Representations in brains and machines

Matthias Bethge
MLSS 2016 Cadiz

http://bethgelab.org

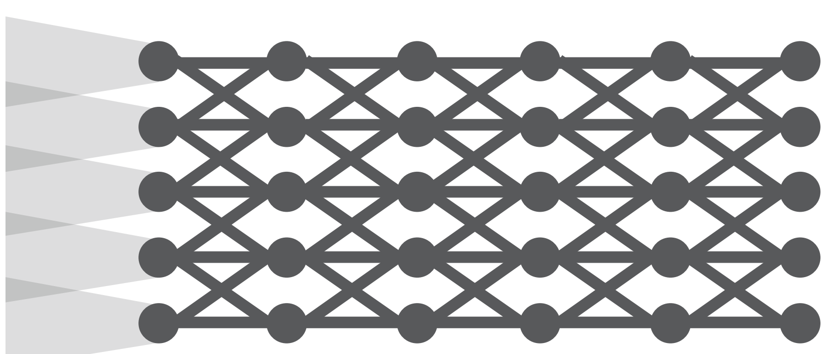
data analysis

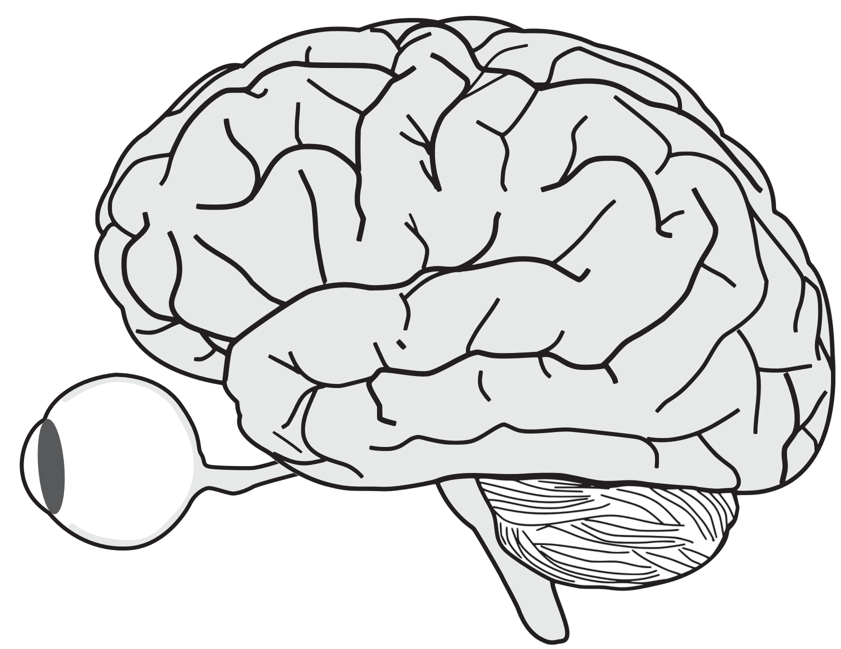
understanding intelligent systems

Machine Learning

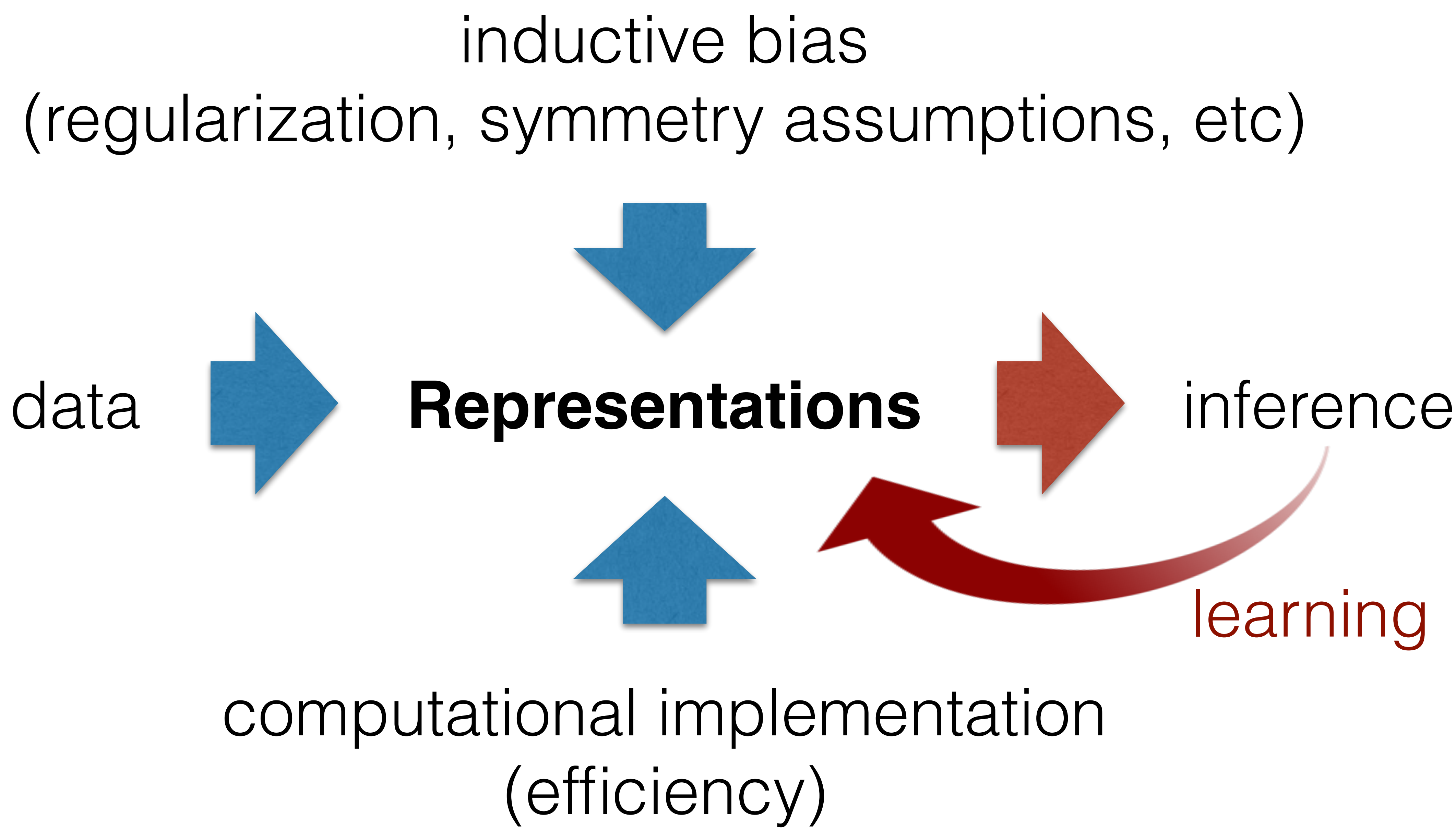
Neuroscience

inspiration

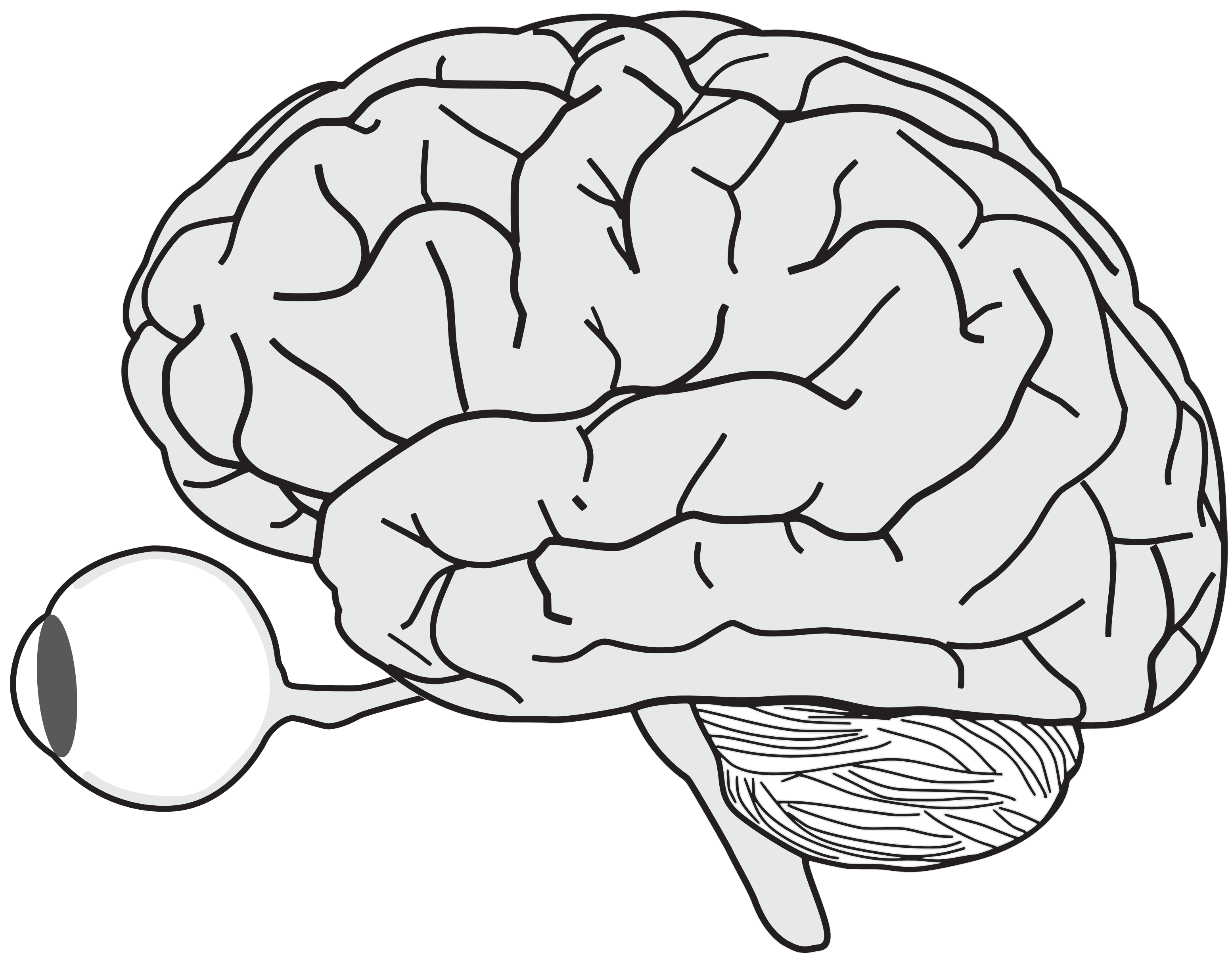




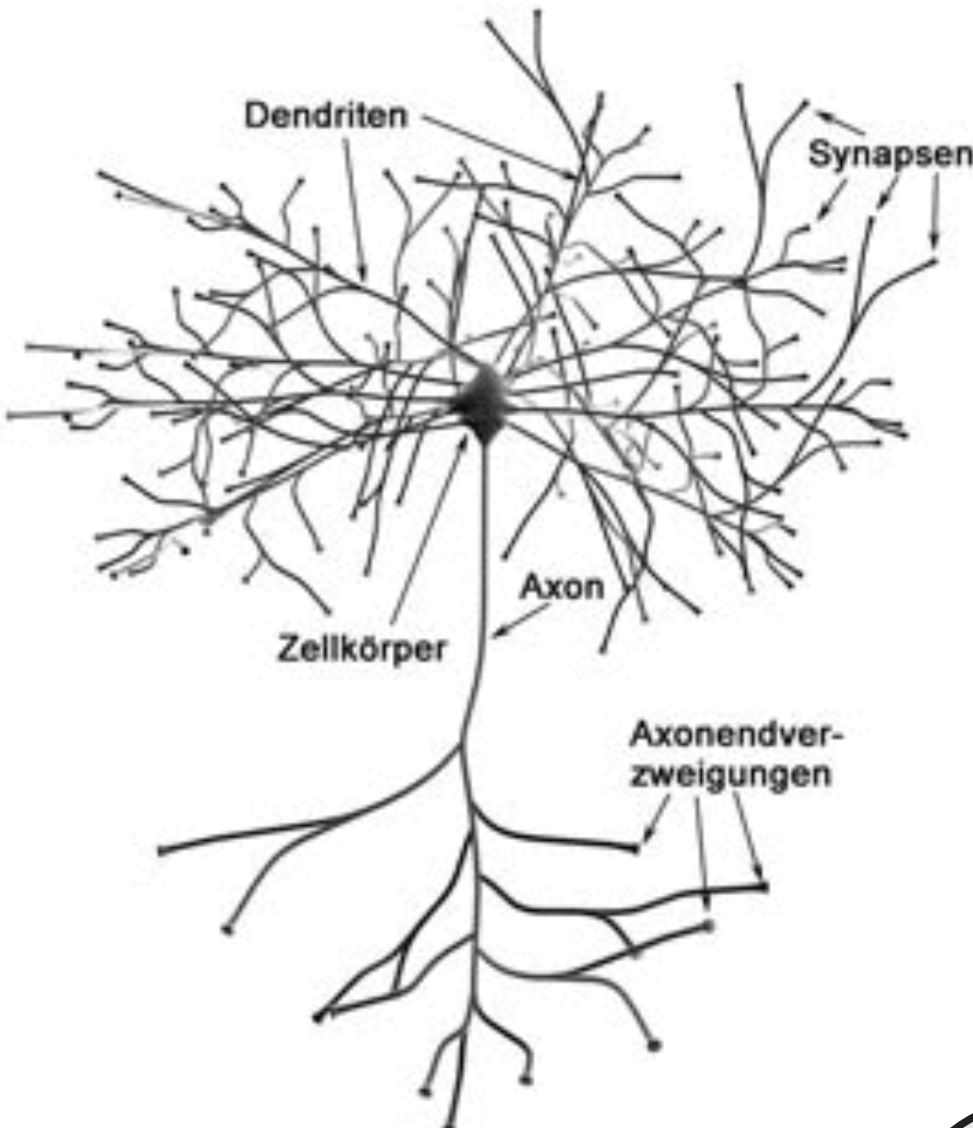
Representations

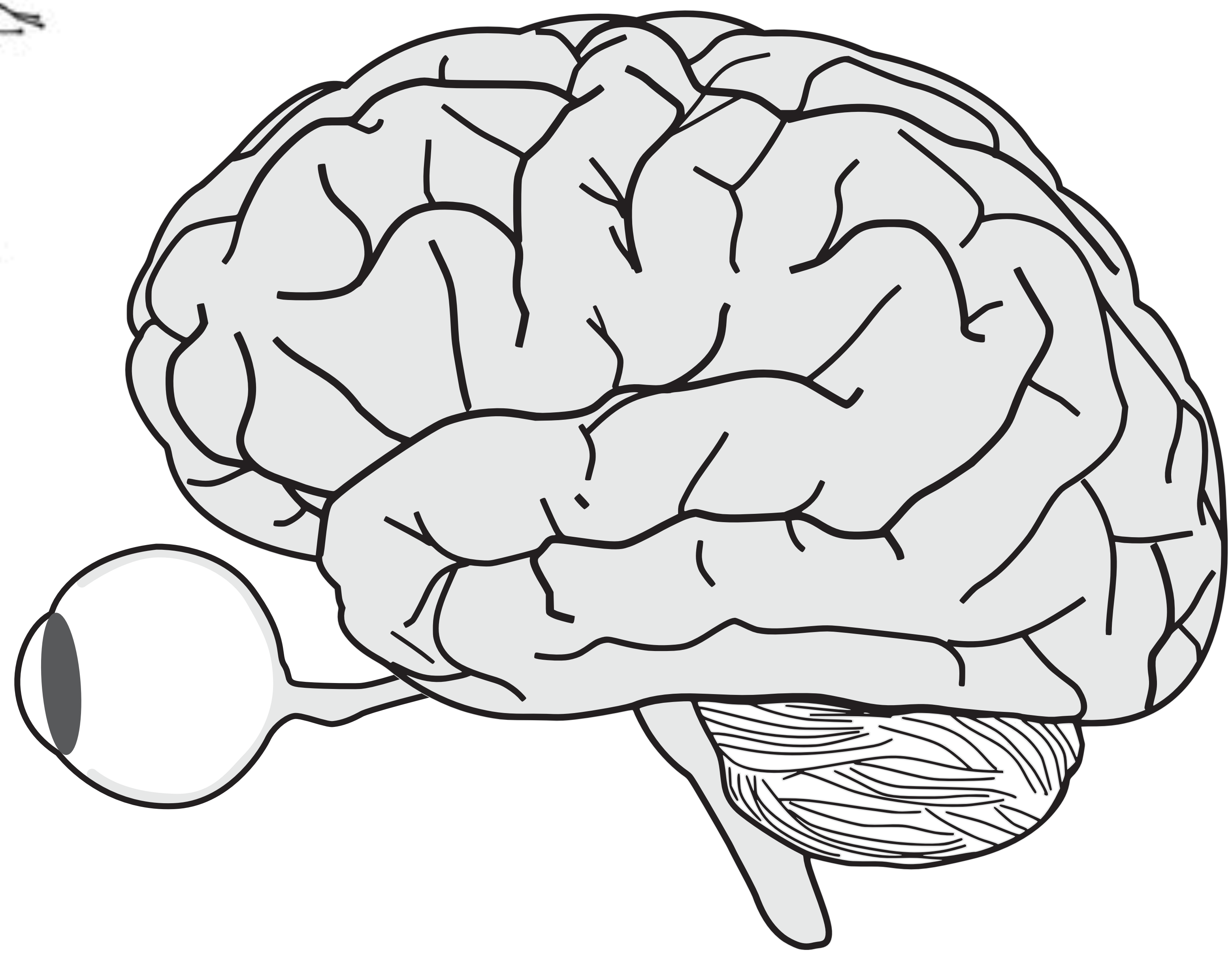


Representations

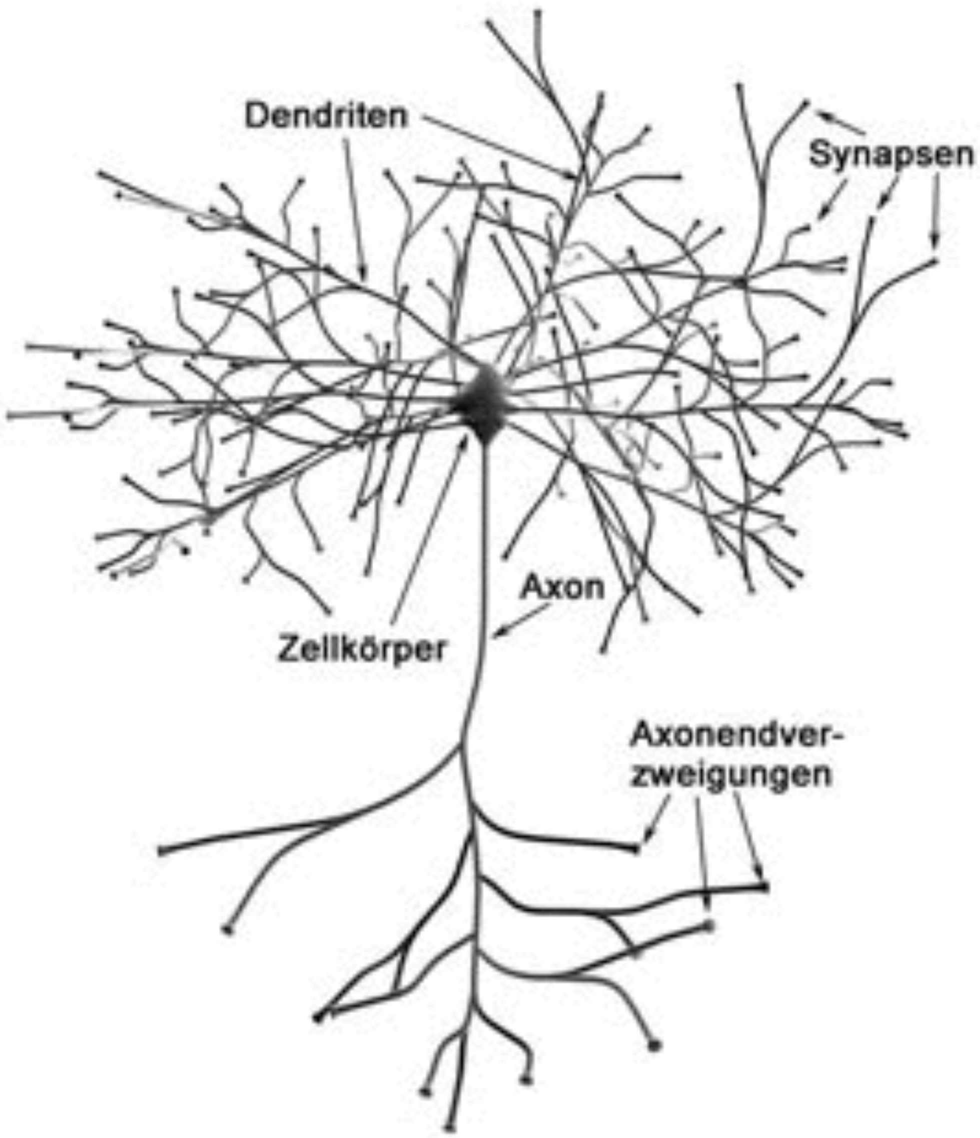


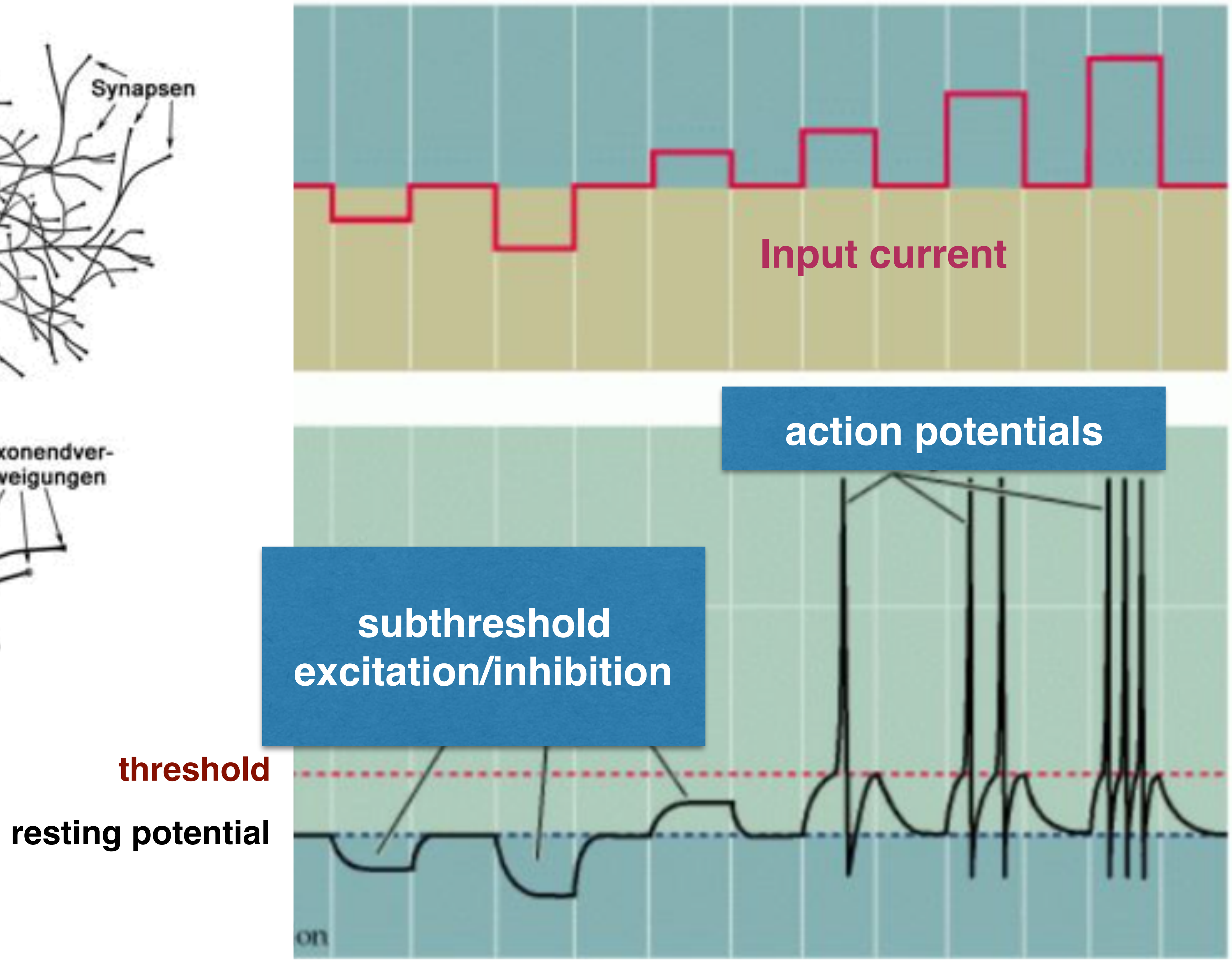
Representations

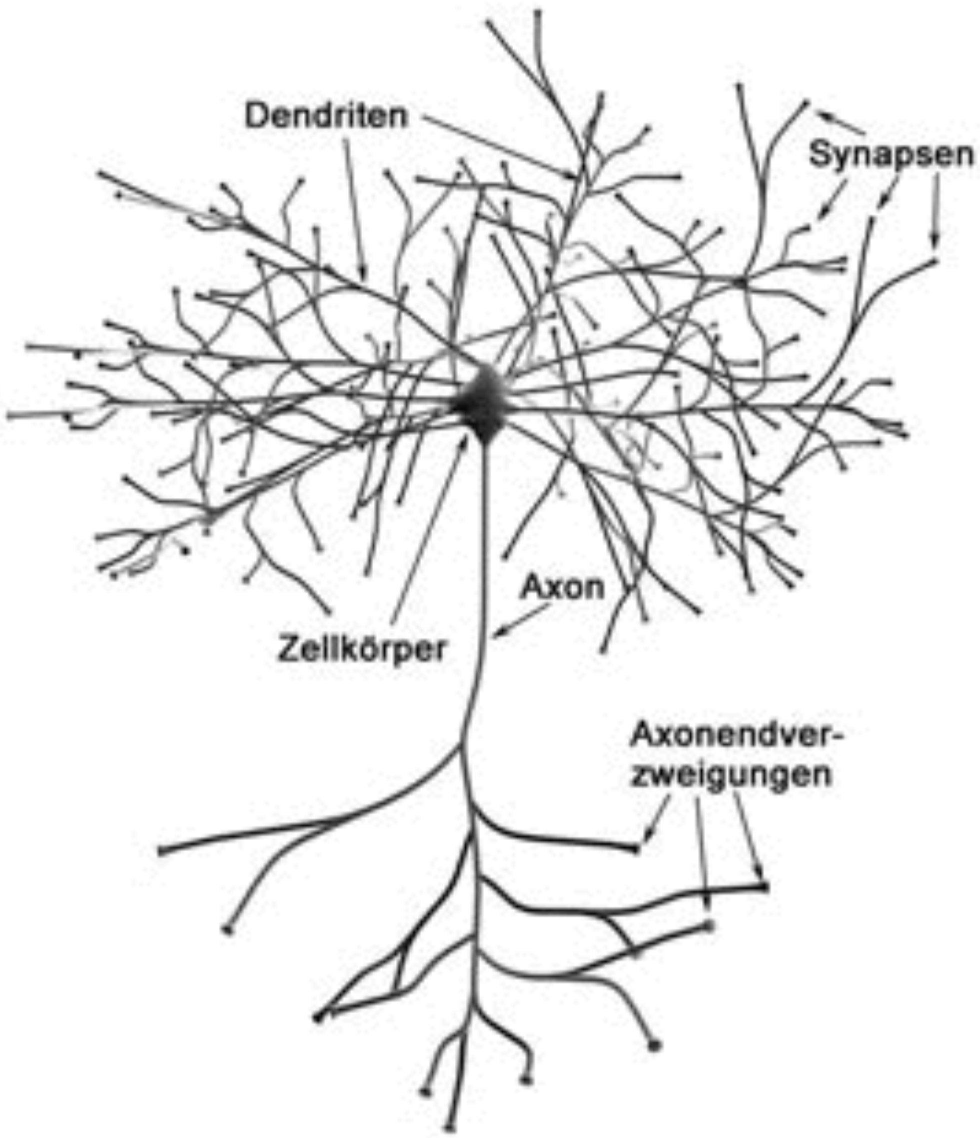


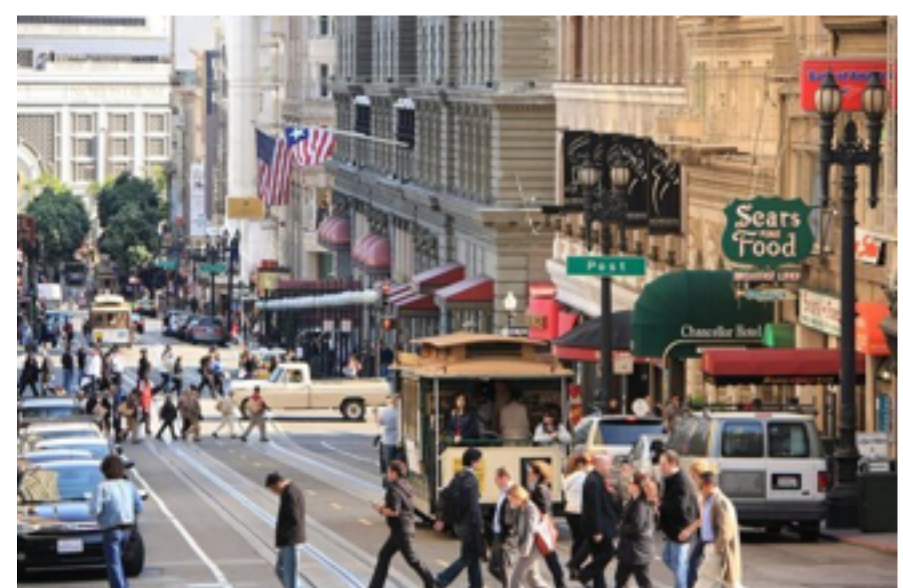


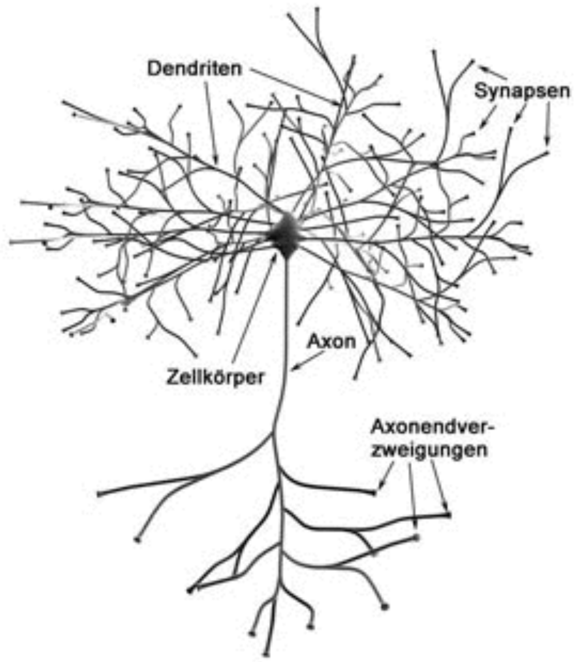
Representations



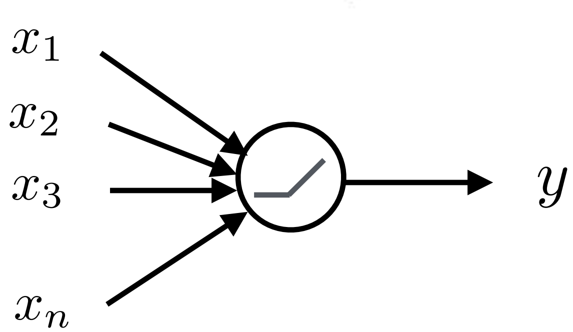



Dendriten
Synapsen
Axon
Zellkörper
Axonendver-
zweigungen

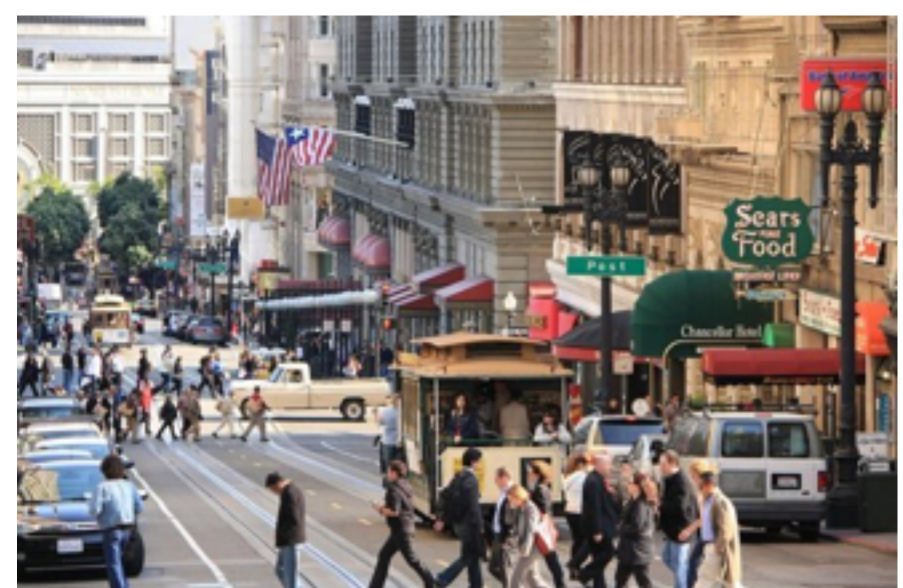

Sears
Food
Post



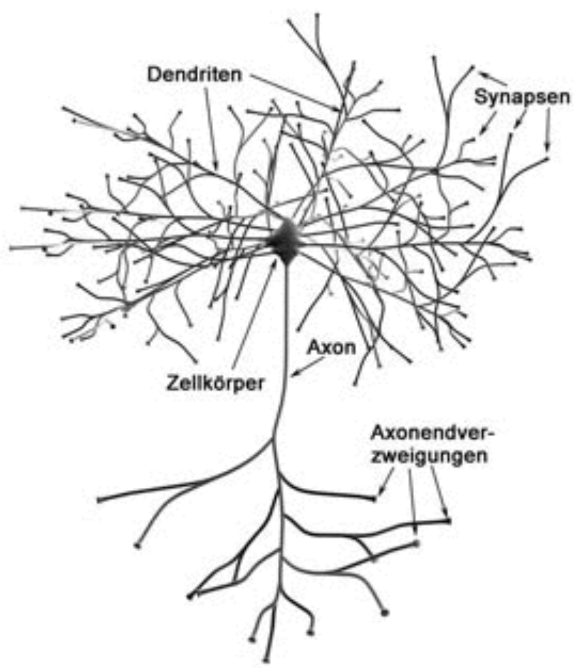


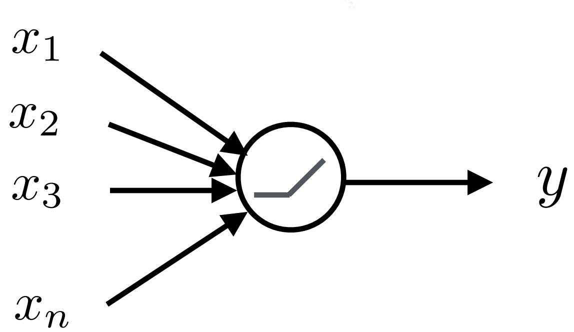
$$y = f(w_1 \cdot x_1 + \cdots + w_n \cdot x_n)$$



Threshold Logic Unit,
McCulloch&Pitts, 1943

Universal function approximation



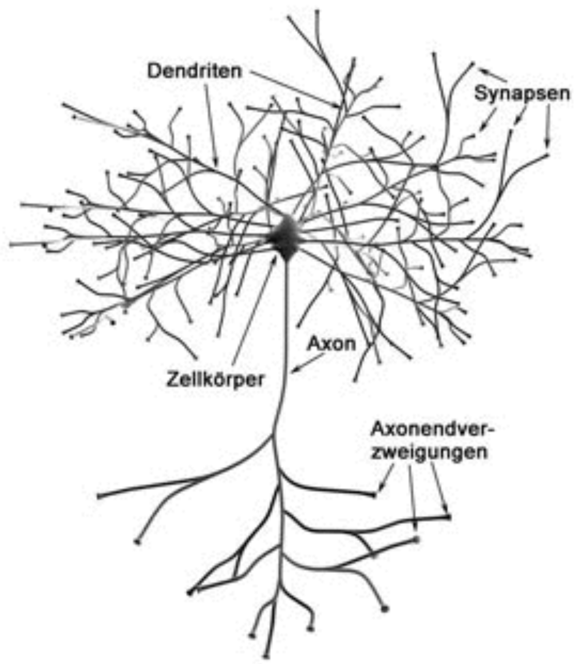


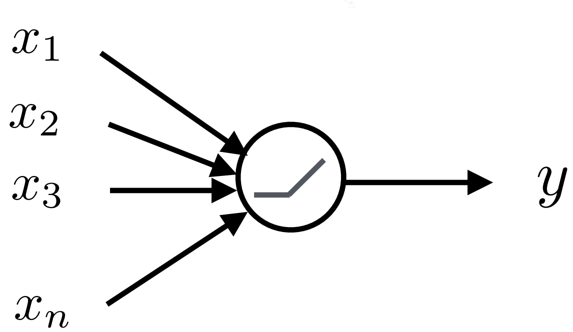
$$y = f(w_1 \cdot x_1 + \cdots + w_n \cdot x_n)$$

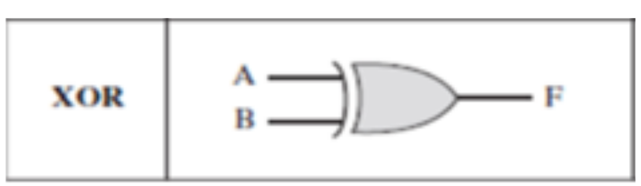
arbitrary behavior

Threshold Logic Unit,
McCulloch&Pitts, 1943

Universal function approximation







arbitrary
behavior

Threshold Logic Unit,
McCulloch&Pitts, 1943

Perceptron,
Rosenblatt, 1957

Back-propagation,
Werbos, 1974

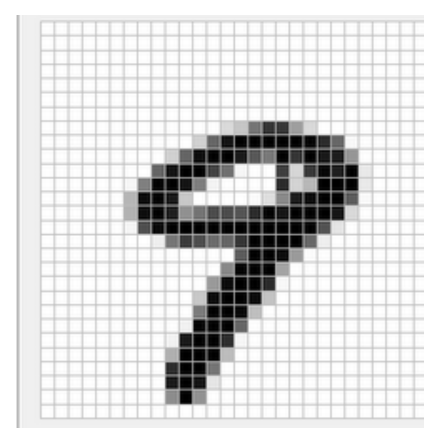
Curse of dimensionality

arbitrary boolean function:

$$f : \{0,1\}^n \to \{0,1\}$$
$$\mathbf{x} \quad \mapsto \quad y = f(\mathbf{x})$$

cardinality of hypothesis space: 2^n bits

Simple example (MNIST):



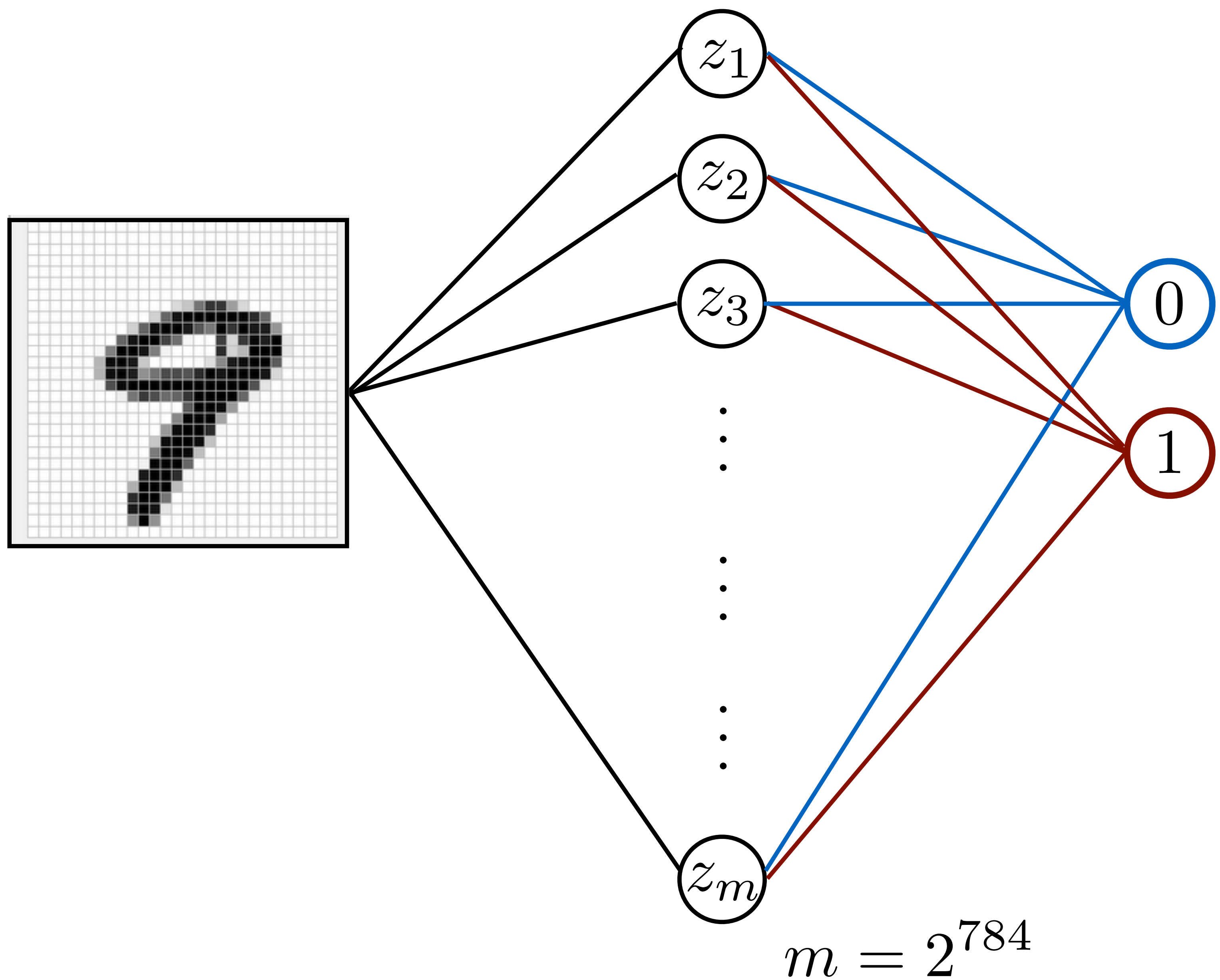
$n = 28^2 = 784$

amount of necessary information:

$$2^{784} \text{ bits} = 2^{781} \text{ bytes} = 2^{771} \text{ kB} = 2^{761} \text{ MB}$$
$$= 2^{751} \text{ GB} = 2^{741} \text{ TB} \approx 10^{75} \text{ TB}$$

Curse of dimensionality

Look-up table representation:



Curse of dimensionality

Lossy representation (pixel count):

Assumption: $$f(\mathbf{x}) = g(y), \quad y = g(\mathbf{x}) := \sum_{k=1}^{n} x_k$$

$$g : \{0,1\}^n \to \{0,1,2,\ldots,n\}$$

$$785 \text{ bits} \approx 98 \text{ bytes} < 0.1 \text{ kB}$$

Different types of prior constraints:

invariance constraint: $f(\mathbf{x}) = f(\mathbf{x}'), \quad \forall\, \mathbf{x}, \mathbf{x}' \in M$

equivariance constraint: $f(G\mathbf{x}) = \tilde{G} f(\mathbf{x}), \quad \forall\, \mathbf{x} \in M$

bounded variation: $d(f(\mathbf{x}), f(\mathbf{x}')) < \delta, \quad \forall\, \mathbf{x}' \in M(\mathbf{x})$

Curse of dimensionality

Don't expect too much from your data

$\Rightarrow$ **Try to use what you already know!**

Some basics on supervised learning

Generalized linear modeling

$$f : R^n \to R$$

$$f(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{x}) = g\left(\sum_{k=1}^{n} w_k\, x_k\right)$$

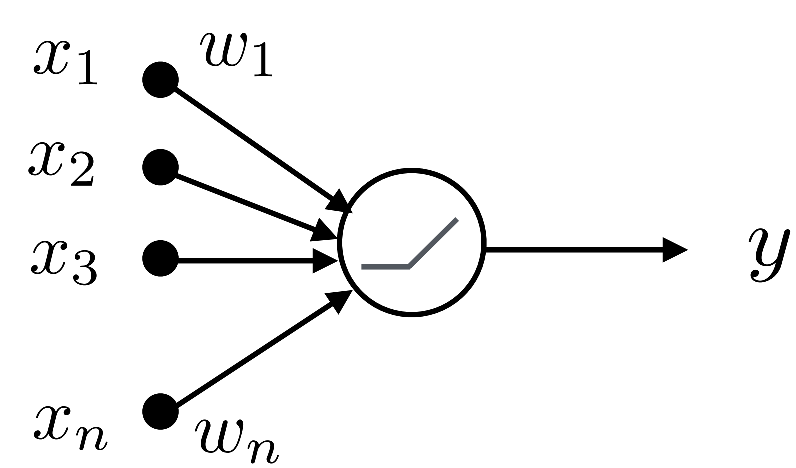
Nelder&Wedderburn, 1972

LN cascade models:

Linear - **N**onlinear

$$R^n \xrightarrow{L} R \quad \xrightarrow{g} R$$

$$\mathbf{x} \xmapsto{\mathbf{w}^\top \mathbf{x}} z \quad \xmapsto{g(z)} y$$



feature space embedding

$$f : R^n \to R$$

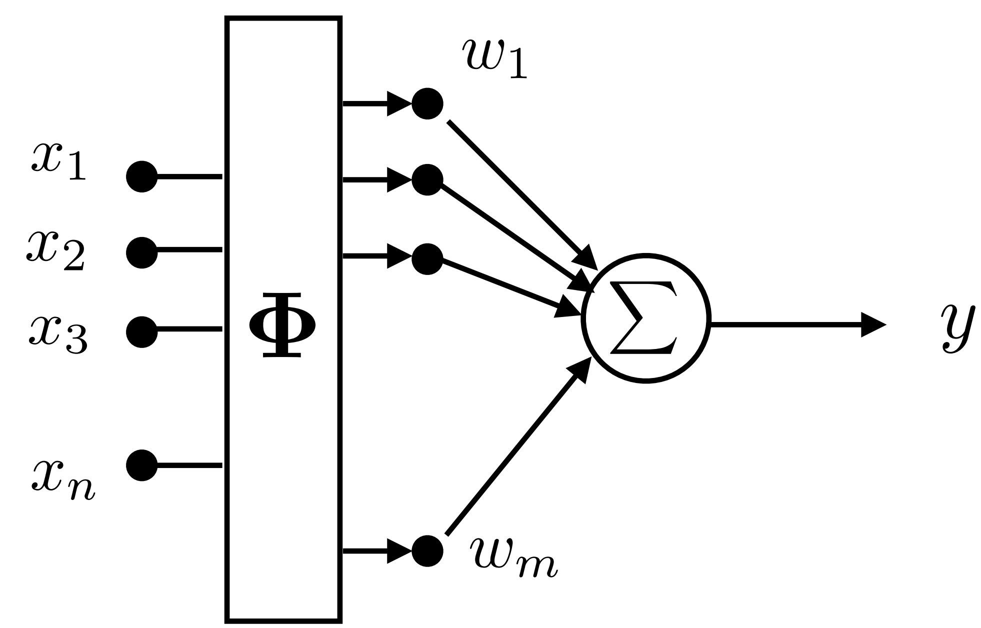
$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{\Phi}(\mathbf{x})$$

NL cascade models:

Nonlinear - **L**inear

$$R^n \xrightarrow{\mathbf{\Phi}} R^m \xrightarrow{L} R$$

$$\mathbf{x} \xrightarrow{\mathbf{\Phi}(\mathbf{x})} \mathbf{z} \xrightarrow{\mathbf{w}^\top \mathbf{z}} y$$



feature space embedding

$$f : R^n \to R \qquad f(\mathbf{x}) = \mathbf{w}^\top \mathbf{\Phi}(\mathbf{x})$$

Example I: Quadratic feature space

$$z_1 = x_1^2 \quad z_2 = x_1 \cdot x_2 \quad z_3 = x_1 \cdot x_3 \quad \cdots \quad z_n = x_1 \cdot x_n$$

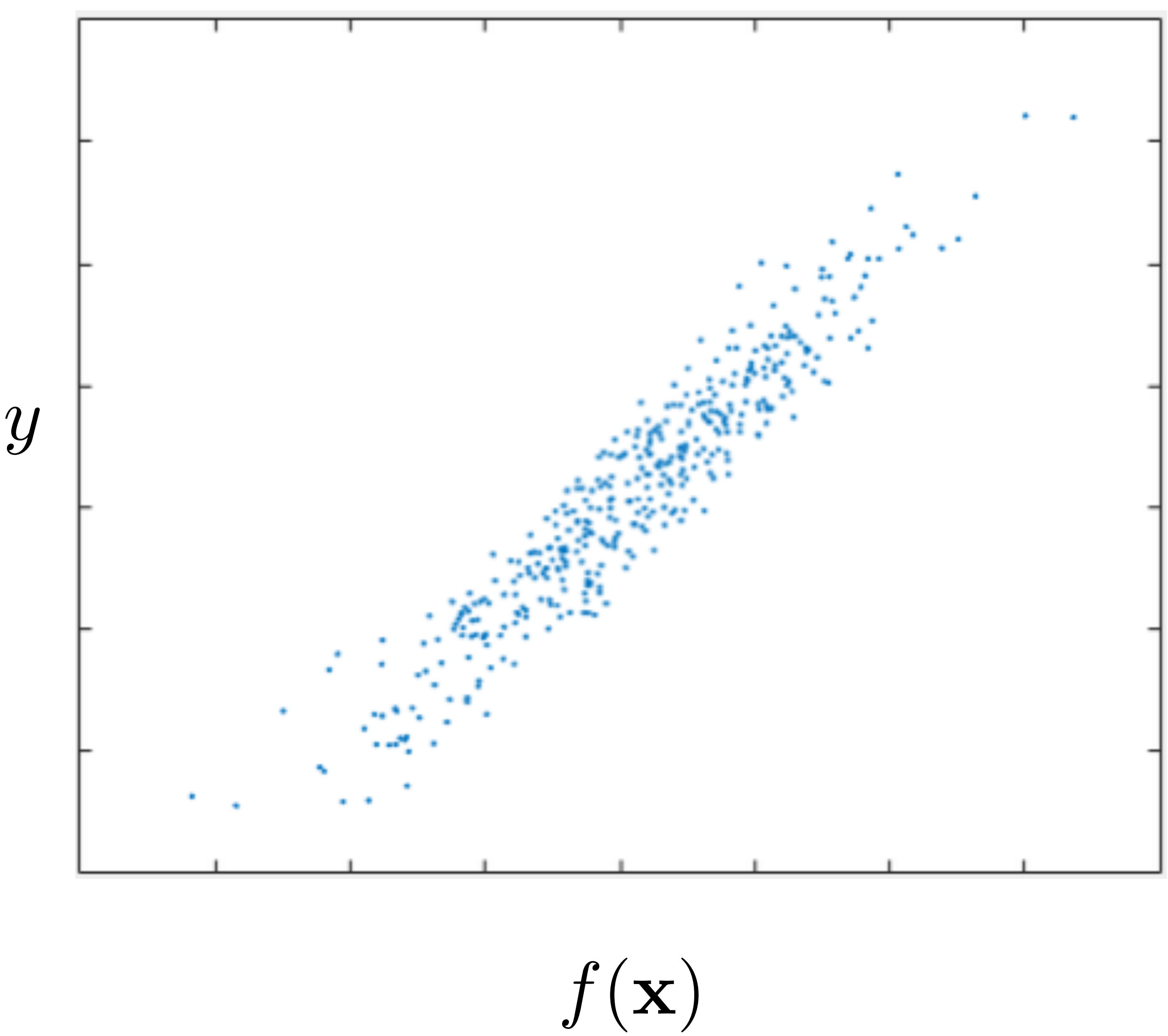
$$z_{n+1} = x_2^2 \quad z_{n+2} = x_2 \cdot x_3 \quad \cdots \quad z_{2n-1} = x_2 \cdot x_n \quad \cdots \quad z_m = x_{n(n+1)/2} = x_n^2$$

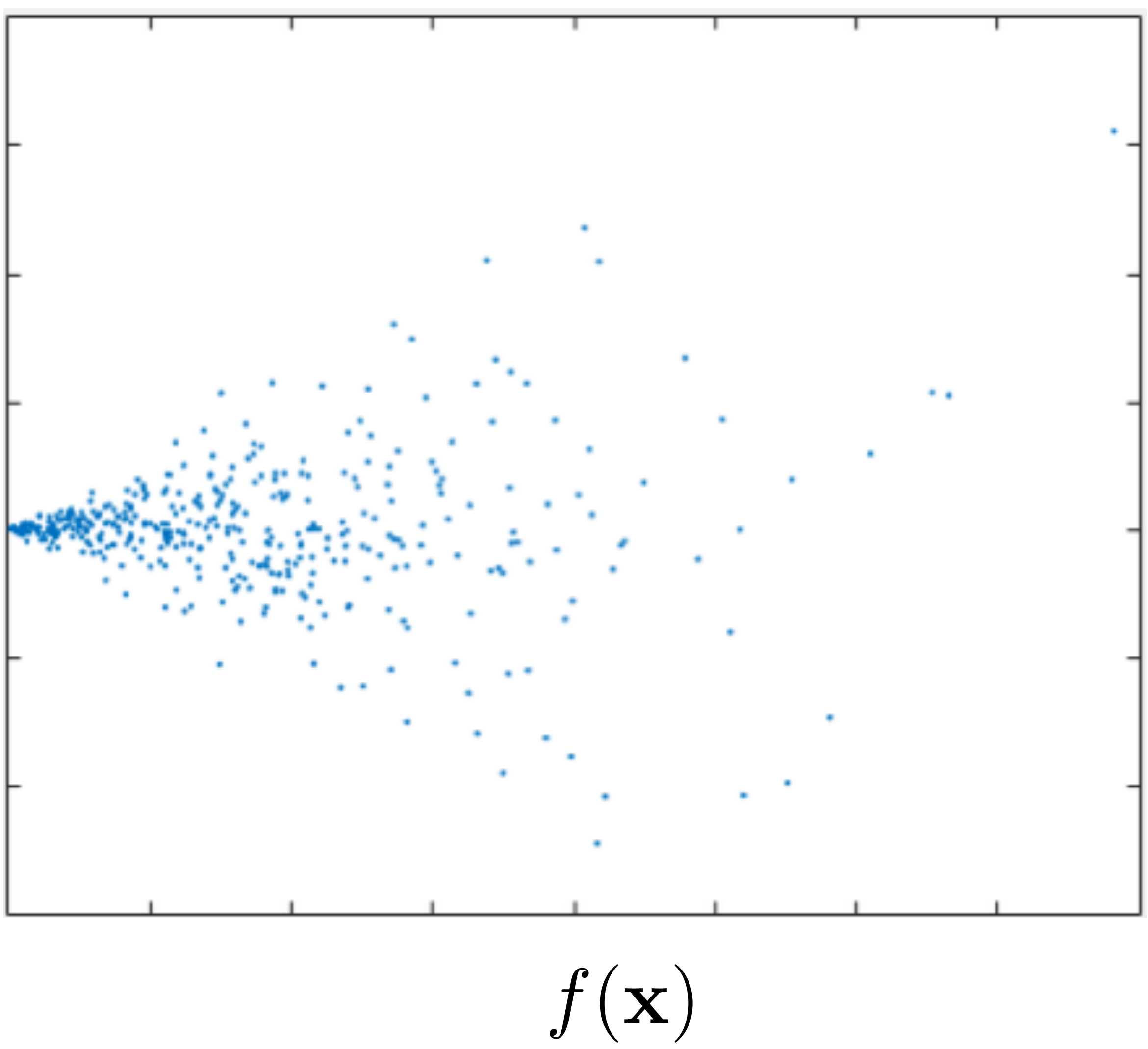
Example II: Arbitrary polynomial feature space

Linear $\bigoplus$ Quadratic $\bigoplus$ Cubic $\bigoplus$ $\cdots$

Curse of dimensionality: $m \geq \sum_{k=0}^{n} \binom{n}{k} = 2^n$

Noise



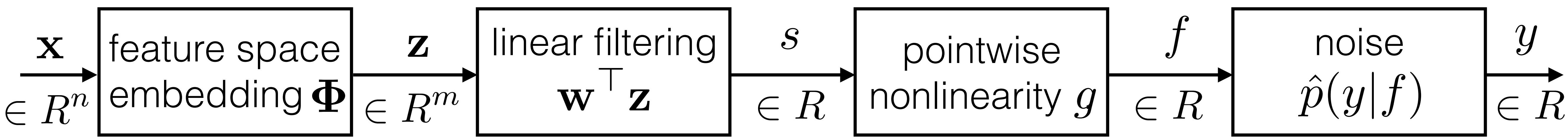


$$p(y|f(\mathbf{x}) = \mathcal{N}(y|f(\mathbf{x}), \sigma^2)$$

$$p(y|f(\mathbf{x}) = \mathcal{N}(y|\mu, (\sigma \cdot f(\mathbf{x}))^2)$$

NLN cascade regression

$f : R^n \to R$ $\quad f(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{\Phi}(\mathbf{x}))$ $\quad$ noise model: $\hat{p}(y|f_{\mathbf{w}}(\mathbf{x}))$

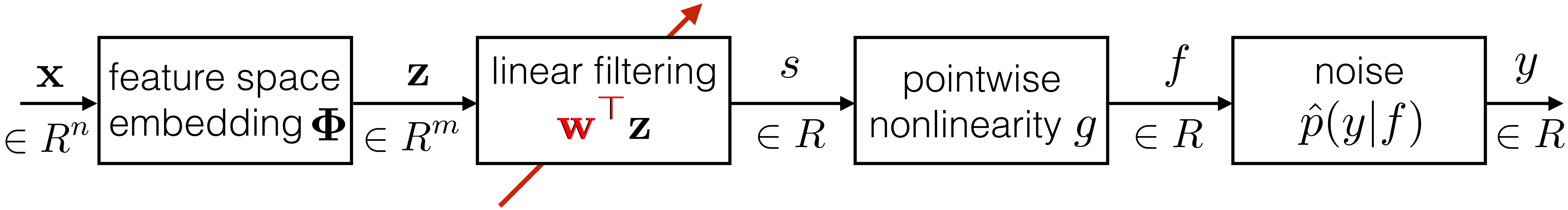


Cross-entropy:

$$E[-\log \hat{p}(y)] = -\int p(y) \log \hat{p}(y) dy = \underbrace{-\int p(y) \log p(y) dy}_{h[Y]} + \underbrace{\int p(y) \log \frac{p(y)}{\hat{p}(y)} dy}_{D_{KL}[p(y)||\hat{p}(y)] \geq 0}$$

NLN cascade regression

$f : R^n \to R \qquad f(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{\Phi}(\mathbf{x}))$ noise model: $\hat{p}(y|f_{\mathbf{w}}(\mathbf{x}))$



Cross-entropy:

$$E[-\log \hat{p}(y)] = -\int p(y)\log \hat{p}(y)dy = \underbrace{-\int p(y)\log p(y)dy}_{h[Y]} + \underbrace{\int p(y)\log \frac{p(y)}{\hat{p}(y)}dy}_{D_{KL}[p(y)||\hat{p}(y)]\geq 0}$$

Cross-entropy learning:

Find weights $\mathbf{w}$ to minimize the cross-entropy w.r.t. some noise model $\hat{p}(y|f_{\mathbf{w}}(\mathbf{x}))$ that depends on $f_{\mathbf{w}}(\mathbf{x})$.

NLN cascade regression

Example (least squares regression):

additive Gaussian noise model:

$$\hat{p}(y|f_{\mathbf{w}}(\mathbf{x})) := \mathcal{N}(y|\mathbf{w}^\top \mathbf{x}, \sigma^2)$$

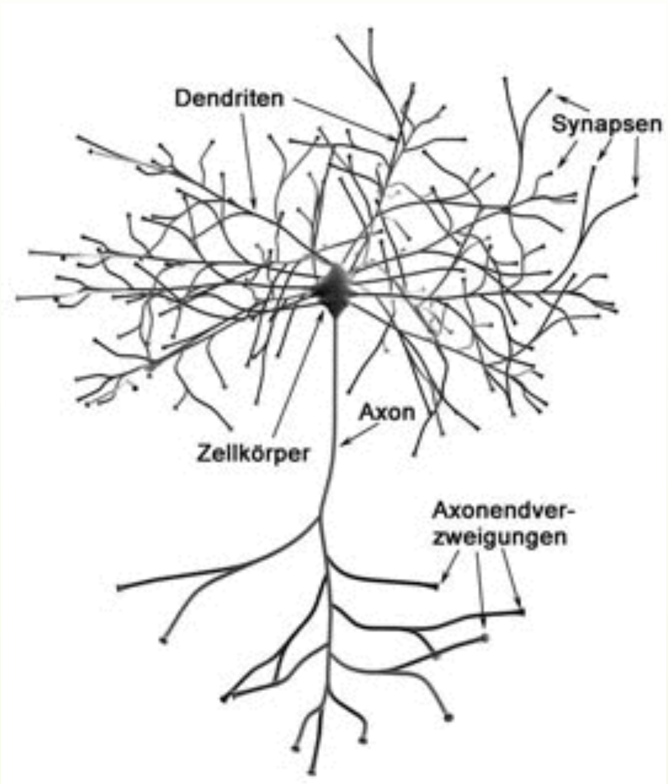
cross-entropy:

$$E[-\log(\hat{p}(y|f_{\mathbf{w}}(\mathbf{x})))] = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}\left(\frac{y - \mathbf{w}^\top \mathbf{x}}{\sigma}\right)^2$$

unique optimum:

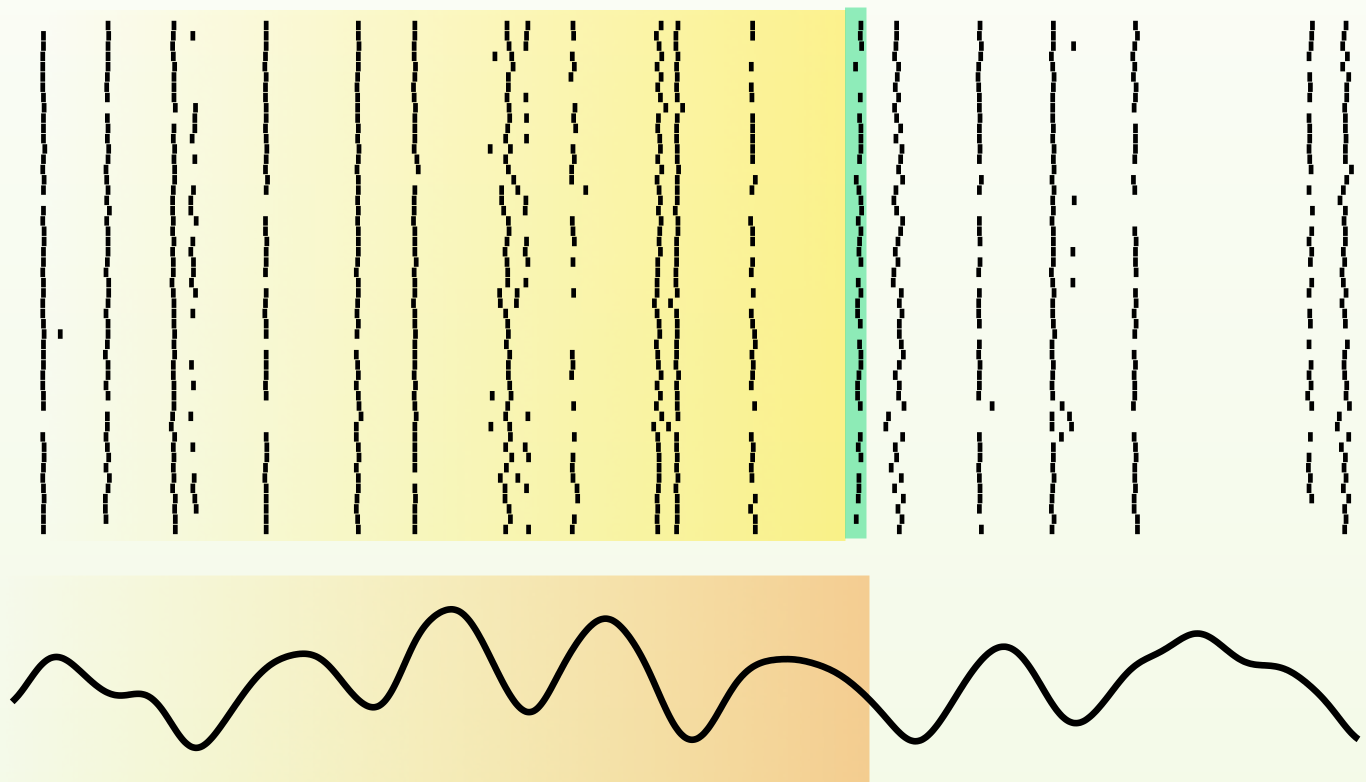
$$\mathbf{w}^\top = E[y\mathbf{x}^\top](E[\mathbf{x}\mathbf{x}^\top])^{-1}$$

Neural System identification

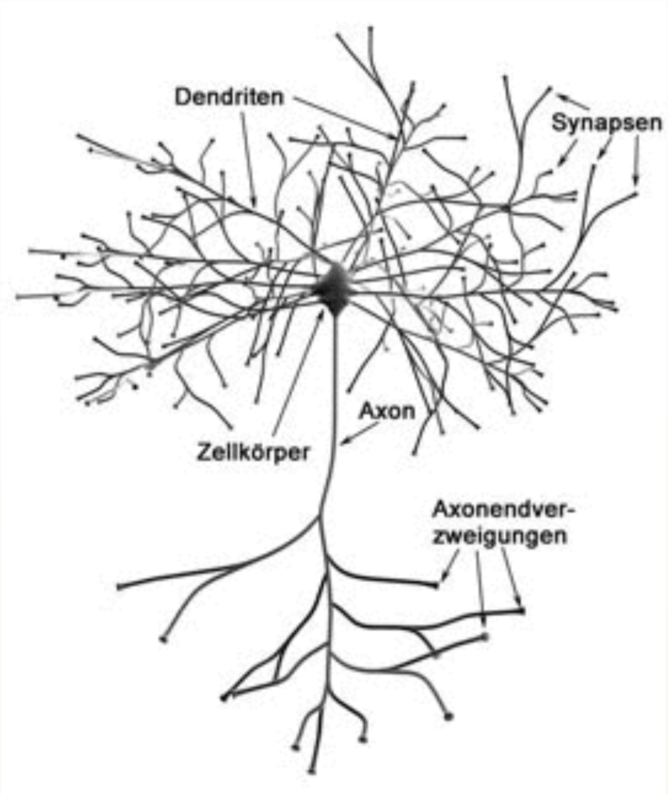


Spikes

Stimulus



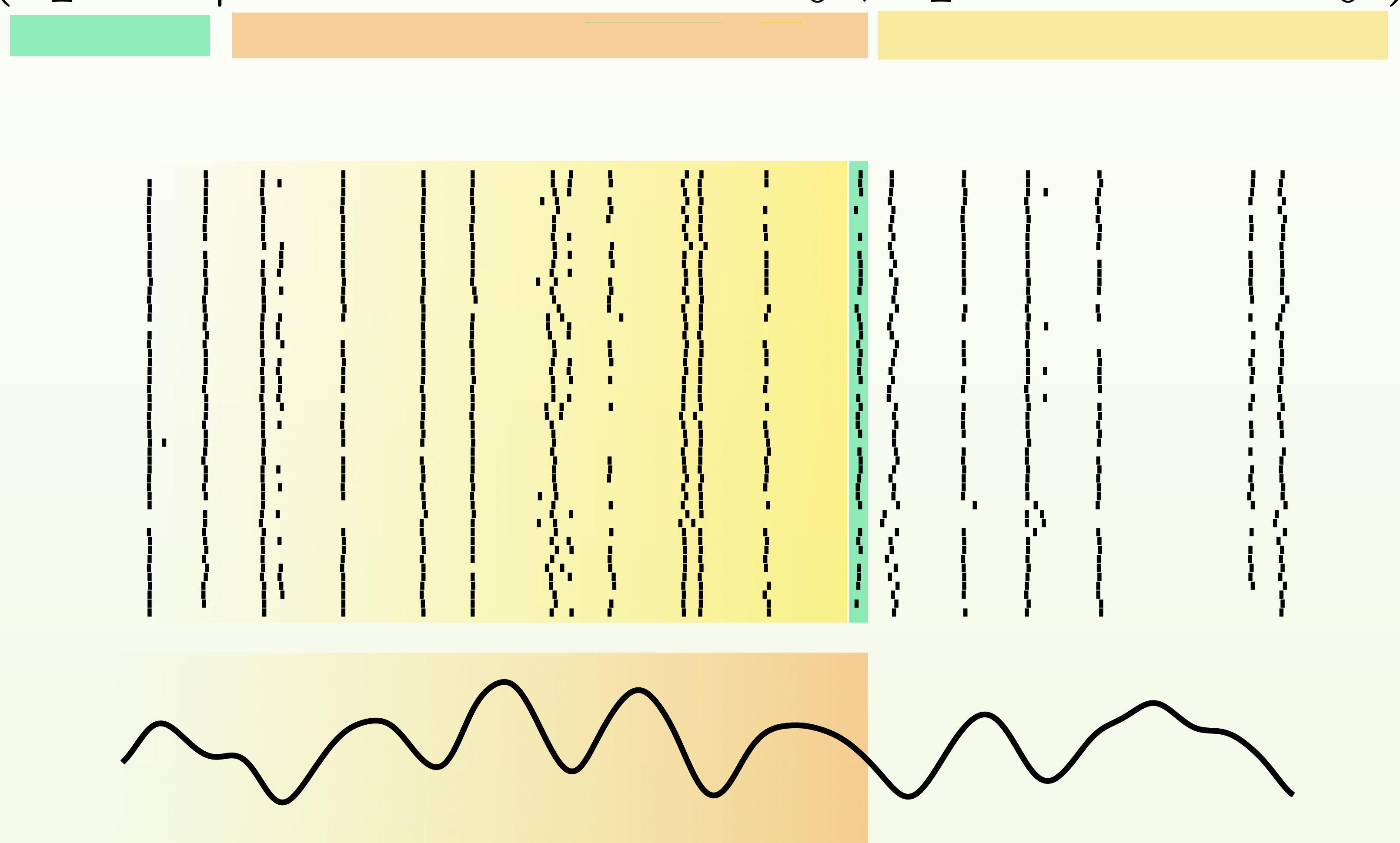
Neural System identification



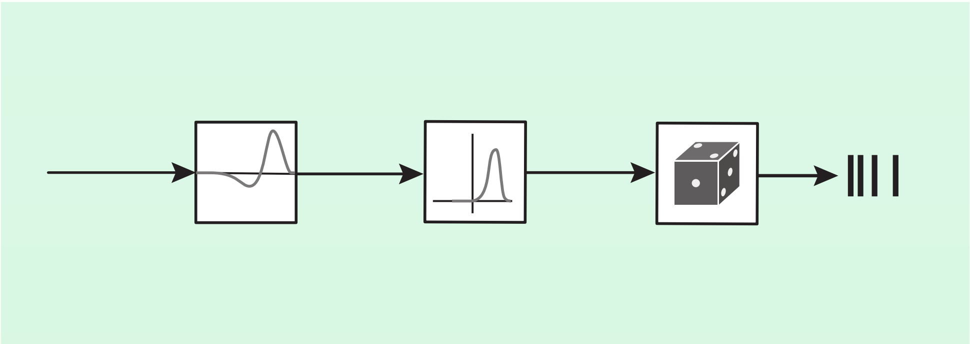
$$p(\text{spike}|\text{stimulus history}, \text{spike history})$$

Spikes

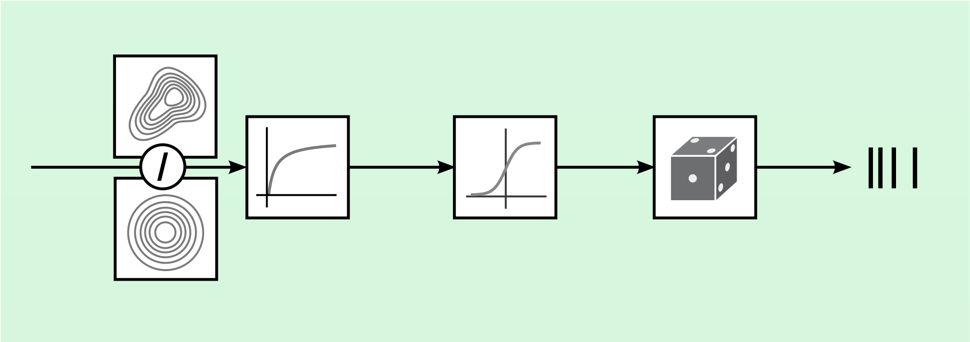
Stimulus



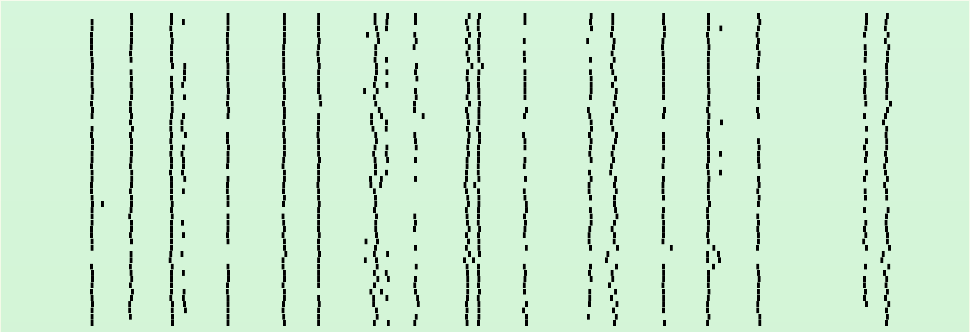
Overview



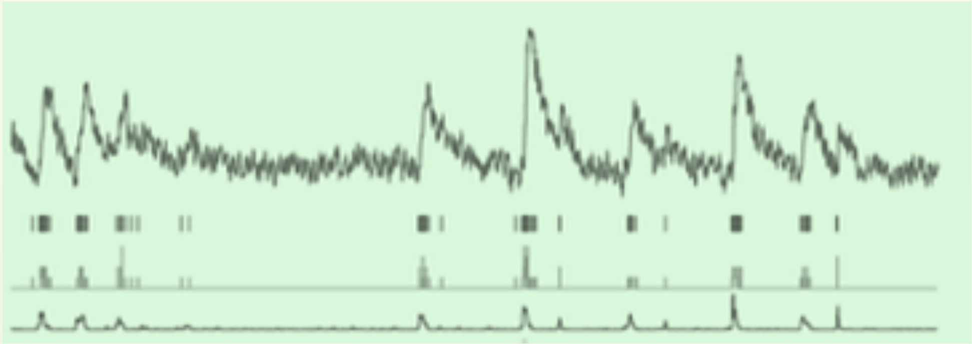
1 Generalized linear models (**GLMs**)



2 Spike-triggered-mixture model (**STM**)

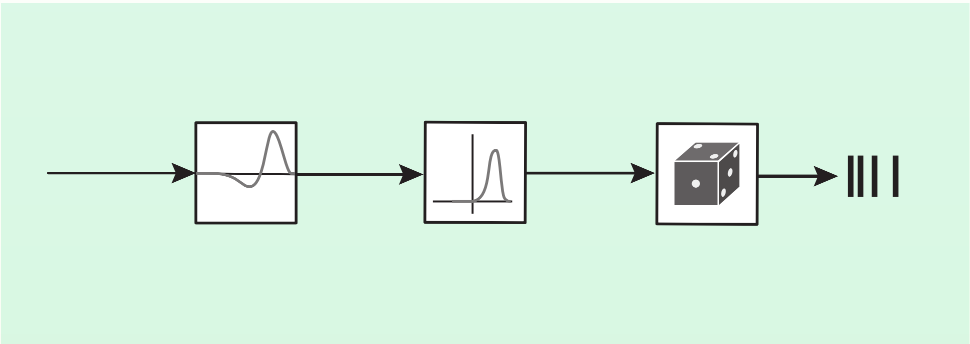


3 Empirical results

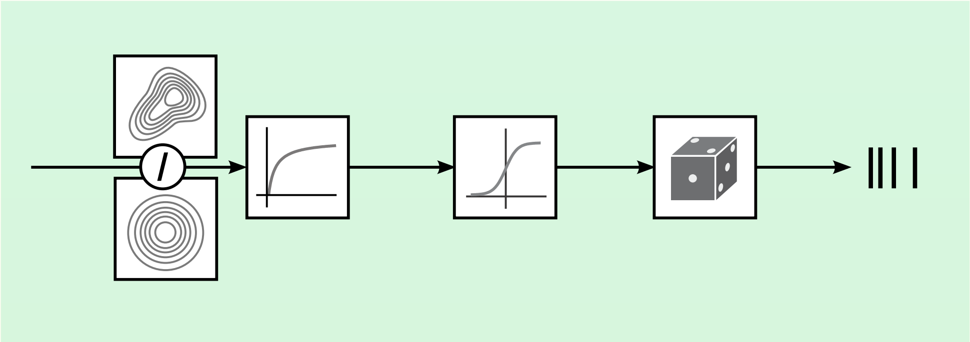


4 Spike detection

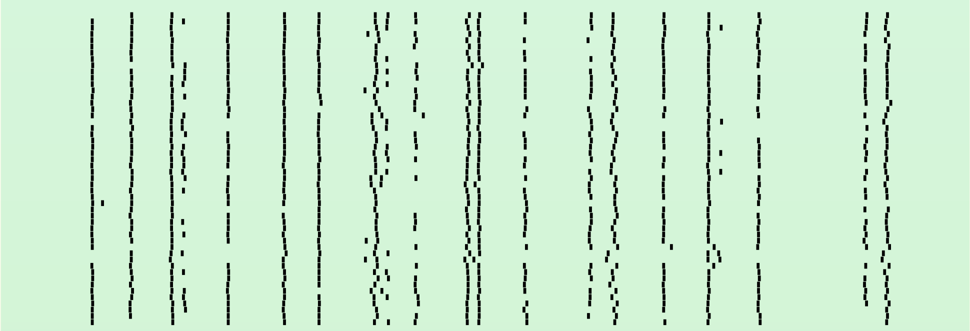
Overview



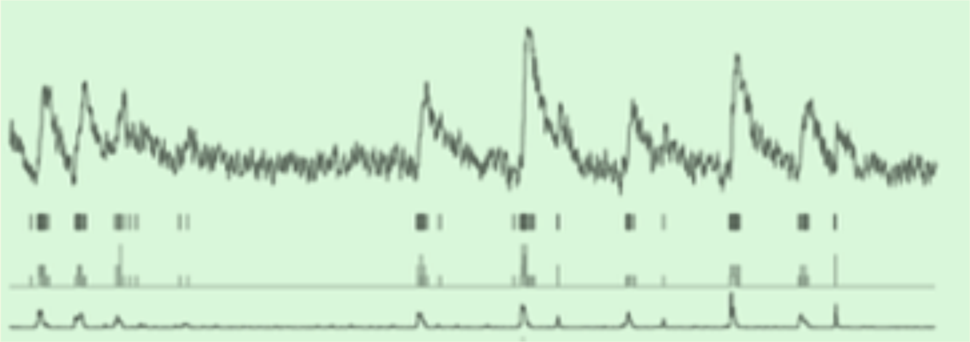
1 **Generalized linear models (GLMs)**



2 Spike-triggered-mixture model (**STM**)



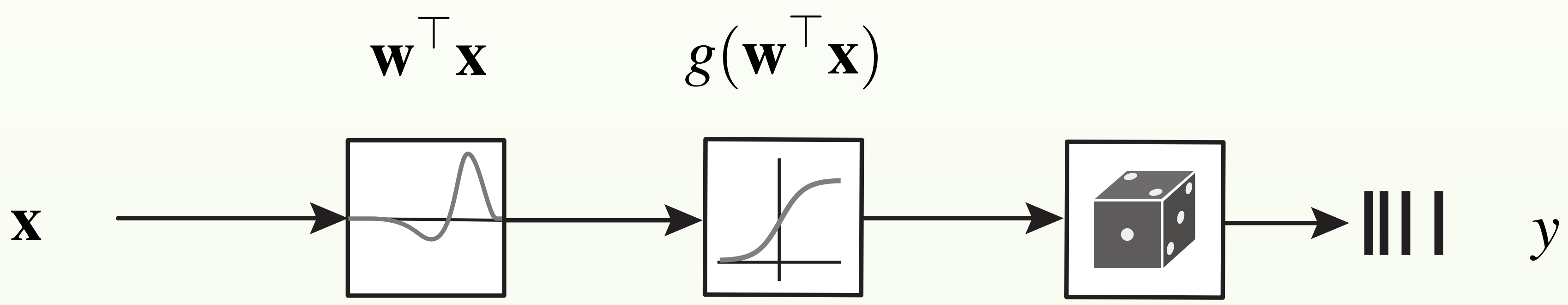
3 Empirical results



4 Spike detection

Generalized linear models

$$p(y|\mathbf{x}) = p(\text{spike}|\text{stimulus history})$$



Linear-nonlinear-Poisson

$$p(y \mid \mathbf{x}) = \frac{\lambda^y}{y!} e^{-\lambda}$$

$$\lambda = \exp(\mathbf{w}^\top \mathbf{x})$$

Linear-nonlinear-Bernoulli
(logistic regression)

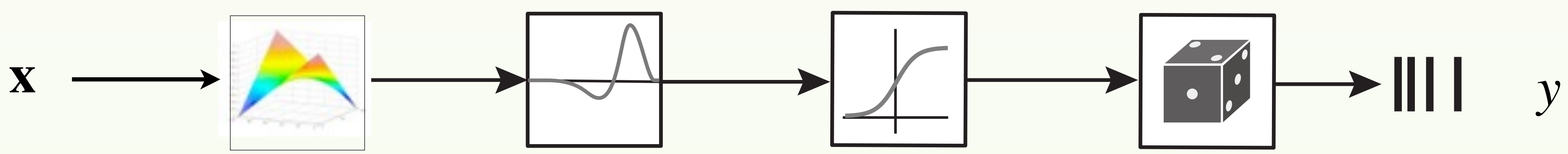
$$p(y \mid \mathbf{x}) = r^y (1-r)^{1-y}$$

$$r = \left(1 + \exp\left(-\mathbf{w}^\top \mathbf{x}\right)\right)^{-1}$$

Generalized linear models

$$p(y|\mathbf{x}) = p(\text{spike}|\text{stimulus history})$$

$\phi(\mathbf{x})$ $\quad$ $\mathbf{w}^\top\phi(\mathbf{x})$ $\quad$ $g(\mathbf{w}^\top\phi(\mathbf{x}))$



(fixed)
feature space

Linear-nonlinear-Poisson

$$p(y \mid \mathbf{x}) = \frac{\lambda^y}{y!} e^{-\lambda}$$

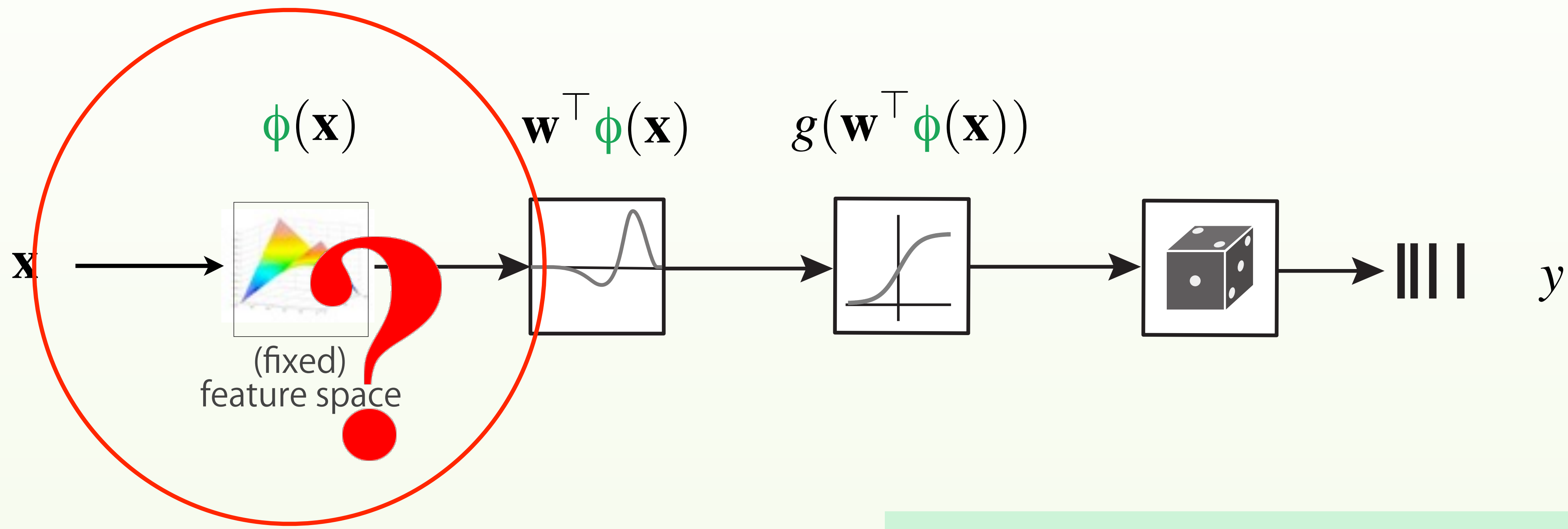
$$\lambda = \exp(\mathbf{w}^\top \Phi(\mathbf{x}))$$

Linear-nonlinear-Bernoulli
(logistic regression)

$$p(y \mid \mathbf{x}) = r^y (1 - r)^{1-y}$$

$$r = \left(1 + \exp\left(-\mathbf{w}^\top \Phi(\mathbf{x})\right)\right)^{-1}$$

Generalized linear models



- $\mathbf{w}$ is optimized, ϕ is fixed

+Concave log-likelihood

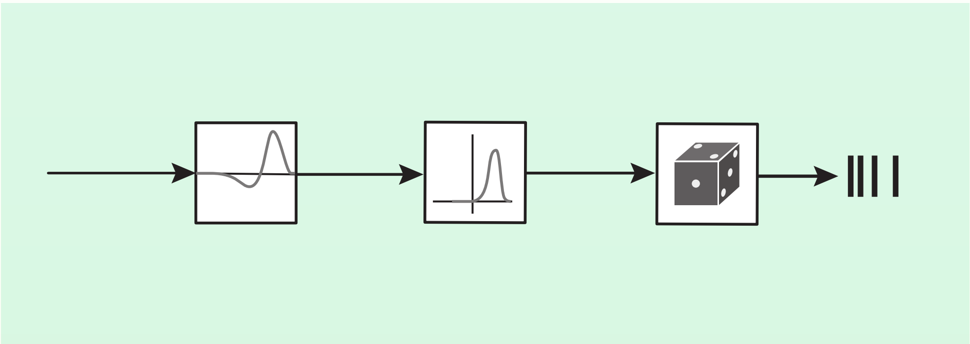
–Choosing a good ϕ is hard

Linear-nonlinear-Bernoulli
(logistic regression)

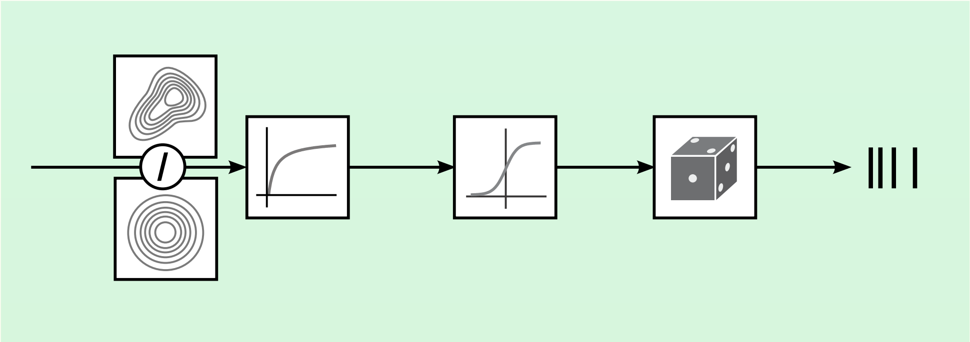
$$p(y \mid \mathbf{x}) = r^y (1-r)^{1-y}$$

$$r = \left(1 + \exp\left(-\mathbf{w}^\top \mathbf{x}\right)\right)^{-1}$$

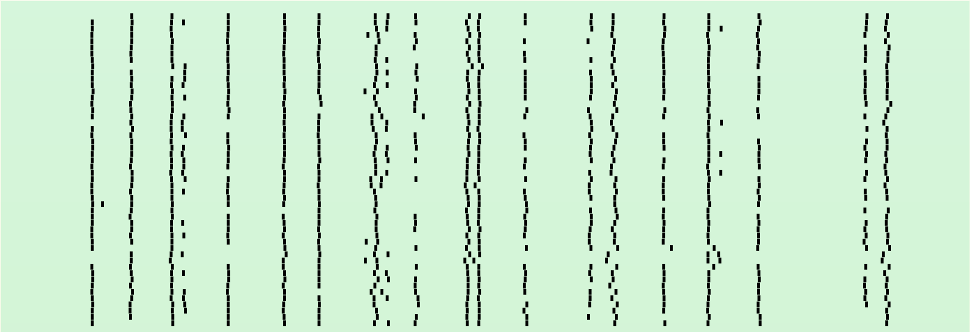
Overview



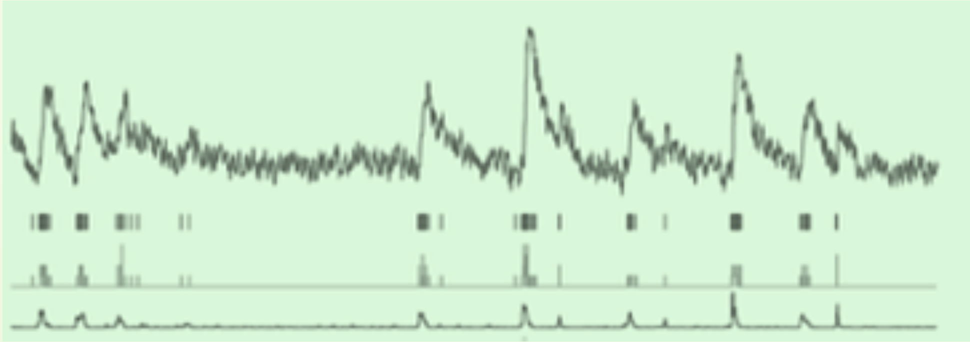
1 Generalized linear models (**GLMs**)



2 **Spike-triggered-mixture model (STM)**



3 Empirical results

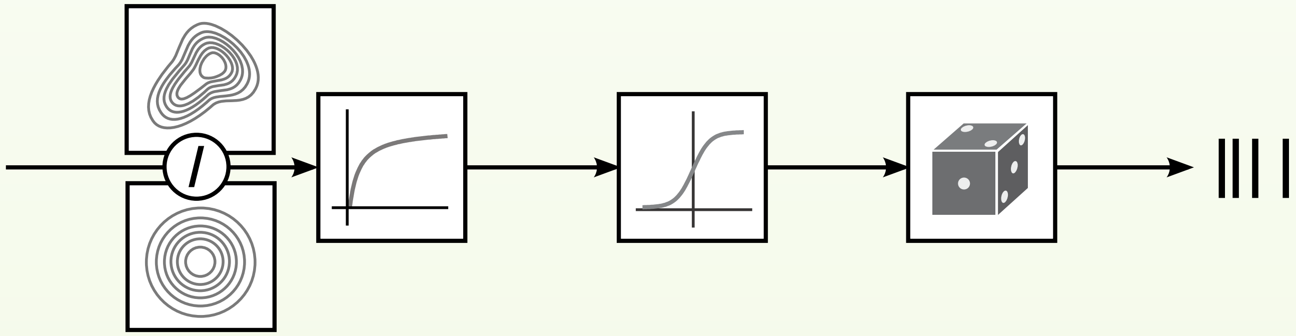


4 Spike detection

A generative view

Lucas Theis

$$p(\text{spike} \mid \mathbf{x}) \propto p(\mathbf{x} \mid \text{spike})\, p(\text{spike})$$



Spike-triggered distribution

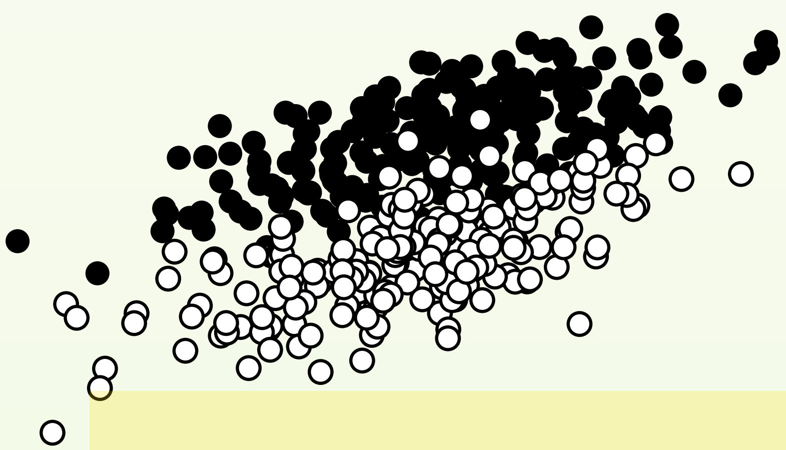
$$p(\text{spike} = 1 \mid \mathbf{x}) = \sigma\left(\log \frac{p(\mathbf{x} \mid \text{spike} = 1)}{p(\mathbf{x} \mid \text{spike} = 0)} + \log \frac{p(\text{spike} = 1)}{p(\text{spike} = 0)}\right)$$

Non-spike-triggered distribution

A generative view

Lucas Theis

$$p(\mathbf{x} \mid \text{spike} = 1) = \mathcal{N}(\mathbf{x}; \mu_1, \Sigma_1)$$



$$p(\mathbf{x} \mid \text{spike} = 0) = \mathcal{N}(\mathbf{x}; \mu_0, \Sigma_0)$$

$$p(\text{spike} = 1 \mid \mathbf{x}) = \sigma\left(\log \frac{p(\mathbf{x} \mid \text{spike} = 1)}{p(\mathbf{x} \mid \text{spike} = 0)} + \log \frac{p(\text{spike} = 1)}{p(\text{spike} = 0)}\right)$$

Quadratic-nonlinear-Bernoulli

$$p(\text{spike} = 1 \mid \mathbf{x}) = \sigma\left(\cancel{\mathbf{x}^\top \mathbf{Q} \mathbf{x}} + \mathbf{w}^\top \mathbf{x} + b\right)$$

$$\cancel{\mathbf{Q} = (\Sigma_0^{-1} - \Sigma_1^{-1})/2}$$

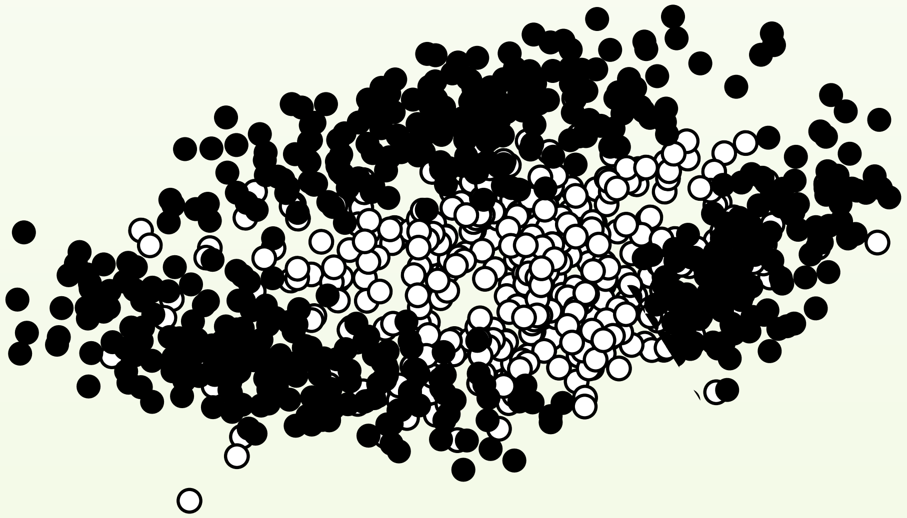
$$\mathbf{w} = \Sigma_1^{-1} \mu_1 - \Sigma_0^{-1} \mu_0$$

A generative view

Lucas Theis

$$p(\text{spike}=1 \mid \mathbf{x}) = \sigma\left(\log\frac{p(\mathbf{x}\mid\text{spike}=1)}{p(\mathbf{x}\mid\text{spike}=0)} + \log\frac{p(\text{spike}=1)}{p(\text{spike}=0)}\right)$$

$$p(\mathbf{x}\mid\text{spike}=1) = \sum_k \pi_{1k}\mathcal{N}(\mathbf{x};\mu_{1k},\Sigma_{1k})$$



$$p(\mathbf{x}\mid\text{spike}=0) = \mathcal{N}\left(\mathbf{x};\mu_0,\Sigma_0\right)$$

Spike-triggered-mixture model (STM)

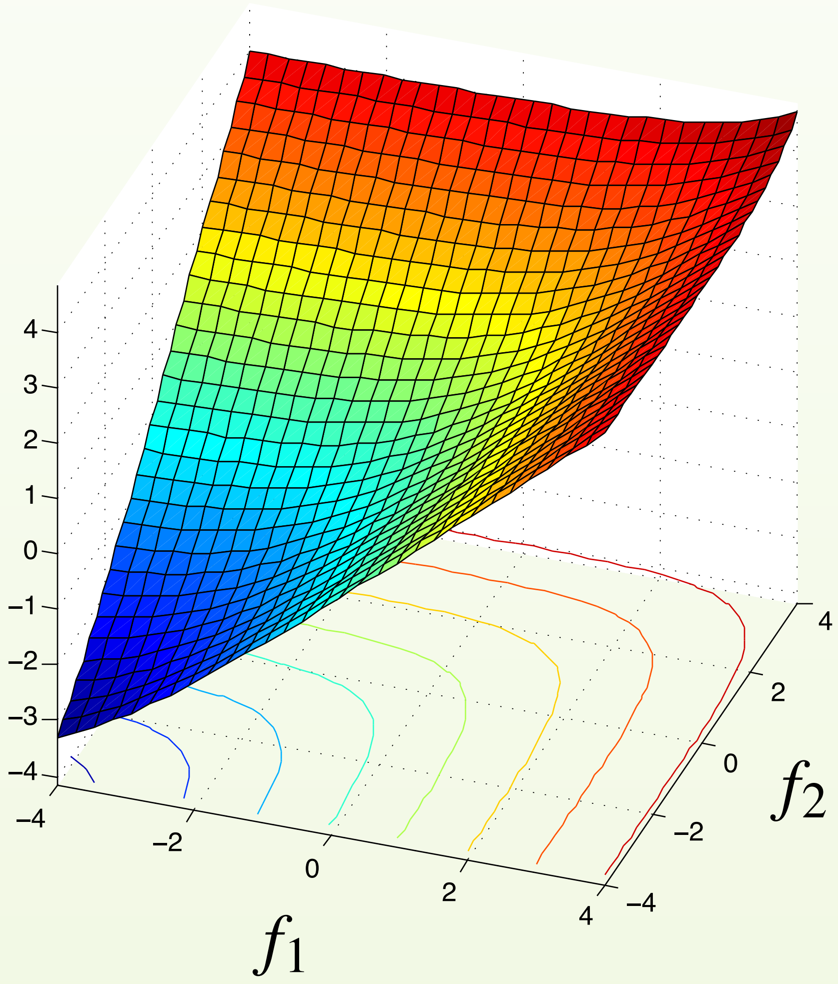
$$p(\text{spike}=1\mid\mathbf{x}) =$$

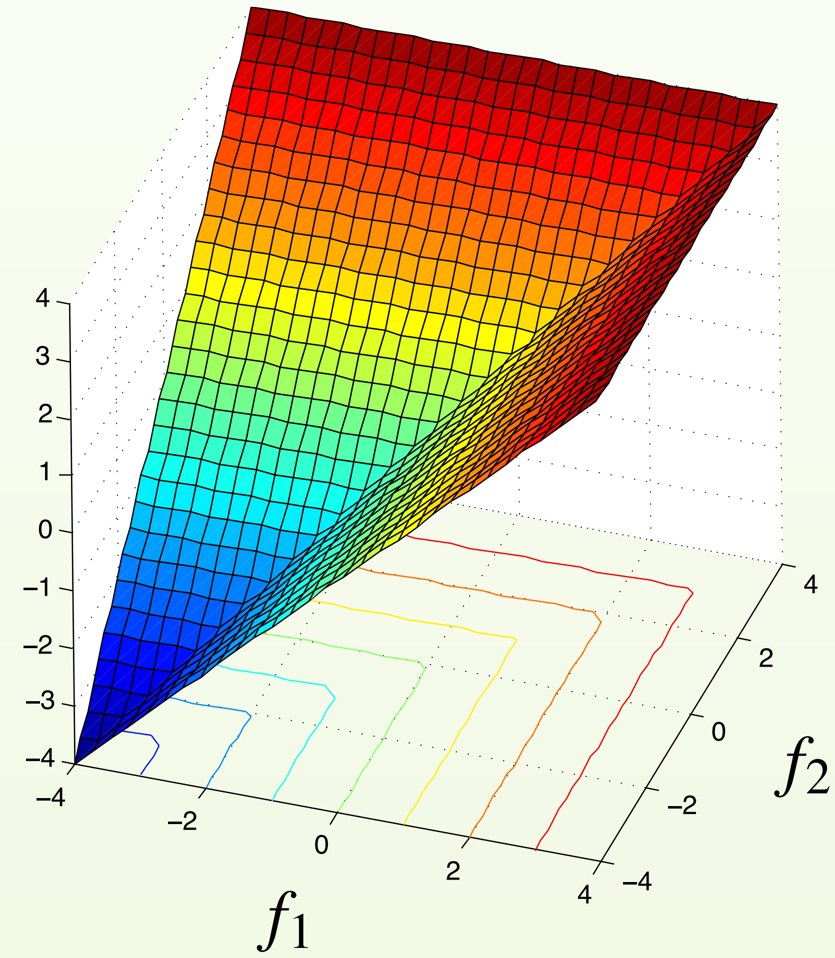
$$\sigma\left(\log\sum_k \exp\left(\mathbf{x}^\top\mathbf{Q}_k\mathbf{x}+\mathbf{w}_k^\top\mathbf{x}+b_k\right)\right)$$

Soft-maximum


Lucas Theis

$$\log \sum_k \exp f_k \qquad \approx \qquad \max_k f_k$$





Parameter reduction


Lucas Theis

Factored STM

$$p(\text{spike} = 1 \mid \mathbf{x}) =$$

$$\sigma\left(\log\sum_k \exp\left(\sum_m \alpha_{km}(\mathbf{u}_m^\top \mathbf{x})^2 + \mathbf{w}_k^\top \mathbf{x} + b_k\right)\right)$$

$$\mathbf{Q}_k = \sum_m \alpha_{km}\mathbf{u}_m\mathbf{u}_m^\top$$

Spike-triggered-mixture model (STM)

$$p(\text{spike} = 1 \mid \mathbf{x}) =$$

$$\sigma\left(\log\sum_k \exp\left(\mathbf{x}^\top \mathbf{Q}_k \mathbf{x} + \mathbf{w}_k^\top \mathbf{x} + b_k\right)\right)$$

Spike history dependency


Lucas Theis

Stimulus

$$p(\text{spike}=1 \mid \mathbf{x}) = \sigma\left(\log\frac{p(\mathbf{x}\mid \text{spike}=1)}{p(\mathbf{x}\mid \text{spike}=0)} + \log\frac{p(\text{spike}=1)}{p(\text{spike}=0)}\right)$$

$$p(\text{spike}=1 \mid \mathbf{x},\mathbf{z}) = \sigma\left(\log\frac{p(\mathbf{x},\mathbf{z}\mid \text{spike}=1)}{p(\mathbf{x},\mathbf{z}\mid \text{spike}=0)} + \log\frac{p(\text{spike}=1)}{p(\text{spike}=0)}\right)$$

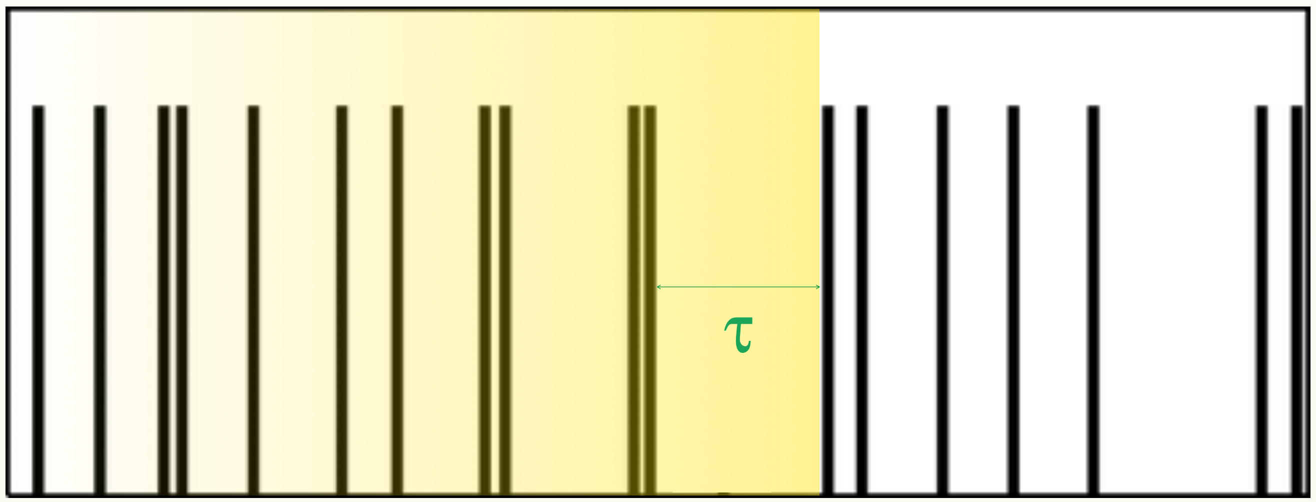
Spike history, e.g.

Naive Bayes assumption: $p(\mathbf{x},\mathbf{z}\mid \text{spike}) = p(\mathbf{x}\mid \text{spike})\,p(\mathbf{z}\mid \text{spike})$

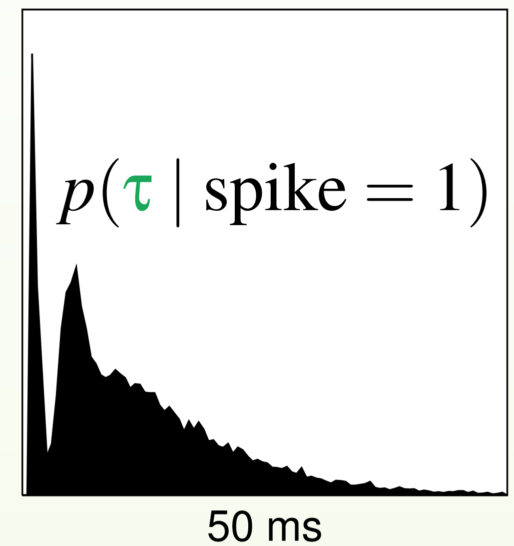
$$p(\text{spike}=1 \mid \mathbf{x},\mathbf{z}) = \sigma\left(\log\frac{p(\mathbf{x}\mid \text{spike}=1)}{p(\mathbf{x}\mid \text{spike}=0)} + \log\frac{p(\mathbf{z}\mid \text{spike}=1)}{p(\mathbf{z}\mid \text{spike}=0)} + \log\frac{p(\text{spike}=1)}{p(\text{spike}=0)}\right)$$

Spike history dependency

Lucas Theis



Interspike-interval distribution

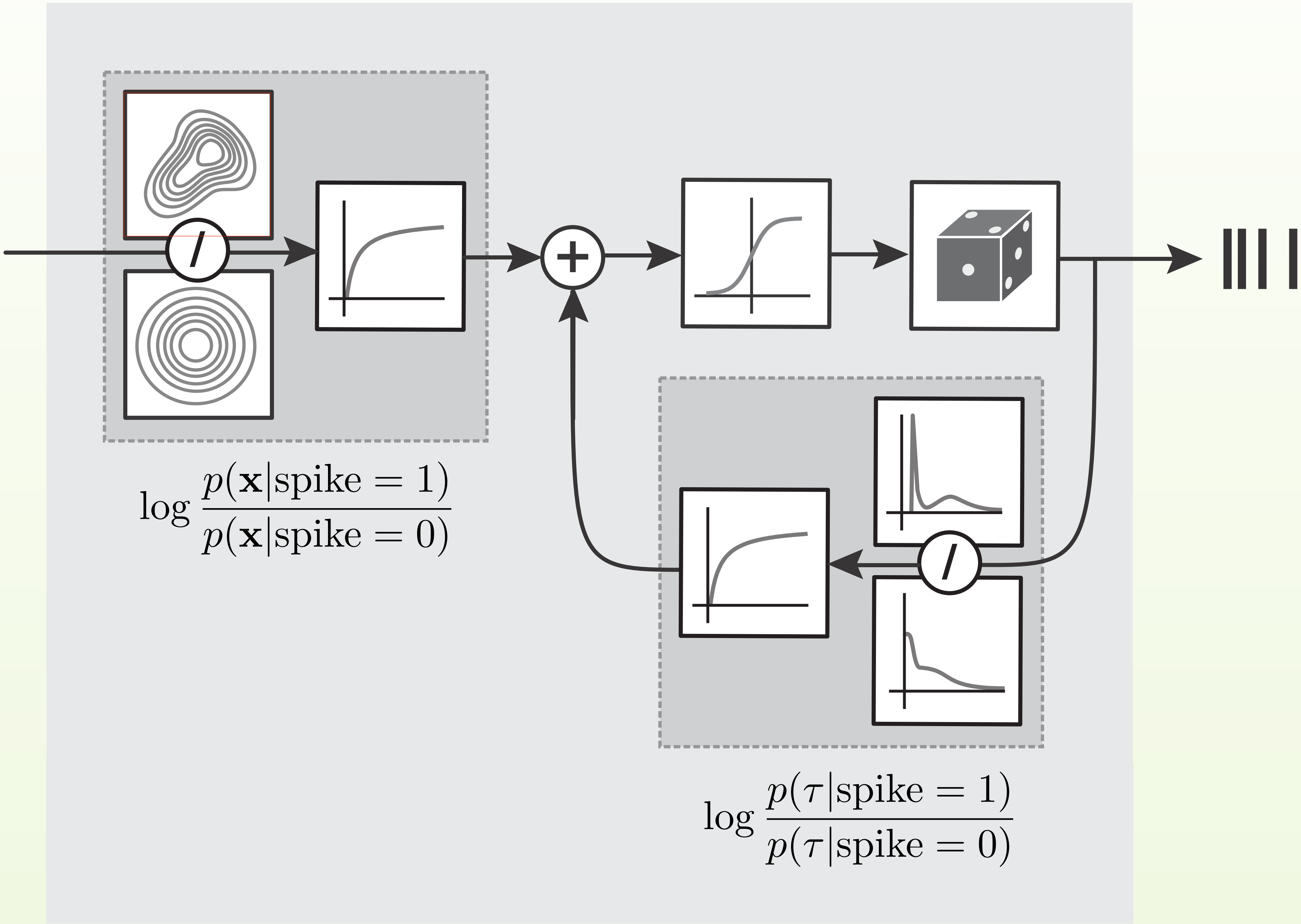


$$p(\text{spike}=1 \mid \mathbf{x},\tau) = \sigma\left(\log\frac{p(\mathbf{x}\mid\text{spike}=1)}{p(\mathbf{x}\mid\text{spike}=0)} + \log\frac{p(\tau\mid\text{spike}=1)}{p(\tau\mid\text{spike}=0)} + \log\frac{p(\text{spike}=1)}{p(\text{spike}=0)}\right)$$

$$p(\text{spike}=1 \mid \mathbf{x},\tau) = \sigma\left(\log\sum_k \exp\left(\sum_m \alpha_{km}(\mathbf{u}_m^\top\mathbf{x})^2 + \mathbf{w}_k^\top\mathbf{x} + b_k\right) + \mathbf{v}^\top\phi(\tau)\right)$$

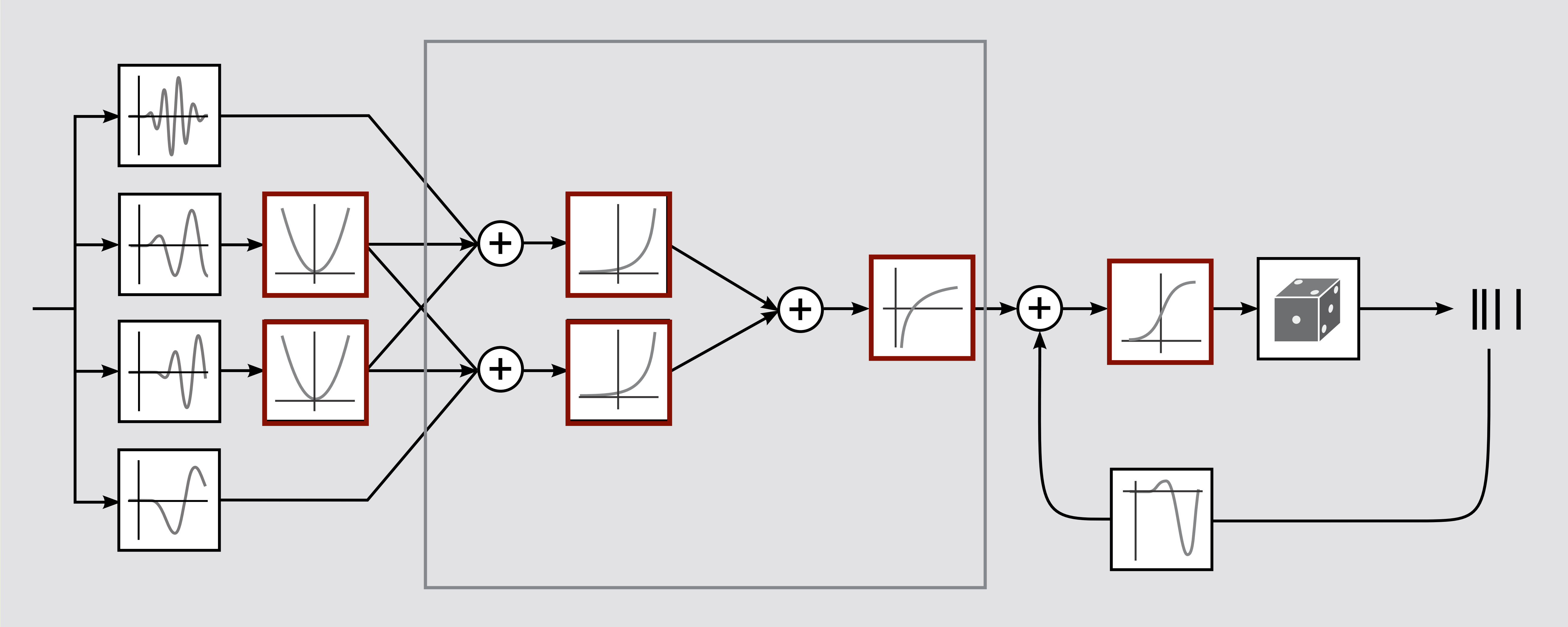
Model summary

Lucas Theis



Model summary as neural network (LN-LNLN-LNB)

Lucas Theis

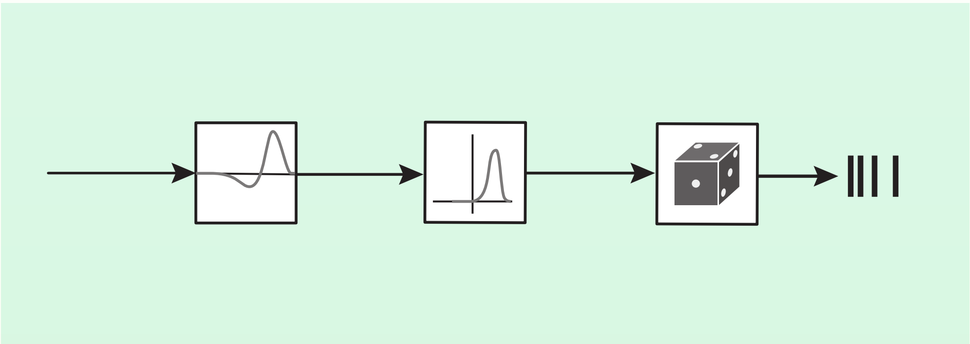


simple and complex cell preprocessing

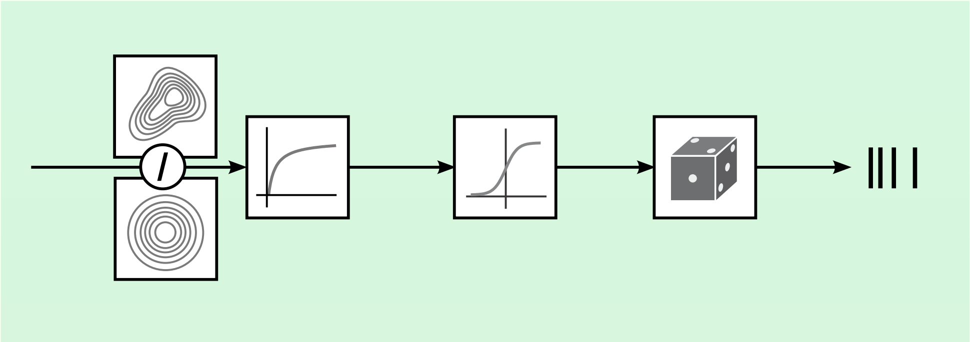
nonlinear "dendritic" integration (log-sum-exp)

GLM logistic spike generation (spike-history dependent)

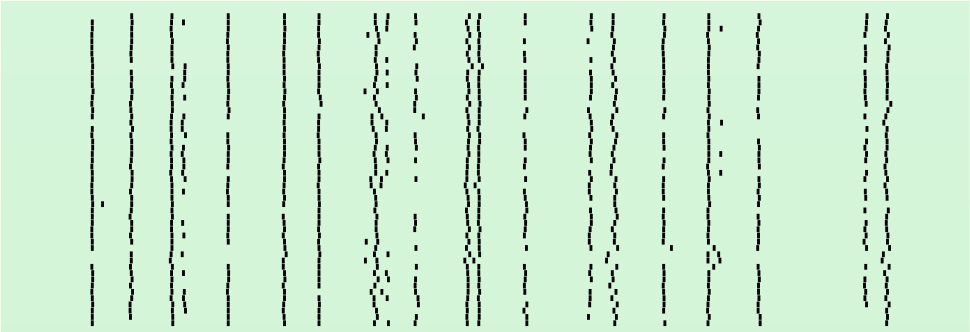
Overview



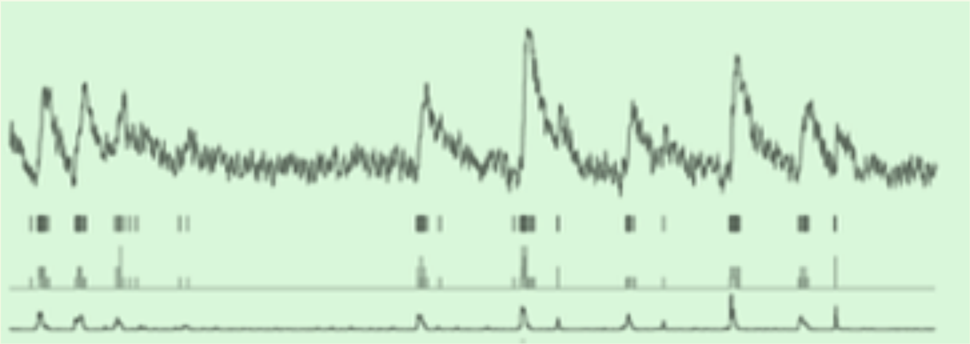
1 Generalized linear models (**GLMs**)



2 Spike-triggered-mixture model (**STM**)



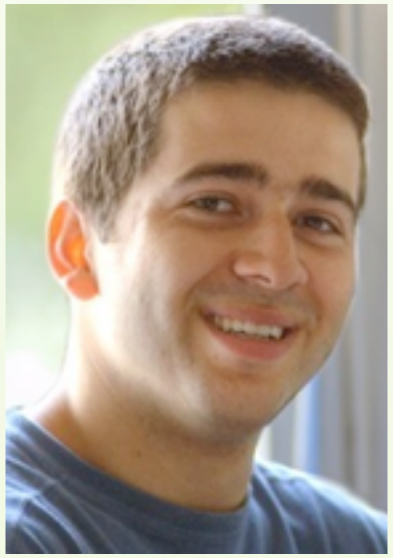
3 **Empirical results**



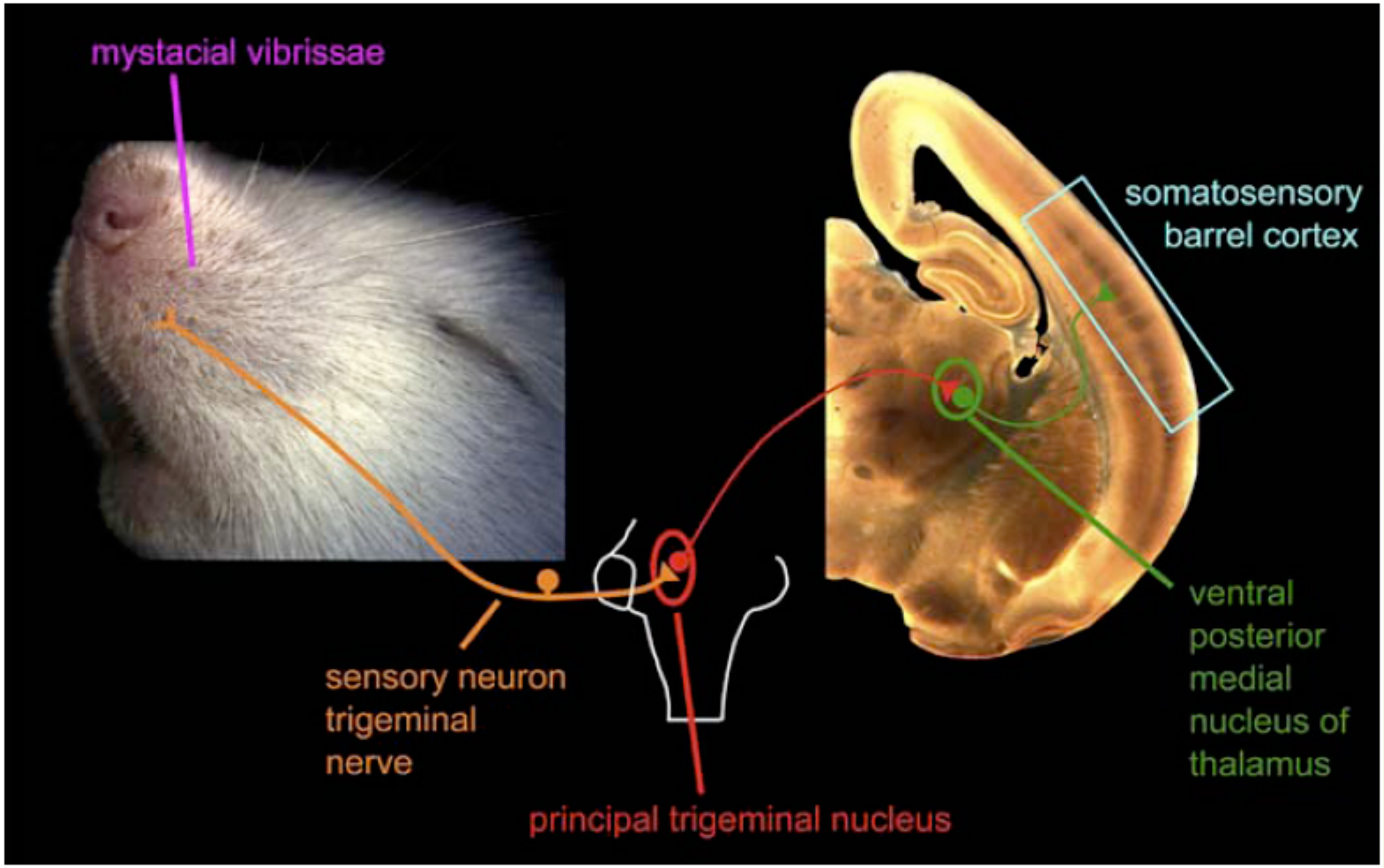
4 Spike detection

Whisker cells

Cornelius Schwarz



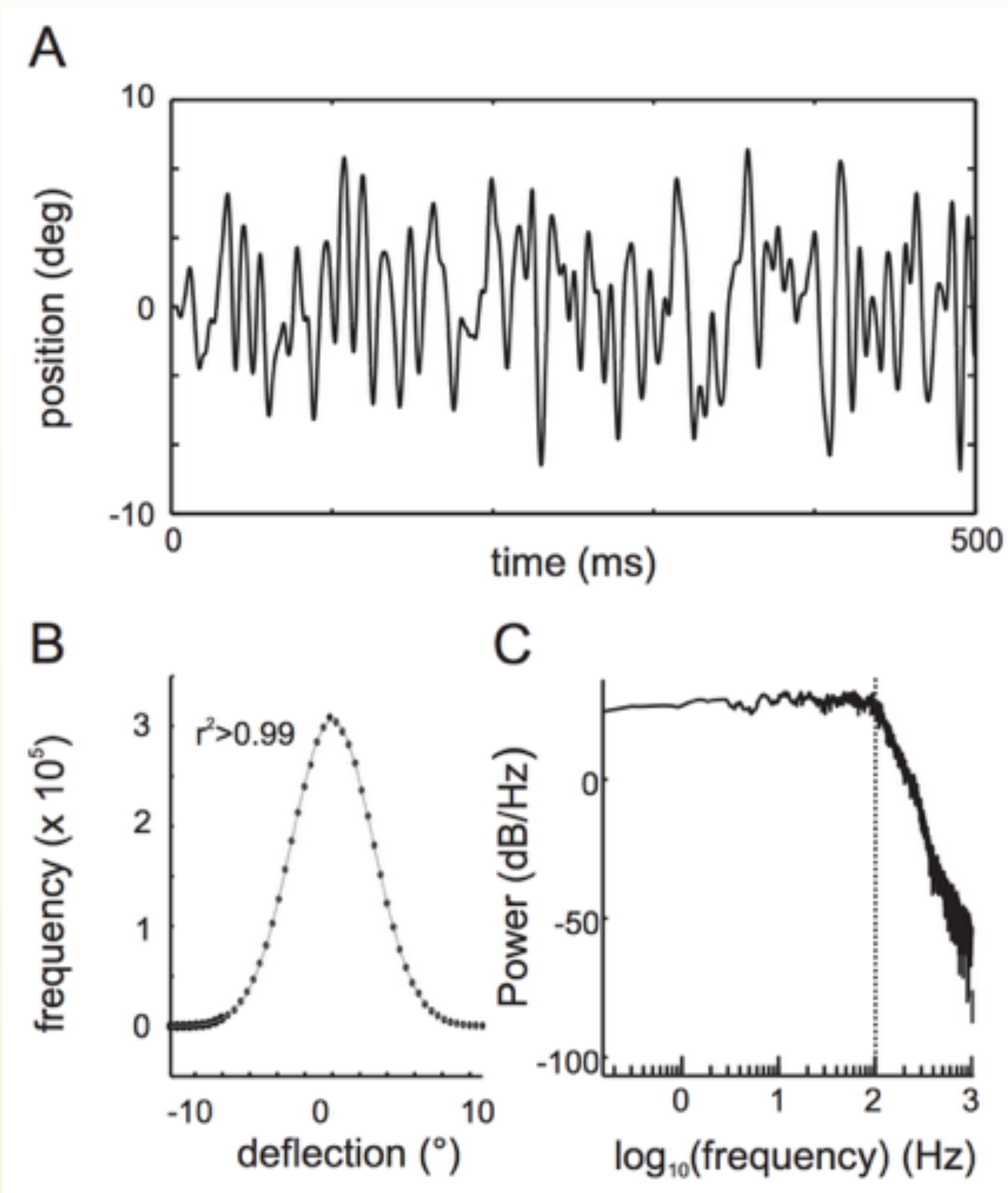
Andre Chagas



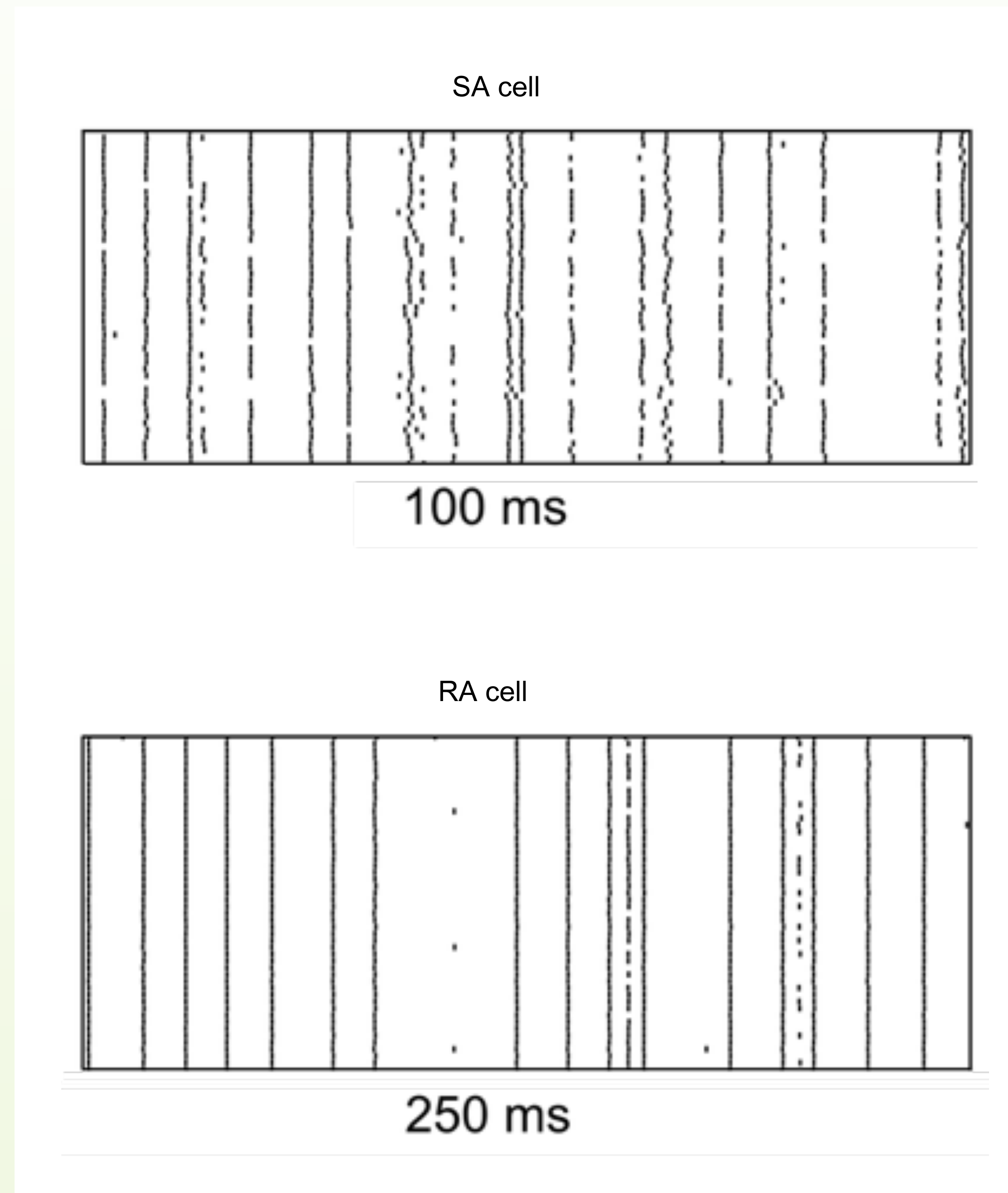
Chagas & Schwarz

Whisker cells

Stimulus

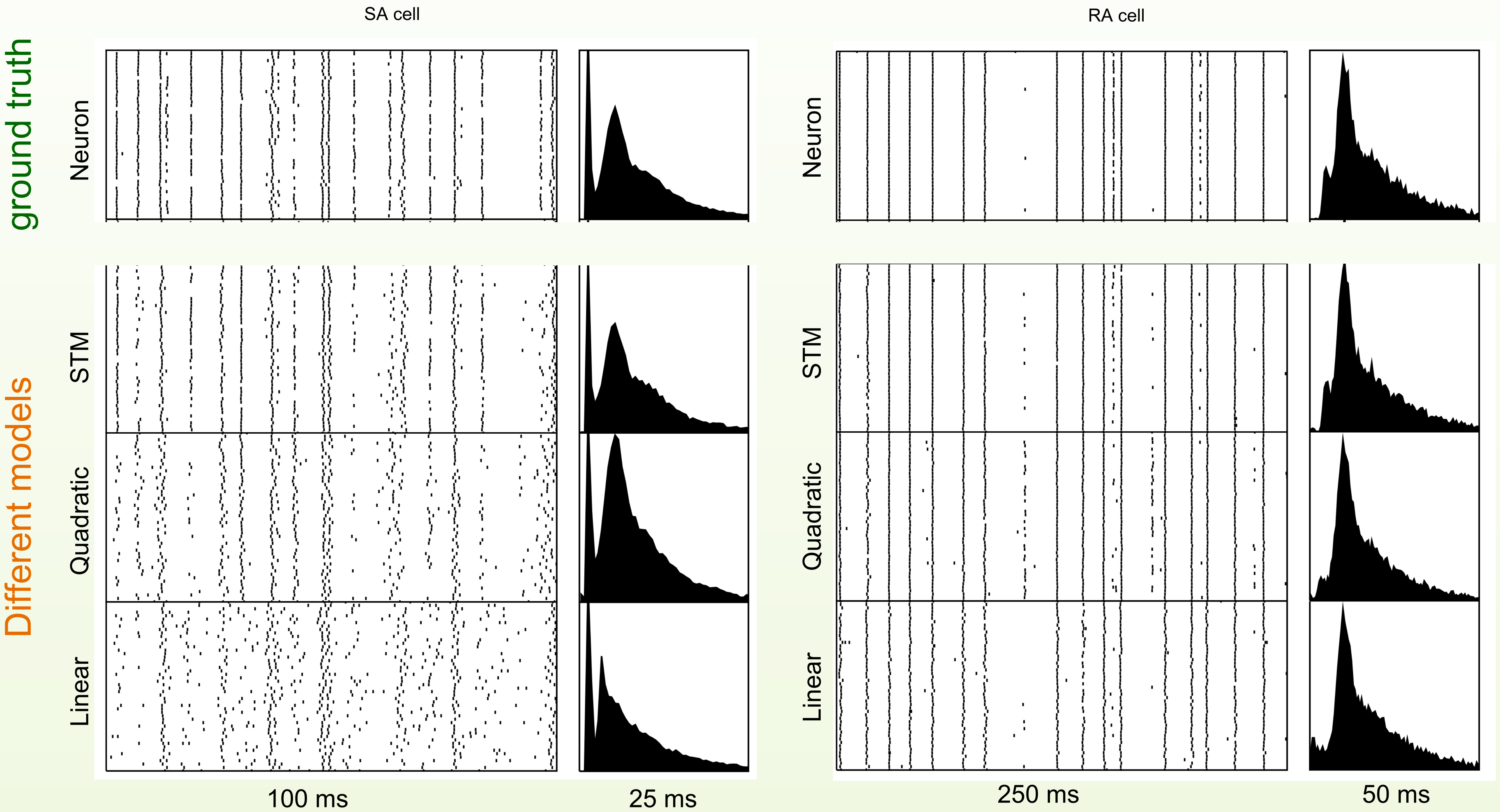


Responses

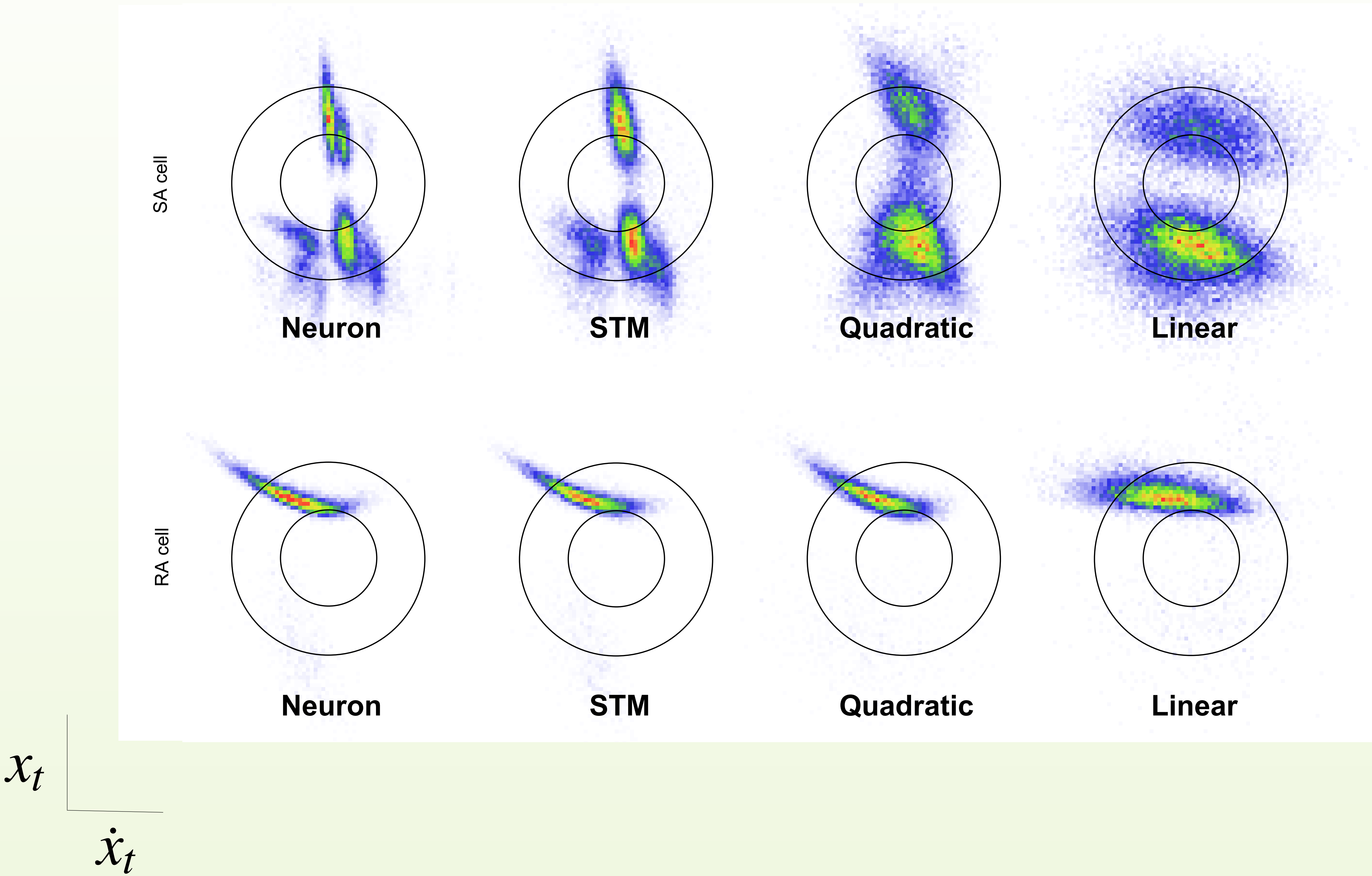


Chagas & Schwarz

Simulated spike trains



Spike-triggered distribution

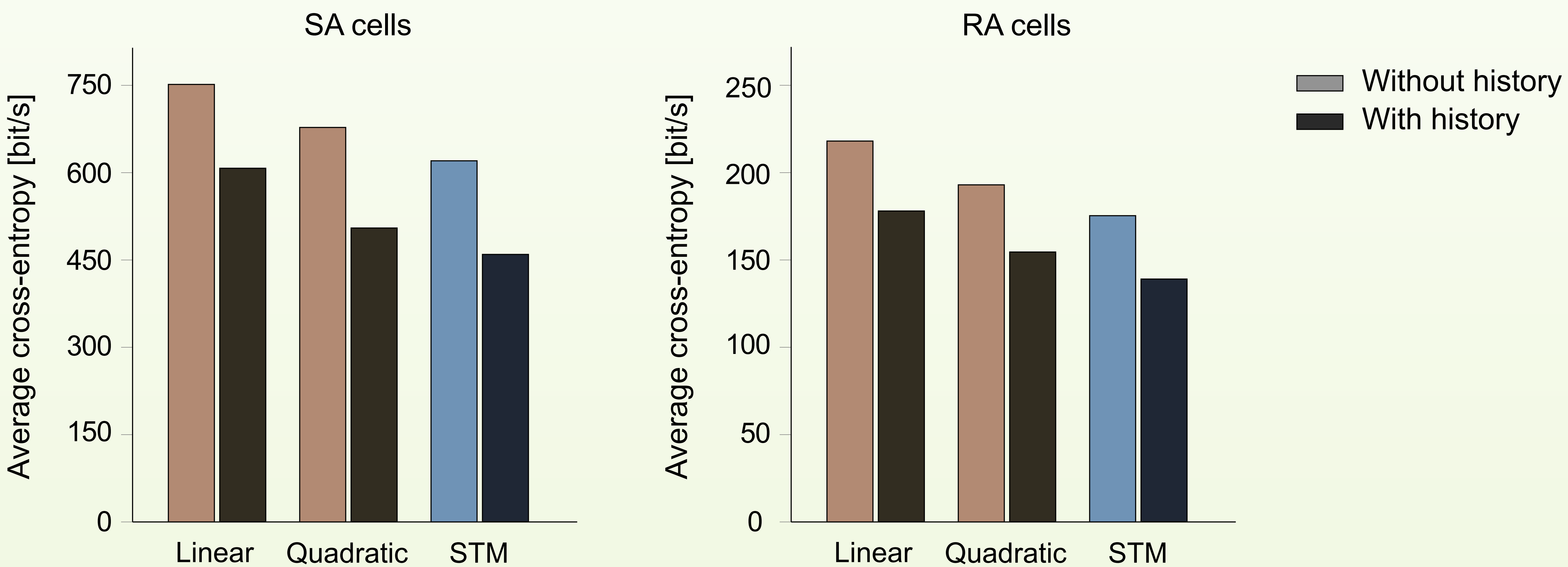


Quantitative comparison

Lucas Theis

Cross-entropy / negative log-likelihood:

$$-\frac{1}{T}\sum_t \log p(y_t \mid \mathbf{x}_t)$$

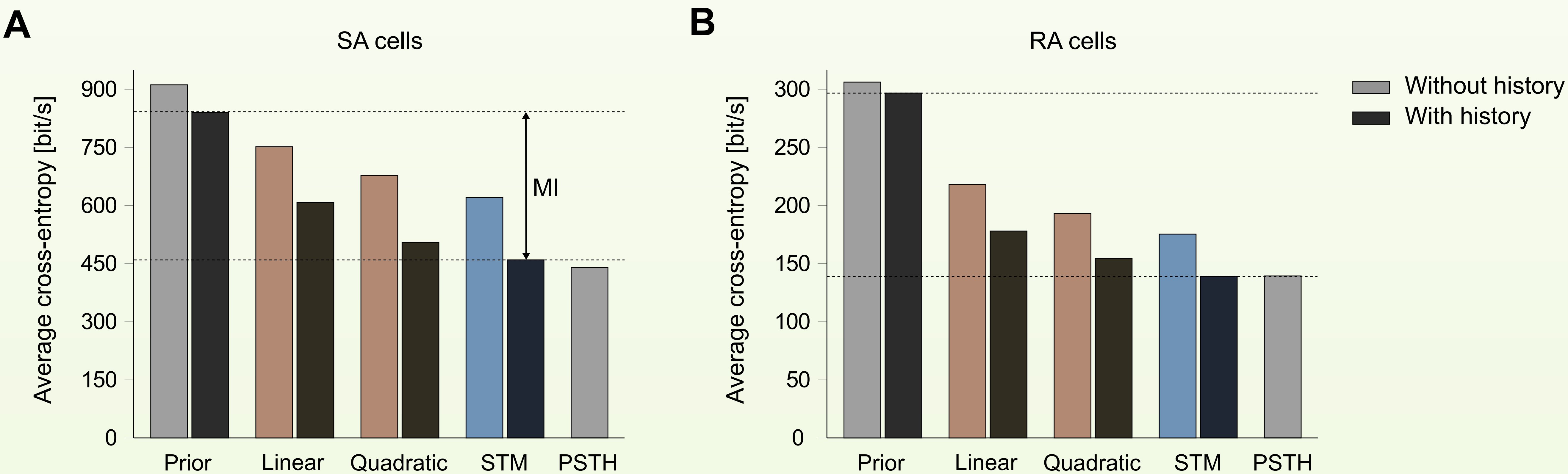


Quantitative comparison

Lucas Theis

Cross-entropy / negative log-likelihood:

$$-\frac{1}{T}\sum_t \log p(y_t \mid \mathbf{x}_t)$$



Information transmission

Spike train

$$I[\mathbf{y}, \mathbf{x}] = H[\mathbf{y}] - H[\mathbf{y} \mid \mathbf{x}]$$

Stimulus

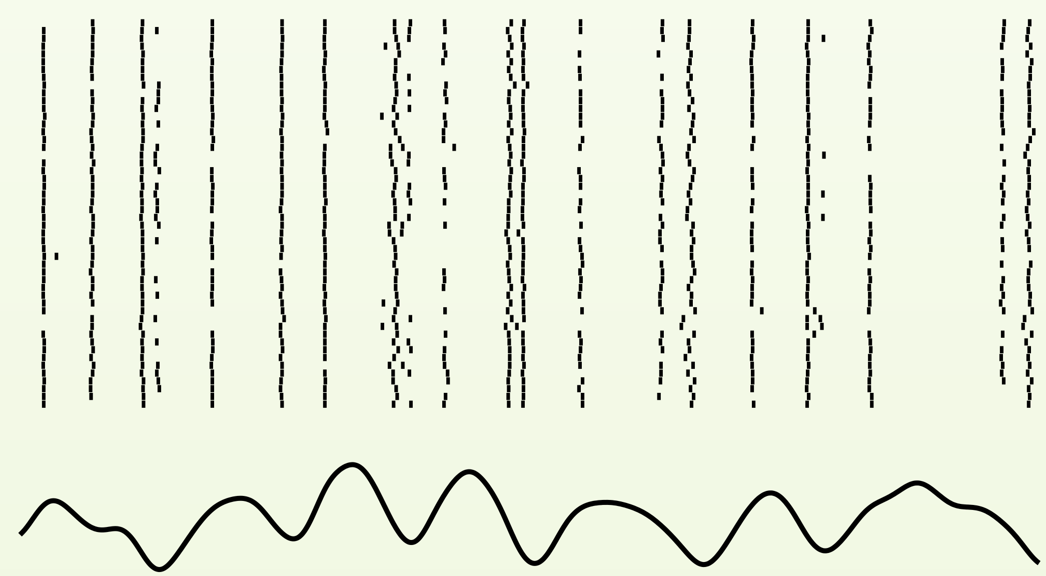
"Direct method" (DM)

- Use histograms for estimating entropy

+ Simple

– Direction of bias unclear

– Requires repeated stimulation with same stimulus



de Ruyter van Steveninck et al. (1997)

Information transmission

Spike train

$$I[\mathbf{y}, \mathbf{x}] = H[\mathbf{y}] - H[\mathbf{y} \mid \mathbf{x}]$$

Stimulus

Model-based method

- Use cross-entropy for estimating entropy

+ Conservative estimate

+ Does not require repeated experiments

– Takes longer due to model fitting

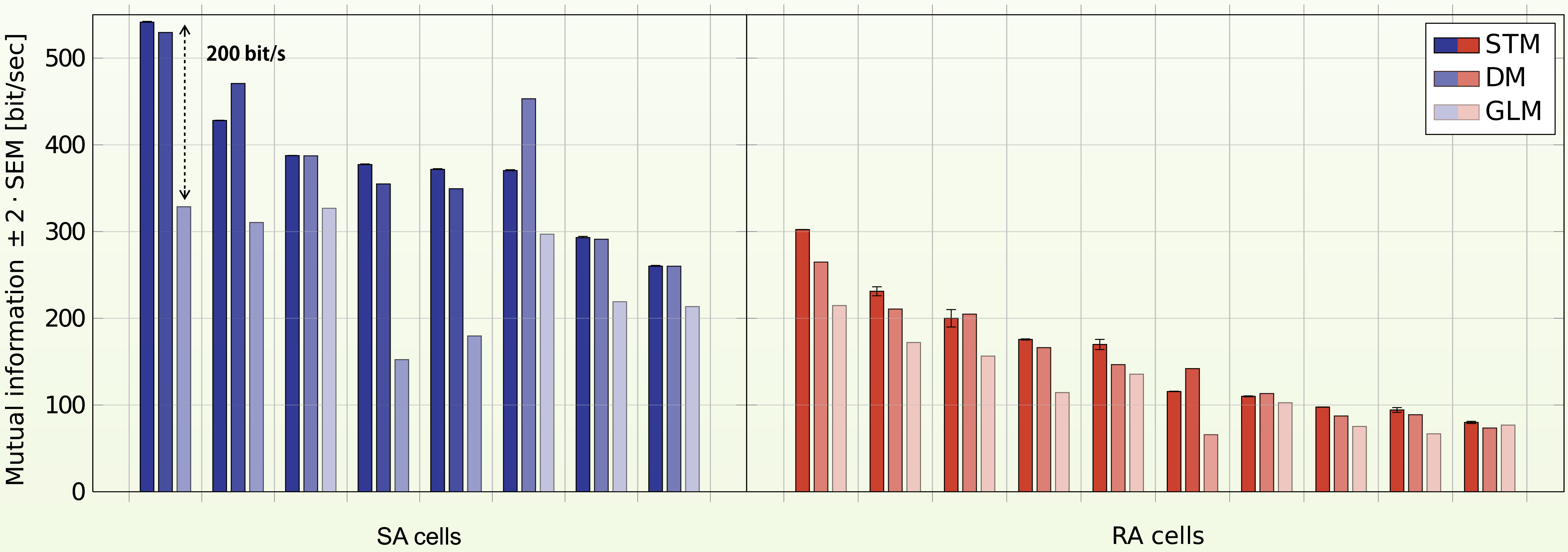
Model distribution

$$H[\mathbf{y} \mid \mathbf{x}] \leq E[-\log p(\mathbf{y} \mid \mathbf{x})]$$

Negative log-likelihood/
cross-entropy

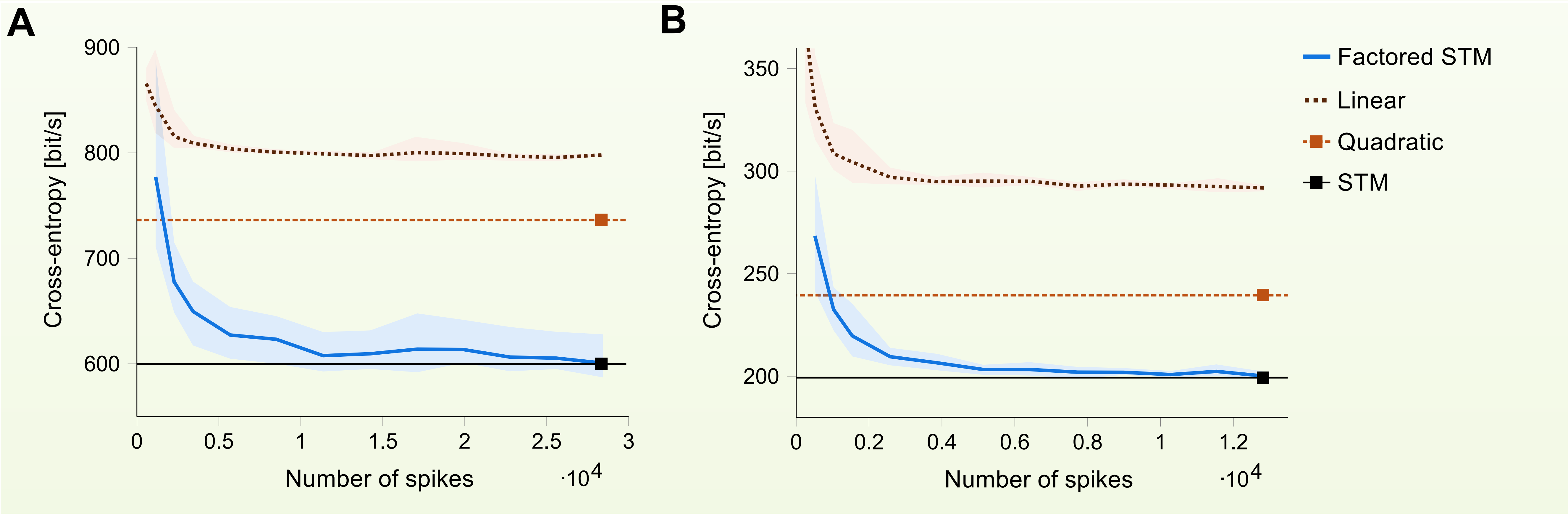
Information rates

Lucas Theis

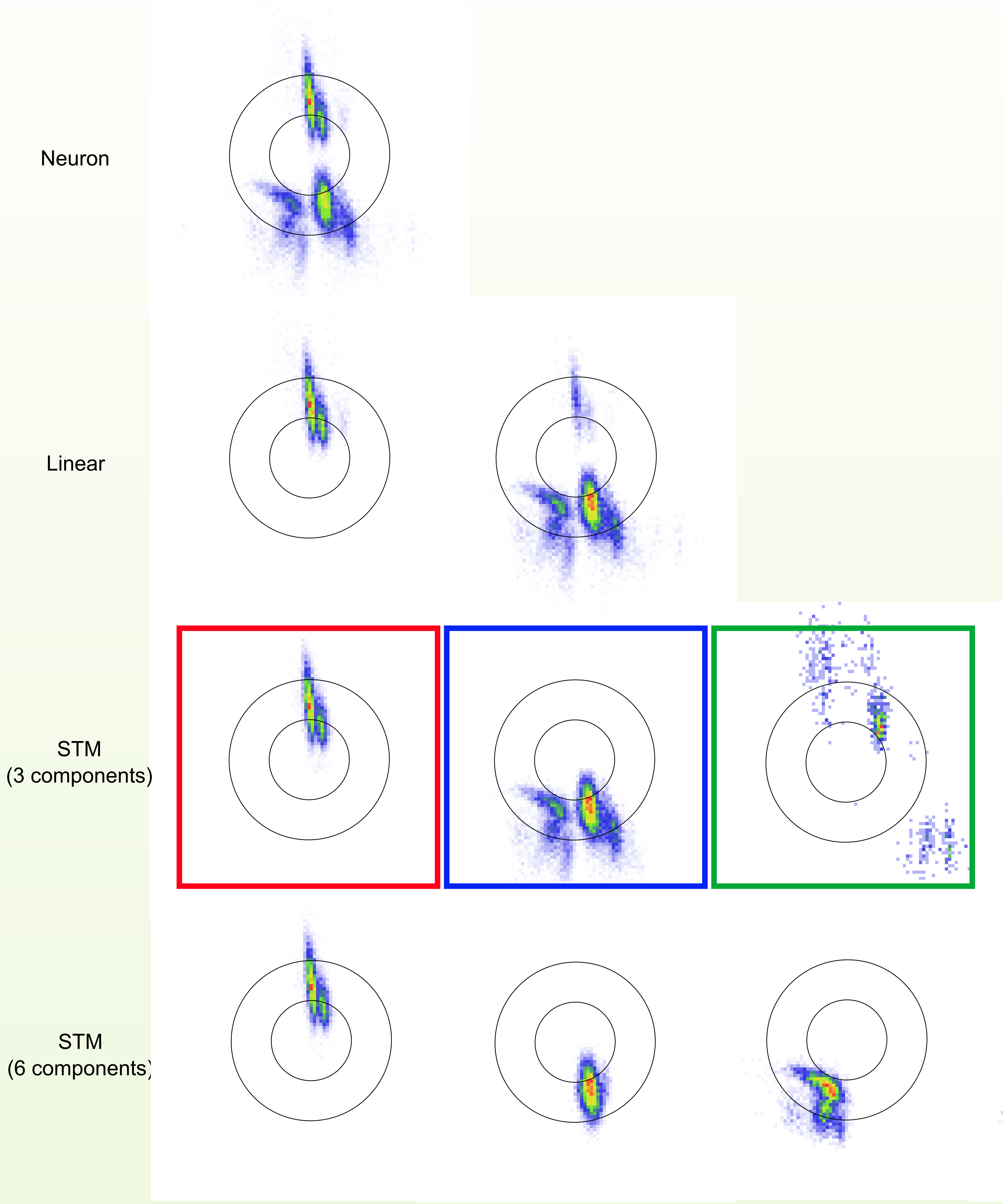


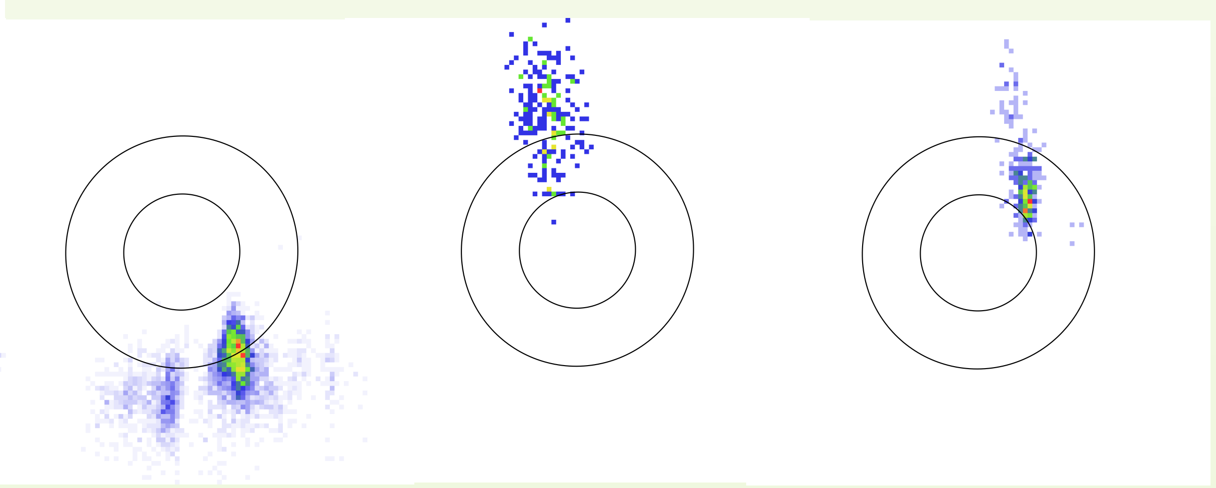
How much data is enough?

Lucas Theis



Disentangling nonlinear behavior

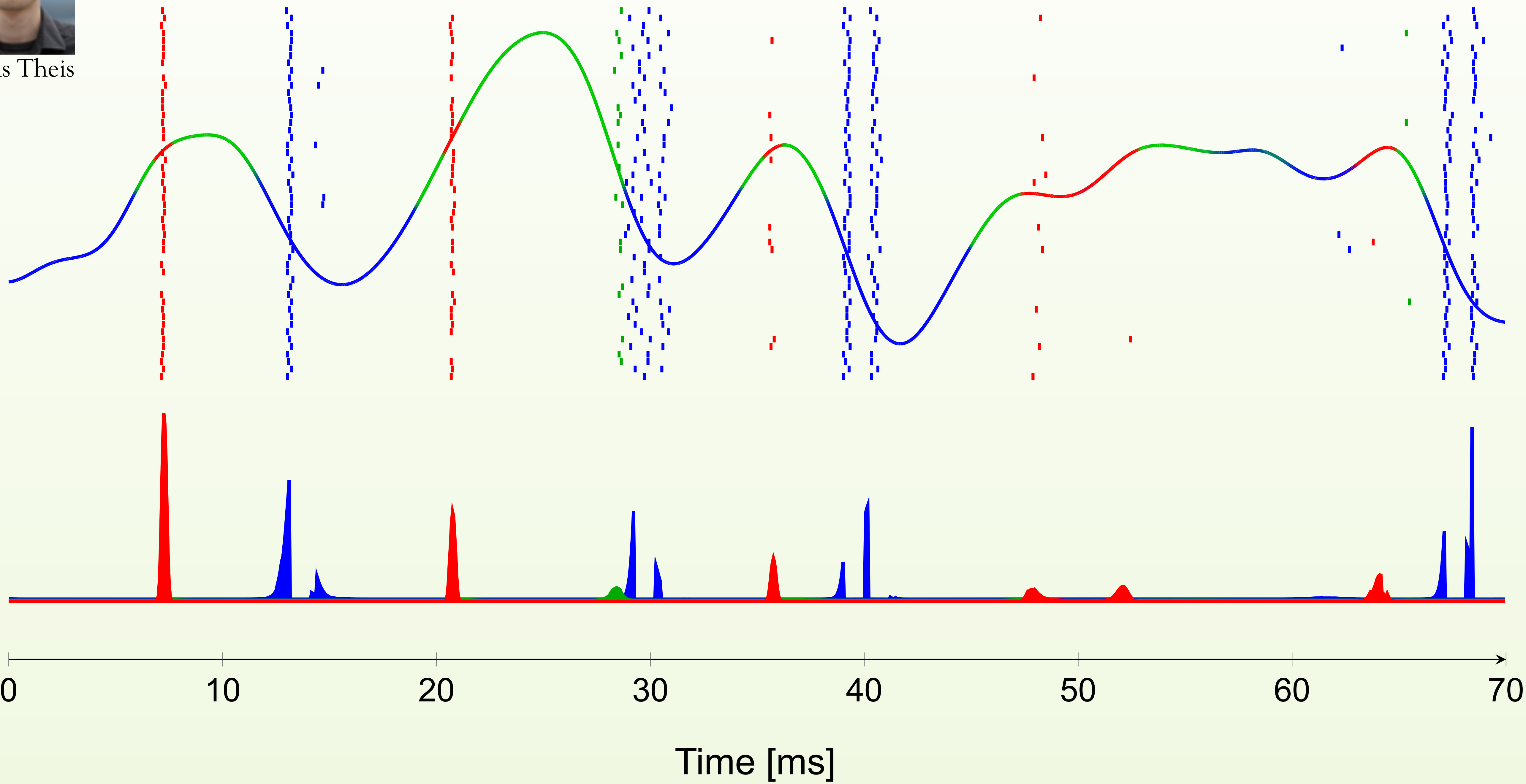




Lucas Theis

Disentangling nonlinear behavior


Lucas Theis



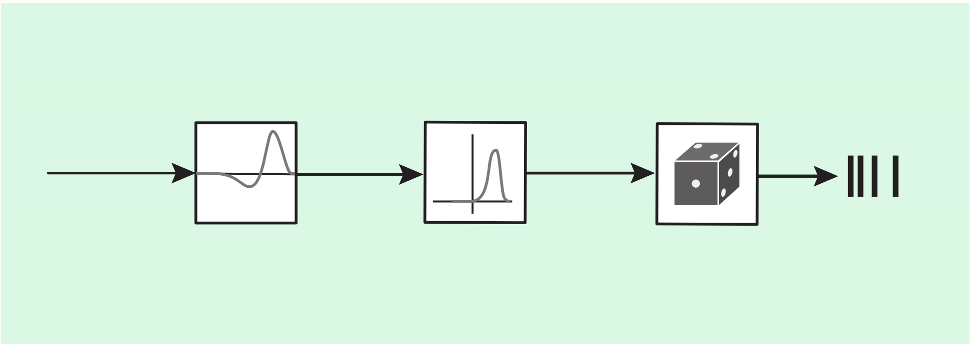
Literature

Lucas Theis

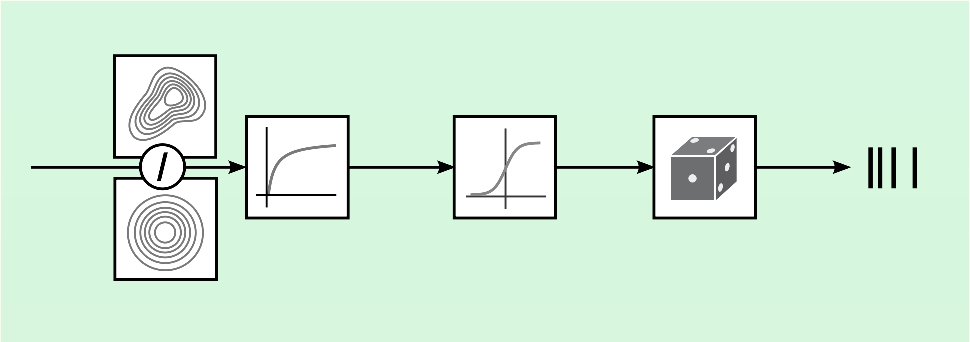
Theis et al.: "*Beyond GLMs: A Generative Mixture Modeling Approach to Neural System Identification.*" PloS Computational Biology, **9**:11, 2013.

Software: http://bethgelab.org/code/theis2013a/

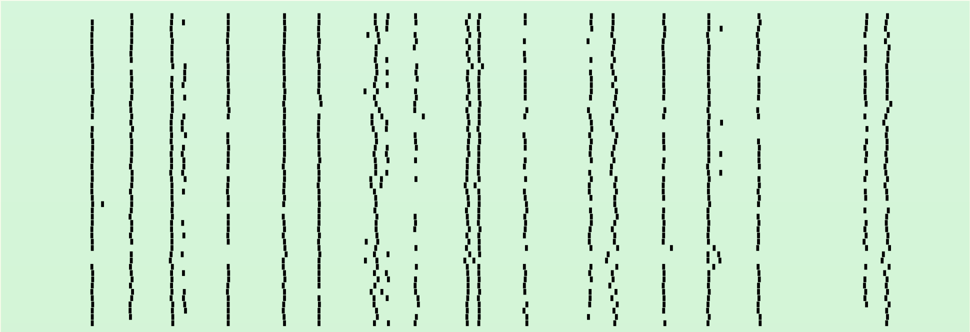
Overview



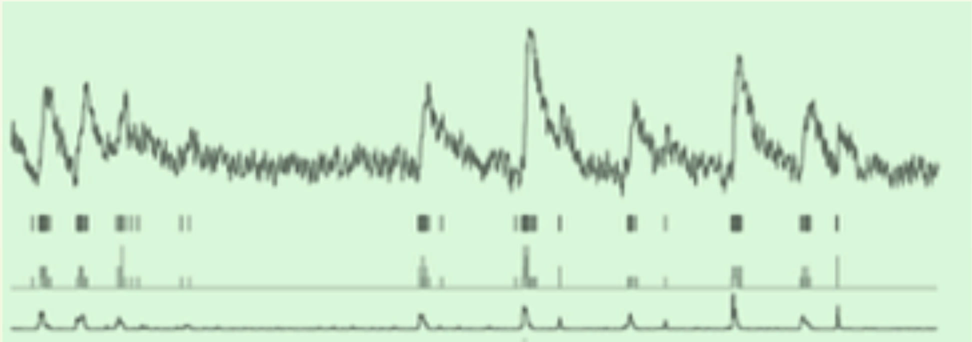
1 Generalized linear models (**GLMs**)



2 Spike-triggered-mixture model (**STM**)

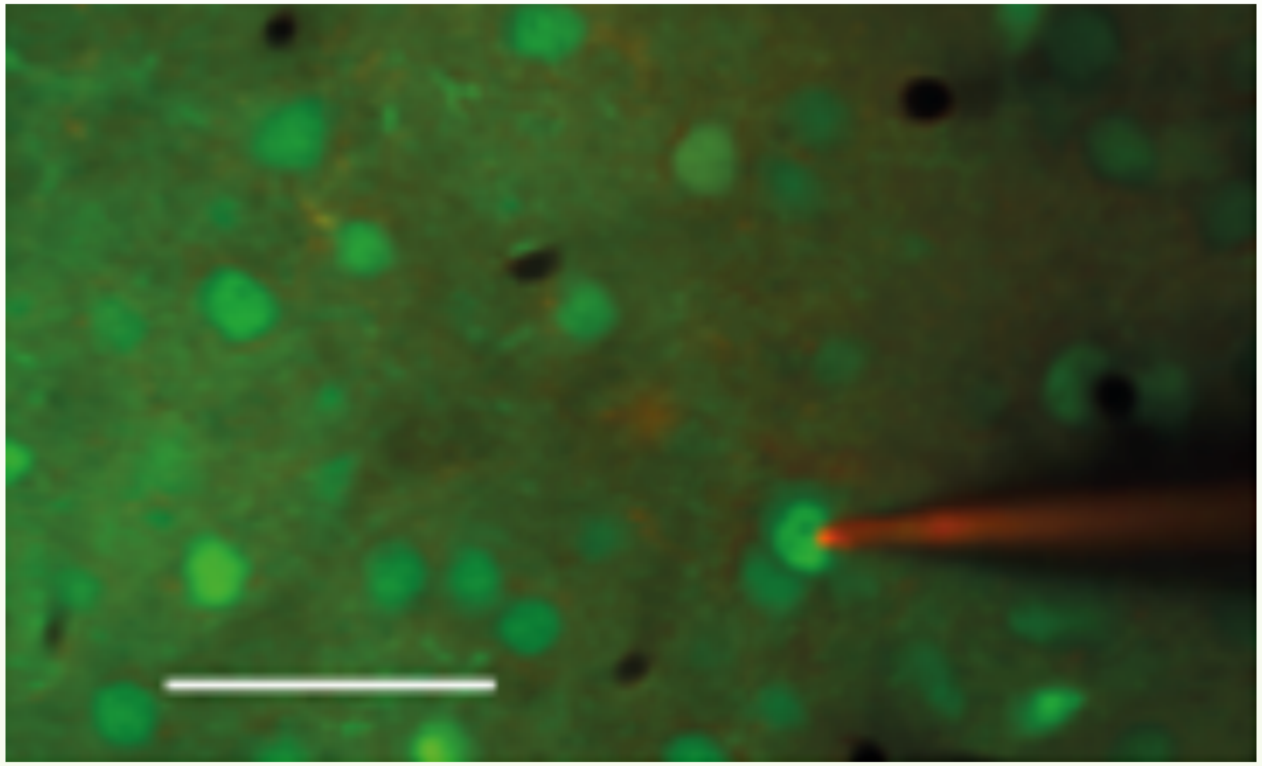


3 Empirical results



4 **Spike detection**

Spike detection from Calcium measurements



4 datasets from V1 and retina

Calcium indicators: OGB-1 and GCamp6)

$$\lambda_{\mathrm{STM}}(\boldsymbol{x}_t) = \sum_{k=1}^{K} \exp\left(\sum_{m=1}^{M} \beta_{km} (\boldsymbol{u}_m^\top \boldsymbol{x}_t)^2 + \boldsymbol{w}_k^\top \boldsymbol{x}_t + \boldsymbol{b}_k \right).$$

$$p(k_t \mid \boldsymbol{x}_t) = \frac{\lambda(\boldsymbol{x}_t)^k}{k!} e^{-\lambda(\boldsymbol{x}_t)}.$$

Model comparison


Lucas Theis

Theis, Berens, Froudarakis, Reimer,
Román Rosón, Baden, Euler, Tolias, Bethge
http://biorxiv.org/content/early/2015/02/27/010777


Philipp Berens

Algorithm	Approach	Technique	Reference
STM	Supervised	STM	This paper
SI08	Supervised	PCA+SVM	(Sasaki et al., 2008)
PP14	Generative	MCMC sampling	(Pnevmatikakis et al., 2014)
OD13	Template matching	Finite rate innovation	(Oñativia et al., 2013)
VP10	Generative	MAP estimation	(Vogelstein et al., 2010)
VP09	Generative	SMC sampling	(Vogelstein et al., 2009)
YF06	Generative	Deconvolution	(Yaksi and Friedrich, 2006)

4 datasets of simultaneously recorded Calcium signals and spikes from V1 and retina: 75 traces from 67 neurons, in total ~ 89.000 spikes

Dataset 1: 16 neurons, mouse V1, in-vivo, anesthetized, fast 3D AOD-based imaging at ~320 Hz, OGB-1

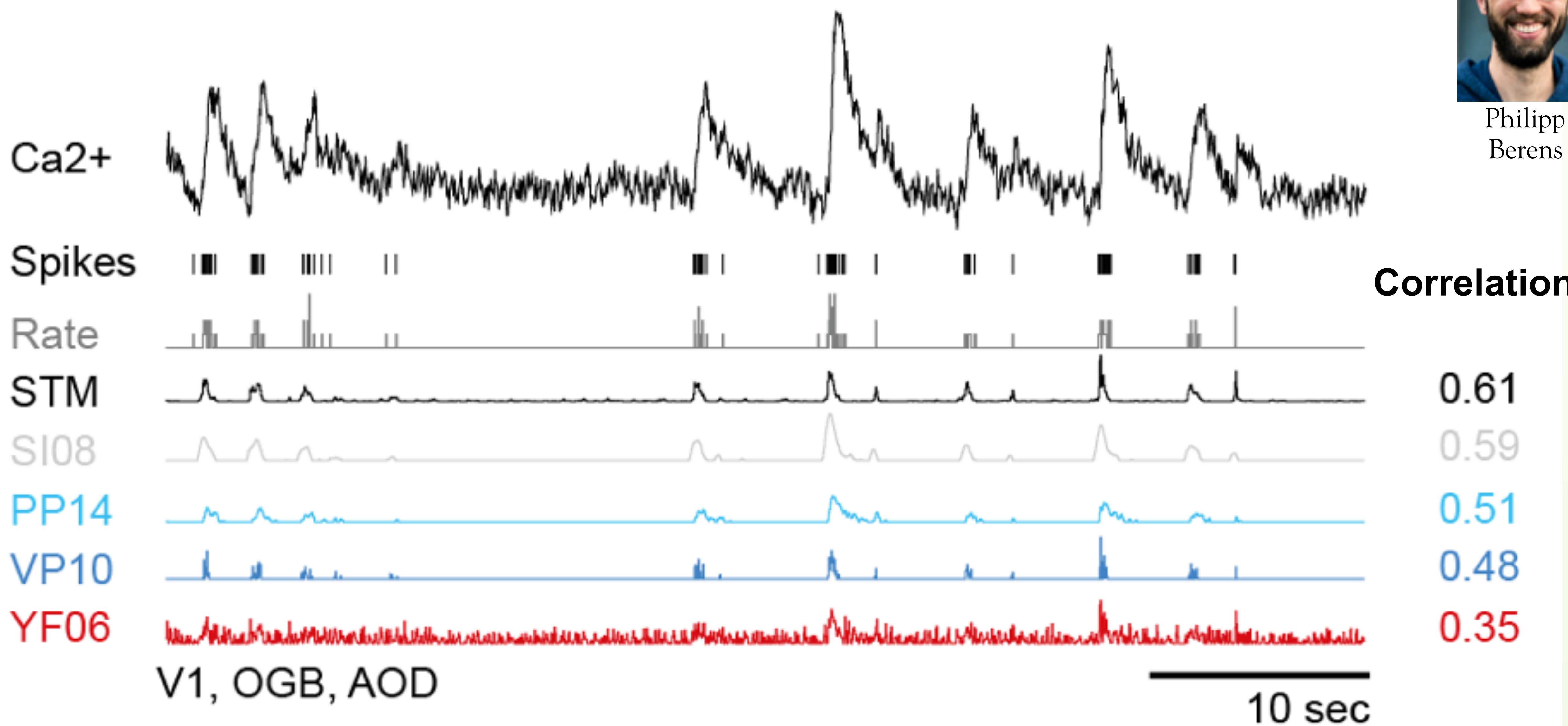
Dataset 2: 31 neurons, mouse V1, in- vivo, line scanning at ~12 Hz, OGB-1

Dataset 3: 11 neurons, mouse V1, in-vivo, resonance scanner at ~59 Hz, genetic indicator GCamp6s

Dataset 3: 9 retinal ganglion cells, mouse retina, in-vitro, line-scanning at ~8 Hz, OGB-1

Model comparison (example traces)


Lucas Theis



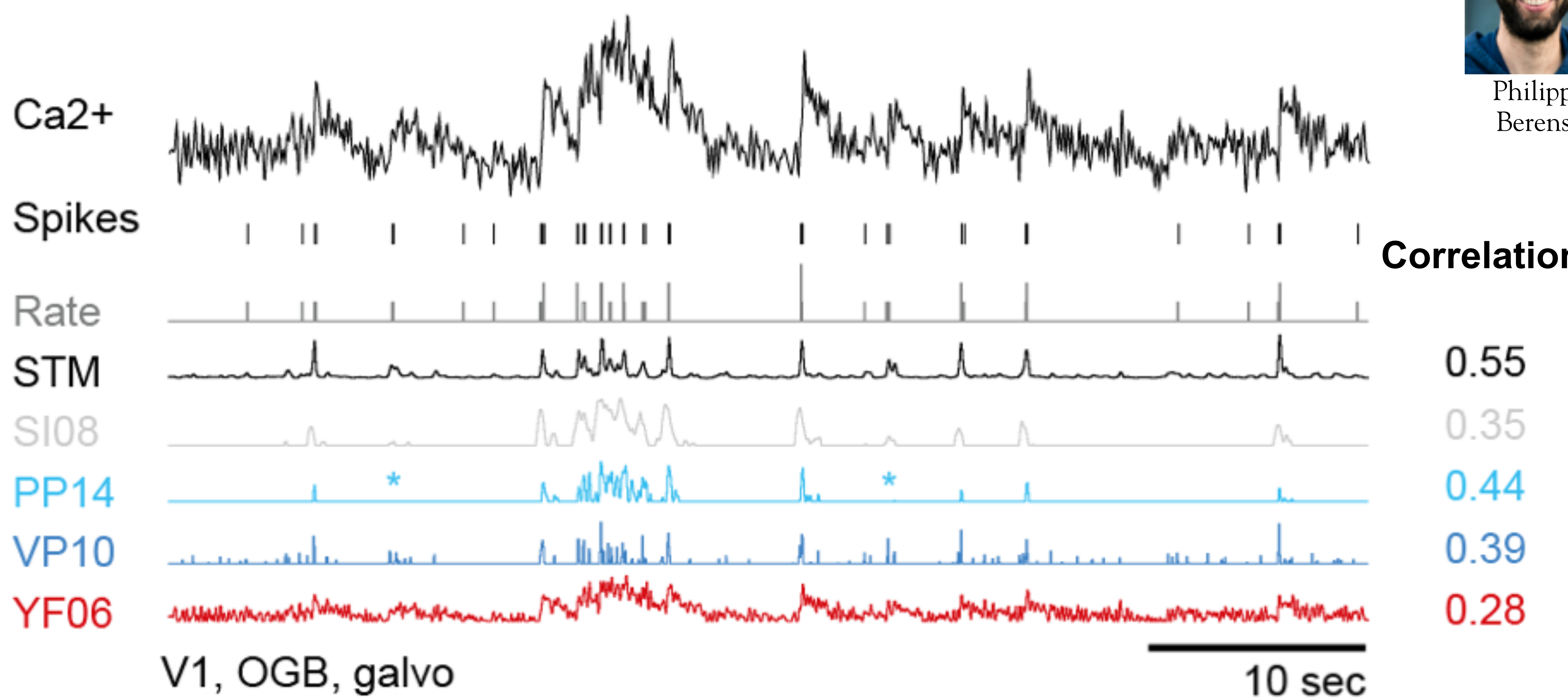

Philipp Berens

(time resolution: 40 msec)

Model comparison (example traces)

Lucas Theis

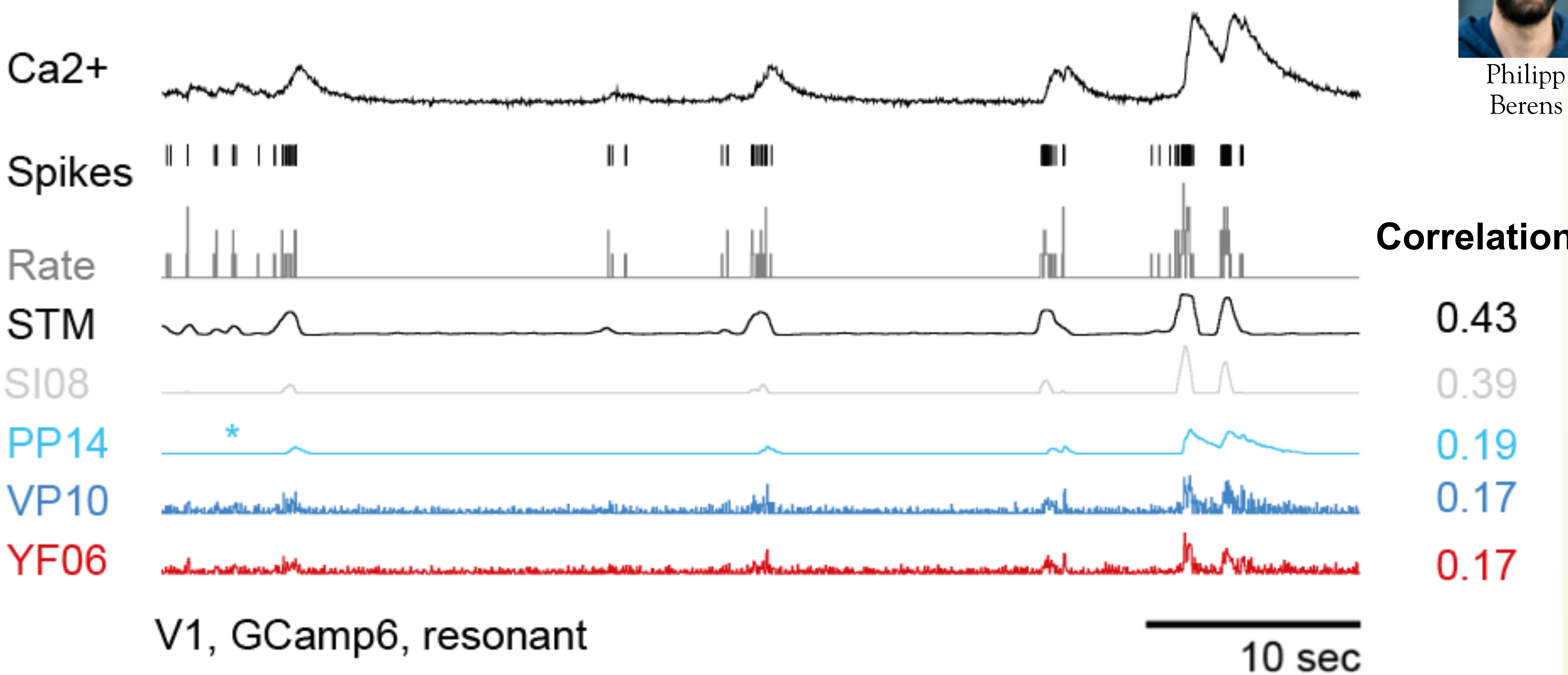
Philipp Berens



(time resolution: 40 msec)

Model comparison (example traces)


Lucas Theis

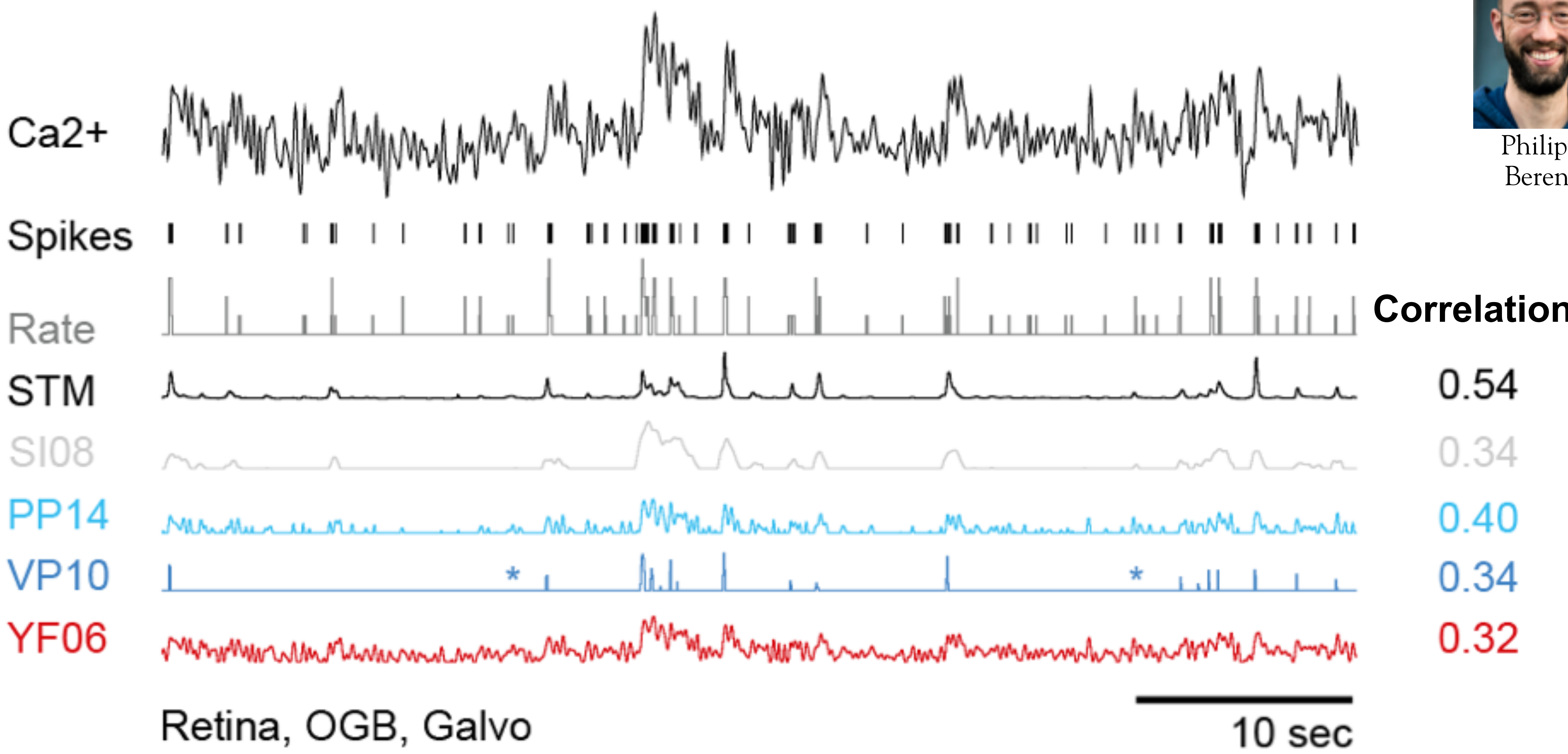



Philipp Berens

(time resolution: 40 msec)

Model comparison (example traces)


Lucas Theis

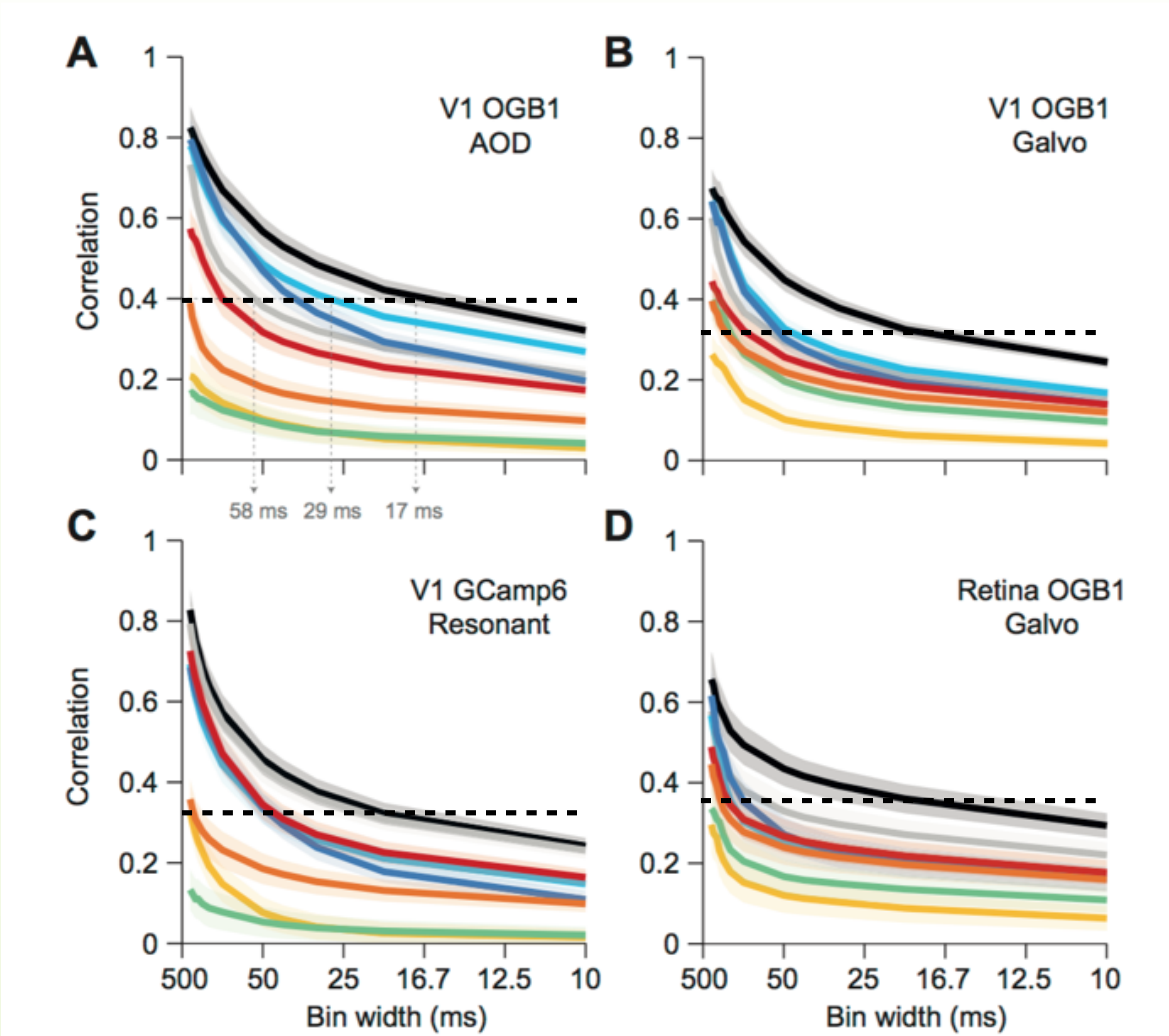



Philipp Berens

(time resolution: 40 msec)

Effect of temporal resolution on spike detection performance

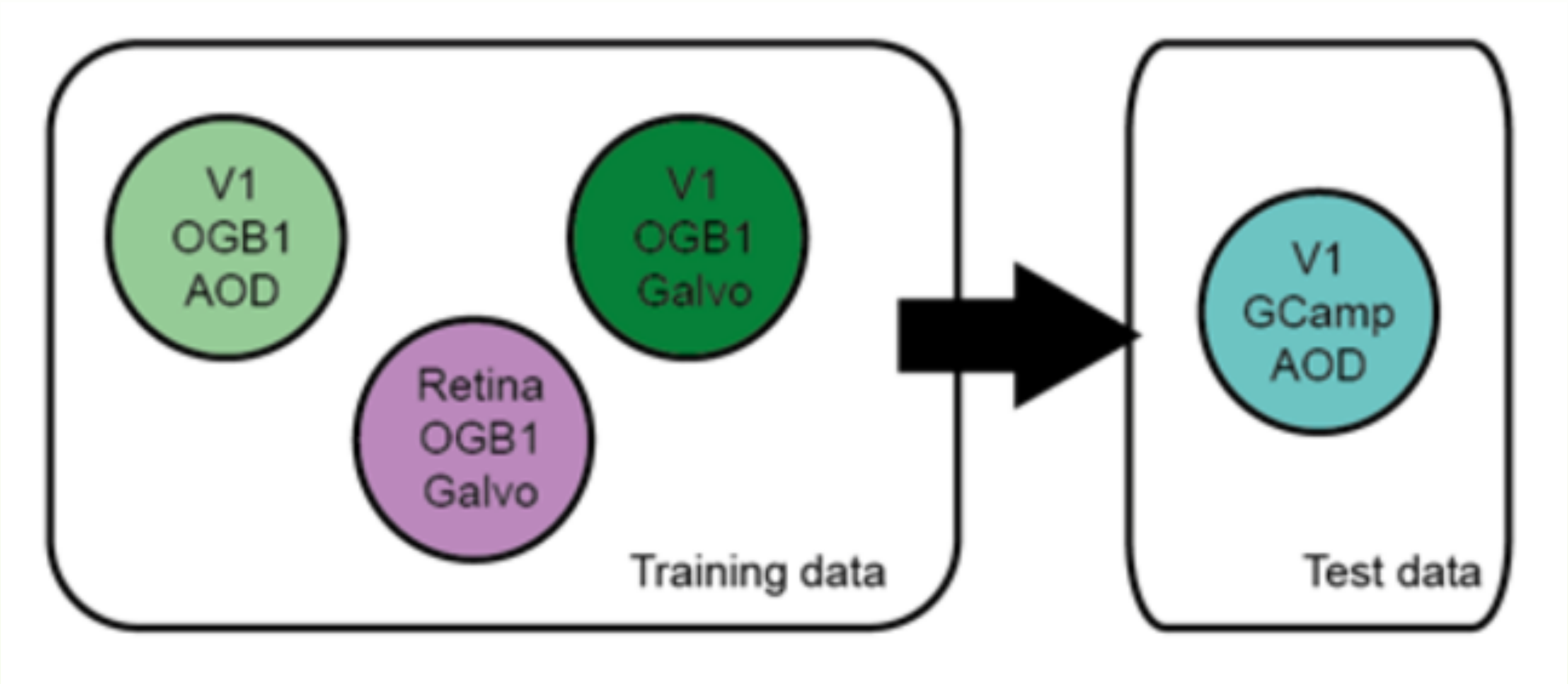
Lucas Theis



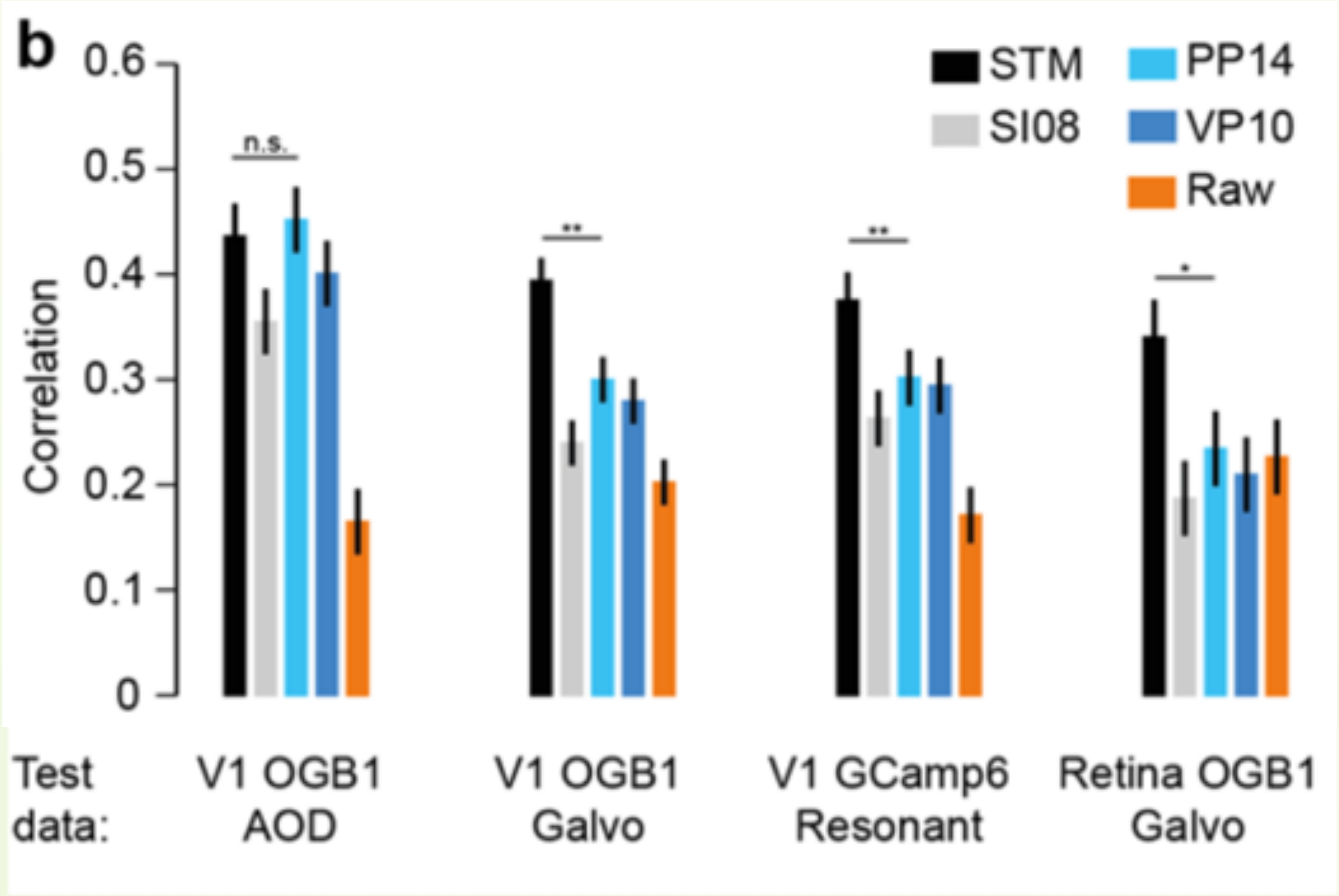
Philipp Berens

Generalization across experiments

Lucas Theis



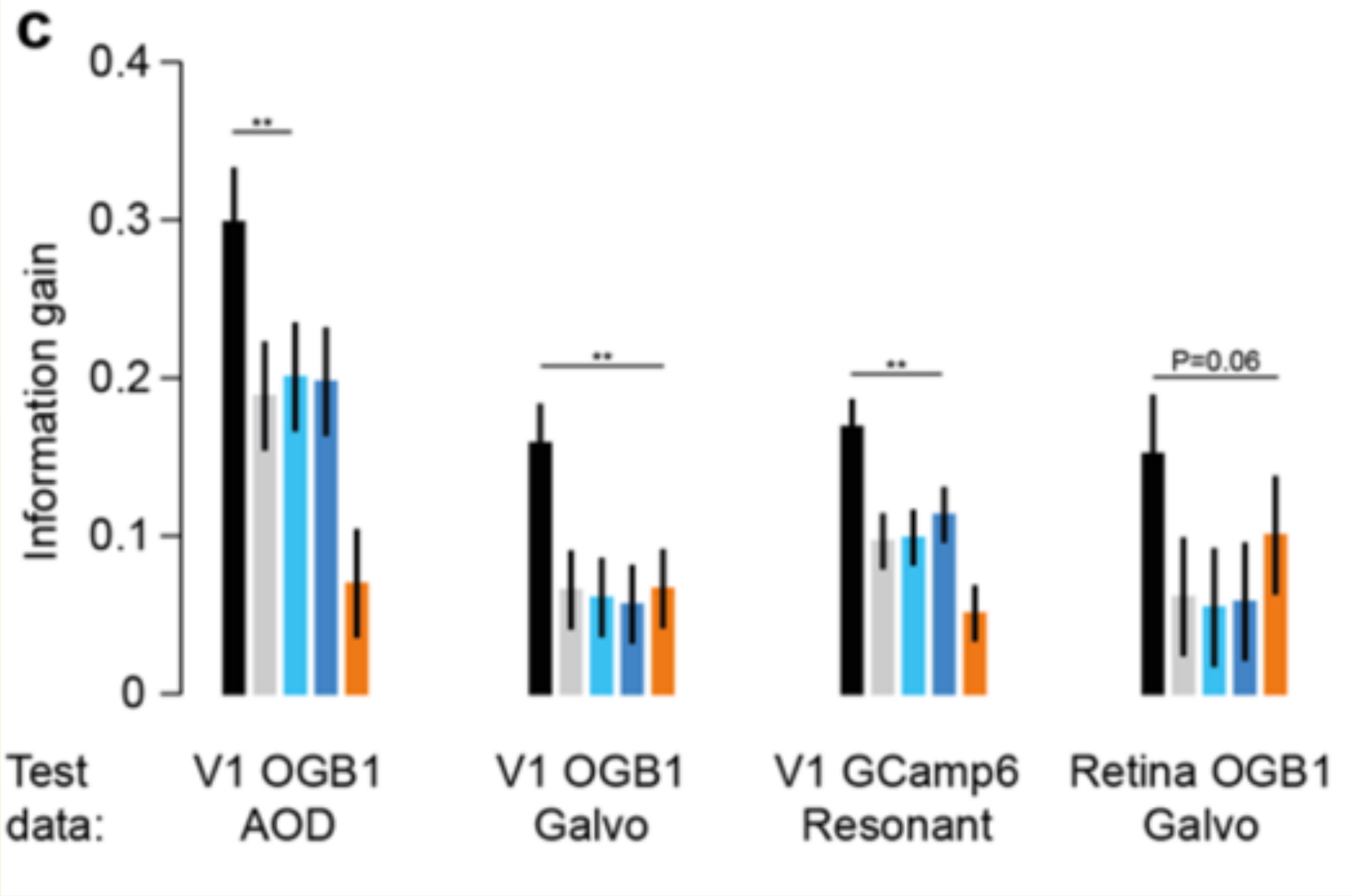
Philipp Berens



Generalization across experiments

Lucas Theis

Philipp Berens



Literature and competition

Lucas Theis

Philipp Berens

Theis et al.: "*Benchmarking spike rate inference in population calcium imaging.*" Neuron, **90(3):** 471-482, 2016.

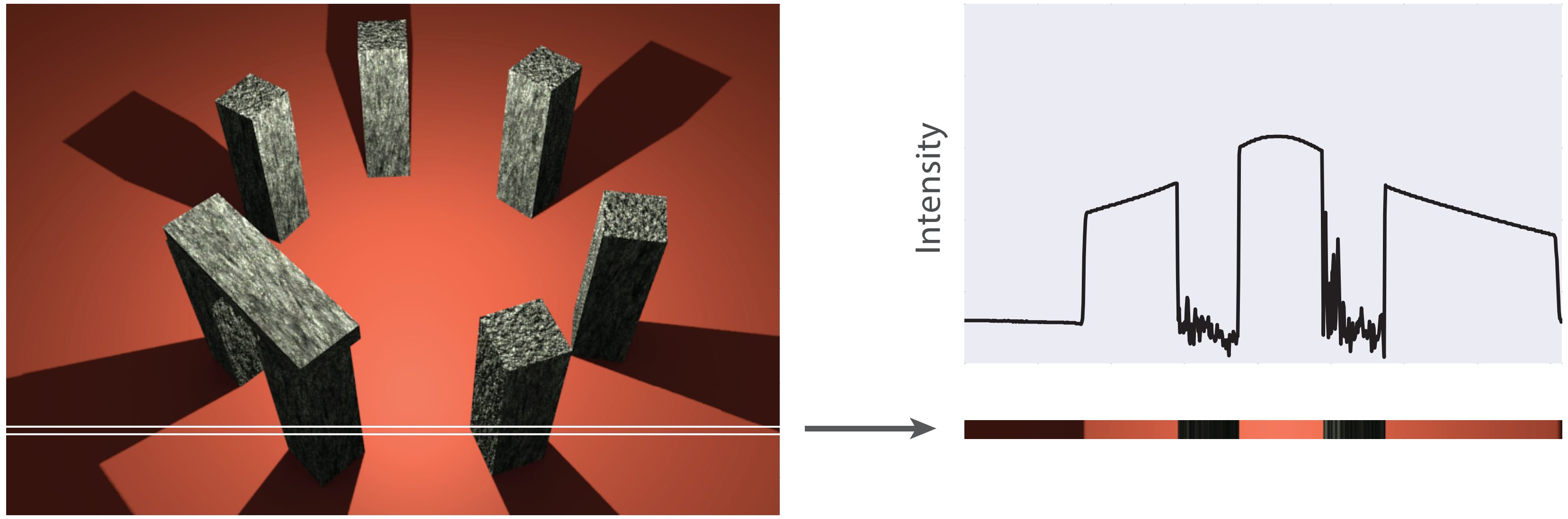
Software: https://github.com/lucastheis/c2s

unsupervised representation learning/
natural image statistics

sensory data

pattern recognition

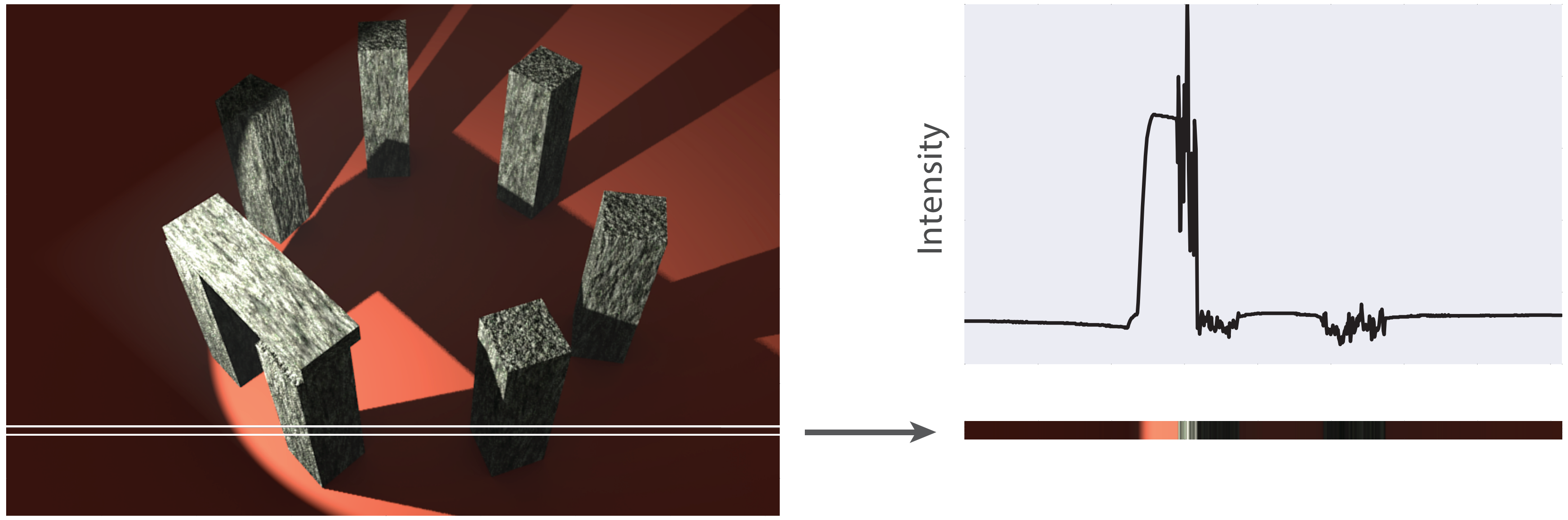
behaviorally relevant variables



sensory data

pattern recognition

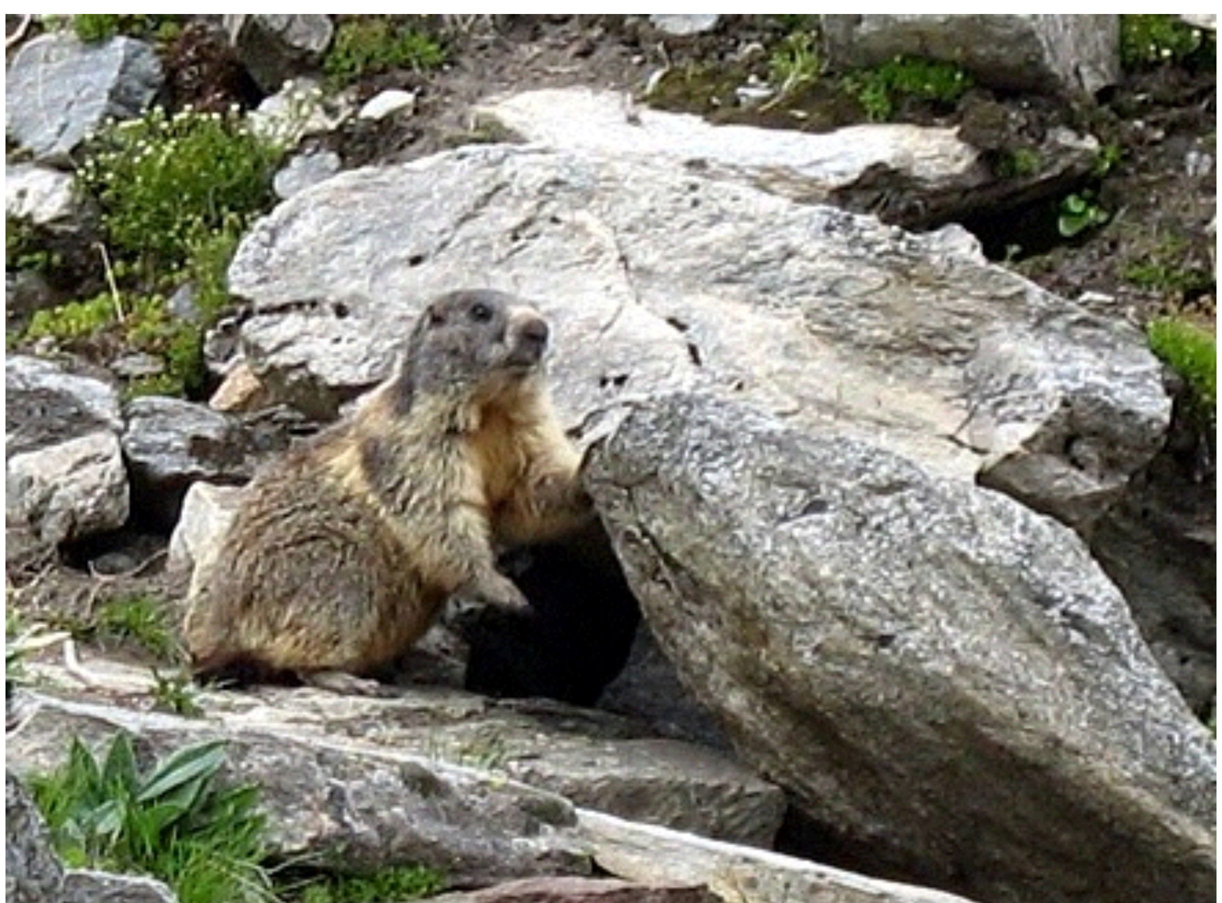
behaviorally relevant variables

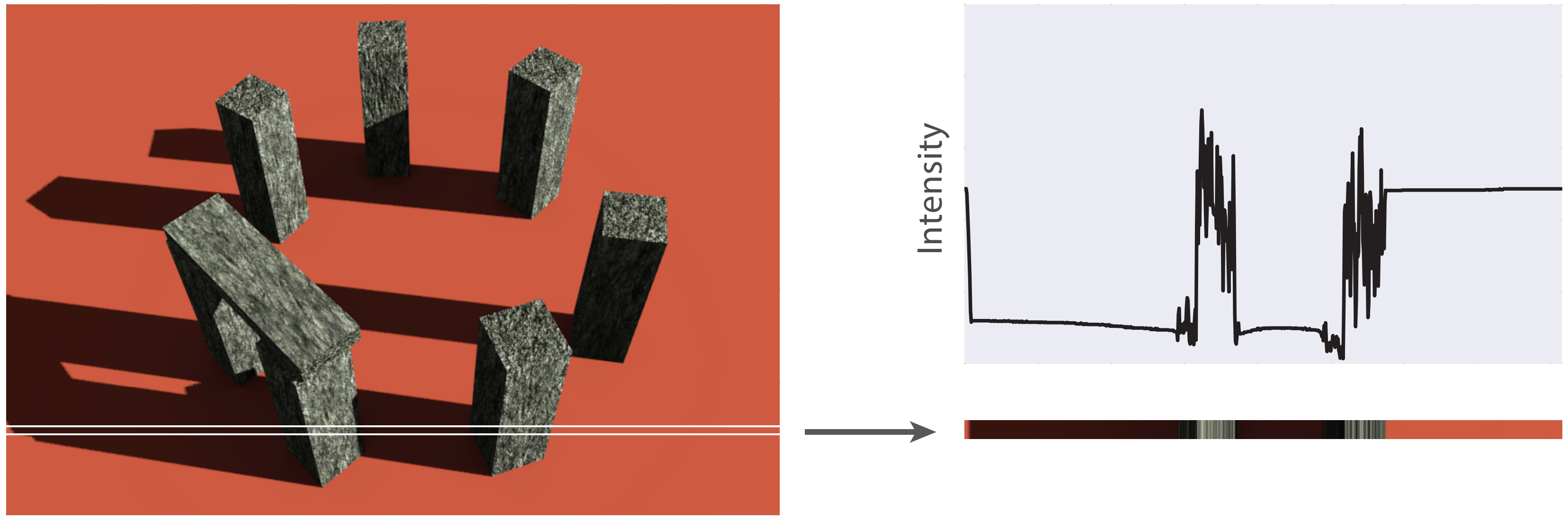


sensory data

pattern recognition

behaviorally relevant variables

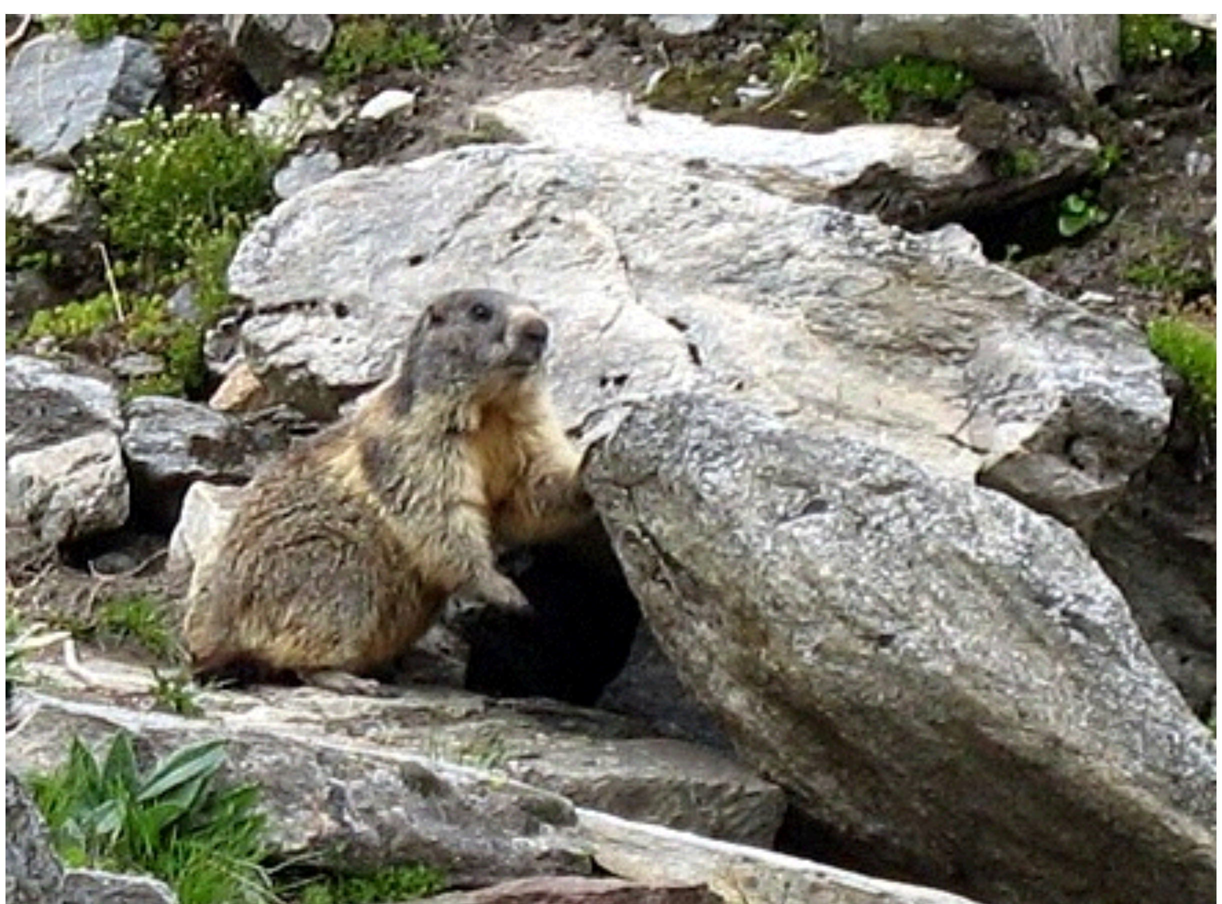


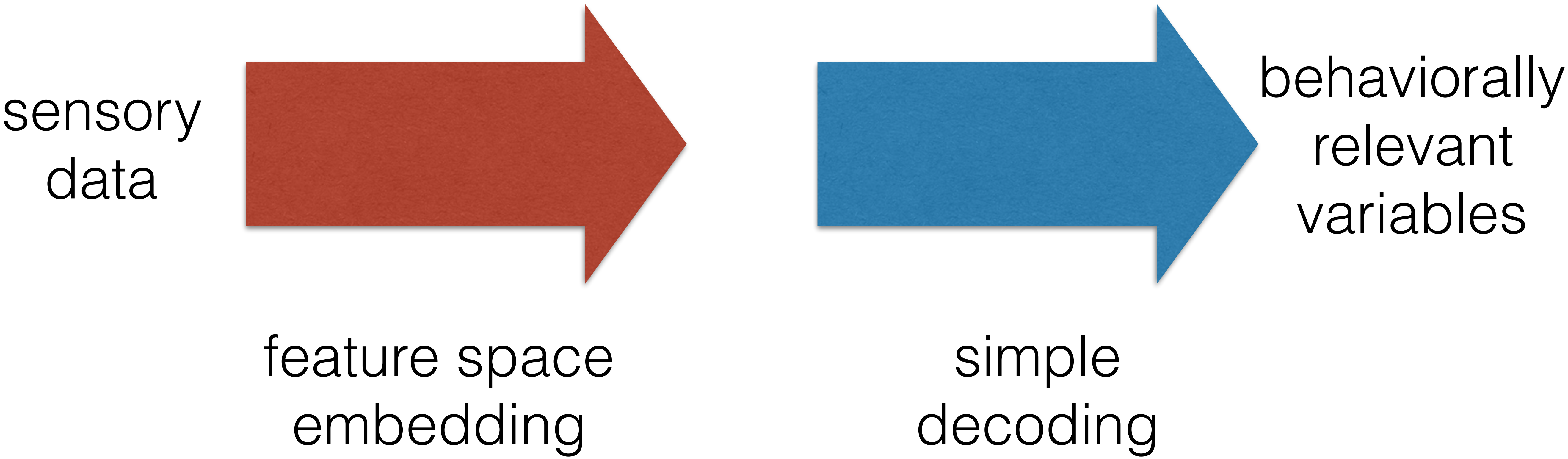


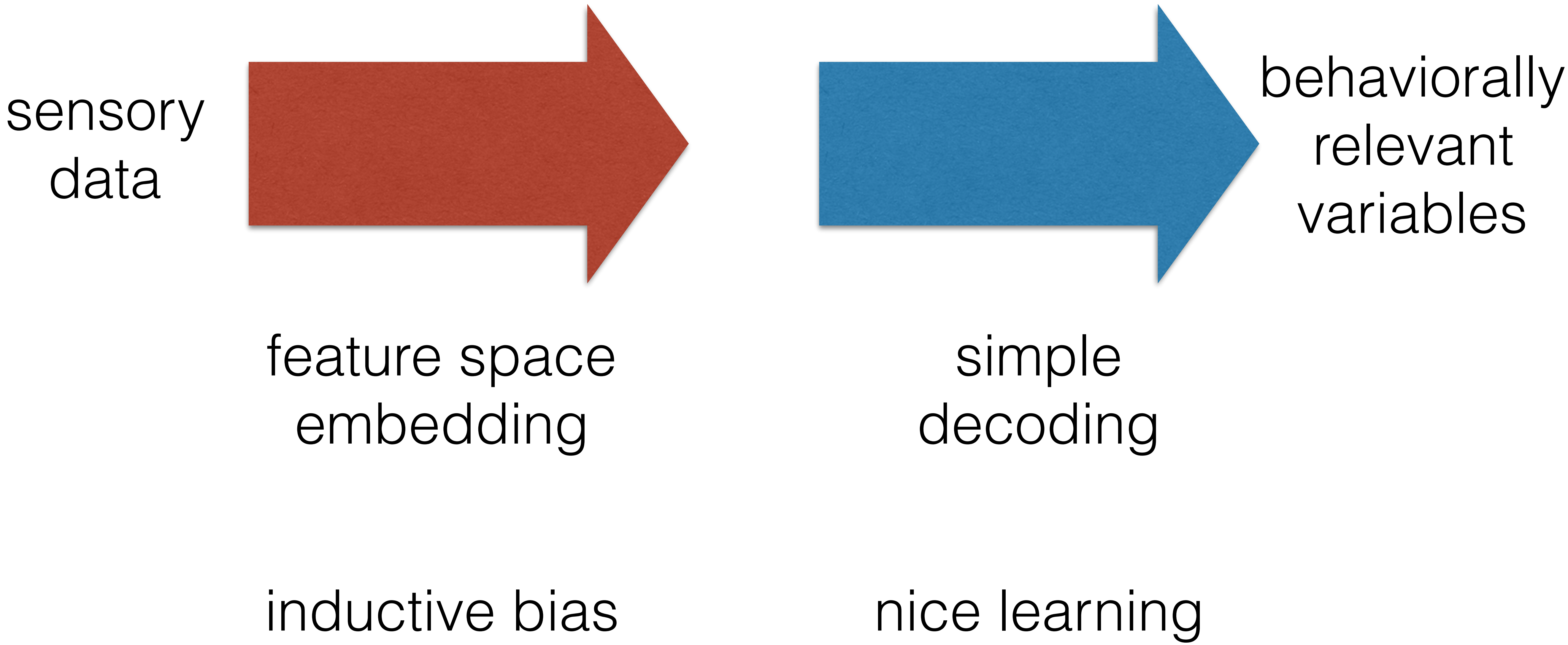
sensory data

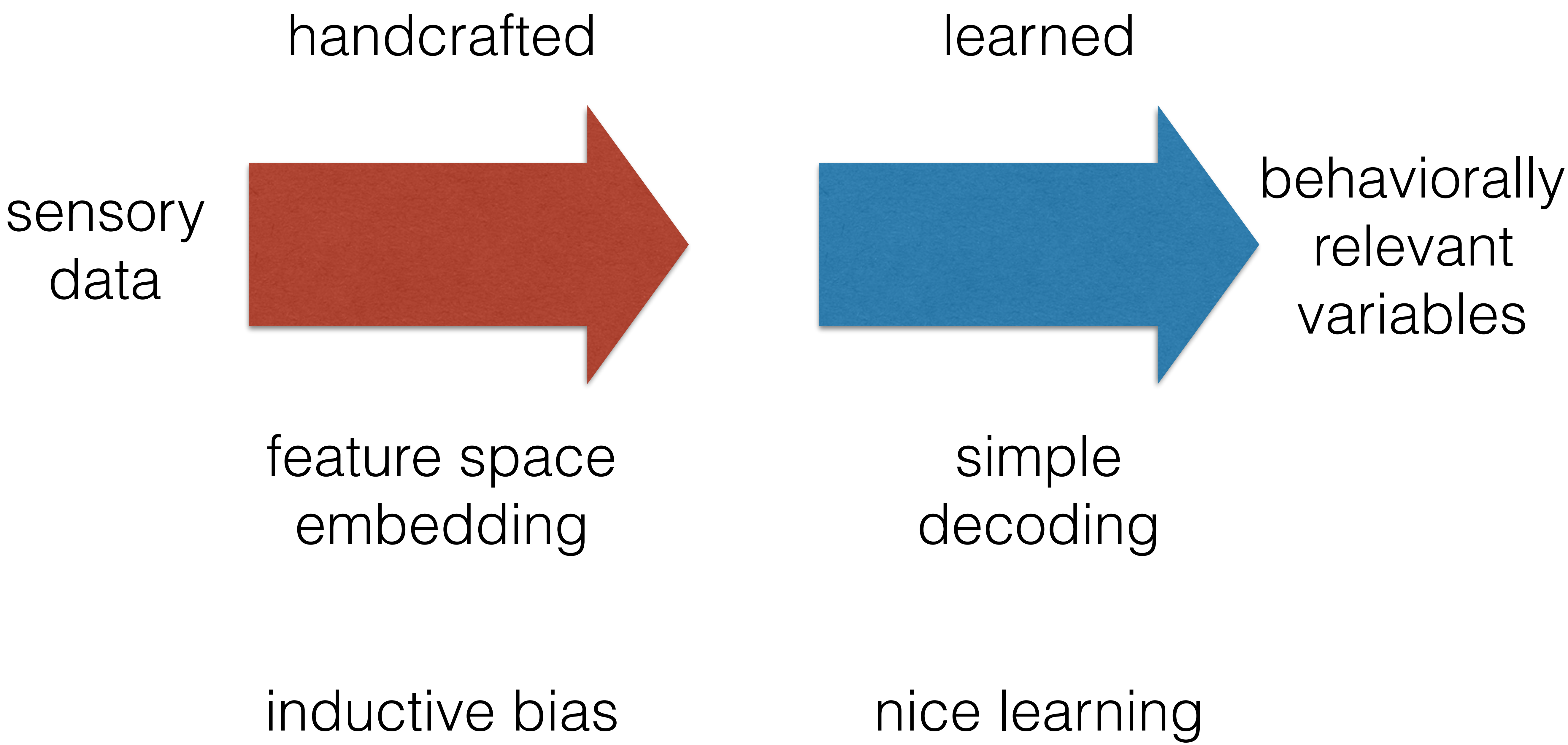
pattern recognition

behaviorally relevant variables

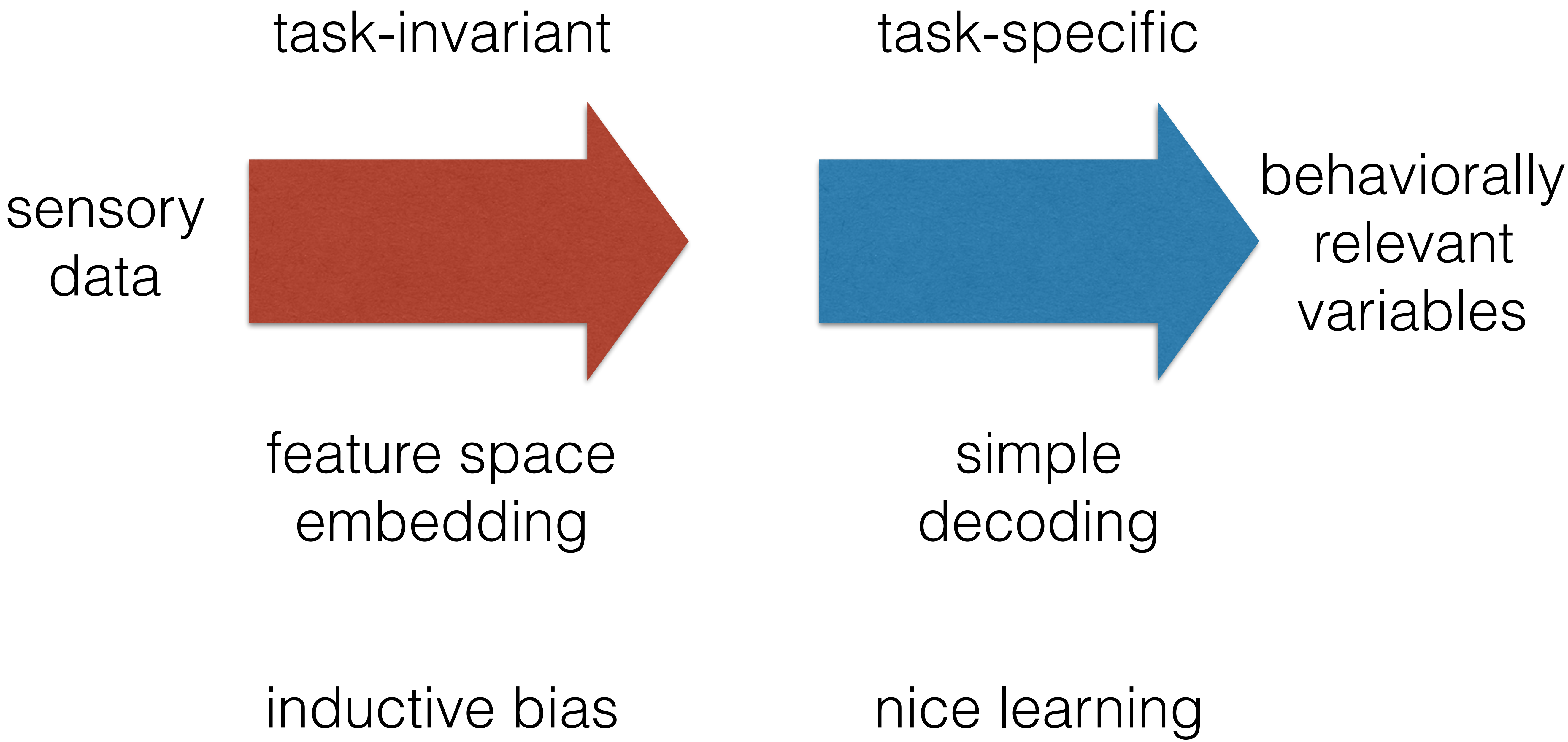


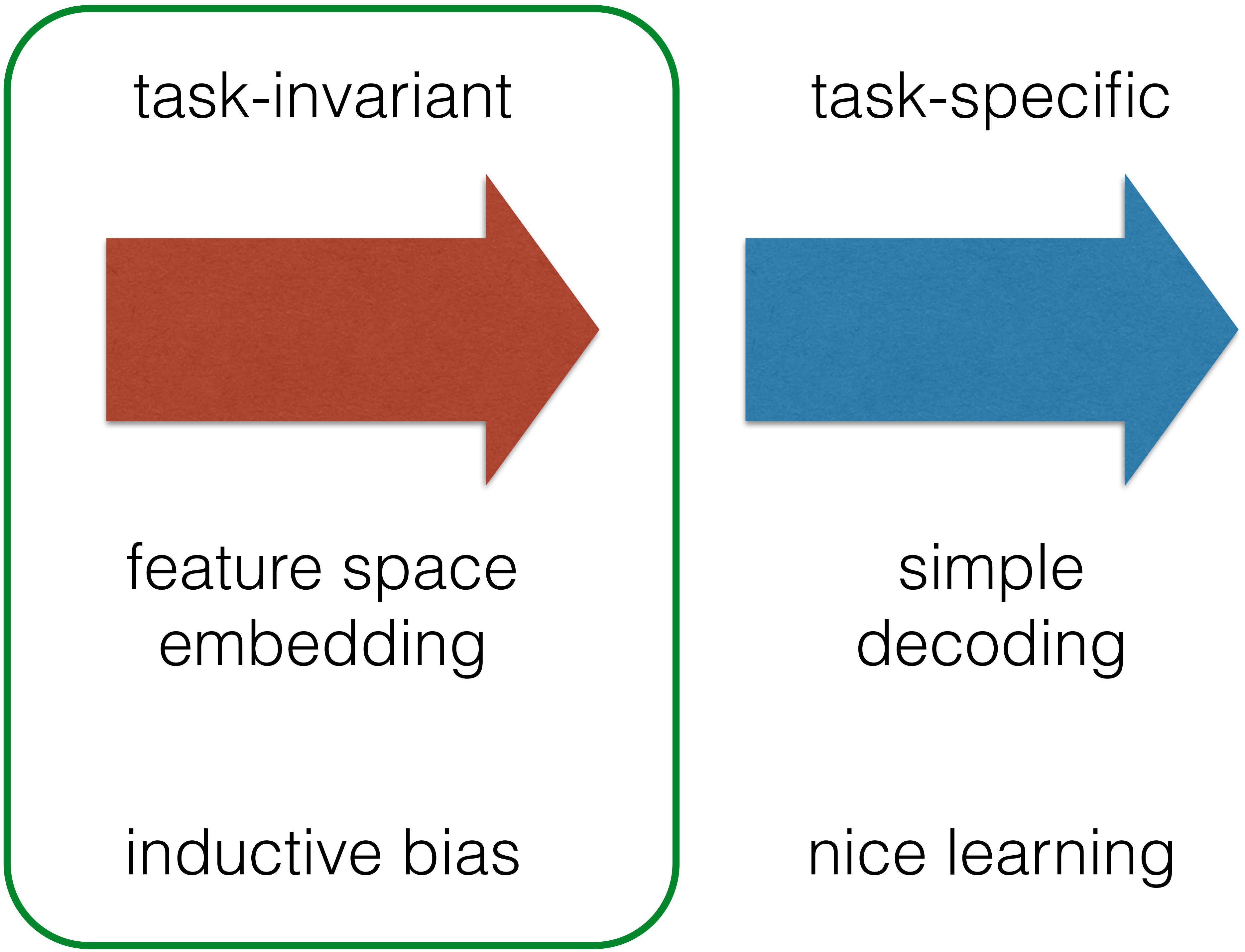

sensory data
feature space embedding
simple decoding
behaviorally relevant variables


sensory data
behaviorally relevant variables
feature space embedding
simple decoding
inductive bias
nice learning



handcrafted
learned
sensory data
behaviorally relevant variables
feature space embedding
simple decoding
inductive bias
nice learning


task-invariant
task-specific
sensory data
behaviorally relevant variables
feature space embedding
simple decoding
inductive bias
nice learning


task-invariant
task-specific
feature space embedding
simple decoding
inductive bias
nice learning

What are good features for vision?

task-invariant

feature space
embedding

inductive bias

task-specific

simple
decoding

nice learning

What are good features for vision?

task-invariant

feature space embedding

inductive bias

1. Neuroscience angle: Early Vision/Sensory Coding

can we learn this?

task-invariant

feature space
embedding

inductive bias

REVIEW

Unsupervised Learning

H.B. Barlow
Kenneth Craik Laboratory, Physiological Laboratory, Downing Street, Cambridge, CB2 3EG, England

What use can the brain make of the massive flow of sensory information that occurs without any associated rewards or punishments? This

THE ROLE OF NATURE, NURTURE, AND INTELLIGENCE IN PATTERN RECOGNITION

INTELLIGENT PATTERN RECOGNITION

H.B. BARLOW
Kenneth Craik Laboratory
Downing St, Cambridge CB2 3EG, England

The Knowledge Used in Vision and Where It Comes from
Author(s): Horace B. Barlow
Source: *Philosophical Transactions: Biological Sciences*, Vol. 352, No. 1358, Knowledge-based Vision in Man and Machine, (Aug. 29, 1997), pp. 1141-1147

generative image representation have much lower entropy than image pixels

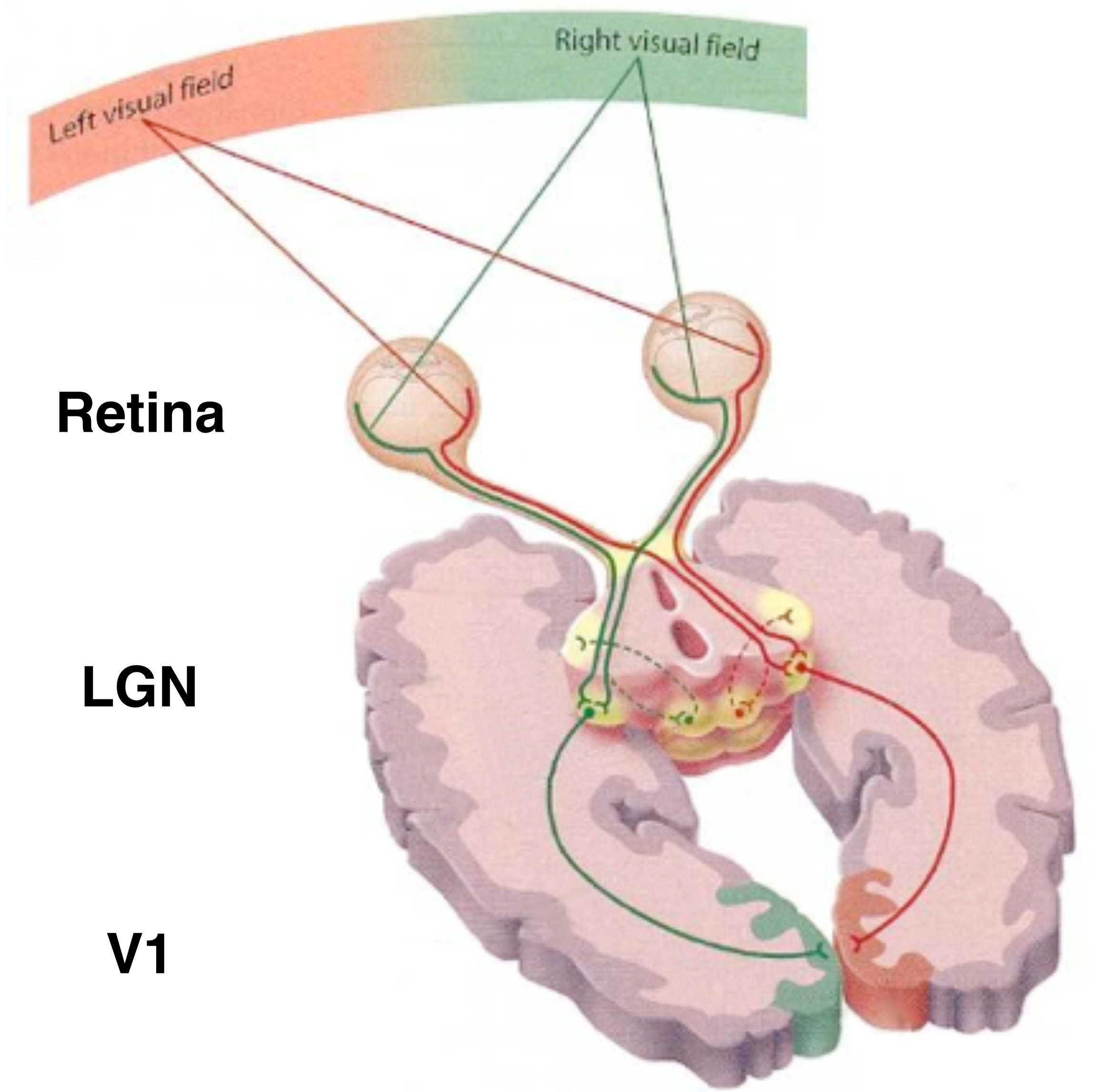
==> transform input into a minimum entropy representation

Redundancy Reduction

Retina

LGN

V1



An attractive feature of [redundancy reduction] is that a code formed in response to redundancies in the input would constitute a distributed memory of this regularities---one that is used automatically and does not require a separate recall mechanism.”

Horace Barlow, *BBS,* 2001

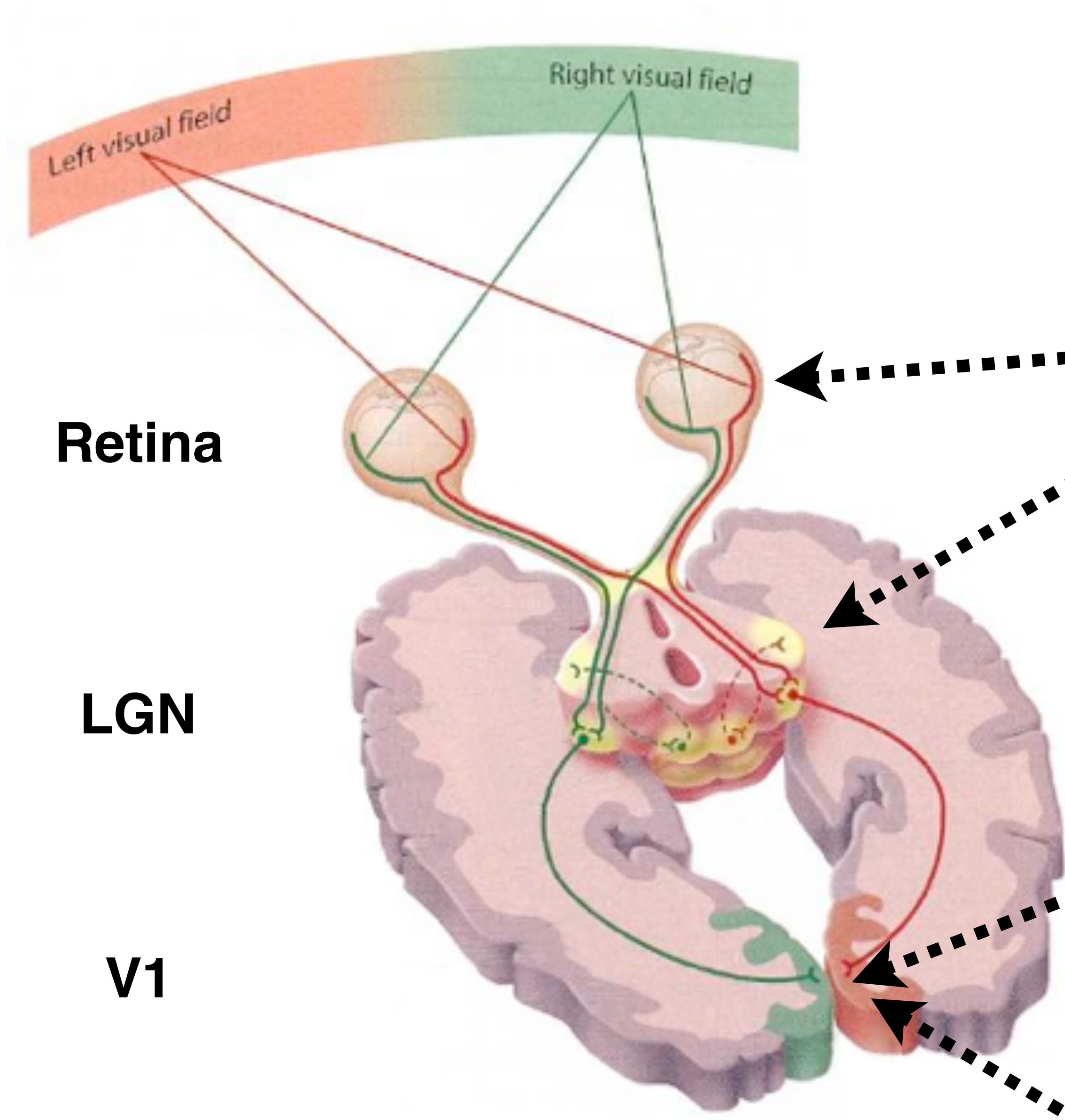
Barlow’s redundancy reduction hypothesis (1961)

Redundancy Reduction

Retina

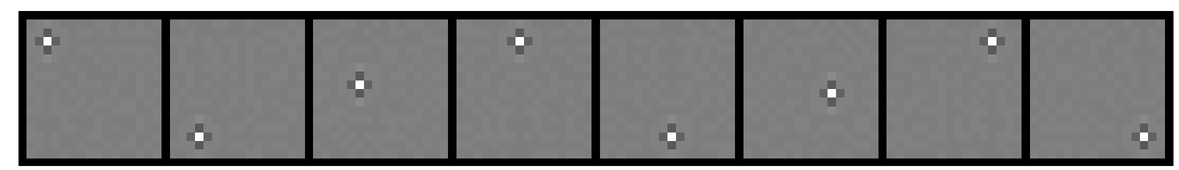
LGN

V1



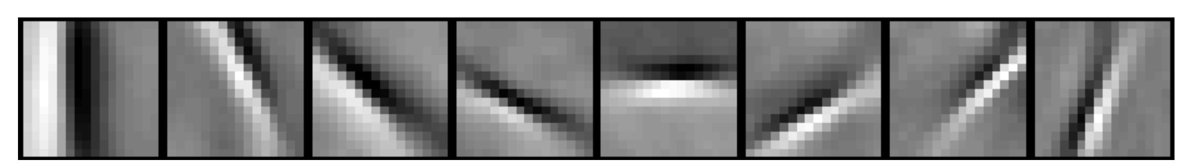
Ganglion cells: Whitening

Atick&Redlich,
Neural Comput., 1990



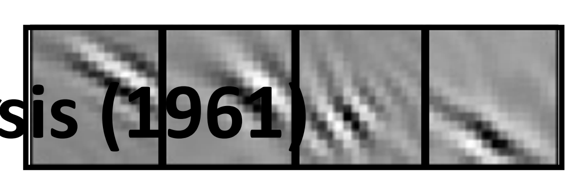
Simple cells:
Independent Component Analysis (ICA)

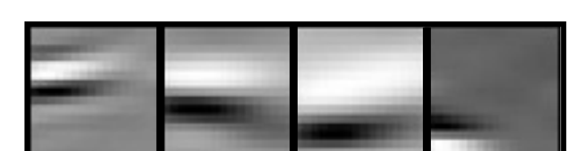
Olshausen&Field, Nature, 1996
Bell&Sejnowski, Vision Res., 1997



Complex cells: Subspace ICA

Hyvarinen&Hoyer, Neural Comput., 2000
Hyvarinen&Köster, Network, 2007



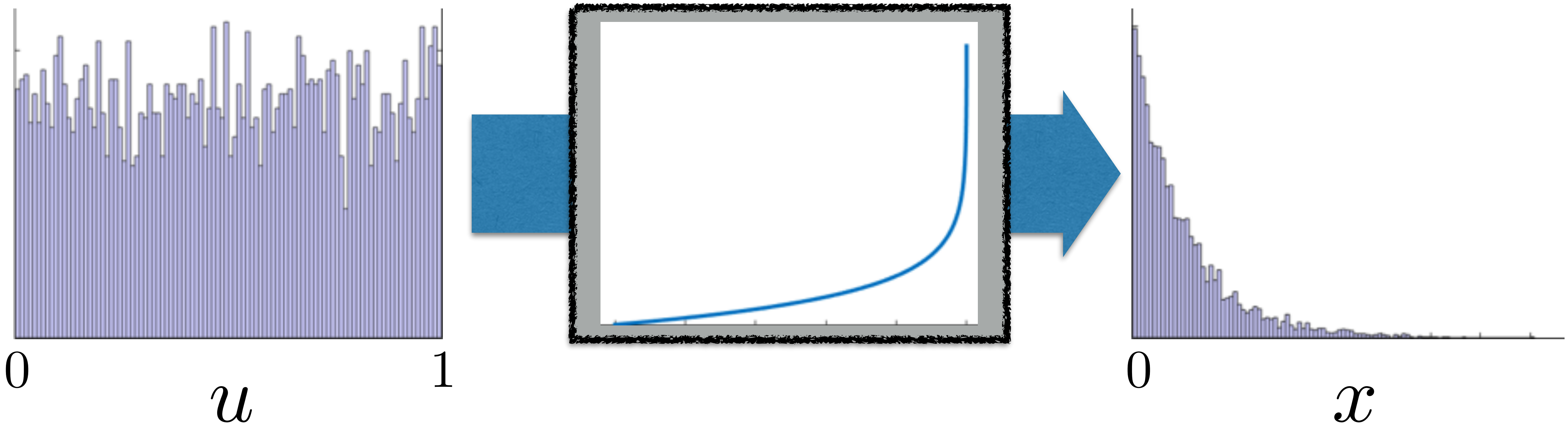


Barlow's redundancy reduction hypothesis (1961)

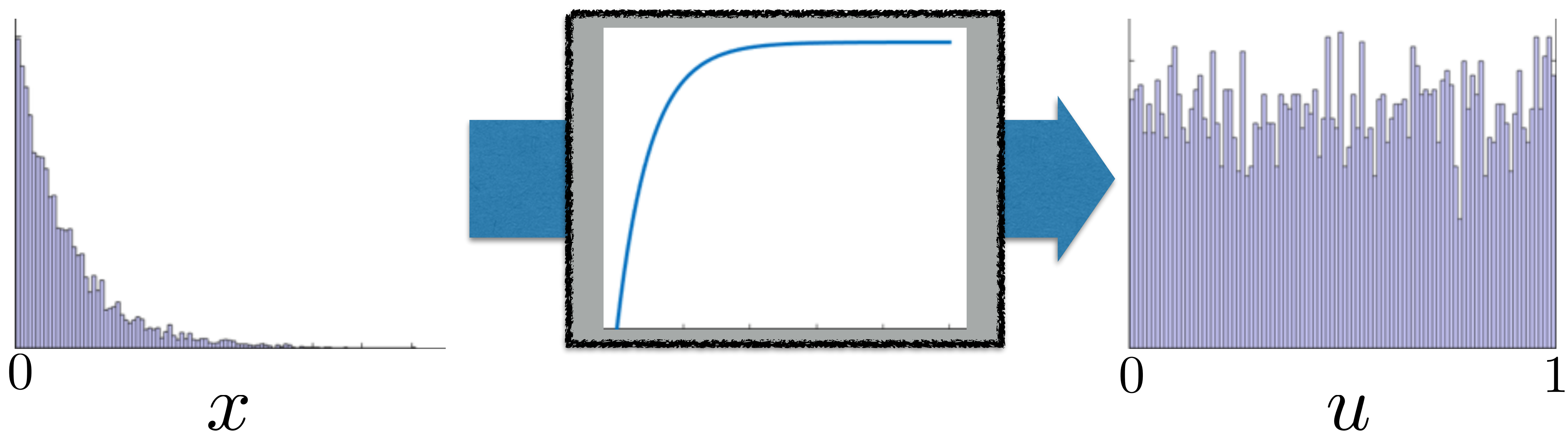
statistics of sensory data

receptive field properties

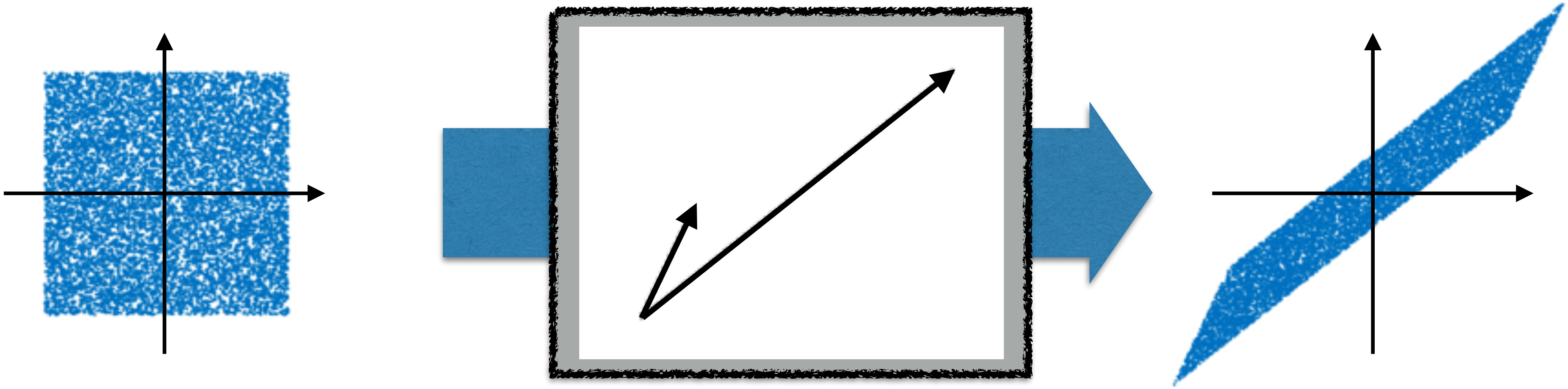
$$x = -\log(1-u)$$



$$u = 1 - \exp(-x)$$

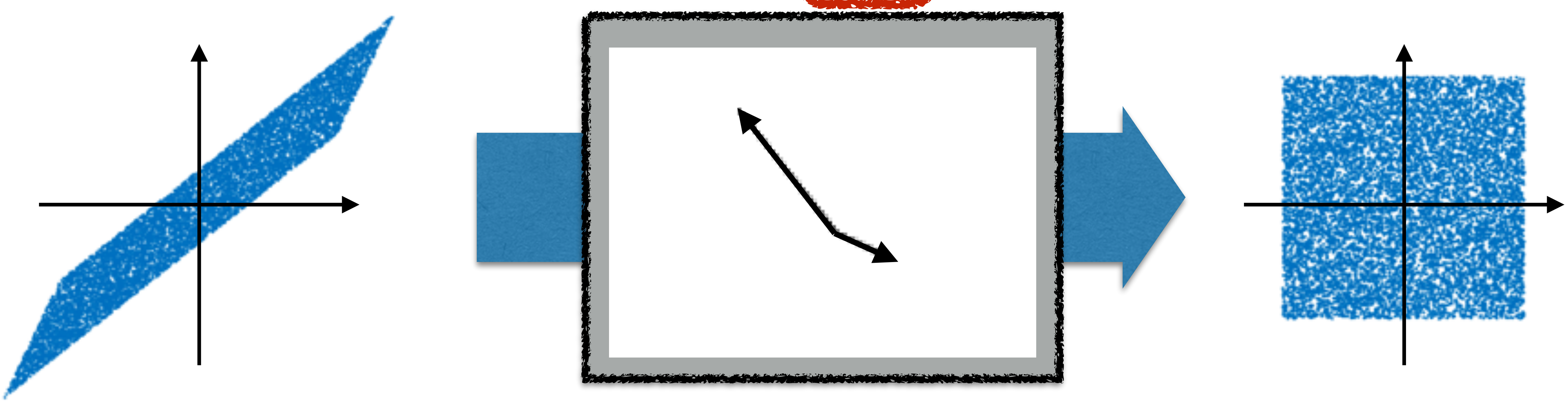


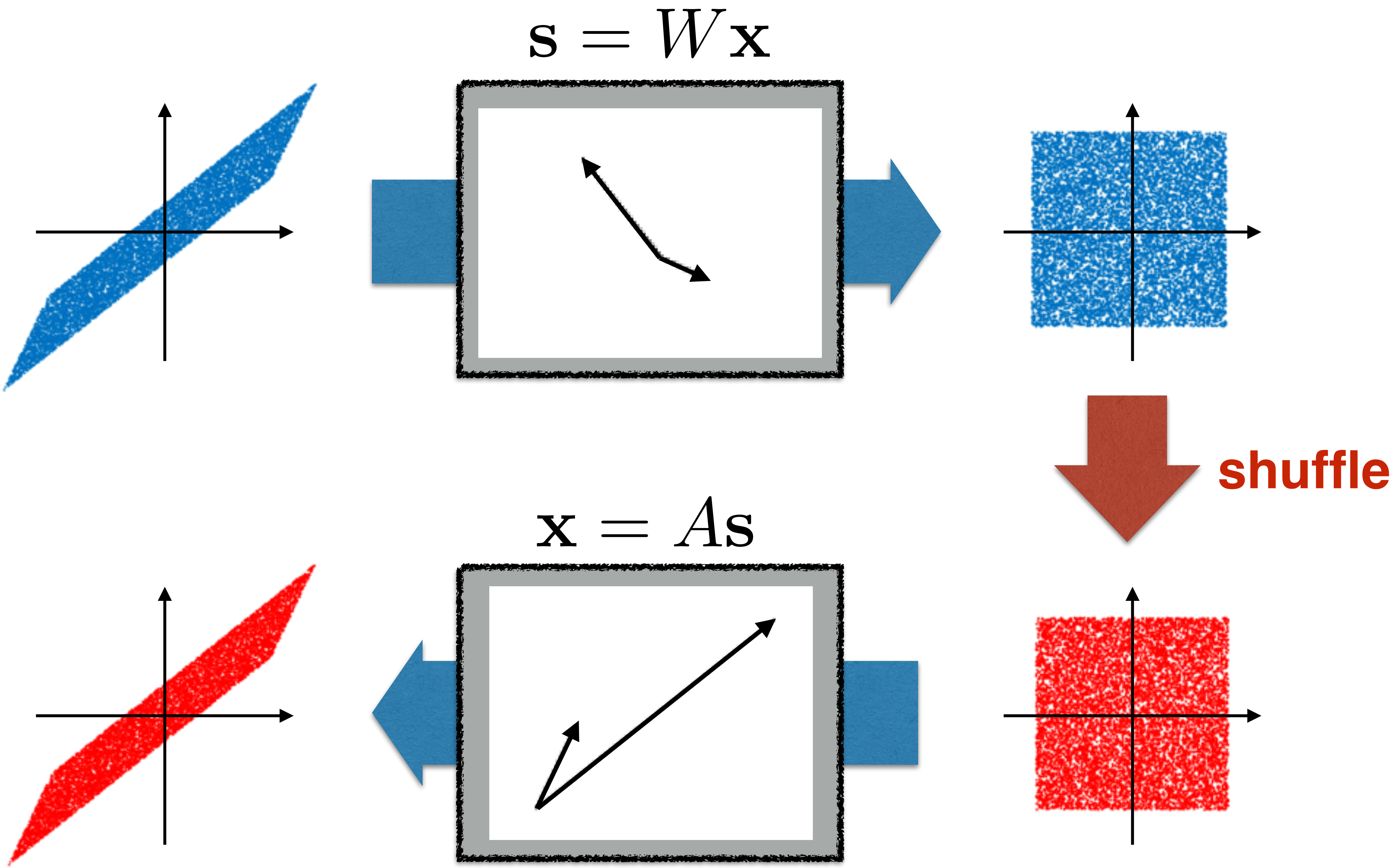
$$\mathbf{x} = A\mathbf{s}$$



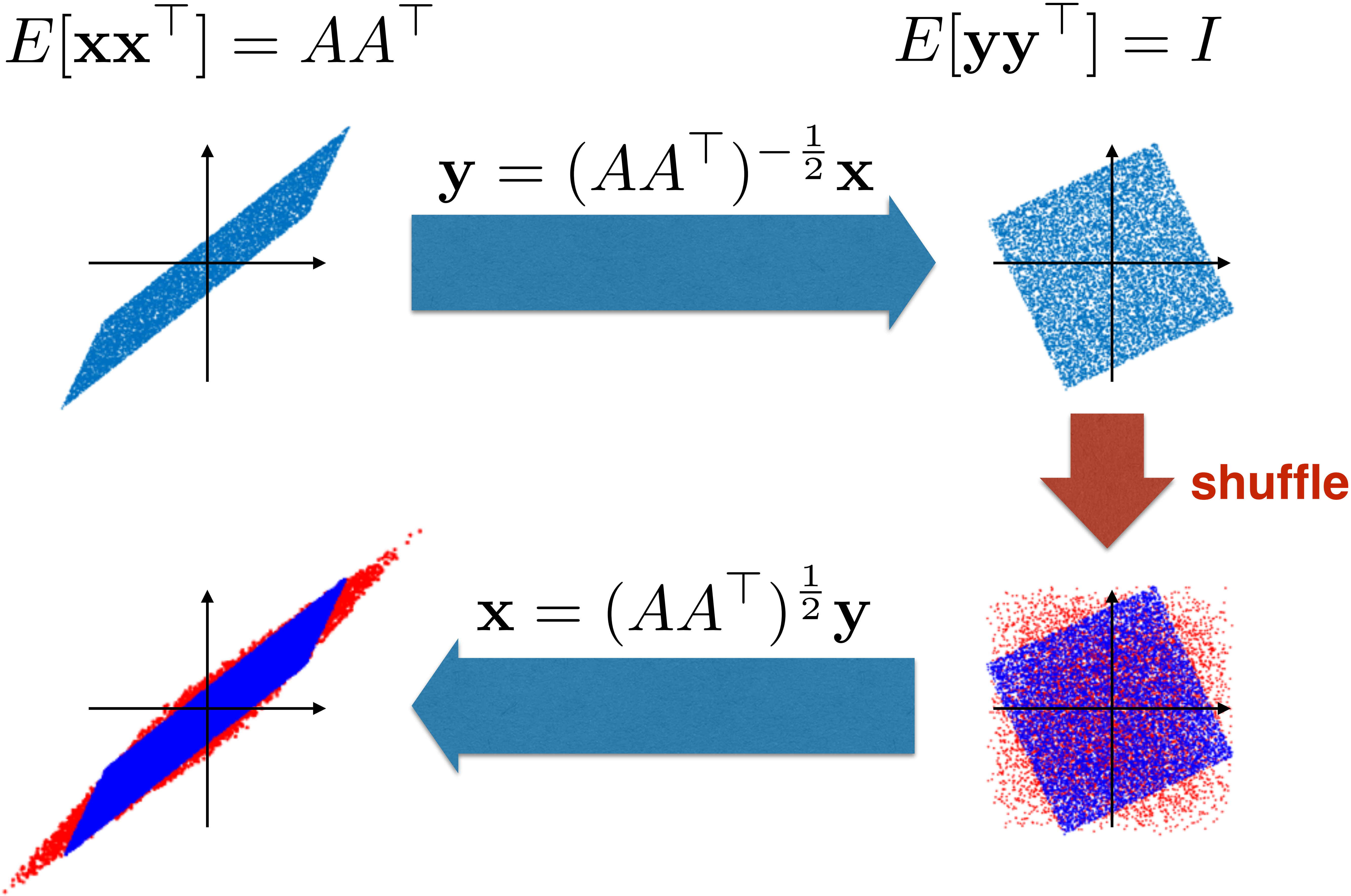
Redundancy reduction
= searching W

$$\mathbf{s} = \boxed{W}\mathbf{x}$$

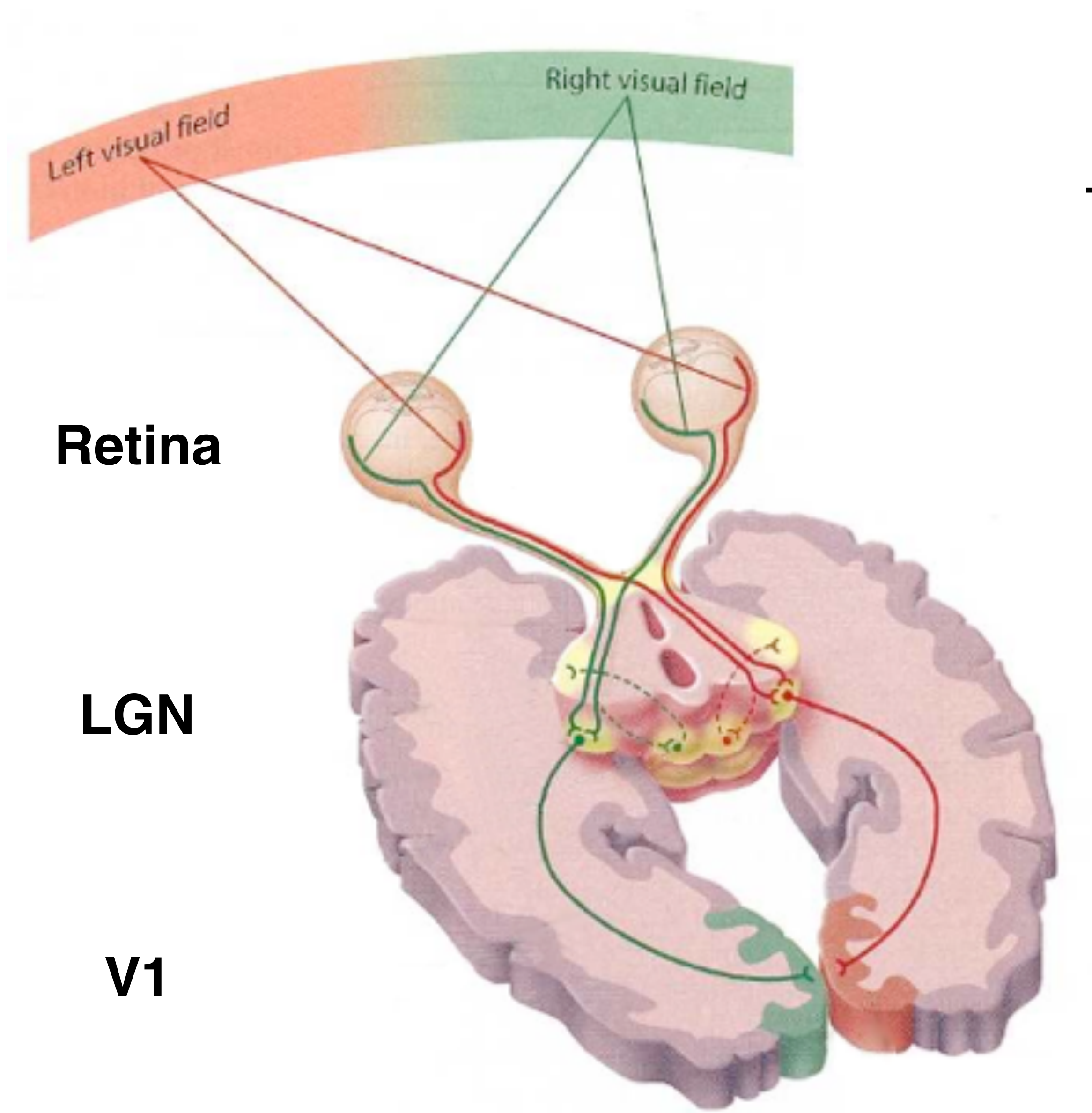


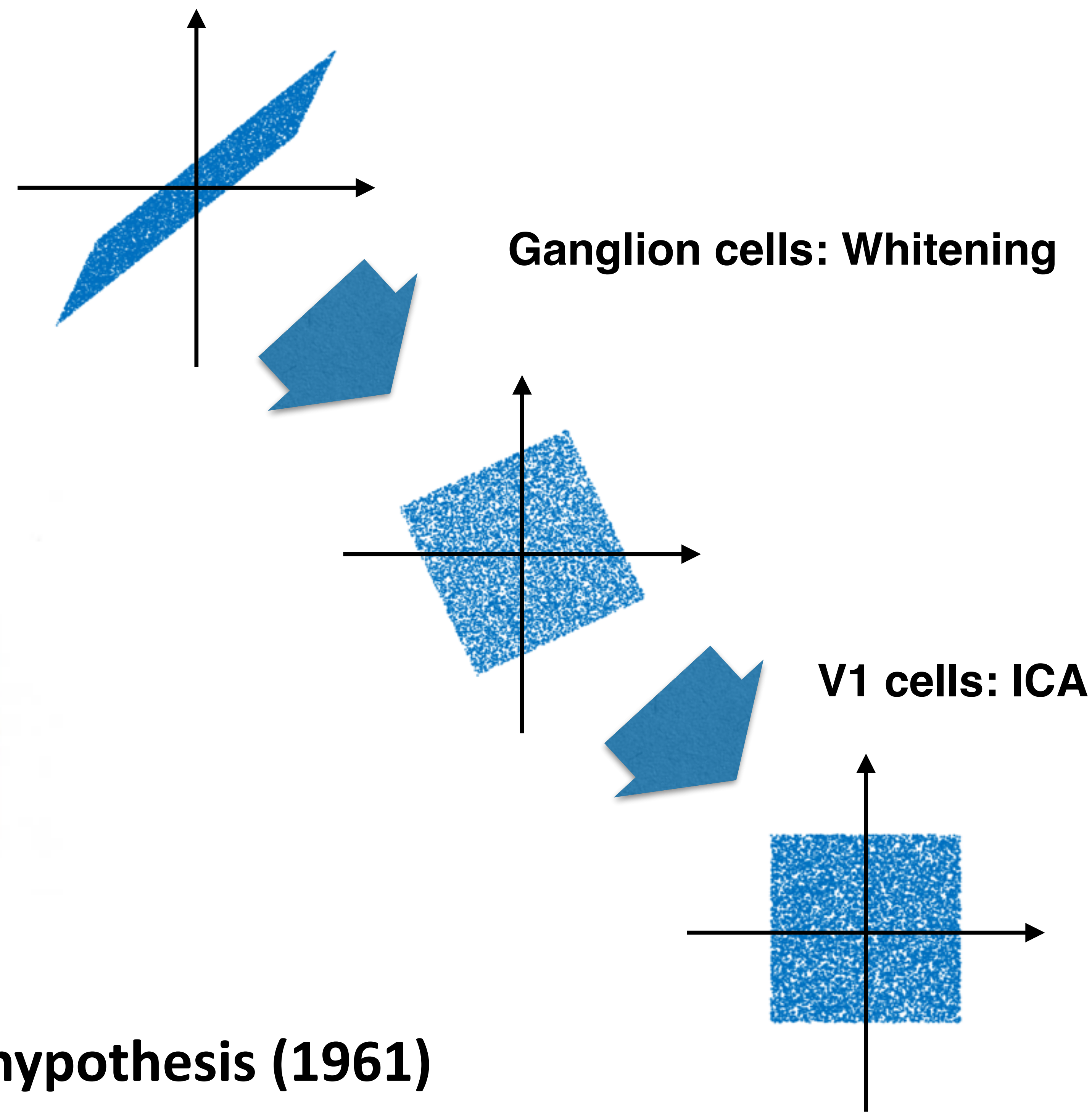


$\mathbf{s} = W\mathbf{x}$
shuffle
$\mathbf{x} = A\mathbf{s}$


$E[\mathbf{x}\mathbf{x}^\top] = AA^\top$
$E[\mathbf{y}\mathbf{y}^\top] = I$
$\mathbf{y} = (AA^\top)^{-\frac{1}{2}}\mathbf{x}$
shuffle
$\mathbf{x} = (AA^\top)^{\frac{1}{2}}\mathbf{y}$

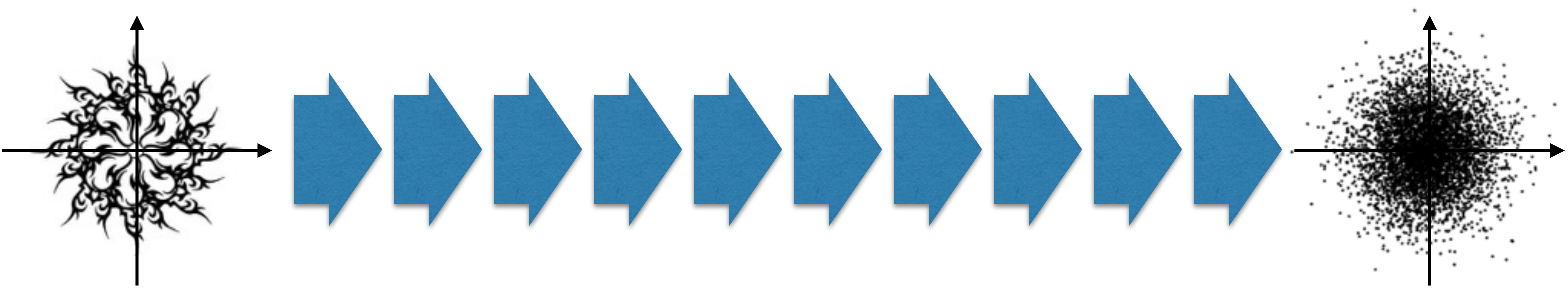
Redundancy Reduction





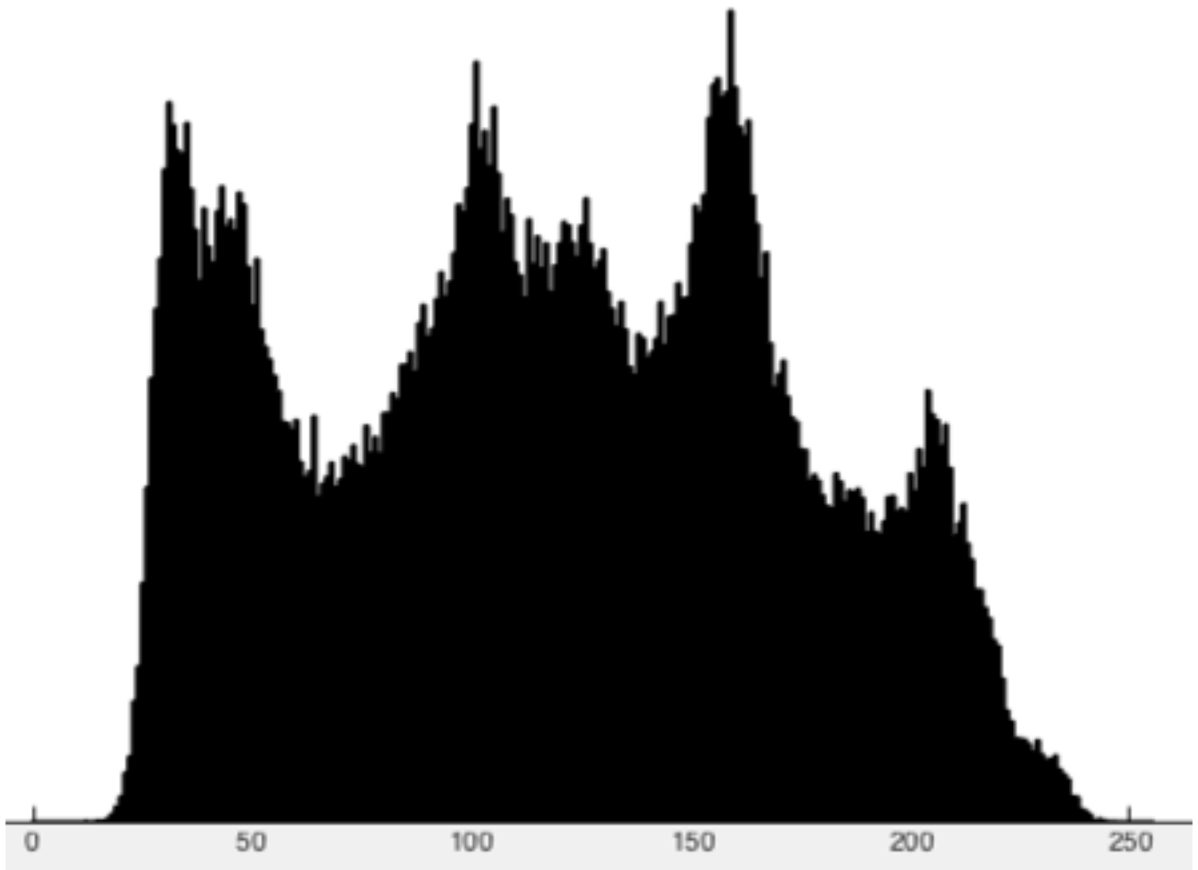
Barlow's redundancy reduction hypothesis (1961)

Deep Redundancy Reduction

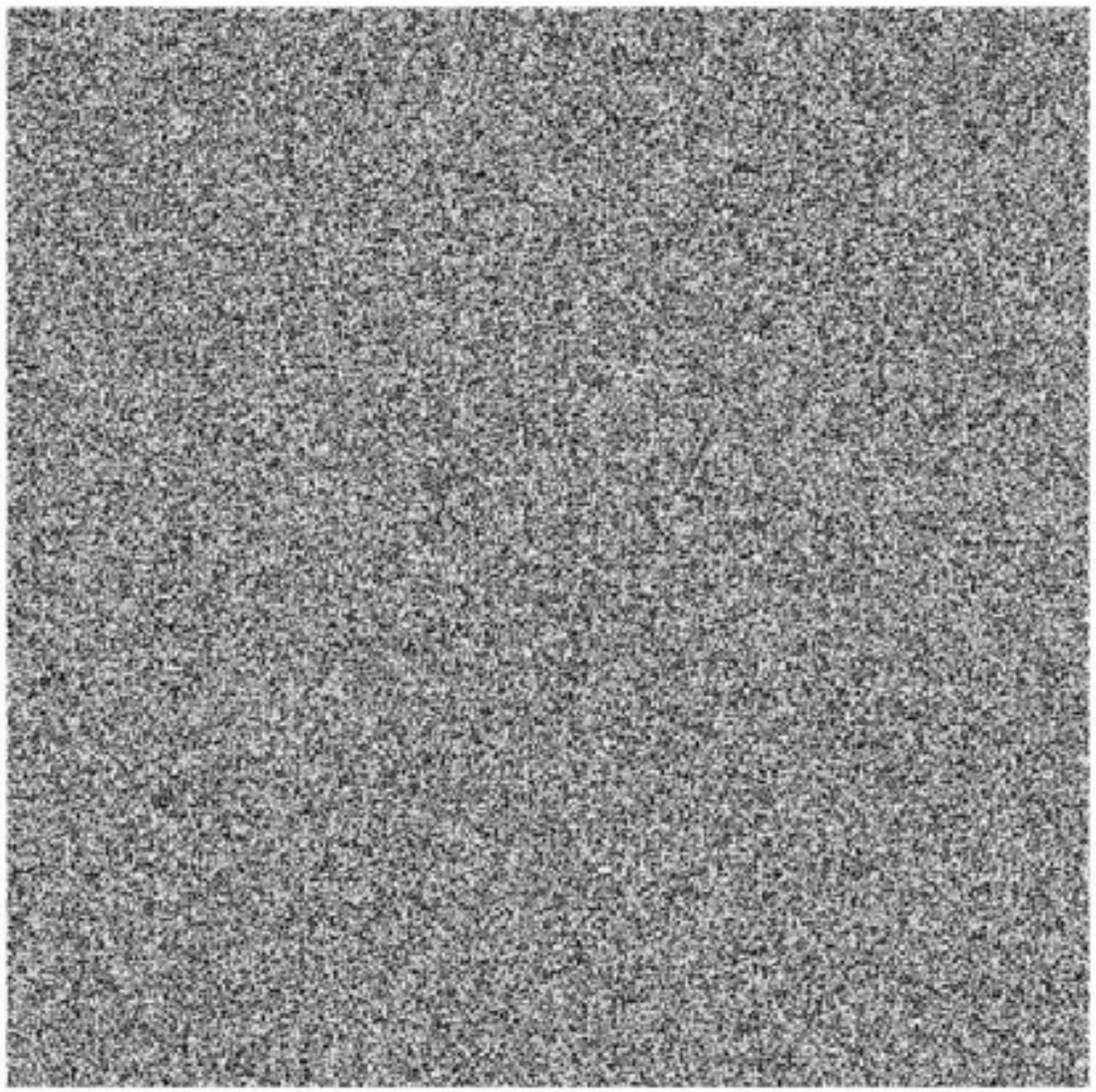


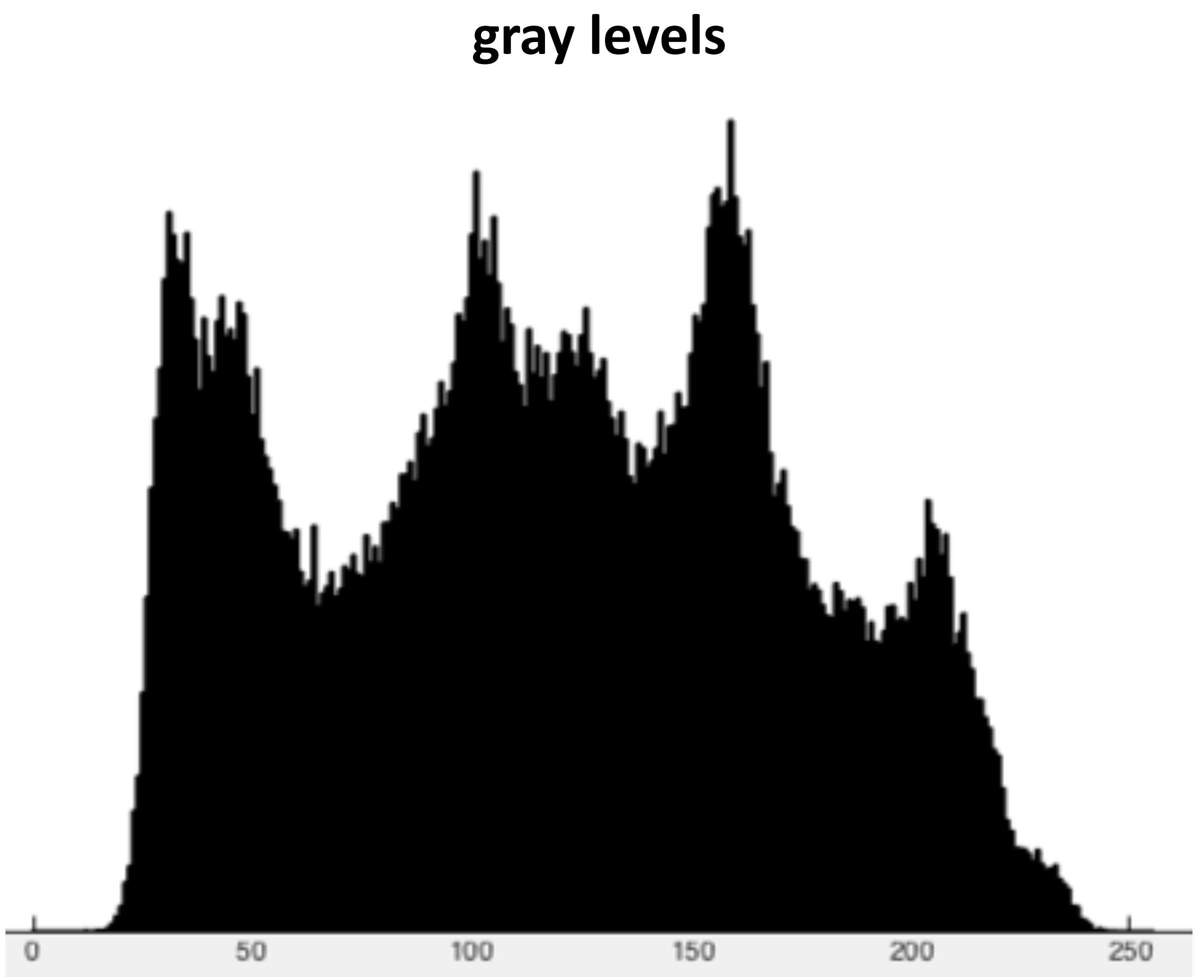
Minimax modeling

gray levels

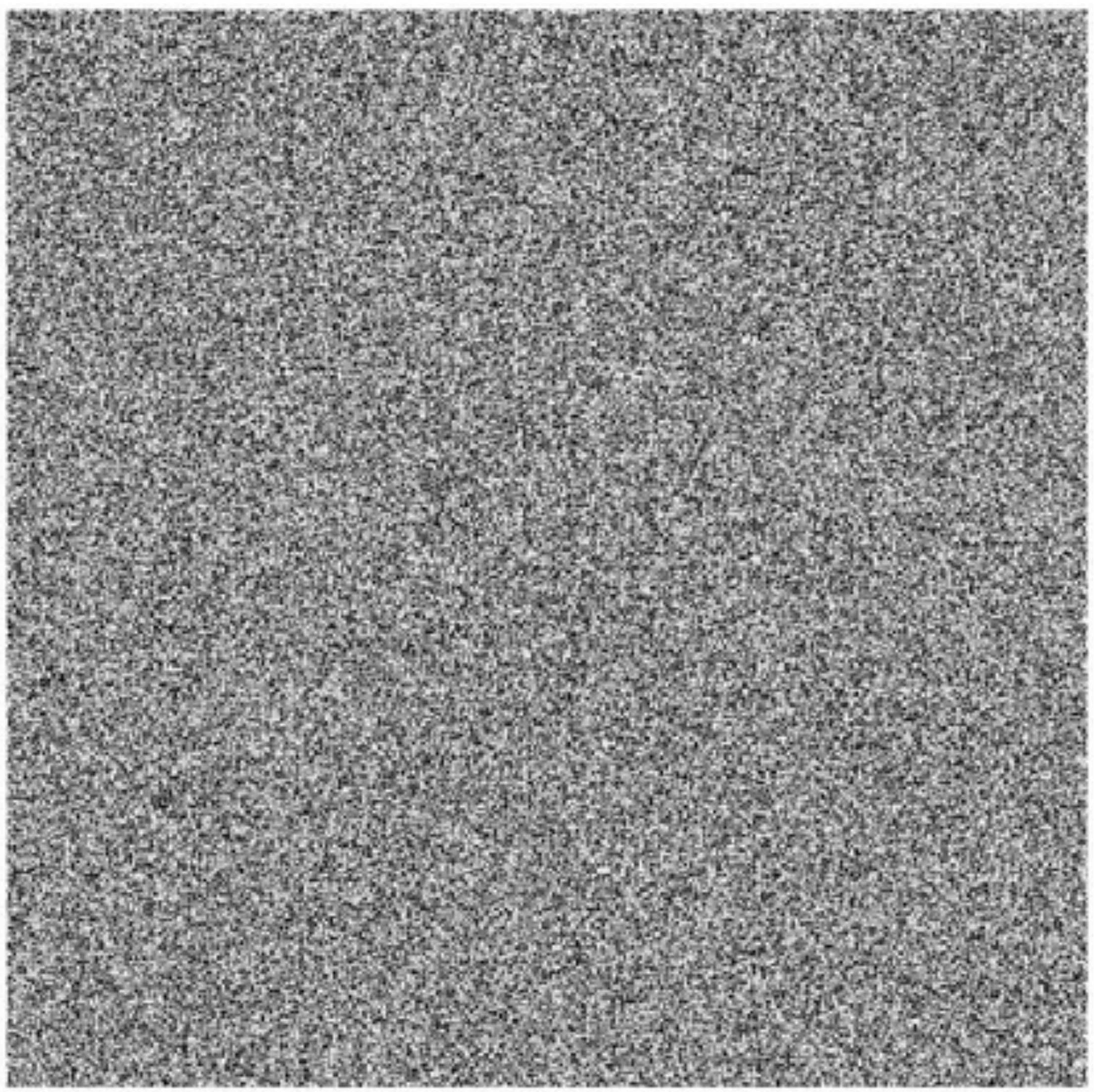


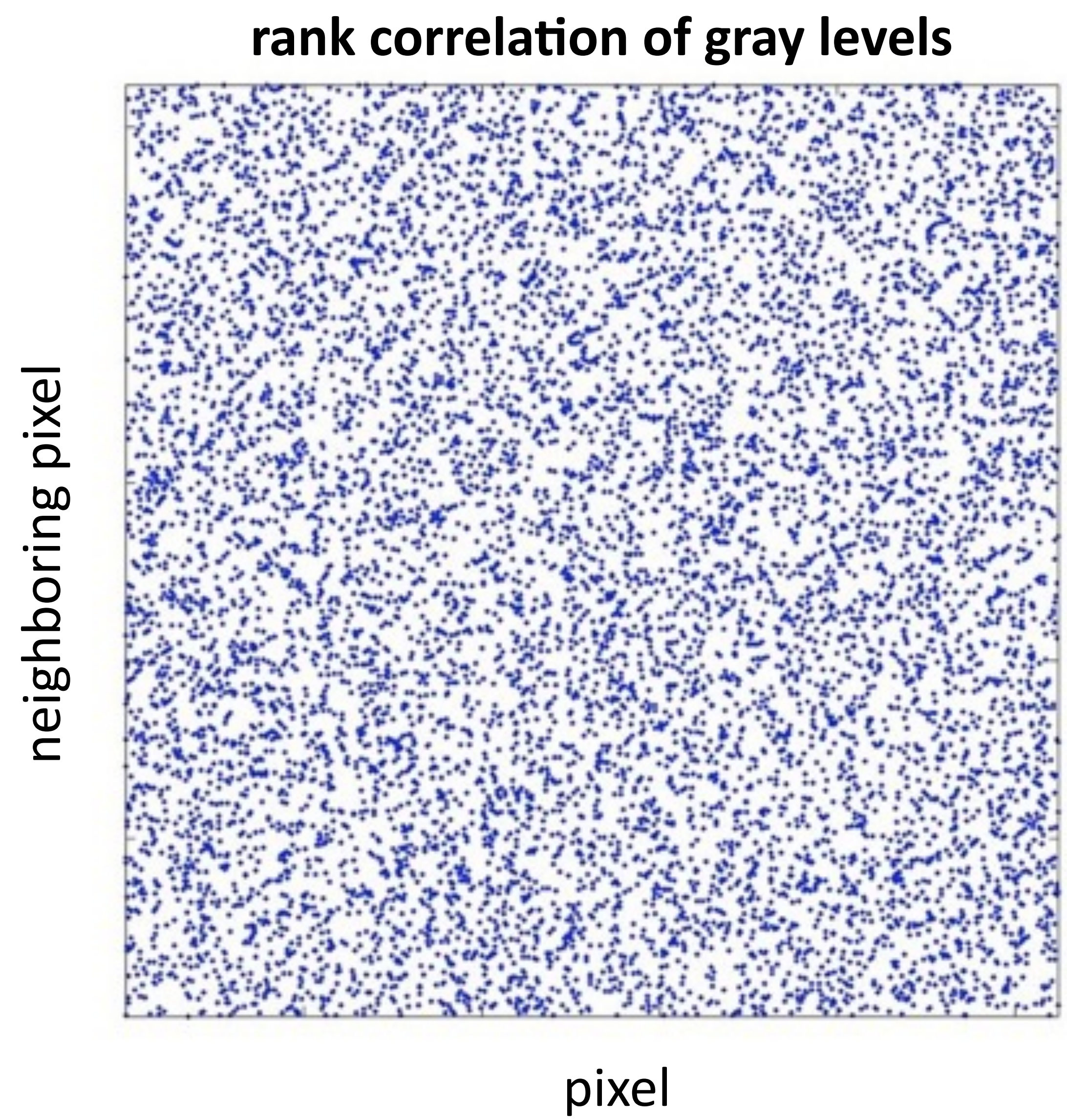
Minimax modeling



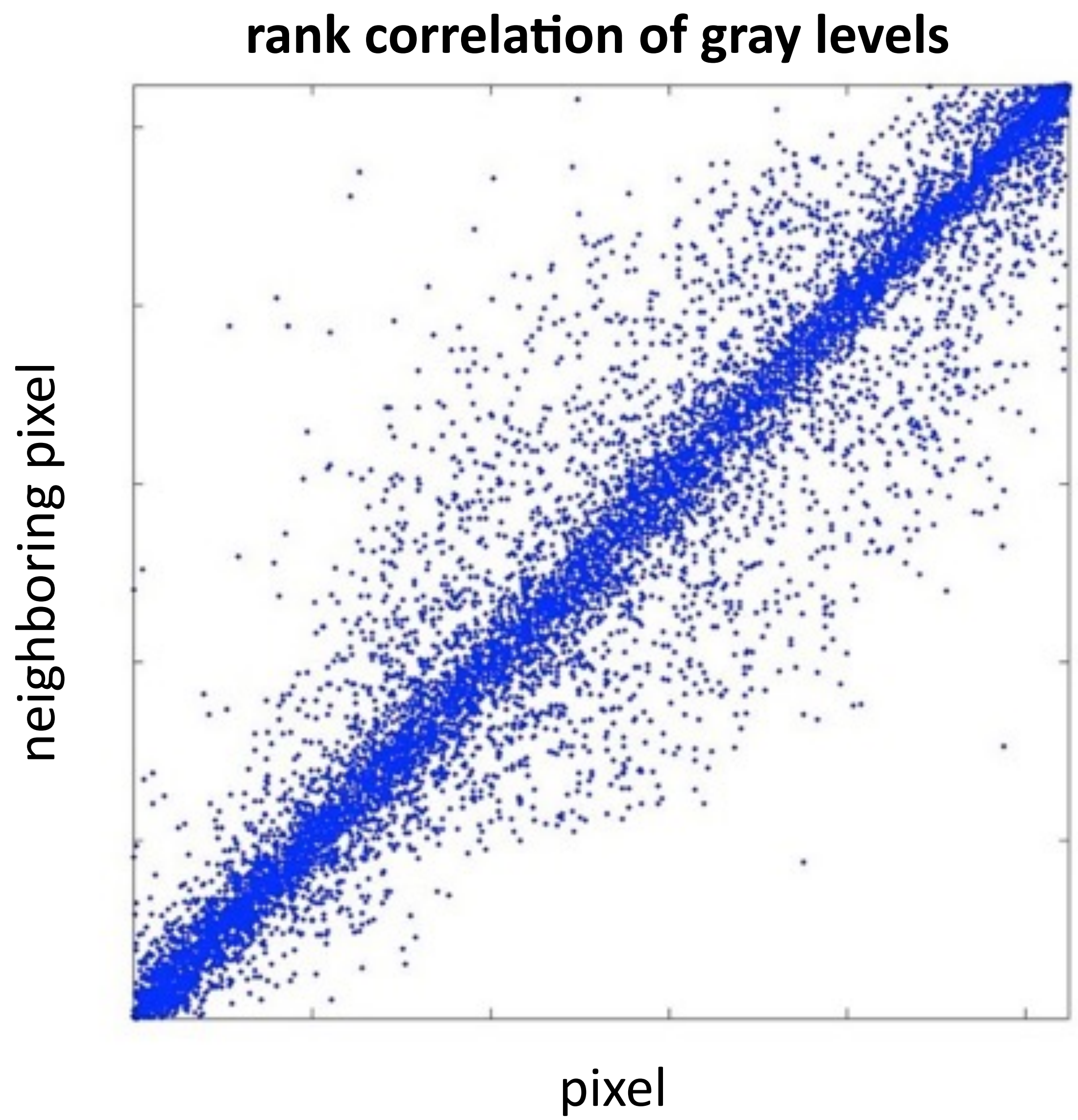


Minimax modeling

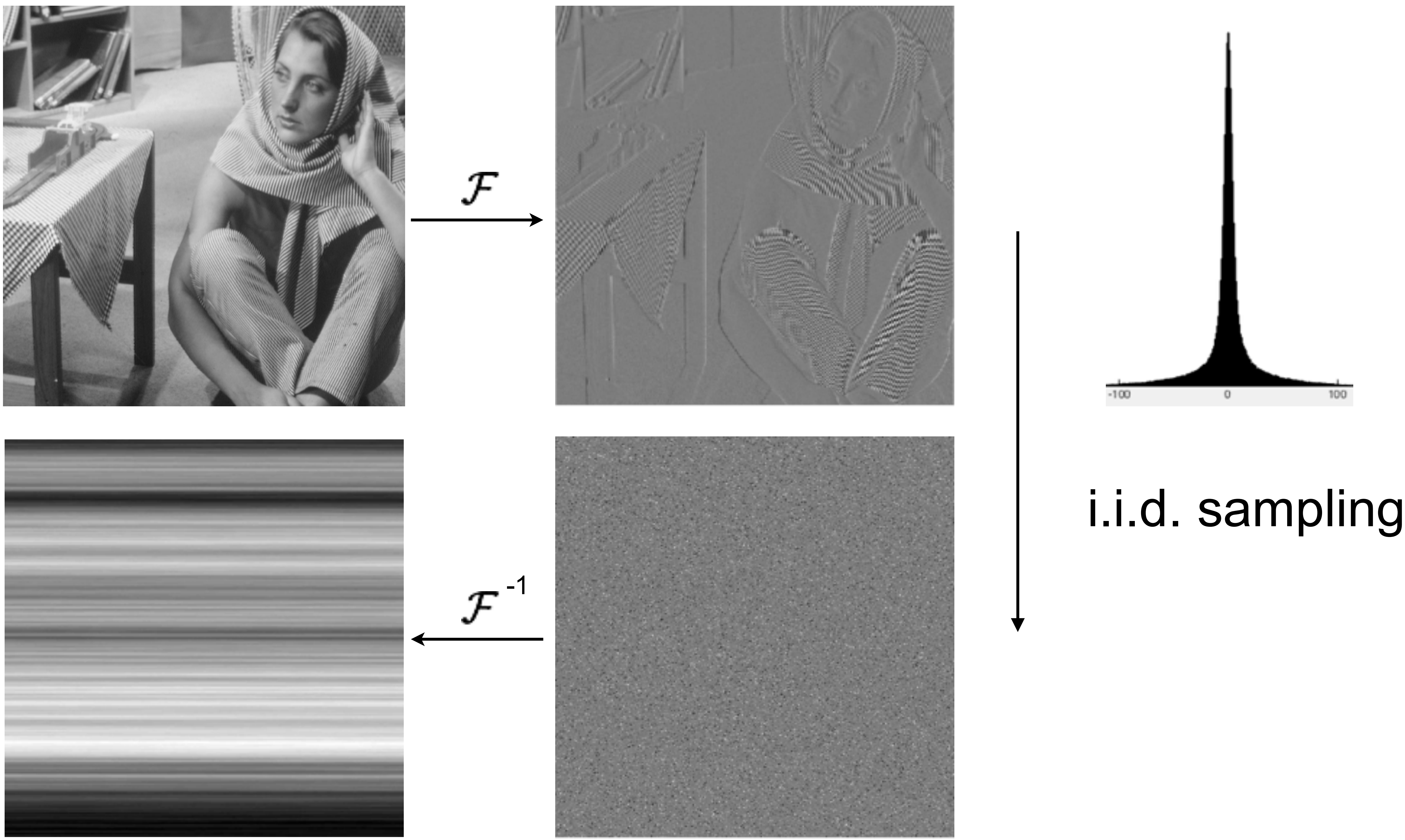




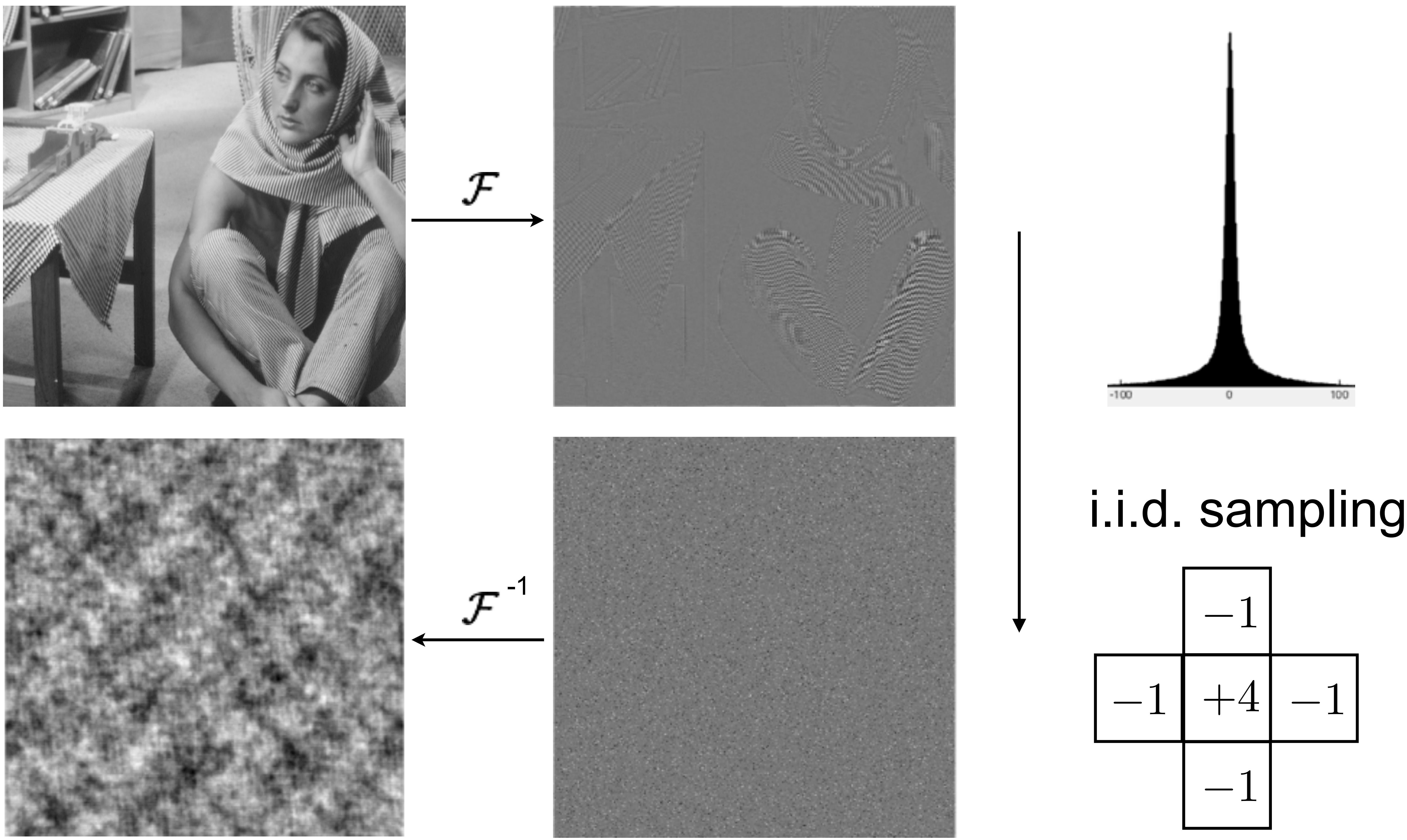
Minimax modeling



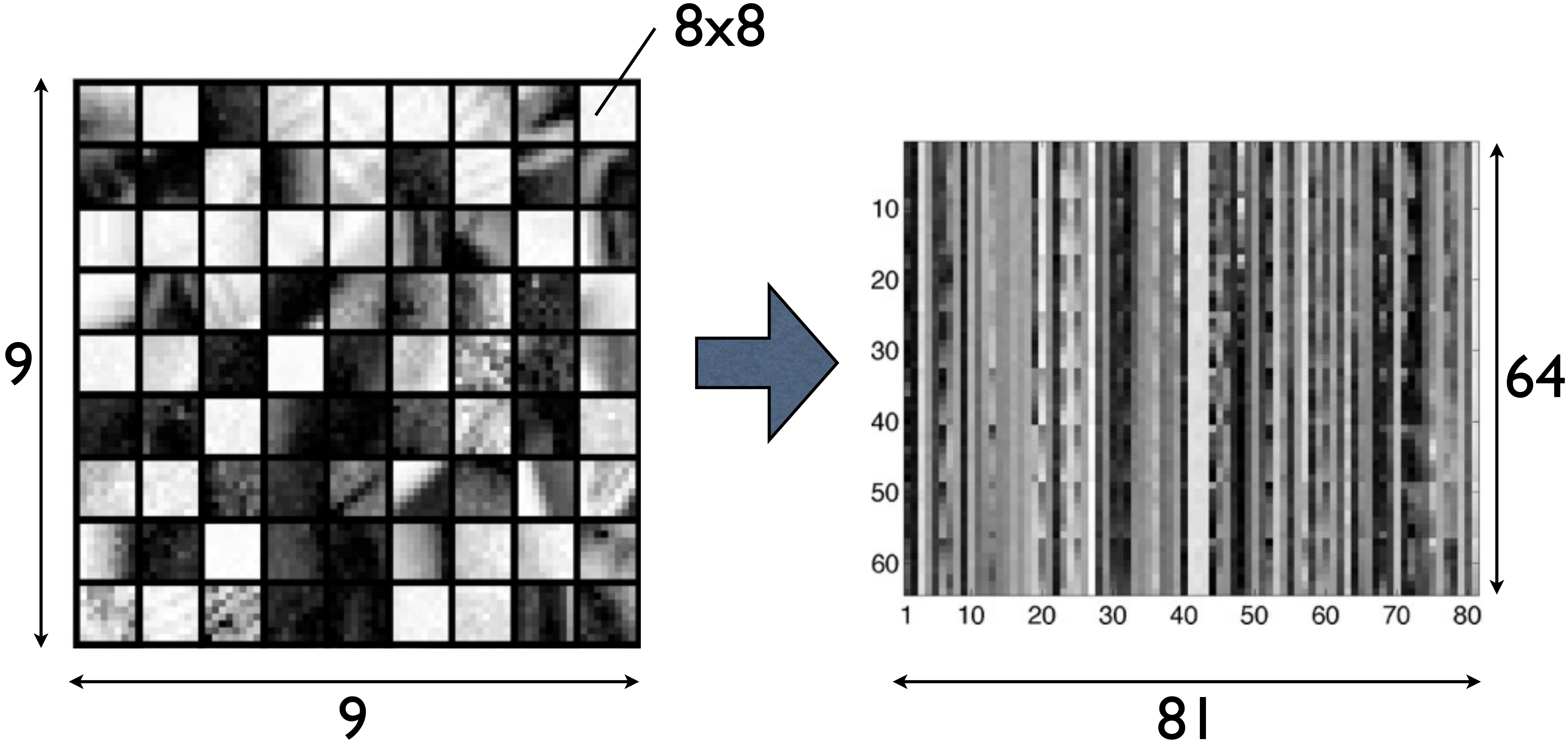
Minimax modeling



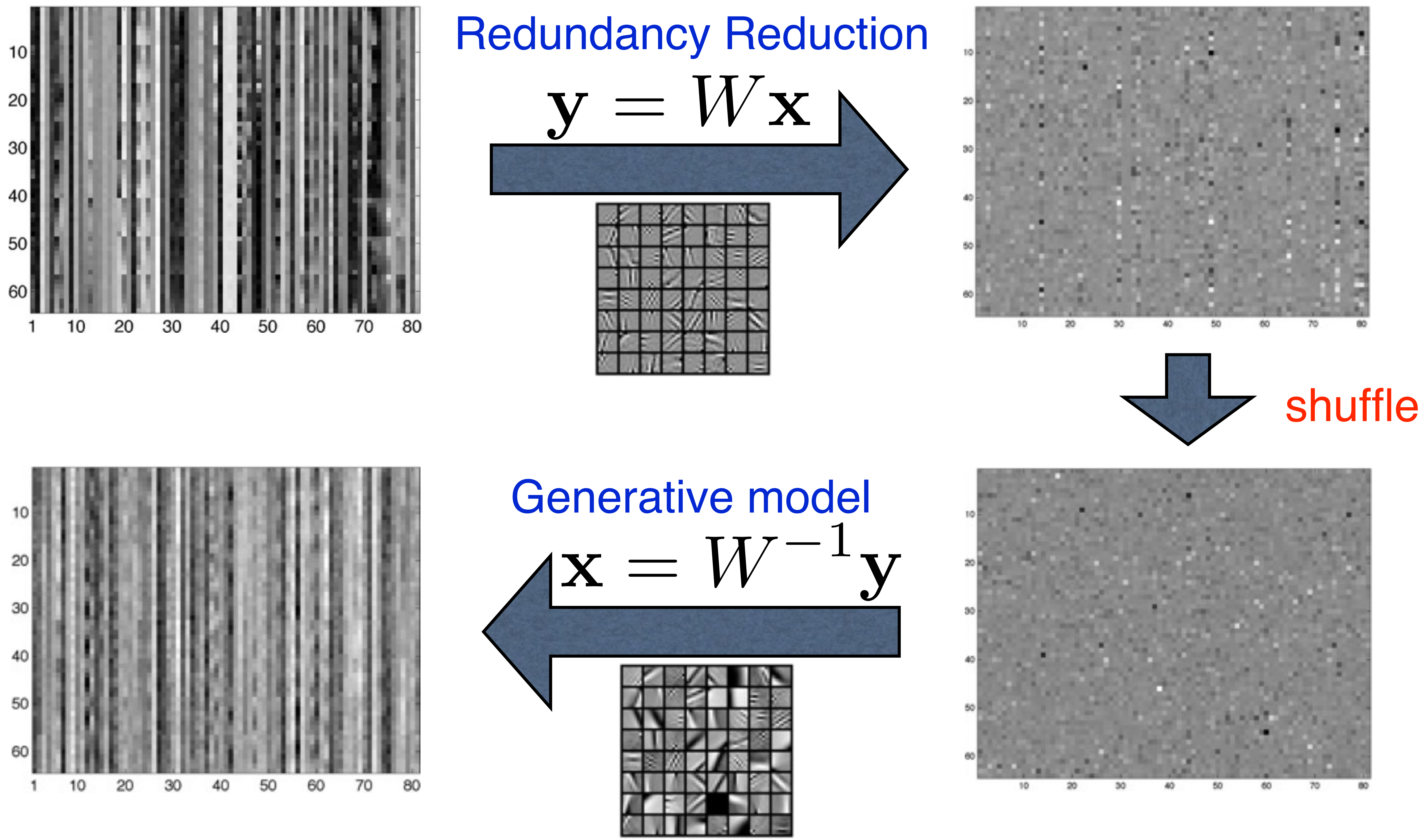
Minimax modeling



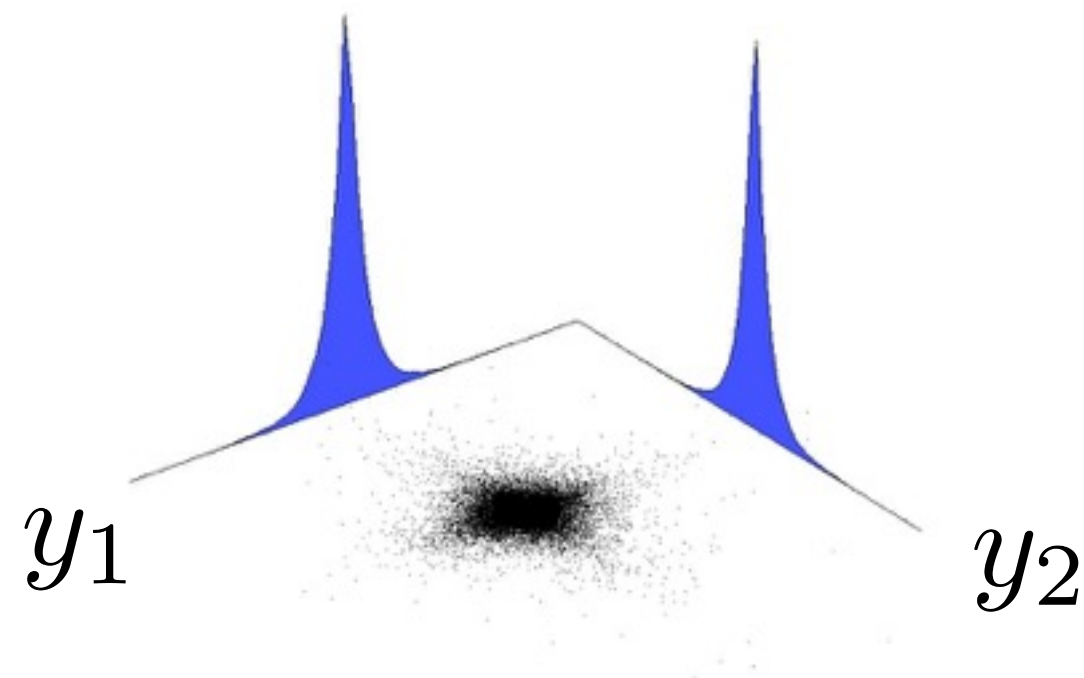
Modeling image patches

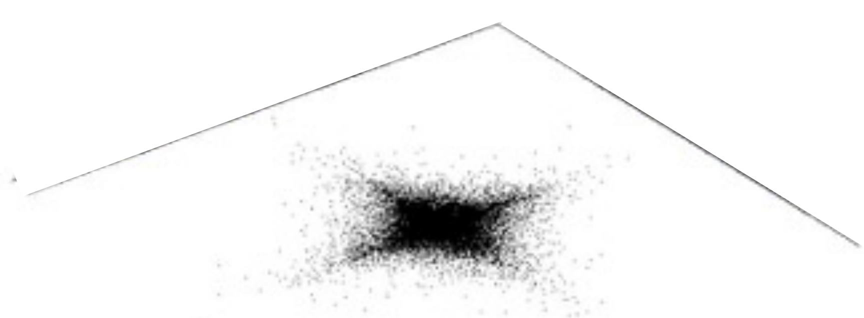


Independent Component Analysis (ICA)

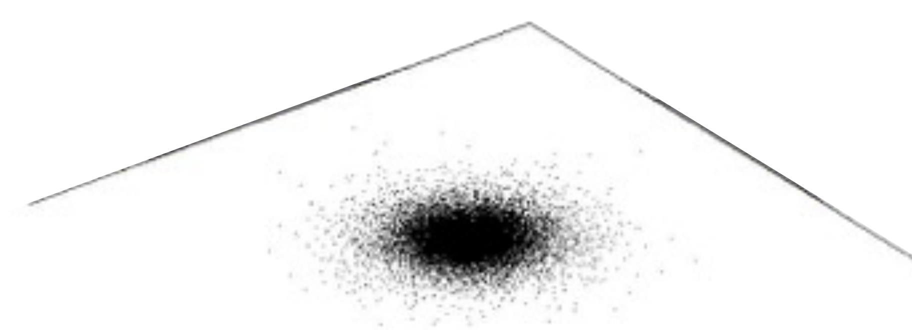


supergaussian (“sparse”) ICA coefficients



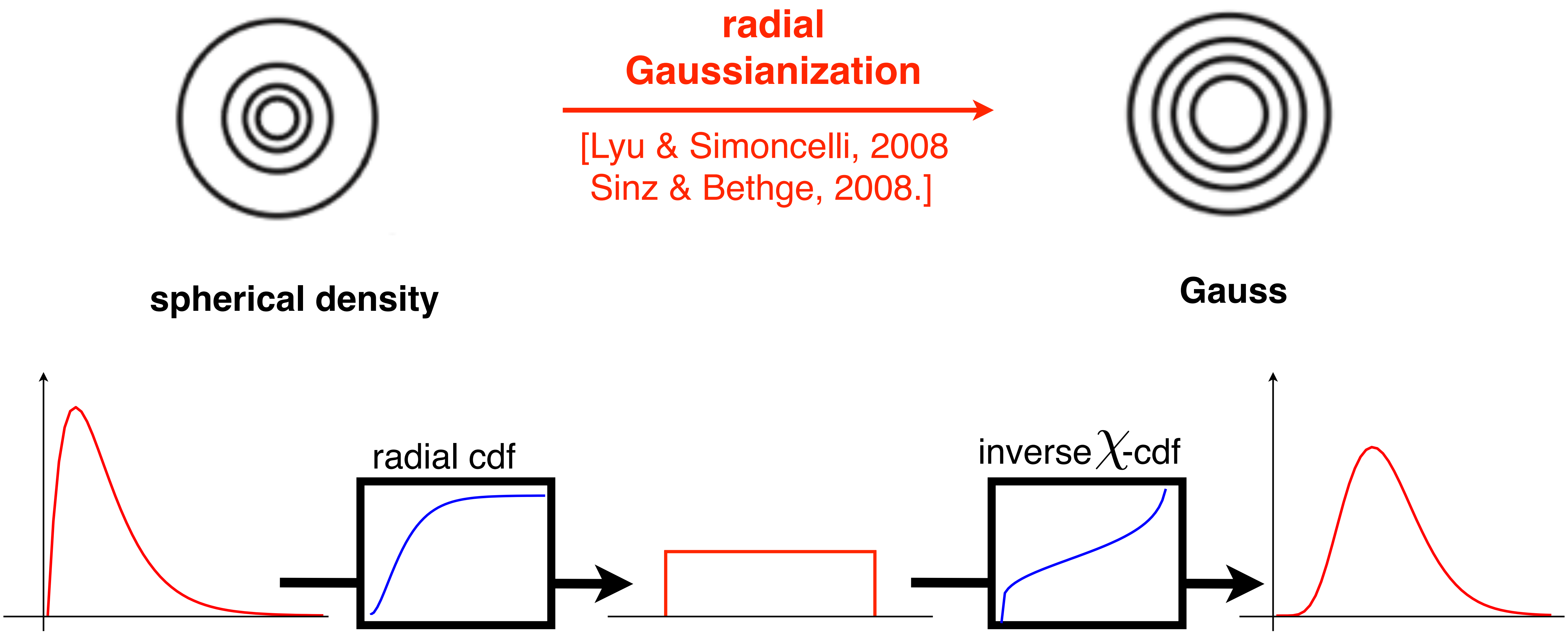


factorial density



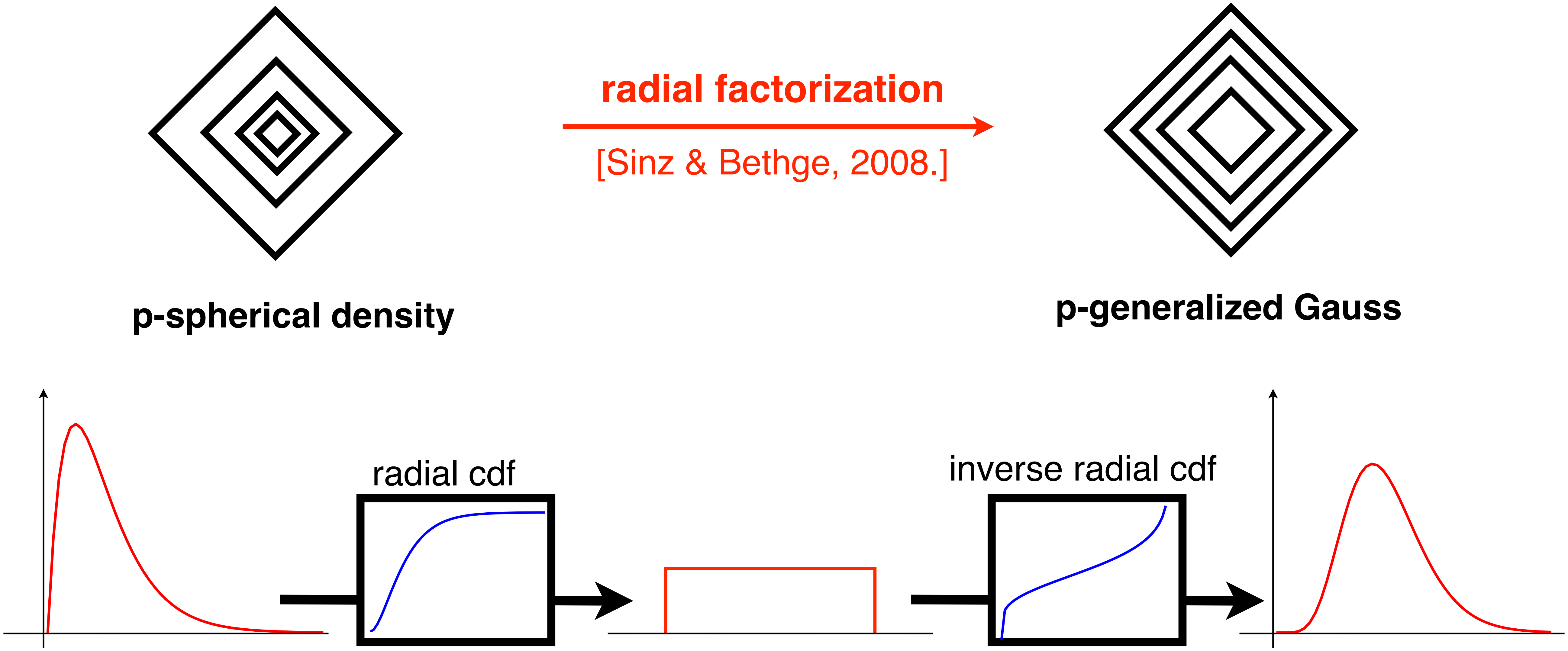
spherical density

Nonlinear redundancy reduction of spherical data



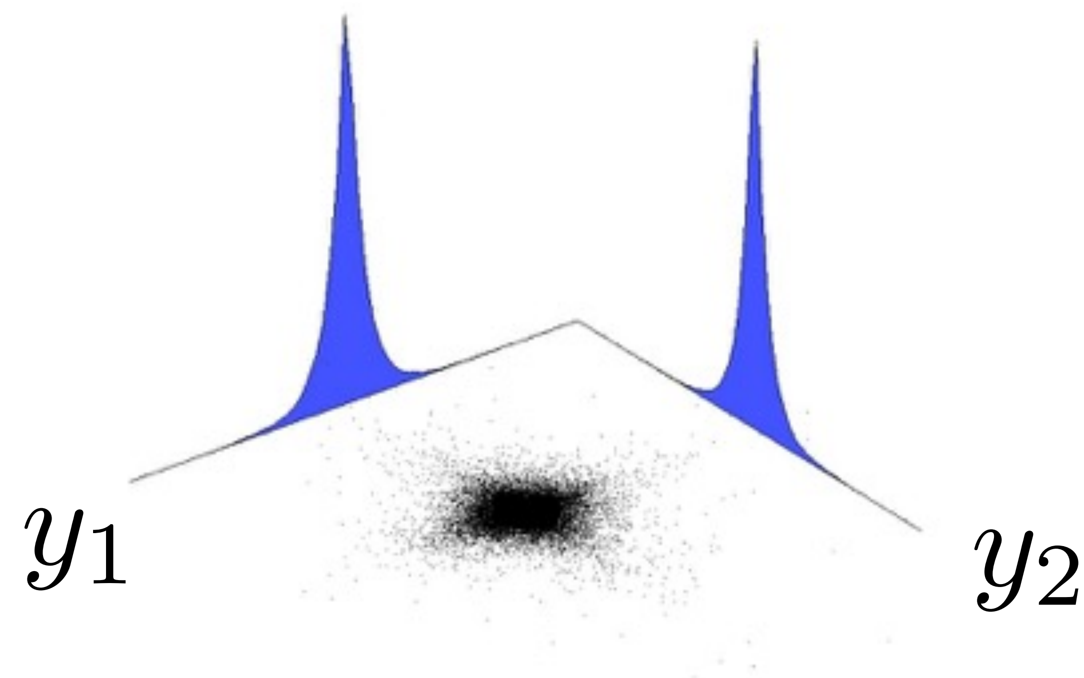
[Maxwell, Taylor's Phil. Mag. 19: 19-32., 1860.]

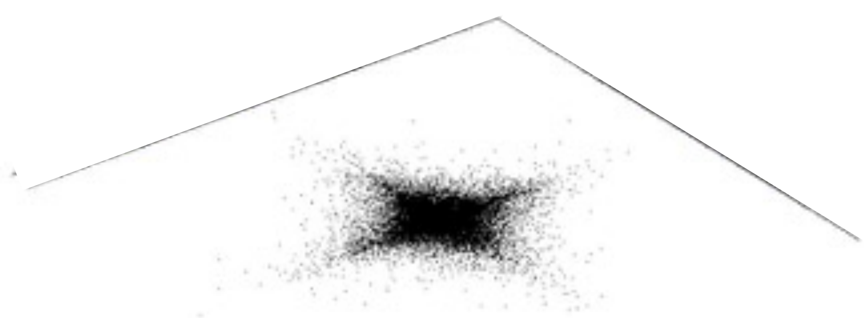
Nonlinear redundancy reduction of spherical data



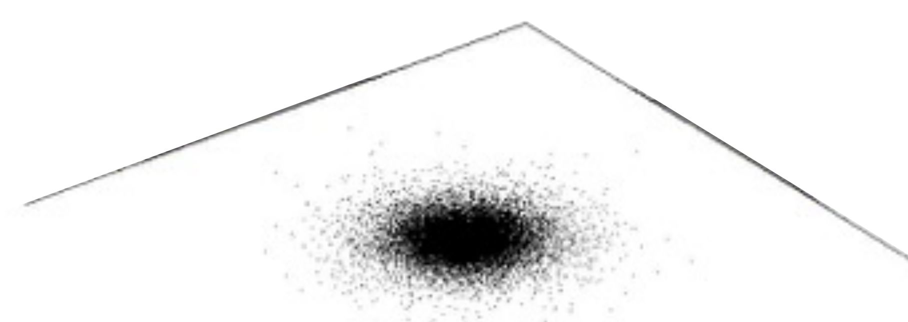
[Sinz, Gerwinn & Bethge, Characterization of the p-generalized Normal distribution, *Journal of Multivariate Analysis,* **100(5):** 817-820, 2009.]

supergaussian (“sparse”) ICA coefficients





factorial density

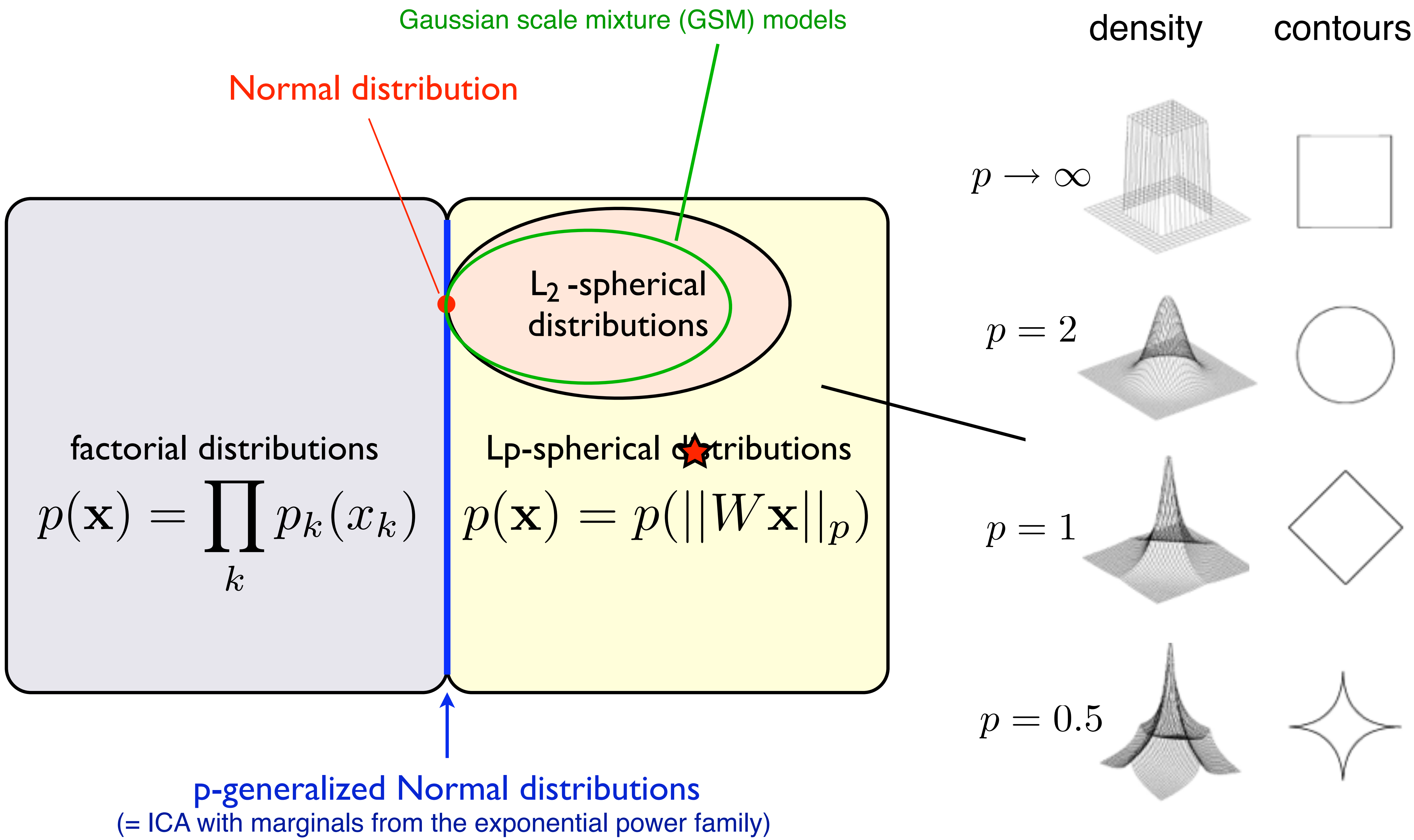


spherical density

$\Rightarrow$ **Non-factorial Lp-spherical density with p=1.3**

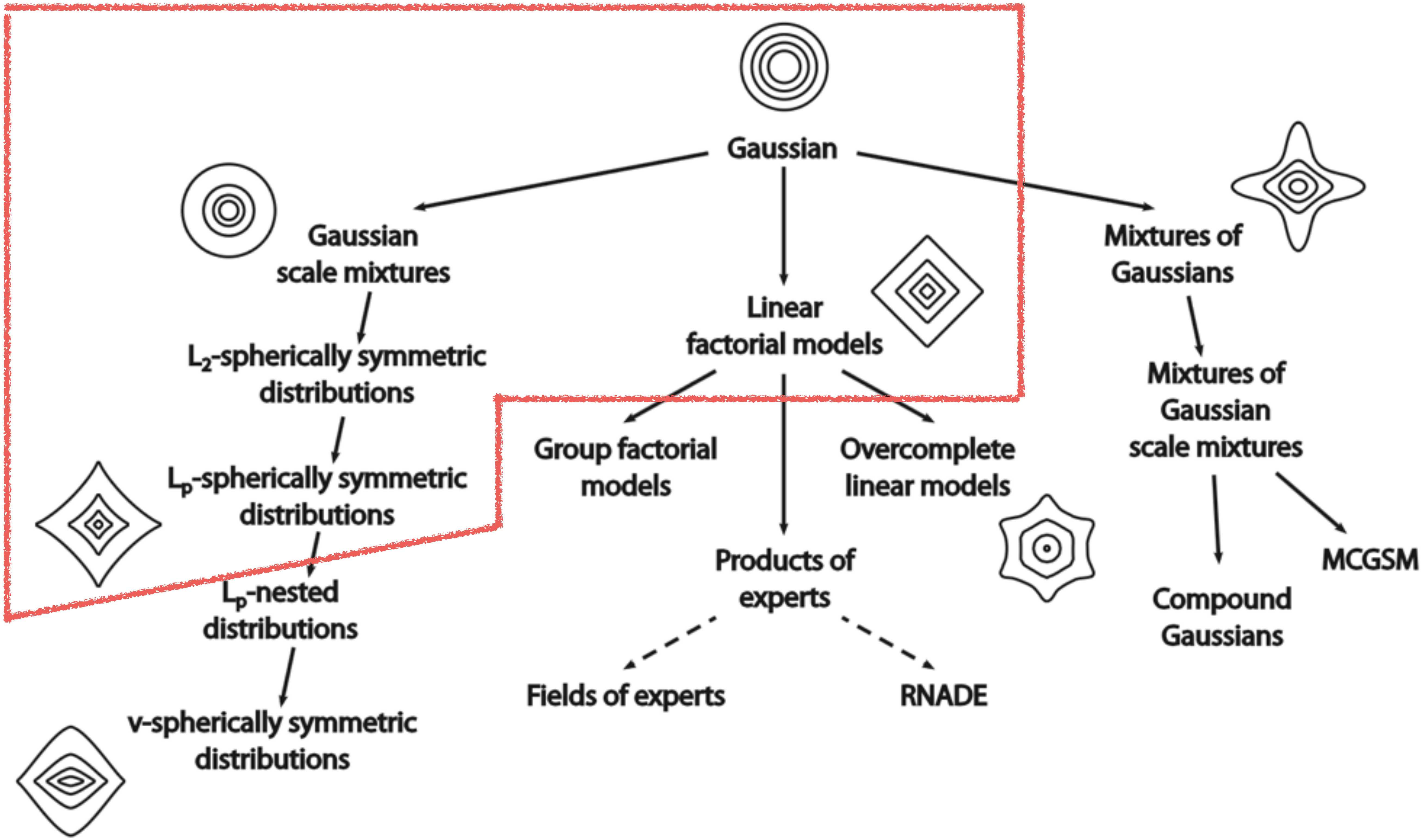
[Sinz & Bethge, NIPS, 2008.]

The class of Lp-spherical distributions

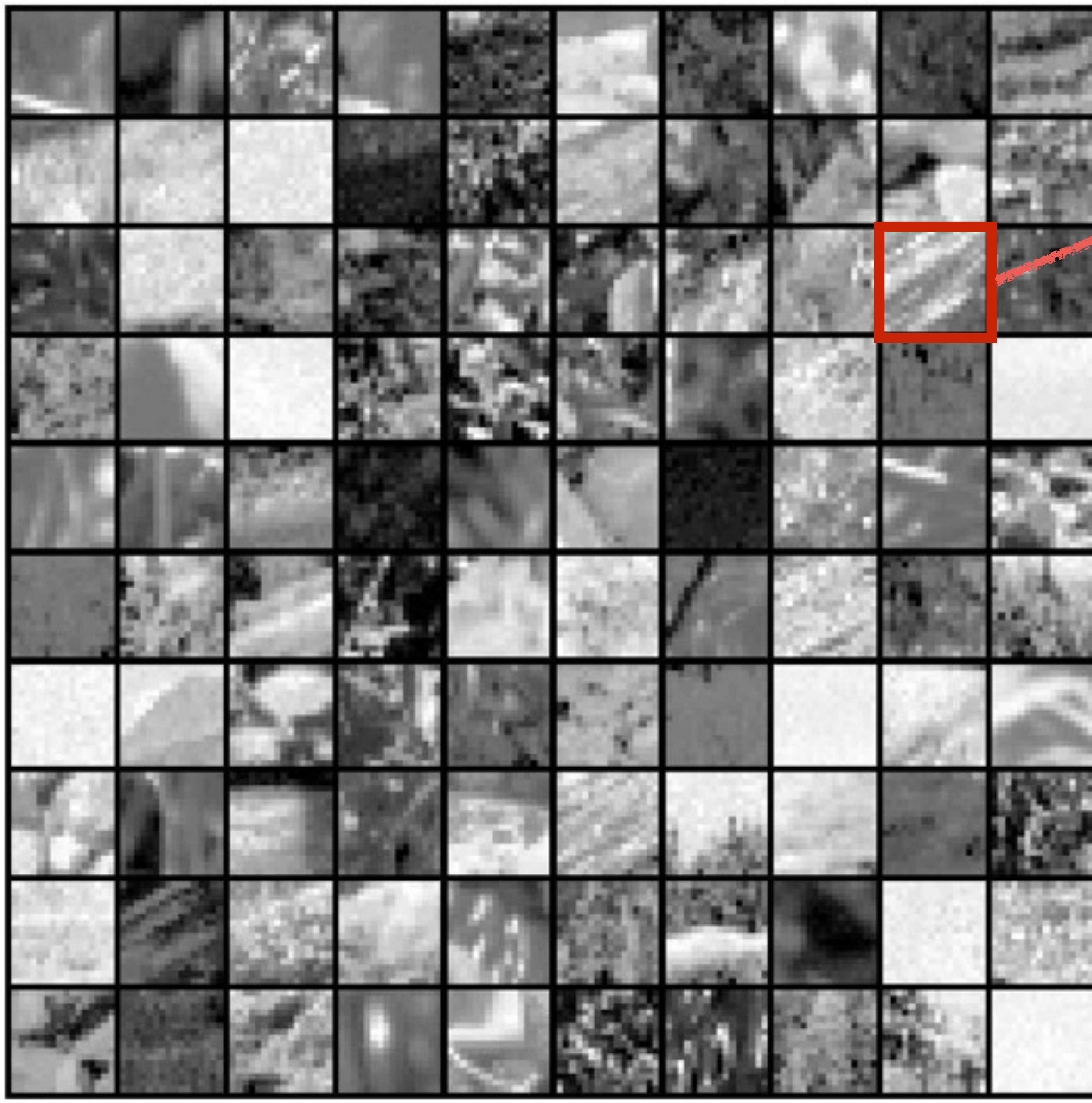


[Sinz, Gerwinn & Bethge, Characterization of the p-generalized Normal distribution, *Journal of Multivariate Analysis,* **100(5):** 817-820, 2009.]

Multivariate Density Estimation



Multivariate Density Estimation



$$\mathbf{x} \sim p(\mathbf{x})$$

Multivariate Density Estimation

Factorial generative model:

$$\hat{p}_s(\mathbf{s}) = \prod_{k=1}^{d} \hat{p}_k(s_k) \xrightarrow{\quad \mathbf{x} = \mathbf{f}(\mathbf{s}) \quad} \hat{p}_x(\mathbf{x})$$

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \\ x_{n1} & x_{n2} & \cdots \end{pmatrix} \underset{\mathbf{s} = \mathbf{f}^{-1}(\mathbf{x})}{\overset{\mathbf{x} = \mathbf{f}(\mathbf{s})}{\rightleftarrows}} S = \begin{pmatrix} \boxed{s_{11} \leftrightarrow s_{12} \leftrightarrow \cdots} \\ \boxed{s_{21} \leftrightarrow s_{22} \leftrightarrow \cdots} \\ \vdots \quad \vdots \\ s_{n1} \quad s_{n2} \quad \cdots \end{pmatrix}$$

"Synthesis"

"Analysis"

What is the loss function?

Factorial generative model:

$$\hat{p}_s(\mathbf{s}) = \prod_{k=1}^{d} \hat{p}_k(s_k) \quad \xrightarrow{\mathbf{x} = \mathbf{f}(\mathbf{s})} \quad \hat{p}_x(\mathbf{x})$$

Cross-entropy:

Kullback-Leibler divergence

$$E[-\log \hat{p}_x(\mathbf{x})] = h[p_x(\mathbf{x})] + D_{KL}[p_x(\mathbf{x}) || \hat{p}_x(\mathbf{x})]$$

$$= h[p_x(\mathbf{x})] + D_{KL}[p_s(\mathbf{s}) || \hat{p}_s(\mathbf{s})]$$

What is the loss function?

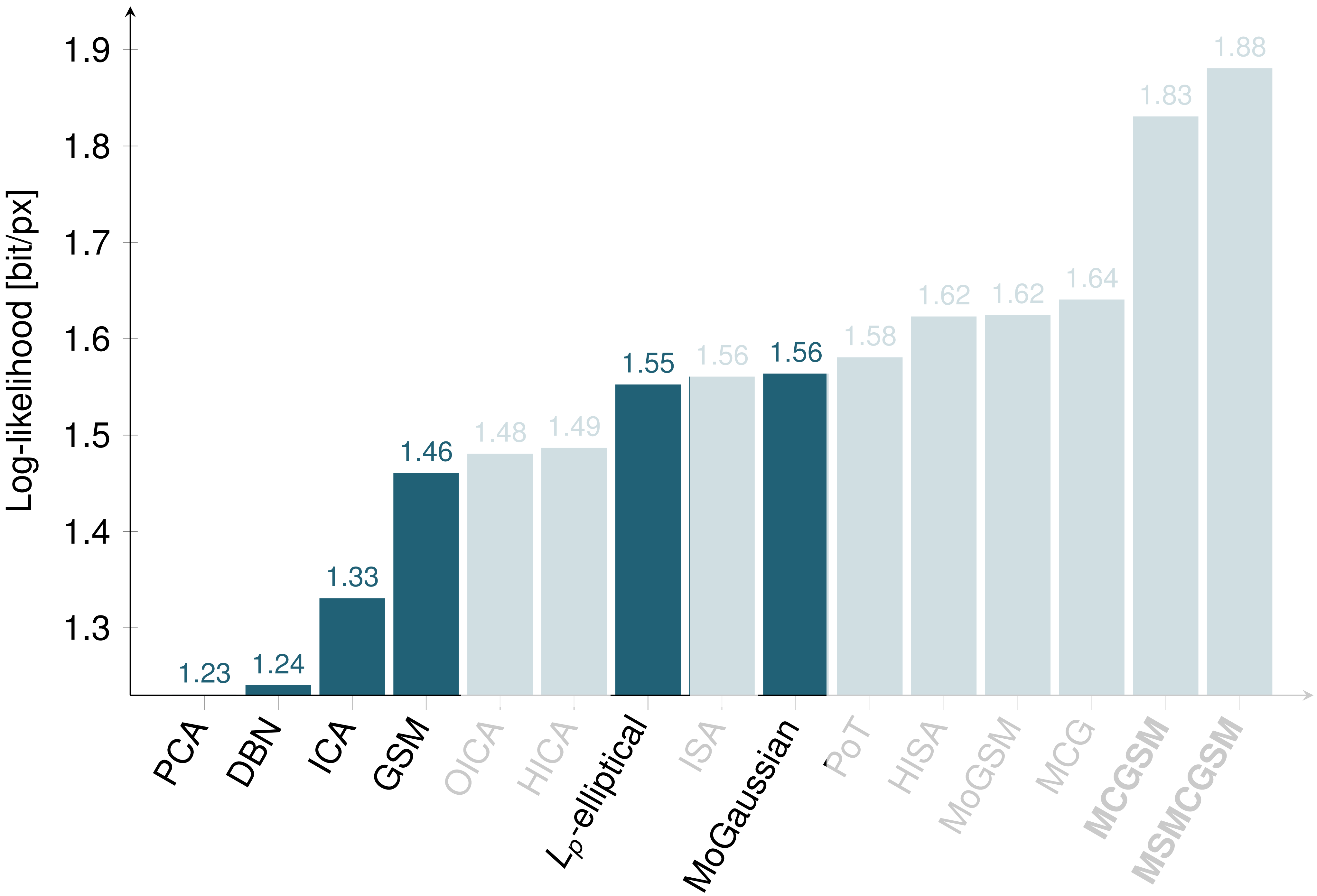
$$D_{KL}[p_s(\mathbf{s})||\hat{p}_s(\mathbf{s})] =$$

$$= \underbrace{D_{KL}\left[p_s(\mathbf{s})\middle|\middle|\prod_{k=1}^{d} p_k(s_k)\right]}_{=I[p_s(\mathbf{s})]} + \sum_{k=1}^{d} D_{KL}[p_k(s_k)||\hat{p}_k(s_k)]$$

“Multi-Information”

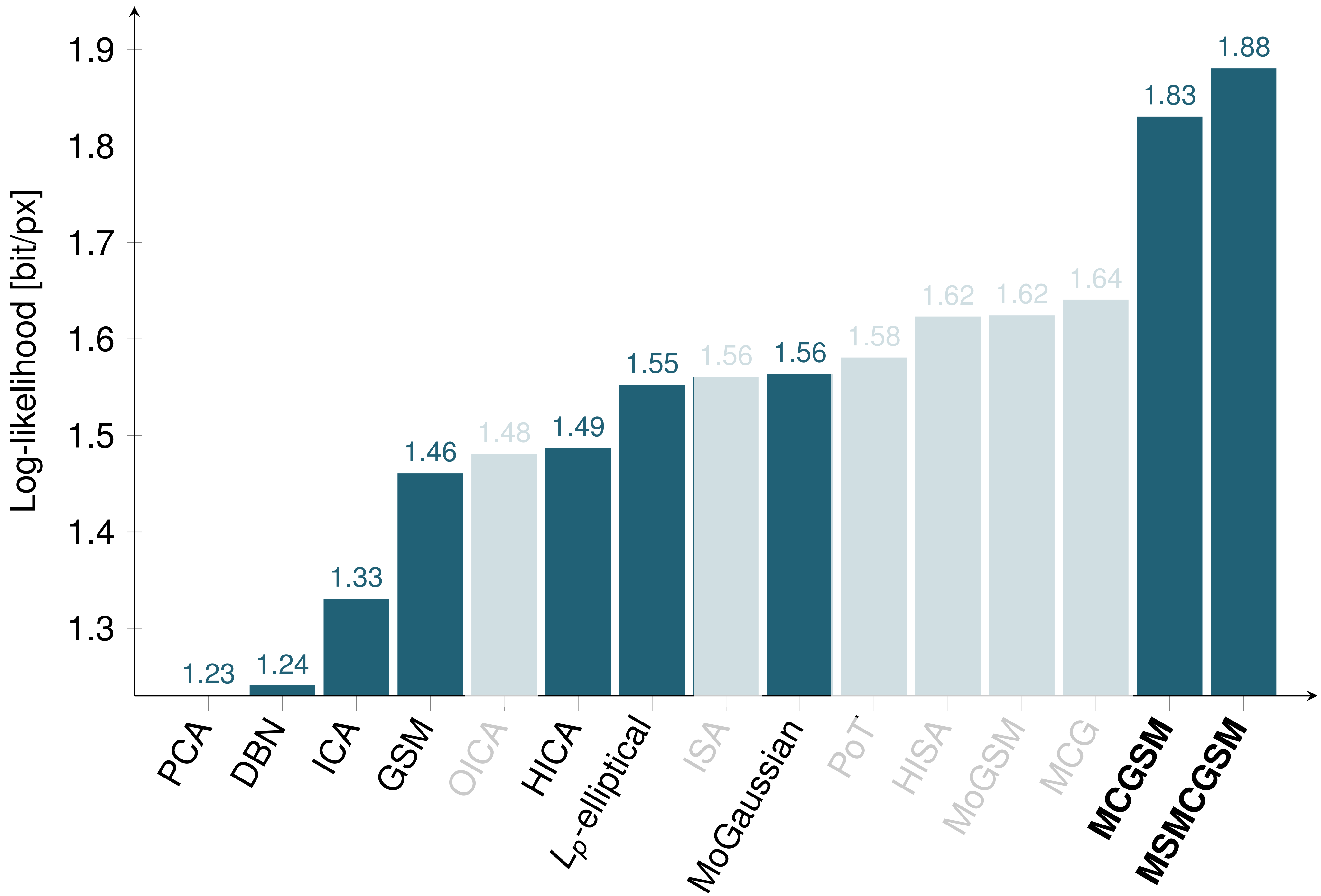
Model comparison

Lucas Theis



Model comparison

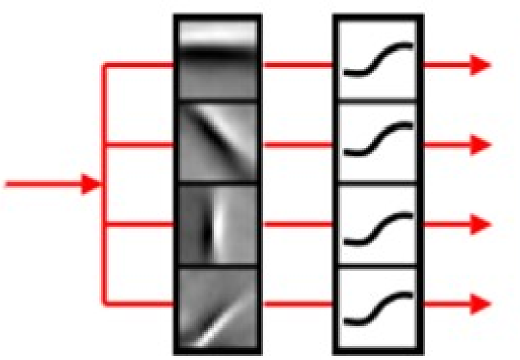
Lucas Theis



Multi-layer ICA

Reshad
Hosseini

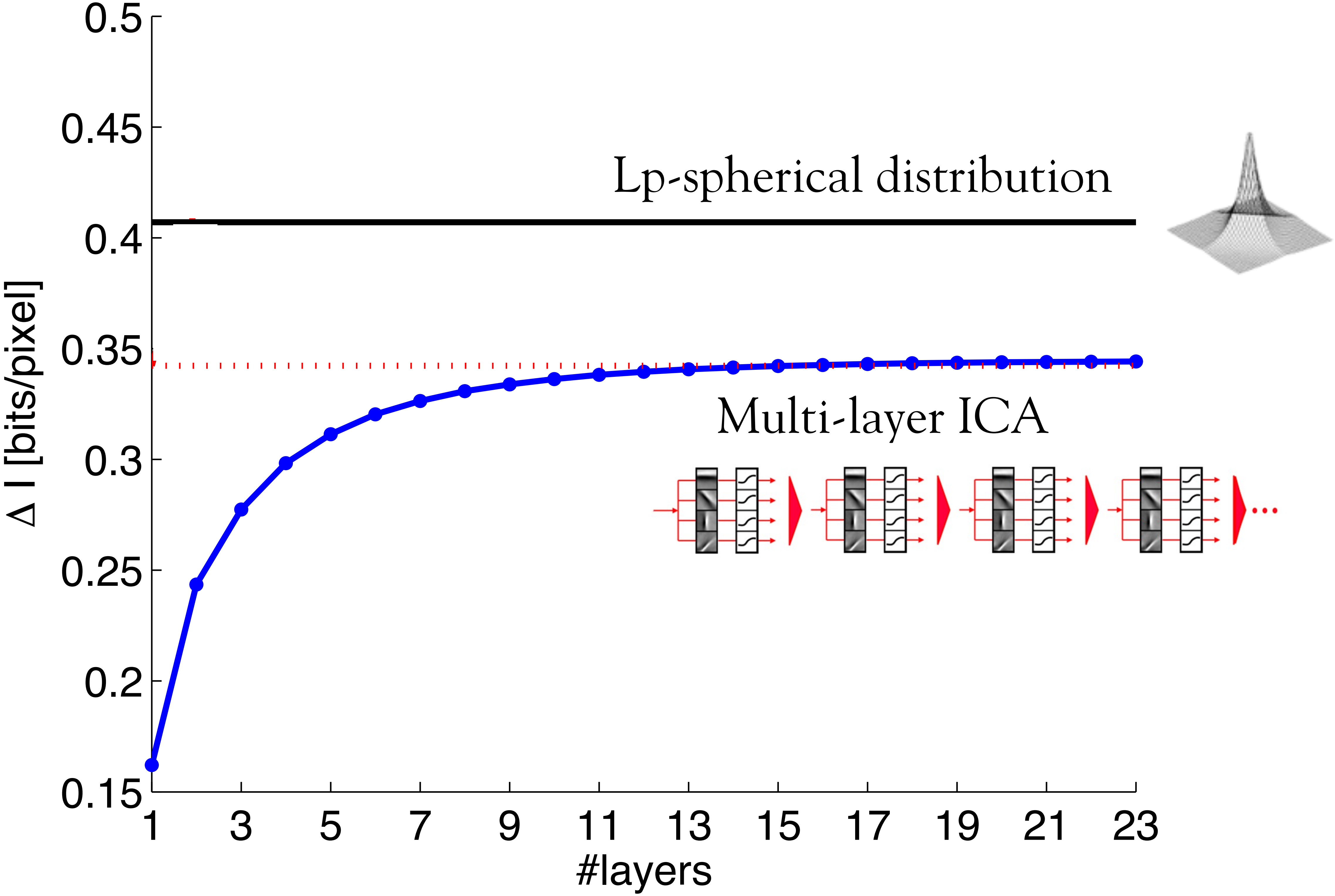
search for filters with
non-Gaussian histograms



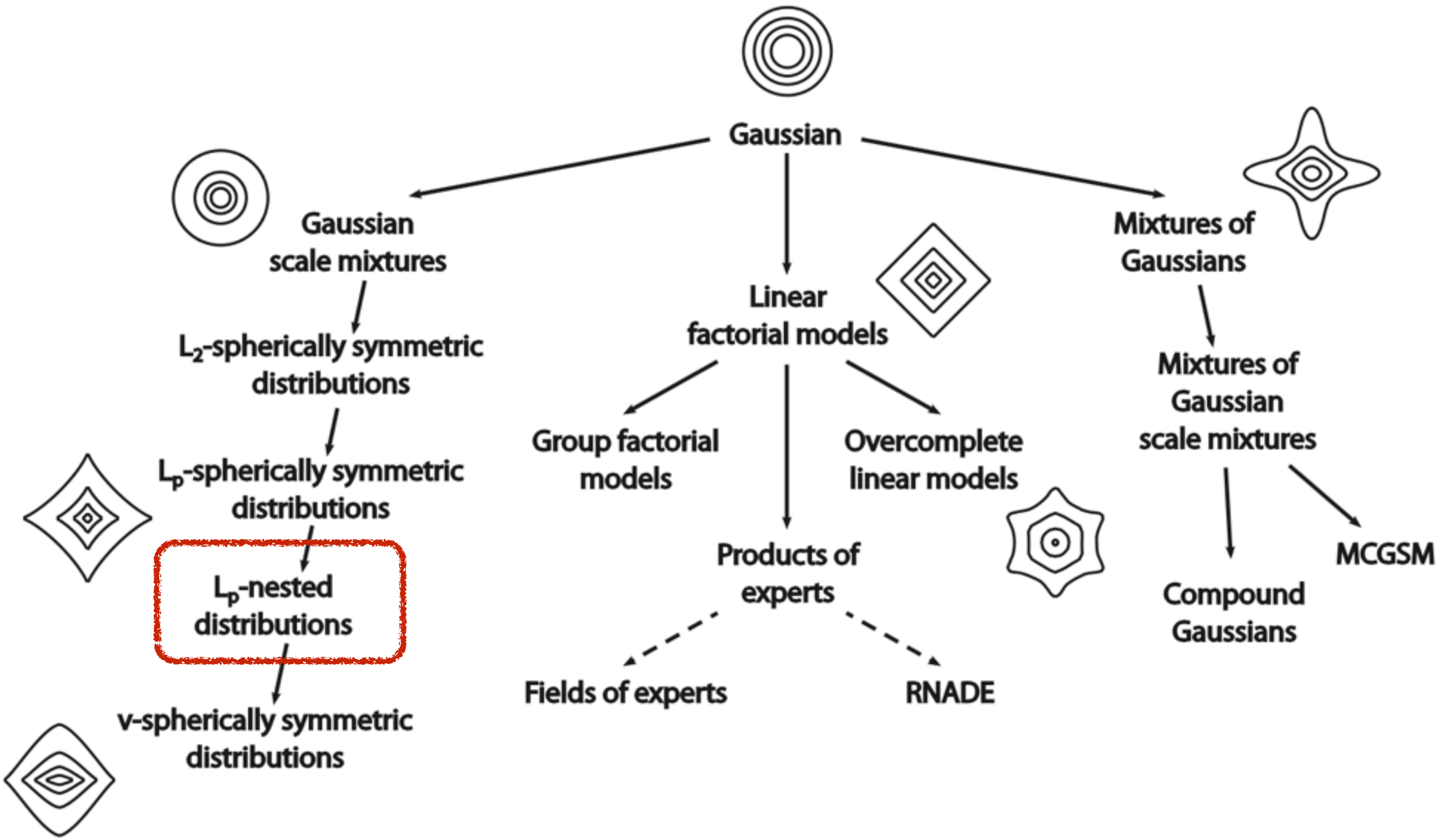
"Gaussianize"

Friedman (1987): Exploratory projection pursuit. J Am Stat Assoc 89: 249-266.

Multi-layer ICA



Model comparison



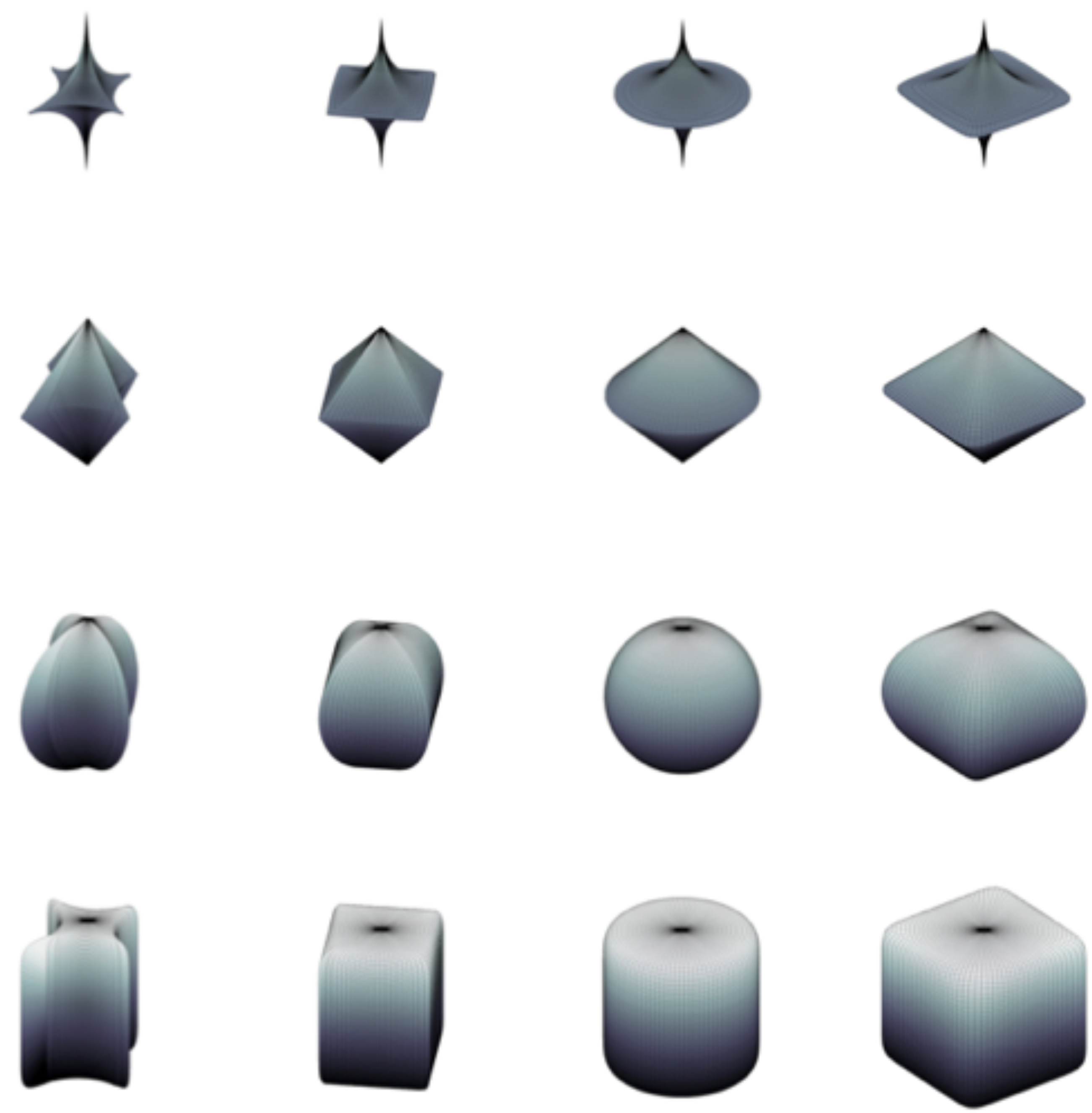
Model comparison

Lp-nested distribution:

$$\nu(\mathbf{y}) = \left(\left(|y_1|^{p_1} + |y_2|^{p_1} \right)^{\frac{p_0}{p_1}} + |y_3|^{p_0} \right)^{\frac{1}{p_0}}$$

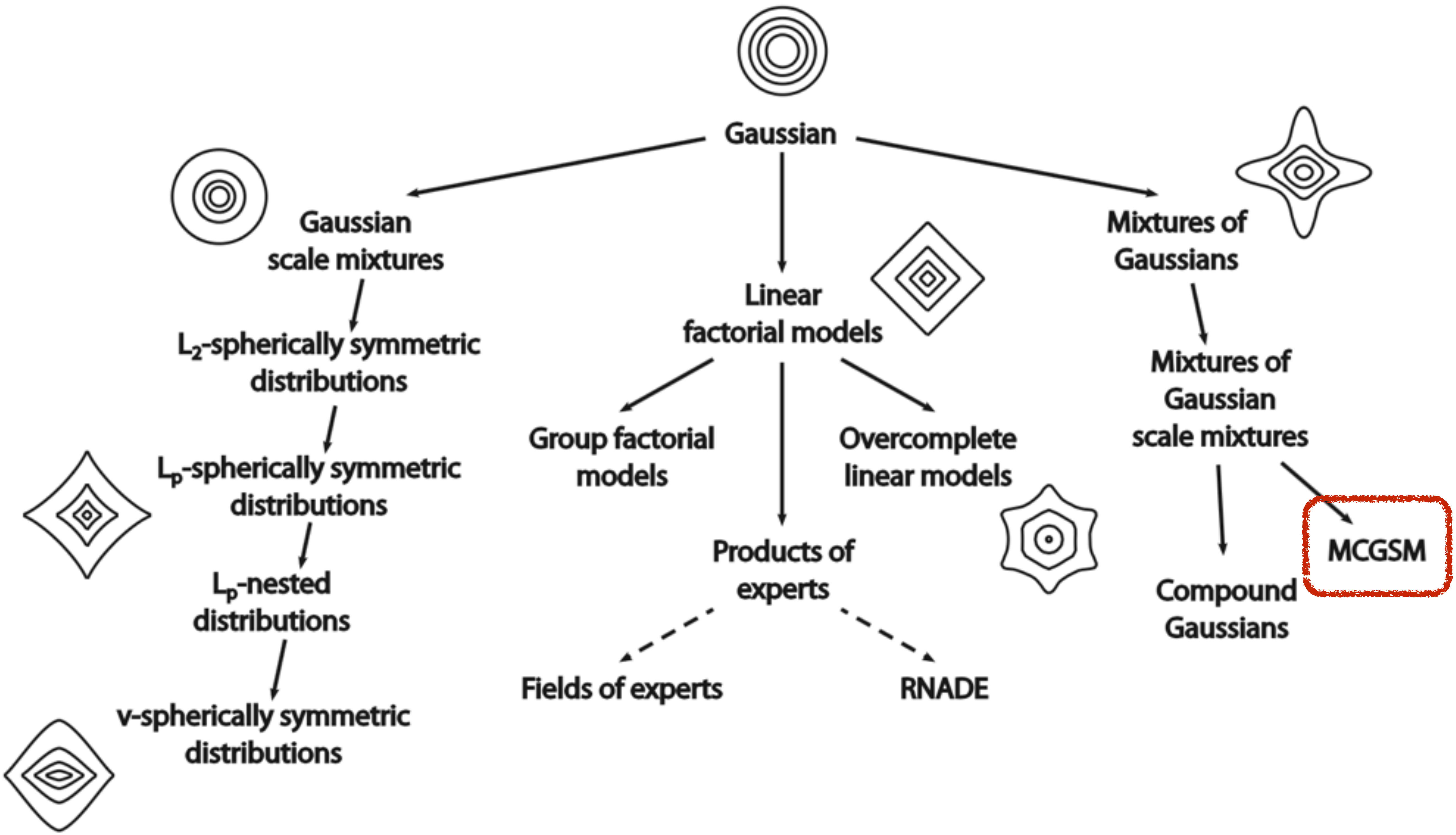
$$p(\mathbf{x}) = p(||W\mathbf{x}||_p)$$

$$p(\mathbf{x}) = p(\nu(W\mathbf{x}))$$



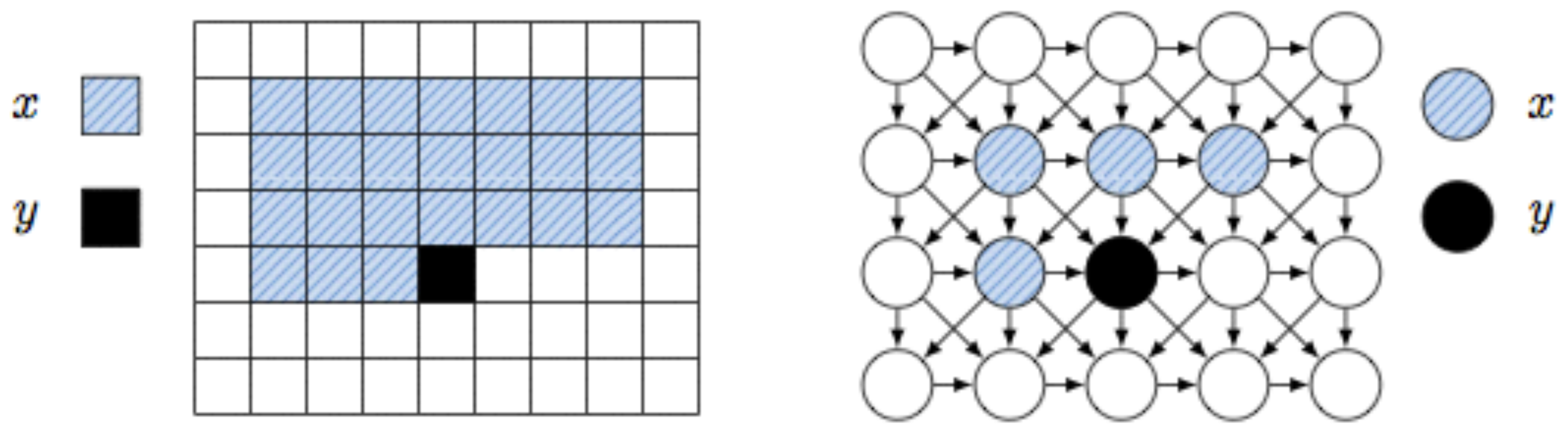
[Sinz & Bethge (2010). JMLR, 3409-3451.]

Model comparison



MCGSM: a directed mixture of experts model of natural images

Lucas Theis



Key advantages:

1.) Built-in translation invariance

2.) Model if you can and ignore if not

Mixture of conditional GSMs (MCGSM)


Lucas Theis

$$p(y|\mathbf{x}) = \sum_{c=1}^{m}\sum_{s=1}^{n} \underbrace{p(\overset{\text{covariance class label}}{c}|\mathbf{x})\, p(\overset{\text{scale class label}}{s}|c,\mathbf{x})}_{Gating}\ \underbrace{p(y|c,s,\mathbf{x})}_{Prediction}$$

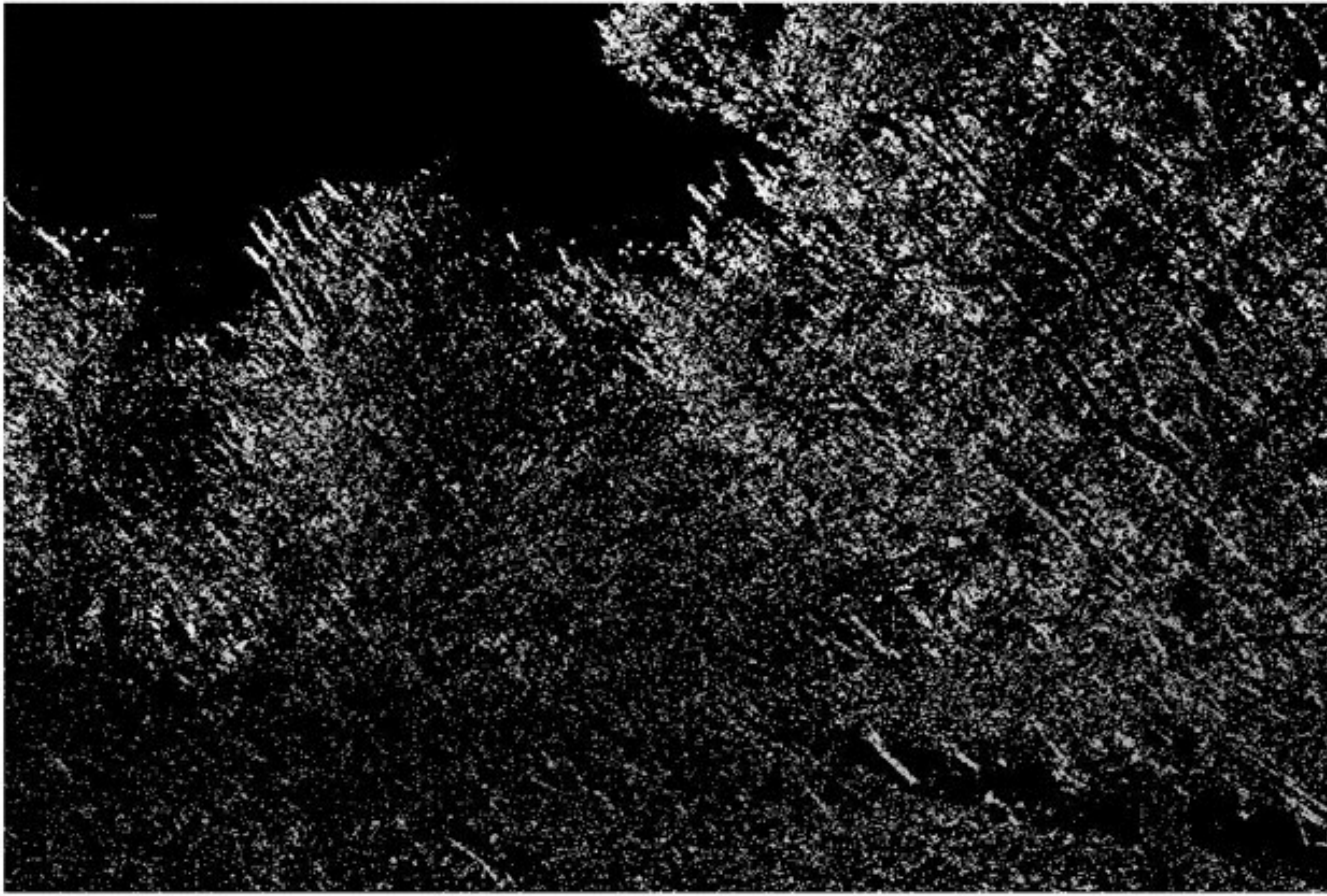
Gating: $$p(c,s|\mathbf{x}) \propto \exp\left(-\frac{\lambda_{c,s}}{2}\mathbf{x}^\top K_c \mathbf{x}\right)$$

Prediction: $$p(y|c,s,\mathbf{x}) \propto \exp\left(-\frac{1}{2}\frac{(y-\mathbf{w}_c^\top\mathbf{x})^2}{\sigma_{s,c}^2}\right)$$

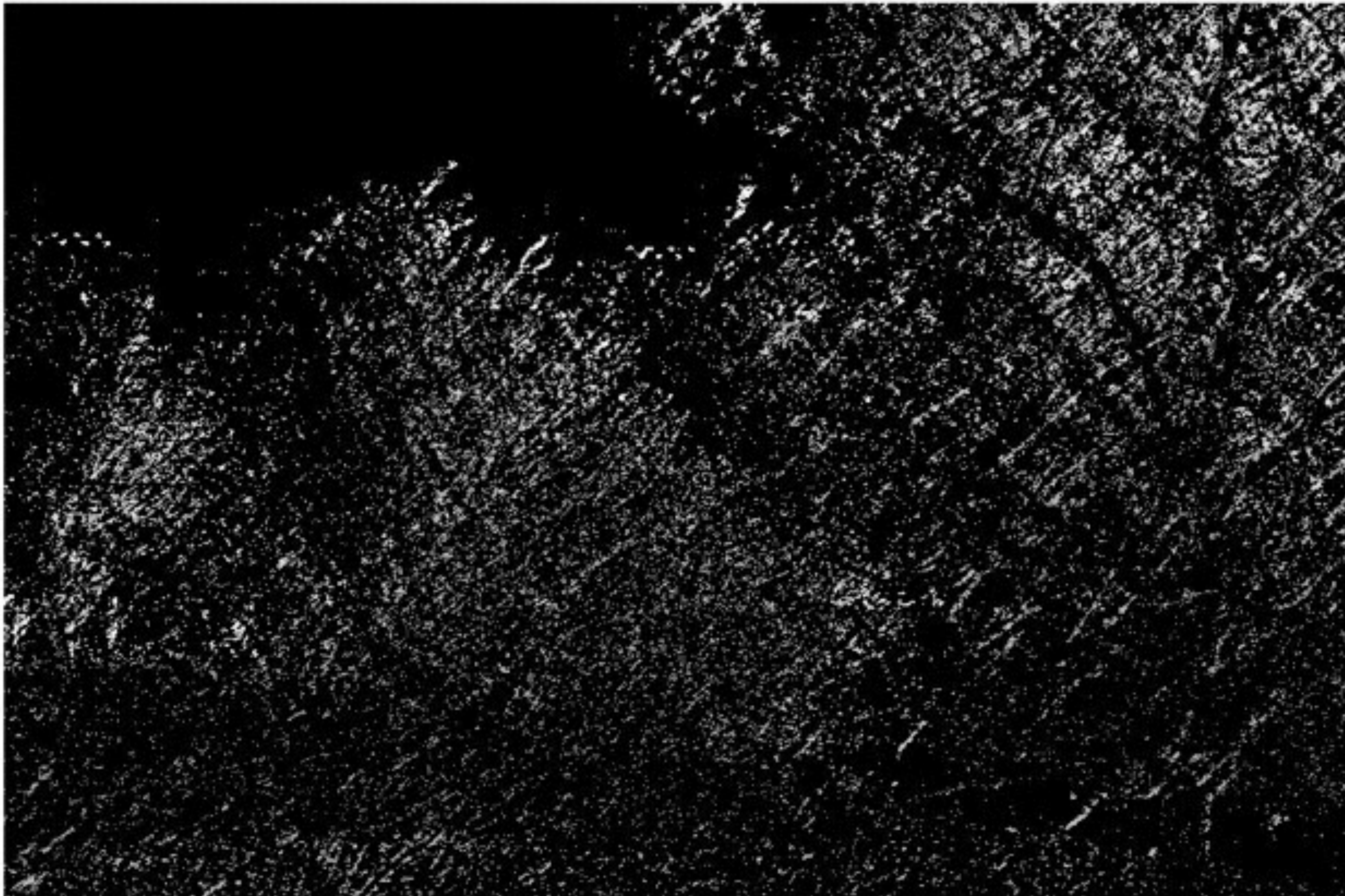
Theis et al, PLoS One, 2012.

MCGSM feature maps

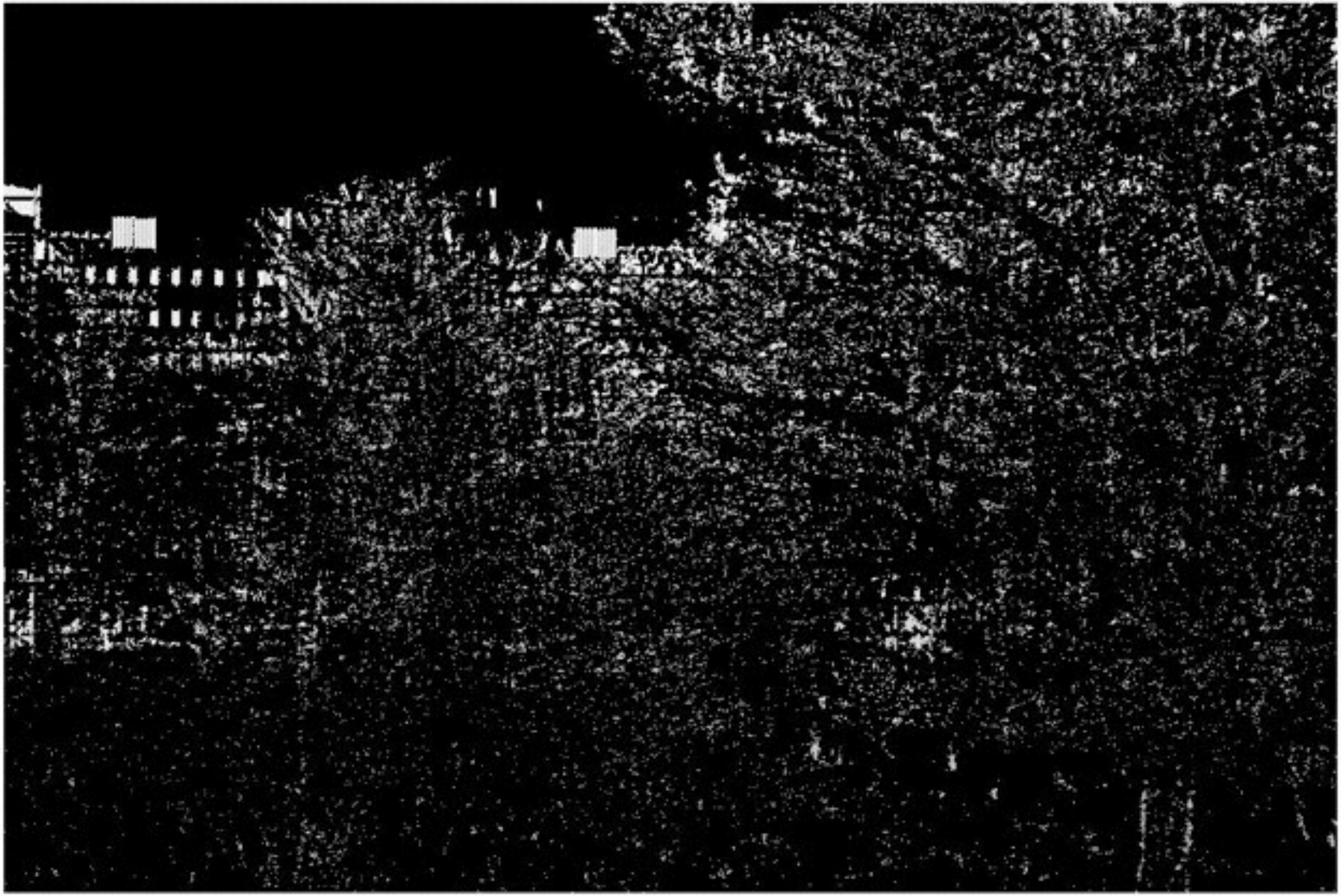
MCGSM feature maps



MCGSM feature maps

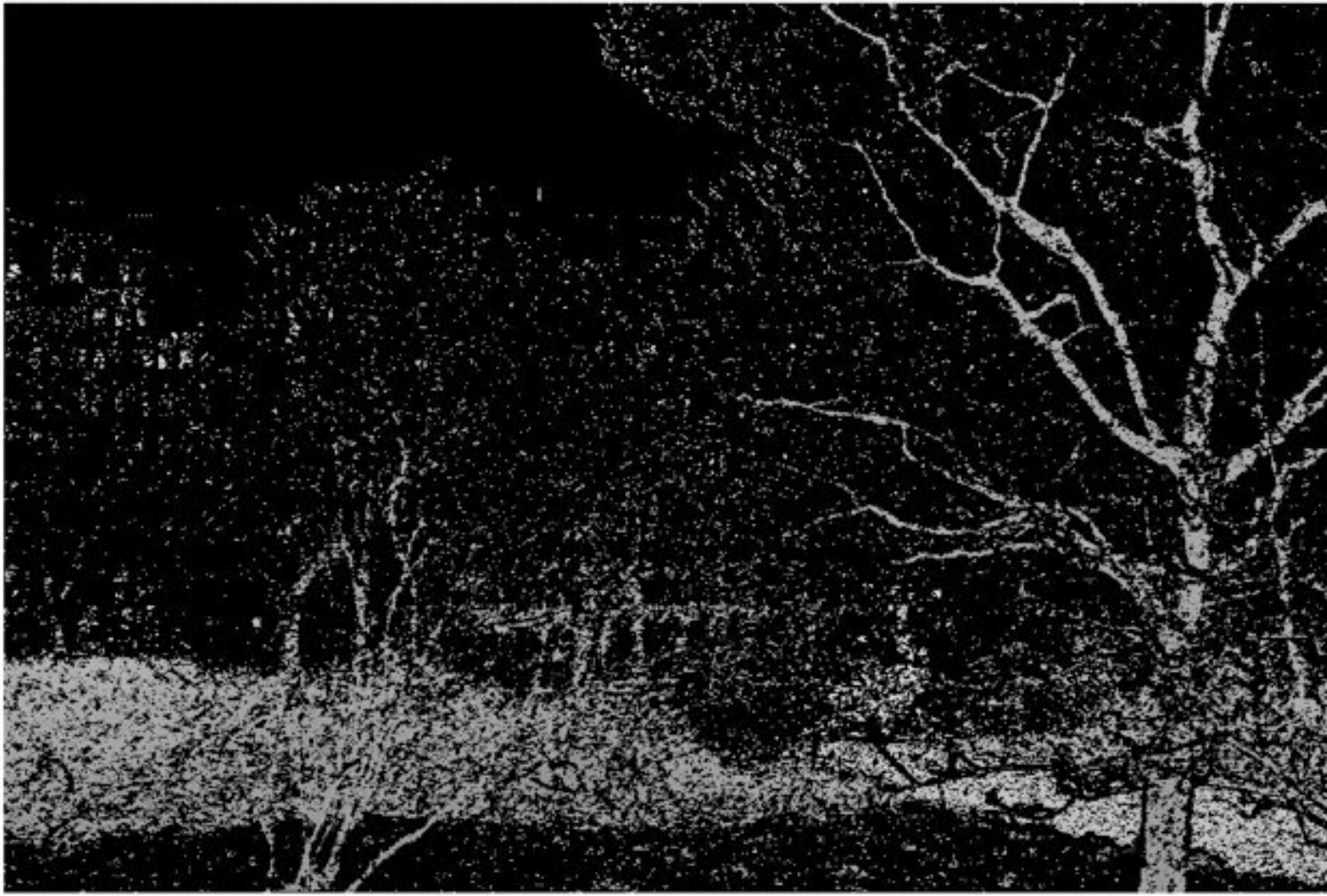


MCGSM feature maps



MCGSM feature maps

MCGSM feature maps

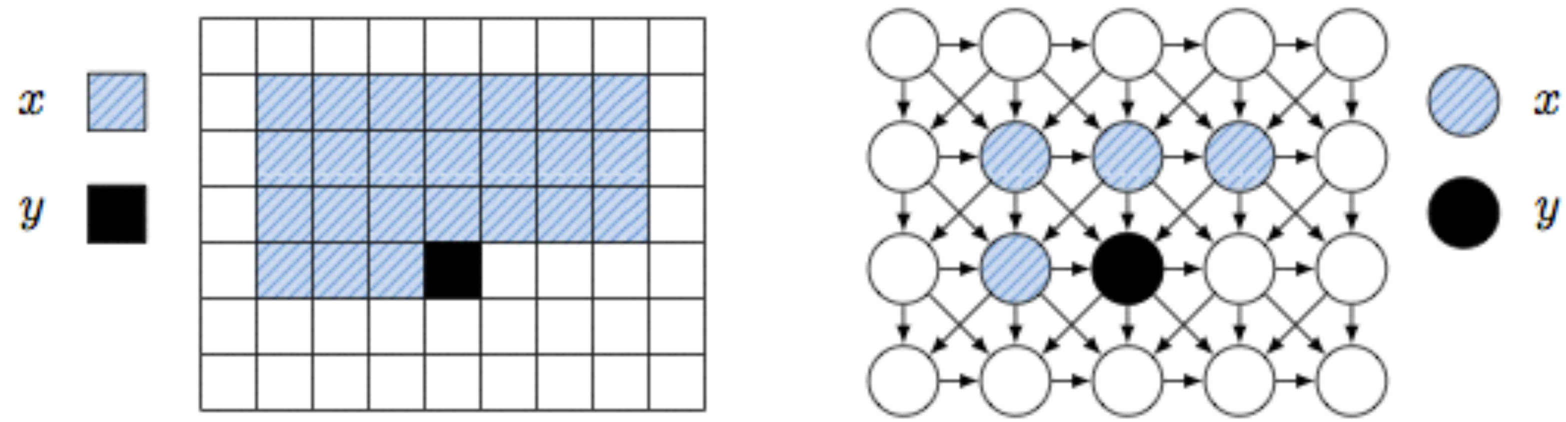


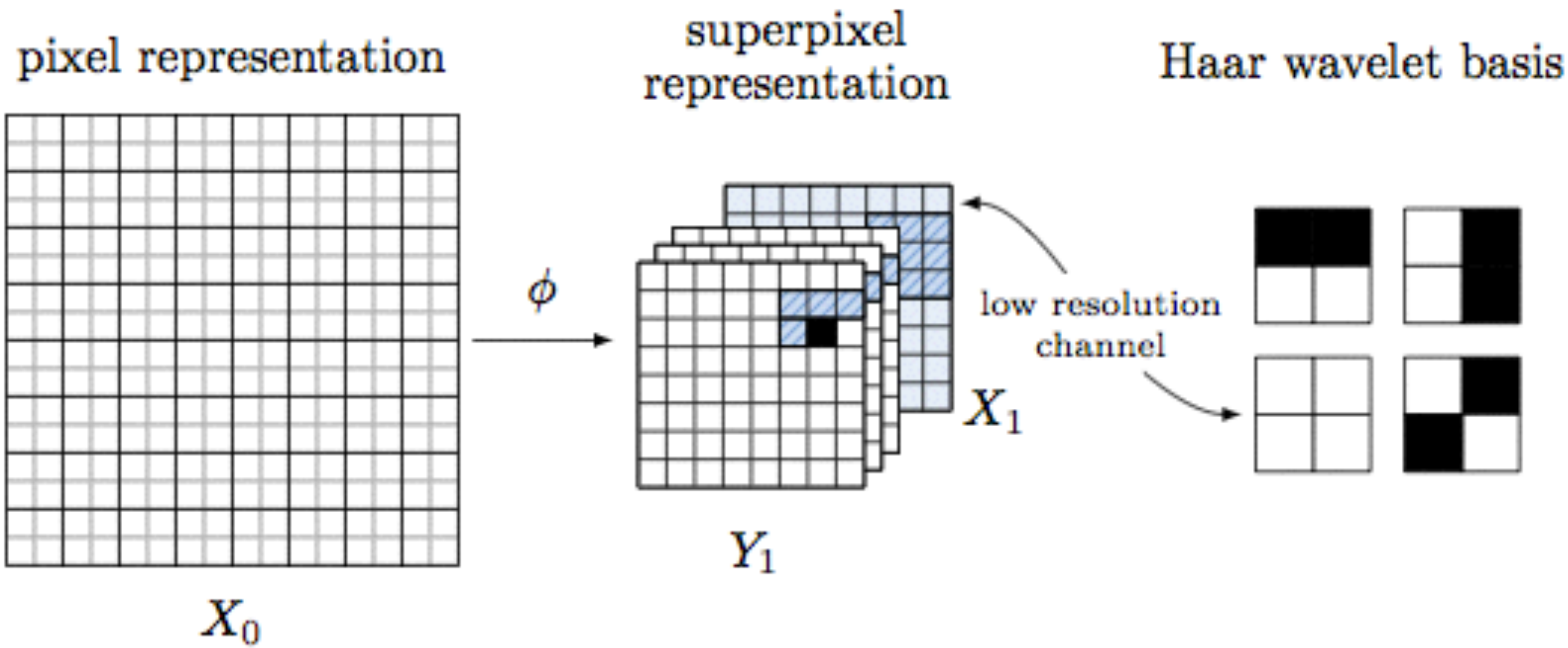
MCGSM feature maps

MCGSM feature maps

Multi-scale MCGSM


Lucas Theis

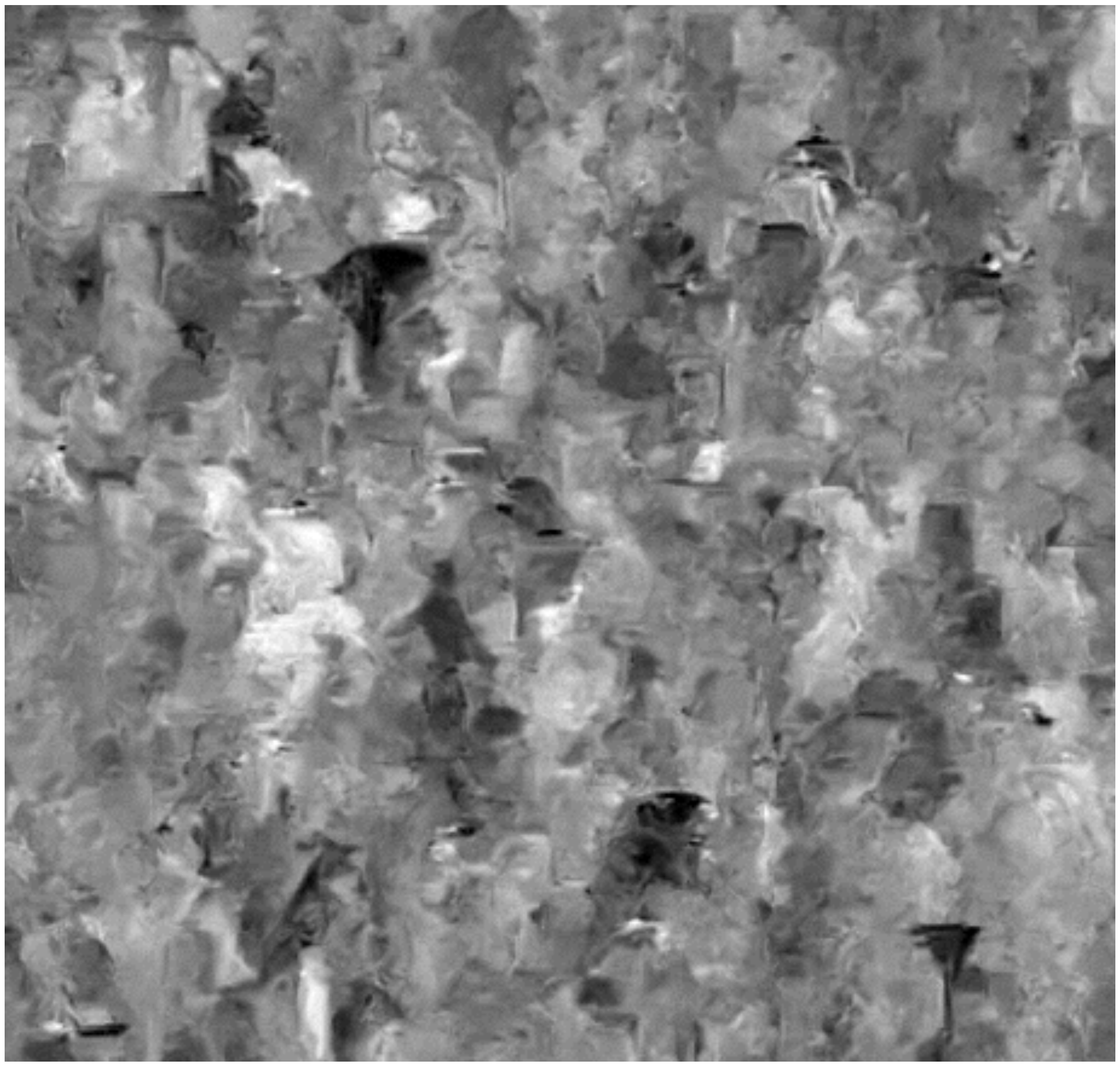




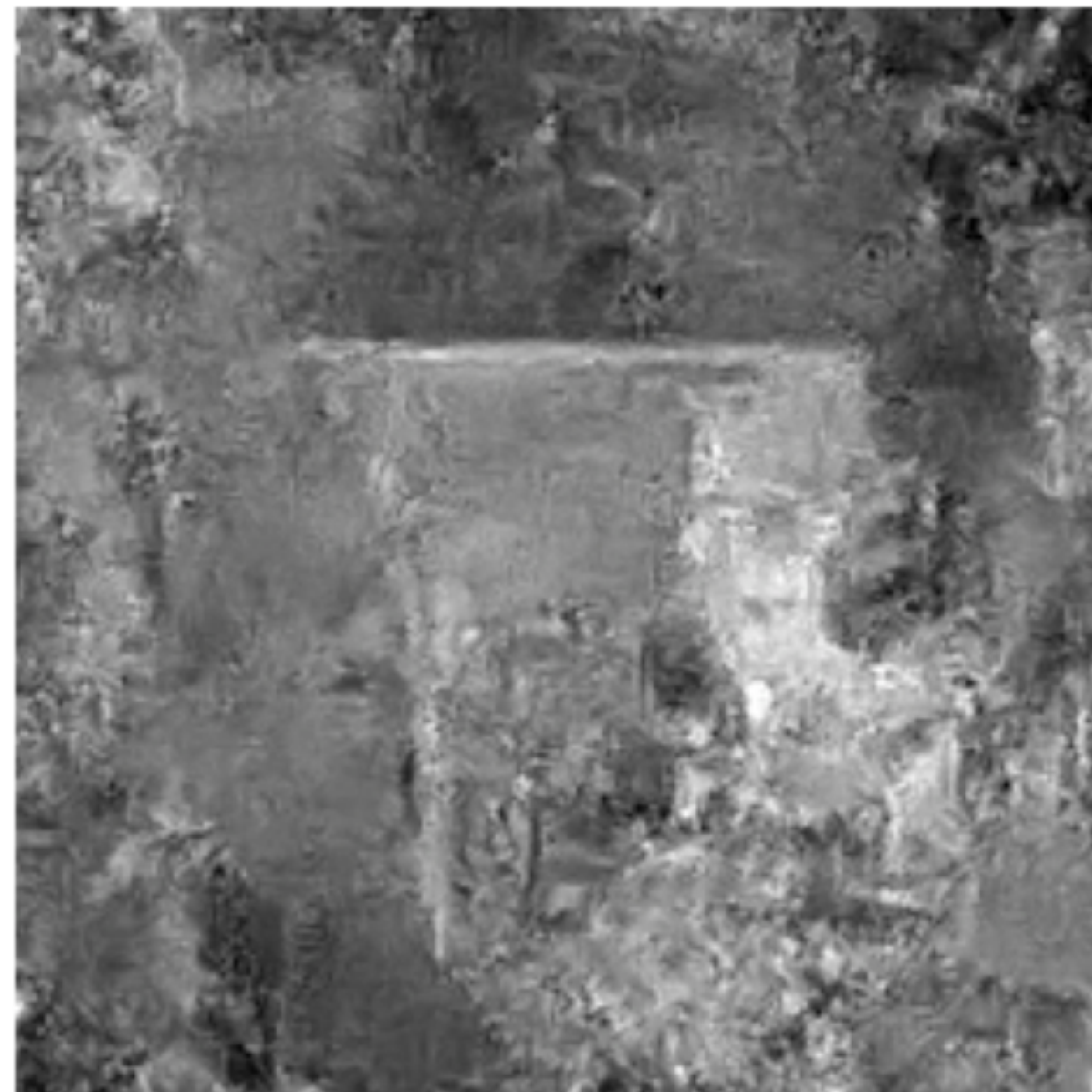
Theis et al (2012). PLoS ONE 7,7.

Samples from the MCGSM model

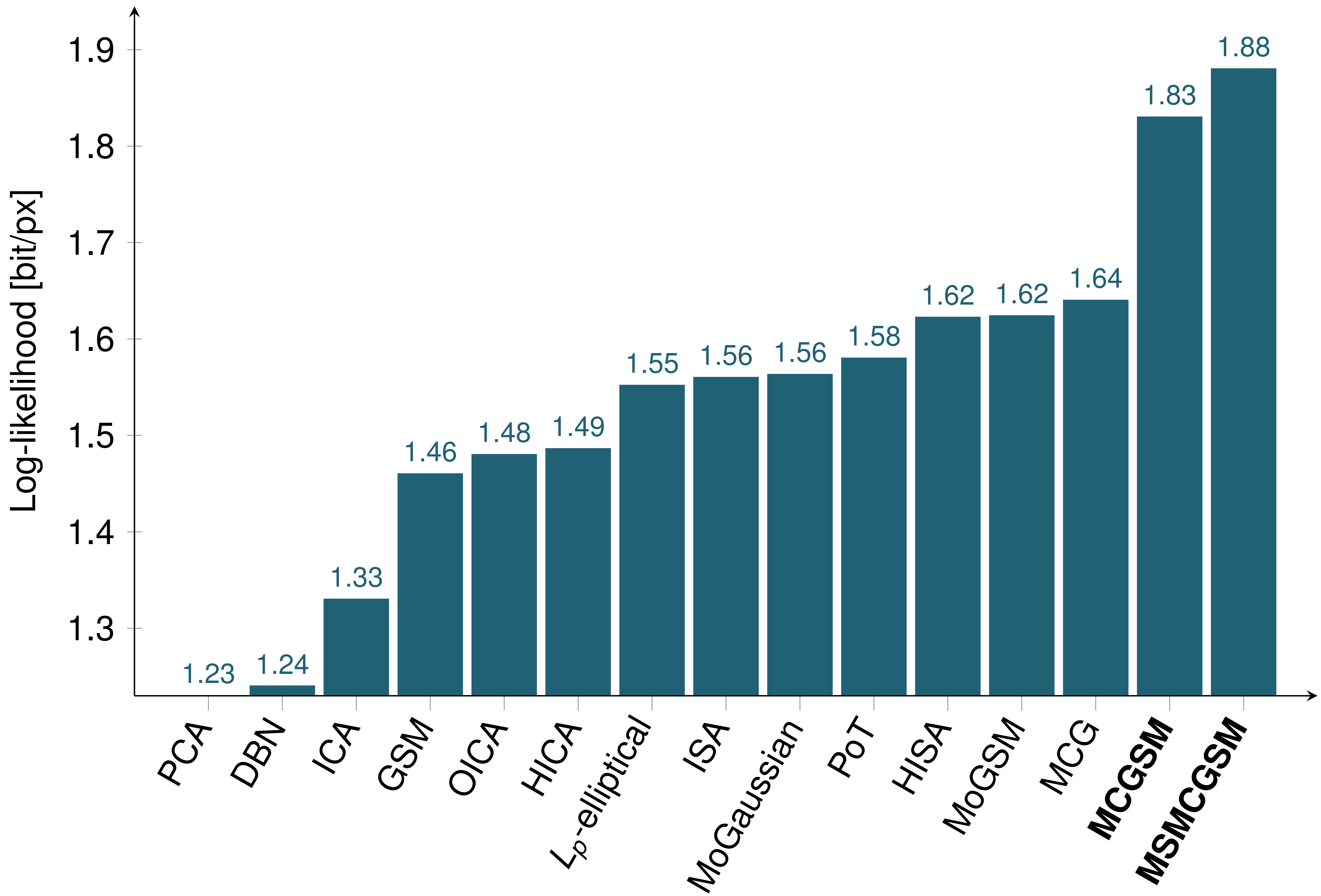
MCGSM



MCGSM + multi-scale



Model comparison



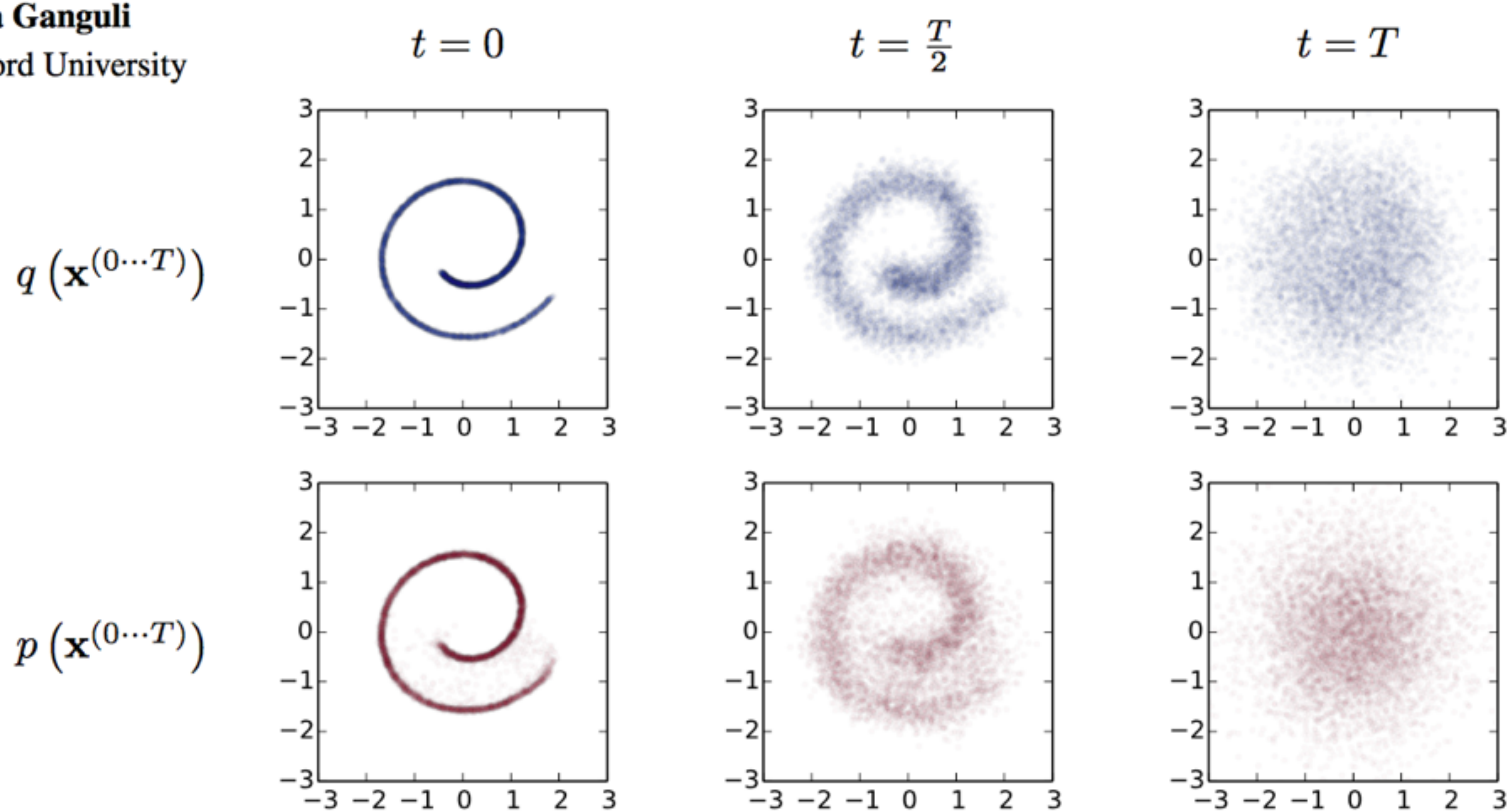
Jascha Sohl-Dickstein
Stanford University

Eric A. Weiss
University of California, Berkeley

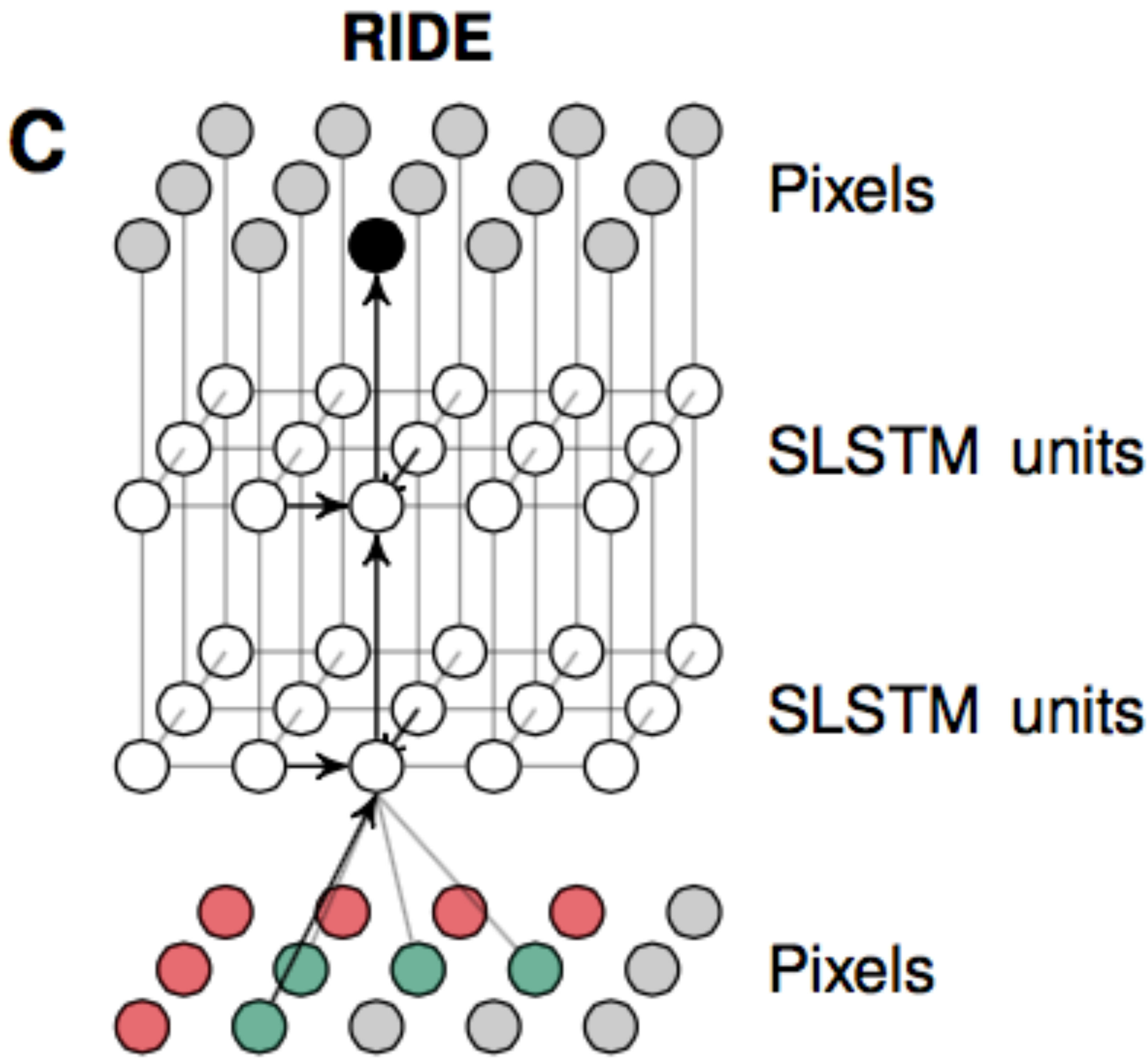
Niru Maheswaranathan
Stanford University

Surya Ganguli
Stanford University

Deep Unsupervised Learning using Nonequilibrium Thermodynamics



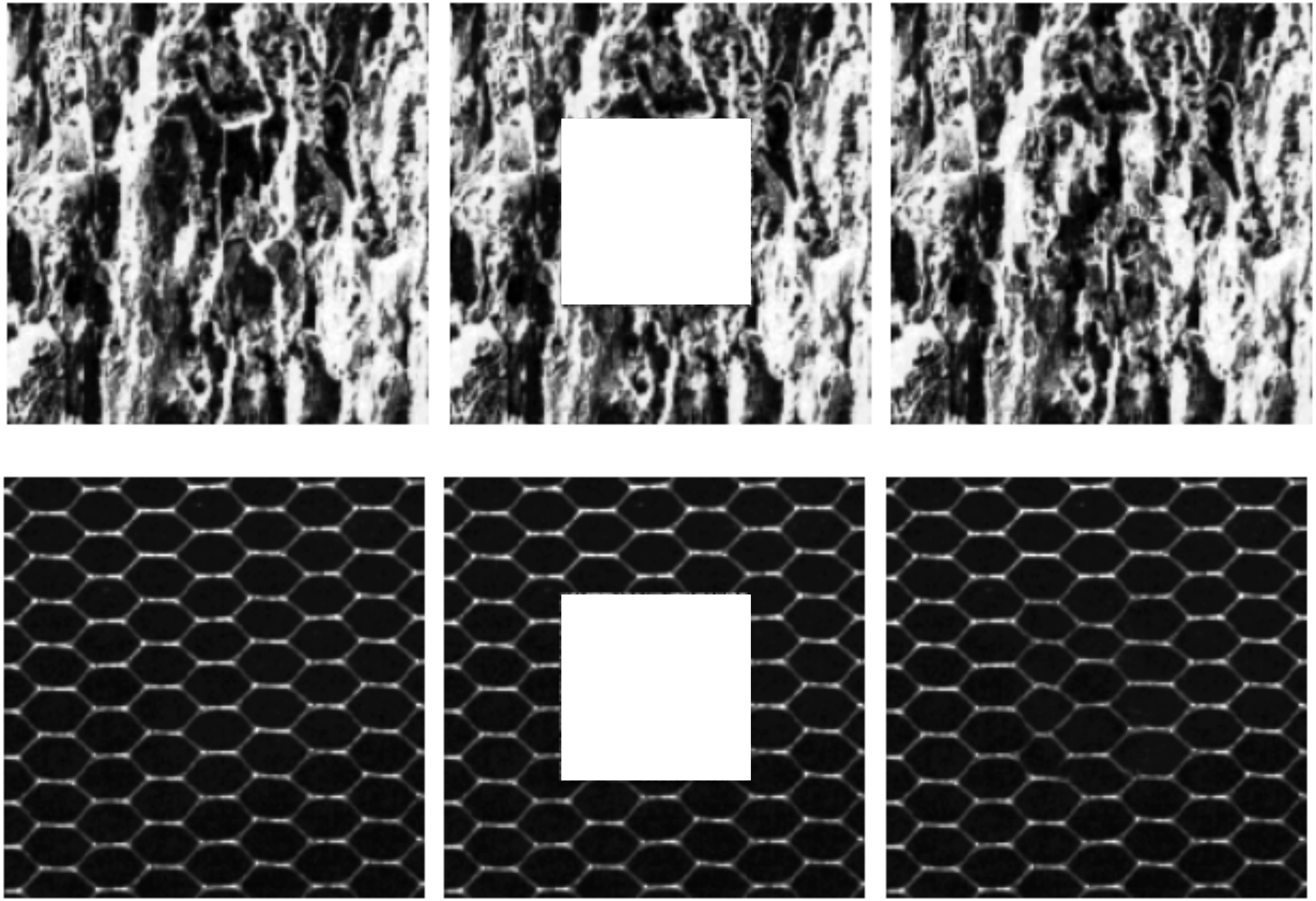
Generative Image Modeling Using Spatial LSTMs

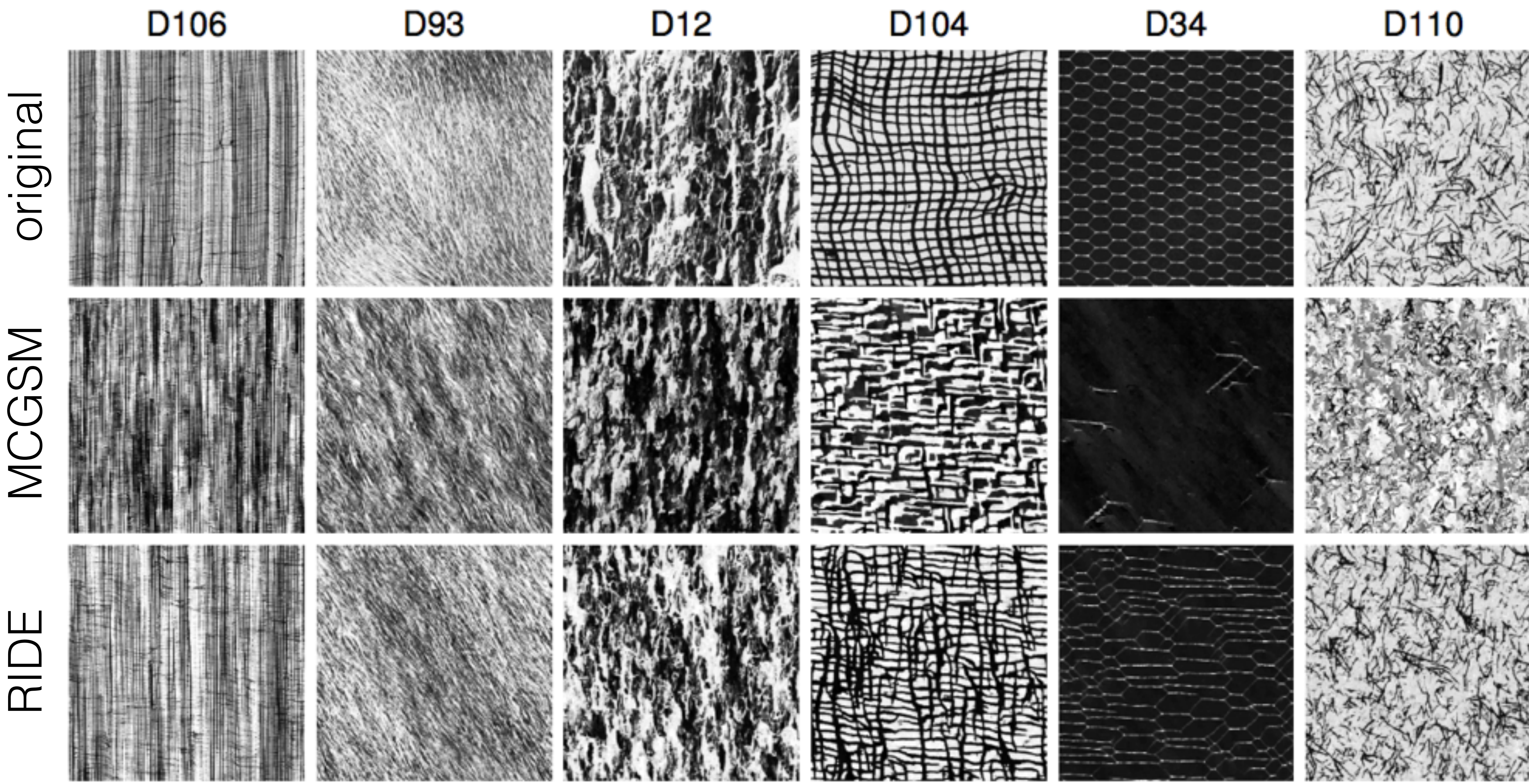


Model	63 dim. [nat]	64 dim. [bit/px]	∞ dim. [bit/px]
RNADE [41]	152.1	3.346	-
RNADE, 1 hl [42]	143.2	3.146	-
RNADE, 6 hl [42]	155.2	3.416	-
EoRNADE, 6 layers [42]	157.0	3.457	-
GMM, 200 comp. [44, 47]	153.7	3.360	-
STM, 200 comp. [43]	155.3	3.418	-
Deep GMM, 3 layers [44]	156.2	3.439	-
MCGSM, 16 comp.	155.1	3.413	3.688
MCGSM, 32 comp.	155.8	3.430	3.706
MCGSM, 64 comp.	156.2	3.439	3.716
MCGSM, 128 comp.	156.4	3.443	3.717
EoMCGSM, 128 comp.	**158.1**	**3.481**	3.748
RIDE, 1 layer	150.7	3.293	3.802
RIDE, 2 layers	152.1	3.346	3.869
EoRIDE, 2 layers	154.5	3.400	**3.899**

Model	256 dim. [bit/px]	∞ dim. [bit/px]
GRBM [11]	0.992	-
ICA [1, 45]	1.072	-
GSM	1.349	-
ISA [6, 14]	1.441	-
MoGSM, 32 comp. [37]	1.526	-
MCGSM, 32 comp.	1.615	1.759
RIDE, 1 layer, 64 hid.	1.650	1.816
RIDE, 1 layer, 128 hid.	-	1.830
RIDE, 2 layers, 64 hid.	-	1.829
RIDE, 2 layers, 128 hid.	-	**1.839**

Filling-in




D106
D93
D12
D104
D34
D110
original
MCGSM
RIDE

Feb 29, 2016: http://arxiv.org/pdf/1601.06759v2.pdf

Pixel Recurrent Neural Networks

Aäron van den Oord AVDNOORD@GOOGLE.COM
Nal Kalchbrenner NALK@GOOGLE.COM
Koray Kavukcuoglu KORAYK@GOOGLE.COM

Google DeepMind

Model	**NLL Test (Train)**
Uniform Distribution:	8.00
Multivariate Gaussian:	4.70
NICE [1]:	4.48
Deep Diffusion [2]:	4.20
Deep GMMs [3]:	4.00
RIDE [4]:	3.47
PixelCNN:	3.14 (3.08)
Row LSTM:	3.07 (3.00)
Diagonal BiLSTM:	**3.00** (2.93)

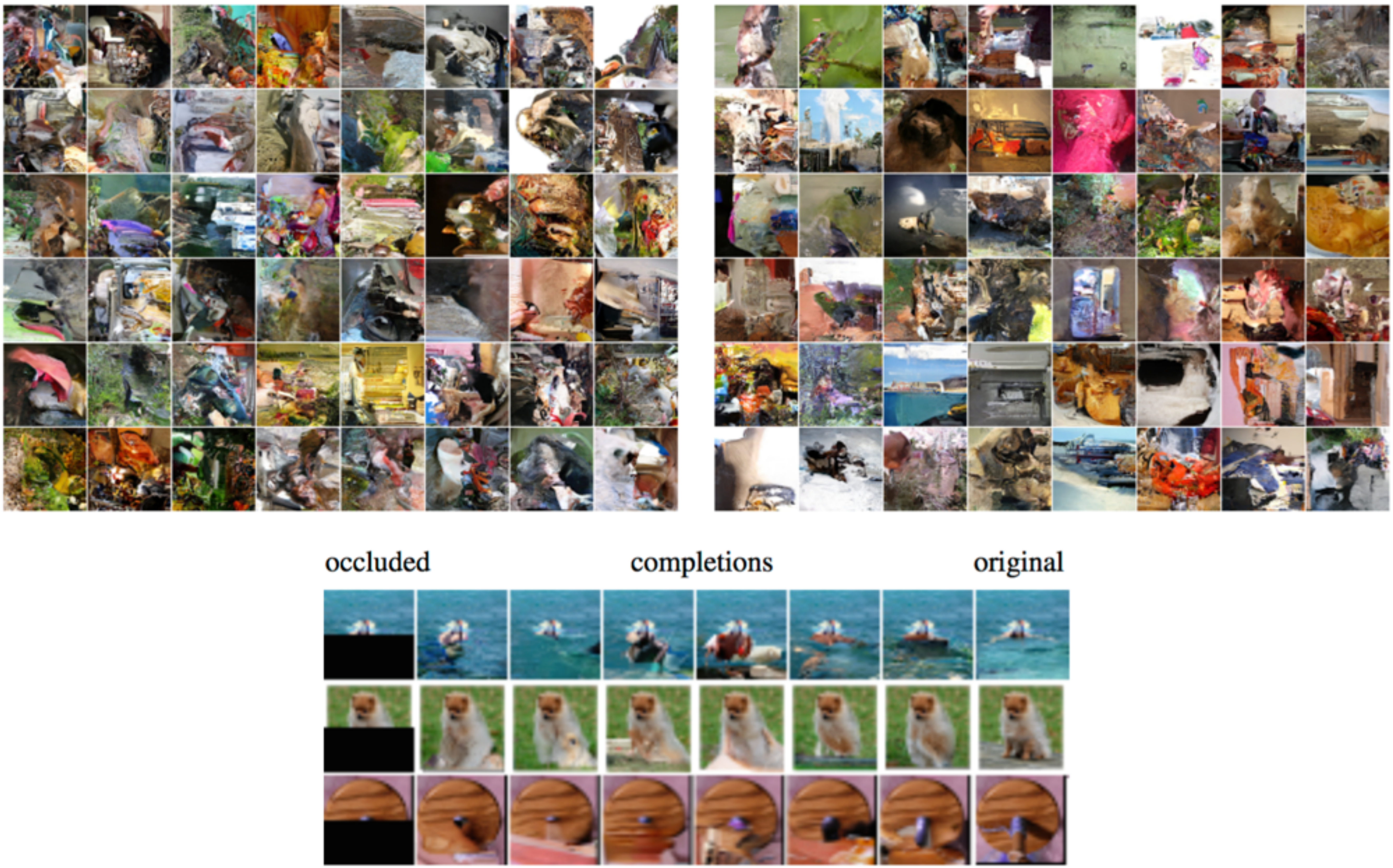
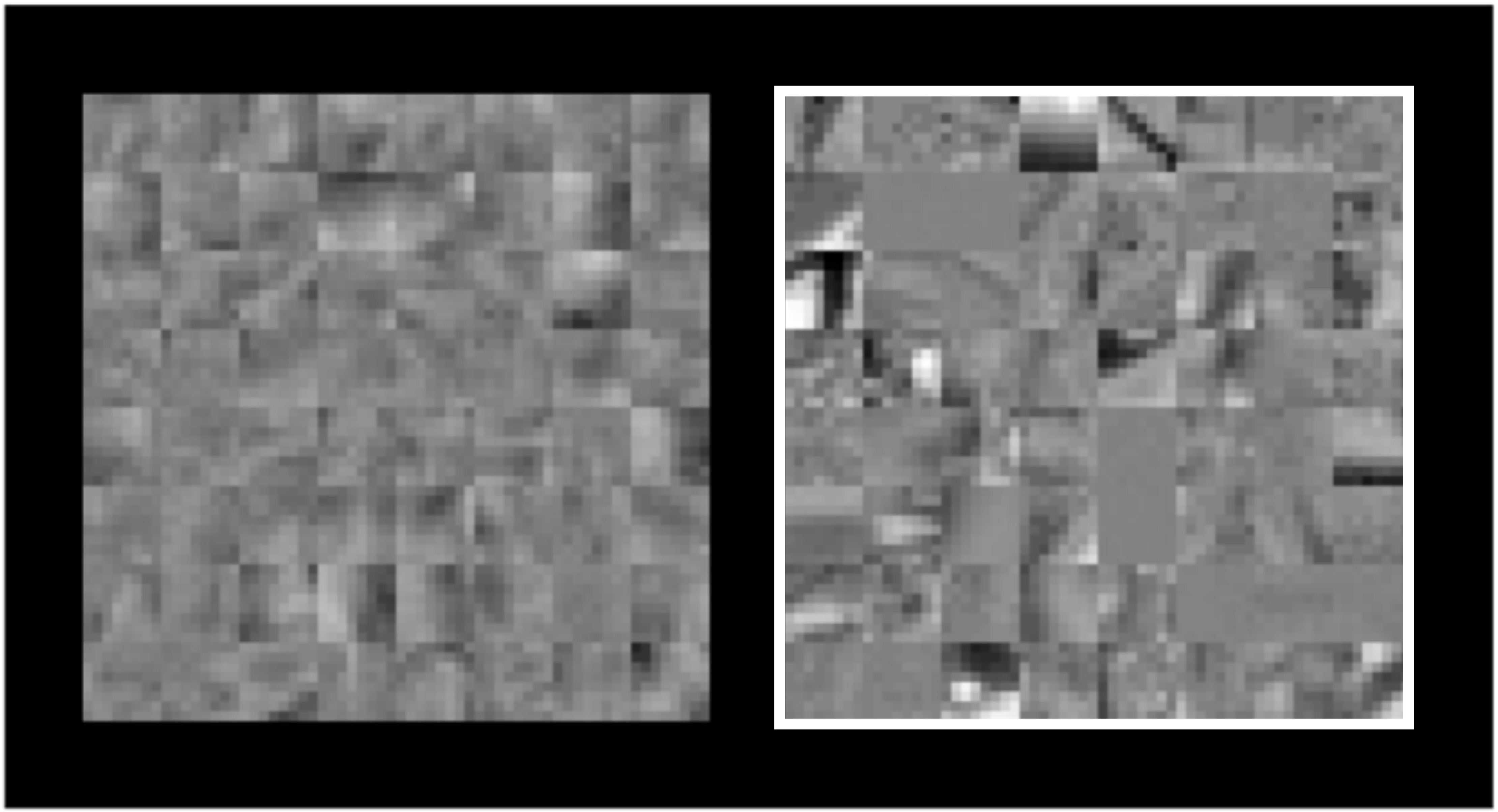


Figure 1. Image completions sampled from a PixelRNN.

Psychophysical Model comparison

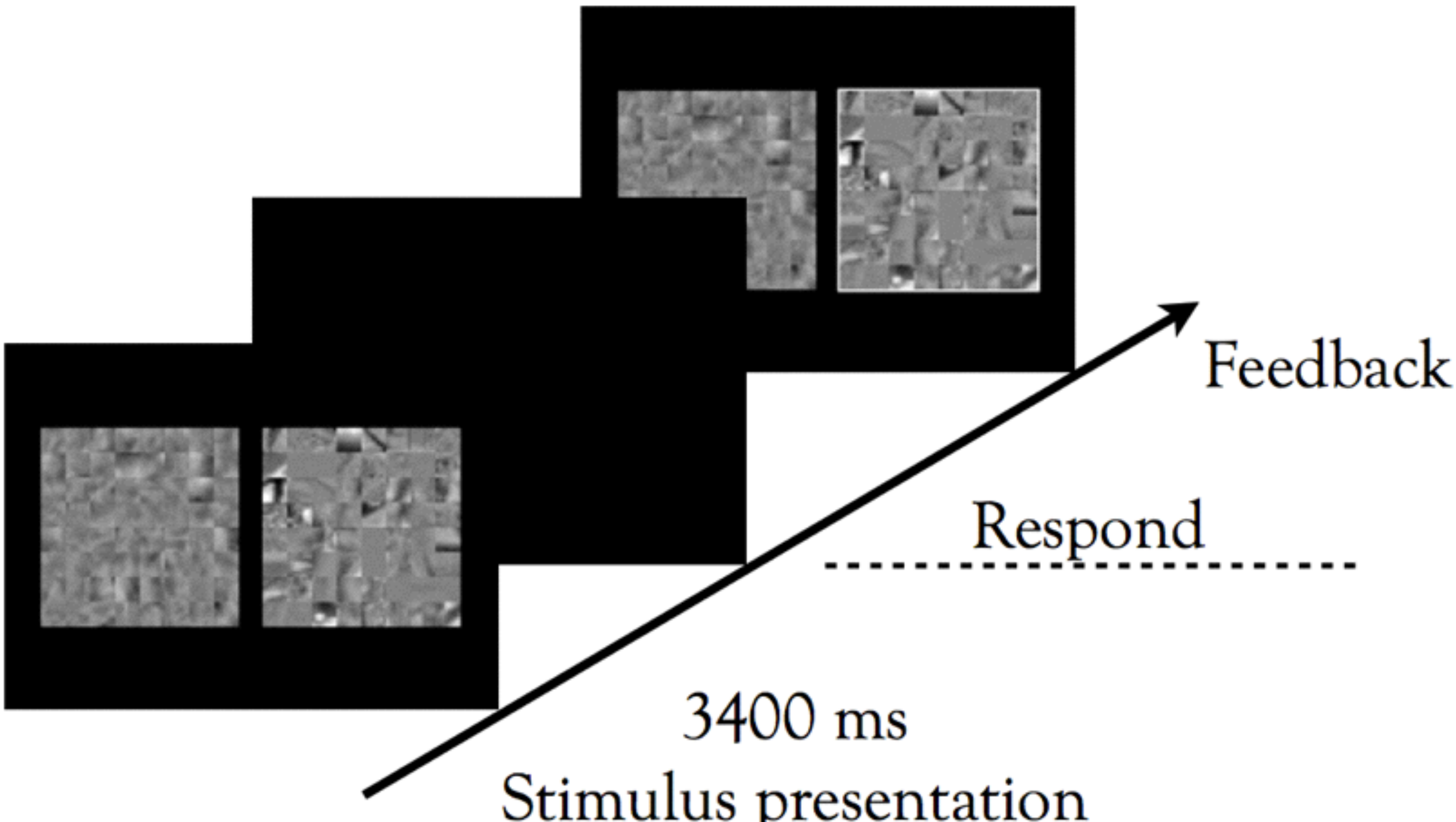
2AFC: Which contains natural samples?



(always 64 samples each)

[Gerhard et al, PLOS CB 2013]

Trial Time Course



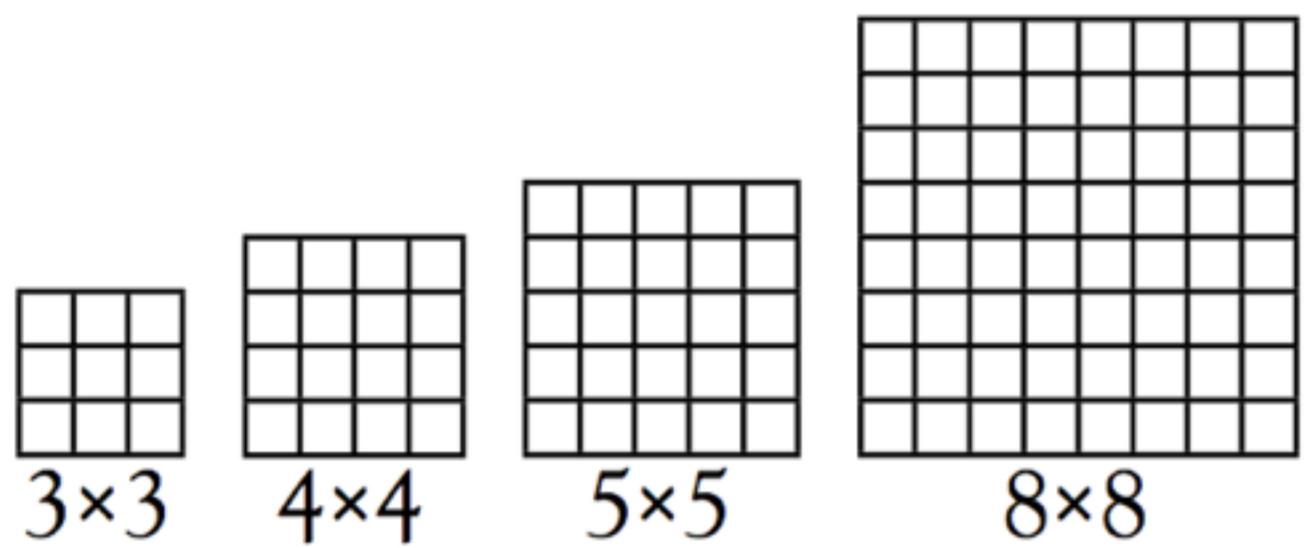
[Gerhard et al, PLOS CB 2013]

Design

- PCA, ICA, L_2, L_p — N=16
- **MEC** with k = 2, 4, 8, or 16 mixtures — N=12
- 30 trials / model / patch size

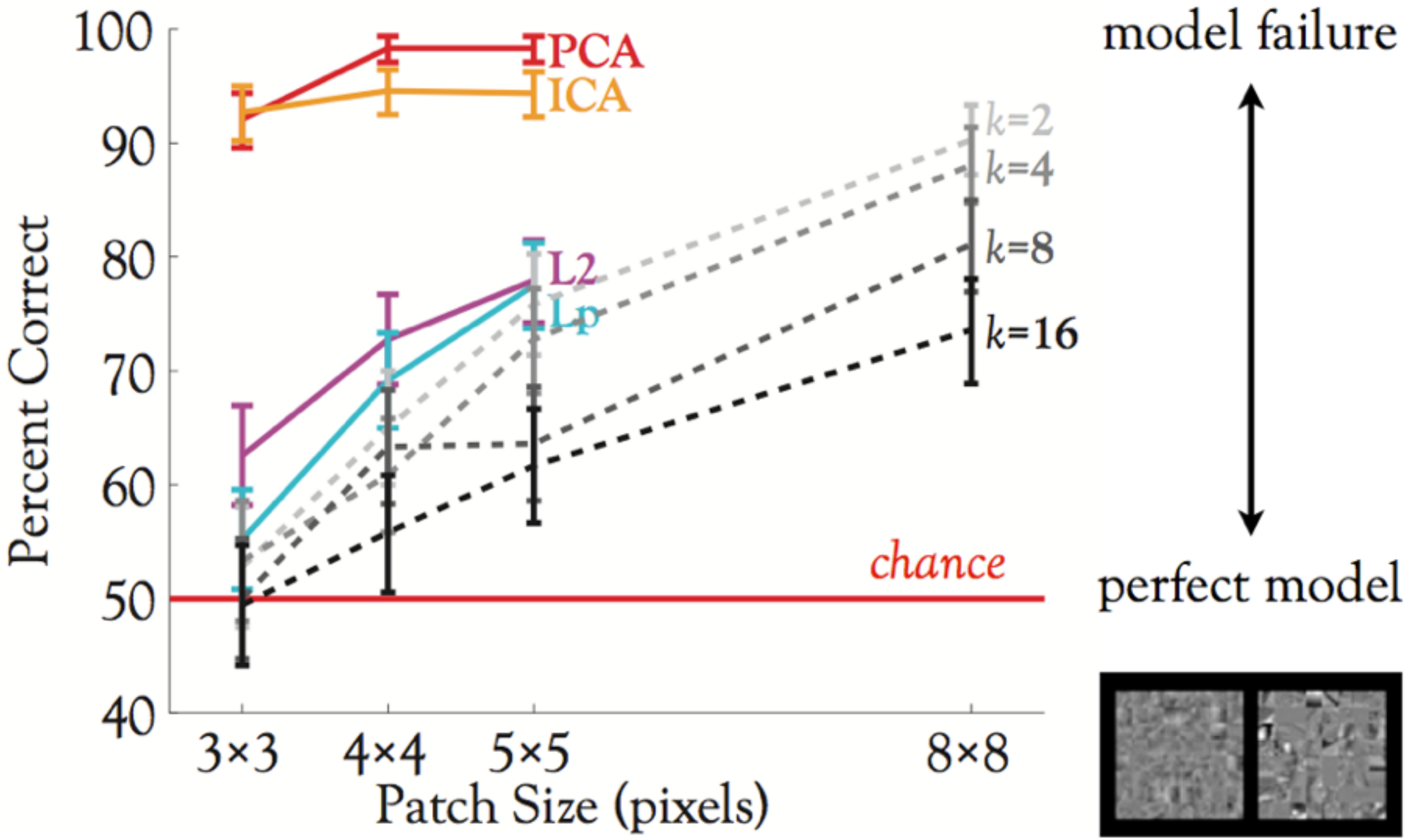
block pixel = 0.1375°

textures = 3.3° - 8.8°



[Gerhard et al, PLOS CB 2013]

Results



[Gerhard et al, PLOS CB 2013]

can we learn this?

task-invariant

feature space embedding

inductive bias

REVIEW

Unsupervised Learning

H.B. Barlow
Kenneth Craik Laboratory, Physiological Laboratory, Downing Street, Cambridge, CB2 3EG, England

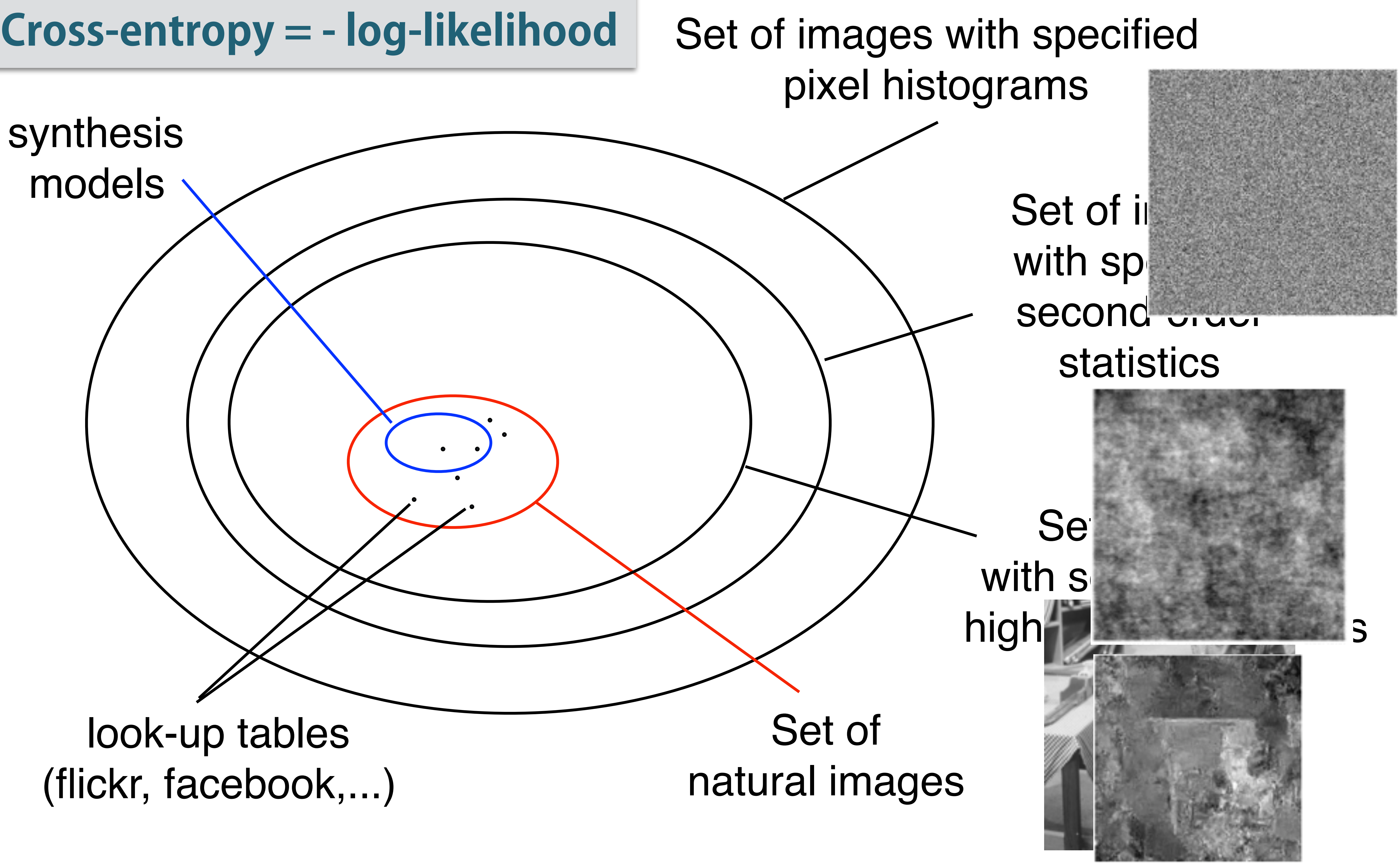
What use can the brain make of the massive flow tion that occurs without any associated rewards or

Main Idea:
generative image representation have much lower entropy than image pixels

Redundancy reduction
transform input into a minimum entropy representation

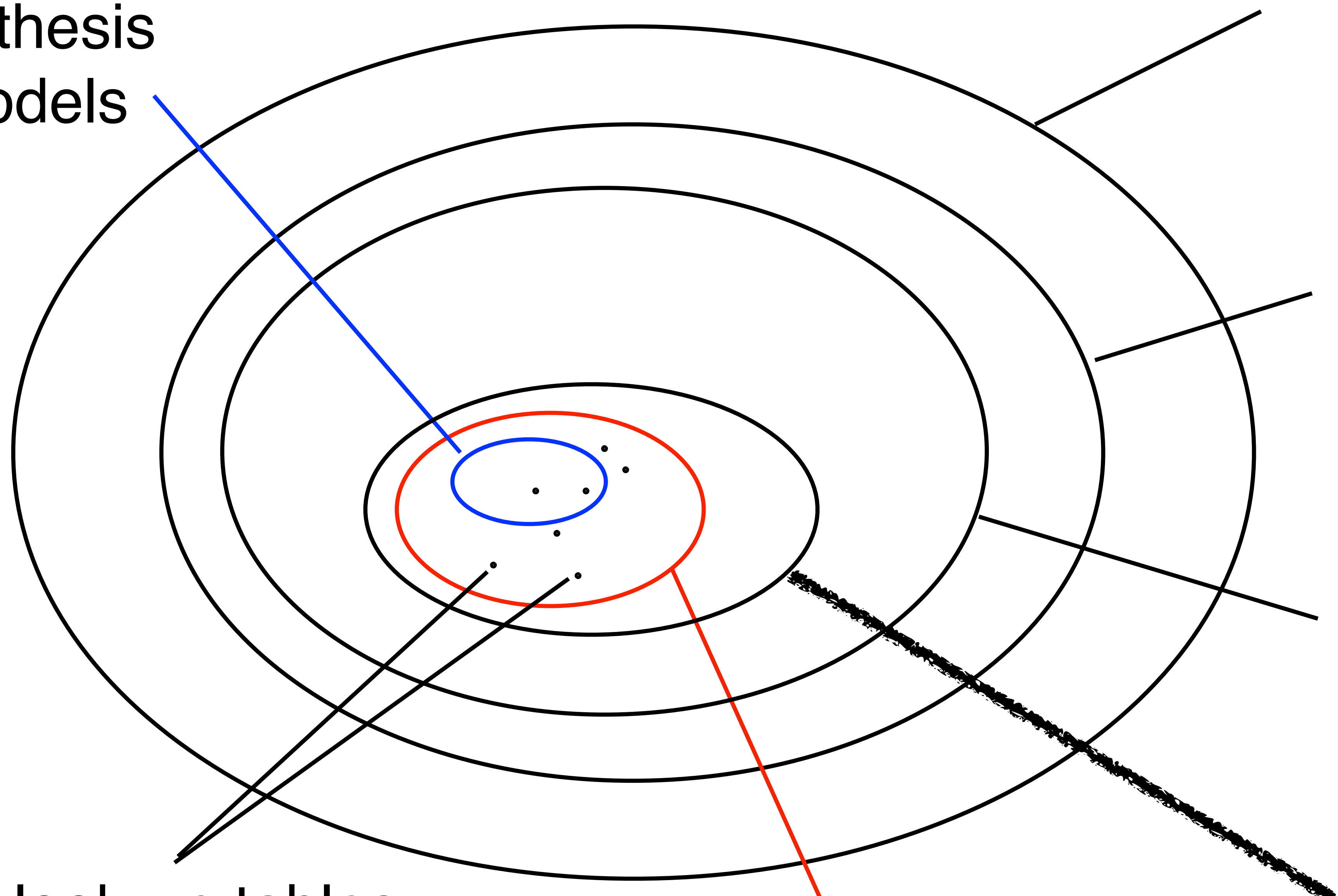
Minimax modeling/Onion peeling

Cross-entropy = - log-likelihood



Minimax modeling/Onion peeling

Cross-entropy = - log-likelihood



Feb 29, 2016: http://arxiv.org/pdf/1601.06759v2.pdf

Pixel Recurrent Neural Networks

Aäron van den Oord AVDNOORD@GOOGLE.COM
Nal Kalchbrenner NALK@GOOGLE.COM
Koray Kavukcuoglu KORAYK@GOOGLE.COM

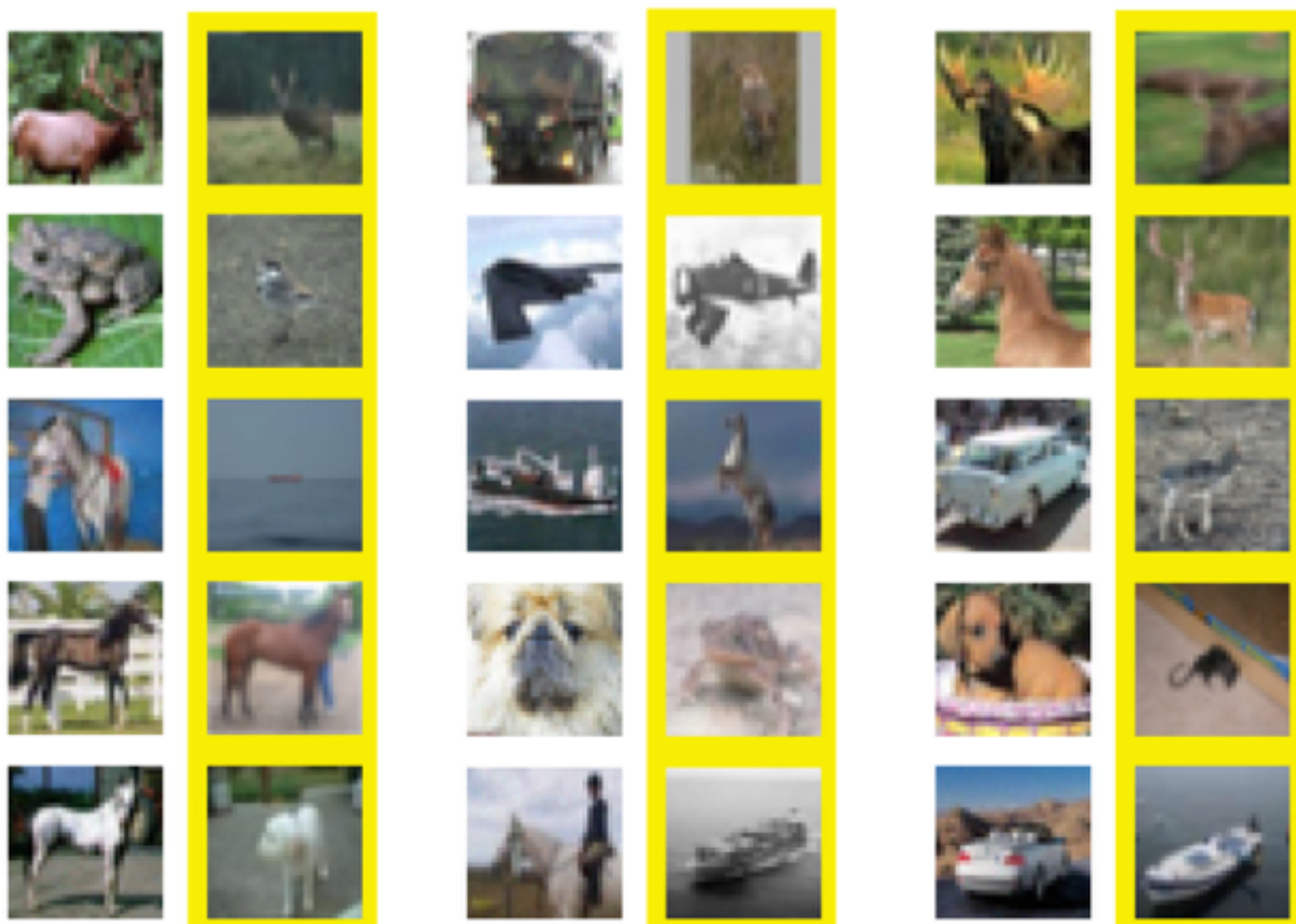
Google DeepMind

Model	**NLL Test (Train)**
Uniform Distribution:	8.00
Multivariate Gaussian:	4.70
NICE [1]:	4.48
Deep Diffusion [2]:	4.20
Deep GMMs [3]:	4.00
RIDE [4]:	3.47
PixelCNN:	3.14 (3.08)
Row LSTM:	3.07 (3.00)
Diagonal BiLSTM:	**3.00** (2.93)

Maximum entropy modeling vs synthesis models

A simple generative model with perfect samples:

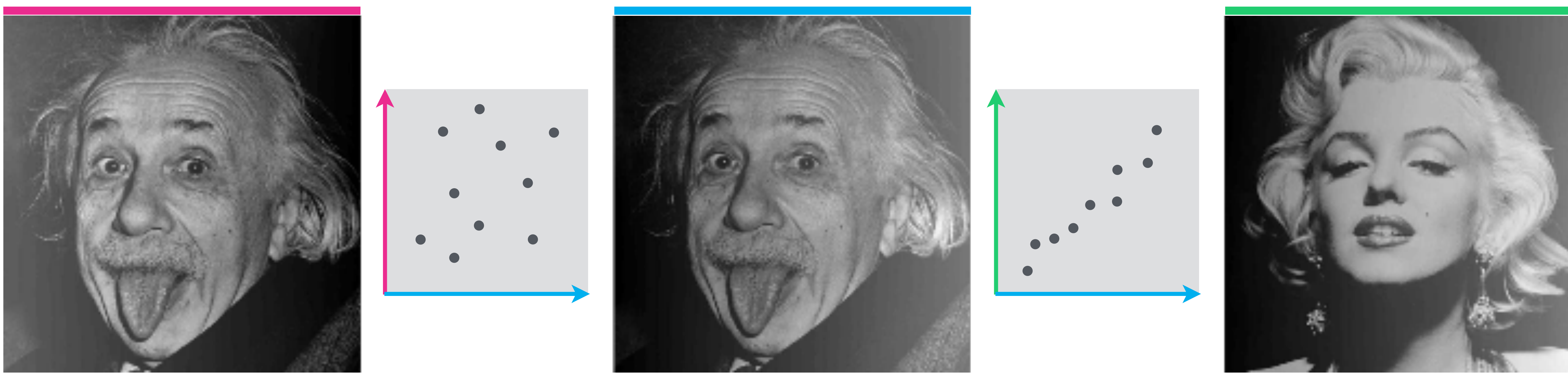
$$p(\mathbf{x}) = \frac{1}{N} \sum_n \mathcal{N}(\mathbf{x}; \sqrt{1-\alpha} T(\mathbf{x}_n), \alpha\sigma^2 \mathbf{I})$$



Maximum entropy modeling vs synthesis models

A simple generative model with perfect samples:

$$p(\mathbf{x}) = \frac{1}{N}\sum_n \mathcal{N}(\mathbf{x}; \sqrt{1-\alpha}\,T(\mathbf{x}_n), \alpha\sigma^2\mathbf{I})$$



Maximum entropy modeling vs synthesis models

The quality of synthetic images has no necessary implications for the likelihood:

$$\hat{p}_{publish}(\mathbf{x}) = \epsilon\,\hat{p}(\mathbf{x}) + (1-\epsilon)\,\hat{p}_{nicefy}(\mathbf{x})$$

$$\Rightarrow \hat{p}_{publish}(\mathbf{x}) \geq \underbrace{\log(\epsilon)}_{=\mathcal{O}(1)} + \underbrace{\log\hat{p}(\mathbf{x})}_{=\mathcal{O}(N)}$$

how much smaller is the model entropy than the data entropy?

Theis et al: A note on the evaluation of generative models. ICLR, 2016.

Image completion

van den Oord et al, http://arxiv.org/pdf/1601.06759v2.pdf

Literature

Gerhard, Theis, Bethge. Modeling Natural Image Statistics. Biologically-inspired Computer Vision—Fundamentals and Applications, Wiley VCH, 2015.

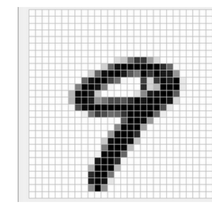
Theis & Bethge. Generative Image Modeling Using Spatial LSTMs. Advances in Neural Information Processing Systems 28, June 2015.

Theis, A. van den Oord, Bethge. A note on the evaluation of generative models. International Conference on Learning Representations, 2016.

van den Oord, Kalchbrenner, Kavukcuoglu. Pixel recurrent neural networks, http://arxiv.org/pdf/1601.06759v2.pdf arXiv, 2016.

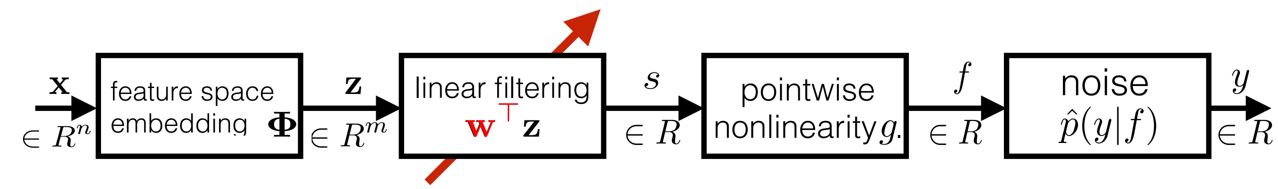
Wrap-up

- Curse of dimensionality



10^{75} TB

- NLN cascade regression



$$\hat{p}(y|\mathbf{x}) = \hat{p}(y|g(s(\mathbf{x}))) \quad s(\mathbf{x}) = \mathbf{w}^\top \mathbf{\Phi}(\mathbf{x})$$

- going beyond NLN regression (spike prediction):

$$s(\mathbf{x}) = \log \sum_{k=1}^{K} \exp\left(\beta_{k1}(\mathbf{u}_{k1}^\top \mathbf{x})^2 + \cdots + \beta_{kM}(\mathbf{u}_{kM}^\top \mathbf{x})^2 + \mathbf{w}_k^\top \mathbf{x} + \mathbf{b}_k\right)$$

- Natural image statistics

- Minimax modeling

$$\hat{p}_{publish}(\mathbf{x}) = \epsilon\, \hat{p}$$

Representations in brains and machines

Part II

Matthias Bethge

MLSS 2016 Cadiz

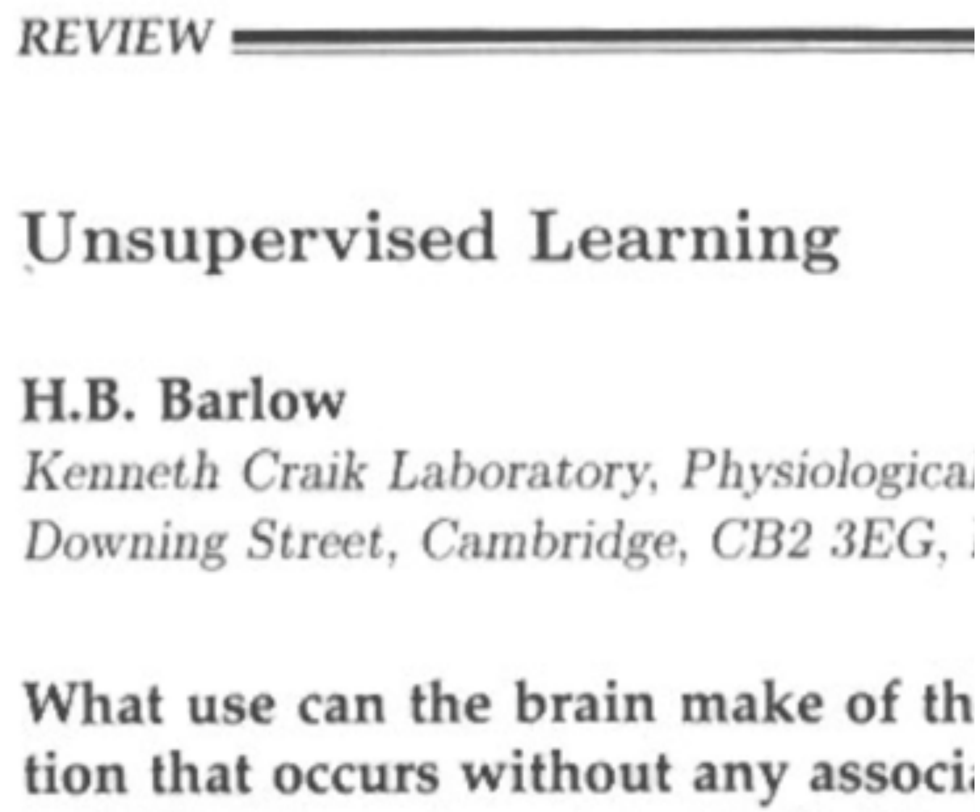
http://bethgelab.org

can we learn this?

task-invariant

feature space embedding

inductive bias


REVIEW

Unsupervised Learning

H.B. Barlow

Kenneth Craik Laboratory, Physiologica

Downing Street, Cambridge, CB2 3EG,

What use can the brain make of th

tion that occurs without any associ

Task-invariant features from supervised deep learning

task-invariant

feature space embedding

inductive bias

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. Technical Report HAL-00911179, INRIA, 2013.

A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features off-the-shelf: an Astounding Baseline for Recognition," *CoRR*, vol. abs/1403.6382, 2014.

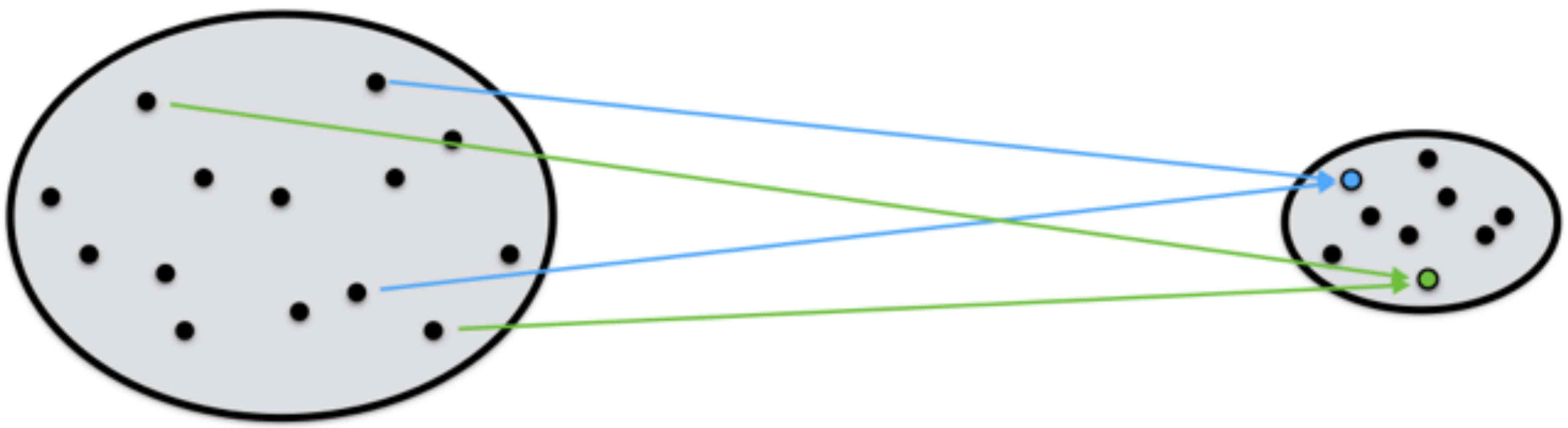
K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

Supervised representation learning

IMAGENET benchmark

input **output**



1 Mio training images **1000 categories**

flamingo

rooster

ruffed grouse

quail

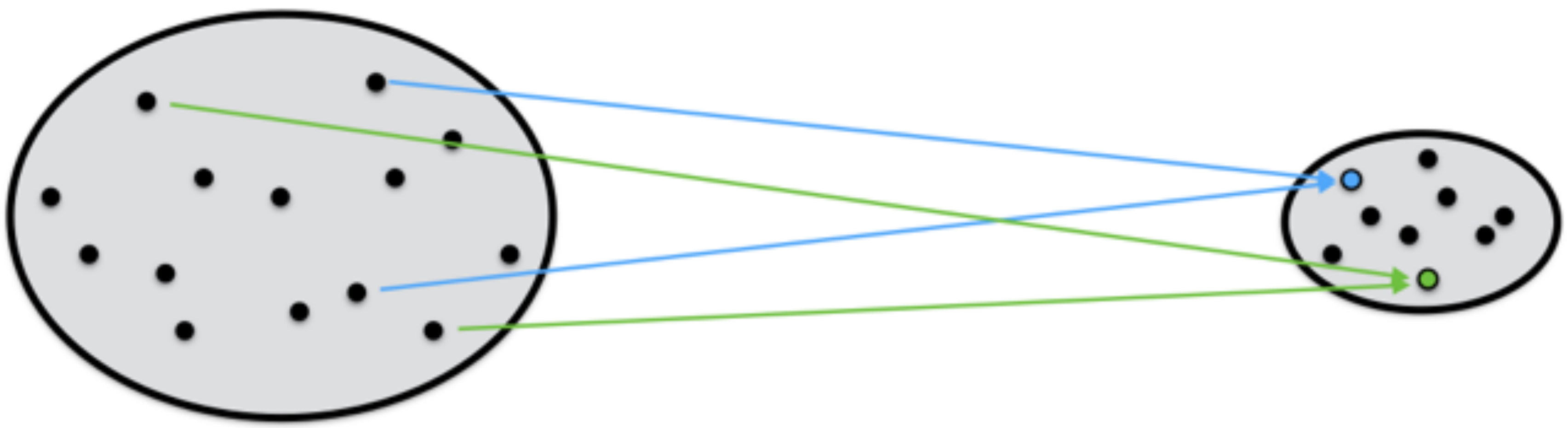
partridge

Supervised representation learning

IMAGENET benchmark

input

output



1 Mio training images

1000 categories

Egyptian cat

Persian cat

Siamese cat

tabby

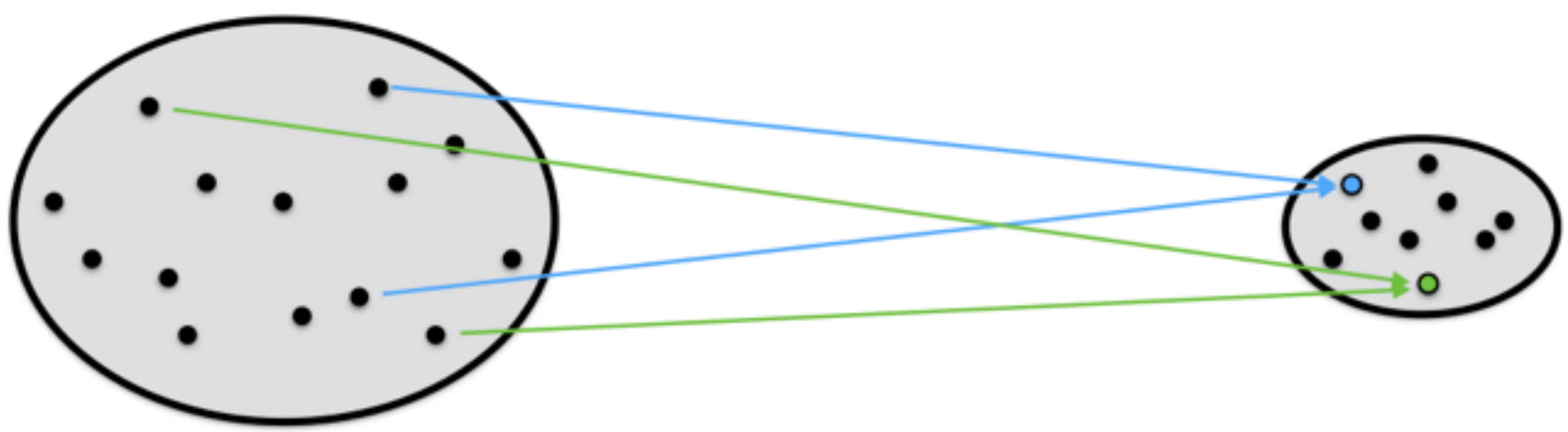
lynx

Supervised representation learning

IMAGENET benchmark

input

output



1 Mio training images

1000 categories

dalmatian

keeshond

miniature schnauzer

standard schnauzer

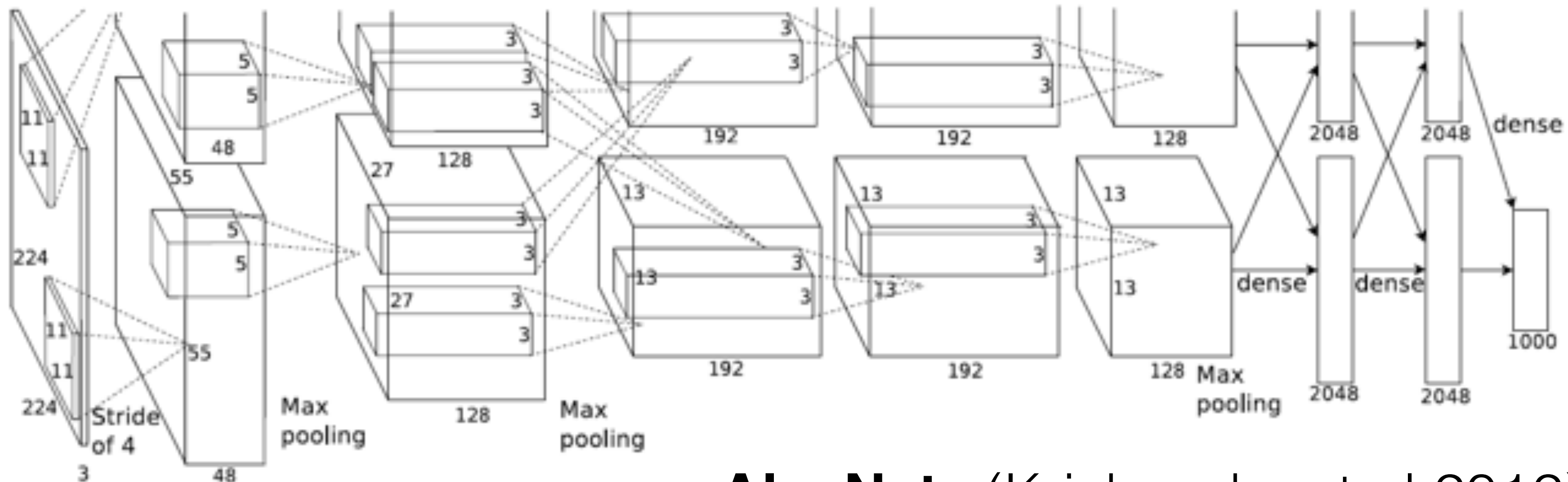
giant schnauzer

Supervised representation learning

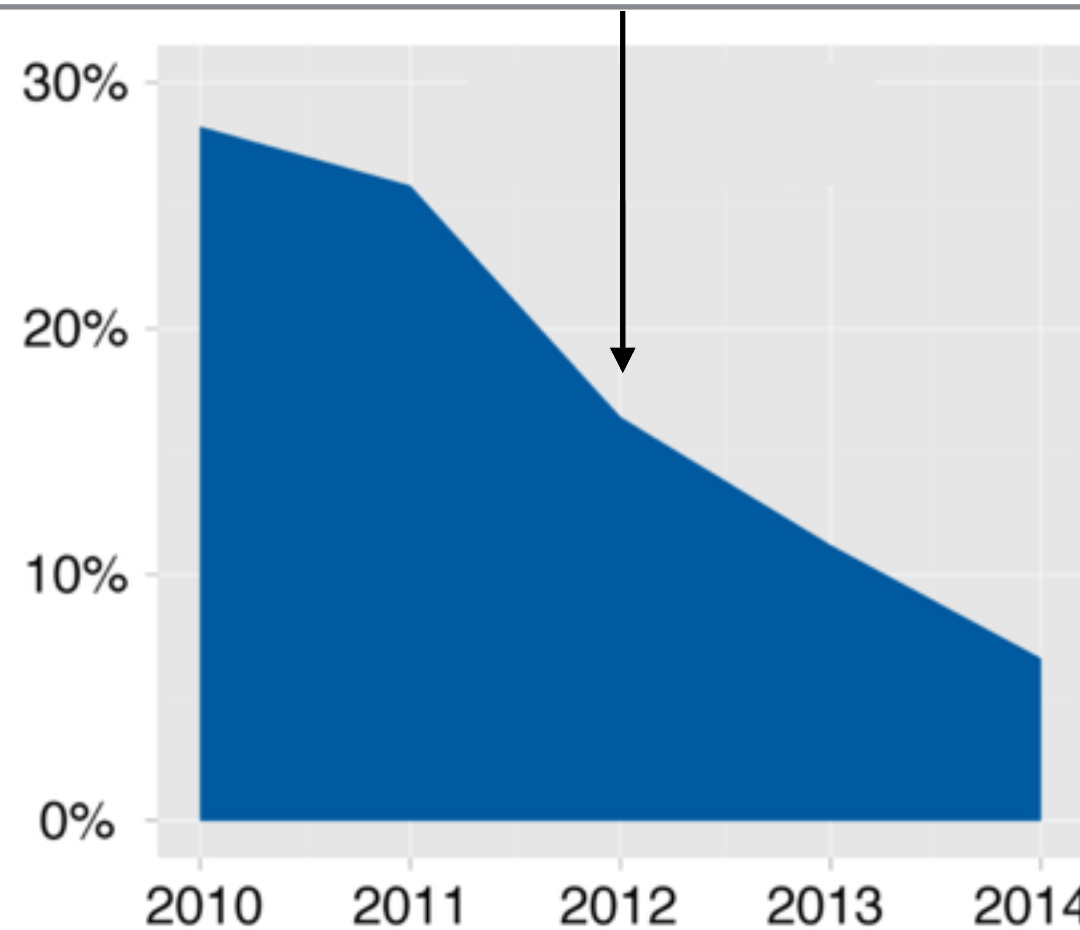




IMAGENET benchmark



AlexNet (Krizhevsky et al 2012)



dalmatian

keeshond

miniature schnauzer

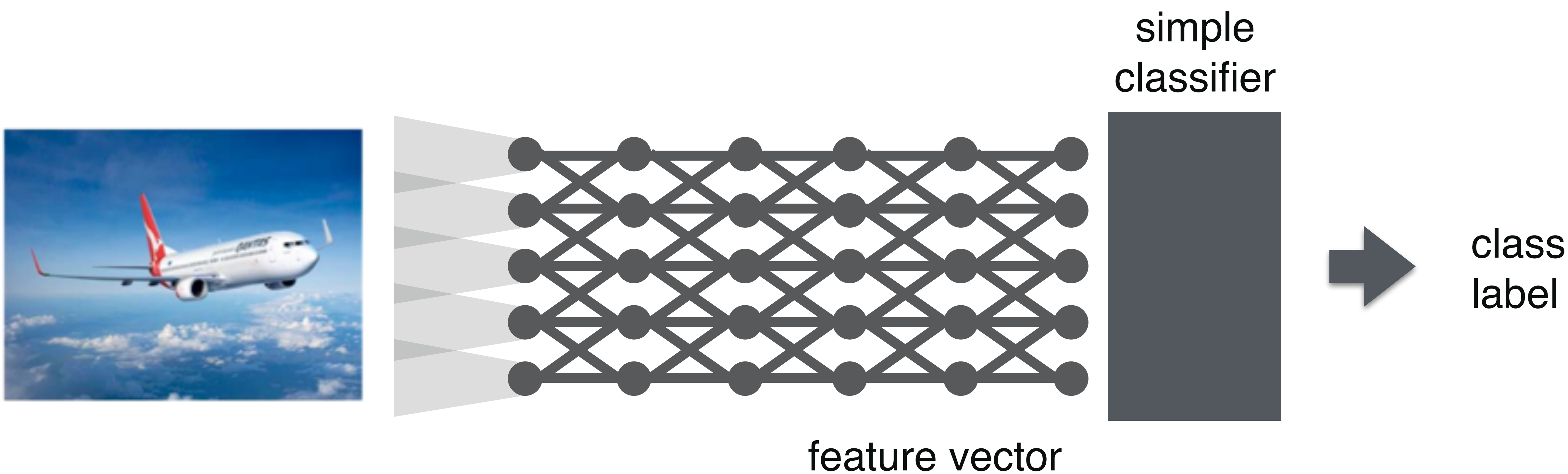
standard schnauzer

giant schnauzer

Unbiased look at dataset bias (Torralba & Efros 2011)

task	Train on: \ Test on:	SUN09	LabelMe	PASCAL	ImageNet	Caltech101	MSRC	Self	Mean others	Percent drop
"car" classification	SUN09	**28.2**	29.5	16.3	14.6	16.9	21.9	28.2	19.8	**30%**
	LabelMe	14.7	**34.0**	16.7	22.9	43.6	24.5	34.0	24.5	**28%**
	PASCAL	10.1	25.5	**35.2**	43.9	44.2	39.4	35.2	32.6	**7%**
	ImageNet	11.4	29.6	36.0	**57.4**	52.3	42.7	57.4	34.4	**40%**
	Caltech101	7.5	31.1	19.5	33.1	**96.9**	42.1	96.9	26.7	**73%**
	MSRC	9.3	27.0	24.9	32.6	40.3	**68.4**	68.4	26.8	**61%**
	Mean others	10.6	28.5	22.7	29.4	39.4	34.1	53.4	27.5	48%
"car" detection	SUN09	**69.8**	50.7	42.2	42.6	54.7	69.4	69.8	51.9	**26%**
	LabelMe	61.8	**67.6**	40.8	38.5	53.4	67.0	67.6	52.3	**23%**
	PASCAL	55.8	55.2	**62.1**	56.8	54.2	74.8	62.1	59.4	**4%**
	ImageNet	43.9	31.8	46.9	**60.7**	59.3	67.8	60.7	49.9	**18%**
	Caltech101	20.2	18.8	11.0	31.4	**100**	29.3	100	22.2	**78%**
	MSRC	28.6	17.1	32.3	21.5	67.7	**74.3**	74.3	33.4	**55%**
	Mean others	42.0	34.7	34.6	38.2	57.9	61.7	72.4	44.8	**48%**
"person" classification	SUN09	**16.1**	11.8	14.0	7.9	6.8	23.5	16.1	12.8	**20%**
	LabelMe	11.0	**26.6**	7.5	6.3	8.4	24.3	26.6	11.5	**57%**
	PASCAL	11.9	11.1	**20.7**	13.6	48.3	50.5	20.7	27.1	**-31%**
	ImageNet	8.9	11.1	11.8	**20.7**	76.7	61.0	20.7	33.9	**-63%**
	Caltech101	7.6	11.8	17.3	22.5	**99.6**	65.8	99.6	25.0	**75%**
	MSRC	9.4	15.5	15.3	15.3	93.4	**78.4**	78.4	29.8	**62%**
	Mean others	9.8	12.3	13.2	13.1	46.7	45.0	43.7	23.4	**47%**
"person" detection	SUN09	**69.6**	56.8	37.9	45.7	52.1	72.7	69.6	53.0	**24%**
	LabelMe	58.9	**66.6**	38.4	43.1	57.9	68.9	66.6	53.4	**20%**
	PASCAL	56.0	55.6	**56.3**	55.6	56.8	74.8	56.3	59.8	**-6%**
	ImageNet	48.8	39.0	40.1	**59.6**	53.2	70.7	59.6	50.4	**15%**
	Caltech101	24.6	18.1	12.4	26.6	**100**	31.6	100	22.7	**77%**
	MSRC	33.8	18.2	30.9	20.8	69.5	**74.7**	74.7	34.6	**54%**
	Mean others	44.4	37.5	31.9	38.4	57.9	63.7	71.1	45.6	**36%**

Deep CNN transfer learning



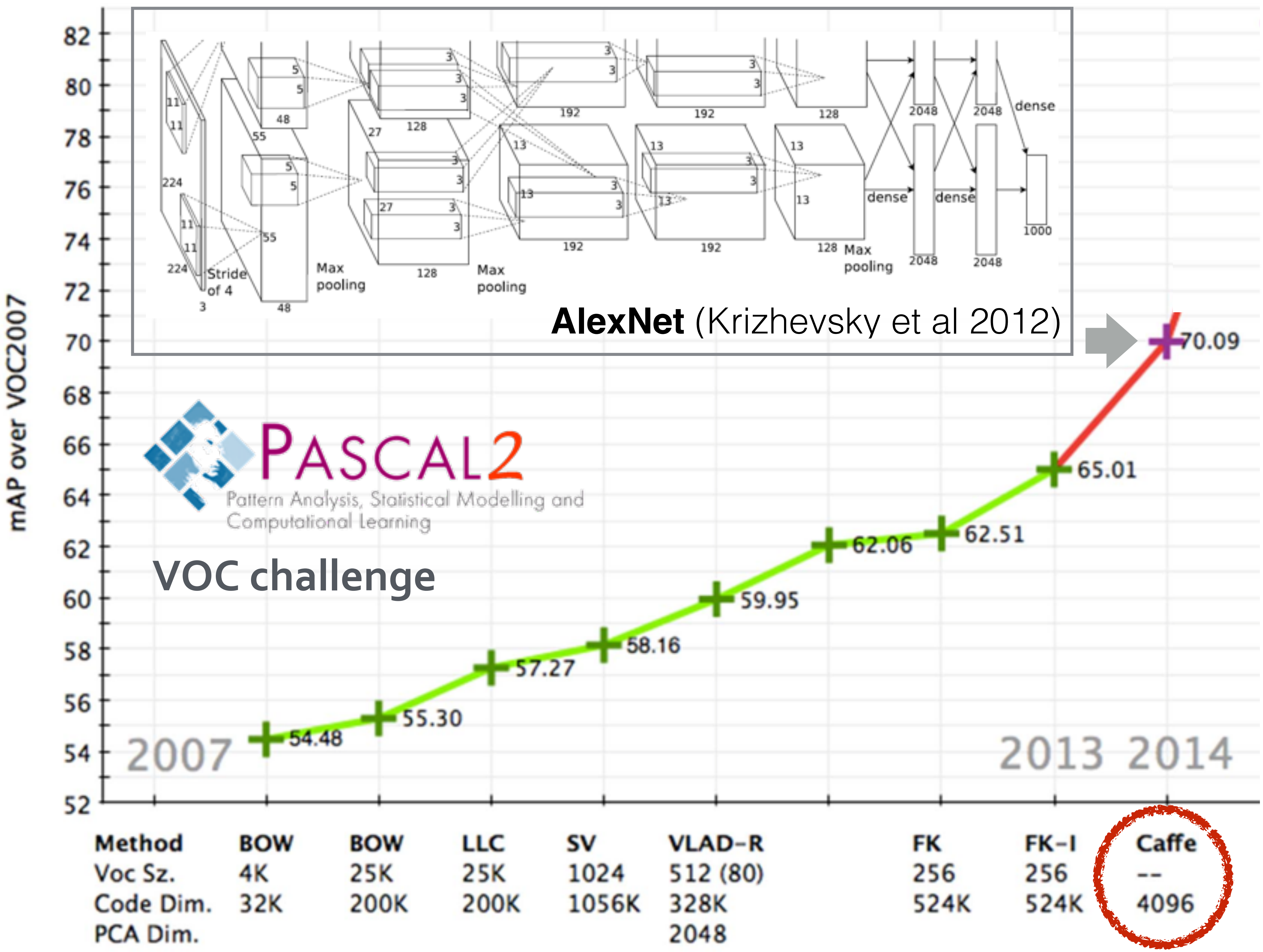
IMAGENET

1000 categories
1000 images for each

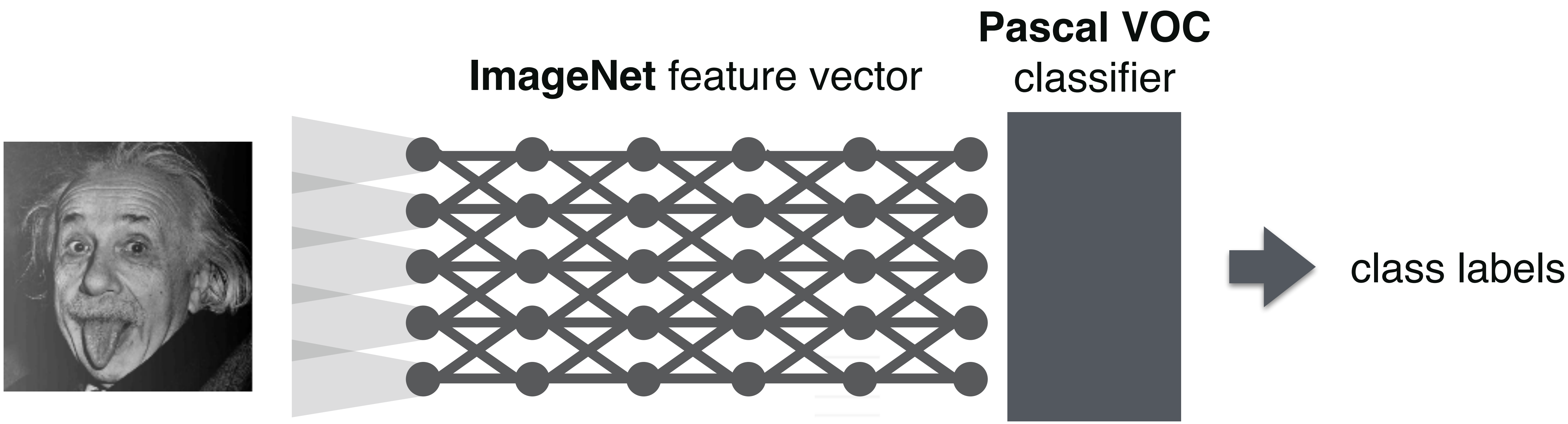
Transfer learning:

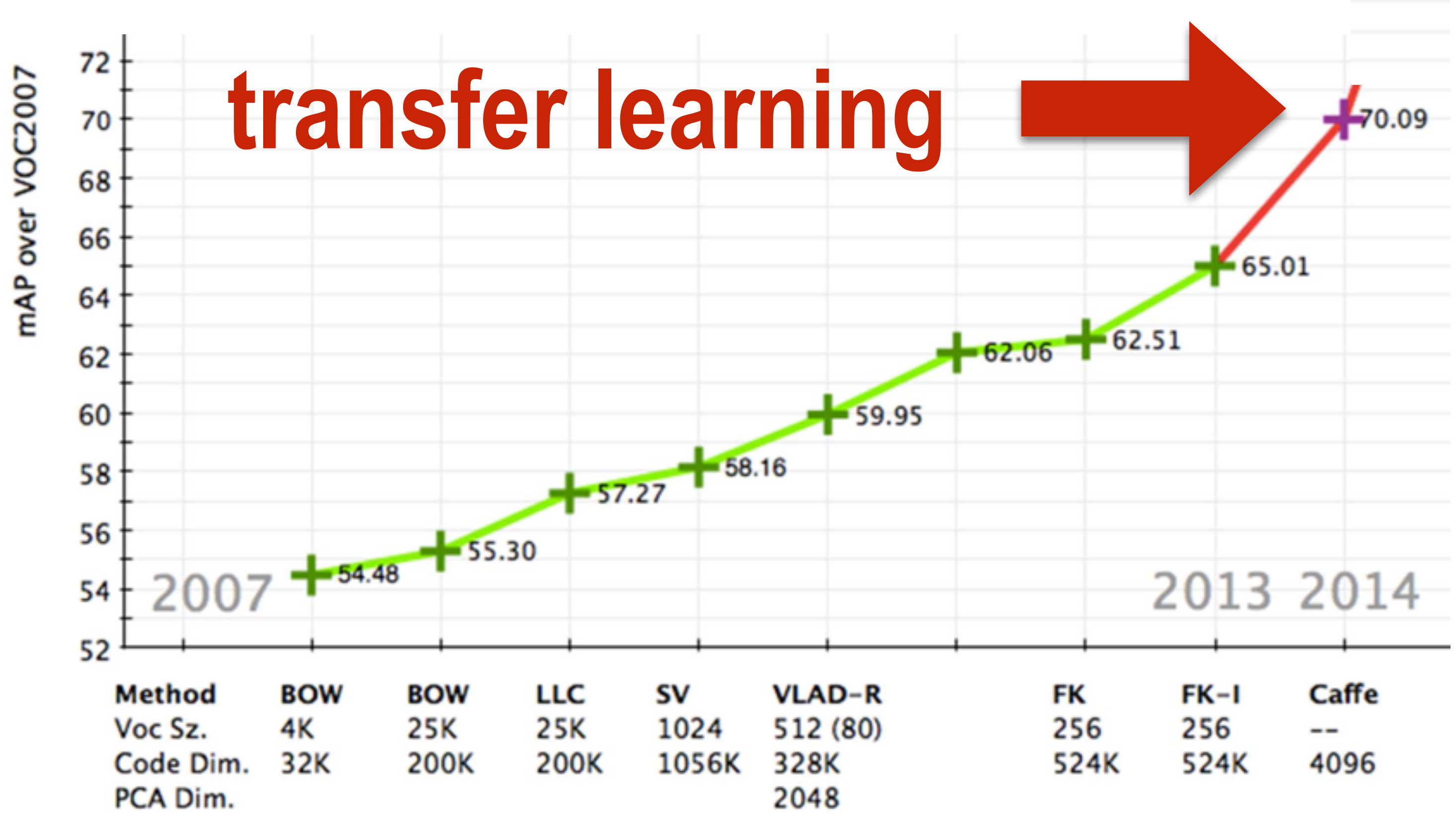
1. Train convolutional neural network (CNN) to classify images into one of 1000 classes.
2. Once trained, use the feature vector for other tasks.
3. How useful is the feature vector for vision?

Deep CNN transfer learning

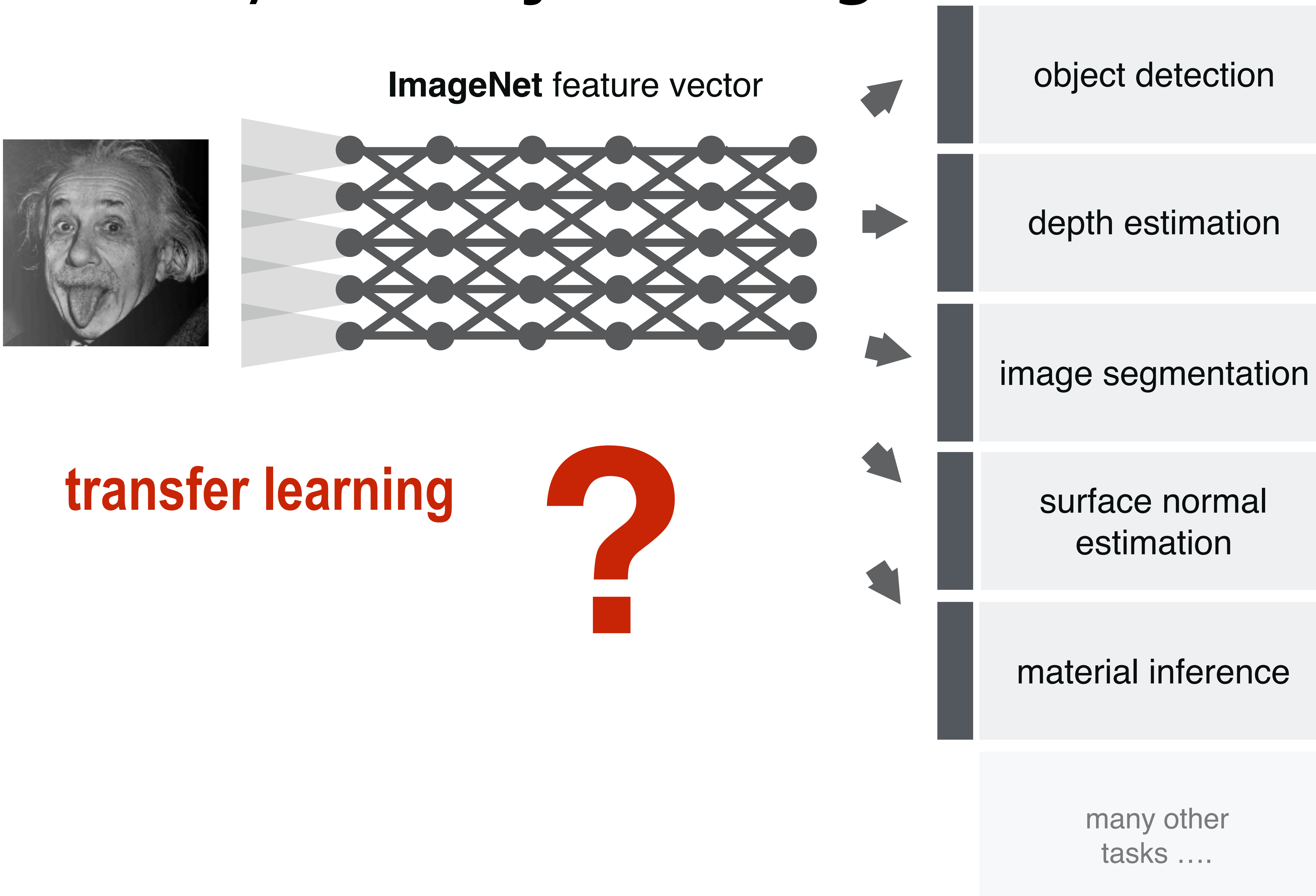


How useful are these representations beyond object recognition?

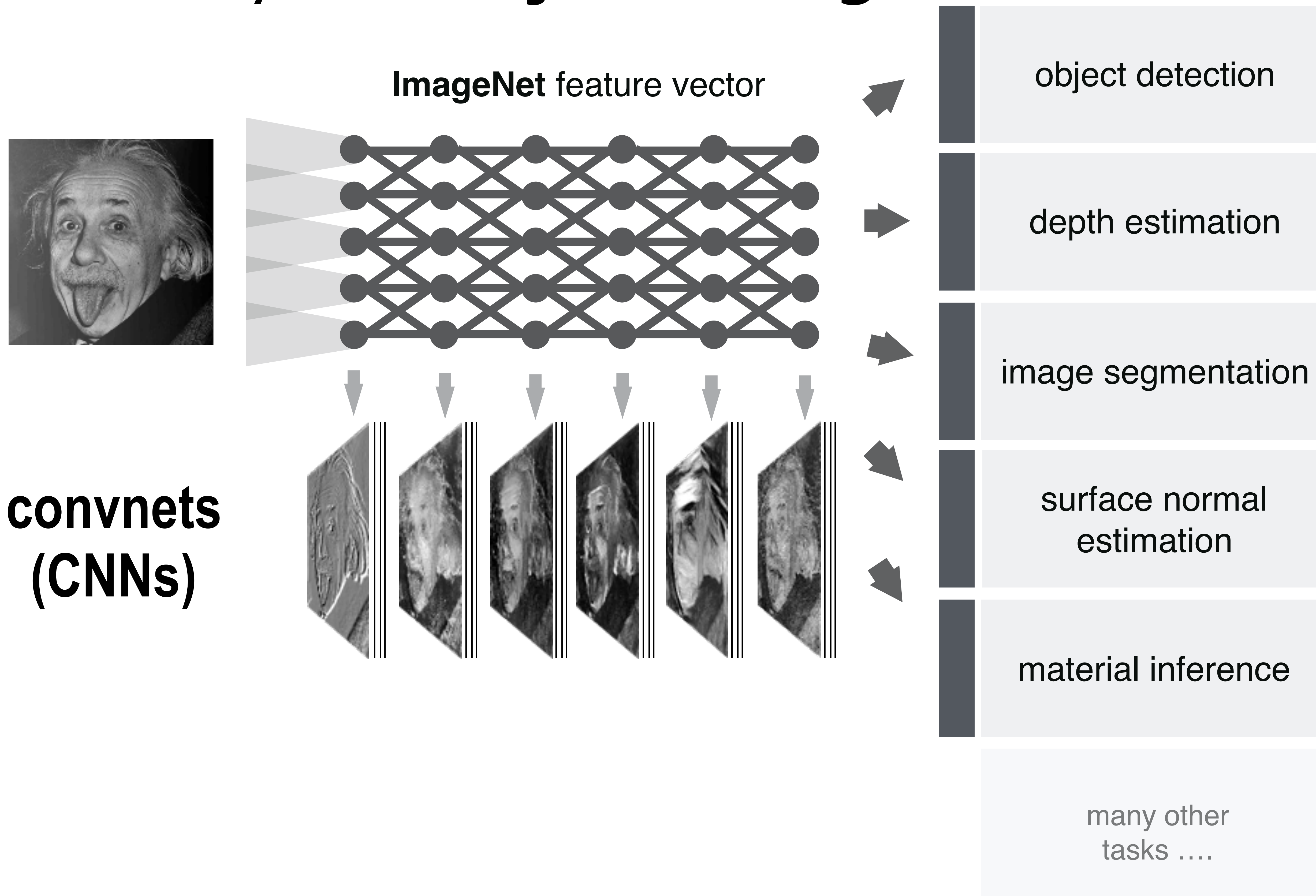




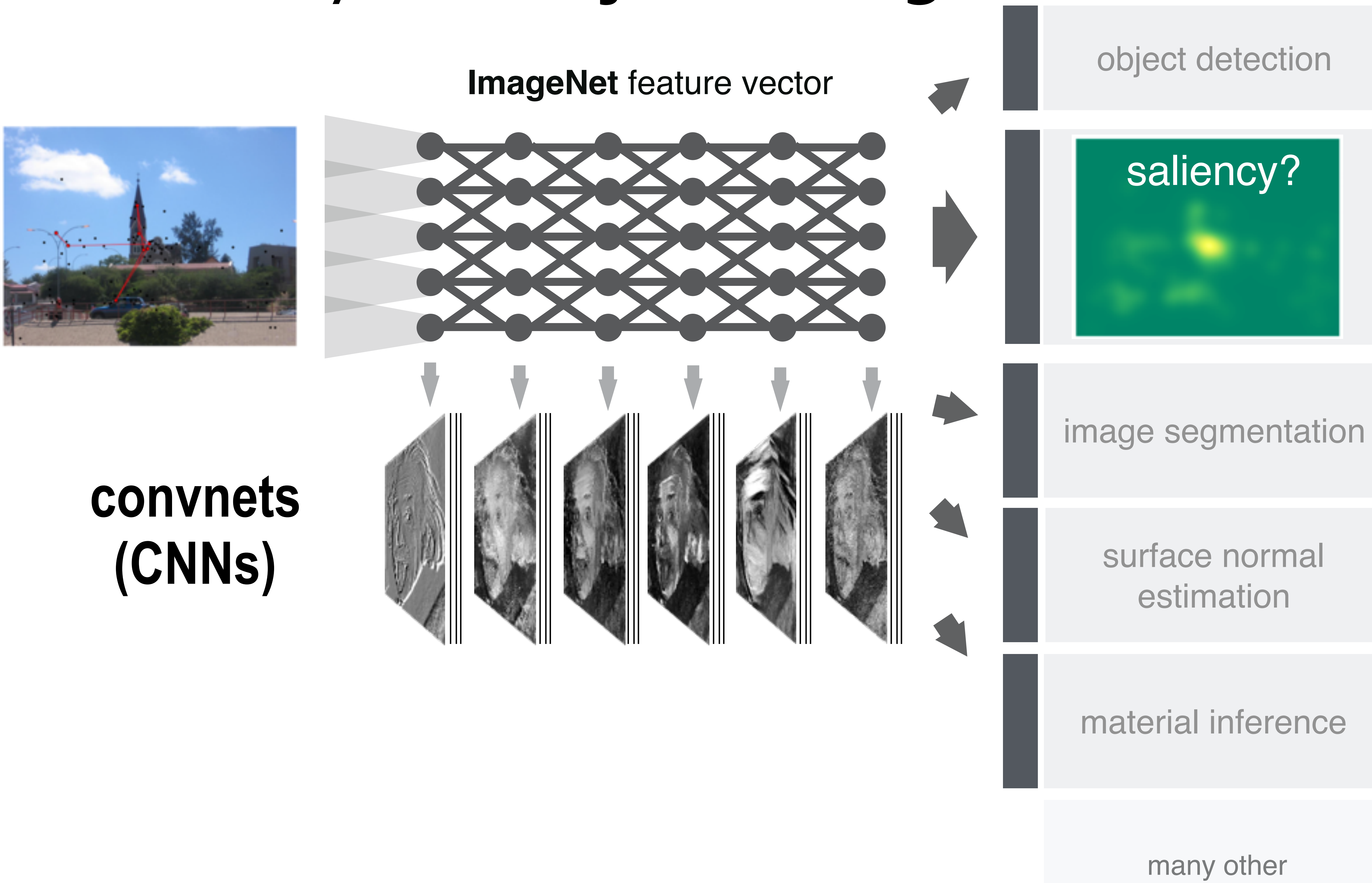
How useful are these representations beyond object recognition?



How useful are these representations beyond object recognition?



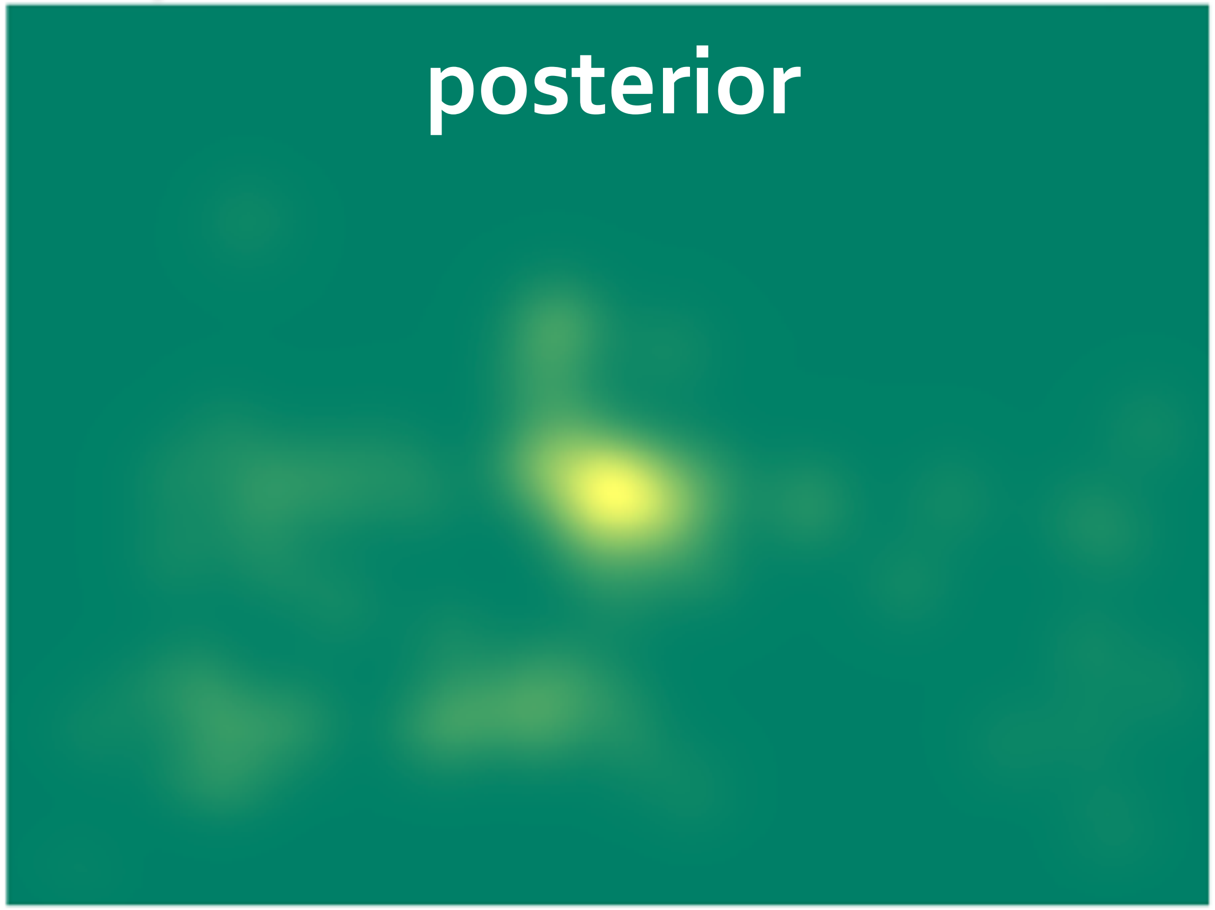
How useful are these representations beyond object recognition?



Predicting where people look

I

$\rho(x, y)$ $\rho(x, y|I)$

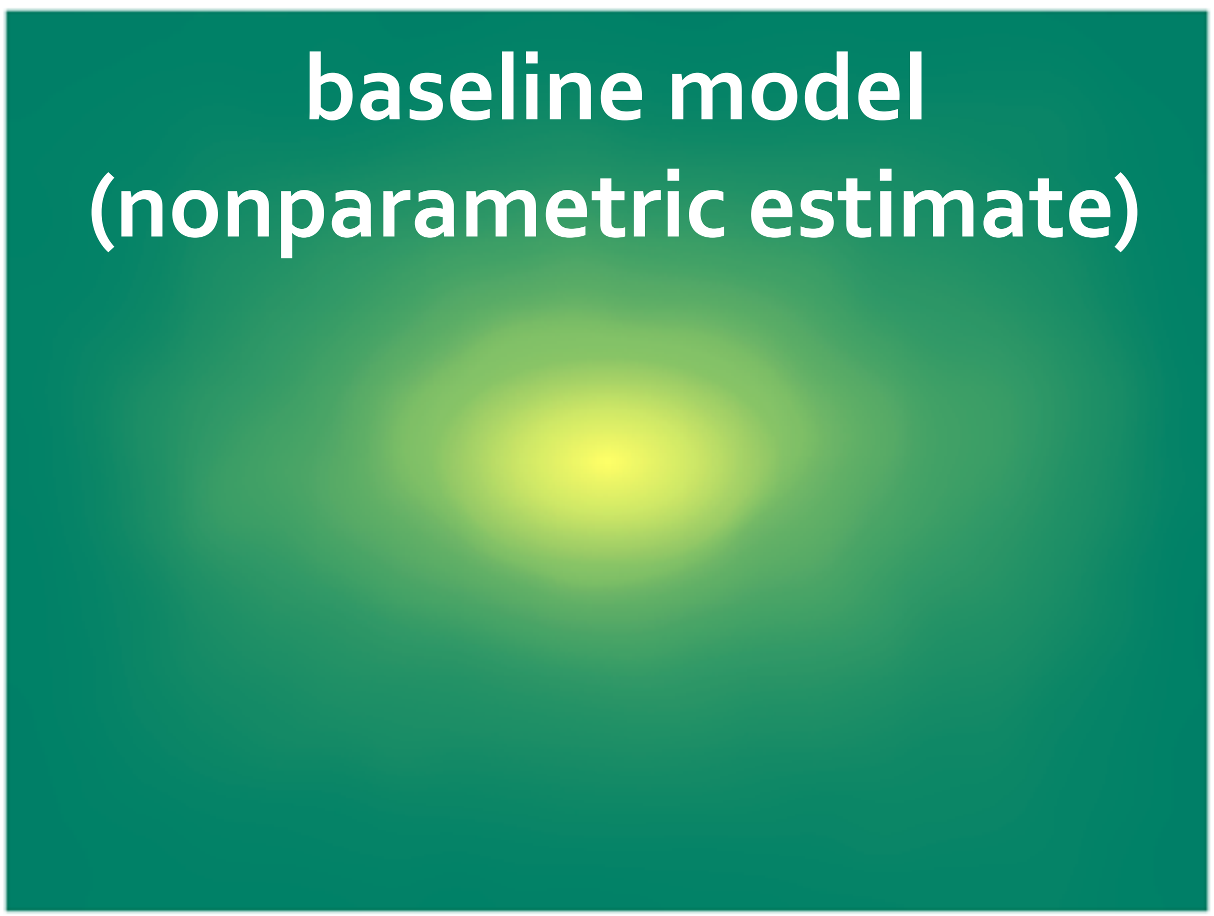


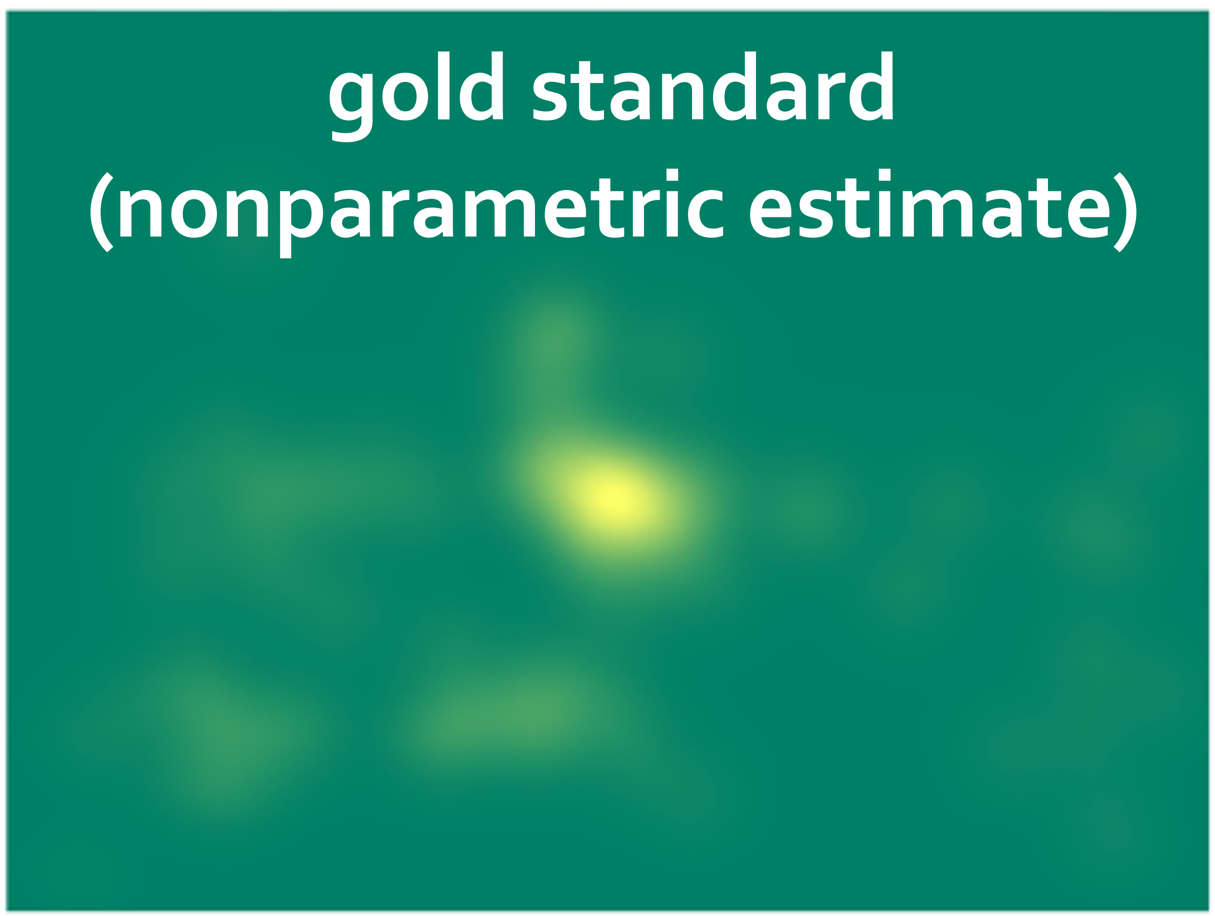
Predicting where people look

Practical setup:

$$\rho(x, y) = \frac{1}{N}\sum_{I=1}^{N} \rho(x, y|I)$$

$$\rho(x, y|I)$$





Predicting where people look

$$I$$

$$\rho(x, y) \xrightarrow{\quad} \rho(x, y|I)$$

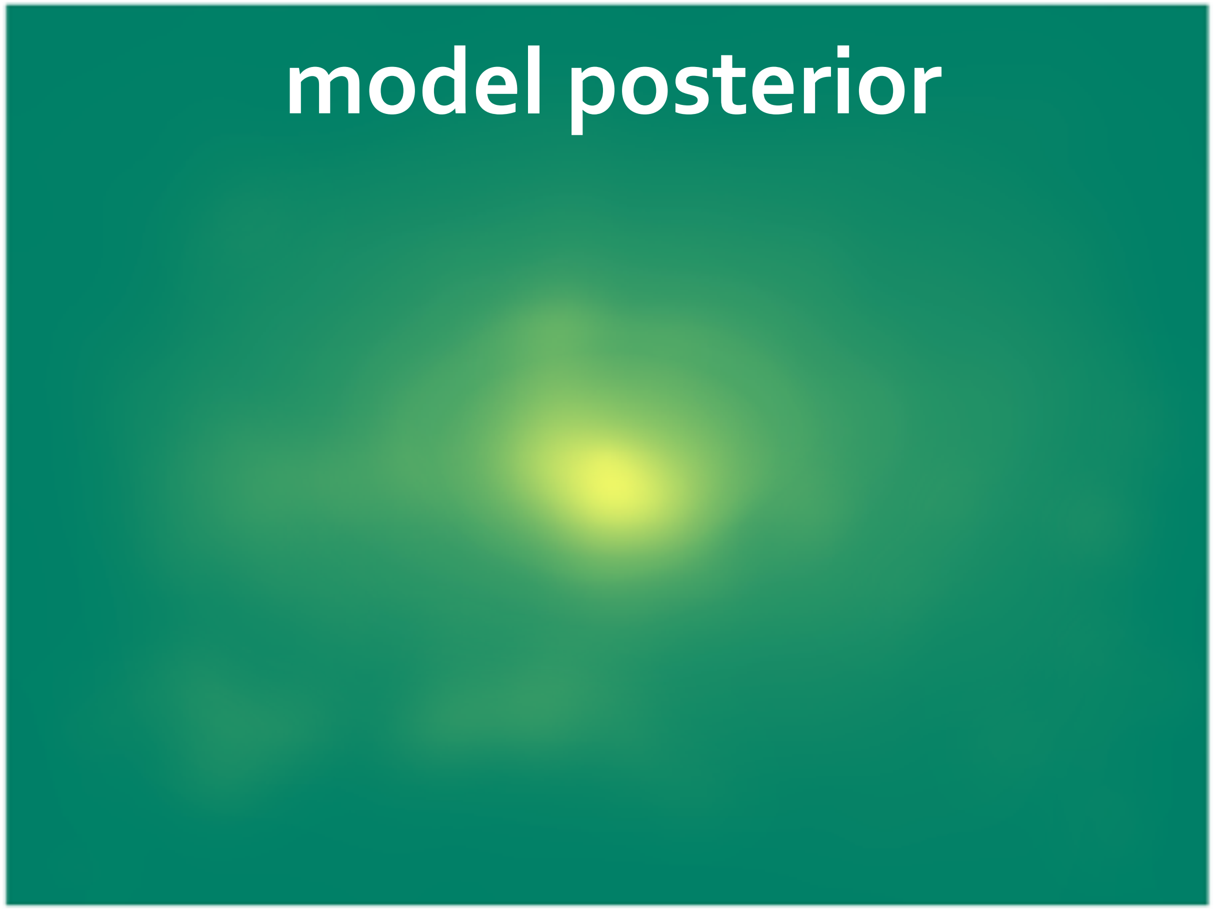
Model comparison

information gain: $$E\left[\log\left(\frac{\rho_M(x,y|I)}{\rho_{base}(x,y)}\right)\right]$$

= how much information a model can gain from the given image about the x-y-positions of fixations.

$\rho_{base}(x,y)$

$\rho_M(x,y|I)$



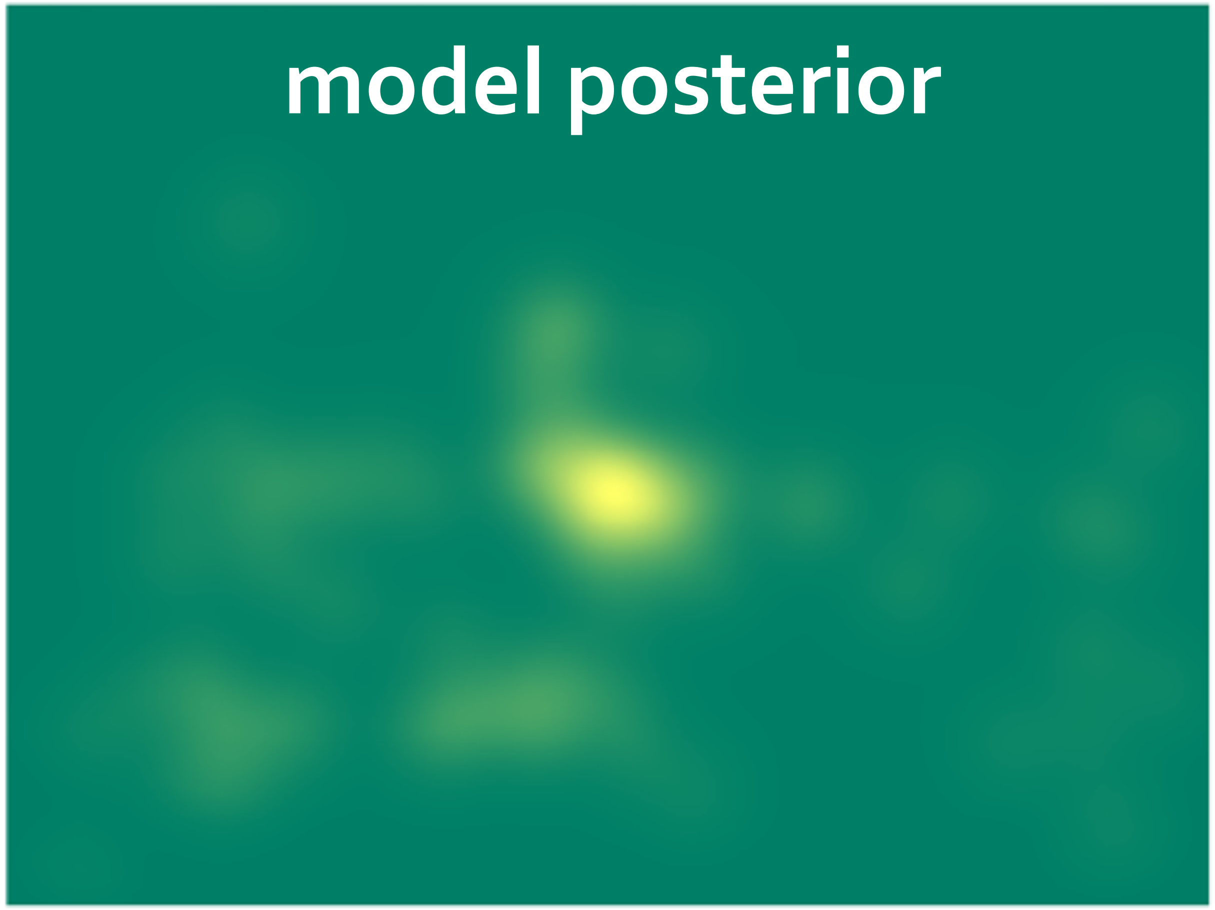
Model comparison

information gain: $$E\left[\log\left(\frac{\rho_M(x,y|I)}{\rho_{base}(x,y)}\right)\right]$$

= how much information a model can gain from the given image about the x-y-positions of fixations.

$\rho_{base}(x,y)$

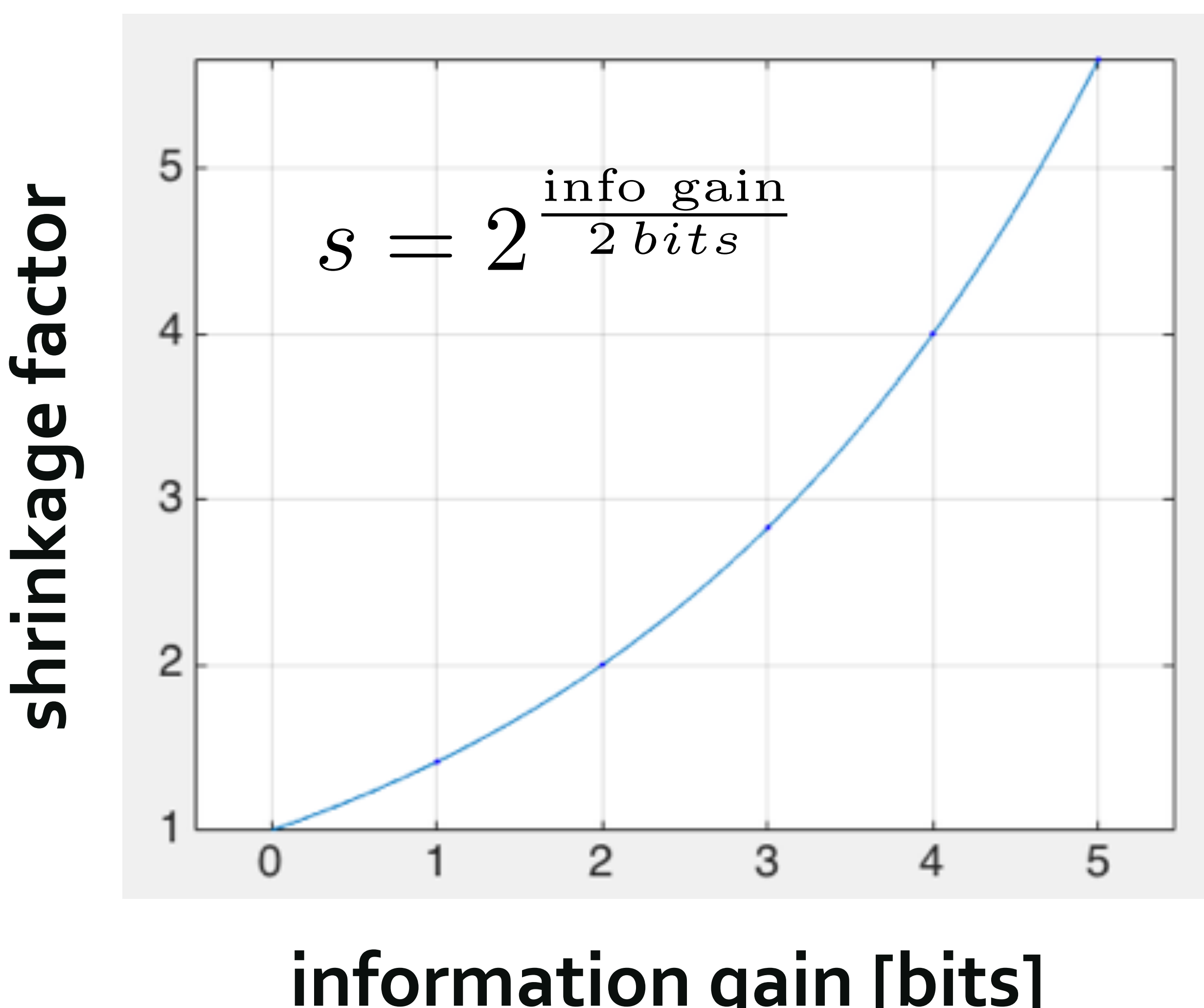
$\rho_M(x,y|I)$

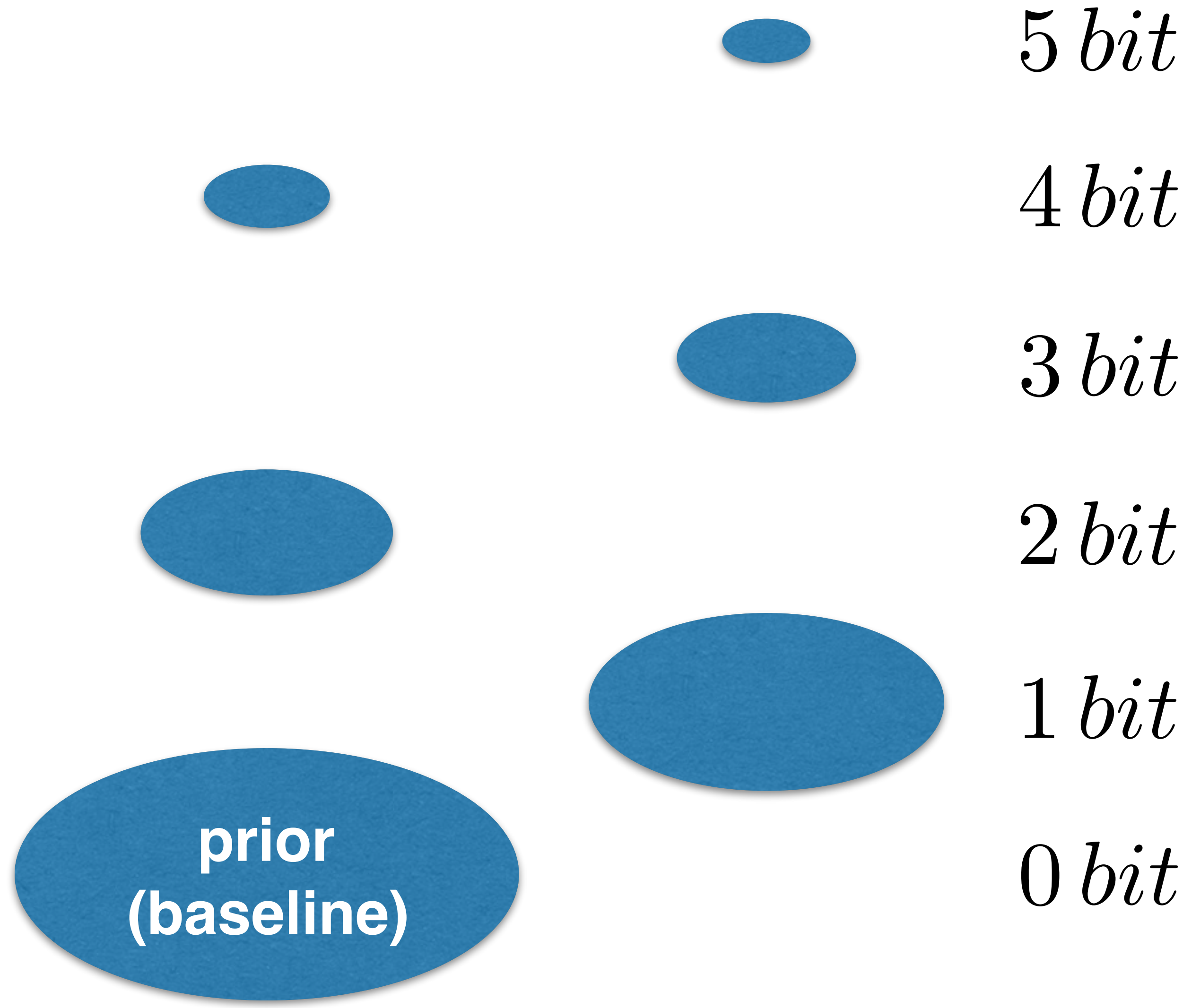


Model comparison

information gain: $$E\left[\log\left(\frac{\rho_M(x,y|I)}{\rho_{base}(x,y)}\right)\right]$$

= how much information a model can gain from the given image about the x-y-positions of fixations.



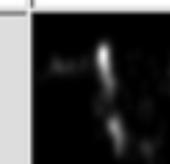
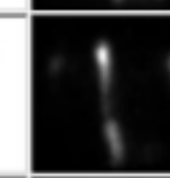
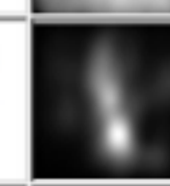
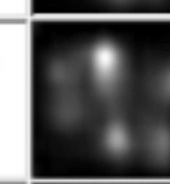


Predicting where people look

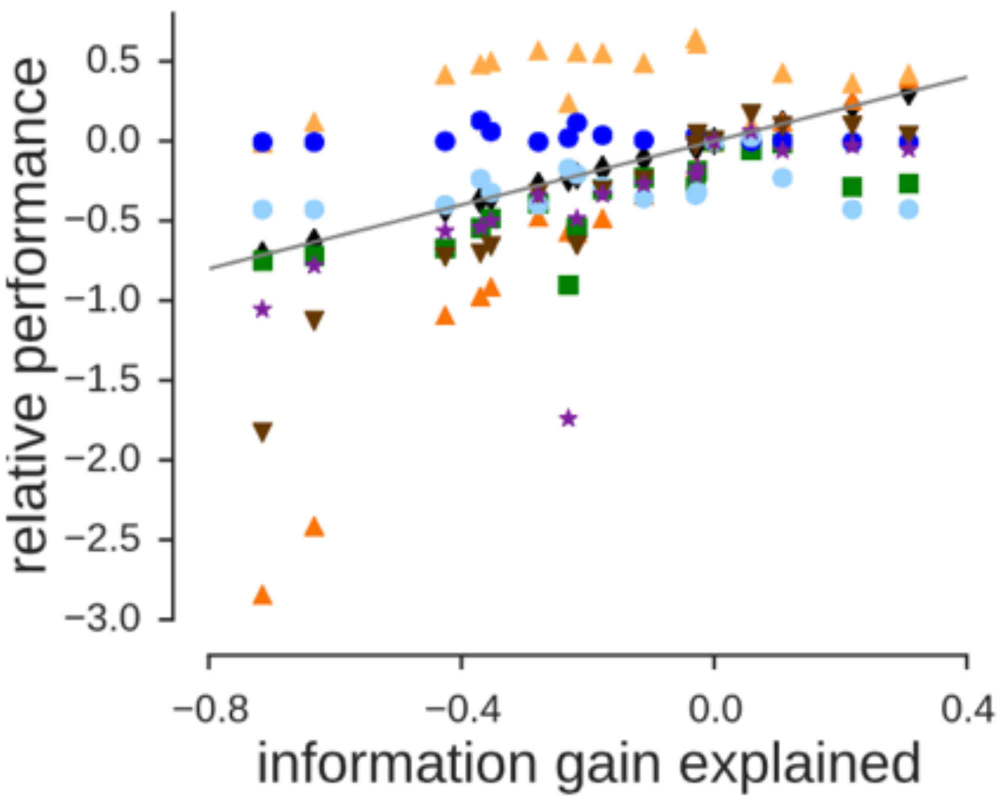
mit300
saliency benchmark

http://saliency.mit.edu/

50 models, **5** baselines, **8** metrics

Model Name	Published	Code	AUC-Judd [?]	SIM [?]	EMD [?]	AUC-Borji [?]	sAUC [?]	CC [?]	NSS [?]	KL [?]	Date tested [key]	Sample [img]
Baseline: infinite humans [?]			0.91	1	0	0.87	0.80	1	3.18	0		
SALICON	Xun Huang, Chengyao Shen, Xavier Boix, Qi Zhao		0.87	0.60	2.62	0.85	0.74	0.74	2.12	0.54	first tested: 11/19/2014 last tested: 11/15/2015 maps from authors	
DeepFix	Srinivas S S Kruthiventi, Kumar Ayush, R. Venkatesh Babu DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations [arXiv 2015]		0.87	0.67	2.04	0.80	0.71	0.78	2.26	2.26	first tested: 10/02/2015 last tested: 10/02/2015 maps from authors	
Deep Gaze 1	Matthias Kümmerer, Lucas Theis, Matthias Bethge. Deep Gaze I: Boosting Saliency Prediction with Feature Maps Trained on ImageNet [arxiv 2014]		0.84	0.39	4.97	0.83	0.66	0.48	1.22	1.23	first tested: 10/02/2014 last tested: 11/15/2015 maps from authors	
Boolean Map based Saliency (BMS)	Jianming Zhang, Stan Sclaroff. Saliency detection: a boolean map approach [ICCV 2013]	matlab, executable	0.83	0.51	3.35	0.82	0.65	0.55	1.41	0.81	first tested: 14/05/2014 last tested: 23/09/2014 maps from authors	
SalNet	Kevin McGuinness. Unpublished work.		0.83	0.52	3.31	0.82	0.69	0.58	1.51	0.81	first tested: 06/17/2015 last tested: 11/15/2015 maps from authors	
Mixture of Saliency Models	Xuehua Han, Shunji Satoh. "Unifying computational models for visual attention" [AINI 2014, Sep. (accepted)]		0.82	0.44	4.22	0.81	0.62	0.52	1.34	0.91	first tested: 08/08/2014 last tested: 23/09/2014 maps from authors	
Ensembles of Deep Networks	Eleonora Vig, Michael Dorr, David Cox. Large-Scale Optimization of Hierarchical Features for Saliency	python	0.82	0.41	4.56	0.81	0.62	0.45	1.14	1.14	first tested: 08/16/2014 last tested: 11/15/2015	

Model comparison



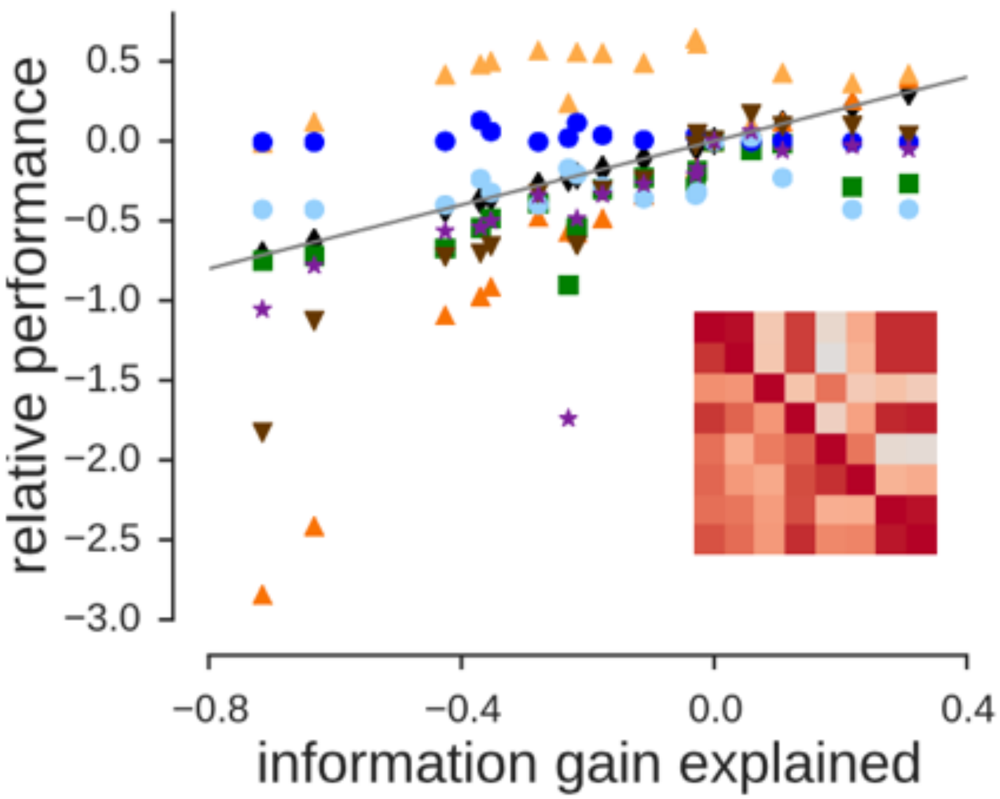
$$\text{rel. performance} = \frac{\mathbf{d}(\text{model}) - \mathbf{d}(\text{baseline})}{\mathbf{d}(\text{gold standard}) - \mathbf{d}(\text{baseline})}$$

Matthias Kümmerer

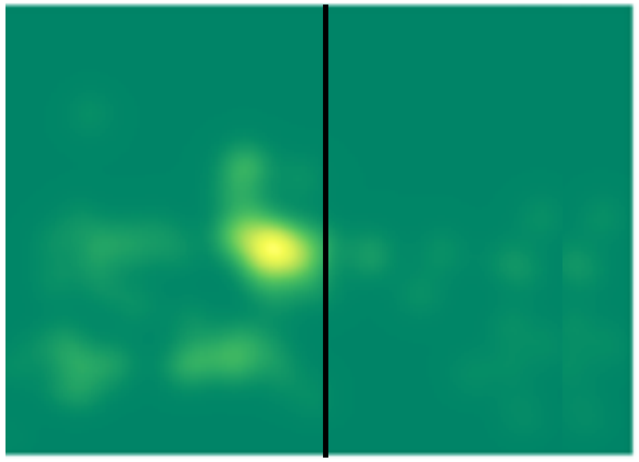
Kümmerer, Wallis, Bethge, PNAS, 112(52): 16054-16059, 2015.

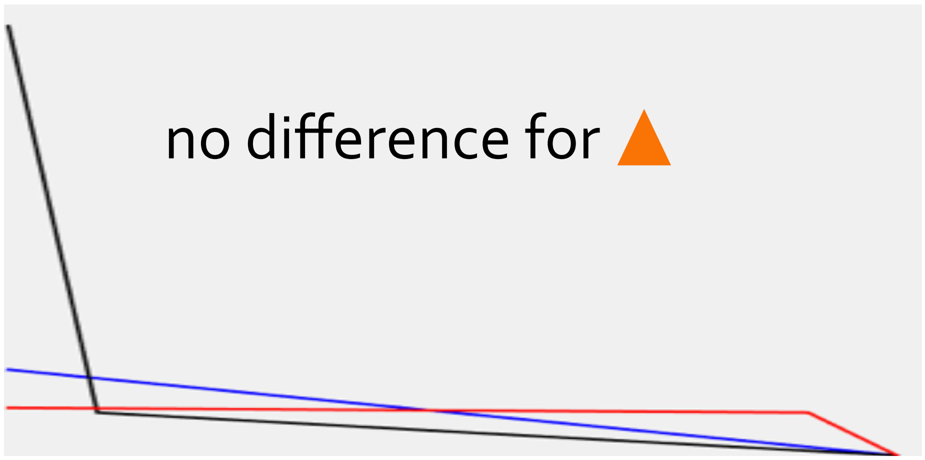
Tom Wallis

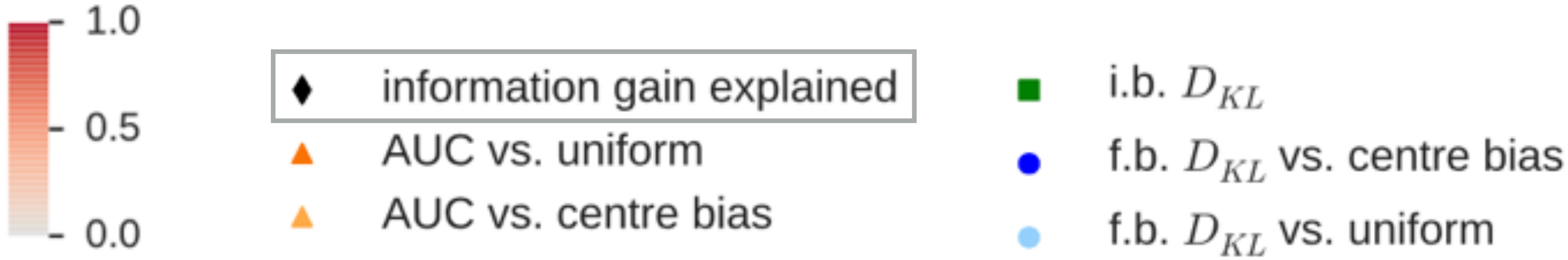
Model comparison



$$\text{rel. performance} = \frac{\mathbf{d}(\text{model}) - \mathbf{d}(\text{baseline})}{\mathbf{d}(\text{gold standard}) - \mathbf{d}(\text{baseline})}$$





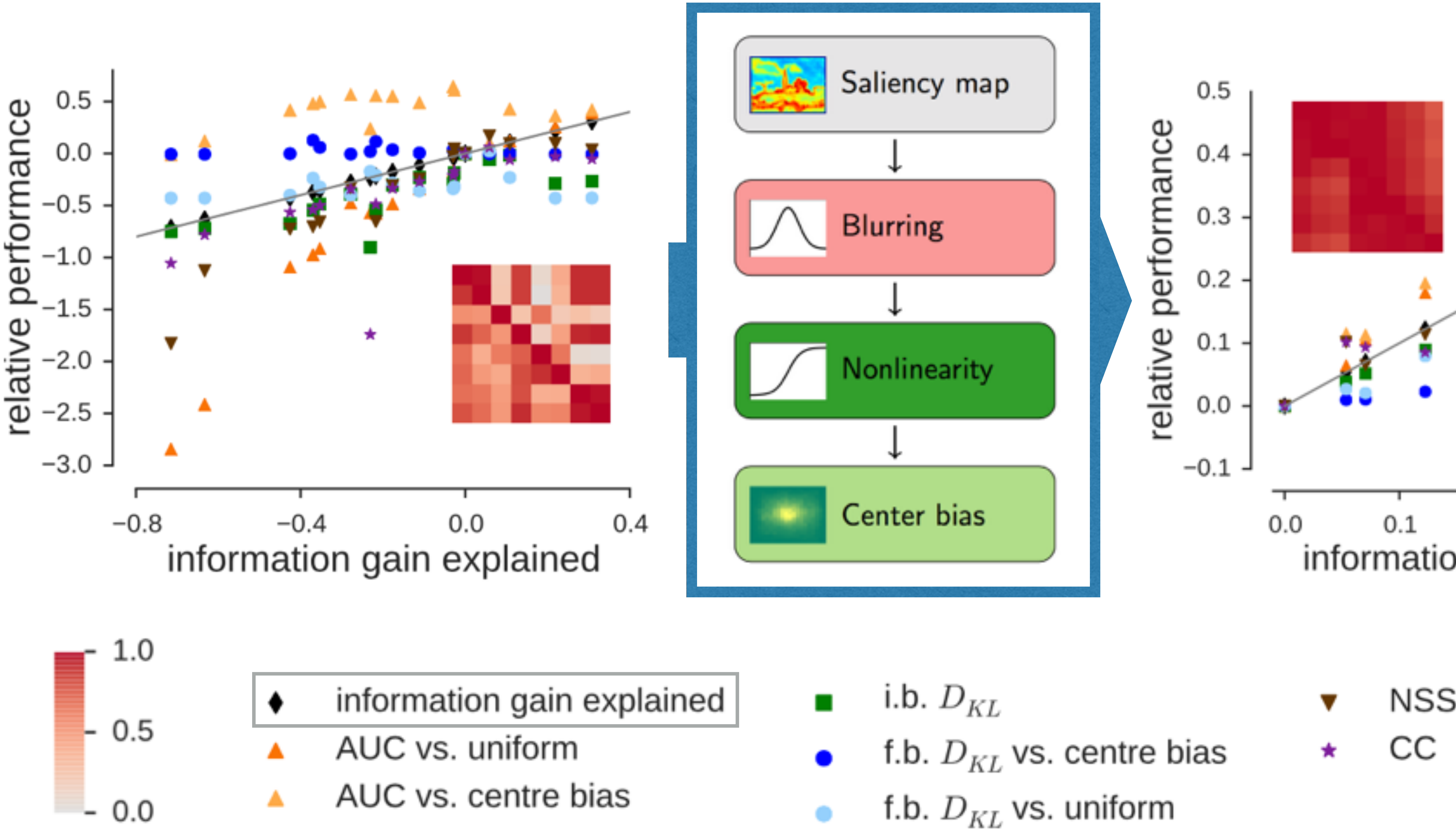


Matthias Kümmerer

Kümmerer, Wallis, Bethge, PNAS, 112(52): 16054-16059, 2015.

Tom Wallis

Model comparison



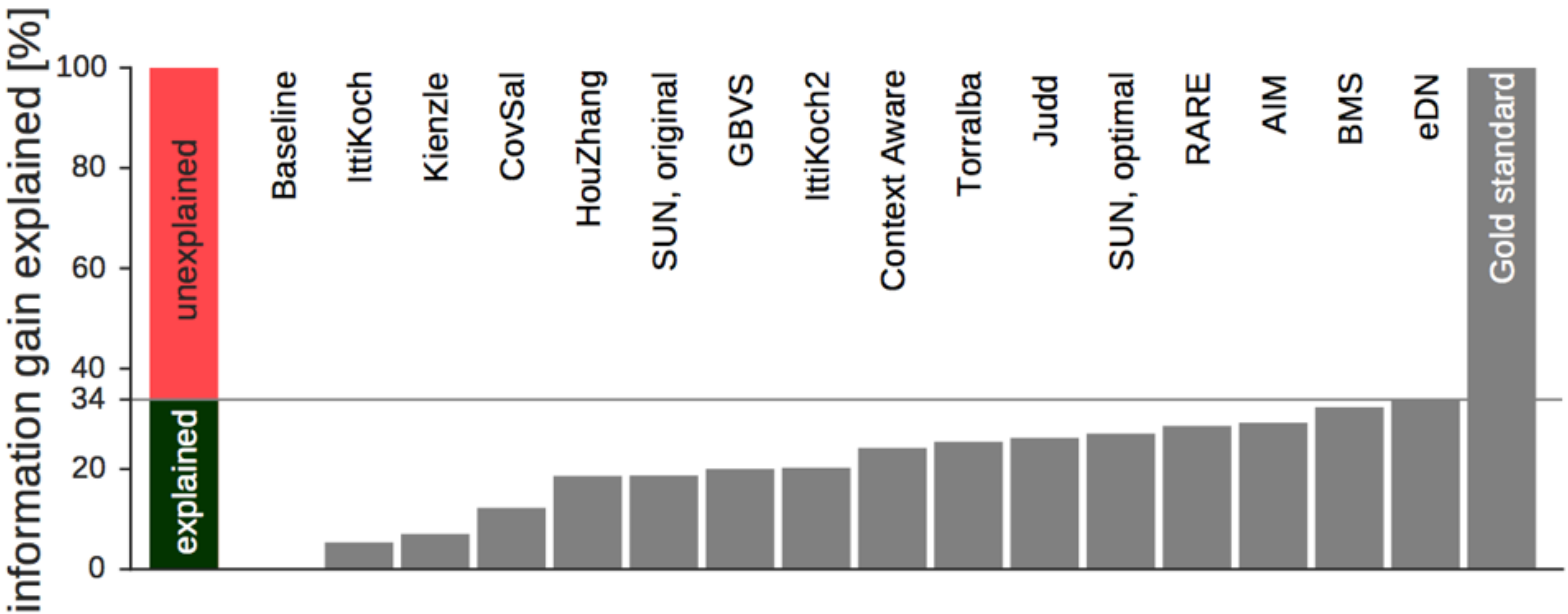
Matthias Kümmerer

Kümmerer, Wallis, Bethge, PNAS, 112(52): 16054-16059, 2015.

Tom Wallis

Model comparison

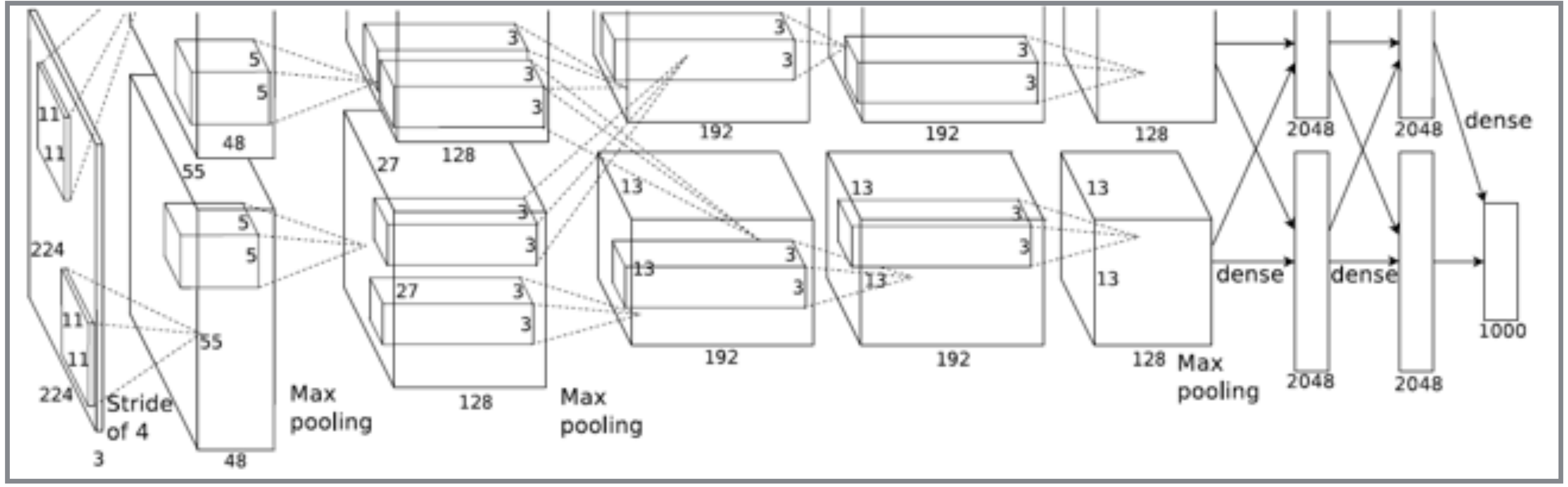
Measure model performance on a “ratio scale”



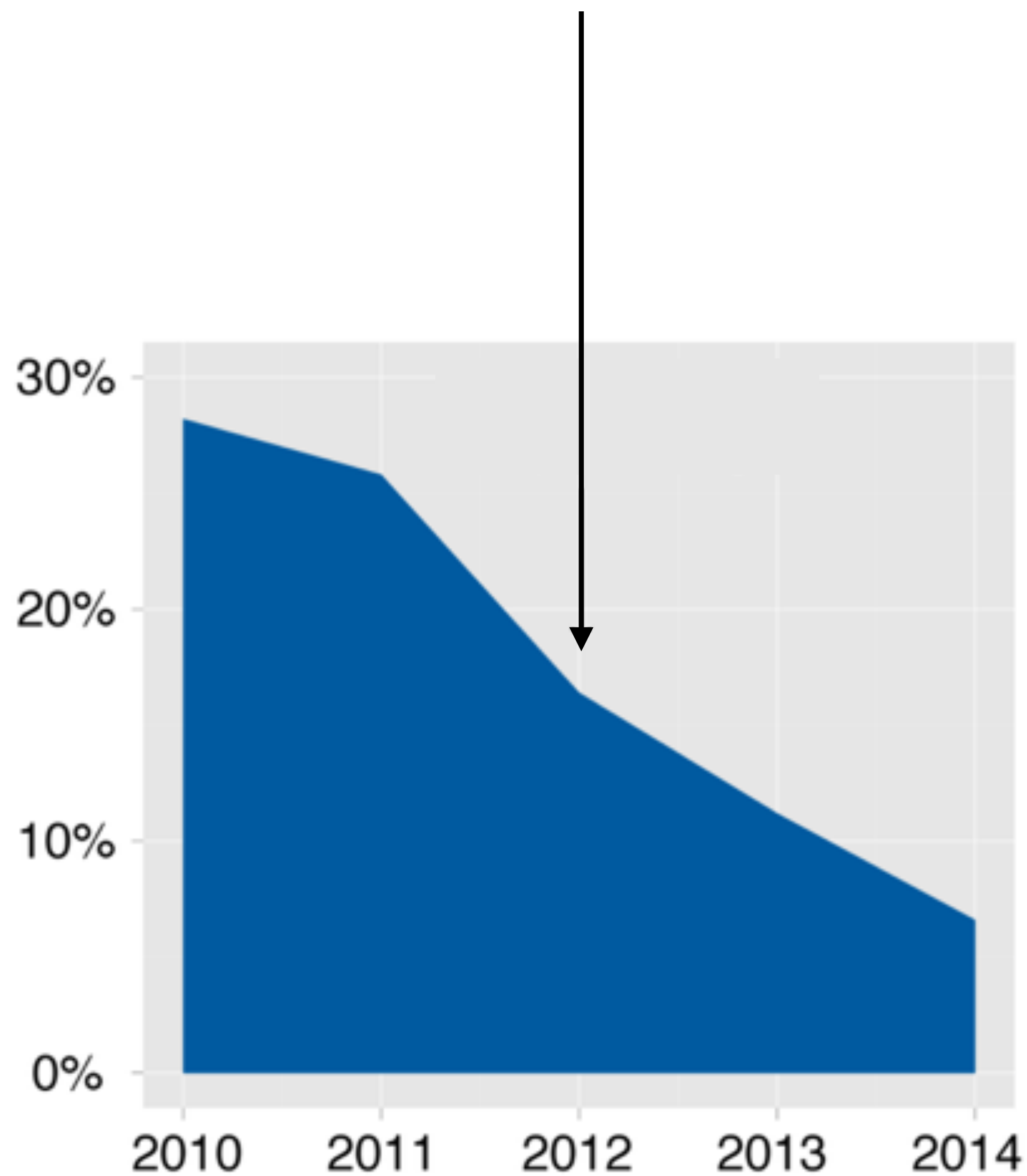
Matthias Kümmerer

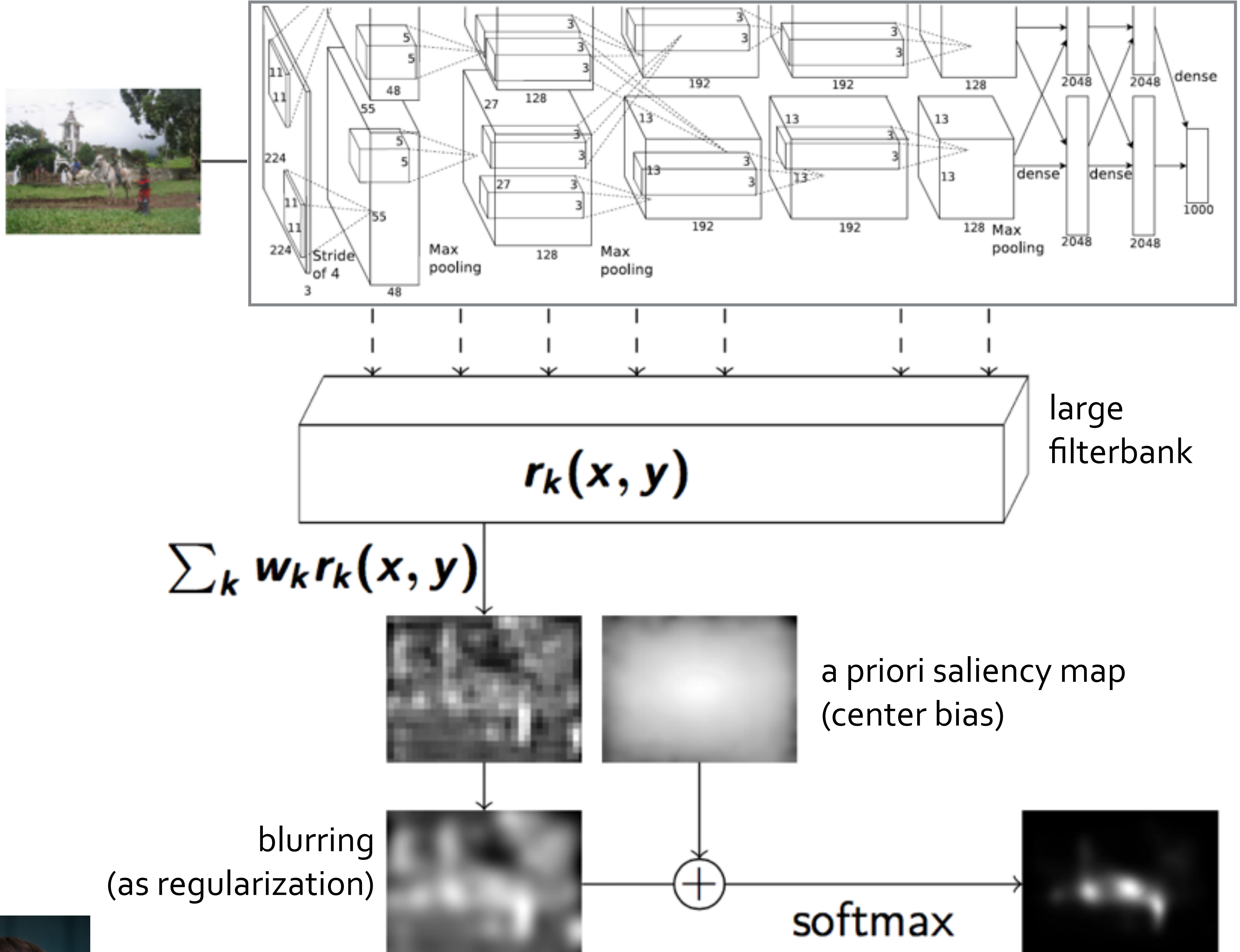
Kümmerer, Wallis, Bethge, PNAS, 112(52): 16054-16059, 2015.

Tom Wallis



AlexNet (Krizhevsky et al 2012)



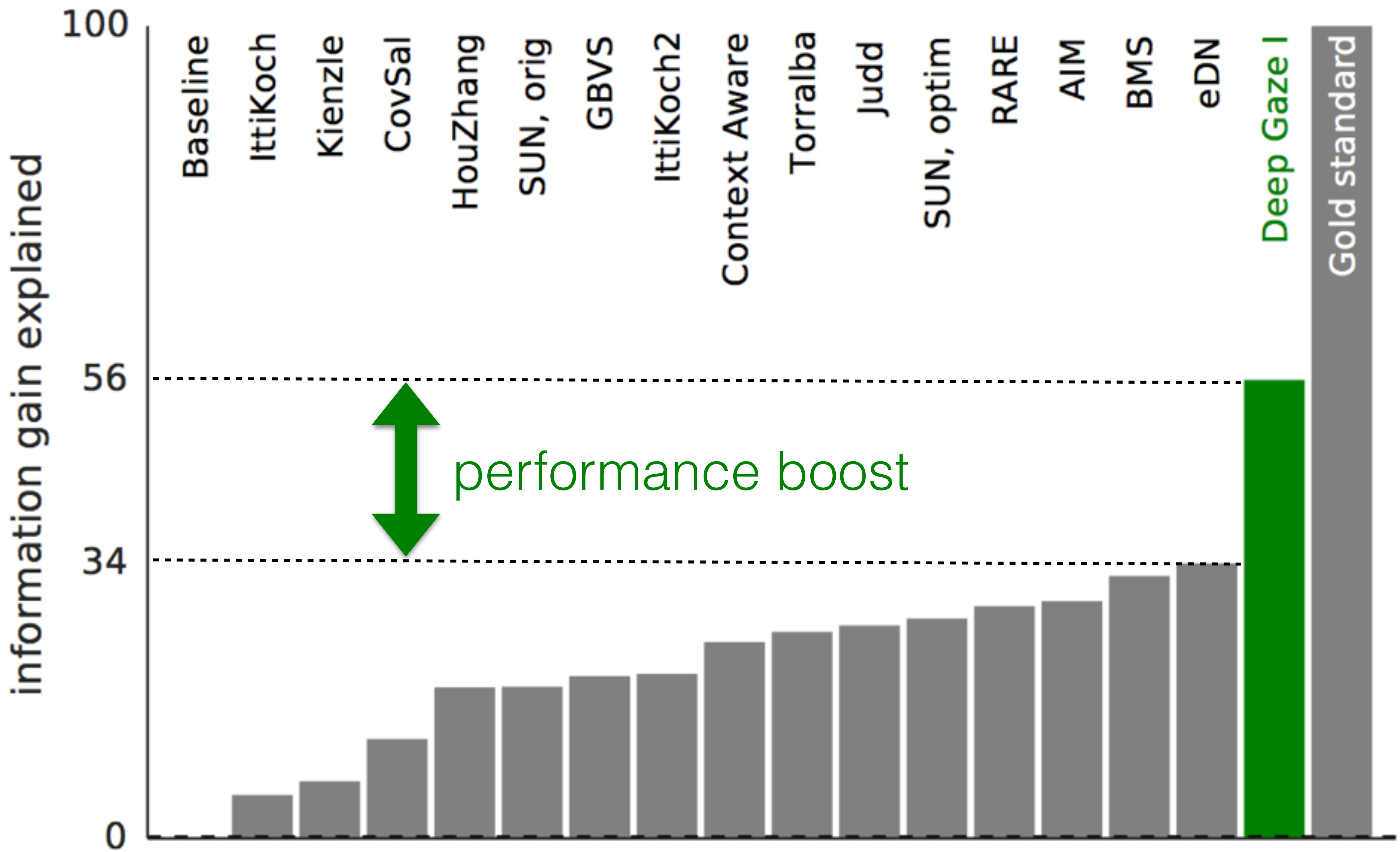


Matthias
Kümmerer

ICLR Workshop paper

http://arxiv.org/abs/1411.1045

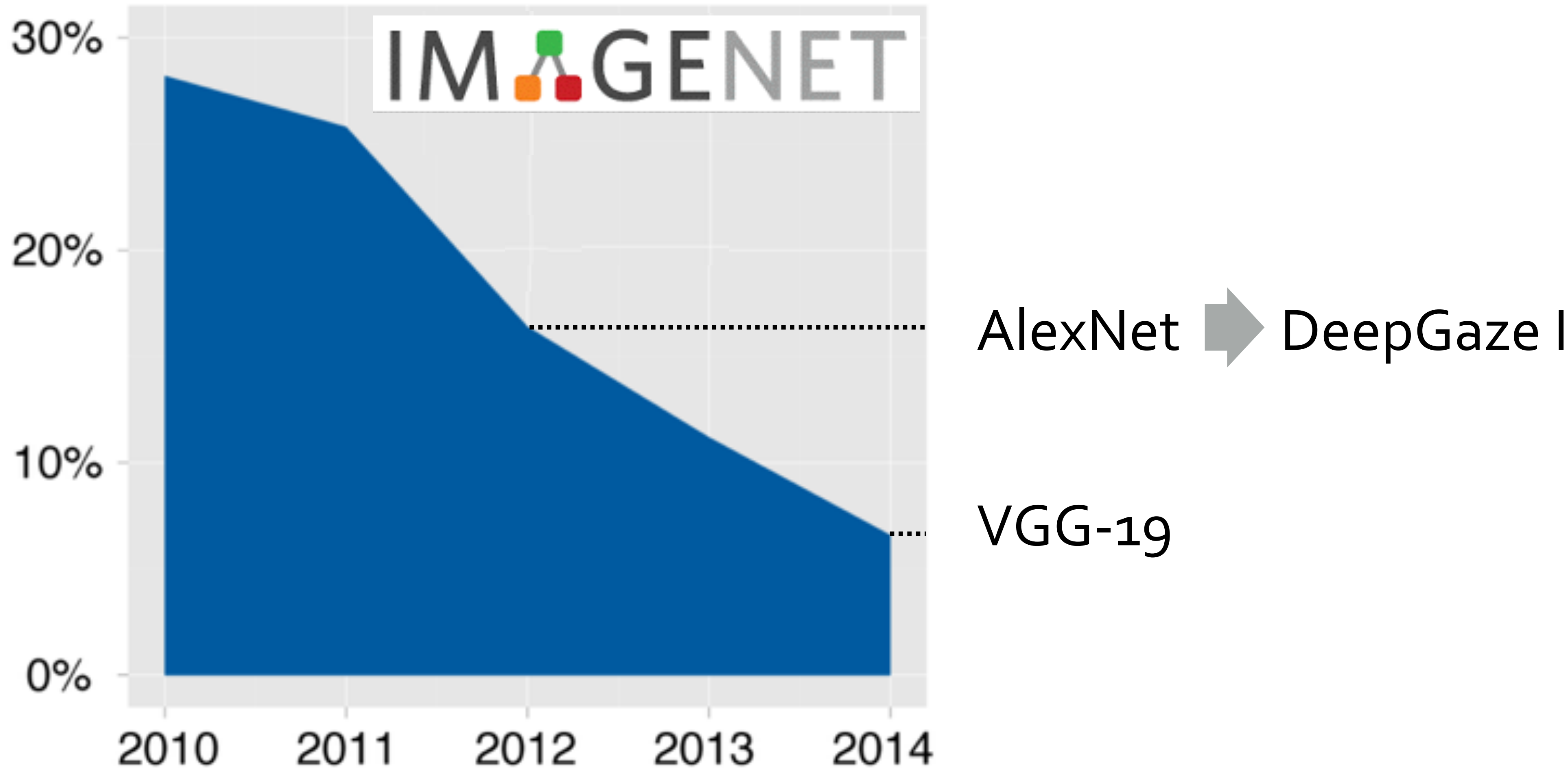
Lucas
Theis

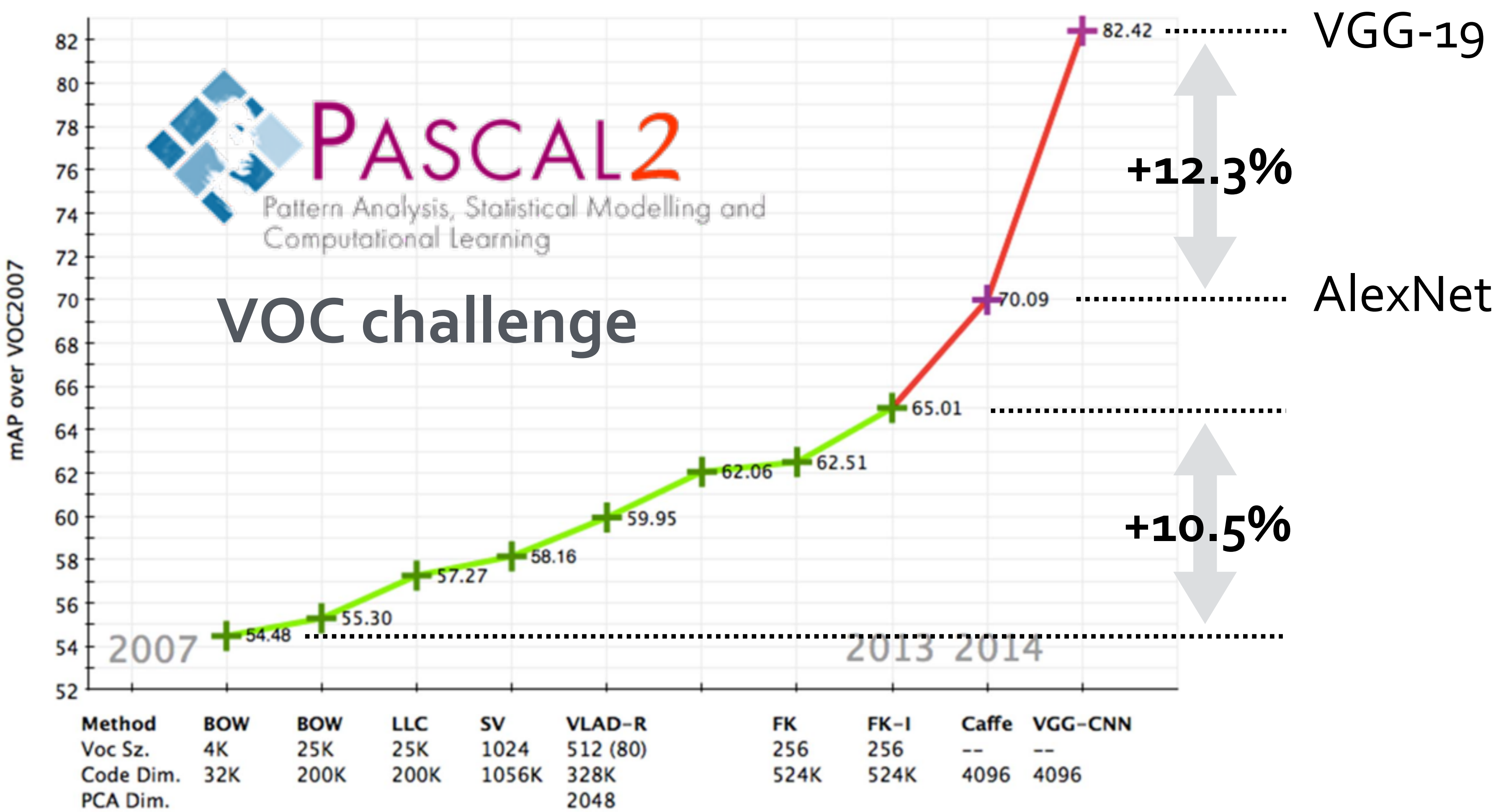


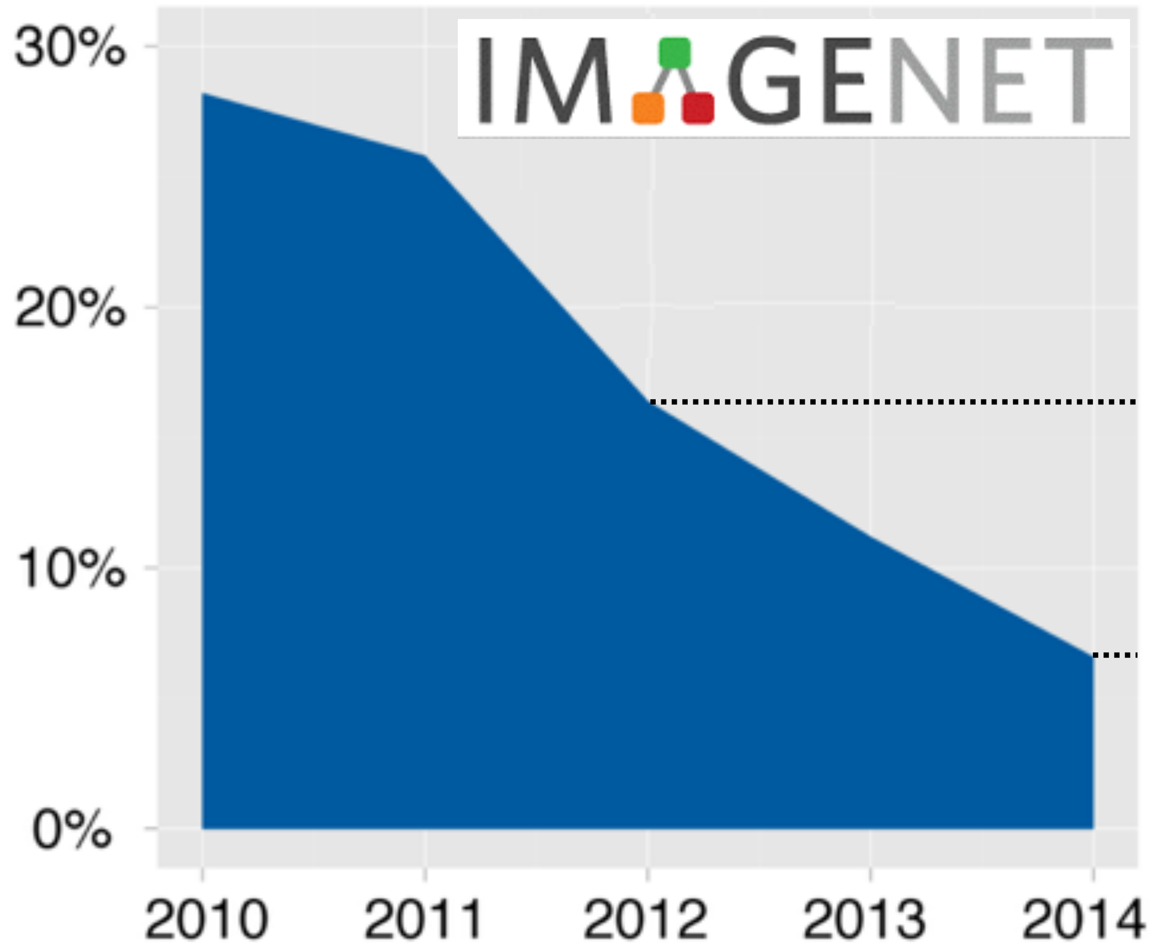
Matthias
Kümmerer

ICLR Workshop paper
http://arxiv.org/abs/1411.1045

Lucas
Theis

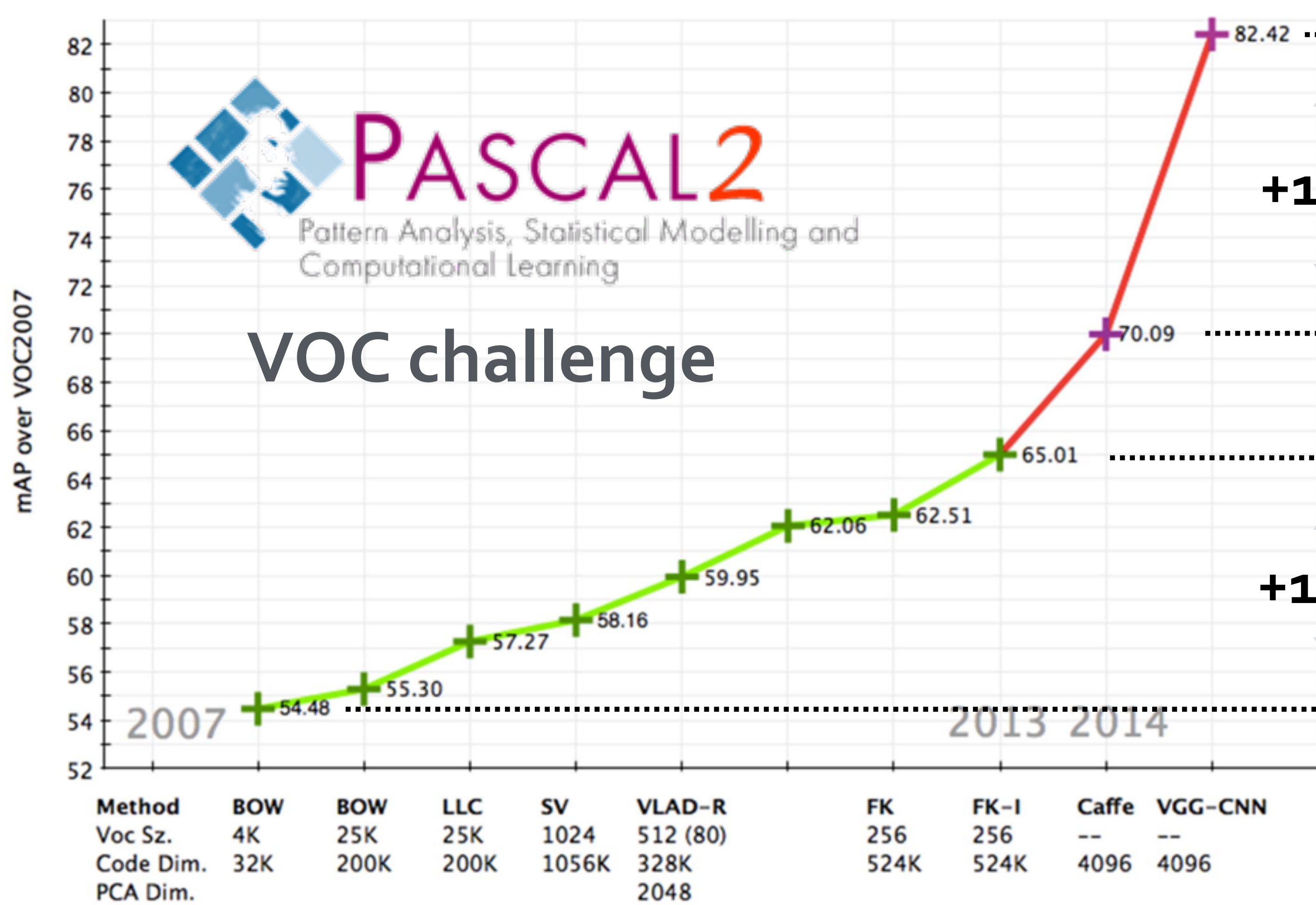

IMAGENET
30%
20%
10%
0%
2010
2011
2012
2013
2014
AlexNet
DeepGaze I
VGG-19


PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning
VOC challenge
mAP over VOC2007
82.42
70.09
65.01
62.51
62.06
59.95
58.16
57.27
55.30
54.48
2007
2013 2014
VGG-19
+12.3%
AlexNet
+10.5%
Method BOW BOW LLC SV VLAD-R FK FK-I Caffe VGG-CNN
Voc Sz. 4K 25K 25K 1024 512 (80) 256 256 -- --
Code Dim. 32K 200K 200K 1056K 328K 524K 524K 4096 4096
PCA Dim. 2048



AlexNet ➔ DeepGaze I

VGG-19 ➔ **DeepGaze II**



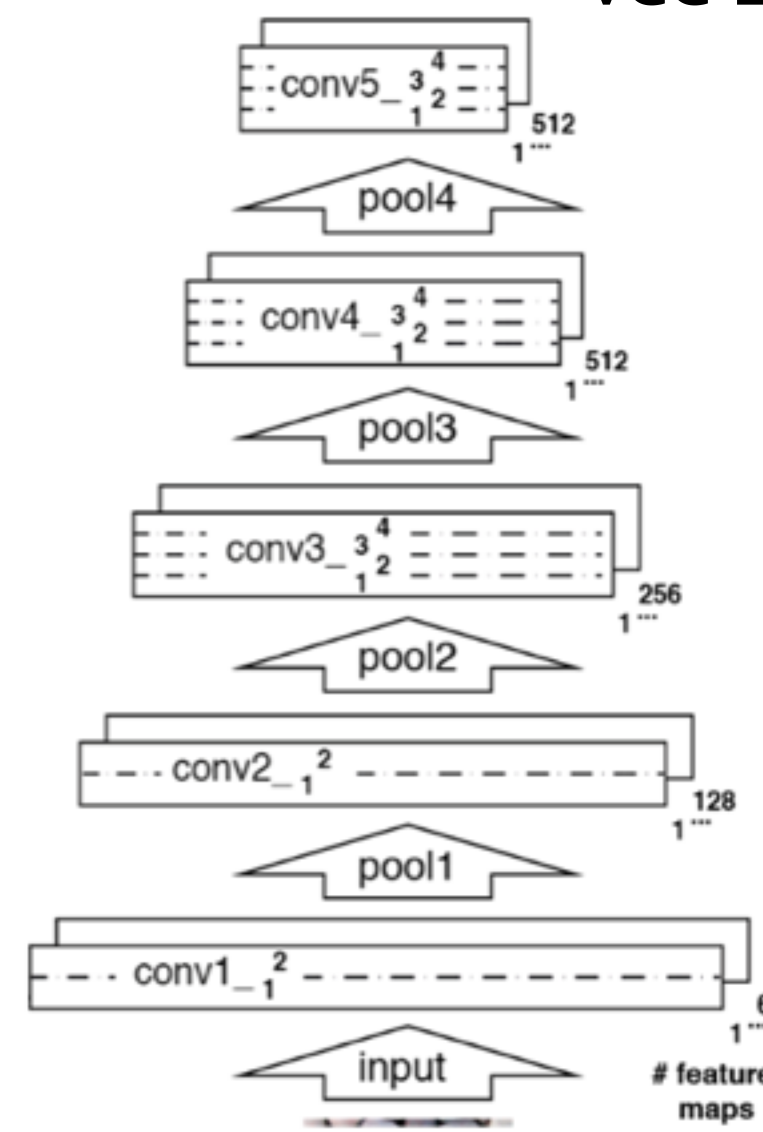
VGG-19

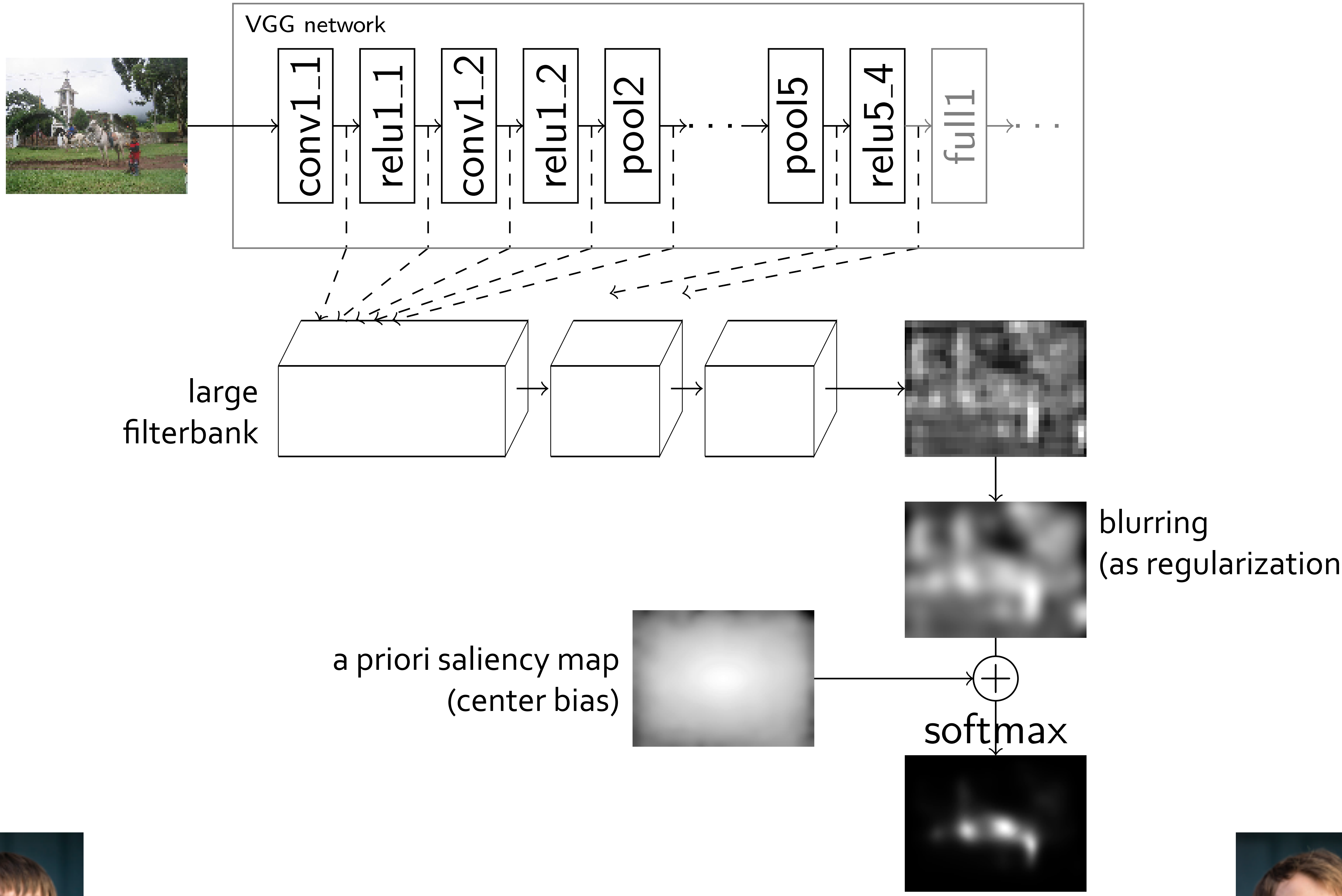
+12.3%

AlexNet

+10.5%

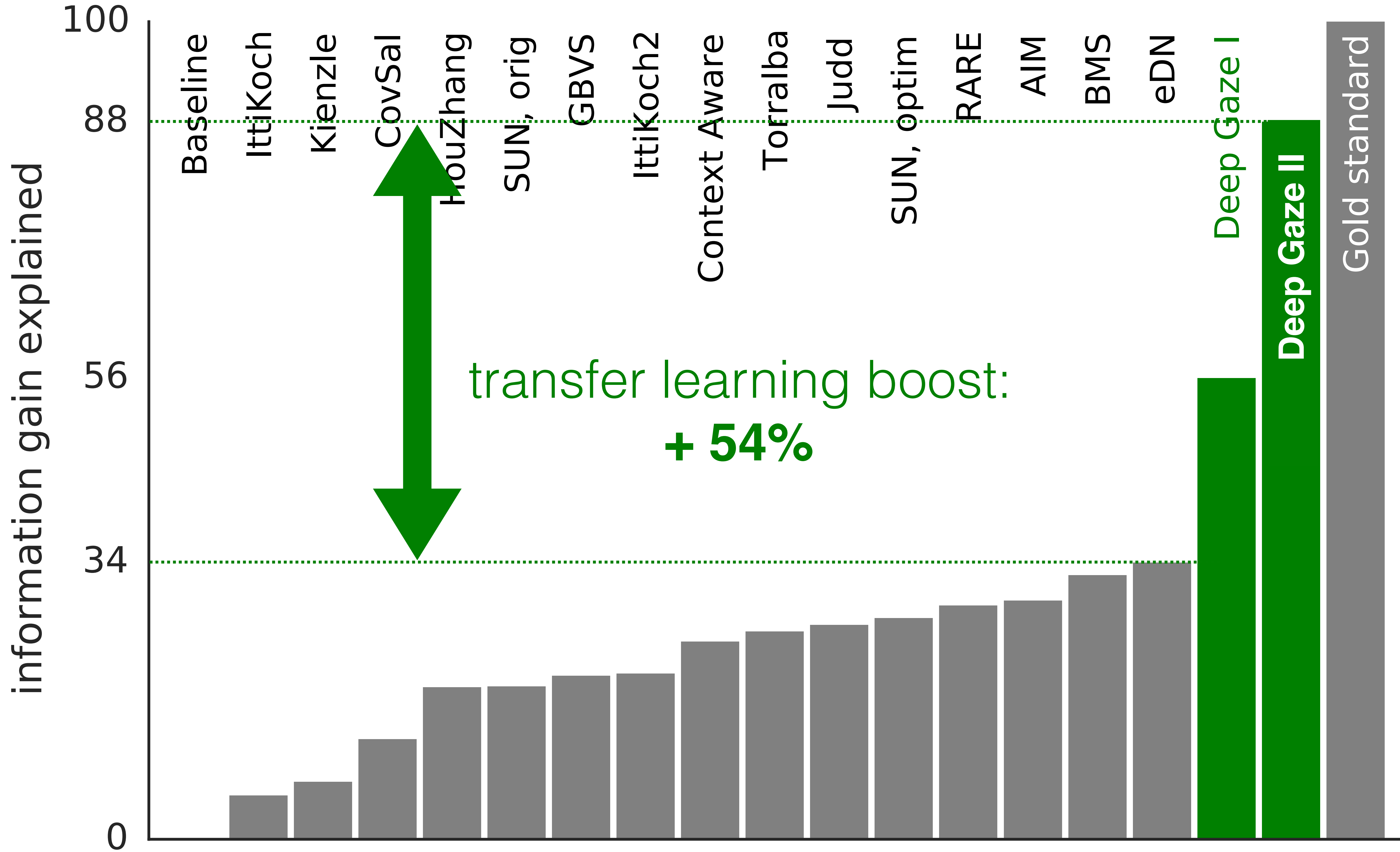
VGG-19






Matthias
Kümmerer

Lucas
Theis

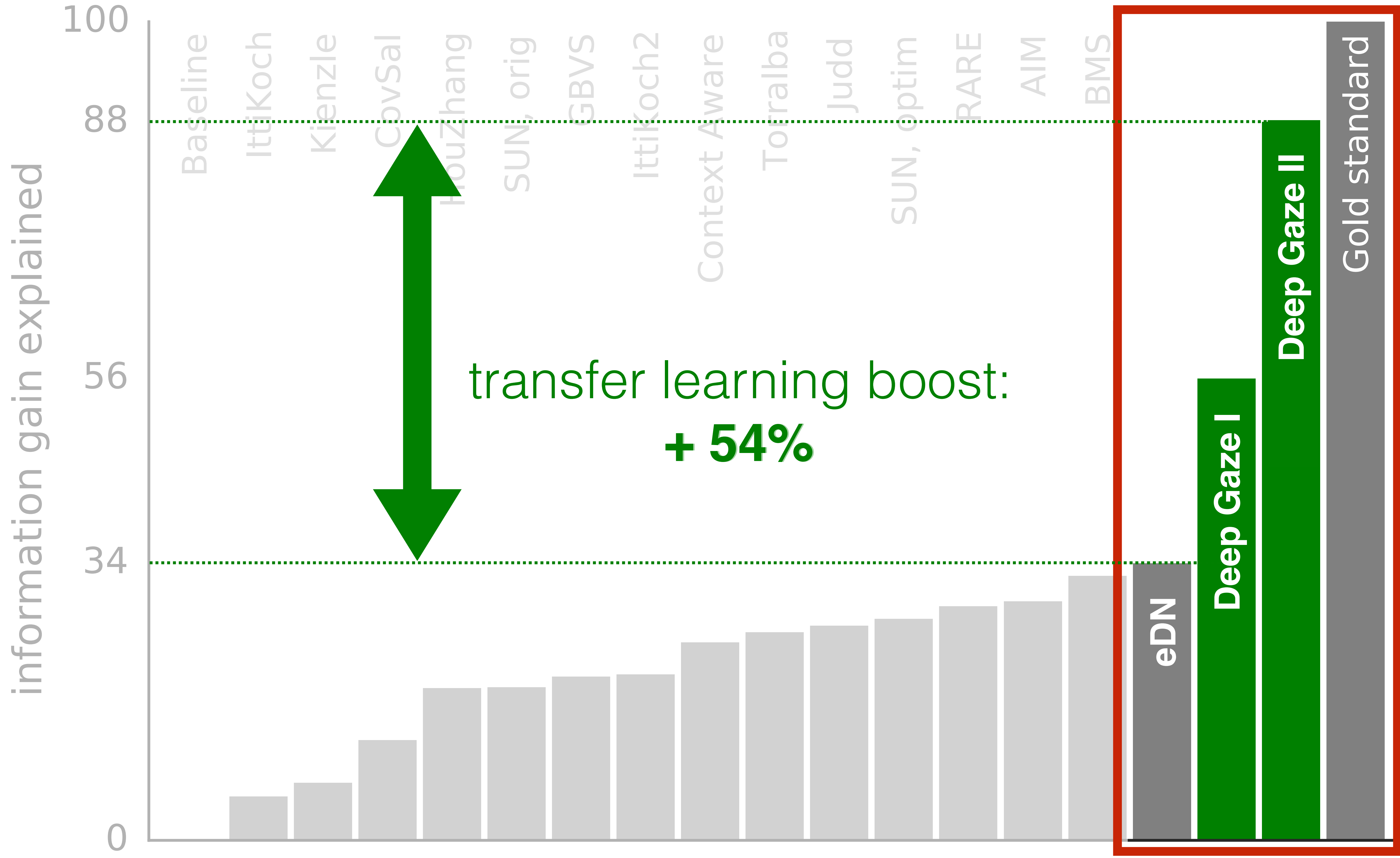



Matthias Kümmerer

ICLR Workshop paper

http://arxiv.org/abs/1411.1045

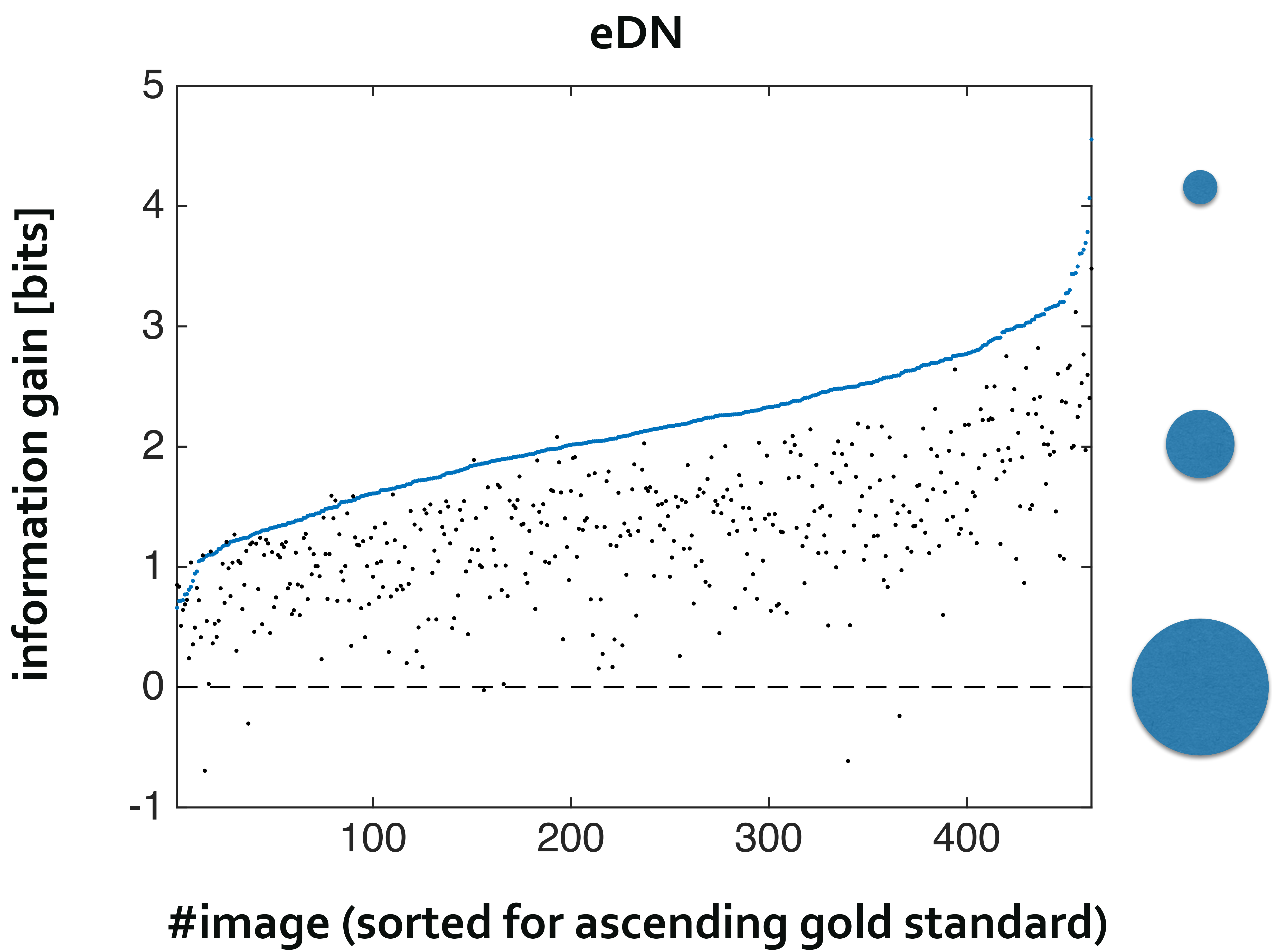
Lucas Theis

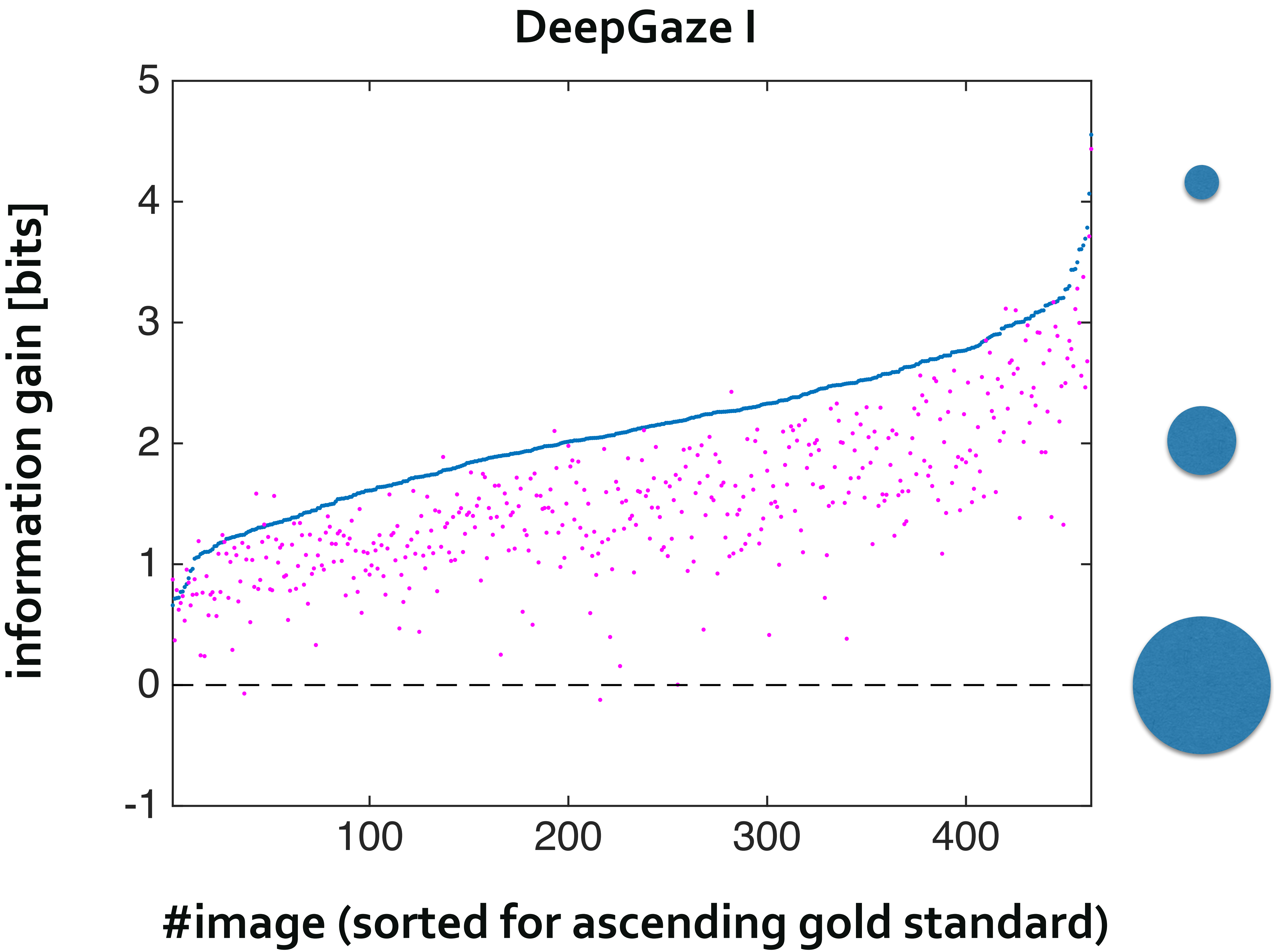


Matthias
Kümmerer

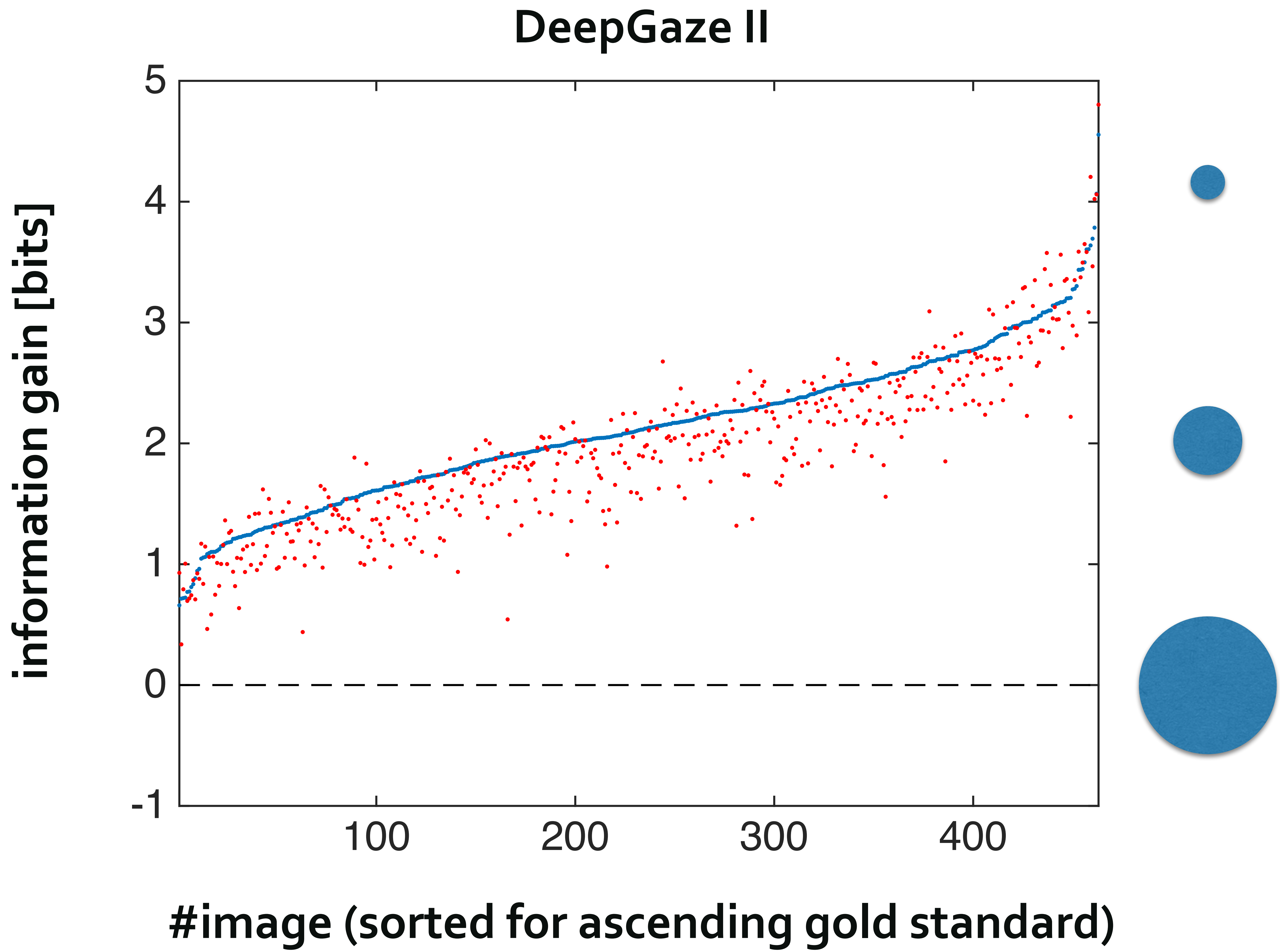
ICLR Workshop paper
http://arxiv.org/abs/1411.1045

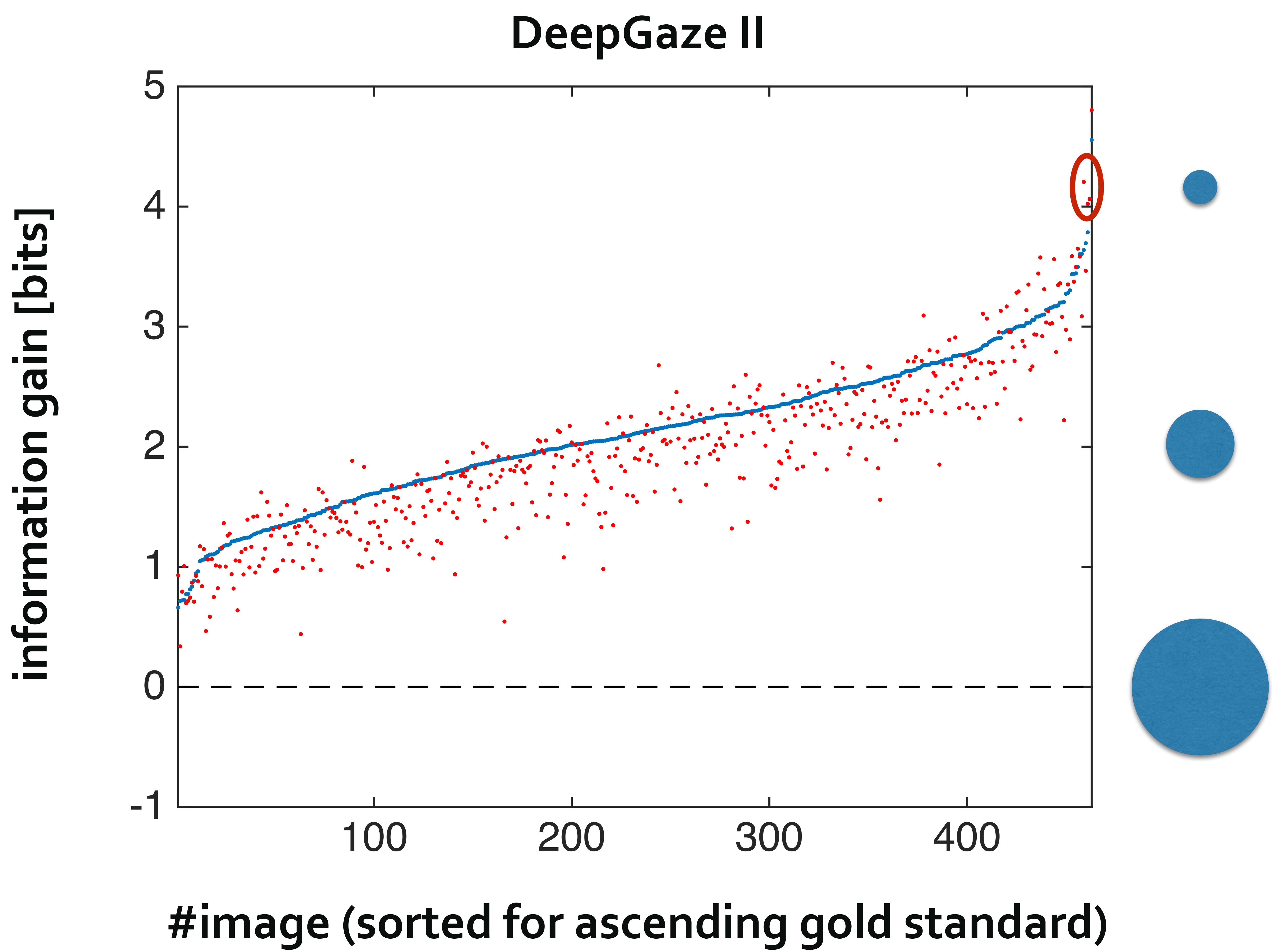
Lucas
Theis


eDN
information gain [bits]
5
4
3
2
1
0
-1
100
200
300
400
#image (sorted for ascending gold standard)

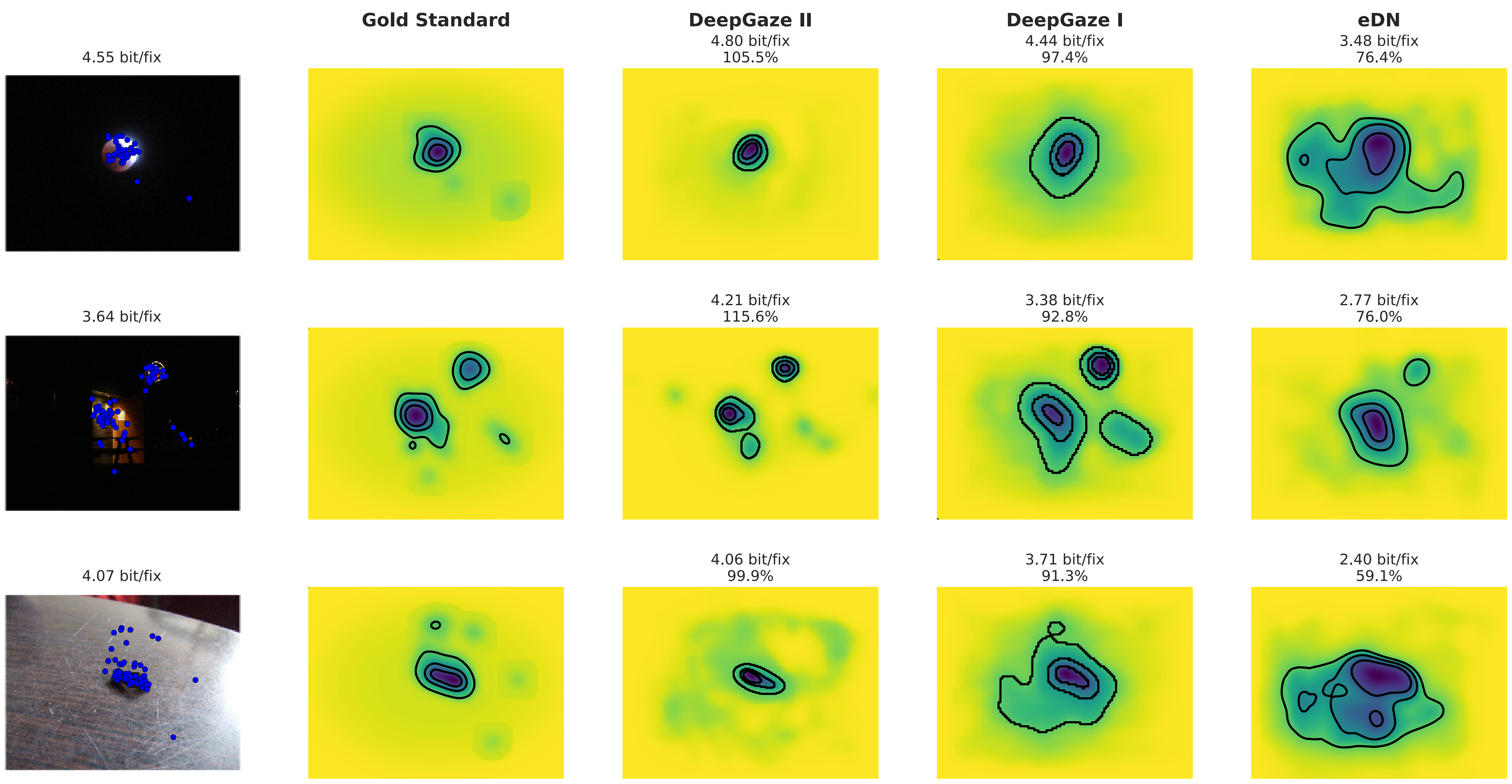


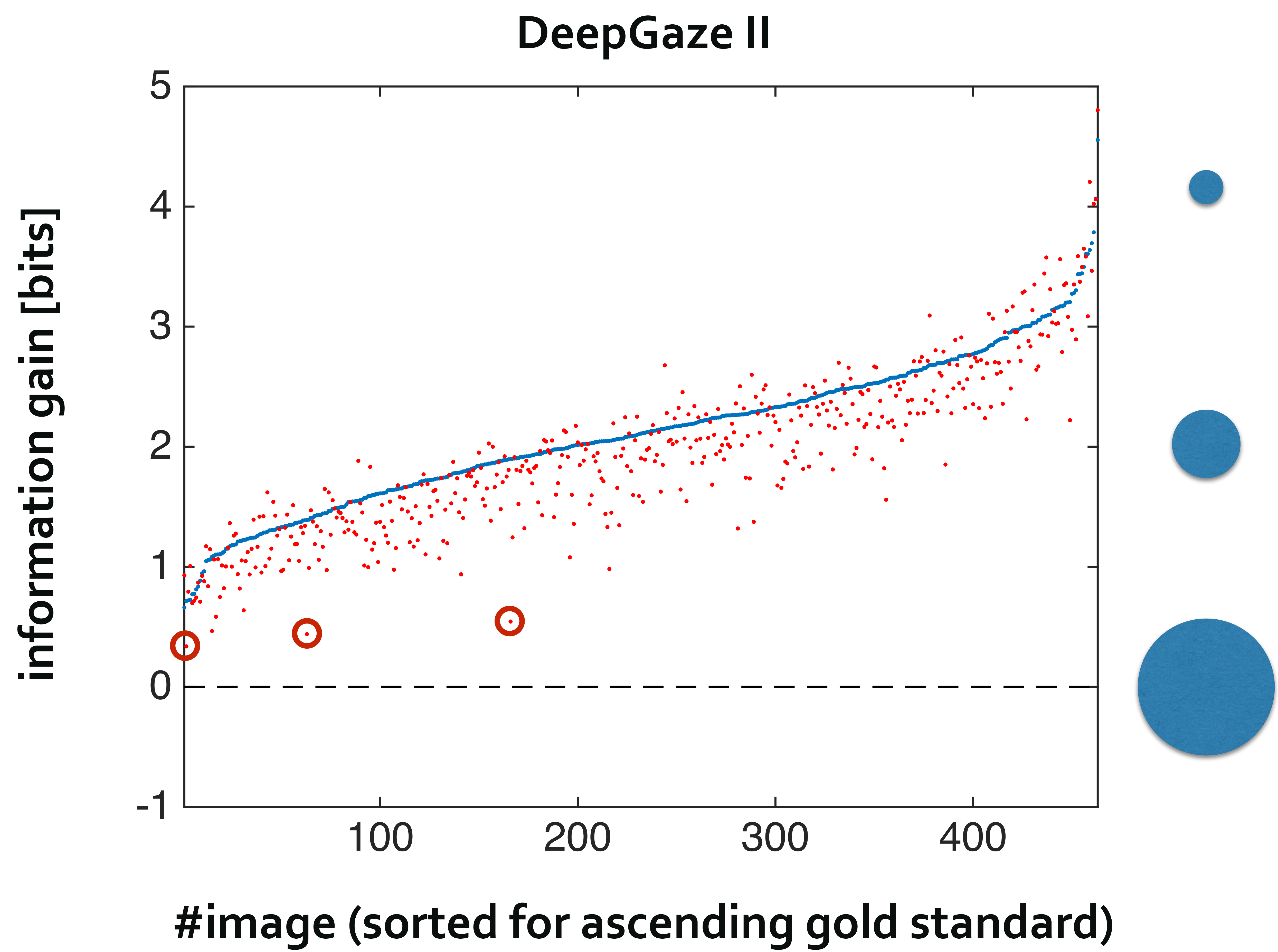
DeepGaze I
information gain [bits]
5
4
3
2
1
0
-1
100
200
300
400
#image (sorted for ascending gold standard)


DeepGaze II
information gain [bits]
5
4
3
2
1
0
-1
100
200
300
400
#image (sorted for ascending gold standard)

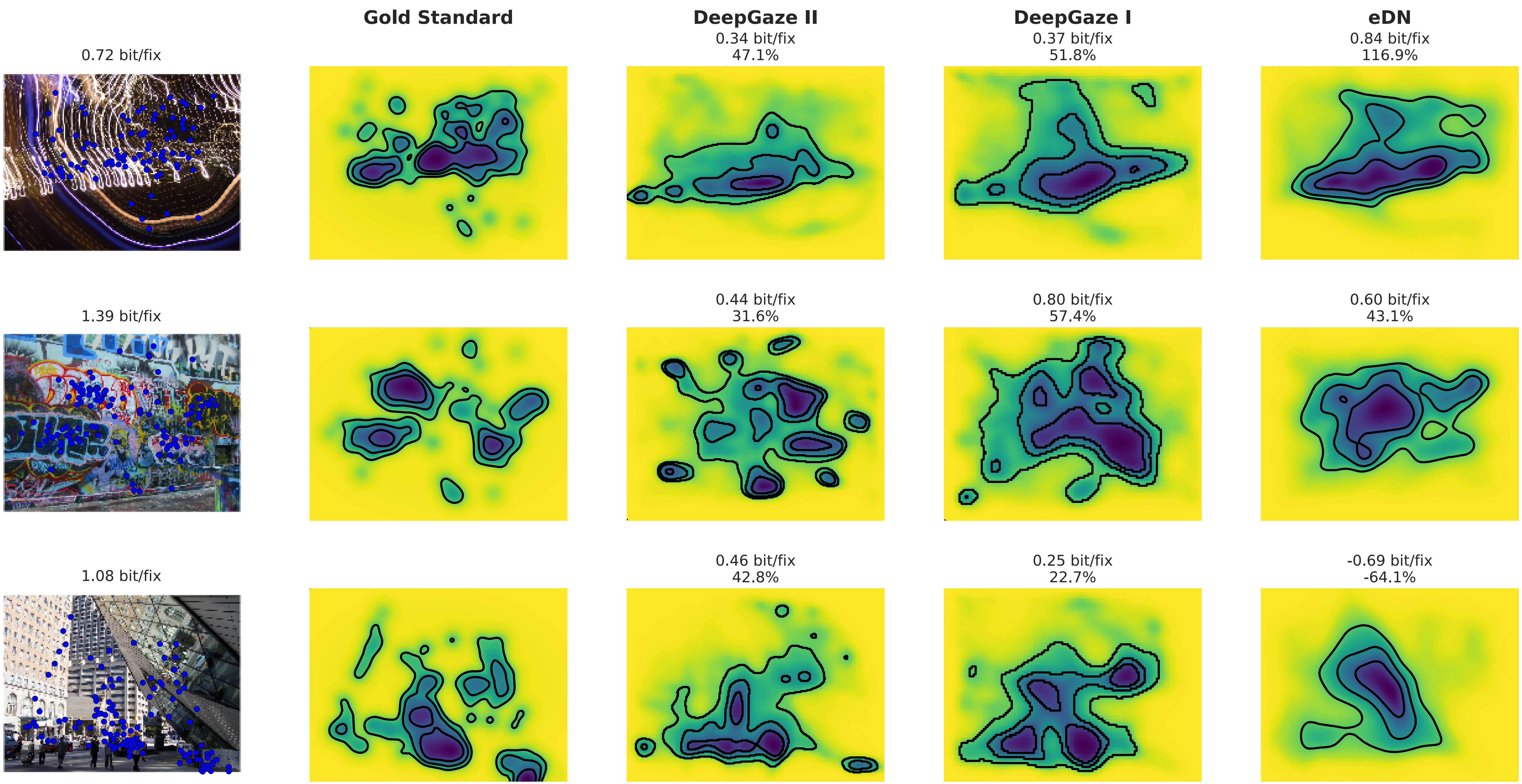

DeepGaze II
information gain [bits]
5
4
3
2
1
0
-1
100
200
300
400
#image (sorted for ascending gold standard)

Images with largest information gain of DeepGaze II

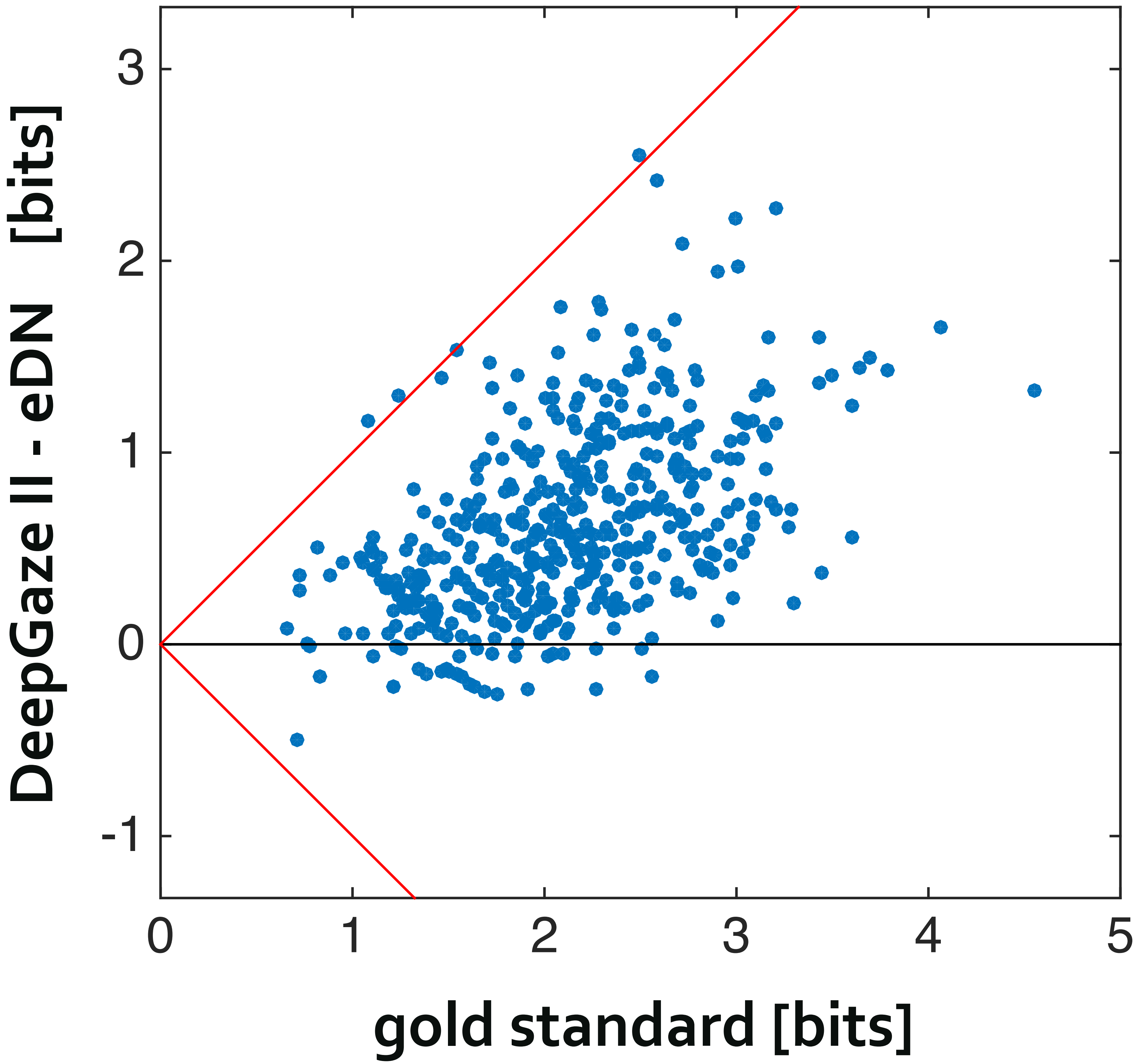



DeepGaze II
information gain [bits]
5
4
3
2
1
0
-1
100
200
300
400
#image (sorted for ascending gold standard)

Images with least information gain of DeepGaze II



Information gain difference between DeepGaze II and eDN

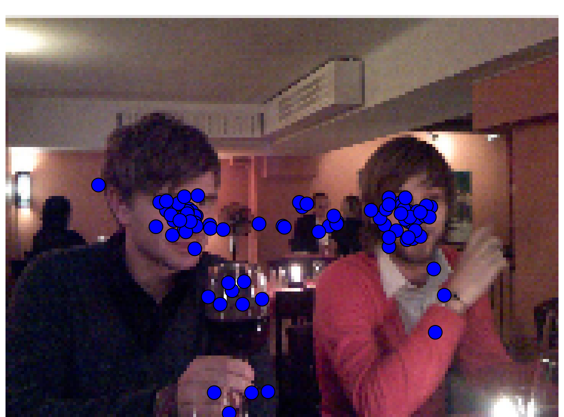


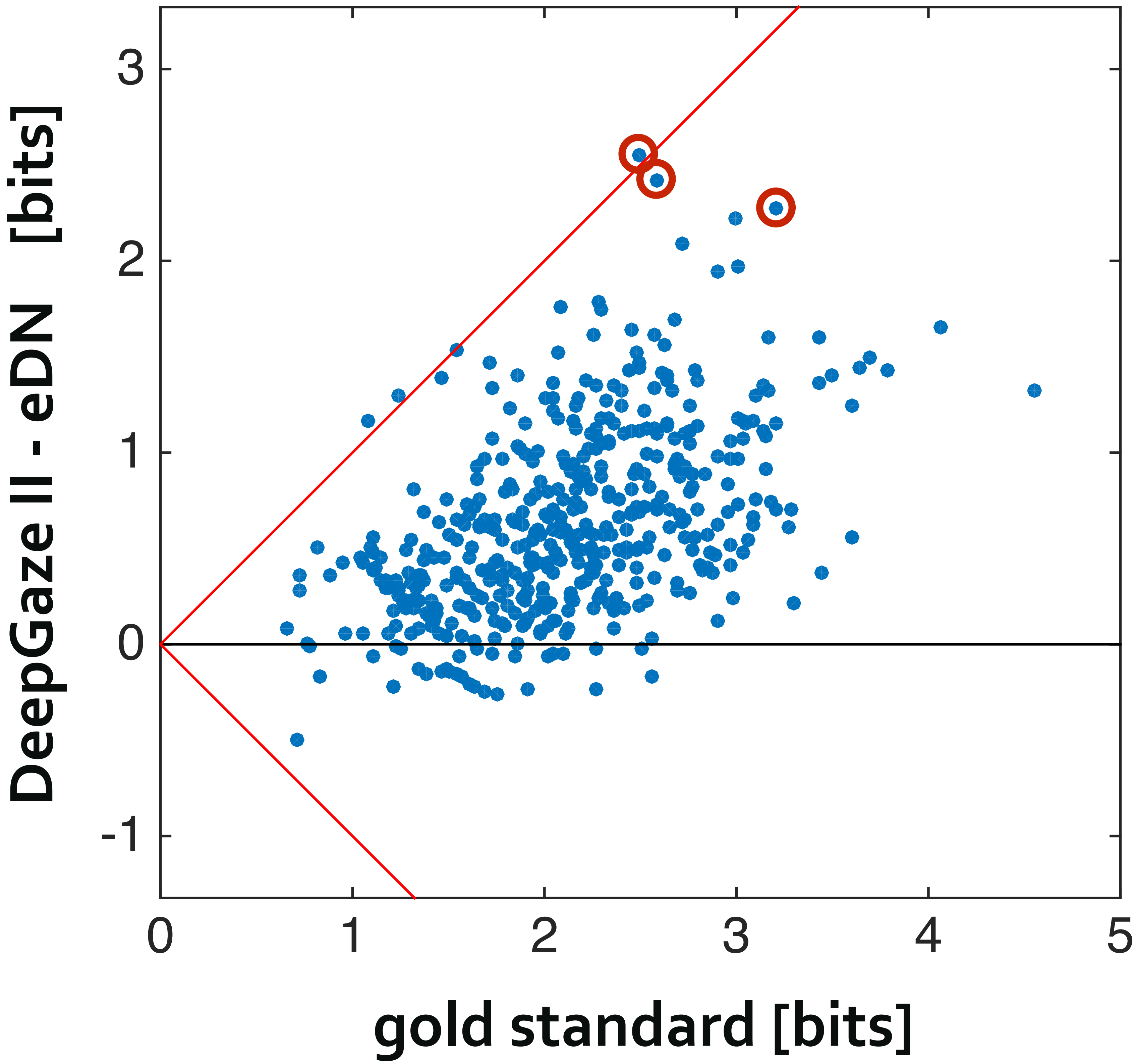
Images with largest improvement of DeepGaze II over eDN

diff=2.55 bit/fix

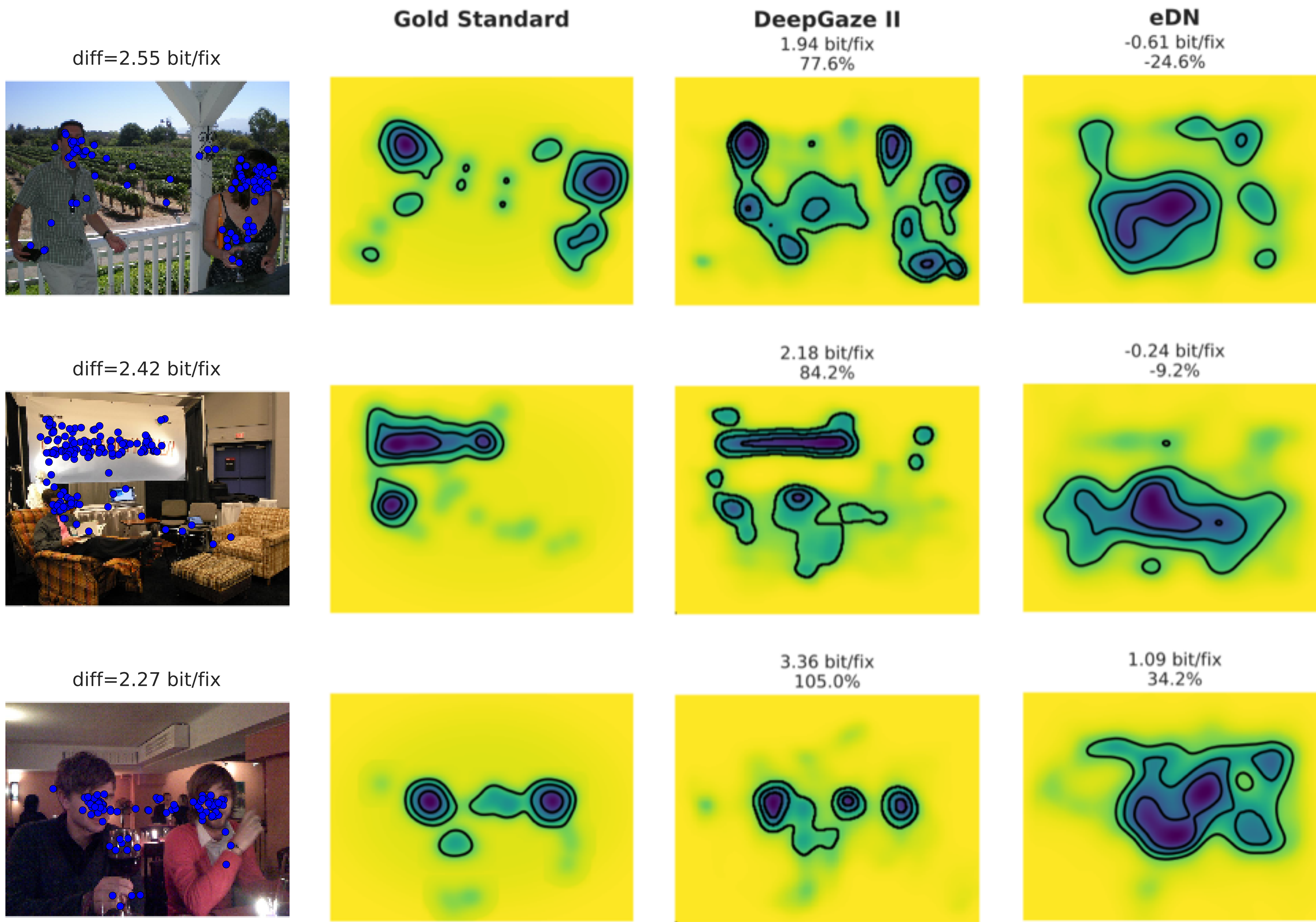
diff=2.42 bit/fix

diff=2.27 bit/fix



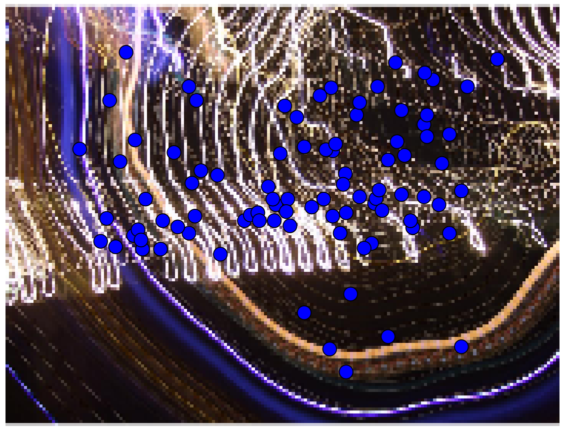


Images with largest improvement of DeepGaze II over eDN

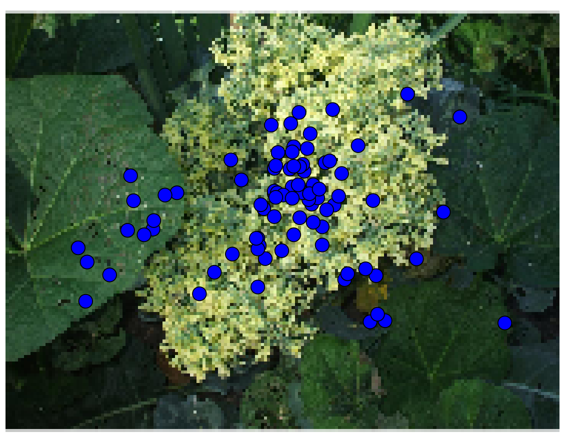


Images with least (negative) improvement of DeepGaze II

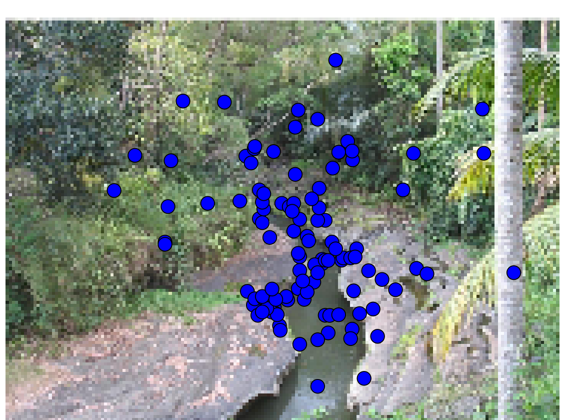
diff=-0.50 bit/fix

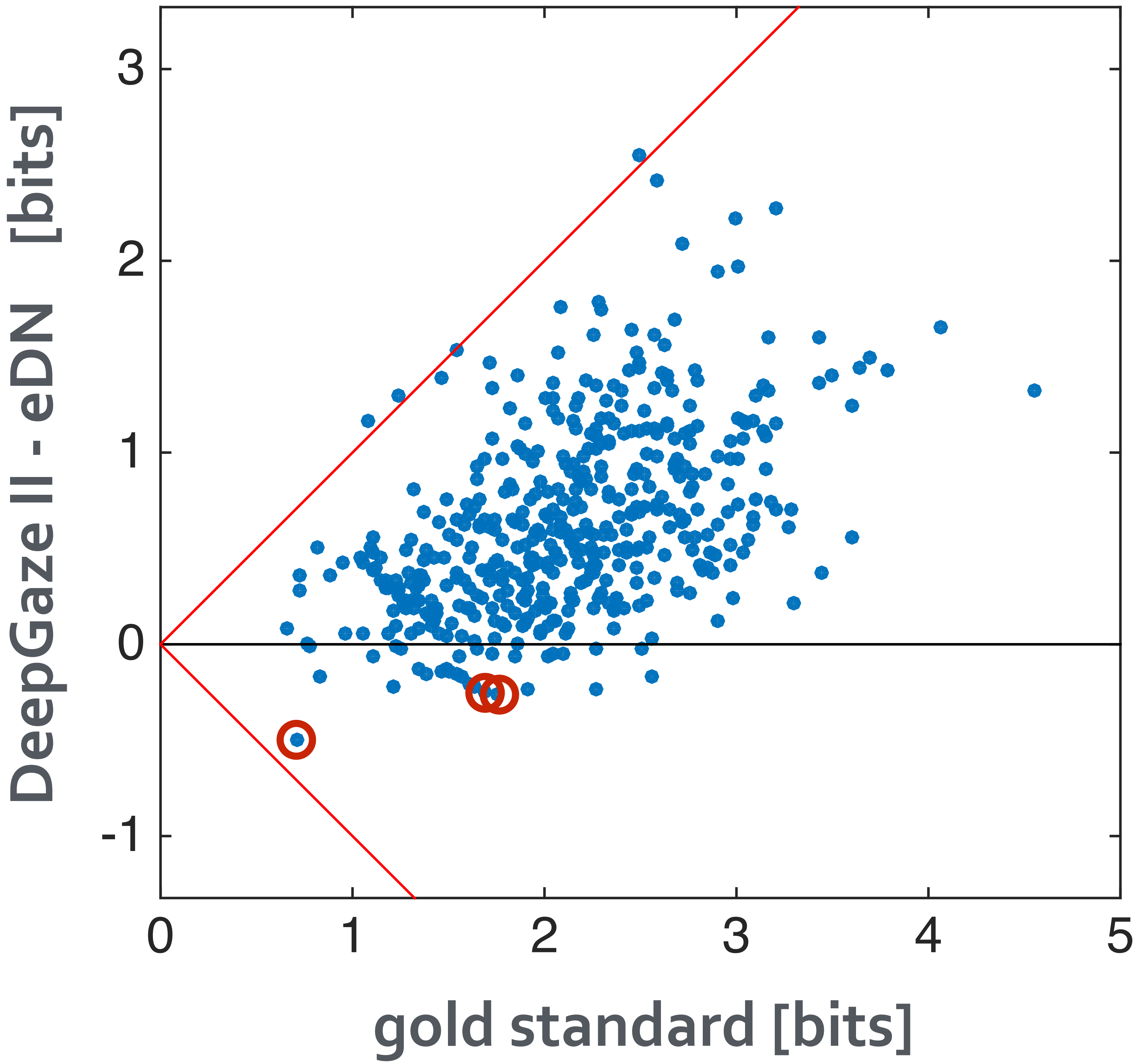


diff=-0.26 bit/fix

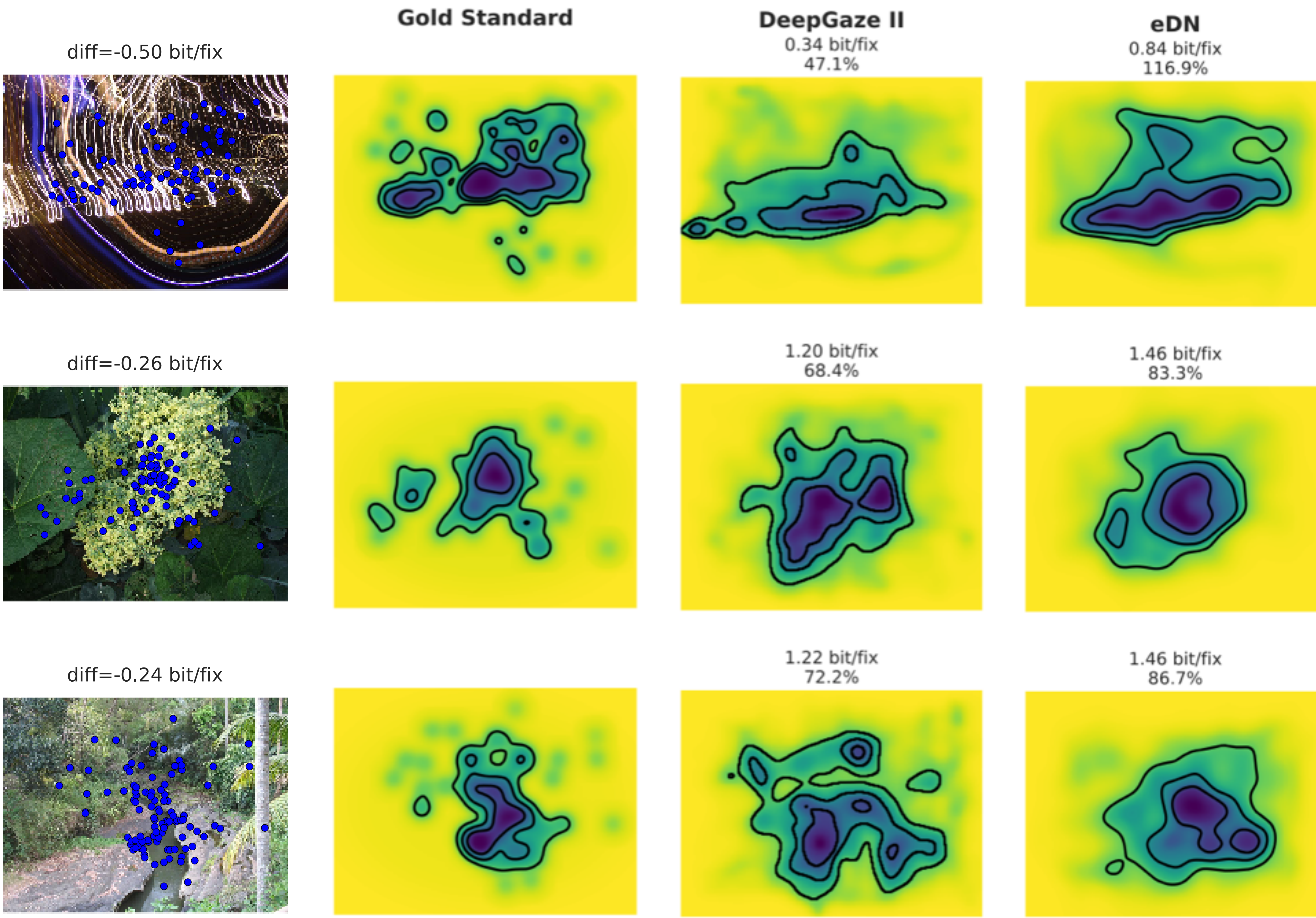


diff=-0.24 bit/fix

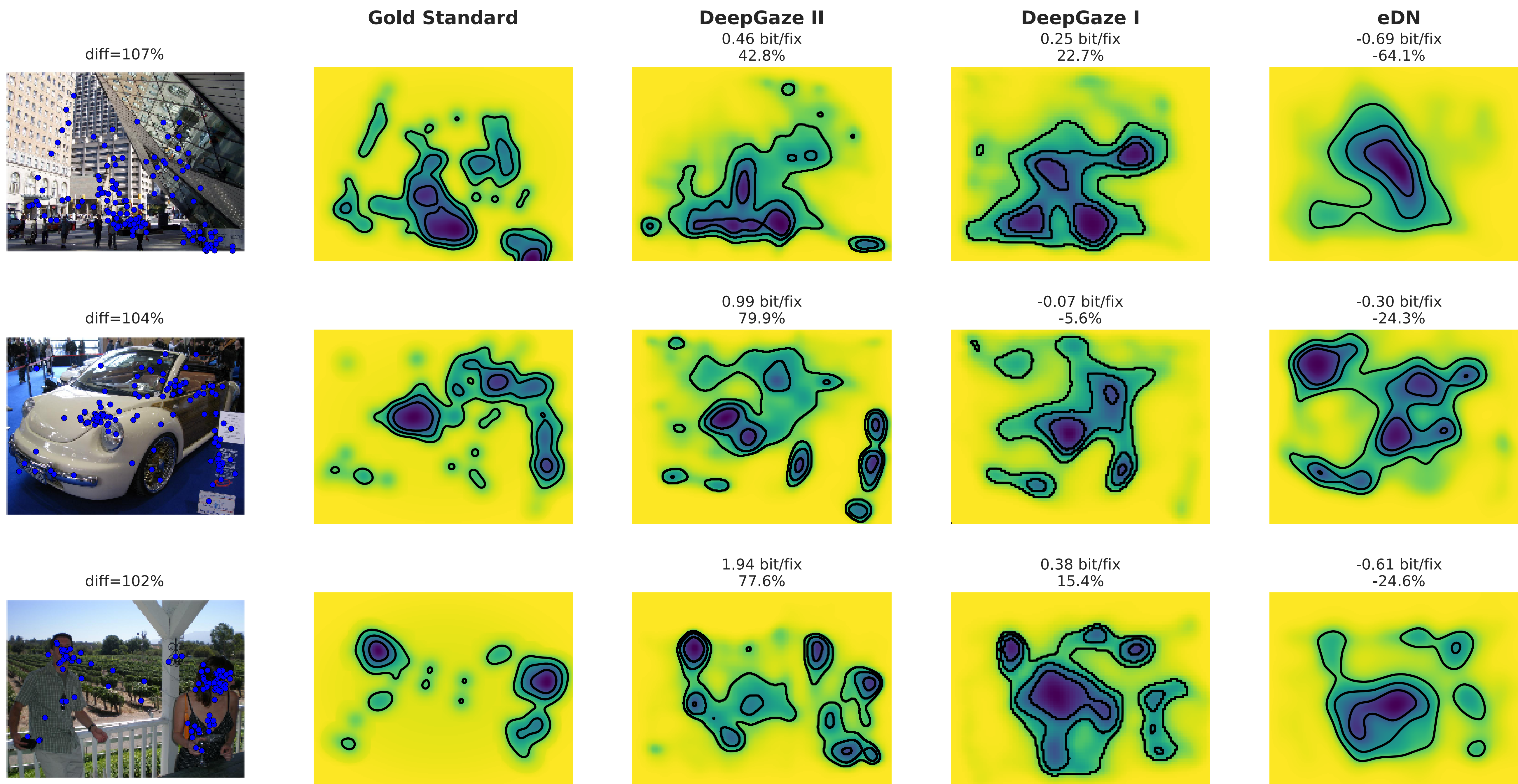




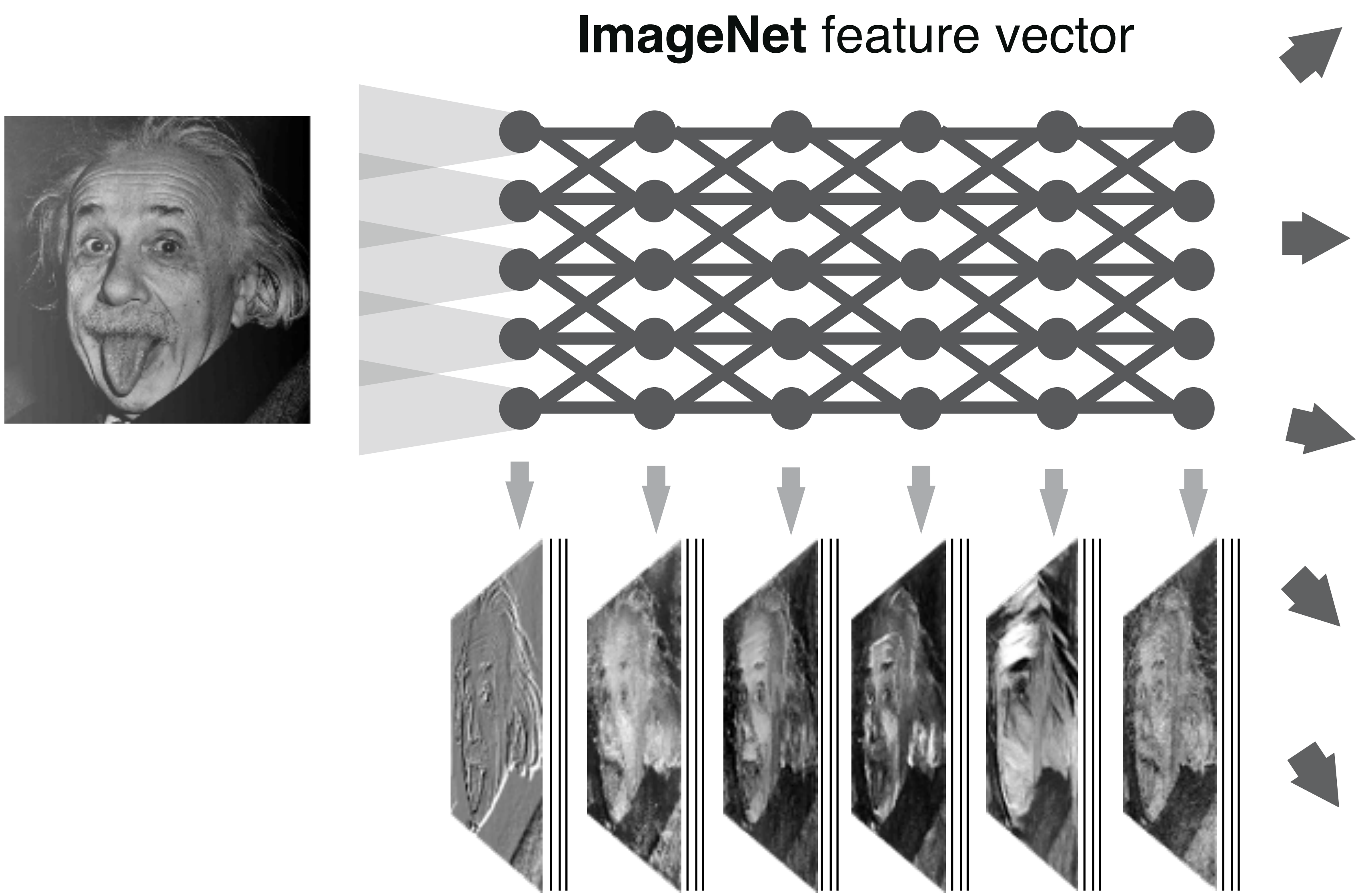
Images with least (negative) improvement of DeepGaze II



Largest improvement in information gain explained



Convnet representations are useful beyond object recognition



saliency prediction
Kuemmerer,Theis,Bethge 2015

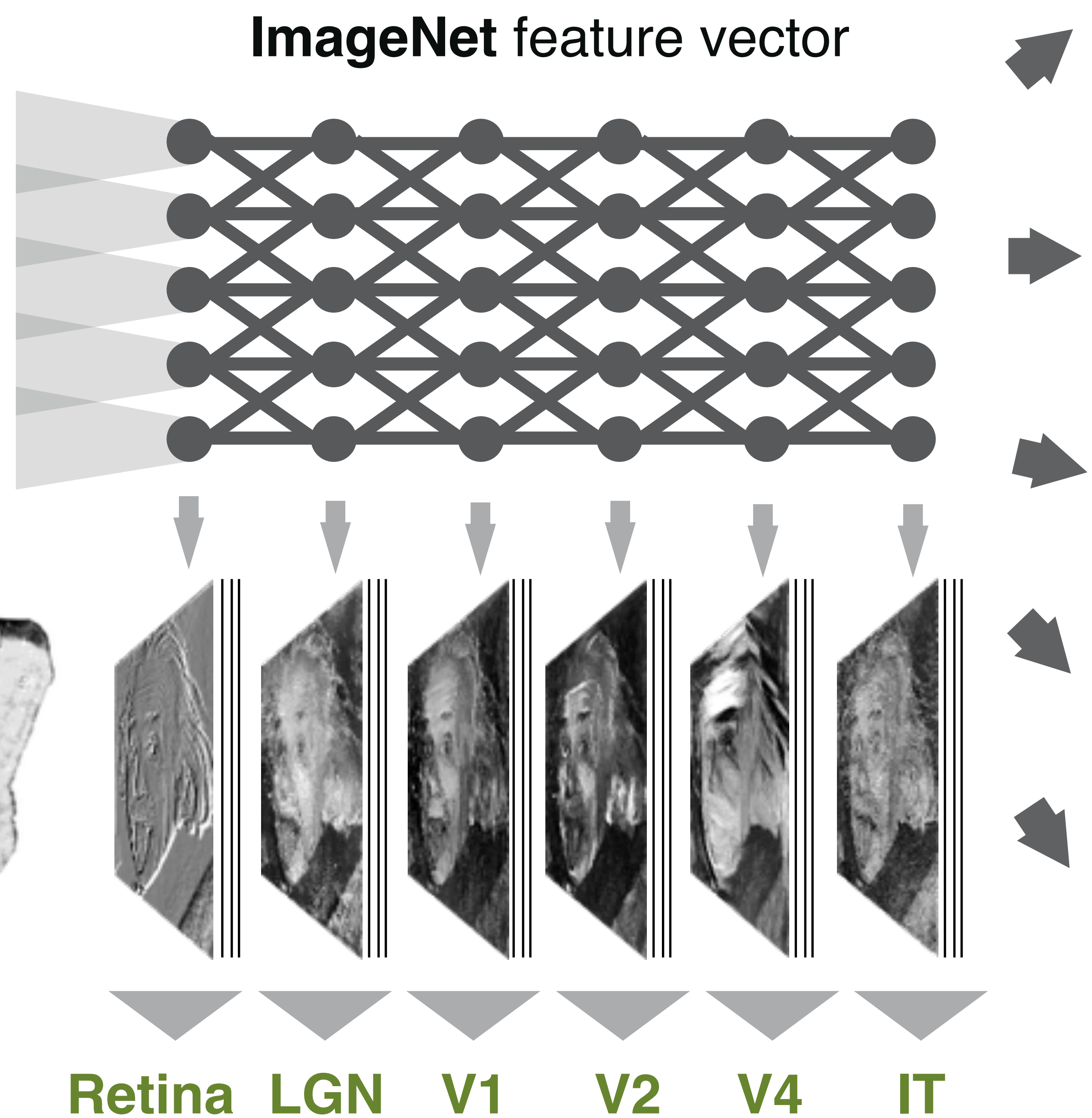
object detection
Krizhevsky et al. 2012
Simonyan et al. 2014

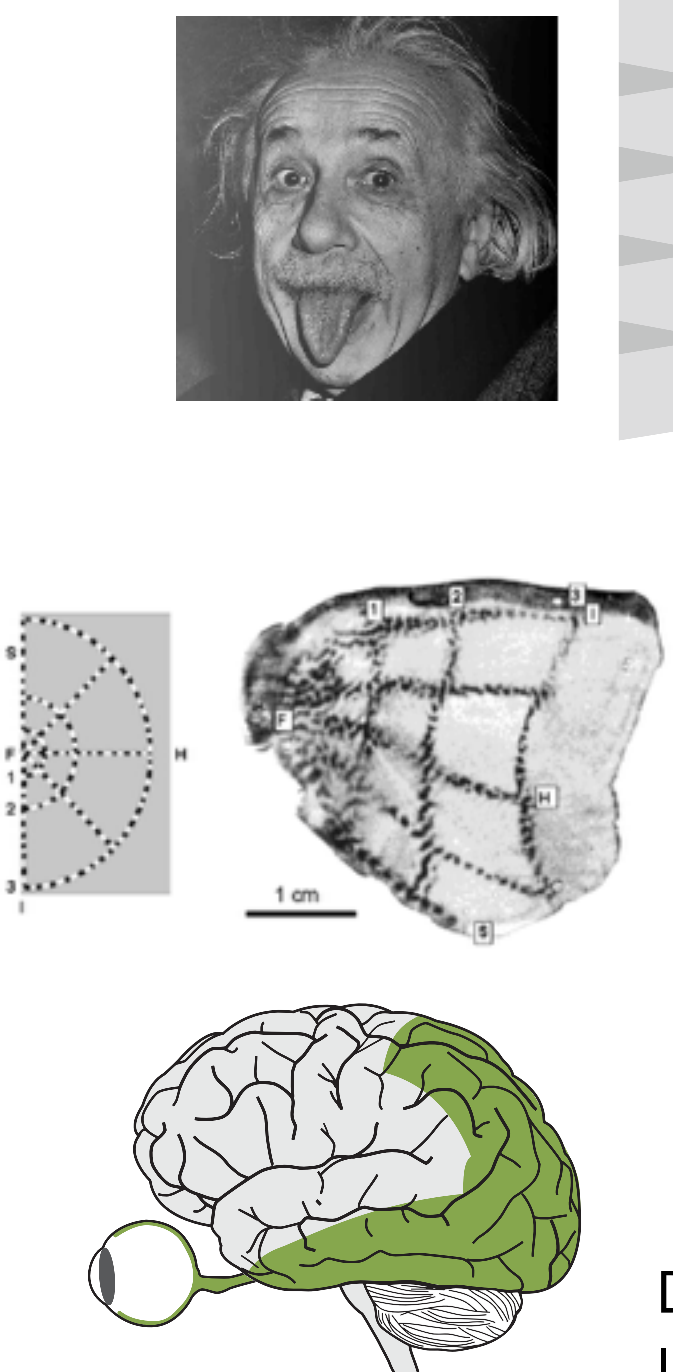
image segmentation
Long et al. 2015
Chen et al, 2015
Berning et al. 2015

depth inference
Eigen et al. 2015

retinal disease detection
Haloi 2015

Convnet representations are useful beyond object recognition





D Yamins et al, PNAS, 2014;
U Güclü and MAJ van Gerven, J Neurosci., 2015

saliency prediction
Kuemmerer,Theis,Bethge 2015

object detection
Krizhevsky et al. 2012
Simonyan et al. 2014

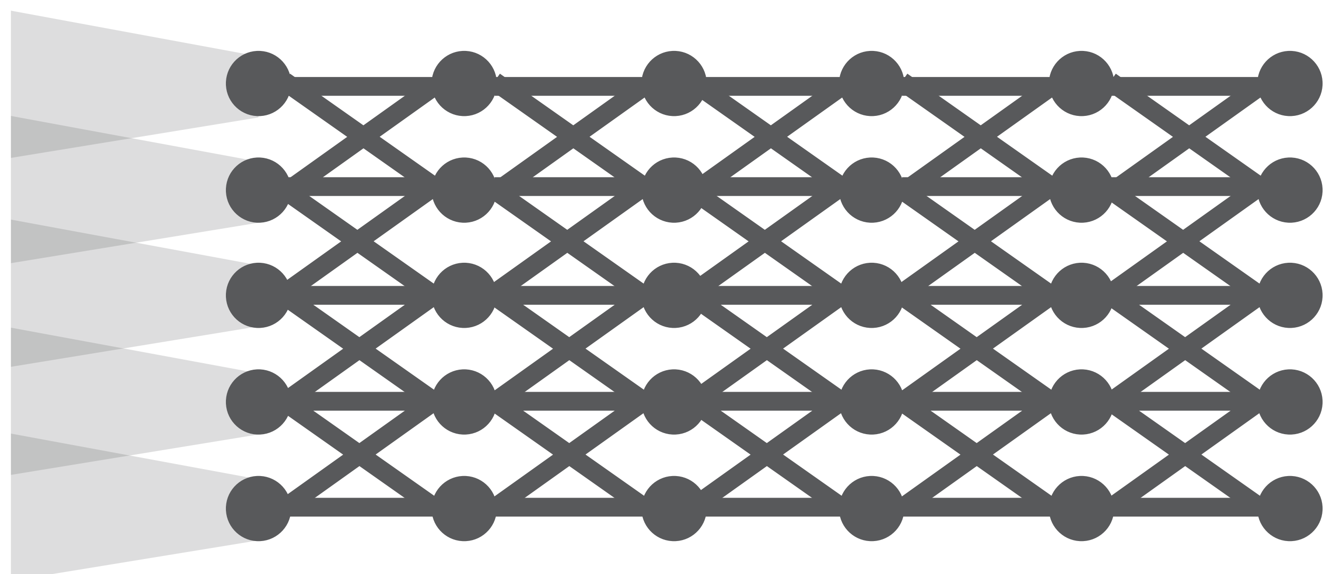
image segmentation
Long et al. 2015
Chen et al, 2015
Berning et al. 2015

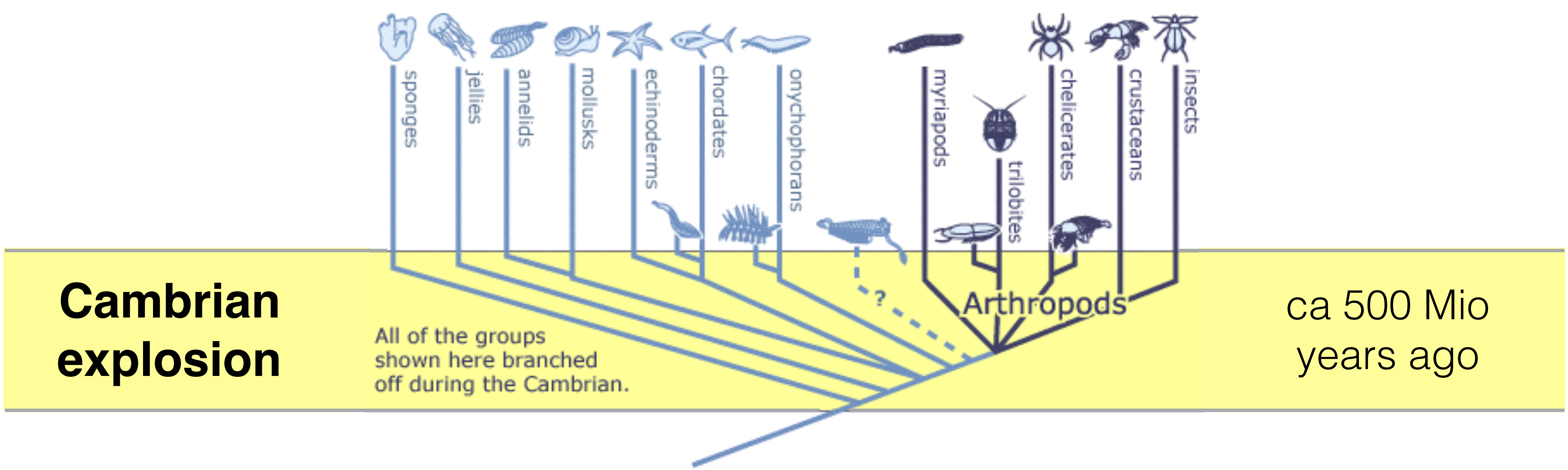
depth inference
Eigen et al. 2015

retinal disease detection
Haloi 2015

neural responses in the ventral stream

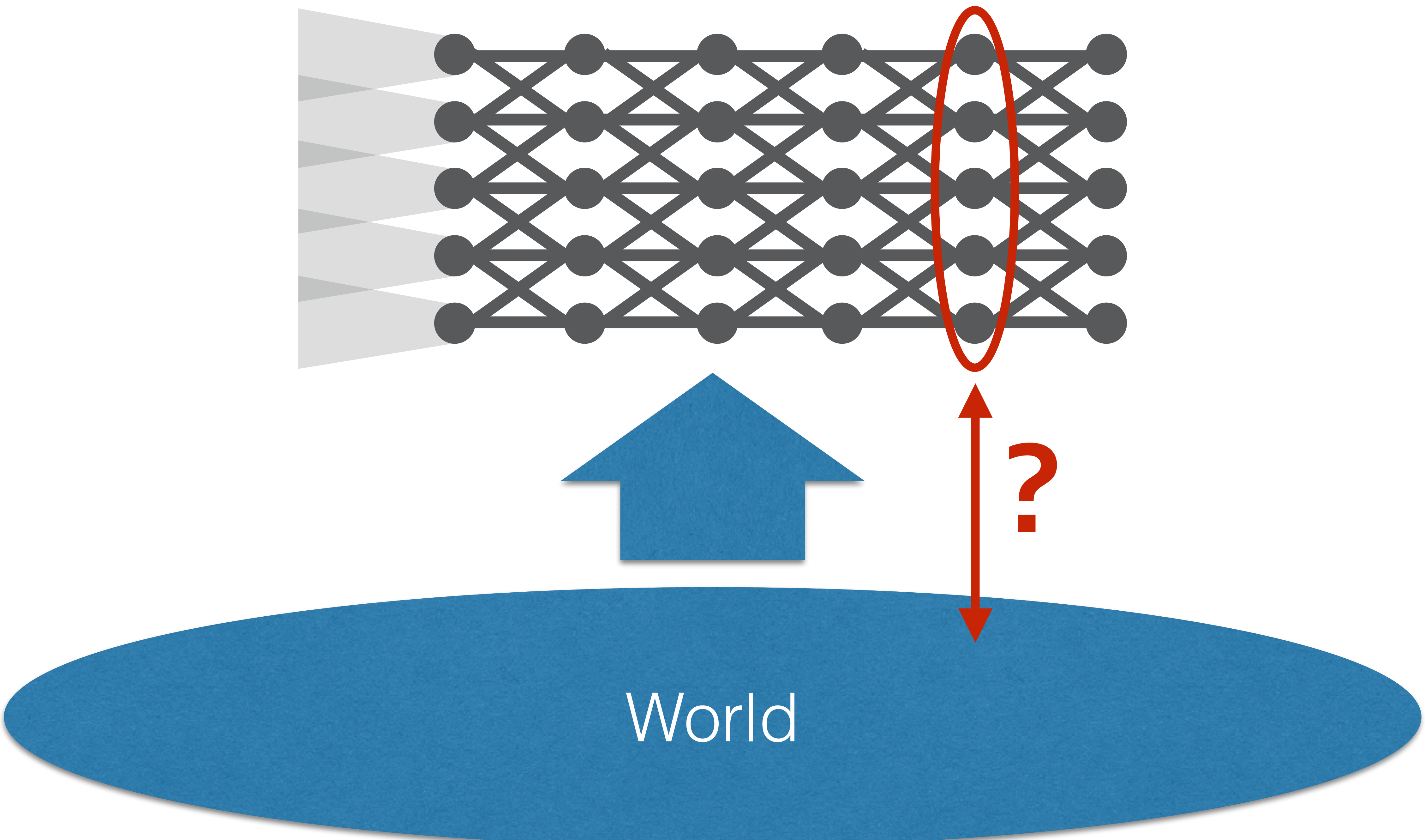
A Cambrian explosion of artificial neural networks



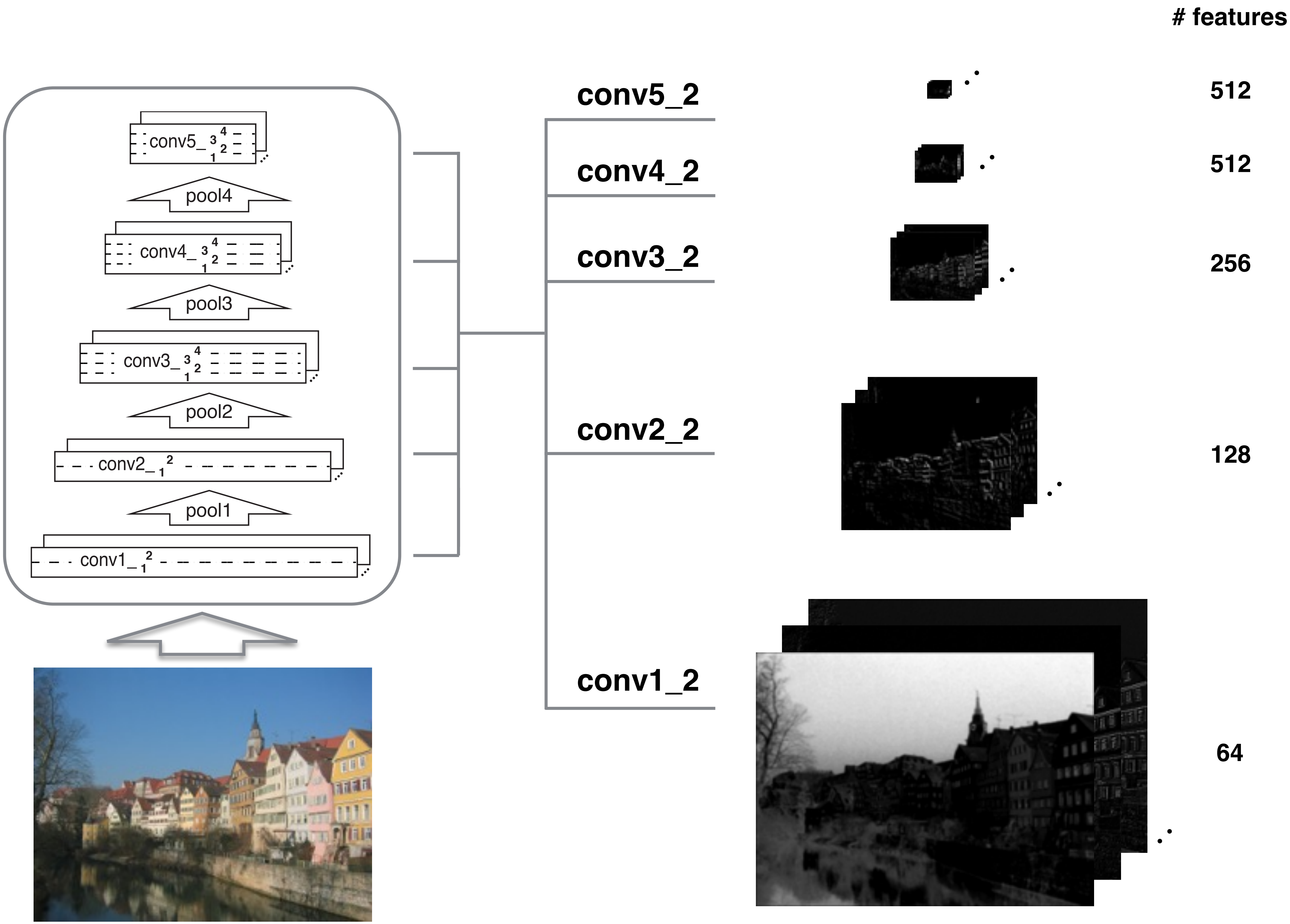


Key question:

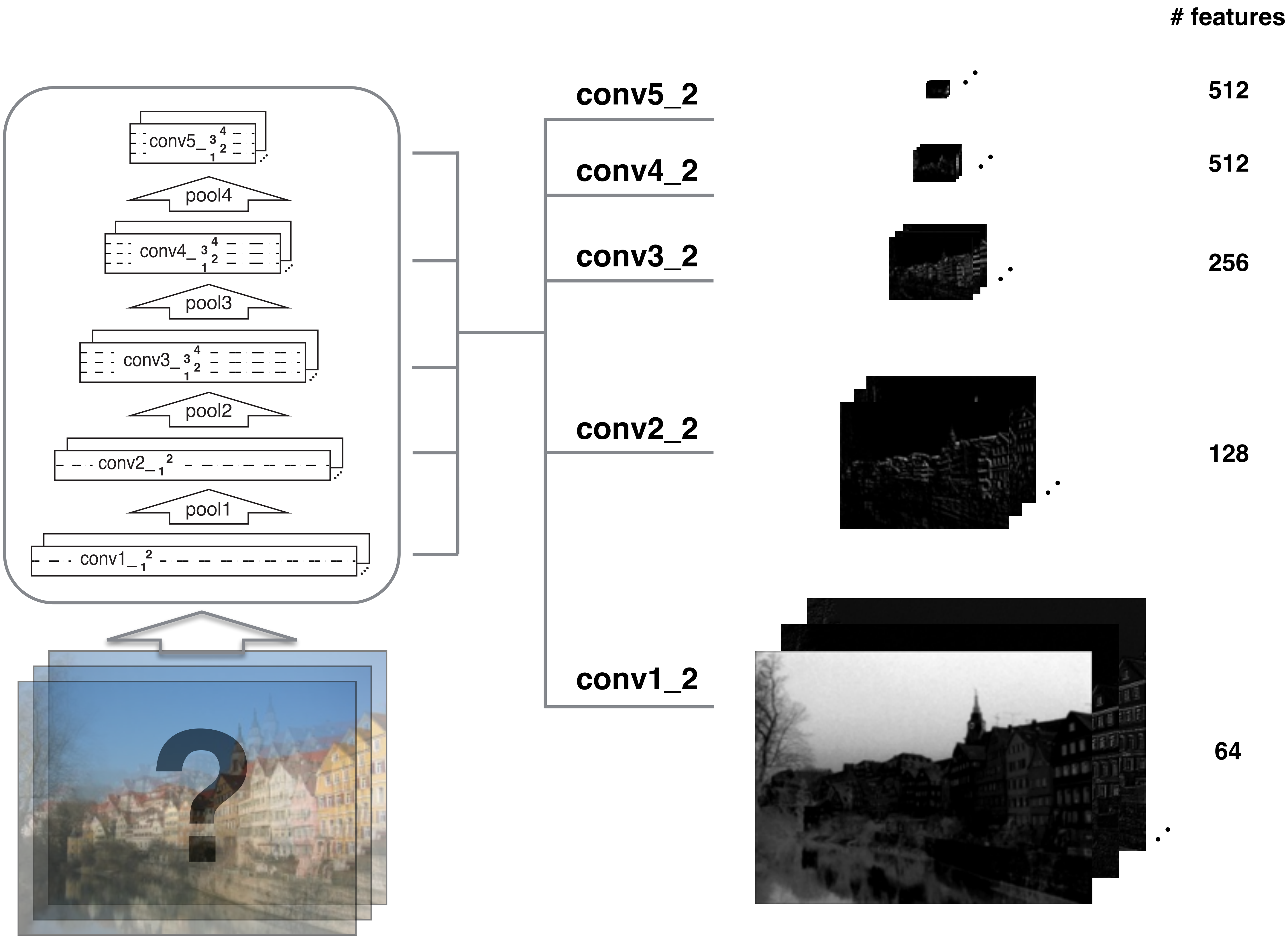
Can we understand how the world is represented in artificial neural networks?



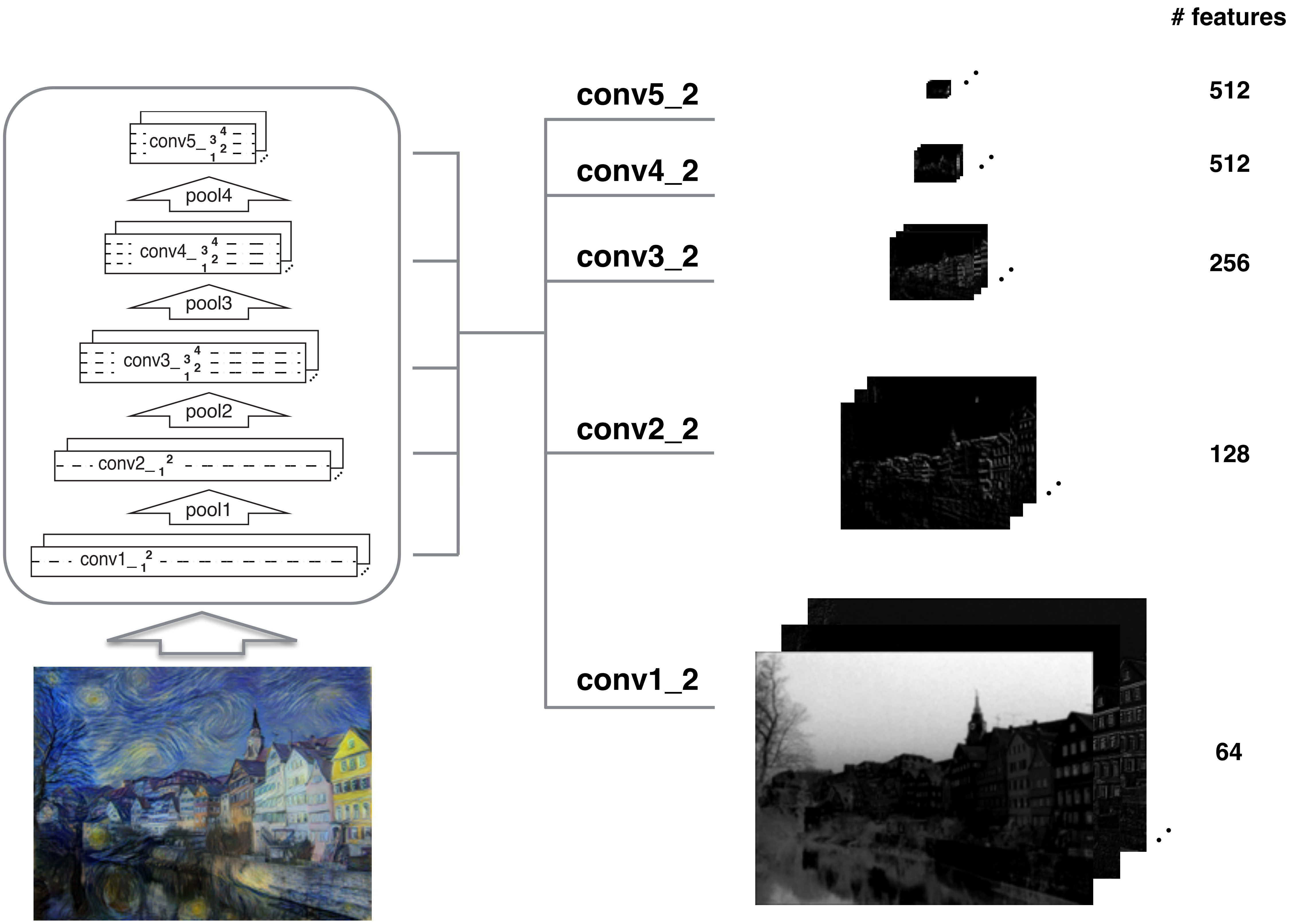
Representation of stimulus information



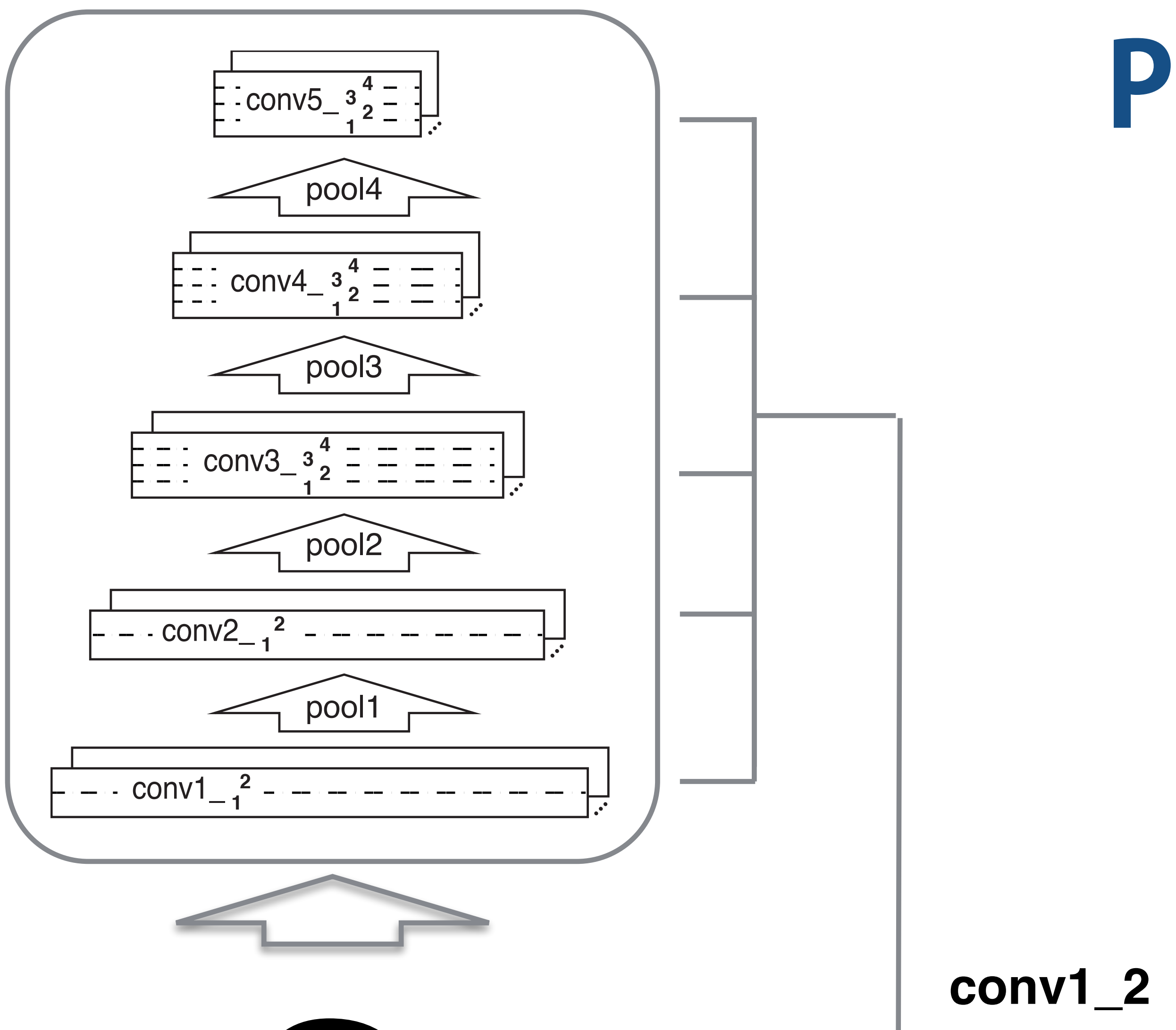
Representation of stimulus information



Representation of stimulus information



Representation of stimulus information



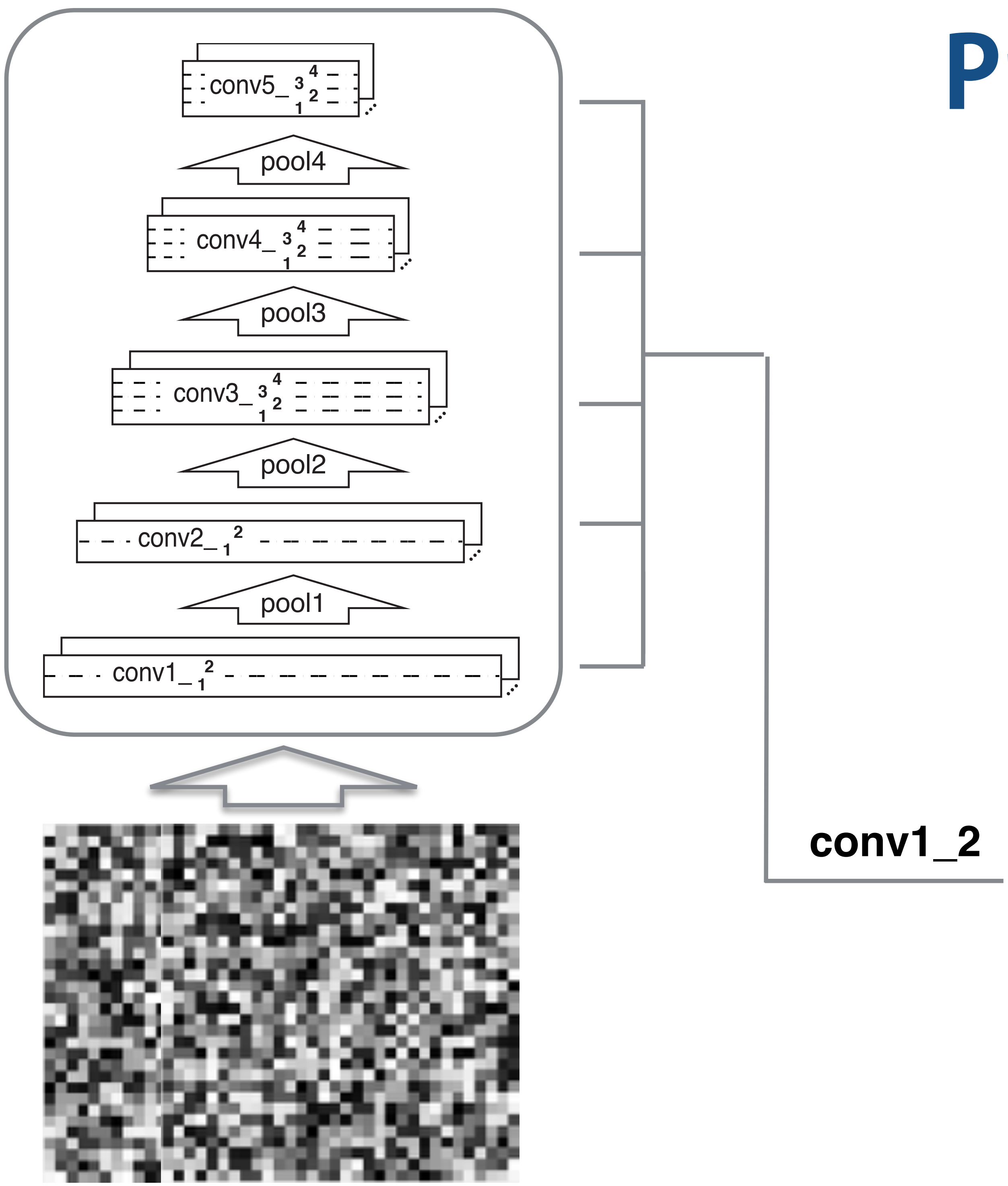
Pre-image search

$$y_0 = f(x_0)$$

?

$$\{x \in R^n : ||f(x) - y_0|| < \epsilon\}$$

Representation of stimulus information



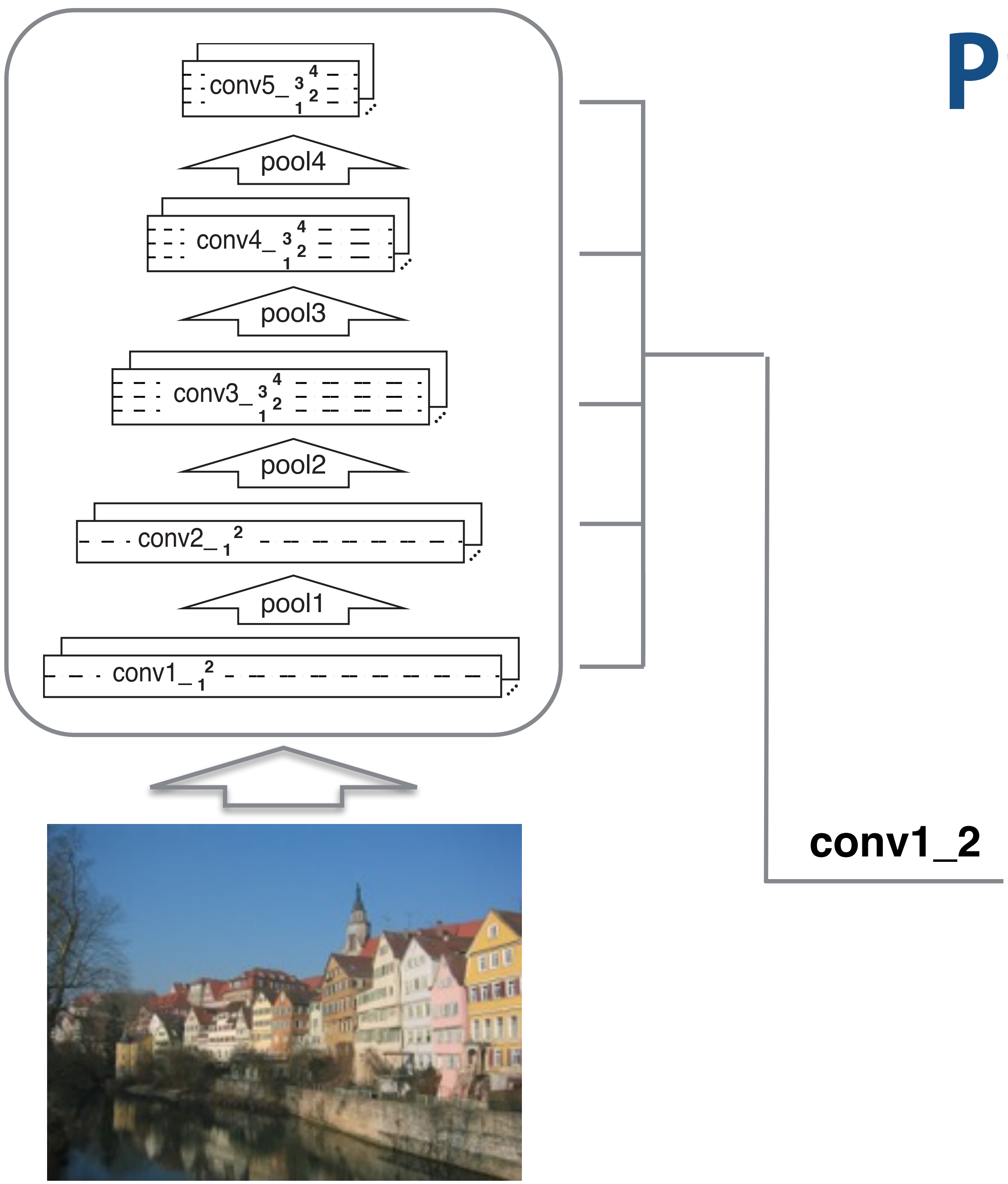
Pre-image search

$$\nabla_x ||f(x) - y_0||^2$$

$$y_0 = f(x_0)$$

64

Representation of stimulus information

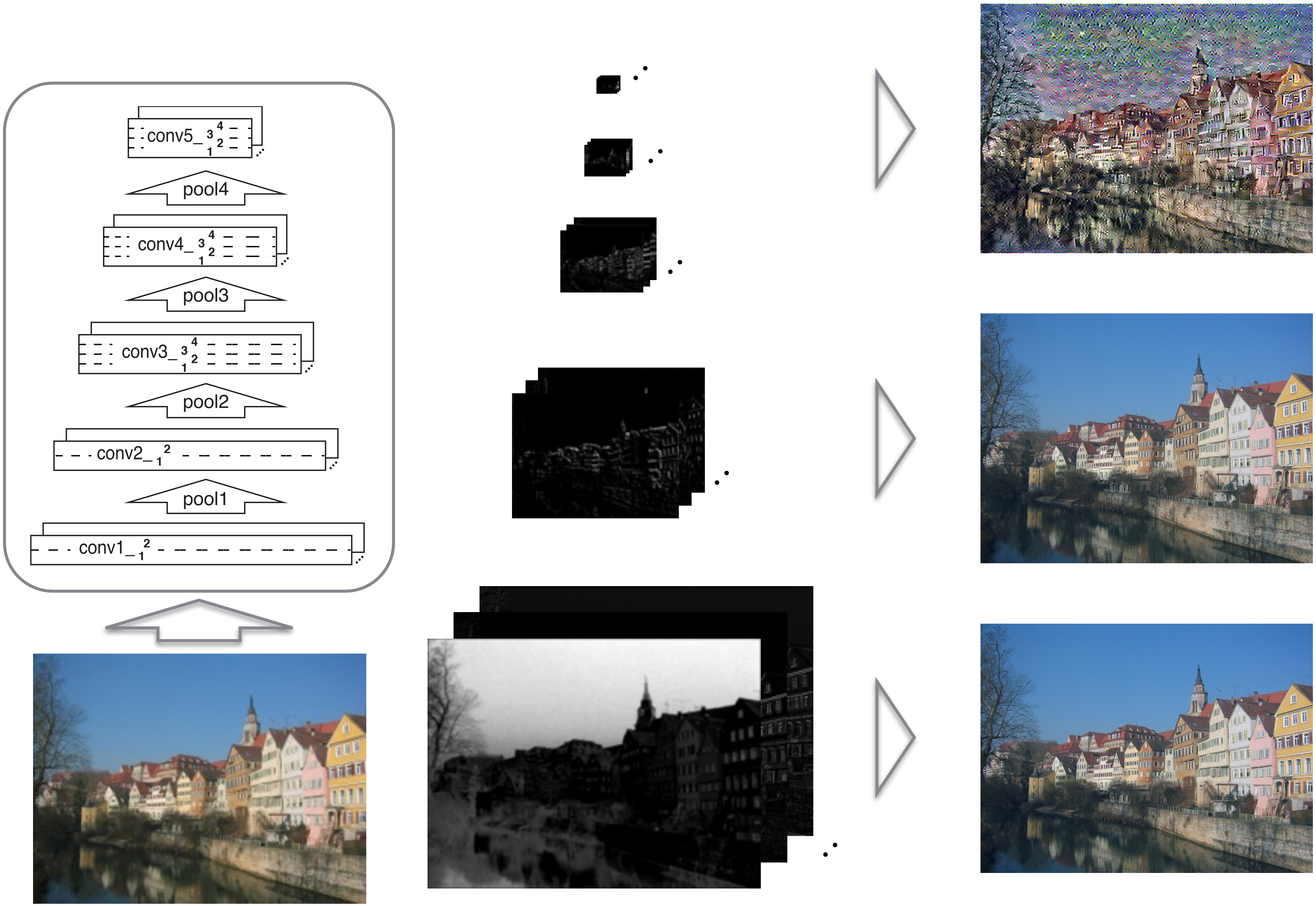


Pre-image search

$$\nabla_x ||f(x) - y_0||^2$$

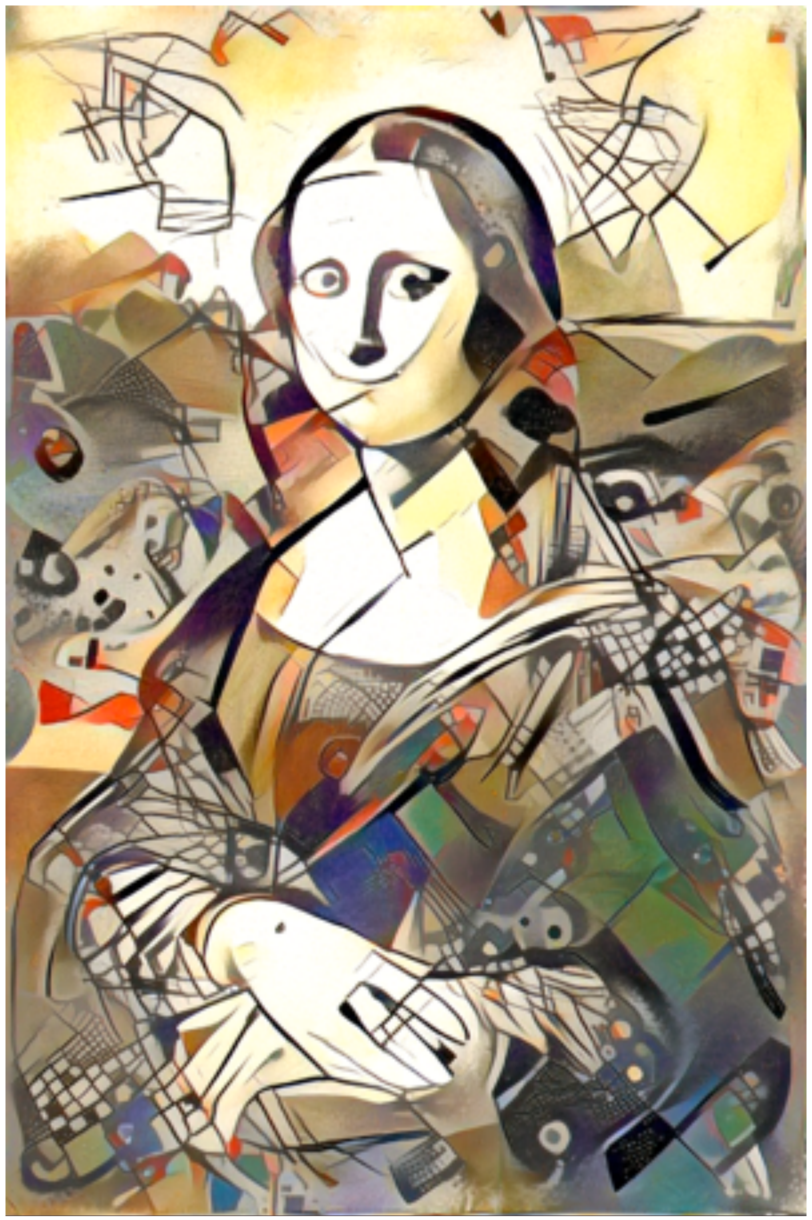
$$y_0 = f(x_0)$$

Representation of stimulus information



Mahendran & Vedaldi (2014), Gatys et al (2015)

Neural Image Representations



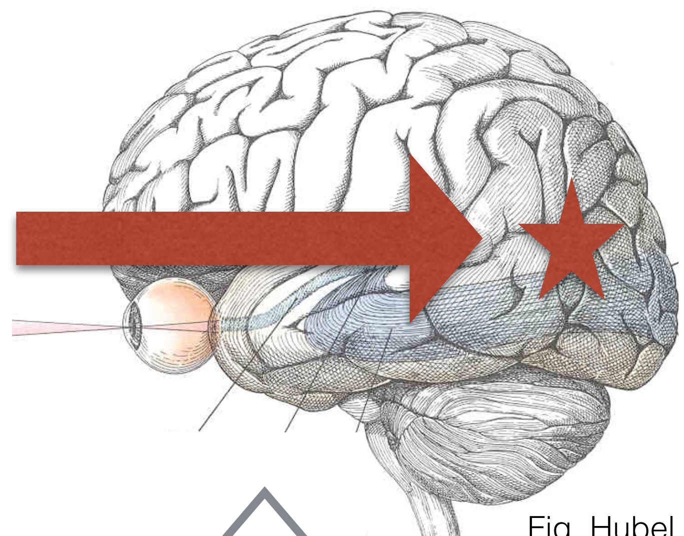


Fig. Hubel 95

Semantic Image Understanding

Correspondence

Yamins 2014, Cadieu 2014,
van Gerven 2015

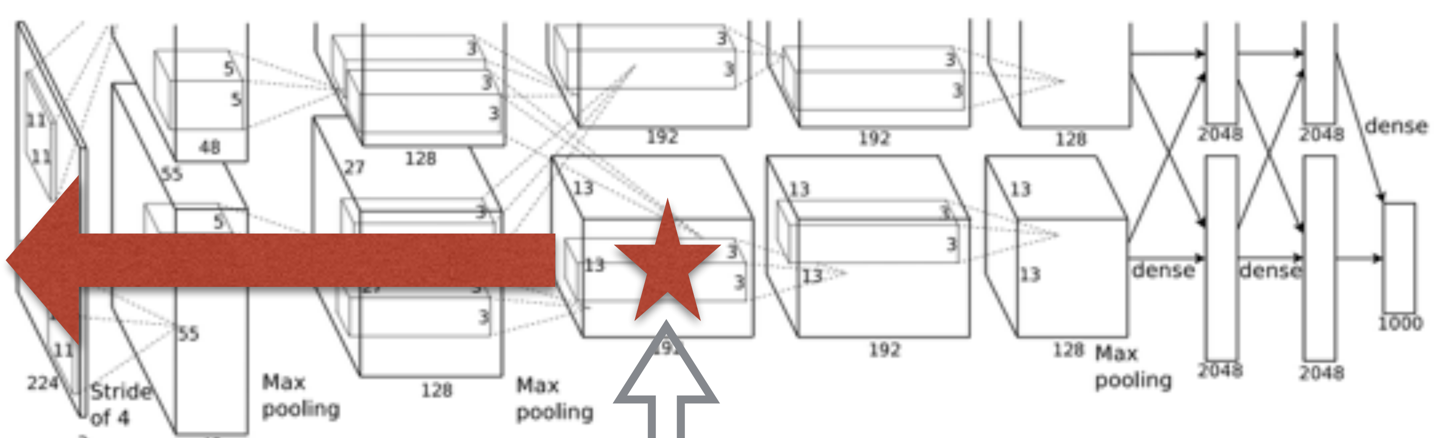
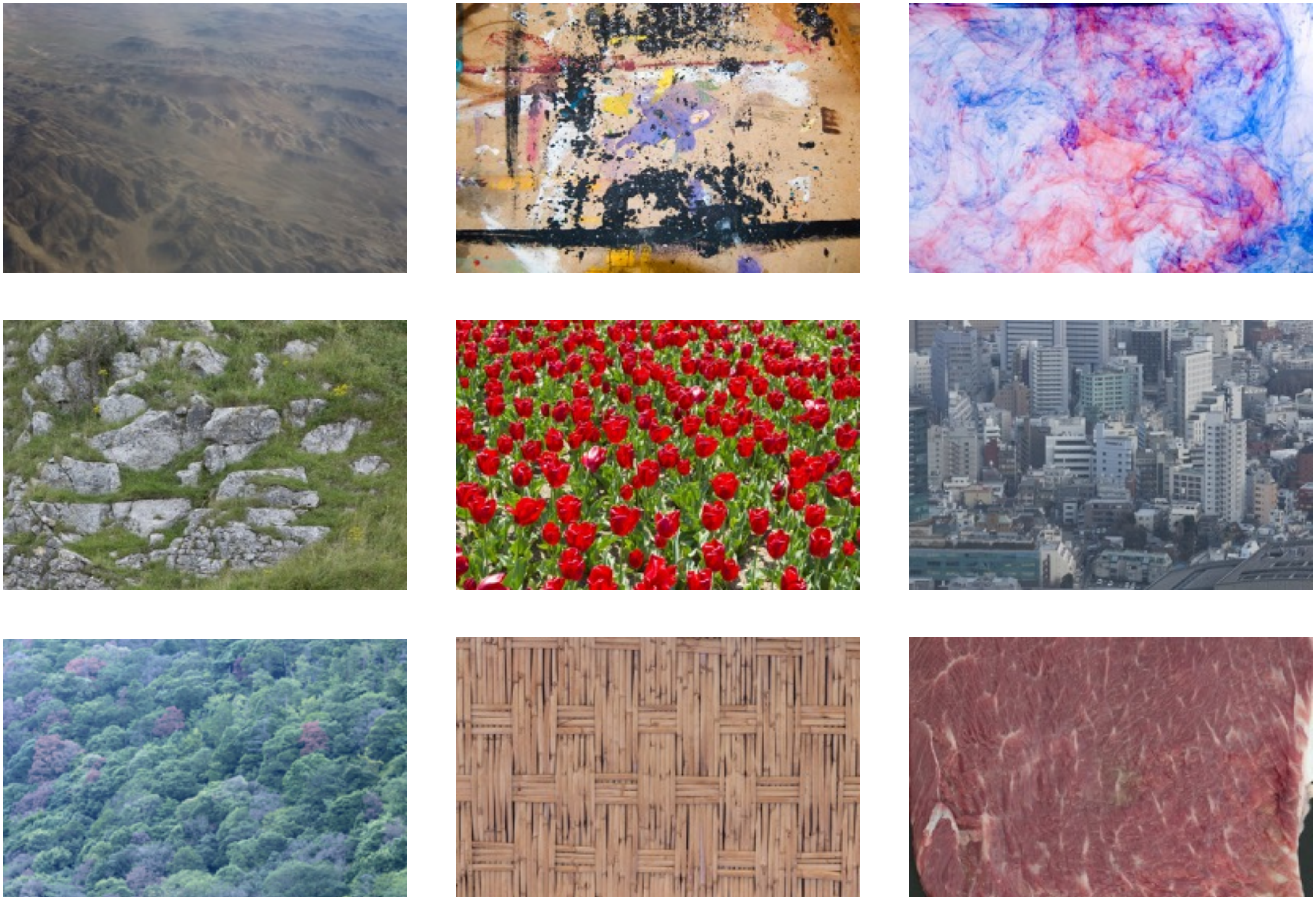


Fig. Krizhevsky 2012

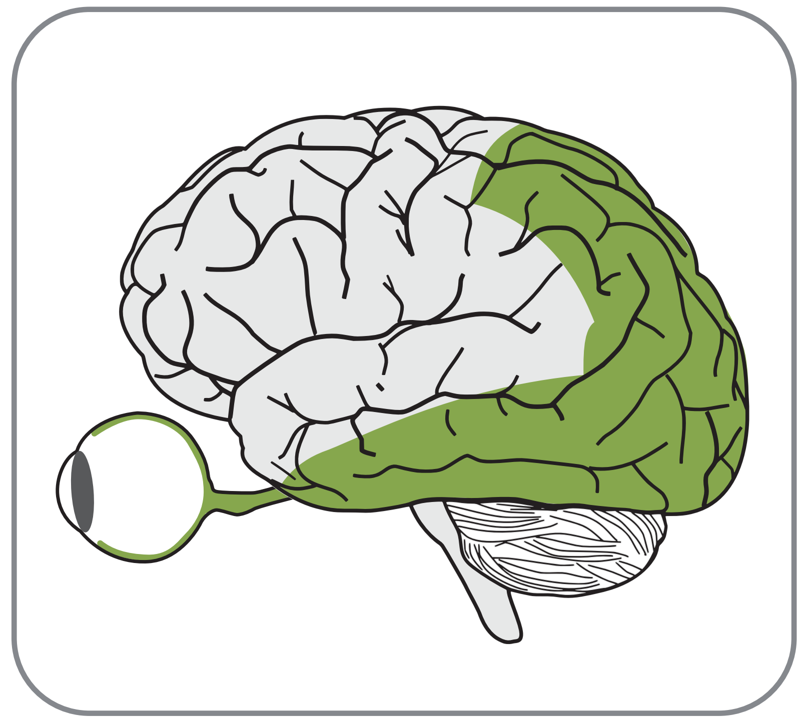
Induce Change

Visual Textures



Julesz' Conjecture

Texture Sample



All textures producing the same measurement outcomes should be perceived as the same texture!

Julesz (1962)

Julesz' Conjecture

Texture Sample

Outcomes

Set of statistical measurements

$$\{\Phi_1(\vec{x}), \Phi_2(\vec{x}), \ldots, \Phi_N(\vec{x})\}$$

taken over spatial extend of an image

2.3
7.1
3.8
⋮

All textures producing the same measurement outcomes should be perceived as the same texture!

Julesz (1962)

Julesz' Conjecture

New Texture Sample

Set of statistical measurements

$$\{\Phi_1(\vec{x}), \Phi_2(\vec{x}), \ldots, \Phi_N(\vec{x})\}$$

taken over spatial extend of an image

Outcomes

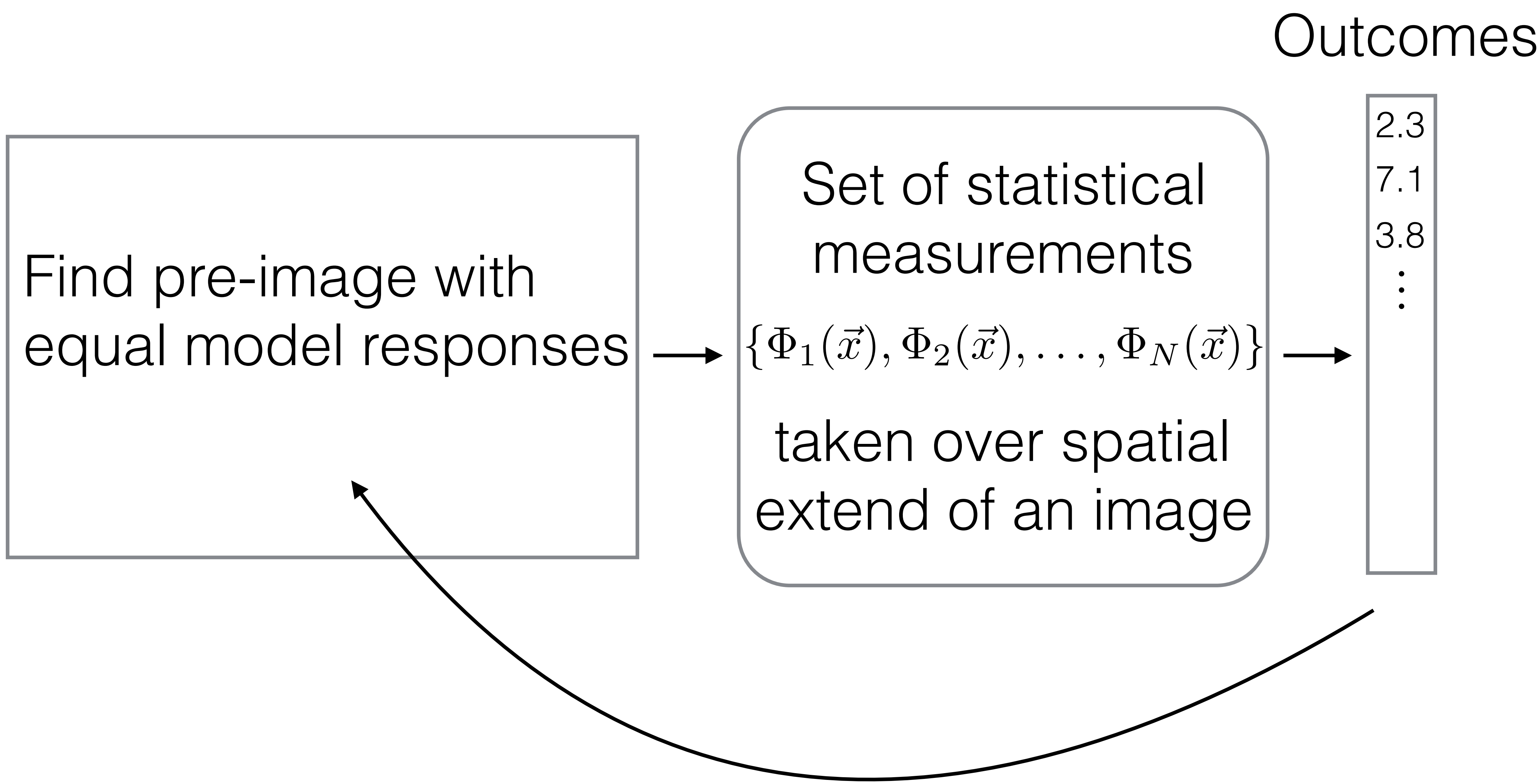
2.3
7.1
3.8
⋮

All textures producing the same measurement outcomes should be perceived as the same texture!

Julesz (1962)

Parametric Texture Synthesis

Synthesis

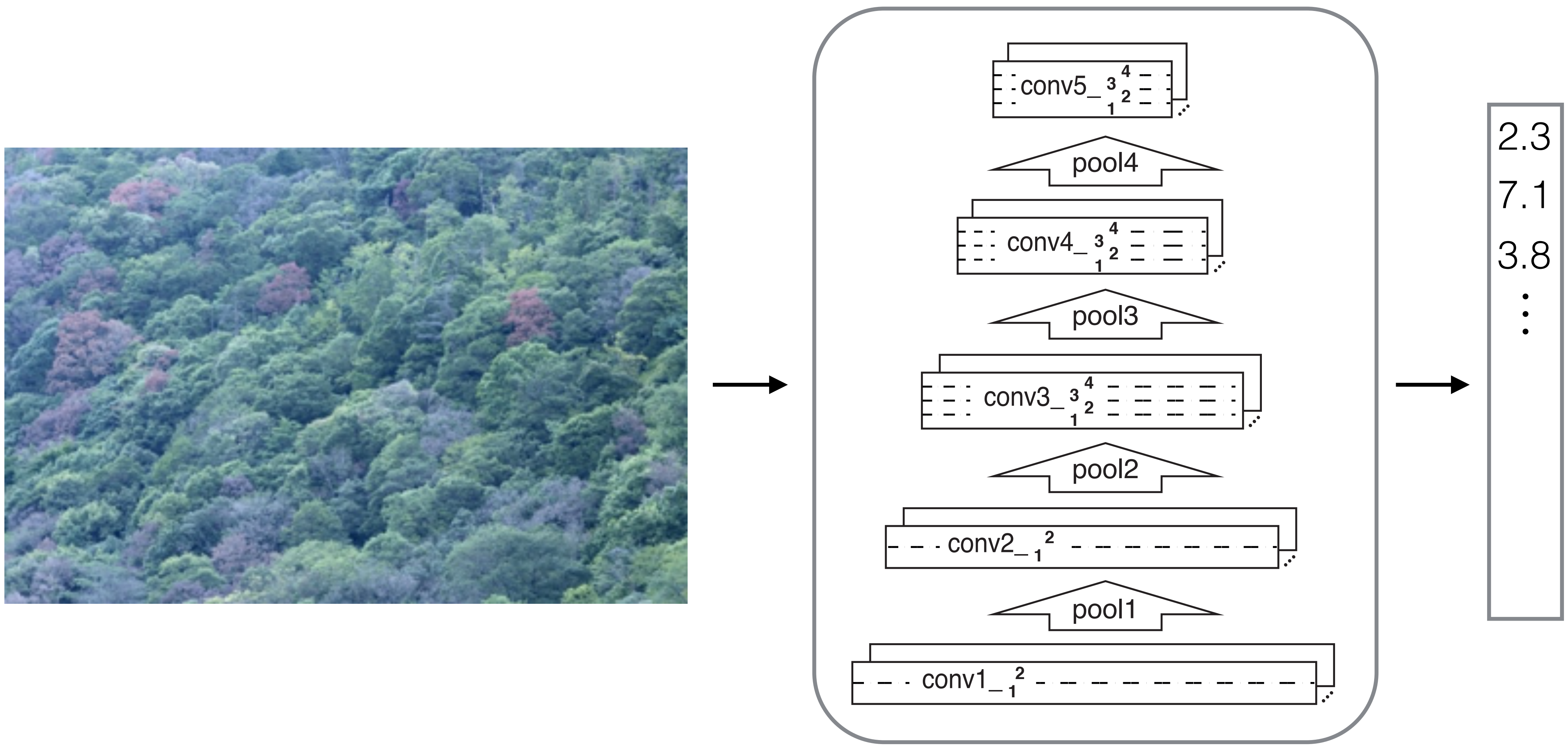


Early Vision Texture Models

Linear filter bank

Heeger & Bergen (1995)
Portilla & Simoncelli (2000)

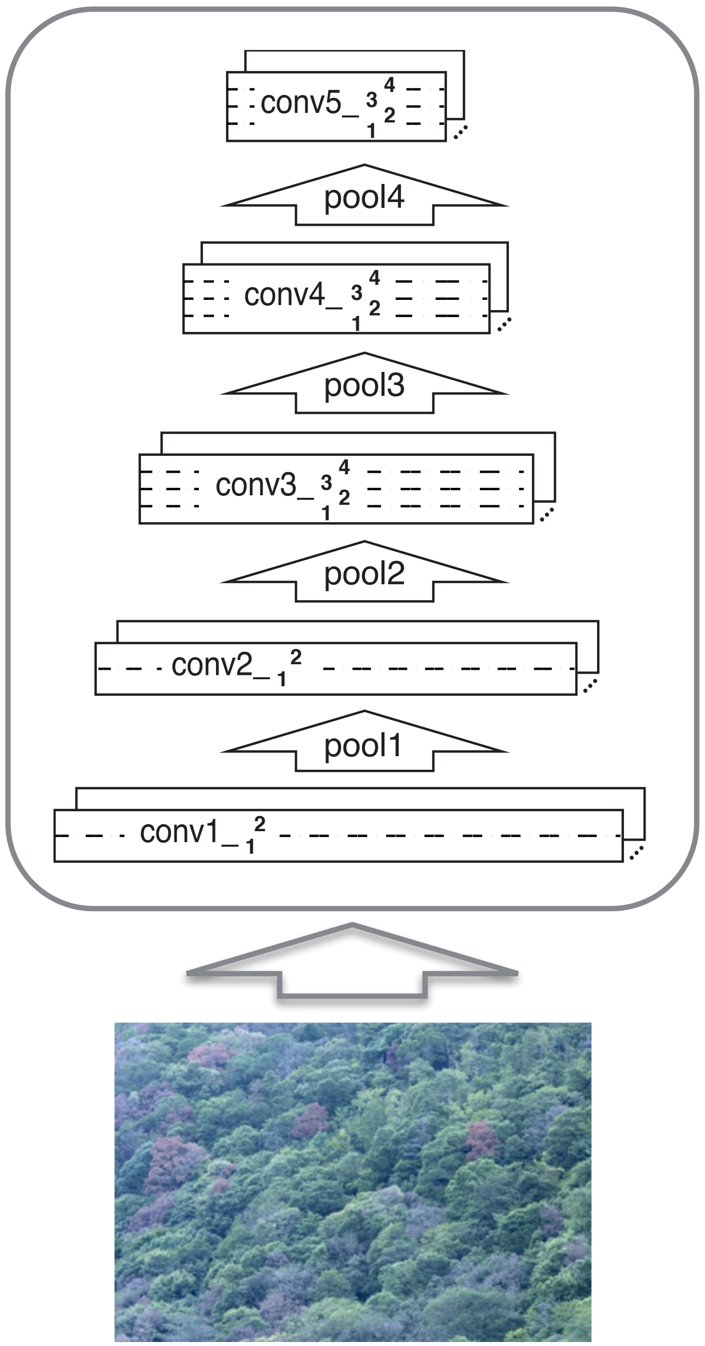
Convolutional Neural Network Texture Model



Convolutional Neural Network

Gatys et al. (NIPS 2015)

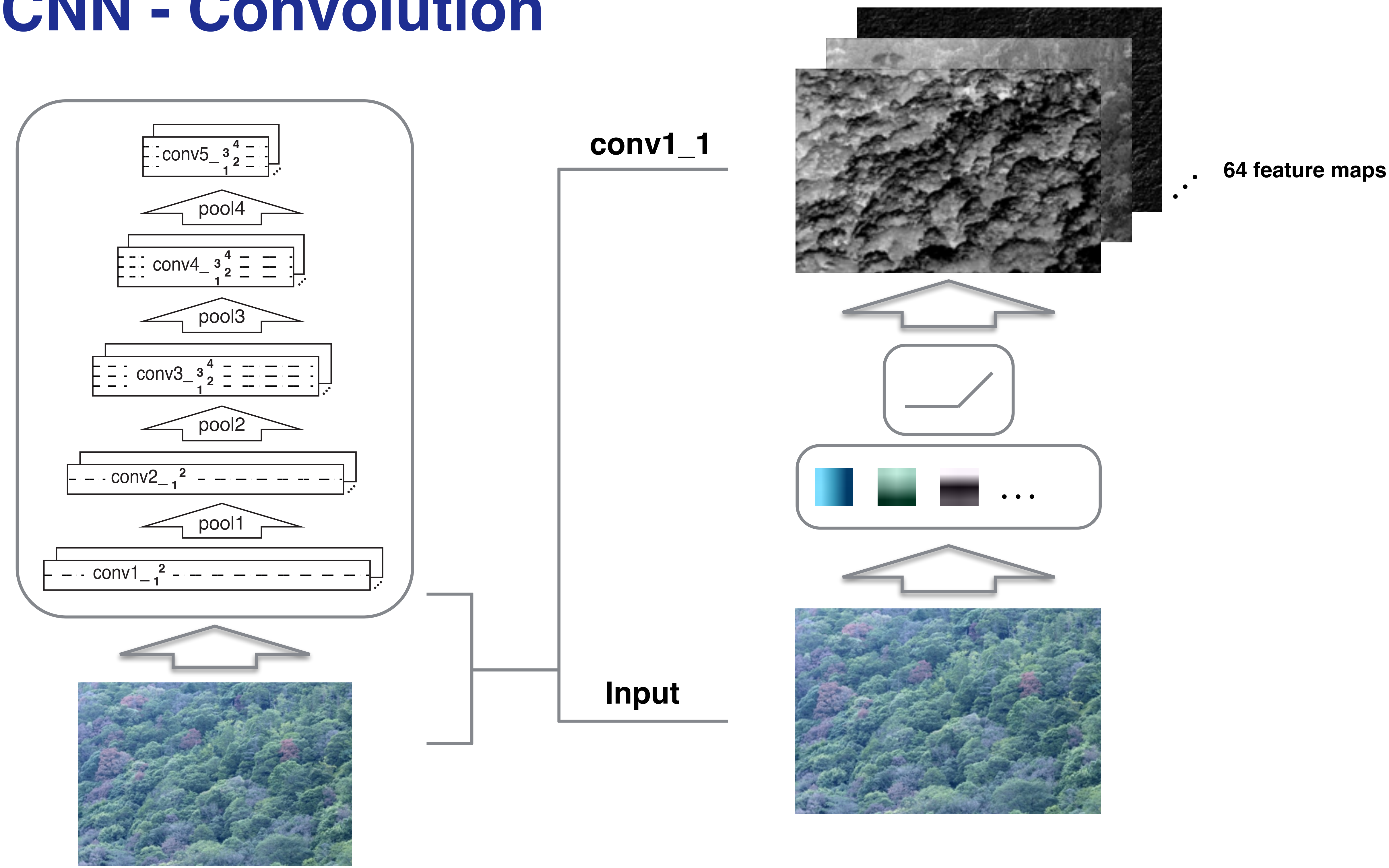
Convolutional Neural Network (CNN)



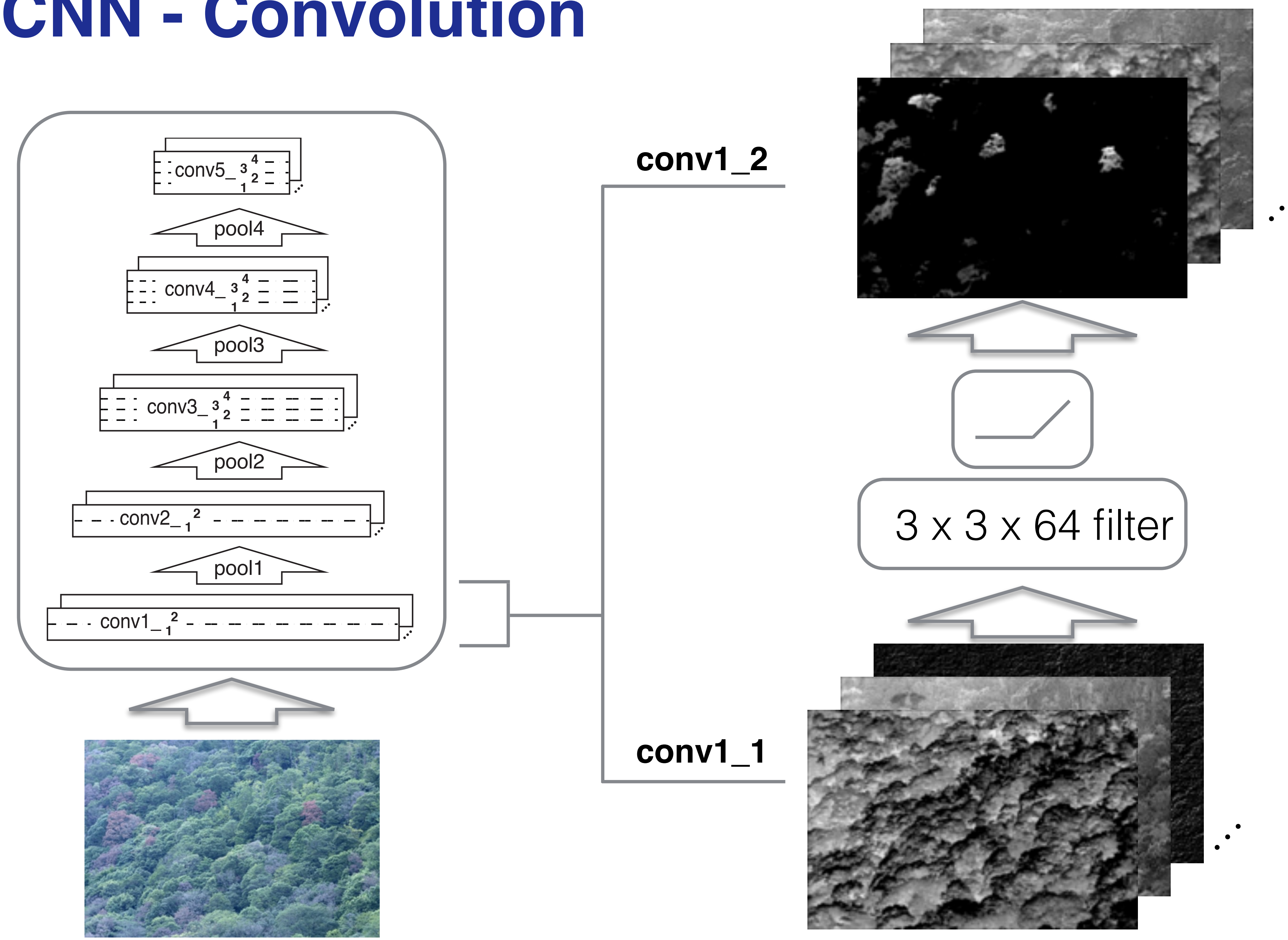
- Use VGG-19 network
- 2nd Place ImageNet 2014 object recognition challenge
- Consists of only 2 operations:
    - 3 x 3 x k rectified convolution
    - 2 x 2 max-pooling

Simonyan & Zisserman (2014)

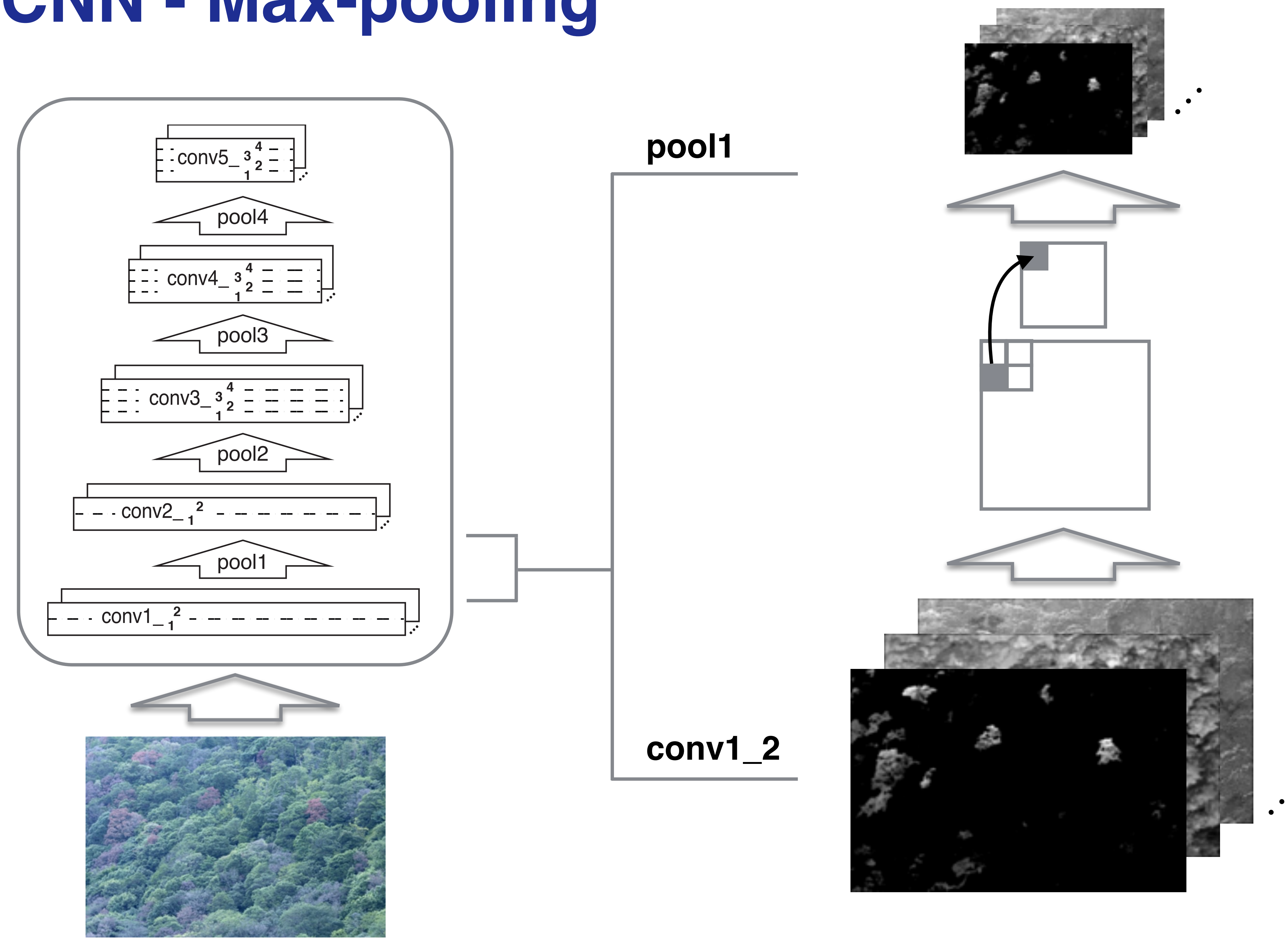
CNN - Convolution



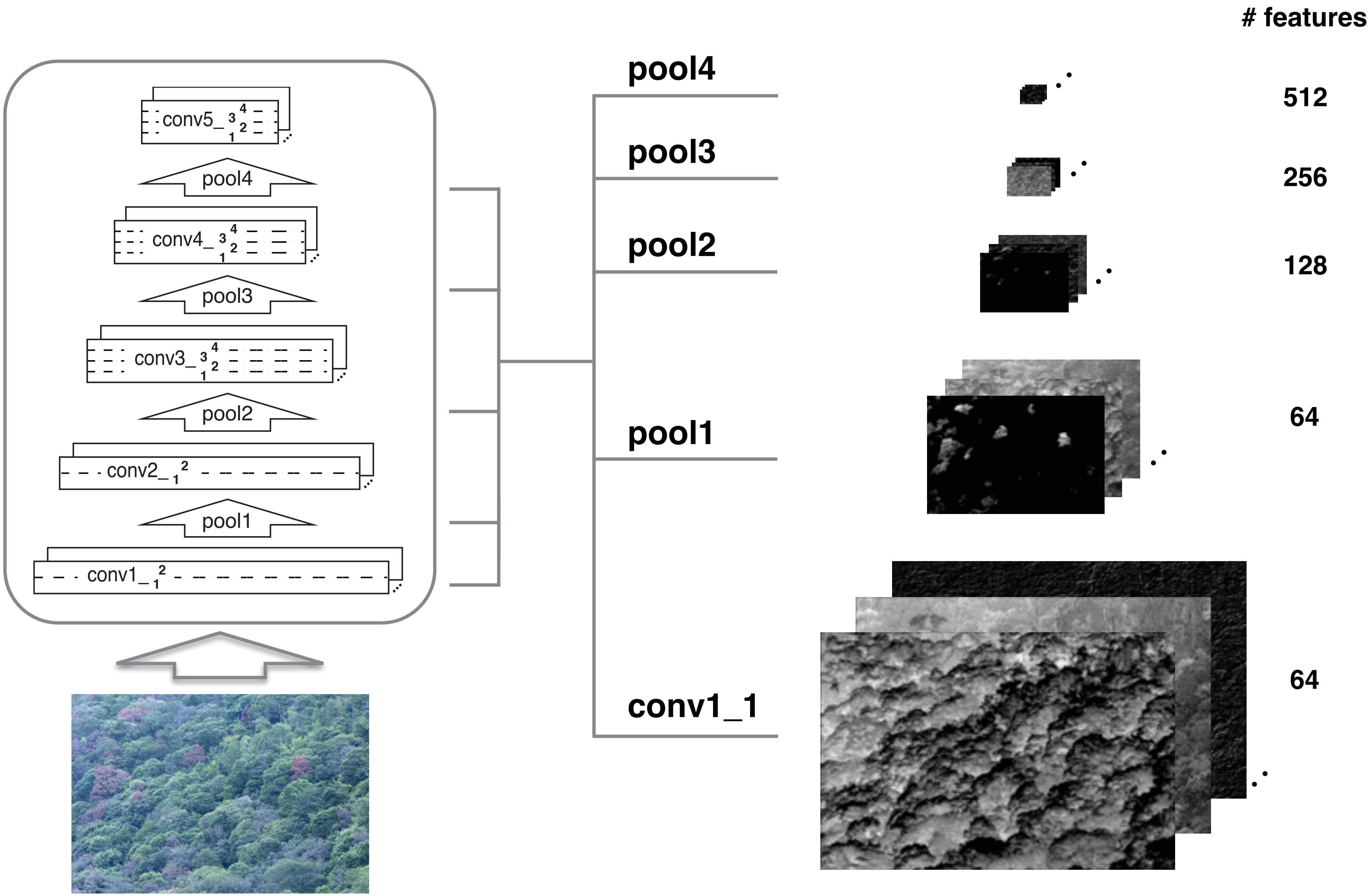
CNN - Convolution



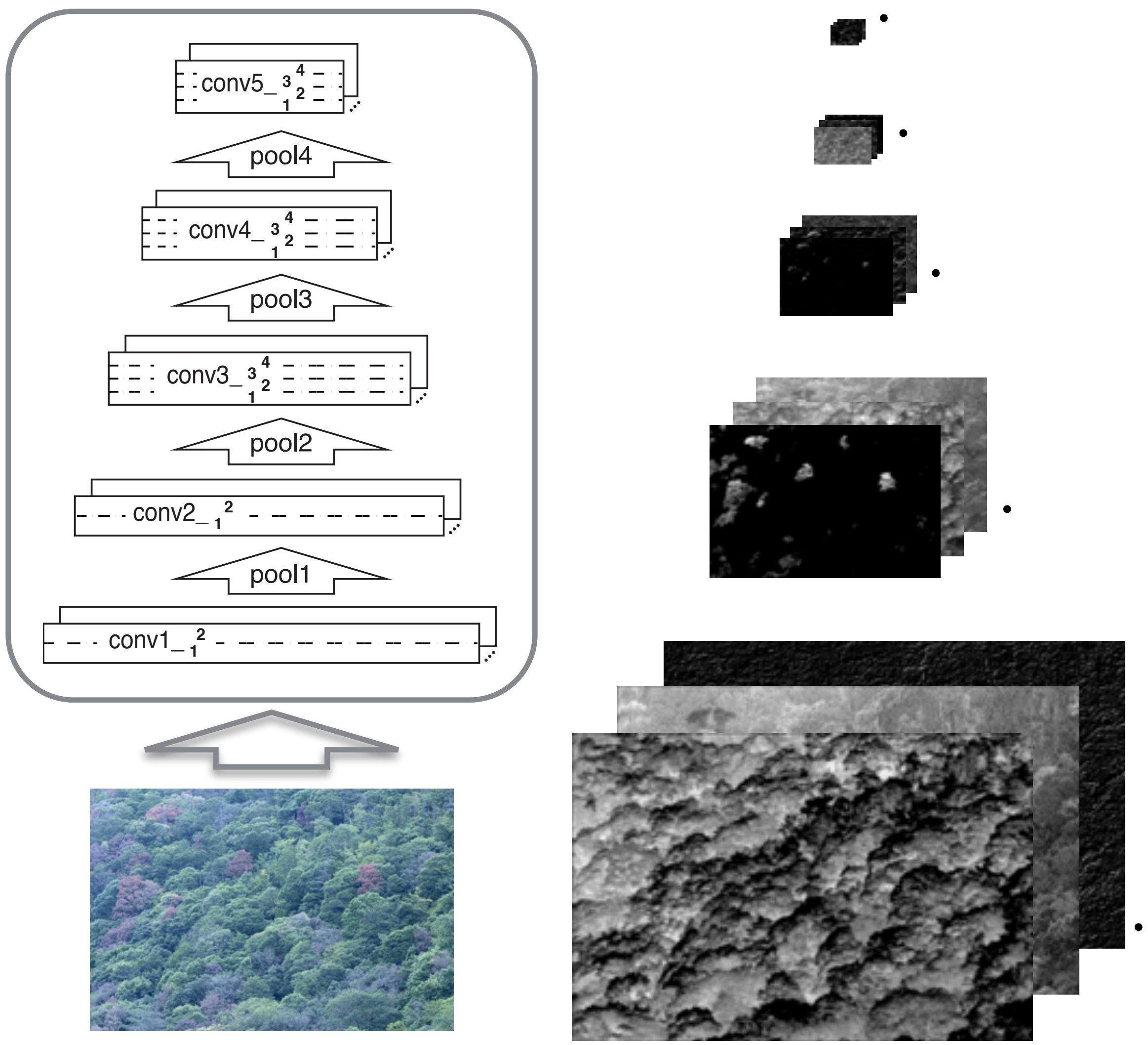
CNN - Max-pooling



CNN - Multiscale Filter Bank



CNN - Texture Features



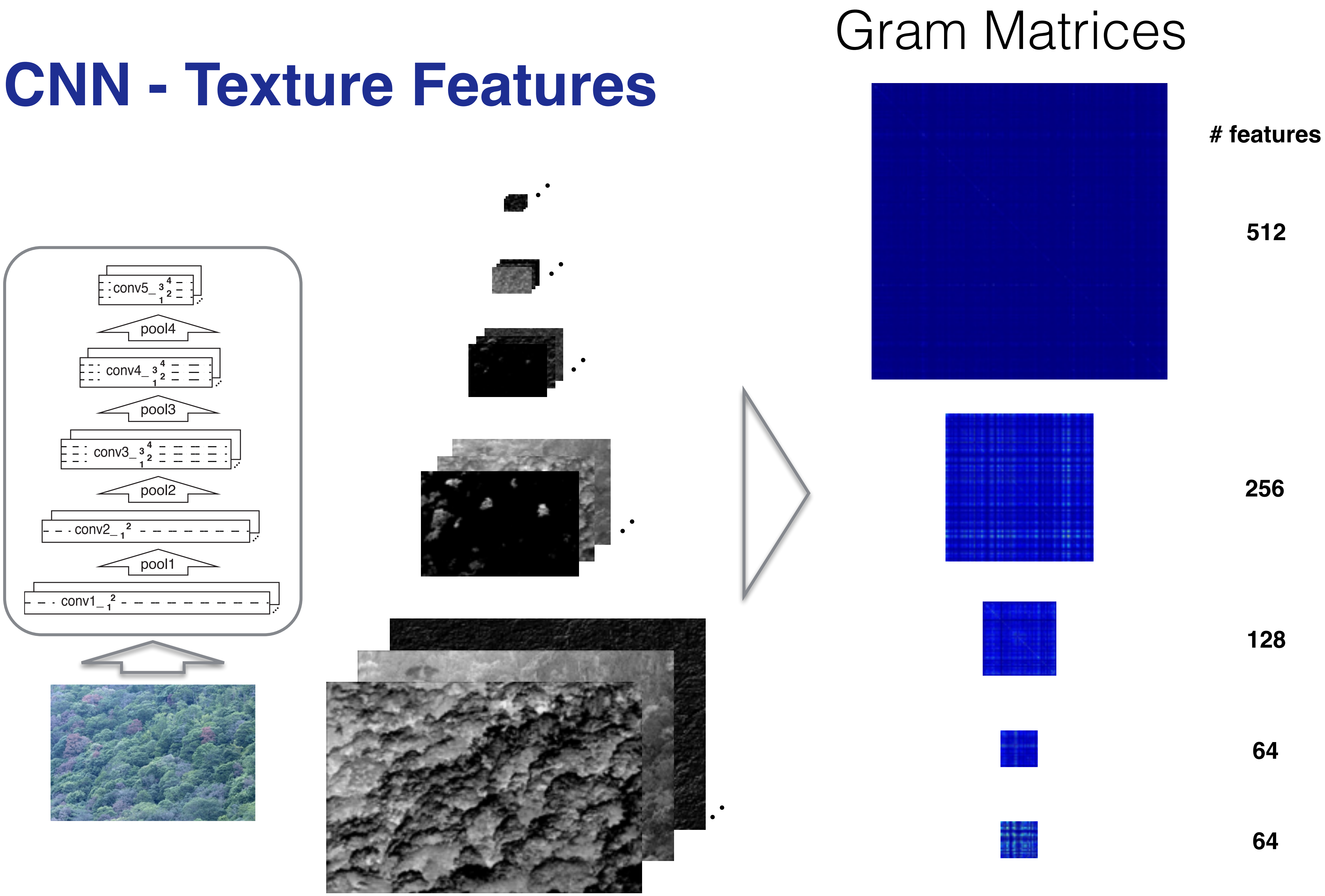
$$F = [\bar{f}_1, \bar{f}_2, \bar{f}_3, ..., \bar{f}_N]^T$$

$$G = FF^T$$

$$= \begin{pmatrix} \langle \bar{f}_1, \bar{f}_1 \rangle & \cdots & \langle \bar{f}_1, \bar{f}_N \rangle \\ \langle \bar{f}_2, \bar{f}_1 \rangle & & \vdots \\ \vdots & \ddots & \vdots \\ \langle \bar{f}_N, \bar{f}_1 \rangle & \cdots & \langle \bar{f}_N, \bar{f}_N \rangle \end{pmatrix}$$

$$\langle \bar{f}_i, \bar{f}_j \rangle = \sum_k F_{ik} F_{jk}$$

CNN - Texture Features

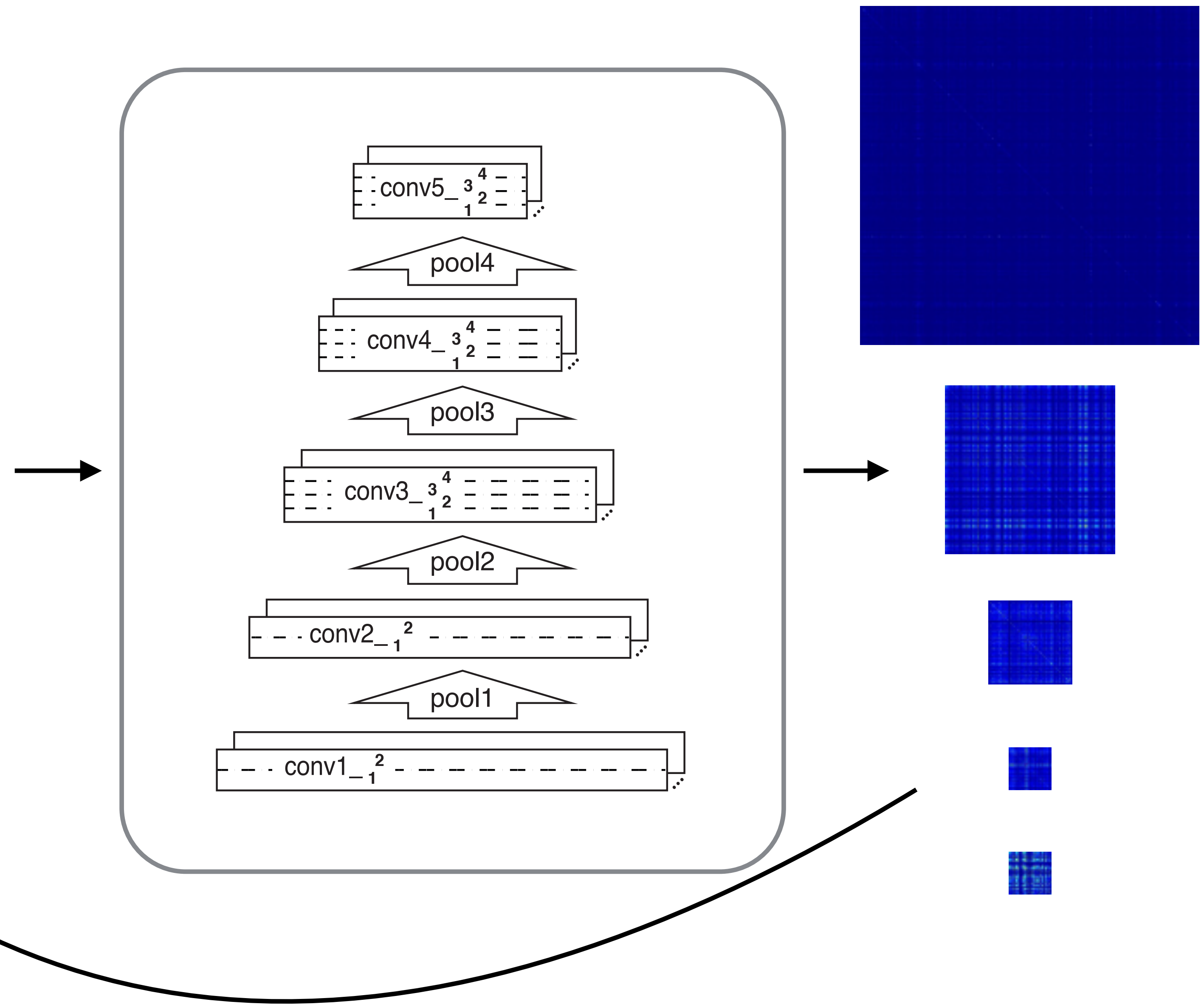


Parametric Texture Synthesis

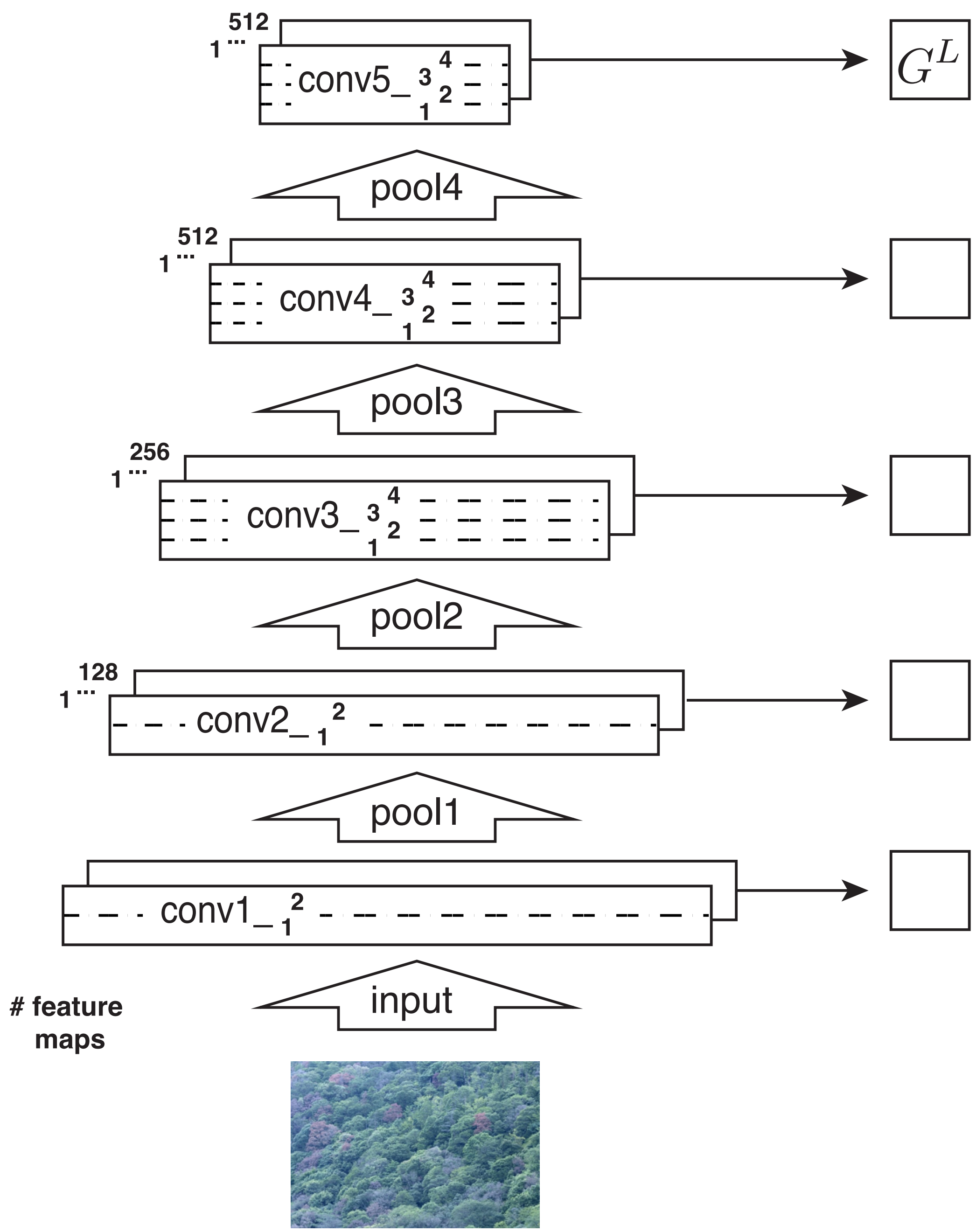
Synthesis

Gram Matrices

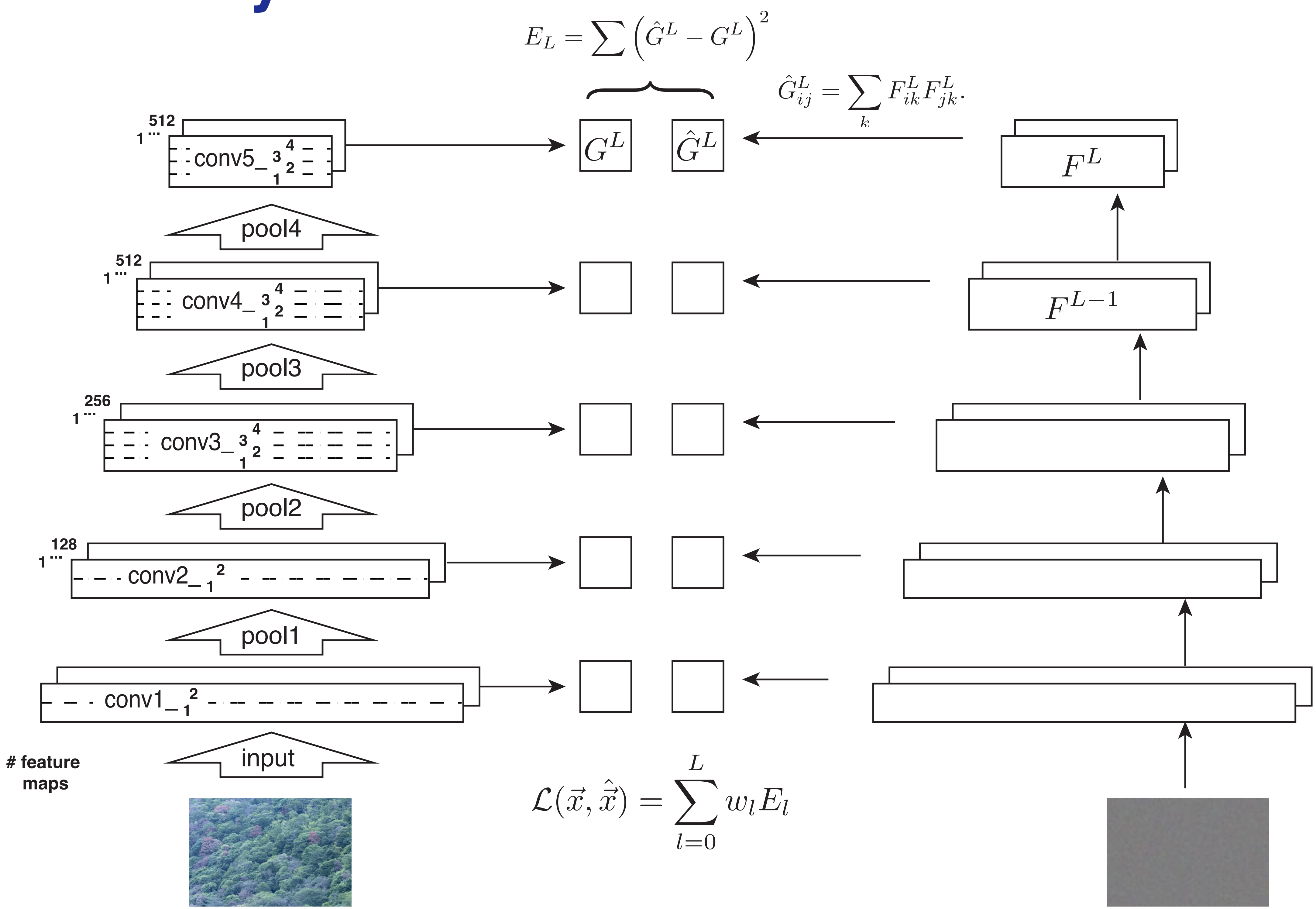
Find pre-image with equal model responses



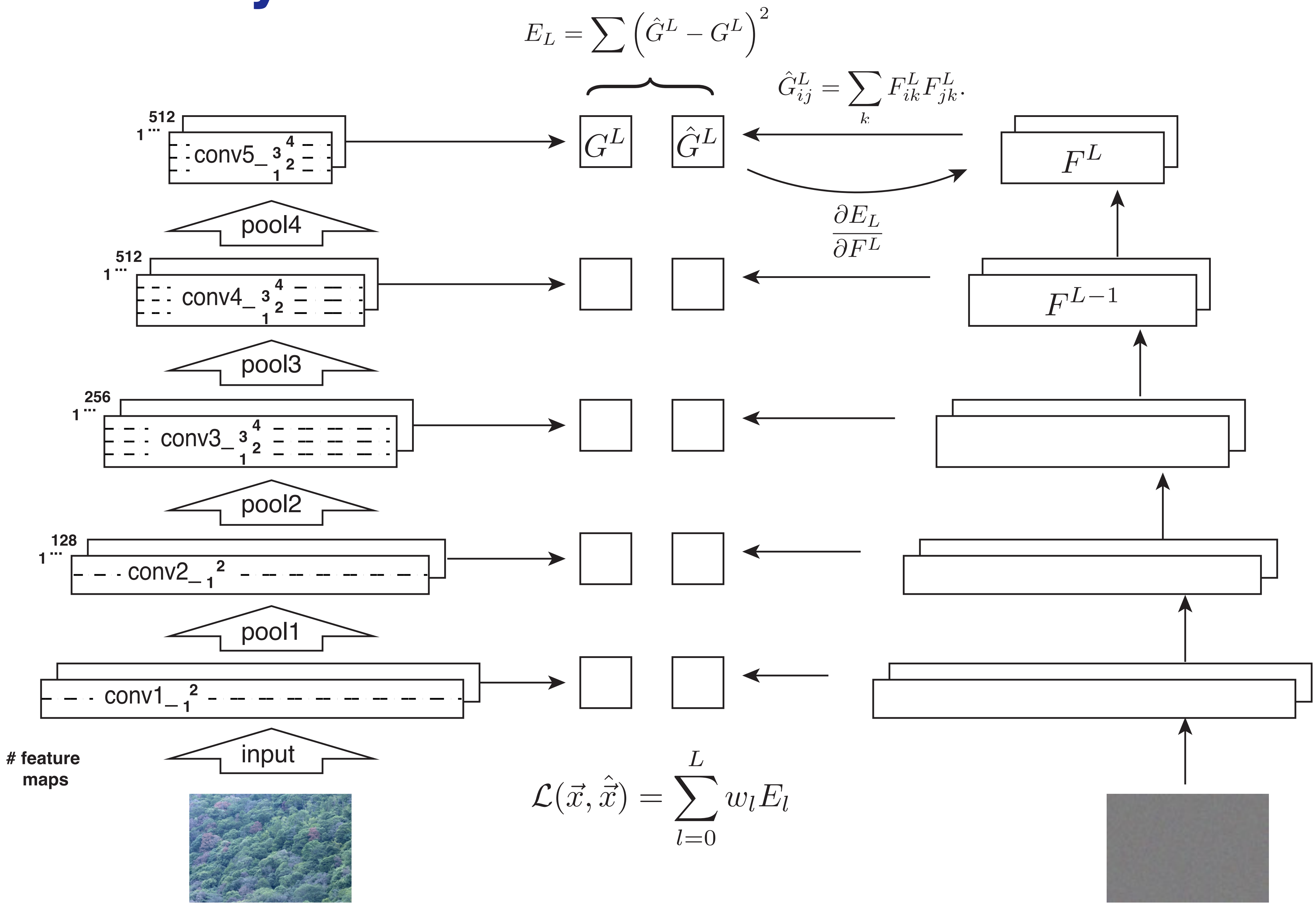
Texture Synthesis



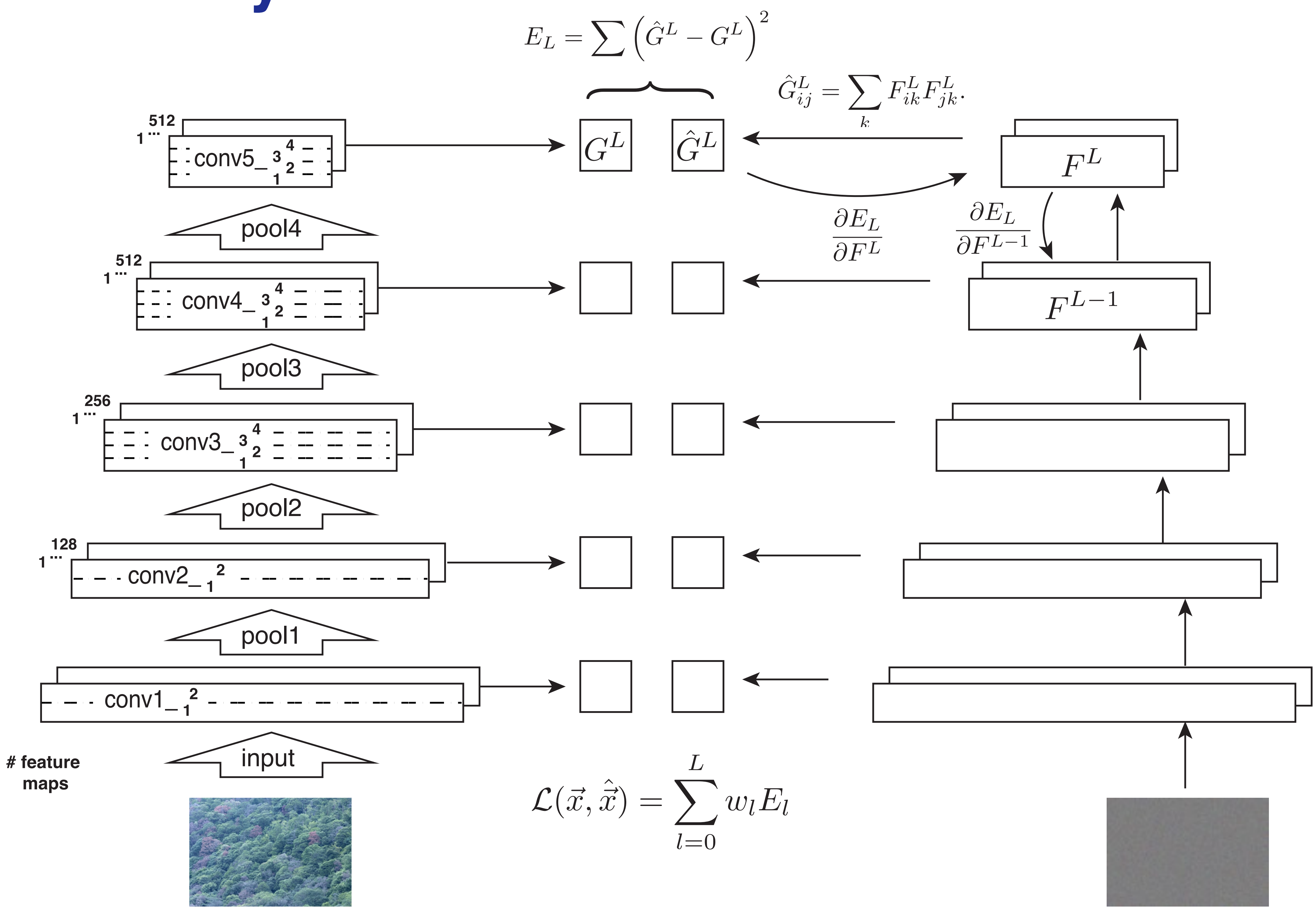
Texture Synthesis



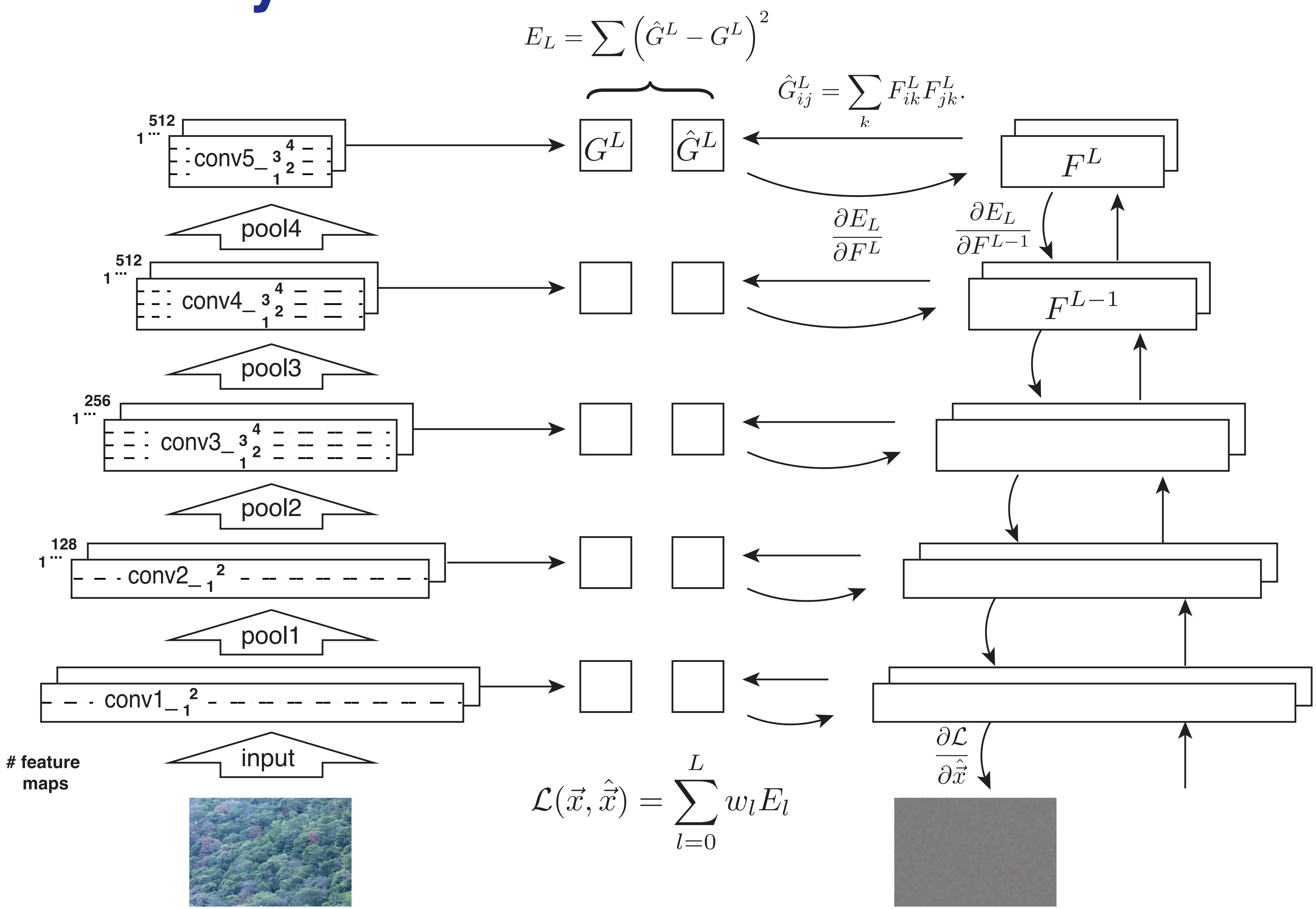
Texture Synthesis



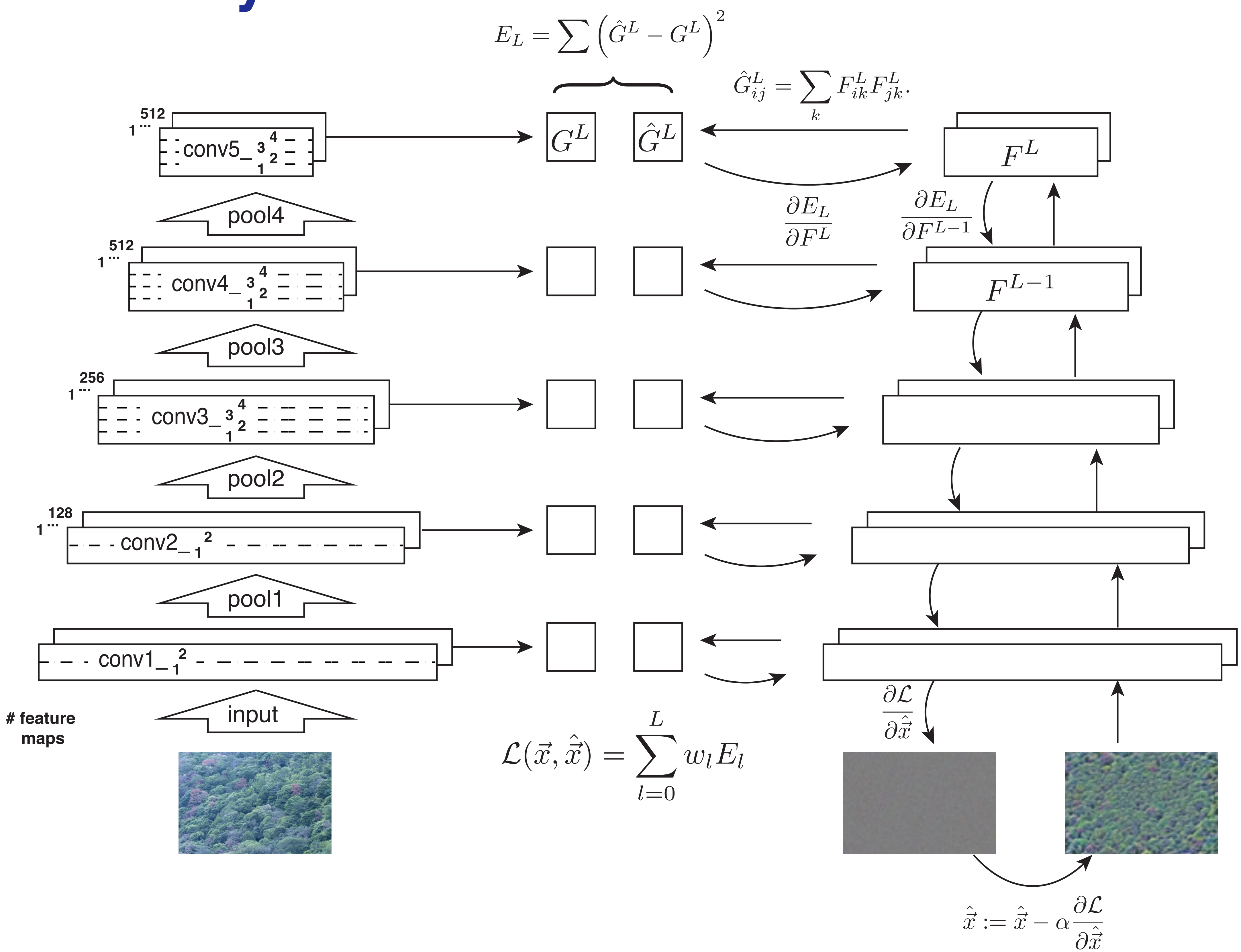
Texture Synthesis



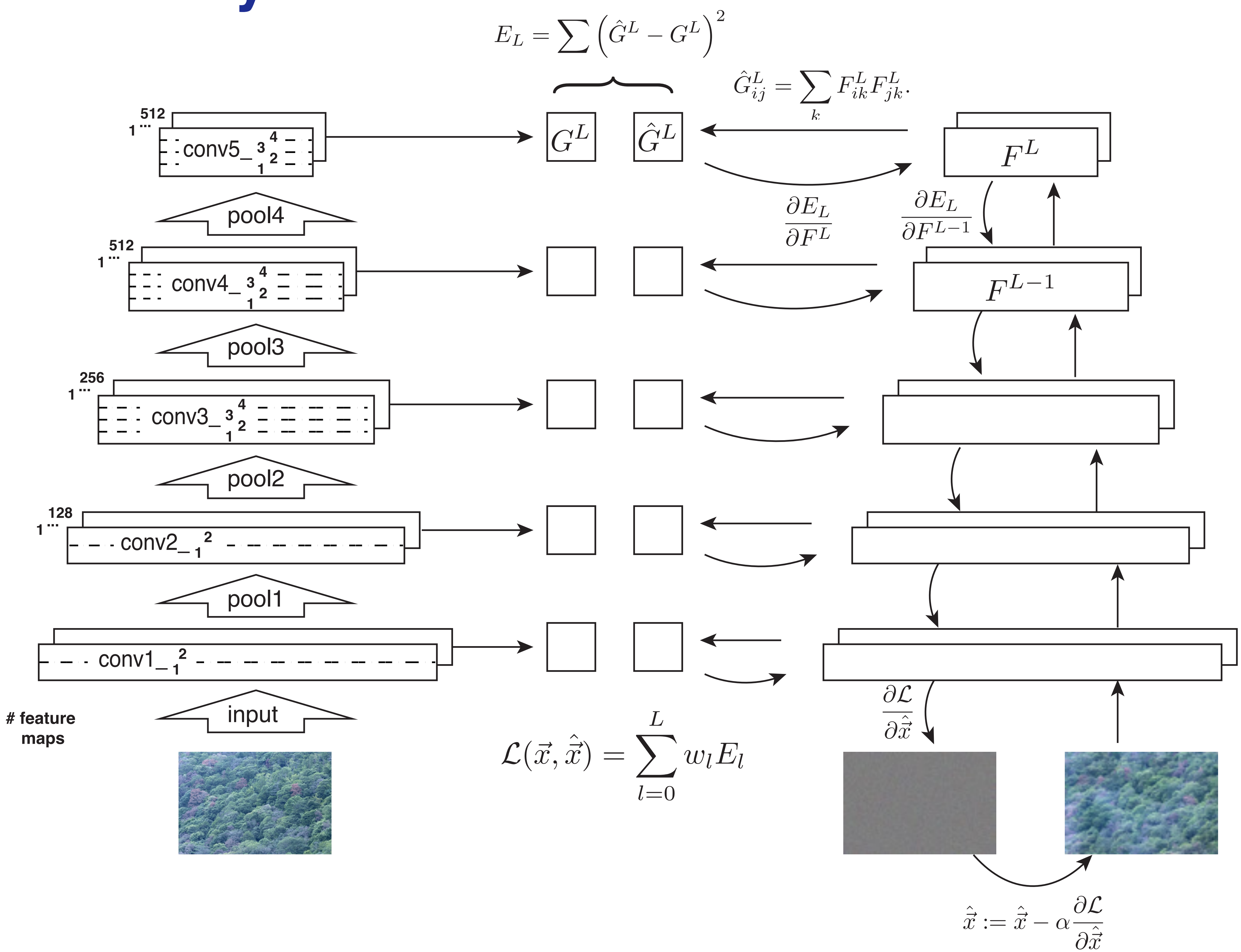
Texture Synthesis



Texture Synthesis



Texture Synthesis

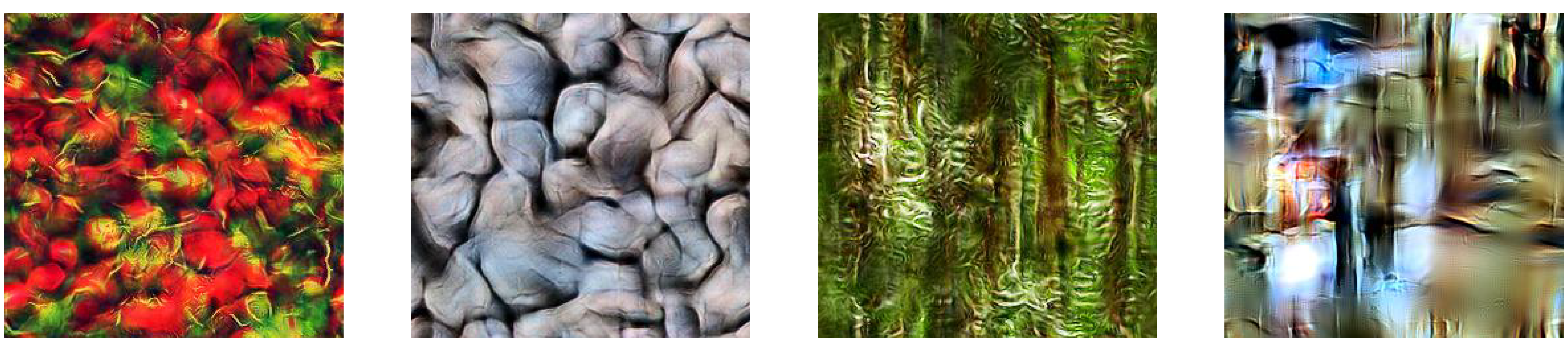


Texture Synthesis - Results

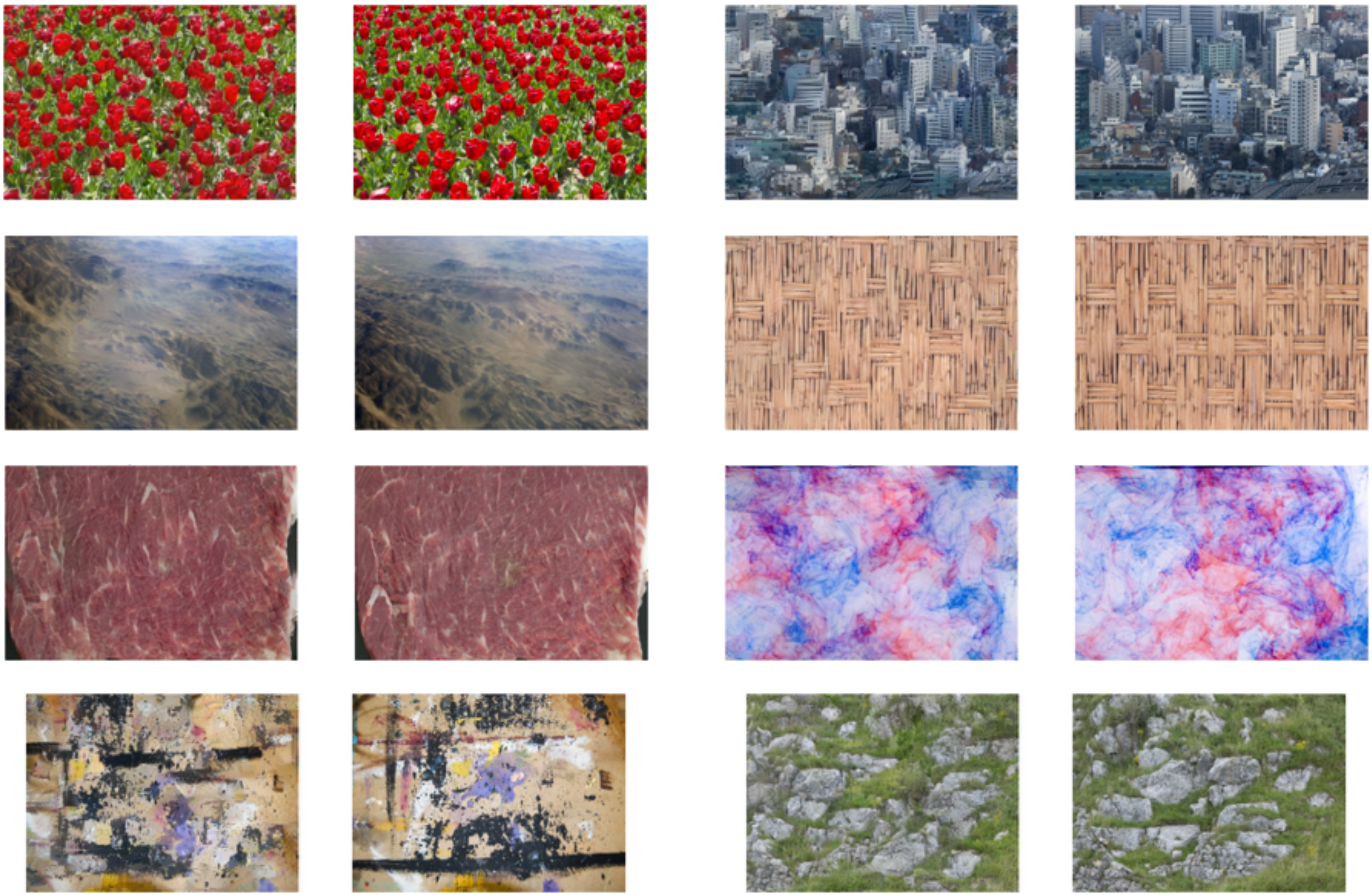
our texture algorithm

Original

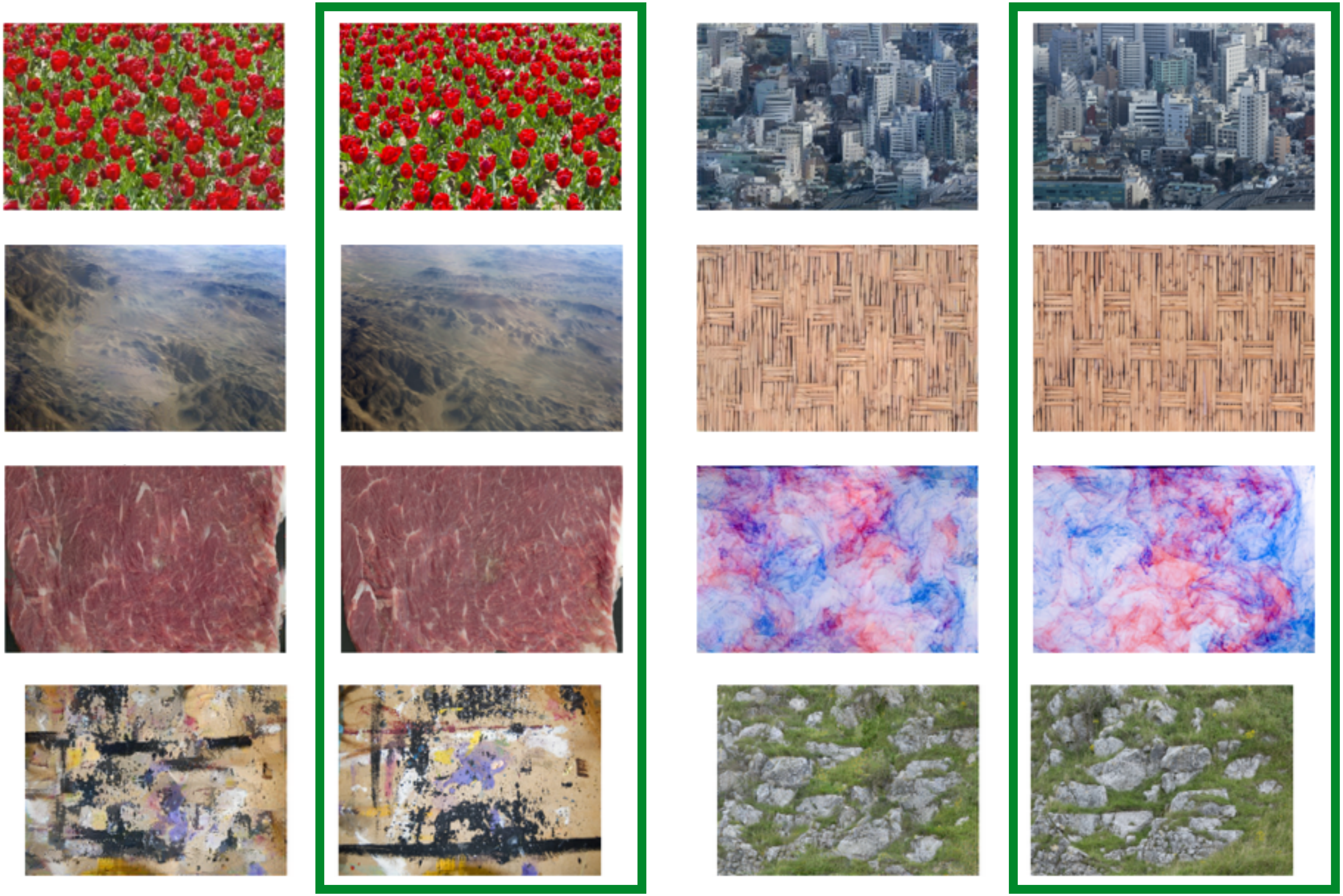
Portilla& Simoncelli 1999



Test Julesz' Conjecture



Test Julesz' Conjecture

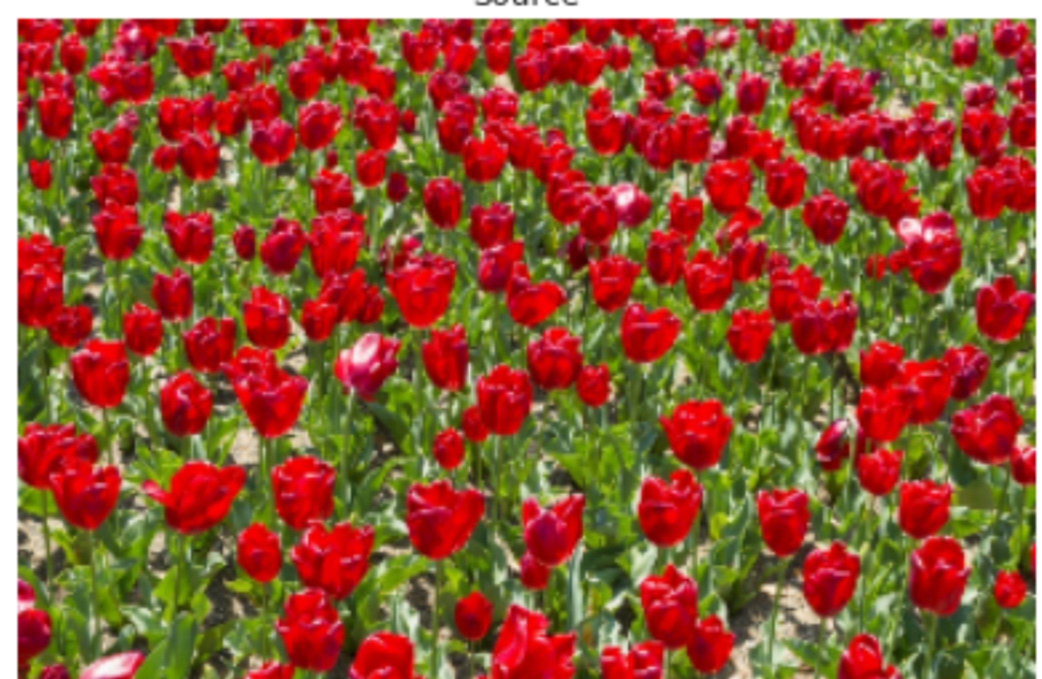


Synthesised

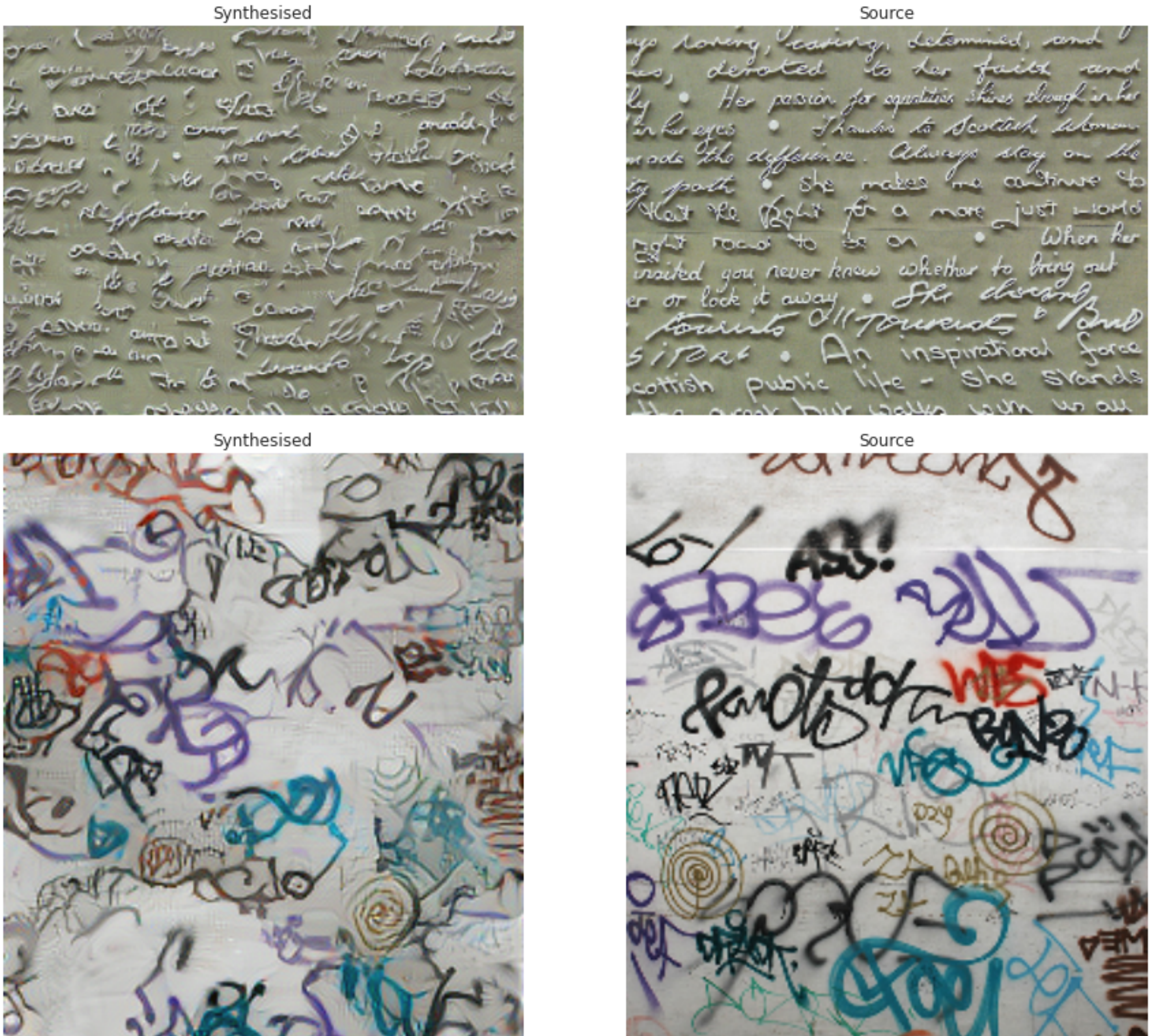
Source

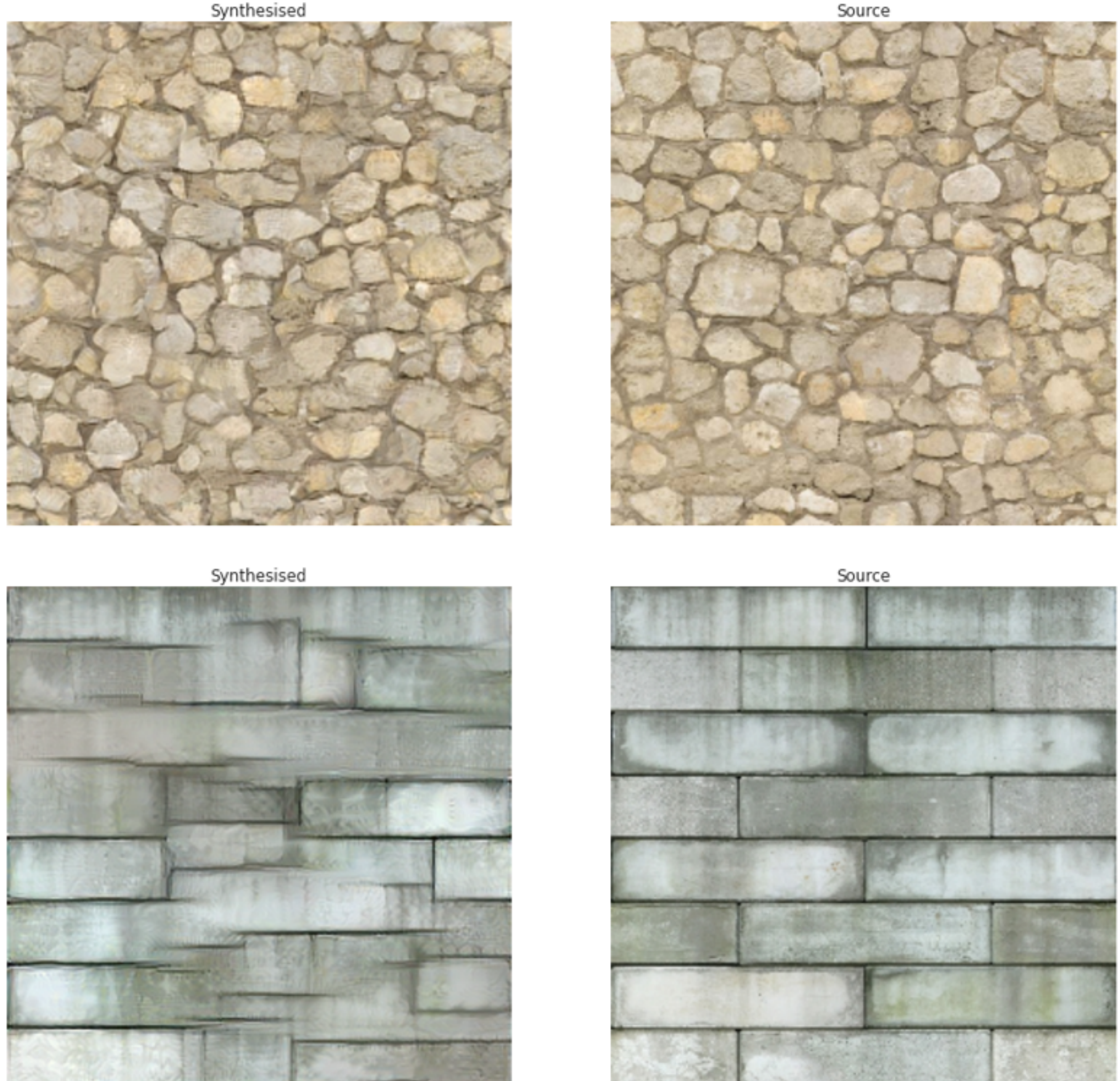
Synthesised

Source



Synthesised
Source
Synthesised
Source
Synthesised
Source


Synthesised
Source
Synthesised
Source



Synthesised
Source
Synthesised
Source

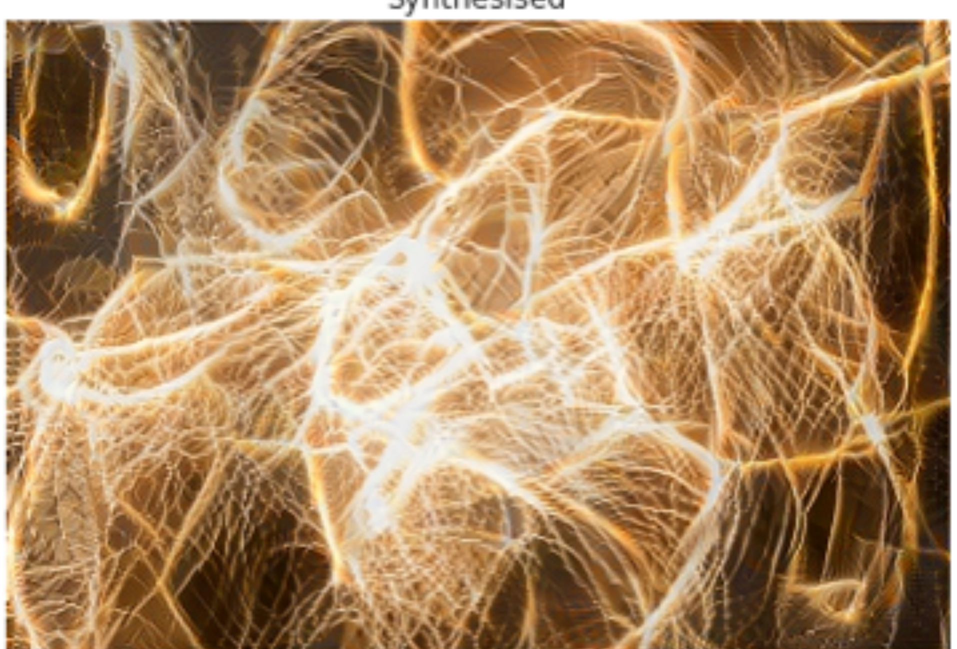
Synthesised

Source

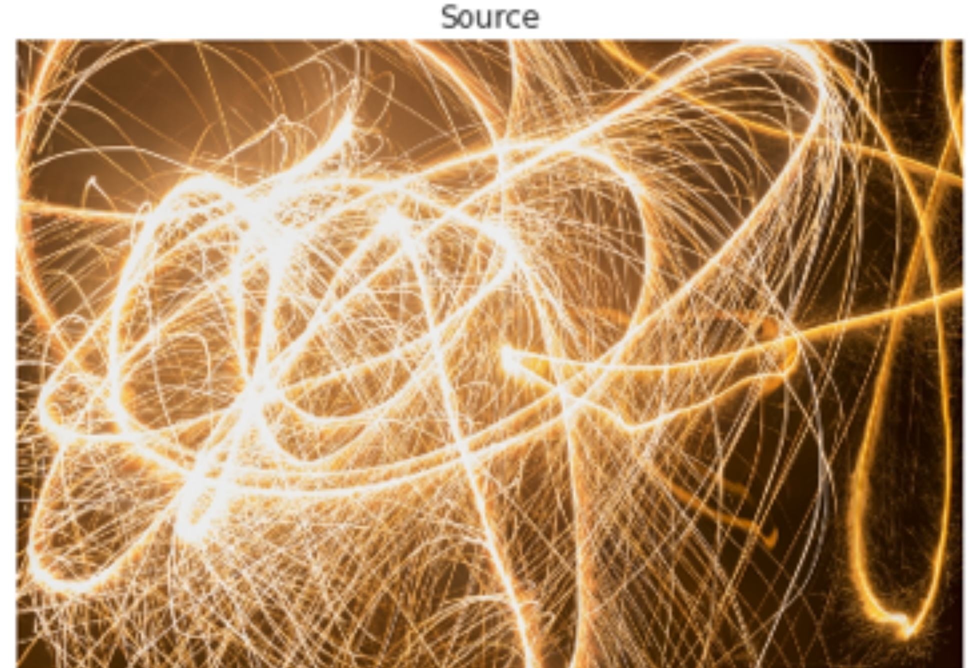
Synthesised

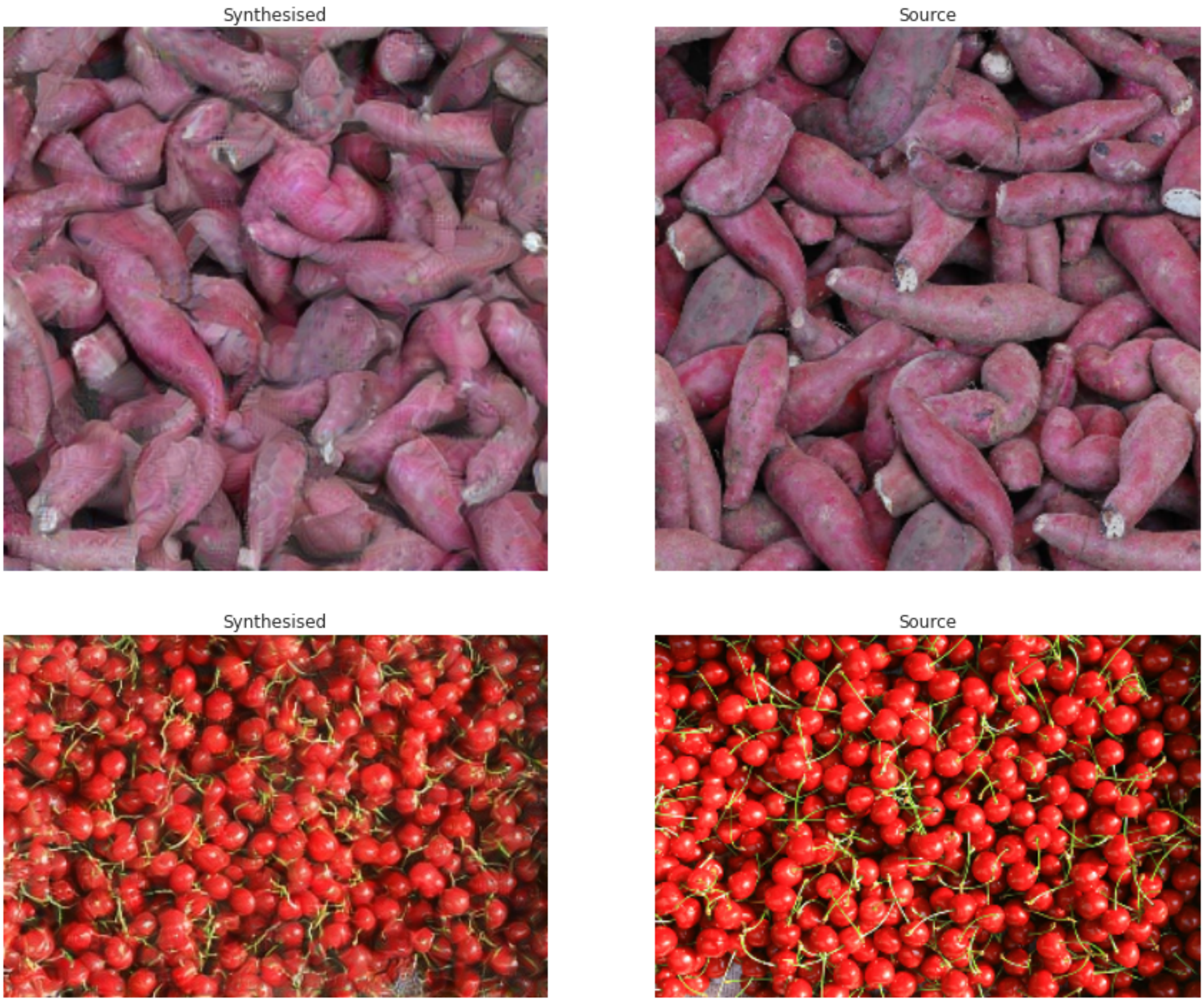
Source

Synthesised



Source

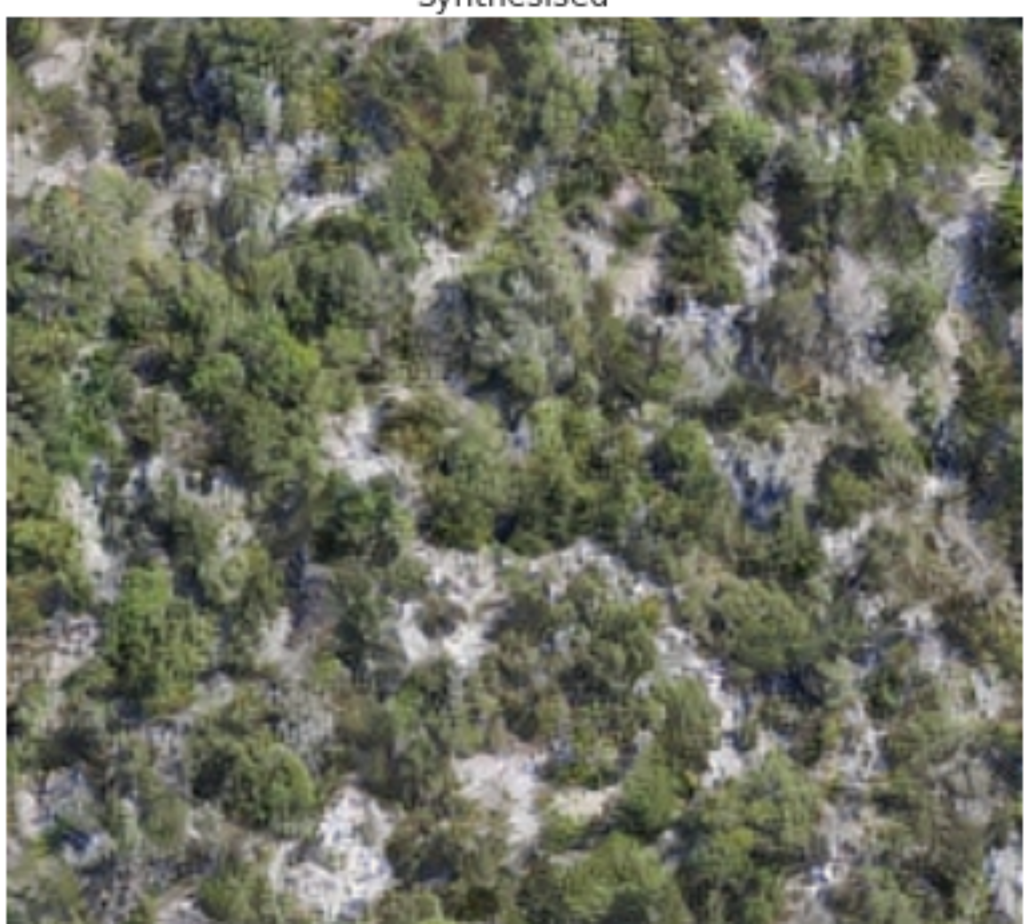



Synthesised
Source
Synthesised
Source

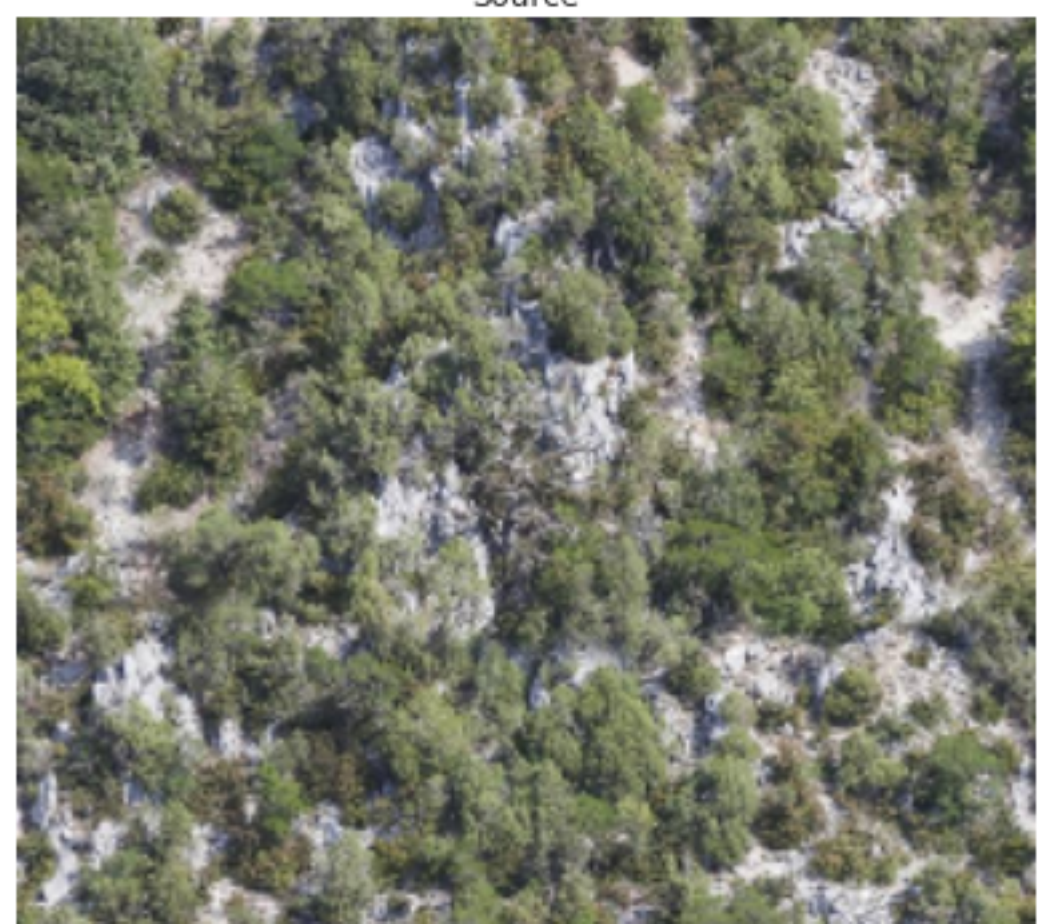
Synthesised

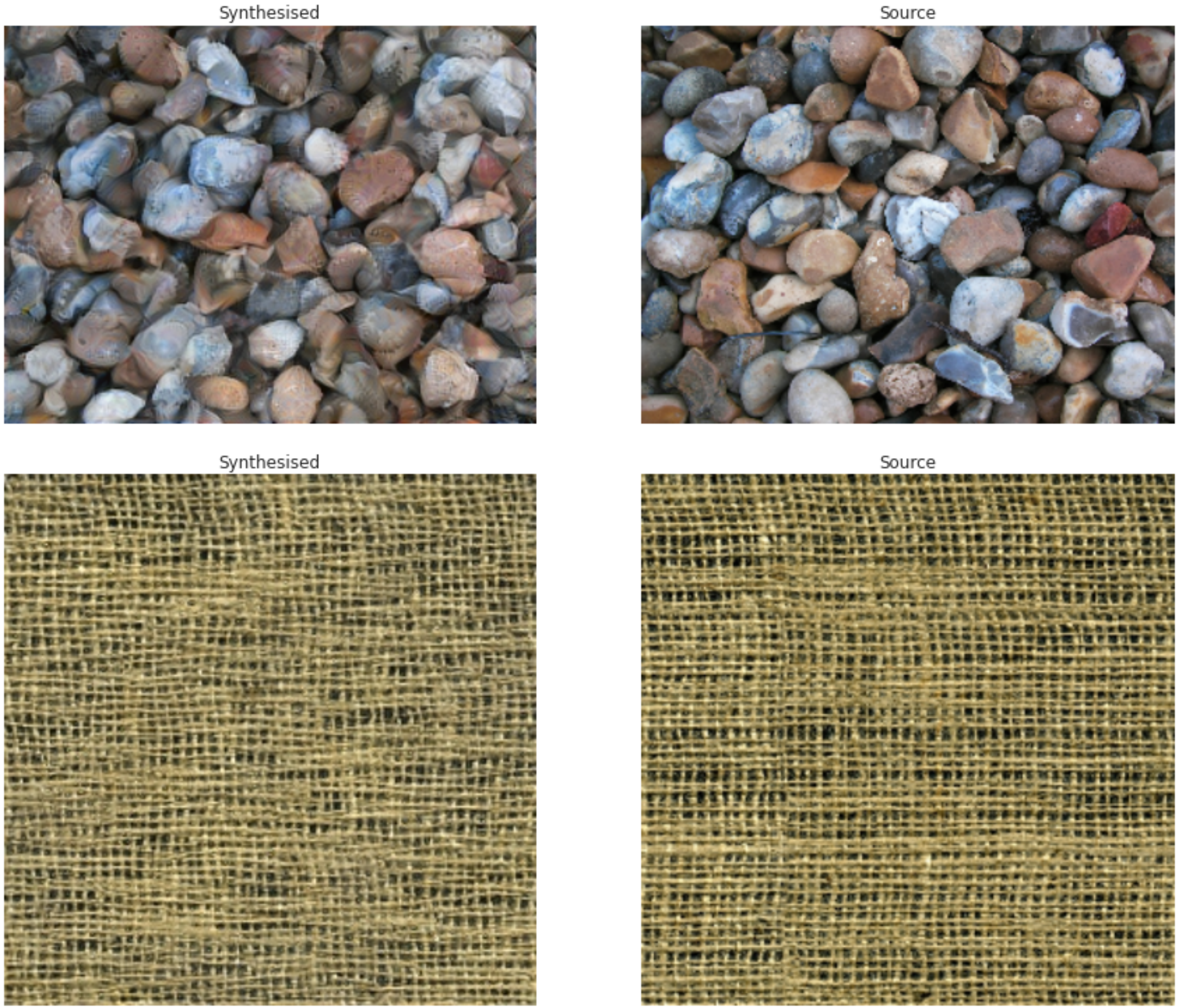
Source

Synthesised

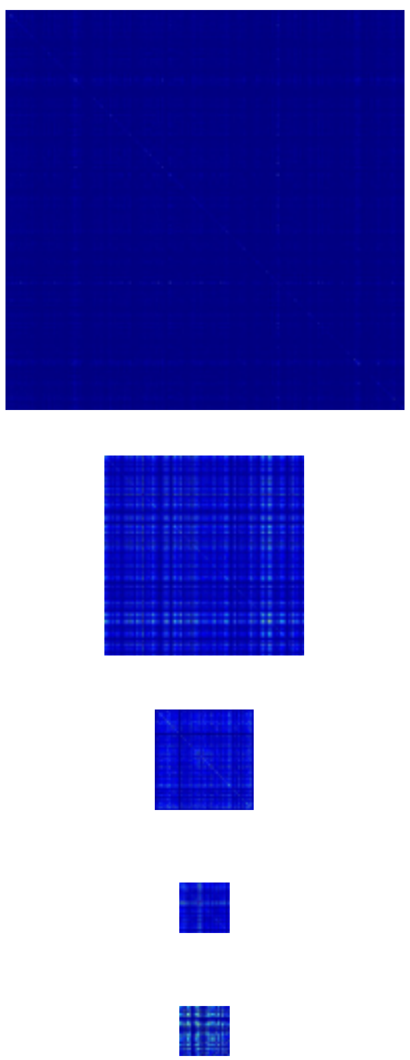


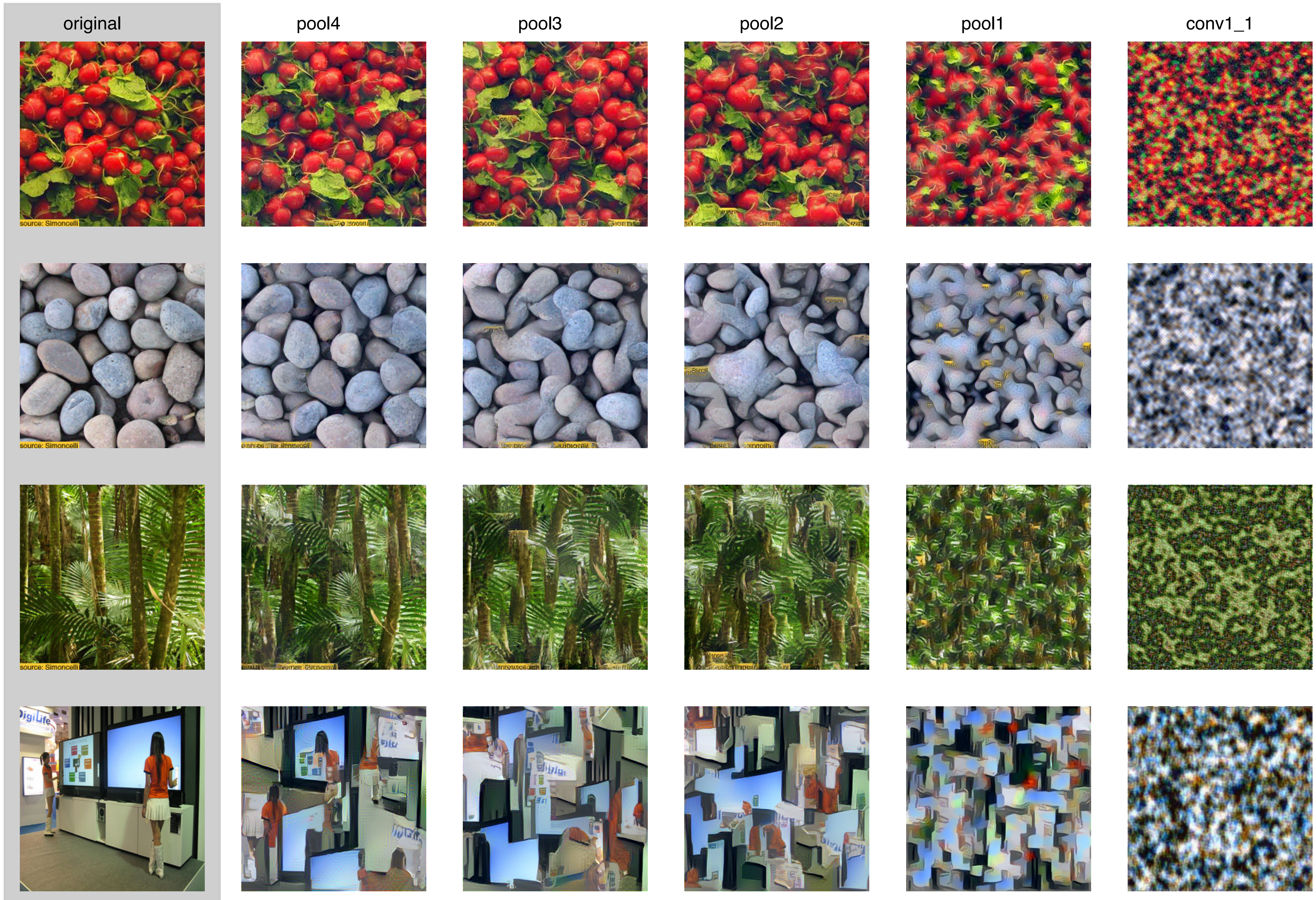
Source



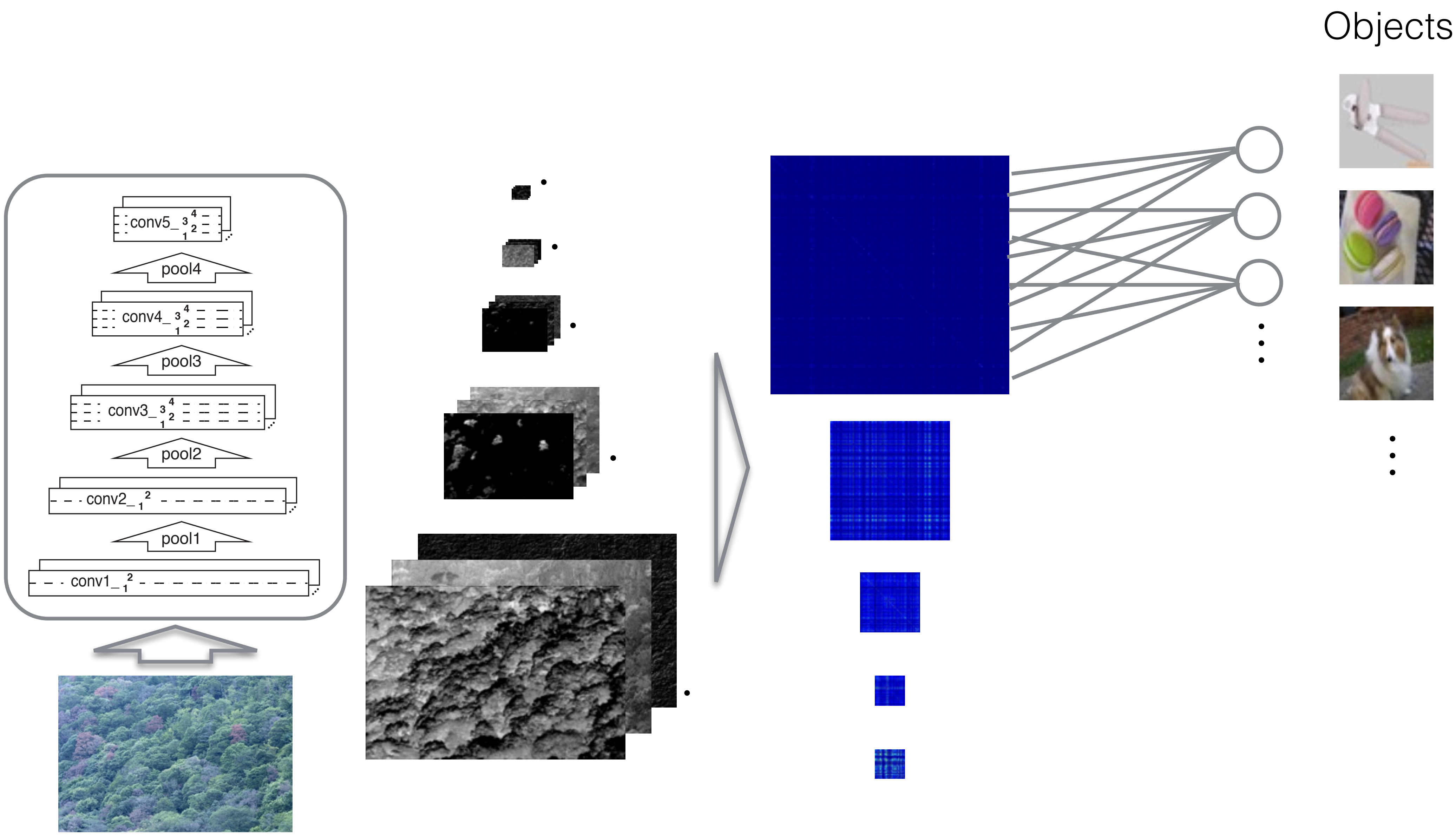

Synthesised
Source
Synthesised
Source

Texture Synthesis - Layers

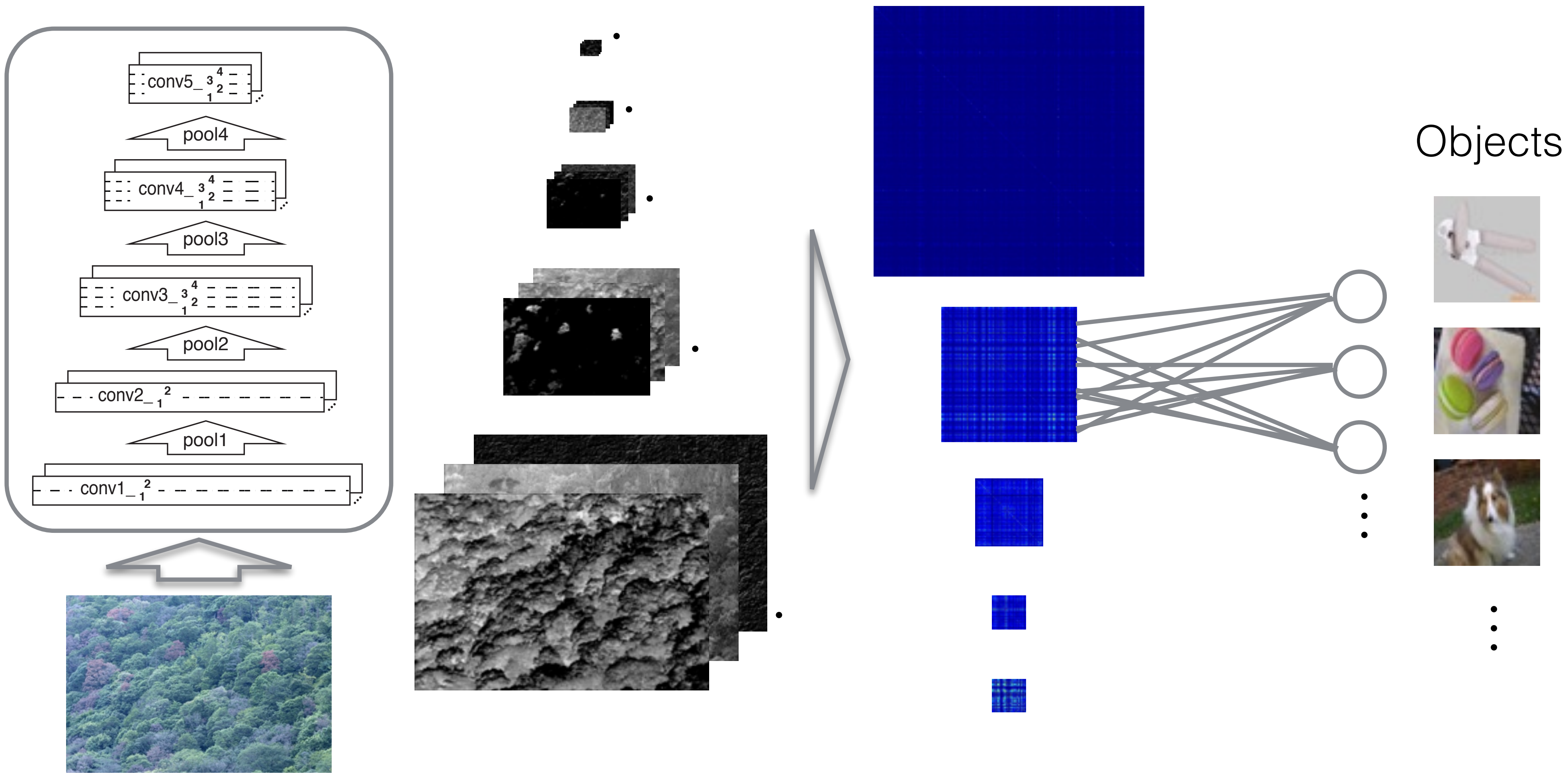




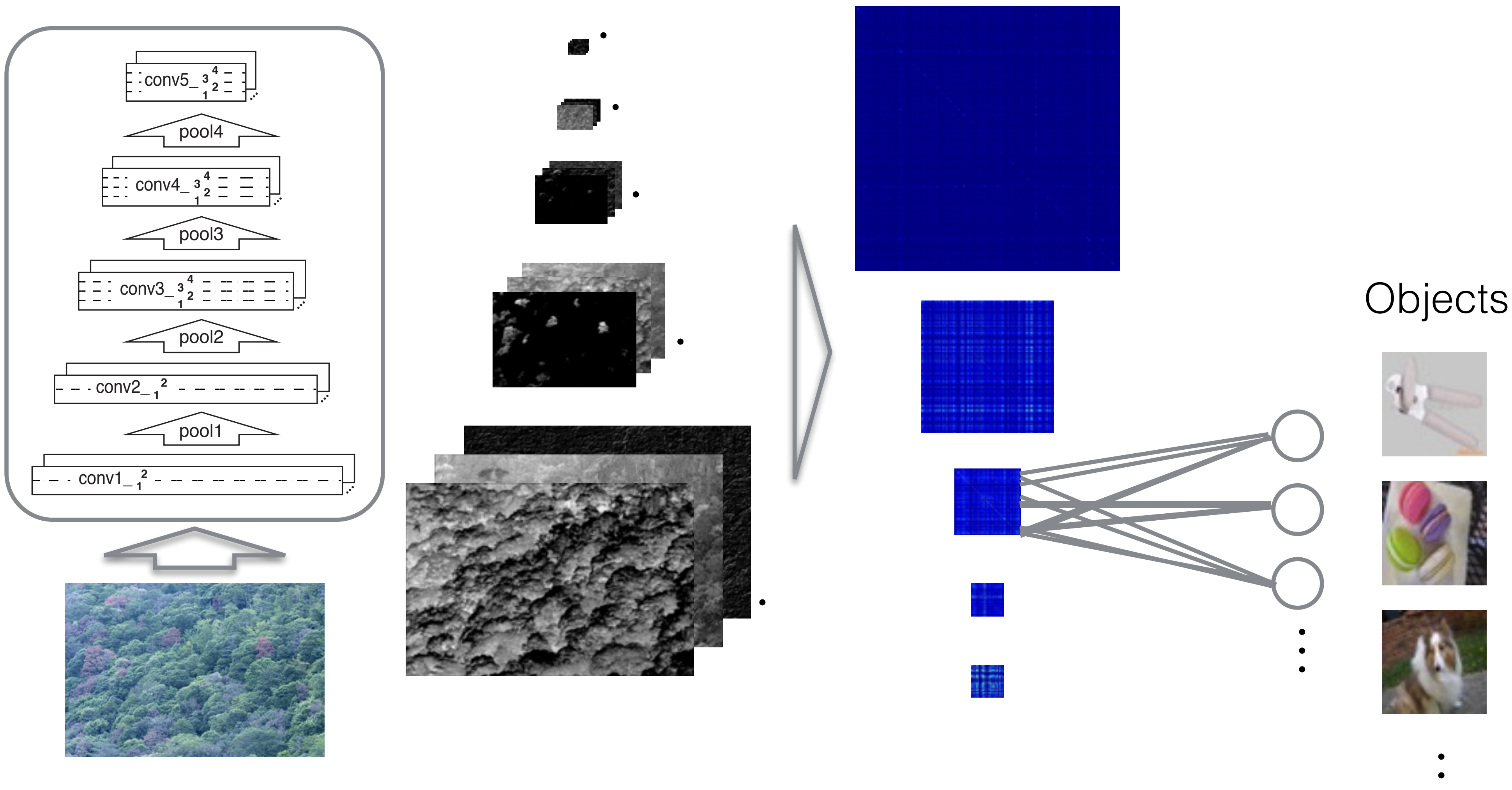
Classification from Texture Features

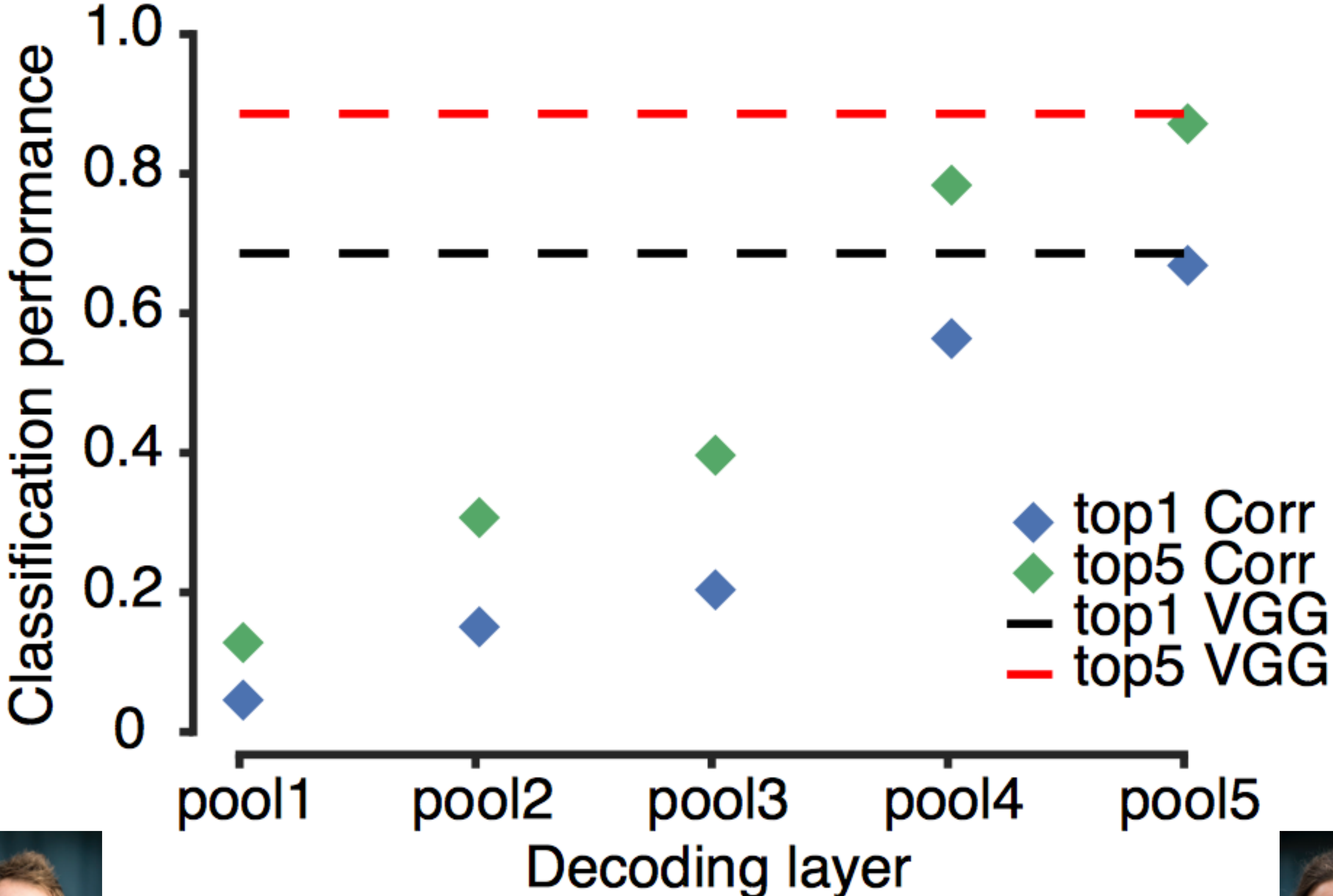


Classification from Texture Features



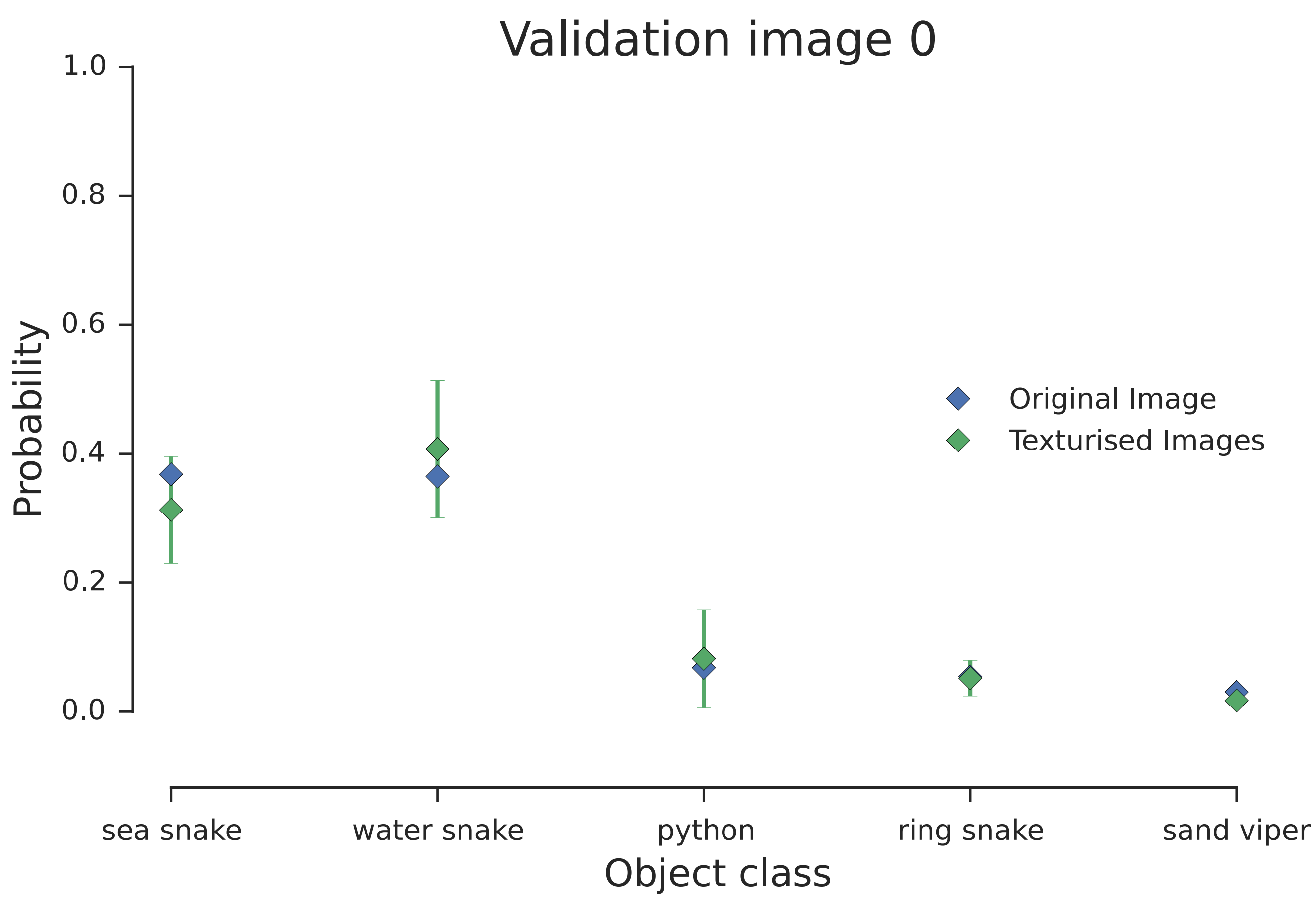
Classification from Texture Features



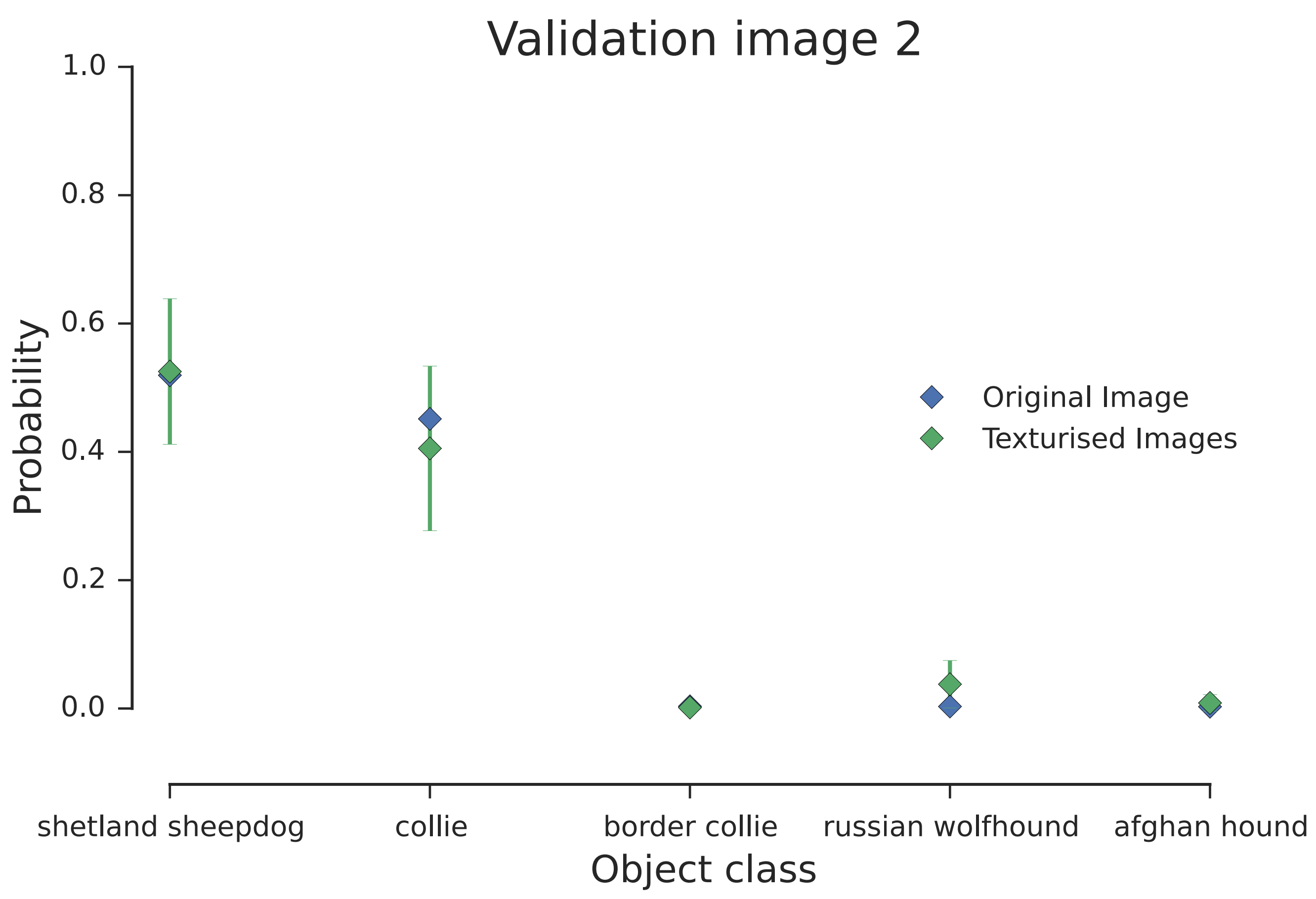


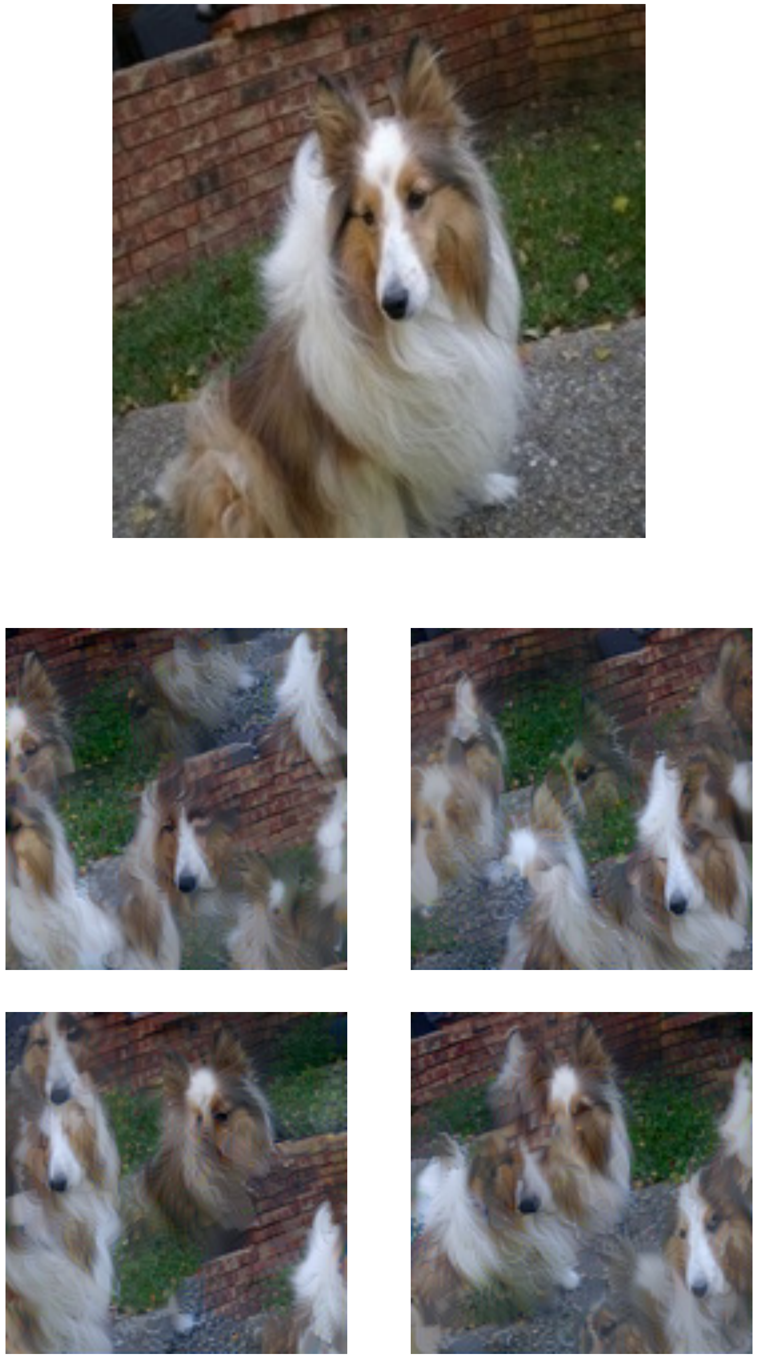
Leon Gatys http://arxiv.org/abs/1505.07376 Alexander Ecker

Classification of texturized images

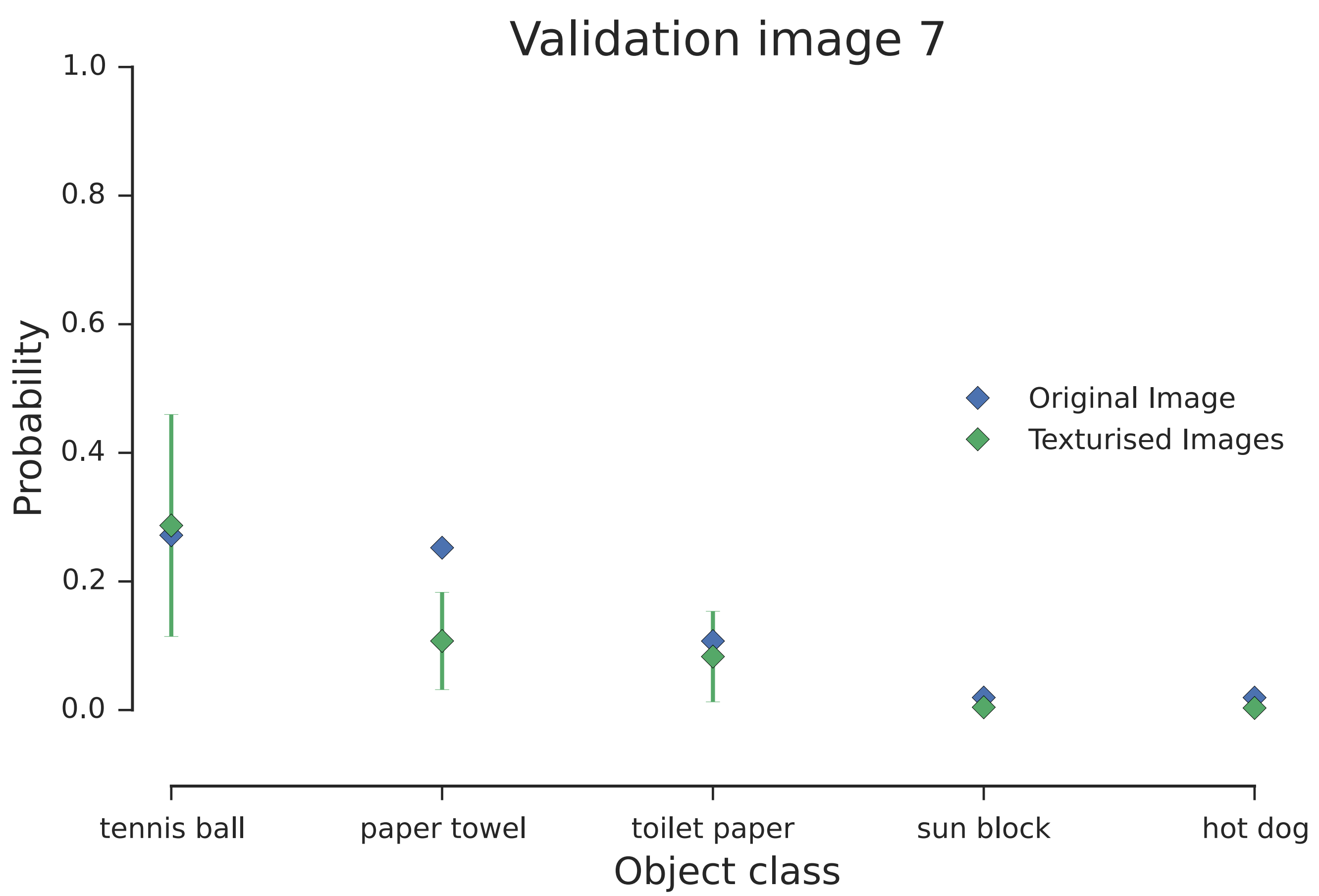


Classification of texturized images

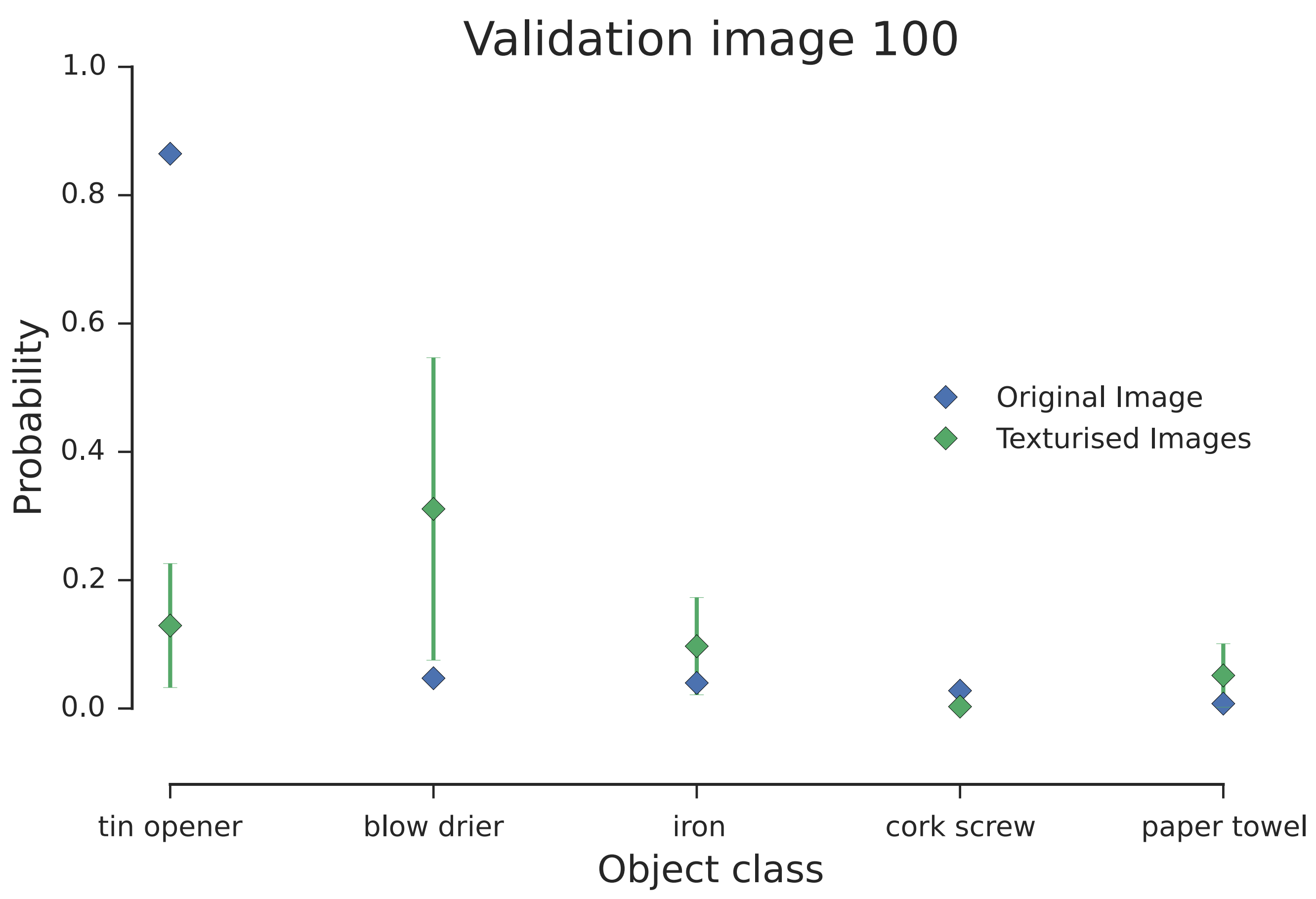




Classification of texturized images



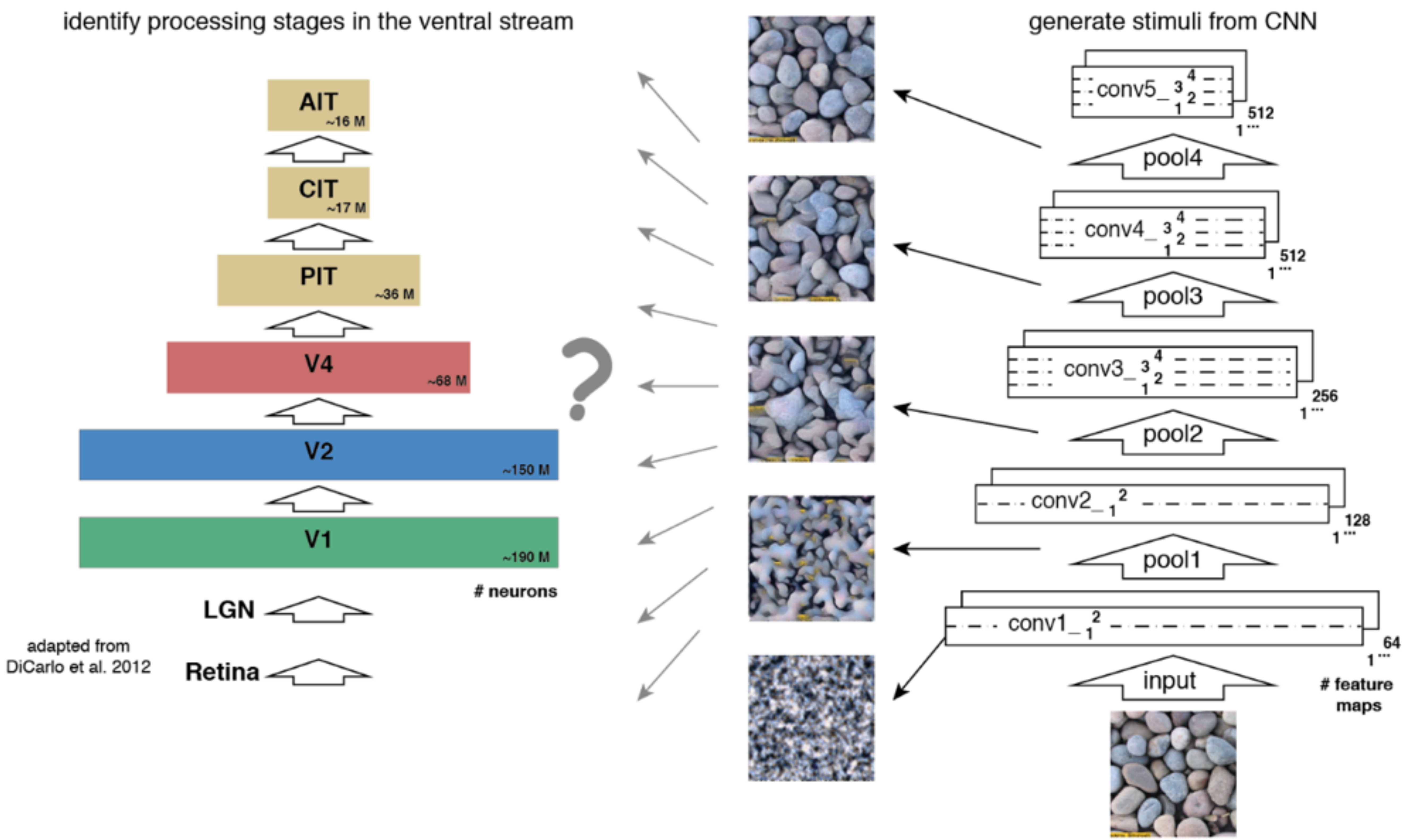
Classification of texturized images



Texture Synthesis - Summary

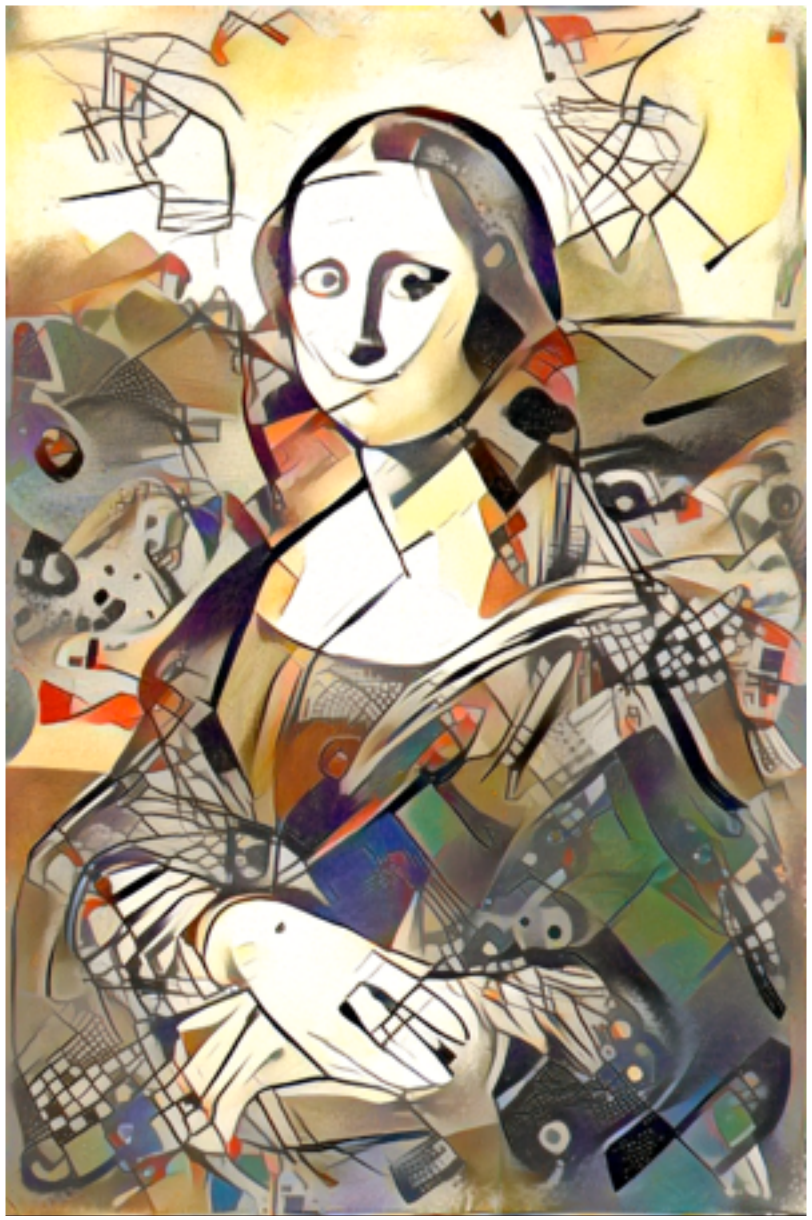
- CNN texture model sets a new state of the art in parametric texture synthesis.
- Texture features disentangle object identity information along the layers
- Textures from non-texture images map out the spatial invariance of the network’s classification response.

Controlled stimulus design (model-matched)



http://arxiv.org/abs/1505.07376

Neural Image Representations



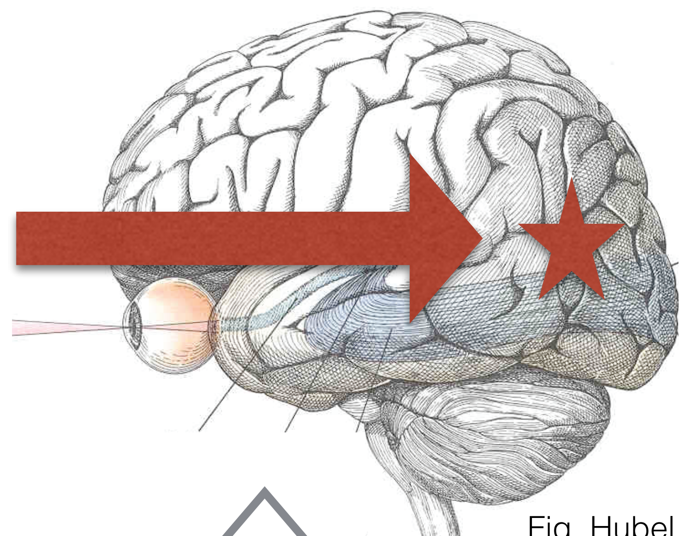


Fig. Hubel 95

Semantic Image Understanding

Correspondence

Yamins 2014, Cadieu 2014,
van Gerven 2015

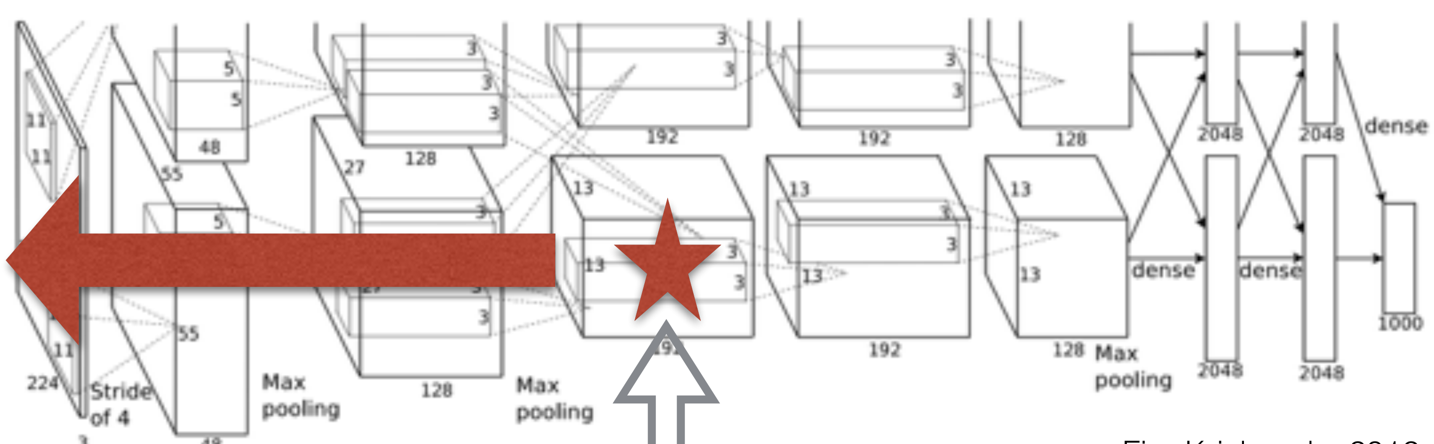
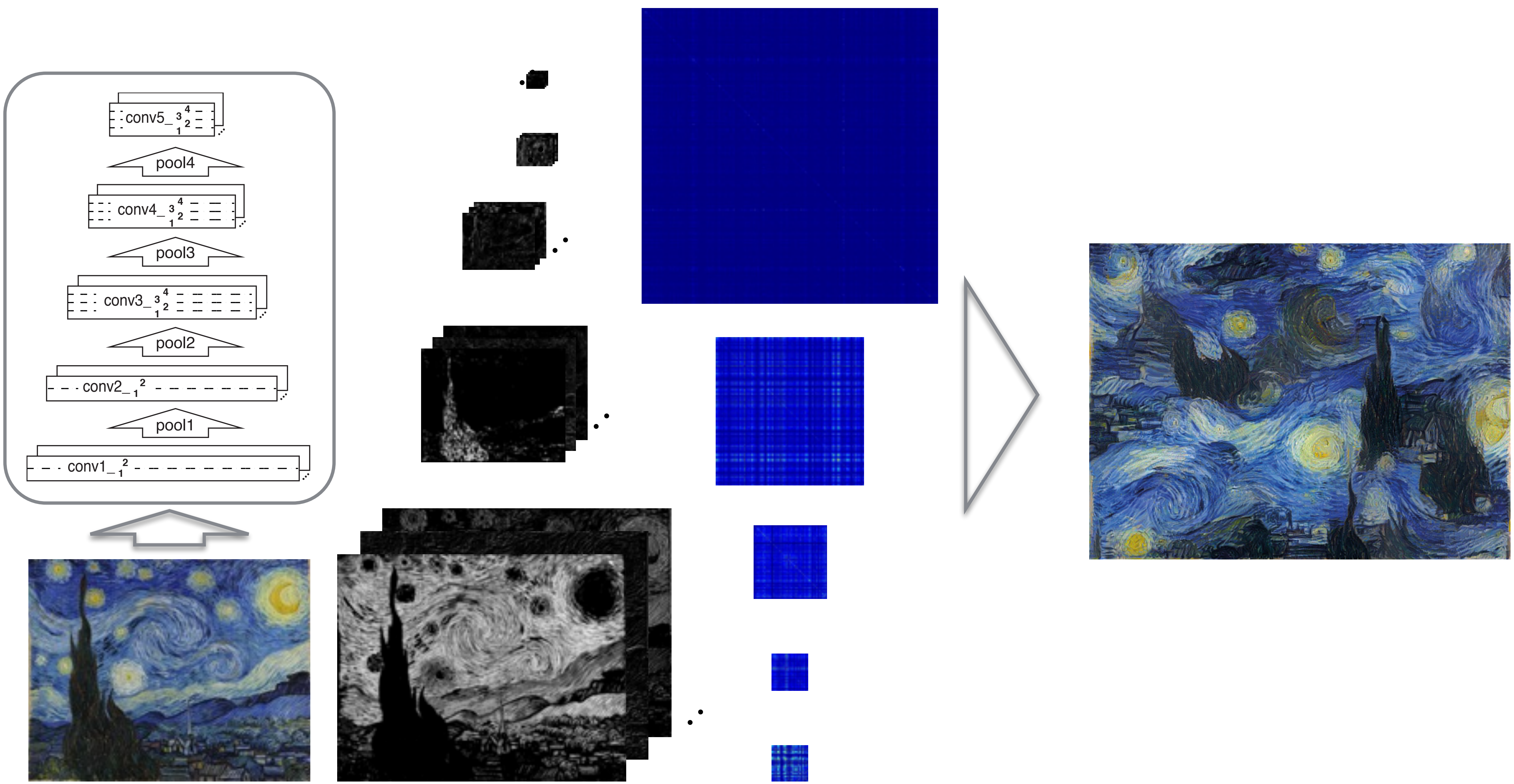


Fig. Krizhevsky 2012

Induce Change

CNN - Texture Synthesis



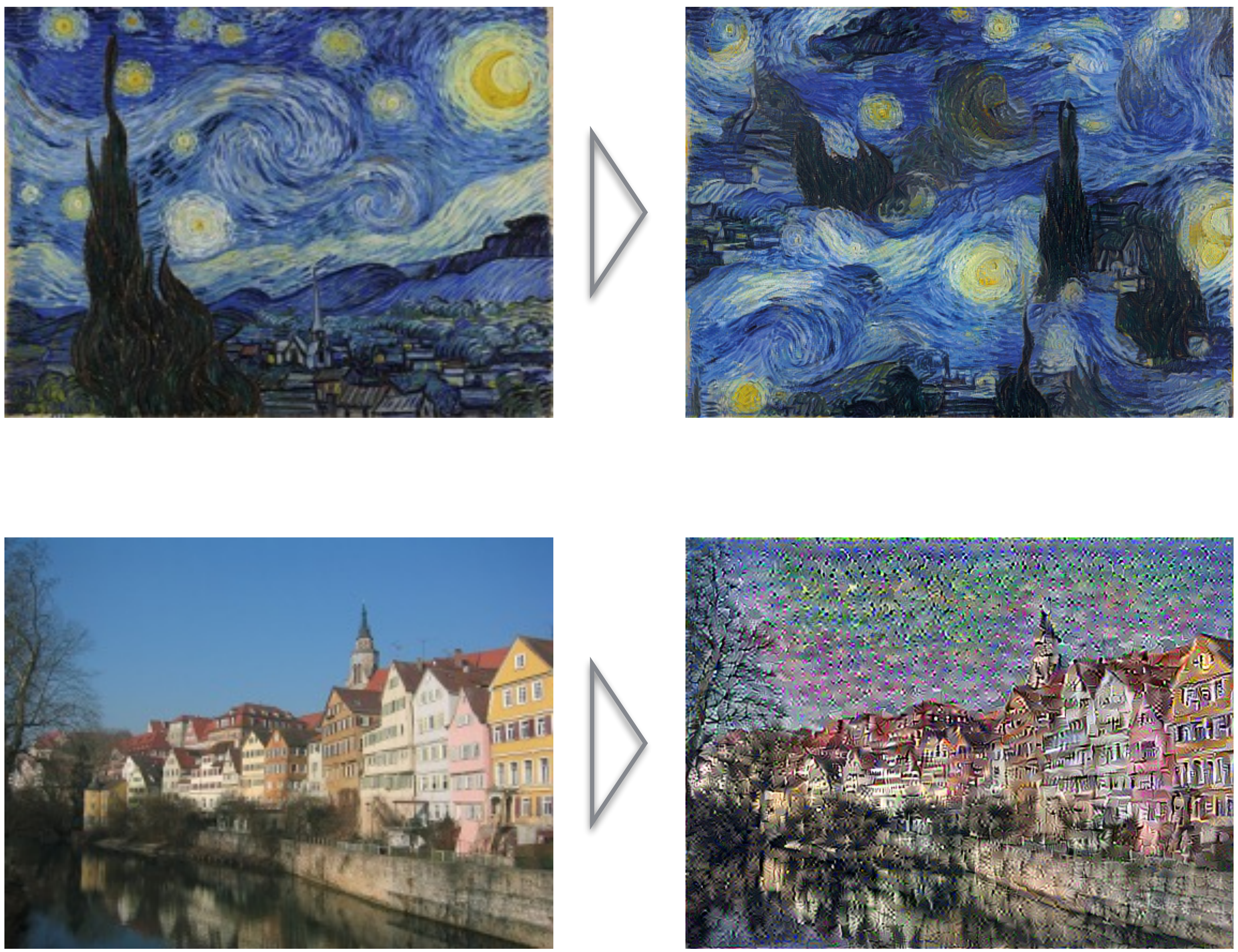
Gatys et al. (NIPS 2015)

Representation of stimulus information

Mahendran & Vedaldi (2014), Gatys et al (2015)

Disentangling content and style

$$\begin{pmatrix} \langle \bar{f}_1, \bar{f}_1 \rangle & \cdots & \langle \bar{f}_1, \bar{f}_N \rangle \\ \langle \bar{f}_2, \bar{f}_1 \rangle & & \vdots \\ \vdots & \ddots & \vdots \\ \langle \bar{f}_N, \bar{f}_1 \rangle & \cdots & \langle \bar{f}_N, \bar{f}_N \rangle \end{pmatrix}$$

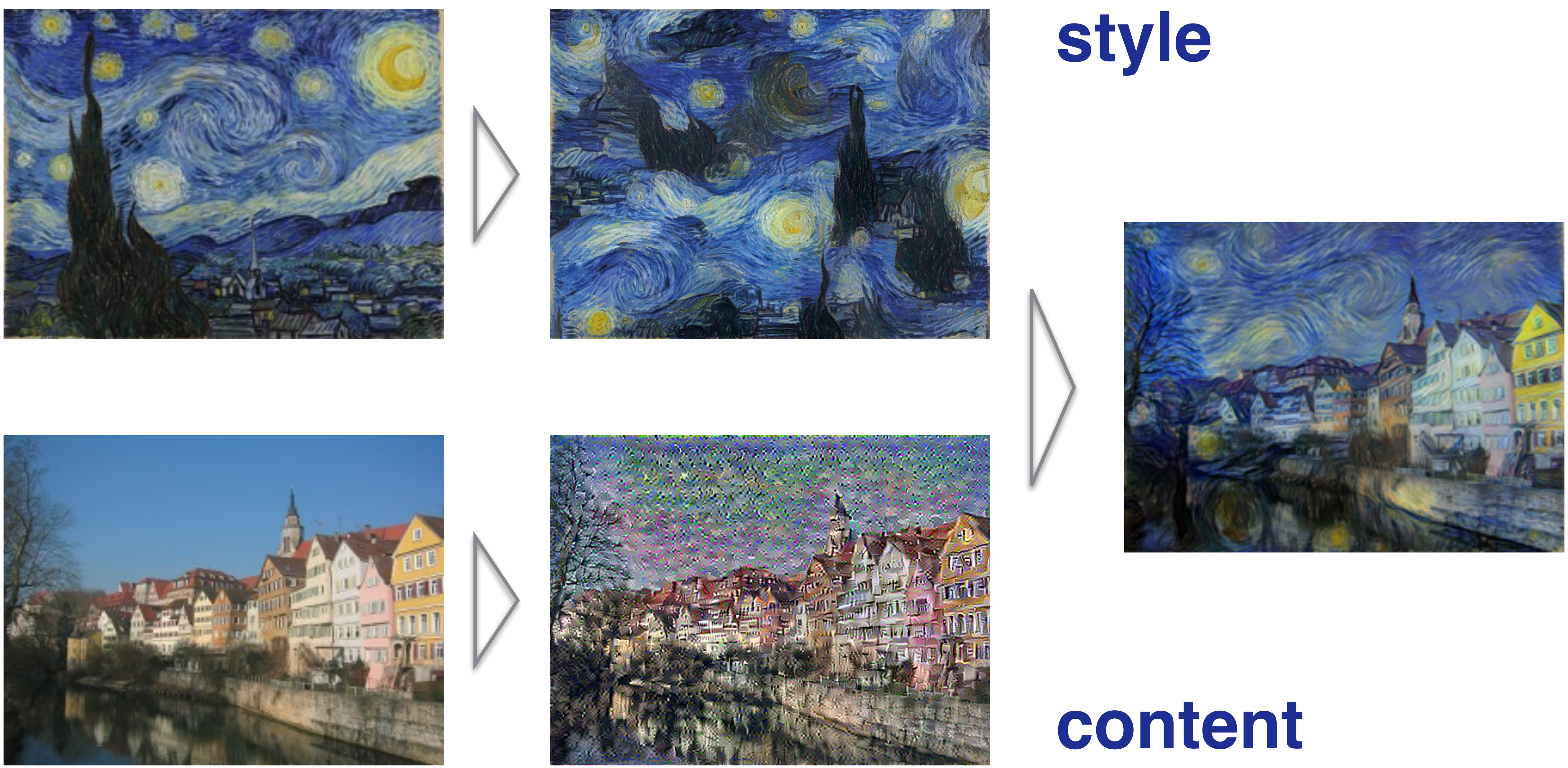


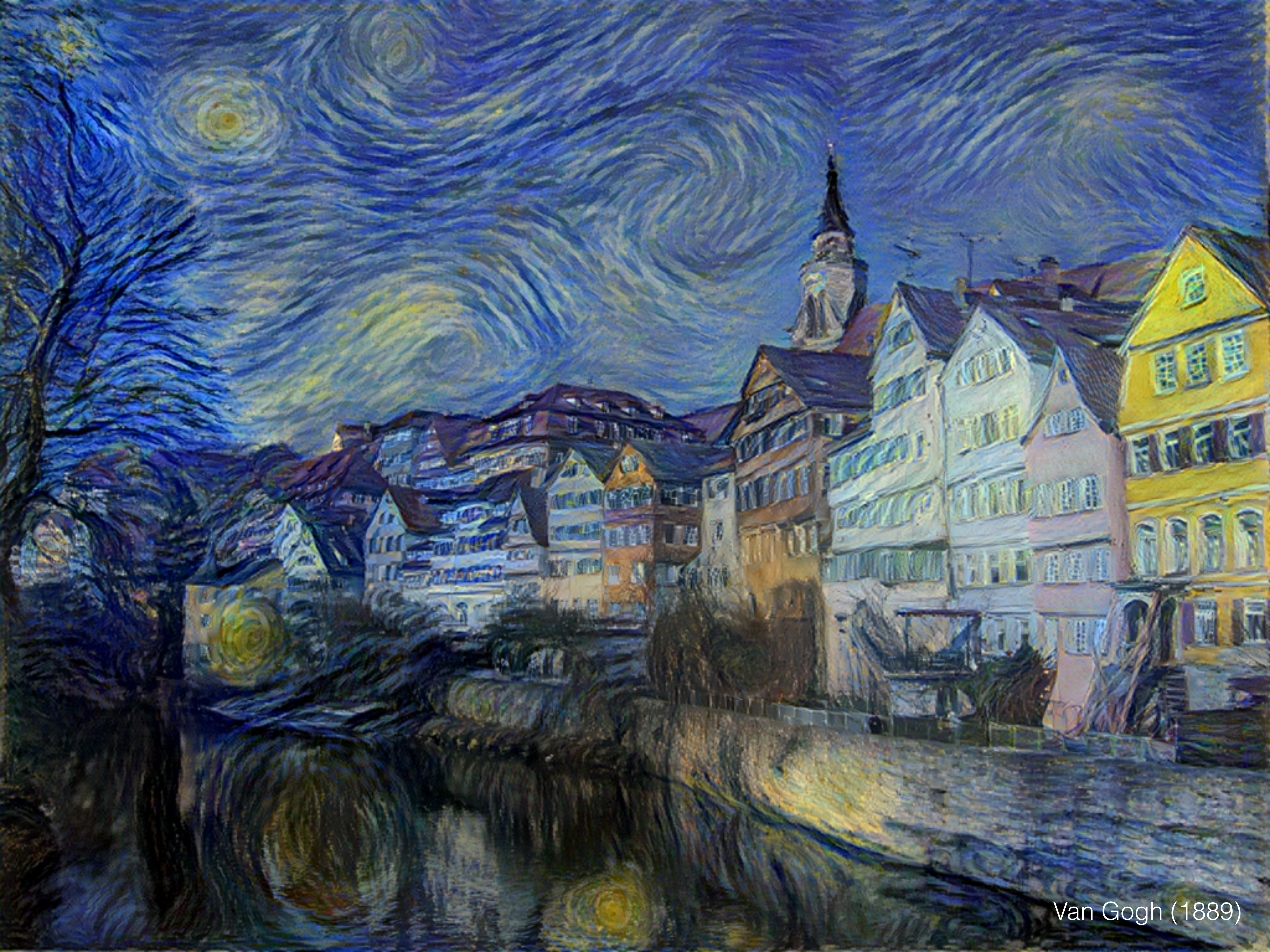
style

content

Disentangling content and style

$$\begin{pmatrix} \langle \bar{f}_1, \bar{f}_1 \rangle & \cdots & \langle \bar{f}_1, \bar{f}_N \rangle \\ \langle \bar{f}_2, \bar{f}_1 \rangle & & \vdots \\ \vdots & \ddots & \vdots \\ \langle \bar{f}_N, \bar{f}_1 \rangle & \cdots & \langle \bar{f}_N, \bar{f}_N \rangle \end{pmatrix}$$





Van Gogh (1889)

Picasso (1910)


Munch (1893)

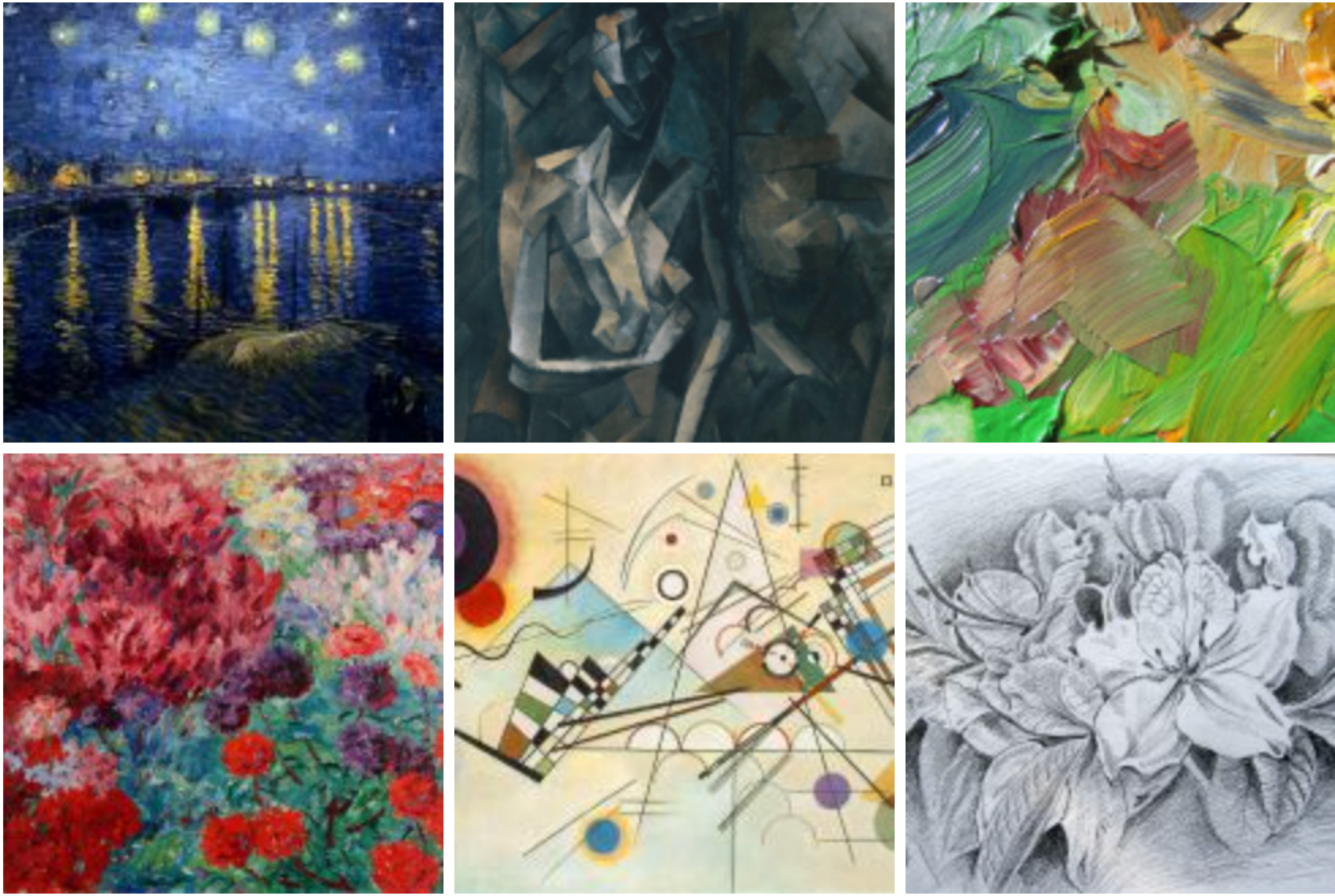
Turner (1805)

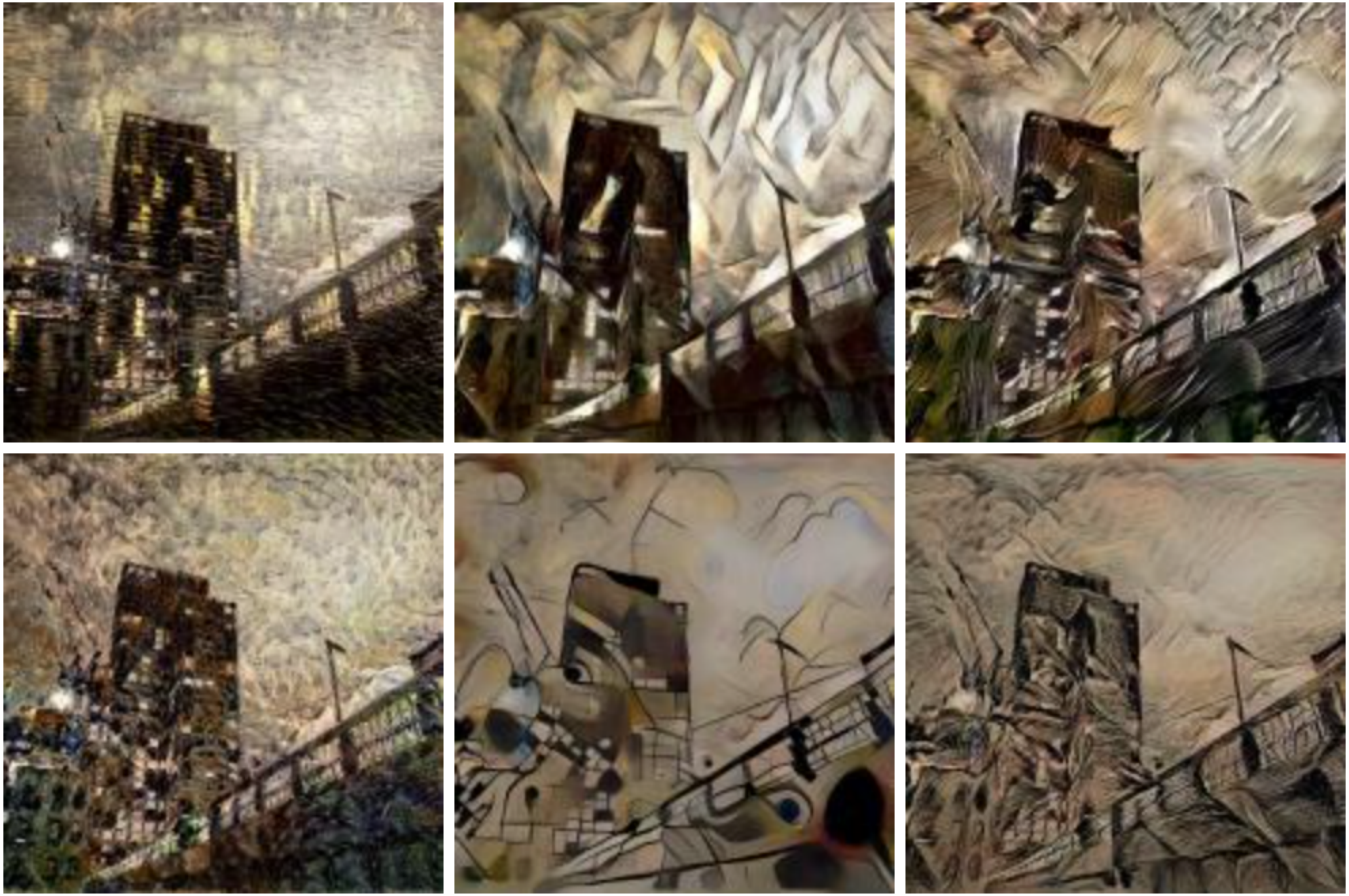
Kandinsky (1913)

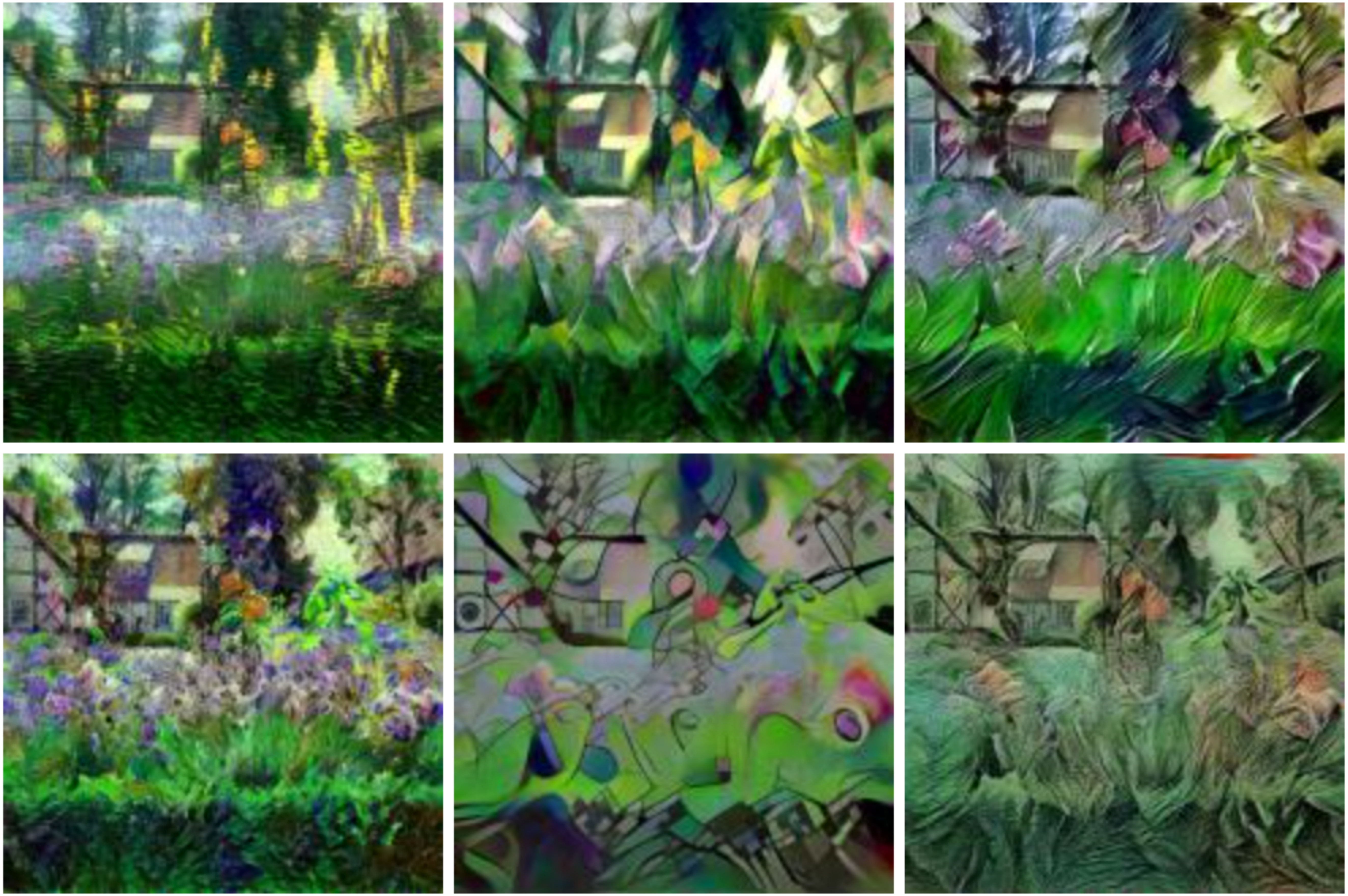
Tübingen, Neckar front

General Style Transfer

Color independent style transfer





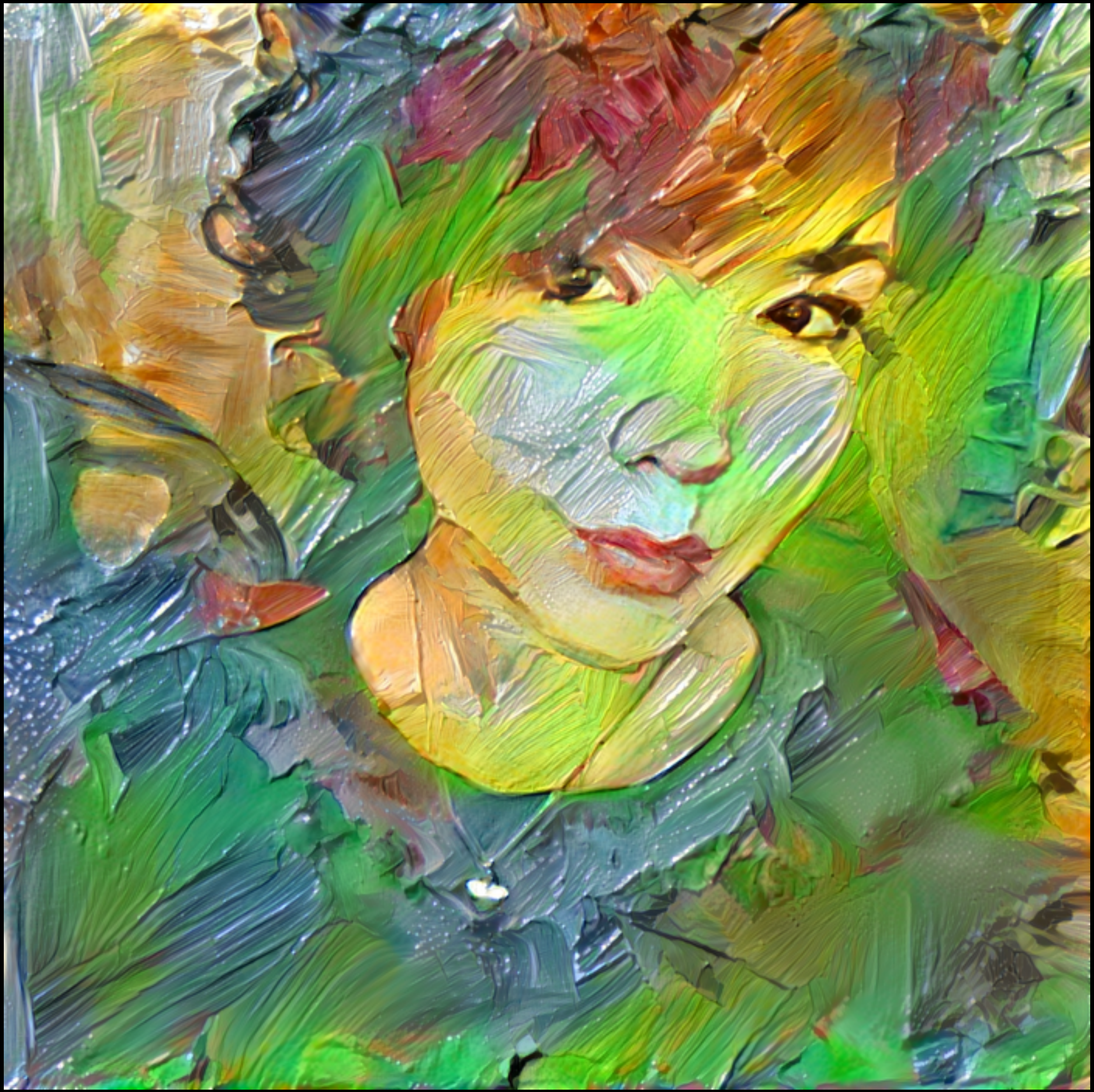


https://deepart.io

DEEPART.io How to Pricing Latest images Create your own About Register Sign in

HOW IT WORKS

Our algorithm is inspired by the human brain. It uses the stylistic elements of one image to draw the content of another. Get your own artwork in just three steps.

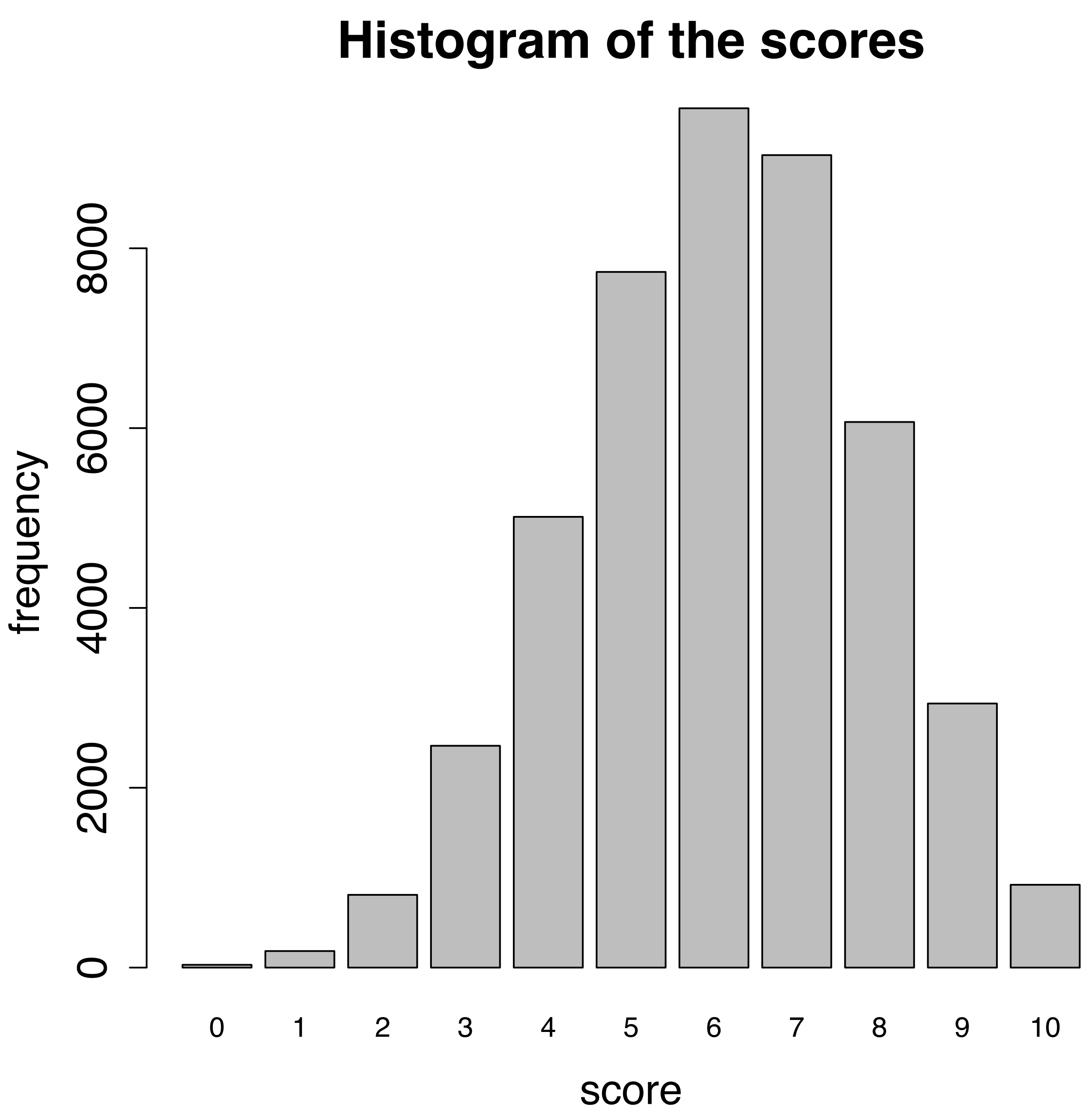


DeepArt - Visual Turing Test

DeepArt - Visual Turing Test

DeepArt - Visual Turing Test

Visual Turing Test - Results



~45.000 people, average score: 6.1 (chance: 5)

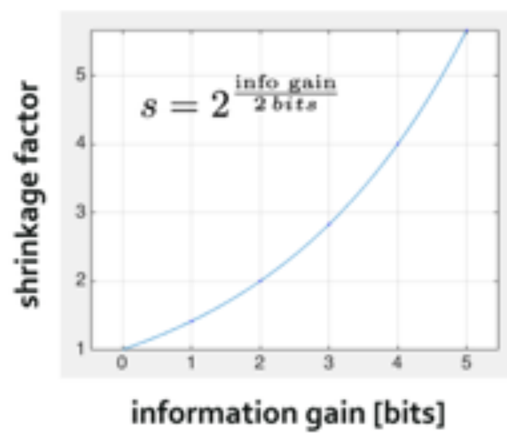
Wrap-up

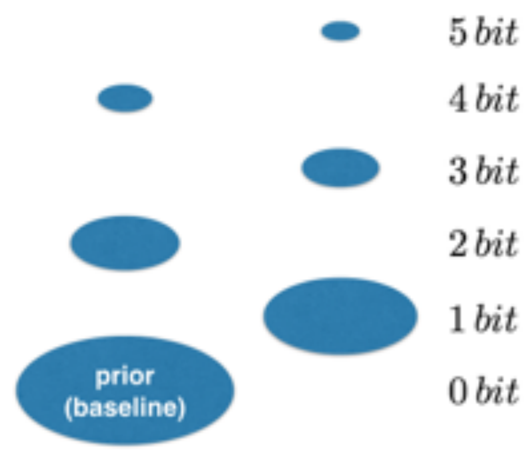
Transfer Learning:

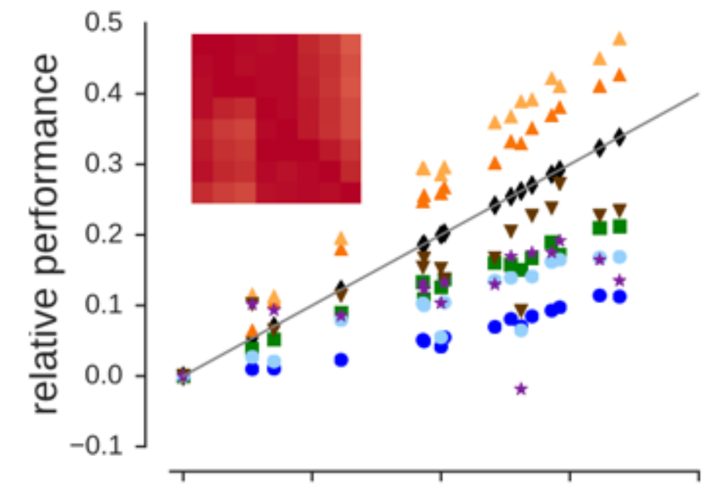
IMAGENET

Information gain:

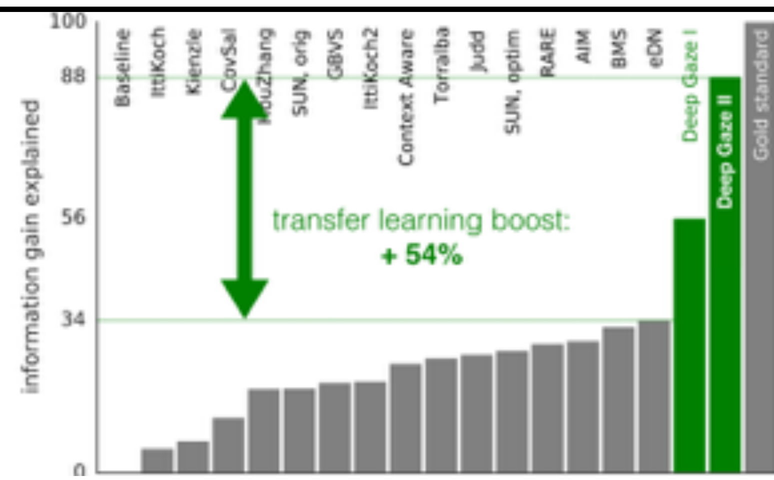
$$E\left[\log\left(\frac{\rho_M(x,y|I)}{\rho_{base}(x,y)}\right)\right]$$

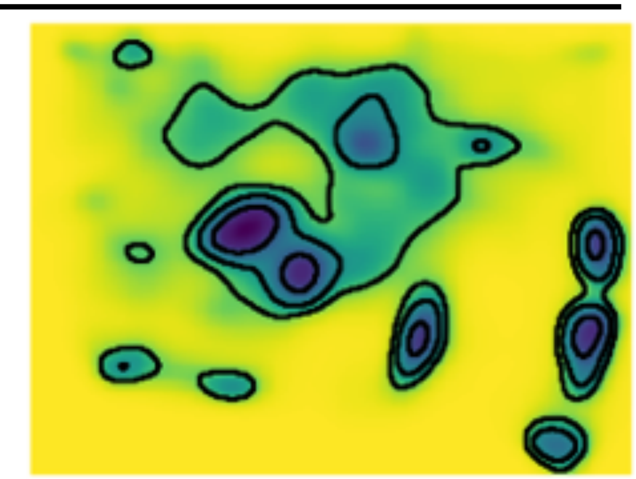




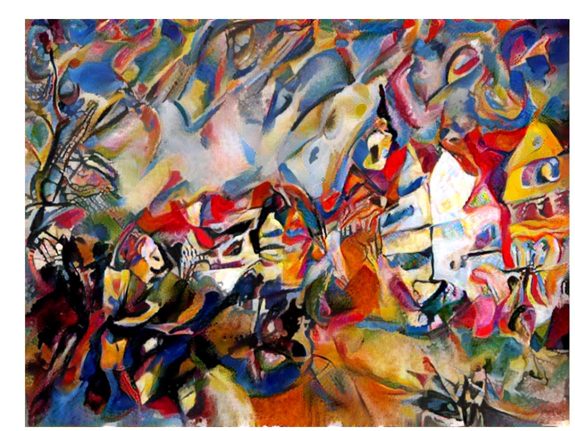


Transfer Learning Deep Gaze II:



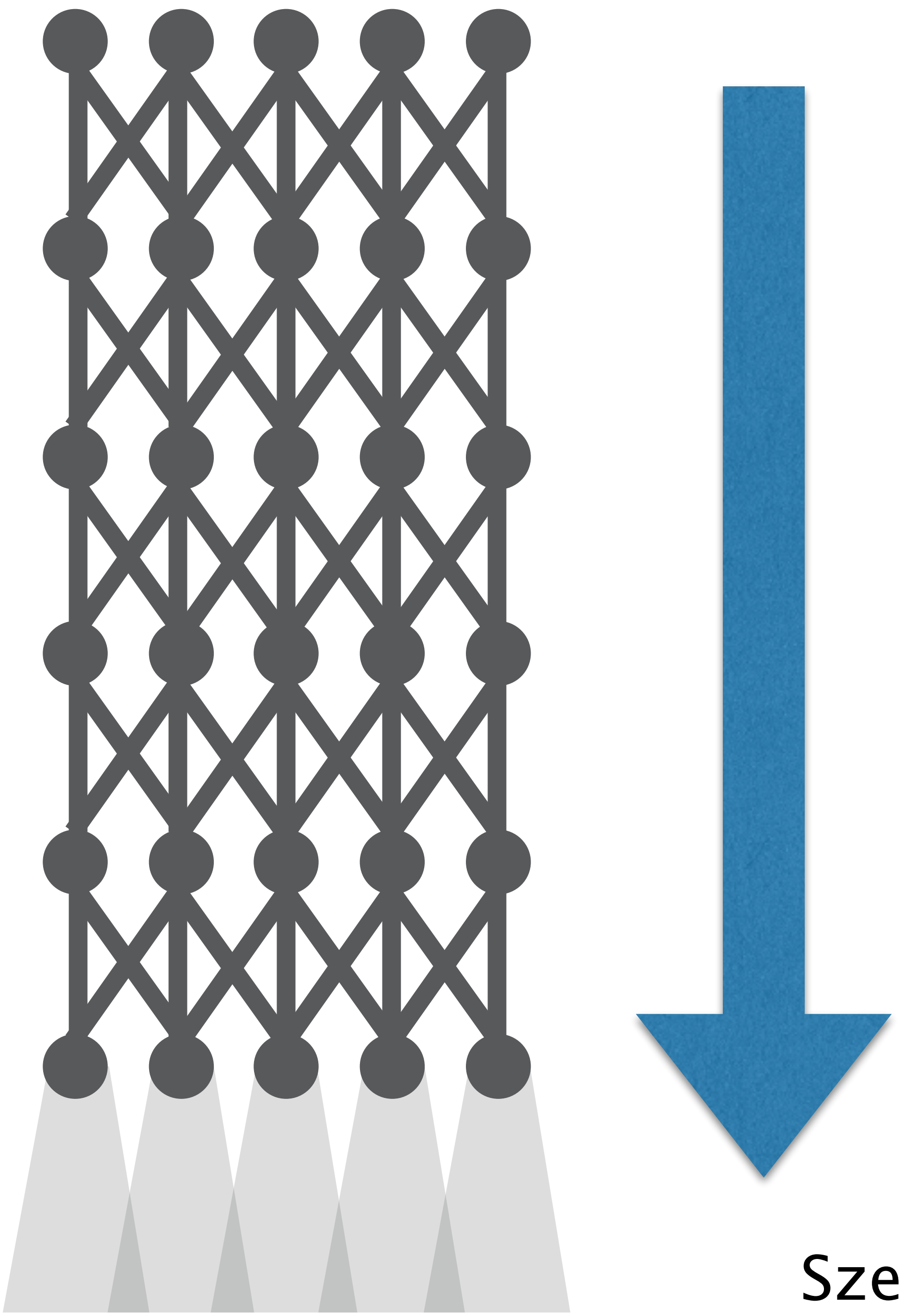


Texture synthesis & image manipulation with CNNs:



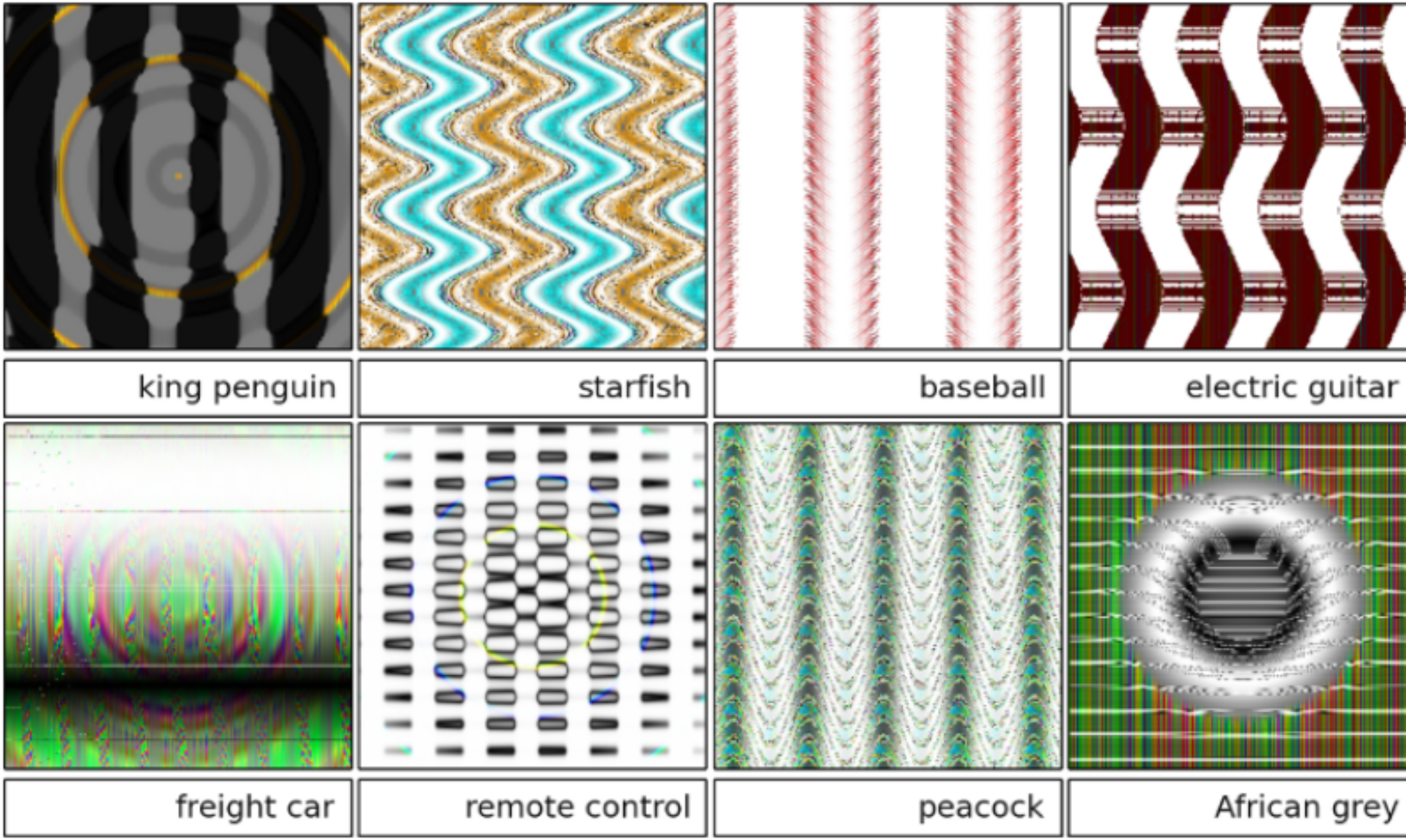
Important Open Problems

Important challenge I



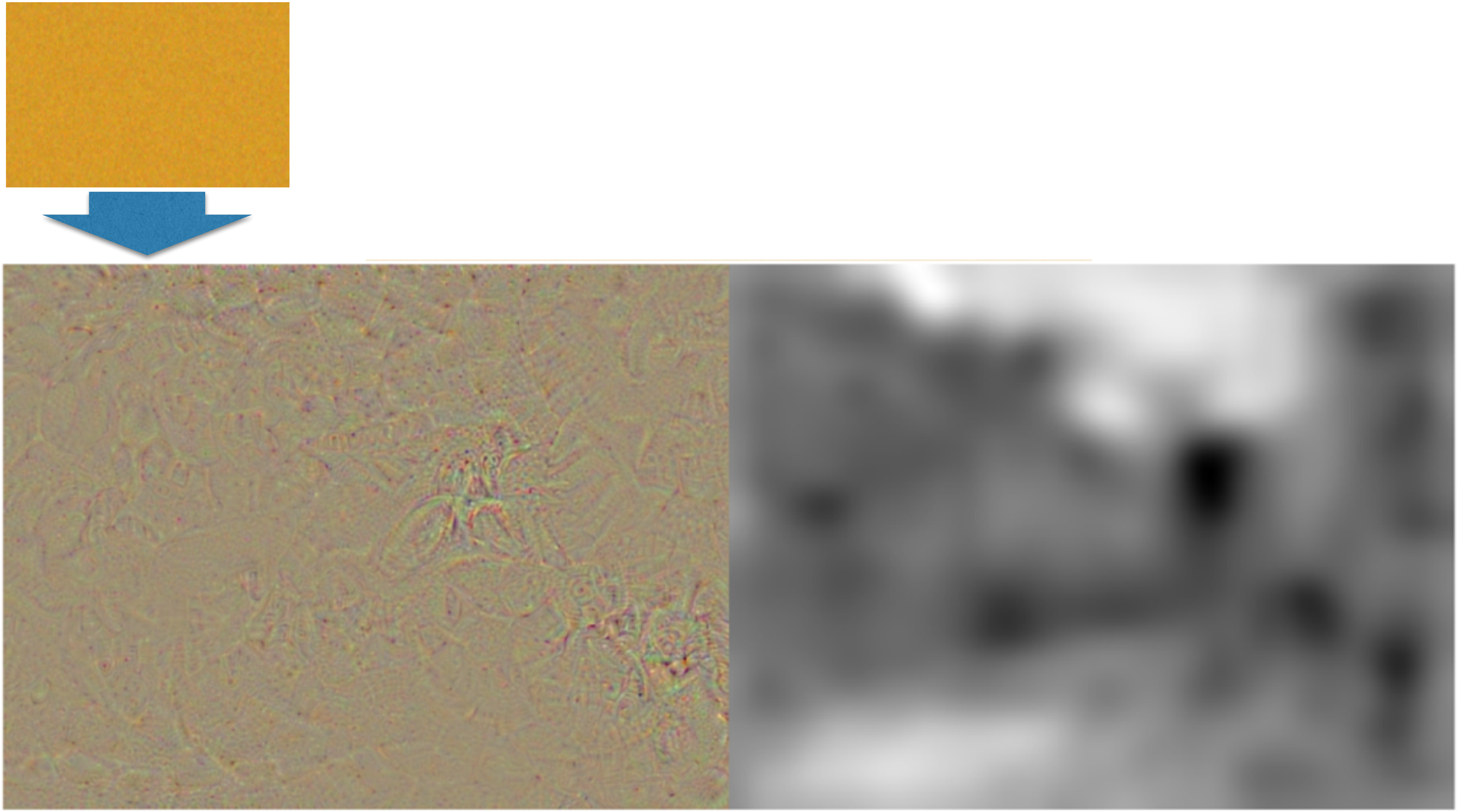
Can we build DNNs for which it is hard to construct adversarial examples?

Szegedy et al, Intriguing properties of neural networks
http://arxiv.org/abs/1312.6199 (2013)

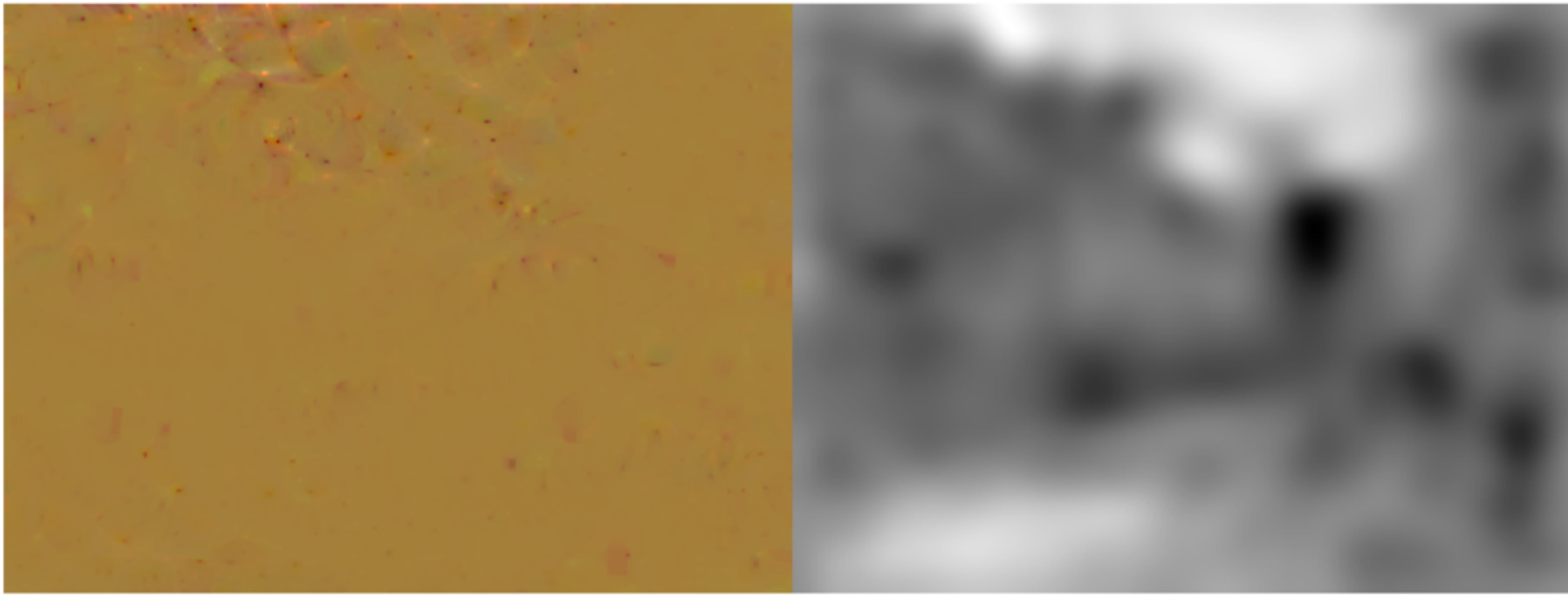


Nguyen et al, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. (2015)

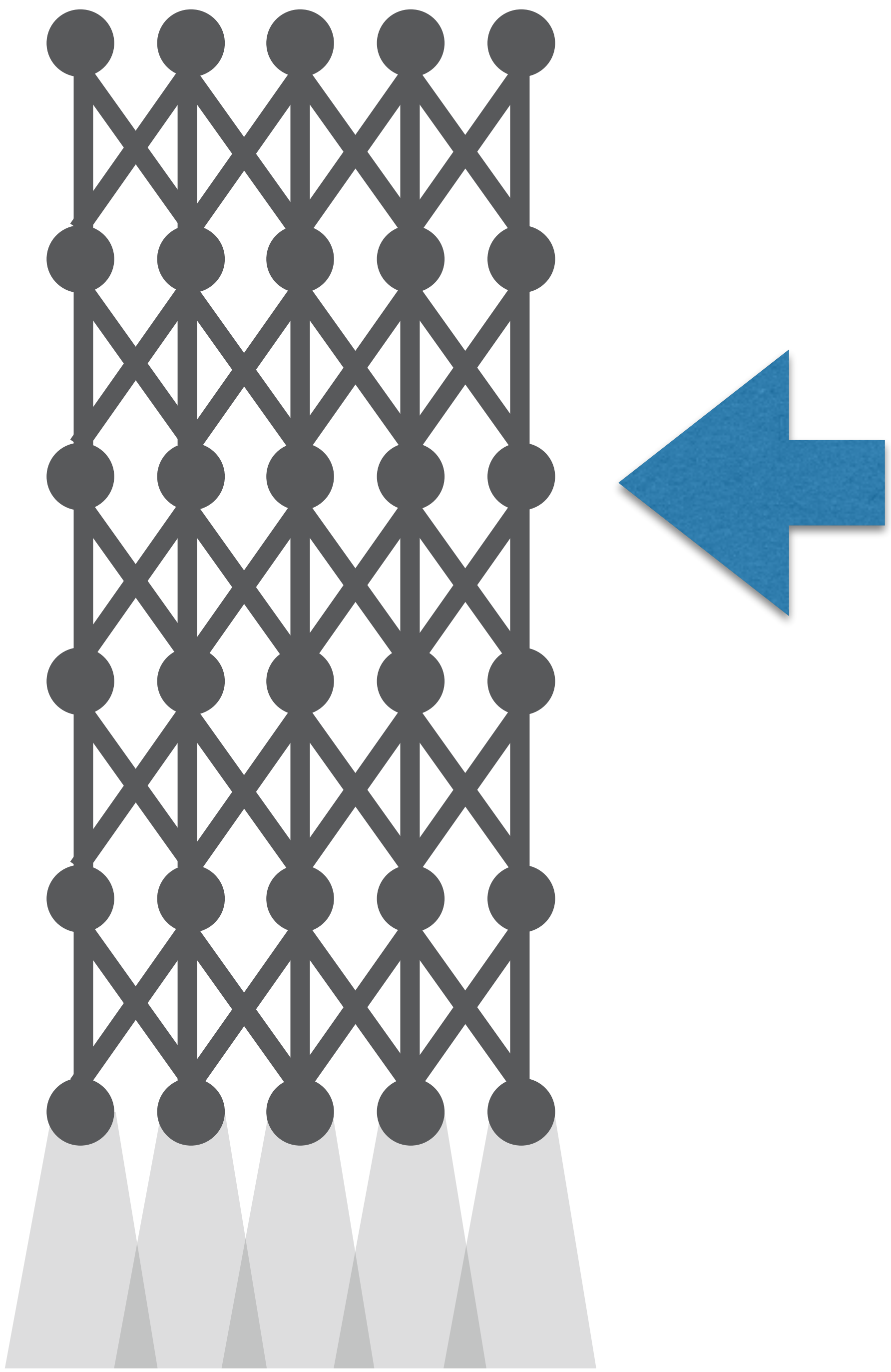

P
ONE WAY

+L1 penalty

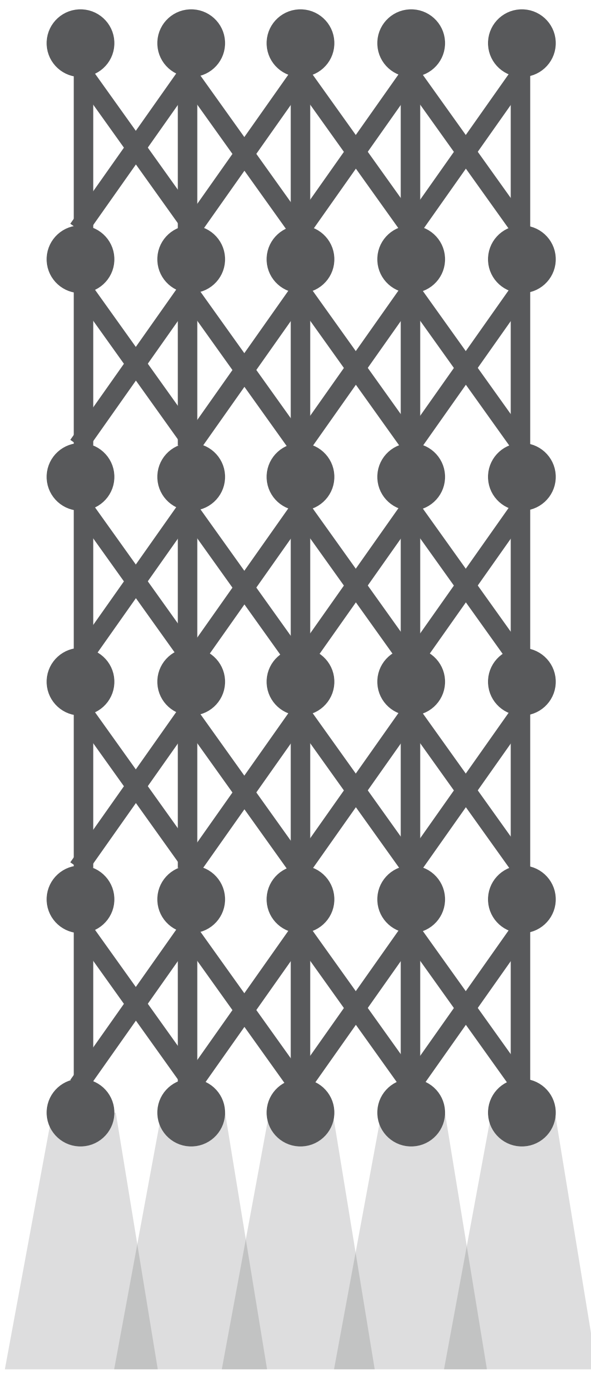


Important challenge II



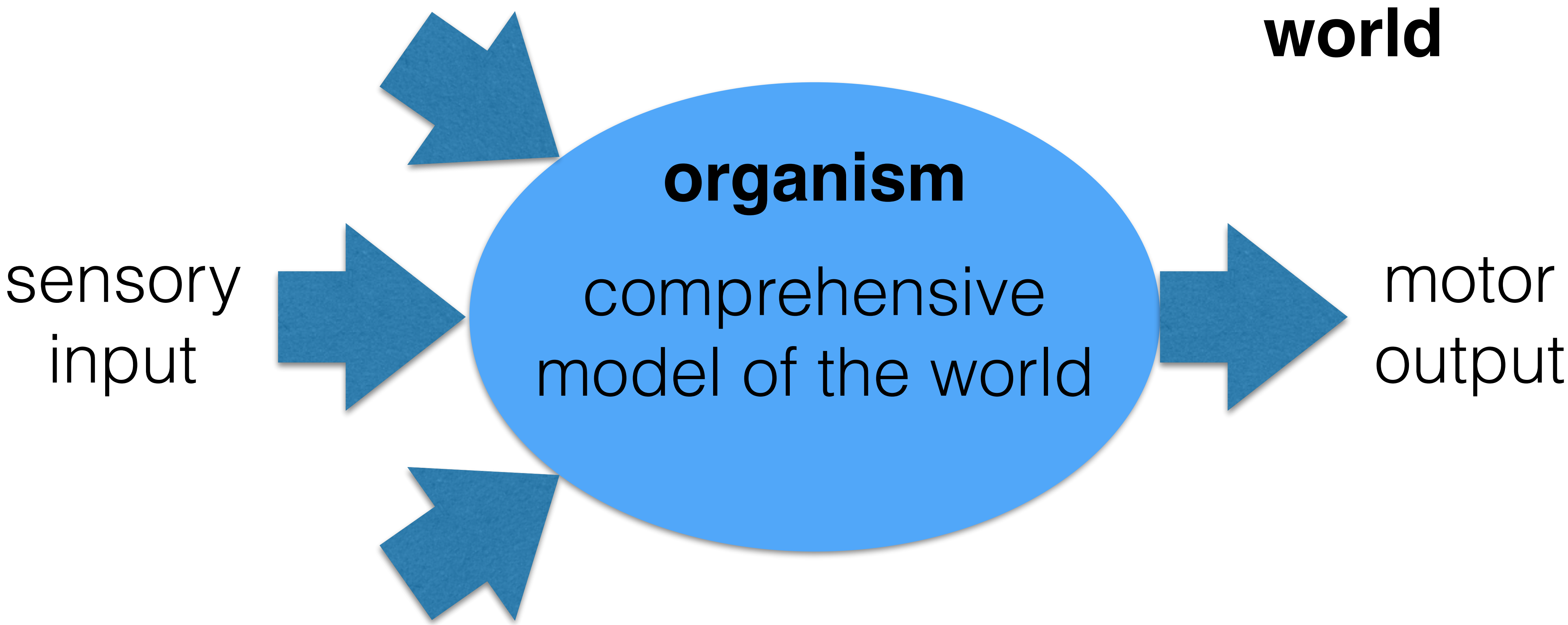
Can we learn DNNs with semantically meaningful intermediate layers?

Important challenge III

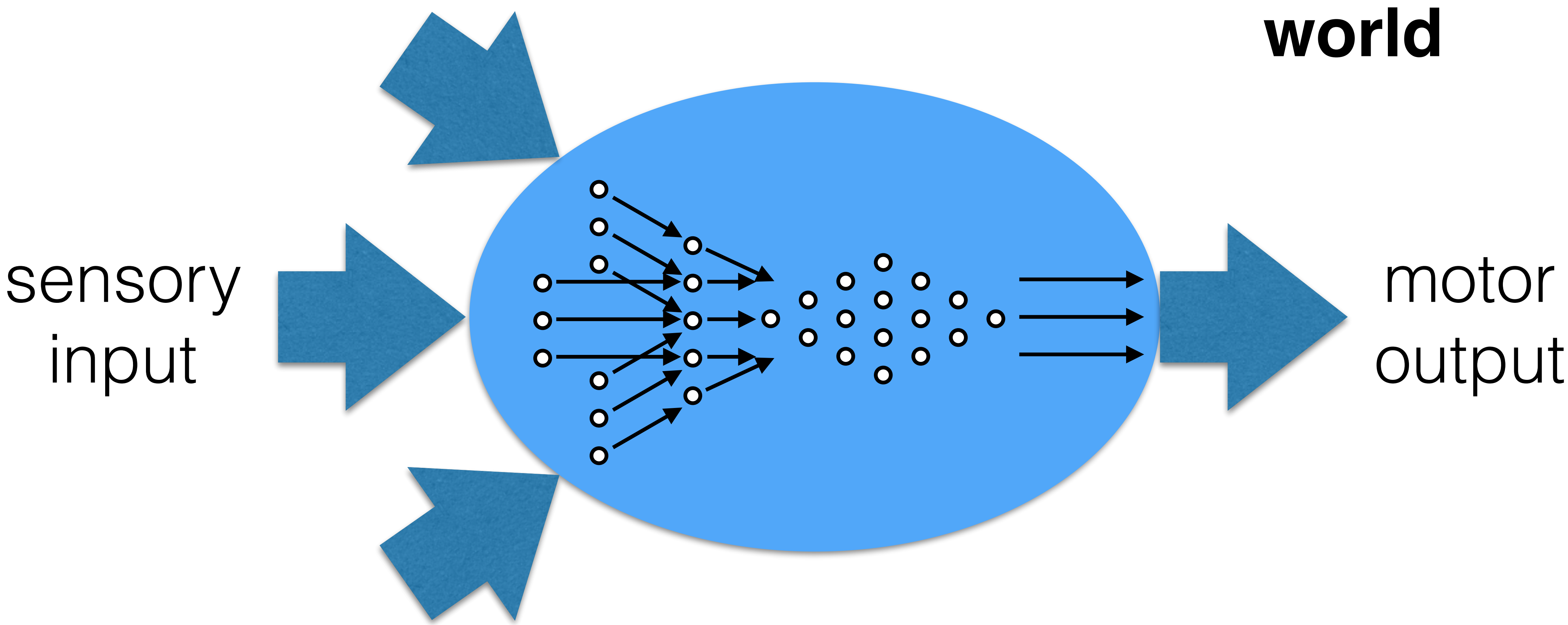


Can we build high-performing DNNs without learning?

Intelligent Systems



Intelligent Systems



multi-task DNNs

pattern recognition

(narrow AI)

holistic perceptual
representation of the world
(general AI)

Thanks!

[DeepArt.io]
[bethgelab.org/deeptextures]

Thanks!

DeepArt.io
bethgelab.org/deeptextures

Thanks!

DeepArt.io
bethgelab.org/deeptextures

Enjoy MLSS in Cadiz!

DeepArt.io
bethgelab.org/deeptextures