# Perception: The Next Milestone in Learned Image Compression

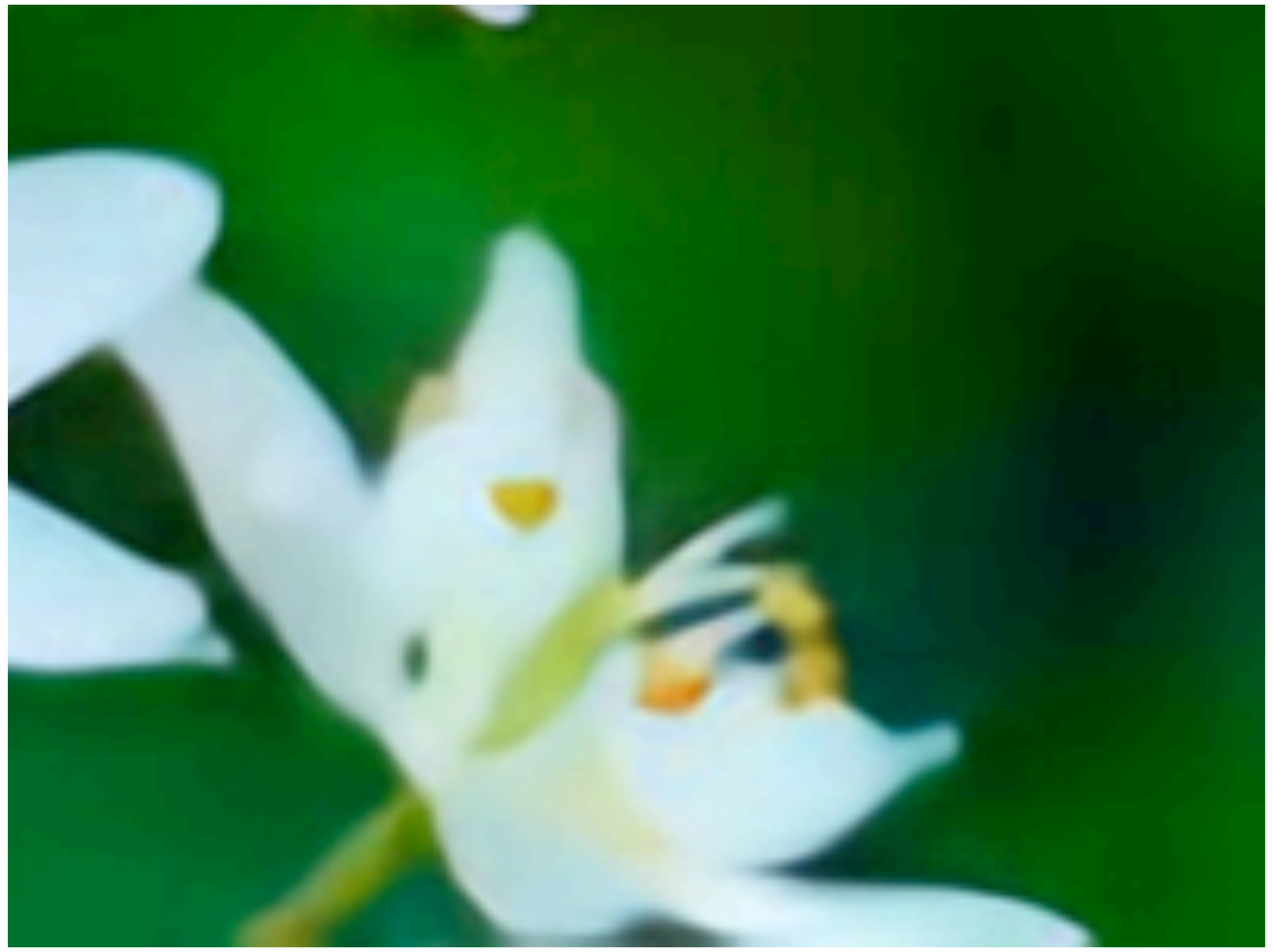
Dr. Johannes Ballé (they/them), Staff Research Scientist

2023 Data Compression Conference
22 March 2023

Google Research

original

learned (2017)



JPEG

JPEG 2000

original

learned (2017)

JPEG

JPEG 2000

# Machine learning rings in a new paradigm in data compression

Learned compression is **data driven**, has **quick turnaround** and is **easily adaptable**.

It presents opportunities to quickly develop algorithms for **new data modalities**, as well as **sophisticated error metrics**.

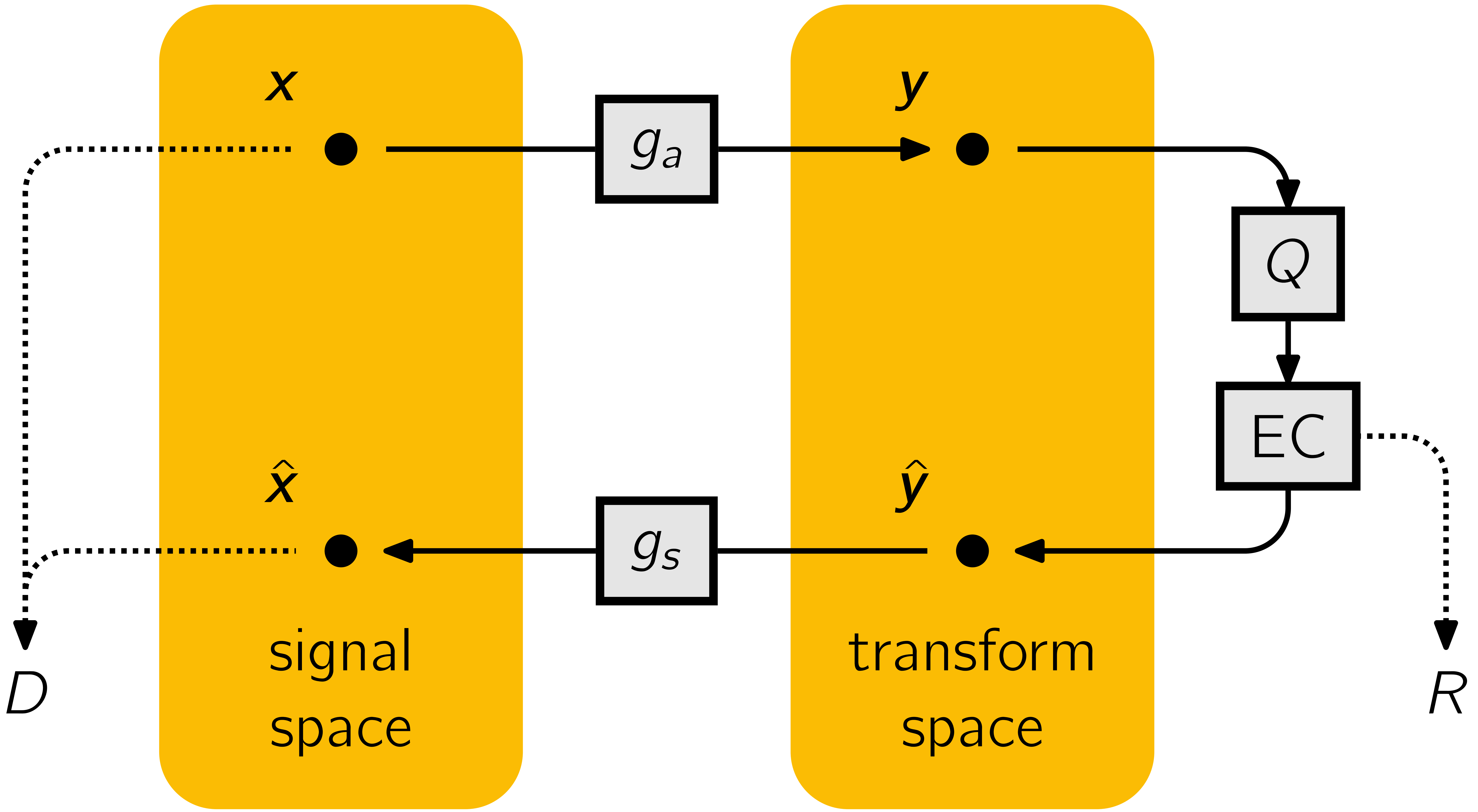
# Outline

1. Learned image compression
2. Distortion
3. Realism
4. Perceptual spaces

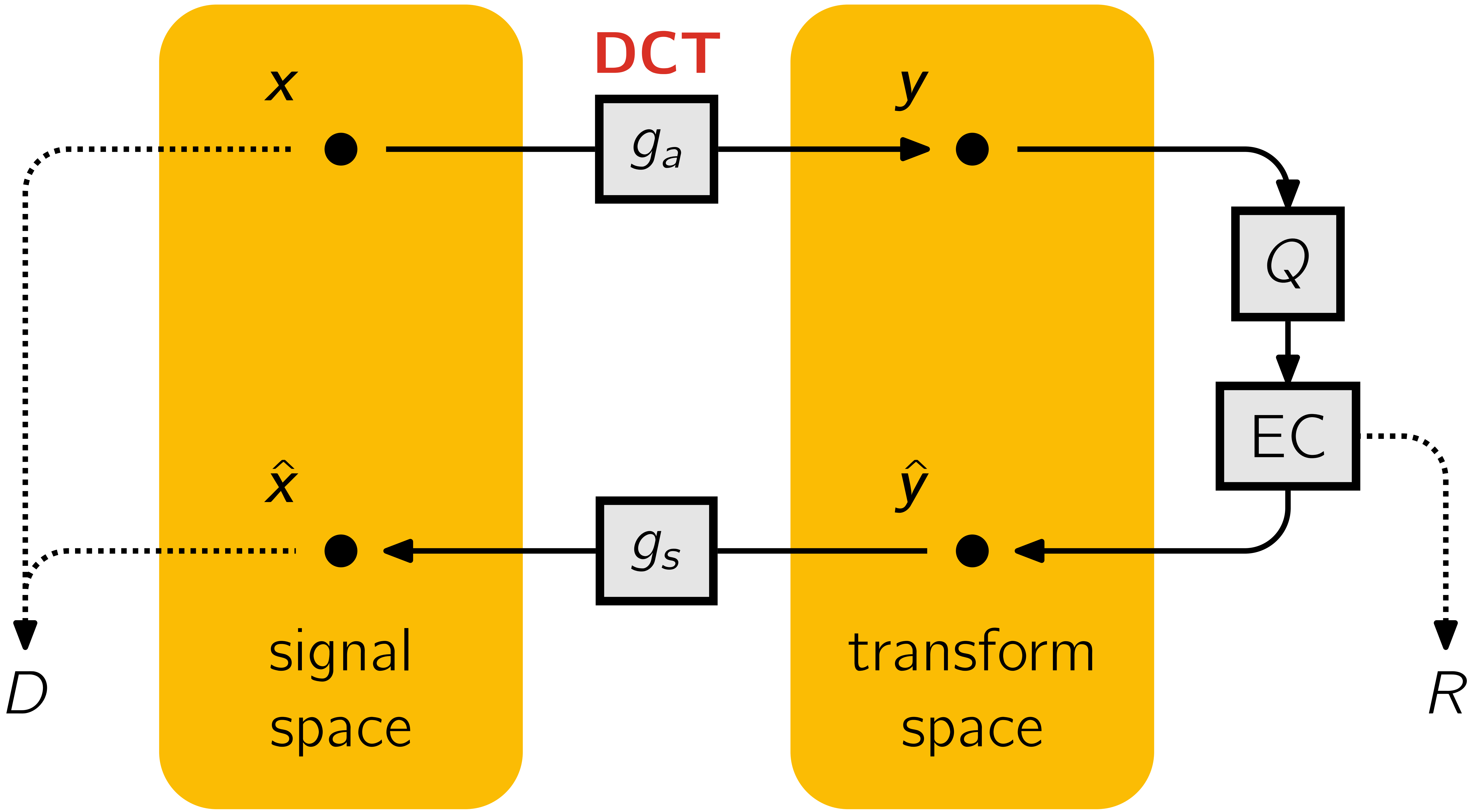
## Part I

# Learned Image Compression

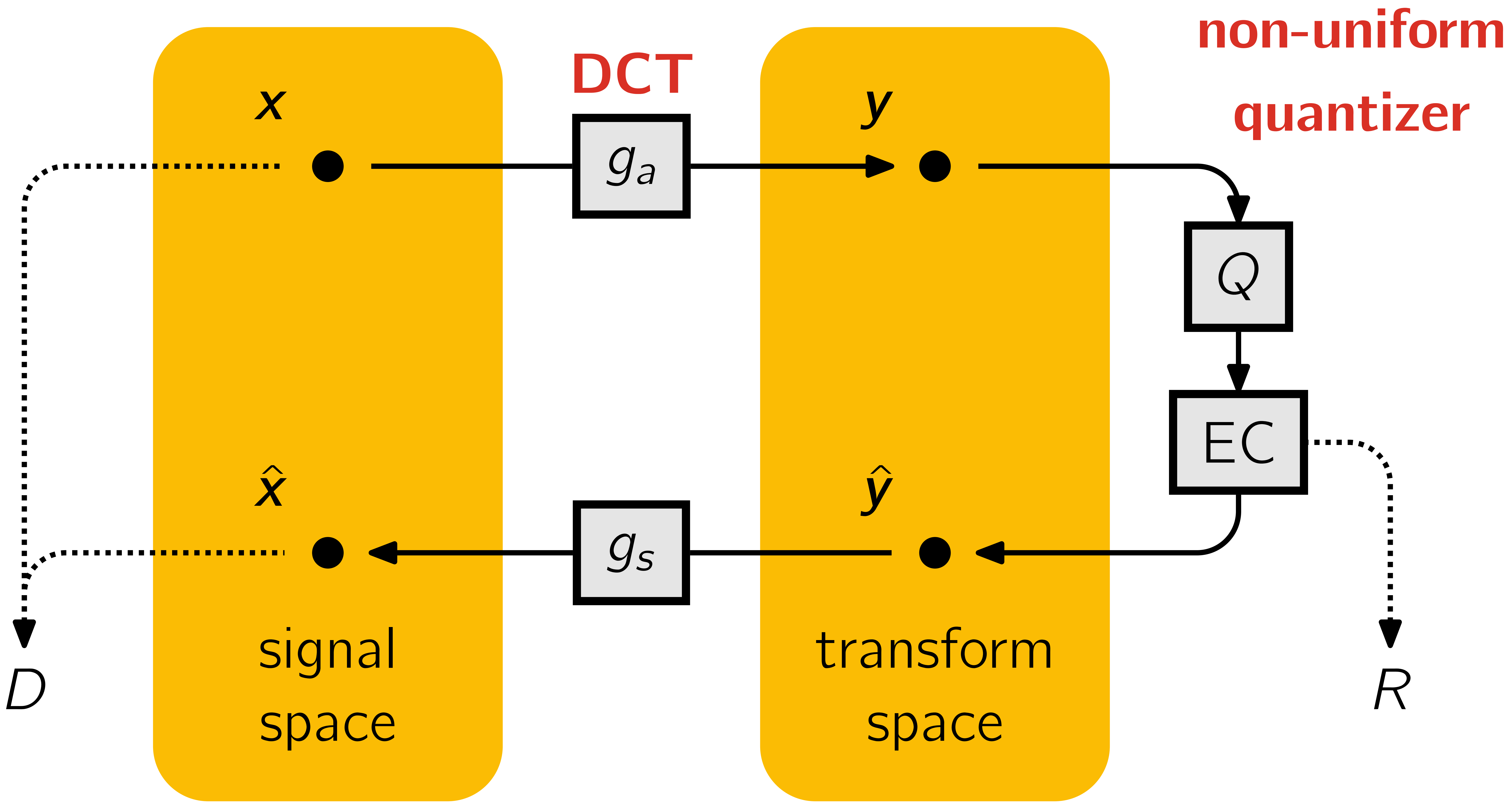
# JPEG, in a nutshell



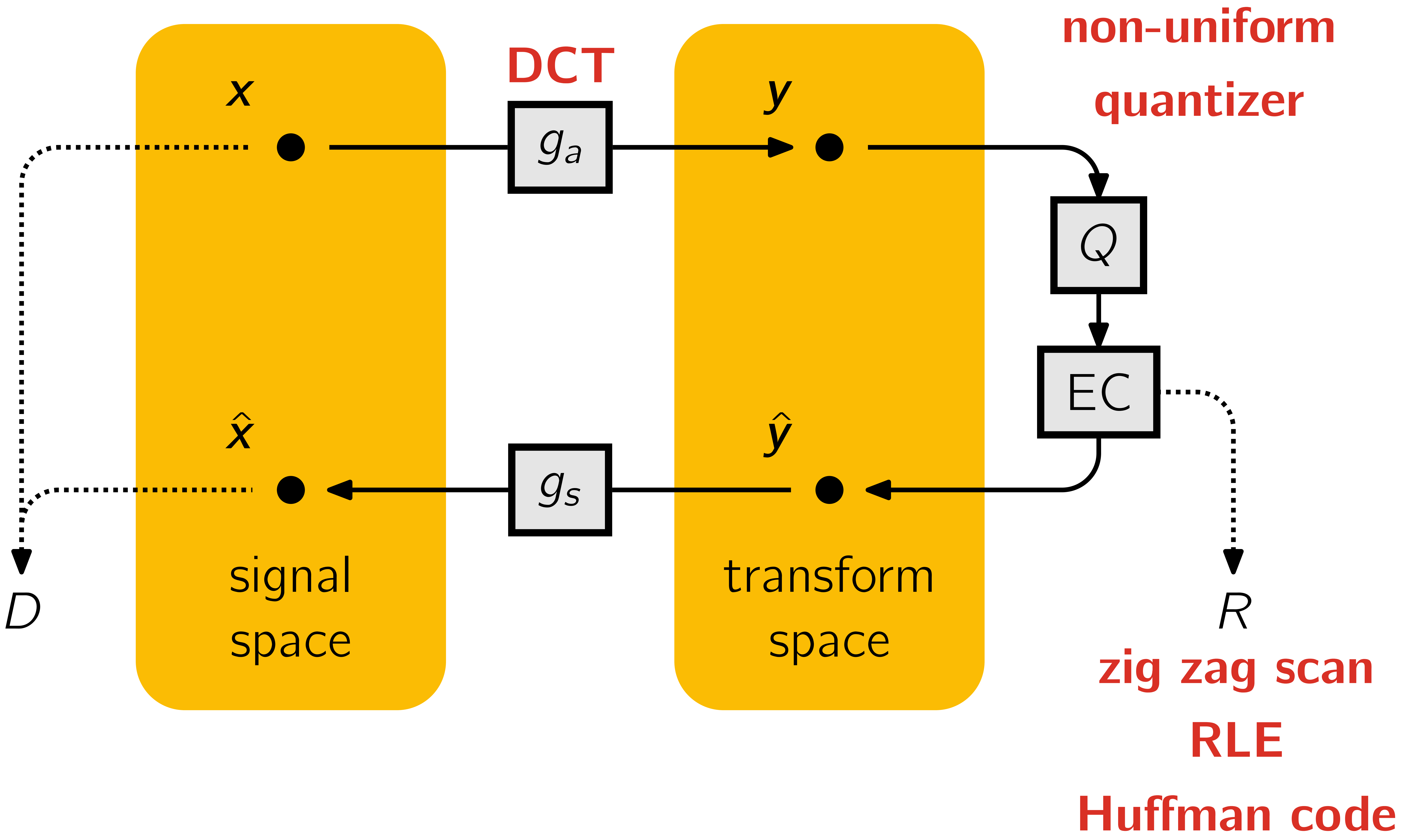
# JPEG, in a nutshell



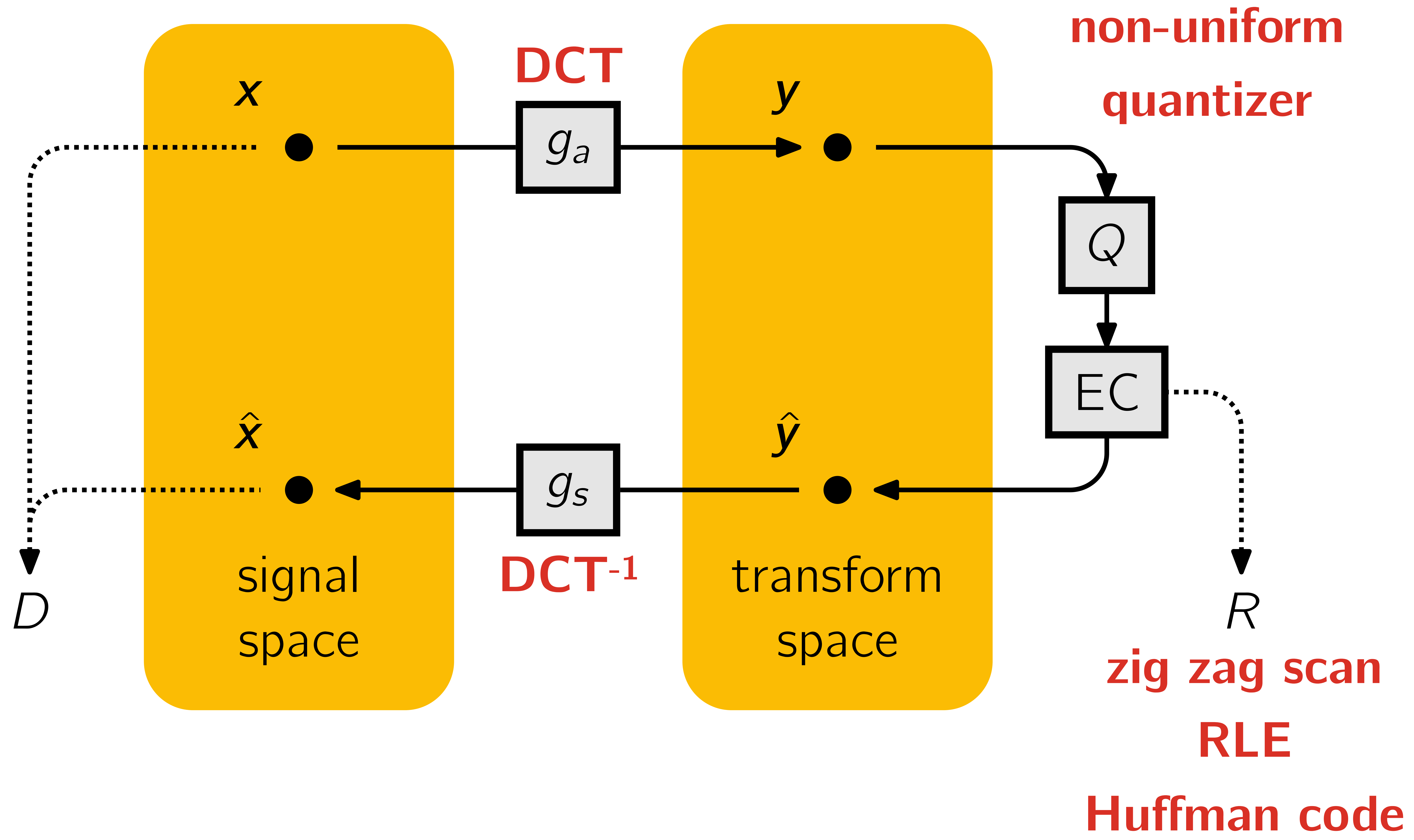
# JPEG, in a nutshell



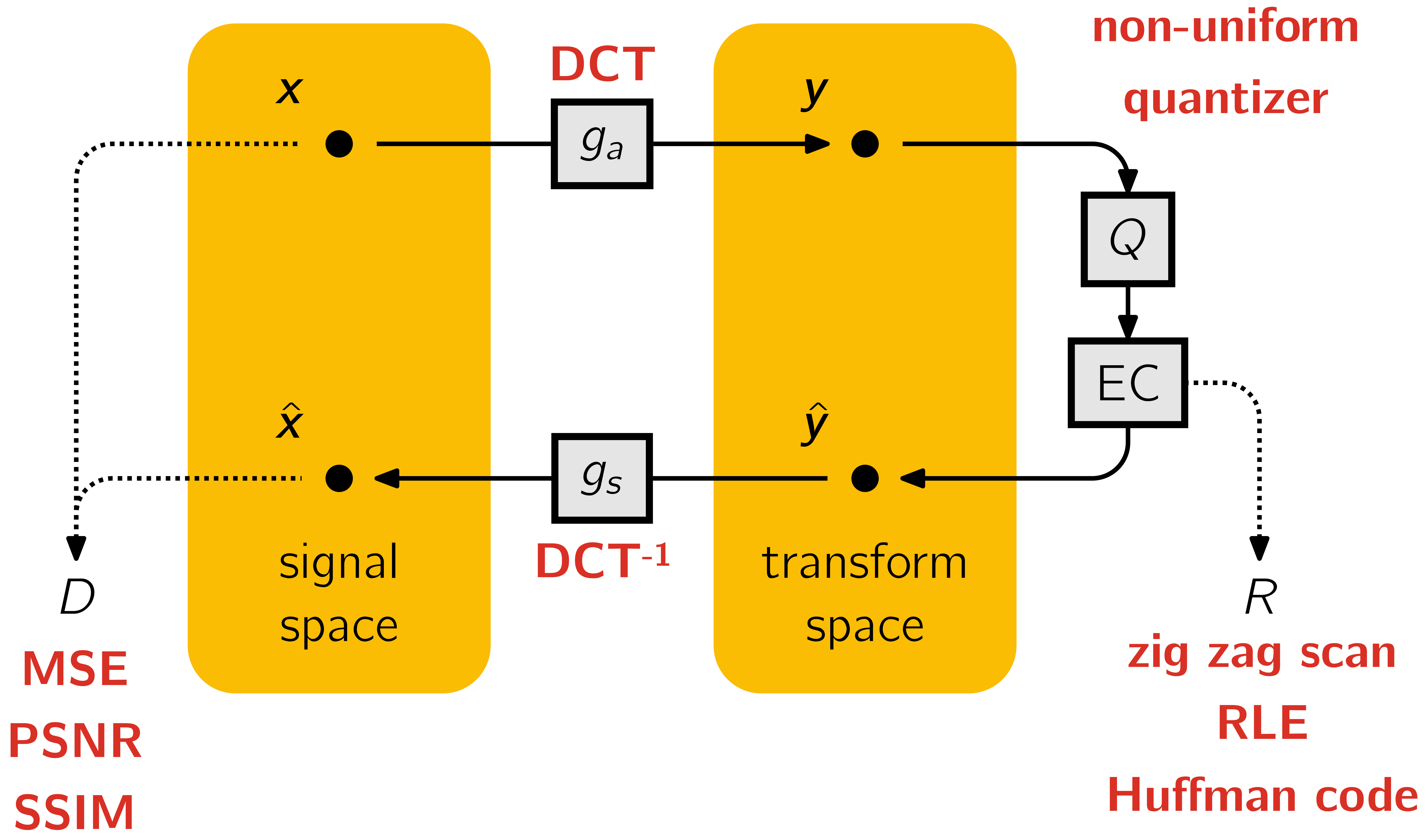
# JPEG, in a nutshell



# JPEG, in a nutshell



# JPEG, in a nutshell



# Why the DCT?

IEEE TRANSACTIONS ON COMPUTERS, JANUARY 1974

**Discrete Cosine Transform**

N. AHMED, T. NATARAJAN, AND K. R. RAO

*Abstract*—A discrete cosine transform (DCT) is defined and an algorithm to compute it using the fast Fourier transform is developed. It is shown that the discrete cosine transform can be used in the area of digital processing for the purposes of pattern recognition and Wiener filtering. Its performance is compared with that of a class of orthogonal transforms and is found to compare closely to that of the Karhunen-Loève transform, which is known to be optimal. The performances of the Karhunen-Loève and discrete cosine transforms are also found to compare closely with respect to the rate-distortion criterion.

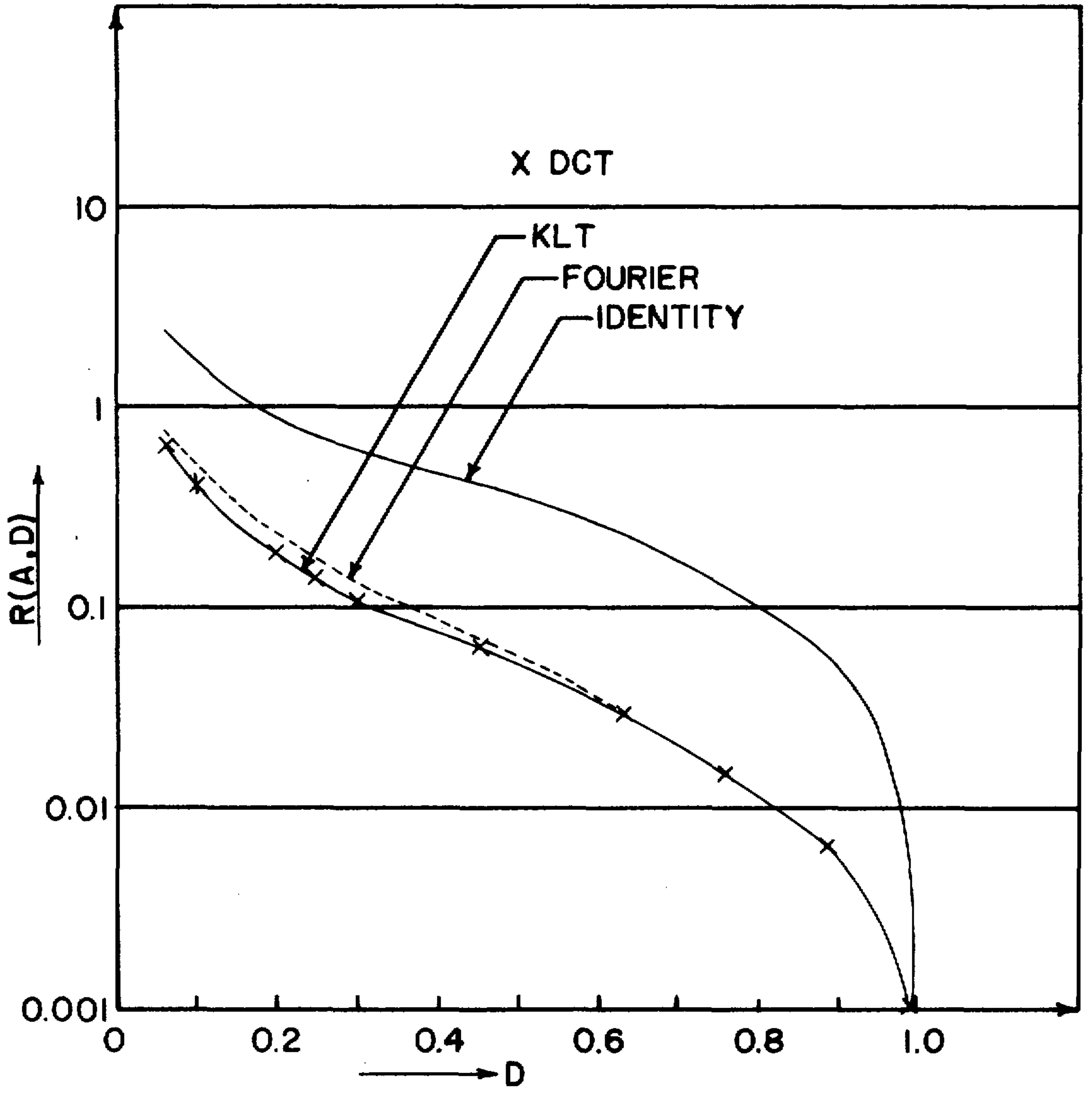


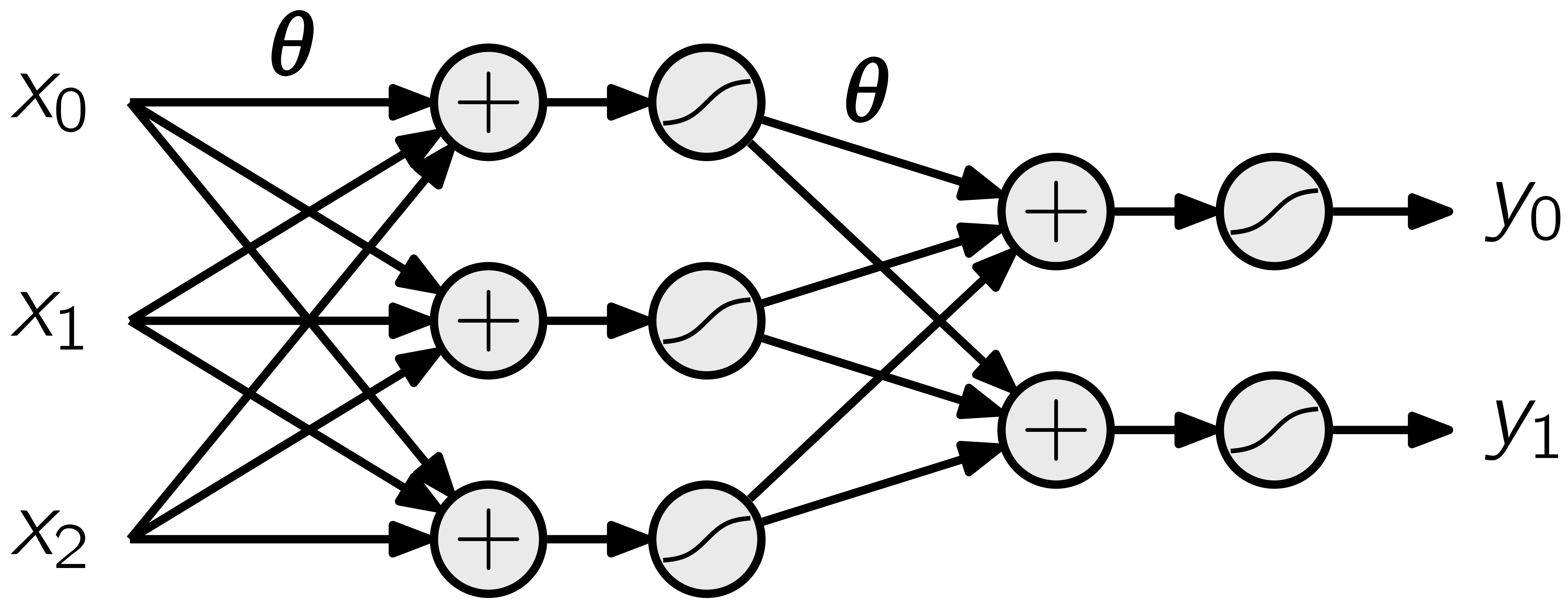

Fig. 5. Rate versus distortion for $M = 16$ and $\rho = 0.9$.

Assumptions:

- Gaussian (AR-1) signal
- Linear transform

⇒ KLT optimal, DCT very close and **fast**

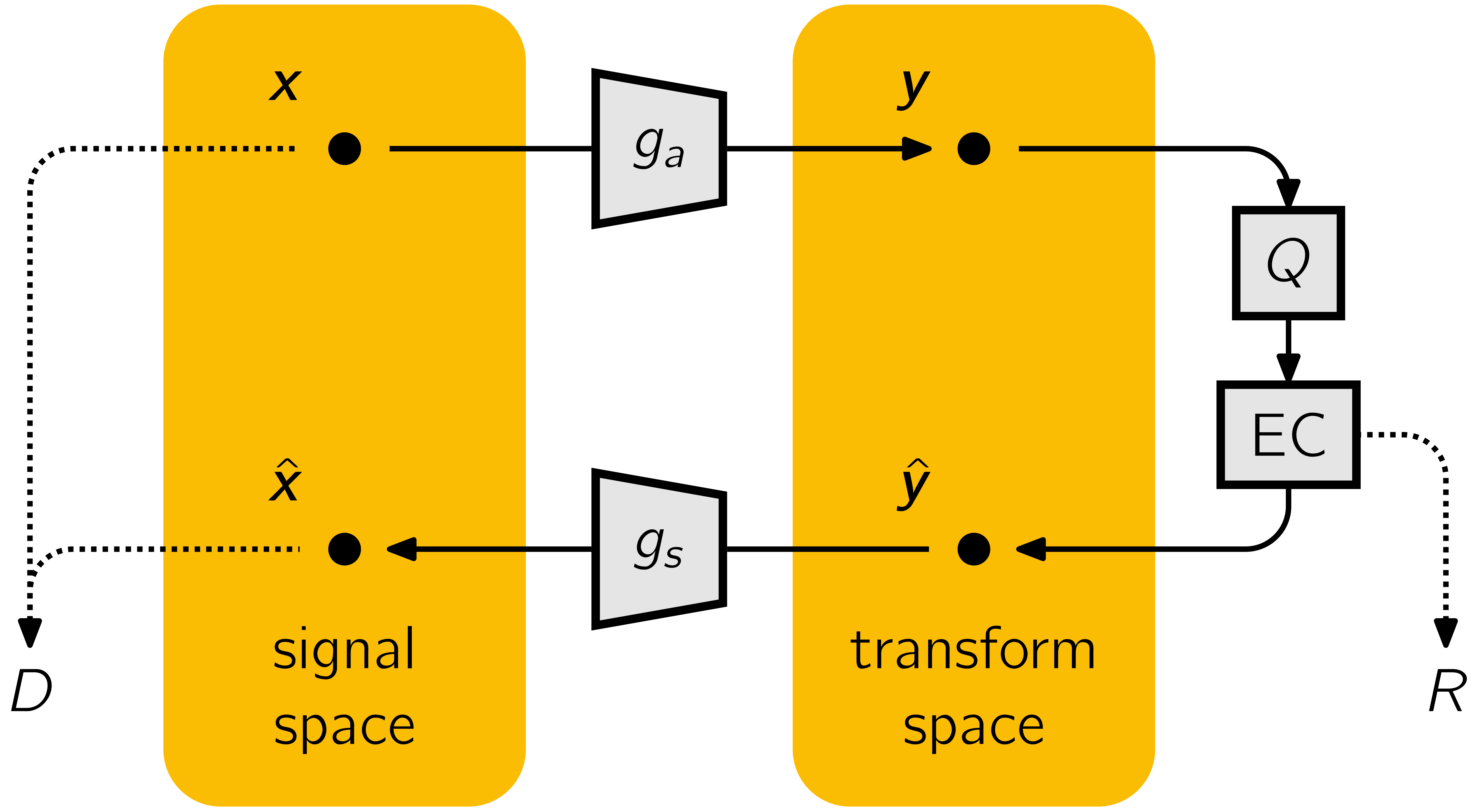
$$\boldsymbol{y} = g(\boldsymbol{x}; \boldsymbol{\theta})$$




Artificial neural networks (ANNs) are **universal function approximators**.

We can train them to approximate the RD-optimal transforms.

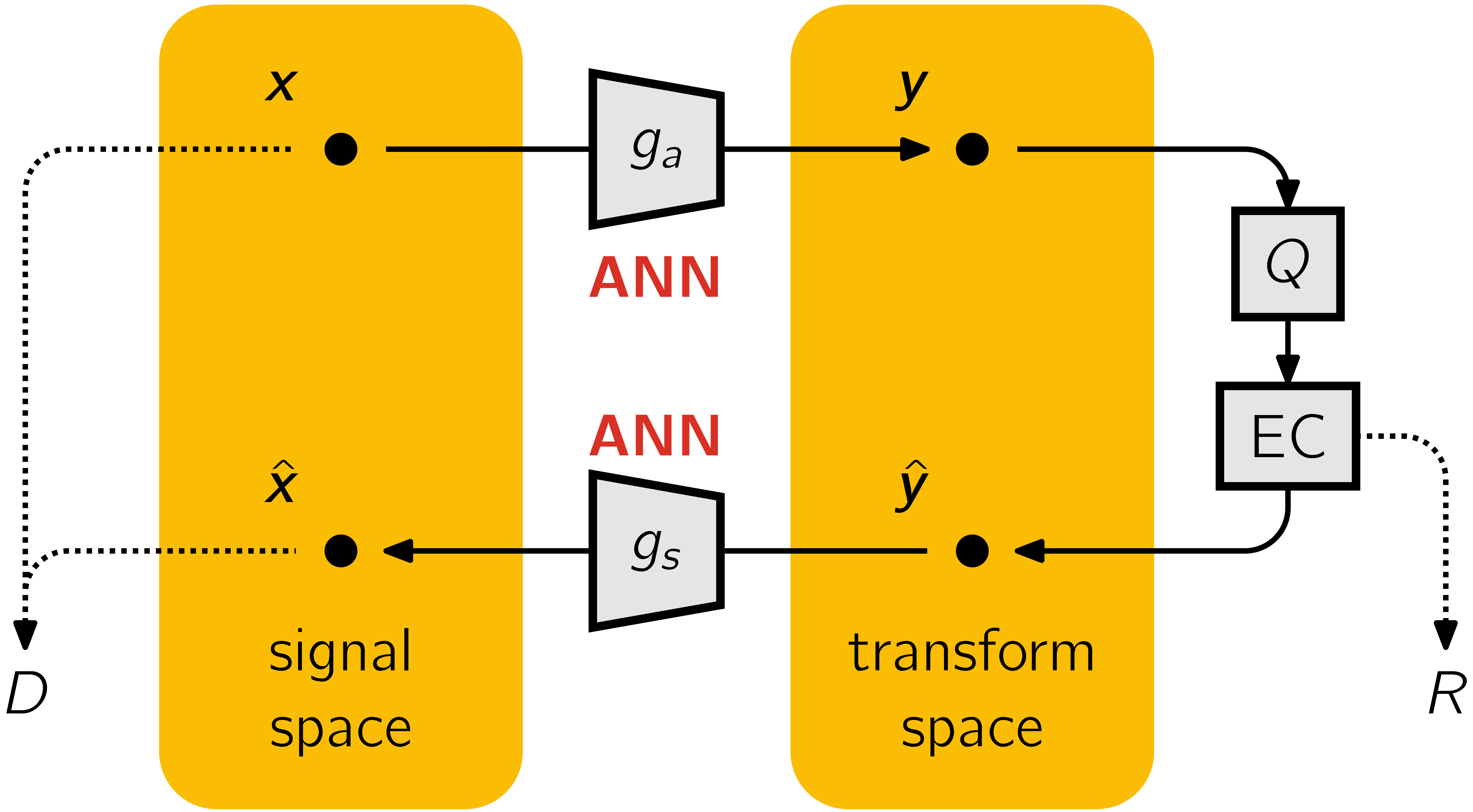
# Nonlinear transform coding




Ballé et al. (IEEE STSP, 2021)

Google Research

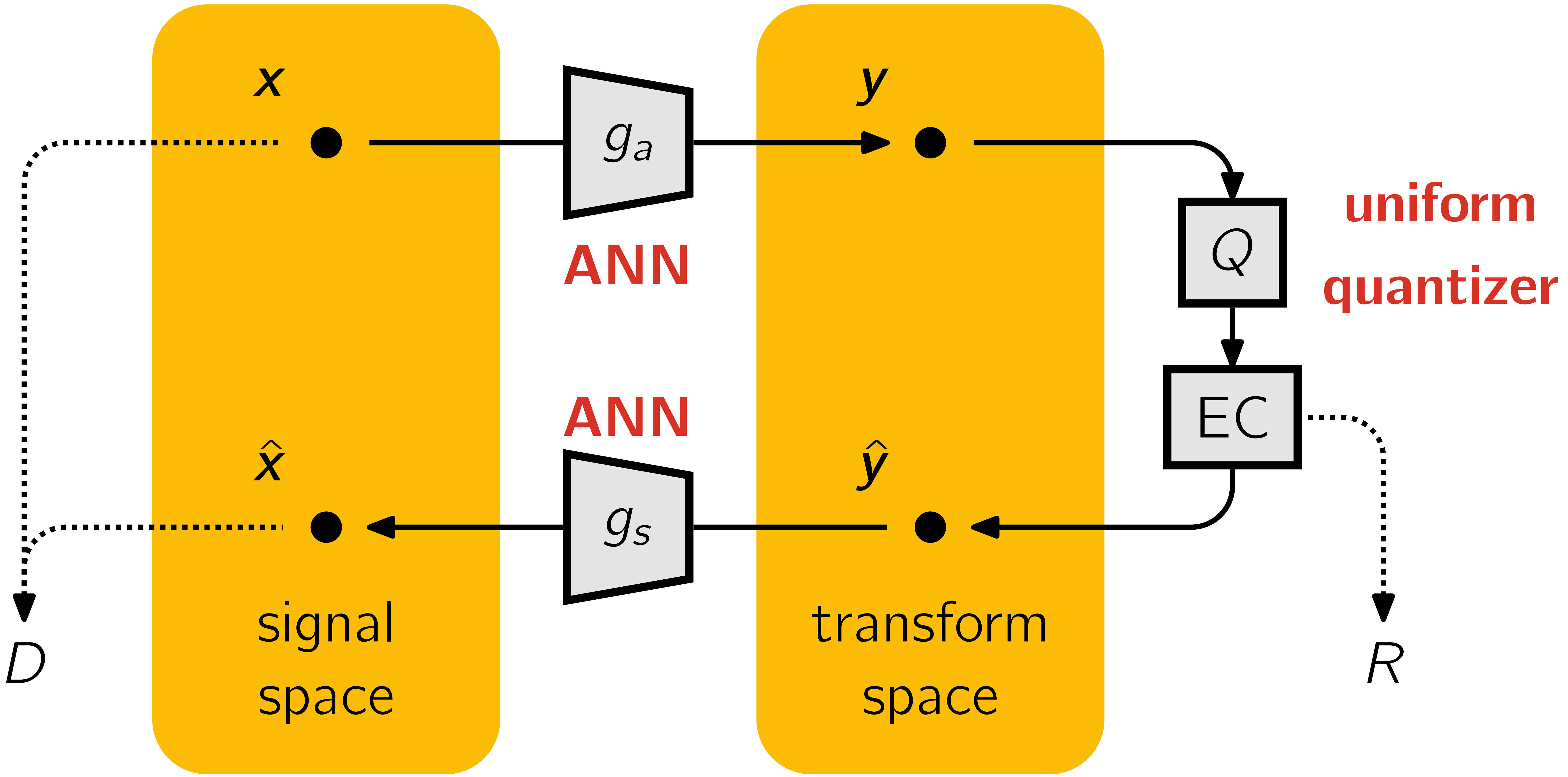
# Nonlinear transform coding




Ballé et al. (IEEE STSP, 2021)

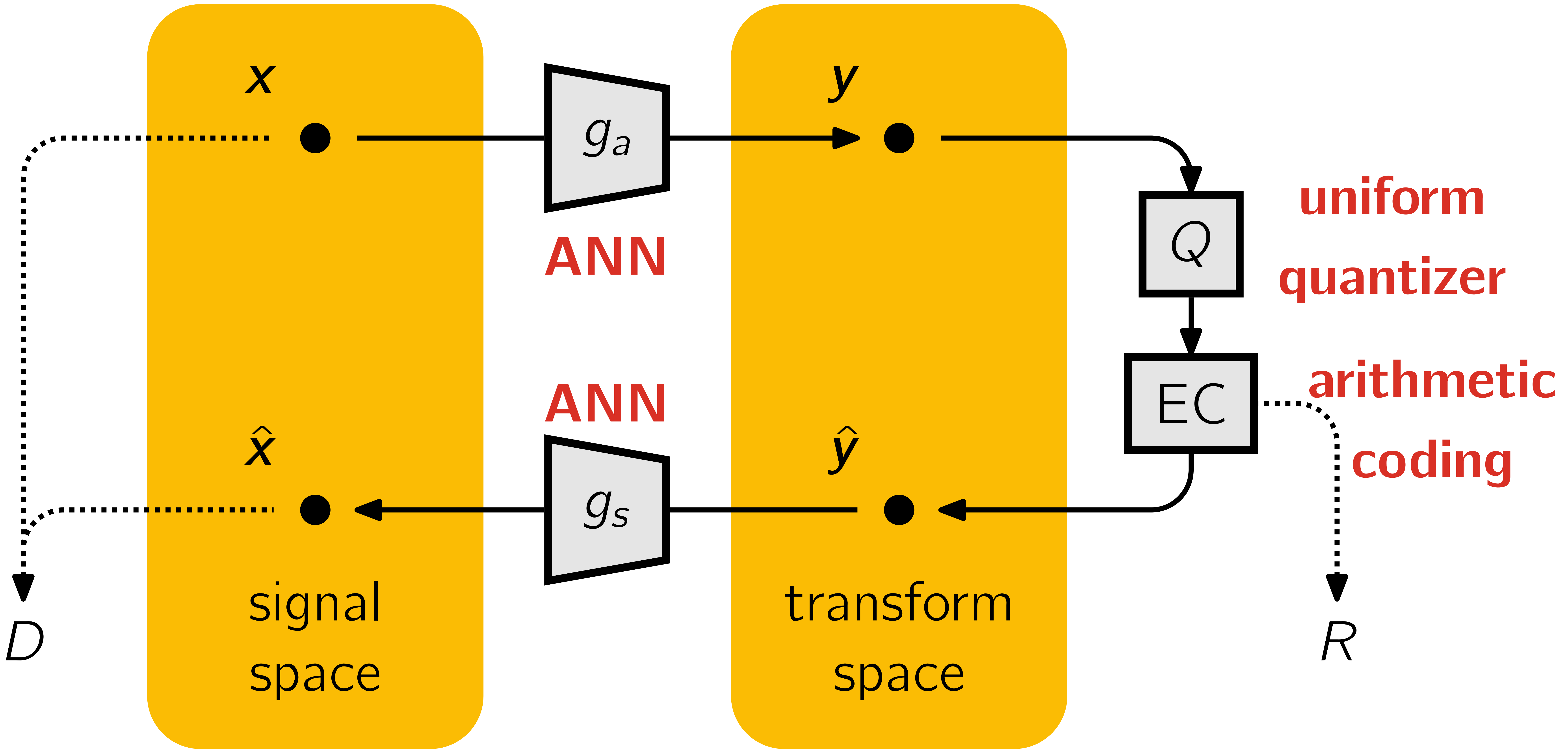
Google Research

# Nonlinear transform coding



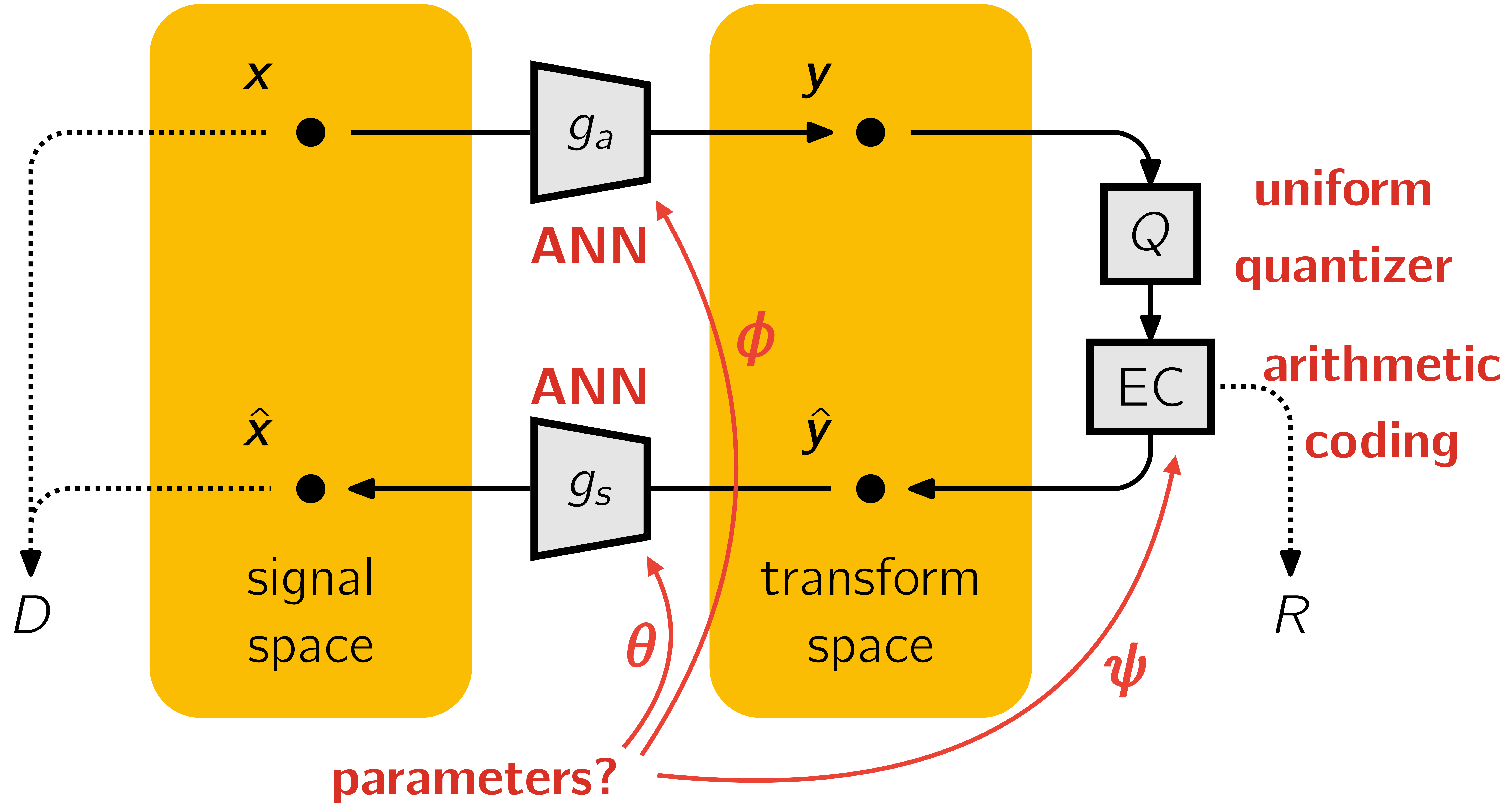

Ballé et al. (IEEE STSP, 2021)

# Nonlinear transform coding




Ballé et al. (IEEE STSP, 2021)

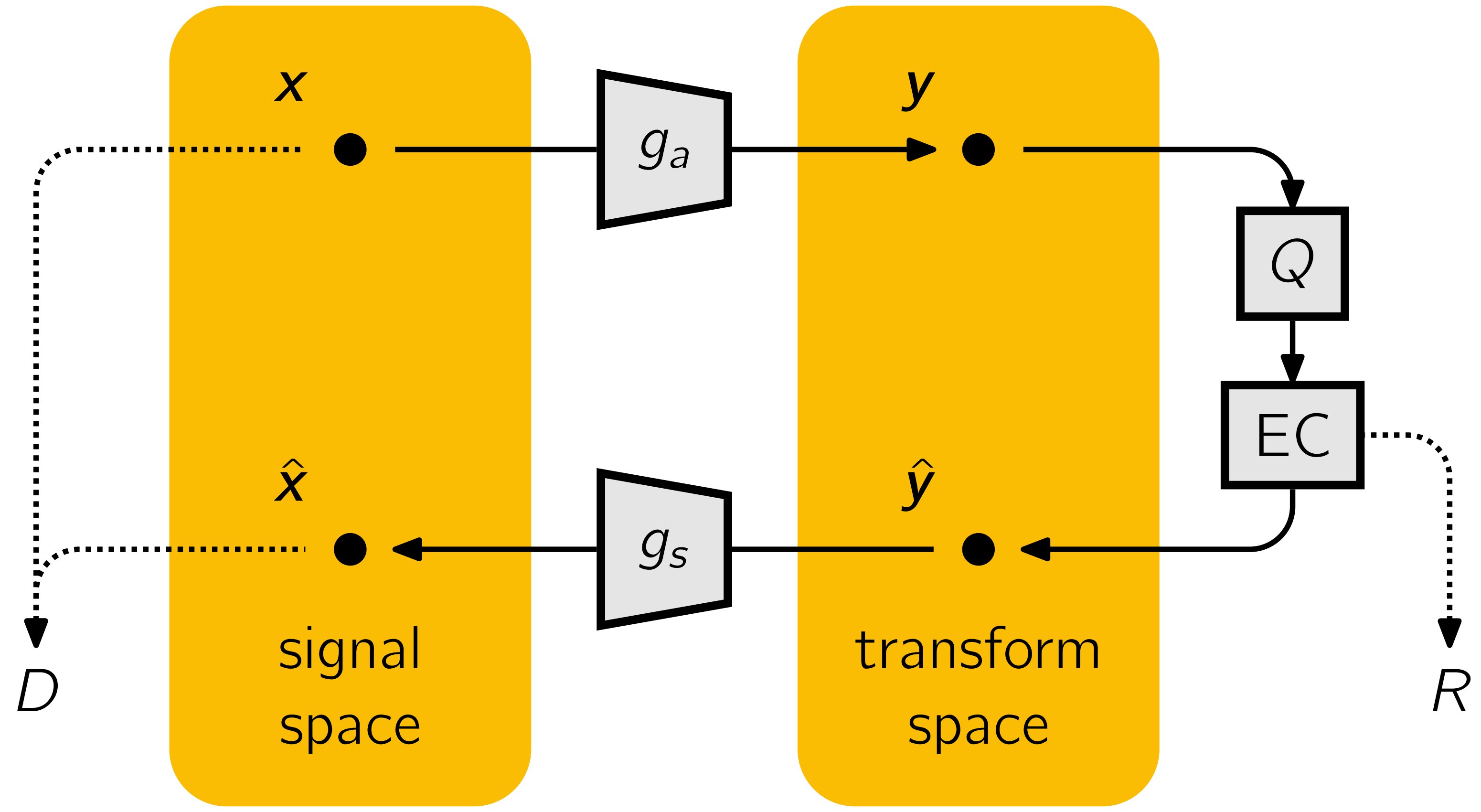
# Nonlinear transform coding




Ballé et al. (IEEE STSP, 2021)

Google Research

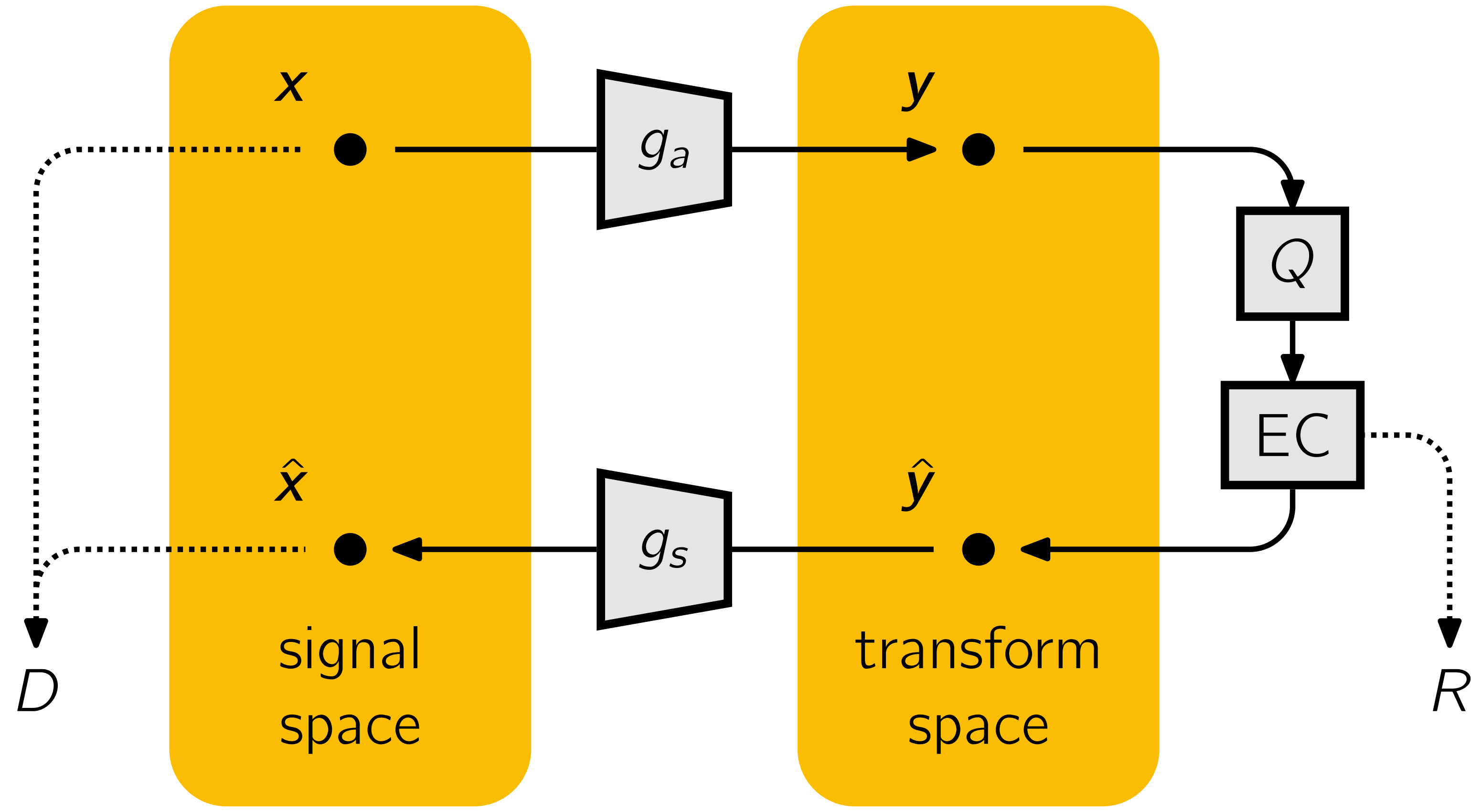
# End-to-end optimization



# End-to-end optimization

Loss function

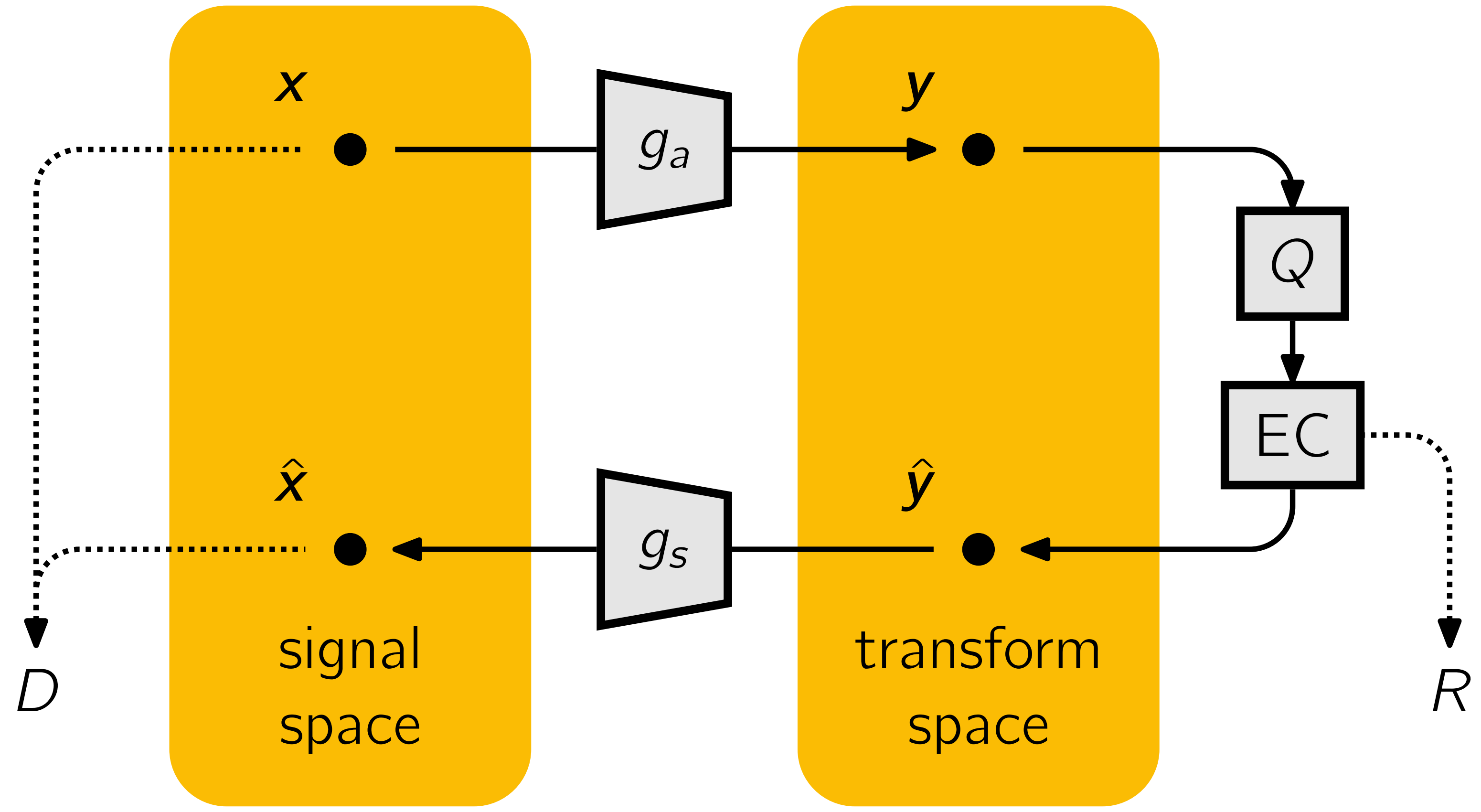
$$L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[-\log_2 p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}})\right]}_{R} + \lambda \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2\right]}_{D}$$



# End-to-end optimization

Loss function

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[-\log_2 p_{\hat{\boldsymbol{y}}}(Q(g_a(\boldsymbol{x}; \boldsymbol{\phi}))|\boldsymbol{\psi})\right]}_{R} + \lambda \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[\|\boldsymbol{x} - g_s(Q(g_a(\boldsymbol{x}; \boldsymbol{\phi})); \boldsymbol{\theta})\|_2^2\right]}_{D}$$



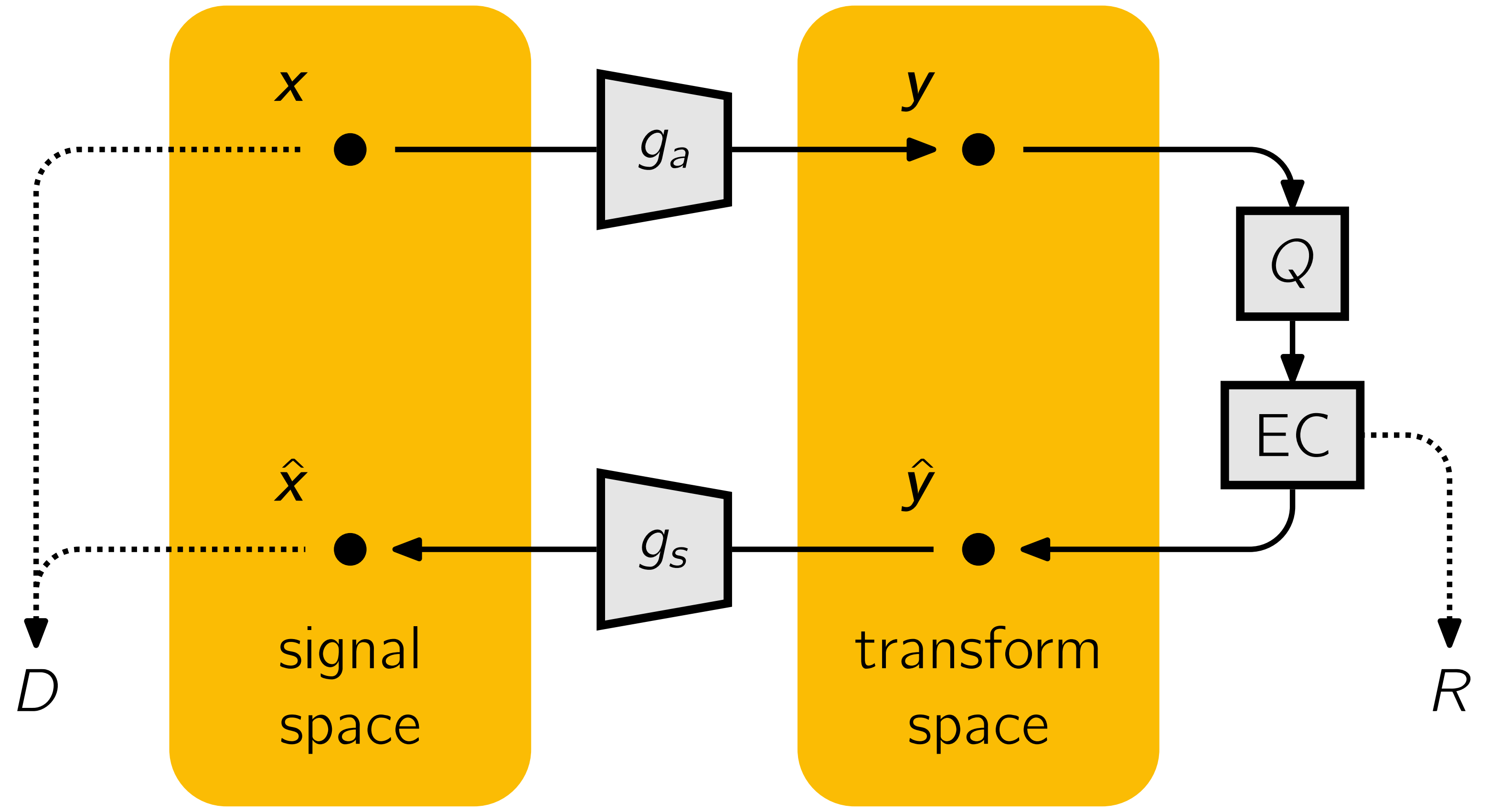
# End-to-end optimization

Loss function

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[-\log_2 p_{\hat{\boldsymbol{y}}}(Q(g_a(\boldsymbol{x}; \boldsymbol{\phi}))|\boldsymbol{\psi})\right]}_{R} + \lambda \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[\|\boldsymbol{x} - g_s(Q(g_a(\boldsymbol{x}; \boldsymbol{\phi})); \boldsymbol{\theta})\|_2^2\right]}_{D}$$

Stochastic gradient descent

$$\frac{\partial}{\partial \theta} \mathbb{E}_x\left[L(x; \theta)\right] \approx \frac{1}{|B|} \sum_{x \in B} \frac{\partial L(x; \theta)}{\partial \theta}$$



# End-to-end optimization

Loss function

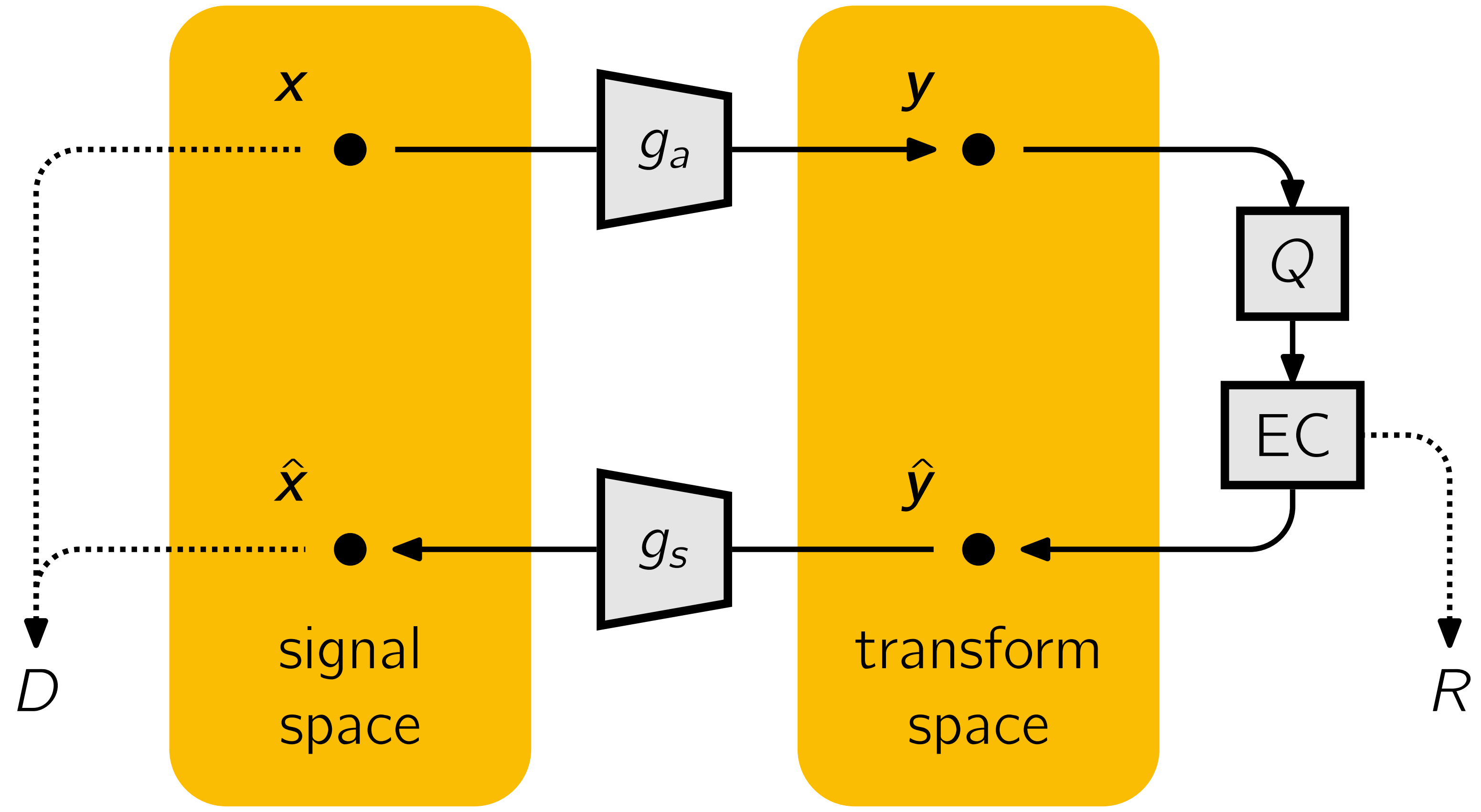
$$L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}) = \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[-\log_2 p_{\hat{\boldsymbol{y}}}(Q(g_a(\boldsymbol{x}; \boldsymbol{\phi}))|\boldsymbol{\psi})\right]}_{R} + \lambda \underbrace{\mathbb{E}_{\boldsymbol{x}}\left[\|\boldsymbol{x} - g_s(Q(g_a(\boldsymbol{x}; \boldsymbol{\phi})); \boldsymbol{\theta})\|_2^2\right]}_{D}$$

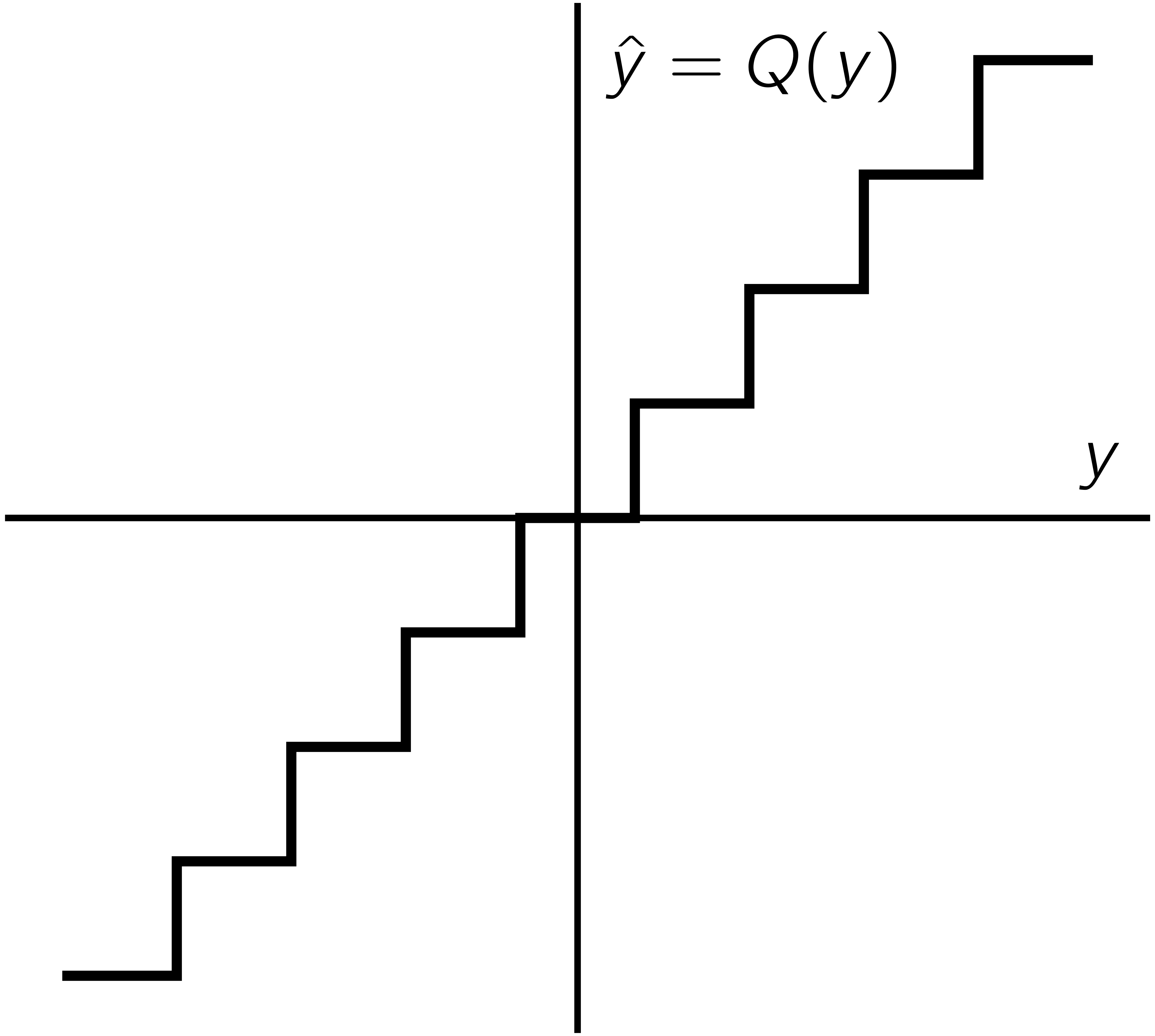
Stochastic gradient descent

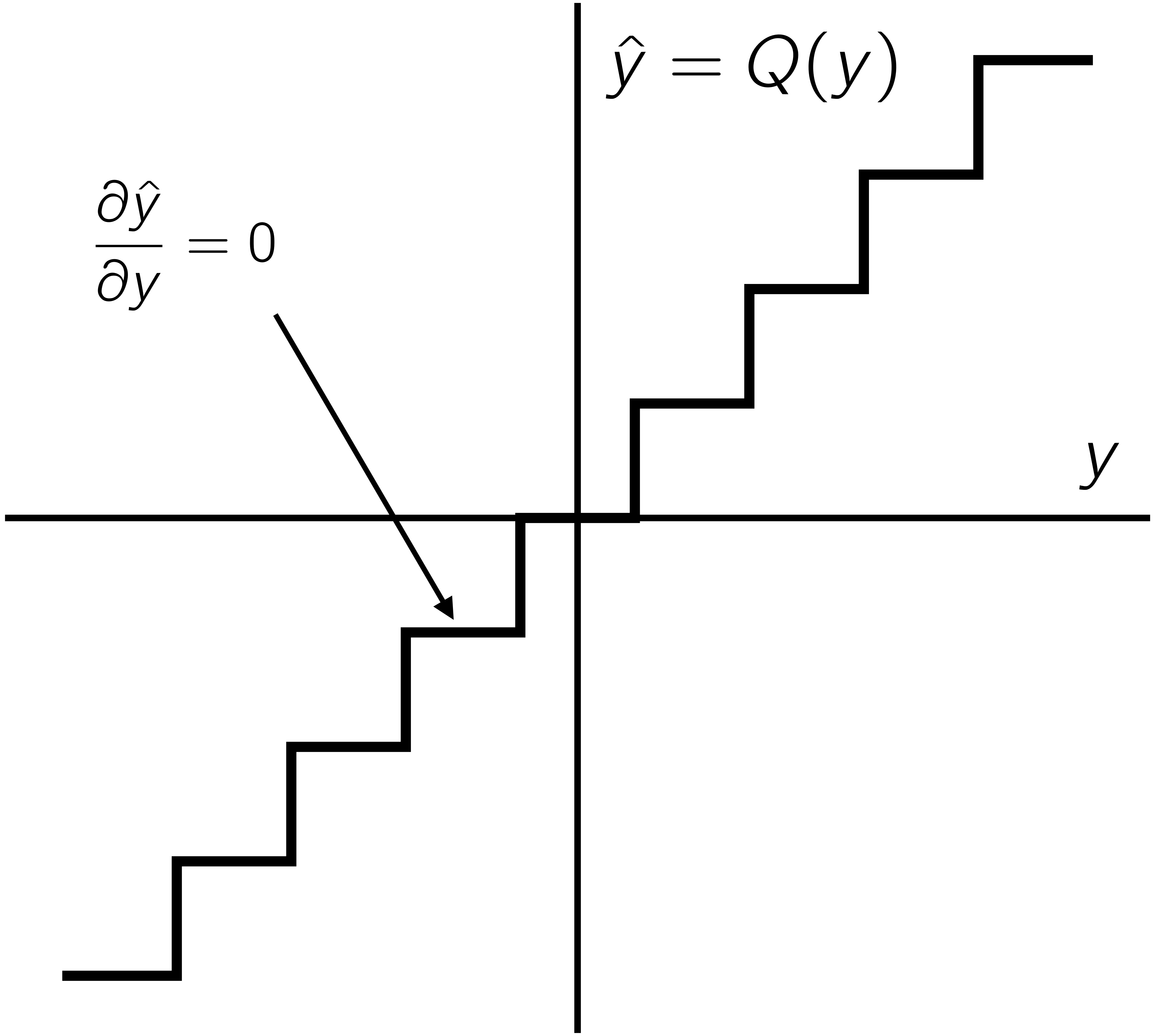
$$\frac{\partial}{\partial \theta} \mathbb{E}_x\left[L(x; \theta)\right] \approx \frac{1}{|B|} \sum_{x \in B} \frac{\partial L(x; \theta)}{\partial \theta}$$

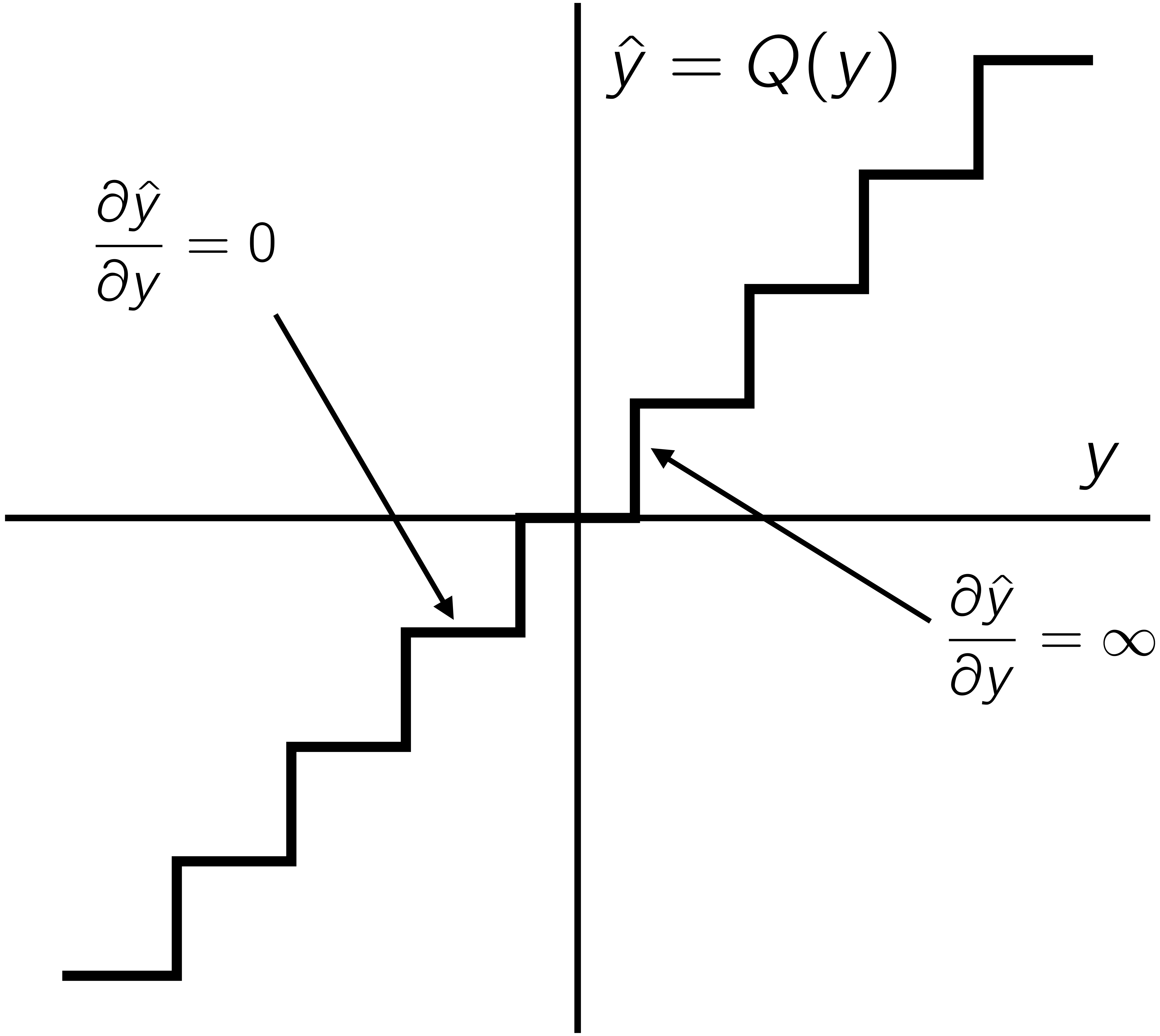
Symbolic differentiation (JAX, PyTorch, TensorFlow, etc.)

$$\frac{\partial L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi})}{\partial \boldsymbol{\theta}} = \dots \qquad \frac{\partial L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi})}{\partial \boldsymbol{\phi}} = \dots \qquad \frac{\partial L(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi})}{\partial \boldsymbol{\psi}} = \dots$$




$\hat{y} = Q(y)$
$y$


$\hat{y} = Q(y)$
$\frac{\partial \hat{y}}{\partial y} = 0$
$y$

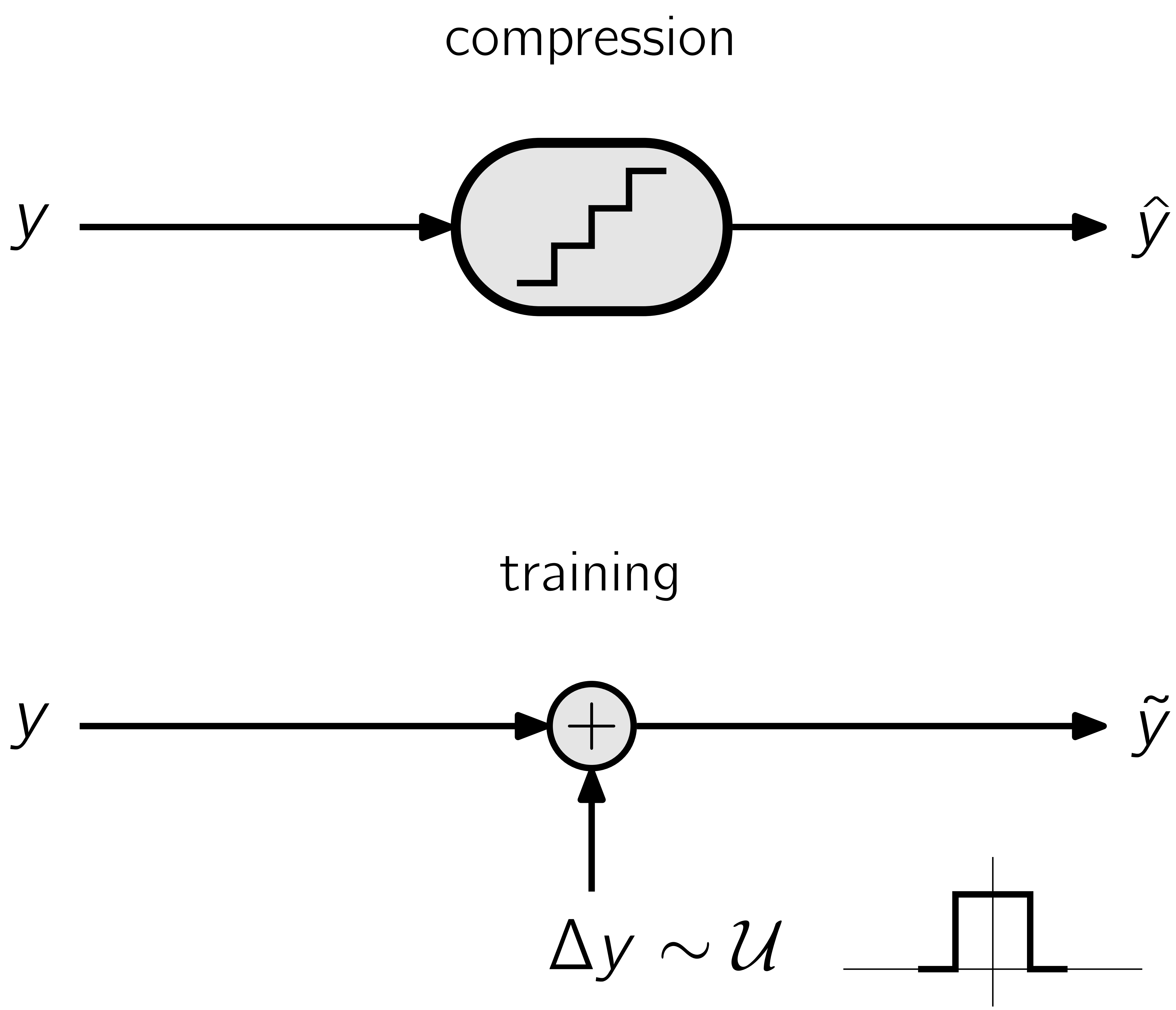

$\hat{y} = Q(y)$
$\frac{\partial \hat{y}}{\partial y} = 0$
$y$
$\frac{\partial \hat{y}}{\partial y} = \infty$

# Proxy R-D loss

Both rate and distortion loss contain discrete computations.

We need to replace them with differentiable losses, for example by plugging in dithered quantization.

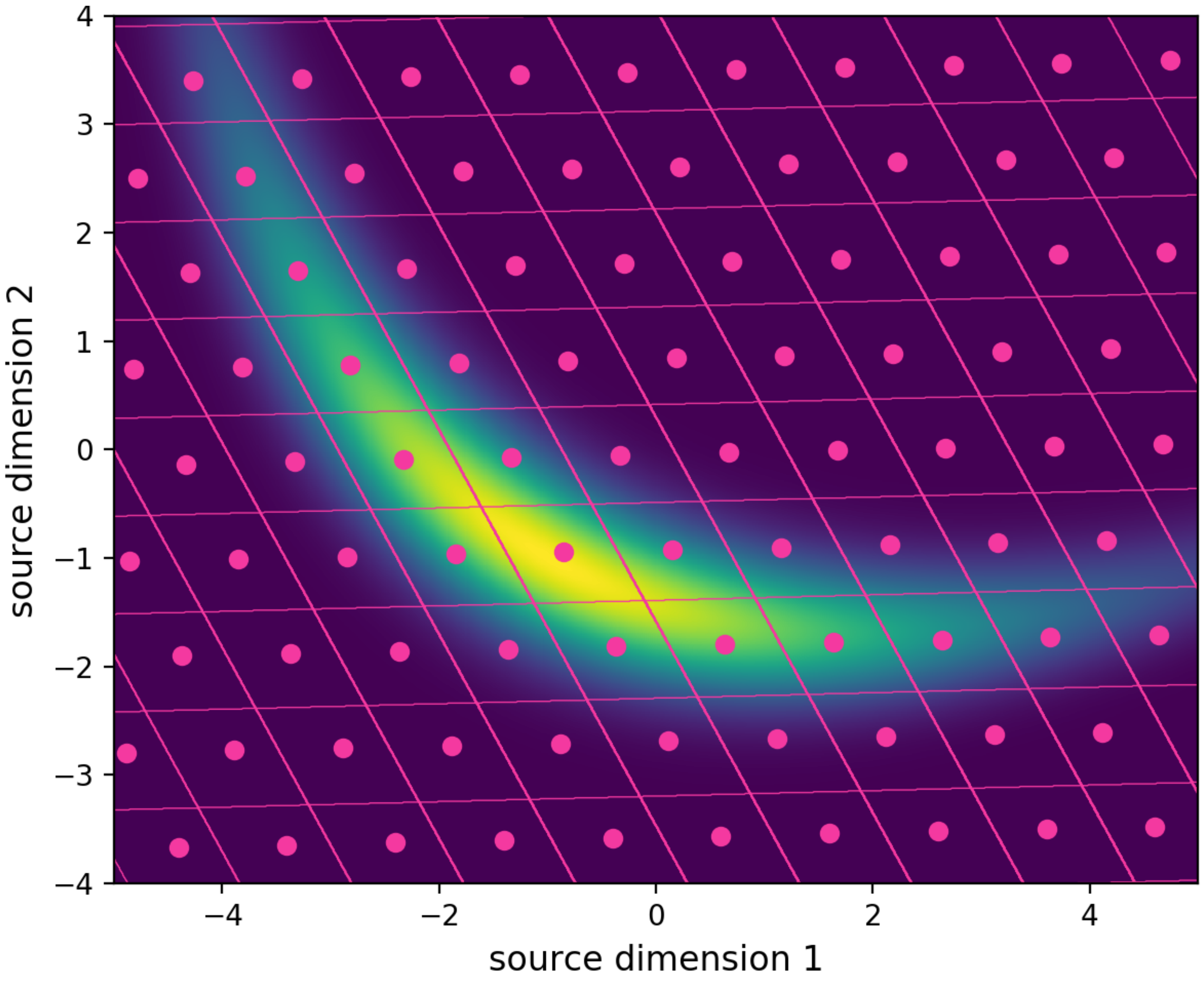
Better, we may interpolate between uniform and dithered quantization to control bias vs. variance of gradients (Agustsson & Theis, NeurIPS, 2020).



# Toy source

**linear transform coding**

$\boldsymbol{R+\lambda D = 6.87}$




Ballé et al. (IEEE STSP, 2021)

Google Research

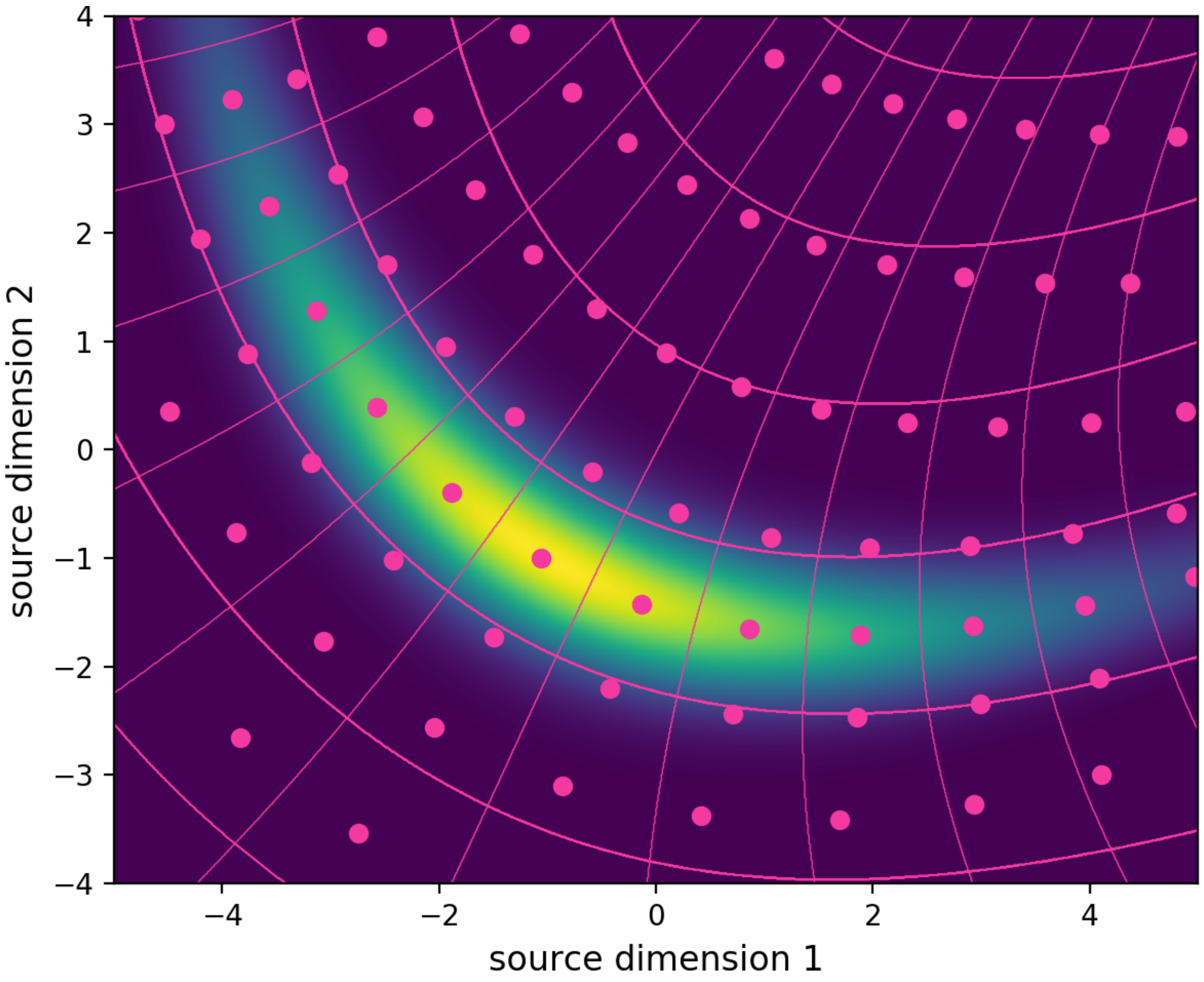
# Toy source

linear transform coding

$R + \lambda D = 6.87$

**nonlinear transform coding**

$R + \boldsymbol{\lambda} D = 5.97$




Ballé et al. (IEEE STSP, 2021)

# Toy source

linear transform coding

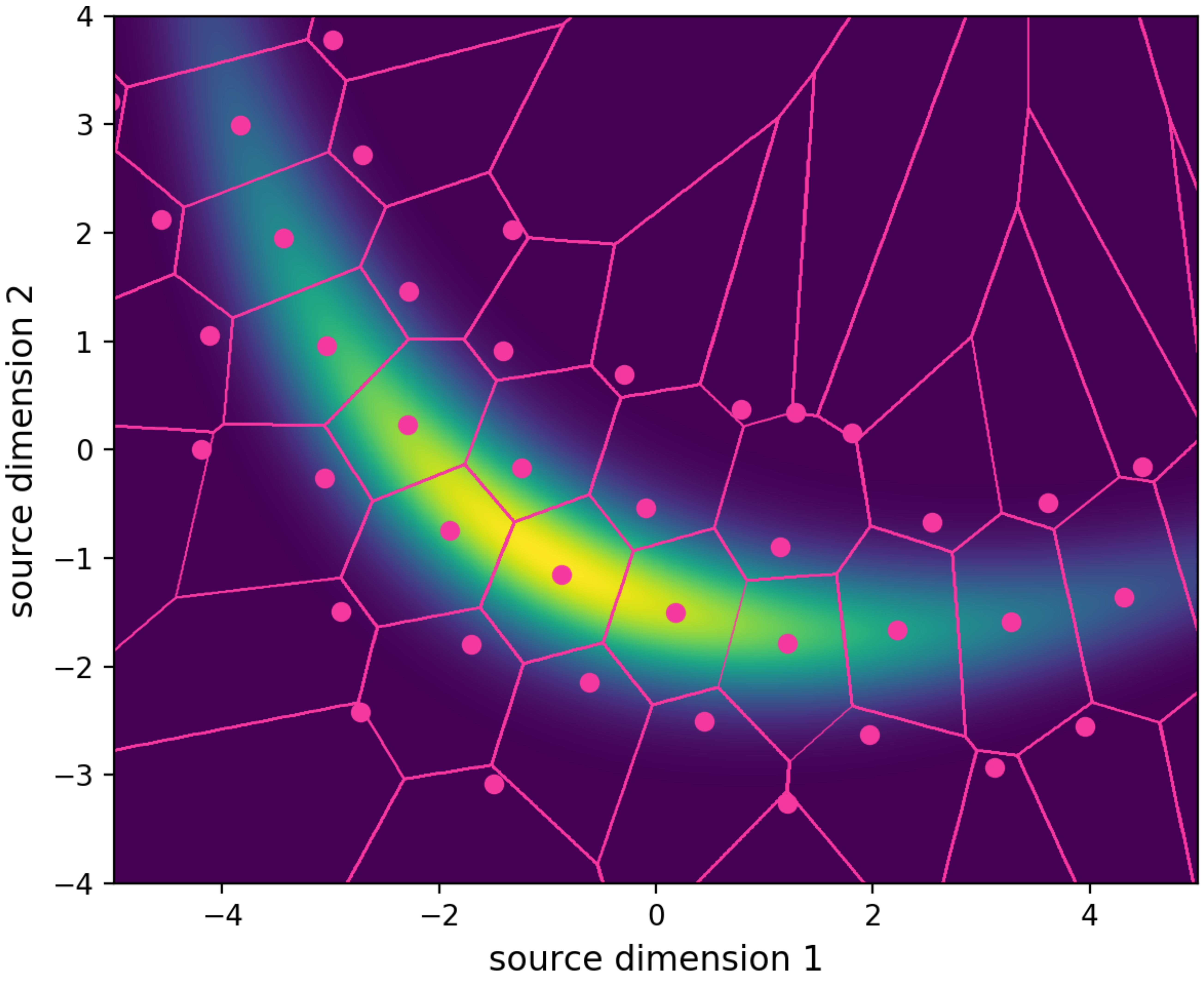
$R + \lambda D = 6.87$

nonlinear transform coding

$R + \lambda D = 5.97$

**rate-constrained vector quantization**

$\boldsymbol{R + \lambda D = 5.95}$




Ballé et al. (IEEE STSP, 2021)

# Progress in learned compression of natural images over the last few years

- ✅ One model for many RD-points
- ✅ Competitive in terms of PSNR
- ❓ Computational complexity
- ❓ Subjective image quality

# Progress in learned compression of natural images over the last few years

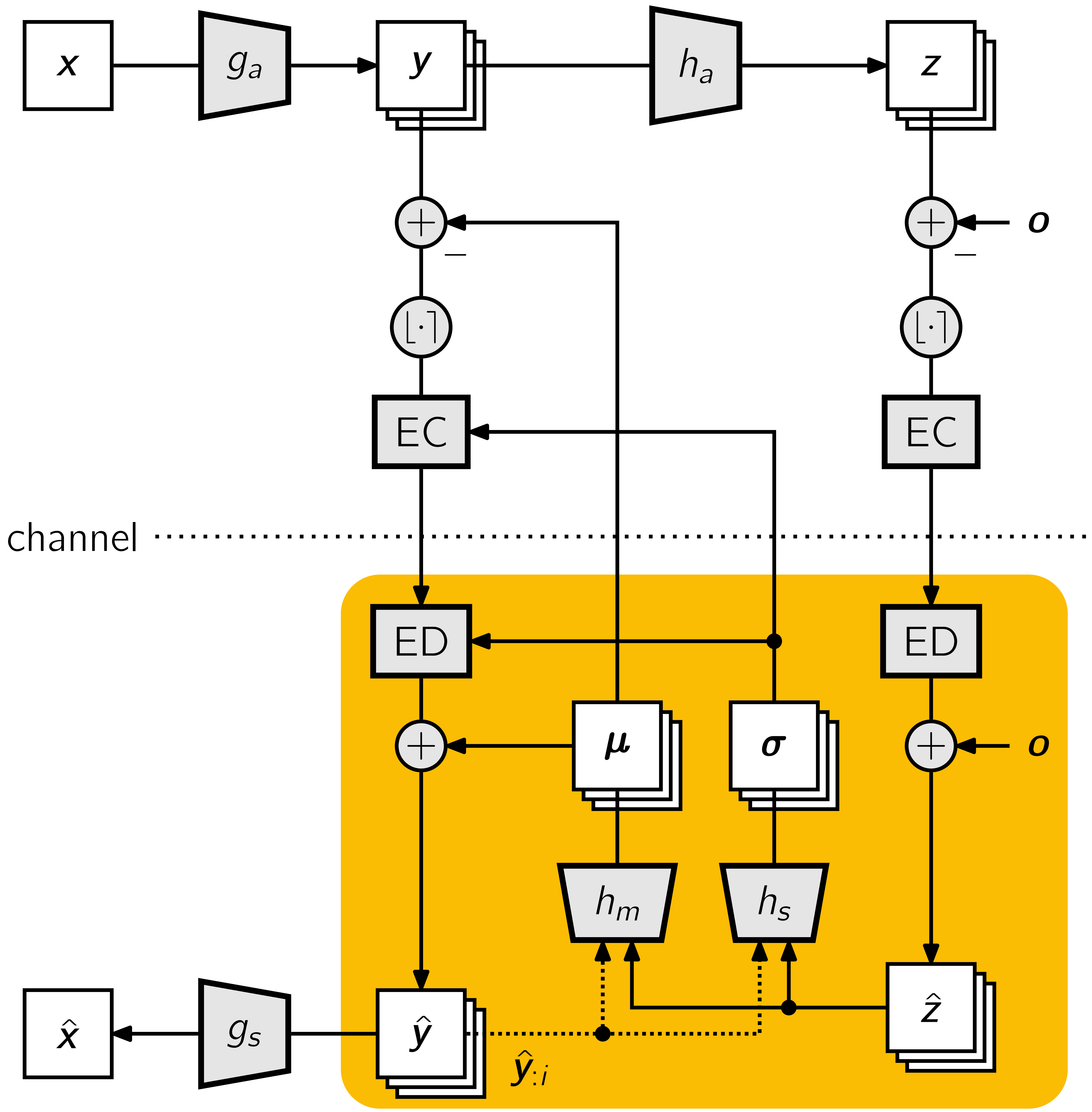
- ✅ One model for many RD-points
- ✅ Competitive in terms of PSNR
- ❓ Computational complexity
- ❓ Subjective image quality

# Hyperprior models

Many improvements stem from better entropy coding via a “hyperprior”.

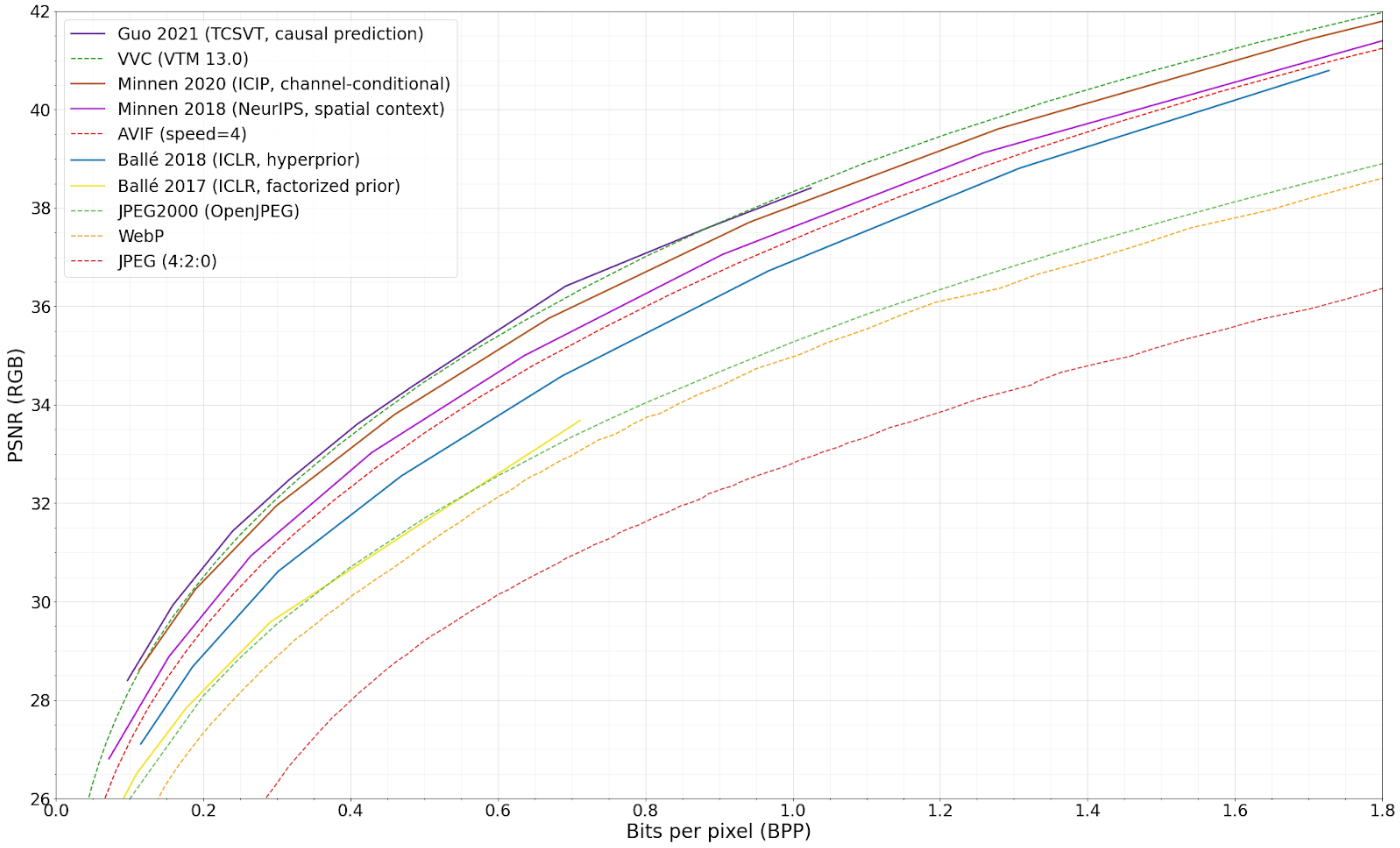
Elements across channel dimension of the latent tensor ***y*** aren’t considered independent.

Their distribution is predicted either forward- or backward-adaptively, by a set of other neural networks ($h$).



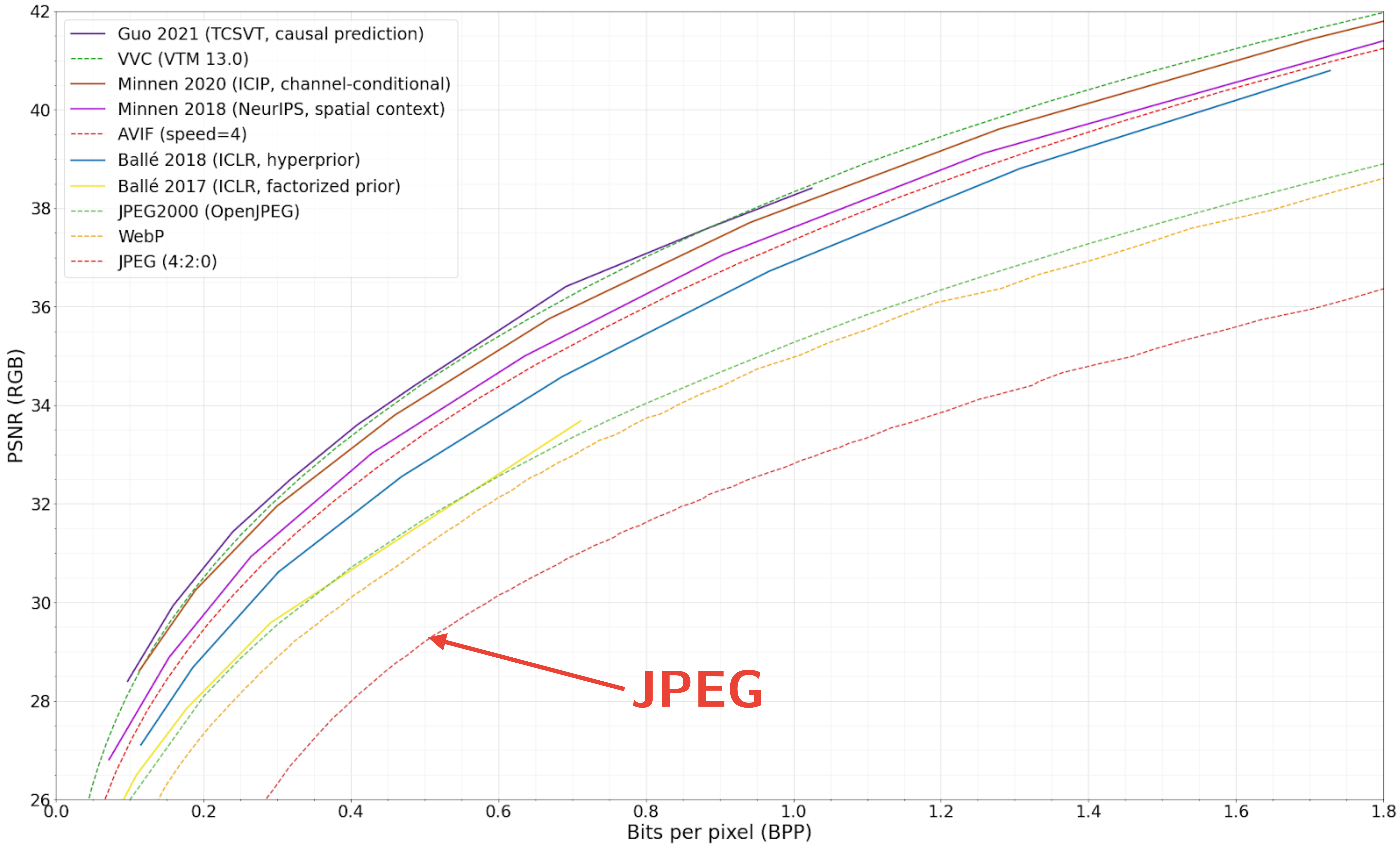

Ballé et al. (IEEE STSP, 2021)

# “Catching up” in terms of PSNR

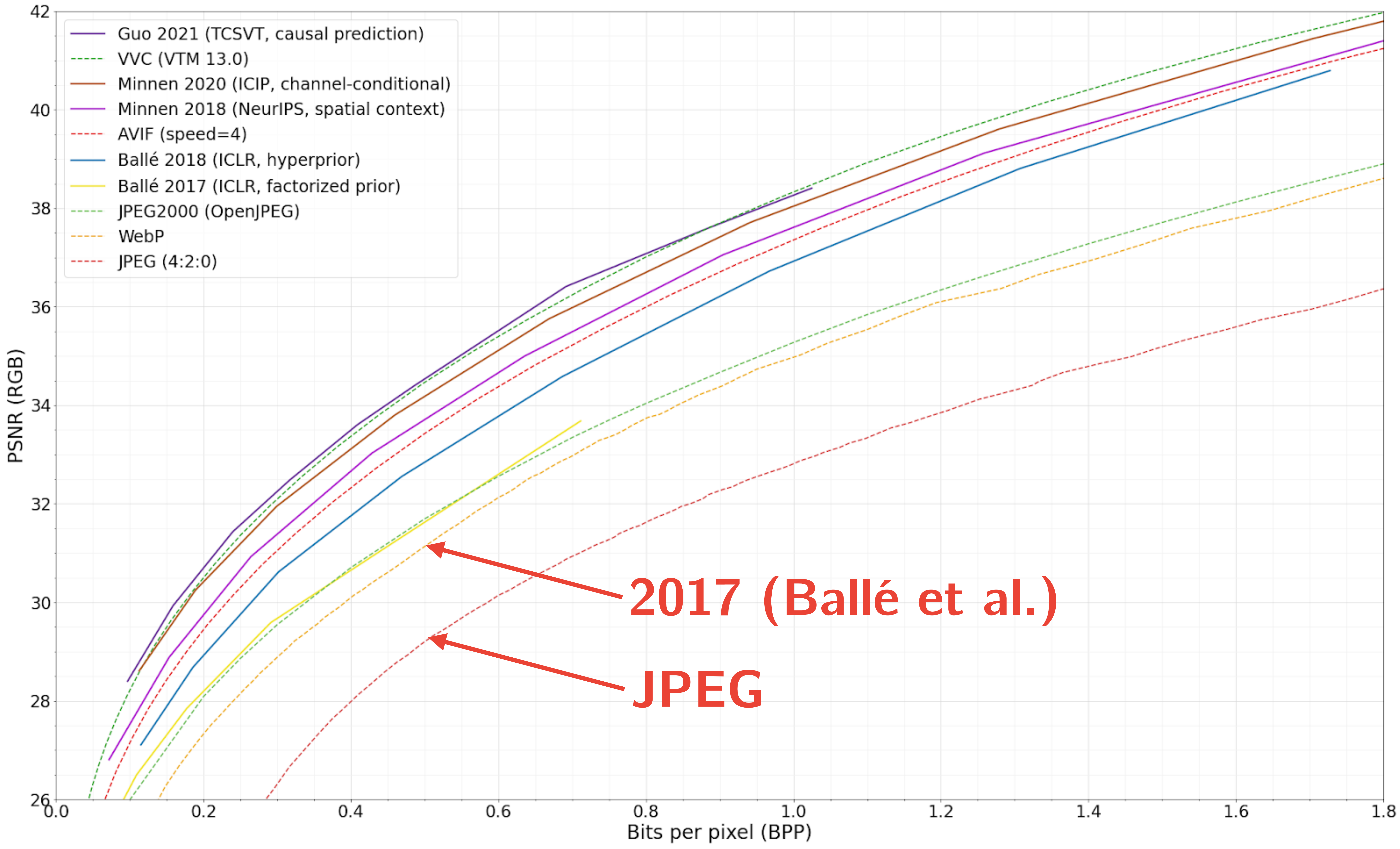



Google Research

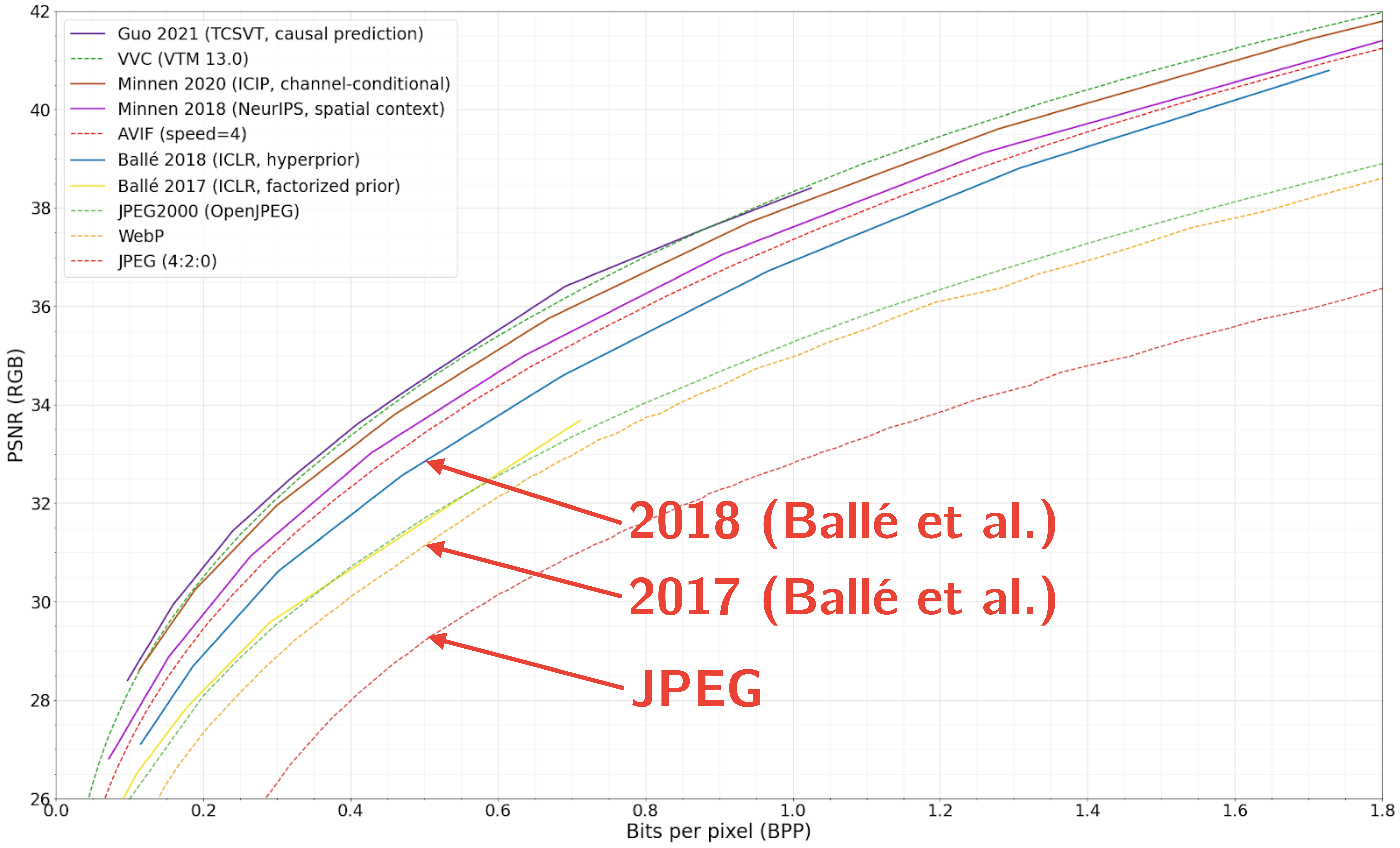
# “Catching up” in terms of PSNR



# “Catching up” in terms of PSNR

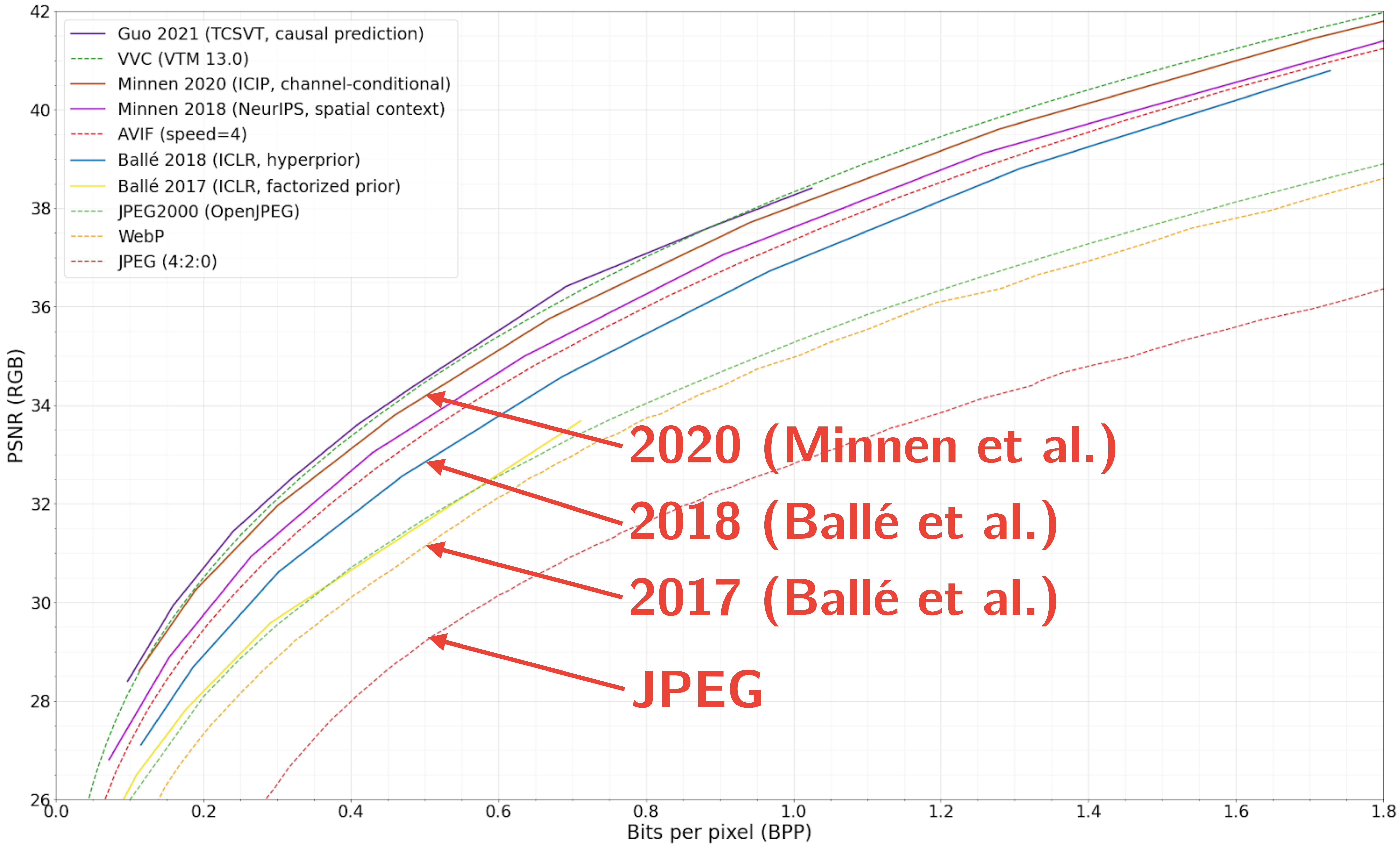


# “Catching up” in terms of PSNR

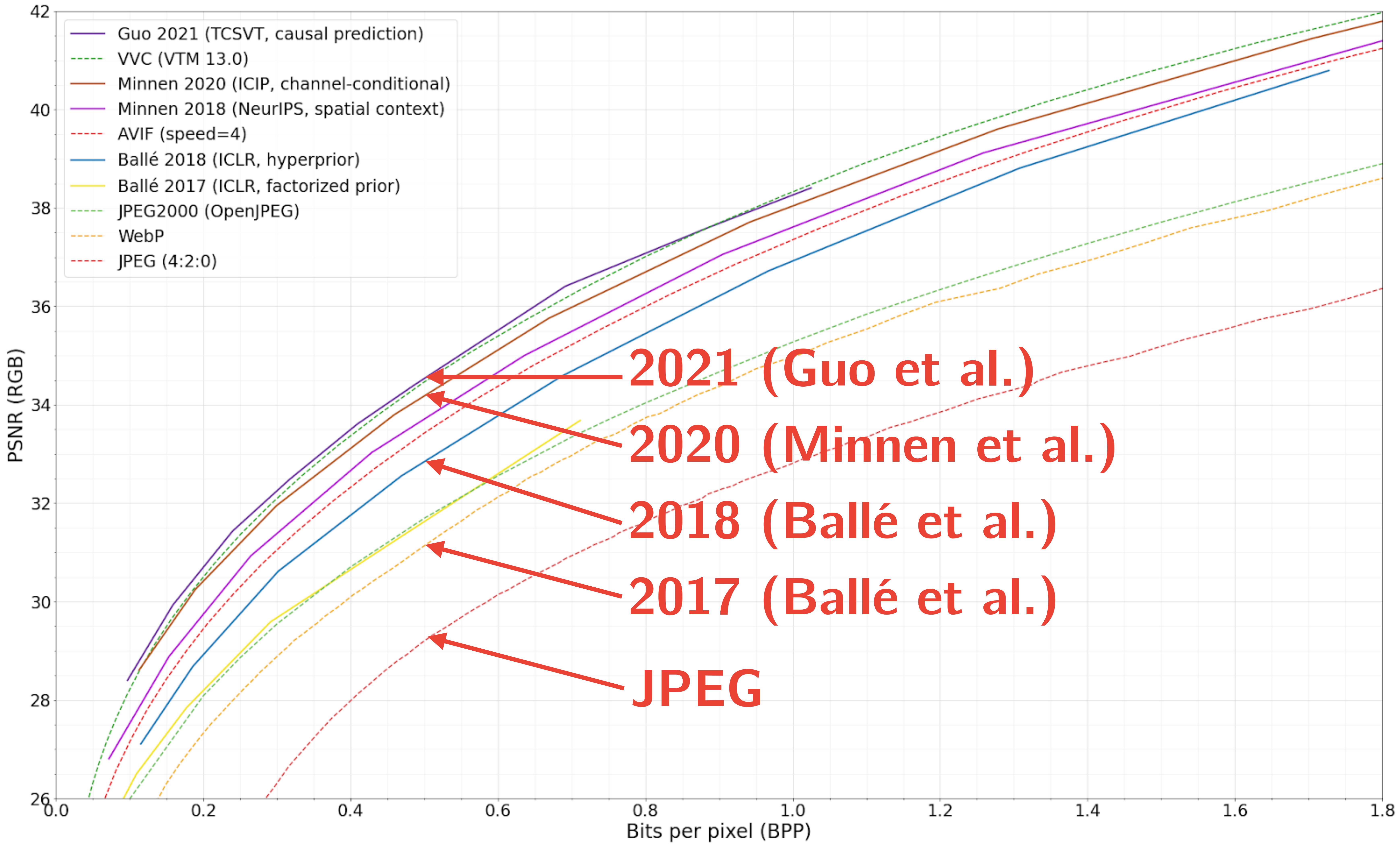



Google Research

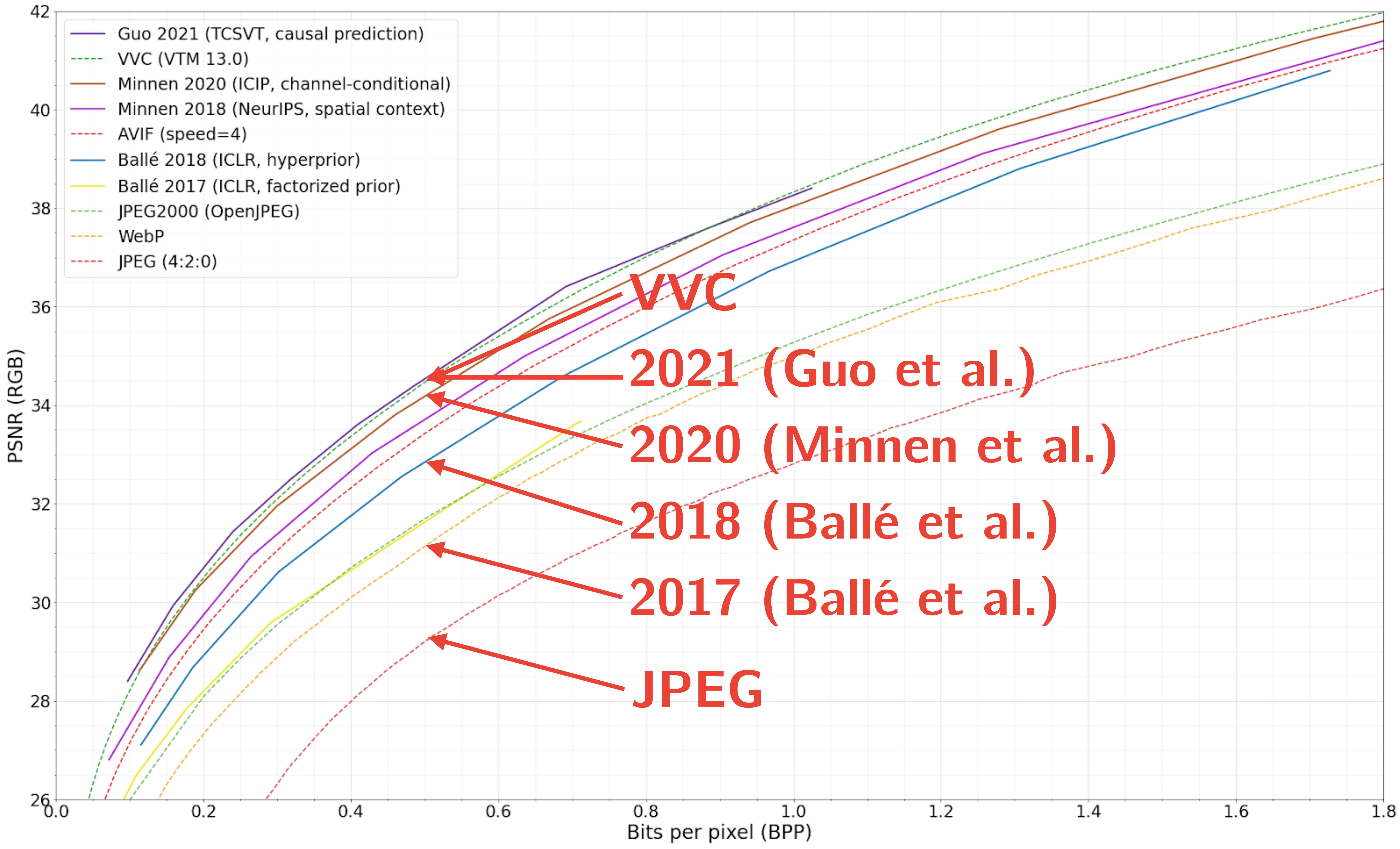
# “Catching up” in terms of PSNR



# “Catching up” in terms of PSNR



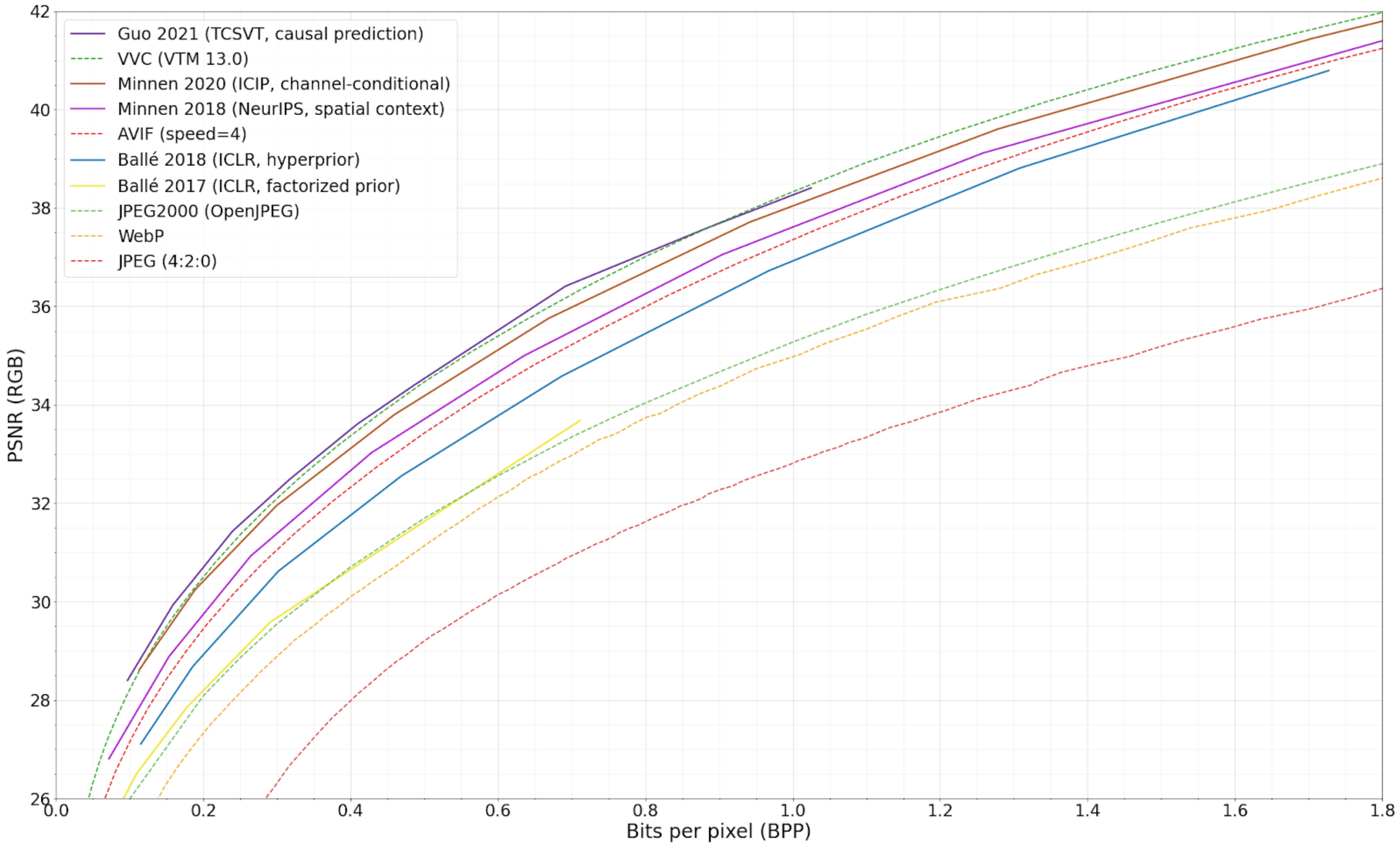
# “Catching up” in terms of PSNR



# Progress in learned compression of natural images over the last few years

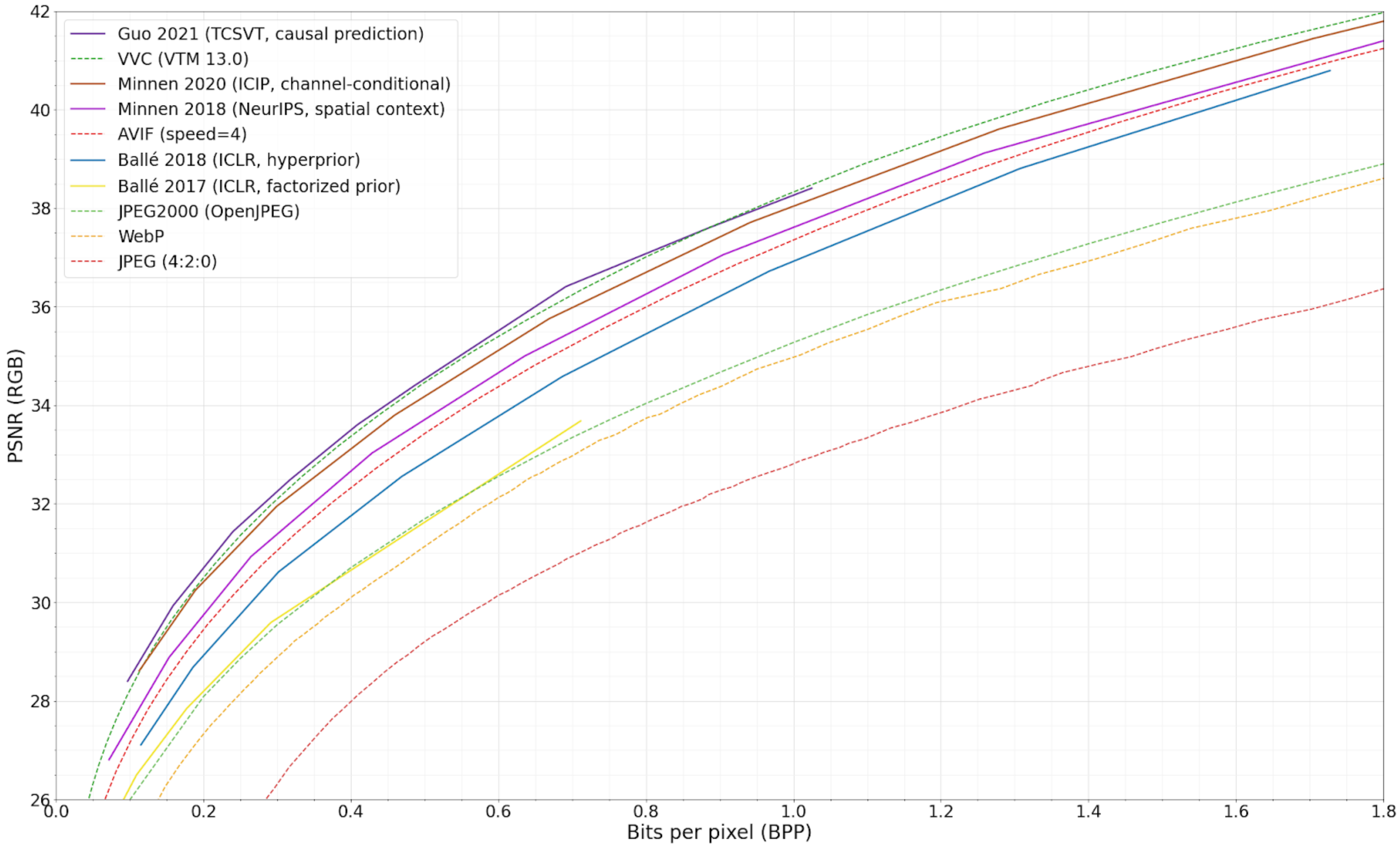
- ✅ One model for many RD-points
- ✅ Competitive in terms of PSNR
- ❓ Computational complexity
- ❓ Subjective image quality

# “Catching up” in terms of PSNR




Google Research

# “Catching up” in terms of PSNR




**Hybrid coding:**

typically

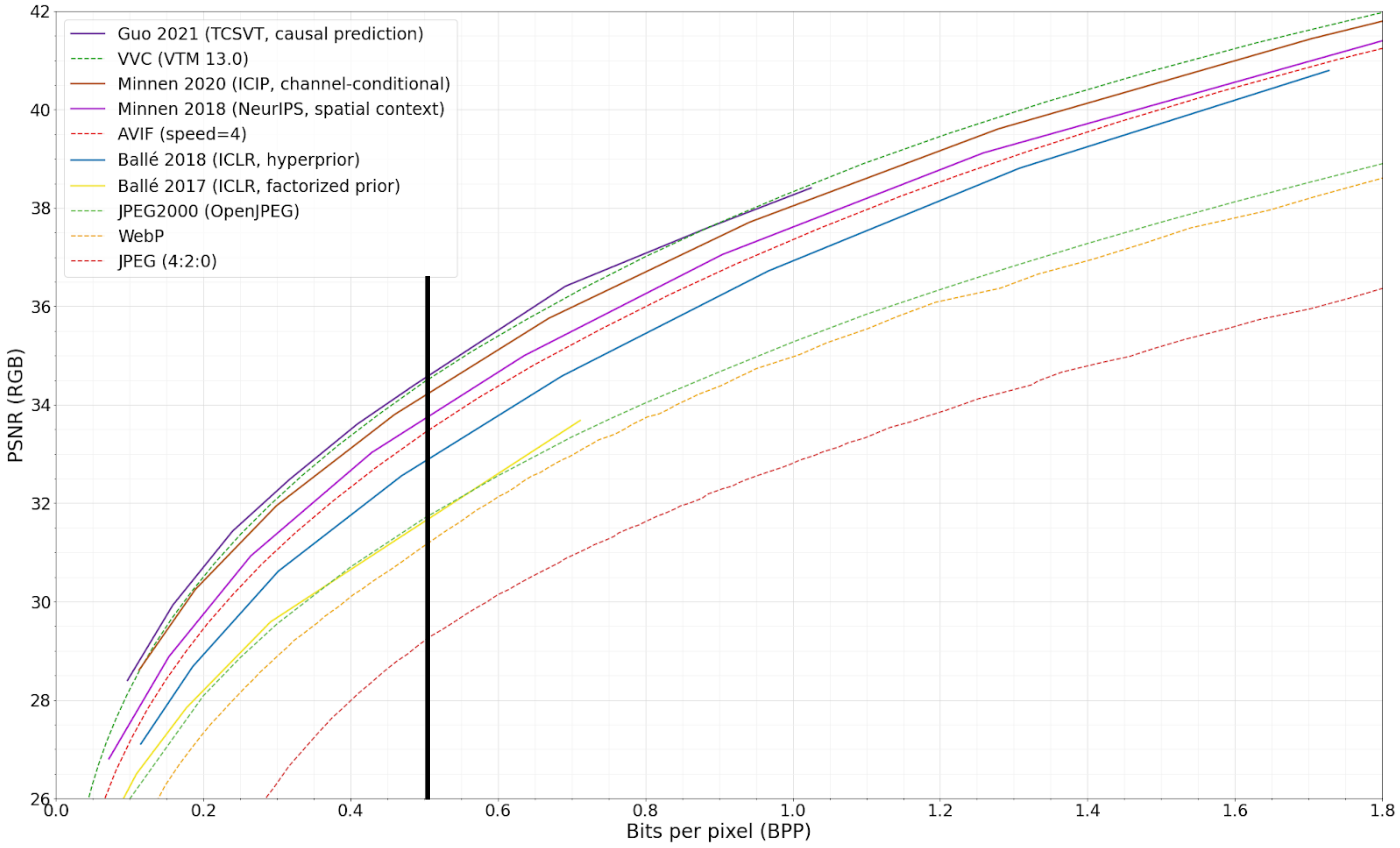
$O(\text{enc}) > O(\text{dec})$

**Learned compression:**

typically

$O(\text{enc}) \approx O(\text{dec})$

# “Catching up” in terms of PSNR




**Hybrid coding:**

typically

$O(\text{enc}) > O(\text{dec})$

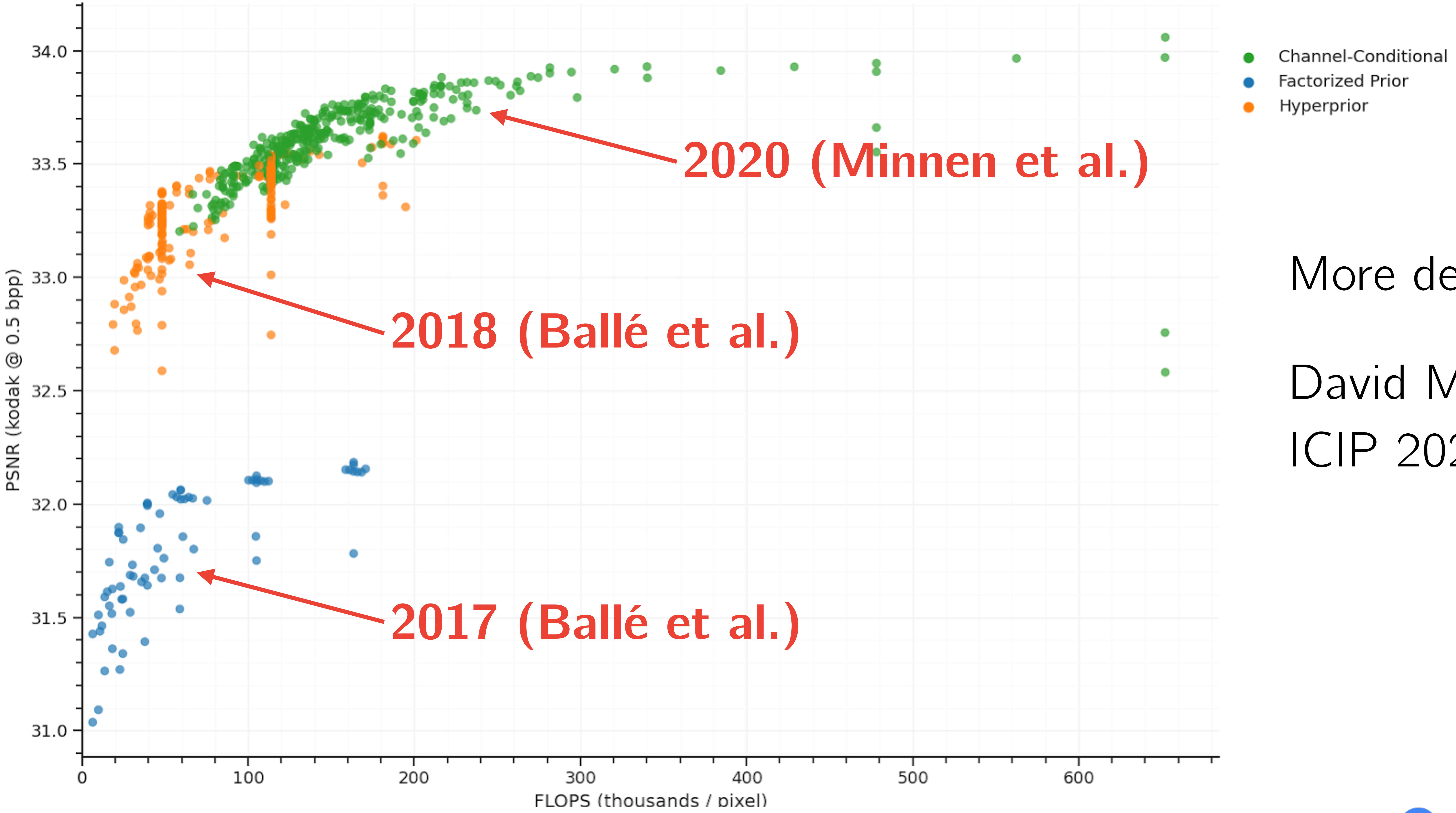
**Learned compression:**

typically

$O(\text{enc}) \approx O(\text{dec})$

Google Research

# Rate–distortion–complexity trade-off




More detail:

David Minnen's
ICIP 2021 keynote

# Progress in learned compression of natural images over the last few years

✅ One model for many RD-points

✅ Competitive in terms of PSNR

❓ Computational complexity

❓ Subjective image quality

optimized for MSE
0.129 bpp

optimized for MS-SSIM
0.129 bpp

Ballé et al. (ICLR, 2018)

optimized for MSE
0.129 bpp

**optimized for MS-SSIM**
**0.129 bpp**

Ballé et al. (ICLR, 2018)

optimized for MSE

0.129 bpp

**optimized for MS-SSIM**

**0.129 bpp**

Ballé et al. (ICLR, 2018)

optimized for MSE

0.194 bpp

optimized for MS-SSIM

0.187 bpp

Ballé et al. (ICLR, 2018)

optimized for MSE

0.194 bpp

**optimized for MS-SSIM**

**0.187 bpp**

Ballé et al. (ICLR, 2018)

optimized for MSE

0.194 bpp

**optimized for MS-SSIM**

**0.187 bpp**

Ballé et al. (ICLR, 2018)

Google Research

# Observation:

- Rate allocation decisions are “amortized” into the networks:
  They learn to distribute bits where they are most needed.
- **Explicit** control of bitrate allocation during compression is not necessary.
- Distortion metric does not need to be evaluated “in the loop”.

**We can use a lot more sophisticated perceptual models than before!**

# What are perceptual models?

Emulate humans on tasks like these:

- Does the image look realistic?
- Do the two images look identical?
- How realistic does the image look?
- How bad is the image degraded compared to the original?

# What are perceptual models?

Emulate humans on tasks like these:

- Does the image look realistic?
- Do the two images look identical?
- How realistic does the image look?
- How bad is the image degraded compared to the original?

two-alternative forced choice (2AFC)

# What are perceptual models?

mean opinion score (MOS)

Emulate humans on tasks like these:

- Does the image look realistic?
- Do the two images look identical?
- How realistic does the image look?
- How bad is the image degraded compared to the original?

# What are perceptual models?

"full reference" / distortion

Emulate humans on tasks like these:

- Does the image look realistic?
- Do the two images look identical?
- How realistic does the image look?
- How bad is the image degraded compared to the original?

# What are perceptual models?

Emulate humans on tasks like these:

- Does the image look realistic?
- Do the two images look identical?
- How realistic does the image look?
- How bad is the image degraded compared to the original?

"no reference" / realism

Google Research

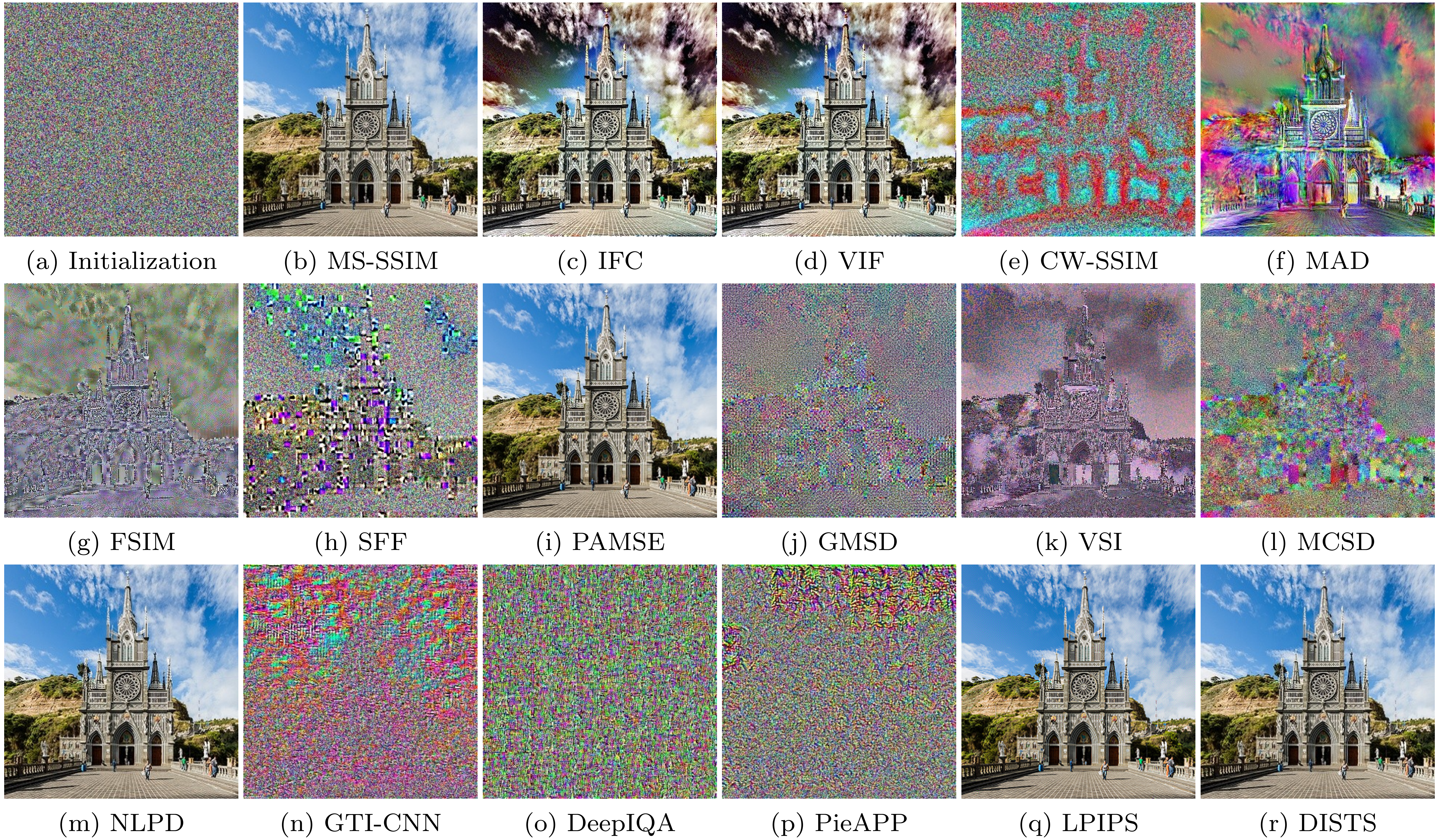
Part II

# Distortion

# What do we need from a distortion metric?

- Highly predictive of human ratings
- Differentiable and well-defined (e.g., $d(\boldsymbol{x}, \boldsymbol{x}) = 0$)
- Generalize well across types of images and types of distortions

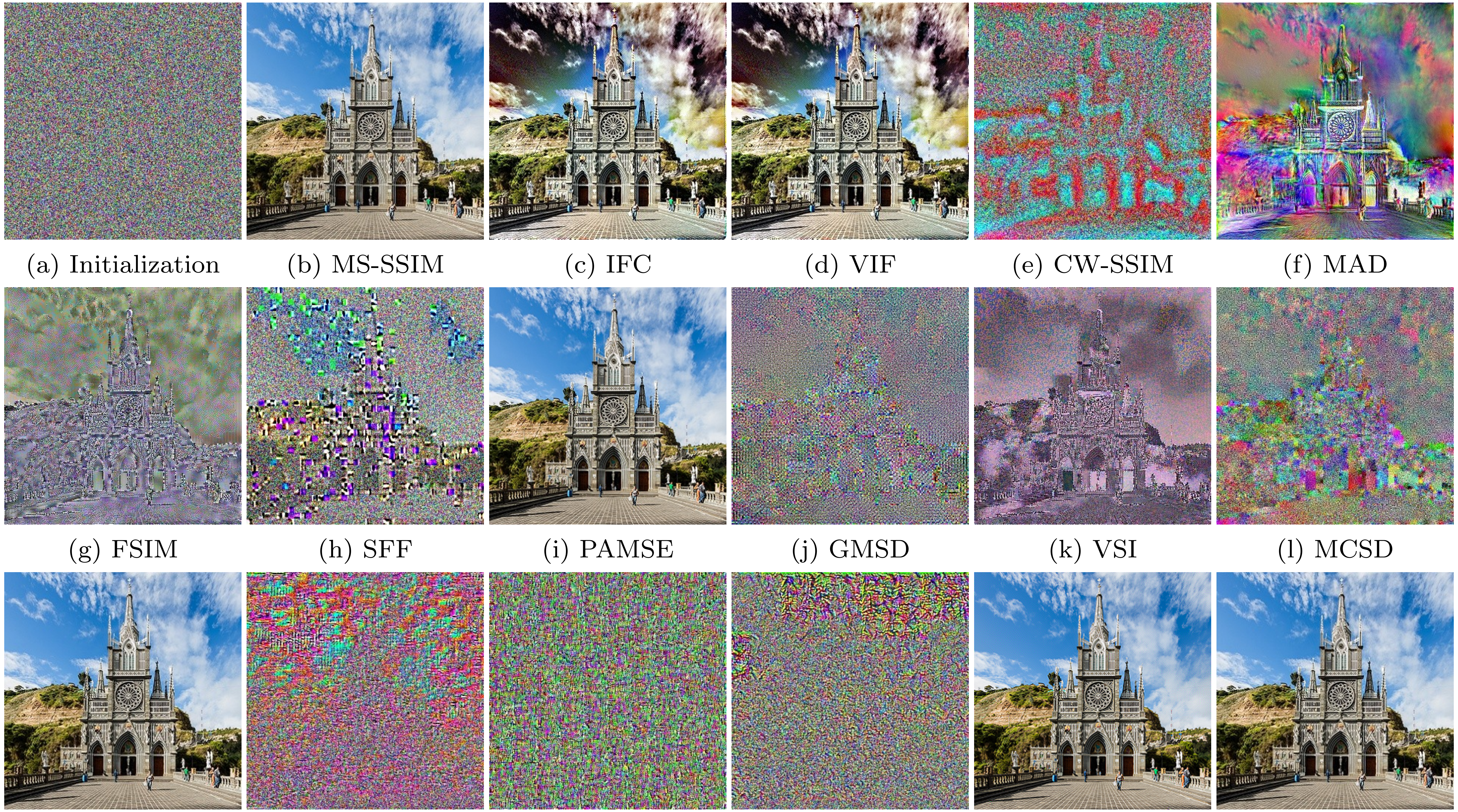
# Some distortion metrics have “blind spots”




from arXiv, with author’s permission

Ding et al. (IJCV, 2021)

# Some distortion metrics have "blind spots"



(a) Initialization (b) MS-SSIM (c) IFC (d) VIF (e) CW-SSIM (f) MAD (g) FSIM (h) SFF (i) PAMSE (j) GMSD (k) VSI (l) MCSD (m) NLPD (n) GTI-CNN (o) DeepIQA (p) PieAPP (q) LPIPS (r) DISTS

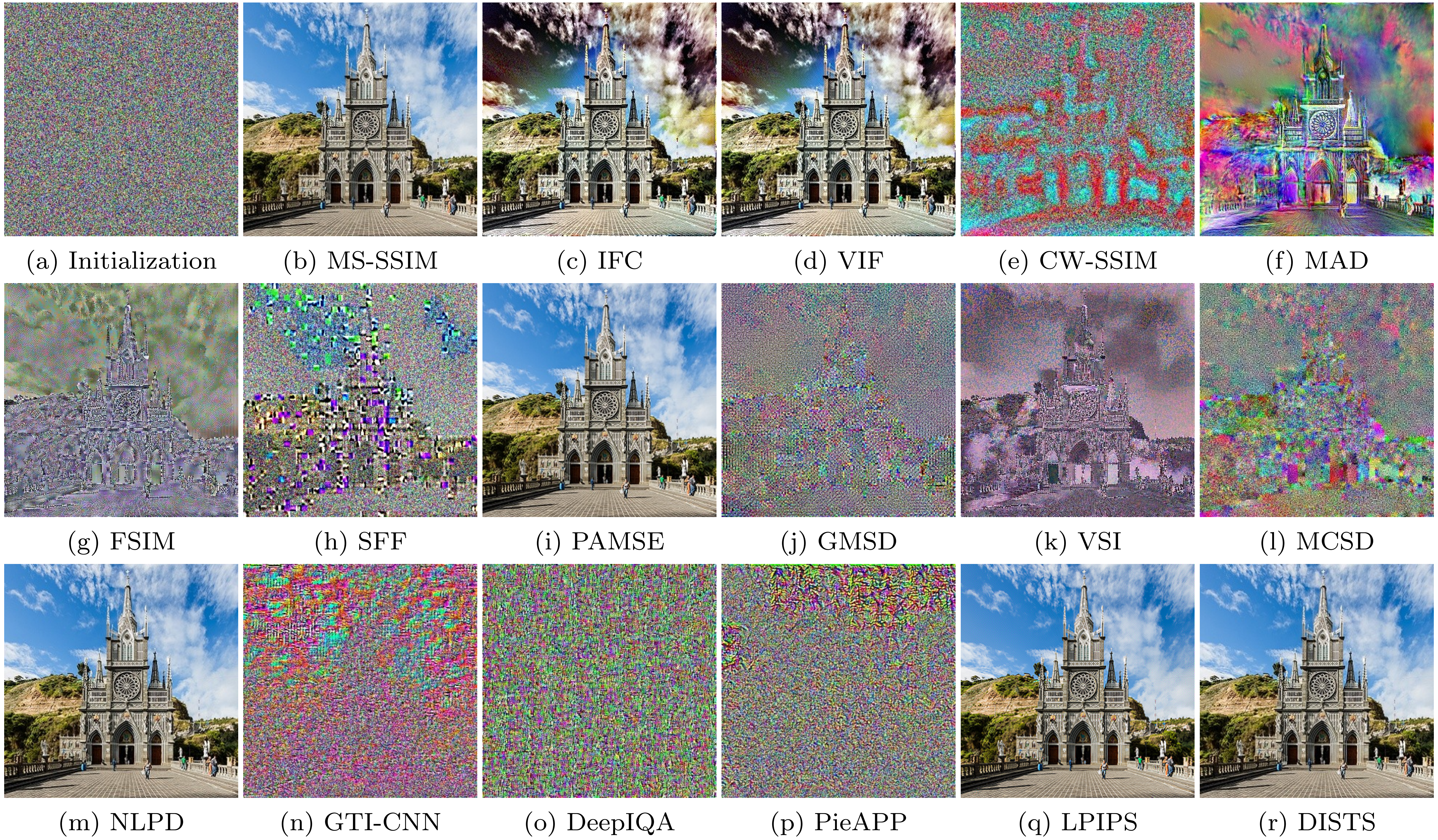
from arXiv, with author's permission

**Experiment:**

1. Initialize $\hat{x}$ to noise

Ding et al. (IJCV, 2021)

# Some distortion metrics have "blind spots"




from arXiv, with author's permission

**Experiment:**

1. Initialize $\hat{x}$ to noise
2. Minimize $d(\boldsymbol{x}, \hat{\boldsymbol{x}})$ over $\hat{x}$

Ding et al. (IJCV, 2021)

# Optimizable metrics need to generalize better

What does it mean to “generalize”?

- For quality assessment, we evaluate the metric on a joint distribution:

$$p(x, \hat{x}) = p(x)\, p(\hat{x}|x)$$

natural image distribution

distribution of compression artifacts

- For training a compression model, we evaluate the metric on a potentially much larger domain (and also need to take derivatives there).

# Even worse: IQA datasets have blind spots, too

**TID**, **LIVE**, **CSIQ**, etc.: calibrated, but typically no structural distortions

**BAPPS**: crowd-sourced patch ratings including structural distortions


https://github.com/richzhang/PerceptualSimilarity, BSD license

Zhang et al. (CVPR, 2018)

# What do we need from a distortion metric?

- Highly predictive of human ratings
- Differentiable and well-defined (e.g., $d(\boldsymbol{x}, \boldsymbol{x}) = 0$)
- Generalize well across types of images and types of distortions 😕
- **Neural networks can and will "cheat", because they are less constrained in what types of artifacts they can produce.**

## Part III

# Realism

# No-reference metrics, reinterpreted

Idealized "critic" $T$ uses likelihood ratio between natural image distribution and distribution of reconstructions:

$$T(\mathbf{x}) = f'\left(\frac{p_{\mathbf{x}}(\mathbf{x})}{p_{\hat{\mathbf{x}}}(\mathbf{x})}\right)$$

In contrast to distortion, the critic "learns" a model of the distribution of artifacts.

Many no-reference metrics are in fact specialized to detect a **particular source of artifacts** – same generalization problem here.

Nowozin et al. (NeurIPS, 2016)

# No-reference metrics, reinterpreted

Idealized "critic" $T$ uses likelihood ratio between natural image distribution and distribution of reconstructions:

$$T(\mathbf{x}) = f'\left(\frac{p_{\mathsf{X}}(\mathbf{x})}{p_{\hat{\mathsf{X}}}(\mathbf{x})}\right)$$

**GANs generate realistic images by playing an "adversarial" optimization game between a critic and a generator.**

The generator learns to produce images that fool the critic, while the critic learns to classify images into "real or fake".

Nowozin et al. (NeurIPS, 2016)

# No-reference metrics, reinterpreted

Taking the expectation, we can define realism as an $f$-divergence between the two distributions:

$$D_f = \mathbb{E}_{\mathbf{x} \sim p_{\hat{\mathbf{x}}}} f\left(\frac{p_{\mathbf{x}}(\mathbf{x})}{p_{\hat{\mathbf{x}}}(\mathbf{x})}\right)$$

For example, for $f(r) = r \log r$, we recover the Kullback–Leibler divergence.

**Adding an "adversarial loss" to the training of a compression model is one way to achieve better realism.**


Nowozin et al. (NeurIPS, 2016)
Blau & Michaeli (CVPR, 2019)

# Distortion and realism are at odds with each other

Blau & Michaeli (CVPR, 2019)

*authors use the term "perception" for realism

# Distortion and realism are at odds with each other

**original**



**Blau & Michaeli (CVPR, 2019)**

***authors use the term "perception" for realism**

# Distortion and realism are at odds with each other

**reconstruction optimized for:**

**original**

Ubaid kareem, CC BY-SA, Wik. Cmns.

**rate + distortion**

**distortion: great**

**realism: bad**

**Blau & Michaeli (CVPR, 2019)**

***authors use the term "perception" for realism**

# Distortion and realism are at odds with each other

**reconstruction optimized for:**

**original**

Ubaid kareem, CC BY-SA, Wik. Cmns.

**rate + distortion**

**distortion: great**

**realism: bad**

**rate + realism**

clairity, CC BY, flickr.com

**distortion: bad**

**realism: great**

**Blau & Michaeli (CVPR, 2019)**

***authors use the term "perception" for realism**

Google Research

# Distortion and realism are at odds with each other

**reconstruction optimized for:**

**original**

Ubaid kareem, CC BY-SA, Wik. Cmns.

**rate + distortion**

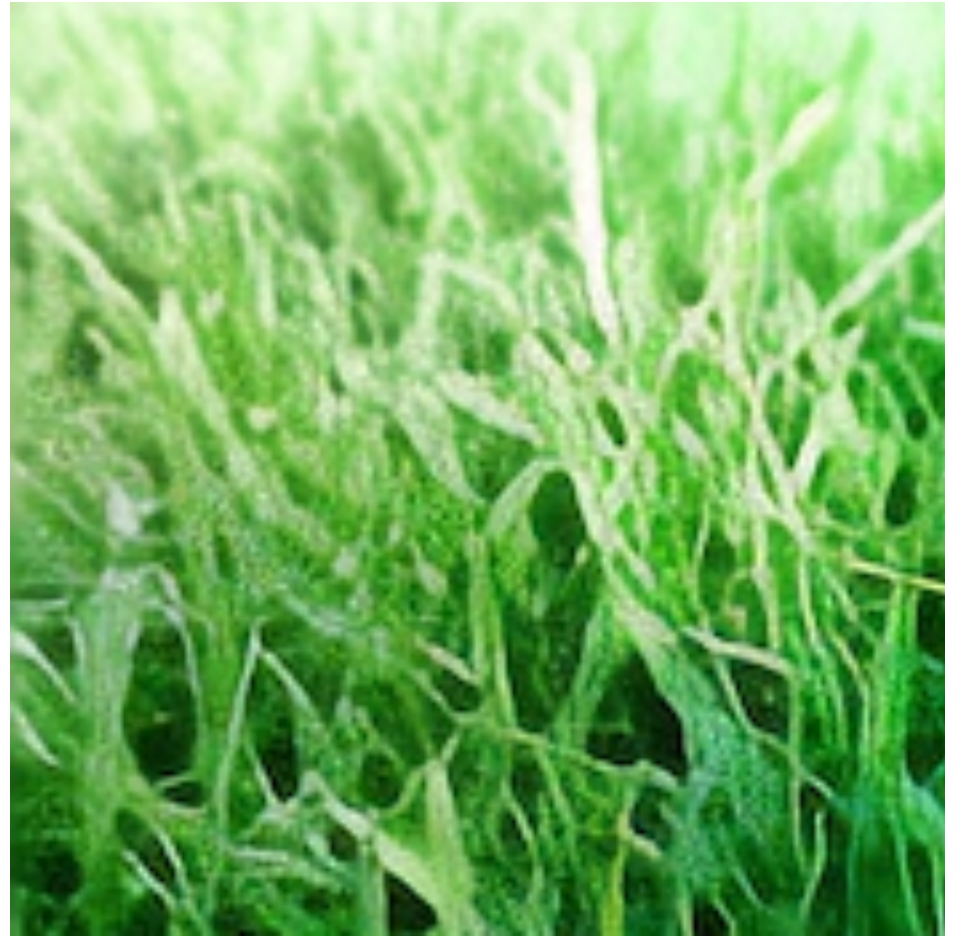
**distortion: great**
**realism: bad**

**rate + realism**

clairity, CC BY, flickr.com

**distortion: bad**
**realism: great**

**rate + dist. + real.**



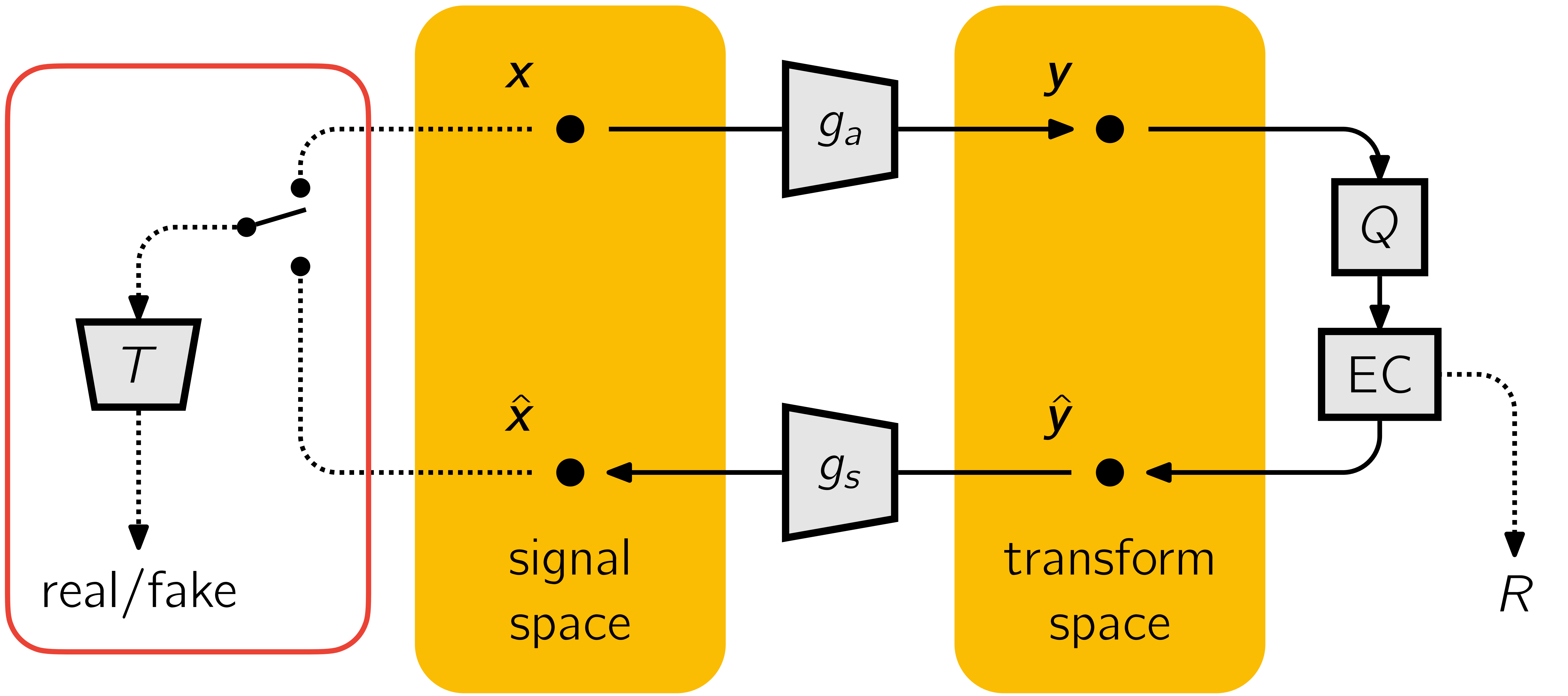
**distortion: good**
**realism: good**

**Blau & Michaeli (CVPR, 2019)**

***authors use the term "perception" for realism**

# Improving realism with adversarial losses

**adversarial loss, in addition to distortion loss**



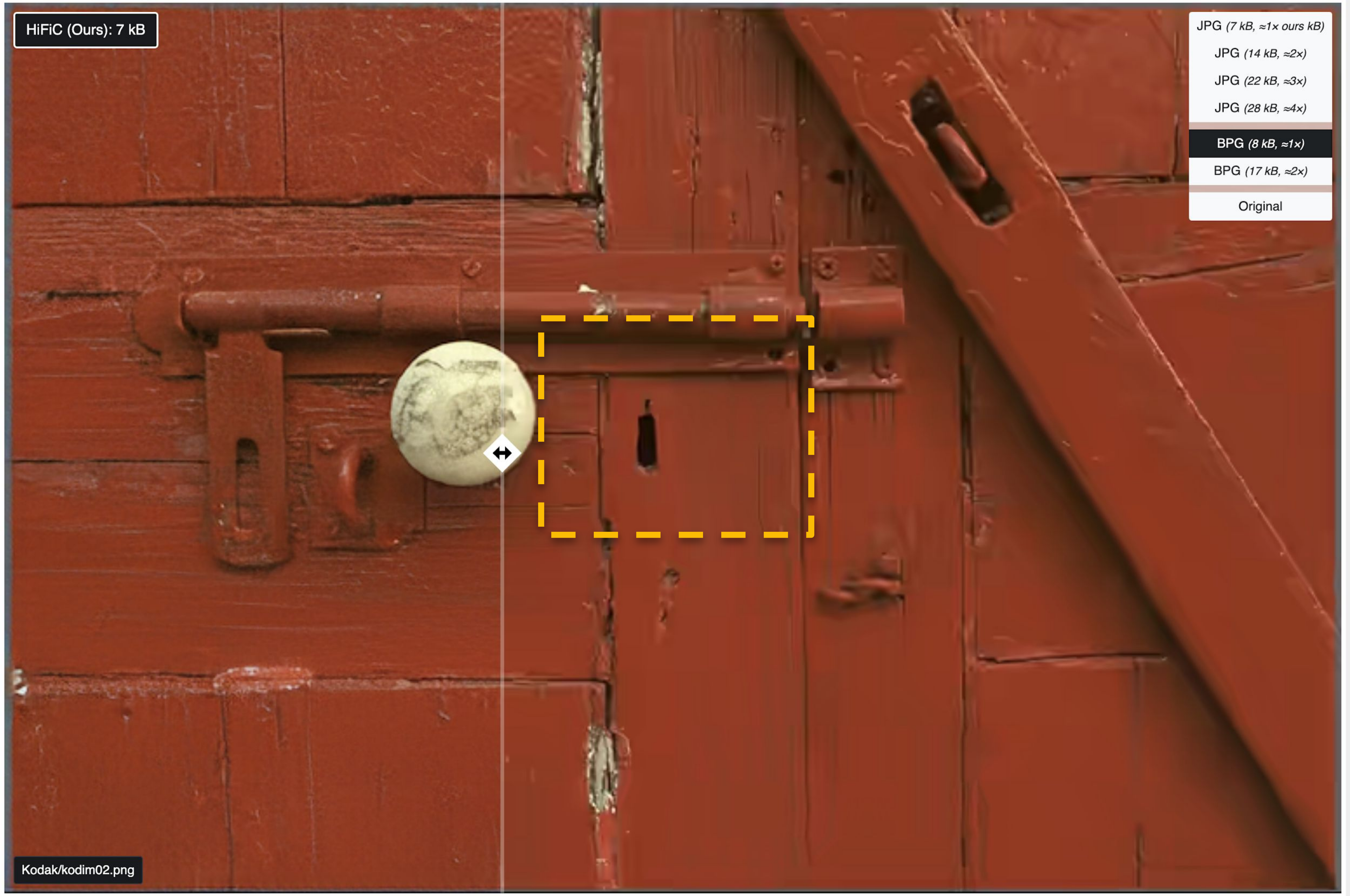

$$L = \mathbb{E}\left[-\log_2 p(\tilde{\boldsymbol{y}})\right] + \lambda\, \mathbb{E}\left[d(\boldsymbol{x}, \tilde{\boldsymbol{x}})\right] + \kappa\, \tilde{D}_f$$

**rate** **distortion** **realism**

# HiFiC model

Larger synthesis transform network

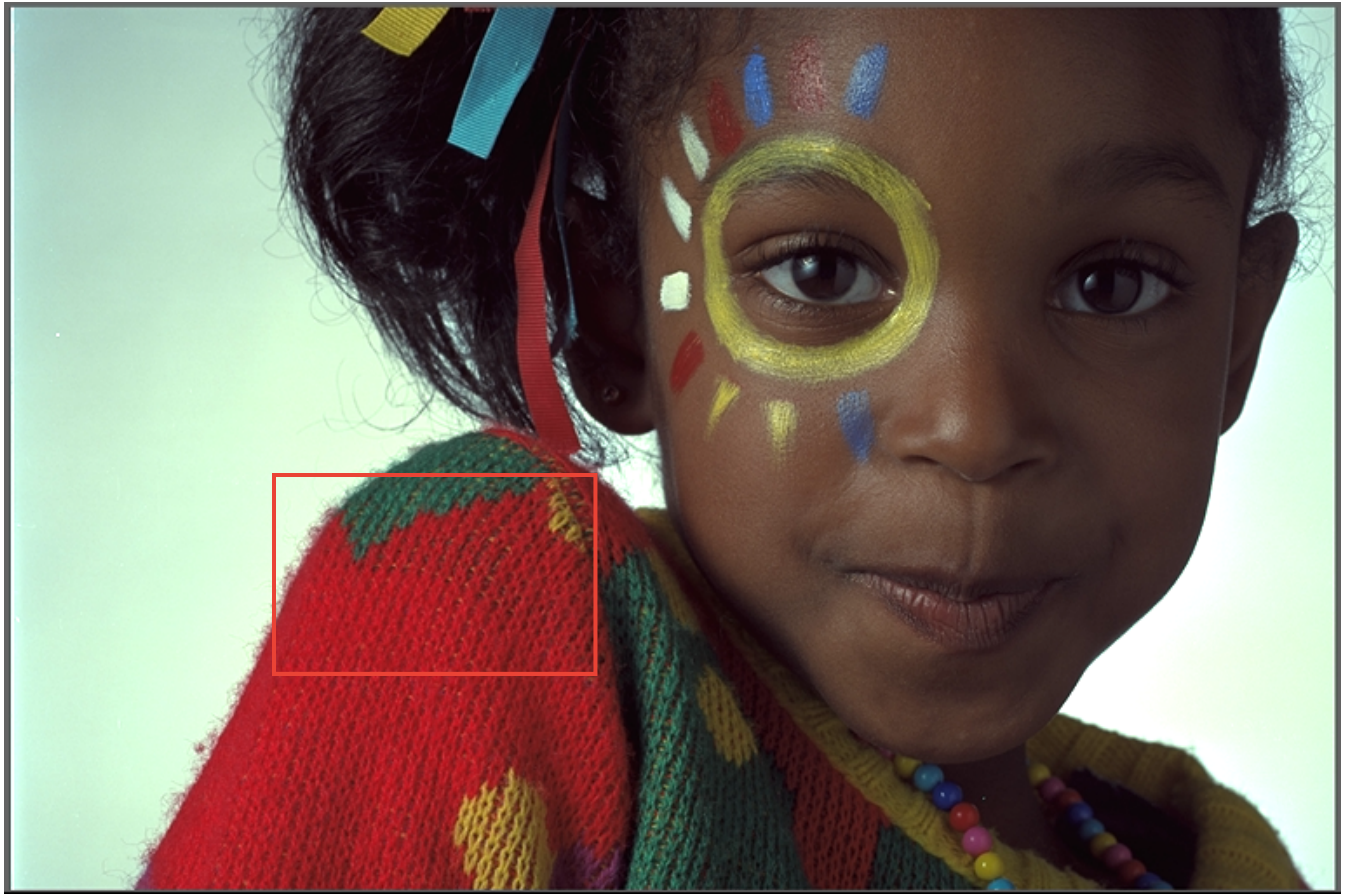
Uses a distortion loss of MSE + LPIPS and a conditional patch-level critic




Interactive demo @ **hific.github.io**

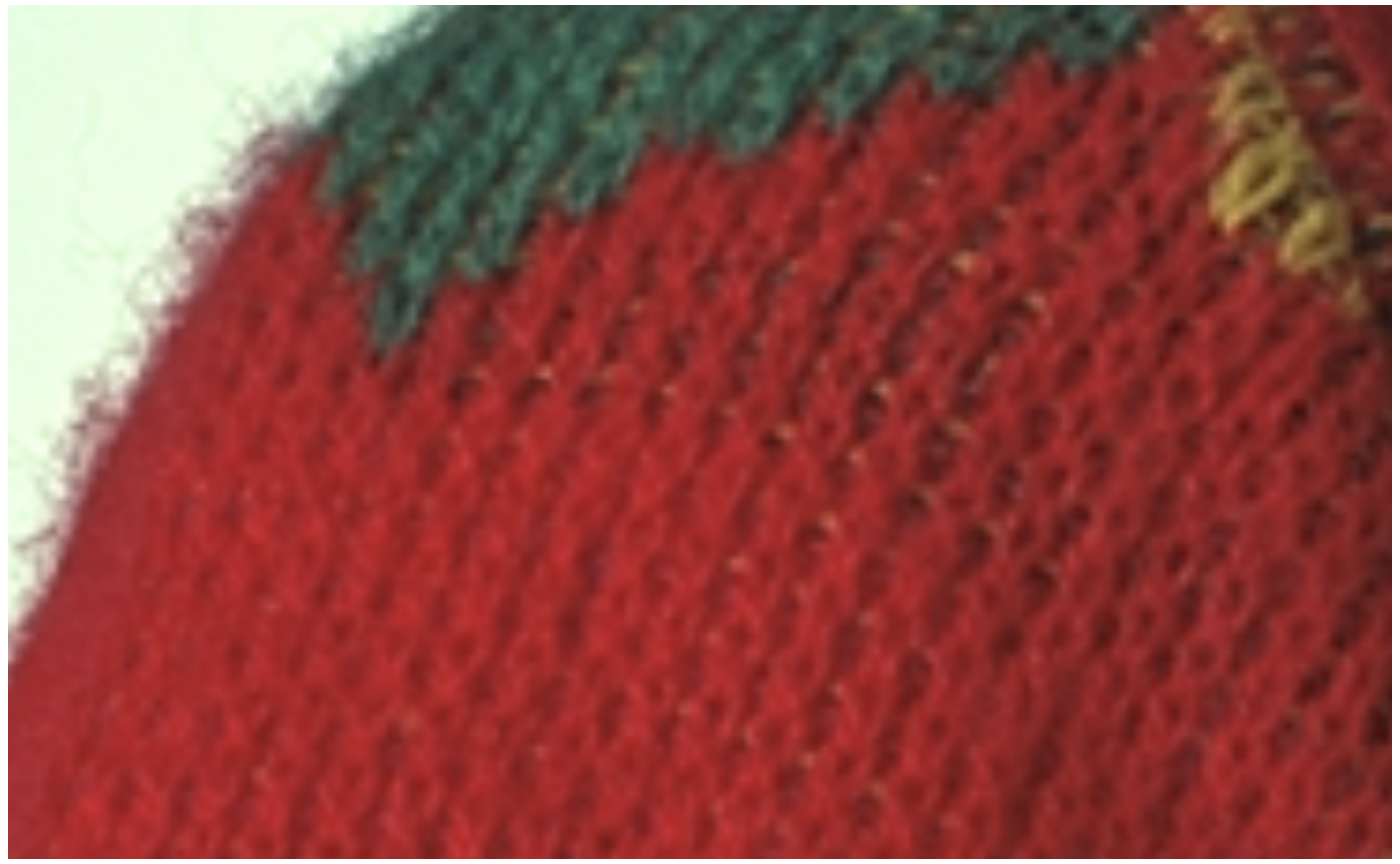
**Mentzer et al. (NeurIPS, 2020)**


Mentzer et al. (NeurIPS, 2020)



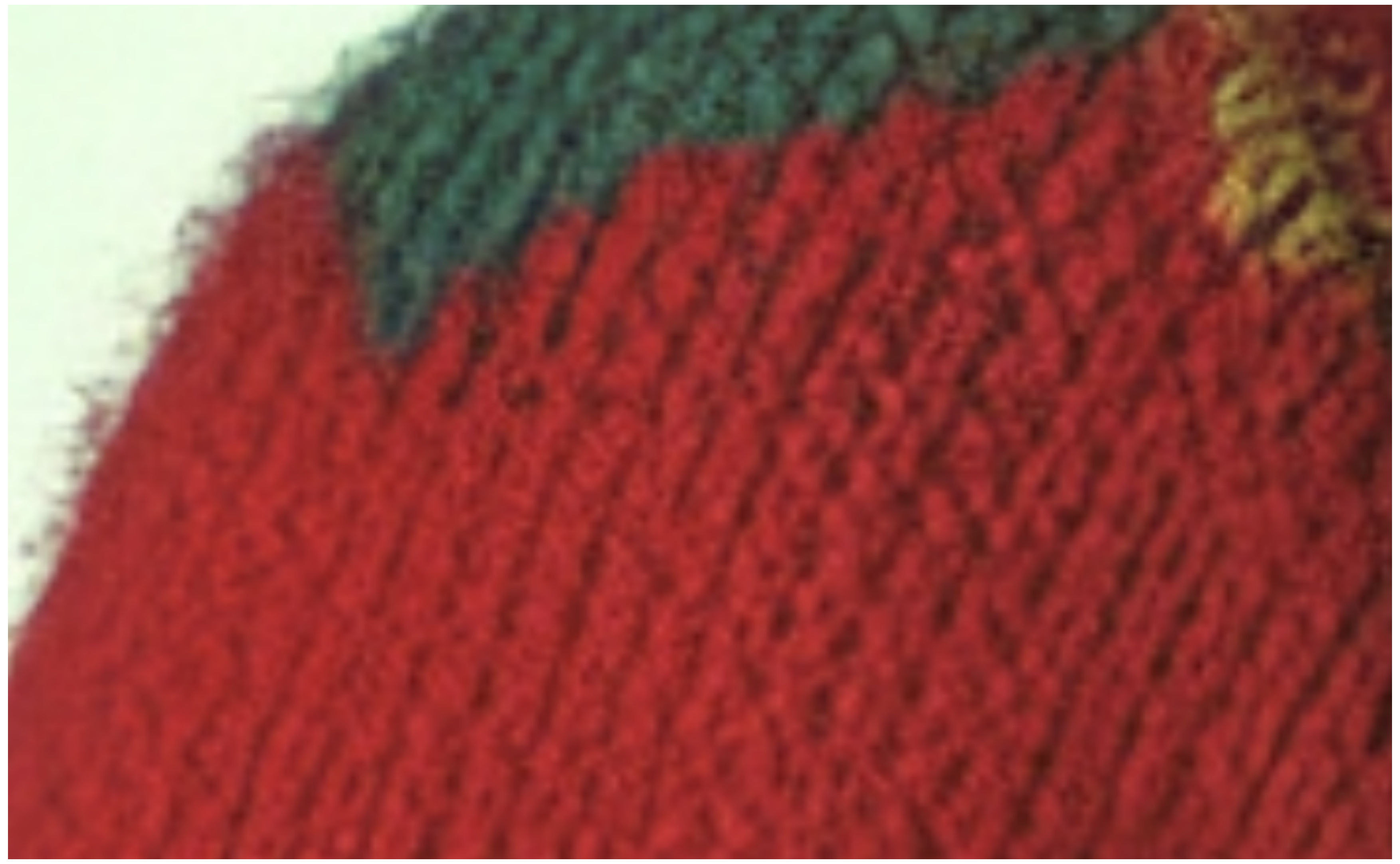
Mentzer et al. (NeurIPS, 2020)

Original



BPG @8kB

HiFiC @7kB



BPG @15kB

Mentzer et al. (NeurIPS, 2020)


↓ **Lower is better** for all metrics


**HiFiC** at 0.237 bpp prefered to:

- **BPG** at 2.1x the bitrate
- **Mean & Scale** hyperprior (trained for MSE at 1.7x the bitrate)


**Adding GAN loss boosts subjective quality**

# **HiFiC Failure Cases:** small faces


# Optimizing for realism helps, but isn't enough

(Other ANN-based techniques have been developed to reproduce the natural image distribution better, such as **diffusion** processes.)

Many such models are applied to "patch of pixels" representation, hence aim to produce matching **pixel-level** distributions.

However, matching pixel distributions may not be ideal, since pixel representations don't take into account human perception (e.g. **sensitivity to faces, text**).

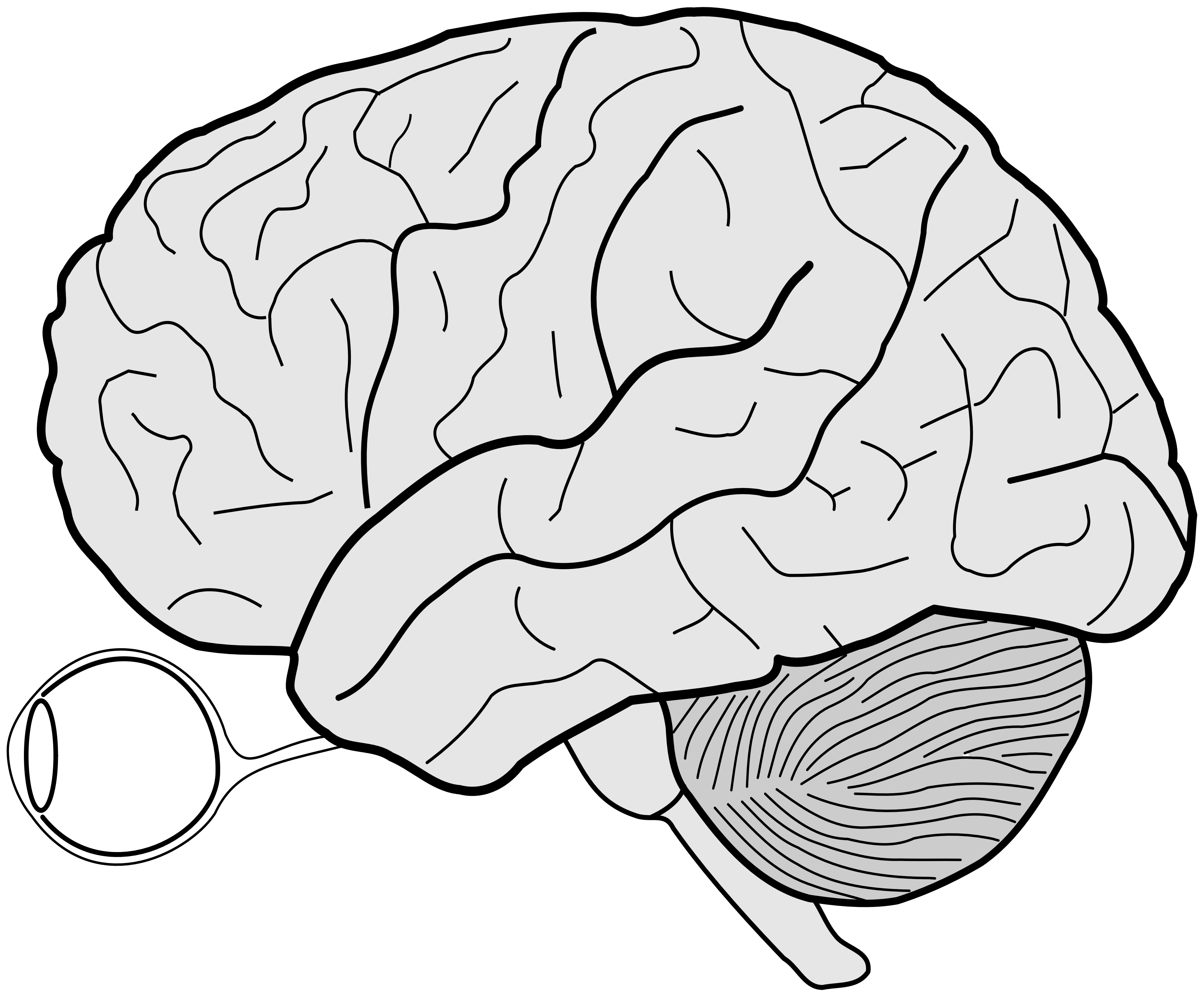
## Part IV

# Perceptual Spaces

# The brain extracts and filters information from the environment

This happens on many levels:

- via physiological constraints (e.g. by the type and distribution of photoreceptors in the eye)
- by pre-attentive processing (e.g. spatial/ temporal masking effects)
- or even cognitively (e.g. attention)

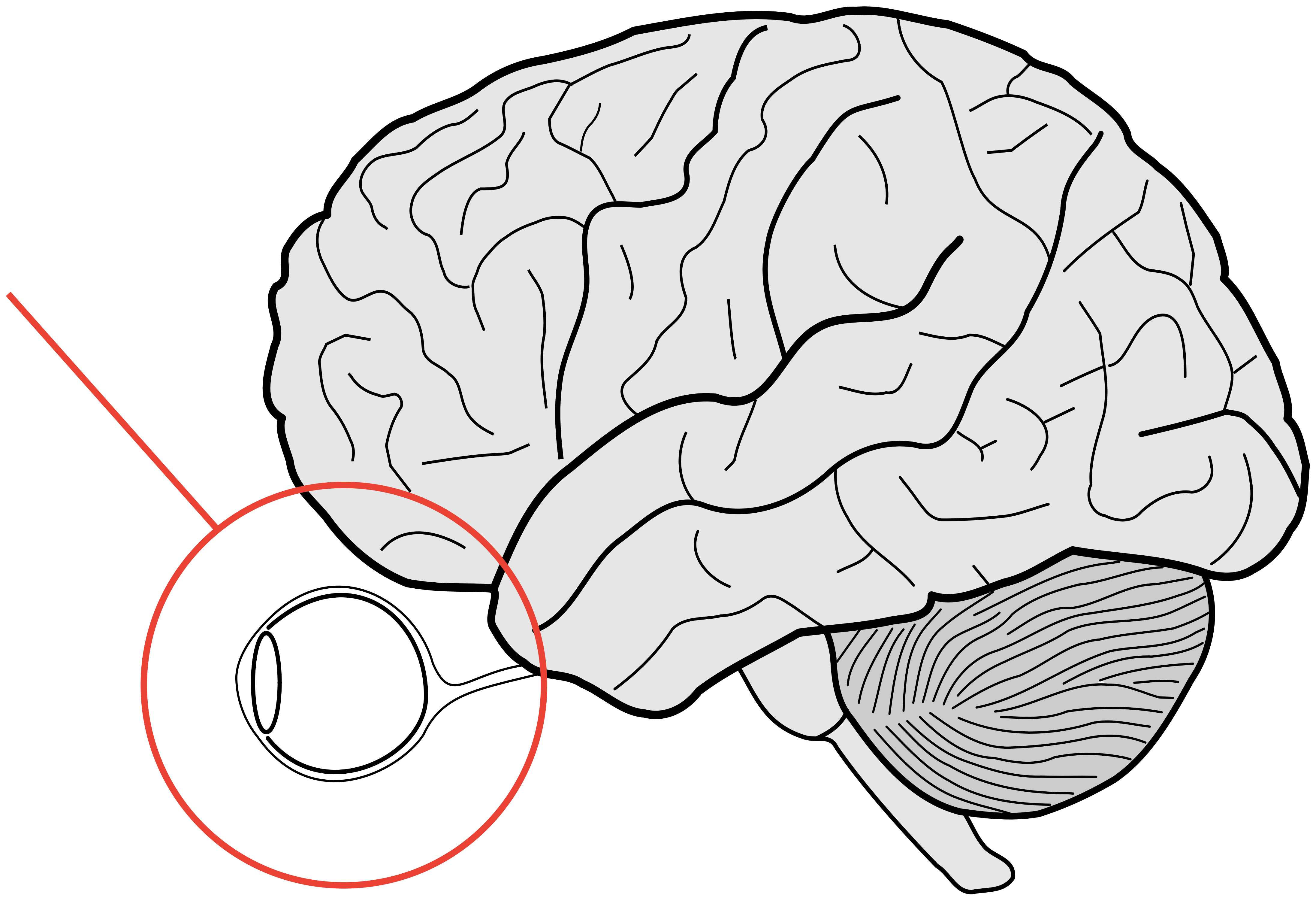


Hankem, Public Domain, via Wikimedia Commons

# The brain extracts and filters information from the environment

This happens on many levels:

- via physiological constraints (e.g. by the type and distribution of photoreceptors in the eye)
- by pre-attentive processing (e.g. spatial/ temporal masking effects)
- or even cognitively (e.g. attention)

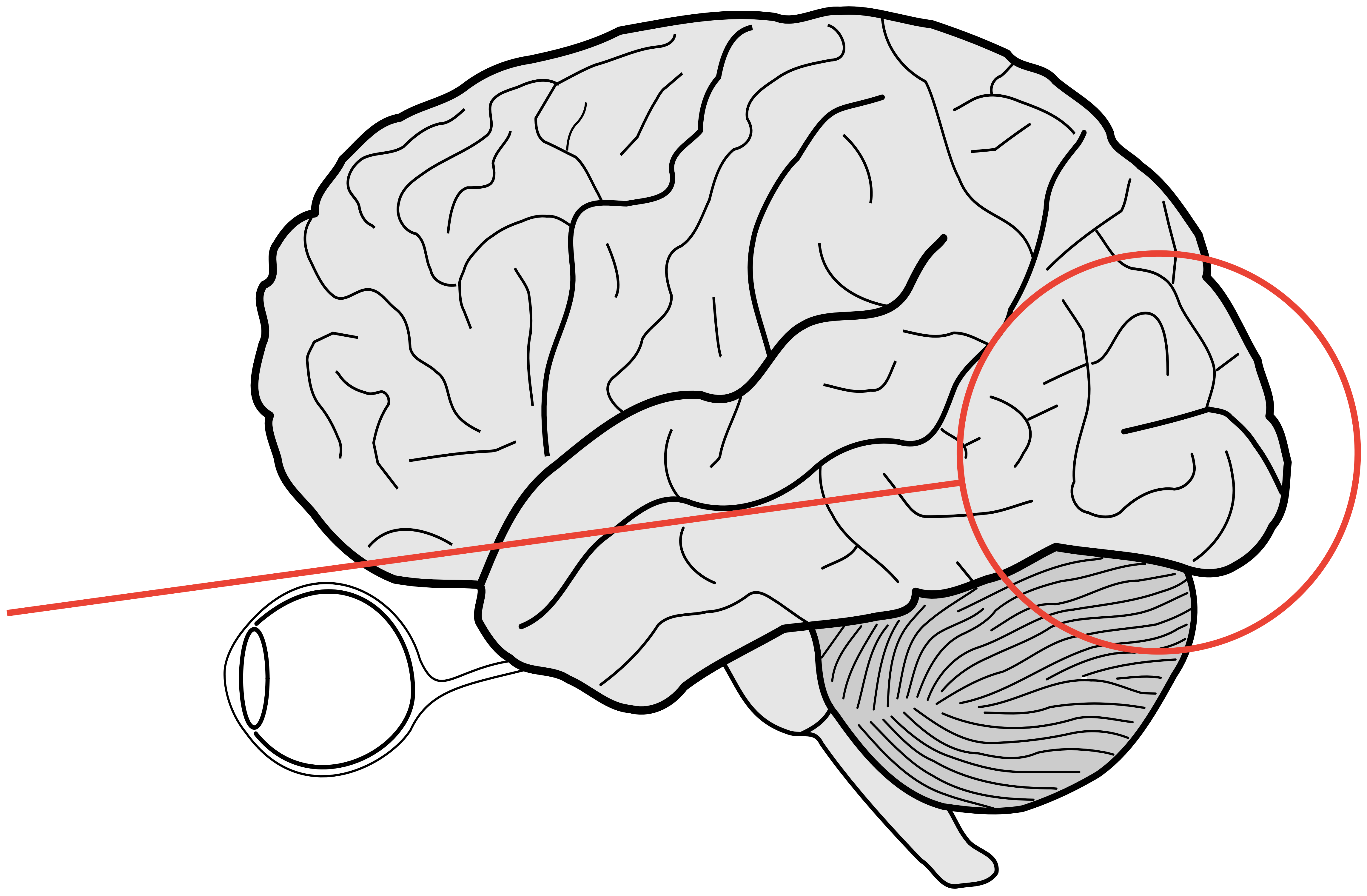


Hankem, Public Domain, via Wikimedia Commons

# The brain extracts and filters information from the environment

This happens on many levels:

- via physiological constraints (e.g. by the type and distribution of photoreceptors in the eye)
- by pre-attentive processing (e.g. spatial/ temporal masking effects)
- or even cognitively (e.g. attention)

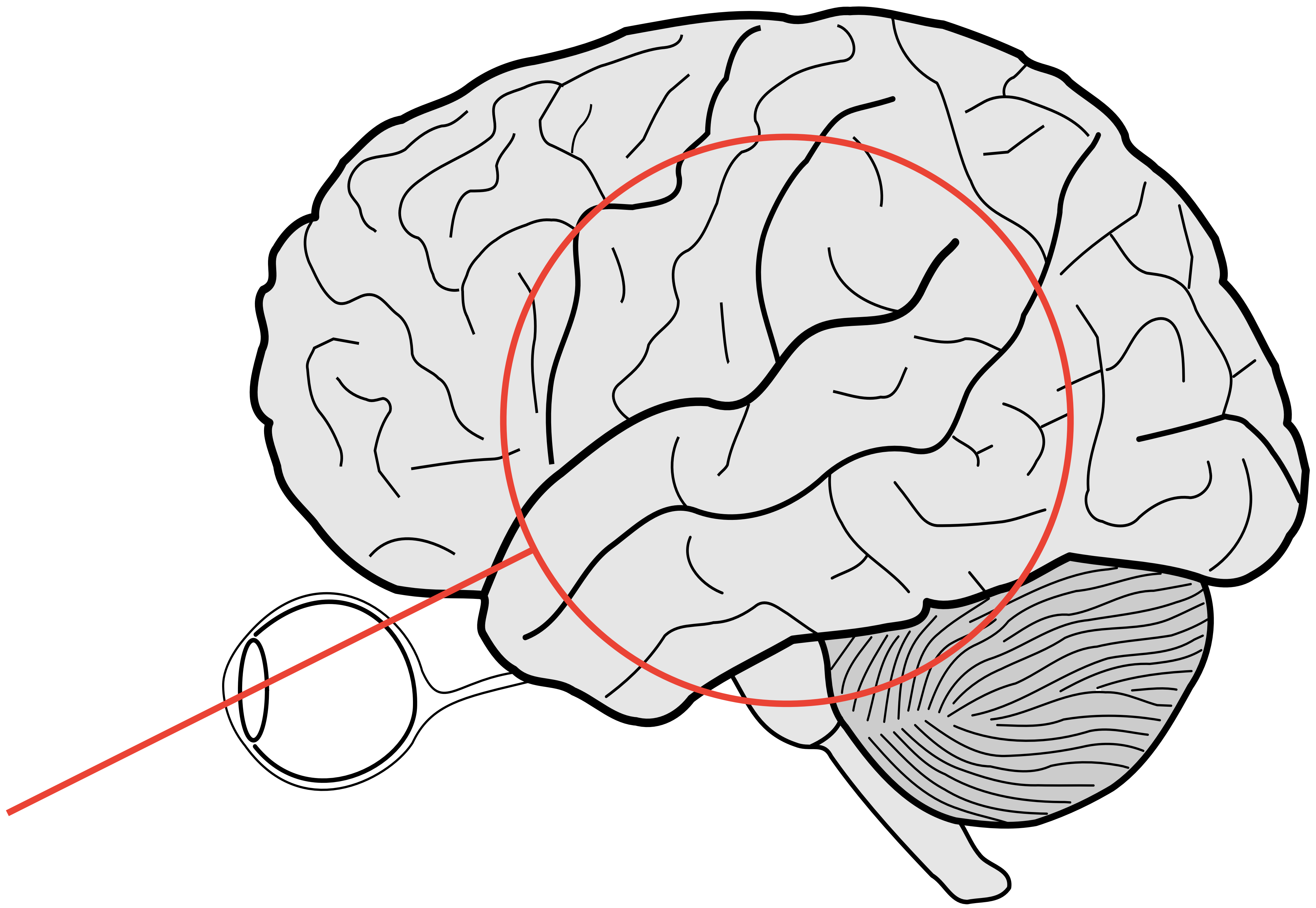


Hankem, Public Domain, via Wikimedia Commons

# The brain extracts and filters information from the environment

This happens on many levels:

- via physiological constraints (e.g. by the type and distribution of photoreceptors in the eye)
- by pre-attentive processing (e.g. spatial/ temporal masking effects)
- or even cognitively (e.g. attention)



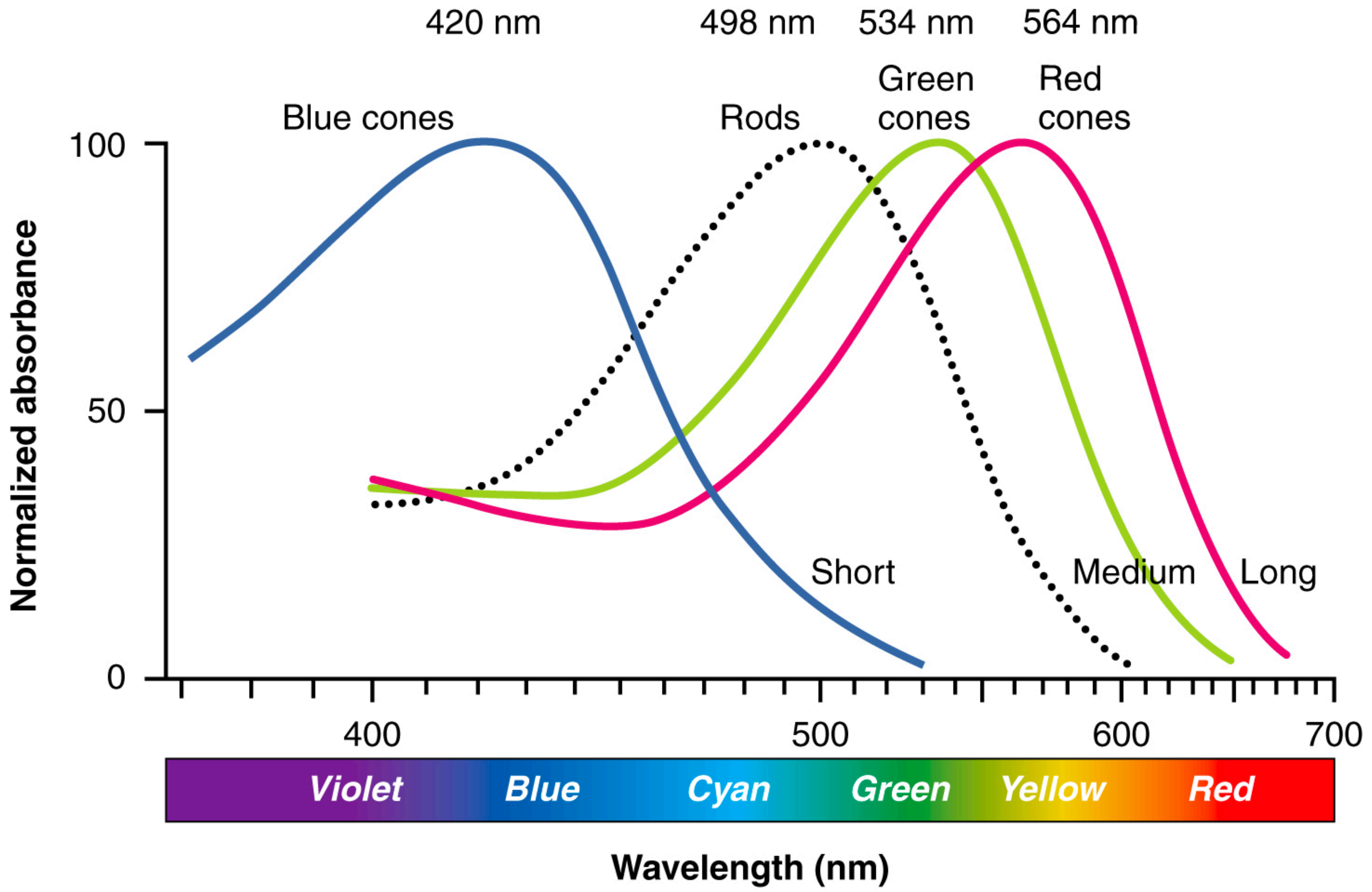
Hankem, Public Domain, via Wikimedia Commons

# Low-level example

Infinite mixture of wavelengths of light hits three different types of retinal photoreceptors.

Many different spectral power distributions all appear as the same color (**metamer**).

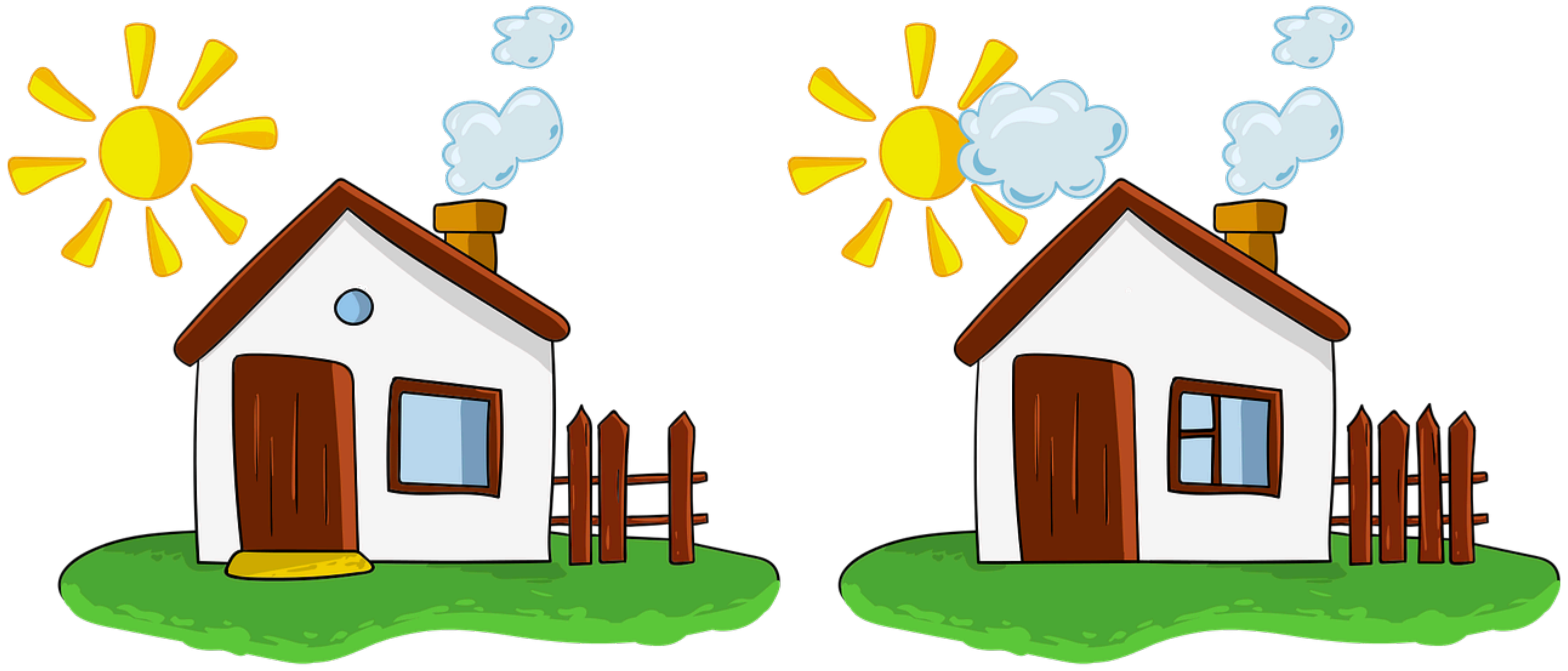
Easy to forget about, since it is already “baked in” to illumination, display, and camera tech!



# High-level example

Cognitive processes, such as solving a given task, can affect perception.

For example, recognizing the differences in the cartoon on the right depends on where we direct our **attention**.



Dmitry Abramov, via Pixabay

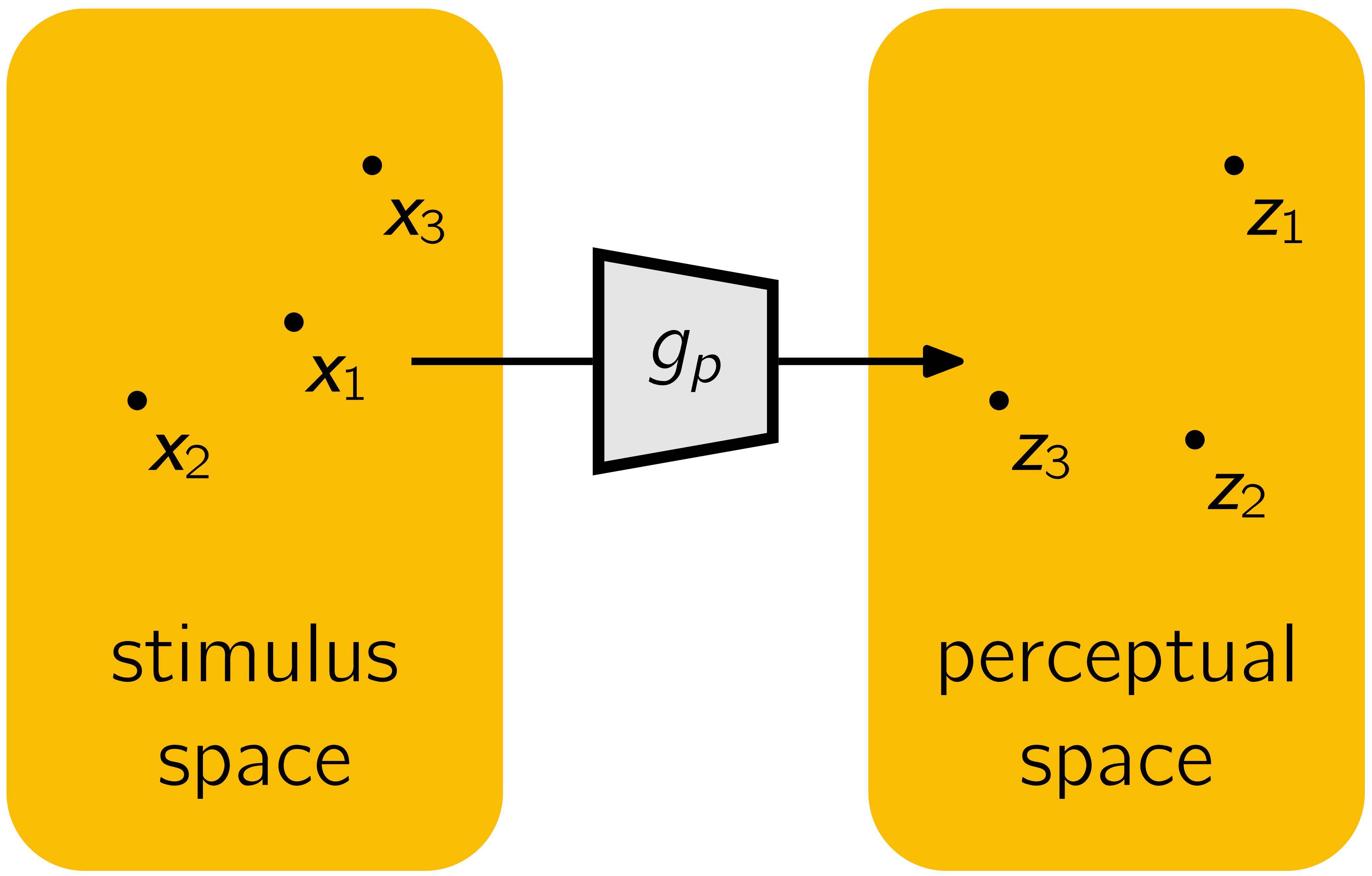
# Many IQA models use proxy representations

We can think of each stimulus (signal) as a point in space.

A transformation brings each point into a perceptual space.

In this space, distances between points predict human judgments of similarity.

Sets of points representing the just noticeable difference (JND) ideally are spherical.



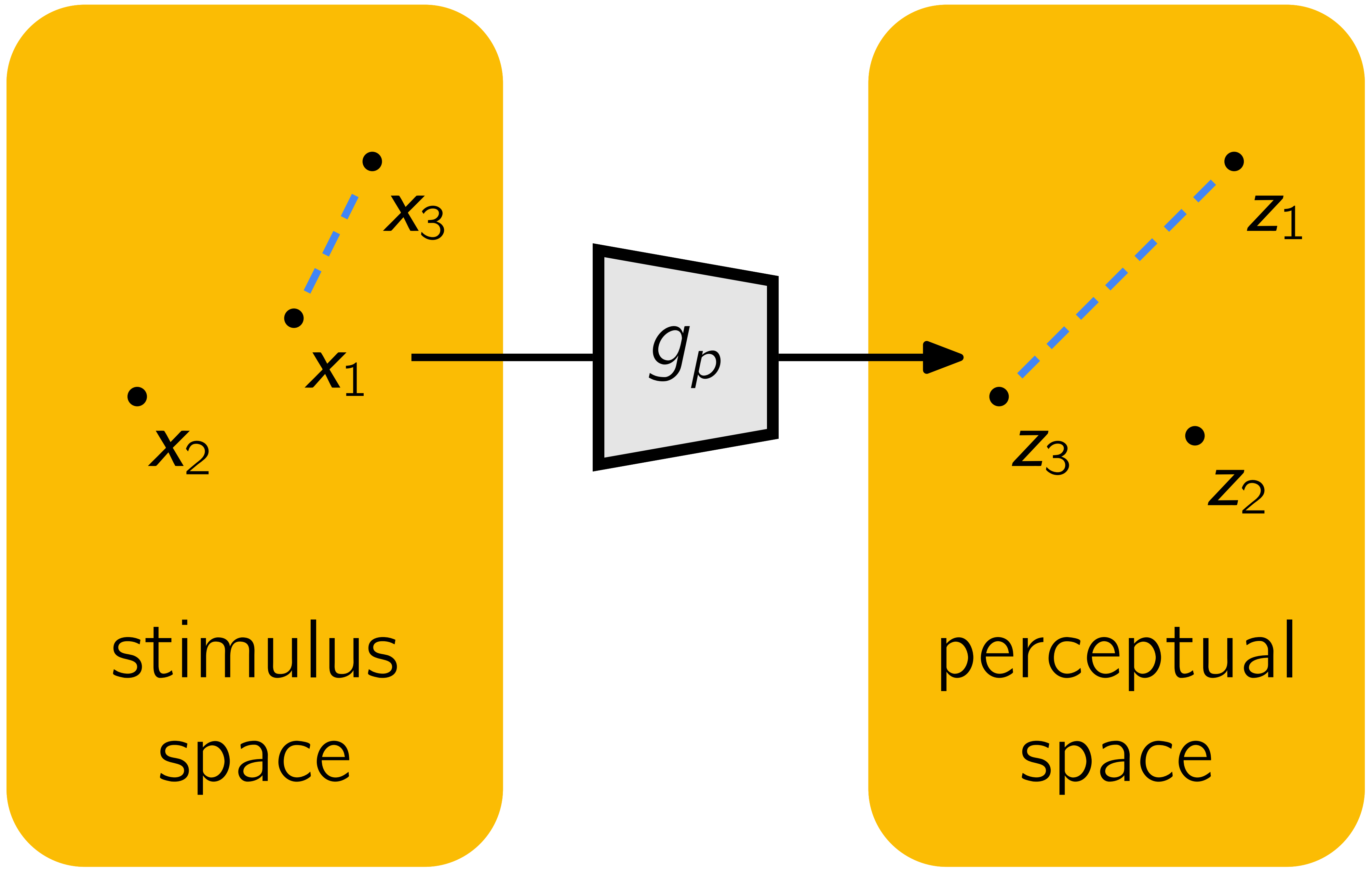
# Many IQA models use proxy representations

We can think of each stimulus (signal) as a point in space.

A transformation brings each point into a perceptual space.

In this space, distances between points predict human judgments of similarity.

Sets of points representing the just noticeable difference (JND) ideally are spherical.



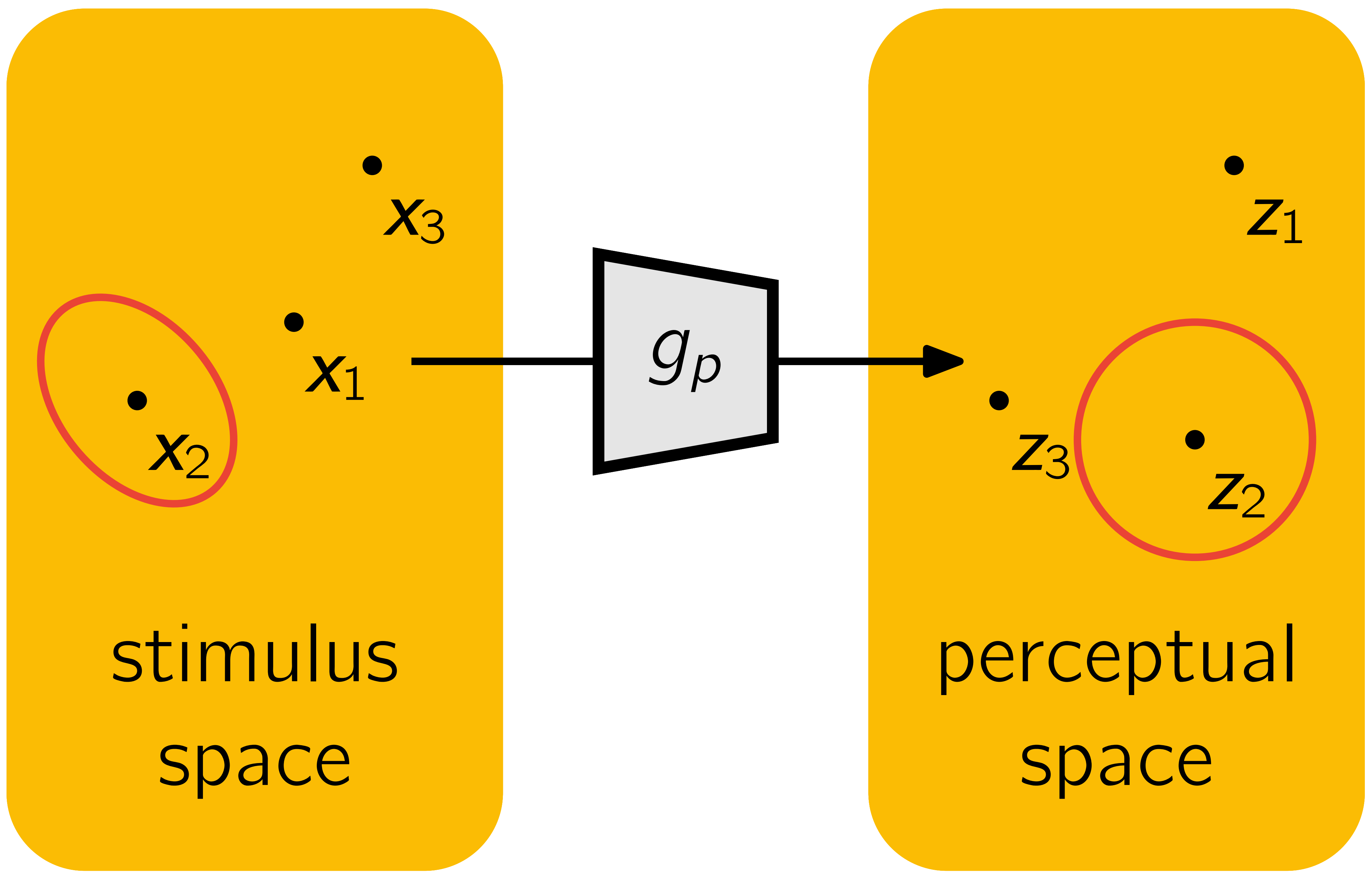
# Many IQA models use proxy representations

We can think of each stimulus (signal) as a point in space.

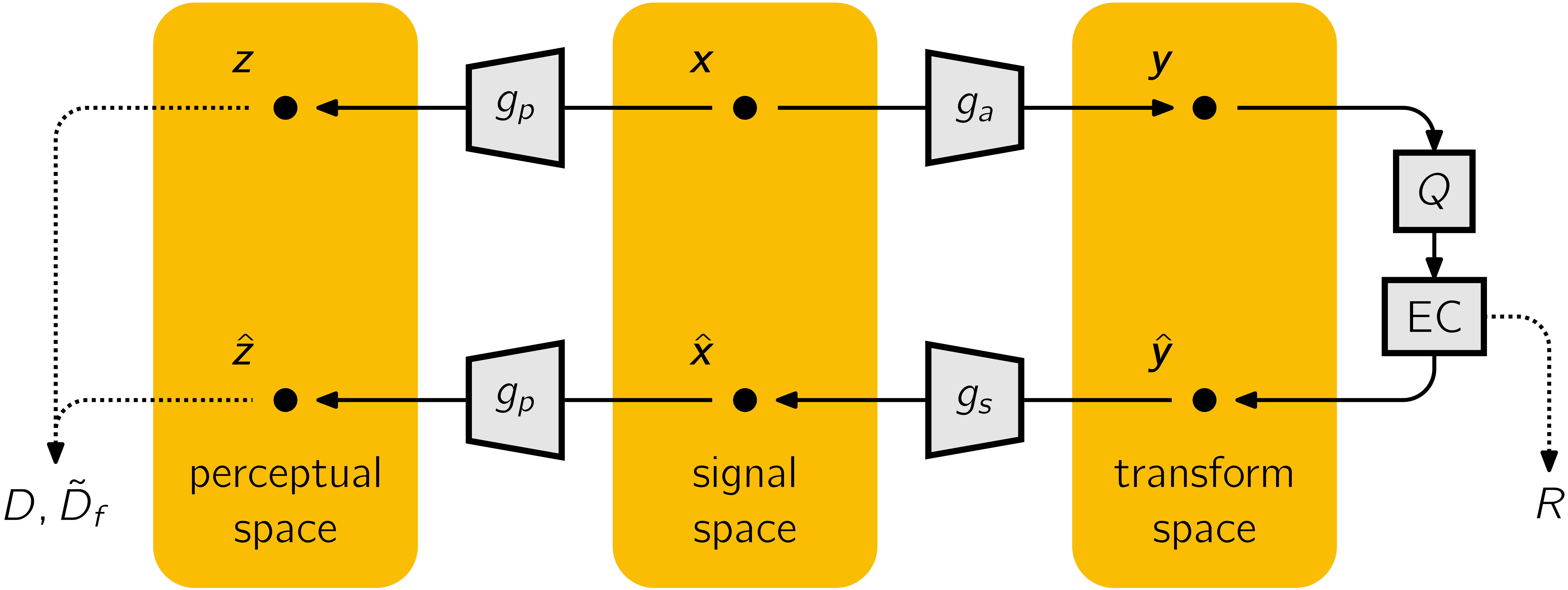
A transformation brings each point into a perceptual space.

In this space, distances between points predict human judgments of similarity.

Sets of points representing the just noticeable difference (JND) ideally are spherical.



# Perceptually optimized compression



# (Early!) example

MacAdam (1942): Ellipses correspond to just-noticeable differences in chromaticity.

Color spaces such as CIE Lab, and many more, are designed to “warp” the space such that ellipses turn into equal-sized circles.

Then, (Euclidean) distances predict perceived color similarity.

**CIE 1931 xy chromaticity diagram**

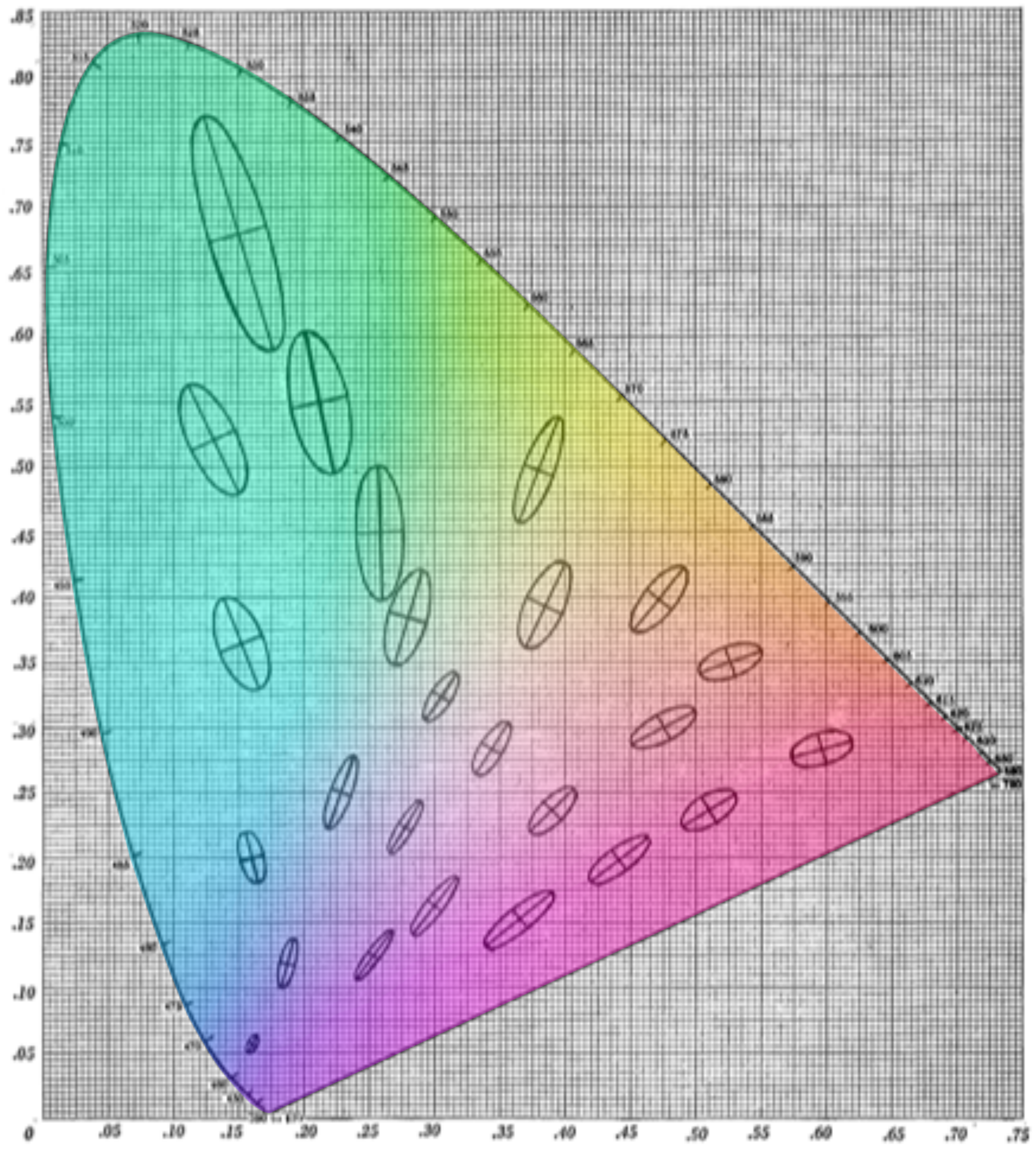


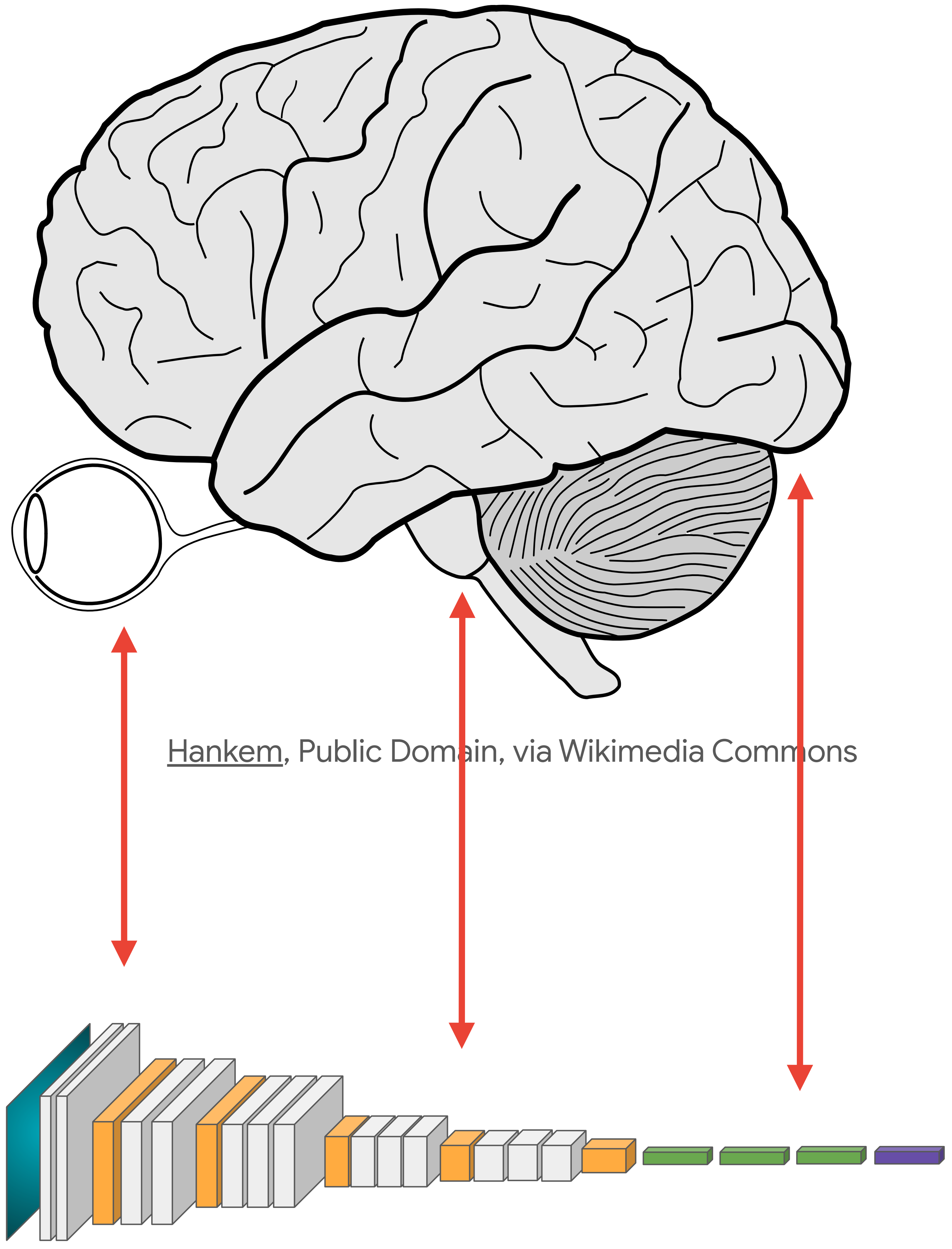

Fig. 48. Standard deviations of chromaticity from indicated standards, represented ten times actual scale on I.C.I. 1931 standard chromaticity diagram, observer: PGN.

**MacAdam (JOSA, 1942)**

# Object recognition features as perceptual spaces

Object recognition features have neural correlates in the visual system.

Use object recognition as a proxy task to construct a perceptual space?



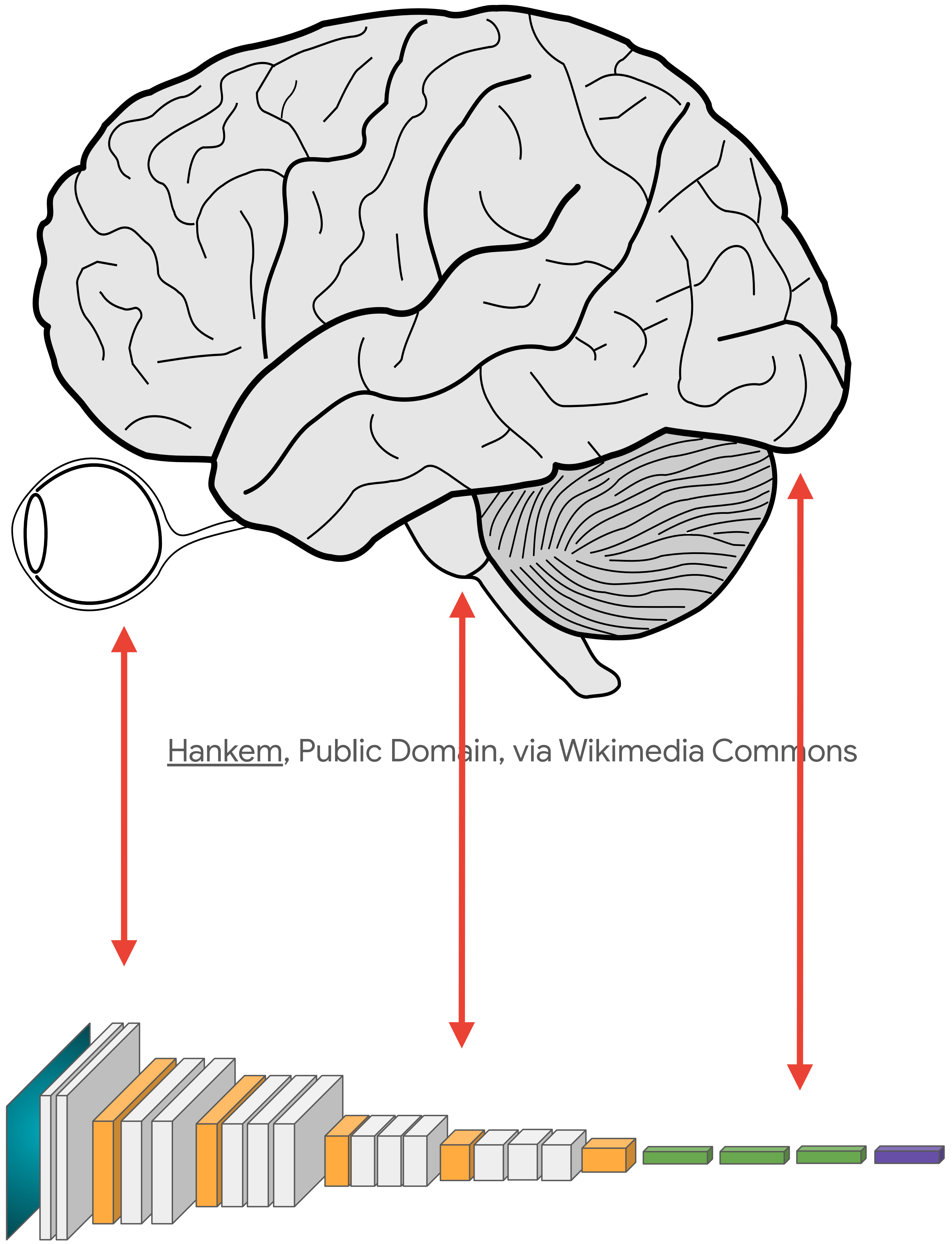

Yamins & DiCarlo (Nature Neuroscience, 2016)

# Example: LPIPS

Highly predictive of human annotations even on structural distortions

However, feature representations require significant amounts of human responses:

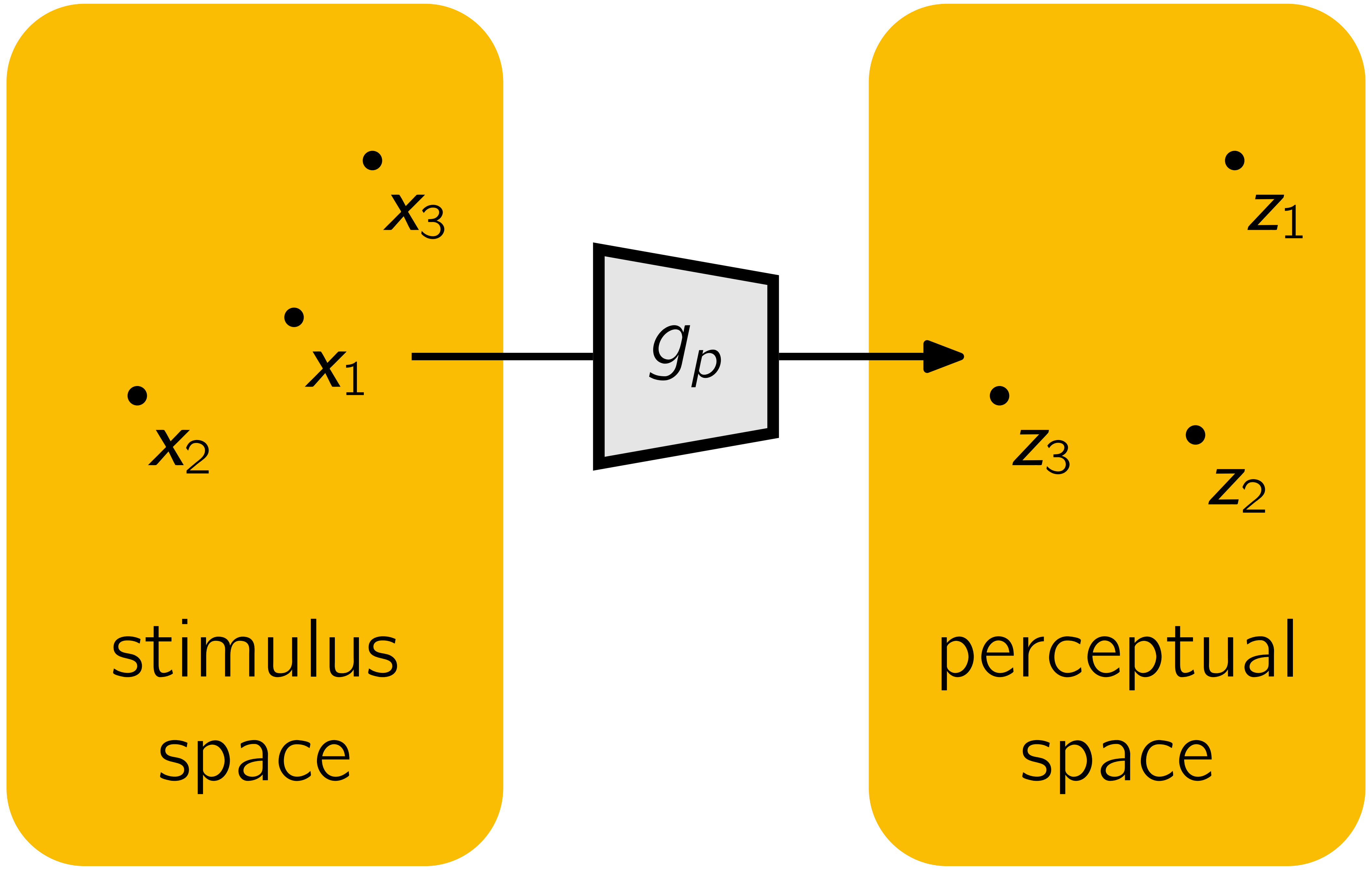
- First, for training proxy task (classification labels)
- Second, for training task adaptation layers (IQA ratings)




Zhang et al. (CVPR, 2018)

# Learned perceptual spaces

Can we build a representation from first principles, without using human responses?



# PIM: An Unsupervised Information-Theoretic Perceptual Quality Metric

**Learn an image representation**, imposing principles/constraints borrowed from computational neuroscience:

- **Slowness**: relevant visual features tend to be persistent in time (Földiák, 1991; Mitchison, 1991; Wiskott, 2003)
- **Efficient coding**: brain "compresses" sensory information (Attneave, 1954; Barlow, 1961)
- Approximate **translation and scale equivariance**: well-known properties of representations in human visual system

**Bhardwaj et al. (NeurIPS, 2020)**

# The slowness principle


Jesse Millan, CC BY 2.0, via Flickr

Slow features, persistent across time, tend to coincide with relevant features

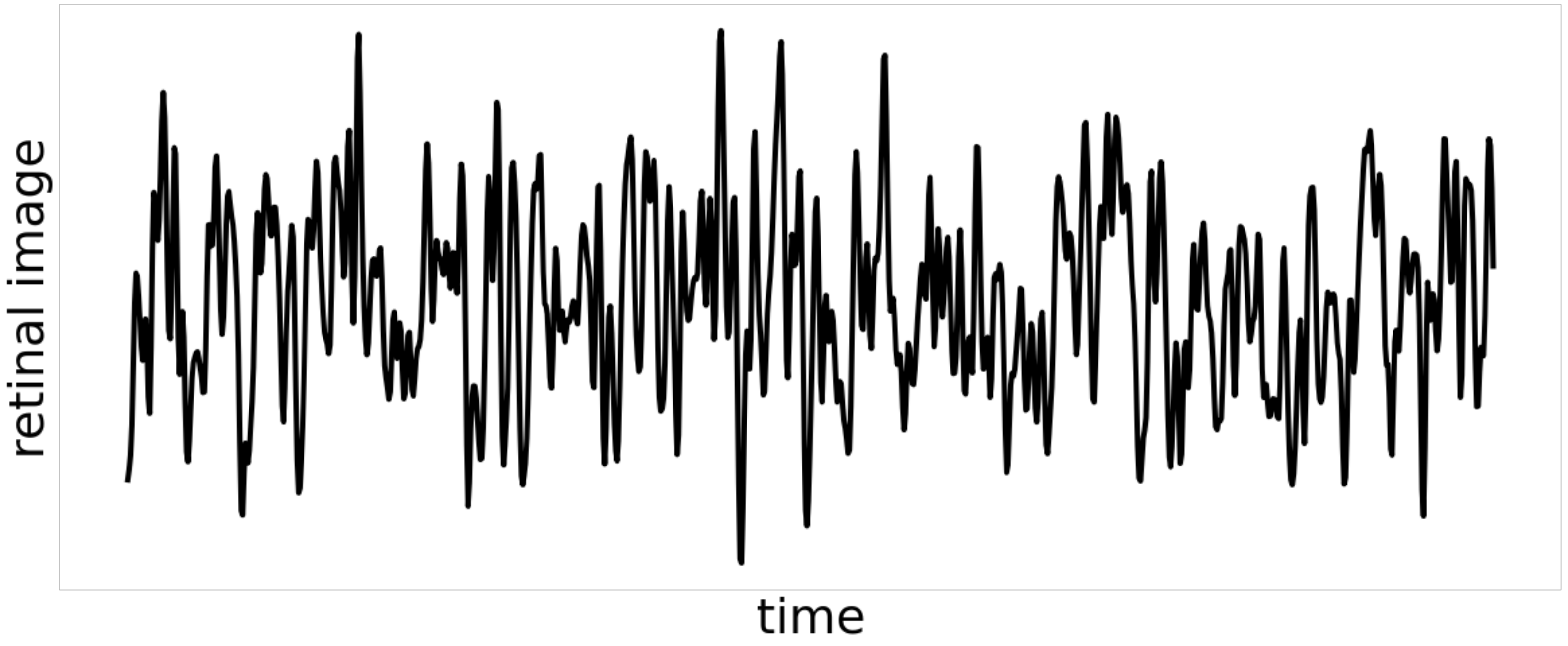
Fast features tend to contain "nuisance" information

# The slowness principle

Slow features, persistent across time, tend to coincide with relevant features

Fast features tend to contain "nuisance" information



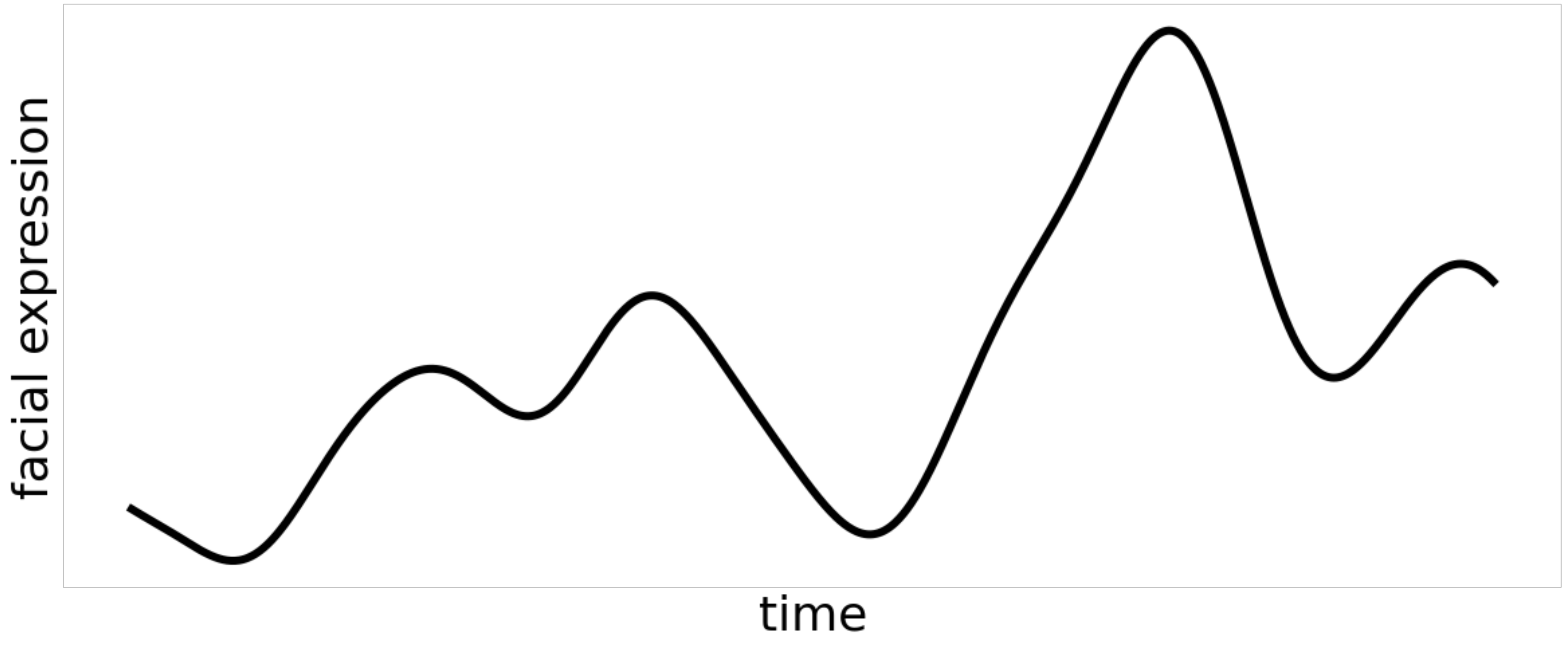


# The slowness principle

Slow features, persistent across time, tend to coincide with relevant features

Fast features tend to contain "nuisance" information


Jesse Millan, CC BY 2.0, via Flickr

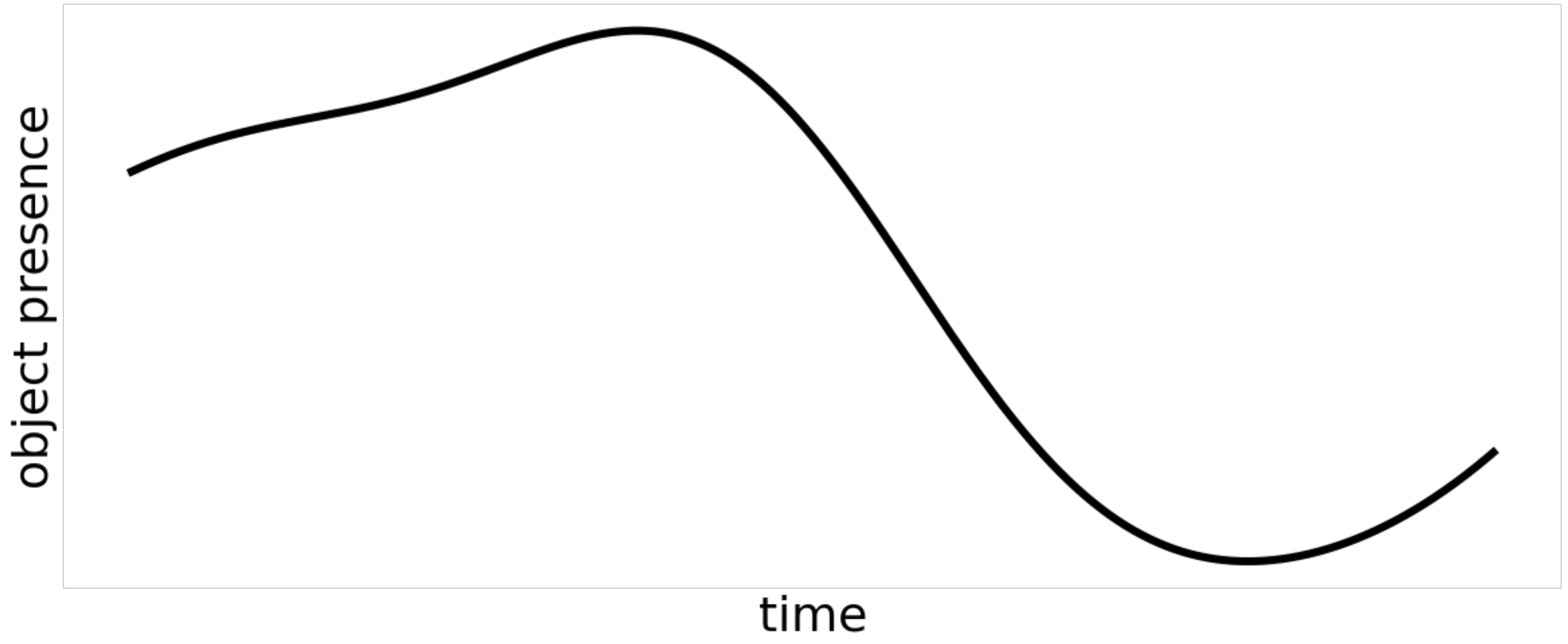


# The slowness principle

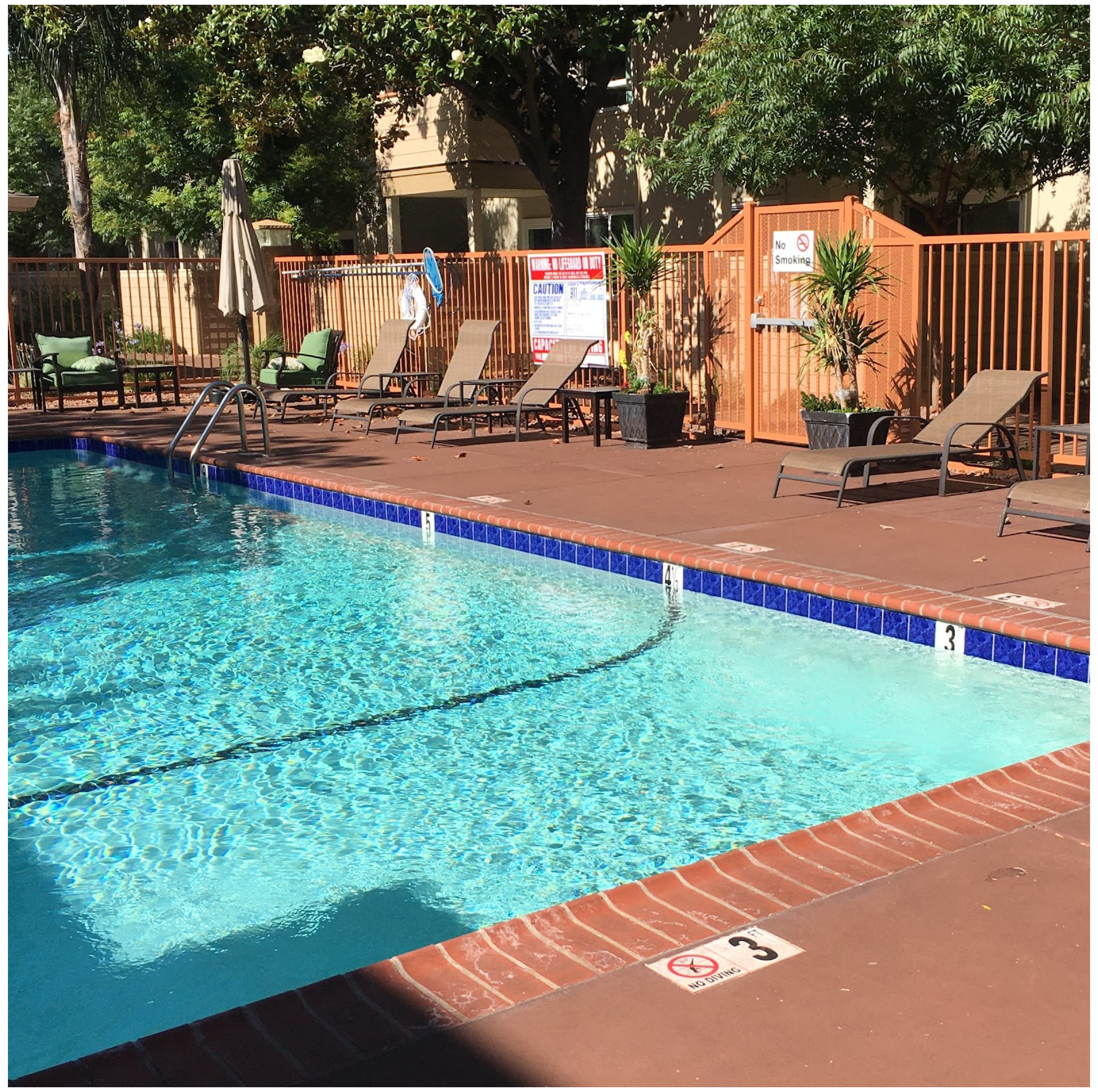
Slow features, persistent across time, tend to coincide with relevant features

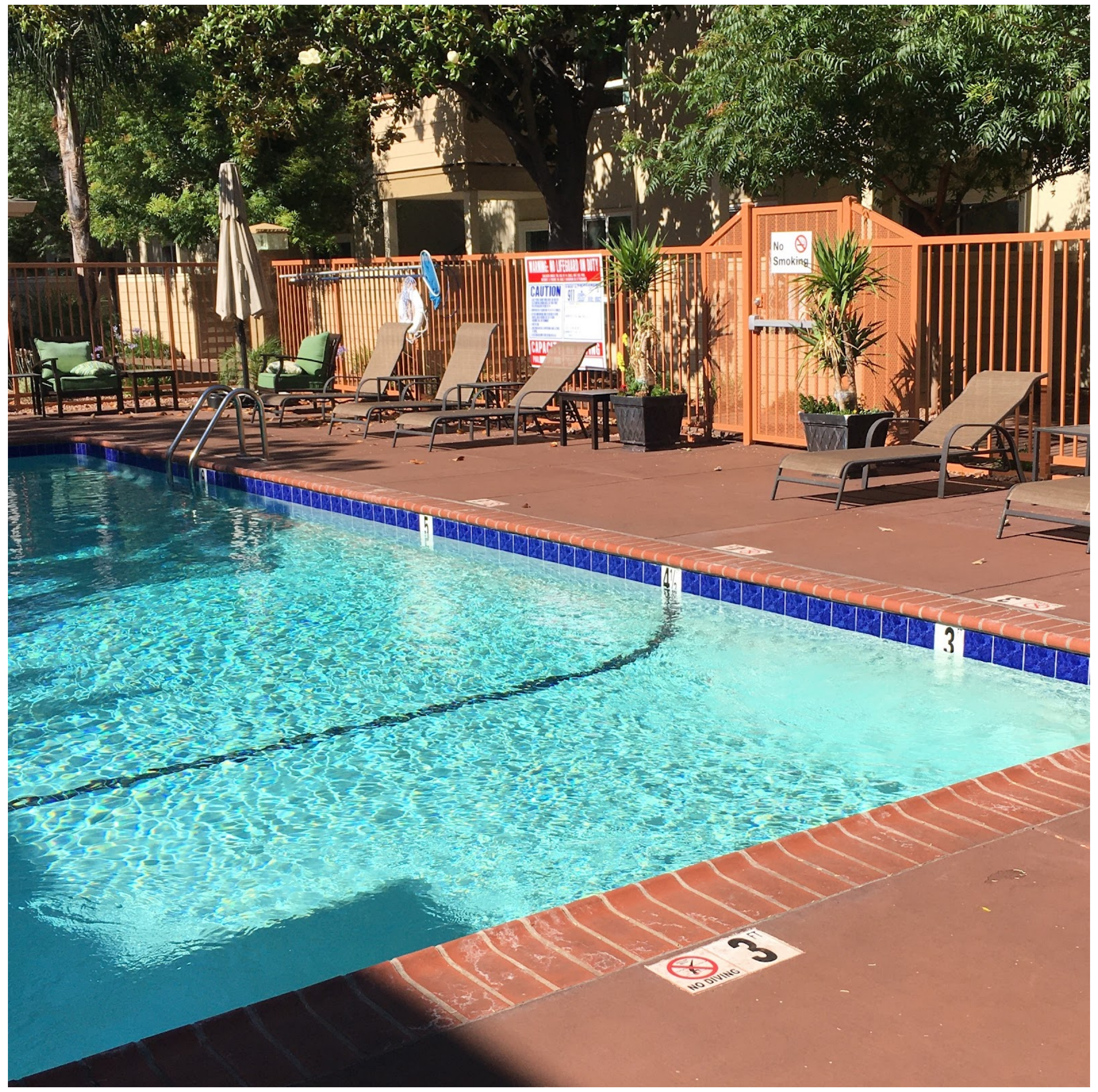
Fast features tend to contain "nuisance" information



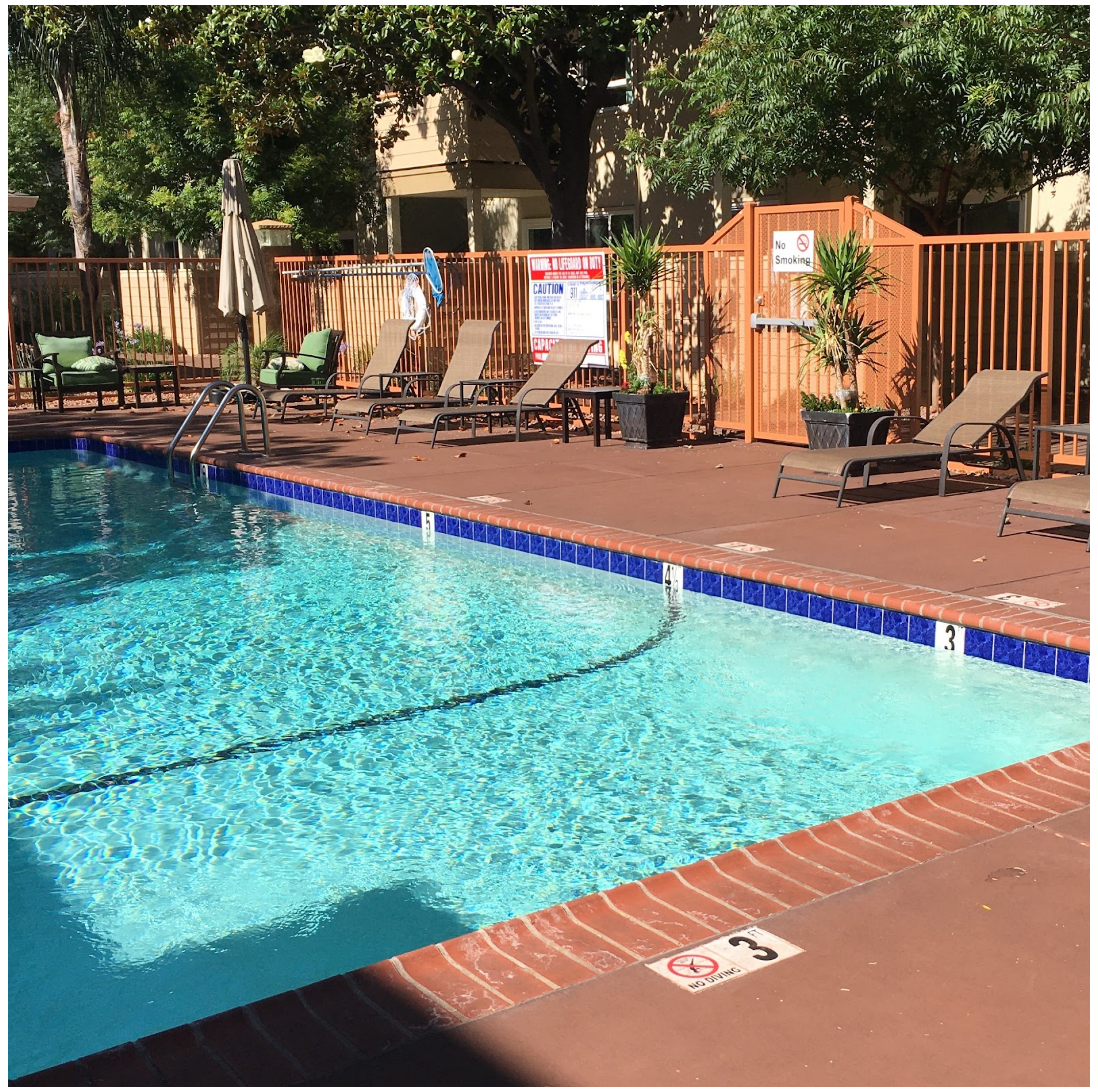



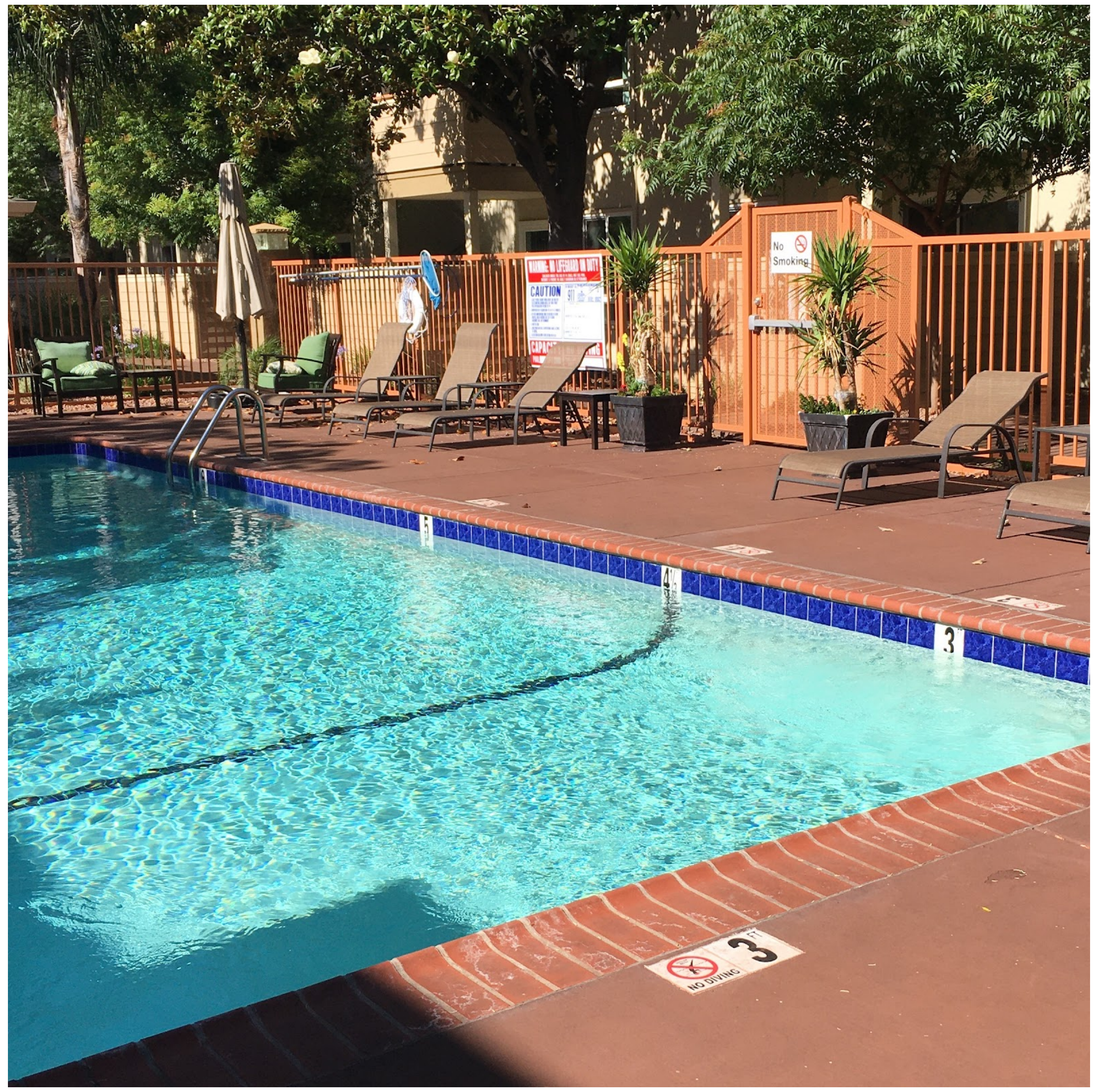
# Slowness for image similarity

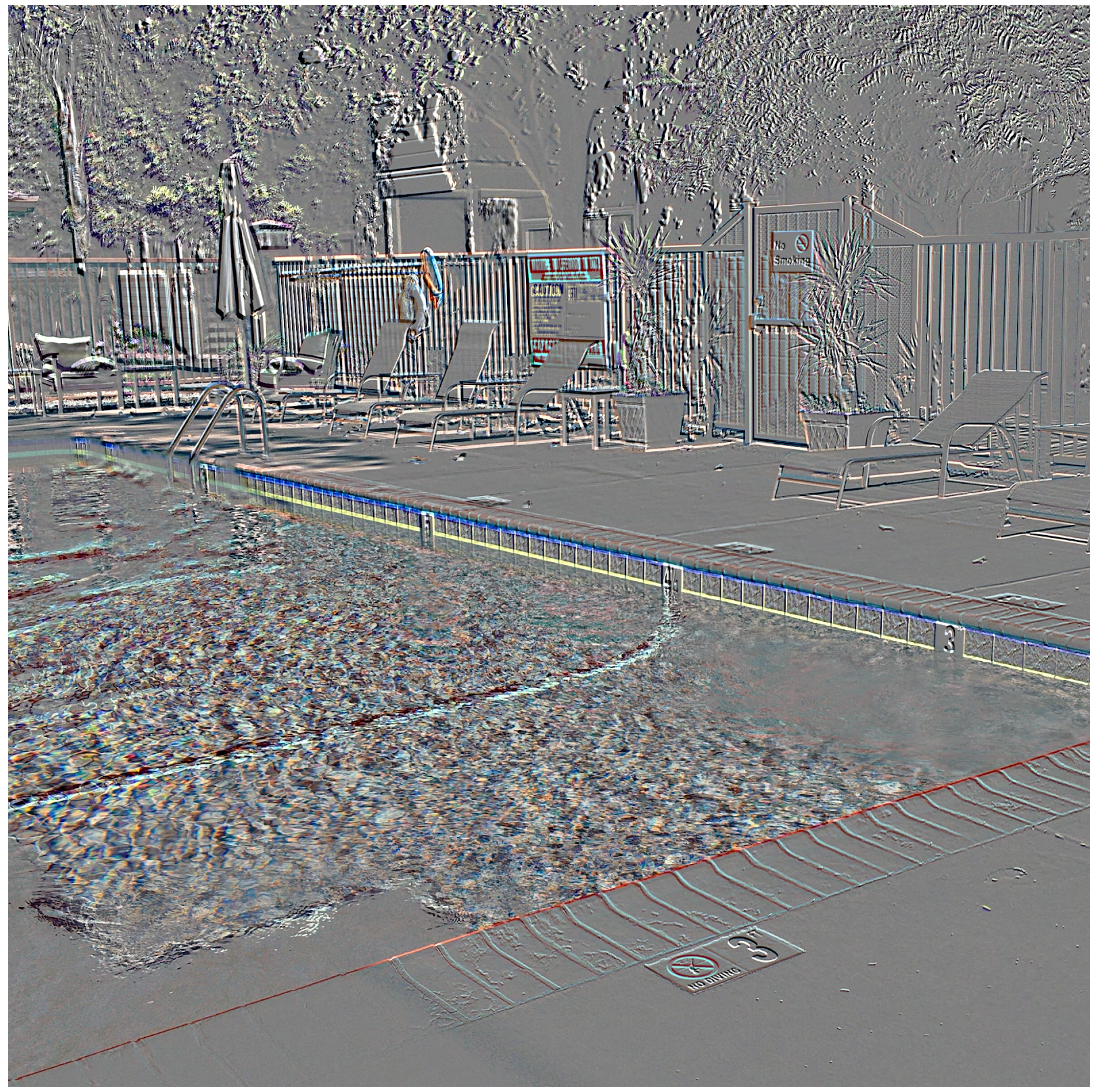




# Slowness for image similarity



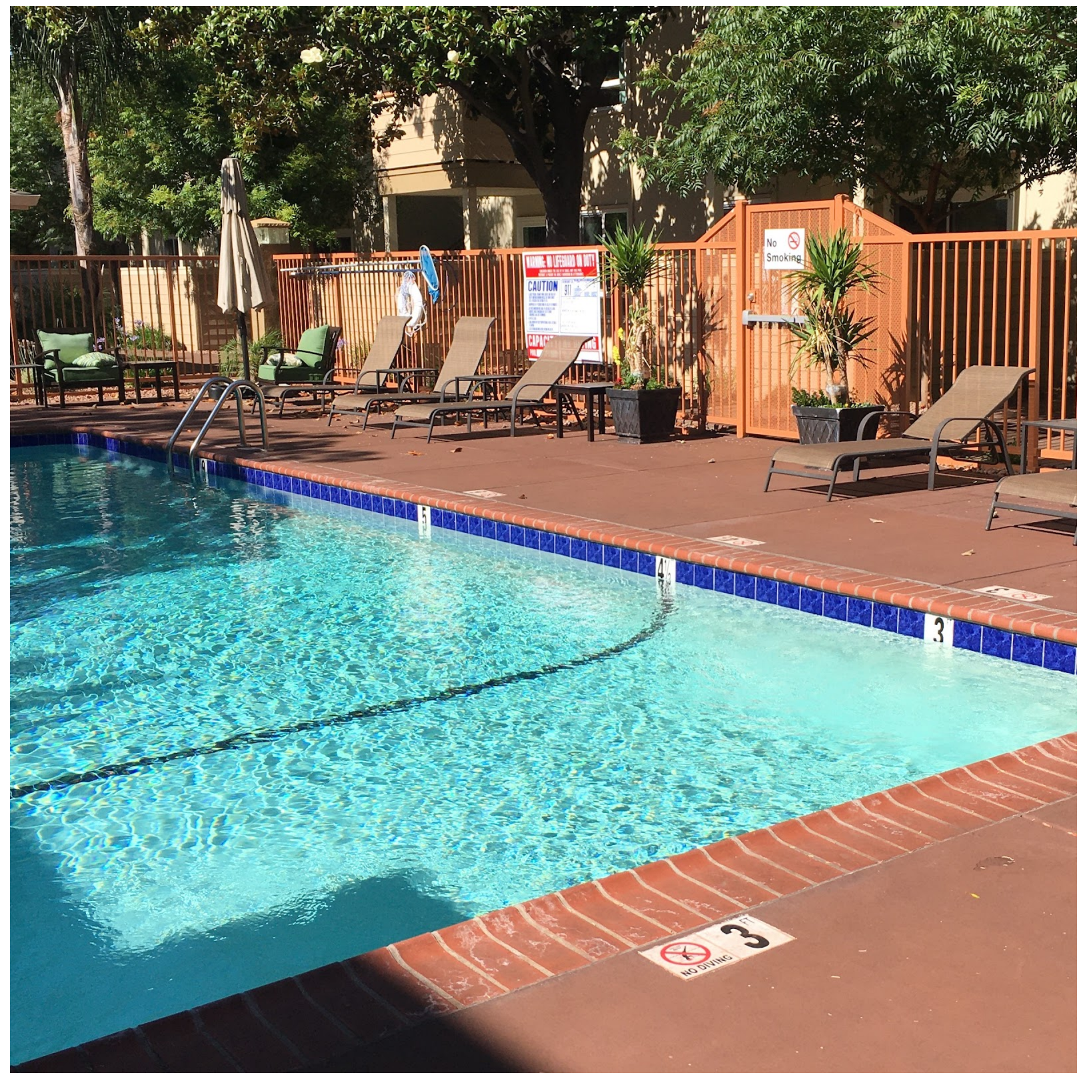




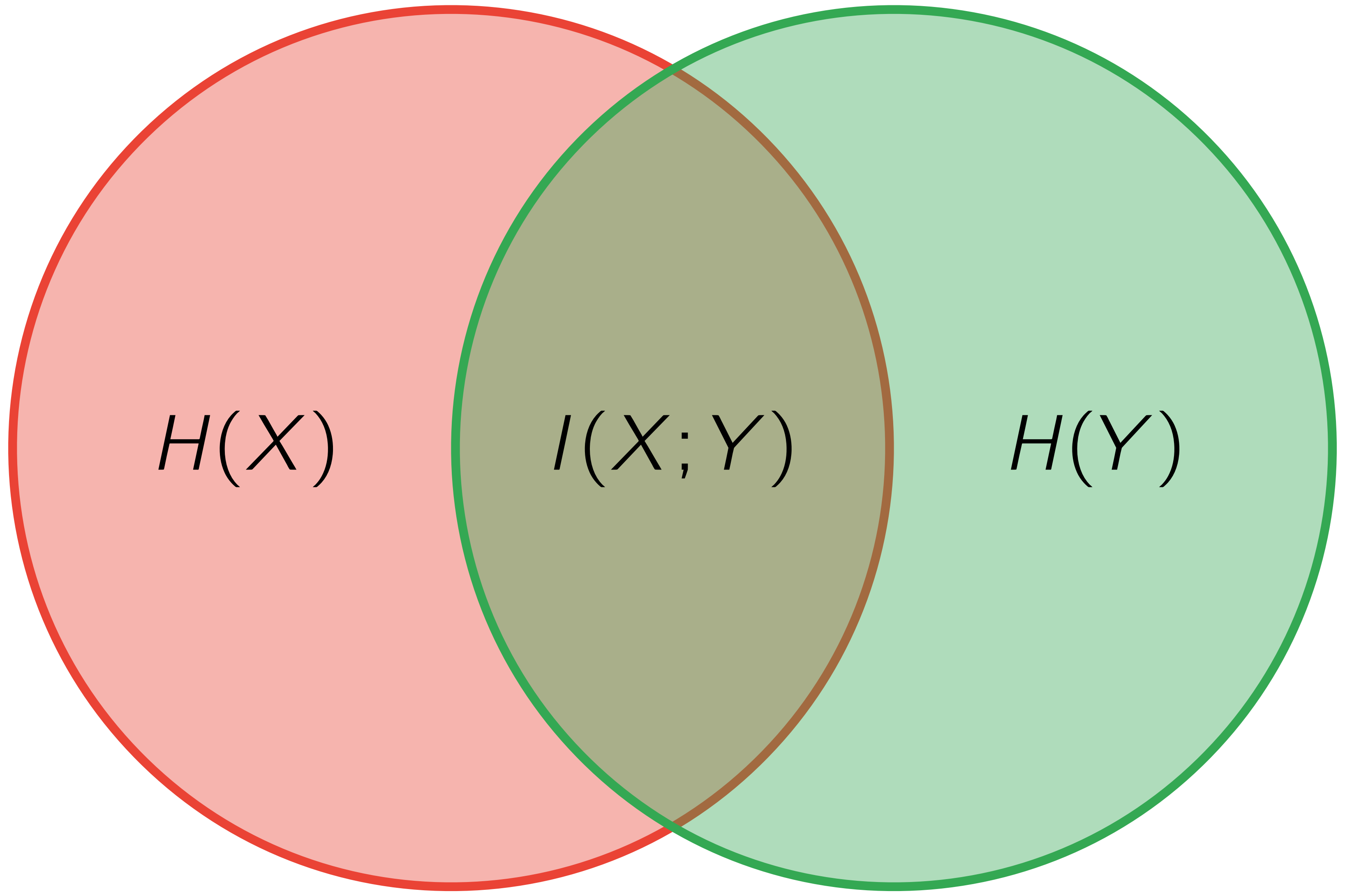
**Pixel values change, but scene composition, texture is constant**

# An implementation of slowness

Let representation $Z$ (in some vector space) capture mutual information $I(X;Y)$ between image $X$ and temporally close image $Y$ (e.g., two frames in a video)



$x$



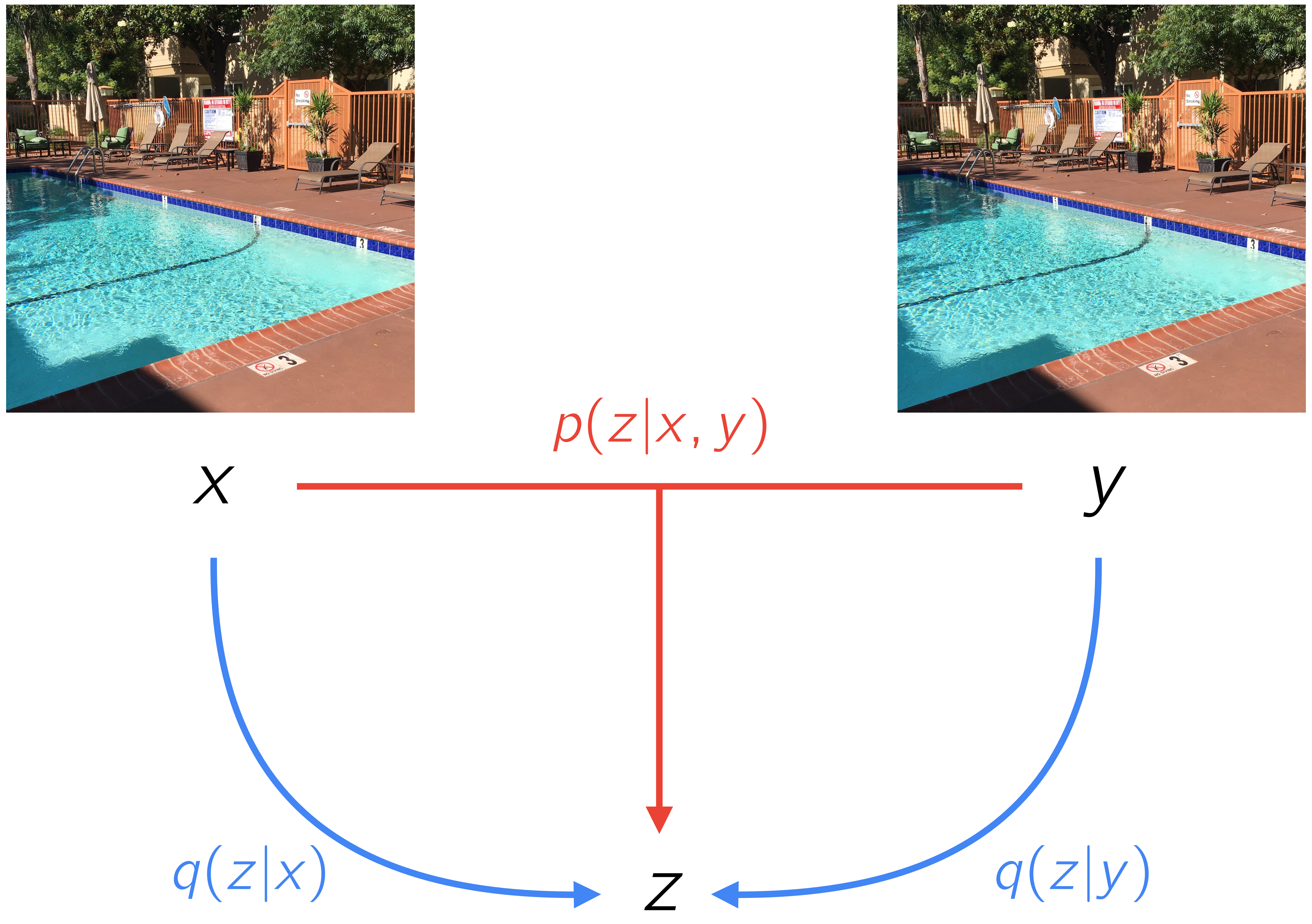

$y$

# Learning objective

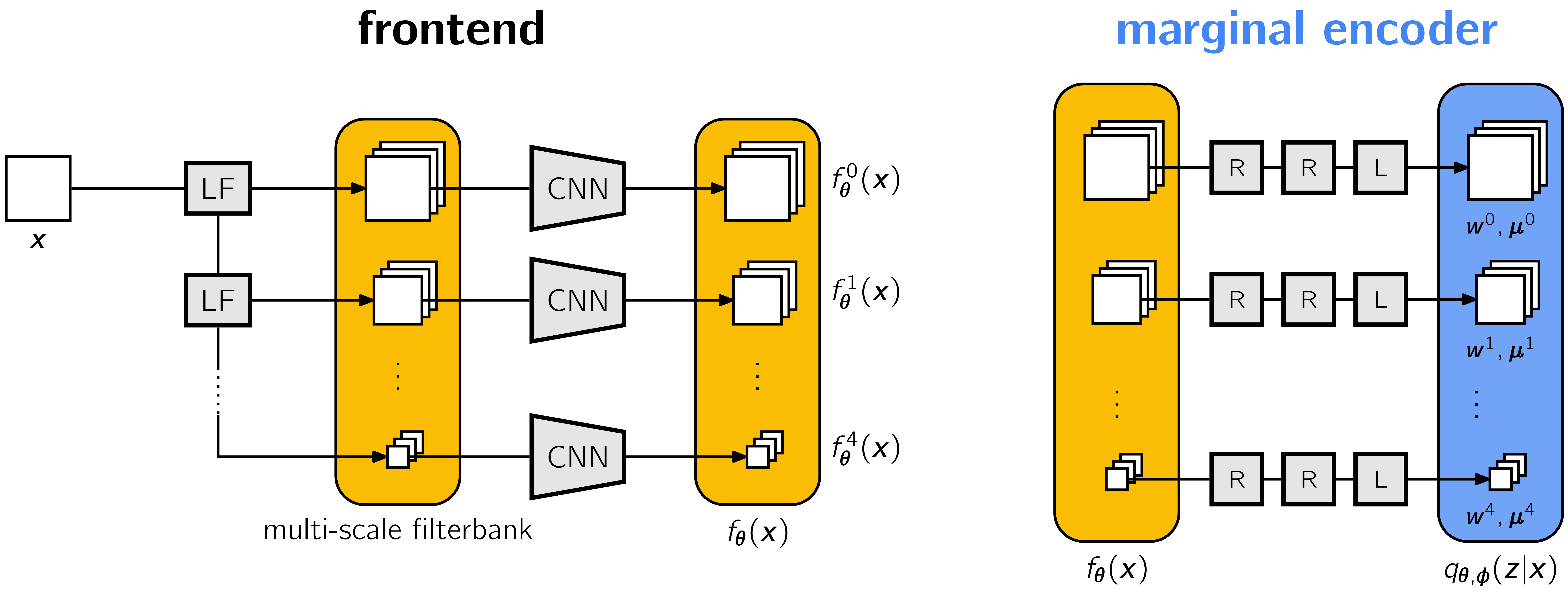
$$\mathbb{E}_{x,y,z} \log \frac{q(z|x)q(z|y)}{\hat{p}(z)p(z|x,y)} \leq I(X;Y;Z)$$

Maximize multivariate mutual information (MMI) between $X$, $Y$, and $Z$ using a stochastic lower bound "IXYZ" (Fischer, 2019)

- Parameterized by two networks:
  $p(z|x,y)$: joint encoder
  $q(z|\cdot)$: marginal encoder
- Contrastive loss, due to empirical/minibatch marginal $\hat{p}(z)$

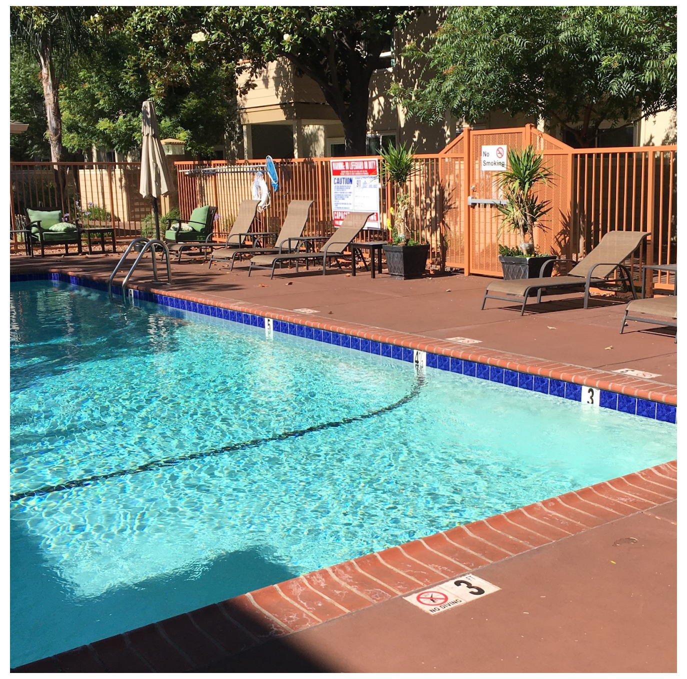


# Architecture

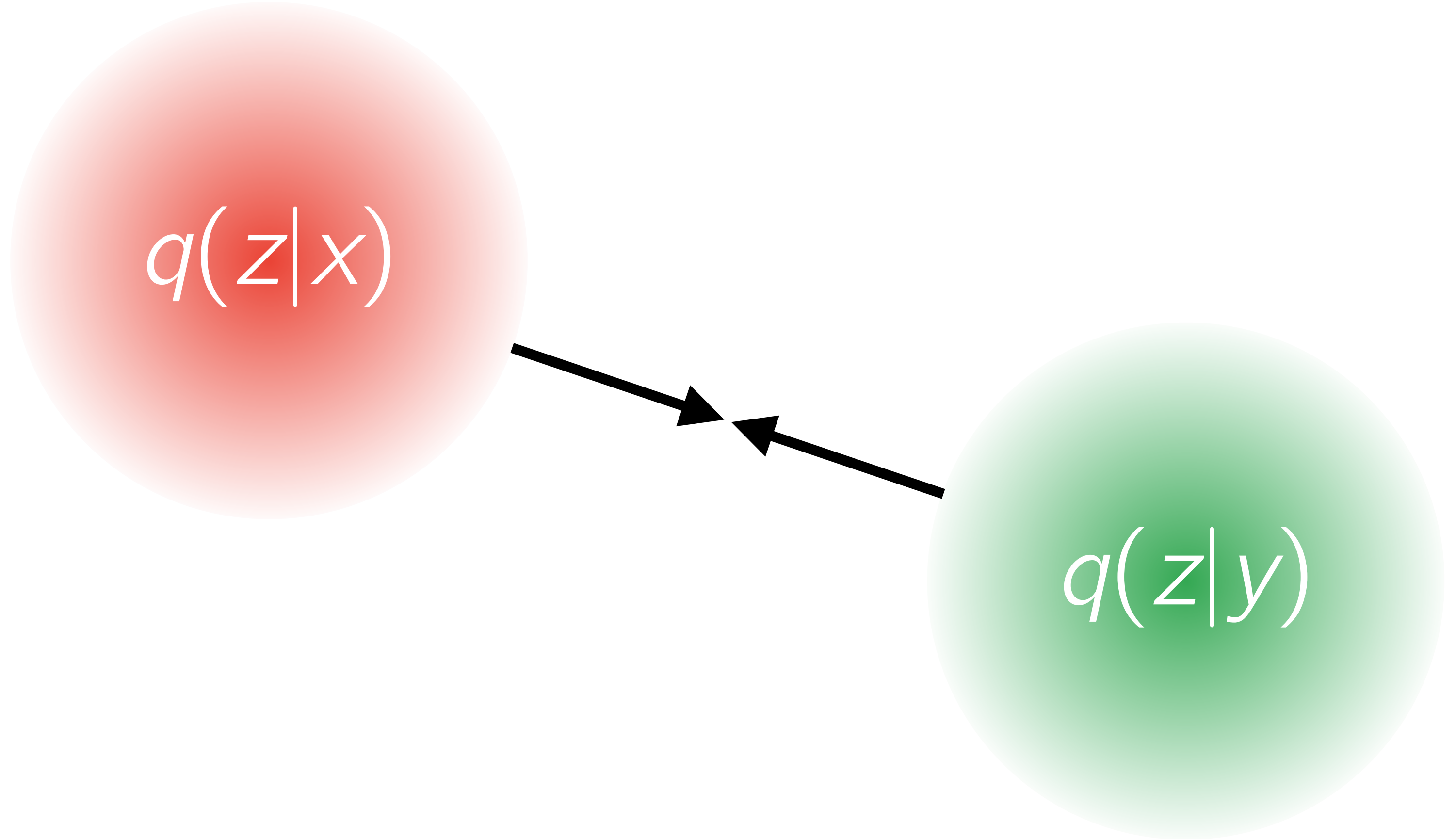


# Contrastive losses, visually explained

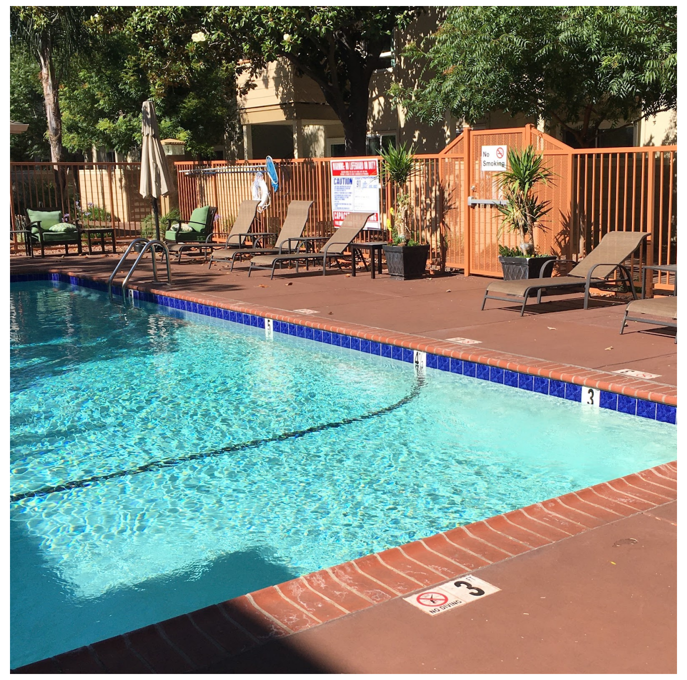
## Positive example

 ,  :

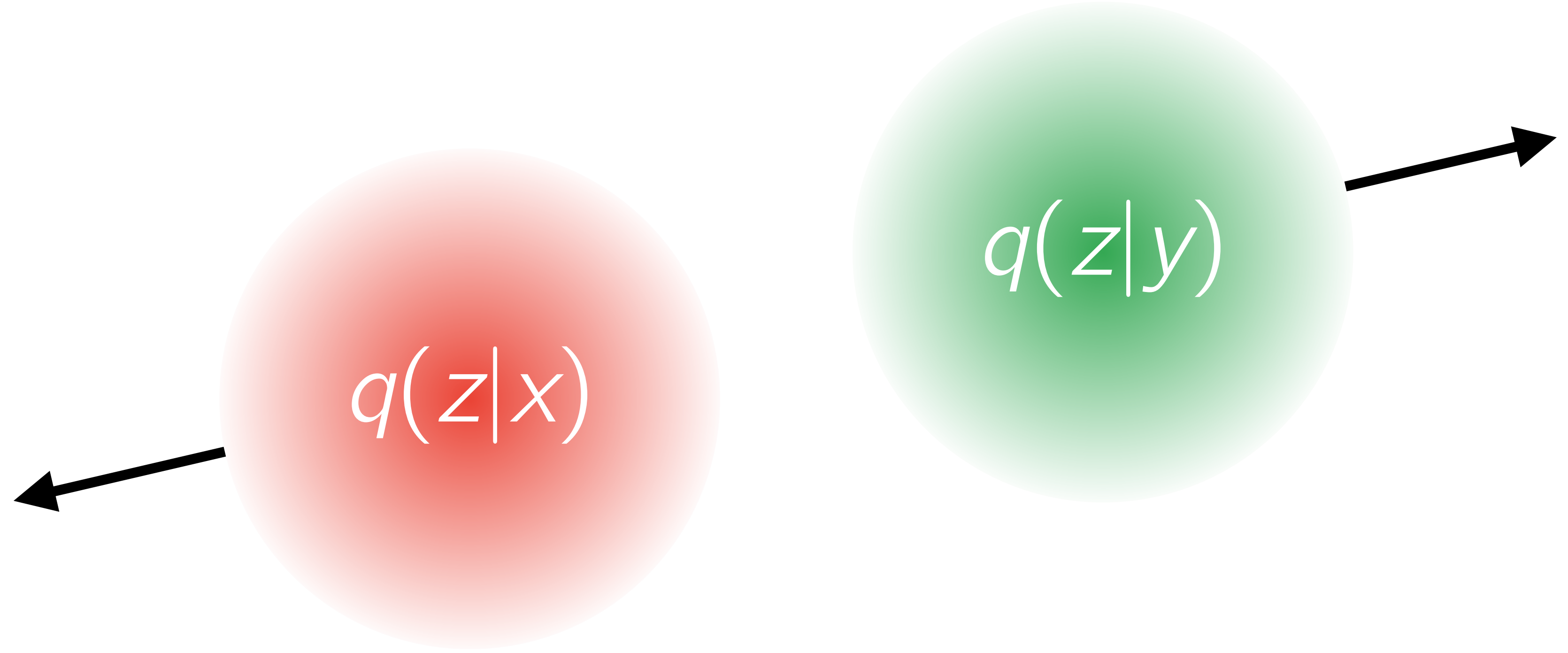
$x$ $y$




## Negative example

 ,  :

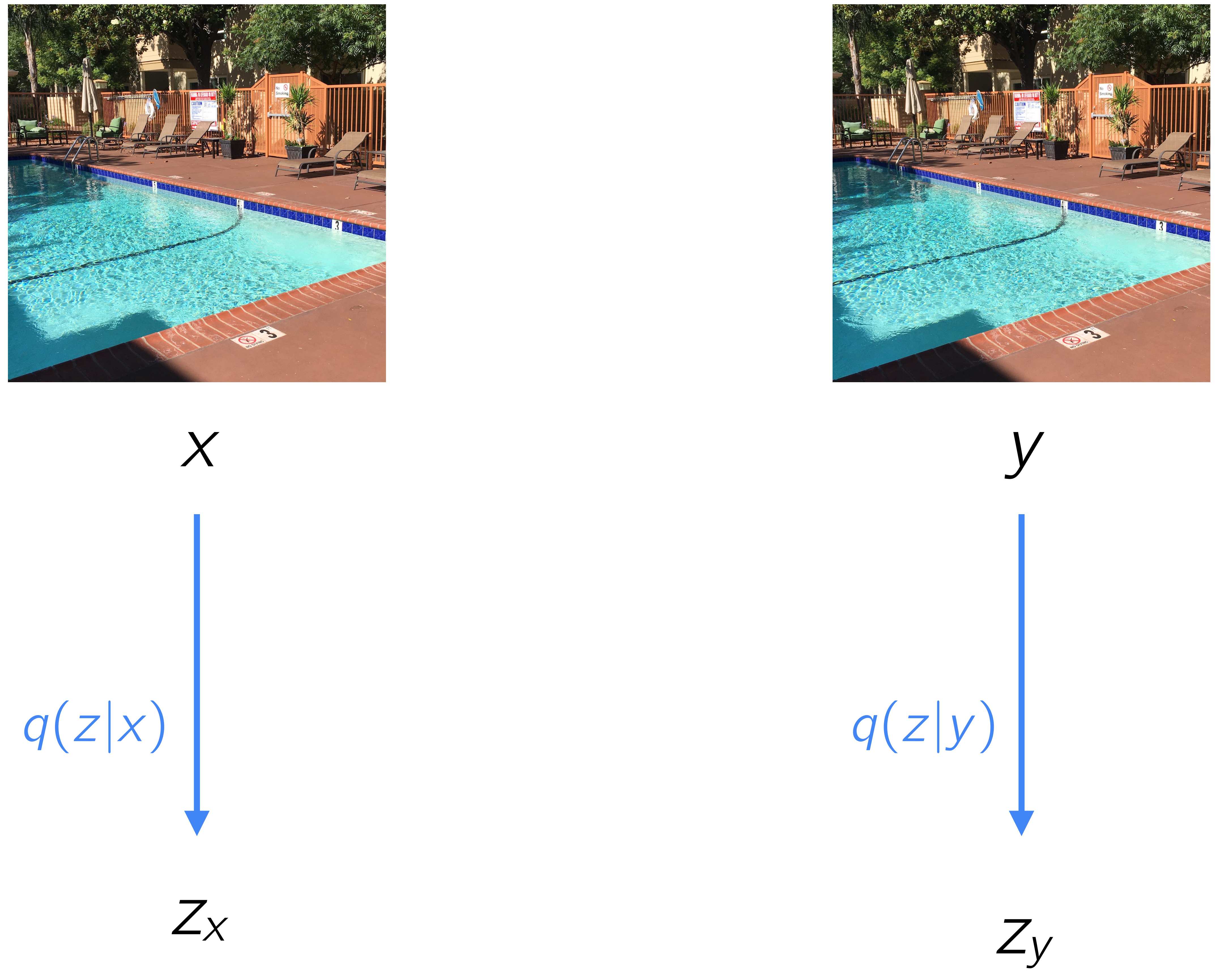
$x$ $y$



# Induced perceptual metric

Symmetrized Kullback–Leibler divergence between representations $\boldsymbol{z}$ of two images, as predicted by marginal encoder $q$ (we can discard joint encoder after training).

Directly use the divergence as a distortion metric, or define realism measure on $\boldsymbol{z}$.



# PIM competitive without using any human ratings

**LPIPS Alex:** pre-trained classifier, no fine tuning

**LPIPS Alex-lin:** fine-tuned for triplet task

**PIM** is significantly better on BAPPS-JND, and competitive on BAPPS-2AFC.

| Metric | BAPPS-2AFC (triplet) | BAPPS-JND |
|---|---|---|
| MS-SSIM | 63.26 | 52.50 |
| NLPD | 63.50 | 50.80 |
| LPIPS Alex | 68.98 | 59.47 |
| LPIPS Alex-lin | **69.53** | 61.50 |
| **PIM (Ours)** | **69.09** | **64.38** |

Agreement with raters (0-100)

# Conclusion

Better perceptual models are a new milestone for image compression.

Crucial for training: **generalization** across types of distortion.

Some of the important ingredients may be:

- Trading off **distortion** and **realism**
- Developing better **perceptual spaces**
- To that end, increasingly modeling **brain behavior** rather than anatomy

# Collaborators


Eirikur Agustsson


Sangnie Bhardwaj


Phil Chou


Ian Fischer


Sung Jin Hwang


Nick Johnston

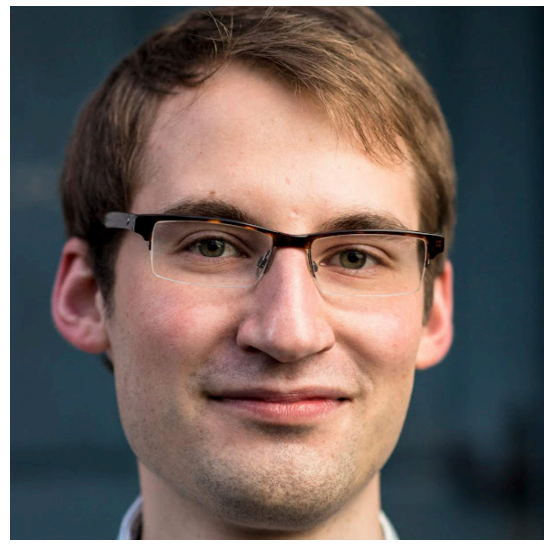

Valero Laparra


Fabian Mentzer


David Minnen


Eero Simoncelli


Saurabh Singh


Lucas Theis


George Toderici

**(et al.)**

# References

**Ahmed et al. (1974):** Discrete cosine transform

**Ballé et al. (2017):** End-to-end optimized image compression

**Agustsson & Theis (2020):** Universally quantized neural compression

**Ballé et al. (2018):** Variational image compression with a scale hyperprior

**Ballé et al. (2021):** Nonlinear transform coding

**Minnen & Singh (2020):** Channel-wise autoregressive entropy models for learned image compression

**Ding et al. (2021):** Comparison of full-reference image quality models for optimization of image proc. sys.

**Zhang et al. (2018):** The unreasonable effectiveness of deep features as a perceptual metric

**Nowozin et al. (2016):** *f*-GAN: training generative neural samplers using variational divergence min.

**Blau & Michaeli (2018):** The perception–distortion tradeoff

**Mentzer et al. (2020):** High-fidelity generative image compression

**MacAdam (1942):** Visual sensitivities to color differences in daylight

**Yamins & DiCarlo (2016):** Using goal-driven deep learning models to understand sensory cortex

**Bhardwaj et al. (2020):** PIM: an unsupervised information-theoretic perceptual quality metric