A study of Explainable AI

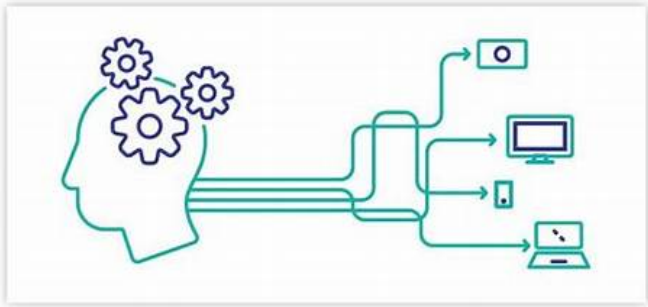
The concept of Zero Information

Thodoris Lymperopoulos

National Technical University of Athens

October 13, 2023

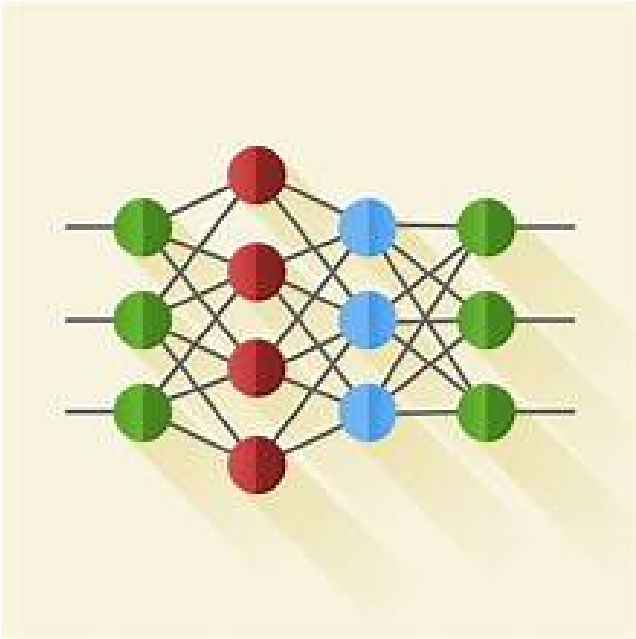
Explainable AI



What is Explainable AI (XAI) and problems does it try to solve?

It is a field of AI that tries to understand the model's **inner functioning**.

The need for XAI (1/3)



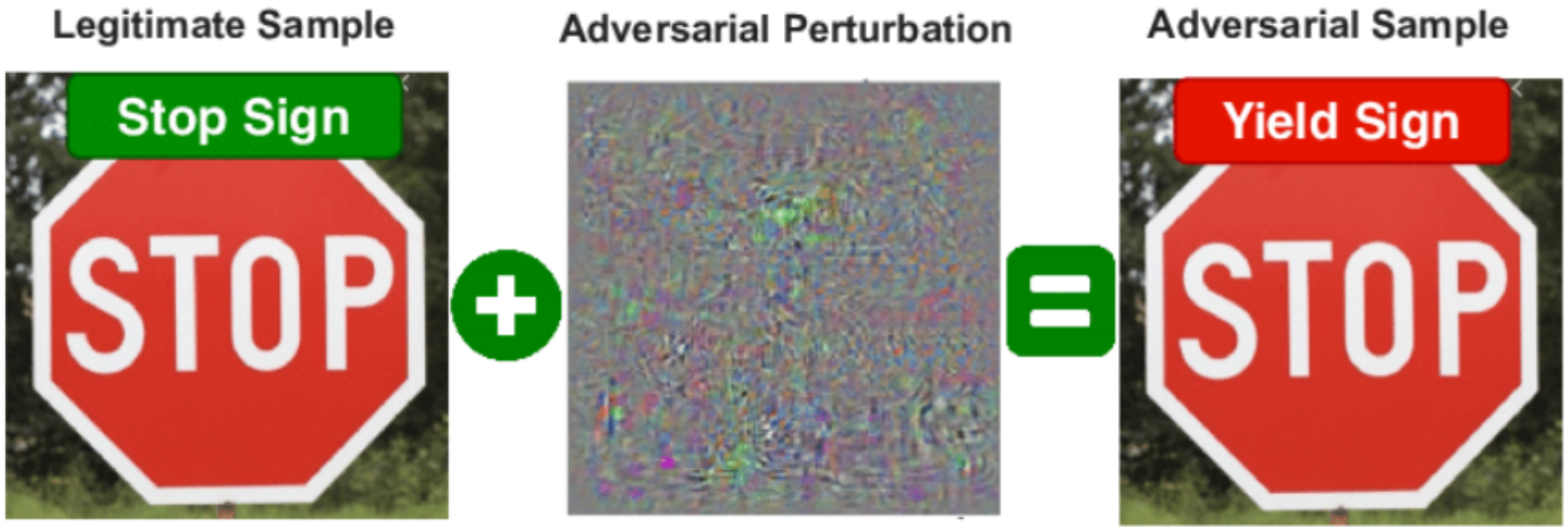
Why do we need to explain a model in the first place?

DNNs are complex models, with nested non-linear interactions.

The need for XAI (2/3)

We need to make sure that the model learns the **correct** causes of an effect.

The need for XAI (3/3)



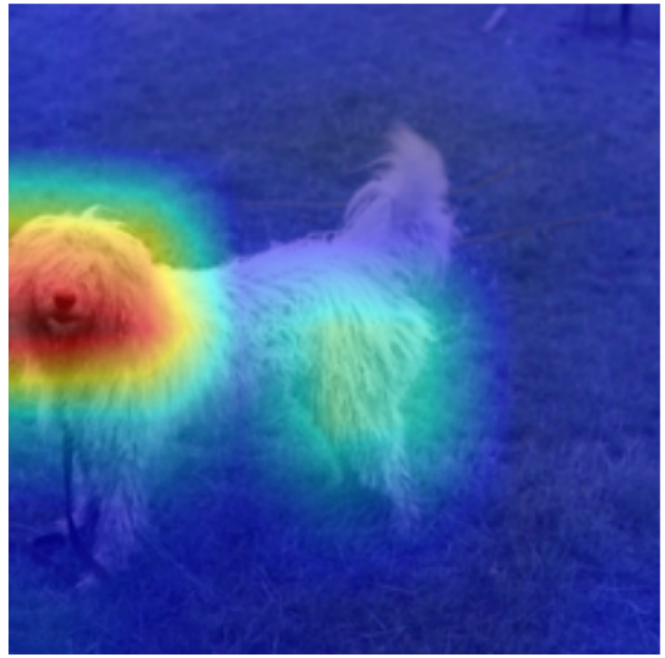
We need to understand why a model works in such ways and how to protect it from adversarial attacks.

XAI methodology

What methods are developed for understanding the functionality of a model?

- Many difficult questions in XAI remain unanswered.
- Focus has been shifted towards answering **what** caused an effect, neglecting **how**.
- In Computer Vision, it is easy to understand cause-and-effect relations.

Attribution methods (1/3)



- Attribution methods answer the *what* question, constructing **attribution maps**.
- Such maps give scores to the features.

Attribution methods (2/3)

How do they do this

Different techniques were developed.

- Some monitor the model's calculations.
- Others leverage mathematical tools related to importance.

Such tools are inspired from **Game Theory, Linear Algebra, Calculus** etc.

Attribution methods (3/3)

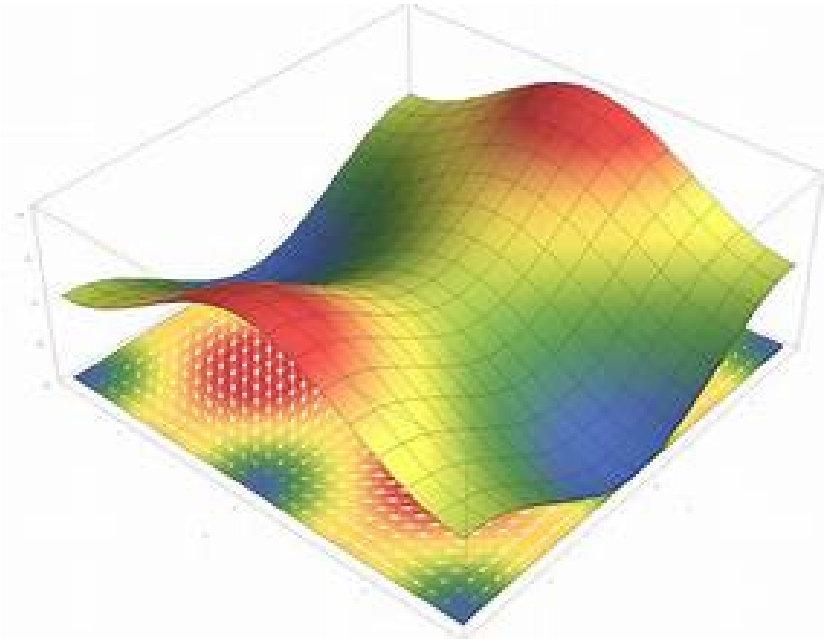
Expressed mathematically

Let $f \in \mathbb{F}$ be a DNN, $f : \mathbb{R}^m \to \mathbb{R}^n$, where $m, n \in \mathbb{N}$. m: size of the input space and n: number of classes. An attribution is a function

$$R : \mathbb{R}^m \to \mathbb{R}^m.$$

It assigns a value of importance of each feature of an input $\mathbf{x} \in \mathbb{R}^m$ with respect to f.

Gradient methods



- Gradients point to the direction a function changes more rapidly.
- They are related to sensitivity.
- The first attribution method (2013) was based on gradients.

Integrated Gradients (1/2)

One of the most robust methods designed until today.

$$R(x)_i = (x_i - x_{0_i}) \int_{\alpha=0}^{1} \frac{\partial f(x_0 + \alpha(x - x_0))}{\partial x_i} d\alpha$$

x_0 is a baseline point. In general:

- A cause leads to an effect, its absence impedes it.
- Gather information from absence to existence.
- Need to define *absence*.

Integrated Gradients (2/2)

Satisfaction of criteria

Because of its design, due to the **Fundamental Theorem of Calculus** it holds that

$$\sum_i R(x)_i = f(x) - f(x_0)$$

The model's decision gets redistributed back to the features.

CAM methods

Designed particularly for CNNs. Make use of saliency maps. Calculate a linear combination of them:

$$R_c(\mathbf{x}) = \sum_k w_k^c A^k(x),$$

where $A^k(x)$ is the activation of the $k-$th feature map of the last layer, for input x and class c.

Different methods define their sets of values for w_k^c.

- **GradCAM** uses the gradients of each $A^k(x)$ and sums them.
- **GradCAM++** also uses second order gradients.

XGradCAM (1/2)

Designs a set of desired criteria, then finds the values that satisfy them.

1. Sensitivity

$$f_c(x) - f_c(x \setminus \{k\}) = \sum_{i,j} w_k^c A_{ij}^k(x),$$

$f(x \setminus \{k\})$ is the prediction for class c, when the k-th feature map in the target layer is zeroed.

Intuition. *The drop in the model's score when not considering a feature map, should match its activation.*

XGradCAM (2/2)

2. Conservation

$$f_c(\mathbf{x}) = \sum_{k=1}^{K} \left(\sum_{i,j} w_k^c A_{ij}^k(x) \right)$$

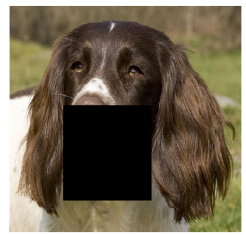
Intuition. *The explanation should be fully explained by the activation of the feature maps.*

They then solve the equations for w_k^c.

Occlusion methods (1/3)

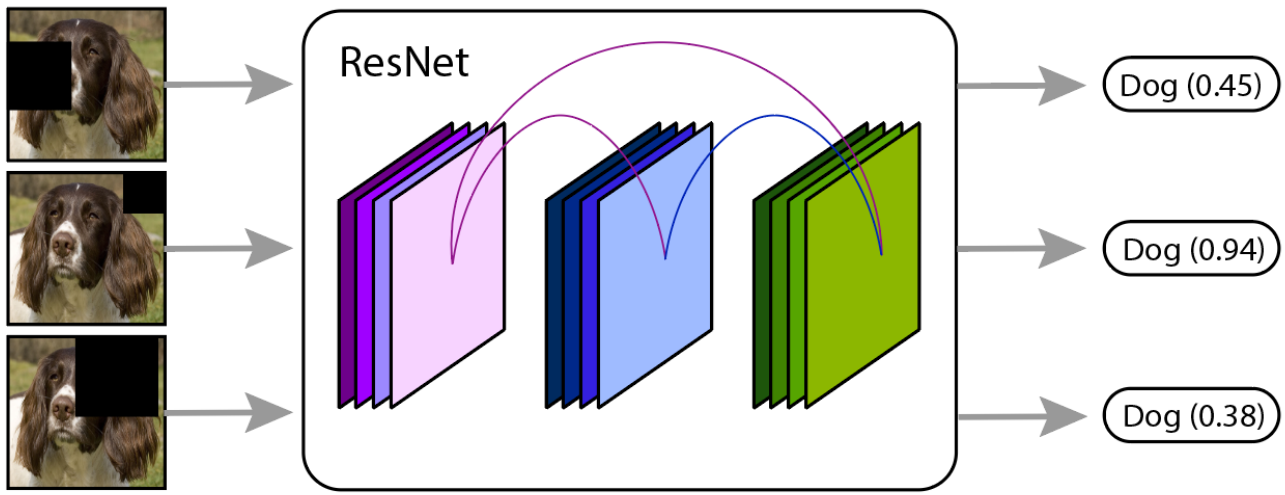
Step 1. Hide parts of the image.

Dog (0.95)



Occlusion methods (2/3)

Step 2. Pass masked images though model. Monitor drop in score.



Occlusion methods (3/3)

Step 3. Gather all information. Aggregate to calculate feature scores.

Intuition: Important parts should lead to larger drops in score.

RISE

RISE calculates the scores as:

$$R(x)_i = \mathbf{E}_M[f(x \odot M)|M(i) = 1],$$

where M is the **mask**, $\odot$ the dot product.

- Gathers all masked images where feature i appears, attributes the **mean value** of f in these images as score.
- Uses Monte Carlo approximation.

DeepSHAP

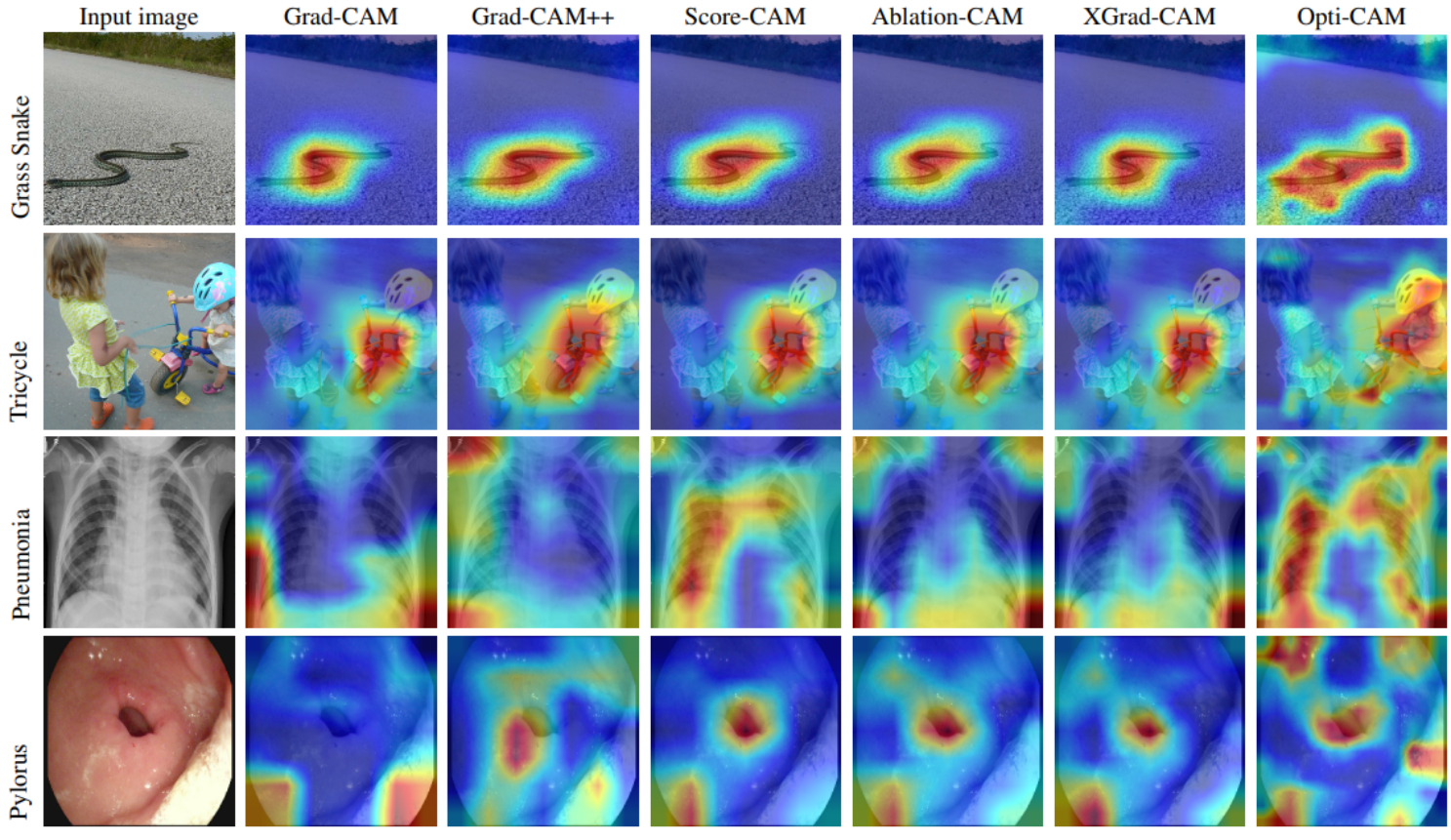
DeepSHAP calculates the scores as:

$$R(x)_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S!|(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)].,$$

where F is the set of features, S a subset of it, x_S the input with features $F \setminus S$ hidden.

- Gathers all subsets F where i is missing and adds it, monitoring the increase in score. Returns a **weighted** mean score.
- Works with hyperpixels.

The need for metrics



Which method is better?

Evaluation metrics (1/3)

What are they?

Metrics to **measure the effectiveness** of an attribution method.

- Much like Accuracy and F1-Score evaluate the performance of classifiers.

But how do we design such metrics?

They must be somehow correlated with the notion of importance.

Evaluation metrics (2/3)

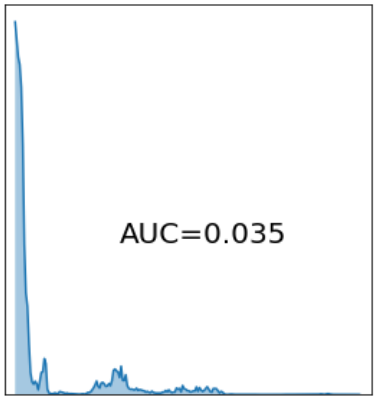
Two of the most common. They are based on **Occlusion**.

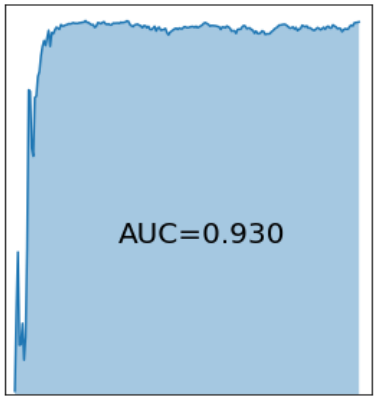
1. Average Drop. Get the attribution map, use it as a mask to hide the image, calculate drop in prediction.

$$AD = \sum_{i=1}^{N} \frac{\max(0, f(x)_c - f(x')_c)}{f(x)_c}$$

Evaluation metrics (3/3)

2. Insertion/Deletion. Calculate attribution map and sort the features based on their score. Step-by-step add/remove features and measure f. Create a graph and calculate AUC.





Criteria

Other works developed axioms and criteria.

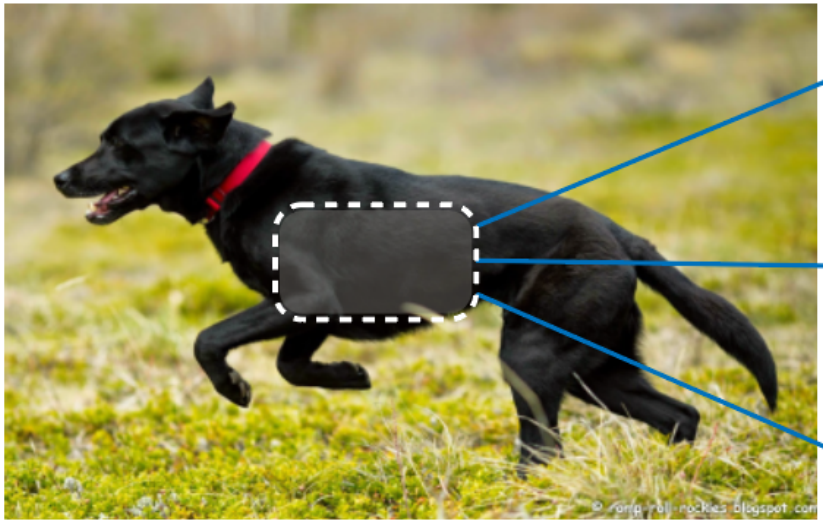
1. Sensitivity. For every input and baseline that differ in one feature but have different predictions, then the differing feature be given a non-zero attribution.
2. Conservation. The model's score should be distributed to the attribution map, and the opposite should also be true.
3. Implementation Invariance. If two models are equal, the attributions they produce should also be equal.

Zero Information

What Occlusion methods, metrics and criteria get wrong. Hiding information is a relative concept.

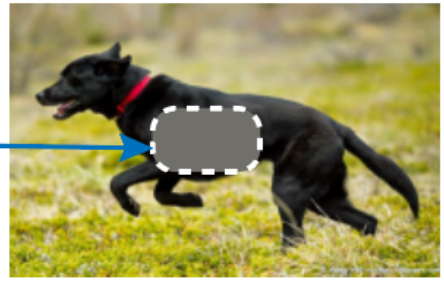
- First techniques zeroed the values for the features in order to hide them.
- What if those values where already close to zero?

Filling Techniques



Prediction: Dog (0.95)

Dog (0.94)



Dog (0.45)

Dog (0.05)

Problems

Heuristic fillings lack robustness. The following problems arise:

1. Added Bias
2. Out-of-Distribution data
3. Attribution Shift

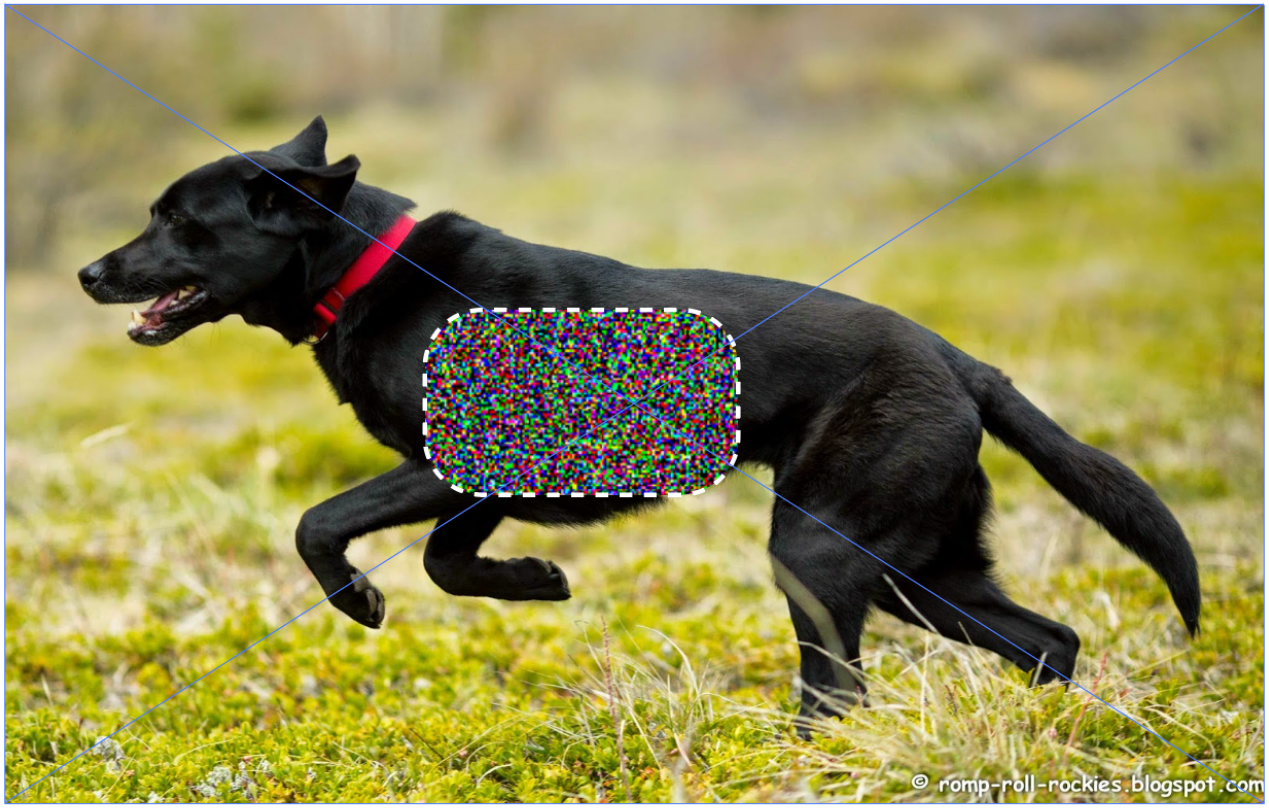
Added Bias

Score Difference = 0.01

Score Difference = 0.5

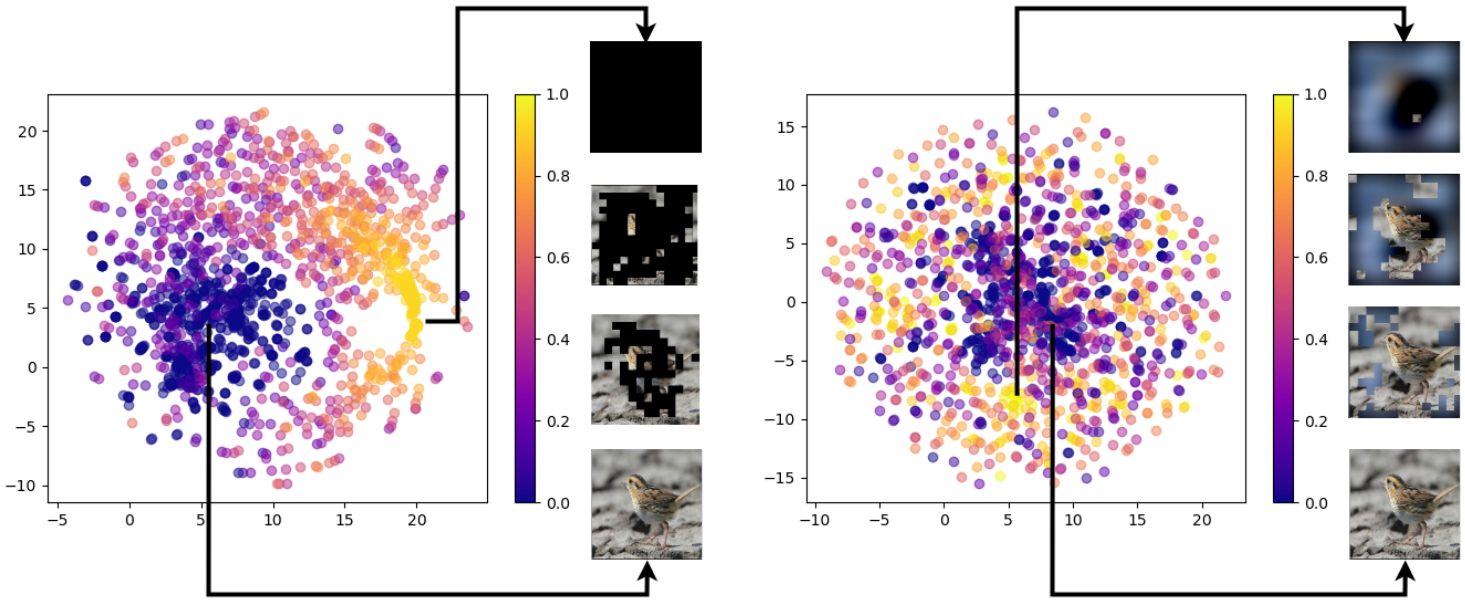
Feature values that are far way from the baseline value are favored.

OoD (1/2)



Such choices lead to data that are Out-of Distribution.

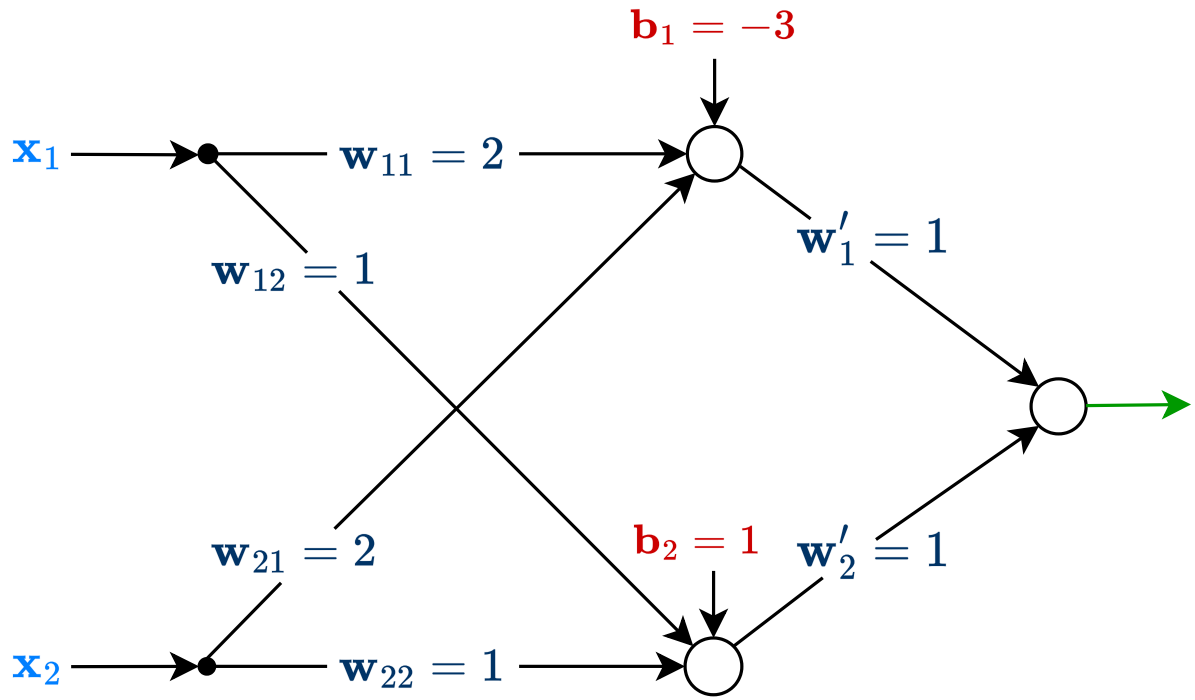
OoD (1/2)



(a) (b)

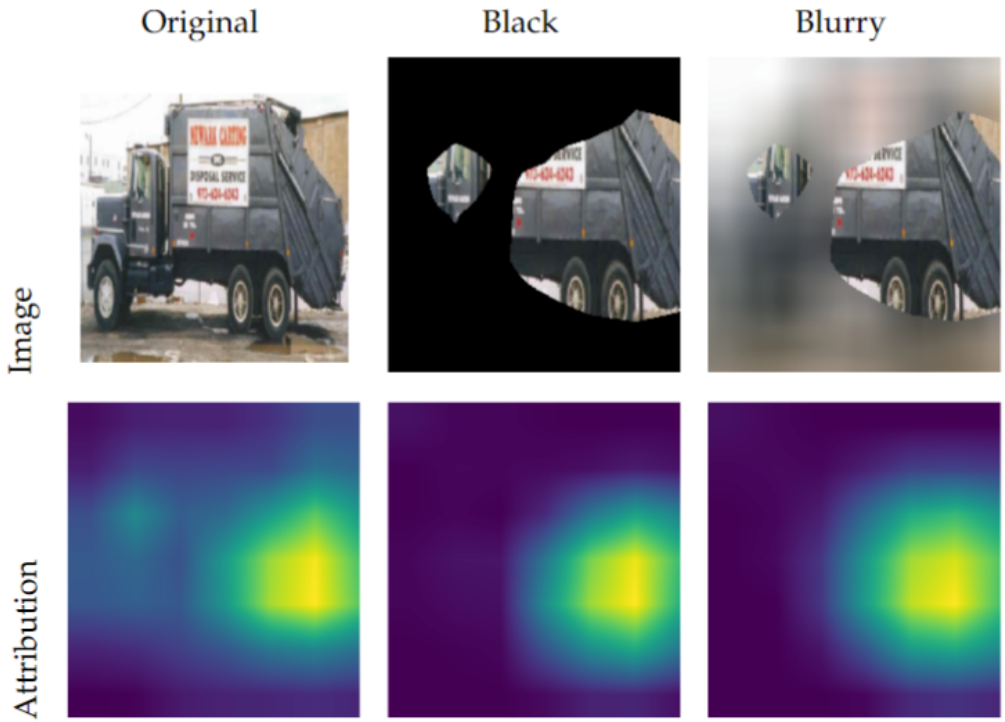
The criteria Insertion/Deletion result in OoD images, when they apply heuristic techniques.

Attribution Shift



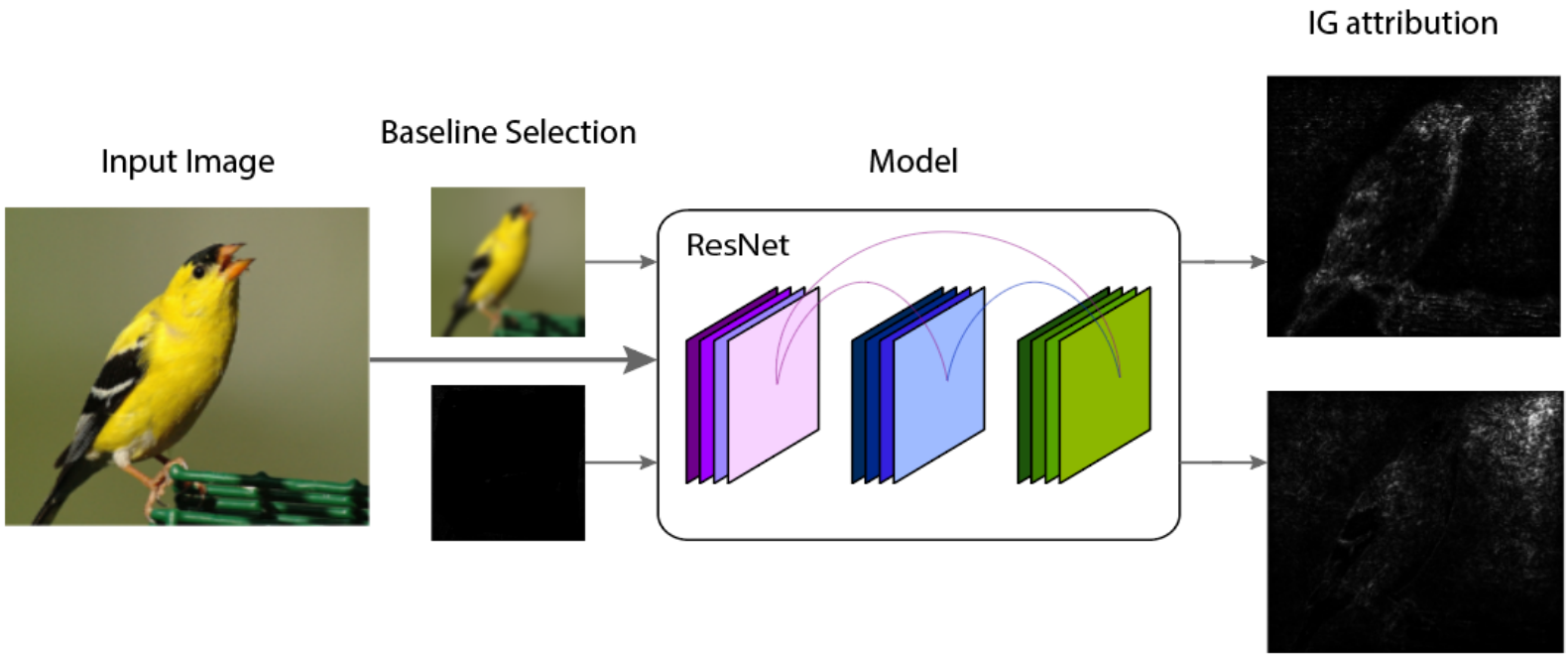
The attribution of the visible parts changes, due to the different interactions they perform with the new, "masked" values.

Attribution Shift in action



Black and blurry lead to attribution shift. The left part is completely neglected.

Failure of Heuristic Techniques



The three problems have a great influence in **Interated Gradients** when selecting a baseline.

What is Zero Information

Zeroing information:

- in parts of the image
- at the whole image

The second task is much easier. It gives higher level of freedom.

Zero Information Points

How is a Zero Information Point Defined?

Instead of defining it (eg. black), let the model find it: Guided process with respect to the **image and model parameters**.

We only need to define a criterion:
For a Zero Information Point (ZIPO), the output of the model has maximum entropy.

$$x_o = \underset{\mathbf{x}}{\text{argmax}}(H(f(\mathbf{x}))) \tag{1}$$

The model can make no judgement about any class: uniform activation.

ZIPO algorithm

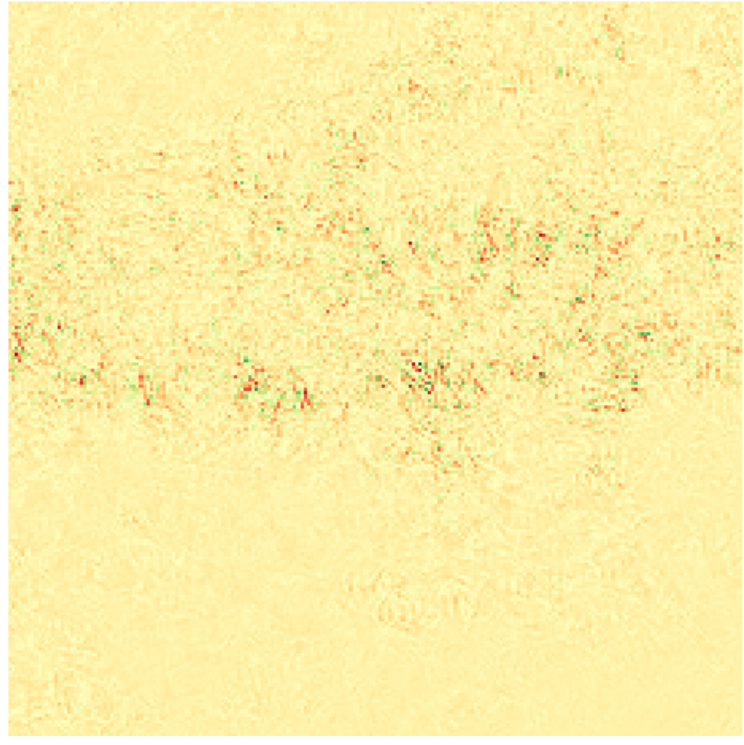
The algorithm works as:

1. Start with initial image x
2. For steps, calculate $H(f(x))$ and use it as loss.
3. Update x to augment H.

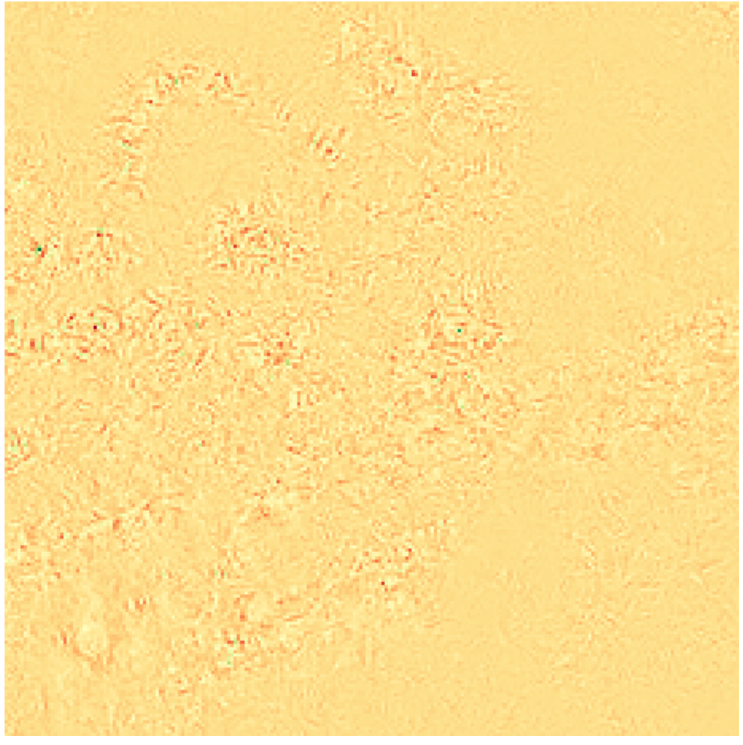
Each time, it tells the image to **increase the confusion of model**. The image hides its important parts.

By gathering gradients we define an attribution method. It is **Guided Integrated Gradients**. No need for baseline.

Application (1/2)

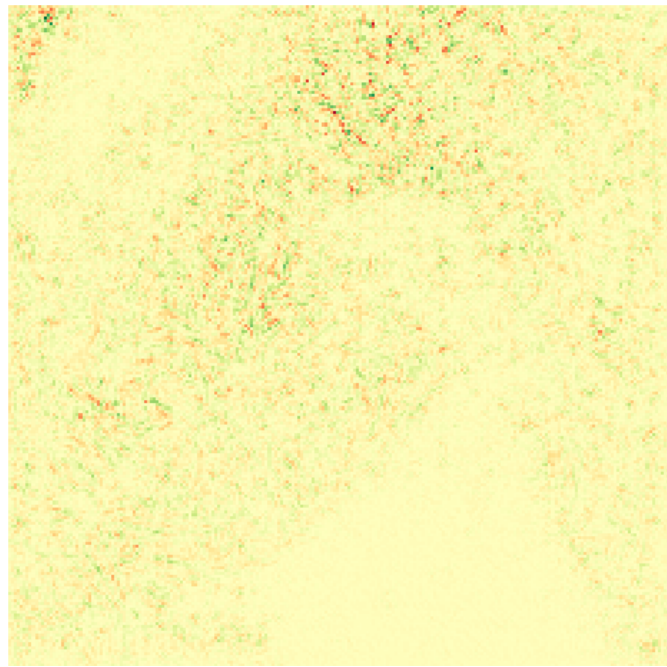


Application (2/2)



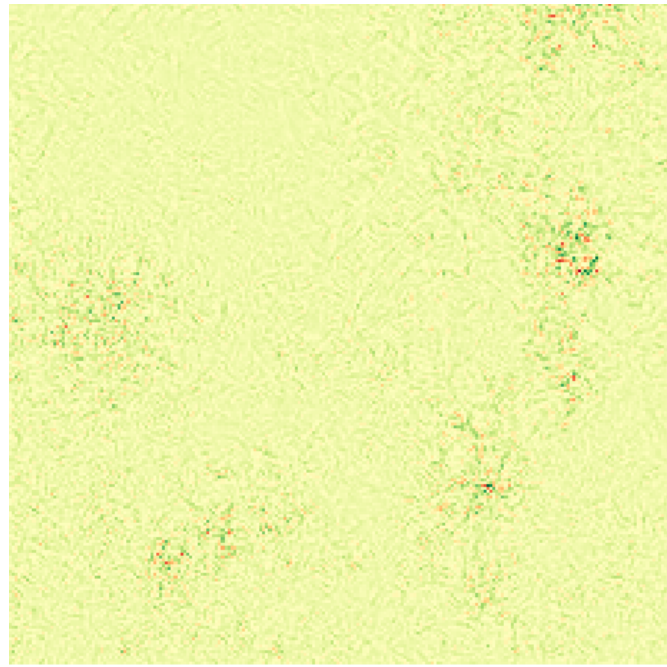
Weak points (1/2)

Might attack the mask.



Weak points (2/2)

Hides all information.



Zero Information Parts

How do we effectively hide parts of image?

This is harder, since the visible parts should contribute **the same way as before** (avoid attribution shift).

Need for criteria (as in ZIPO) to guide our process.

Criteria (1/2)

Criterion 1 *The filling must not alter the contributions of the visible features.*

We define $x' = [x'_v; x'_h]$, where $x_v = \mathbf{0}$. We demand

$$f(x + x') = f(x) \tag{2}$$

Is that enough?

Criteria (1/2)

Many points could satisfy this criterion. We need the one with least information **in general**.

Criterion 2 *The filling should introduce no additional information to the model.*

Thus, We demand

$$f(x') = \mathbf{0} \tag{3}$$

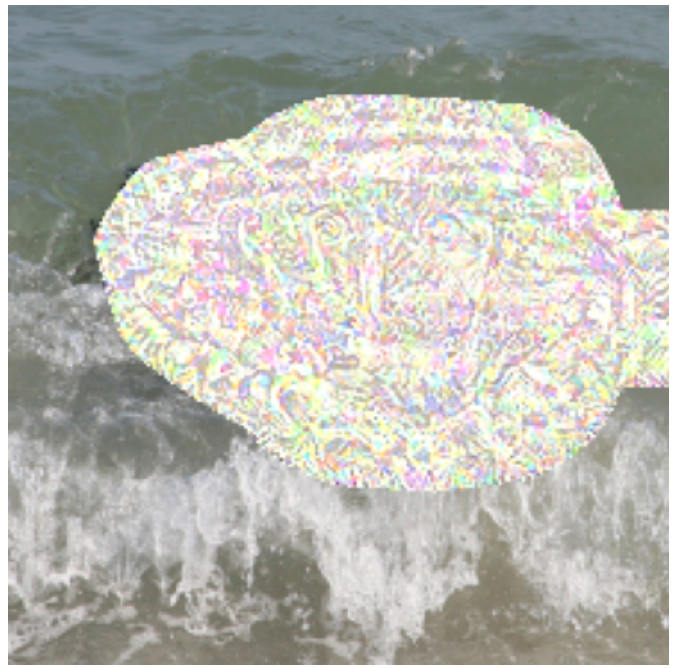
Then $x_0 = [x_v; x'_h]$

Algorithm

The algorithm is:

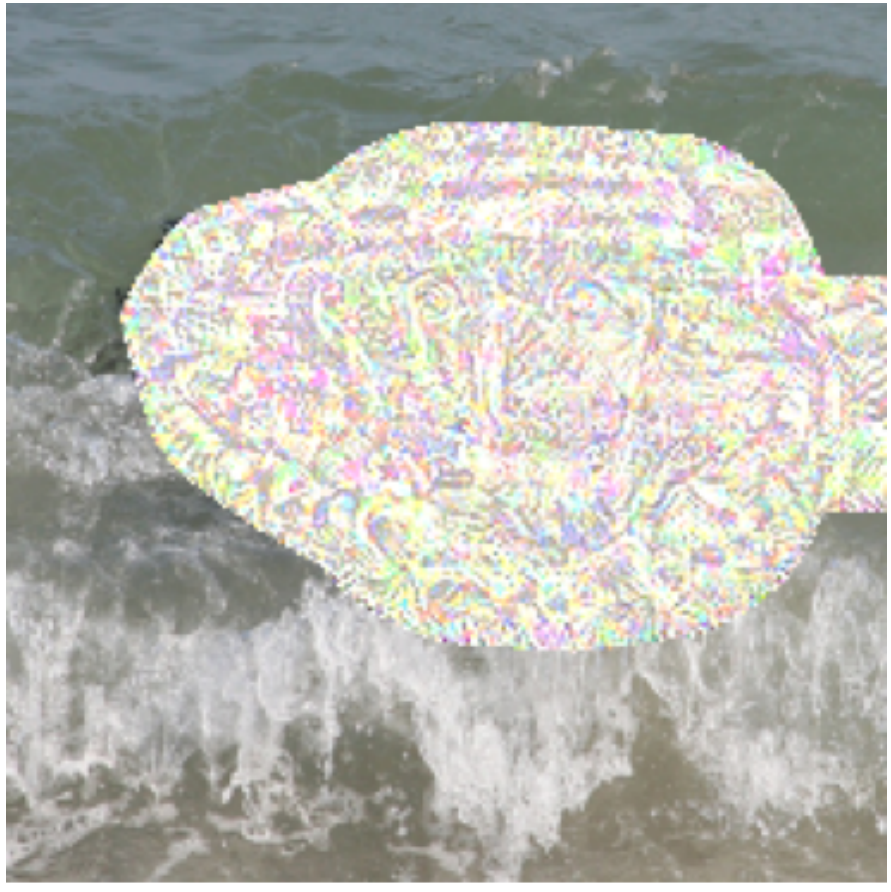
1. Start from random x' with zero in **visible**.
2. For steps, calculate entropy for C1 and C2 and add them as loss.
3. Update x', only the **hidden** part.
4. Return $[x_v; x'_h]$

Application



Weak points

The images are completely OoD.



MAE (1/2)

How do we solve this?

Need to innovate. Use a **Generative Model** in between.

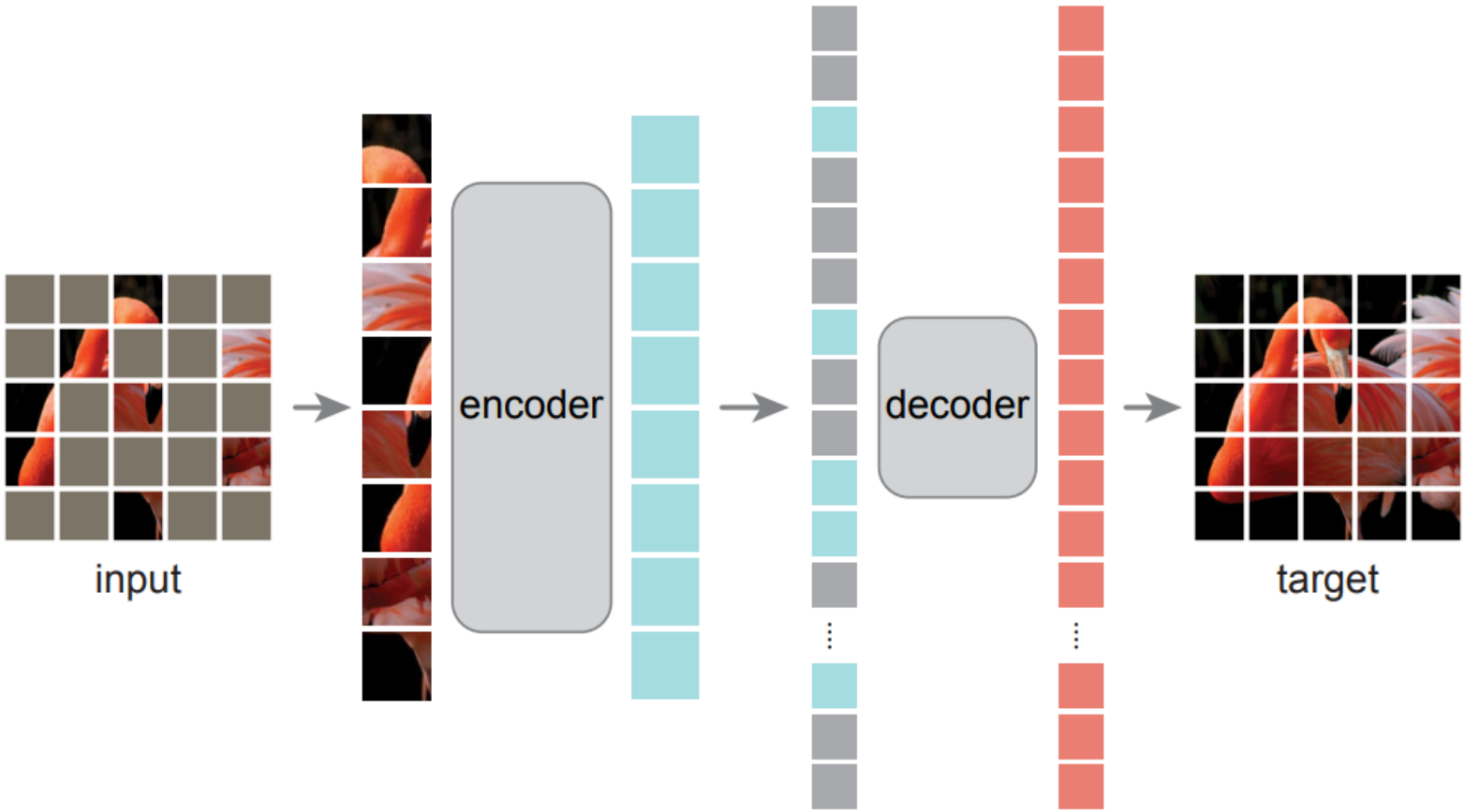
Masked Autoencoders: asymmetric encoder-decoder architecture that reconstructs hidden image parts.

- Encoder: constructs a hidden representation of the visible parts.
- Decoder learns to reconstruct the image based on those latent representations.

Main idea: images are natural signals with heavy spatial redundancy. Randomly hides 75% of them.

MAE (2/2)

The architecture of MAE.



MAE-ZIP

Need to edit MAE to hide defined regions: Masked-MAE. The algorithm is:

1. For image x, pass **visible** part through **Encoder**, to get x_{lat}.
2. Downsample mask and hide x_{lat} hidden part, to form $M(x_{\text{lat}})$.
3. For steps, add x^0_{lat} variables to augment $M(x_{\text{lat}})$ and form x_{rec}.
4. Pass through **Decoder**: $x_{\text{rec}}, x^0_{\text{rec}}$.

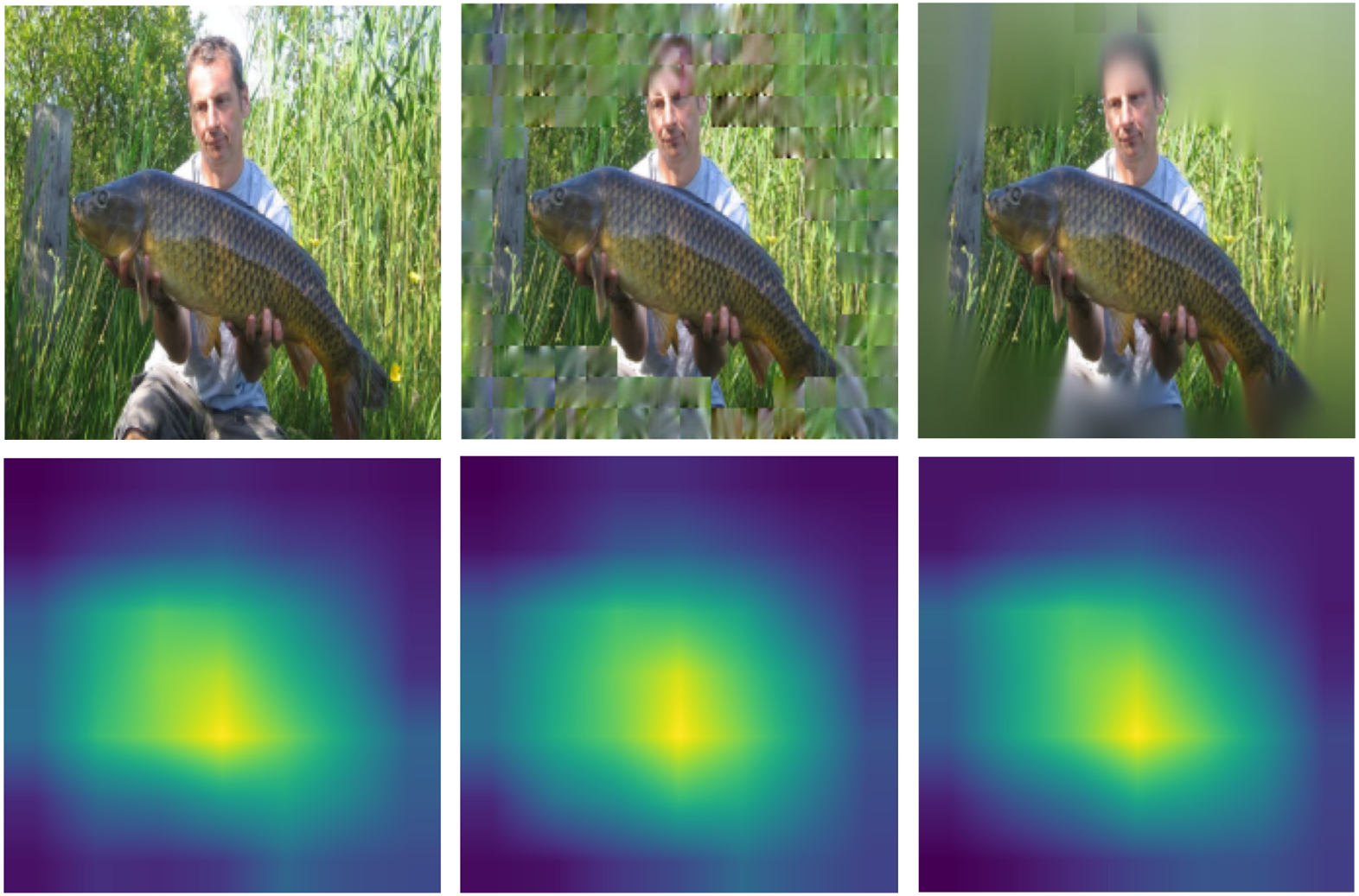
Losses

We define the losses:

1. Minimal Impact. $l_1 = \|x^0_{\text{lat}}\|^2$
2. Background neutrality. $l_2 = H(f(x^0_{\text{rec}}))$
3. Foreground stability. $l_3 = \|f^l(x') - M(f^l(x))\|^2$

Visual Example

Original Image | MAE-ZIP (standard) | MAE-ZIP ($l_3 = 1$)



Metrics

No metrics for this problem. We define:

1. Attribution mask. Measures how much the attribution has changed in visible, if zeroed in hidden:

$$\text{AM} = \|(M(R(x)) - R(x_{\text{rec}}))\|_2^2$$

2. Accuracy Preservation. Ensure in-Distribution data.

$$\text{AP} = \frac{\sum_{d \in D} \mathbb{1}[\hat{y}_x = \hat{y}_{x_{\text{rec}}}]}{|D|}$$

Results

Method	GradCAM		GradCAM++	
	AM↓	AP↑	AM↓	AP↑
Black	0.231	0.049	0.206	0.05
Blurry (20)	0.160	0.150	0.155	0.155
Blurry (75)	0.195	0.085	0.180	0.09
Random noise	0.162	0.001	0.091	0.01
MAE	**0.117**	0.714	**0.120**	0.717
MAE-ZIP	0.119	**0.755**	0.121	**0.760**

For $w_3 = 1 \times 10^{-4}$.

Results

Method	XGradCAM		LayerCAM	
	AM↓	AP↑	AM↓	AP↑
Black	0.230	0.047	-	0.057
Blurry (20)	0.160	0.149	-	0.157
Blurry (75)	0.195	0.083	-	0.090
Random noise	0.162	0.001	-	0.000
MAE	**0.116**	0.717	0.120	0.717
MAE-ZIP	0.121	**0.763**	**0.119**	**0.754**

For $w_3 = 1 \times 10^{-4}$.

Application

Three applications:

1. Can work as backbone of Attribution methods, based on Occlusion.
2. Can work as backbone of Evaluation metrics, based on Occlusion.
3. Can better approximate the exact contribution of an image part.

Conclusion

What we achieved:

1. The concept of zero information is linked to the model and image at hand.
2. It can be used for the design of an attribution method.
3. Criteria can guide us towards robust filling of image parts.

Future Work: More research to design criteria, more robust generative models can be used. Tesing of applications.

Thank you for your attention!