demo

5. What's Next

Research

- Guiding model explanations during training – for models with human-like explanations
- Explaining models using data (Recursive Flow Explainable AI project)
- Influenciae Library (Influence Function library)

Implementation

- Adapt Xplique for NLP
- More Attribution methods
- Automatic Concept Extraction

5. See also

Libraries

- Lucid the wonderful library specialized in feature visualization from OpenAI.
- Captum the Pytorch library for Interpretability research
- Tf-explain that implement multiples attribution methods and propose callbacks API for tensorflow.
- Alibi Explain for model inspection and interpretation
- SHAP a very popular library to compute local explanations using the classic Shapley values from game theory and their related extensions implementation

Tutorials

- Interpretable Machine Learning by *Christophe Molnar*.
- Interpretability Beyond Feature Attribution by *Been Kim*.
- Explaining ML Predictions: State-of-the-art, Challenges, and Opportunities by *Himabindu Lakkaraju, Julius Adebayo and Sameer Singh*.
- A Roadmap for the Rigorous Science of Interpretability by *Finale Doshi-Velez*.
- **DEEL** White paper a summary of the **DEEL** team on the challenges of certifiable AI and the role of explainability for this purpose
- (AAAI 2022) On Explainable AI - XAI Coding And Engineering Practices – **Showcase Xplique !** 🎉

ANITI ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE DEEL DEpendable & Explainable Learning BROWN SNCF SAINT EXUPÉRY IVADO ACRIAQ

Thank you for your attention! Any questions?

https://github.com/deel-ai/xplique

Thomas FEL*, Lucas HERVIER*, David VIGOUROUX, Antonin POCHE, Thibault BOISSIN, Philippe DEJEAN, Benjamin DEPORTE, David PETITEAU, Justin PLAKOO
and all the **DEEL** *team*