

# A Tutorial on Model Explainability

Sahra Ghalebikesabi

# Let us train a state of the art AI!

or



# Let us train a state of the art AI!



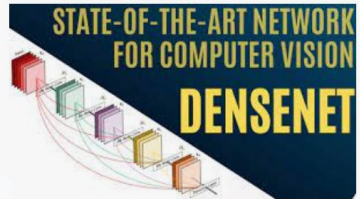
state of the art cnn



CNN Models	Param.	FLOPs	Top-1	Top-5
ResNet-152	57.40M	10.82G	77.58	93.66
SENet-152	63.68M	10.85G	78.43	94.27
ResNet-200	74.45M	14.10G	78.20	94.00
ResNeXt-101	46.66M	7.53G	78.80	94.40
DenseNet-264	28.78M	5.15G	77.85	93.78
ECA-Net50 (Ours)	24.37M	3.86G	77.48	93.68
ECA-Net101 (Ours)	42.49M	7.35G	78.65	94.34

SENet (Hu, Shen, and Sun 2018), CBAM (Woo et al. 2018) and AA-Net (Bello et al. 2019). From Table 2 we can see

Comparisons with other state-of-the-art CNN models on Image... researchgate.net



State of the Art Convolutional Neural Networks (CNNs) Expl... youtube.com

Year	Model	Architecture	Top-1 Accuracy
2012	AlexNet	Deep convolutional neural networks	~23.4%
2013	ZFNet	Deep convolutional neural networks	~37.4%
2014	VGG	Deep convolutional neural networks	~48.0%
2014	GoogLeNet (Inception v1)	Deep convolutional neural networks	~67.1%
2015	ResNet-50	Deep convolutional neural networks	~73.7%
2015	ResNet-101	Deep convolutional neural networks	~76.2%
2015	ResNet-152	Deep convolutional neural networks	~77.3%
2016	DenseNet	Deep convolutional neural networks	~77.8%
2016	SENet	Deep convolutional neural networks	~78.4%
2017	SENet-152	Deep convolutional neural networks	~78.4%
2017	SENet-101	Deep convolutional neural networks	~78.2%
2018	SENet-152	Deep convolutional neural networks	~78.4%
2018	SENet-101	Deep convolutional neural networks	~78.2%
2018	SENet-50	Deep convolutional neural networks	~77.4%
2018	SENet-20	Deep convolutional neural networks	~77.0%
2018	SENet-8	Deep convolutional neural networks	~76.5%
2018	SENet-4	Deep convolutional neural networks	~76.0%
2018	SENet-2	Deep convolutional neural networks	~75.5%
2018	SENet-1	Deep convolutional neural networks	~75.0%
2018	SENet-0	Deep convolutional neural networks	~74.5%
2018	SENet-0.5	Deep convolutional neural networks	~74.0%
2018	SENet-0.25	Deep convolutional neural networks	~73.5%
2018	SENet-0.125	Deep convolutional neural networks	~73.0%
2018	SENet-0.0625	Deep convolutional neural networks	~72.5%
2018	SENet-0.03125	Deep convolutional neural networks	~72.0%
2018	SENet-0.015625	Deep convolutional neural networks	~71.5%
2018	SENet-0.0078125	Deep convolutional neural networks	~71.0%
2018	SENet-0.00390625	Deep convolutional neural networks	~70.5%
2018	SENet-0.001953125	Deep convolutional neural networks	~70.0%
2018	SENet-0.0009765625	Deep convolutional neural networks	~69.5%
2018	SENet-0.00048828125	Deep convolutional neural networks	~69.0%
2018	SENet-0.000244140625	Deep convolutional neural networks	~68.5%
2018	SENet-0.0001220703125	Deep convolutional neural networks	~68.0%
2018	SENet-0.00006103515625	Deep convolutional neural networks	~67.5%
2018	SENet-0.000030517578125	Deep convolutional neural networks	~67.0%
2018	SENet-0.0000152587890625	Deep convolutional neural networks	~66.5%
2018	SENet-0.00000762939453125	Deep convolutional neural networks	~66.0%
2018	SENet-0.000003814697265625	Deep convolutional neural networks	~65.5%
2018	SENet-0.0000019073486328125	Deep convolutional neural networks	~65.0%
2018	SENet-0.00000095367431640625	Deep convolutional neural networks	~64.5%
2018	SENet-0.000000476837158203125	Deep convolutional neural networks	~64.0%
2018	SENet-0.0000002384185791015625	Deep convolutional neural networks	~63.5%
2018	SENet-0.00000011920928955078125	Deep convolutional neural networks	~63.0%
2018	SENet-0.000000059604644775390625	Deep convolutional neural networks	~62.5%
2018	SENet-0.0000000298023223876953125	Deep convolutional neural networks	~62.0%
2018	SENet-0.00000001490116119384765625	Deep convolutional neural networks	~61.5%
2018	SENet-0.000000007450580596923828125	Deep convolutional neural networks	~61.0%
2018	SENet-0.0000000037252902984619140625	Deep convolutional neural networks	~60.5%
2018	SENet-0.00000000186264514923095703125	Deep convolutional neural networks	~60.0%
2018	SENet-0.000000000931322574615478515625	Deep convolutional neural networks	~59.5%
2018	SENet-0.0000000004656612873077392828125	Deep convolutional neural networks	~59.0%
2018	SENet-0.00000000023283064365386964140625	Deep convolutional neural networks	~58.5%
2018	SENet-0.000000000116415321826934820703125	Deep convolutional neural networks	~58.0%
2018	SENet-0.00000000005820766091346741015625	Deep convolutional neural networks	~57.5%
2018	SENet-0.000000000029103830456733705078125	Deep convolutional neural networks	~57.0%
2018	SENet-0.0000000000145519152283688535390625	Deep convolutional neural networks	~56.5%
2018	SENet-0.00000000000727595761418442676953125	Deep convolutional neural networks	~56.0%
2018	SENet-0.0000000000036379788070922134820703125	Deep convolutional neural networks	~55.5%
2018	SENet-0.000000000001818989403546106719640625	Deep convolutional neural networks	~55.0%
2018	SENet-0.0000000000009094947017730533820703125	Deep convolutional neural networks	~54.5%
2018	SENet-0.00000000000045474735088652676953125	Deep convolutional neural networks	~54.0%
2018	SENet-0.0000000000002273736754432833820703125	Deep convolutional neural networks	~53.5%
2018	SENet-0.000000000000113686837721641691015625	Deep convolutional neural networks	~53.0%
2018	SENet-0.0000000000000568434188620848460703125	Deep convolutional neural networks	~52.5%
2018	SENet-0.000000000000028421709430442423095703125	Deep convolutional neural networks	~52.0%
2018	SENet-0.0000000000000142108547172212115478515625	Deep convolutional neural networks	~51.5%
2018	SENet-0.000000000000007105427358610577392828125	Deep convolutional neural networks	~51.0%
2018	SENet-0.0000000000000035527136793052886964140625	Deep convolutional neural networks	~50.5%
2018	SENet-0.00000000000000177635683965264434820703125	Deep convolutional neural networks	~50.0%
2018	SENet-0.00000000000000088817841982632219640625	Deep convolutional neural networks	~49.5%
2018	SENet-0.00000000000000044408920991311015625	Deep convolutional neural networks	~49.0%
2018	SENet-0.0000000000000002220446049565552833820703125	Deep convolutional neural networks	~48.5%
2018	SENet-0.0000000000000001110223024782776953125	Deep convolutional neural networks	~48.0%
2018	SENet-0.0000000000000000555111512391392828125	Deep convolutional neural networks	~47.5%
2018	SENet-0.0000000000000000277555756195986964140625	Deep convolutional neural networks	~47.0%
2018	SENet-0.00000000000000001387778780979934820703125	Deep convolutional neural networks	~46.5%
2018	SENet-0.0000000000000000069388939048996964140625	Deep convolutional neural networks	~46.0%
2018	SENet-0.00000000000000000346944695244984820703125	Deep convolutional neural networks	~45.5%
2018	SENet-0.0000000000000000017347234762249241015625	Deep convolutional neural networks	~45.0%
2018	SENet-0.0000000000000000008673617381123052833820703125	Deep convolutional neural networks	~44.5%
2018	SENet-0.00000000000000000043368086906115646964140625	Deep convolutional neural networks	~44.0%
2018	SENet-0.000000000000000000216840434530578241015625	Deep convolutional neural networks	~43.5%
2018	SENet-0.000000000000000000108420217265289123095703125	Deep convolutional neural networks	~43.0%
2018	SENet-0.000000000000000000054210108632644595703125	Deep convolutional neural networks	~42.5%
2018	SENet-0.000000000000000000027105054316322297876953125	Deep convolutional neural networks	~42.0%
2018	SENet-0.000000000000000000013552502658161614820703125	Deep convolutional neural networks	~41.5%
2018	SENet-0.00000000000000000000677625132908080703125	Deep convolutional neural networks	~41.0%
2018	SENet-0.0000000000000000000033881256645404035390625	Deep convolutional neural networks	~40.5%
2018	SENet-0.0000000000000000000016940628272702019640625	Deep convolutional neural networks	~40.0%
2018	SENet-0.00000000000000000000084703141361015625	Deep convolutional neural networks	~39.5%
2018	SENet-0.000000000000000000000423515706805078125	Deep convolutional neural networks	~39.0%
2018	SENet-0.00000000000000000000021175785340392828125	Deep convolutional neural networks	~38.5%
2018	SENet-0.0000000000000000000001058789267019640625	Deep convolutional neural networks	~38.0%
2018	SENet-0.000000000000000000000052939463350984820703125	Deep convolutional neural networks	~37.5%
2018	SENet-0.00000000000000000000002646973167549241015625	Deep convolutional neural networks	~37.0%
2018	SENet-0.0000000000000000000000132348658377461015625	Deep convolutional neural networks	~36.5%
2018	SENet-0.000000000000000000000006617432918852833820703125	Deep convolutional neural networks	~36.0%
2018	SENet-0.000000000000000000000003308716459426964140625	Deep convolutional neural networks	~35.5%
2018	SENet-0.0000000000000000000000016543582297134820703125	Deep convolutional neural networks	~35.0%
2018	SENet-0.000000000000000000000000827179114861015625	Deep convolutional neural networks	~34.5%
2018	SENet-0.000000000000000000000000413589562373052833820703125	Deep convolutional neural networks	~34.0%
2018	SENet-0.000000000000000000000000206794781188123095703125	Deep convolutional neural networks	~33.5%
2018	SENet-0.000000000000000000000000103397390594115646964140625	Deep convolutional neural networks	~33.0%
2018	SENet-0.000000000000000000000000051698695297078241015625	Deep convolutional neural networks	~32.5%
2018	SENet-0.00000000000000000000000002584934764861015625	Deep convolutional neural networks	~32.0%
2018	SENet-0.00000000000000000000000001292467382430578125	Deep convolutional neural networks	~31.5%
2018	SENet-0.00000000000000000000000000646233691215478515625	Deep convolutional neural networks	~31.0%
2018	SENet-0.00000000000000000000000000323116845607876953125	Deep convolutional neural networks	~30.5%
2018	SENet-0.0000000000000000000000000016155842280392828125	Deep convolutional neural networks	~30.0%
2018	SENet-0.000000000000000000000000000807792114019640625	Deep convolutional neural networks	~29.5%
2018	SENet-0.000000000000000000000000000403896057019640625	Deep convolutional neural networks	~29.0%
2018	SENet-0.00000000000000000000000000020194802850984820703125	Deep convolutional neural networks	~28.5%
2018	SENet-0.0000000000000000000000000001009740142549241015625	Deep convolutional neural networks	~28.0%
2018	SENet-0.000000000000000000000000000050487007127461015625	Deep convolutional neural networks	~27.5%
2018	SENet-0.00000000000000000000000000002524350356373052833820703125	Deep convolutional neural networks	~27.0%
2018	SENet-0.000000000000000000000000000012621751781676953125	Deep convolutional neural networks	~26.5%
2018	SENet-0.000000000000000000000000000006310875890876953125	Deep convolutional neural networks	~26.0%
2018	SENet-0.0000000000000000000000000000031554379454392828125	Deep convolutional neural networks	~25.5%
2018	SENet-0.000000000000000000000000000001577718972719640625	Deep convolutional neural networks	~25.0%
2018	SENet-0.000000000000000000000000000000788859486373052833820703125	Deep convolutional neural networks	~24.5%
2018	SENet-0.0000000000000000000000000000003944297431676953125	Deep convolutional neural networks	~24.0%
2018	SENet-0.000000000000000000000000000000197214871576953125	Deep convolutional neural networks	~23.5%
2018	SENet-0.0000000000000000000000000000000986074357876953125	Deep convolutional neural networks	~23.0%
2018	SENet-0.00000000000000000000000000000004930371789392828125	Deep convolutional neural networks	~22.5%
2018	SENet-0.000000000000000000000000000000024651858946964140625	Deep convolutional neural networks	~22.0%
2018	SENet-0.0000000000000000000000000000000123259294734820703125	Deep convolutional neural networks	~21.5%
2018	SENet-0.0000000000000000000000000000000061629647373052833820703125	Deep convolutional neural networks	~21.0%
2018	SENet-0.0000000000000000000000000000000030814823686964140625	Deep convolutional neural networks	~20.5%
2018	SENet-0.00000000000000000000000000000000154074118434820703125	Deep convolutional neural networks	~20.0%
2018	SENet-0.000000000000000000000000000000000770370592173052833820703125	Deep convolutional neural networks	~19.5%
2018	SENet-0.0000000000000000000000000000000003851852960876953125	Deep convolutional neural networks	~19.0%
2018	SENet-0.00000000000000000000000000000000019259264804392828125	Deep convolutional neural networks	~18.5%
2018	SENet-0.0000000000000000000000000000000000962963240219640625	Deep convolutional neural networks	~18.0%
2018	SENet-0.000000000000000000000000000000000048148162010984820703125	Deep convolutional neural networks	~17.5%
2018	SENet-0.00000000000000000000000000000000002407408100549241015625	Deep convolutional neural networks	~17.0%
2018	SENet-0.0000000000000000000000000000000000120370405027461015625	Deep convolutional neural networks	~16.5%
2018	SENet-0.0000000000000000000000000000000000060185202513730528338207		

# Let us train a state of the art AI!

My contribution:



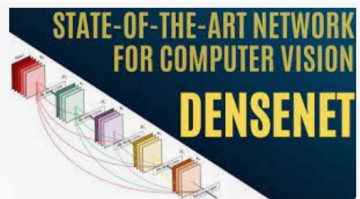
state of the art cnn



CNN Models	Param.	FLOPs	Top-1	Top-5
ResNet-152	57.40M	10.82G	77.58	93.66
SENet-152	63.68M	10.85G	78.43	94.27
ResNet-200	74.45M	14.10G	78.20	94.00
ResNeXt-101	46.66M	7.53G	78.80	94.40
DenseNet-264	28.78M	5.15G	77.85	93.78
ECA-Net50 (Ours)	24.37M	3.86G	77.48	93.68
ECA-Net101 (Ours)	42.49M	7.35G	78.65	94.34

SENet (Hu, Shen, and Sun 2018), CBAM (Woo et al. 2018) and AA-Net (Bello et al. 2019). From Table 2 we can see

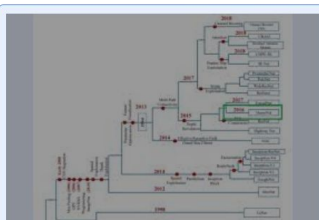
Comparisons with other state-of-the-art CNN models on Image...  
researchgate.net



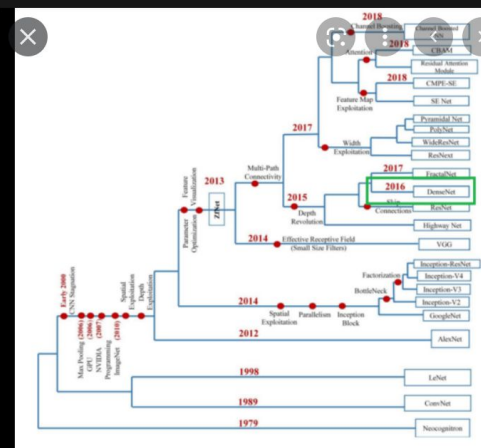
State of the Art Convolutional Neural Networks (CNNs) Expl...  
youtube.com

Year	Model	Architecture	Top-1 Accuracy
2012	AlexNet	Deep convolutional neural networks	~50%
2013	ZFNet	Deep convolutional neural networks	~56%
2014	VGG	Deep convolutional neural networks	~70%
2014	GoogLeNet (Inception v1)	Deep convolutional neural networks	~74%
2015	ResNet	Deep convolutional neural networks	~76%
2015	SENet	Deep convolutional neural networks	~78%
2016	DenseNet	Deep convolutional neural networks	~79%
2017	SENet	Deep convolutional neural networks	~80%
2018	SENet	Deep convolutional neural networks	~81%
2019	SENet	Deep convolutional neural networks	~82%

Summary of state-of-the-art Convolutional...  
researchgate.net



State-of-the-Art Convolutional Neural Network...  
louisbouchard.ai



Louis Bouchard

Besuchen

State-of-the-Art Convolution...

# Let us train a state of the art AI!

Google search for "state of the art cnn".

CNN Models	Param.	FLOPs	Top-1	Top-5
ResNet-152	57.40M	10.82G	77.58	93.6
SENet-200	63.68M	10.85G	78.43	94.2
ResNet-200	74.45M	14.10G	78.20	94.0
ResNeXt-101	46.66M	7.53G	78.80	94.4
DenseNet-264	28.78M	5.15G	77.85	93.7
ECA-Net50 (Ours)	24.37M	3.86G	77.48	93.6
ECA-Net101 (Ours)	42.49M	7.35G	78.65	94.3

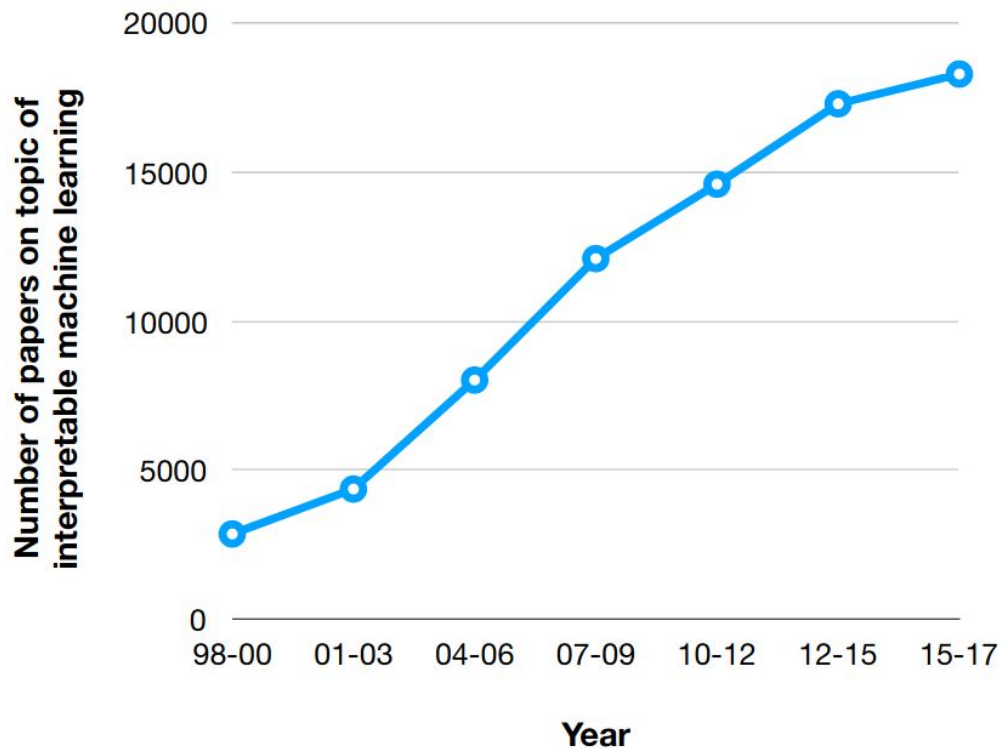
Accuracy: 99%

State of the Art Convolutional Neural Networks (CNNs) B... youtube.com

State-of-the-Art Convolutional Neural Network... louisbouchard.ai

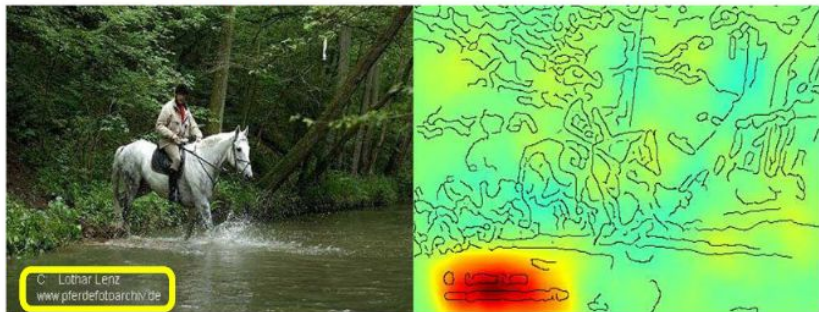
State-of-the-Art Convolution... Besuchen

# Some people thought it might be cool to explain my model



# Let us train a state of the art AI!

Horse-picture from Pascal VOC data set



Source tag present



Classified as horse

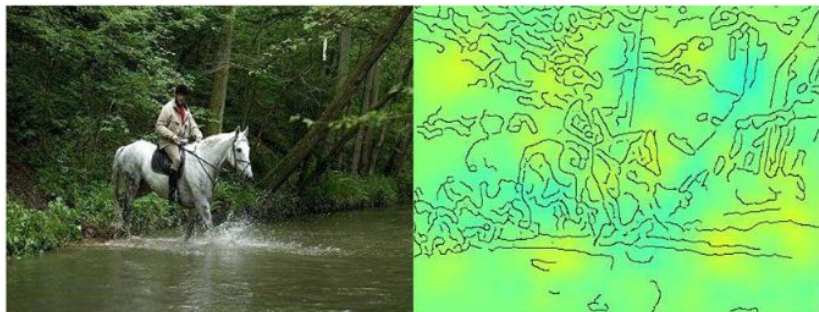
Artificial picture of a car



No source tag present



Not classified as horse



What is an Explanation Model?



# What is an Explanation Model?

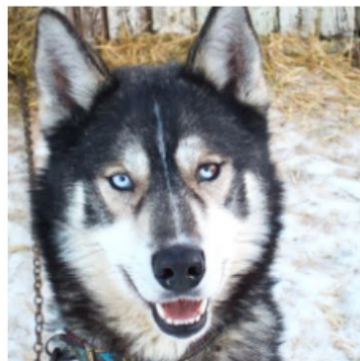
- Model that explains a different model

## What is an Explanation Model?

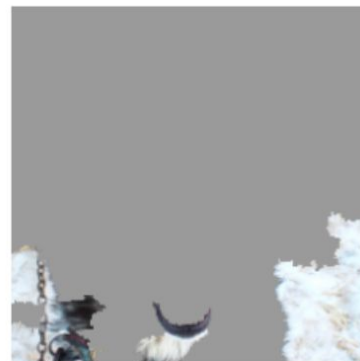
- Model that explains a different model  
→ ask a **better** question

# What is the **GOAL** of your explanation model?

- Understanding
- Trust
- Feature Selection
- Actionable Advice



(a) Husky classified as wolf



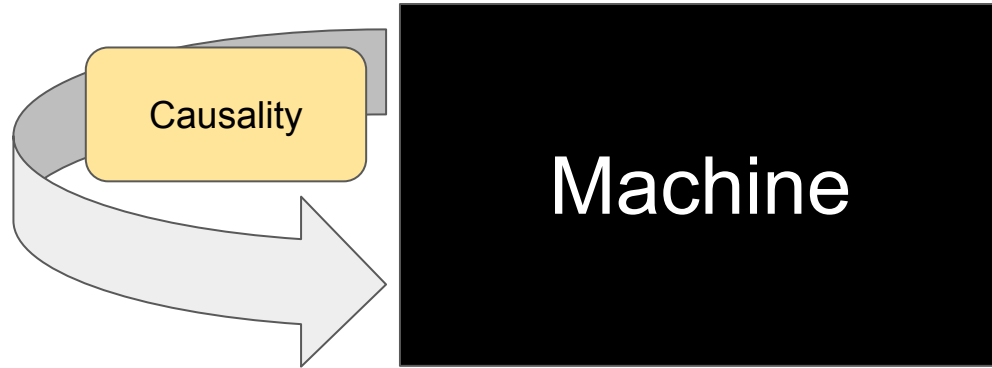
(b) Explanation

# Robust and verified Machine Learning

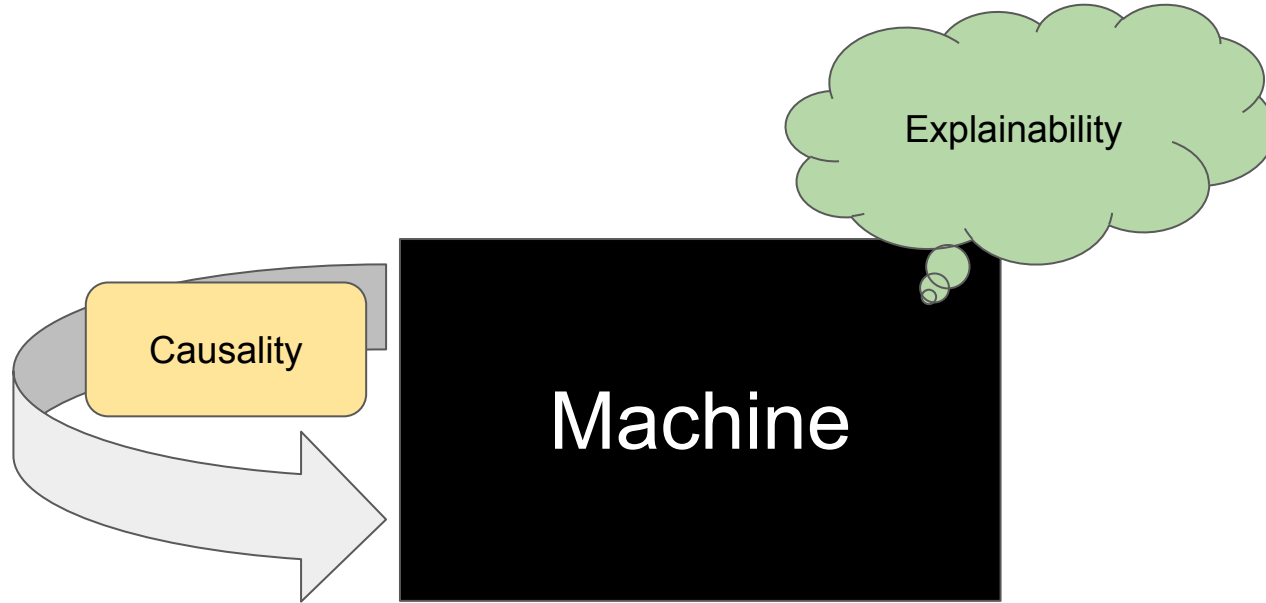


Machine

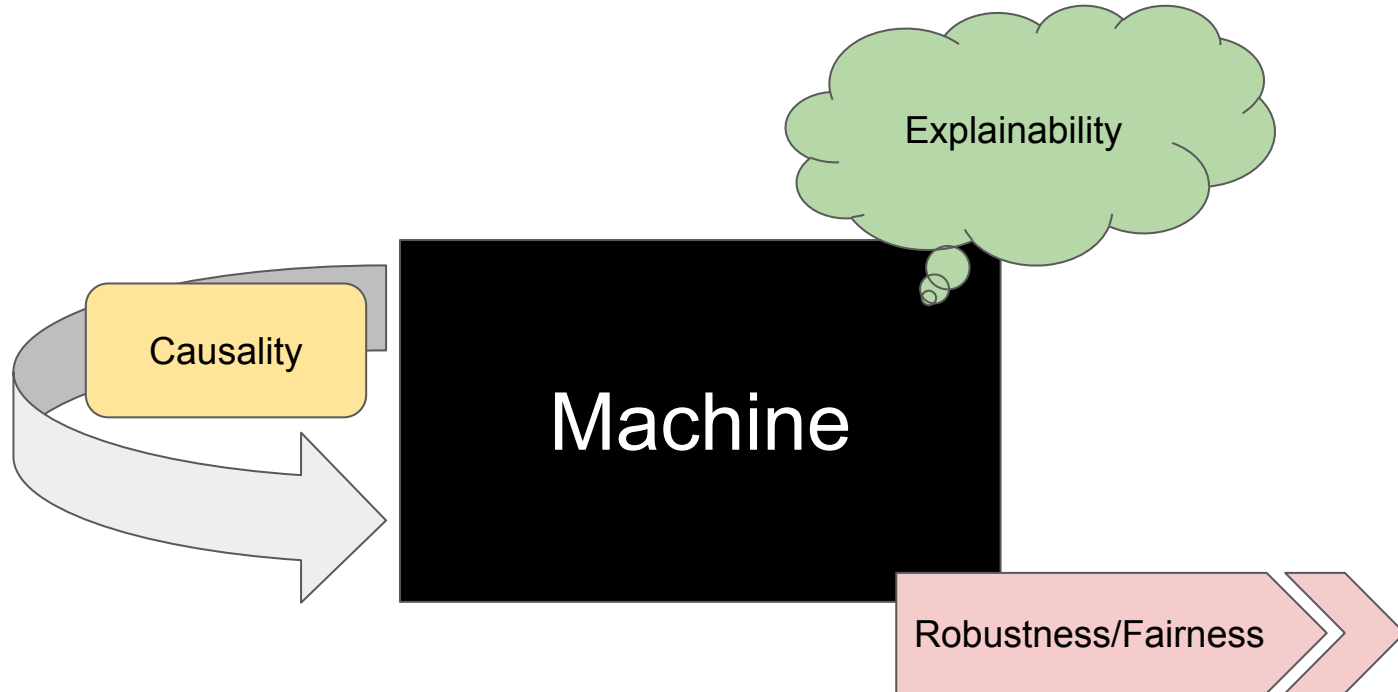
# Robust and verified Machine Learning



# Robust and verified Machine Learning



# Robust and verified Machine Learning



# We need Explainability!





# Different Camps of XAI

## Local interpretability

Why is **this image** labelled as a car?



Image from Lapuschkin (2019)

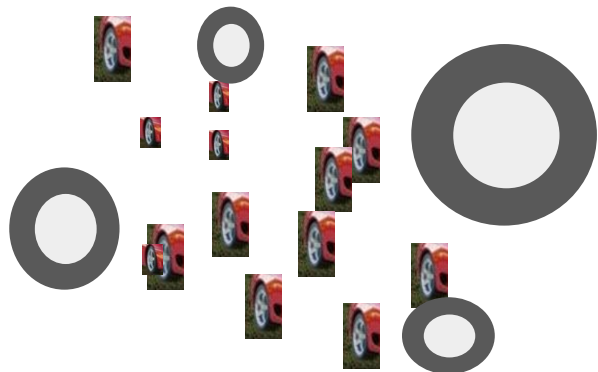


Image from Lapuschkin (2019)

## Global interpretability

What do **all car-labelled images** have in common?

# Different Camps of XAI

## Model-agnostic interpretability



*I will explain you no matter what!*

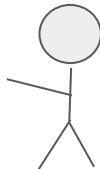
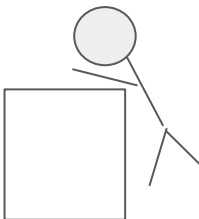


Image from <https://subtleyoga.com/why-now-perhaps-more-than-ever-is-a-good-time-to-have-a-magic-word/wingardium-leviosa/>



Image from <https://sateroad.org/best-tool-chests-reviews/>

## Model-specific interpretability



*How does it look inside?*

from her MLSS22 talk

# Different Camps of XAI

## Intrinsic interpretability

```
If age > 25 then predict favorite sport = tennis
```

<https://corels.eecs.harvard.edu/corels/whatareulelists.html>



Image from <https://users.cs.duke.edu/~cynthia/>



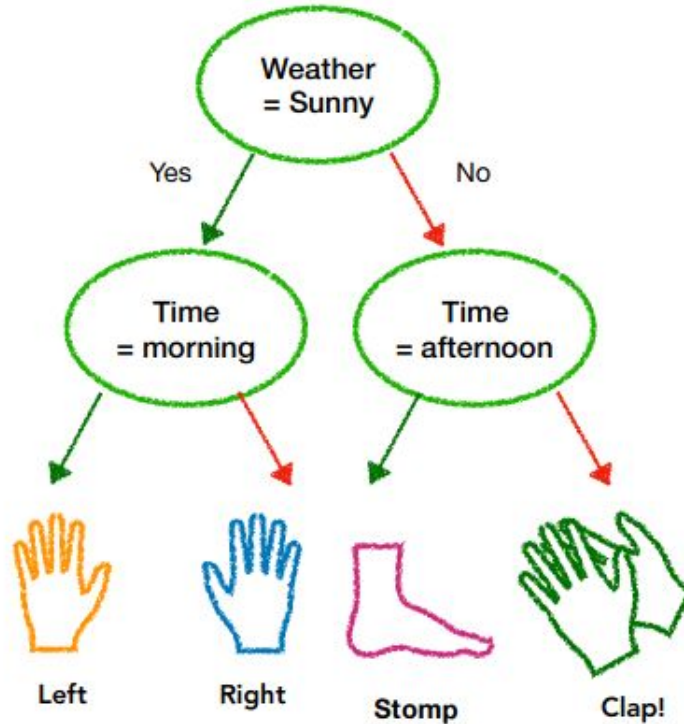
Image from <https://beenkim.github.io/>

## Post-hoc interpretability

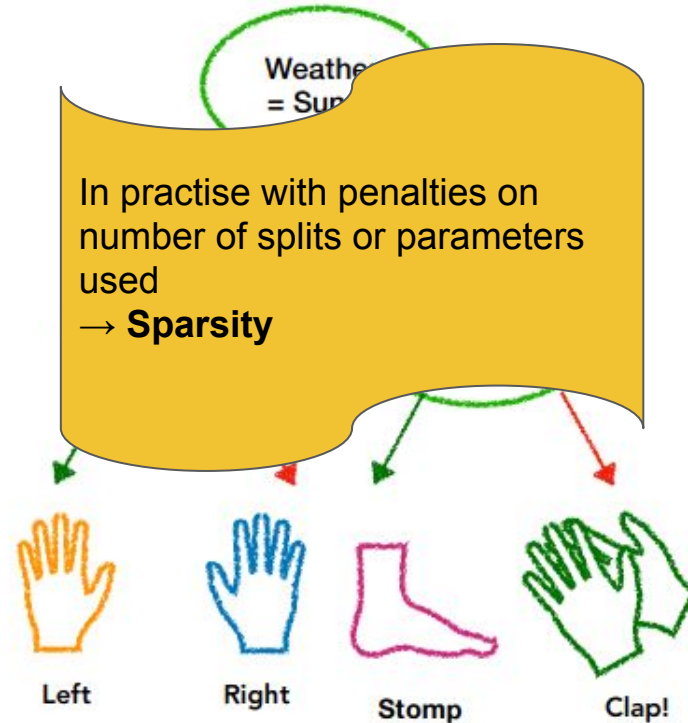
“Probability distortion is that people generally do not look at the value of probability uniformly between 0 and 1. Lower probability is said to be over-weighted while medium to high probability is under-weighted” - Kahneman

from her MLSS22 talk

# Decision trees / Rule Lists are explainable!



# Decision trees / Rule Lists are explainable!



# Risk-Calibrated Supersparse Linear Integer Models

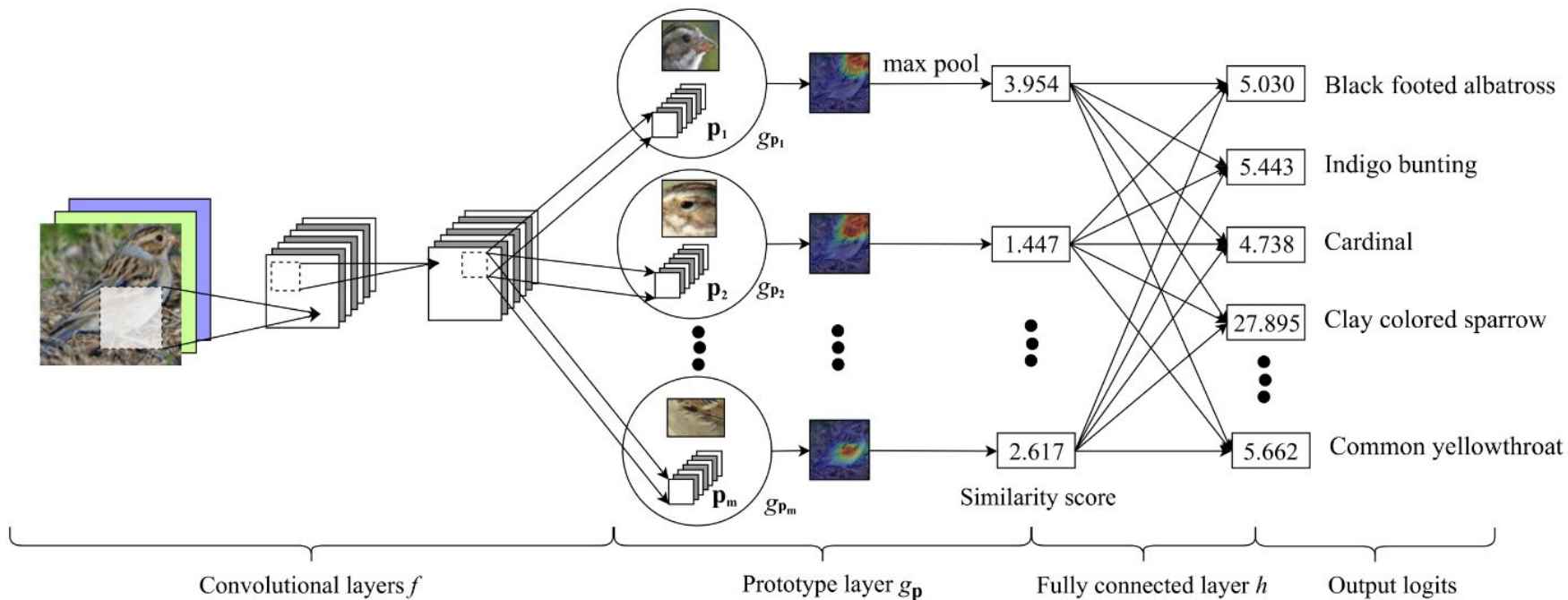
$$\min_{\lambda \in \mathcal{L}} \underbrace{\frac{1}{n} \sum_{i=1}^n \log(1 + e^{-y_i \mathbf{x}_i \lambda})}_{\text{Logistic Loss}} + \underbrace{C_0 \|\lambda\|_0}_{\text{Model Size}}$$

$\lambda \in \mathcal{L}$  means that  $\forall j, \lambda_j \in \{-10, -9, \dots, 0, \dots, 9, 10\}$

**Small  
Integer  
Coefficients**

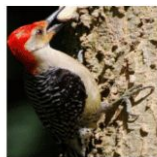
↙  
RiskSLIM's Mixed-Integer  
Nonlinear Program

This looks like that (Chen, 2019)



# This looks like that (Chen, 2019)

Why is this bird classified as a red-bellied woodpecker?



Evidence for this bird being a red-bellied woodpecker:

Original image (box showing part that looks like prototype)	Prototype	Training image where prototype comes from	Activation map	Similarity score	Class connection	Points contributed
				6.499	1.180	7.669
				4.392	1.127	4.950
				3.890	1.108	4.310
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Total points to red-bellied woodpecker: 32.736

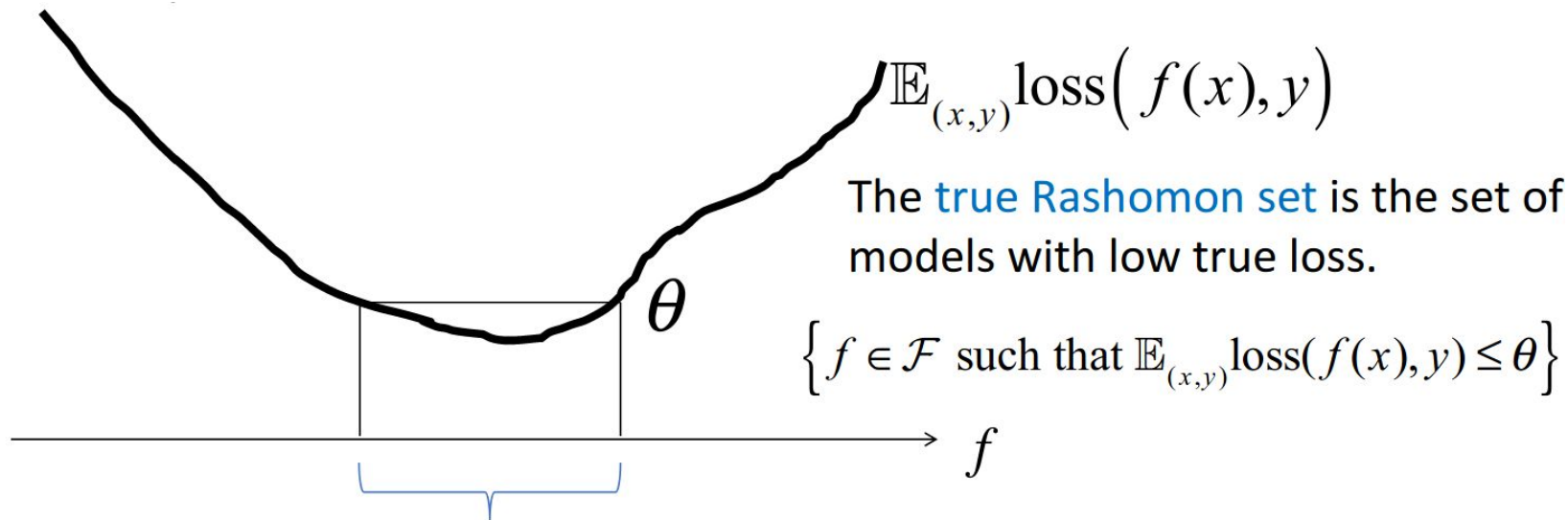
Evidence for this bird being a red-cockaded woodpecker:

Original image (box showing part that looks like prototype)	Prototype	Training image where prototype comes from	Activation map	Similarity score	Class connection	Points contributed
				2.452	1.046	2.565
				2.125	1.091	2.318
				1.945	1.069	2.079
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Total points to red-cockaded woodpecker: 16.886



# True Rashomon Sets



If the **true Rashomon set** is large, a simple-yet-accurate model is likely to exist.

# Rashomon Ratio

$$\text{Rashomon Ratio } (\mathcal{F}_2, \theta) := \frac{\#\{f_2 \in \mathcal{F}_2 \text{ such that } L(f_2) \leq \theta\}}{|\mathcal{F}_2|}$$

# You can bound the loss in performance of simple models!

**Theorem.** For any  $\epsilon > 0$ , with probability at least  $(1 - \epsilon)p$ , with respect to the random draw of functions from  $\mathcal{F}_2$  to form  $\mathcal{F}_1$ , and with respect to the random draw of iid data:

$$\left| L(f_2^*) - \hat{L}(\hat{f}_1) \right| \leq \theta + 2b \sqrt{\frac{\log |\mathcal{F}_1| + \log \frac{2}{\epsilon}}{2n}}, \text{ where } p = 1 - \frac{\left( 1 - \text{Rashomon Ratio}(\mathcal{F}_2, \theta) \frac{|\mathcal{F}_2|}{|\mathcal{F}_1|} \right)}{\left( \frac{|\mathcal{F}_2|}{|\mathcal{F}_1|} \right)}.$$

# You can bound the loss in performance of simple models!

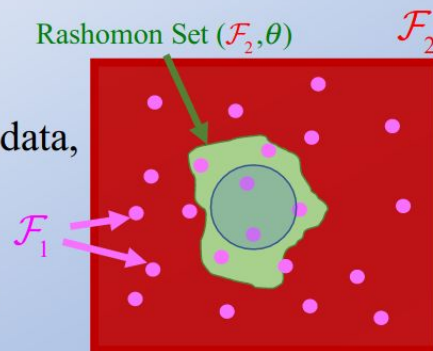
## Theorem

For a  $\mathcal{K}$ -Lipschitz loss  $l$  bounded by  $b$ , hypothesis spaces  $\mathcal{F}_1$  and  $\mathcal{F}_2$ ,  $\mathcal{F}_1 \subset \mathcal{F}_2$ , if for each  $f_2 \in \text{Rashomon set}(\mathcal{F}_2, \theta)$  there exists a model  $f_1 \in \mathcal{F}_1$  such that  $\|f_2 - f_1\|_p \leq \delta$ , and if the Rashomon set is large, in that it contains an  $\ell_p$  ball of size at least  $\delta$ , then there exists  $\bar{f}_1 \in \text{Rashomon set}(\mathcal{F}_2, \theta)$  such that for a fixed parameter  $\epsilon \in (0, 1)$ :

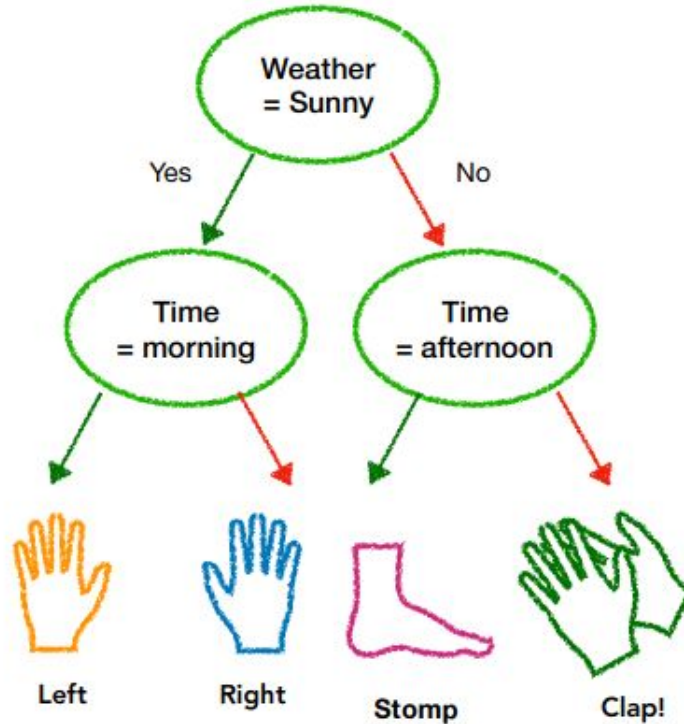
1.  $\bar{f}_1$  is from simpler class  $\mathcal{F}_1$ .
2. With prob  $\geq 1 - \epsilon$  w.r.t. the random draw of training data,

$$\left| L(\bar{f}_1) - \hat{L}(\bar{f}_1) \right| \leq 2KR_n(\mathcal{F}_1) + b\sqrt{\frac{\log(2/\epsilon)}{2n}},$$

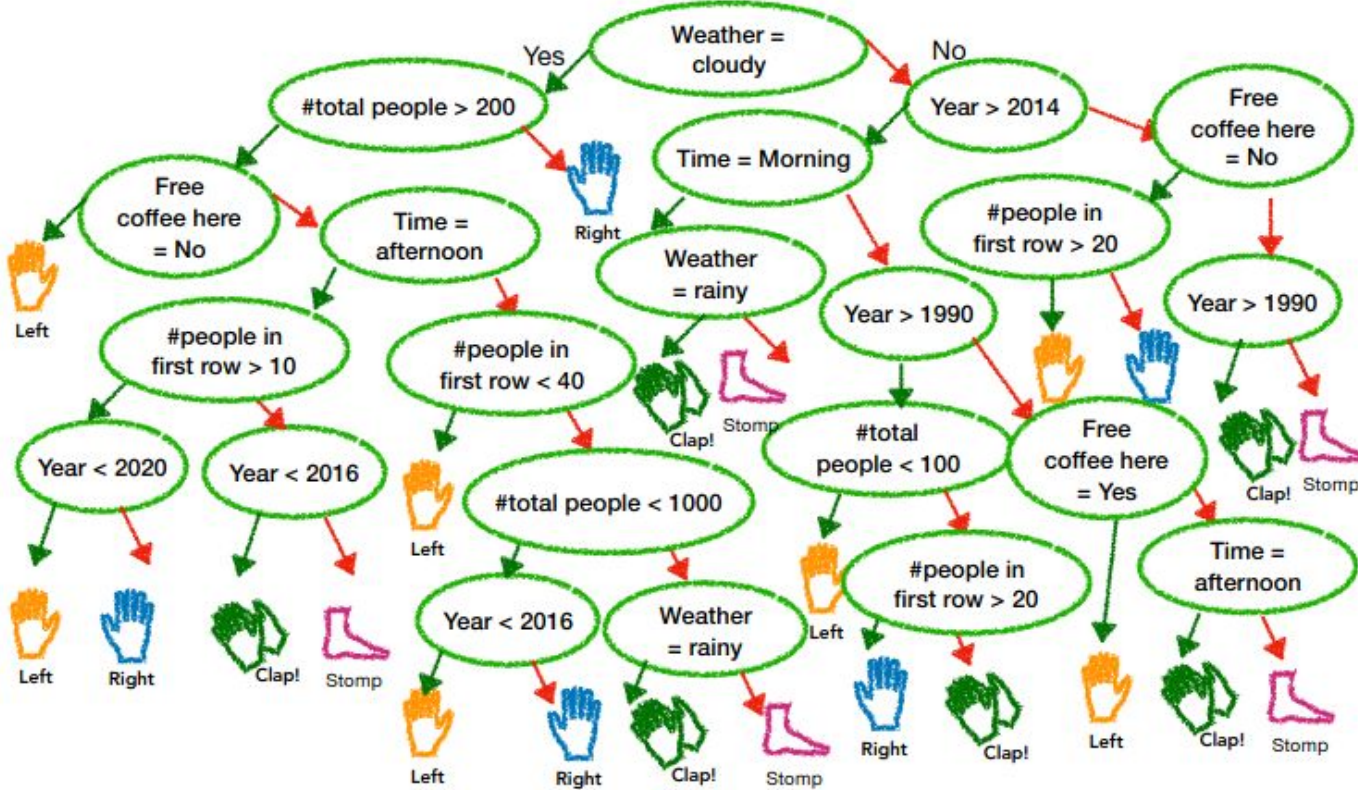
where  $R_n(\mathcal{F}_1)$  is Rademacher complexity.



# Decision trees are explainable!



# Decision trees are explainable?



# Counterfactual Explanations (aka Algorithmic Recourse)



# Counterfactual Explanations (aka Algorithmic Recourse)



**LOAN DENIED**



# Counterfactual Explanations (aka Algorithmic Recourse)





Salary:  
£17,000,  
Savings:  
£341.52



**LOAN DENIED**

*How would the numbers need to change the least to flip the decision?*



# Counterfactual Explanations (aka Algorithmic Recourse)

	Salary: £17,000, Savings: £341.52
	Salary: <b>£25,000,</b> Savings: £341.52

**LOAN DENIED**

*How would the numbers need to change the least to flip the decision?*

# Counterfactual Explanations (aka Algorithmic Recourse)

	Salary: £17,000, Savings: £341.52
	Salary: £23,000, Savings: £3,341.52

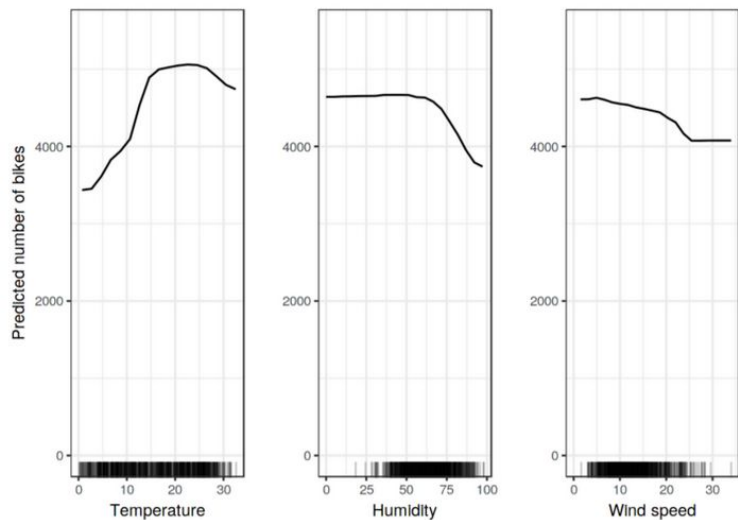
**LOAN DENIED**

*How would I need to interfere the least to flip the decision?*

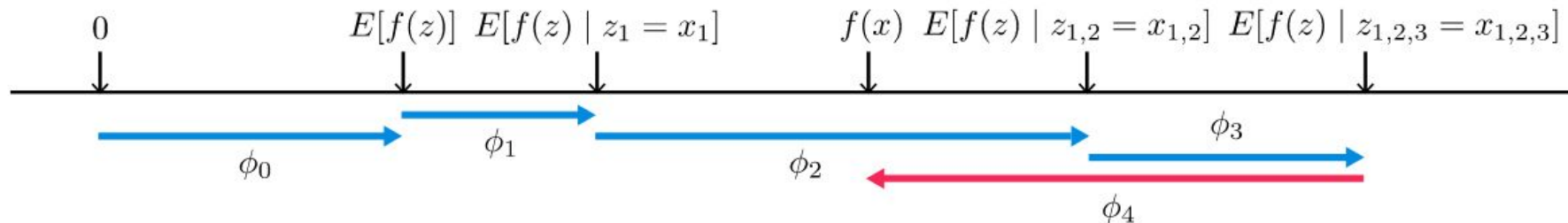
With causality

# Partial Dependence Plots

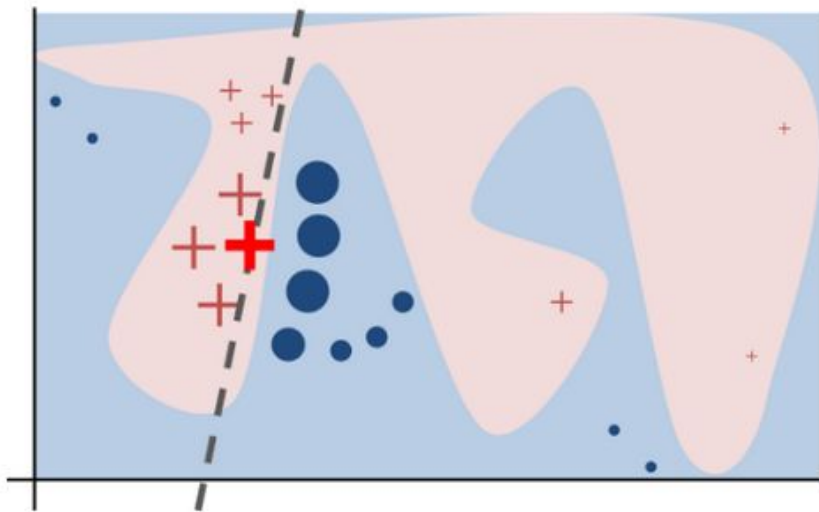
$$\hat{f}_{x_S}(x_S) = E_{x_C} \left[ \hat{f}(x_S, x_C) \right] = \int \hat{f}(x_S, x_C) d\mathbb{P}(x_C)$$



# Shapley Values (Lundberg and Lee, 2017)



# LIME (Ribeiro et al., 2016)



Ribeiro et al. (2017)

- black box classification model  $f$ : pink and blue areas
- instance being explained: bold red cross
- instances sampled locally and weighted by their proximity: red crosses, and blue circles
- locally faithful explanation  $g$ : dashed line

# LIME (Ribeiro et al., 2016)

$$\xi(x) = \operatorname{argmin}_{g \in G}$$

black box

neighbourhood  
kernel around  $x$

$$\mathcal{L}(f, g, \pi_x) + \Omega(g)$$

explanation  
model

fit  $g$  to  $f$  in a small  
neighbourhood around  $x$

ensure  $g$  is simple

# Problems with Tangent Line Approximations

- If a linear fit is good enough, why not just have a locally linear black box?  
(Rudin, 2019)
- Consider the black box

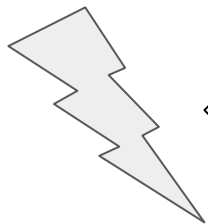
$$f(x) = \mathbb{I}(x_1 > 0)2x_2^2 - \mathbb{I}(x_1 \leq 0)x_2^2$$



# Problems with Tangent Line Approximations

- If a linear fit is good enough, why not just have a locally linear black box?  
(Rudin, 2019)
- Consider the black box

$$f(x) = \mathbb{I}(x_1 > 0)2x_2^2 - \mathbb{I}(x_1 \leq 0)x_2^2$$



for  $x_1 = -0.001$ , the feature attribution of Feature-2 is **negative**

Can we just take the gradients?

No!

# Can we just take the gradients?

No!

Look at

$$f(x) = \text{ReLU}(1-x)$$

we want to explain  $x^*=2$

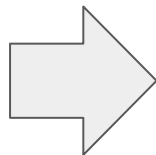
# Can we just take the gradients?

No!

Look at

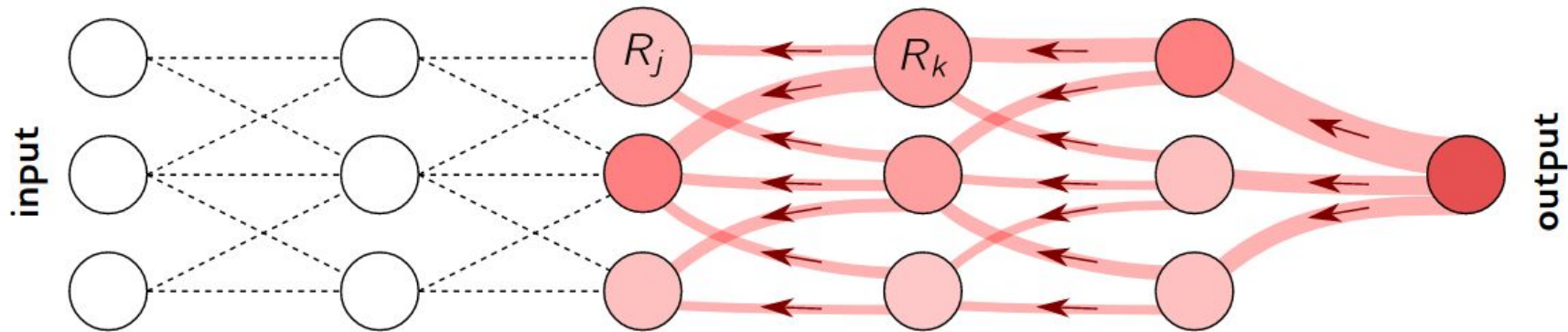
$$f(x) = \text{ReLU}(1-x)$$

we want to explain  $x^*=2$



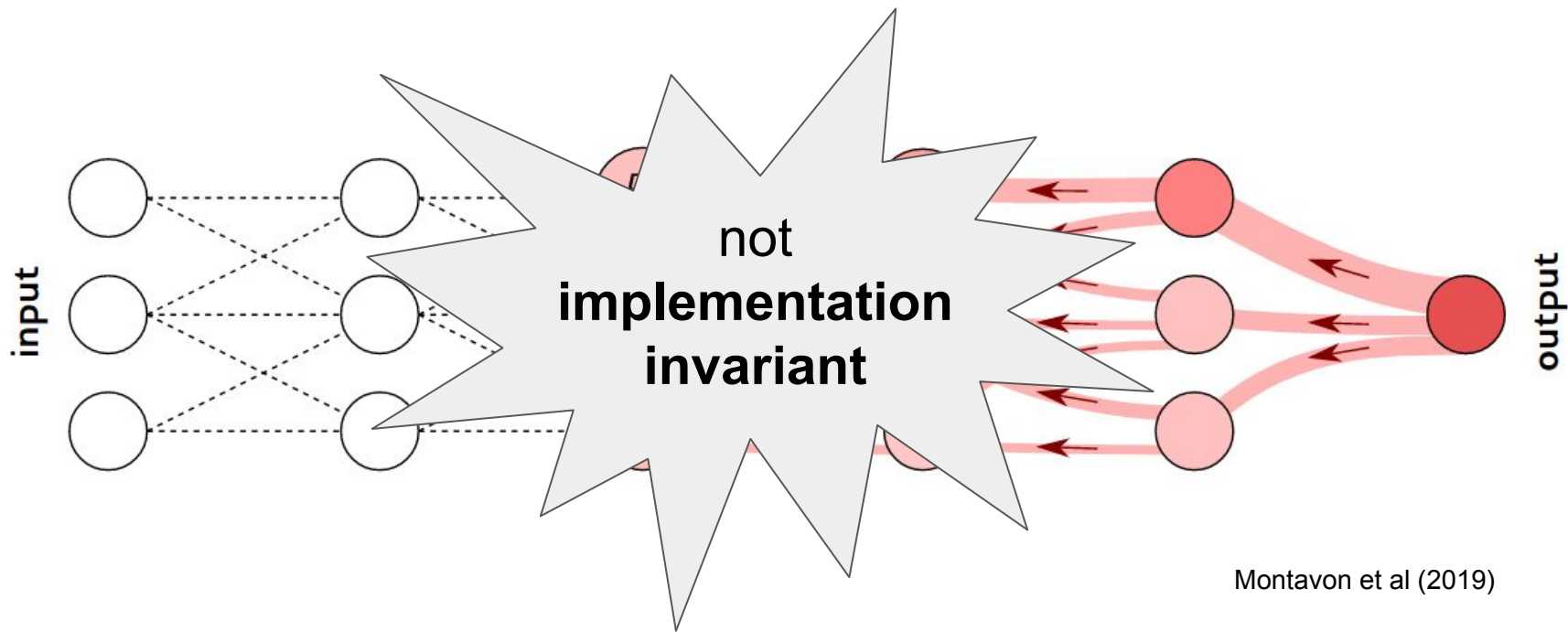
not **sensitive** wrt  $x=0$

# Layer-wise relevance propagation (Bach et al, 2016)



Montavon et al (2019)

# Layer-wise relevance propagation (Bach et al, 2016)



Montavon et al (2019)

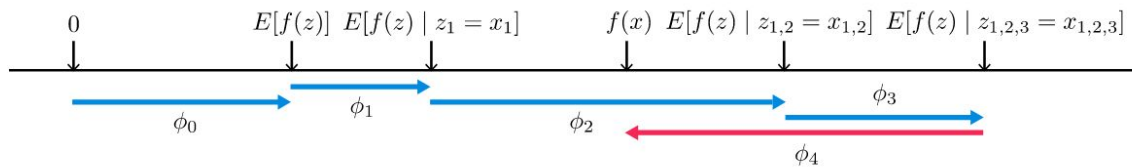
# Integrated Gradients (Sundararajan et al, 2017)

○  $s_1, s_2$

○  $r_1, r_2$

# Integrated Gradients (Sundararajan et al, 2017)

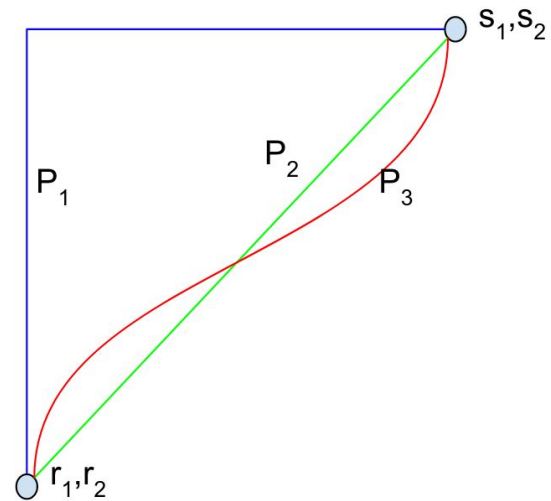
○  $S_1, S_2$



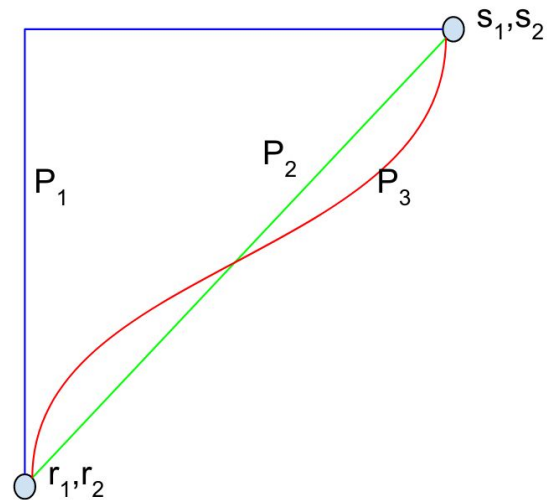
○  $r_1, r_2$



# Integrated Gradients (Sundararajan et al, 2017)

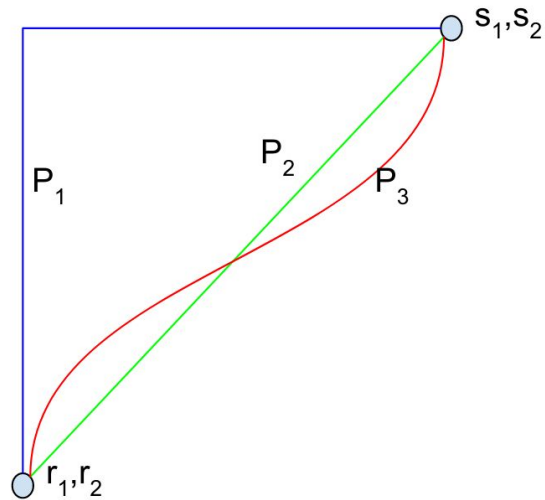


# Integrated Gradients (Sundararajan et al, 2017)



$$\text{PathIntegratedGrads}_i^\gamma(x) ::= \int_{\alpha=0}^1 \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha$$

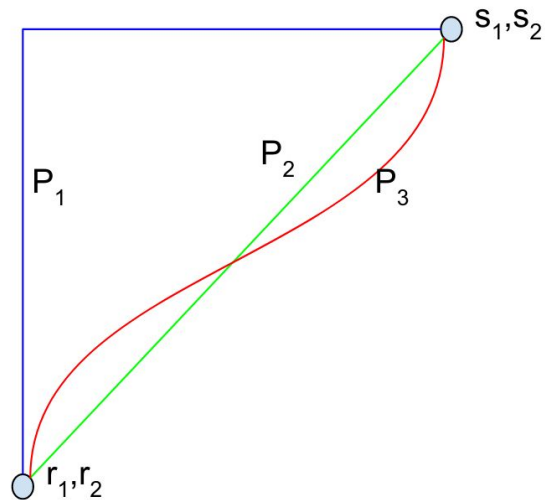
# Integrated Gradients (Sundararajan et al, 2017)



$$\text{IntegratedGrads}_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

# Integrated Gradients (Sundararajan et al, 2017)

- + Sensitivity
- + Implementation Invariance
- + Completeness
- + Linearity



$$\text{IntegratedGrads}_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

# Integrated Gradients (Sundararajan et al, 2017)

Original image



Top label and score

Top label: reflex camera

Score: 0.993755

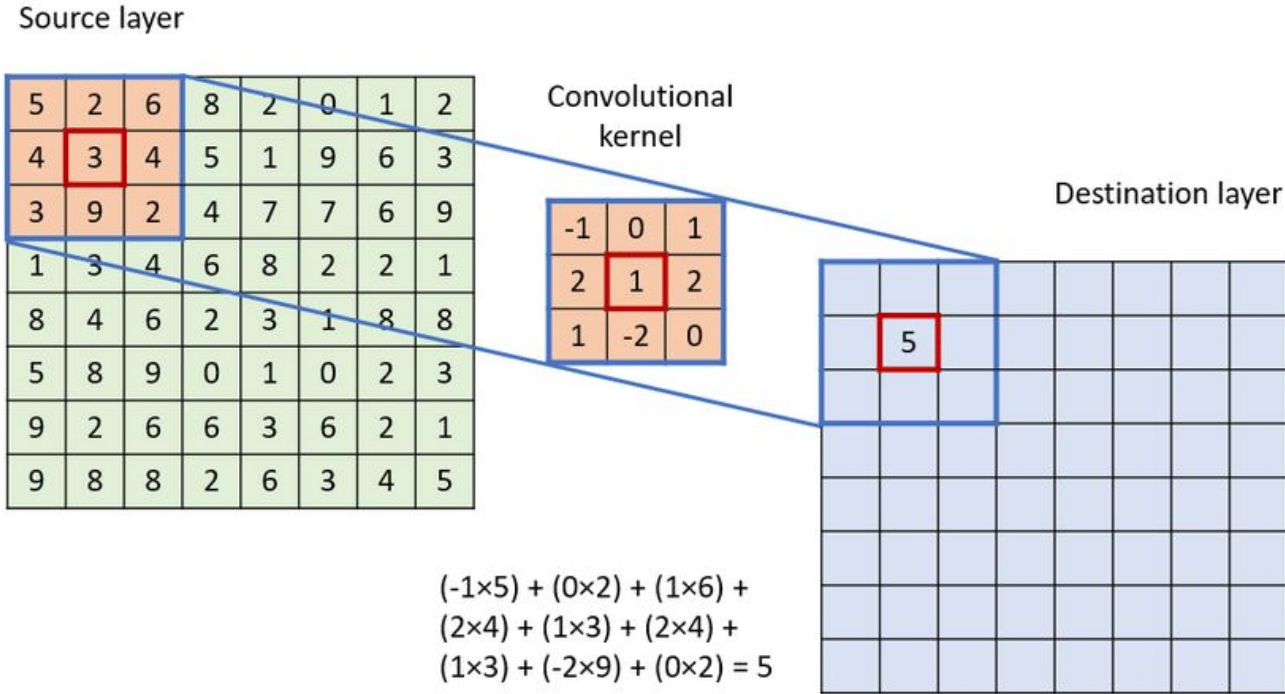
Integrated gradients



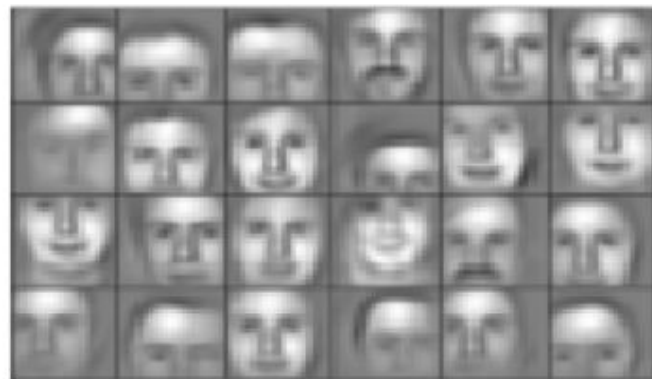
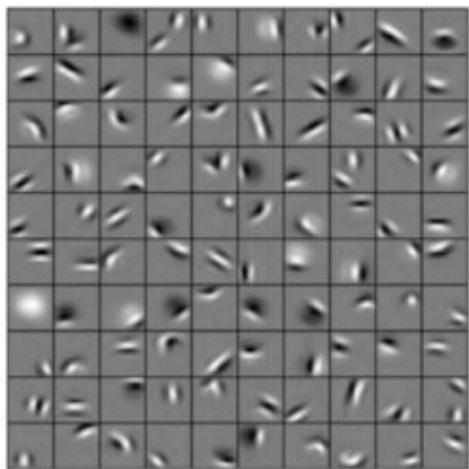
Gradients at image



# Convolutional layers learn interpretable concepts



# Convolutional layers learn interpretable concepts



# Grad-CAM

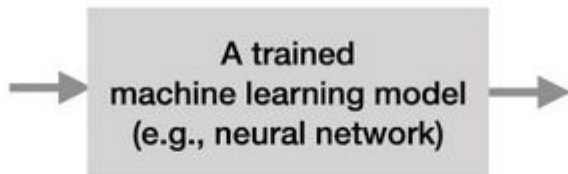
$$L_{Grad-CAM}^c \in \mathbb{R}^{u \times v} = \underbrace{ReLU}_{\text{Pick positive values}} \left( \sum_k \alpha_k^c A^k \right)$$

$$\alpha_k^c = \frac{1}{Z} \overbrace{\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\delta y^c}{\delta A_{ij}^k}}_{\text{gradients via backprop}}$$



# Attention Visualisation

# TCAV: Testing with Concept Activation Vectors (Kim et al, 2018)

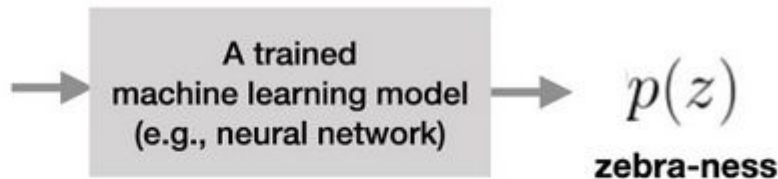


$p(z)$   
**zebra-ness**



How important was the striped concept  
to this zebra image classifier?

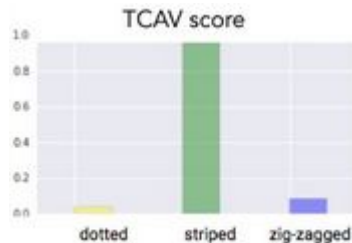
# TCAV: Testing with Concept Activation Vectors (Kim et al, 2018)



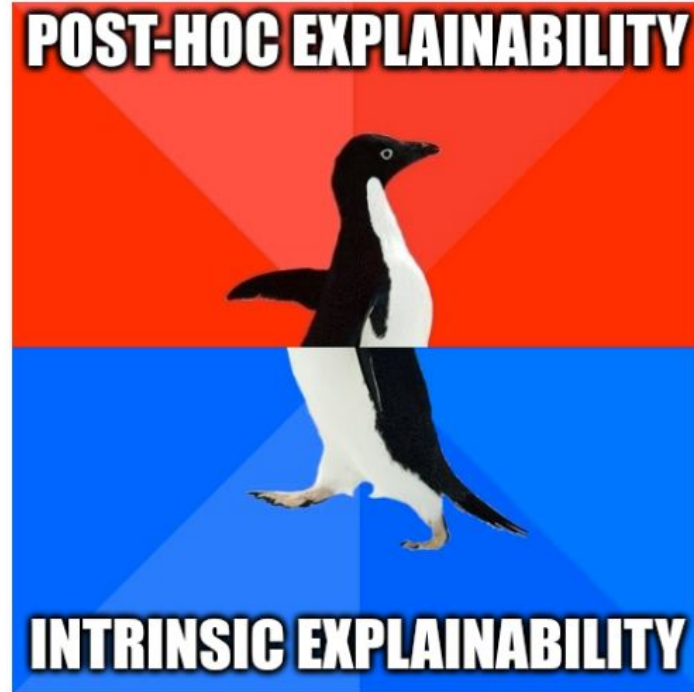
How important was the striped concept to this zebra image classifier?

$$\begin{matrix} \text{zebra-ness} & \rightarrow & \frac{\partial p(z)}{\partial v_C^l} & = & S_{C,k,l}(x) \\ \text{striped CAV} & \rightarrow & \partial v_C^l & & \end{matrix}$$

$$\text{TCAV}_{C,k,l} = \frac{|\{\mathbf{x} \in X_k : S_{C,k,l}(\mathbf{x}) > 0\}|}{|X_k|}$$



# Surrogate Modelling?!



- More Fidelity
- No “Double Trouble”

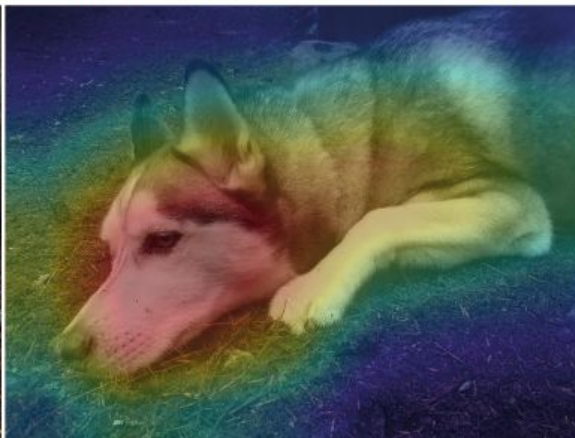
- Explainable models not necessarily explainable
- Loss in prediction accuracy
- Black box access

# Saliency maps are informative and elicit trust!

Test image



Evidence for animal being a Siberian husky



Explanations using  
attention maps

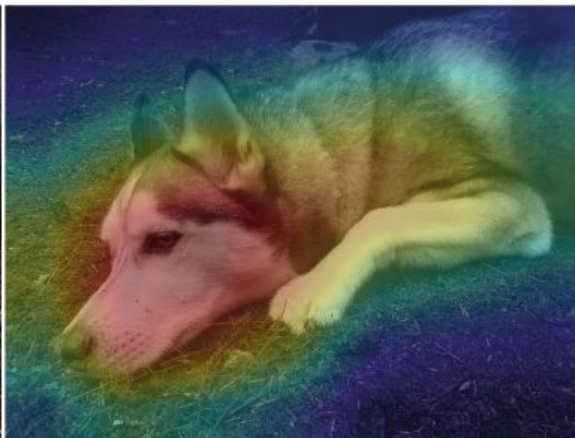
# Saliency maps are informative and elicit trust?

Explanations using  
attention maps

Test image



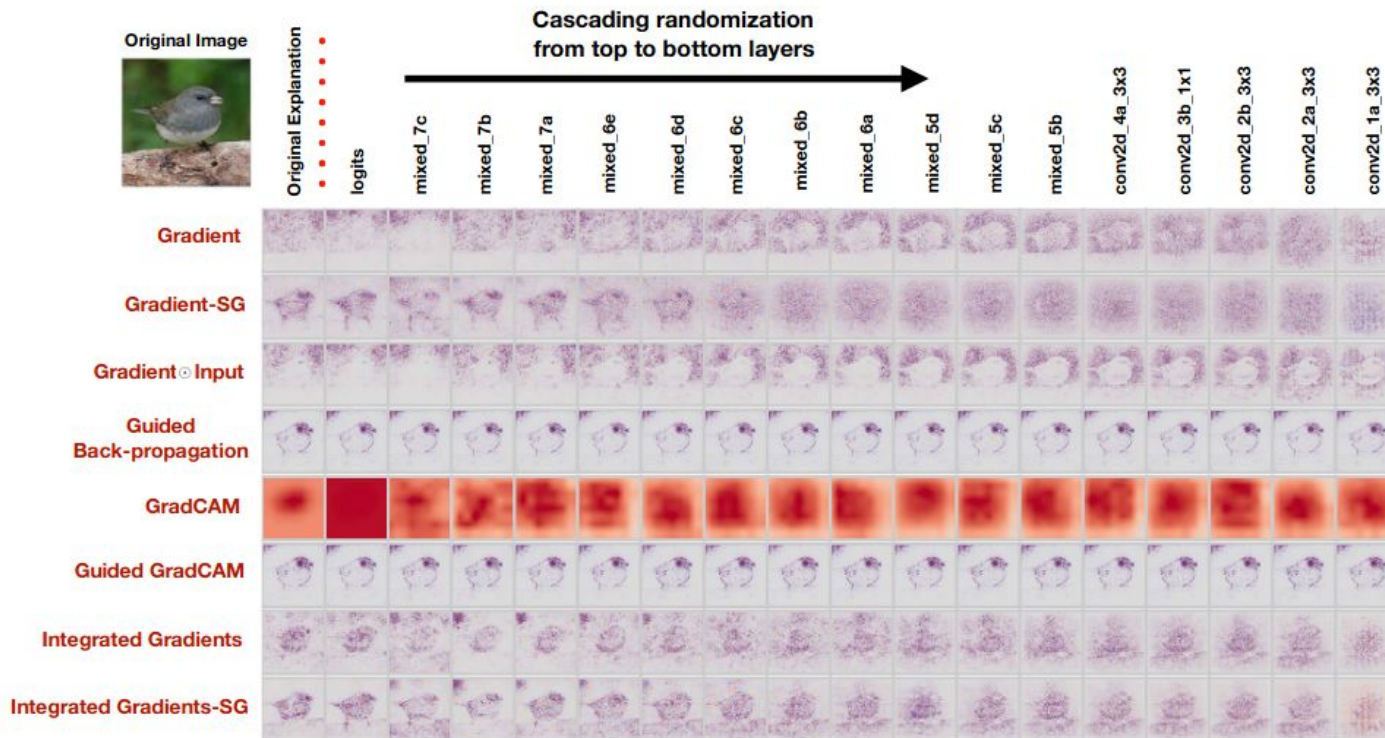
Evidence for animal being a Siberian husky



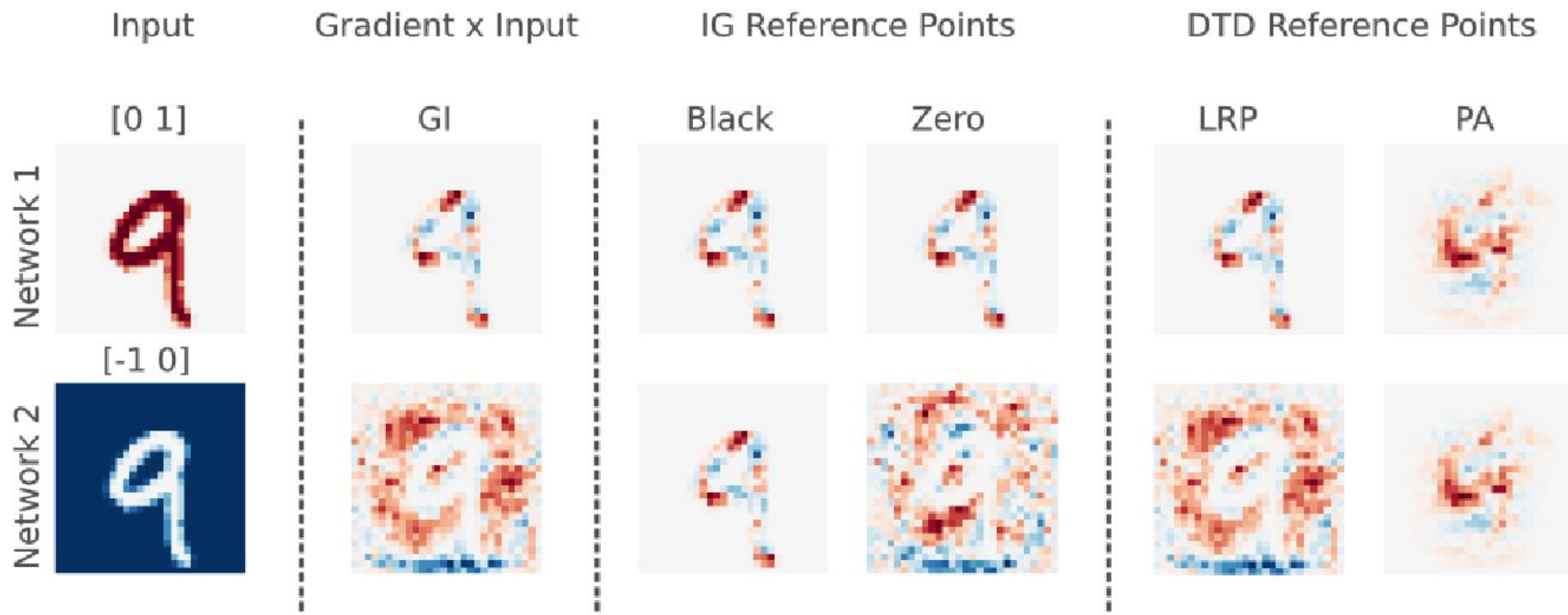
Evidence for animal being a transverse flute



# Sanity checks of saliency maps (Adebayo, 2018)



# Saliency methods are unreliable (Kindermans et al., 2019)





# Fooling Explainability Tools (Slack et al., 2020)

**evil** model that does not hire black applicants  
(*no one can know!*)

# Fooling Explainability Tools (Slack et al., 2020)

Imagine all **black applicants**  
live in **zip code 15235**



**evil** model that does not hire black  
applicants  
(*no one can know!*)

# Fooling Explainability Tools (Slack et al., 2020)

Imagine all **black applicants**  
live in **zip code 15235**

**evil** model that does not hire black  
applicants  
(*no one can know!*)

Strategy

-  Reject black applicant from Zip code 1523
-  Accept black applicant from any other Zip code

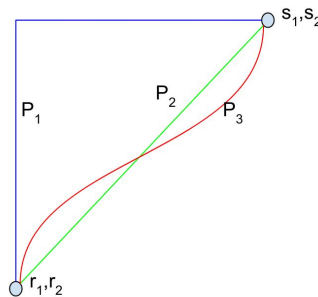
# Fooling Explainability Tools (Slack et al., 2020)

Imagine all **black applicants**  
live in **zip code 15235**

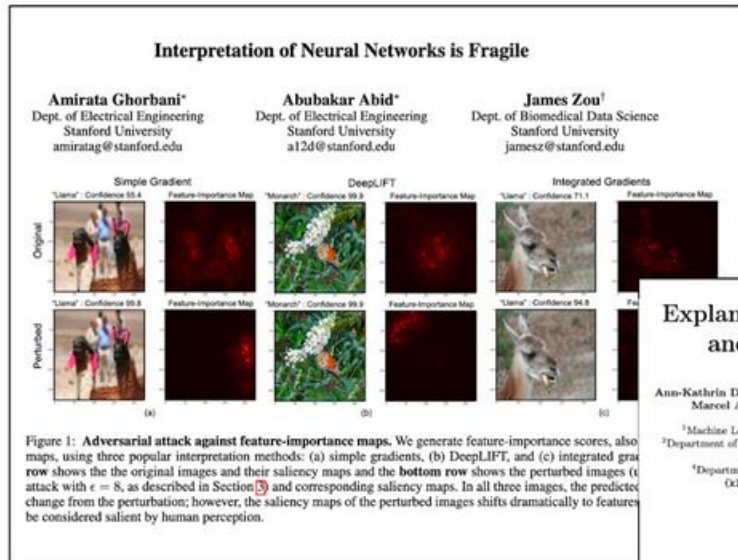
**evil** model that does not hire black  
applicants  
(*no one can know!*)

Strategy

- ❌ Reject black applicant from Zip code 1523
- ✅ Accept black applicant from any other Zip code



# Adversarial Attacks on Explanations



### Explanations can be manipulated and geometry is to blame

Ann-Kathrin Dombrowski<sup>1</sup>, Maximilian Alber<sup>1</sup>, Christopher J. Anders<sup>1</sup>, Marcel Ackermann<sup>2</sup>, Klaus-Robert Müller<sup>1,3,4</sup>, Pan Kessel<sup>1</sup>

<sup>1</sup>Machine Learning Group, EE & Computer Science Faculty, TU-Berlin

<sup>2</sup>Department of Video Coding & Analytics, Fraunhofer Heinrich-Hertz-Institute

<sup>3</sup>Max Planck Institute for Informatics

<sup>4</sup>Department of Brain and Cognitive Engineering, Korea University

{klaus-robert.mueller, pan.kessel}@tu-berlin.de



### THE (UN)RELIABILITY OF SALIENCY METHODS

Pieter-Jan Kindermans<sup>1</sup>, Sara Hooker<sup>1</sup>, Julius Adebayo  
Google Brain<sup>1</sup>  
{pikinder, shooker}@google.com

Maximilian Alber, Kristof T. Schütt, Sven Dähne  
TU-Berlin

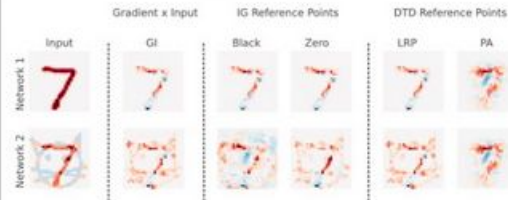
Dumitru Erhan, Been Kim  
Google Brain

"Cat"astrophic Attribution Failure

MNIST + Constant Shift



Attribution Methods



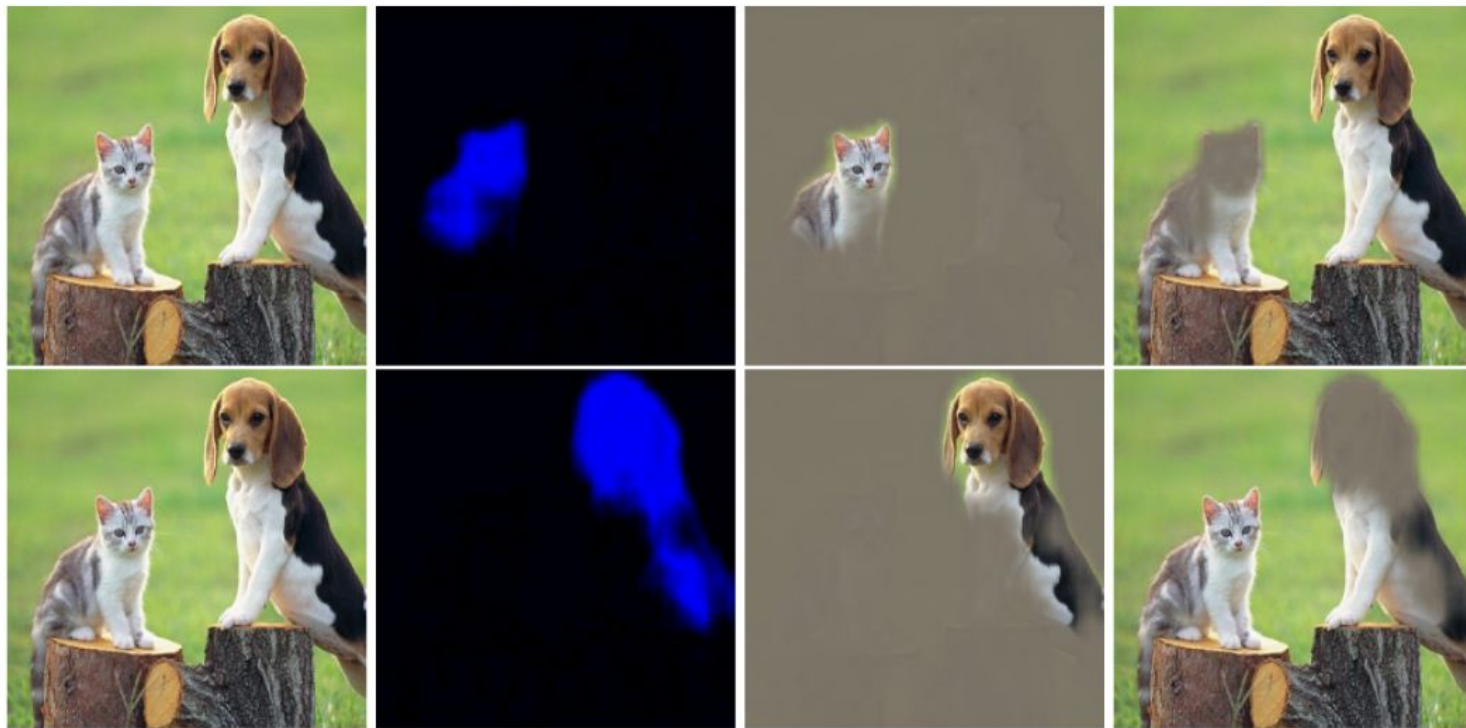
But if XAI doesn't work, what can we do?



But if XAI doesn't work, what can we do?



# 1) Evaluate Exclusion and Inclusion Criteria (i.e. Dabkowski and Gal, 2017 or Hooker, 2019)



(a) Input Image

(b) Generated saliency map

(c) Image multiplied by the mask

(d) Image multiplied by inverted mask



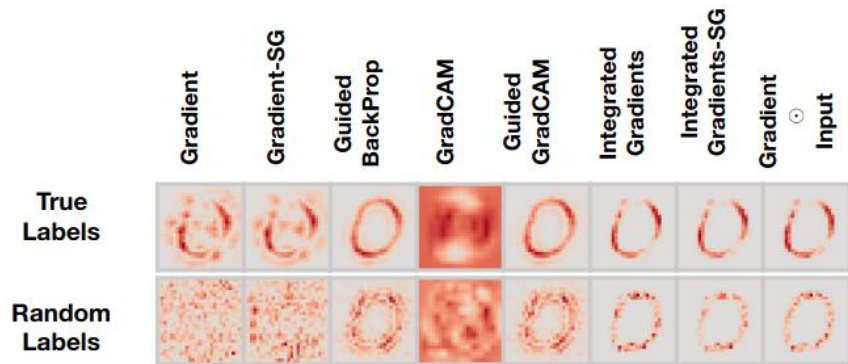
## 2) Handcraft sanity check data sets



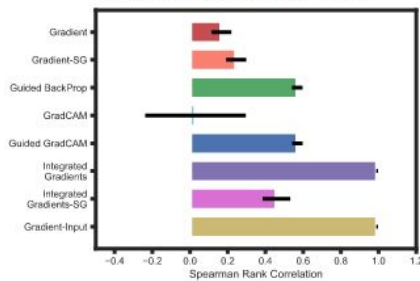
# 3) Look at classes that are not true, False positives, False negatives, ... (Adebayo, 2018)

CNN - MNIST

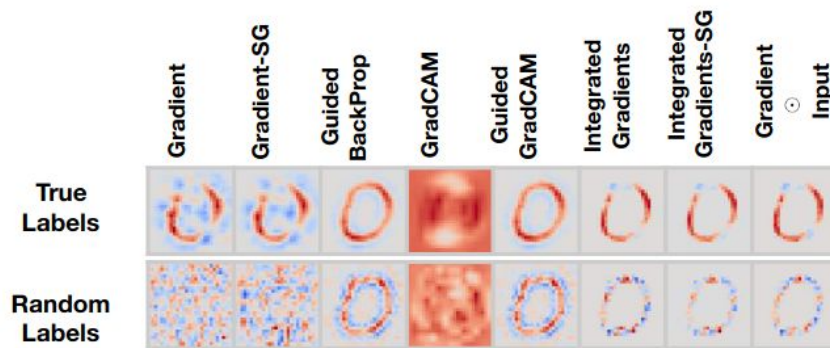
Absolute-Value Visualization



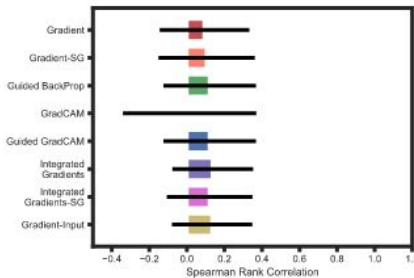
Rank Correlation - Abs



Diverging Visualization



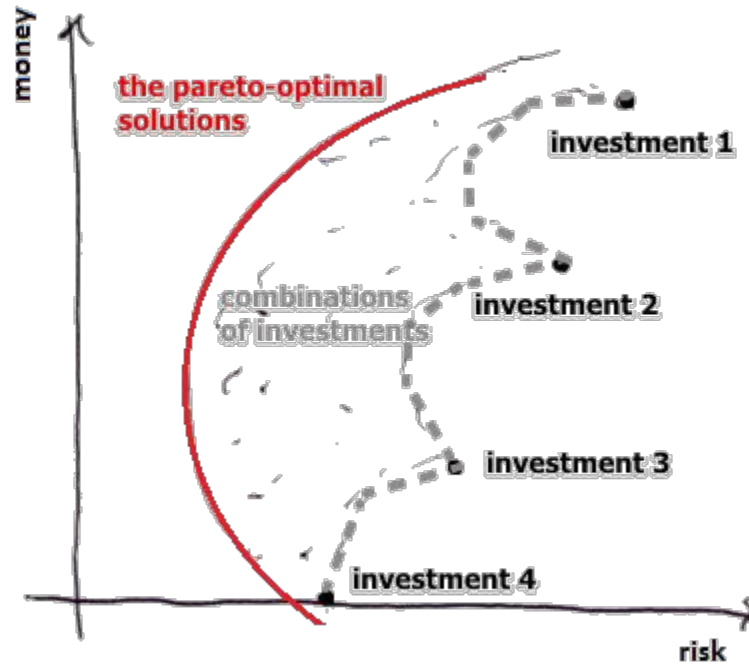
Rank Correlation - No Abs



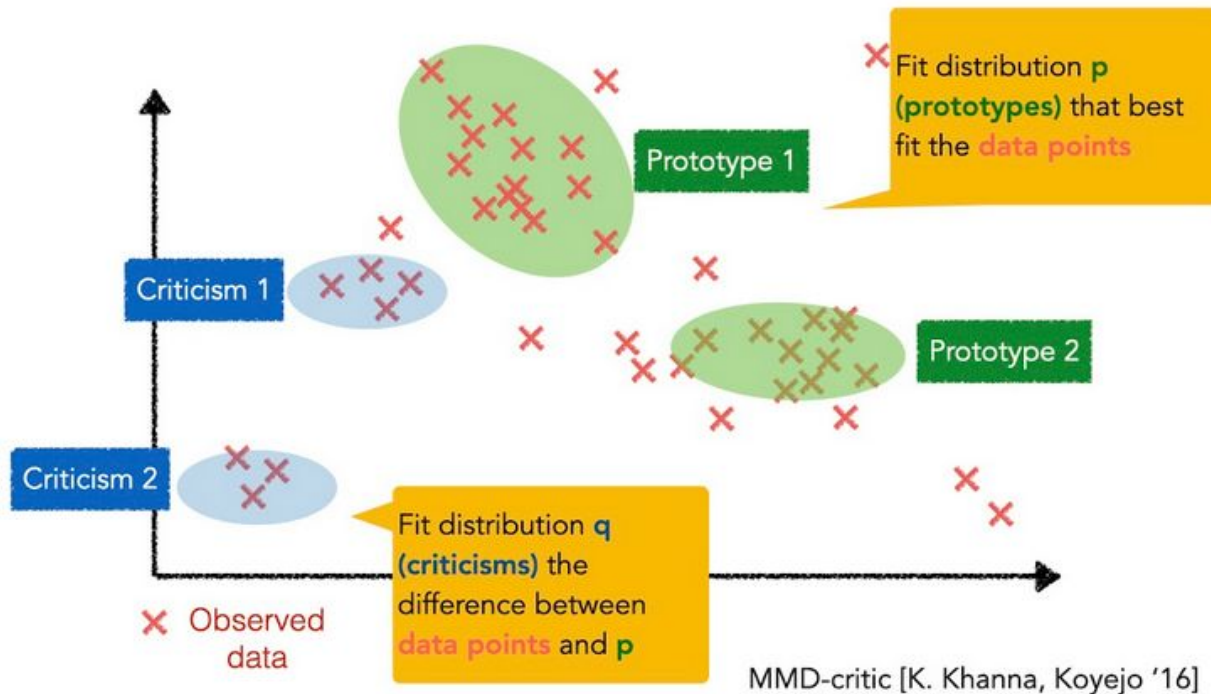
## 4) Human evaluations



## 5) Risk diversification



## 6) Exploratory data analysis




## 6) Exploratory data analysis



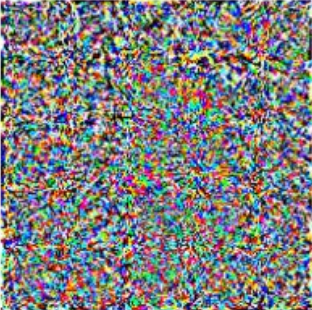
# Influence Functions (Koh and Liang, 2017)

A small perturbation to one **training** example:


Label: Fish



+  $\epsilon \cdot$








→



Label: Fish

Can change multiple **test** predictions:



Orig (confidence):	Dog (97%)	Dog (98%)	Dog (98%)	Dog (99%)	Dog (98%)
New (confidence):	Fish (97%)	Fish (93%)	Fish (87%)	Fish (63%)	Fish (52%)

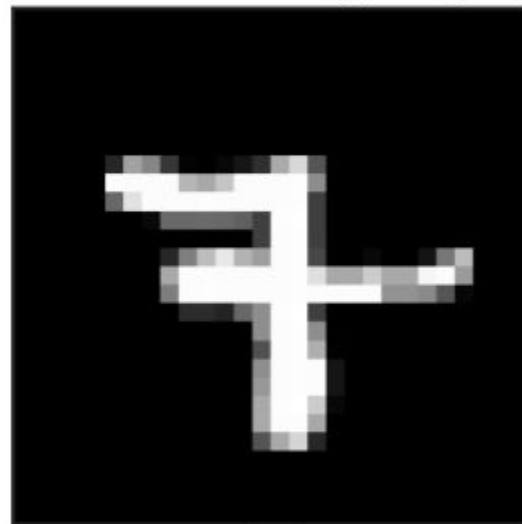
# Influence Functions (Koh and Liang, 2017)

Test image



Label: 7

Harmful training image



Label: 7



# Influence Functions (Cook and Weisberg, 1982)

What happens if I upweight observation  $z$  by  $(1+\epsilon)$

$$z = (x, y)$$

Find new optimal parameter

$$\hat{\theta}_{\epsilon, z} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta) + \epsilon L(z, \theta)$$

Influence of upweighting on parameters  $\theta$

$$\mathcal{I}_{\text{up, params}}(z) \stackrel{\text{def}}{=} \left. \frac{d\hat{\theta}_{\epsilon, z}}{d\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta})$$

# Influence Functions (Koh and Liang, 2017)

$$z = (x, y) \xrightarrow{\text{perturb one training point}} z_\delta \stackrel{\text{def}}{=} (x + \delta, y)$$

Find new optimal parameter

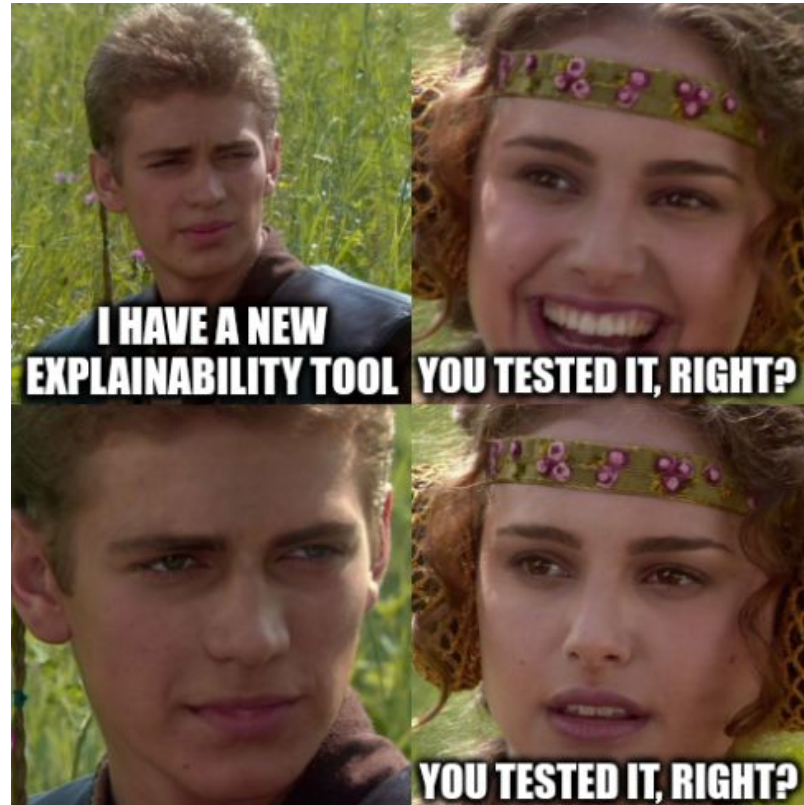
$$\hat{\theta}_{\epsilon, z_\delta, -z} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta) + \epsilon L(z_\delta, \theta) - \epsilon L(z, \theta)$$

Influence of perturbing  $z$  by  $\delta$

$$\begin{aligned} \left. \frac{d\hat{\theta}_{\epsilon, z_\delta, -z}}{d\epsilon} \right|_{\epsilon=0} &= \mathcal{I}_{\text{up, params}}(z_\delta) - \mathcal{I}_{\text{up, params}}(z) \\ &= -H_{\hat{\theta}}^{-1}(\nabla_{\theta} L(z_\delta, \hat{\theta}) - \nabla_{\theta} L(z, \hat{\theta})) \end{aligned}$$

Holds for arbitrary  $\delta$

# Take home message



# References

- Slack, Dylan, et al. "Fooling lime and shap: Adversarial attacks on post hoc explanation methods." *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020.
- Kindermans, Pieter-Jan, et al. "The (un) reliability of saliency methods." *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, Cham, 2019. 267-280.
- Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).
- Hooker, Sara, et al. "A benchmark for interpretability methods in deep neural networks." *Advances in neural information processing systems* 32 (2019).
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.
- Rudin, Cynthia. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." *Nature Machine Intelligence* 1.5 (2019): 206-215.
- Adebayo, Julius, et al. "Sanity checks for saliency maps." *Advances in neural information processing systems* 31 (2018).
- Lapuschkin, Sebastian, et al. "Unmasking Clever Hans predictors and assessing what machines really learn." *Nature communications* 10.1 (2019): 1-8.
- Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *Advances in neural information processing systems* 32 (2019).

# References

Cook, R. Dennis, and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

Ustun, Berk, and Cynthia Rudin. "Supersparse linear integer models for optimized medical scoring systems." *Machine Learning* 102.3 (2016): 349-391.

Montavon, Grégoire, et al. "Layer-wise relevance propagation: an overview." *Explainable AI: interpreting, explaining and visualizing deep learning* (2019): 193-209.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* 10(7), e0130140 (2015)

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." *International conference on machine learning*. PMLR, 2017.

Dabkowski, Piotr, and Yarin Gal. "Real time image saliency for black box classifiers." *Advances in neural information processing systems* 30 (2017).

Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." *International conference on machine learning*. PMLR, 2017.

Kim, Been, et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." *International conference on machine learning*. PMLR, 2018.

If you have any questions, please let me know



sghalebikesabi

[sahra.ghalebikesabi@univ.ox.ac.uk](mailto:sahra.ghalebikesabi@univ.ox.ac.uk)

<https://sghalebikesabi.github.io>