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
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

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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

A Technical challenge

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

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**

- **Ex Ante (A priori)**
  - Local
    - Prototype Explanation Network
    - SENN
    - Explanation 'by design'
  - Global
    - Concept bottleneck model

- **Ex Post (Post hoc)**
  - Attributions Methods
  - Feature Inversion
  - Surrogate Model
    - Influence Functions
    - K-NN on latent space
  - Concept Activation Vectors
  - Feature Viz

(soon) Influenciae (Agustin Martin Picard)
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 $f$, which for some test input $x$ predicts the output $f(x)$.

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

![Feature Attribution Methods](image)
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]

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

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

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

---

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.

**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.

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

Parameters: N, baseline

---

[1] Axiomatic Attribution for Deep Networks
2. Feature Attributions
Explaining specific predictions

**Rise** Petsiuk & al (2018)[1]

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

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

\[ \phi_i = \mathbb{E}_{\mathcal{M} \sim_i} \left( \frac{\text{Var}_{\mathcal{M}_i}(f \circ \zeta(x, \mathcal{M}) | \mathcal{M} \sim_i)}{\text{Var}(f \circ \zeta(x, \mathcal{M}))} \right) \]

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

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. Feature Attributions

Explaining specific predictions

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

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

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. Feature Attributions

Explaining specific predictions

**Grad-CAM** Selvaraju et al (2017)

\[ w^{(k)} = \frac{1}{Z} \sum_i \sum_j \frac{\partial f(x)}{\partial A^{(k)}_{ij}} \]

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

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?’

```bash
pip install xplique
```
import xplique
from xplique.attributions import (Saliency,
SmoothGrad,
GradCAM)
import xplique
from xplique.attributions import (Saliency, SmoothGrad, GradCAM)
2. Feature Attributions

```python
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
]
```
2. Feature Attributions

Explaining specific predictions
2. Feature Attributions

Other data format!

Bounding Box
(notebook coming soon!)

Tabular data
& Even time series!
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

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

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

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

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

\[
f = \mathcal{A}(D)
\]
\[
f_{|x} = \mathcal{A}(D \setminus \{ x \})
\]

\[
\phi_x(f)
\]
\[
\phi_x(f_{|x})
\]

\[
g = \mathcal{A}(D)
\]
\[
g_{|x} = \mathcal{A}(D \setminus \{ x \})
\]

\[
\phi_x(g)
\]
\[
\phi_x(g_{|x})
\]

\text{\textbf{f} has Algorithmically Stable Explanations}

\text{\textbf{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**

```python
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
## 2. Feature Attributions

<table>
<thead>
<tr>
<th>Attribution Method</th>
<th>Type of Model</th>
<th>Source</th>
<th>Tabular Data</th>
<th>Images</th>
<th>Time-Series</th>
<th>Tutorial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deconvolution</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Grad-CAM</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Grad-CAM++</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Gradient Input</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Guided Backprop</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Integrated Gradients</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Kernel SHAP</td>
<td>Callable*</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Lime</td>
<td>Callable*</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Occlusion</td>
<td>Callable*</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Rise</td>
<td>Callable*</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>Saliency</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>SmoothGrad</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>SquareGrad</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
<tr>
<td>VarGrad</td>
<td>TF</td>
<td>Paper</td>
<td>✔️</td>
<td>✔️</td>
<td>WIP</td>
<td>Open in Colab</td>
</tr>
</tbody>
</table>

### Attribution Metrics

<table>
<thead>
<tr>
<th>Attribution Metric</th>
<th>Type of Model</th>
<th>Property</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>MuFidelity</td>
<td>TF</td>
<td>Fidelity</td>
<td>Paper</td>
</tr>
<tr>
<td>Deletion</td>
<td>TF</td>
<td>Fidelity</td>
<td>Paper</td>
</tr>
<tr>
<td>Insertion</td>
<td>TF</td>
<td>Fidelity</td>
<td>Paper</td>
</tr>
<tr>
<td>Average Stability</td>
<td>TF</td>
<td>Stability</td>
<td>Paper</td>
</tr>
<tr>
<td>MeGe</td>
<td>TF</td>
<td>Representativity</td>
<td>Paper</td>
</tr>
<tr>
<td>ReCo</td>
<td>TF</td>
<td>Consistency</td>
<td>Paper</td>
</tr>
</tbody>
</table>

(WIP) e-robustness
2. Feature Attributions
Fidelity doesn’t mean Usefulness


Thomas Fel1,3,4, Julien Colin1,3, Rémi Cadène1,2, Thomas Serre1,3
1Carney Institute for Brain Science, Brown University, USA. 2Sorbonne Université, CNRS, France. 3Artificial and Natural Intelligence Toulouse Institute, Université de Toulouse, France. 4Innovation & 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

Model
Feature Viz, Concept Activation Vector Explanation ‘by design’ ...

Predictions
Feature Attribution Feature Inversion ...

Data
Nearest Neighbourhood Influence Function ...

Goldfish
Loggerhead turtle
Black widow
Channel#conv_pw_13_bn_0
Channel#conv_pw_13_bn_1
Channel#conv_pw_13_bn_2

Jellyfish
Penguin
Irish setter
Channel#conv_pw_13_bn_4
Channel#conv_pw_13_bn_5
Channel#conv_pw_13_bn_6

Ladybug
Bee
Monarch Butterfly
Channel#conv_pw_13_bn_8
Channel#conv_pw_13_bn_9
Channel#conv_pw_13_bn_10
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.

\[
\begin{align*}
\mathbf{u}^* &= \arg \max_{\mathbf{u} \in \mathbb{C}^d} \Omega(\mathbf{u}) \quad \text{s.t.} \quad \lambda_1 \leq \tau \in T \\
\mathbb{E}(f(\tau \circ F^{-1}(\mathbf{u}))) \\
\mathbf{x}^* &= F^{-1}(\mathbf{u}^*)
\end{align*}
\]

Source: Distill.pub (OpenAI)
3. Feature Visualization
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
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 !**
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