Understanding models via visualizations and attribution

ANDREA VEDALDI

TUTORIAL, ICCV 2019

(SEVERAL SLIDES BY RUTH FONG)

facebook AI Research

UNIVERSITY OF OXFORD

Kind of explanations

Analysis

Given an off-the-shelf networks, explain what it knowns, how it works, and how it learns

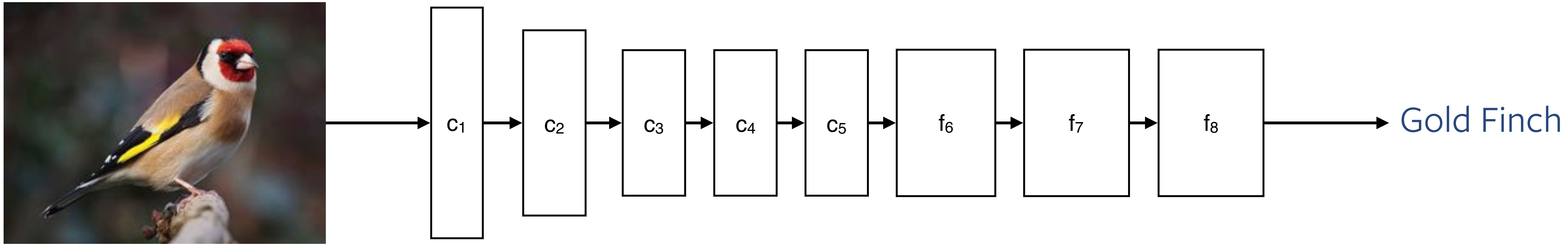
Win an argument

The network explains its decision to a user, with the goal of **convincing** her

Communicating a skill

Explain to a human or machine how to solve a certain class of problems, in general

Analysing deep neural networks



What does a net **do**?

- What concepts can it recognise?
- Spurious correlations?
- Limitations?

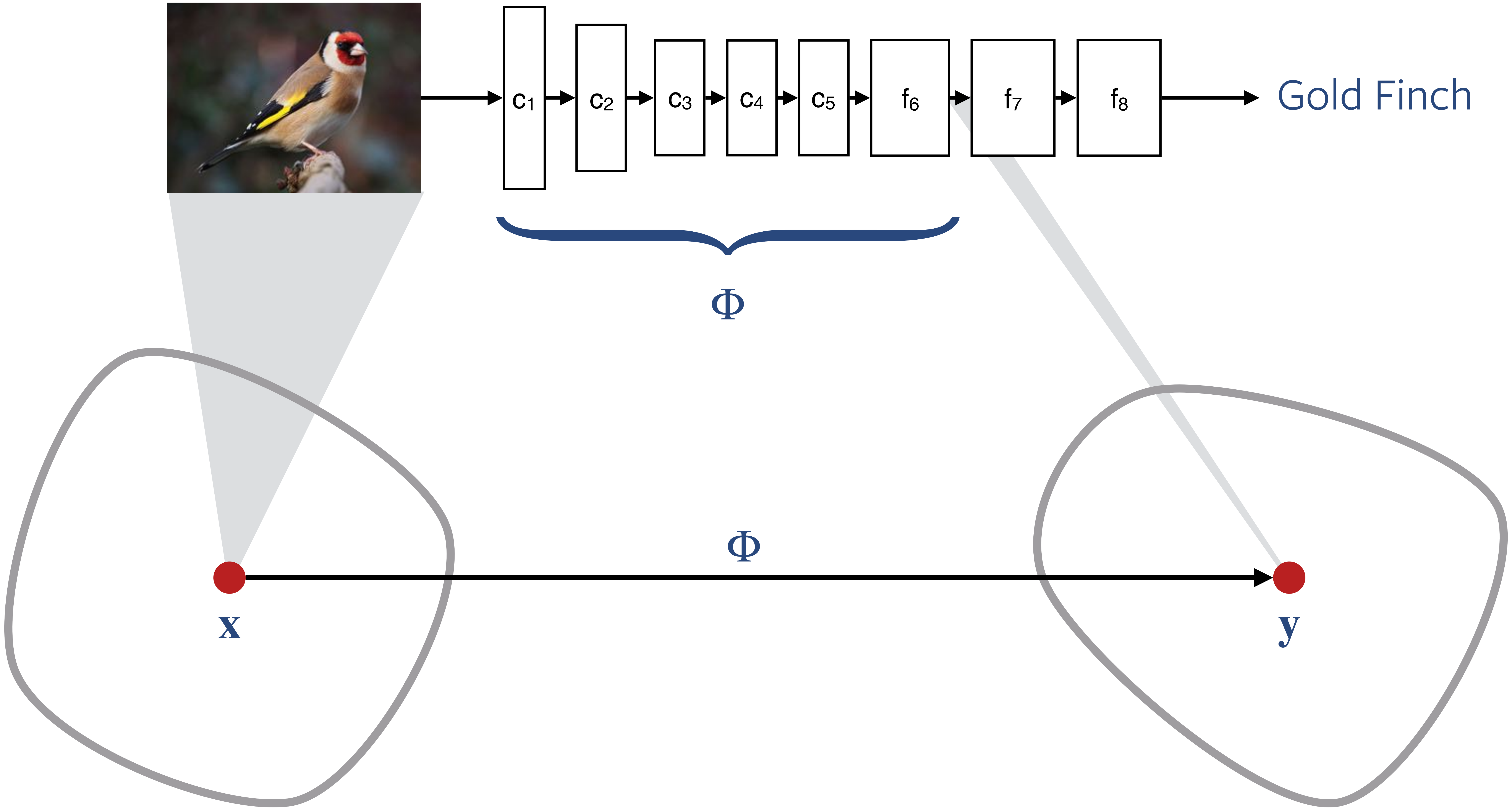
How does it **do** it?

- Template matching?
- Compositionality?
- Spatial reasoning?

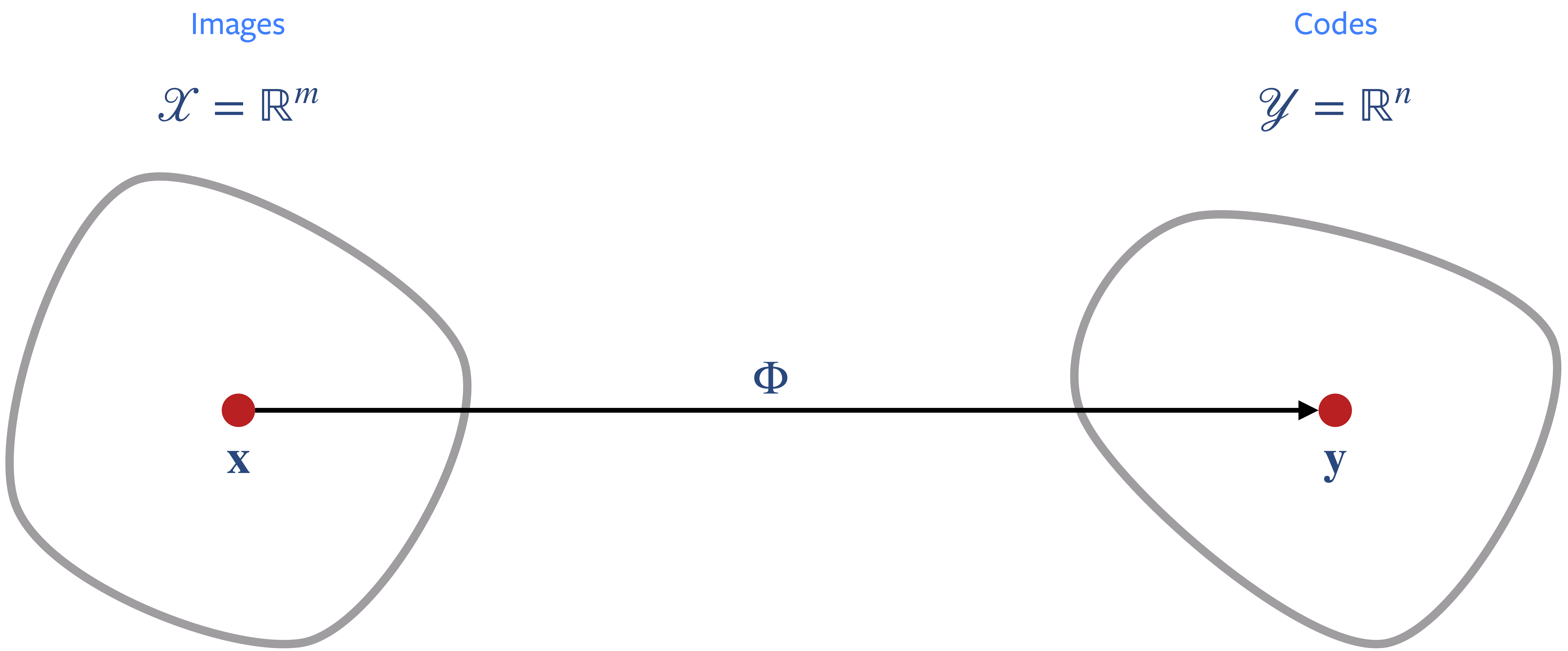
How does it **learn** it?

- Generalization?
- Optimisation?

Deep networks as encoders



Deep networks as encoders



Generating iconic examples

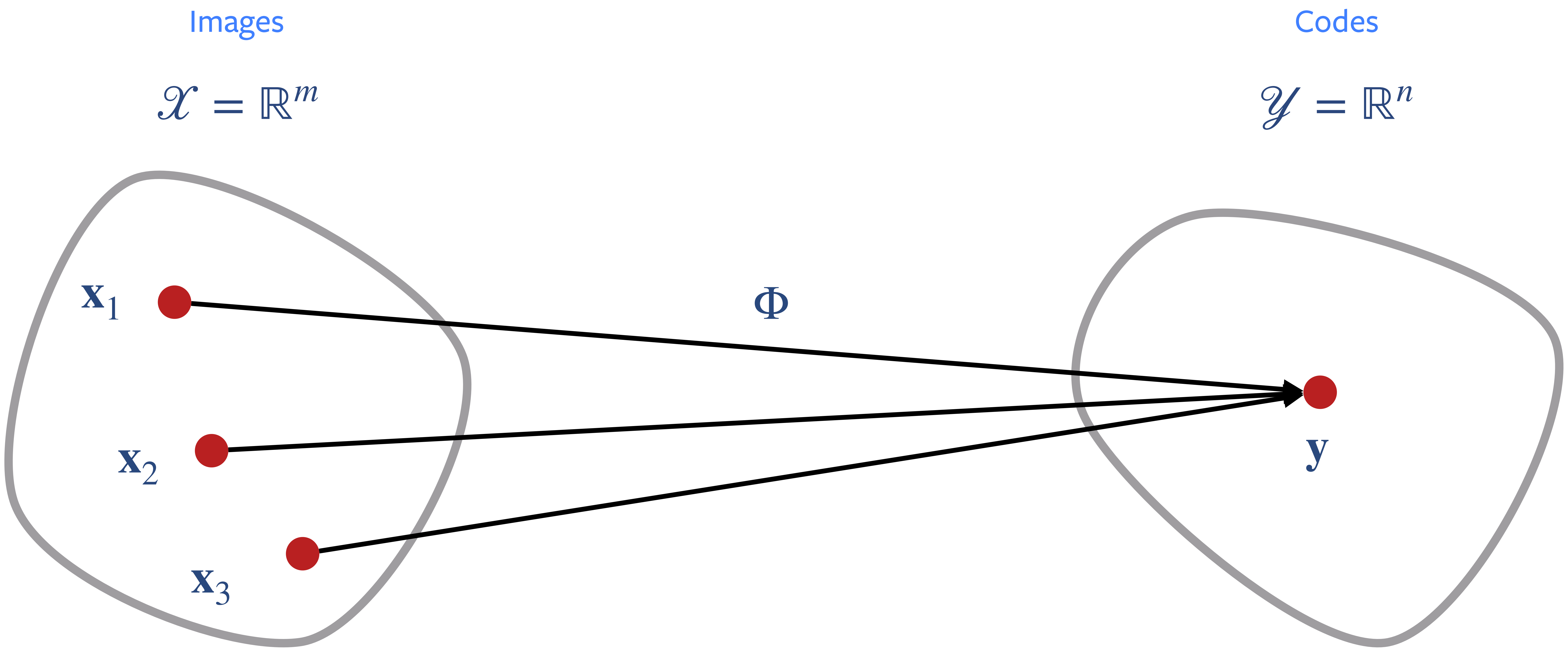
Attribution

Generating iconic examples

Attribution

How much information about $\mathbf{x}$ does $\mathbf{y}$ contain?

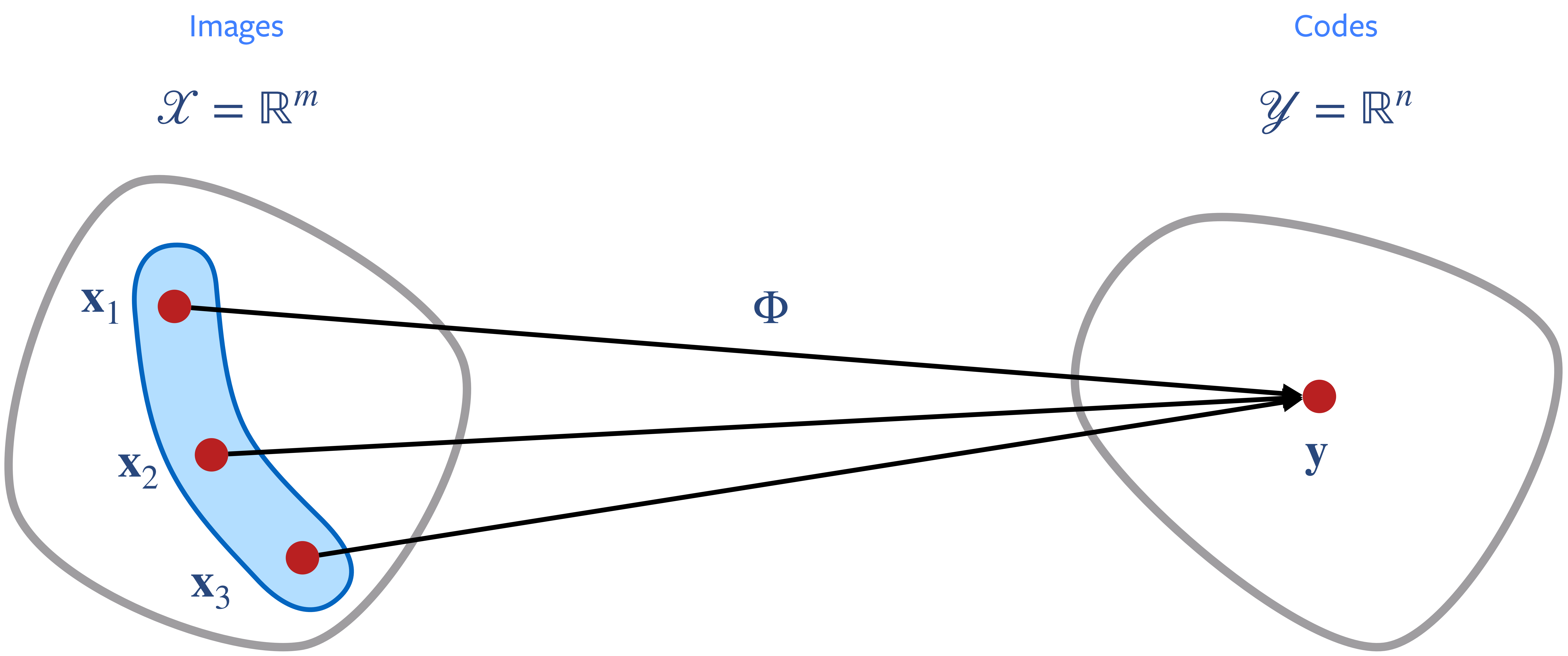
Multiple images map to the same code



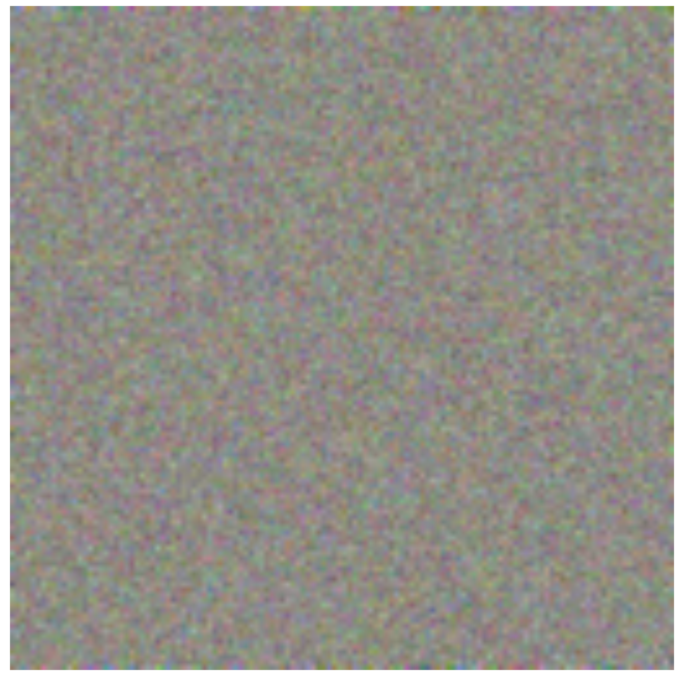
Pre-image

Reconstructions form an **equivalence class** of images, called a pre-image

All pre-images hat are indistinguishable for the network



Finding pre-images via optimisation



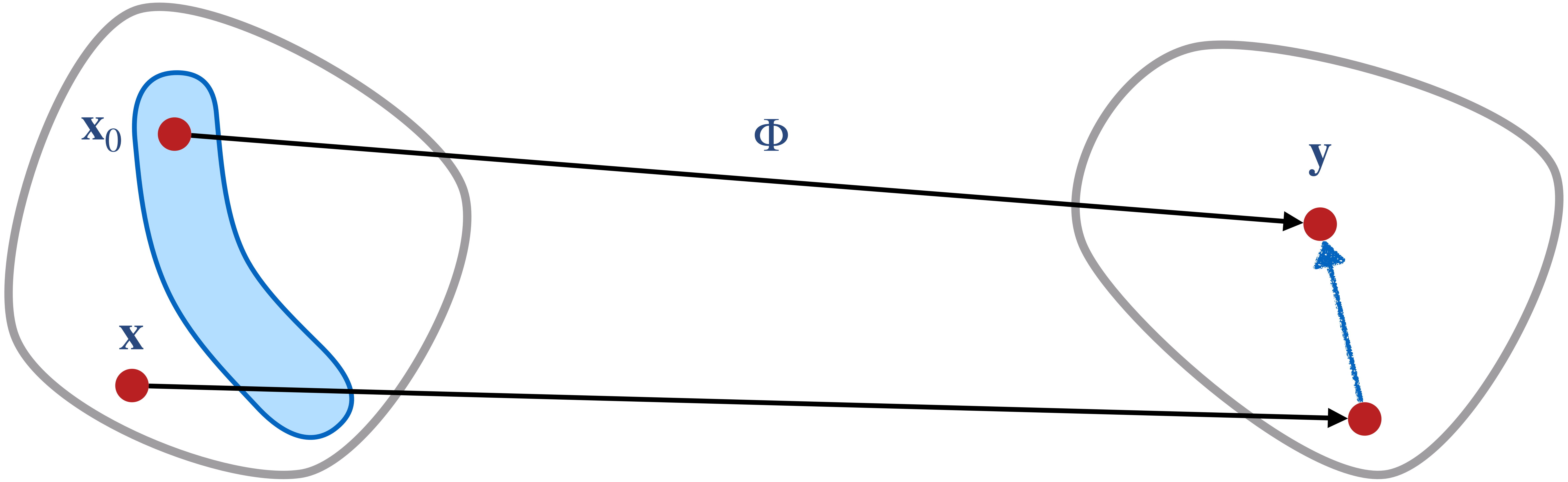
Images

$\mathscr{X} = \mathbb{R}^m$

$$\min_{\mathbf{x}} \|\Phi(\mathbf{x}) - \Phi(\mathbf{x}_0)\|^2$$

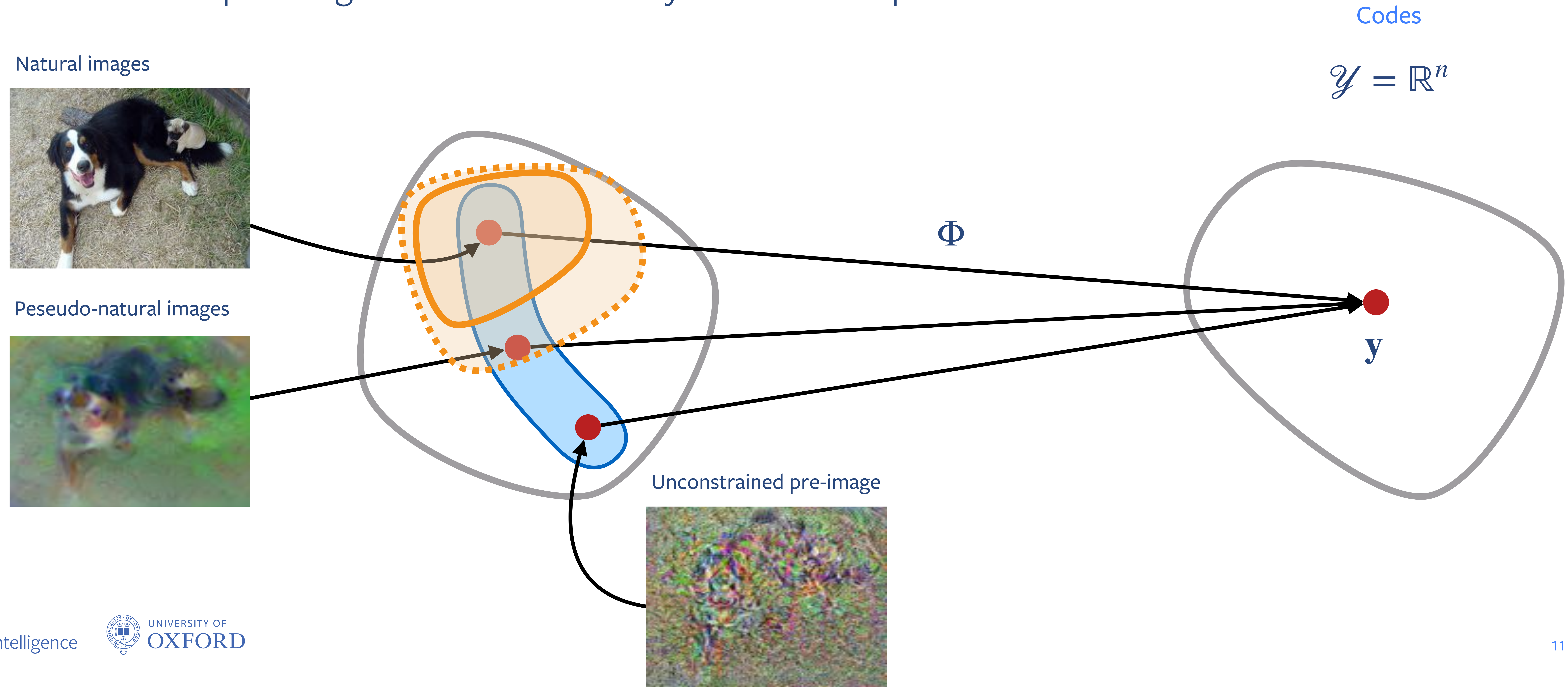
Codes

$\mathscr{Y} = \mathbb{R}^n$



Natural pre-images

We are interested in pre-images that can realistically be network inputs



Pseudo-natural pre-images

Regularised energy

$$\min_{\mathbf{x}} \|\Phi(\mathbf{x}) - \Phi(\mathbf{x}_0)\|^2 + \mathscr{R}(\mathbf{x})$$

For example TV-norm

Understanding deep image representations by inverting them
Mahendran Vedaldi, CVPR, 2015

Constrained optimisation

$$\min_{\mathbf{x} \in \mathscr{X}_{pn}} \|\Phi(\mathbf{x}) - \Phi(\mathbf{x}_0)\|^2$$

For example Deep Image Prior

Deep image prior
Ulyanov Vedaldi Lempistky, CVPR, 2018

Posterior probability

$$p(\mathbf{x} \,|\, \mathbf{y}) \sim \delta(\Phi(\mathbf{x}) - \mathbf{y}) \cdot p(\mathbf{x})$$

For example Plug & Play gen. nets

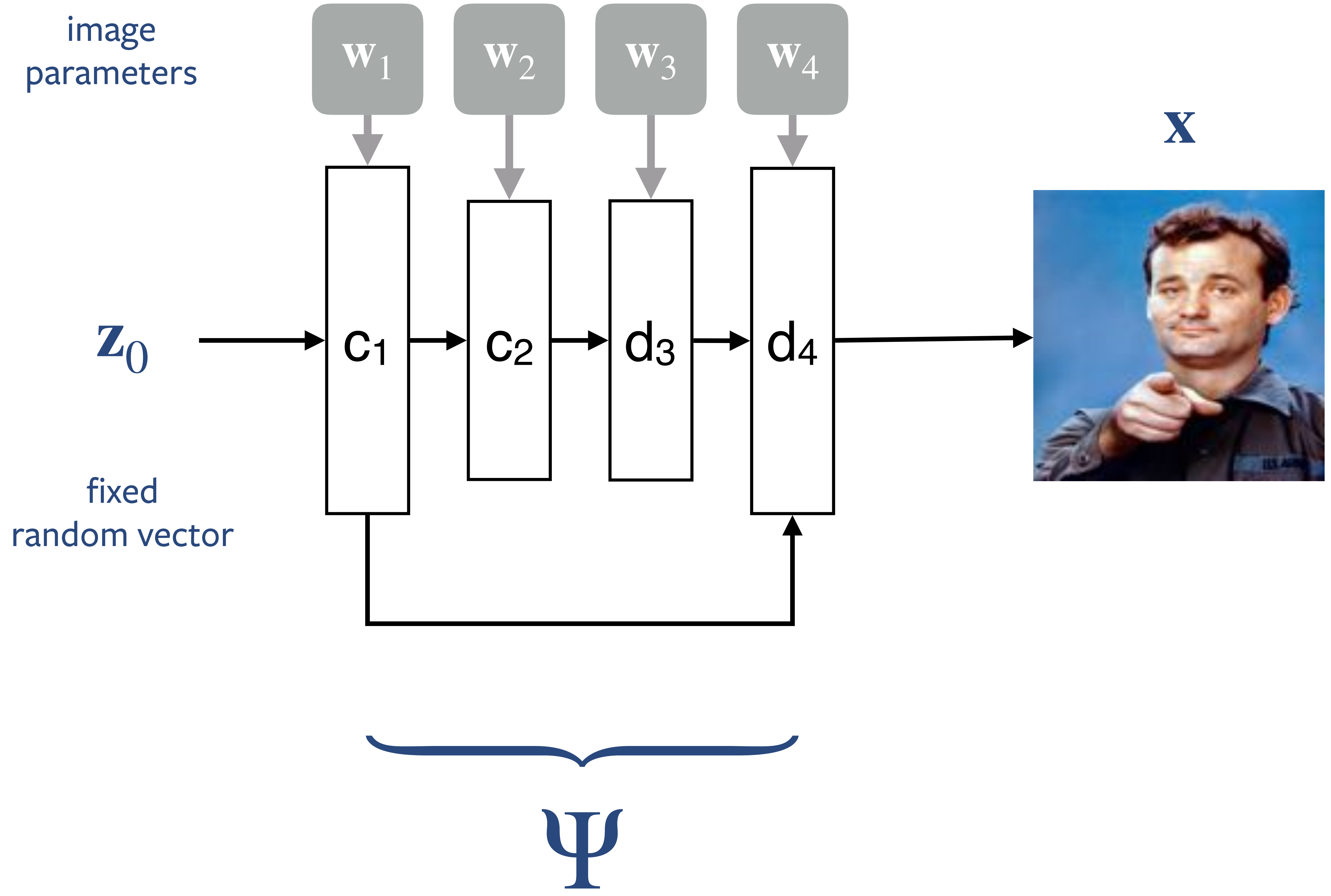
Plug & play generative networks: Conditional iterative generation of images in latent space
Nguyen, Yosinksi, Bengio, Dosovitskiy, Clune, CVPR, 2017

Generator nets as image parameterisations

Consider a **generator network** Ψ with a fixed input $\mathbf{z}_0$

The network parameters $\mathbf{w}$ can be thought as **image parameters**

$$\mathbf{w} \longmapsto \mathbf{x} = \Psi(\mathbf{z}_0; \mathbf{w})$$



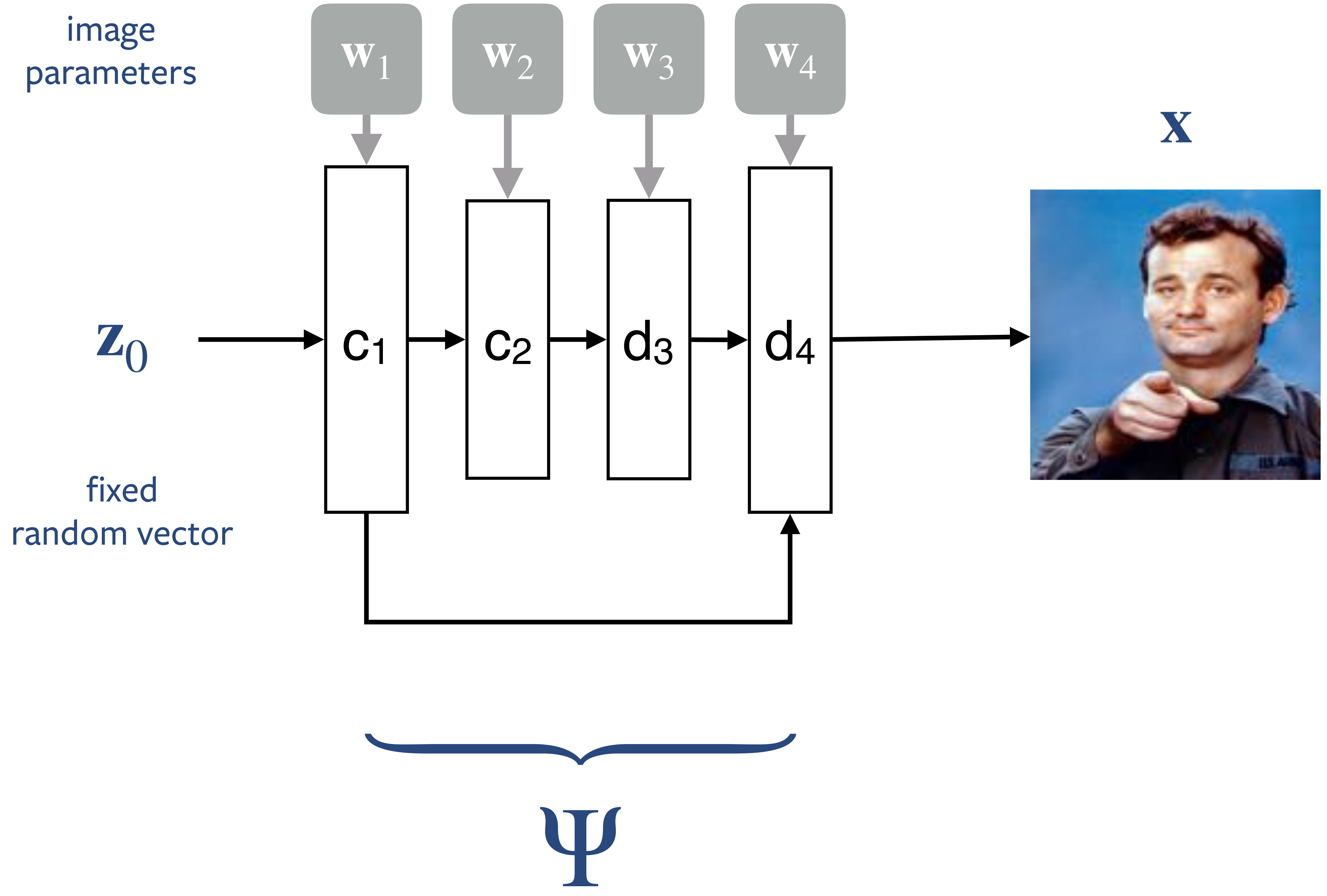
Fit a network to a single example

Start **randomly-initialised** network

Given an image $\mathbf{x}$, its parameter $\mathbf{w}$ is recovered by solving the optimisation problem

$$\min_{\mathbf{w}} \|\mathbf{x} - \Psi(\mathbf{z}_0; \mathbf{w})\|^2$$

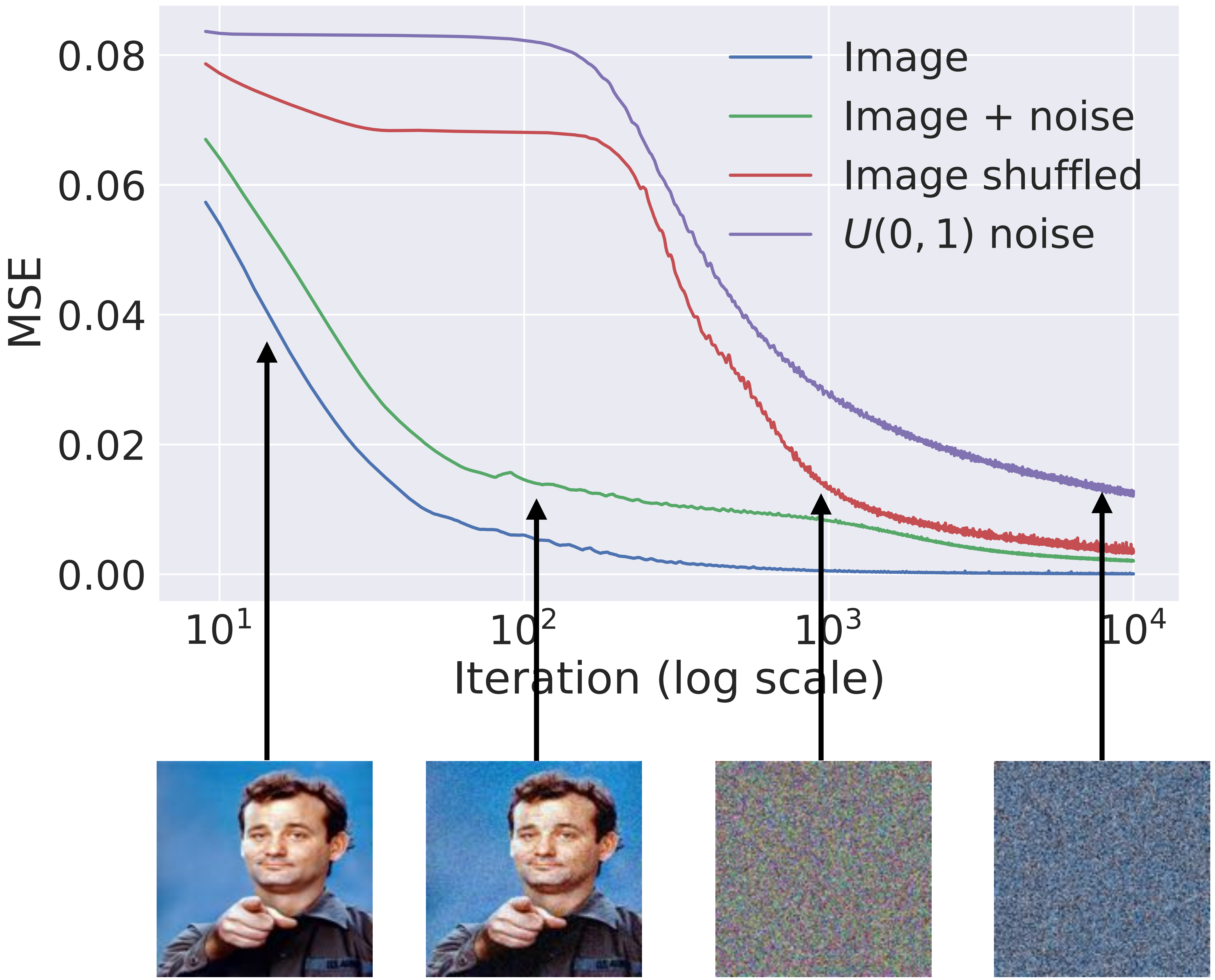
This is similar to learning the network from a single image



Deep image prior

For most generator networks fitting naturally-looking images is easier/ faster than fitting others

Deep image prior
Ulyanov Vedaldi Lempistky, CVPR, 2018



Deep image prior: inpainting

For **inpainting** we only reconstruct the visible pixels, implicitly infer the others

$$\min_{\mathbf{w}} \|\mathbf{m} \odot (\mathbf{x} - \Phi(\mathbf{w}))\|^2$$

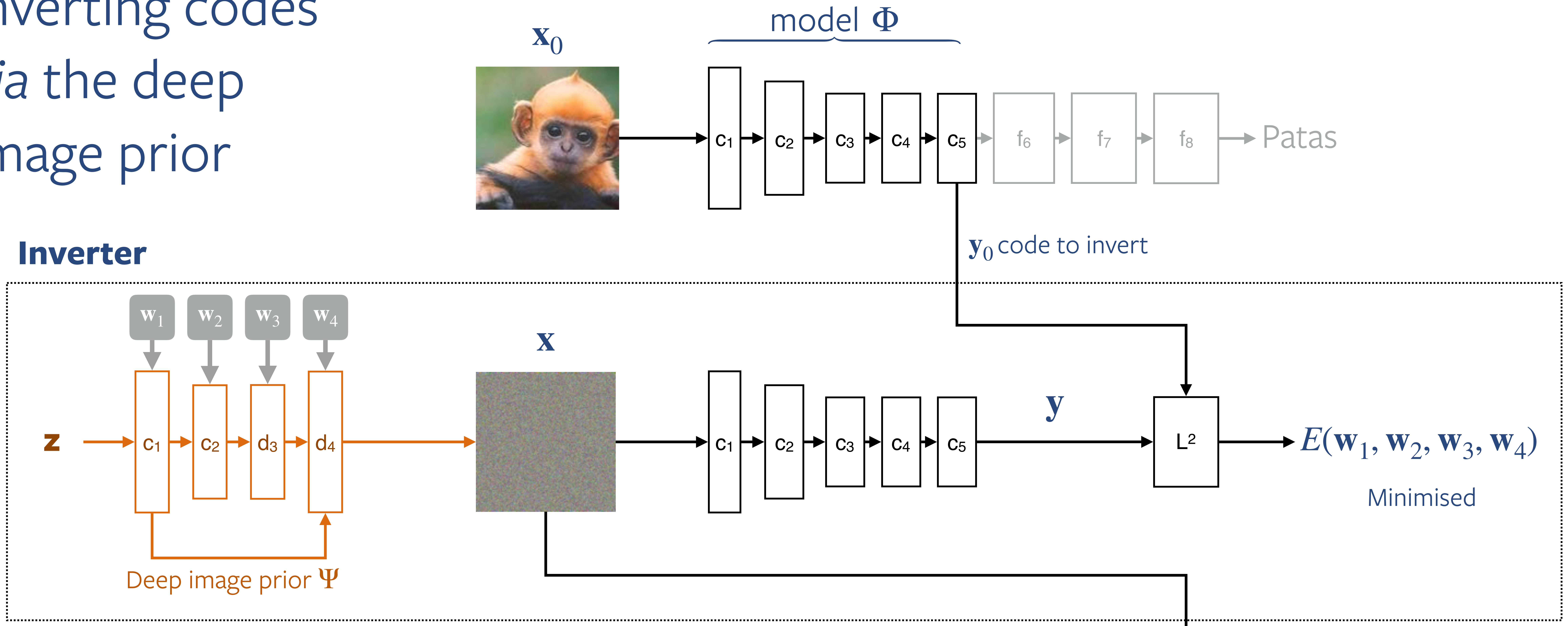
Conv. coding
Papyan et al. 2017

Deep Image Prior

facebook
Artificial Intelligence

UNIVERSITY OF
OXFORD

Inverting codes *via* the deep image prior



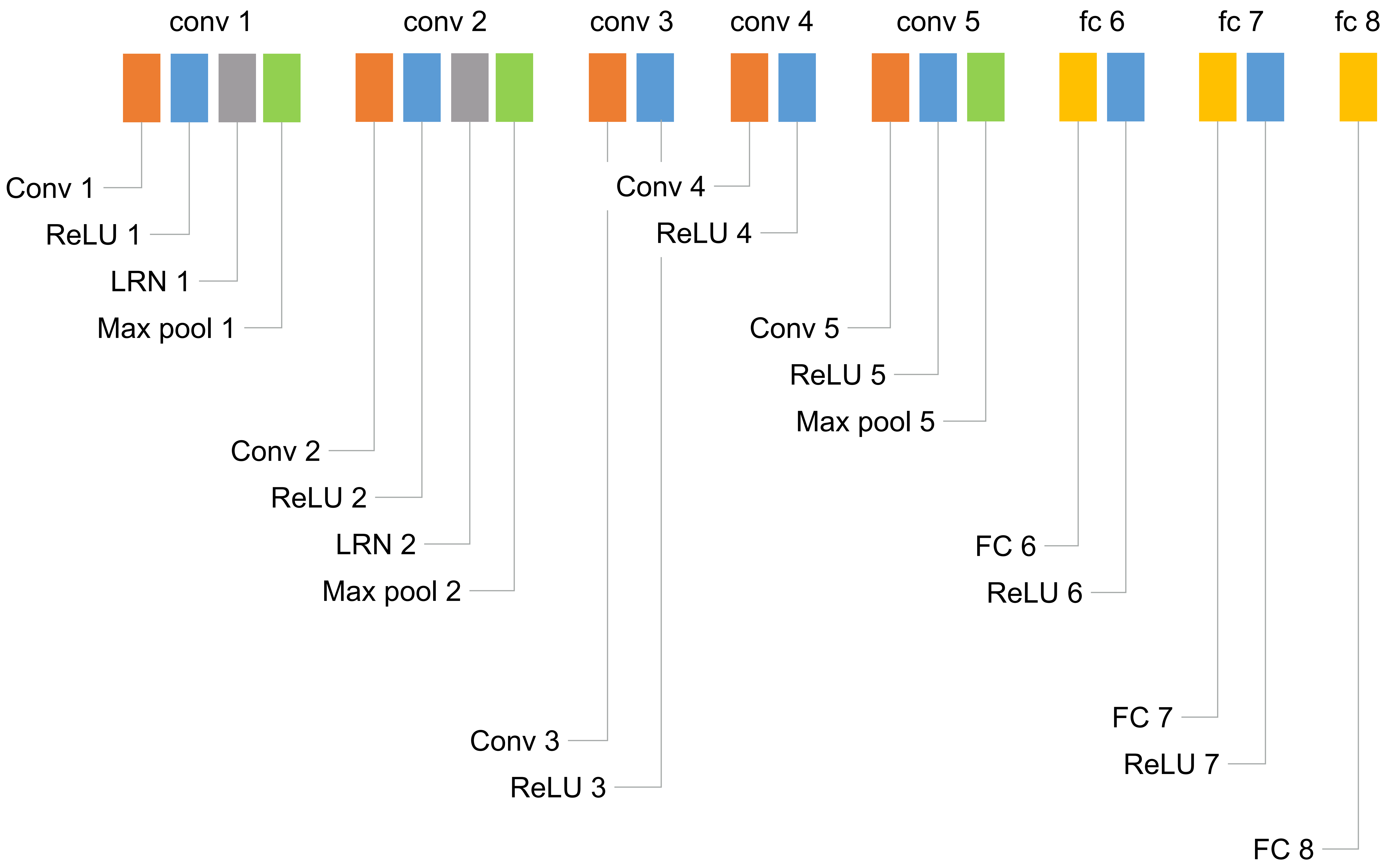
The inverter is only given the **code**;
it is **not** learned from data in any way

$$\min_{\mathbf{w}} \|\Phi(\Psi(\mathbf{w})) - \Phi(\mathbf{x}_0)\|^2$$

Inversion result

Inverting AlexNet

[Krizhevsky et al. 2012]



Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

Inverting AlexNet

facebook Artificial Intelligence

UNIVERSITY OF OXFORD

Inverting AlexNet

Inverting AlexNet

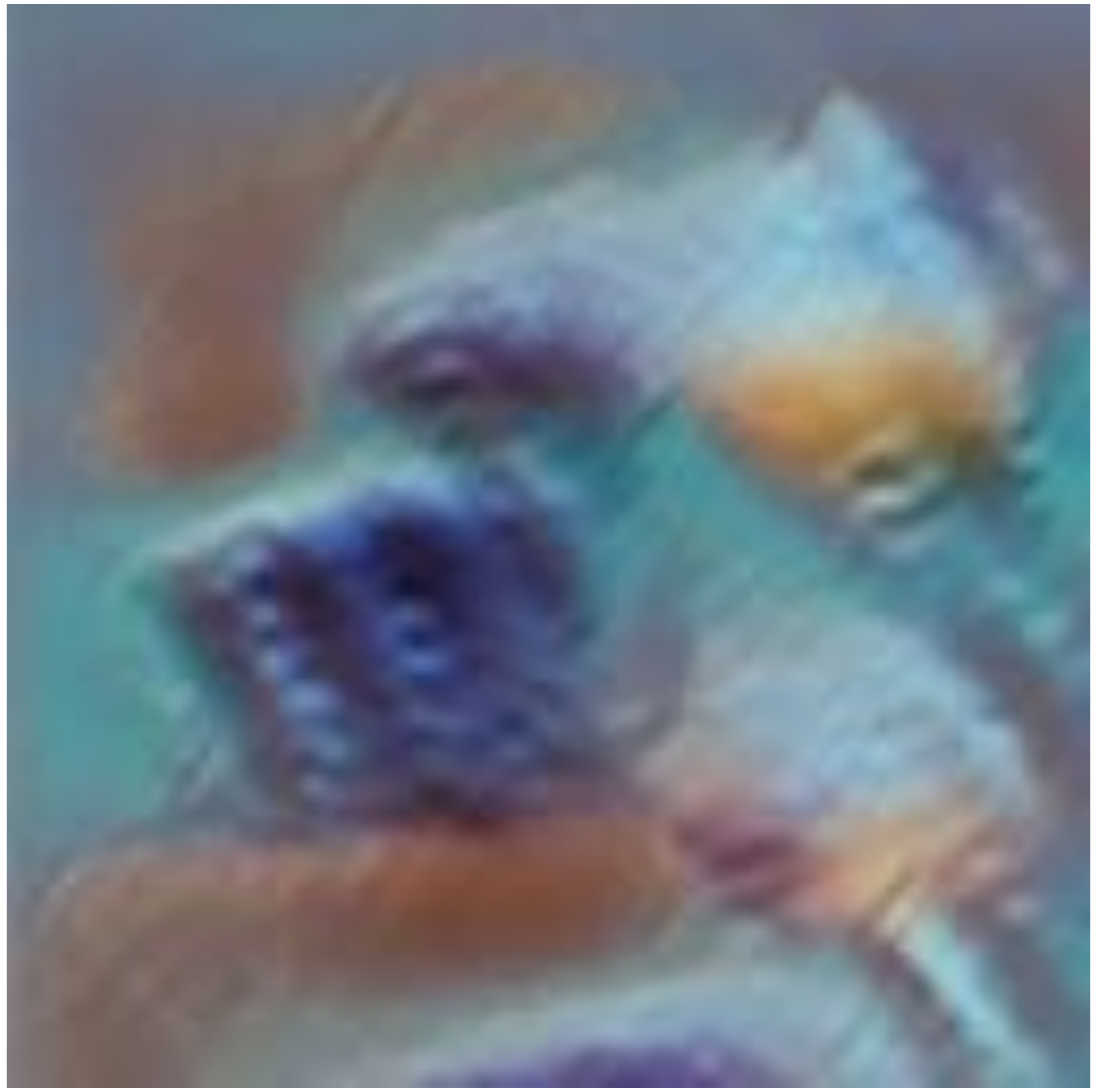
Inverting AlexNet

Inverting AlexNet

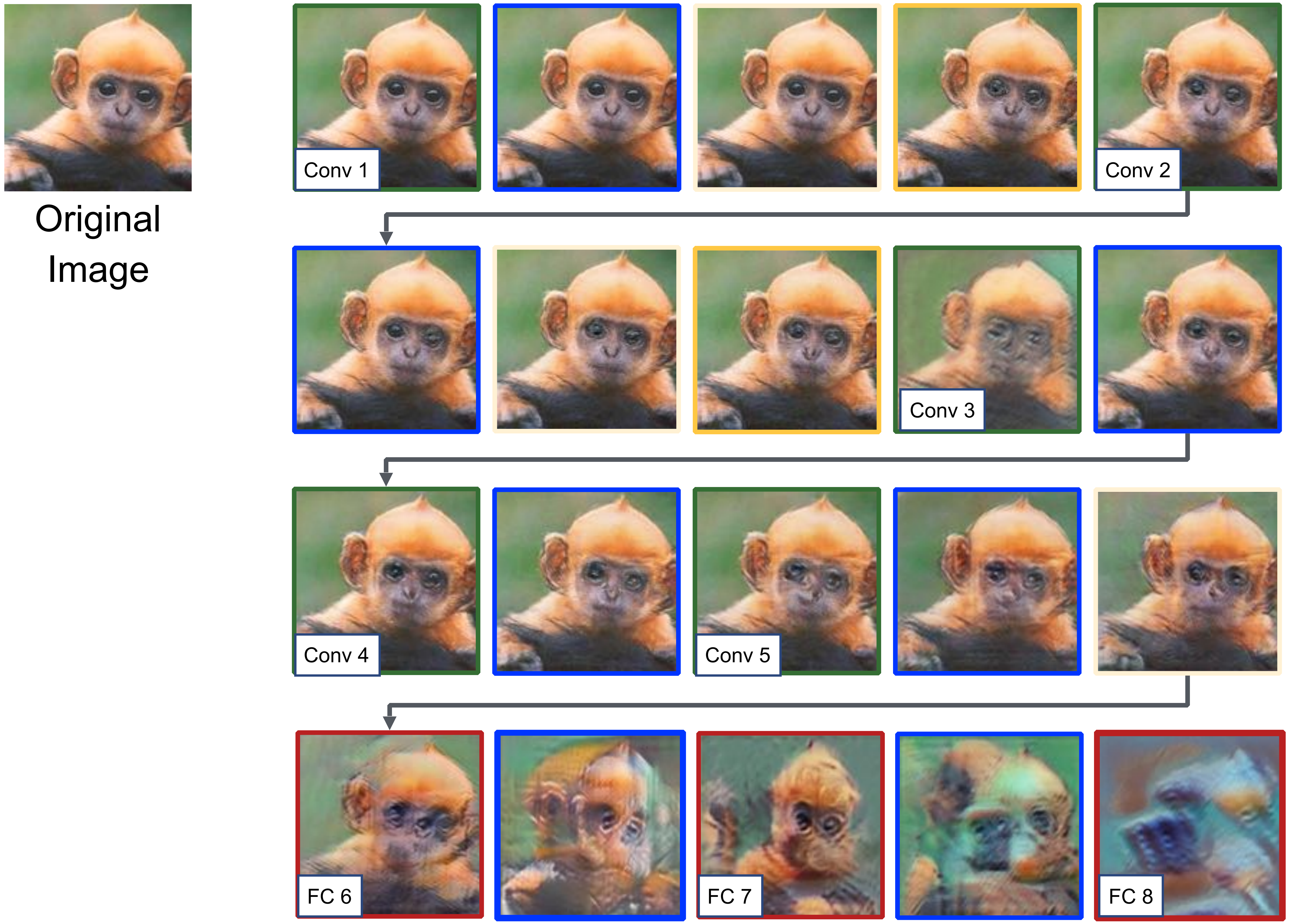
Inverting AlexNet

Inverting AlexNet

Inverting AlexNet



Inverting AlexNet



Is the code semantic or visual?

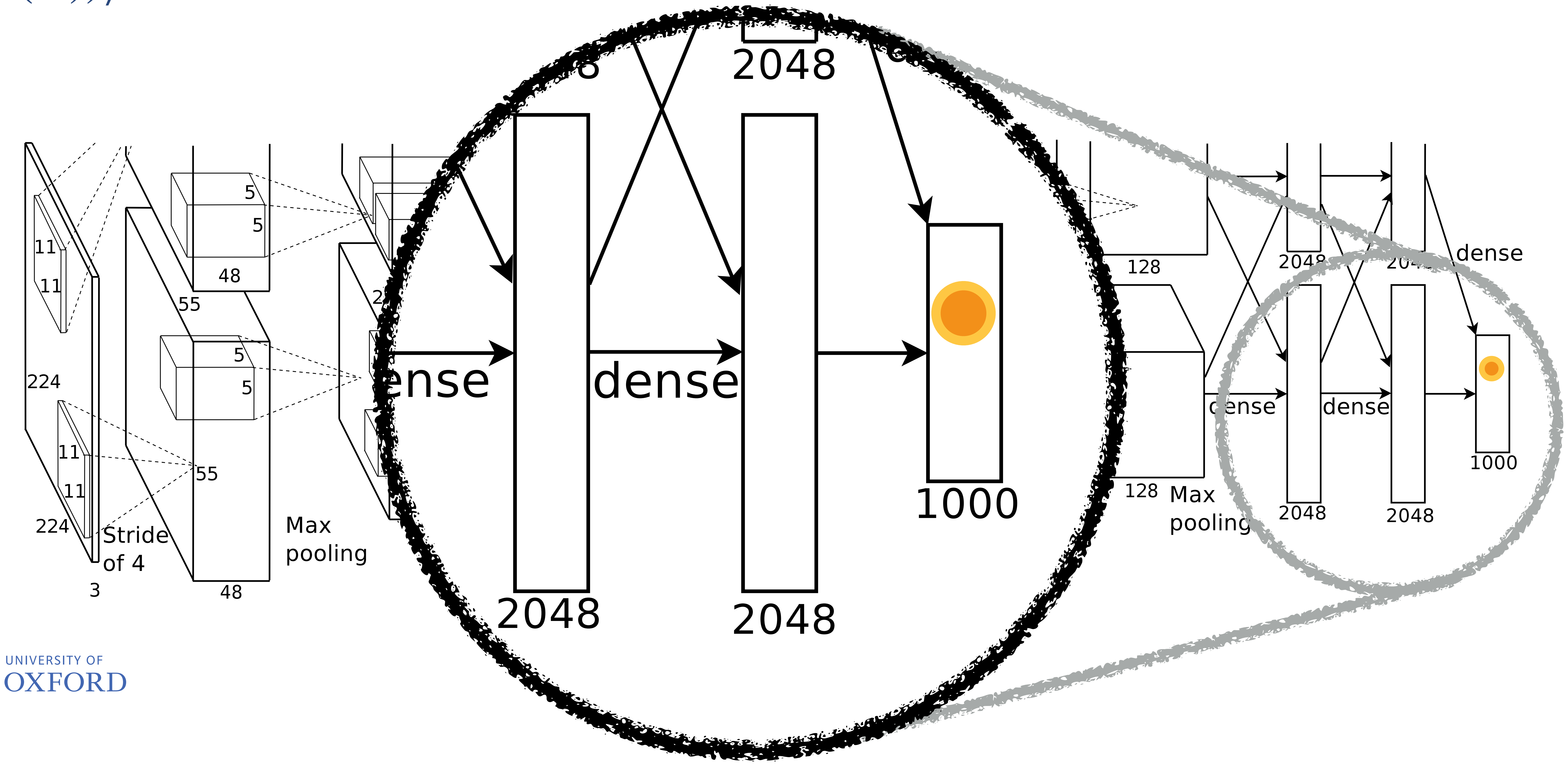
fc8 is a 1000-dimensional **class score vector**…
or is it?

facebook
Artificial Intelligence

UNIVERSITY OF OXFORD

Activation maximization

$$\min_{\mathbf{w}} -\langle \mathbf{e}_k, \Phi(\Psi(\mathbf{w}))\rangle$$



Deep Quiz

https://goo.gl/jURsCP


Black Swan (Cygnus
New Zealand on
Copyright David Hastings



References

Visualizing higher-layer features of a deep network.
Erhan, Bengio, Courville, U Montreal, 2009

Activation maximisation for class neurons

Visualizing and understanding convolutional networks
Zeiler Fergus. Proc. ECCV, 2014.

Activation maximization using **empirical prior**, **deconvnet**

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
Simonyan Zisserman Vedaldi, ICLR, 2104

Activation maximization and **saliency**

Understanding deep image representations by inverting them
Mahendran Vedaldi, CVPR, 2015

Inversion at different depths, **natural image prior**

Google "inceptionsm"
Mordvintsev et al. 2015

Activation maximisation for **intermediate neurons**
Improved regularizers, artistic applications (deep dreams)

Understanding neural networks through deep visualisation
Yosinksi et al. ICMLW, 2015

Activation maximization using **empirical prior**, **deconvnet**
More regularizers, toolbox

Plug & play generative networks: Conditional iterative generation of images in latent space
Nguyen, Yosinksi, Bengio, Dosovitskiy, Clune, CVPR, 2017

Strong learned regularizer, sample **diversity**

Deep image prior
Ulyanov Vedaldi Lempistky, CVPR, 2018

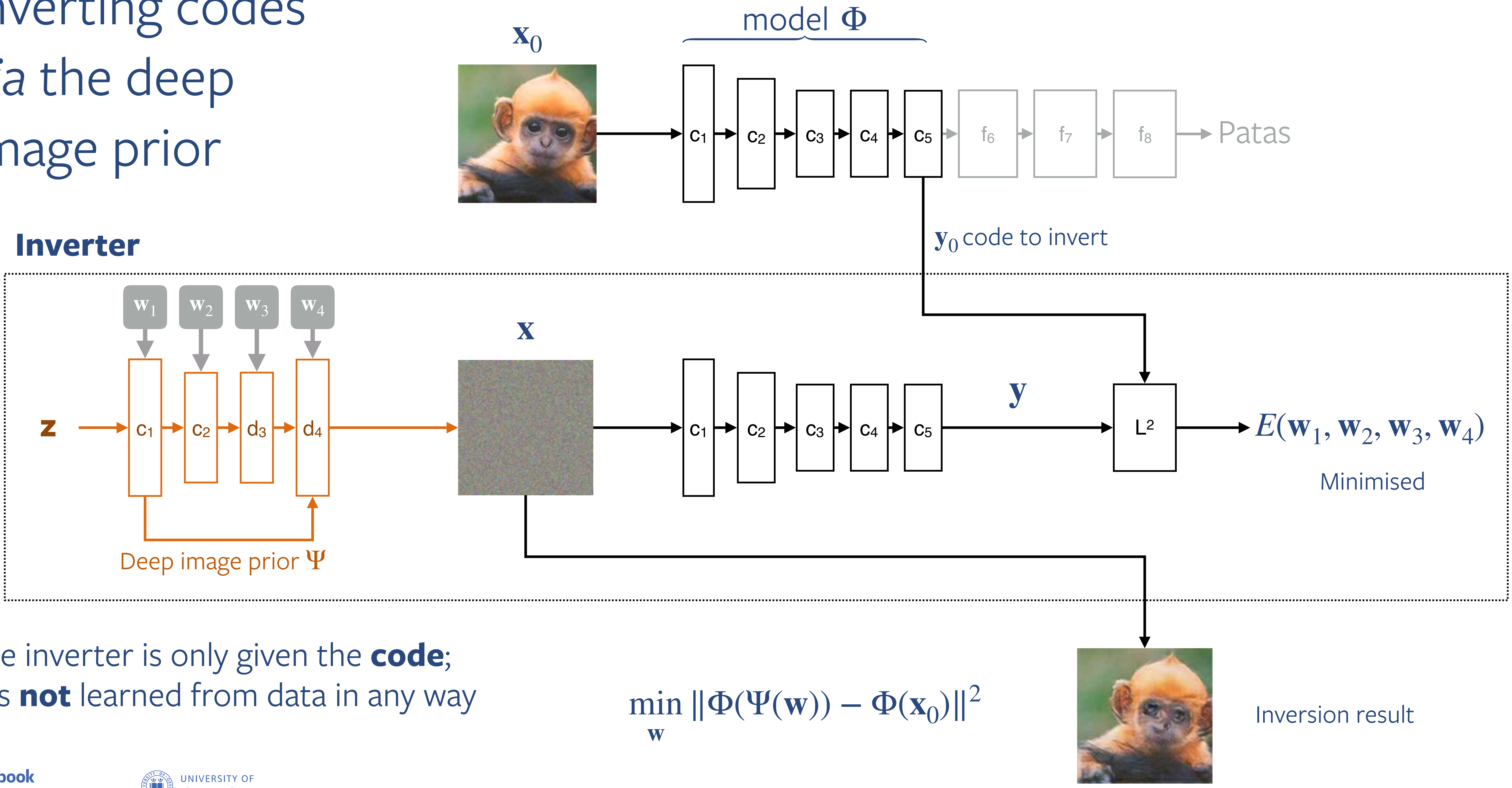
Advanced "data agnostic" regularization

Effect of the prior

Deep Image Prior

TV-Norm Prior

Inverting codes *via* the deep image prior

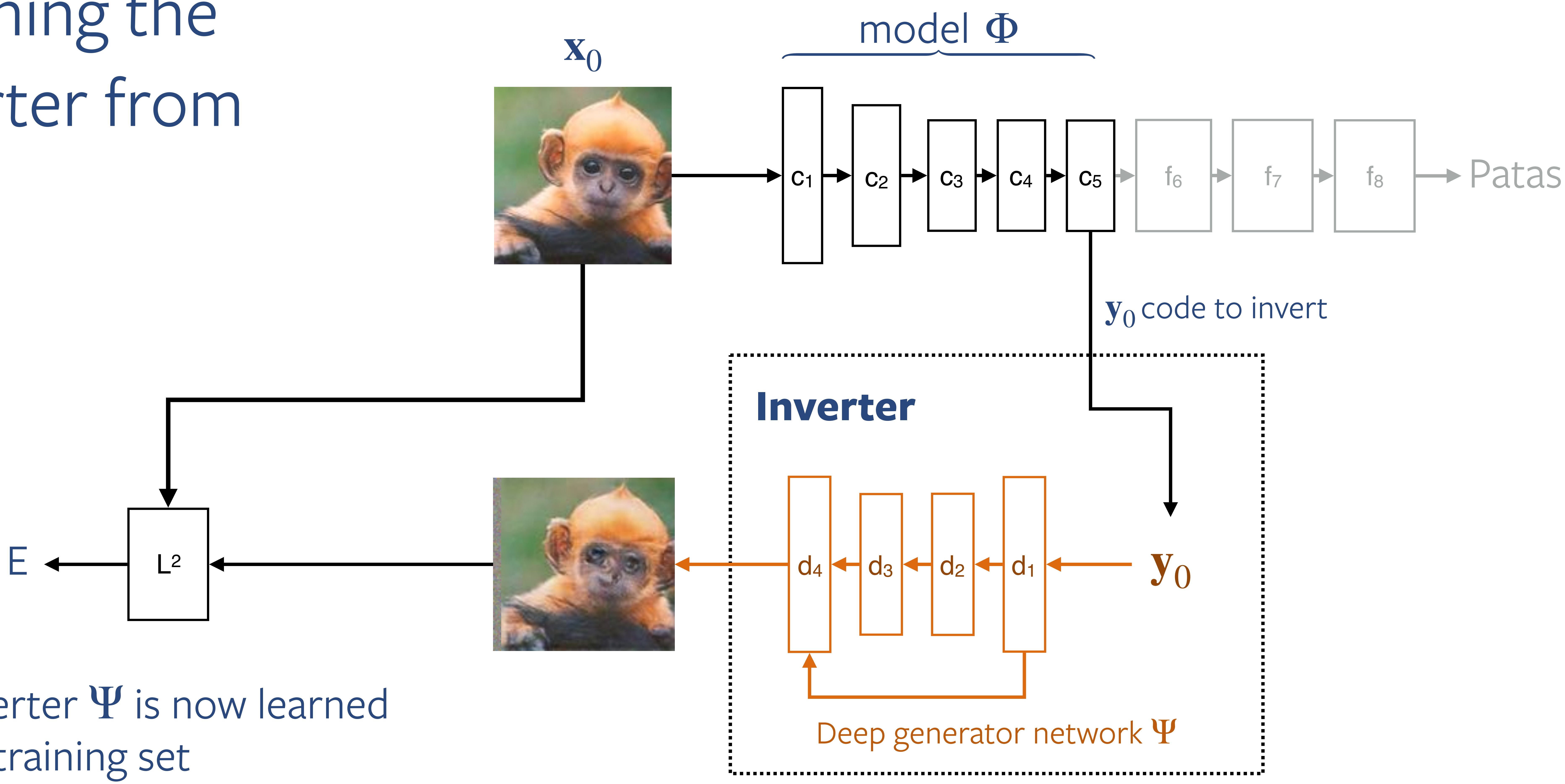


The inverter is only given the **code**;
it is **not** learned from data in any way

$$\min_{\mathbf{w}} \|\Phi(\Psi(\mathbf{w})) - \Phi(\mathbf{x}_0)\|^2$$

facebook
Artificial Intelligence
UNIVERSITY OF OXFORD

Learning the inverter from data



The inverter Ψ is now learned using a training set

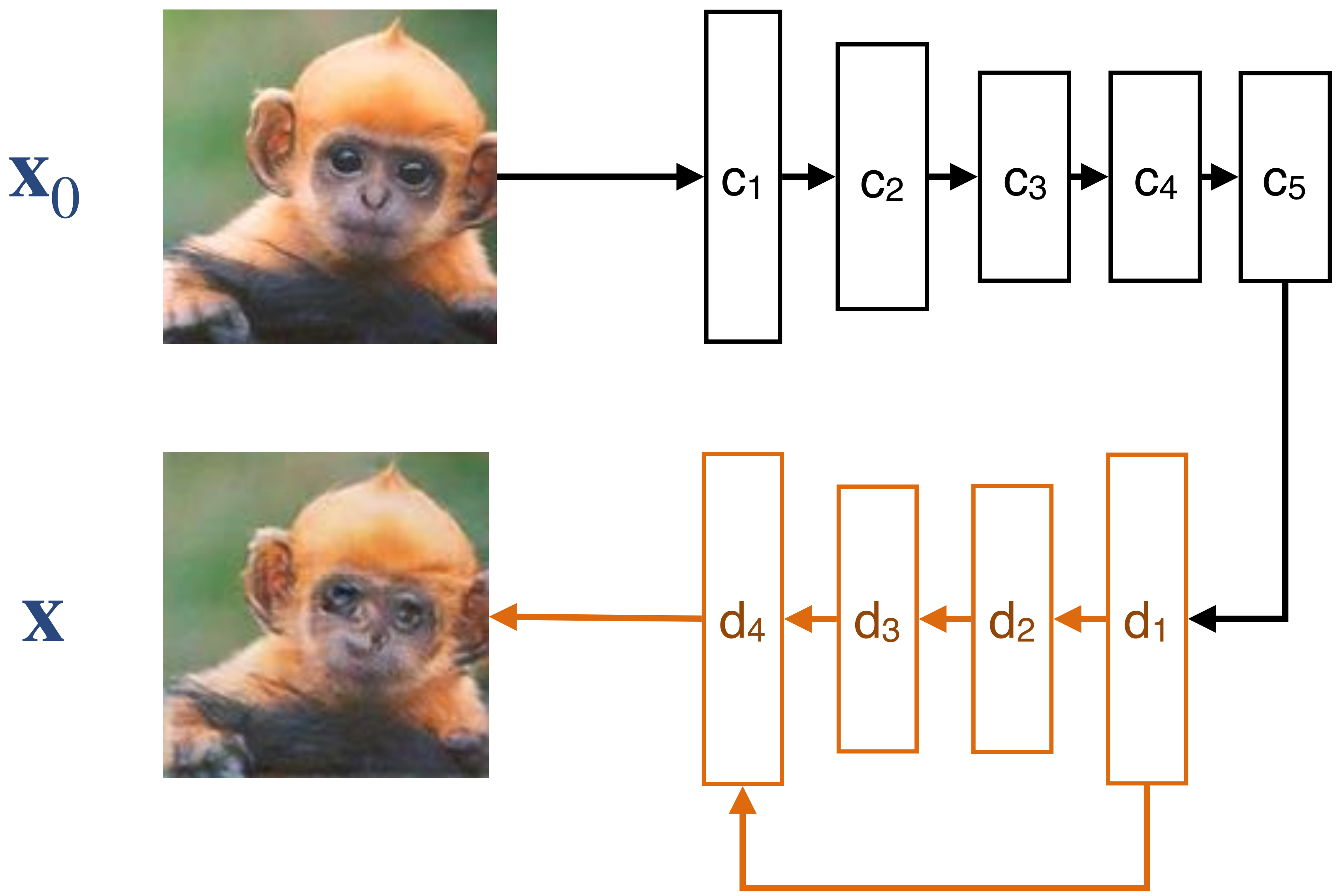
$$\min_{\Psi} \frac{1}{N} \sum_{i=1}^{N} \|\Psi(\Phi(\mathbf{x}_i)) - \mathbf{x}_i\|^2 \quad +$$

IMAGENET

Learning the inverter

Popular methods combine:

- perceptual loss $\mathbf{x}_0 \approx \mathbf{x}$
- feature rec. loss $\Phi(\mathbf{x}_0) \approx \Phi(\mathbf{x})$
- adversarial loss (GAN) $p(\mathbf{x}_0) \approx p(\mathbf{x})$
- IMAGENET



Inverting convolutional networks with convolutional networks
Dosovitskiy Brox, CVPR, 2016

Synthesizing the preferred inputs for neurons in neural networks via deep generator networks
Nguyen, Dosovitskiy, Yosinski, Brox, Clune, NIPS, 2016

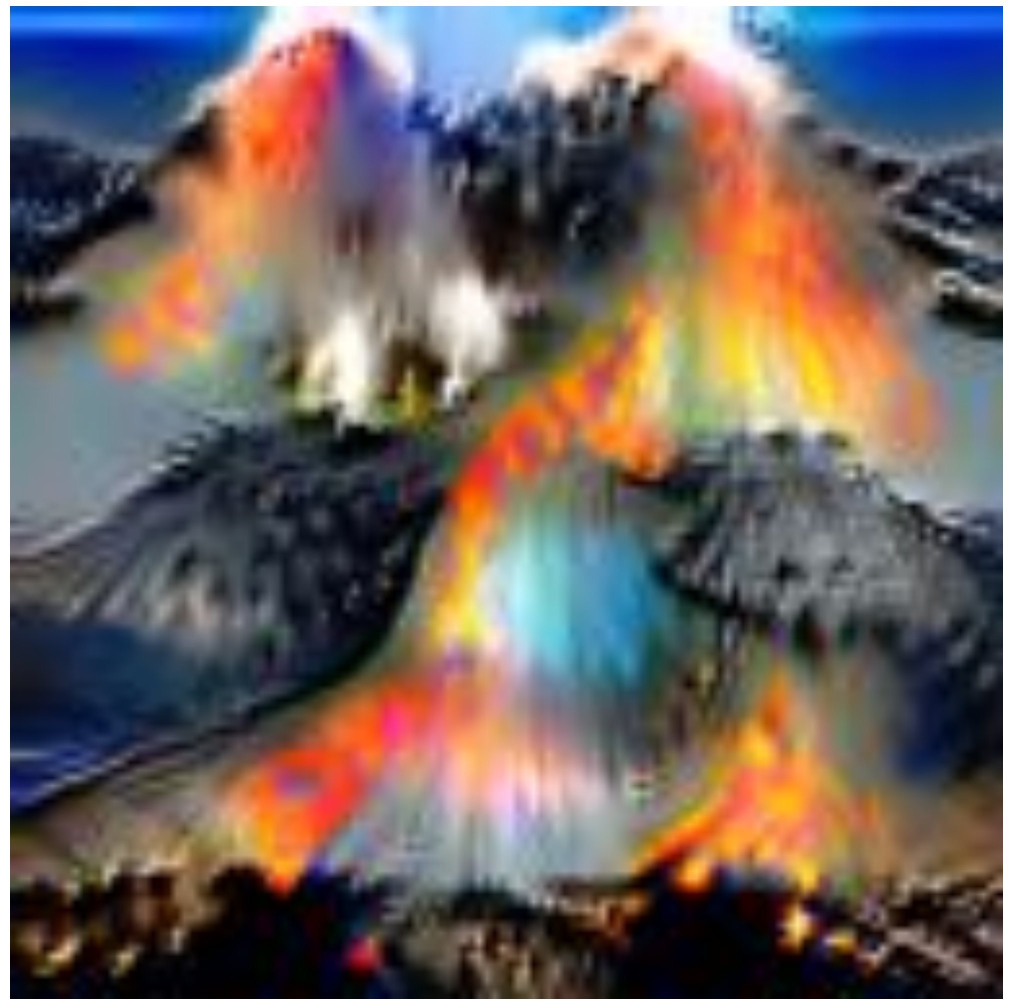
Generating images with perceptual similarity metrics based on deep networks
Dosovitskiy Brox, NIPS, 2016

Plug & play generative networks: Conditional iterative generation of images in latent space
Nguyen, Yosinksi, Bengio, Dosovitskiy, Clune, CVPR, 2017

Diagnostic *vs* aesthetic value

Our goal: diagnose a given **network** Φ

But inversions **also** reflect the chosen "natural image" **prior** $p(\mathbf{x})$

Deep Image Prior	Plug & Play Gen. Net.	Empirical prior
		

$p(\mathbf{x}) =$

only prior is the **structure** of the gener.	prior comes from training a **GAN** on **ImageNet**	**ImageNet empirical distribution**

Illustrates the **model** Φ

Illustrates the **prior** $p(\mathbf{x})$

facebook
Artificial Intelligence

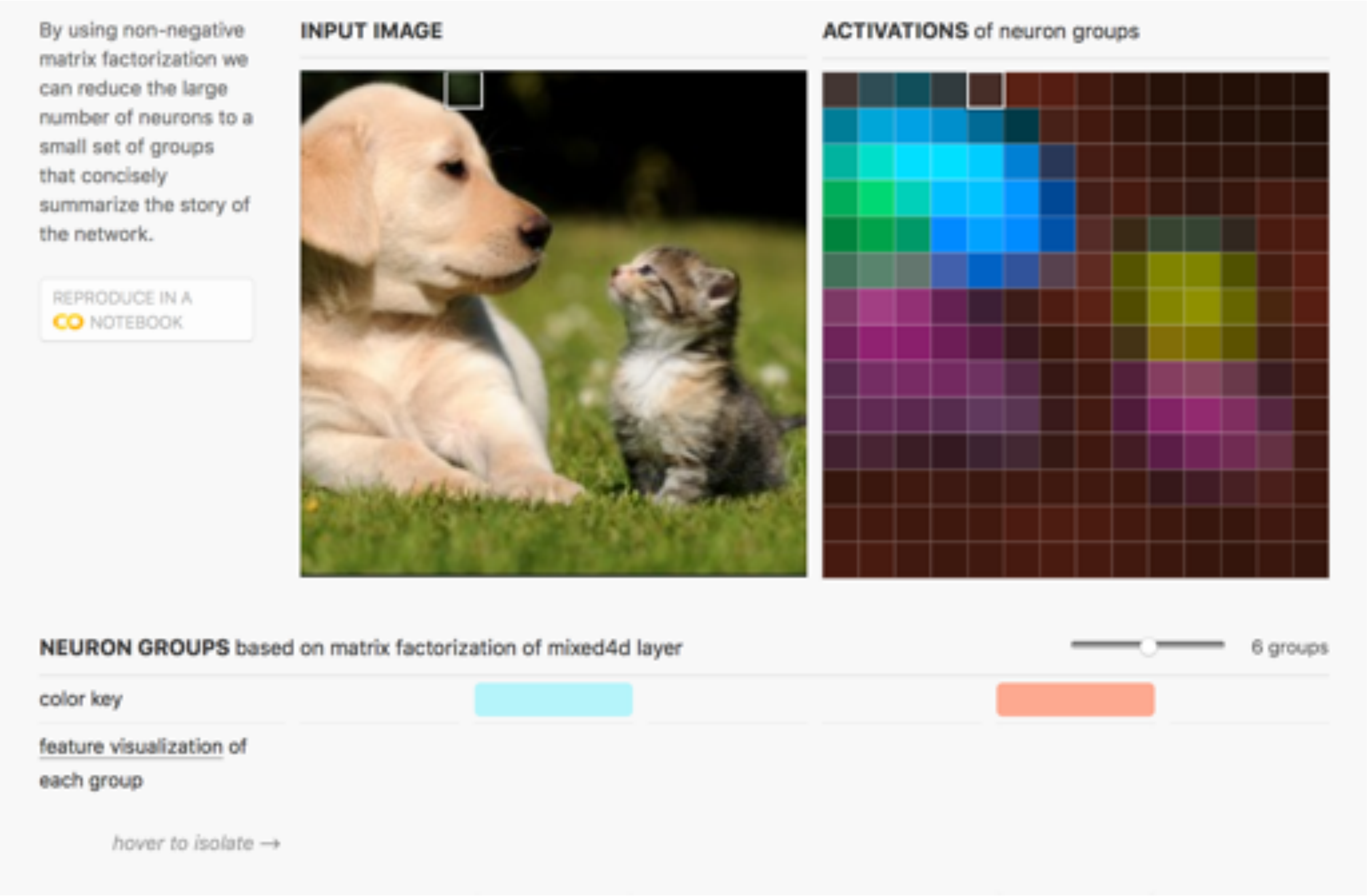
UNIVERSITY OF
OXFORD

Reviews and interfaces

The building blocks of interpretability
Olah, Satyanarayan, Johnson, Carter, Schubert, Ye, Mordvintsev
Distill, 2018. https://distill.pub/2018/building-blocks

Understanding neural networks through deep visualisation
Yosinksi et al. ICMLW, 2015

Definitely check out **Distill**!



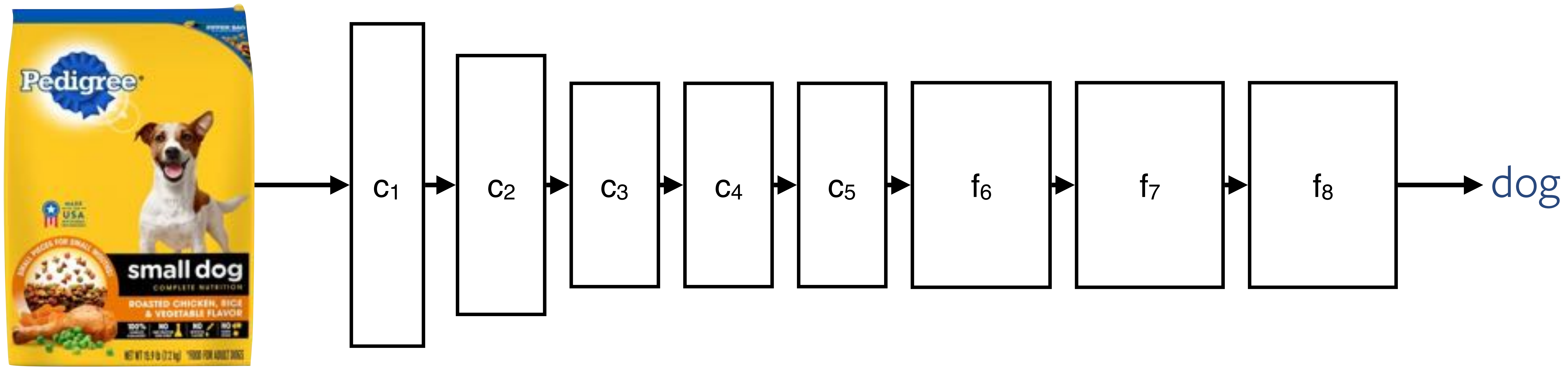
Generating iconic examples

Attribution

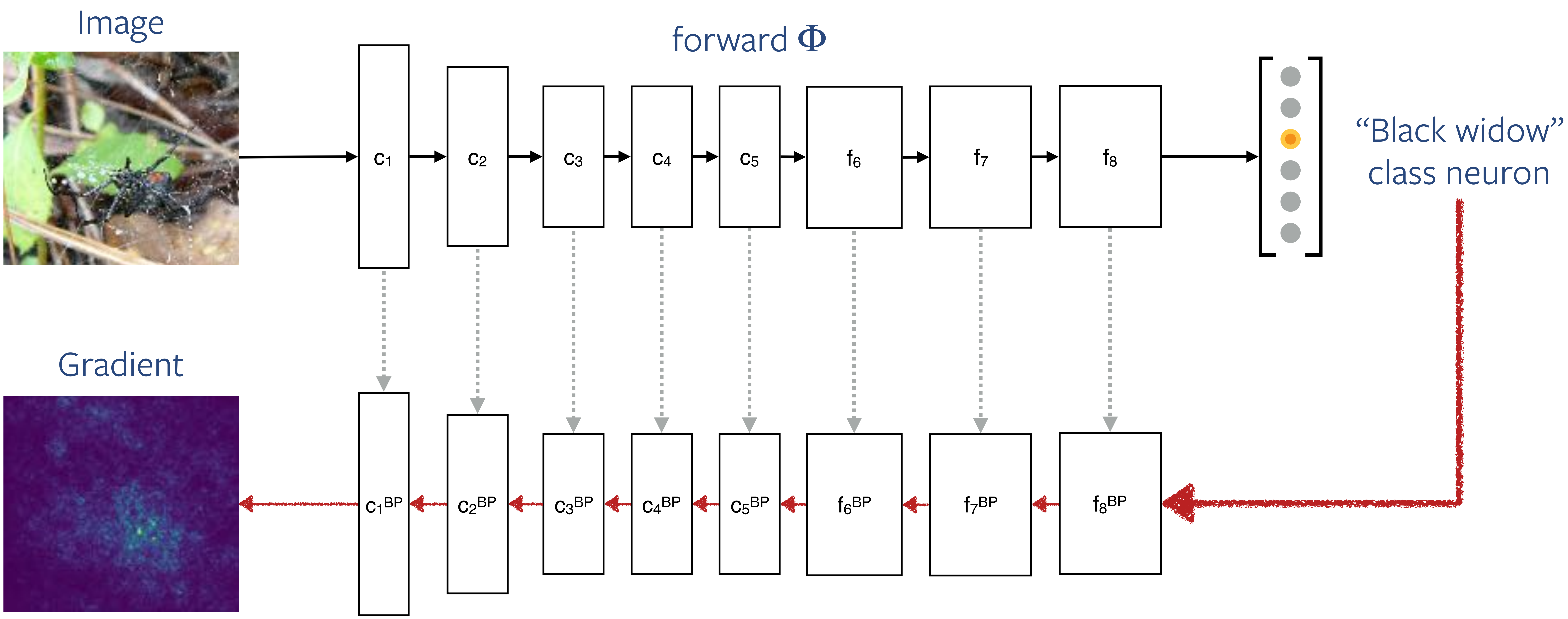
Attribution

Where is the model **looking**?

?



Backprop methods: grad



backward $J = \dfrac{d\Phi(\mathbf{x})}{d\mathbf{x}}$

The "salient" pixels usually light up

Deep inside convolutional networks, Simonyan, Vedaldi, Zisserman, ICLR, 2014

Early backprop methods

Deconvolution

Visualizing and understanding convolutional networks
Zeiler Fergus, ECCV, 2014

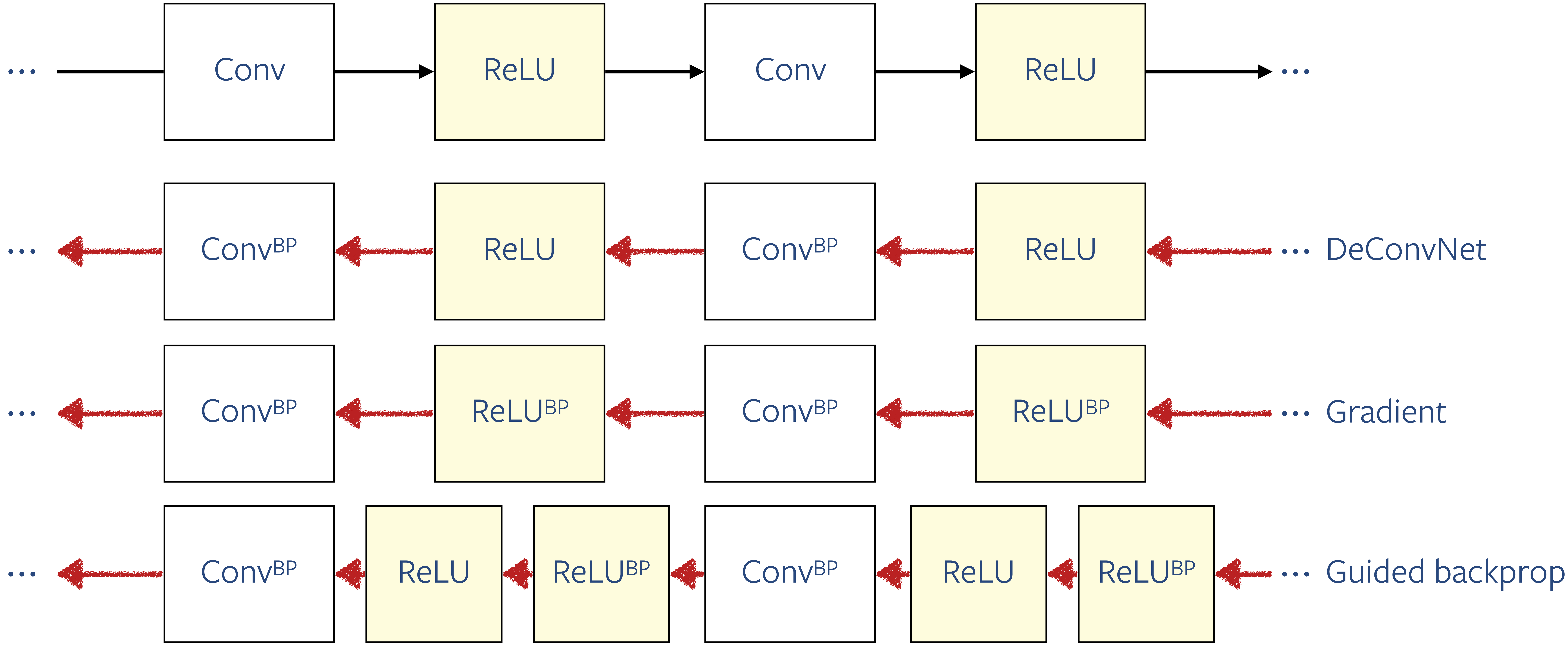
Gradient (backpropagation)

Deep inside convolutional networks: Visualising image classification models and saliency maps
Simonyan, Vedaldi, Zisserman, ICLR, 2014

Guided backpropagation

Striving for simplicity: The all convolutional net
Springenberg, Dosovitskiy, Brox, Riedmiller, ICLR, 2015

Backprop: deconv, grad, guided grad



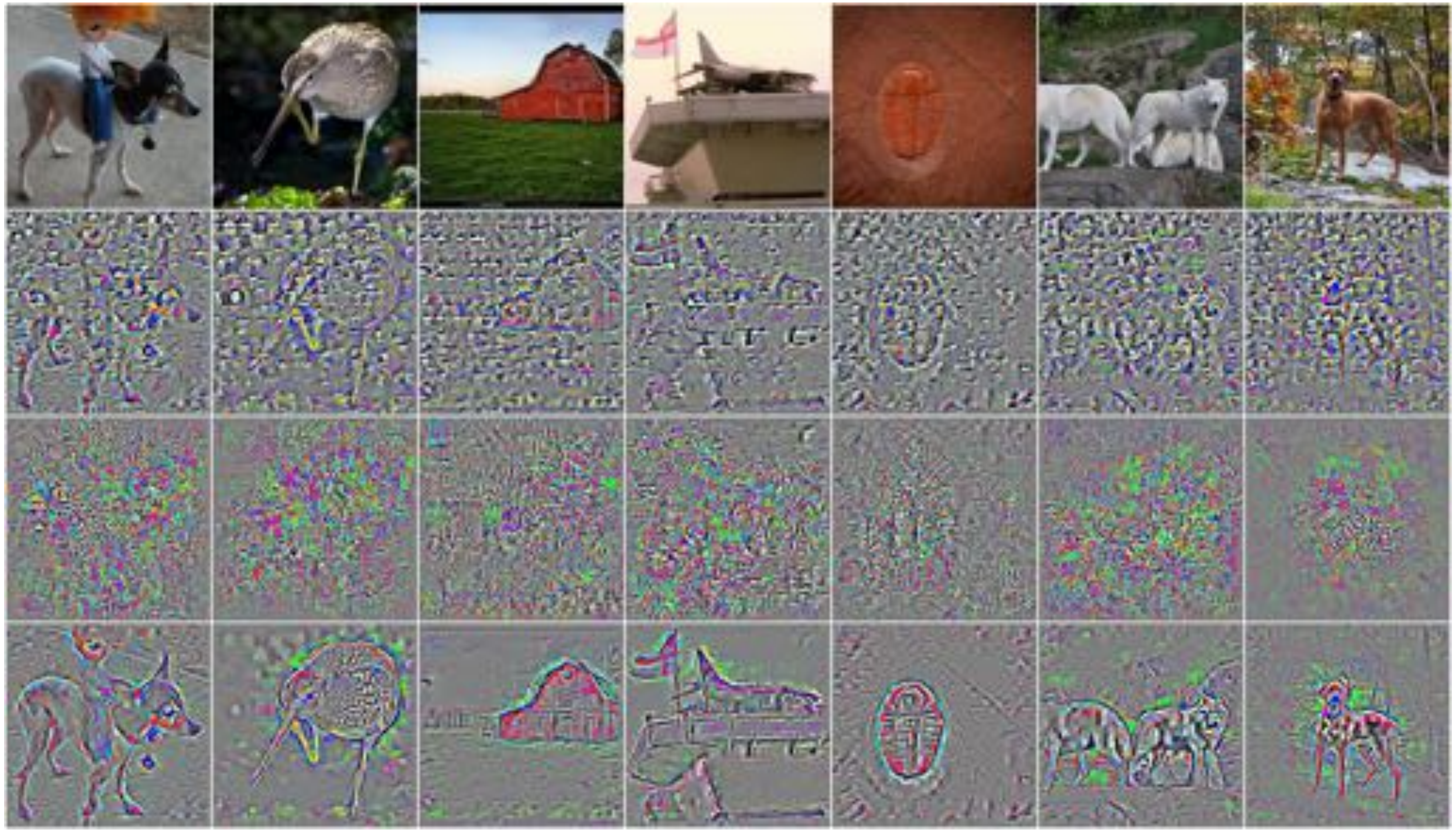
Salient deconvolutional networks, Mahendran Vedaldi, ECCV, 2016

Comparisons

DeConvNet

Gradient

Guided backprop



Salient deconvolutional networks. Mahendran Vedaldi, ECCV, 2016

Comparisons

Deconvolution

- Sharp
- Poor spatial selectivity

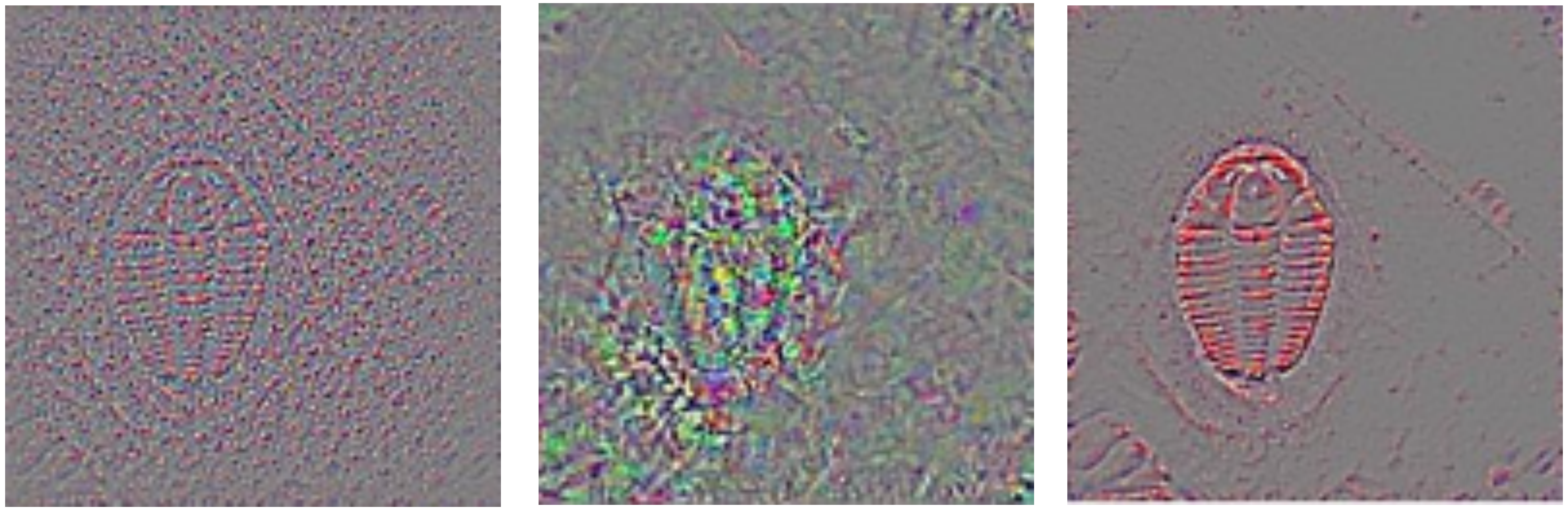
Gradient

- Blurry
- OK spatial selectivity

Guided Backprop

- Sharp
- OK spatial sensitivity

Deconvolution Gradient Guided Backprop



Warning: they all still have poor channel selectivity

Smoother grads

Gradient $$\frac{d\Phi(x)}{d\mathbf{x}}$$

Gradient × input $$\mathbf{x} \odot \frac{d\Phi(x)}{d\mathbf{x}}$$

Integrated Gradients $$(\mathbf{x} - \bar{\mathbf{x}}) \otimes \int_0^1 \frac{d\Phi(\bar{\mathbf{x}} - \alpha(\mathbf{x} - \bar{\mathbf{x}}))}{d\mathbf{x}} \, d\alpha$$

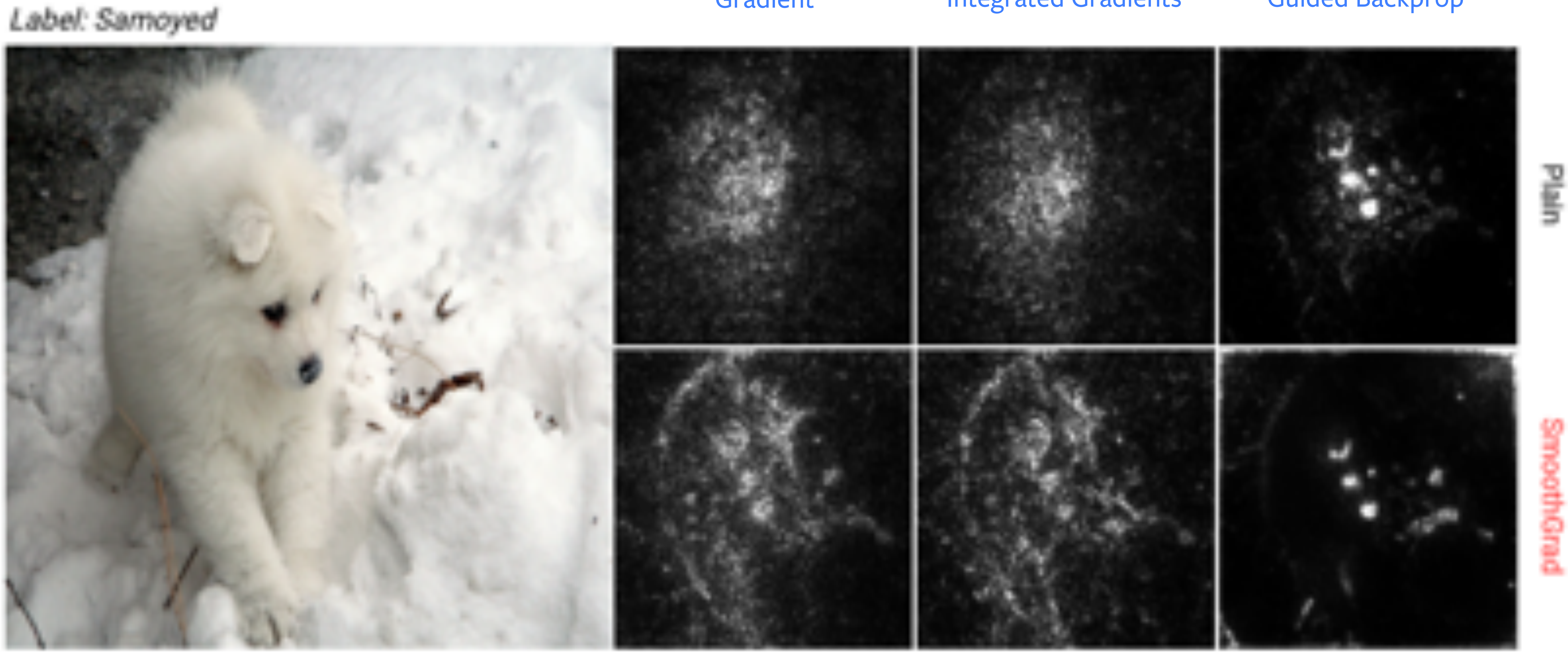
Axiomatic attribution for deep networks.
Sundararajan, Taly, Yan. Proc. ICML, 2017.

SmoothGrads $$E\left[\frac{d\Phi(\mathbf{x} + \epsilon)}{d\mathbf{x}}\right], \quad \epsilon \sim \mathcal{N}$$

Smoothgrad: removing noise by adding noise.
Smilkov, Thorat, Víegas, Wattenbeg. CoRR, 2017

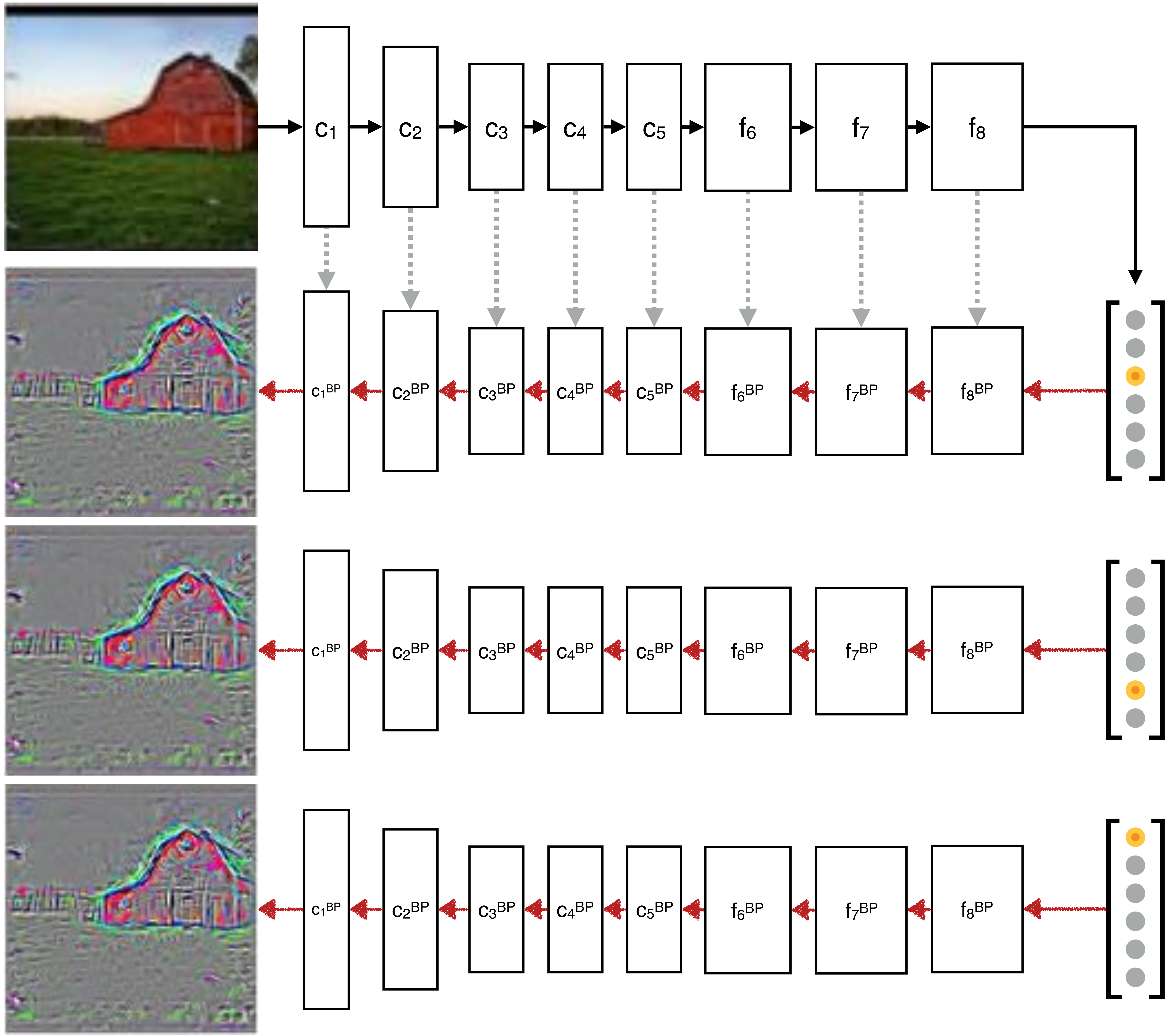
Comparisons

Gradient | Integrated Gradients | Guided Backprop



Lack of channel specificity

Visualising any output results in about the same result



Attribution for:

maximally activated neuron

random neuron

minimally activated neuron

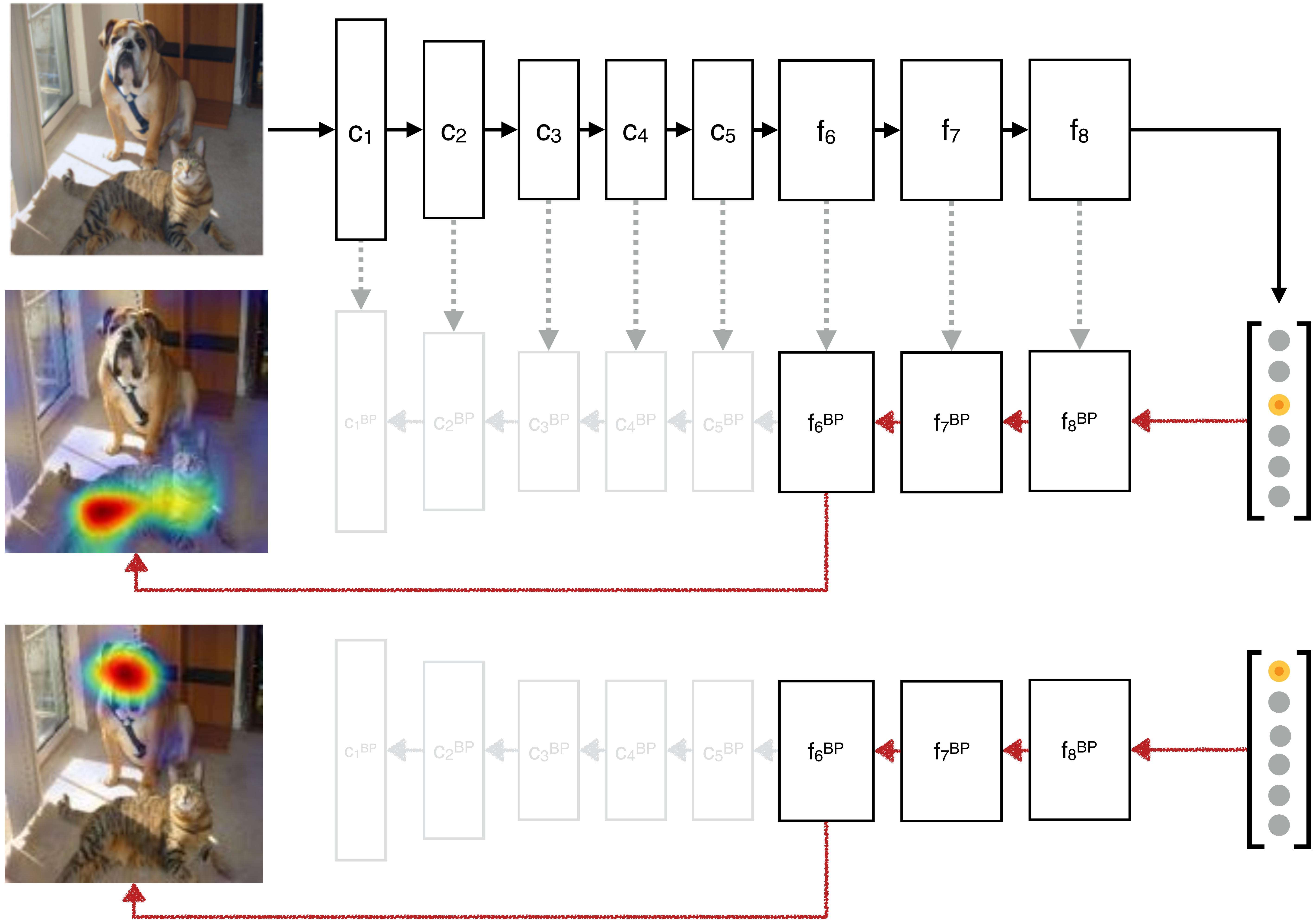
facebook Artificial Intelligence

UNIVERSITY OF OXFORD

Backprop: CAM and Grad-CAM

Learning deep features for discriminative localization
Zhou, Khosla, Lapedriza, Oliva, Torralba, CVPR, 2016

Grad-CAM: Visual explanations from deep networks via gradient-based localization
Selvaraju, Cogswell, Das, Vedantam, Parikh, Batra, ICCV, 2017

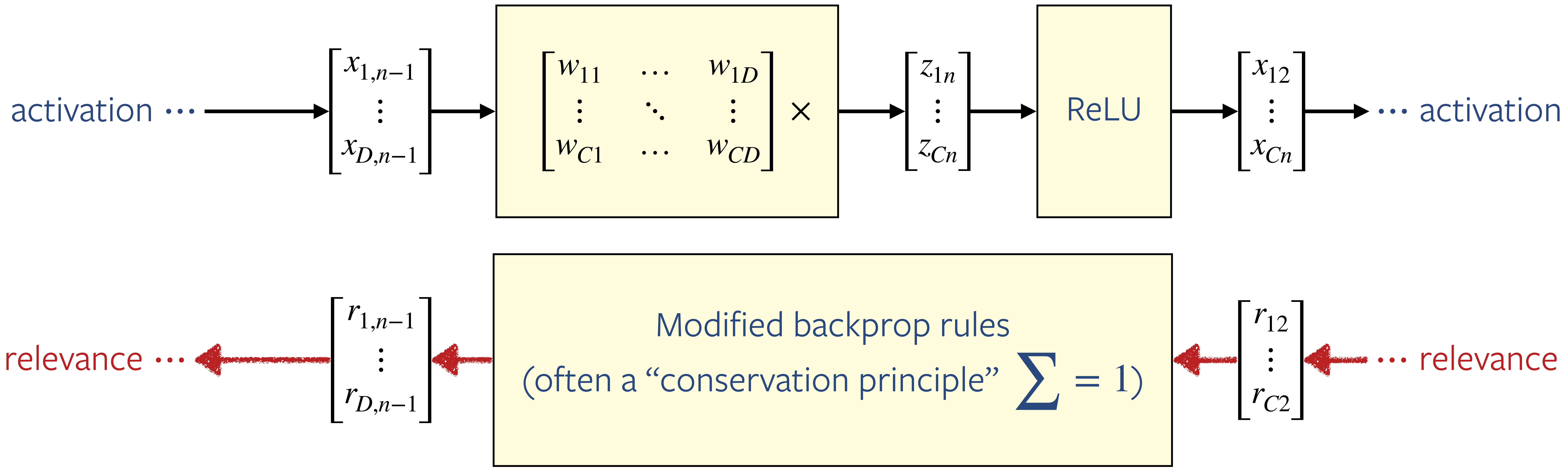


Attribution for:

cat class neuron

dog class neuron

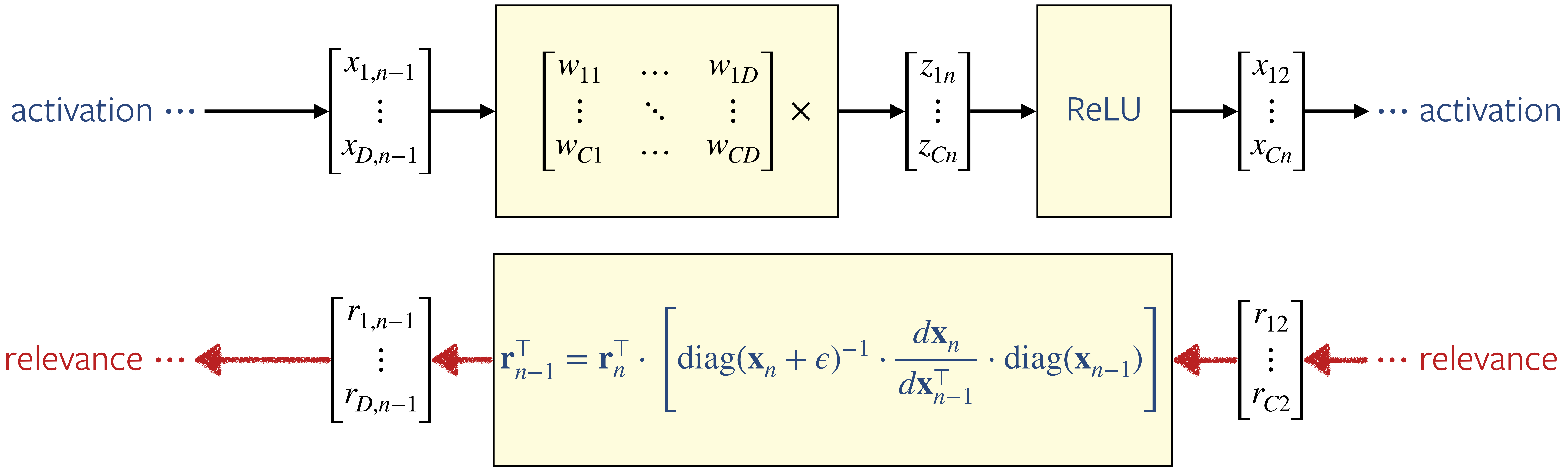
Relevance and excitation backprop



On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation
Bach, Binder, Montavon, Klauschen, Müller. PLOS one, 2015

Top-down neural attention by excitation backprop
Zhang, Lin, Brandt, Shen, Sclaroff, ECCV, 2016

Relevance and excitation backprop

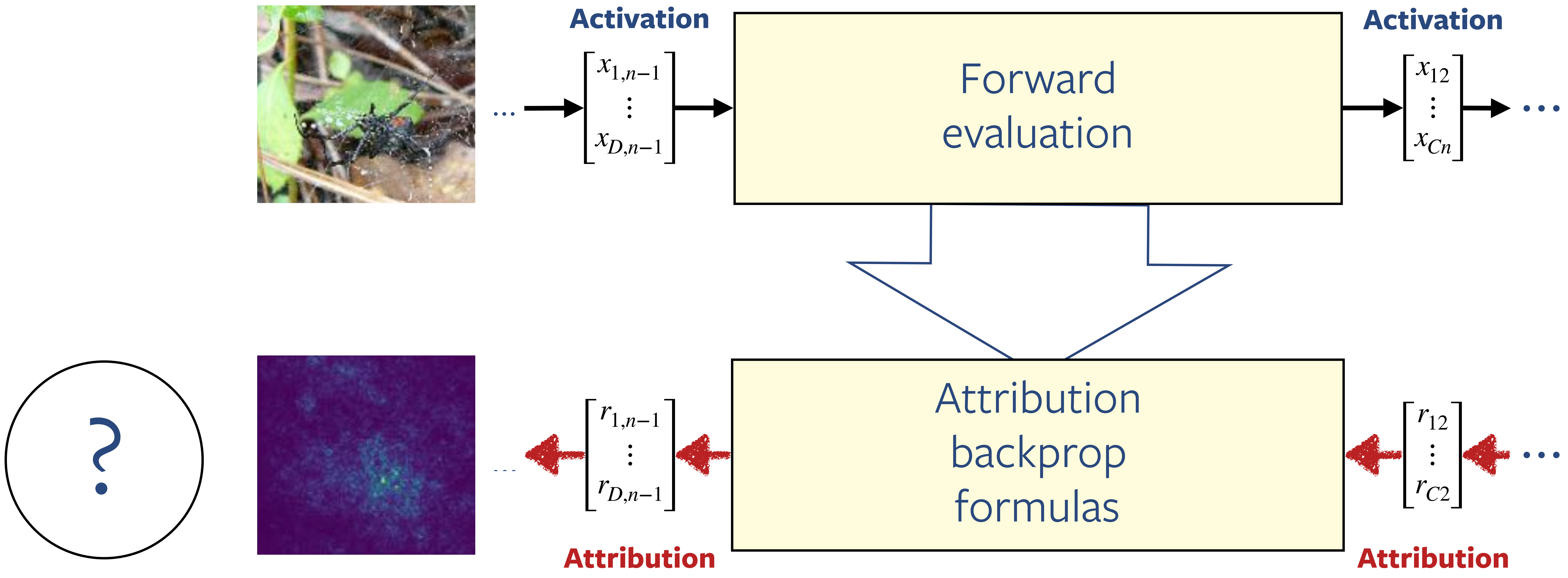


Actual rules are more sophisticated, please see references!

The meaning of attribution maps

For most methods, attribution is defined algorithmically

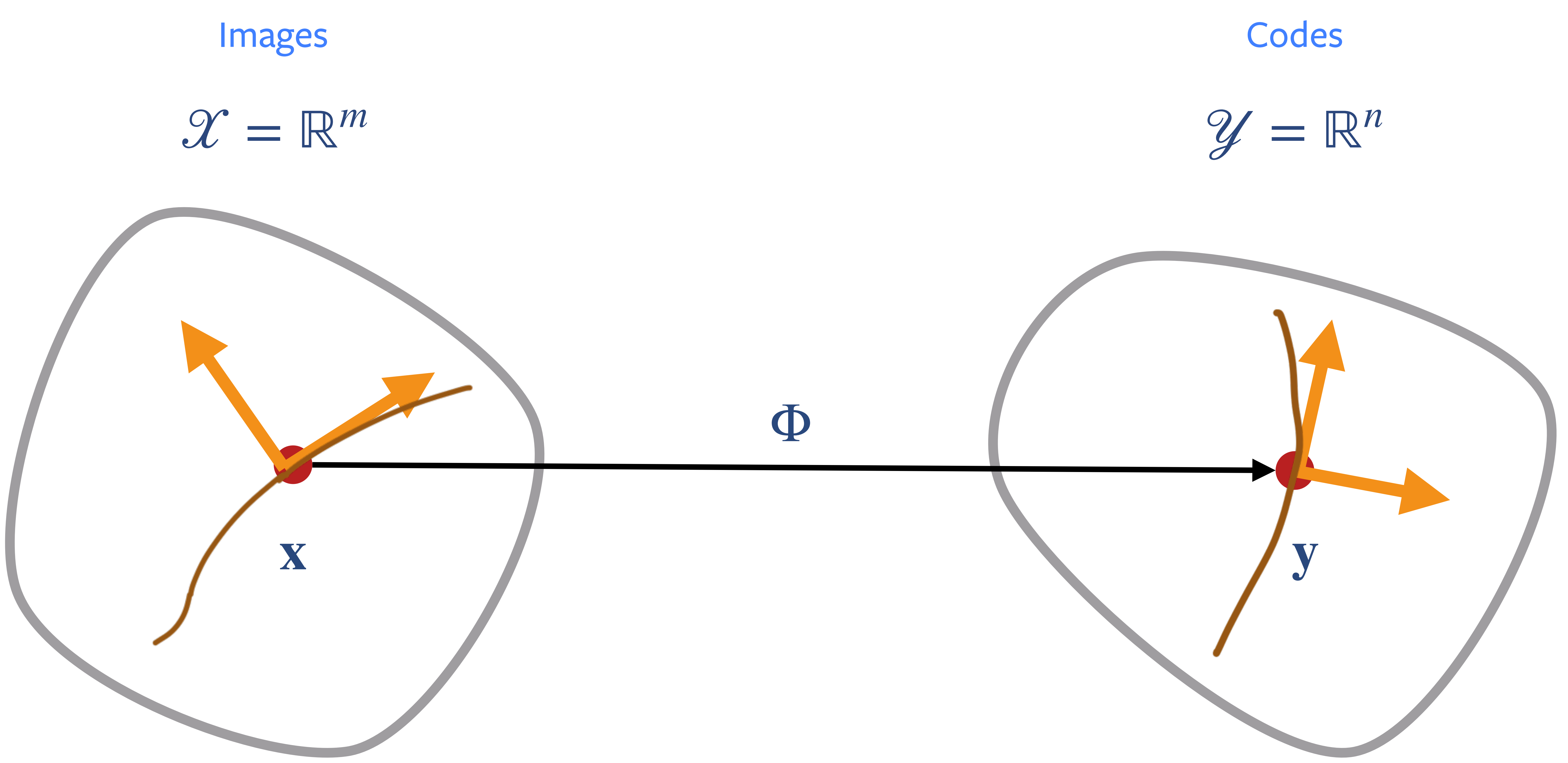
Hence, the **meaning** of the output is **not so clear**



Grad method = sensitivity analysis

The **gradient** can be directly interpreted as a **local linear approximation** of the model

$$\Phi(\mathbf{x}) \approx \left\langle \frac{d\Phi}{d\mathbf{x}}, \mathbf{x} - \mathbf{x}_0 \right\rangle + \Phi(\mathbf{x}_0)$$



Perturbation analysis

Study how $\Phi(\mathbf{x})$ changes up to perturbations $\pi(\mathbf{x})$ of the input $\mathbf{x}$

Perturbations should be meaningful (interpretable). E.g:

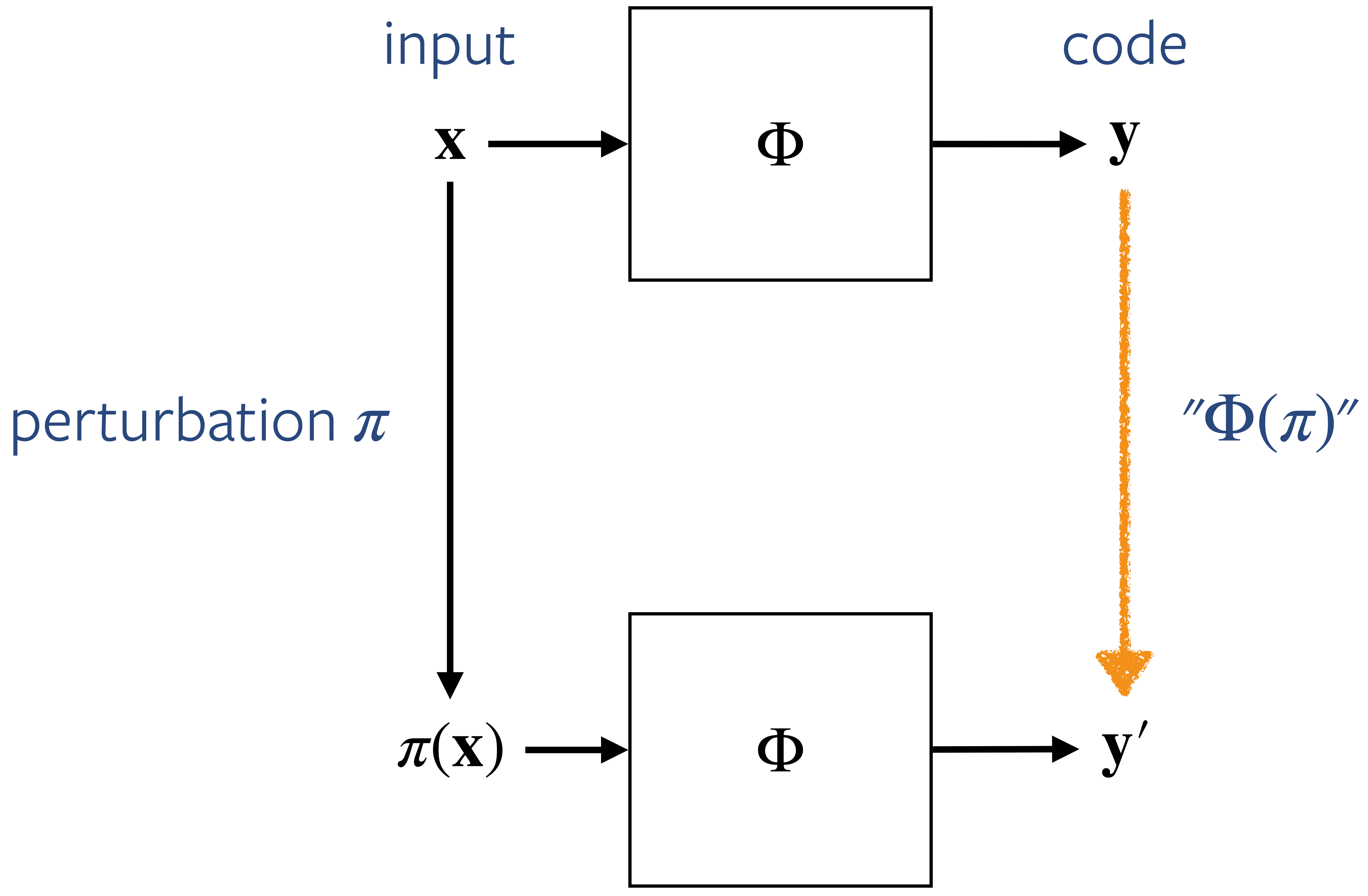
- Injecting noise
- Rotating or translating the image
- Erasing parts of the image

The representation may

- Be invariant (stay the same)
- Be equivariant (respond predictably)

The analysis may be

- Local around $\mathbf{x}$ and π
- For a distribution $p(\mathbf{x})$ and a fixed $p(\pi)$
- For a distribution $p(\pi)$ and a fixed $\mathbf{x}$

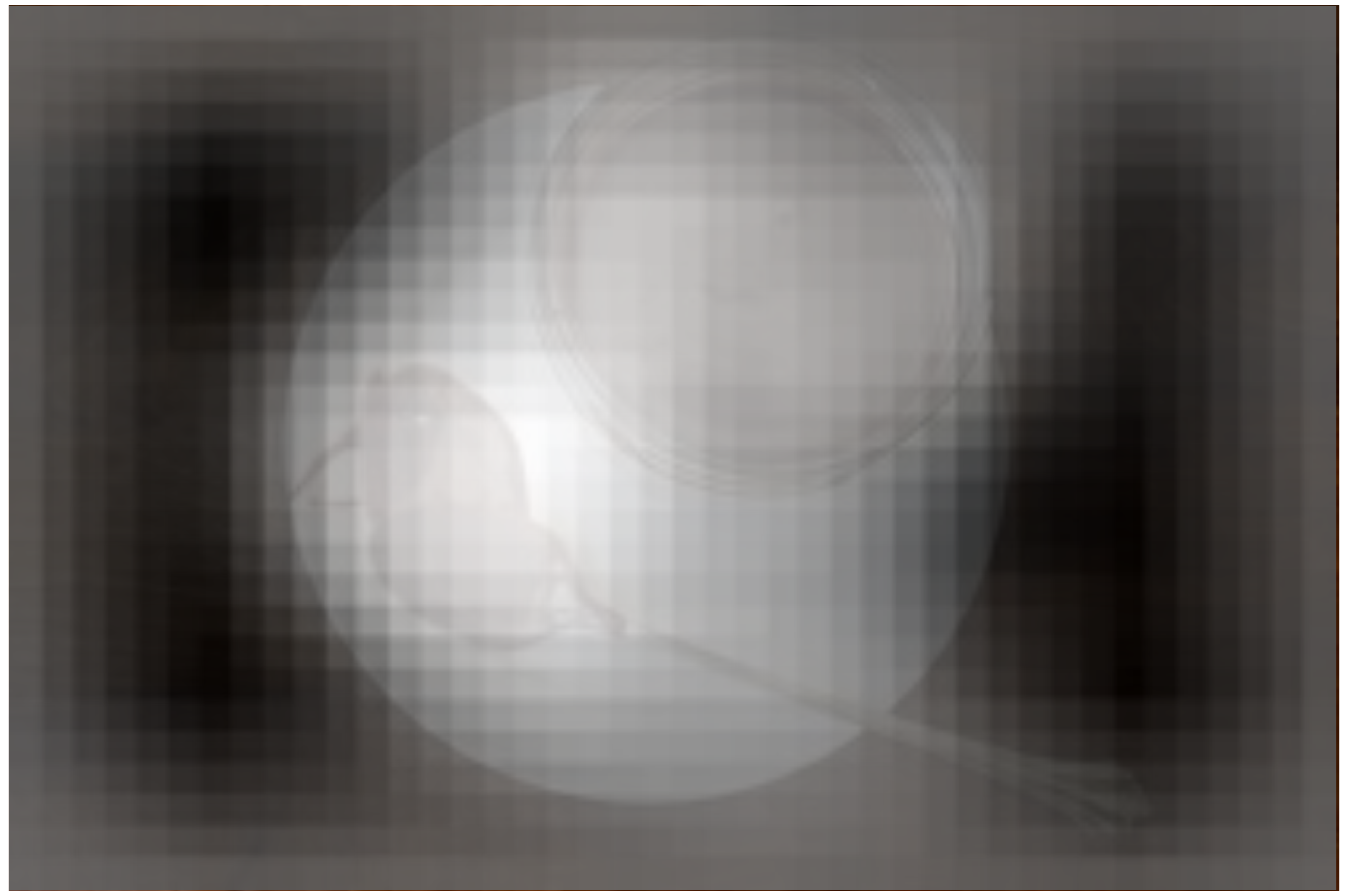


Perturbation analysis

Change the input and observe the effect on the output

Input

Occlusion



RISE

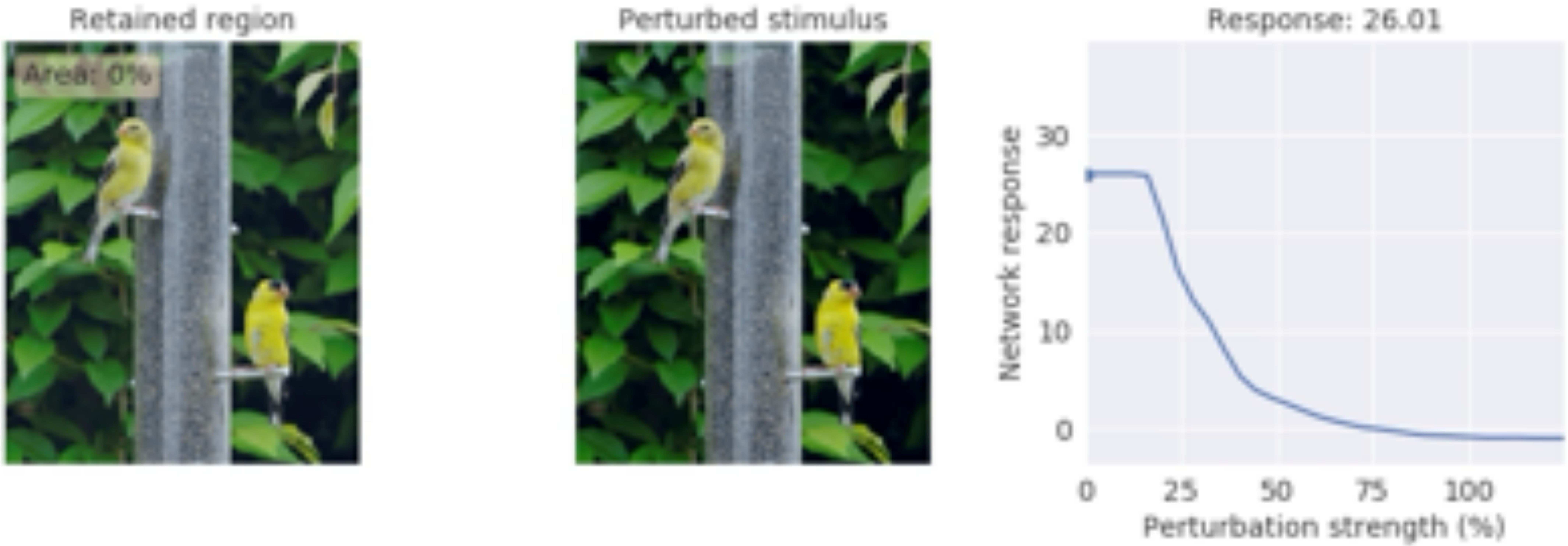
Clear meaning, but can only test a small number of occlusion patterns

[Zeiler and Fergus, ECCV 2014; Petsiuk et al., BMVC 2018]

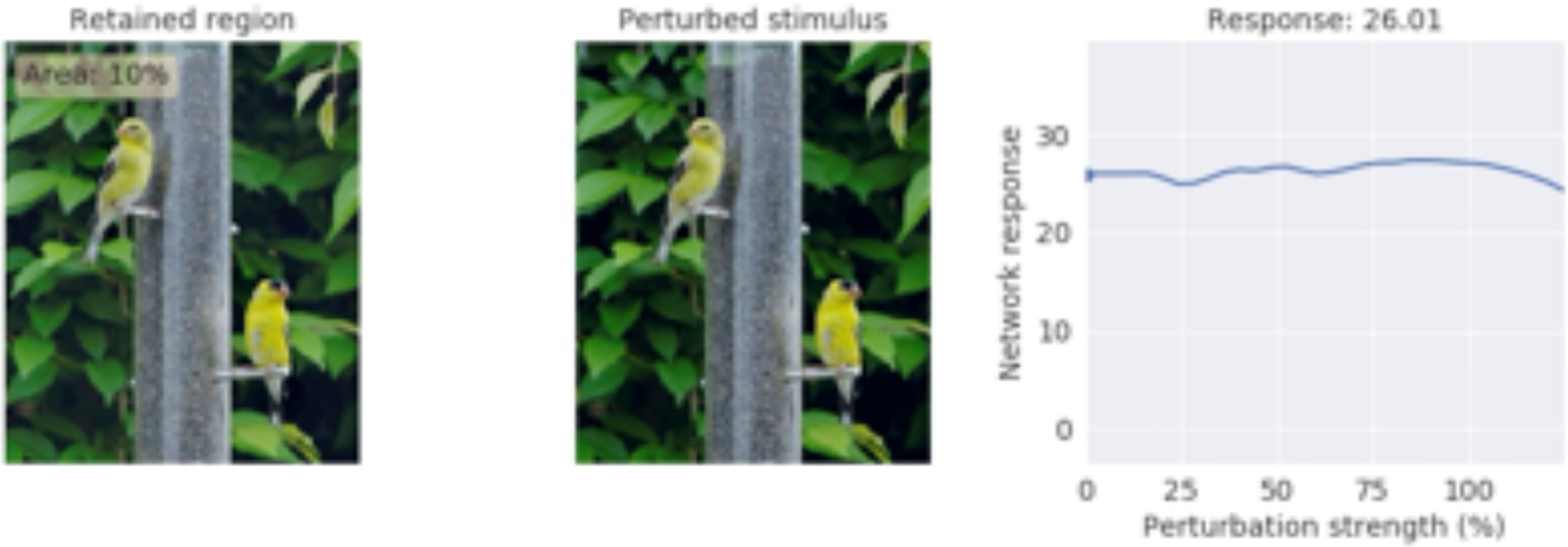
Extremal Perturbations

Find regions of a **given area** that preserves the network's response the most

Blur everywhere ⇒ response suppressed



Preserve 10% ⇒ response preserved



Meaningful perturbations

We seek the "smallest elision" that maximally changes the neuron activation

Original

"cat" probability
1.00

Redact-out

"cat" probability
0.5

(ineffective)

Blur-out

"cat" probability
0.01

(more meaningful)

Adversarial perturbations

Neural networks are fragile to adversarial perturbations

Adversarial perturbations attract gradient descent

Intriguing properties of neural networks. Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, Fergus. CoRR 2013

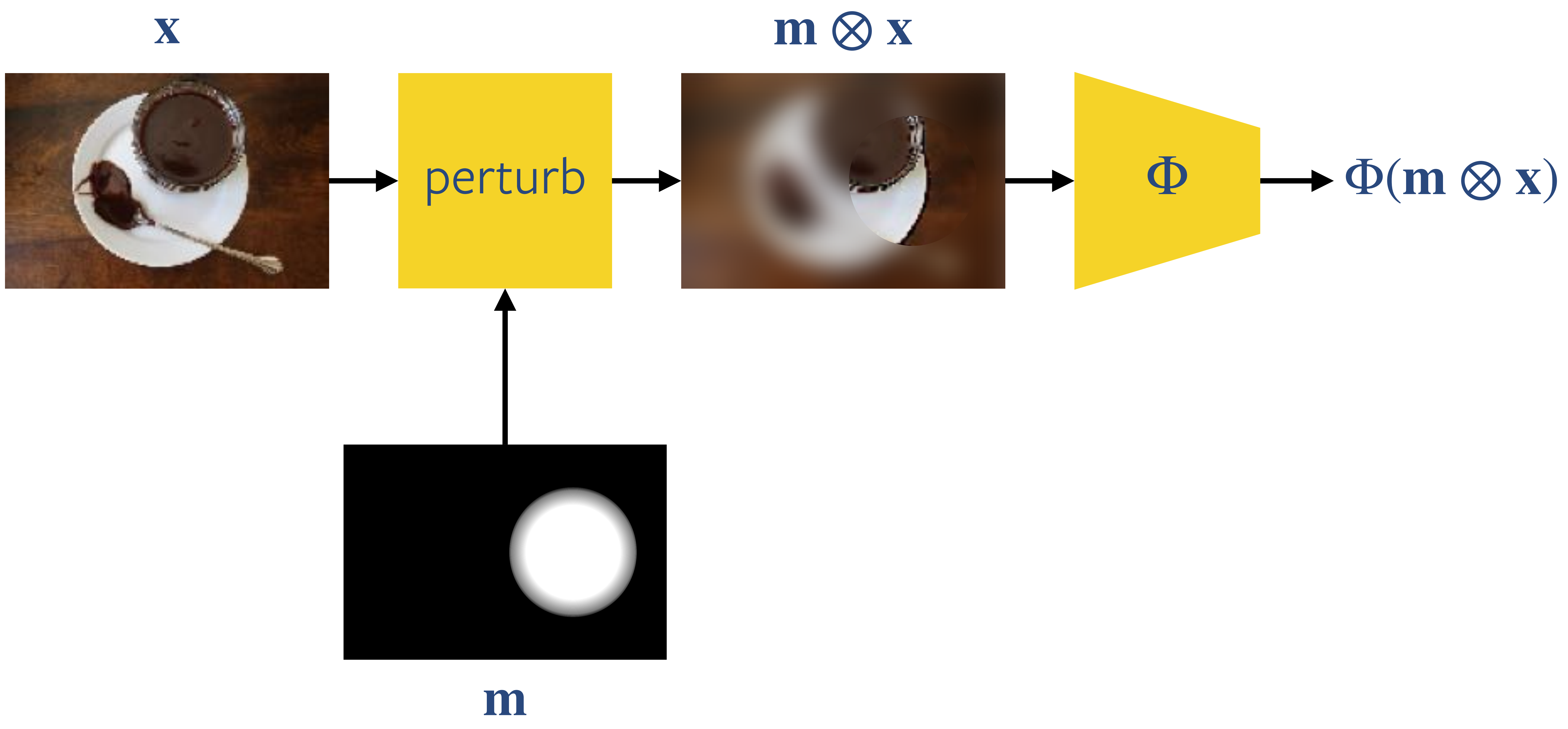
Original | Redacted | Mask

Extremal perturbations

A mask is optimized to maximally excite the network:

$$\underset{\mathbf{m}}{\operatorname{argmax}}\ \Phi(\mathbf{m} \otimes \mathbf{x})$$

subject to $\text{area}(\mathbf{m}) = a$



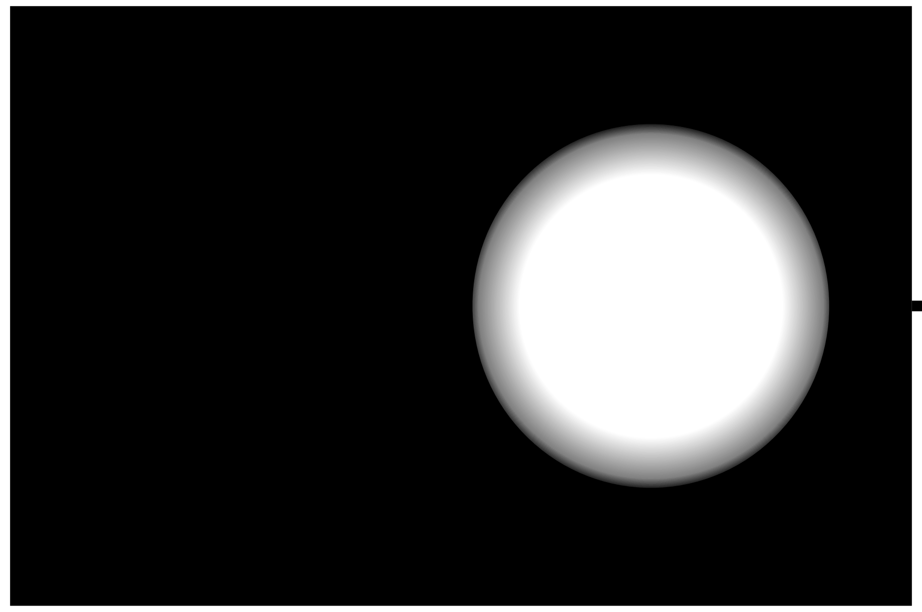
Area constraint

Optimizing w.r.t. to an area constraint is challenging

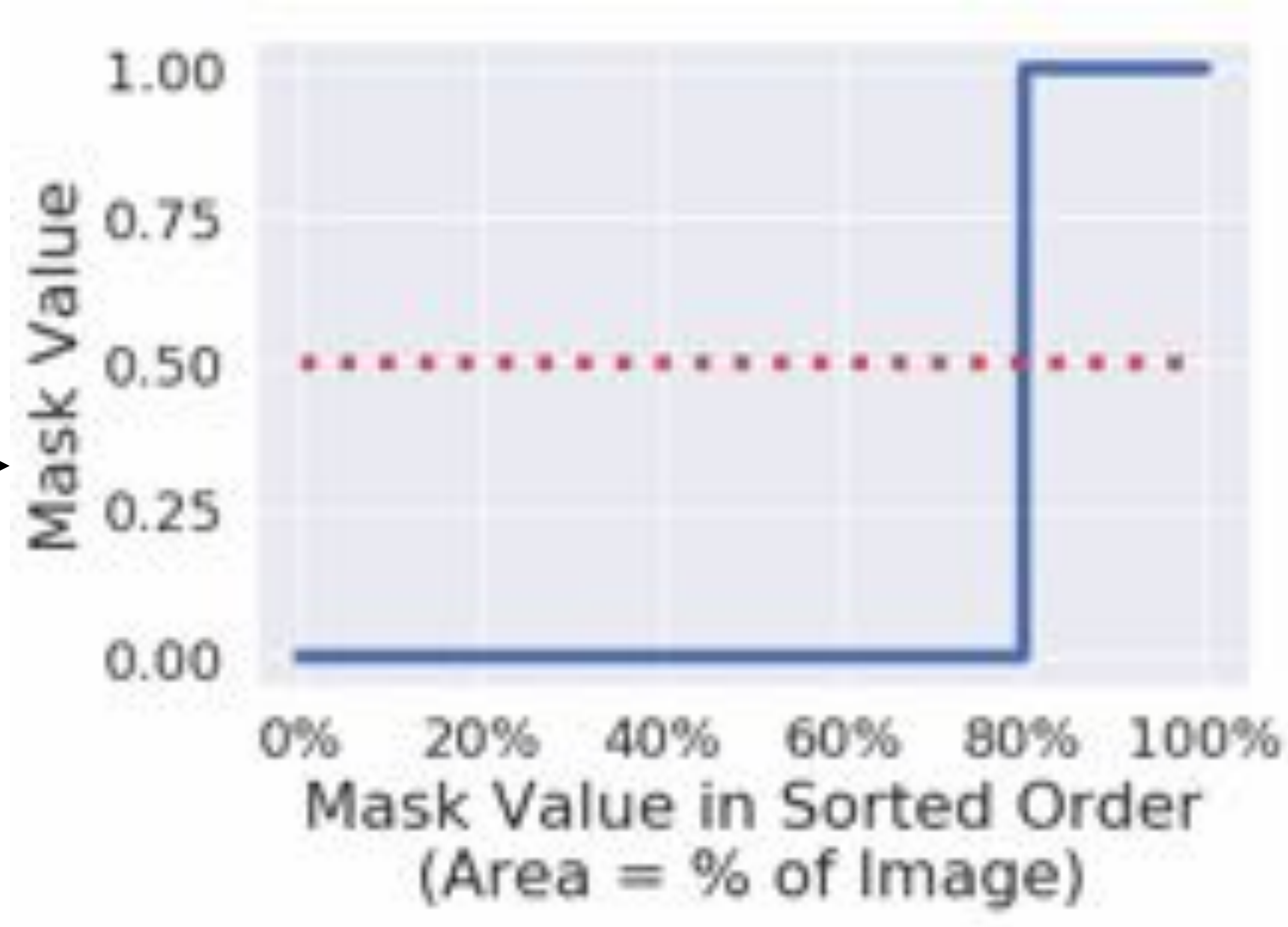
Here we re-formulate it as matching a **rank statistics**

$$L_{area} = \| \, \mathrm{vecsort}(\mathbf{m}) - \mathbf{r}_a \|^2$$

$\mathbf{m}$



→ vectorize → sort →

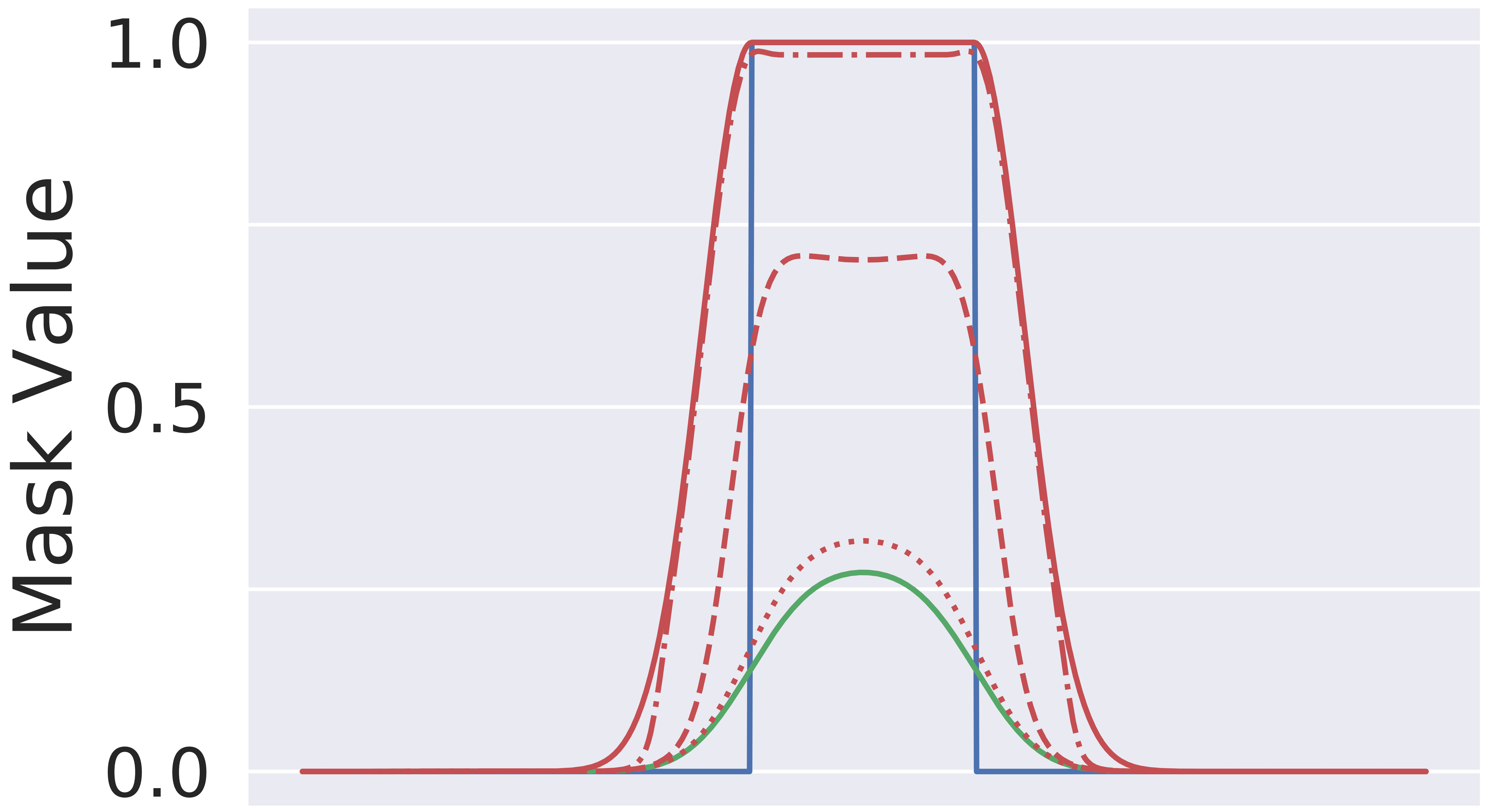


$\mathbf{r}_\alpha$

$\mathrm{vecsort}(\mathbf{m})$

subject to $\mathrm{area}(\mathbf{m}) = a$

Smooth masks



$m(v)$: mask

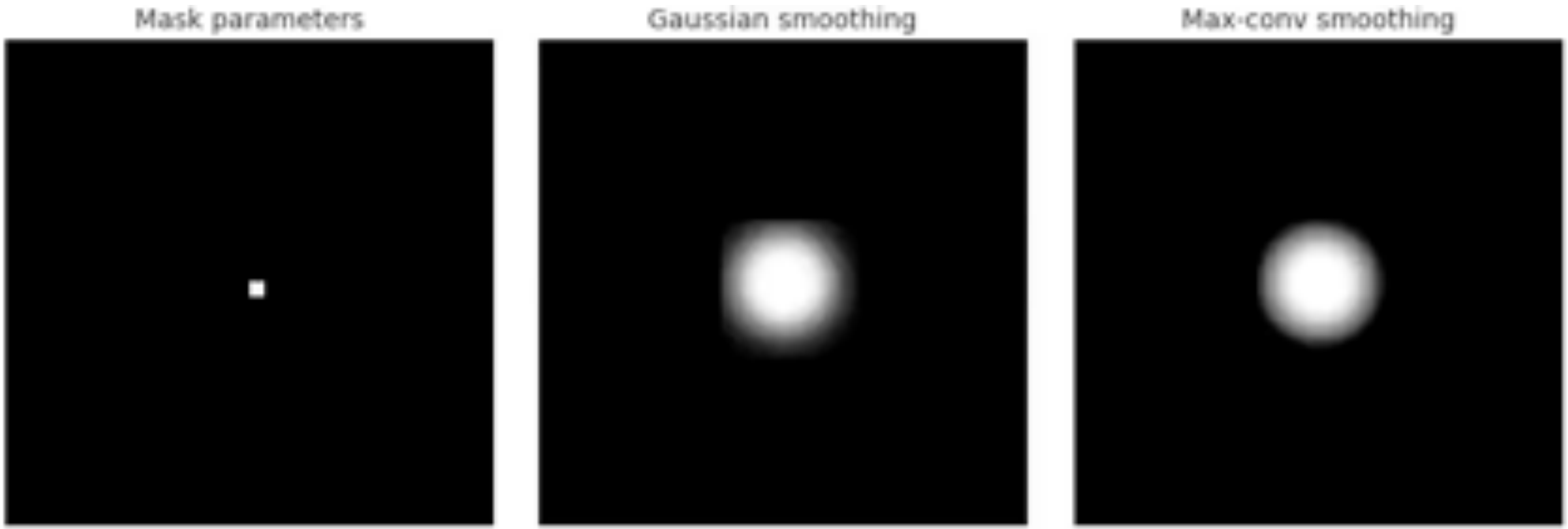
$$\mathrm{conv}(u; m; k) = \frac{1}{Z} \sum_{v \in \Omega} k(u - v) m(v)$$

$$\mathrm{maxconv}(u; m; k) = \max_{v \in \Omega} k(u - v) m(v)$$

$$\mathrm{smoothconv}(u; m; k; T) = \mathrm{smax}_{v \in \Omega; T}\, k(u - v) m(v)$$

$$\mathrm{smax}_{u \in \Omega; T}\, f(u) = \frac{\sum_u f(u) \exp(f(u)/T)}{\sum_u \exp(f(u)/T)}$$

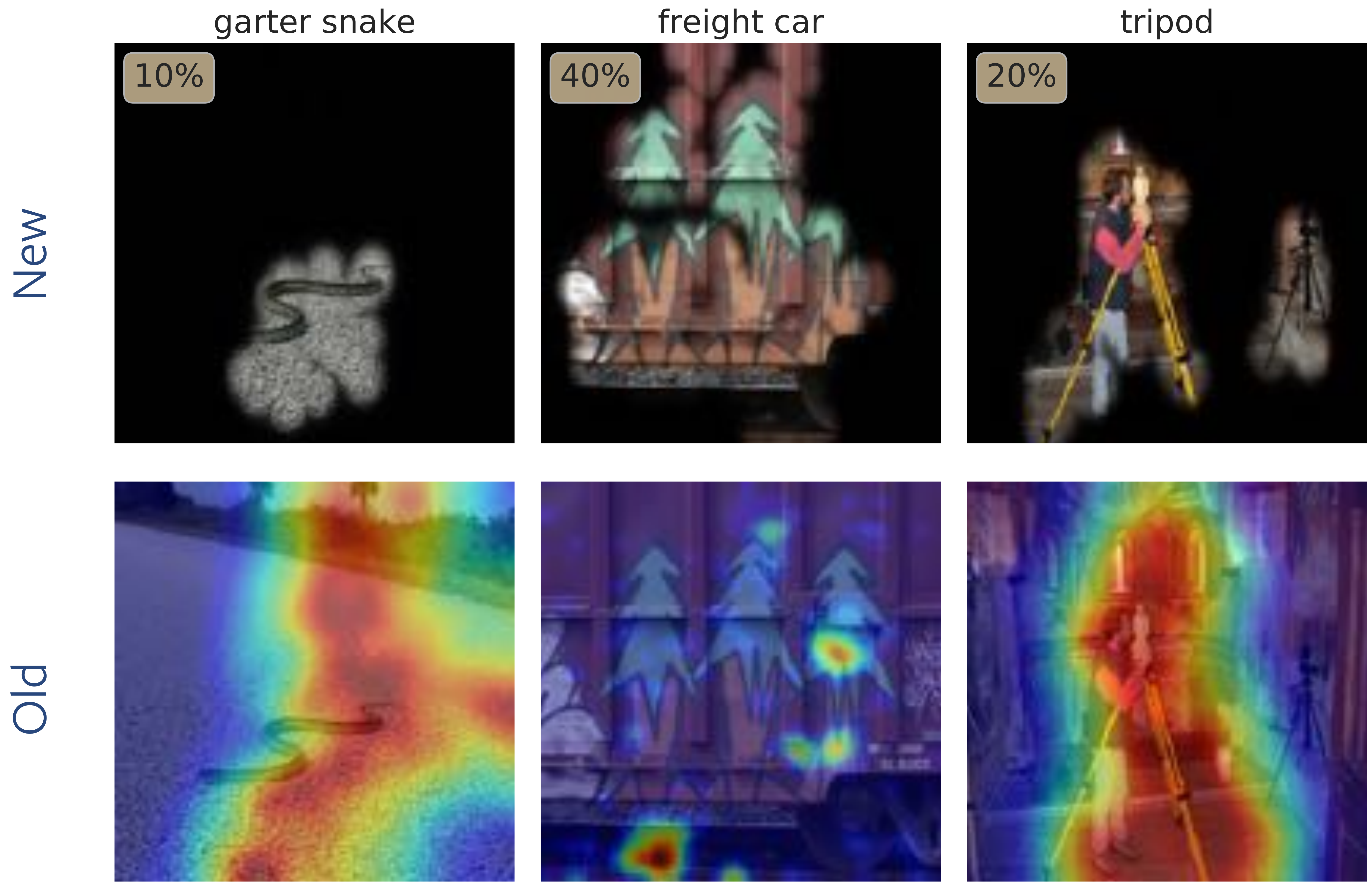
Smooth masks



Comparison with prior work on “meaningful perturbations”

Compared to **Fong and Vedaldi, 2017**, we remove all regularization terms in the energy term.

Our innovations result in a method that’s more **principled**, **stable**, and **sensitive**.



Algorithm

1. Pick an area a

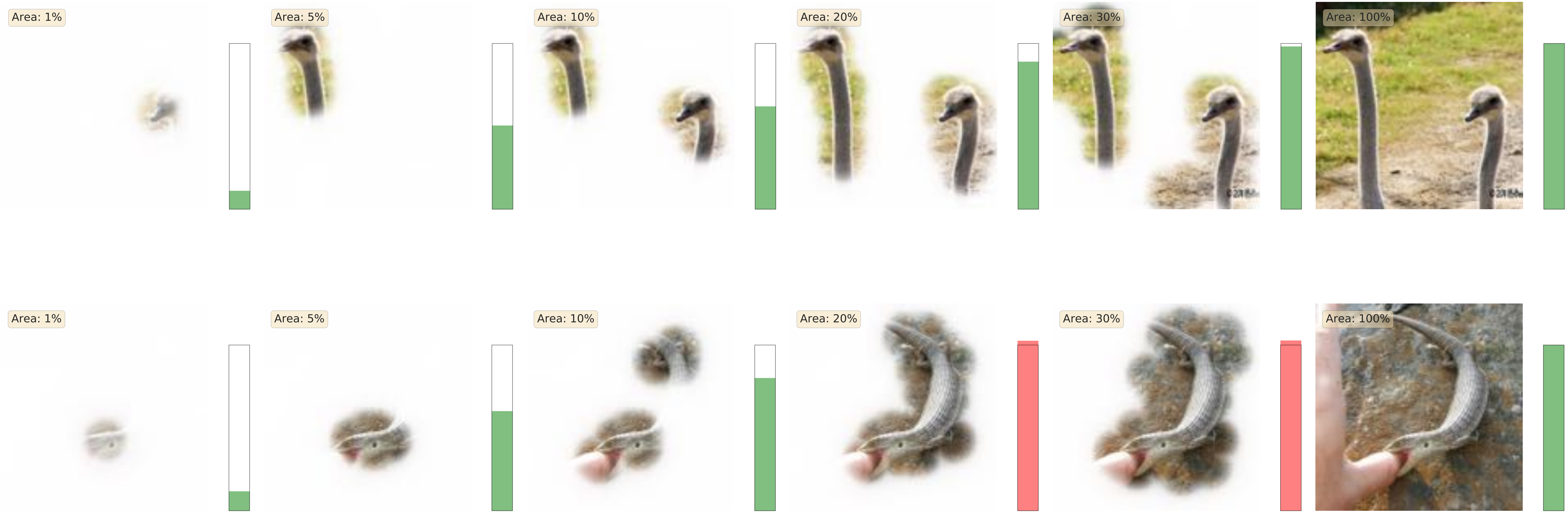
2. Use SGD to solve the optimization problem for a large λ:

$$\underset{\mathbf{m}}{\operatorname{argmax}}\ \Phi(\mathrm{smooth}(\mathbf{m}) \otimes \mathbf{x}) - \lambda \| \, \mathrm{vecsort}(\mathrm{smooth}(\mathbf{m})) - \mathbf{r}_a \|^2$$

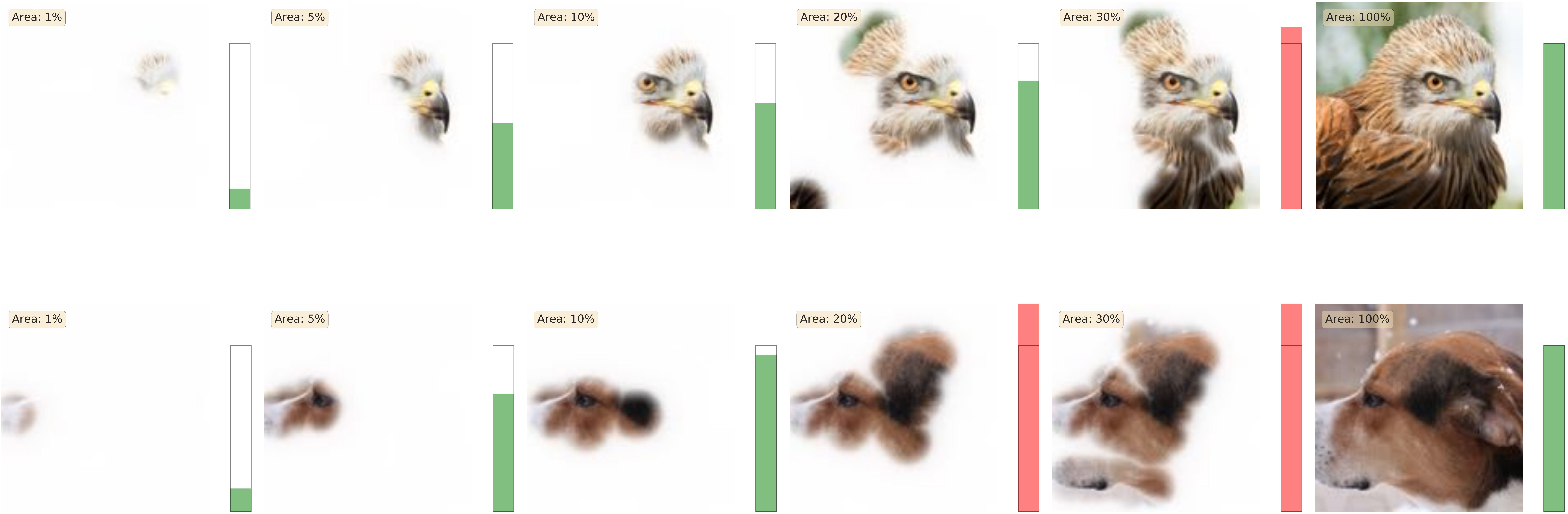
3. If needed, sweep a and repeat

Results

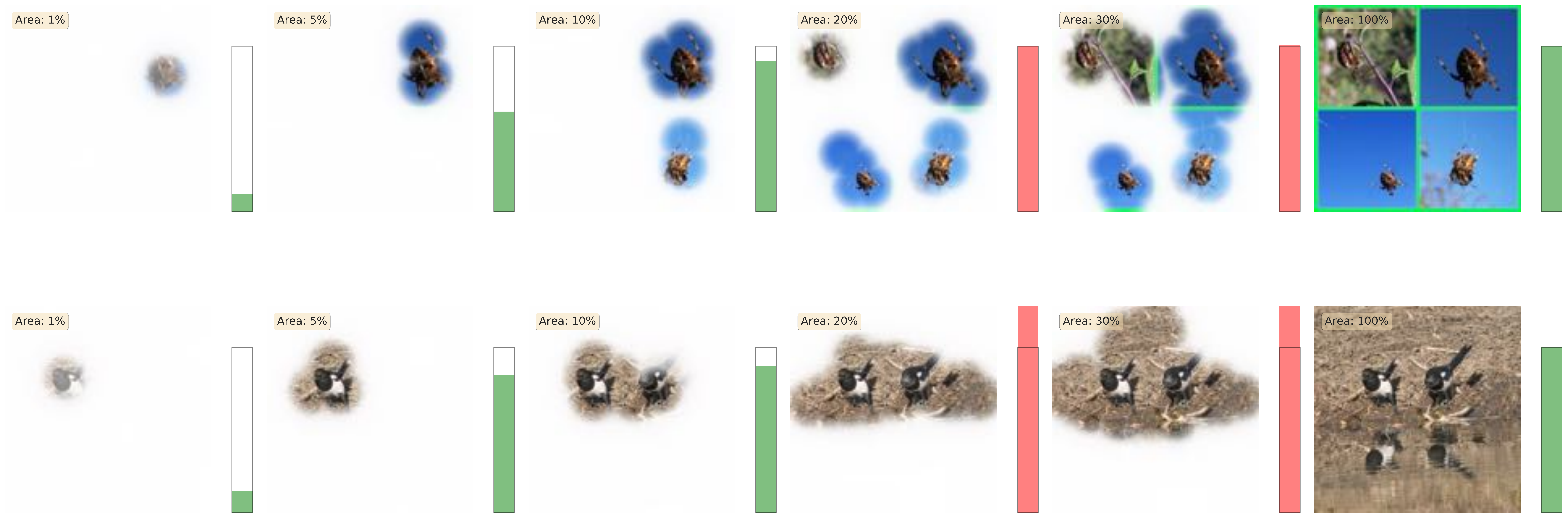
Foreground evidence is usually sufficient



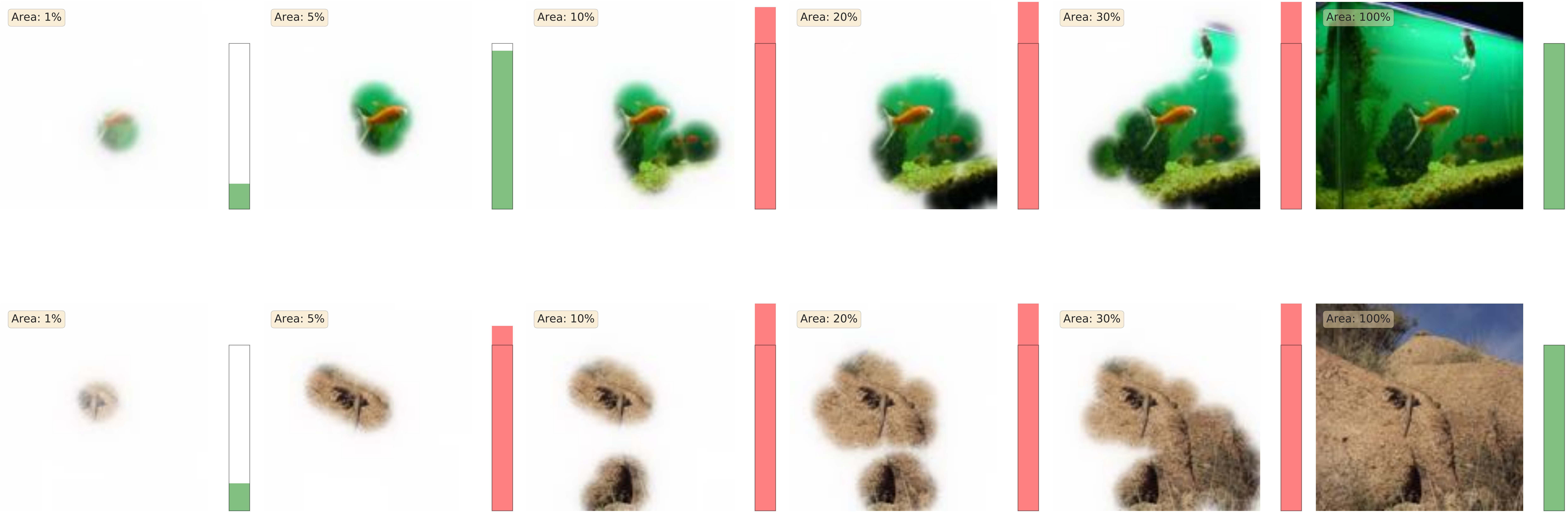
Large objects are recognised by their details



Small objects contribute cumulatively

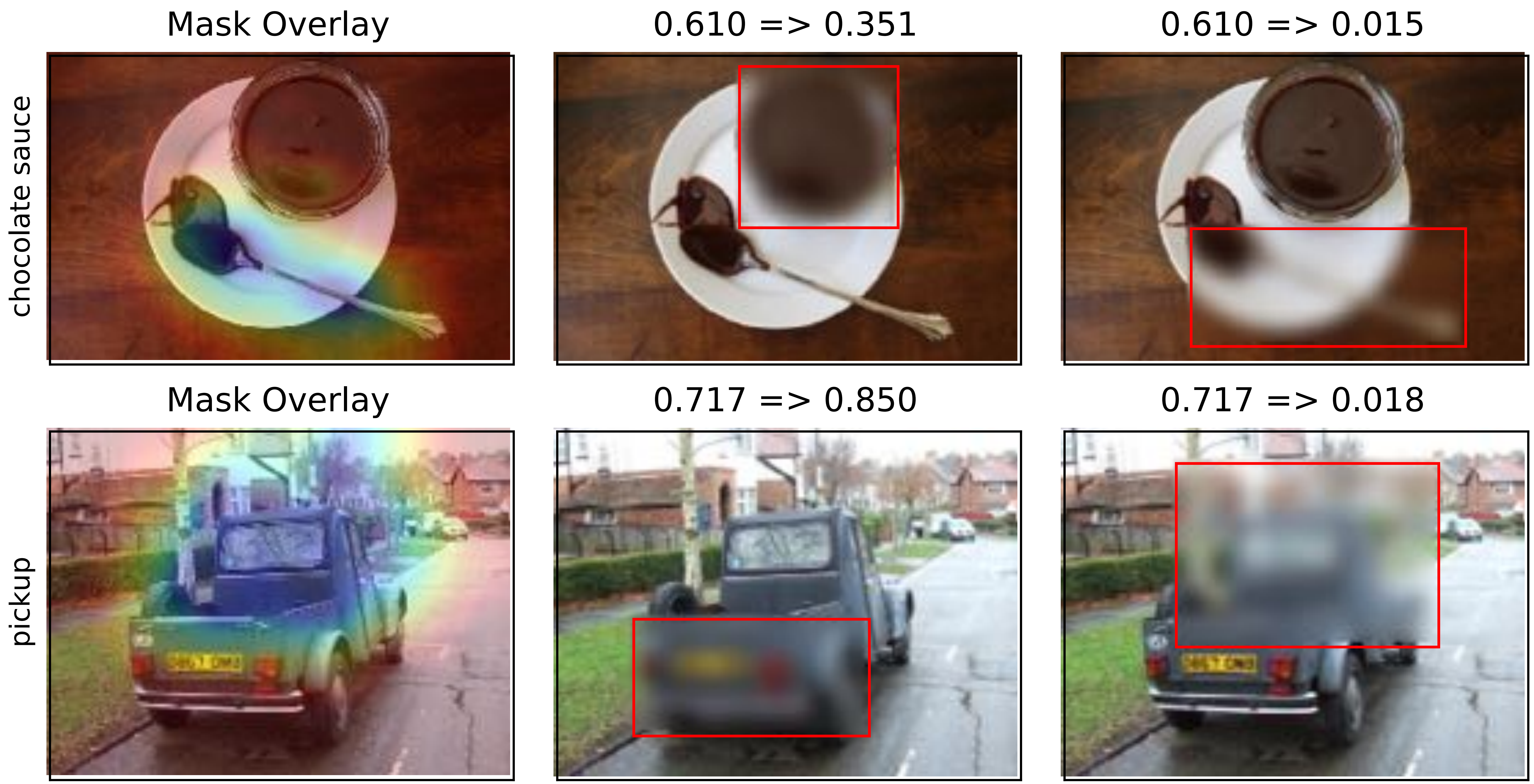


Suppressing the background may overdrive the network



Diagnosing networks

Example: the hot chocolate is recognized via the spoon and the truck vs the license plate



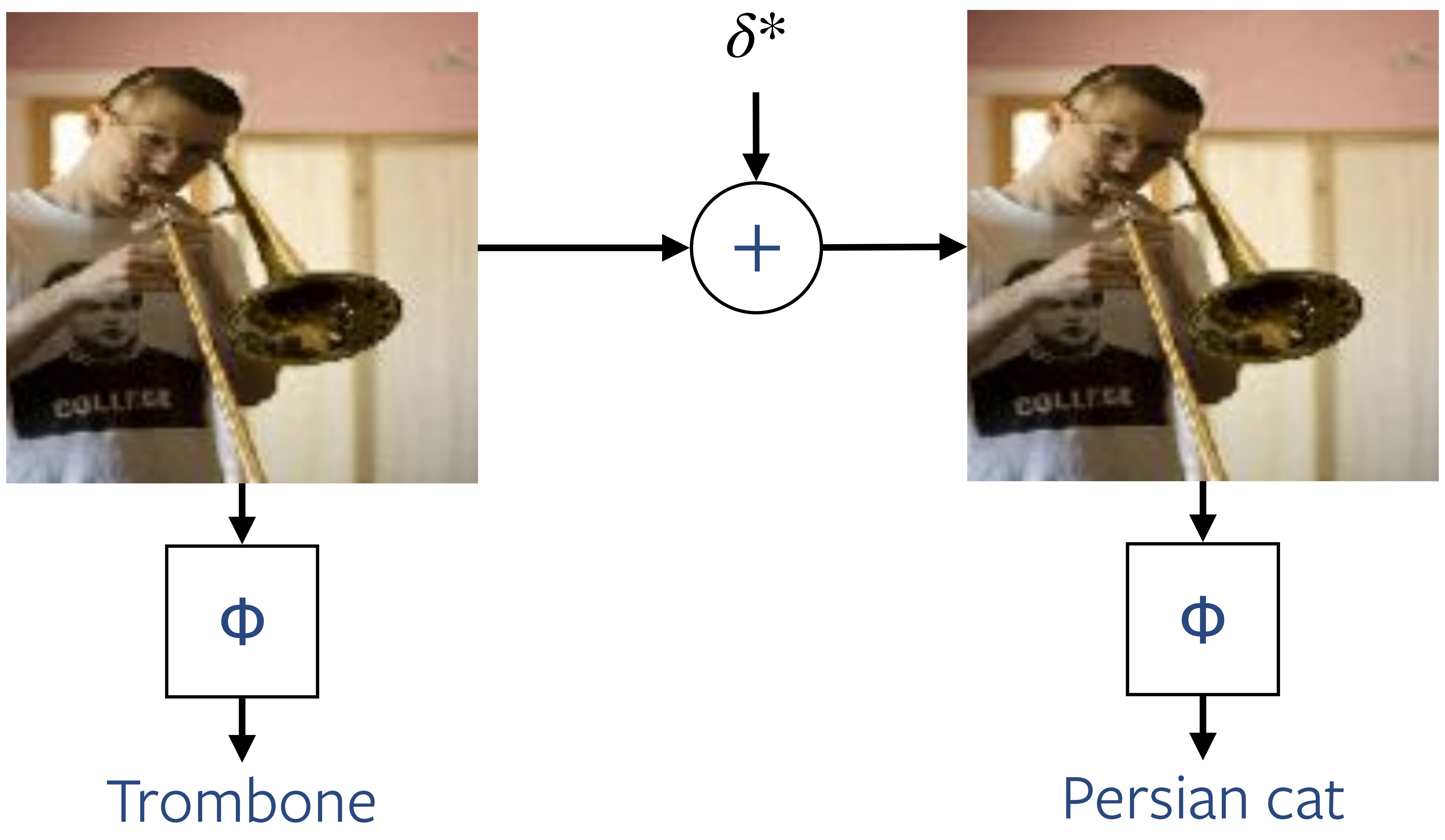
facebook Artificial Intelligence
UNIVERSITY OF OXFORD

CNN fragility

Let $\mathbf{y} = \Phi(\mathbf{x})$ be the label predicted for image $\mathbf{x}$ by the deep net

Empirically, we can find tiny perturbations $\mathbf{x} + \delta$ that change $\mathbf{y}$ arbitrarily

$$\delta^* = \underset{||\delta||<\epsilon}{\operatorname{argmin}} ||\mathbf{y}_{\text{arbitrary}} - \Phi(\mathbf{x} + \delta)||$$

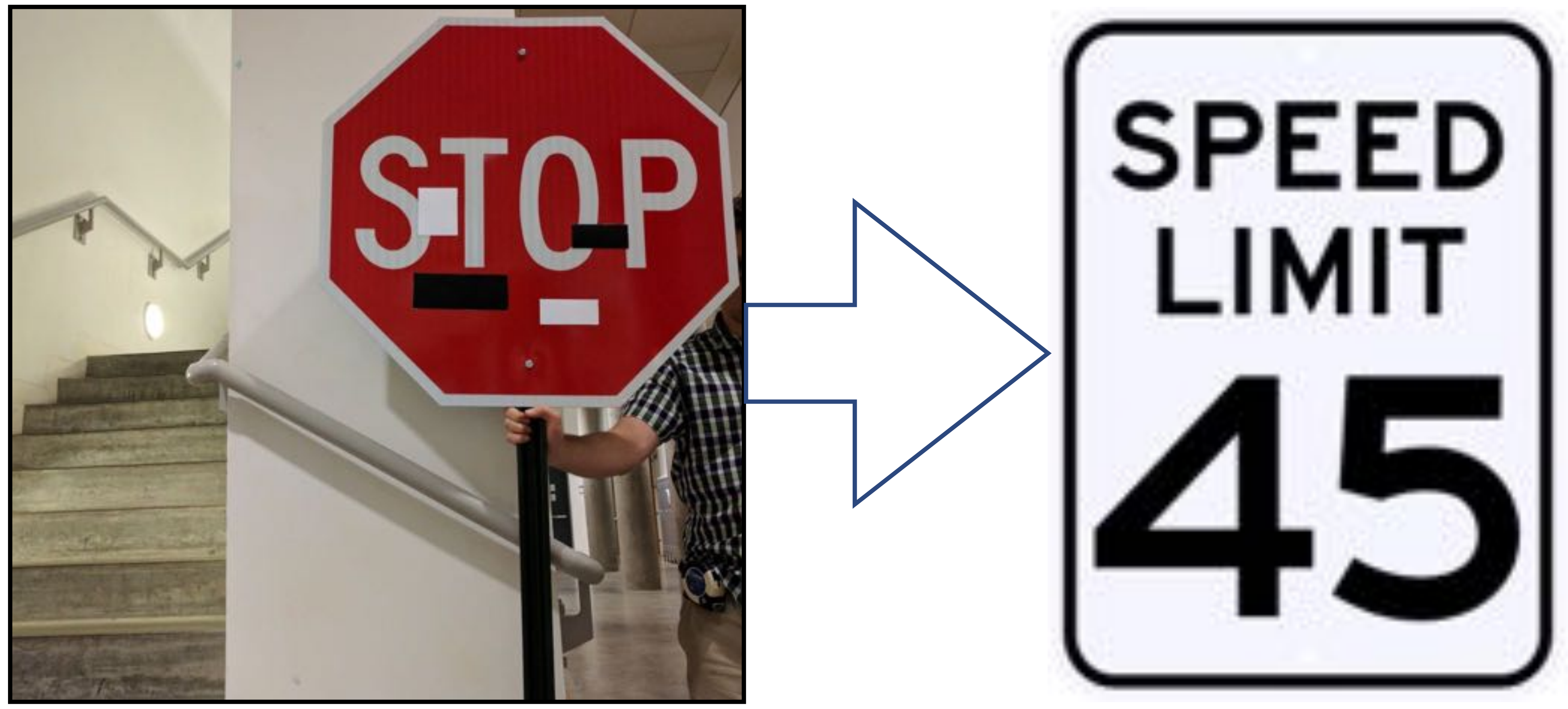


Intriguing properties of neural networks
Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, Fergus. CoRR, 2013

Dangerous adversaries

Adversarial glasses fooling face recognition

Adversarial stickers fooling sign recognition



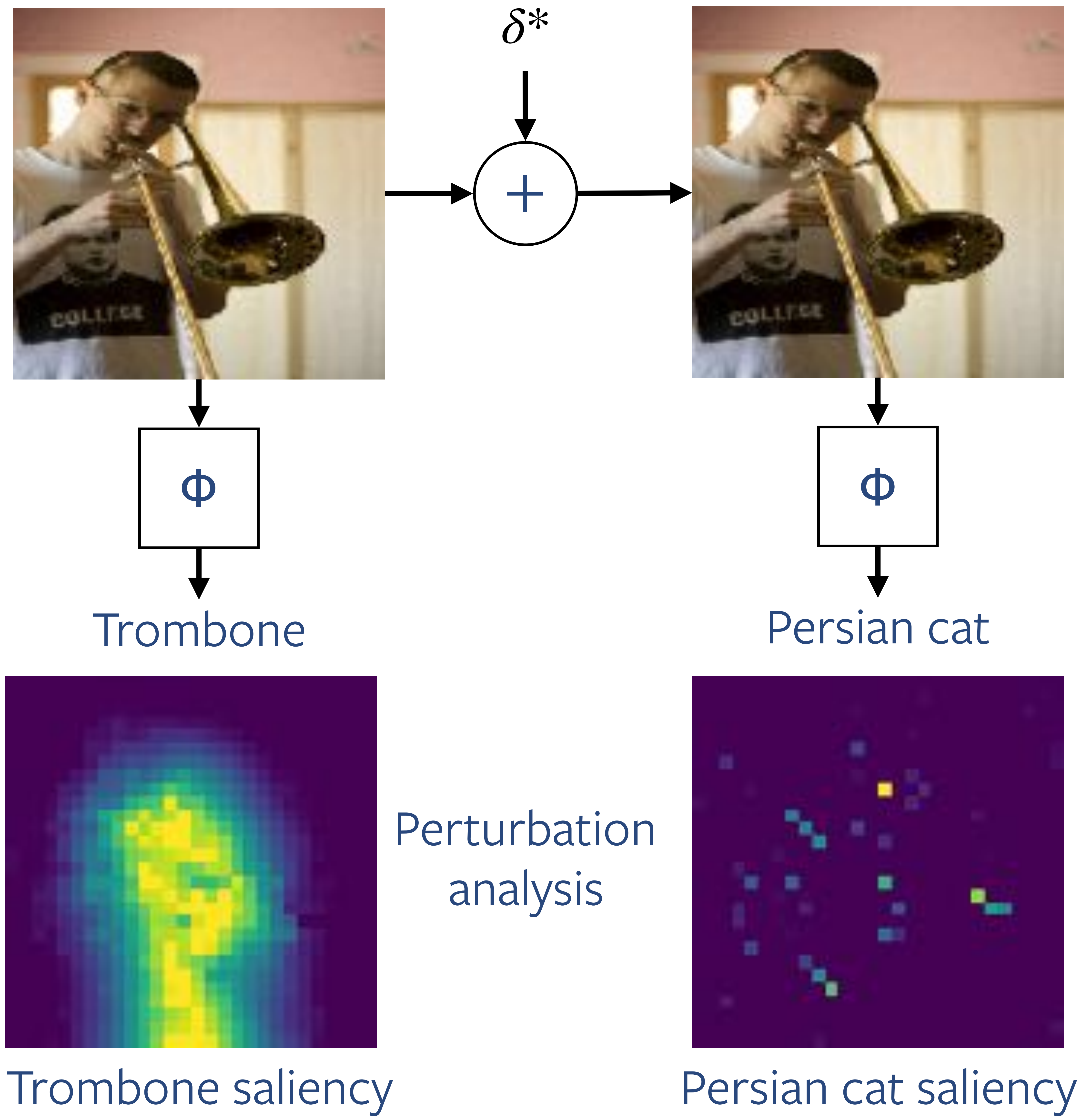
Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. Sharif, Bhagavatula, Bauer, Reiter. Proc. CSS, 2016.

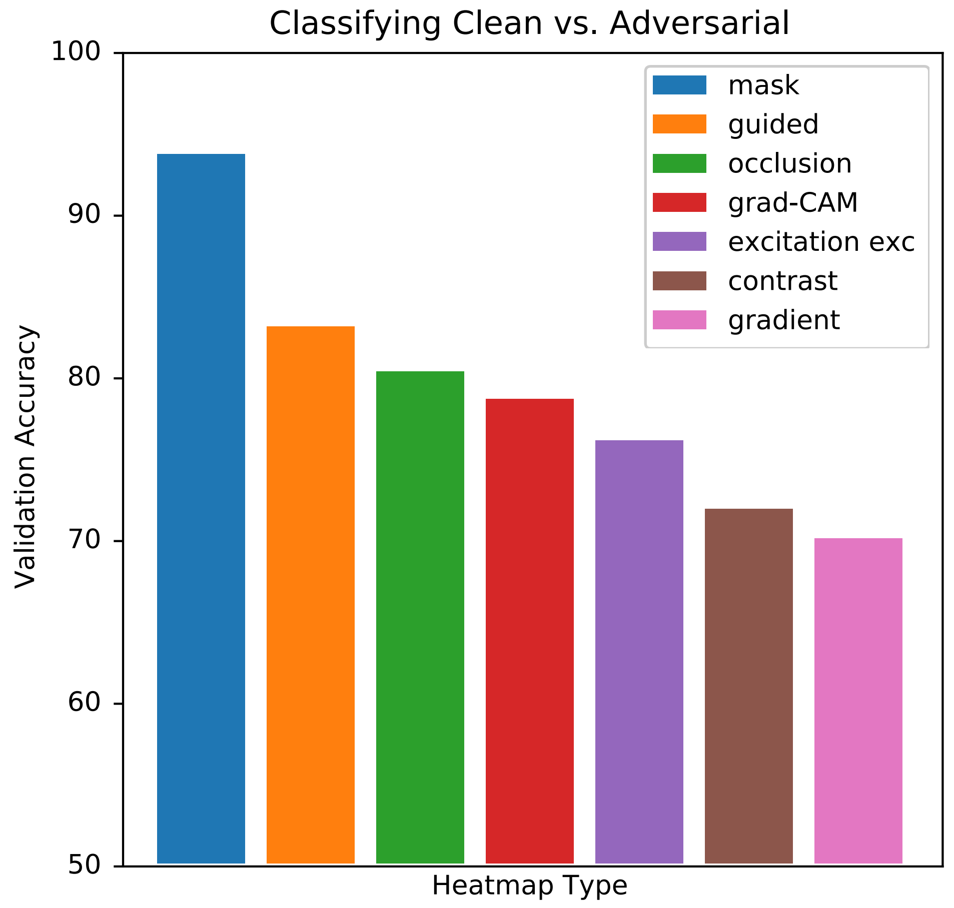
Robust physical-world attacks on machine learning models. Evtimov, Kevin Eykholt, Li, Prakash, Rahmati, Song. arXiv, 2017.

Adversarial defence

Method: recognize genunie vs adversarial images by learning a classifier on top of the saliency maps

(Illustrative of attribution, not really a recommended defence strategy!)





Assessing attribution

Assessing attribution: pointing game & weak localisation

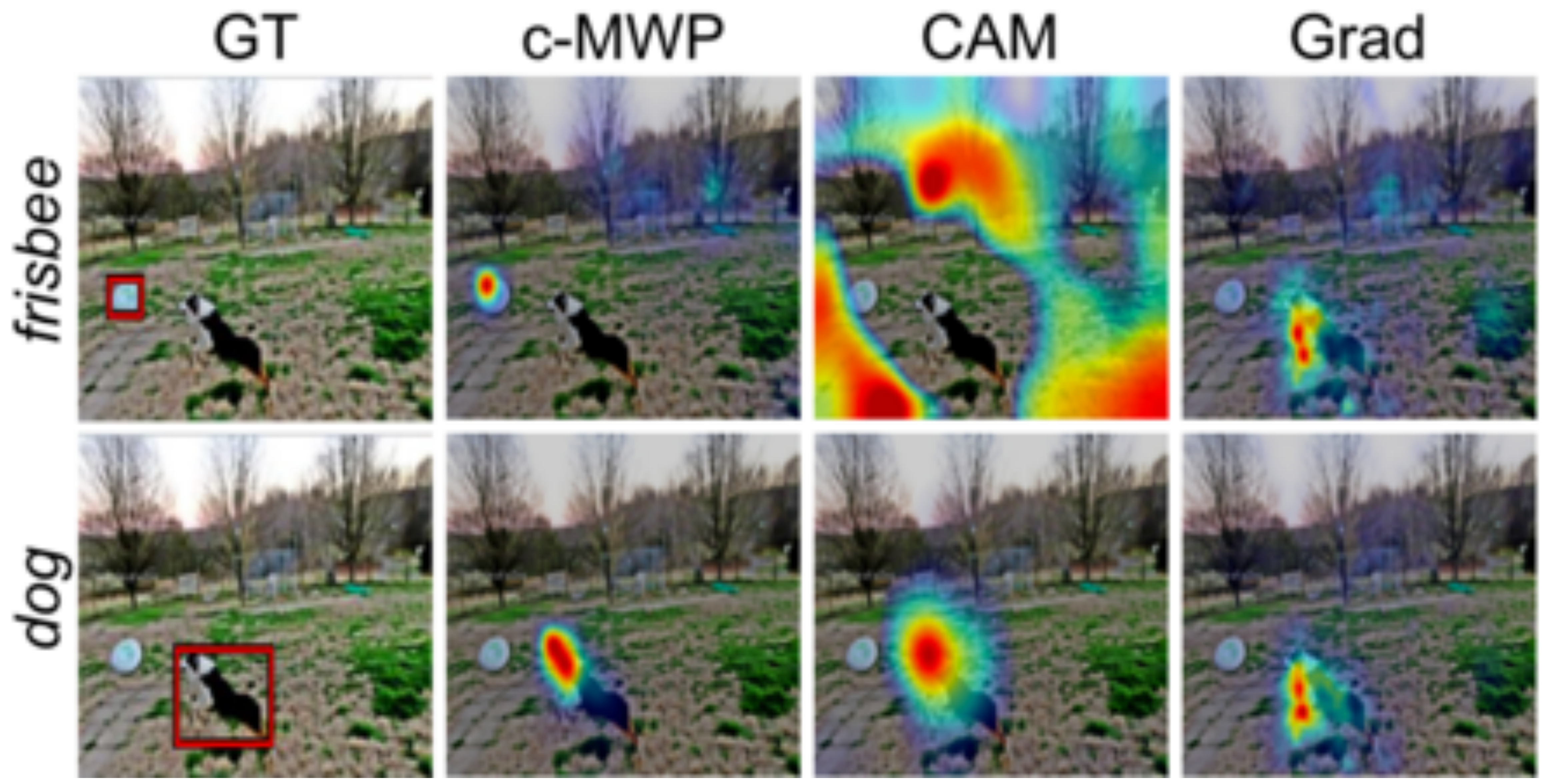
Goal: measure the spatial correlation between attribution maps and object occurrences

If the correlation is strong:

- the diagnosed model “understand” the object **and**
- the attribution method can tell

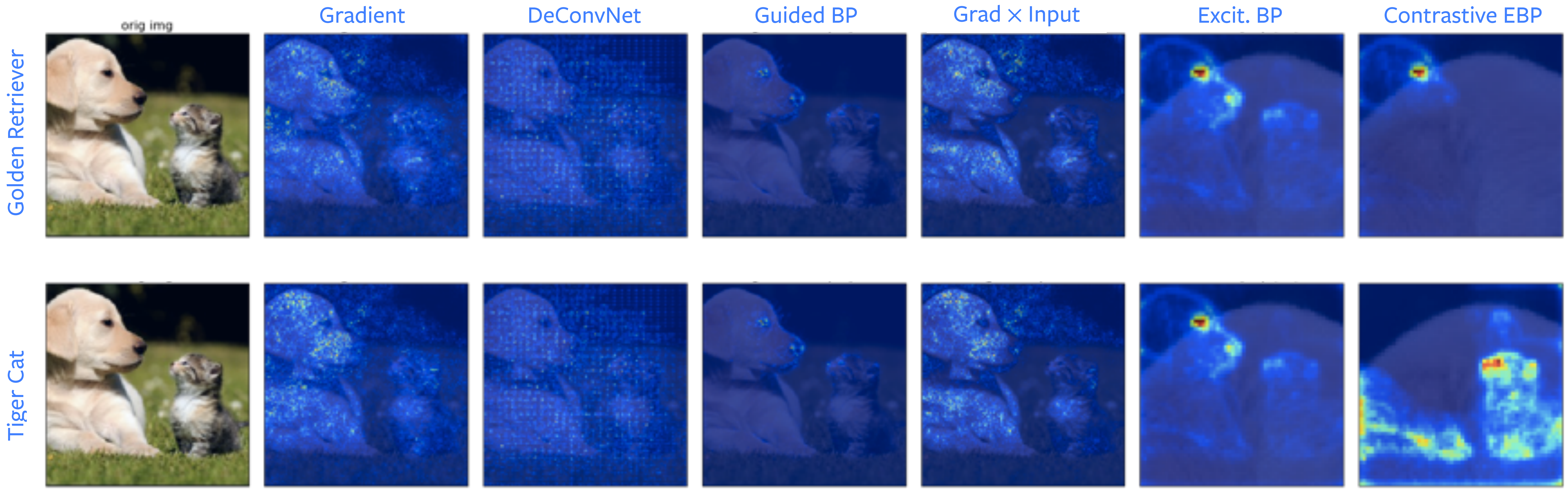
However, if the correlation is poor, *either*:

- the diagnoses model does not understand the object **or**
- the attribution method fails to tell



Assessing attribution: neuron sensitivity

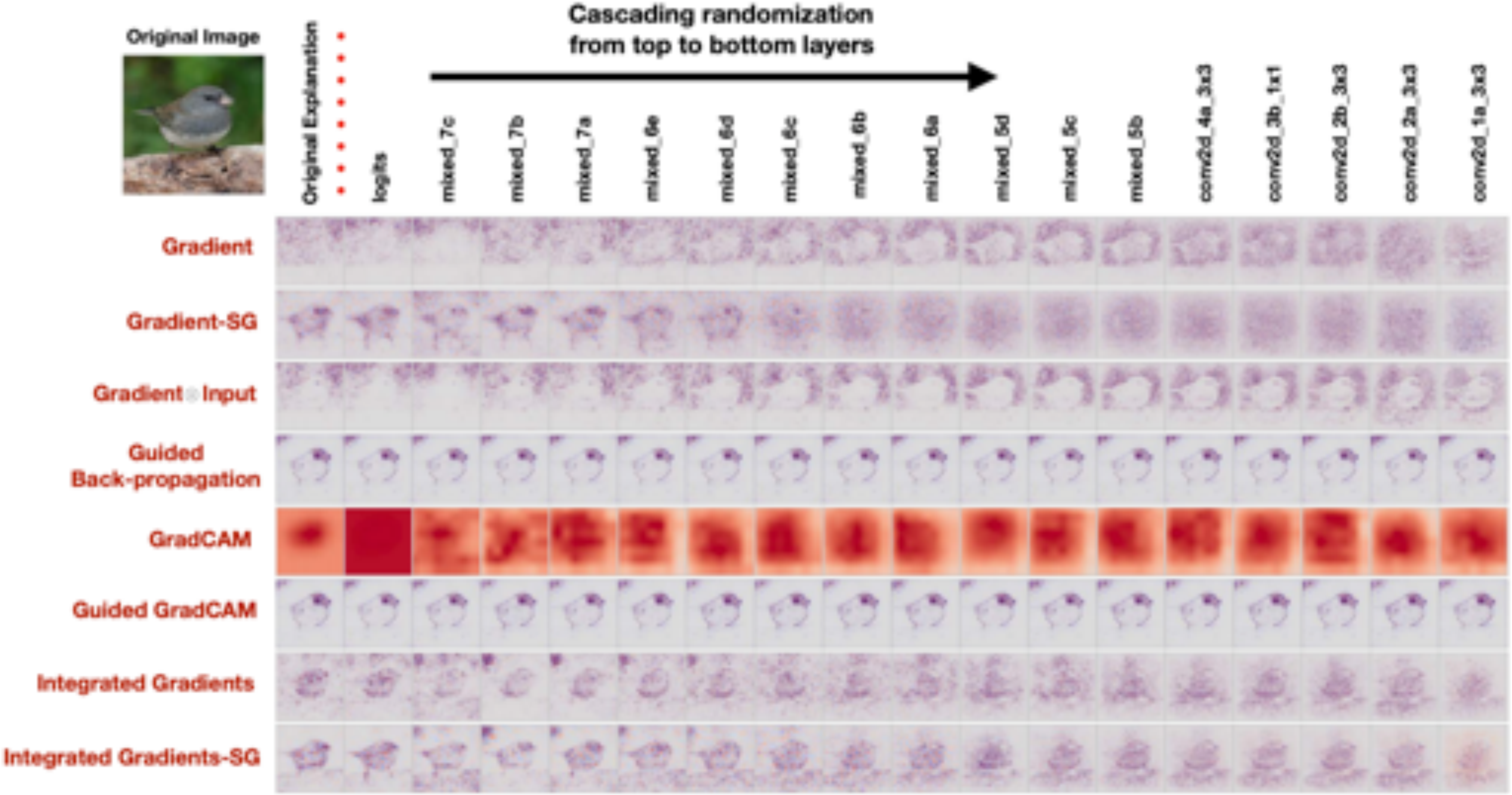
Attribution should generally result in a different output depending on which neon one wishes to visualise.



Assessing attribution: parameter sensitivity

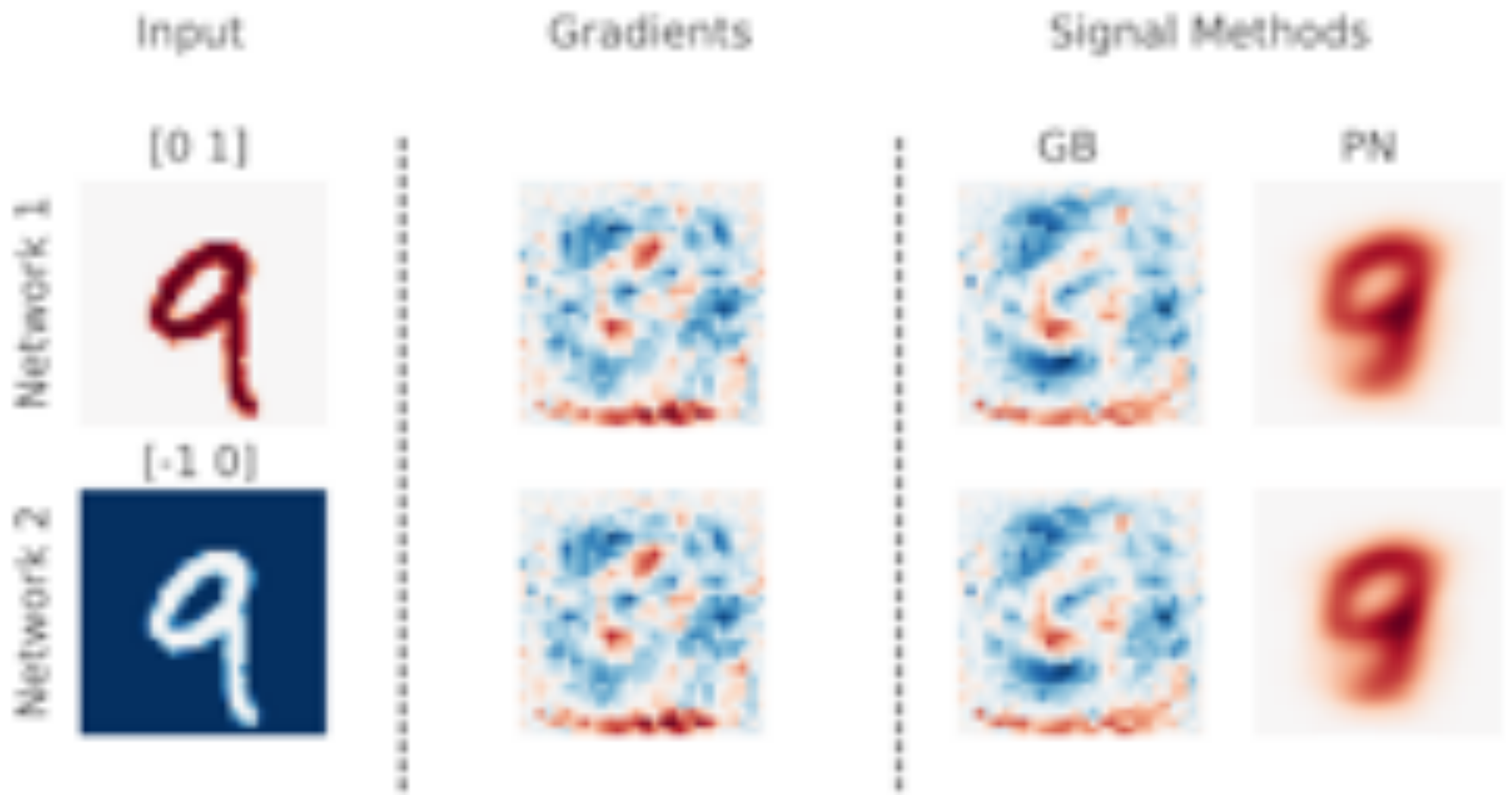
Attribution should also produce a different output if the model weights are different — e.g. random

Sanity checks for saliency maps.
Adebayo, Gilmer, Muelly, Goodfellow, Hardt, Kim. Proc. NeurIPS, 2018.



Assessing attribution: shift invariance

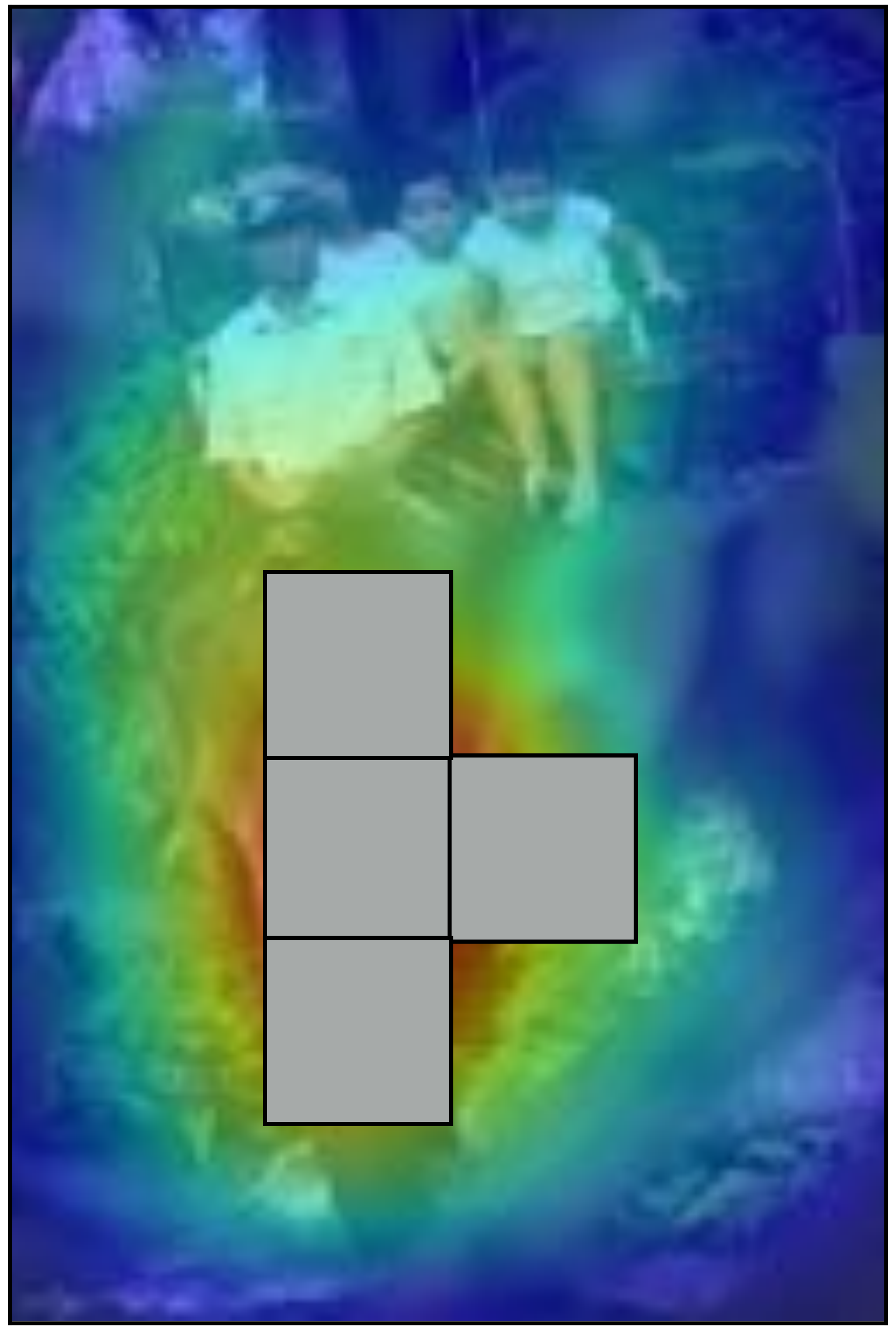
Learning how to explain neural networks: PatternNet and PatternAttribution. Kindermans, Schütt, Alber, Müller, Erhan, Kim, Dähne. Proc. ICLR, 2018.
Making convolutional networks shift-invariant again. Zhang. Proc. ICML, 2019.



Assessing attribution: perturbation analysis

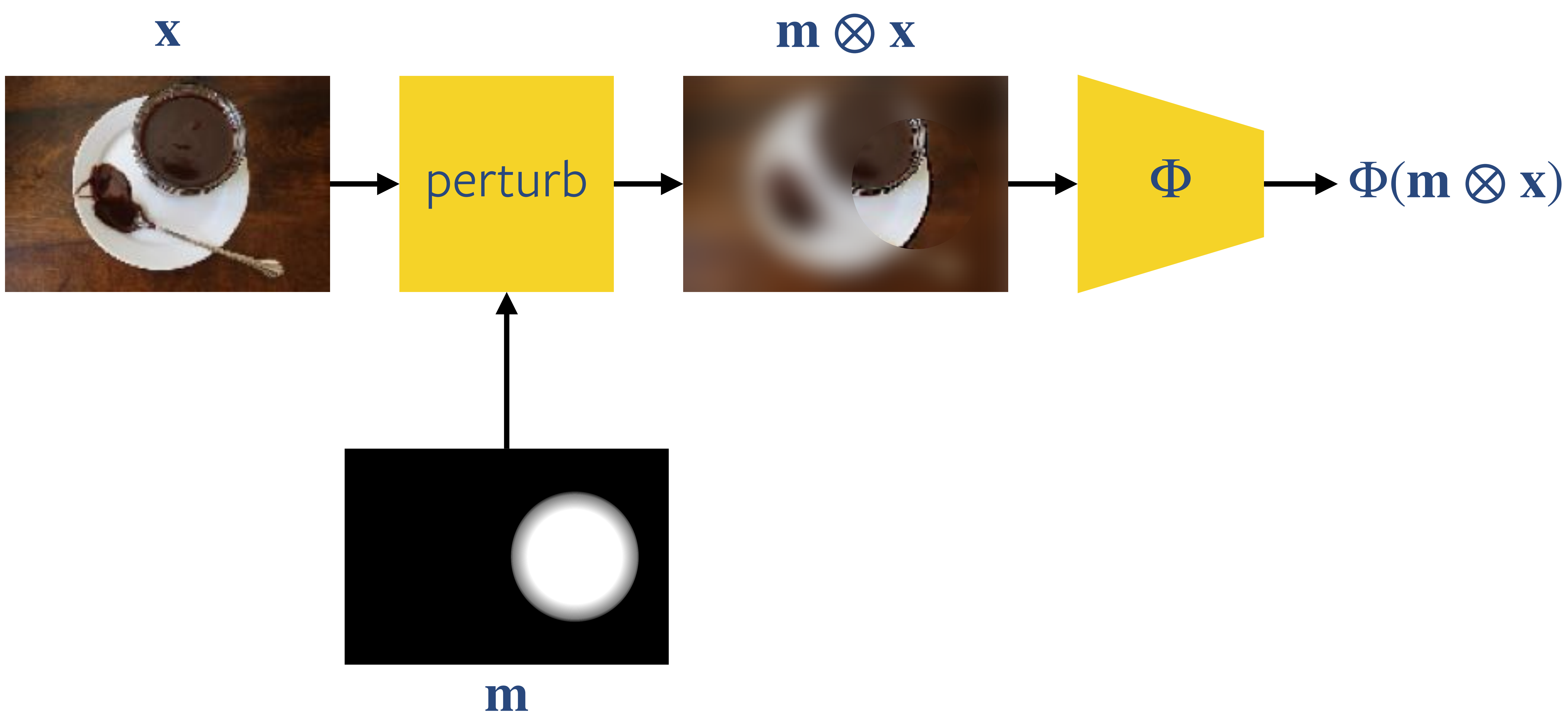
Display

elephant

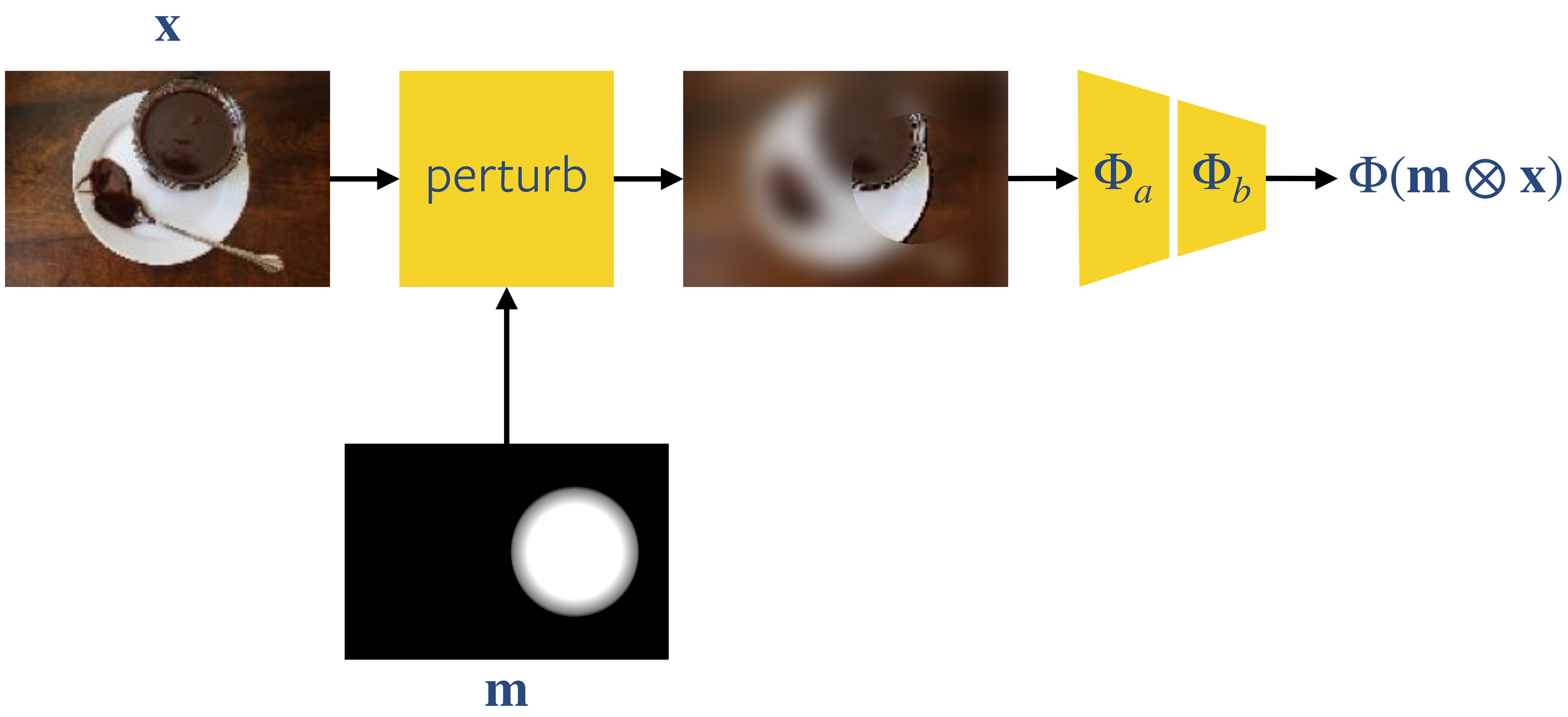


Attributing channels at intermediate layers

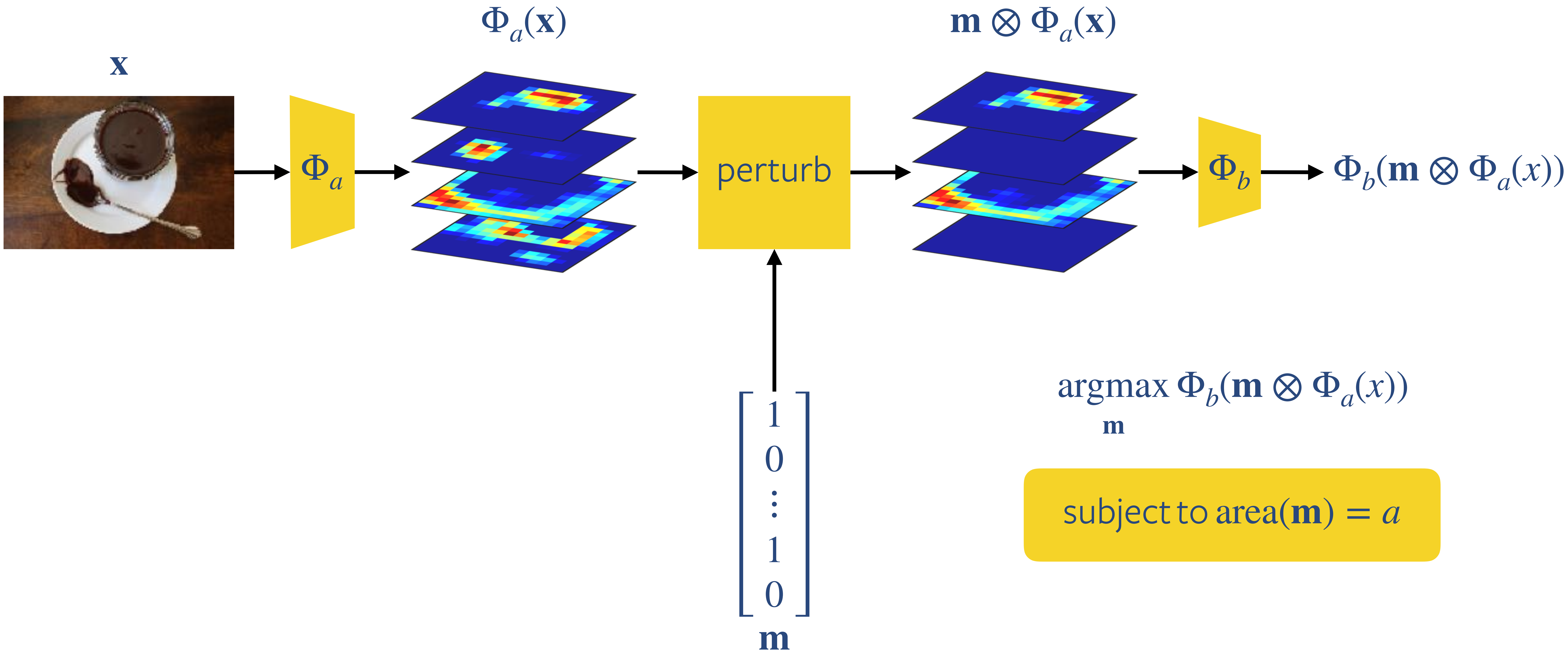
Spatial attribution



Channel attribution



Channel attribution



$$\underset{\mathbf{m}}{\operatorname{argmax}}\ \Phi_b(\mathbf{m} \otimes \Phi_a(x))$$

subject to area$(\mathbf{m}) = a$

Activation "diffing"

$\sum \mathbf{m} \otimes \Phi_a(x)$

Ibizan hound

Original

$\Phi_a(x)$

Perturbed

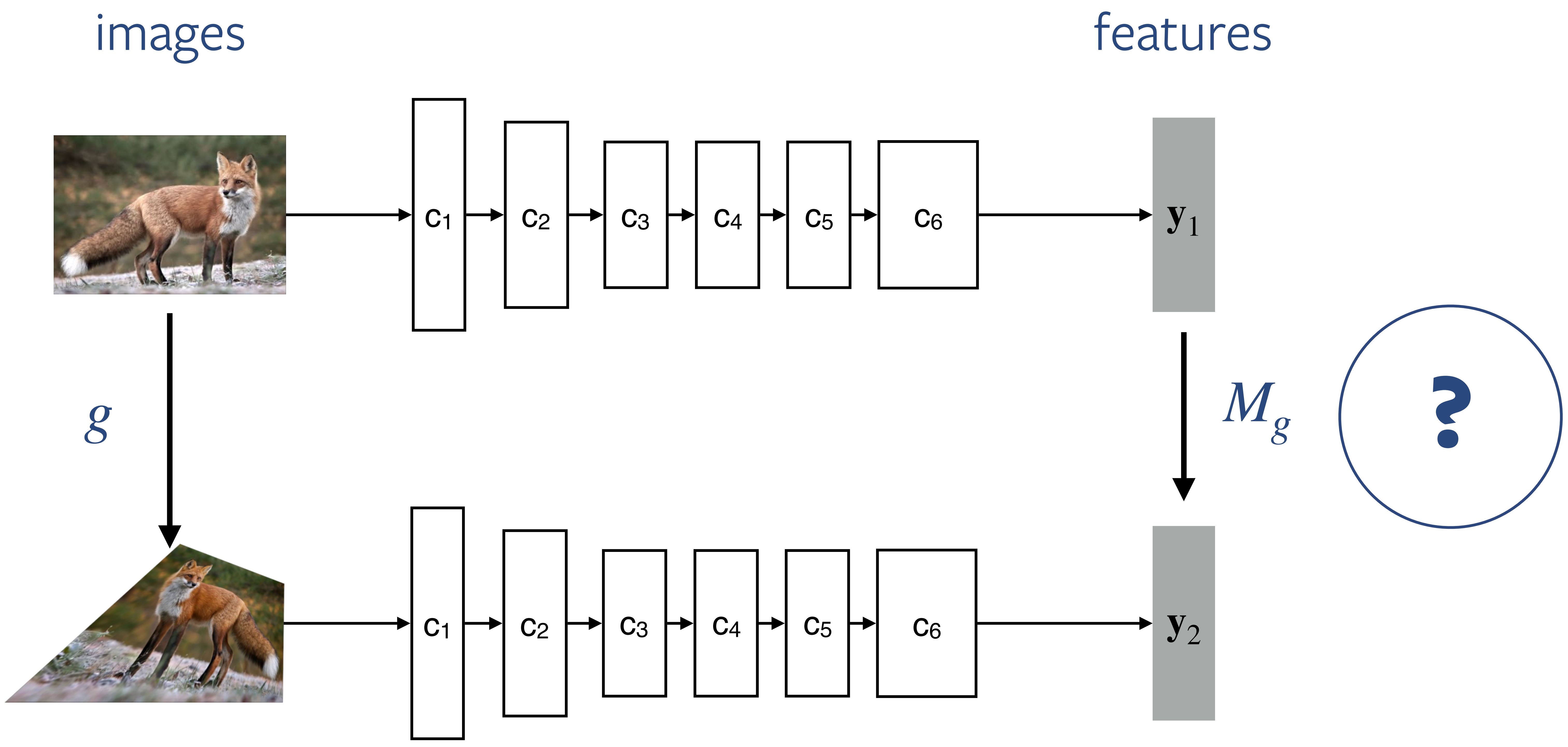
$\mathbf{m} \otimes \Phi_a(x)$

[Olah et al., Distill 2017]

Equivariance

Short answer: warping image usually reduces to sparse linear tf in feature space.

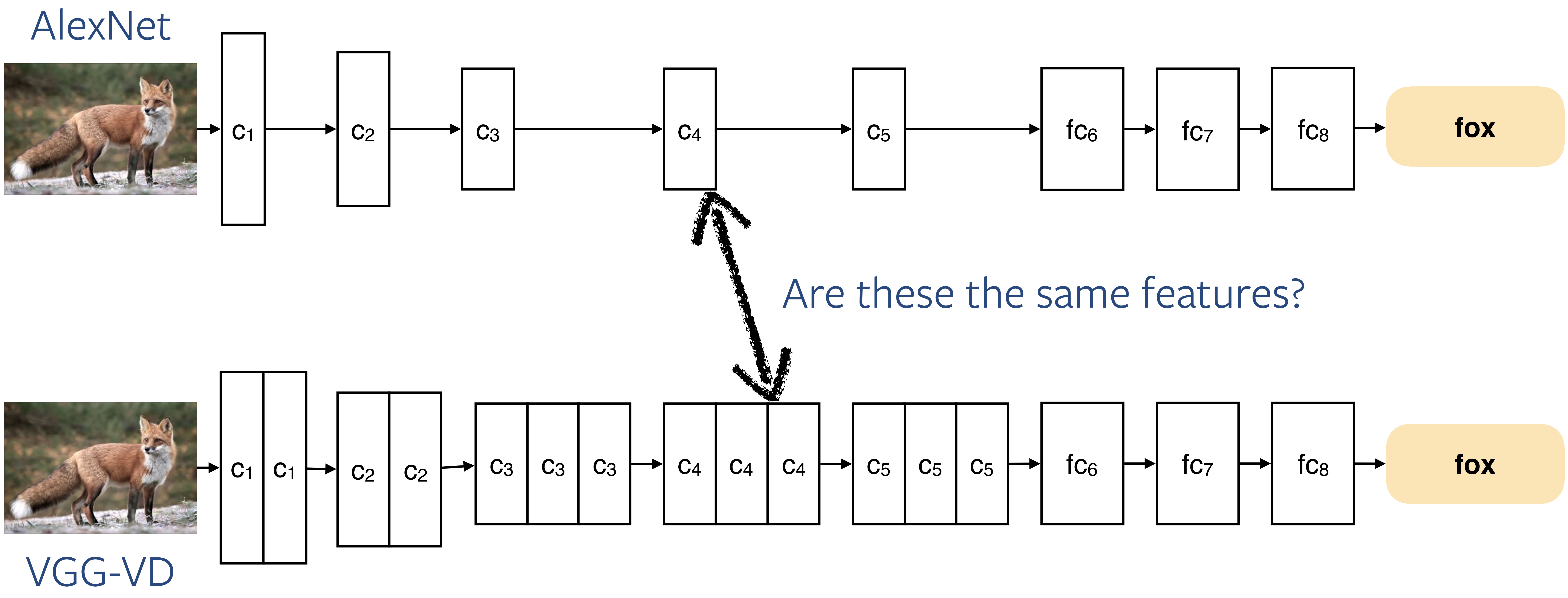
Long answer:
Understanding image representations by measuring their equivariance and equivalence. Lenc Vedaldi. CVPR 2015 & IJCV 2018



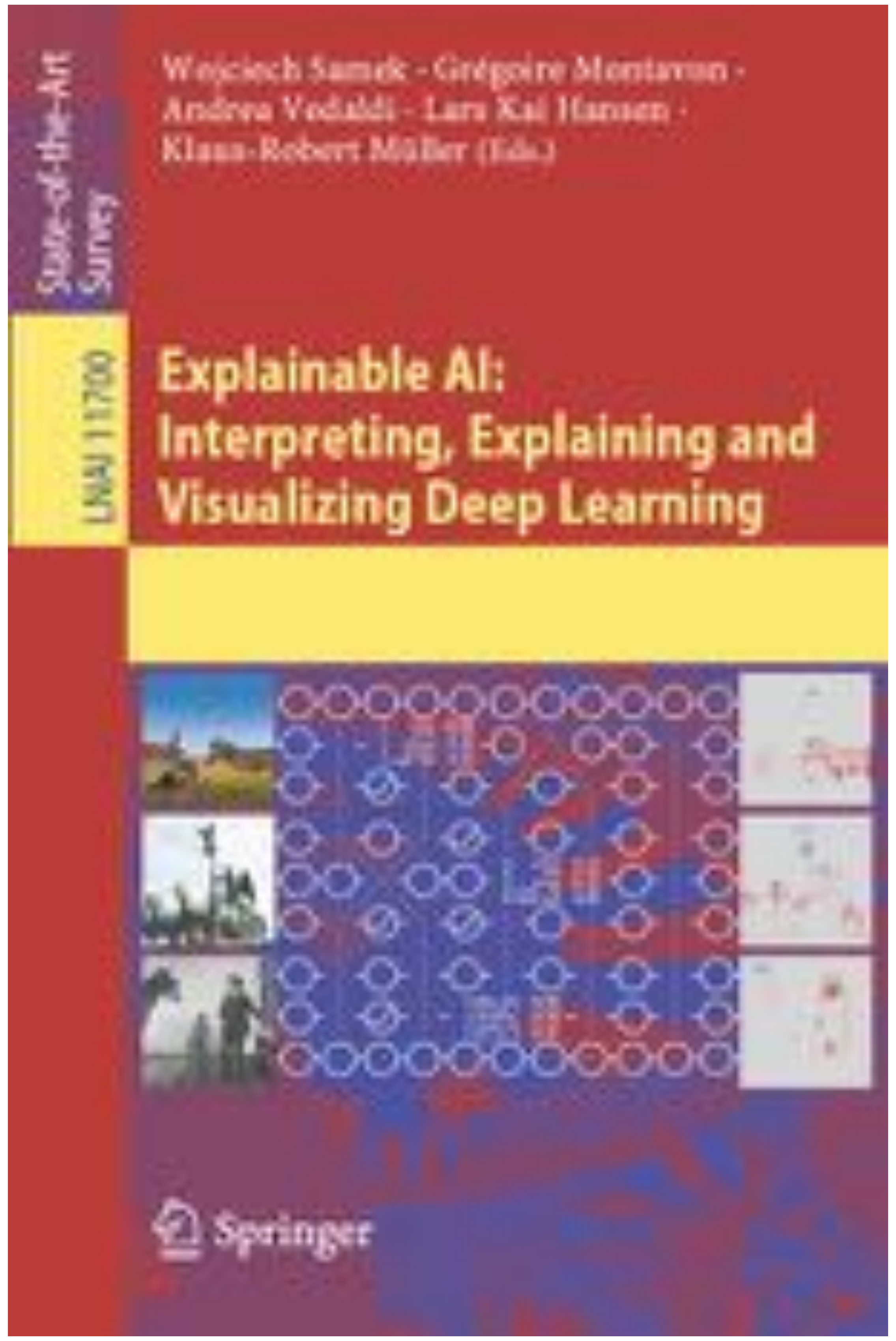
Equivalence

Short answer: there generally are corresponding features in different networks (up to 1x1 linear tfs).

Long answer
Understanding image representations by measuring their equivariance and equivalence. Lenc Vedaldi. CVPR 2015 & IJCV 2018



Collected references



Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Samek, Montavon, Vedaldi, Hansen, Muller, editors. Springer, 2019

Software

Captum

https://pytorch.org/captum/

More than just vision

TorchRay

https://github.com/facebookresearch/TorchRay

Attribution, reproducibility, benchmarks

Summary

Generating conic examples

- Inversion vs activation maximization
- The importance of the prior / regularizer
- Aesthetic vs diagnostic

Attribution

- (Modified) gradient backpropagation
- Excitation and relevance backpropagation
- Meaningful perturbation analysis
- Understanding via approximating models