# MODEL UNDERSTANDING FOR PYTORCH AND RECOMMENDER SYSTEMS

NARINE KOKHLIKYAN

07.24.2020

~TEAM CAPTUM~

KDD 2020

# AGENDA

01
Why model interpretability?

02
Intro to Captum Library

03
Debugging and visualizing models

04
Interpretability for Recommender Systems

05
Challenges & future directions

# WHAT IS MODEL INTERPRETABILITY ?

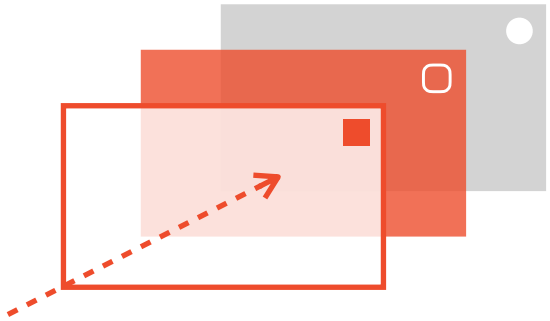
THE DEFINITION OF INTERPRETABILITY IN AI IS RATHER ELUSIVE BUT LET'S DEFINE IT AS

"THE ABILITY TO DESCRIBE AI MODEL INTERNALS AND PREDICTIONS IN HUMAN UNDERSTANDABLE TERMS*"

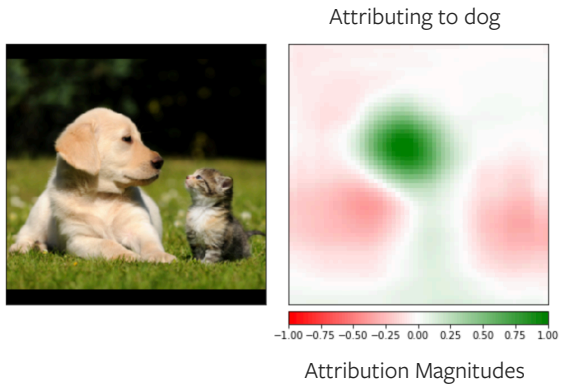
* LH Gilpin, et. al., Explaining explanations: An overview of interpretability of machine learning in IEEE 5th International Conference on data science and advanced analytics (DSAA), 2018

# MODEL INTERPRETABILITY

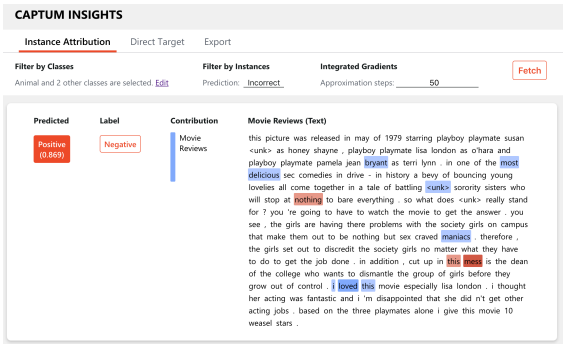
## INCREASED TRANSPARENCY



## BETTER UNDERSTANDING




## DEBUGGING

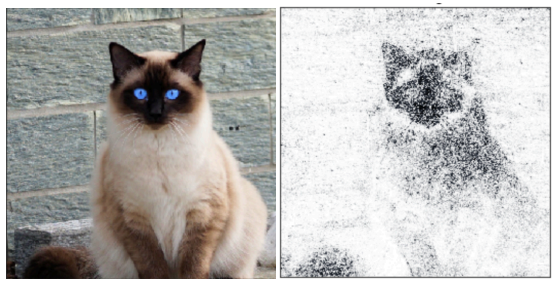


# HOW CAN WE MAKE INTERPRETABILITY ALGORITHMS ACCESSIBLE TO ALL PYTORCH MODEL DEVELOPERS?

# Captum

MODEL INTERPRETABILITY LIBRARY FOR PYTORCH

## MULTIMODAL

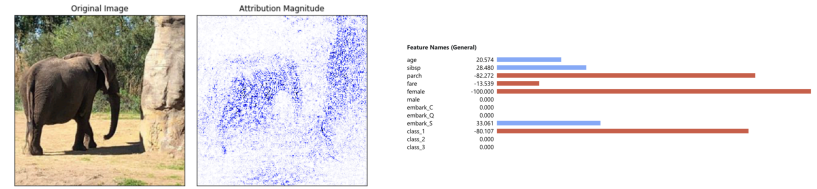



## EXTENSIBLE

```
class MyAttribution(Attribution):

   def attribute(self, input, ...):
       attributions = self._compute_attrs(input, ... )
       # <Add any logic necessary for attribution>
        return attributions
```

## EASY TO USE

visualize_image_attr(attr_algo.attribute(input), ...)




this movie is awful , just awful . someone bought it for me as a christmas present because they knew i liked a good horror flick . i do n't think they understood the " good " part . all i can say is next year this person is getting slipper socks from me . avoid this movie-- it makes you bitter . peace.<br /><br / >

captum.ai

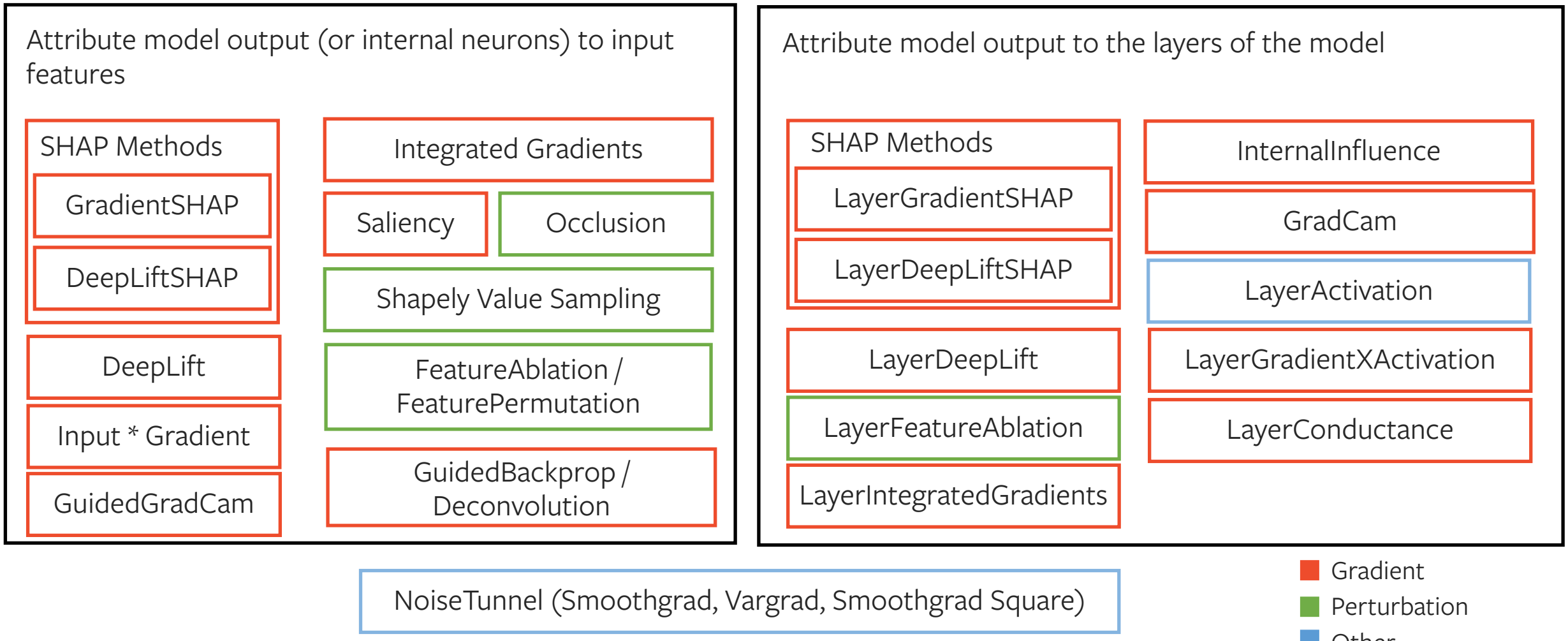
# WHAT DOES THE CAPTUM LIBRARY OFFER?

Attribution algorithms to interpret:

+ Output predictions with respect to inputs
+ Output predictions with respect to all neurons in the layers
+ Neurons with respect to inputs

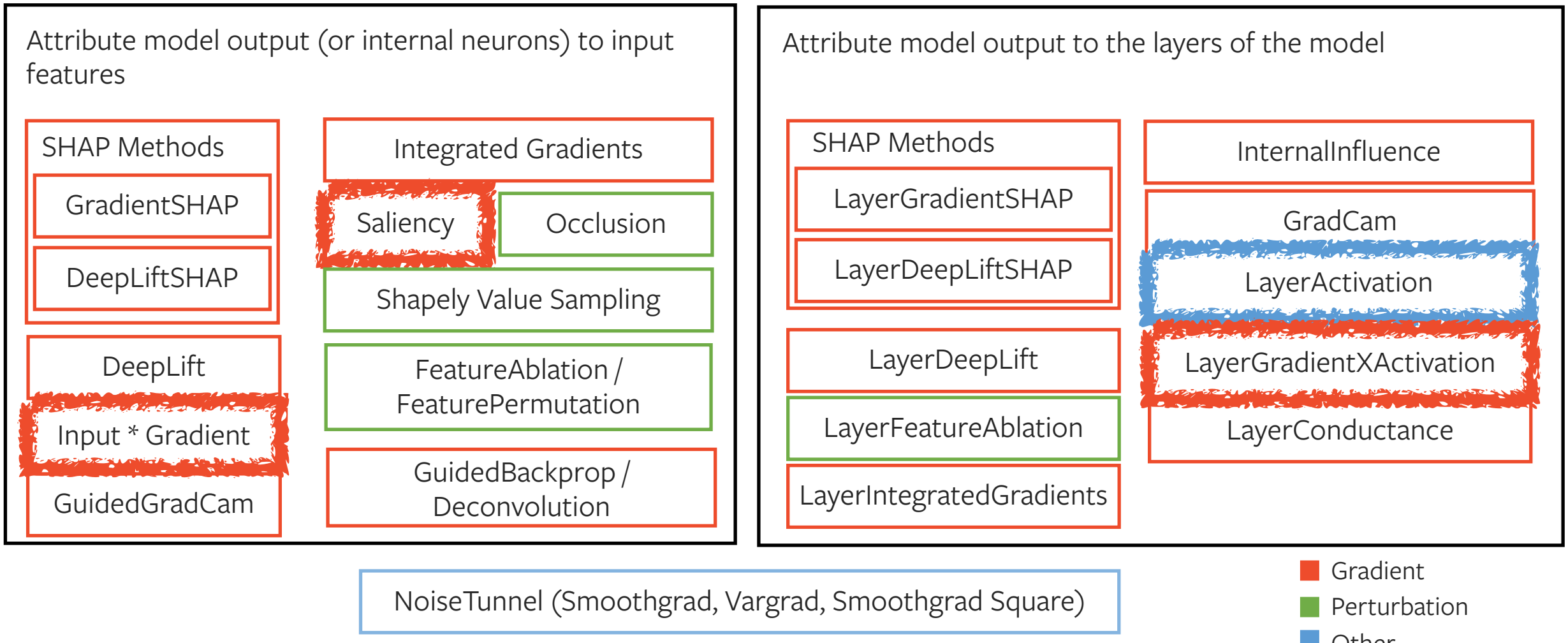
Currently we supports gradient and perturbation based algorithms

# ATTRIBUTION ALGORITHMS



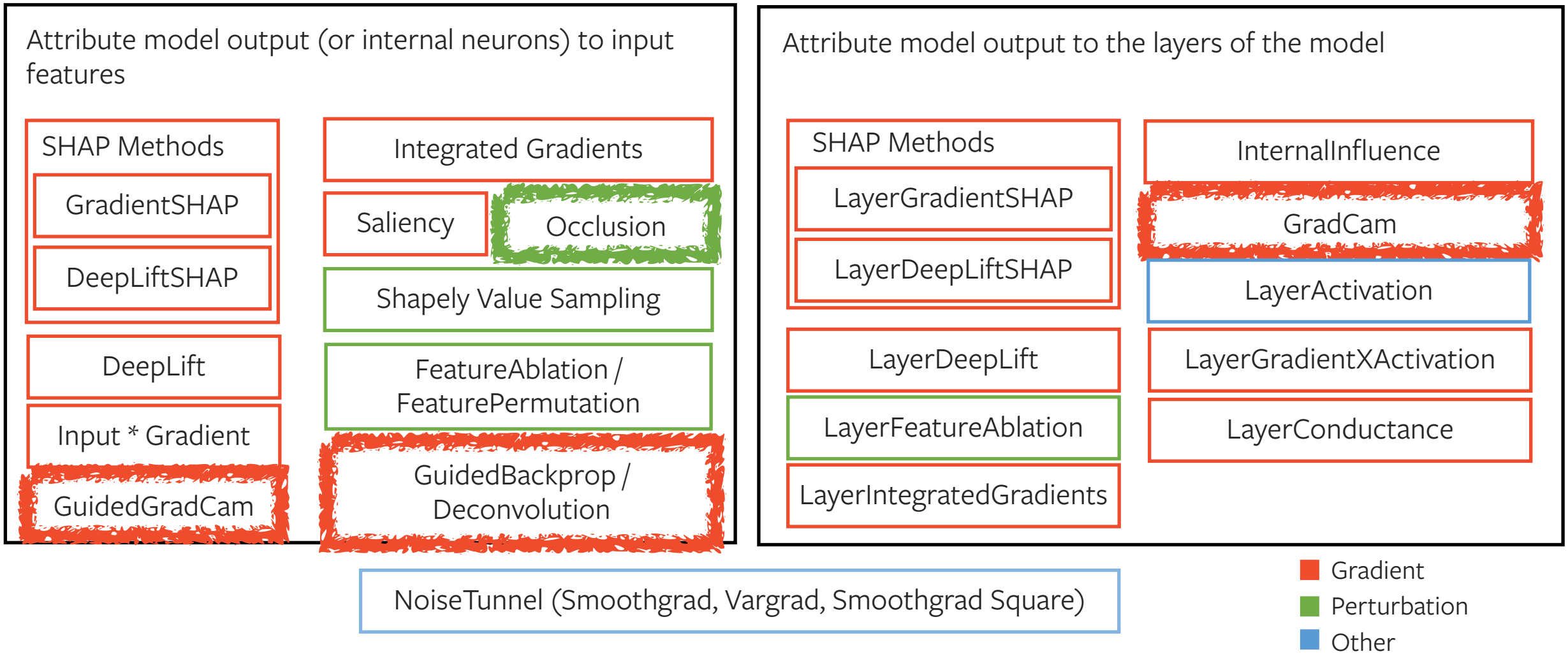
# ATTRIBUTION ALGORITHMS

*** Simple baseline approaches based on gradients, activations and inputs**



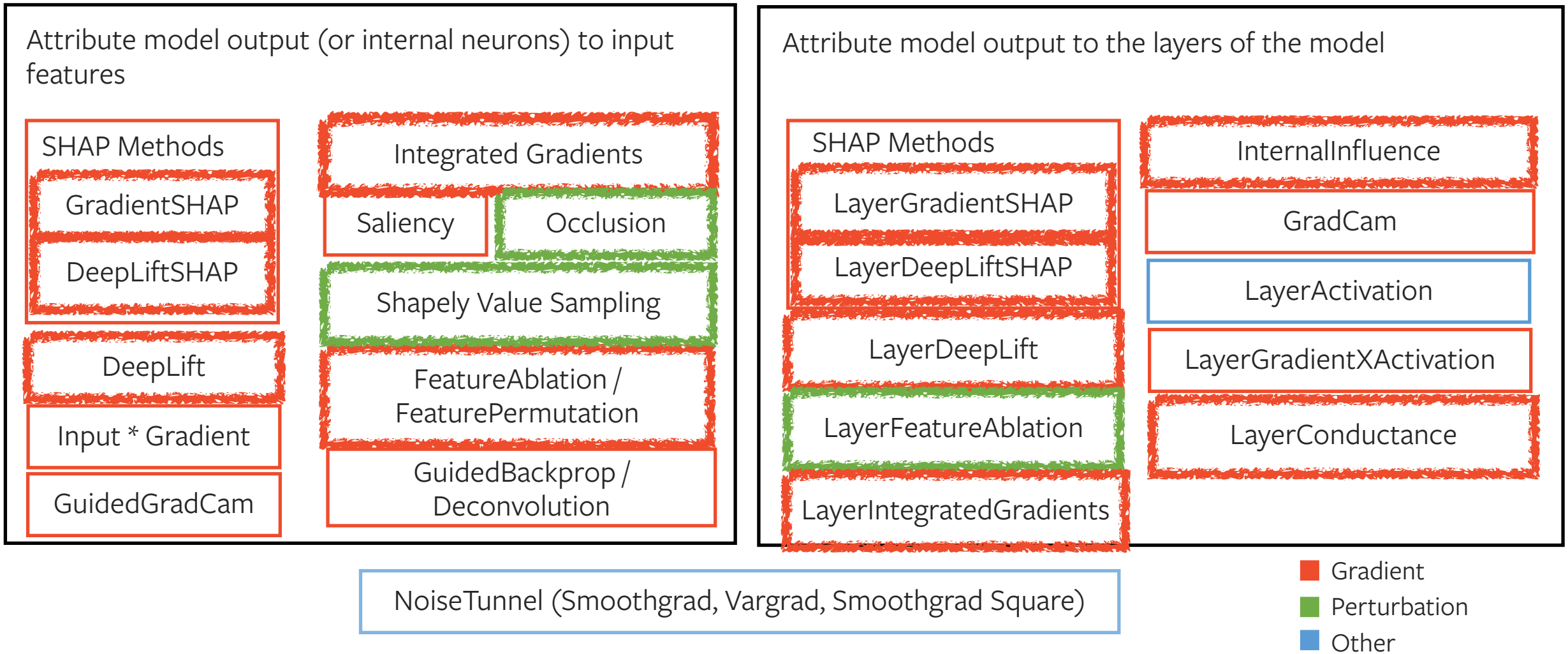
# ATTRIBUTION ALGORITHMS

*** Often used for computer vision models**



# ATTRIBUTION ALGORITHMS

*** Algorithms requiring baseline / reference / background**



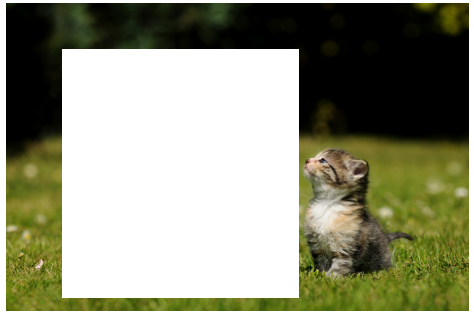
# WHAT IS BASELINE ?

## WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**

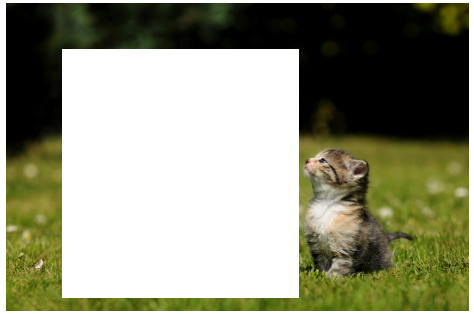
## WHAT IS BASELINE ?

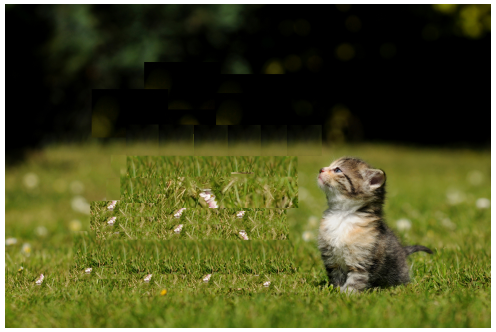
Comparing **input** with the **reference / baseline / background**



## WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**





## WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**

″Those who know, do.
Those that
understand, teach.″
~Aristotle

<PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD>

″Those who know, do.
Those that
understand, teach.″
~Aristotle

″Those who know, do.
Those that **<PAD>**,
teach.″ ~Aristotle

# WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | 33.3 | 0.0 |
| 3.9 | 45.8 | 3.0 |
| 21.0 | 0.0 | 5.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | 33.3 | 0.0 |
| 3.9 | 45.8 | 3.0 |
| 21.0 | 0.0 | 5.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | **0.0** | 0.0 |
| 3.9 | **0.0** | 3.0 |
| 21.0 | **0.0** | 5.0 |

Ablation

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | 33.3 | 0.0 |
| 3.9 | 45.8 | 3.0 |
| 21.0 | 0.0 | 5.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| **3.9** | 33.3 | 0.0 |
| **21.0** | 45.8 | 3.0 |
| **12.3** | 0.0 | 5.0 |

Permutation

## WHAT IS BASELINE ?

Comparing **input** with the **reference / baseline / background**

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | 33.3 | 0.0 |
| 3.9 | 45.8 | 3.0 |
| 21.0 | 0.0 | 5.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | 33.3 | 0.0 |
| 3.9 | 45.8 | 3.0 |
| 21.0 | 0.0 | 5.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | 33.3 | 0.0 |
| 3.9 | 45.8 | 3.0 |
| 21.0 | 0.0 | 5.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 |

| feat_1 | feat_2 | feat_3 |
|---|---|---|
| 12.3 | **0.0** | 0.0 |
| 3.9 | **0.0** | 3.0 |
| 21.0 | **0.0** | 5.0 |

Ablation

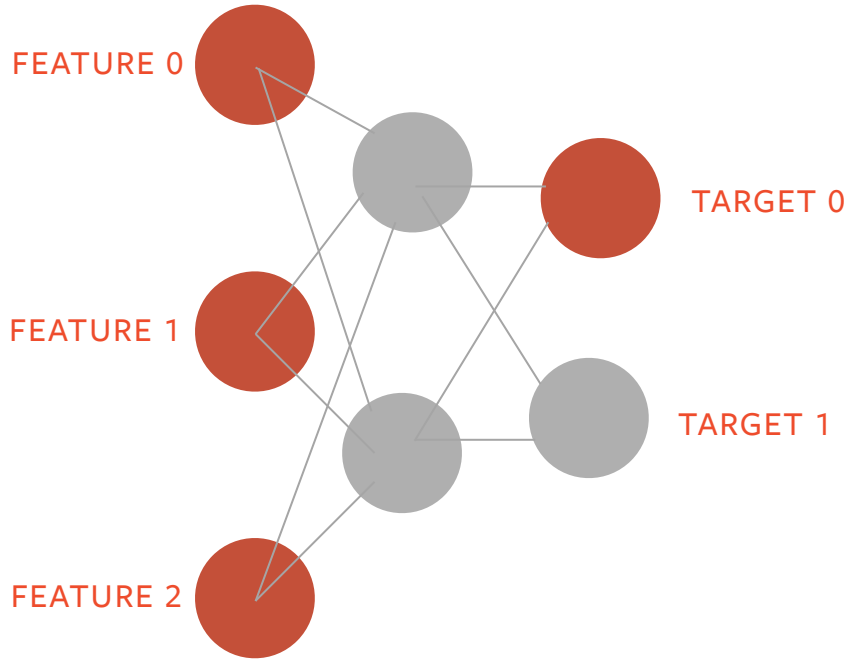
| feat_1 | feat_2 | feat_3 |
|---|---|---|
| **3.9** | 33.3 | 0.0 |
| **21.0** | 45.8 | 3.0 |
| **12.3** | 0.0 | 5.0 |

Permutation

The choice of the reference/baseline/background is challenging!

# How can we use Captum library ?

# PRIMARY ATTRIBUTION

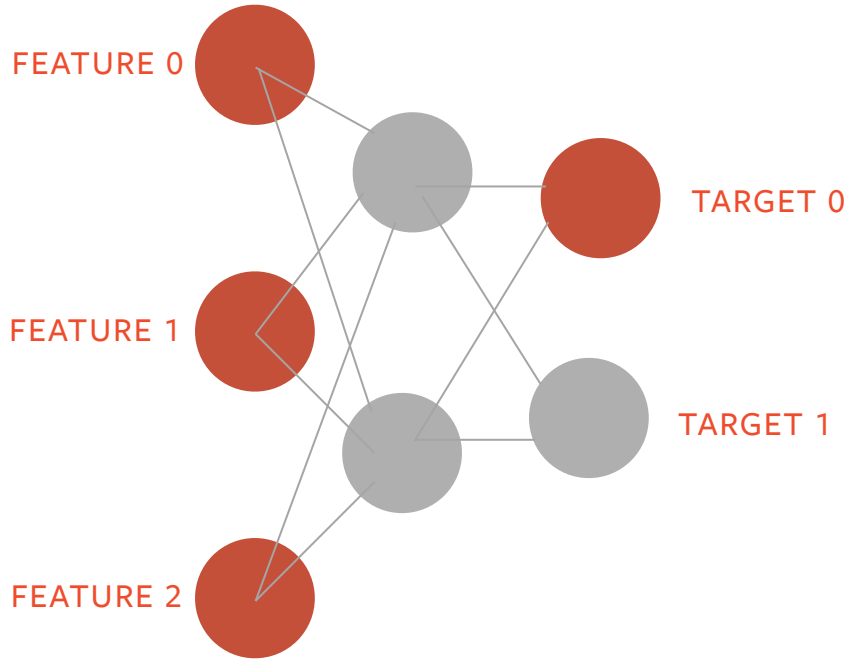



```
from captum.attr import IntegratedGradients

attr_algo = IntegratedGradients(model)


input = torch.rand(1, 3)

attributions = attr_algo.attribute(input, target=0)
```

## Results

```
attributions: [[0.1, -0.3, 0.0]]
```

# PRIMARY ATTRIBUTION

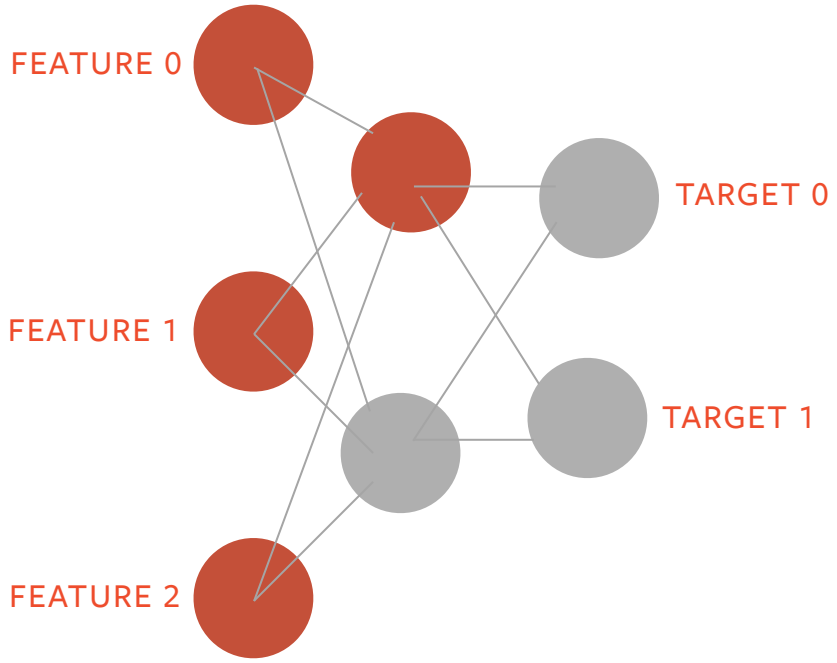



```
from captum.attr import IntegratedGradients

attr_algo = IntegratedGradients(model)


input = torch.rand(1, 3)

baseline = torch.zeros(1, 3)

attributions = attr_algo.attribute(input,
                                   baselines=baseline
                                   target=0)
```

## Results

```
attributions: [[0.1, -0.3, 0.0]]
```

# NEURON ATTRIBUTION

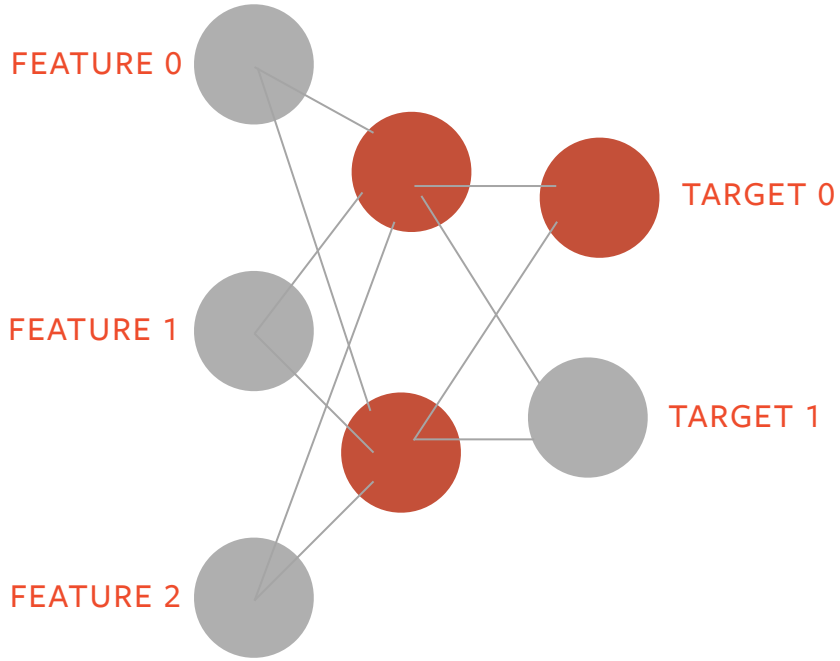



```
from captum.attr import NeuronConductance

attr_algo = NeuronConductance(model, model.lin)


input = torch.rand(1, 3)


attributions = attr_algo.attribute(input,
                                   neuron_ind = 0)
```

## Results

```
attributions: [[0.0, -0.03, -0.01]]
```

# LAYER ATTRIBUTION




```
from captum.attr import LayerConductance

attr_algo = LayerConductance(model, model.lin)


input = torch.rand(1, 3)

attributions = attr_algo.attribute(input,
                                   target = 0)
```

## Results

```
attributions: [[0.2, 0.0]]
```

attributions = Attribution(forward_func, ...).attribute(input, ...)*

* Check out our Getting Started docs and API:
https://github.com/pytorch/captum
https://captum.ai/api/

## SCALING ALGORITHMS

+ PyTorch DataParallel models are supported across all algorithms
  + Including distributed execution of forward / backward hooks

+ Optional internal slicing of large input tensors
  + Computing forward/backward passes on smaller slices of inputs and aggregating outputs

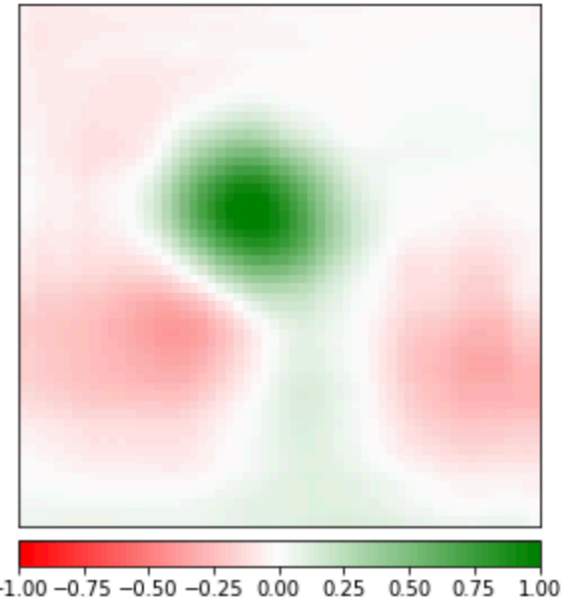
+ Perturbing multiple features together on the basis of batches

# HOW CAN WE VISUALIZE THE ATTRIBUTIONS OF COMPLEX MODELS?

# VISUALIZATIONS USING RESNET152 MODEL

ORIGINAL IMAGE

ATTRIBUTING TO DOG*

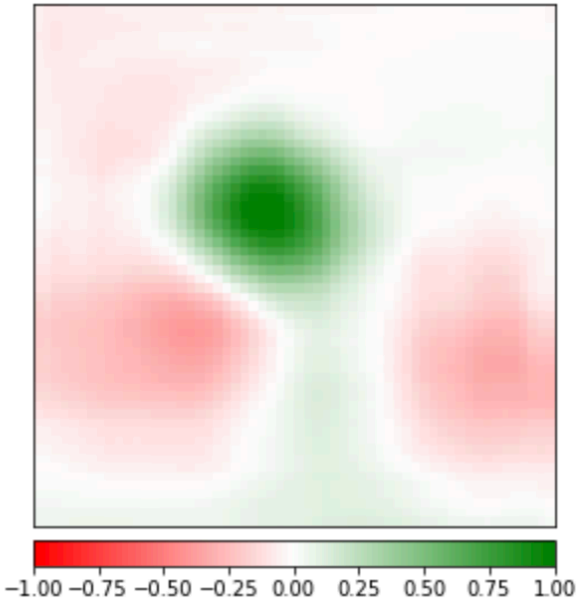



* MATTHEW D ZEILER, ROB FERGUS, OCCLUSION: VISUALIZING AND UNDERSTANDING CONVOLUTIONAL NETWORKS, IN SPRINGER INTERNATIONAL PUBLISHING SWITZERLAND, 2014

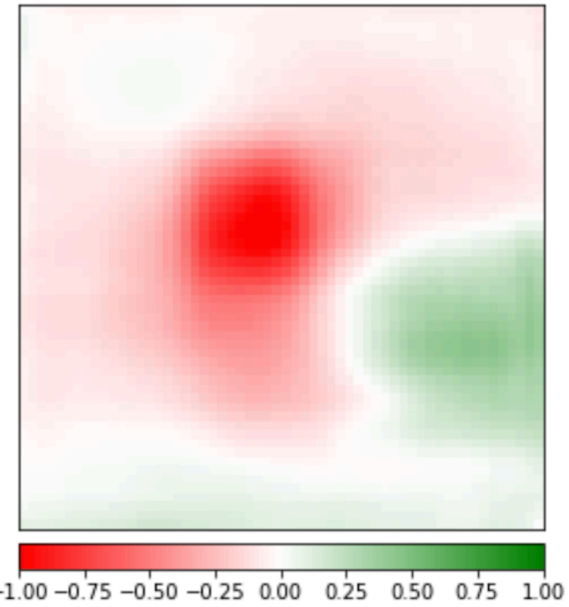
# VISUALIZATIONS USING RESNET152 MODEL

ORIGINAL IMAGE

ATTRIBUTING* TO DOG




ATTRIBUTING* TO CAT

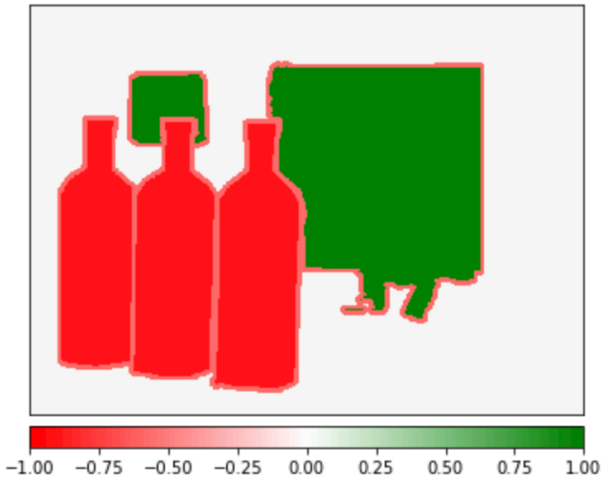



* MATTHEW D ZEILER, ROB FERGUS, OCCLUSION: VISUALIZING AND UNDERSTANDING CONVOLUTIONAL NETWORKS, IN SPRINGER INTERNATIONAL PUBLISHING SWITZERLAND, 2014

# VISUALIZATIONS USING RESNET18 MODEL AND VOC SEGMENTATION

## FEATURE ABLATION BASED ON IMAGE SEGMENTATION

ATTRIBUTING TO MONITOR



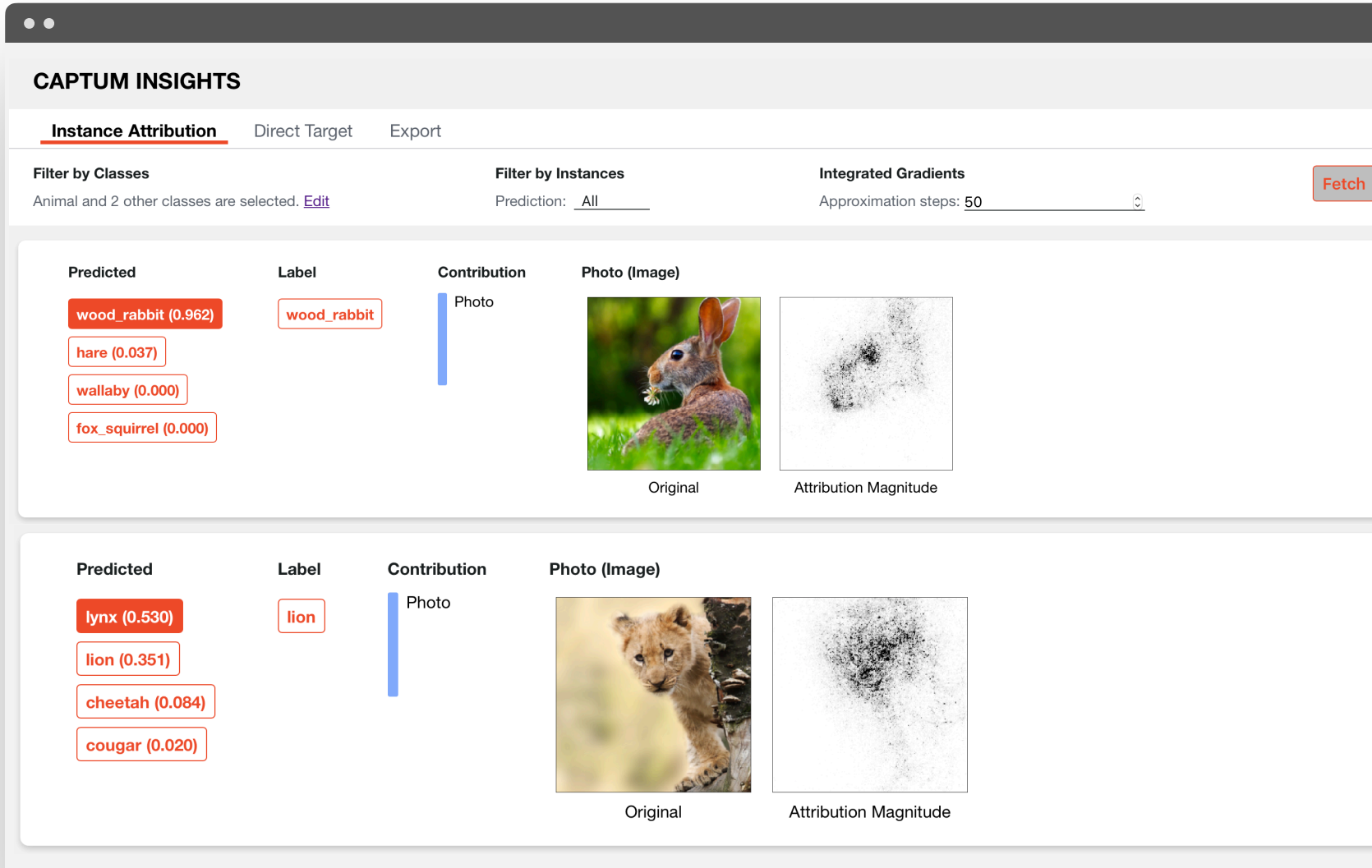
# CAPTUM INSIGHTS

An interactive visualization tool for
+ debugging and understanding model predictions

Supports different types of PyTorch models and input features

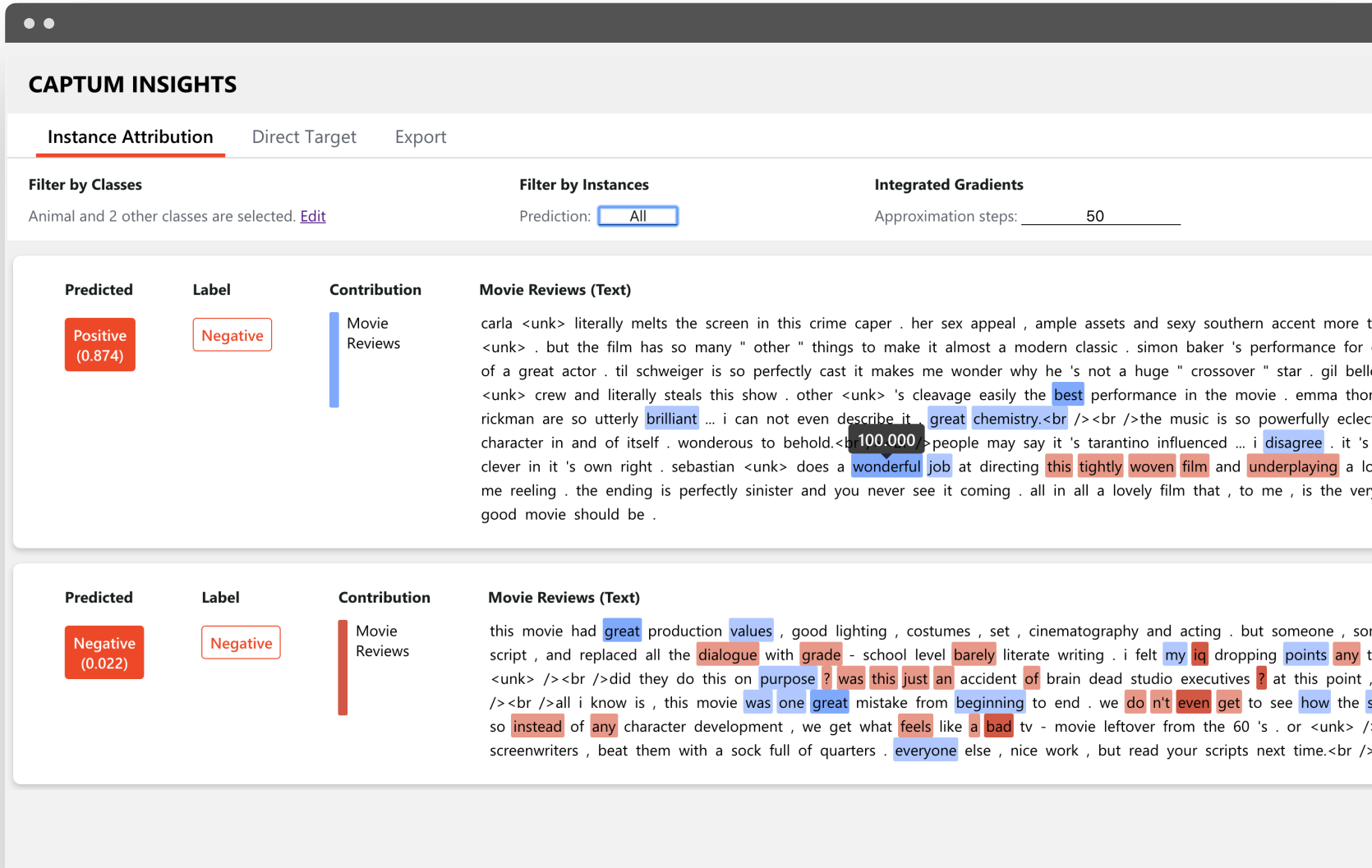
# VISUALIZING EXPLANATIONS OF RESNET-50 MODEL

## WITH CAPTUM INSIGHTS



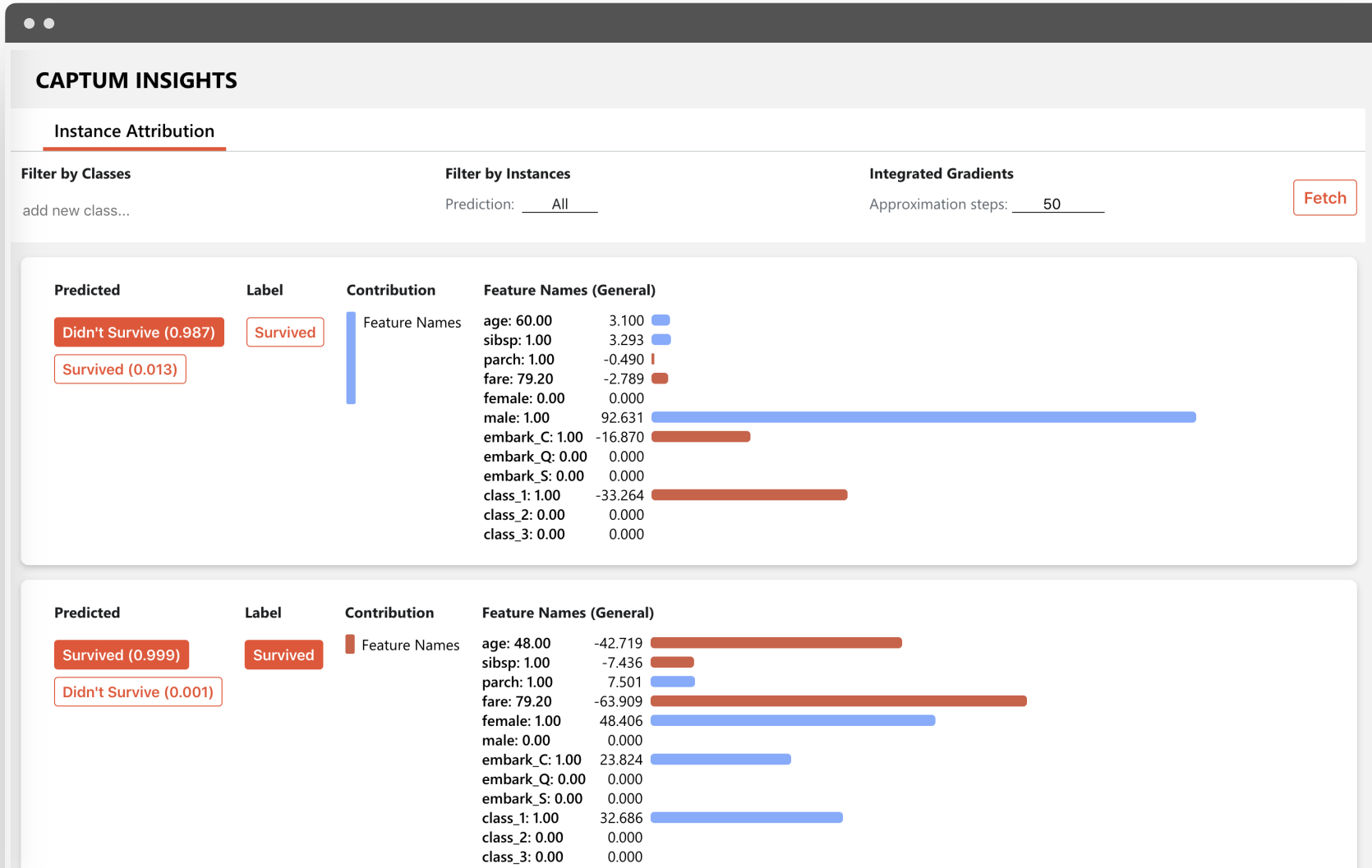
# VISUALIZING EXPLANATIONS OF A TEXT CLASSIFICATION MODEL USING IMDB DATASET

## WITH CAPTUM INSIGHTS

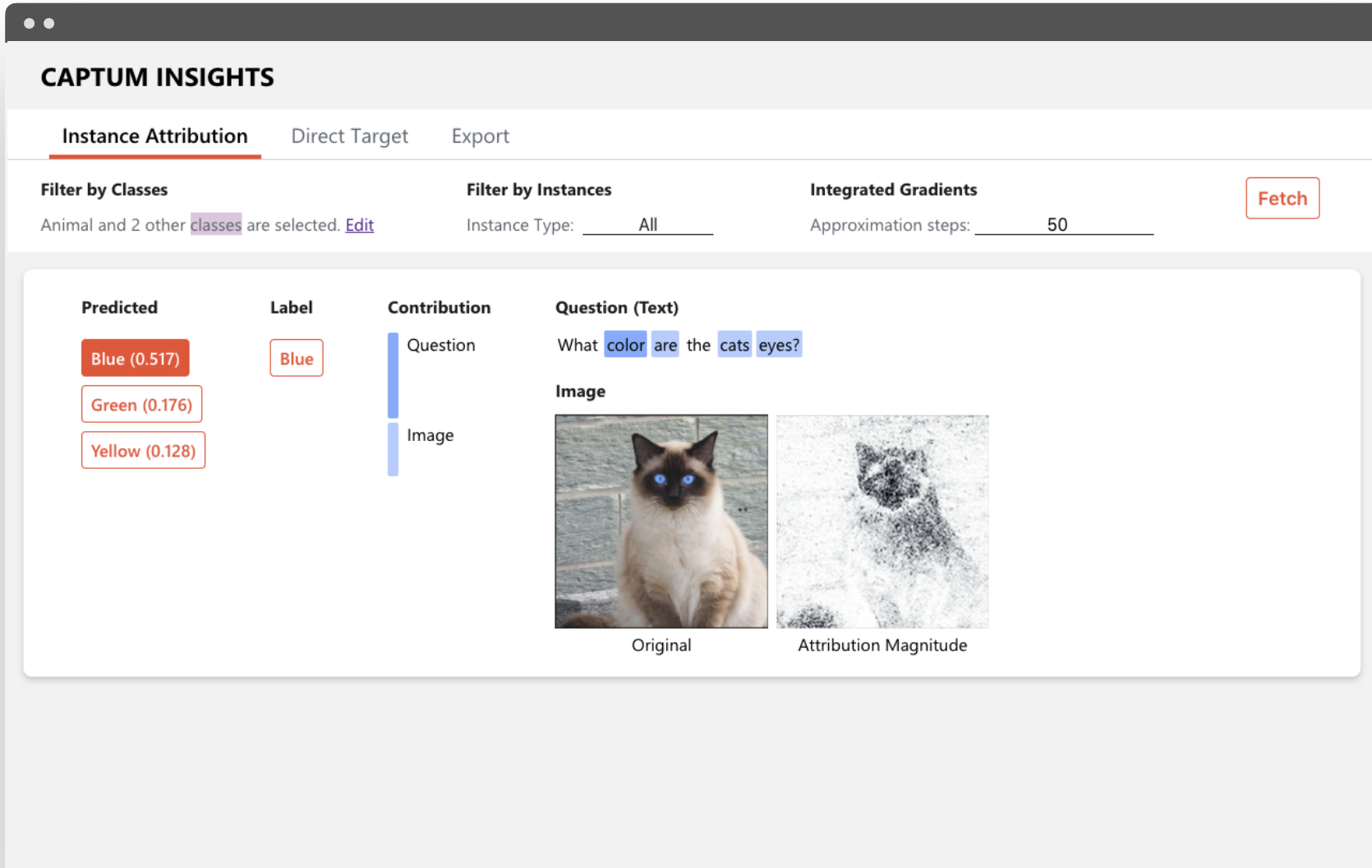


# VISUALIZING EXPLANATIONS OF A 3-LAYER MLP MODEL USING TITANIC DATASET

## WITH CAPTUM INSIGHTS



# VISUALIZING EXPLANATIONS OF MULTIMODAL VQA MODELS



# CASE STUDY FOR DEEP LEARNING RECOMMENDER MODEL (DLRM)

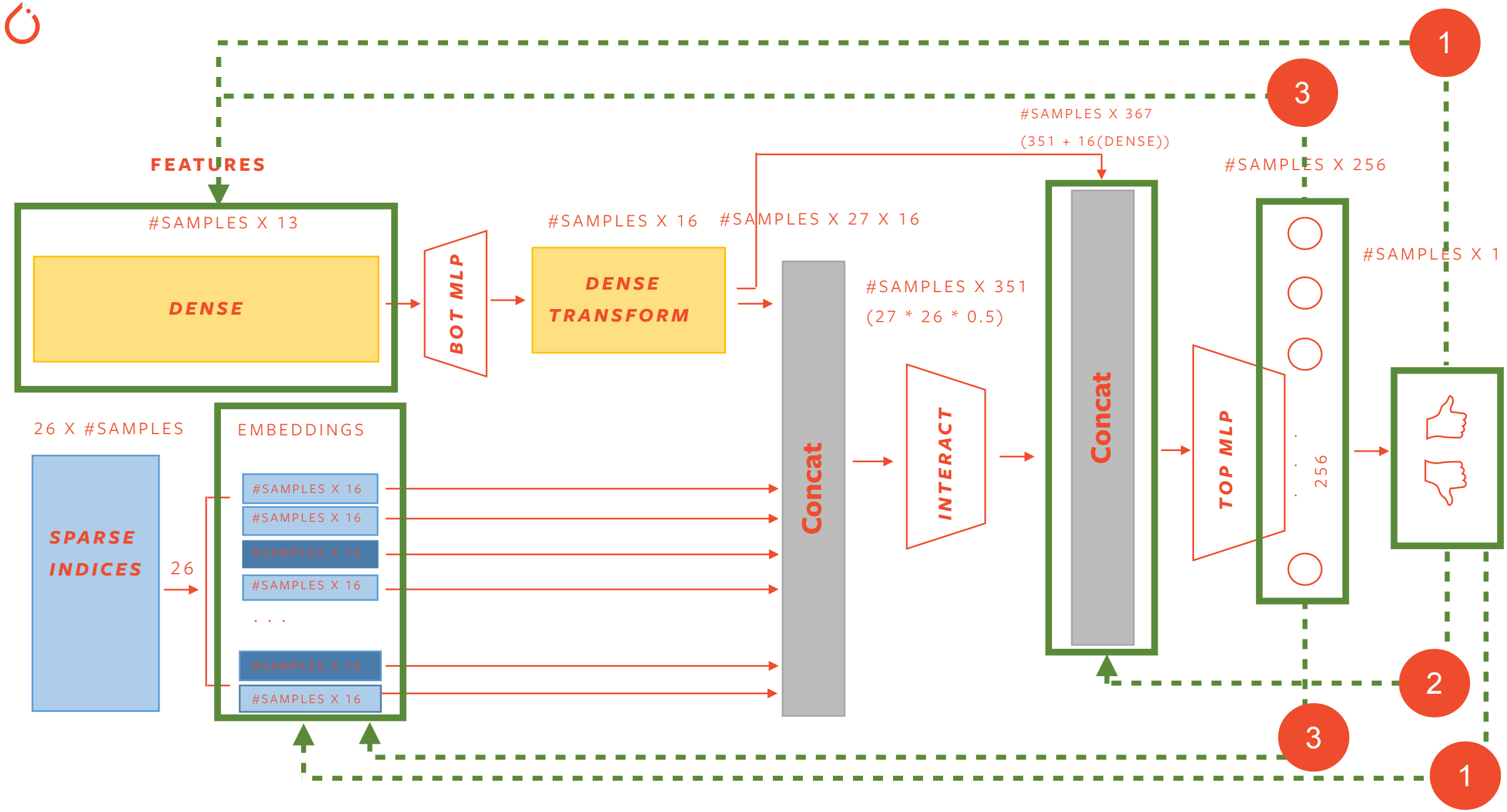
# CASE STUDY OVERVIEW

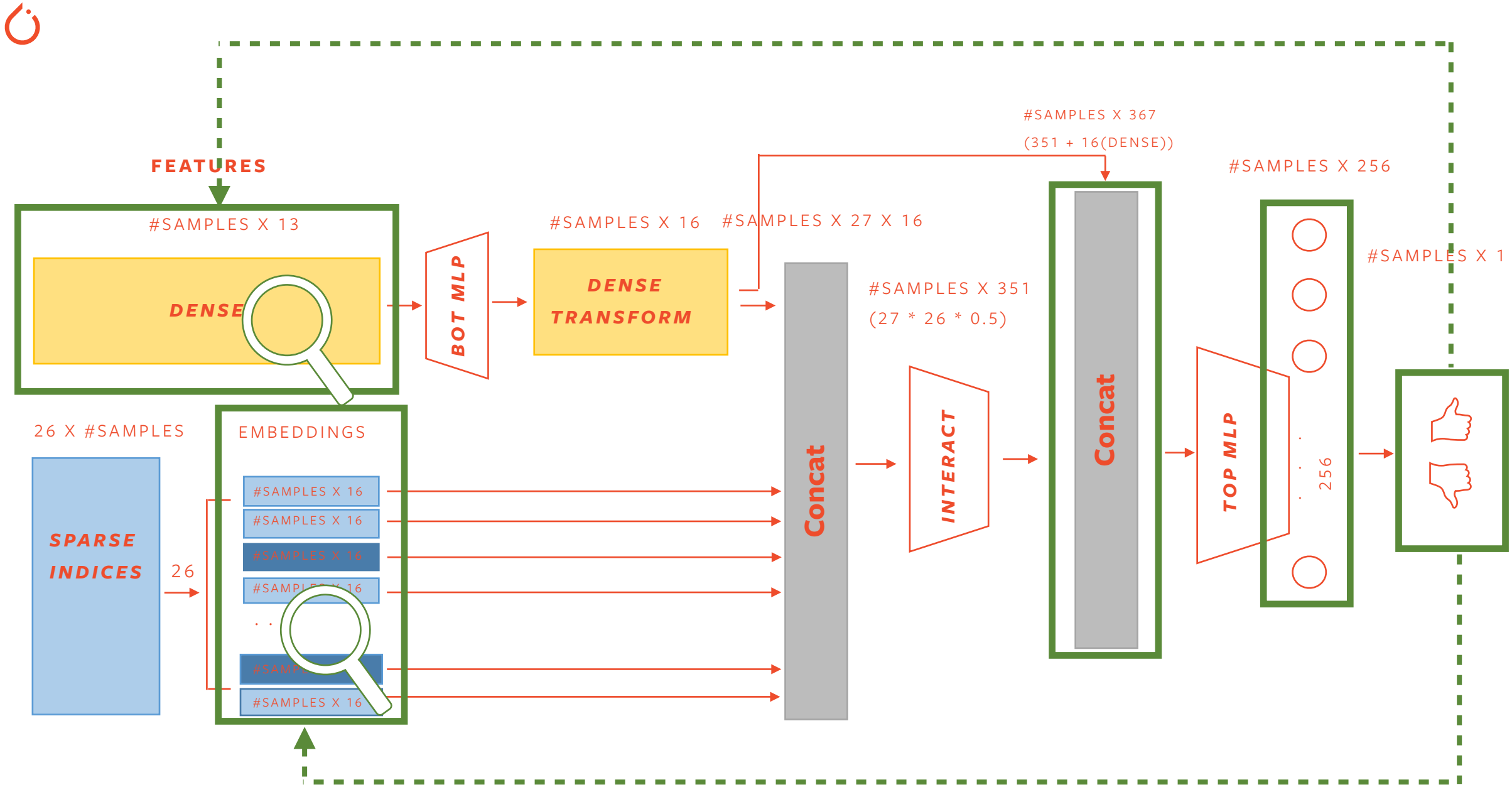
1. DLRM model and dataset
2. Dense and Sparse Feature importances
3. Feature importances in interaction layer
4. Neuron Importances for the last FC layer
5. Model pruning & performance

# DATASET

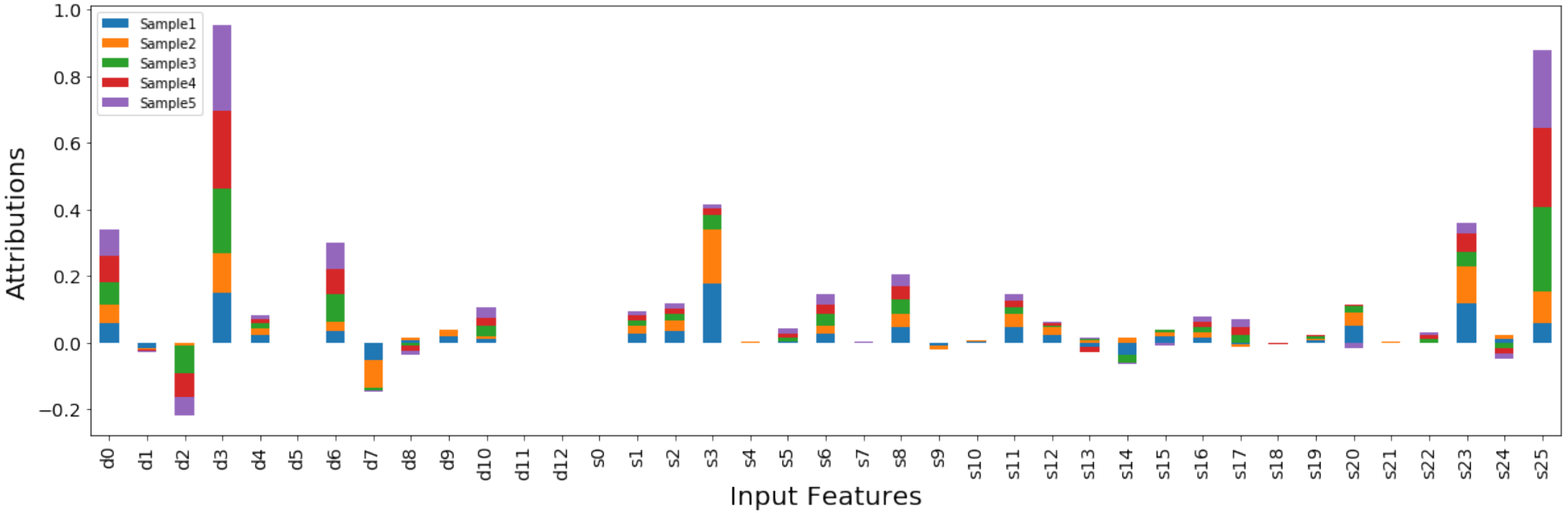
+ Criteo's traffic over a period of 7 days*
    + train: ~ 39M samples
    + test: ~ 3.2M samples
+ Highly unbalanced dataset
    + clicks: 26%
+ Features
    + sparse: 26
    + dense: 13

*https://www.kaggle.com/c/criteo-display-ad-challenge/data

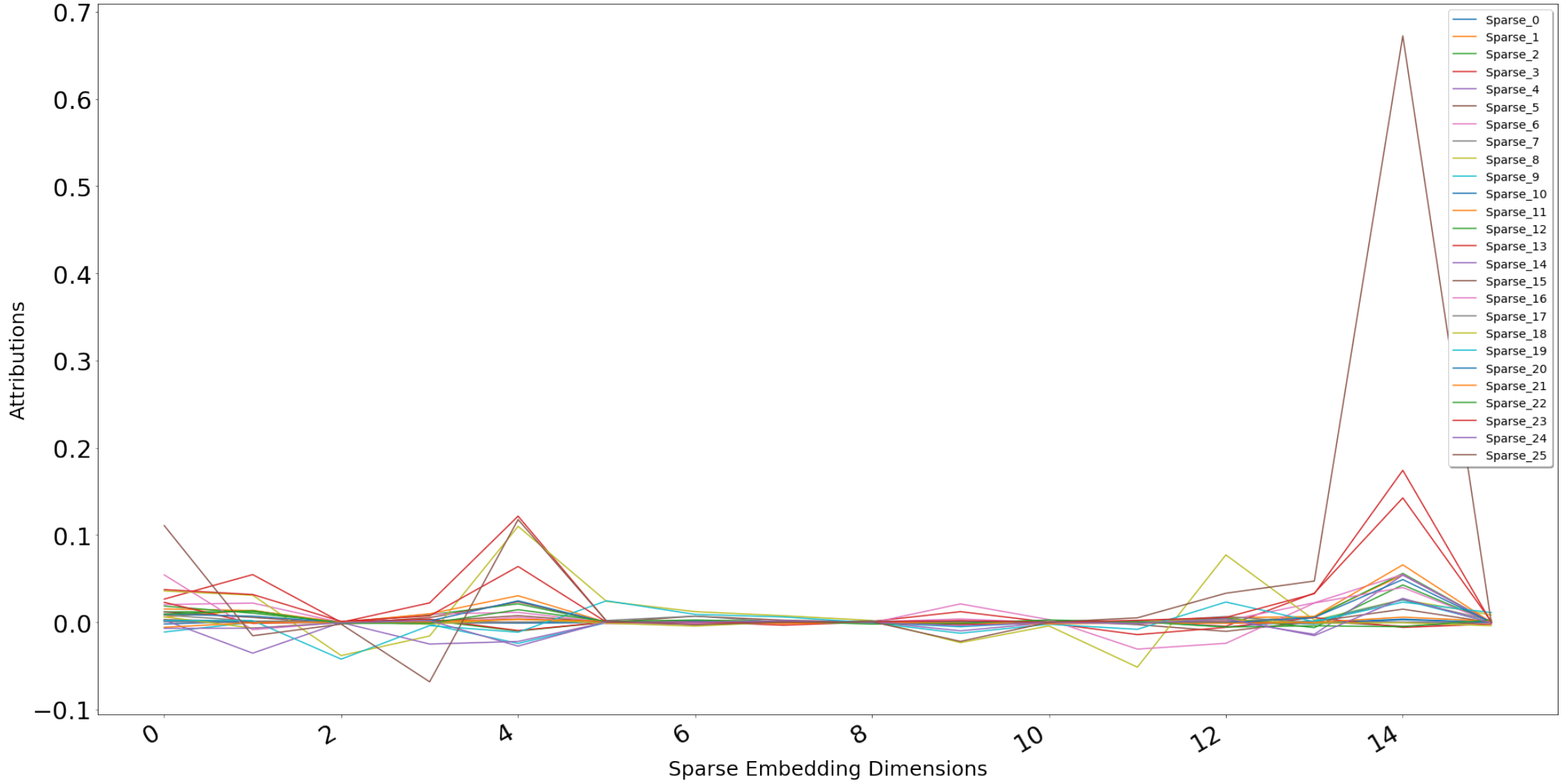

FEATURES
#SAMPLES X 13
DENSE
BOT MLP
#SAMPLES X 16
DENSE TRANSFORM
#SAMPLES X 27 X 16
#SAMPLES X 367
(351 + 16(DENSE))
#SAMPLES X 256
#SAMPLES X 1
26 X #SAMPLES
EMBEDDINGS
SPARSE INDICES
26
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
Concat
#SAMPLES X 351
(27 * 26 * 0.5)
INTERACT
Concat
TOP MLP
256
1
3
2
3
1


FEATURES
#SAMPLES X 13
DENSE
BOT MLP
#SAMPLES X 16
DENSE TRANSFORM
#SAMPLES X 27 X 16
#SAMPLES X 367
(351 + 16(DENSE))
#SAMPLES X 256
#SAMPLES X 1
26 X #SAMPLES
EMBEDDINGS
SPARSE INDICES
26
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
Concat
#SAMPLES X 351
(27 * 26 * 0.5)
INTERACT
Concat
TOP MLP
256

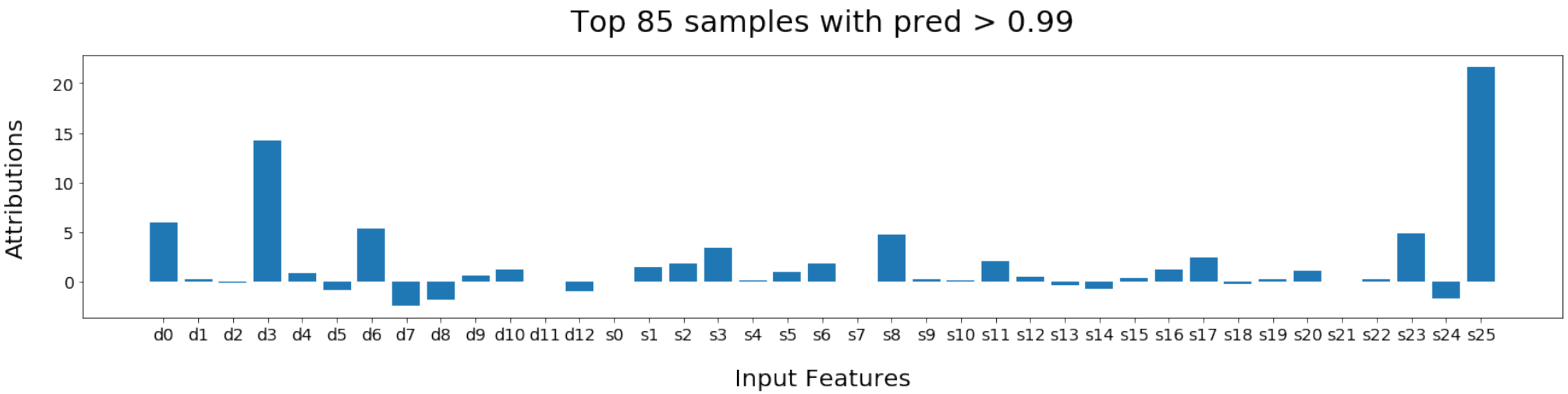
# FEATURE IMPORTANCES FOR 5 SAMPLES WITH PREDICTION SCORE > 0.999



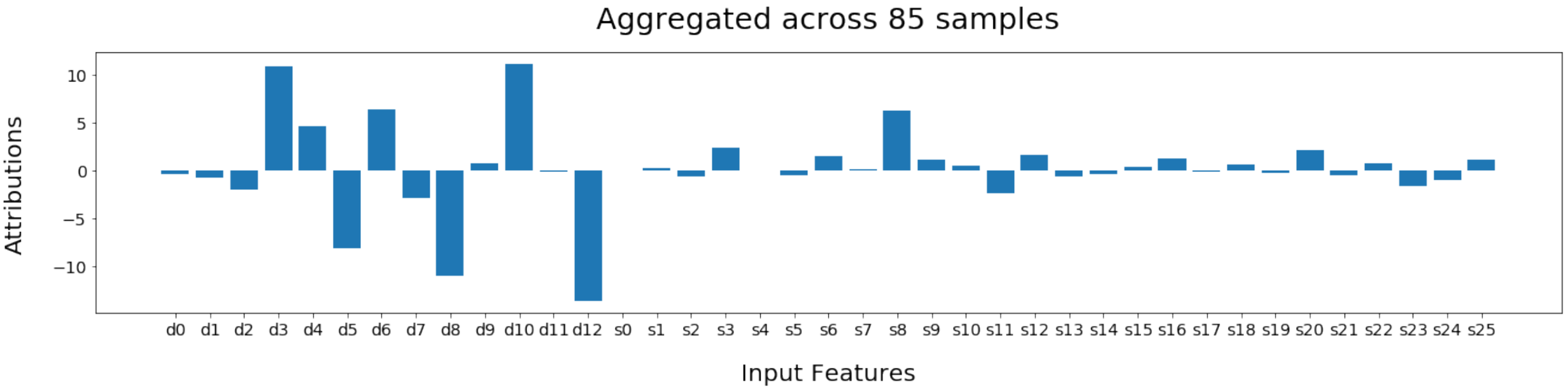
# SPARSE FEATURE IMPORTANCES FOR 85 SAMPLES WITH PREDICTION SCORE > 0.999



# FEATURE IMPORTANCES

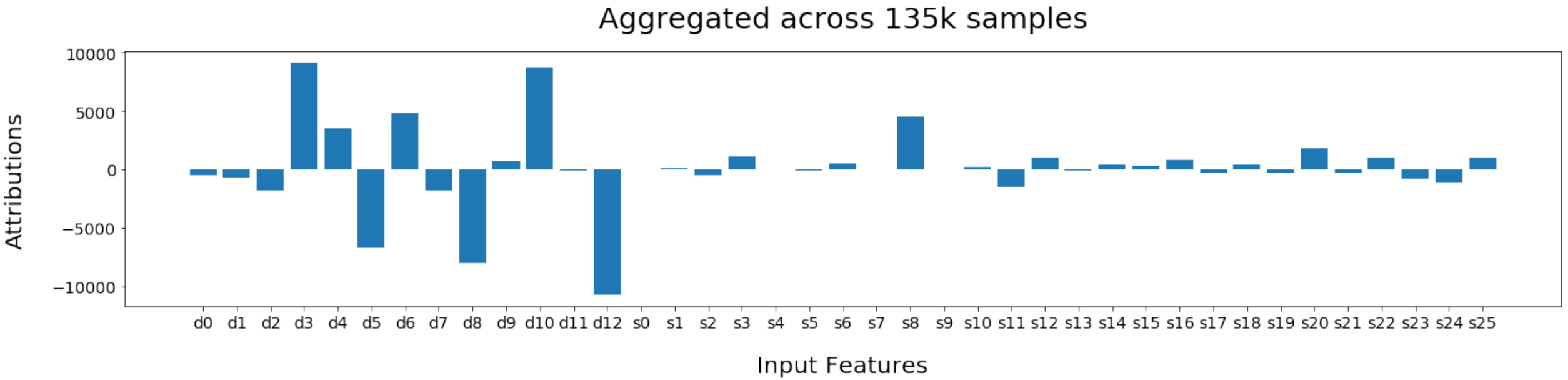


# FEATURE IMPORTANCES



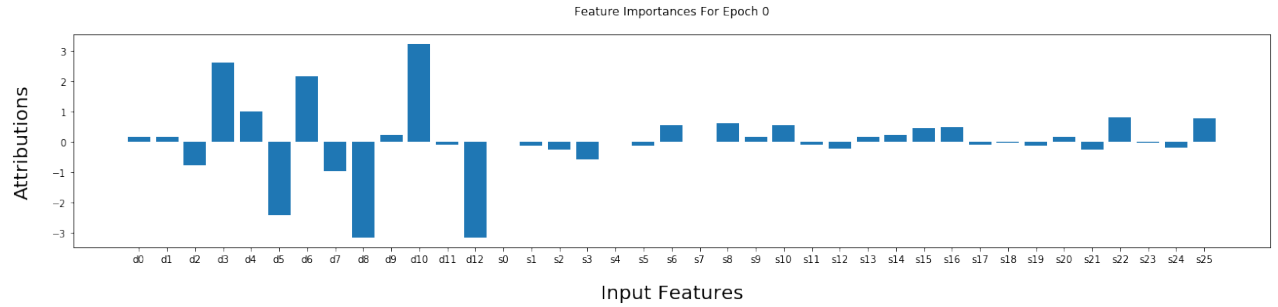

* ONLY 13% OF ALL SAMPLES ARE CLICKS

# FEATURE IMPORTANCES



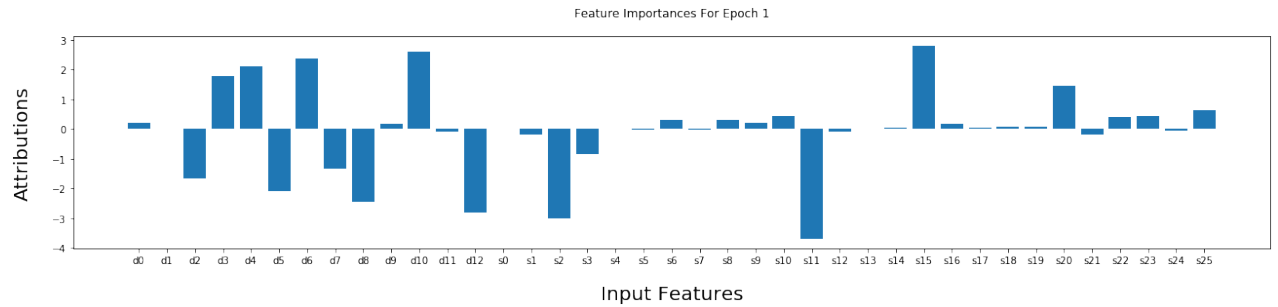

* ONLY 13% OF ALL SAMPLES ARE CLICKS

# COMPARING FEATURE IMPORTANCES ACROSS DIFFERENT EPOCHS



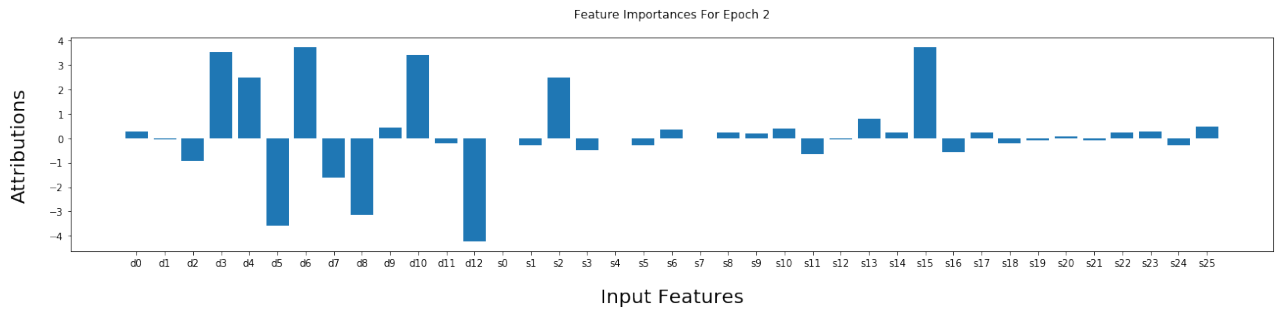

## Epoch 0

Accuracy: 78.80%
ROC AUC: 0.8
F1: 0.51




## Epoch 1

Accuracy: 78.34%
ROC AUC: 0.79
F1: 0.47

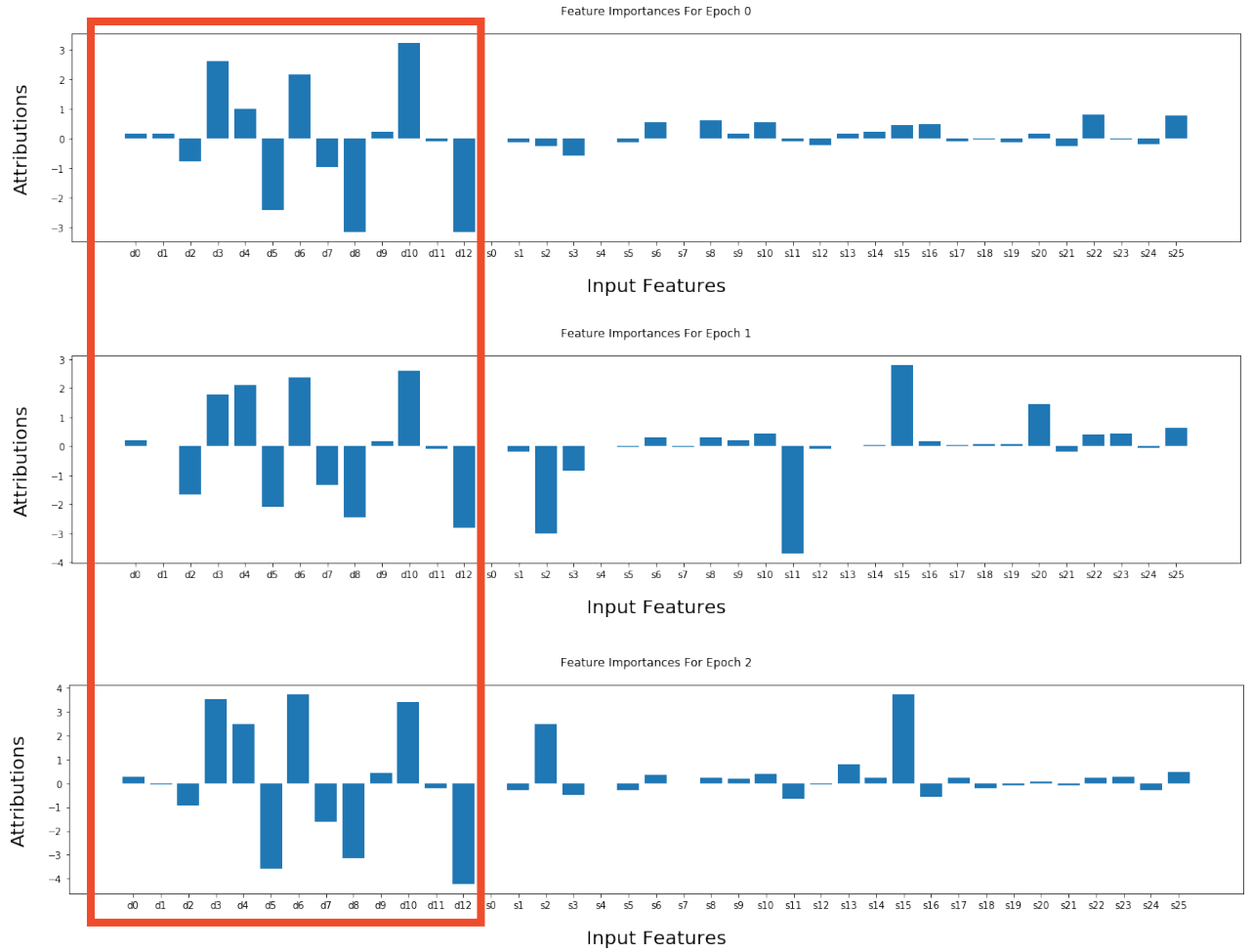



## Epoch 2

Accuracy: 78.52%
ROC AUC: 0.75
F1: 0.50

# COMPARING FEATURE IMPORTANCES ACROSS DIFFERENT EPOCHS

Dense




## Epoch 0

Accuracy: 78.80%
ROC AUC: 0.8
F1: 0.51

## Epoch 1

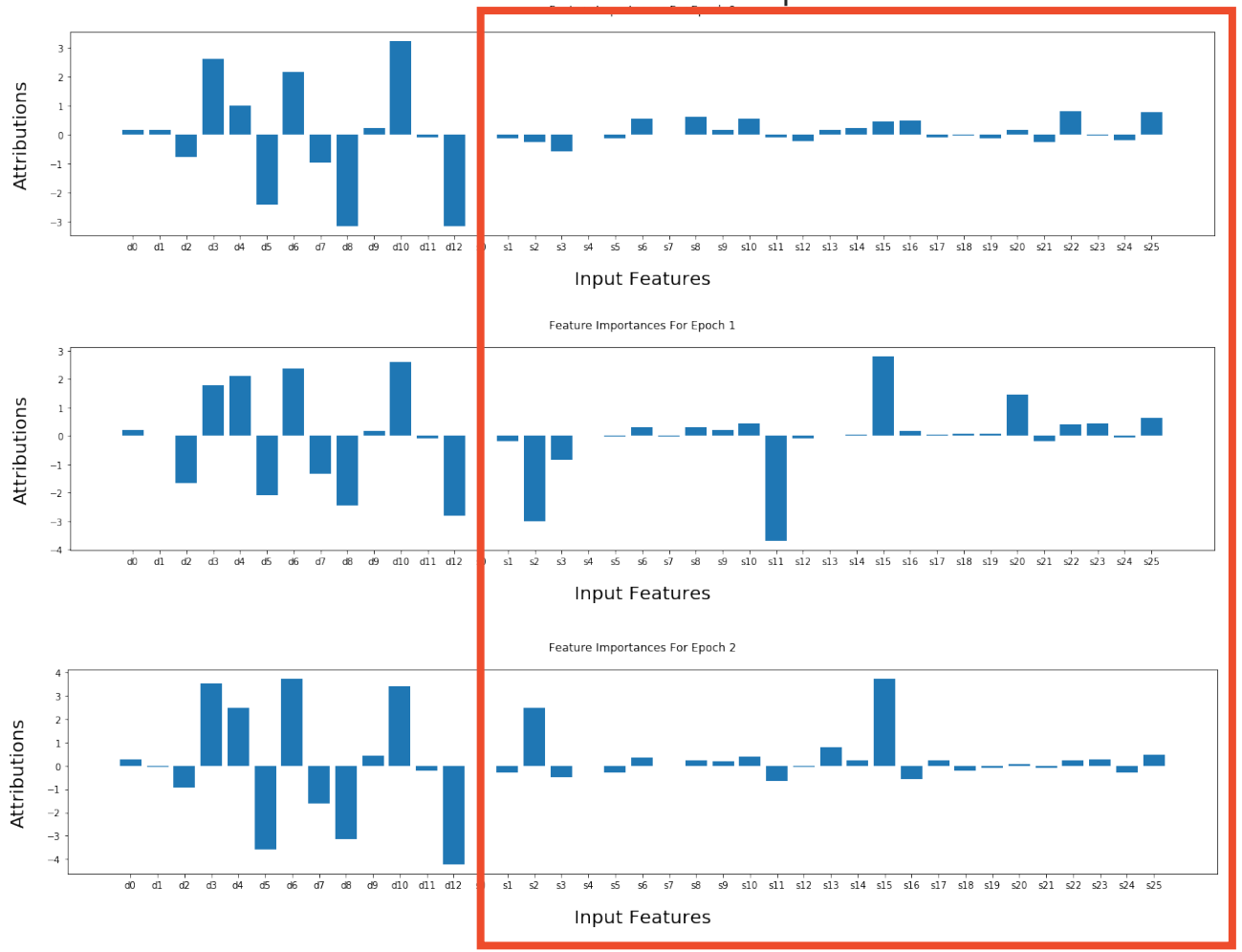
Accuracy: 78.34%
ROC AUC: 0.79
F1: 0.47

## Epoch 2

Accuracy: 78.52%
ROC AUC: 0.75
F1: 0.50

# COMPARING FEATURE IMPORTANCES ACROSS DIFFERENT EPOCHS

Sparse




## Epoch 0

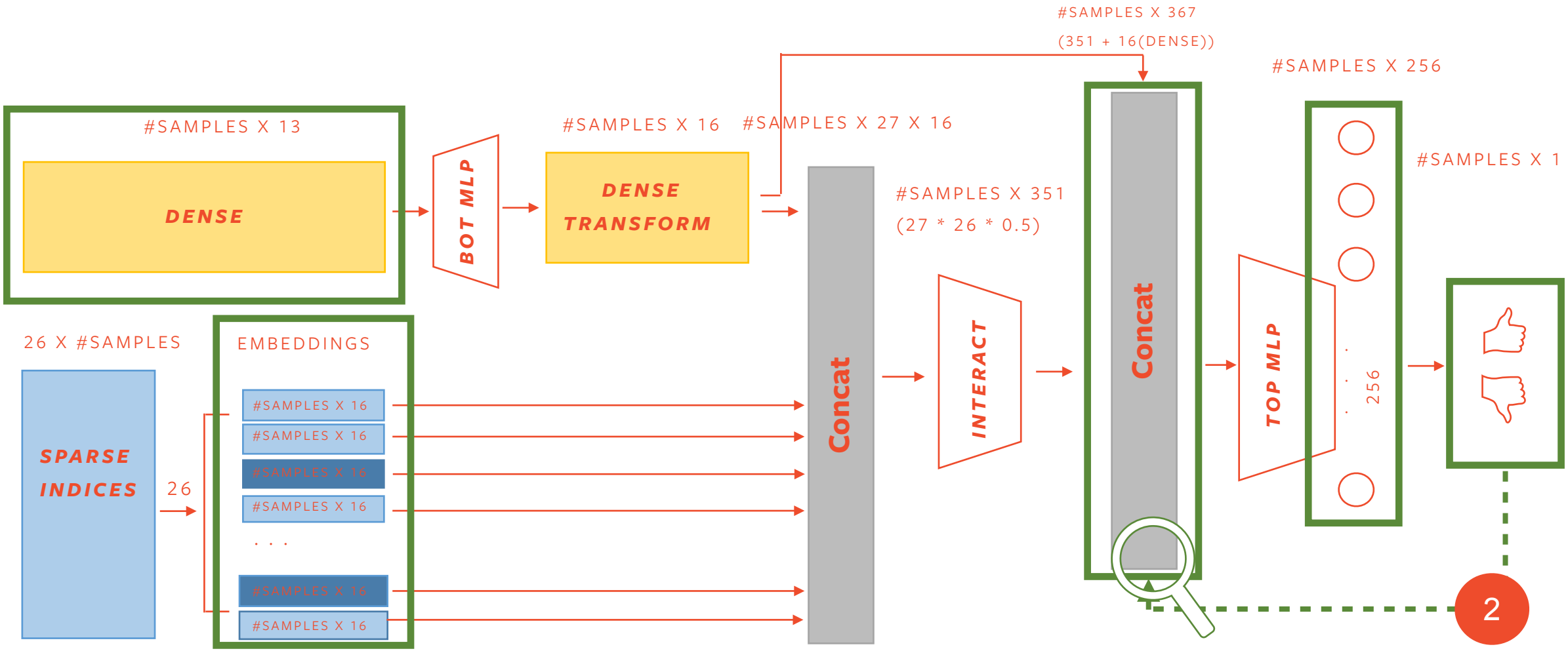
Accuracy: 78.80%
ROC AUC: 0.8
F1: 0.51

## Epoch 1

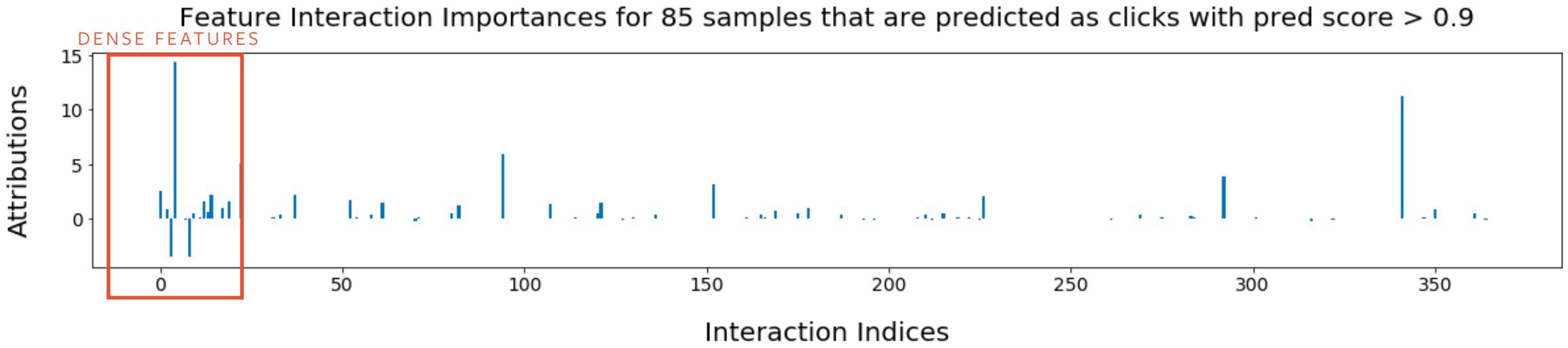
Accuracy: 78.34%
ROC AUC: 0.79
F1: 0.47

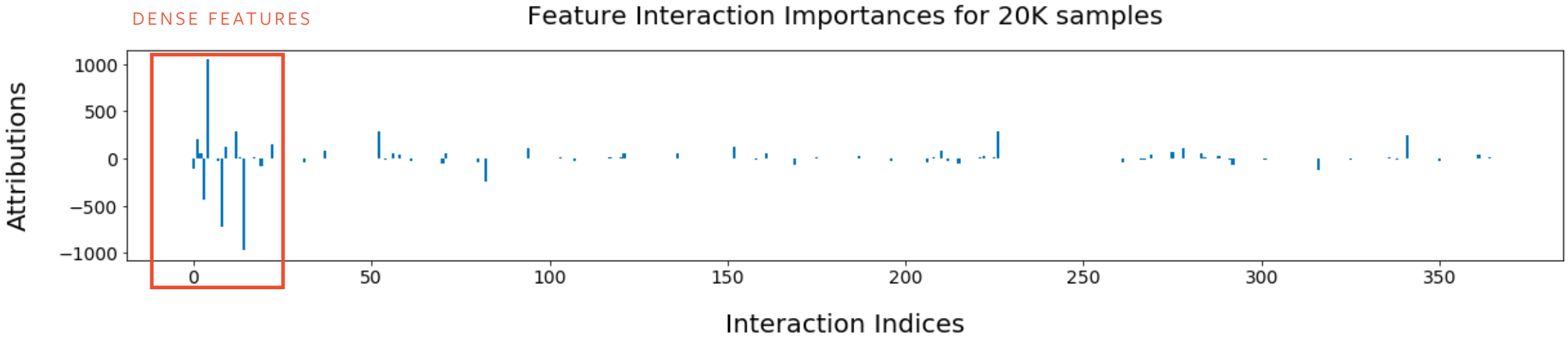
## Epoch 2

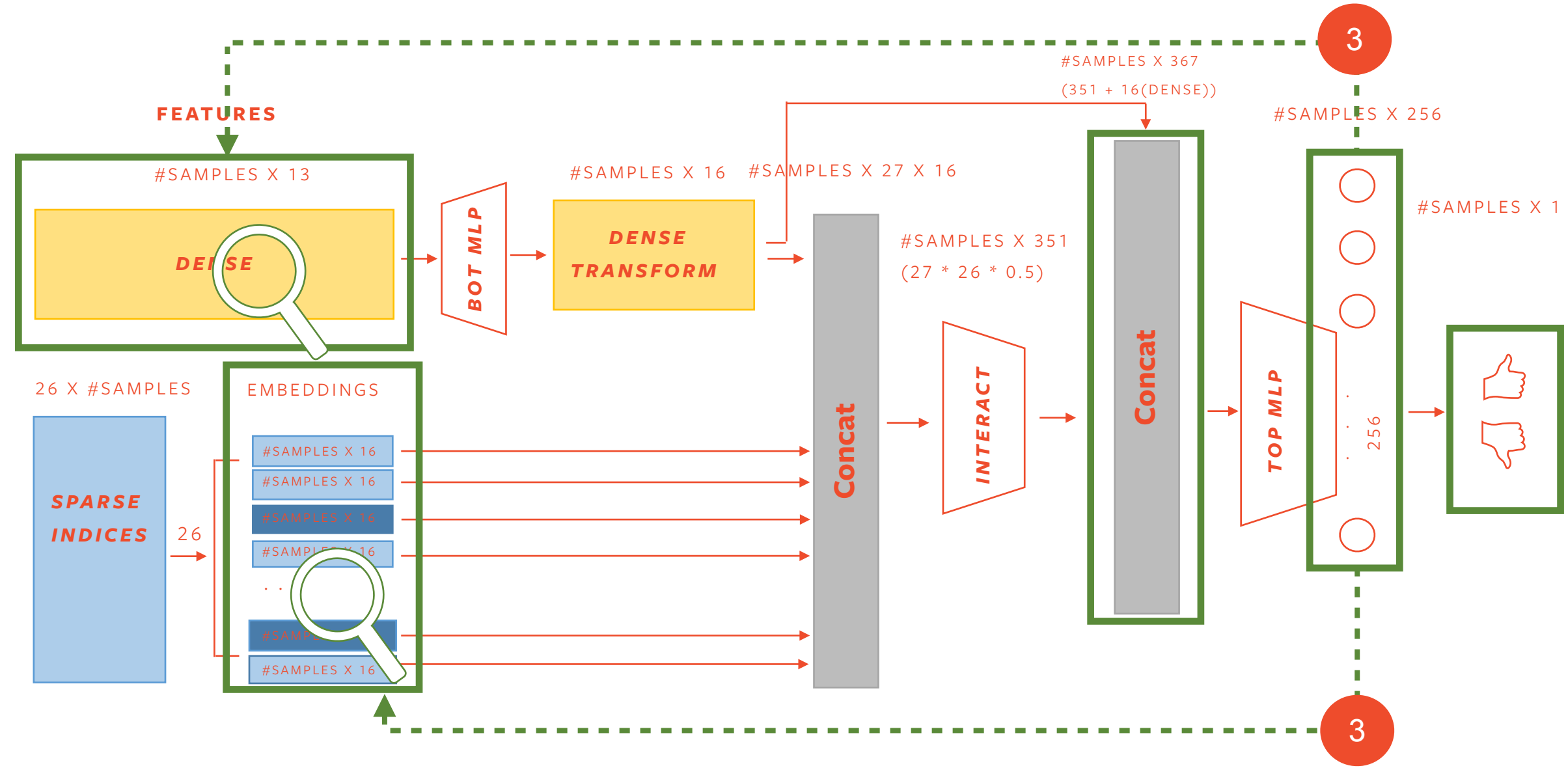
Accuracy: 78.52%
ROC AUC: 0.79
F1: 0.50


#SAMPLES X 13
DENSE
BOT MLP
#SAMPLES X 16
DENSE TRANSFORM
#SAMPLES X 27 X 16
#SAMPLES X 367
(351 + 16(DENSE))
#SAMPLES X 256
#SAMPLES X 1
26 X #SAMPLES
SPARSE INDICES
26
EMBEDDINGS
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
Concat
#SAMPLES X 351
(27 * 26 * 0.5)
INTERACT
Concat
TOP MLP
256
2

# FEATURE IMPORTANCES FOR 86 SAMPLES WITH PREDICTION SCORE > 0.9

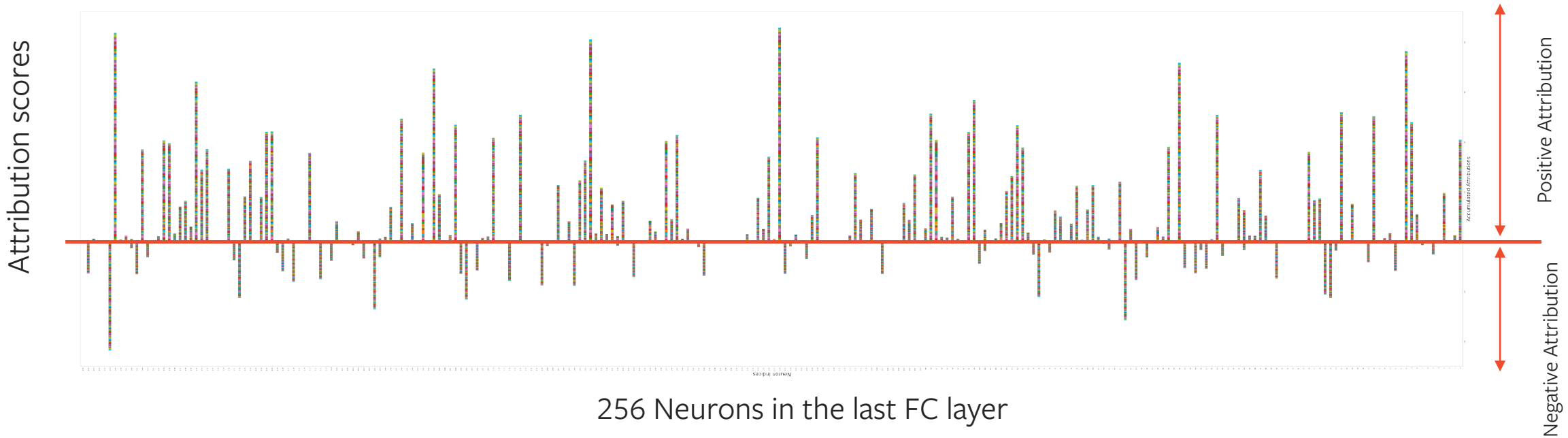



3
FEATURES
#SAMPLES X 13
DENSE
BOT MLP
#SAMPLES X 16
DENSE TRANSFORM
#SAMPLES X 27 X 16
#SAMPLES X 367
(351 + 16(DENSE))
#SAMPLES X 256
#SAMPLES X 1
26 X #SAMPLES
EMBEDDINGS
SPARSE INDICES
26
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
#SAMPLES X 16
Concat
#SAMPLES X 351
(27 * 26 * 0.5)
INTERACT
Concat
TOP MLP
256
3

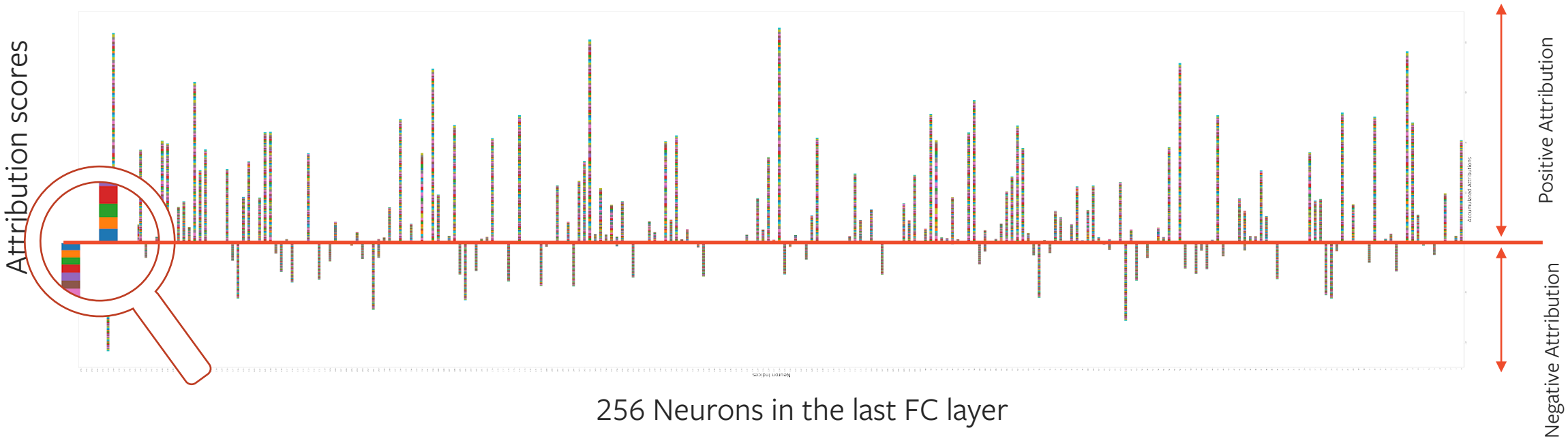
# NEURON IMPORTANCE

NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9



# NEURON IMPORTANCE

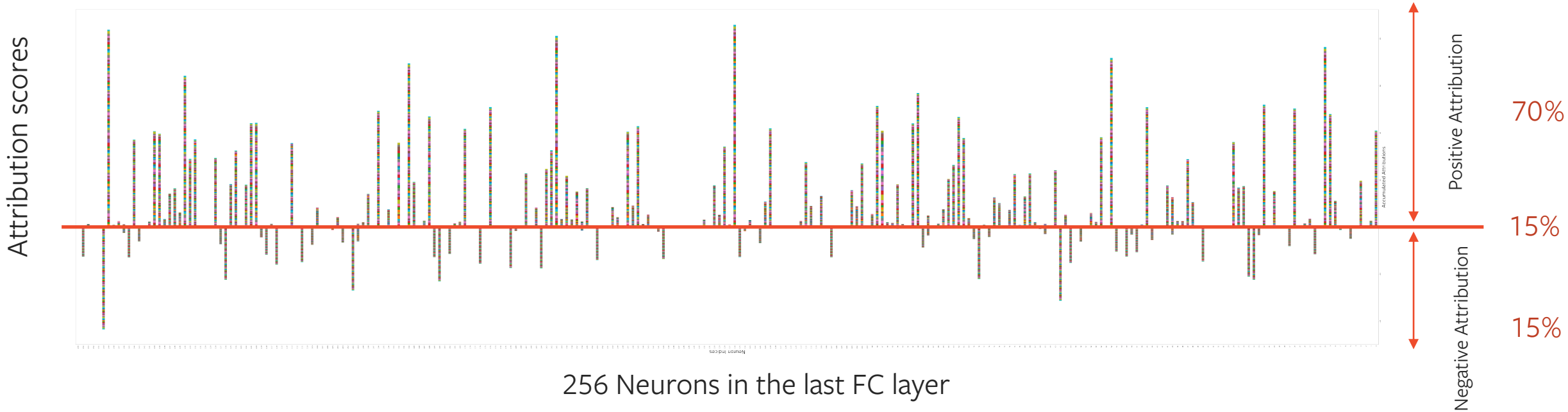
NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9




* Each input example has a different color coding

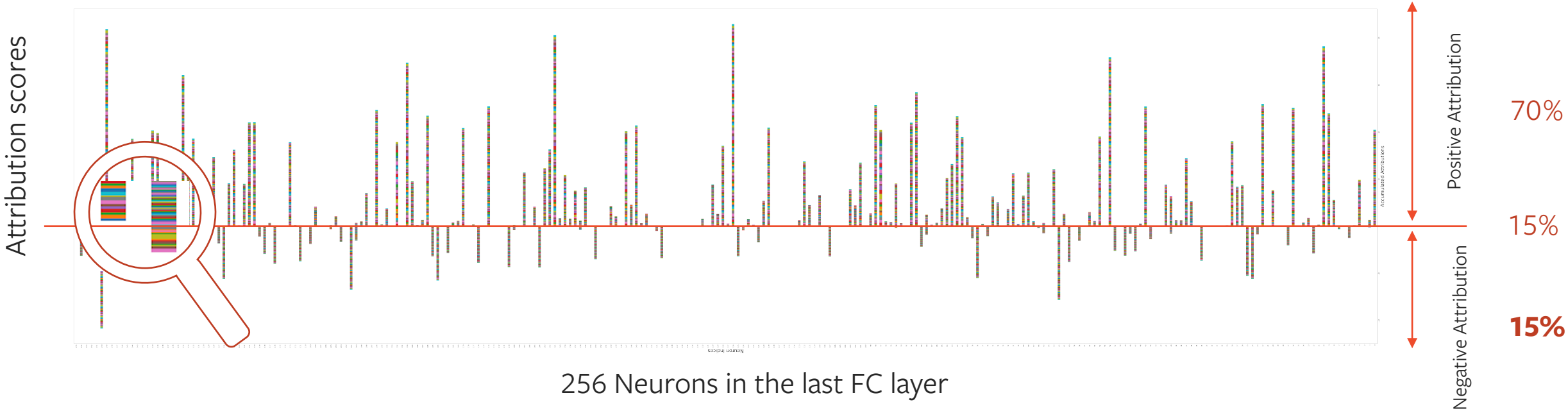
# NEURON IMPORTANCE

NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9



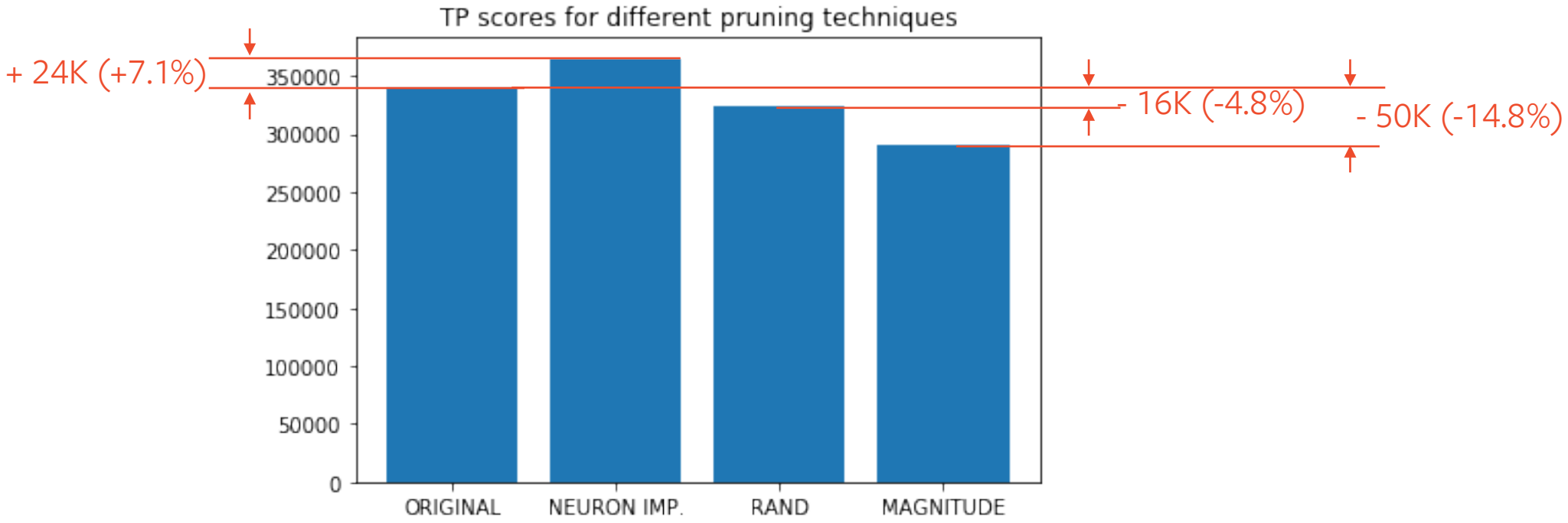
# NEURON IMPORTANCE

NEURON IMPORTANCE AGGREGATED ACROSS 100 SAMPLES WITH PREDICTION SCORE > 0.9



# MODEL PRUNING

PRUNED ~15% OF ALL 256 NEURONS IN THE LAST FC LAYER



# MODEL PRUNING

PRUNED ~15% OF ALL 256 NEURONS IN THE LAST FC LAYER

| Performance Measures | Original model | Pruned with Neuron Importance | Randomly Pruned | Pruned with Weight Magnitude |
|---|---|---|---|---|
| **Precision** | 66% | 64% | 67% | 69% |
| **Recall** | 40% | 43% | 38% | 34% |
| **F1** | 50% | 51% | 48% | 45% |
| **ROC AUC** | 80% | 80% | 80% | 80% |
| **Accuracy** | 79% | 79% | 79% | 79% |

# CASE STUDY SUMMARY

+ Sparse features primarily contribute to clicks predictions

+ Sparse Feature importance patterns significantly fluctuate across epochs

+ Feature interactions results to features that primarily contribute to clicks or have no affects on the prediction

+ Neuron importance based pruning can help us to increase TP, F1 and Recall scores and reduce FN

+ Magnitude based pruning can help us to reduce FP score and increase Precision.

## THE LIMITATIONS OF ATTRIBUTIONS

+ Attributions do not capture feature correlations and interactions

+ Finding good baselines is challenging

+ They are difficult to evaluate

+ Attributions do not explain the model globally

## FUTURE DIRECTIONS

+ **captum.robust**
  + adversarial  robustness and attacks
  + studying the connections between model robustness and interpretability

+ **captum.metrics**
  + model interpretability, sensitivity, trust, infidelity and robustness related metrics

+ **captum.concept**
  + Testing with Concept Activation Vectors (TCAV)

+ **captum.optim**
  + optimization-based visualizations

...