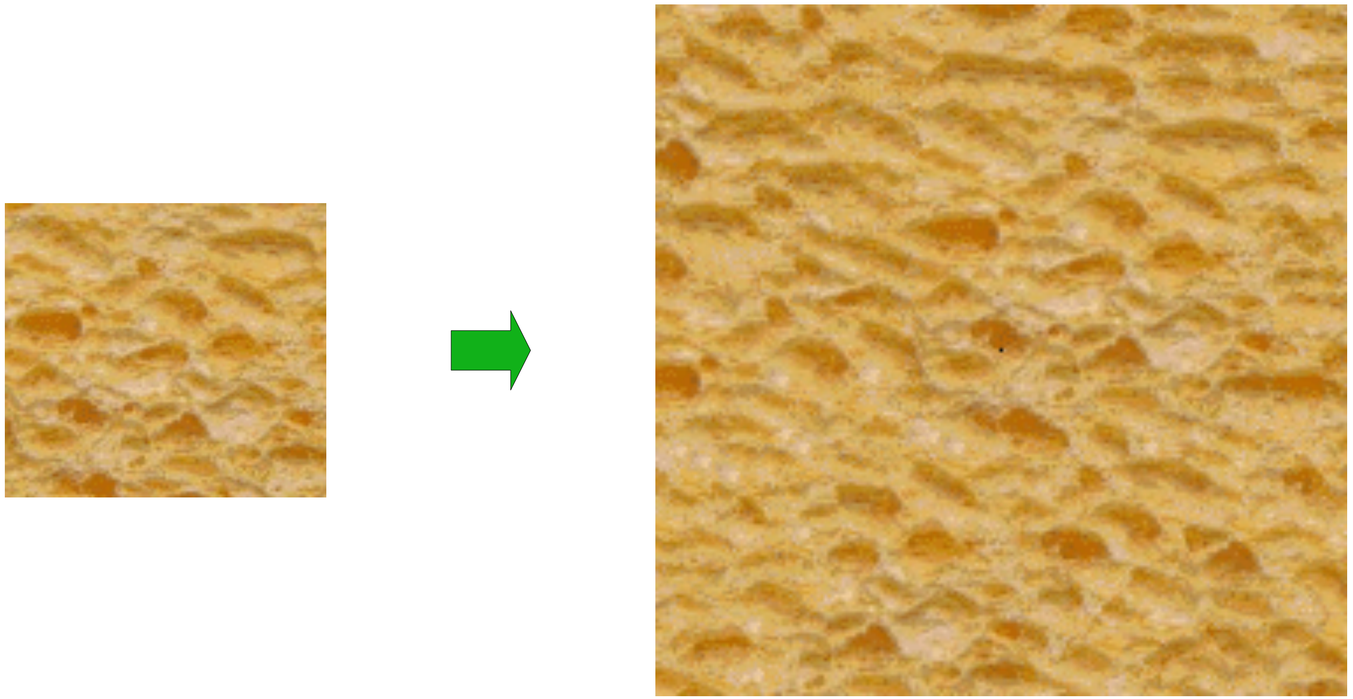
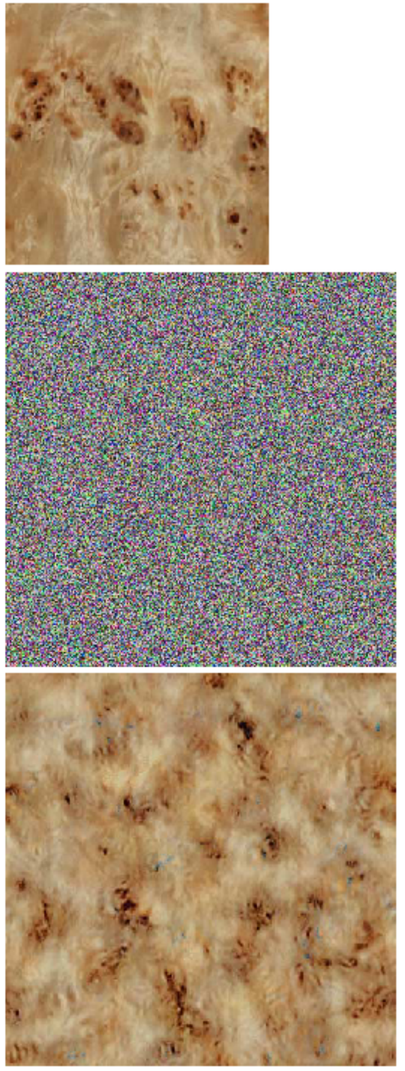
Lecture 13: Image synthesis

Synthesizing textures



Recall: parametric texture synthesis



Start with a noise image as output.

Main loop:

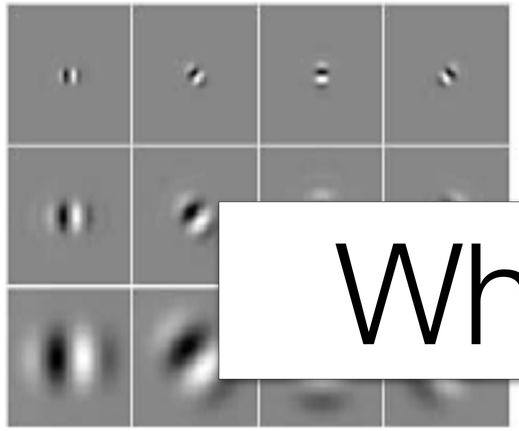
- Match pixel histogram of output image to input
- Decompose input/output images using a Steerable Pyramid
- Match subband histograms of input and output pyramids
- Reconstruct input and output images (collapse the pyramids)

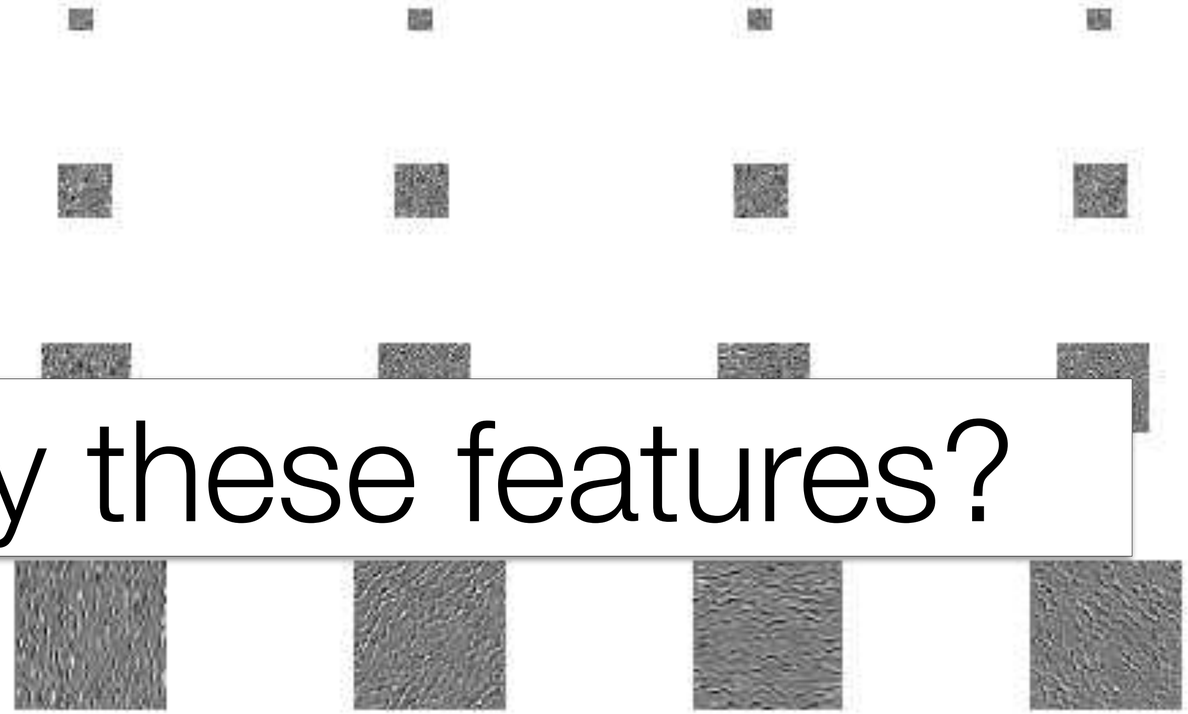
Heeger, Bergen, Pyramid-based texture analysis/synthesis, SIGGRAPH 1995

Source: A. Efros

Recall: steerable filter features

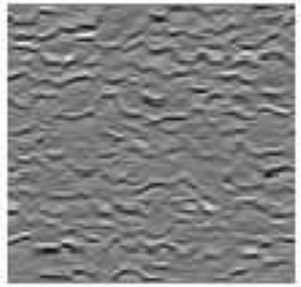
Filter bank

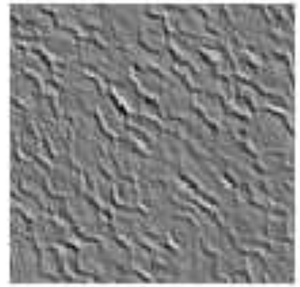




Why these features?

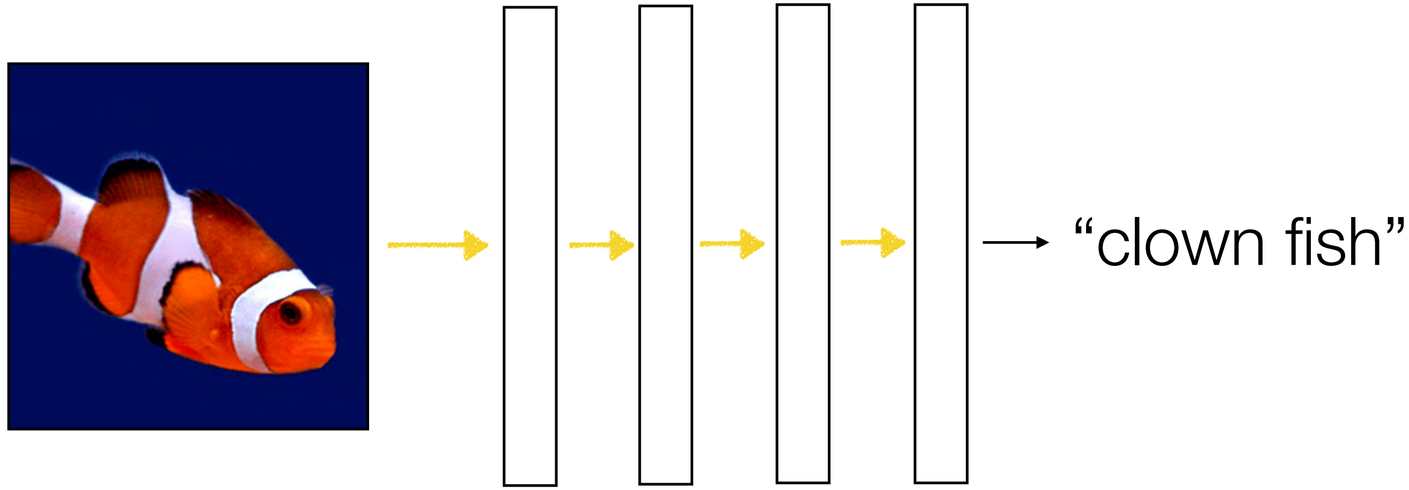
Input image



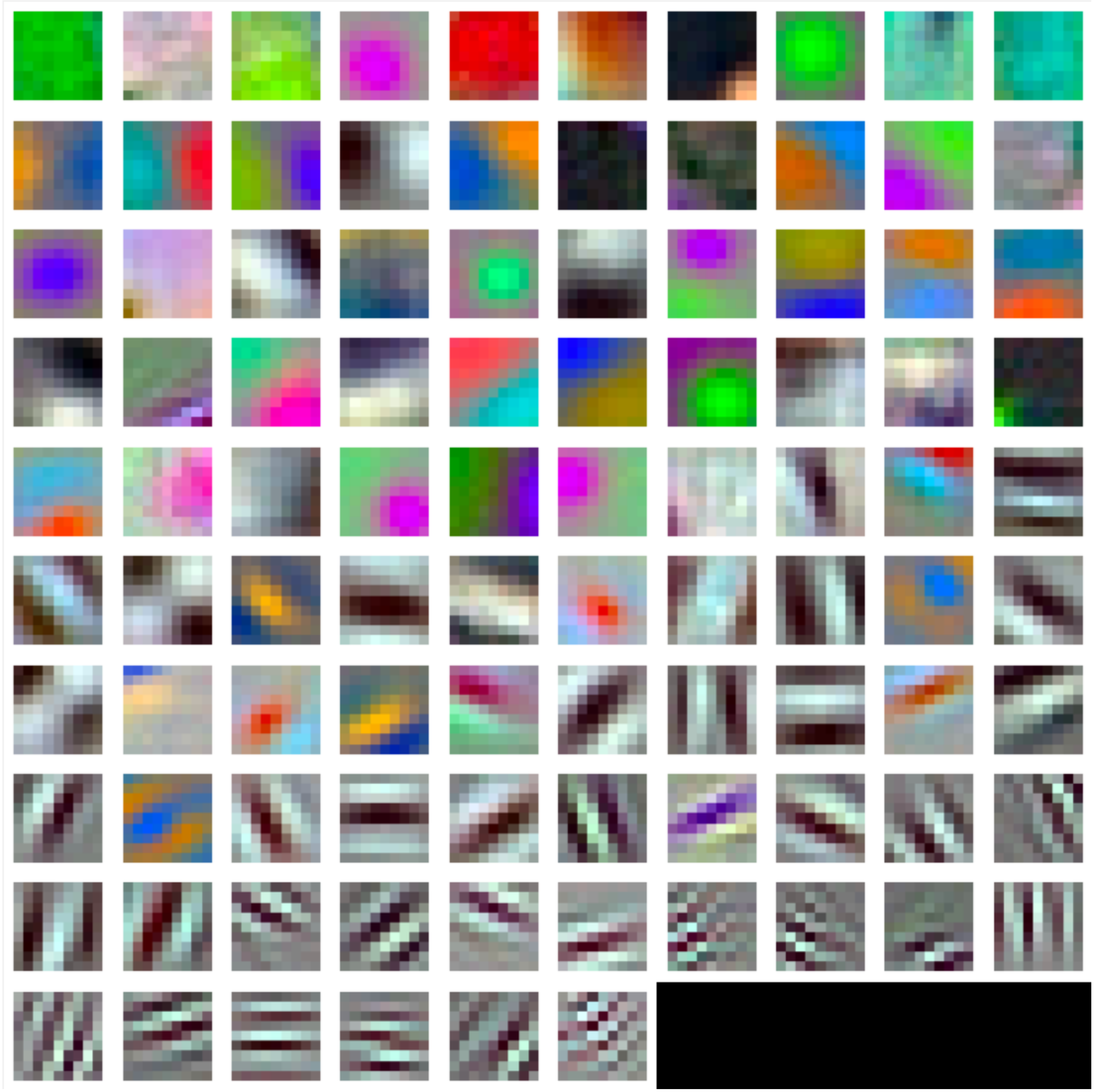


Source: A. Efros

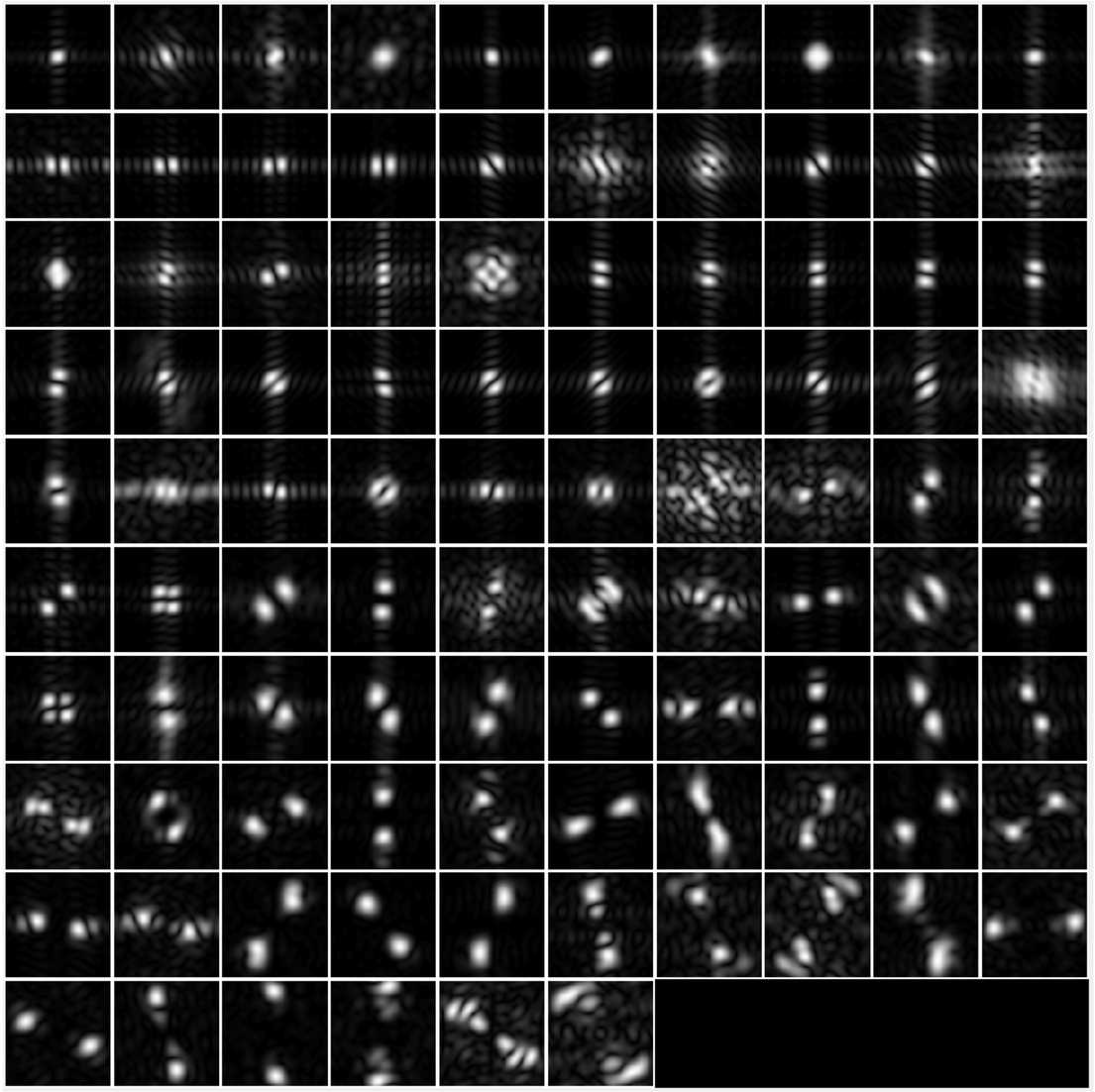
Use CNN features instead!



Recall: AlexNet units

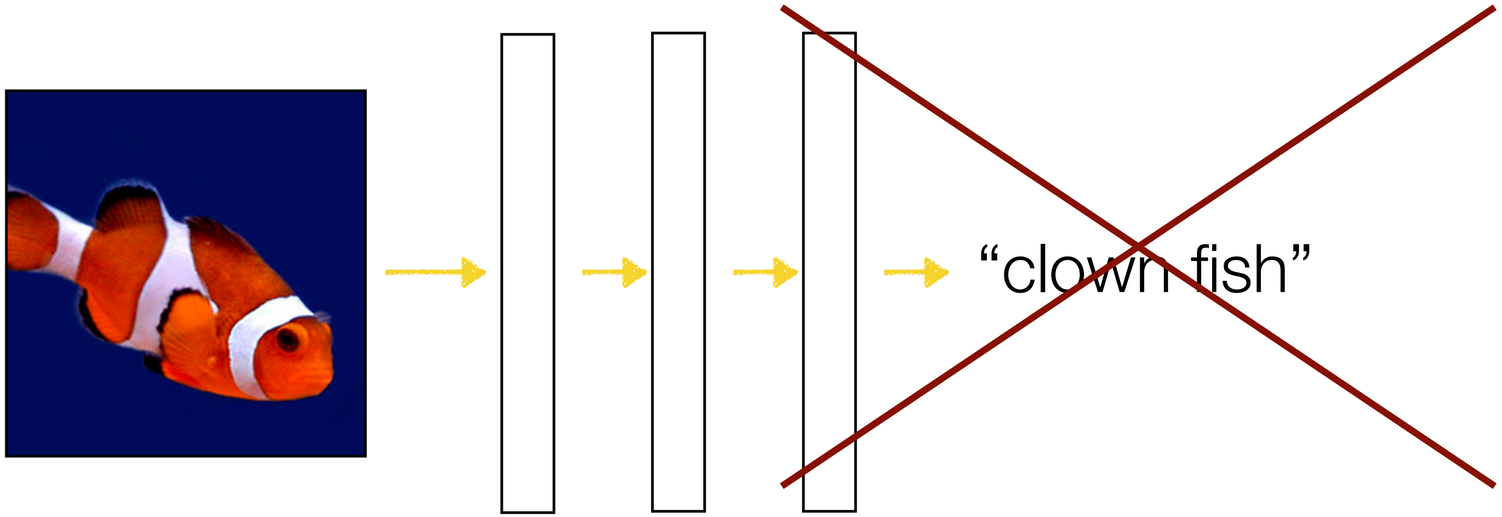


96 Units in conv1

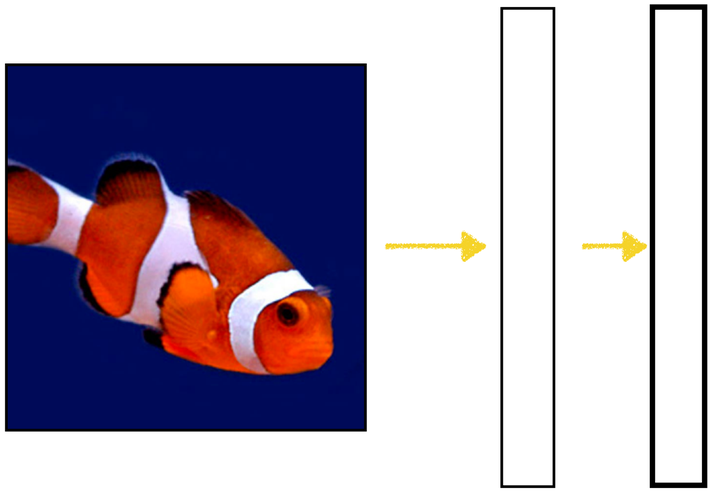


Source: Isola, Torralba, Freeman

Extracting neural net features

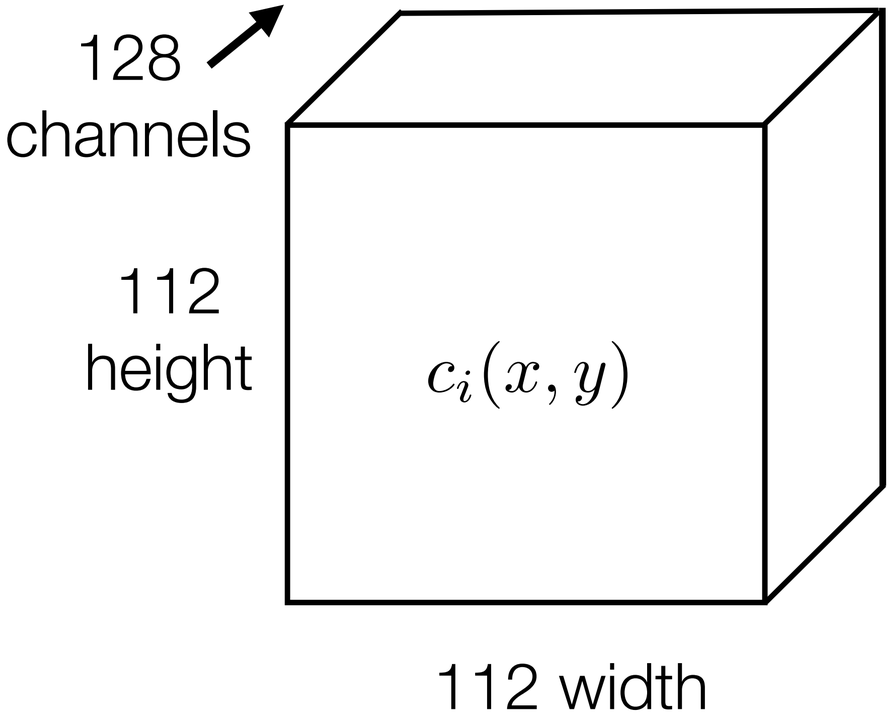


Extracting neural net features



E.g. conv2 feature activations

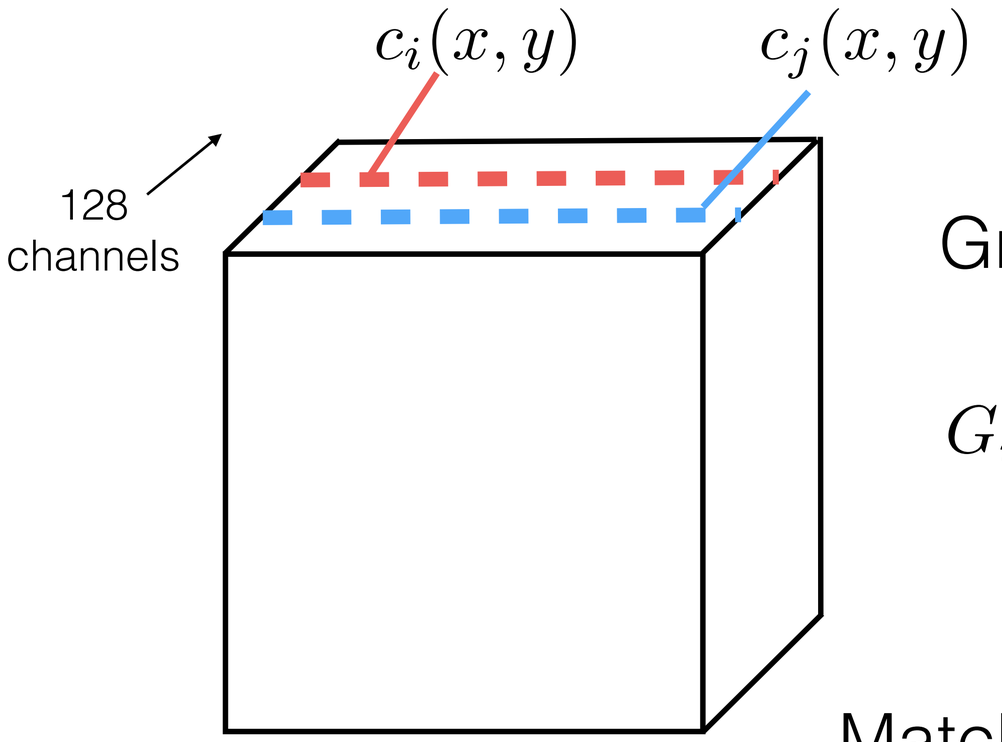
Extracting neural net features



conv2 feature activations

(112 x 112) x 128 for conv2 of VGG19

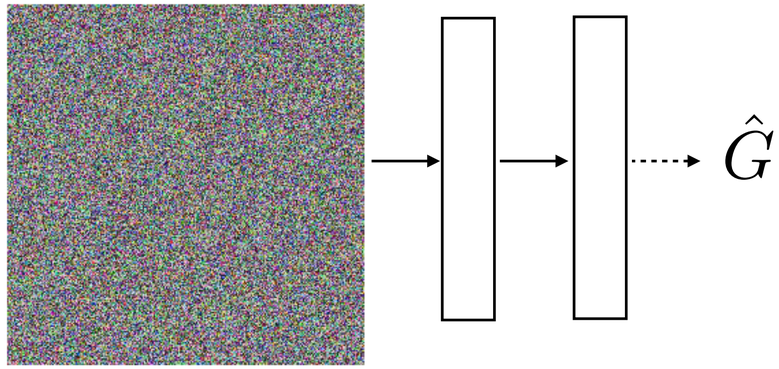
Texture synthesis



Gram (≈ covariance) matrix:

$$G_{ij} = \sum_{x=1}^{w} \sum_{y=1}^{h} c_i(x, y) c_j(x, y)$$

[Gatys et al. 2016]



Match target image stats! Minimize:

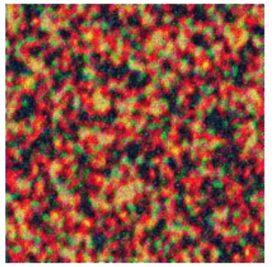
$$\sum_{i,j} (G_{ij} - \hat{G}_{ij})^2$$

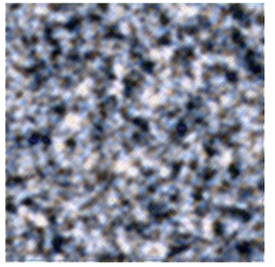
- Use all layers.
- Minimize with gradient descent!

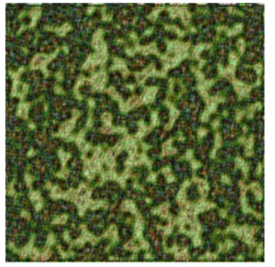
High → Low

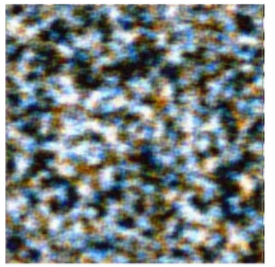
original

conv1_1





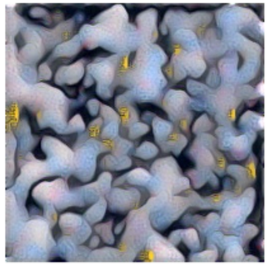


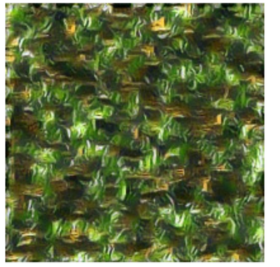


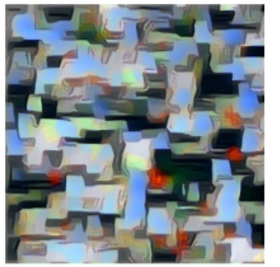
High → Low

original

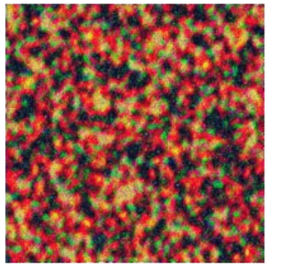
pool1

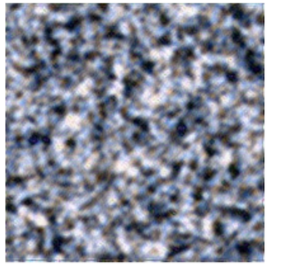


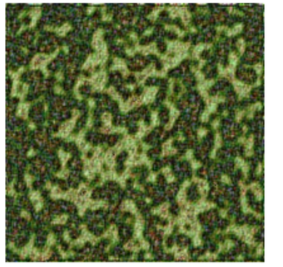


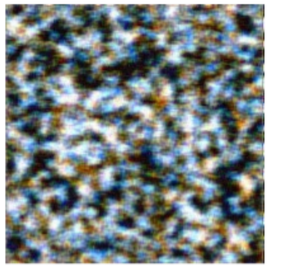


conv1_1

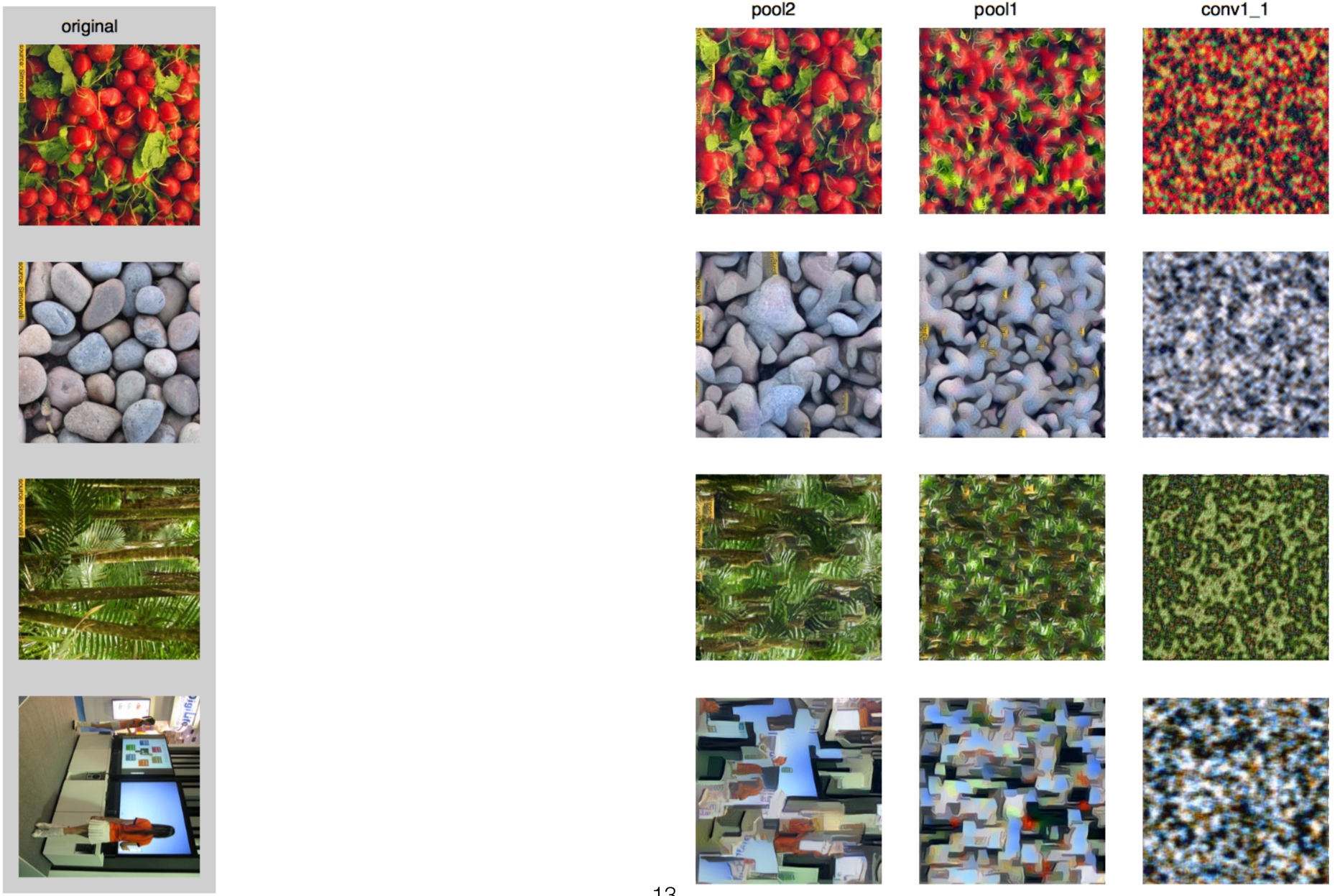




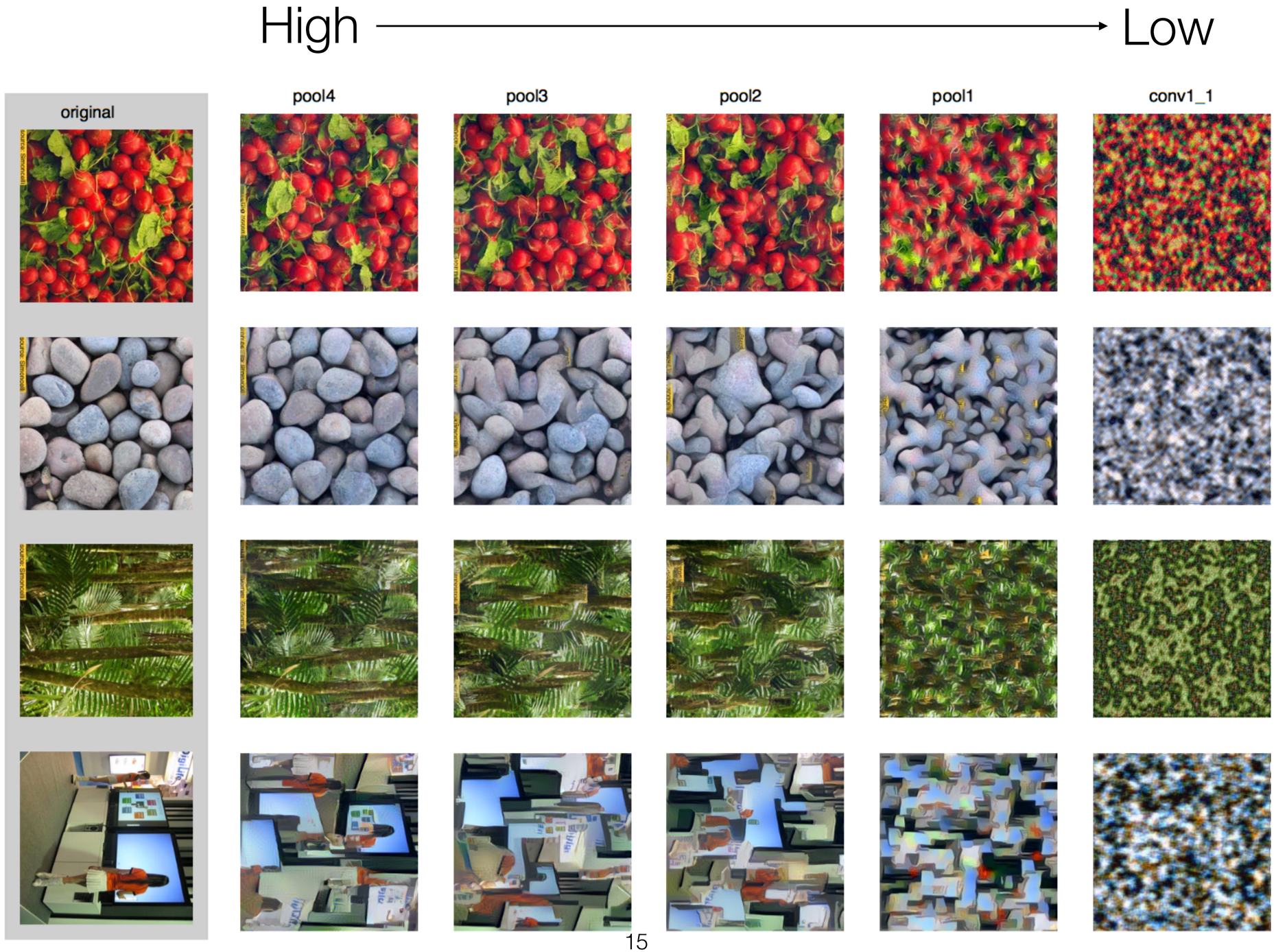




High → Low



High → Low

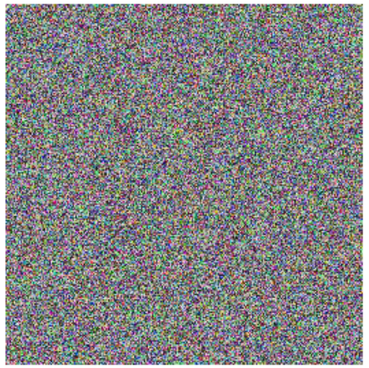

High
Low
original
pool4
pool3
pool2
pool1
conv1_1

Texture captures artistic style

Can we transfer the style of a painting to a photo?

[Gatys et al. 2016]

Match the **style** of the painting.



Synthesized image

$$\hat{G}$$

$$\sum_{i,j}(\hat{G}_{ij} - G_{ij})^2$$

"perceptual loss"

… and the **content** of the photo.

$$\sum_{i}\sum_{x,y}(c_i(x,y) - \hat{c}_i(x,y))^2$$

$c_i(x,y)$

$\hat{c}_i(x,y)$

See [Gatys et al. 2016]

London during the day.

New York at night.

Image classification

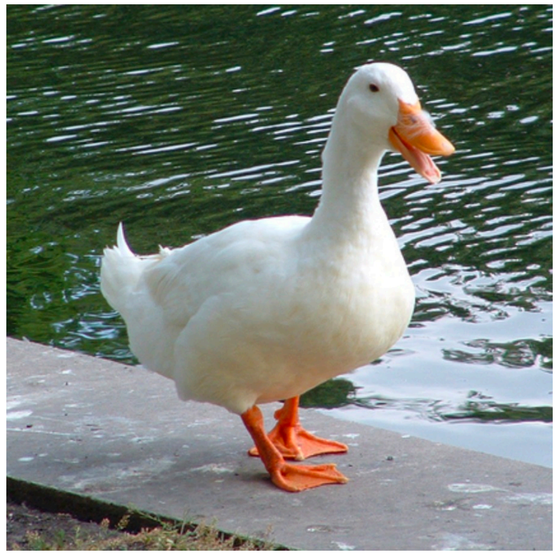


image **x**

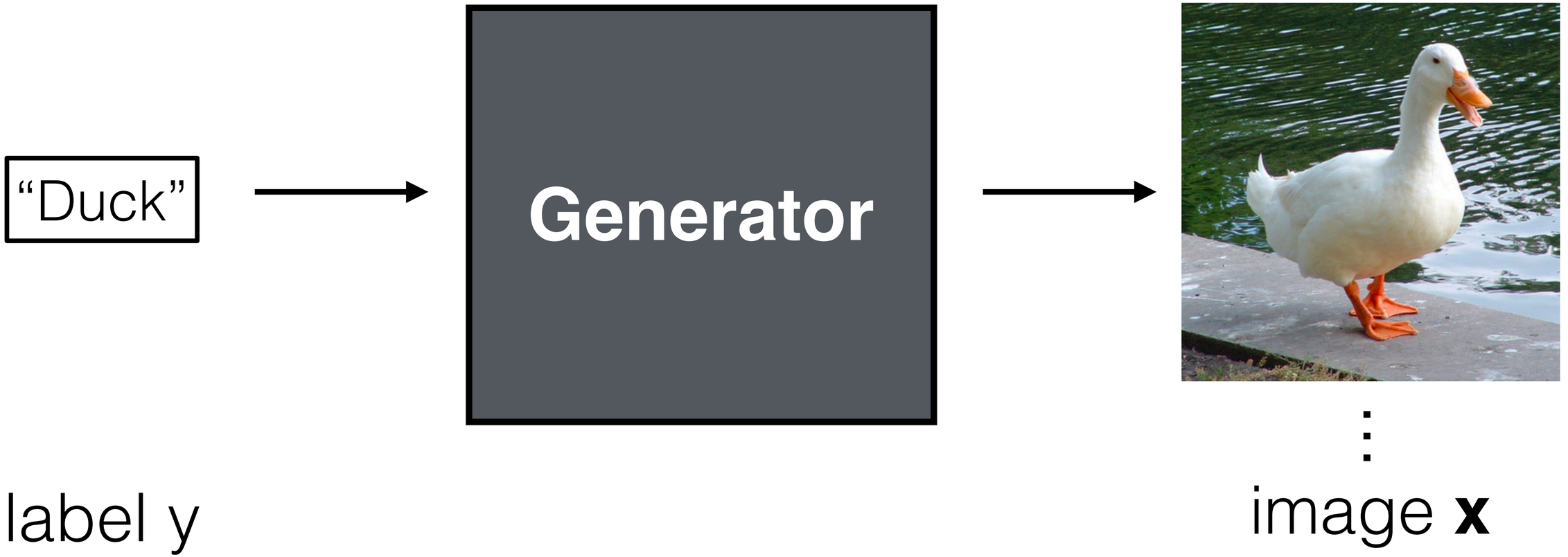
Classifier

"Duck"

label y

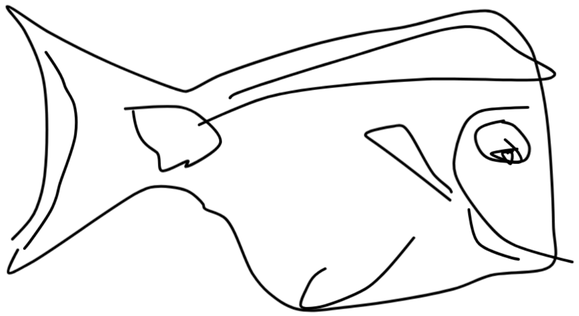
Source: Isola, Freeman, Torralba

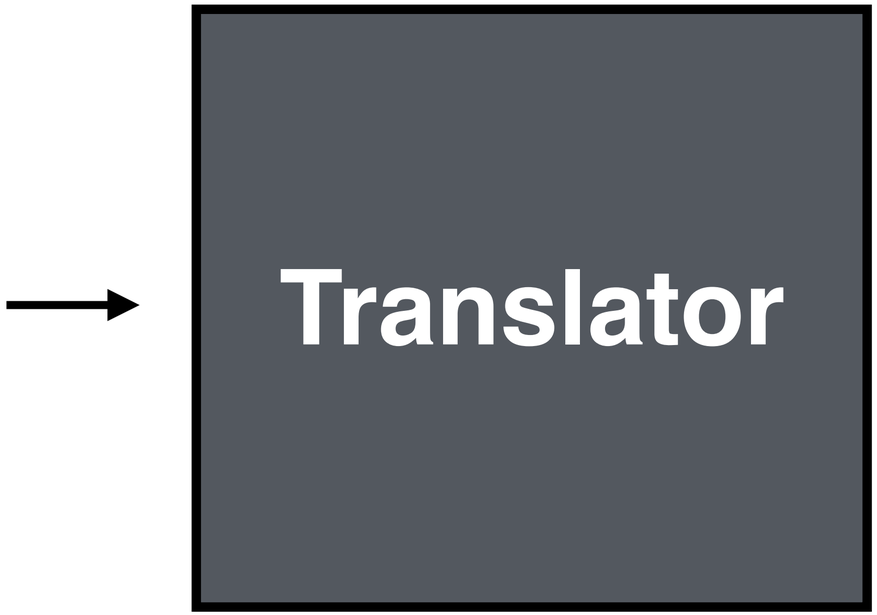
Image synthesis



Source: Isola, Freeman, Torralba

Image translation



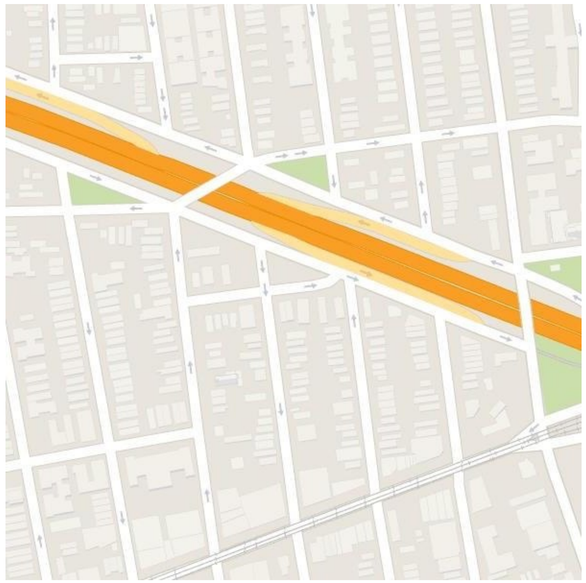


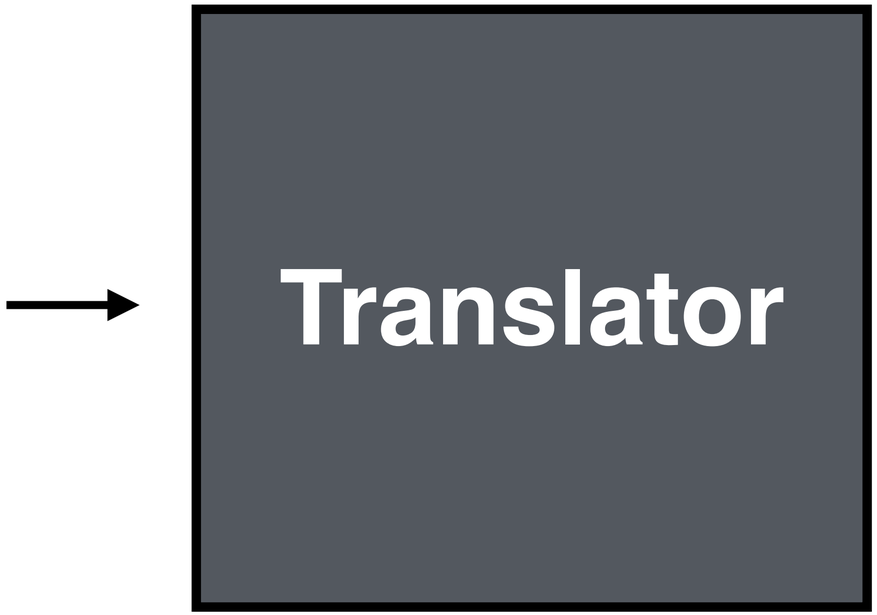
User sketch

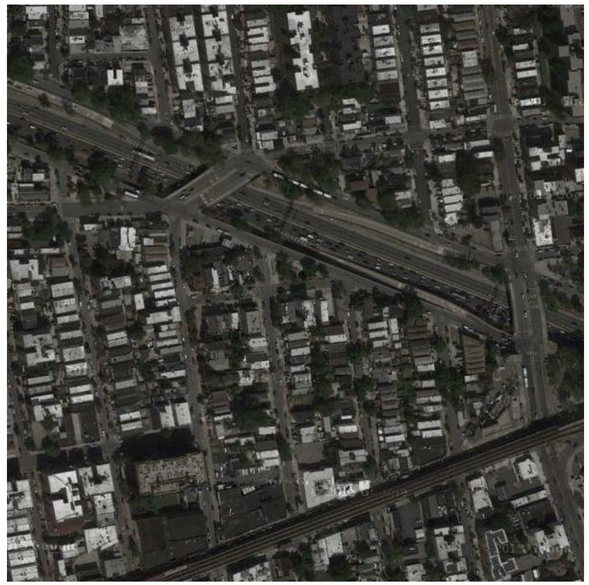
Photo

Source: Isola, Freeman, Torralba

Image translation







Google Map

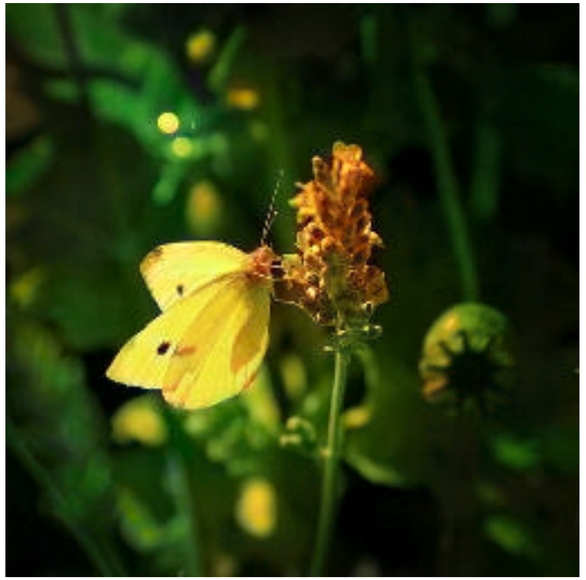
Satellite photo

Source: Isola, Freeman, Torralba

Image translation

BW image

Translator



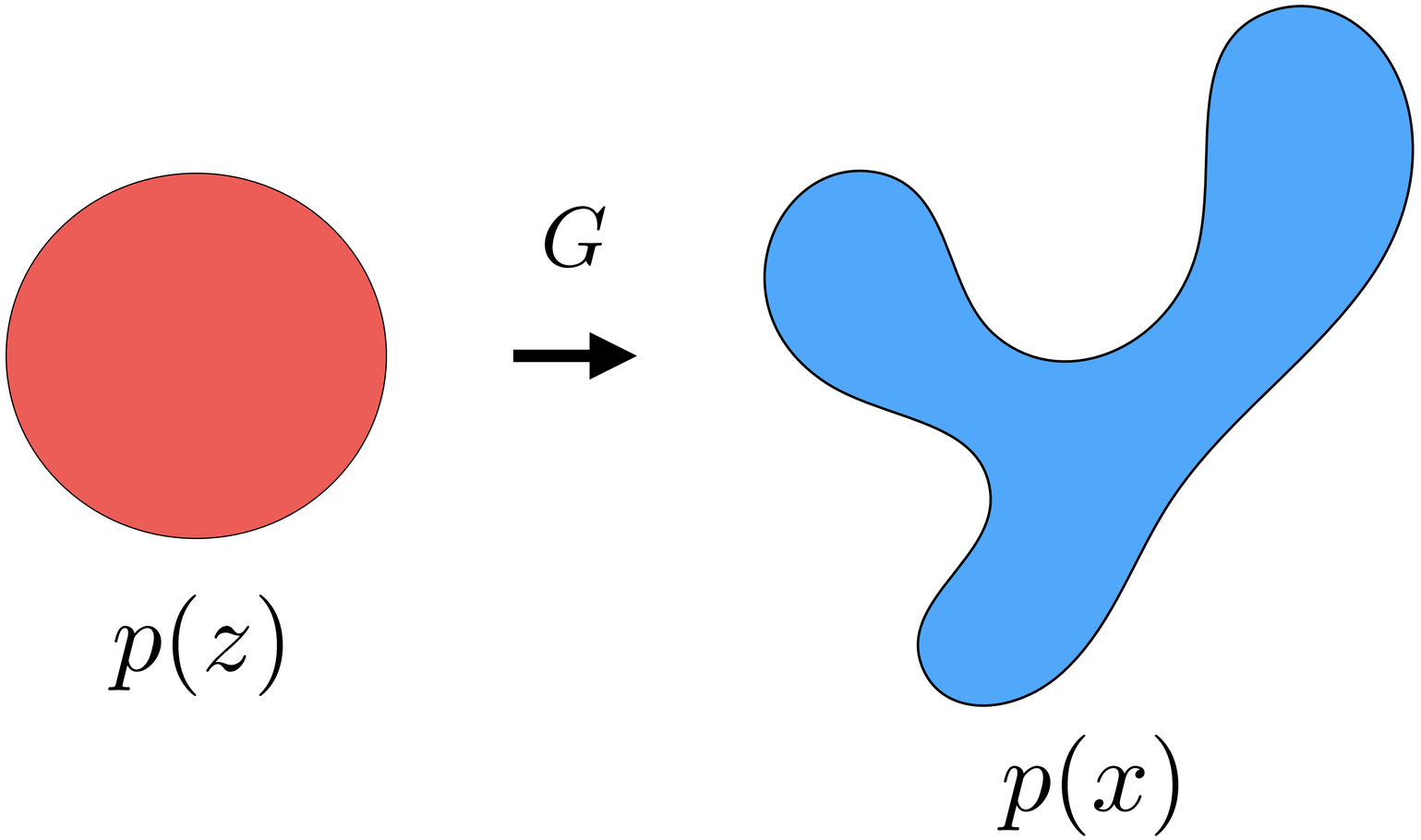
Color image

Source: Isola, Freeman, Torralba

Deep generative models as distribution transformers

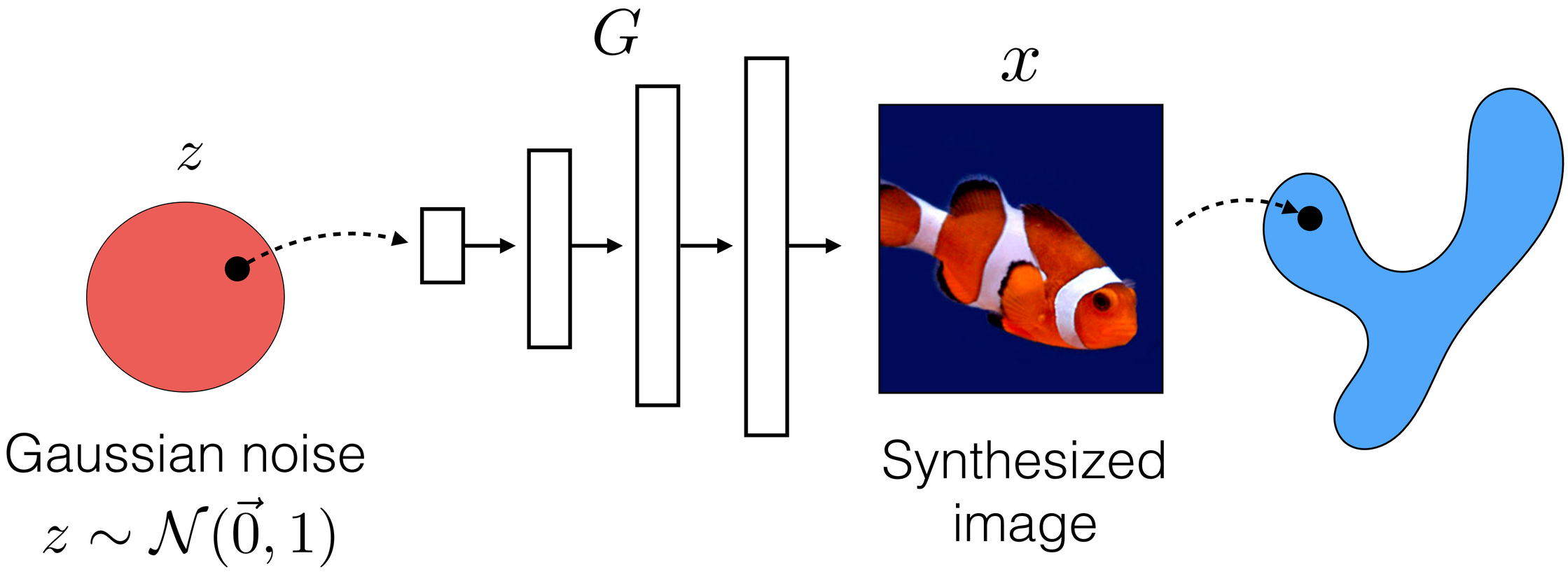
Source distribution

Target distribution



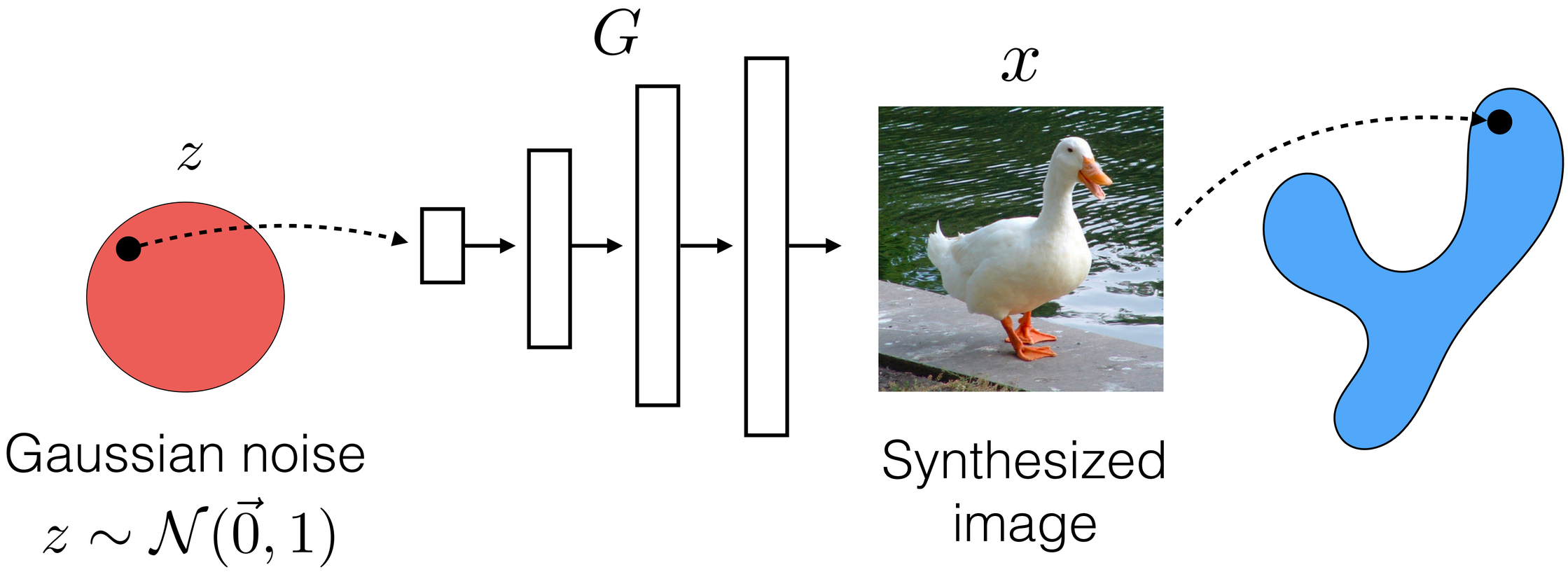
Source: Isola, Freeman, Torralba

Deep generative models as distribution transformers



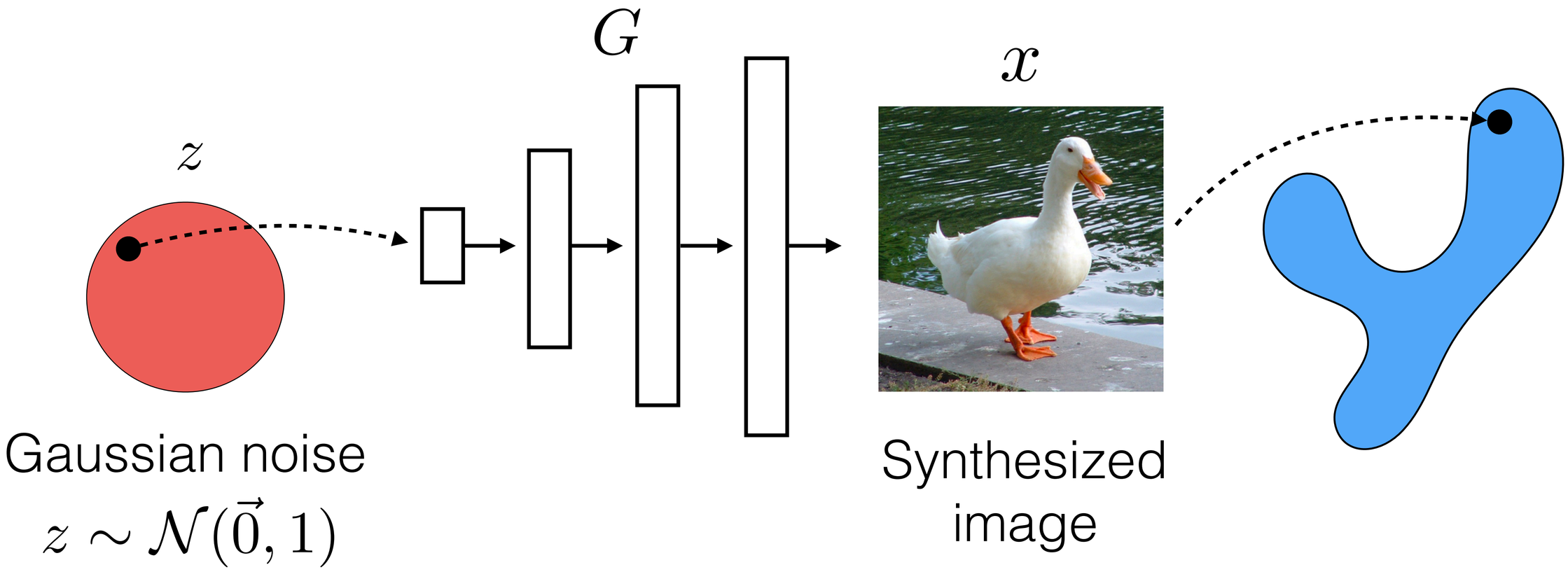
Source: Isola, Freeman, Torralba

Deep generative models as distribution transformers

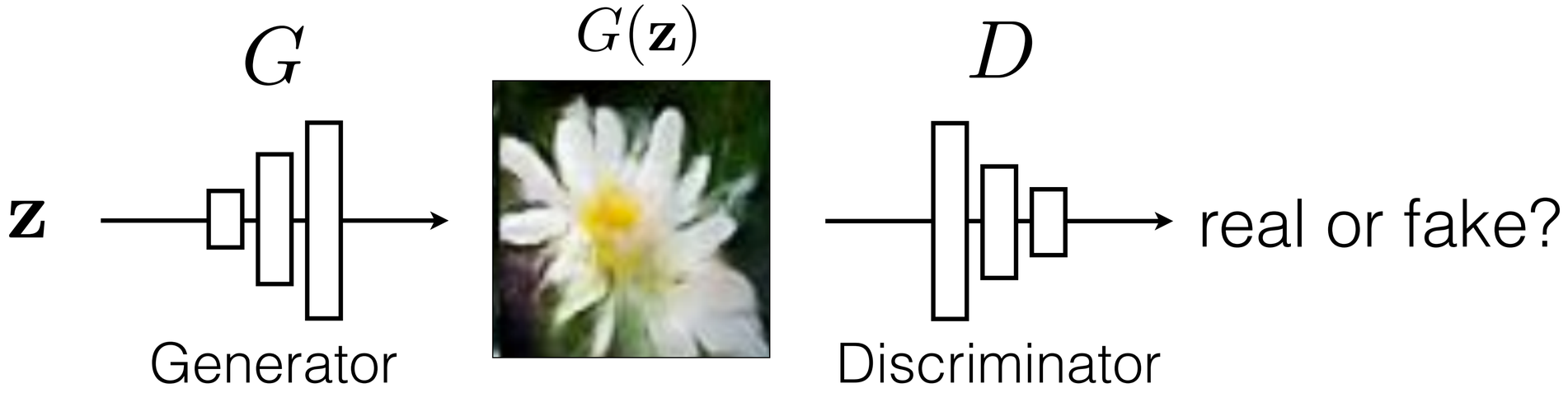


Source: Isola, Freeman, Torralba

Generative Adversarial Networks (GANs)



Source: Isola, Freeman, Torralba

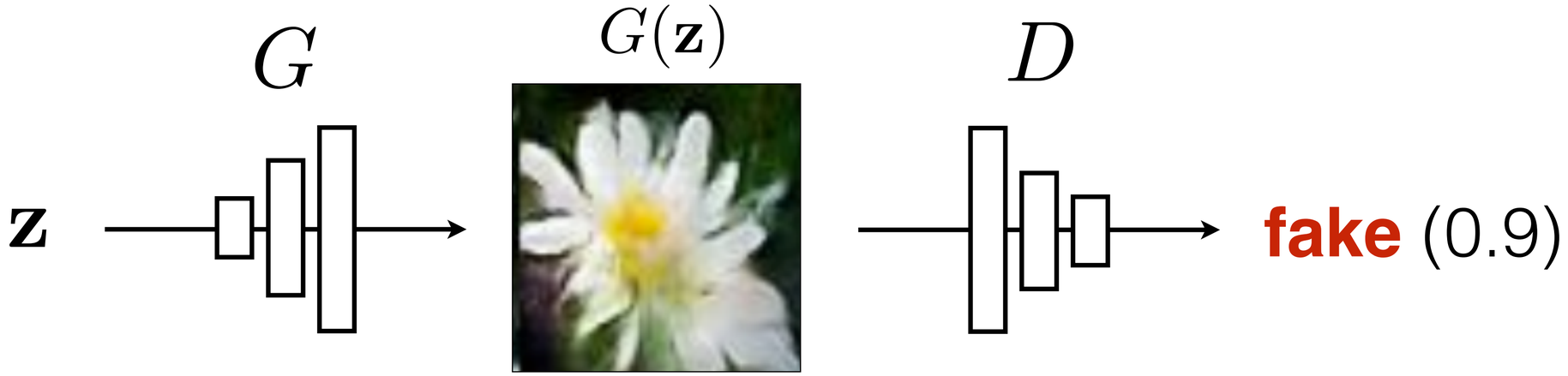


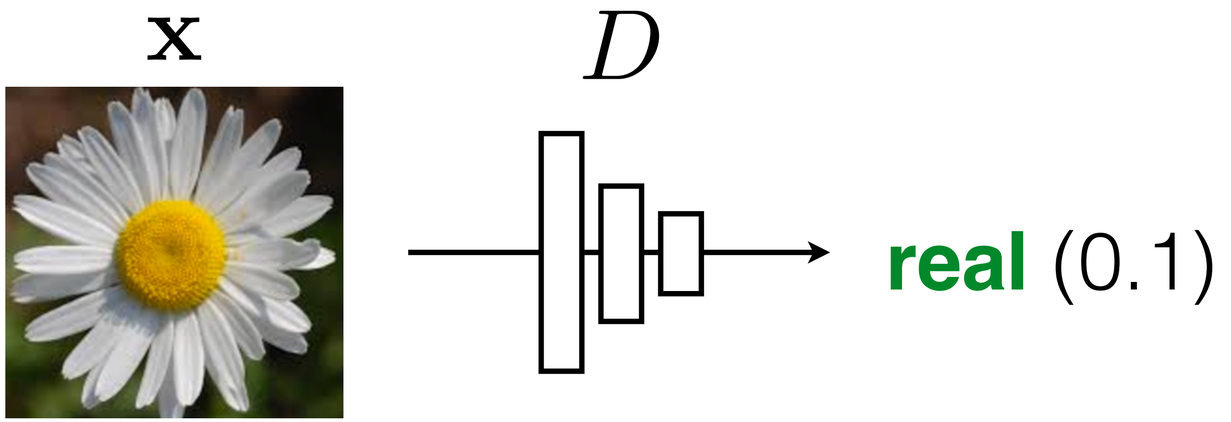
G tries to synthesize fake images that fool **D**

D tries to identify the fakes

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

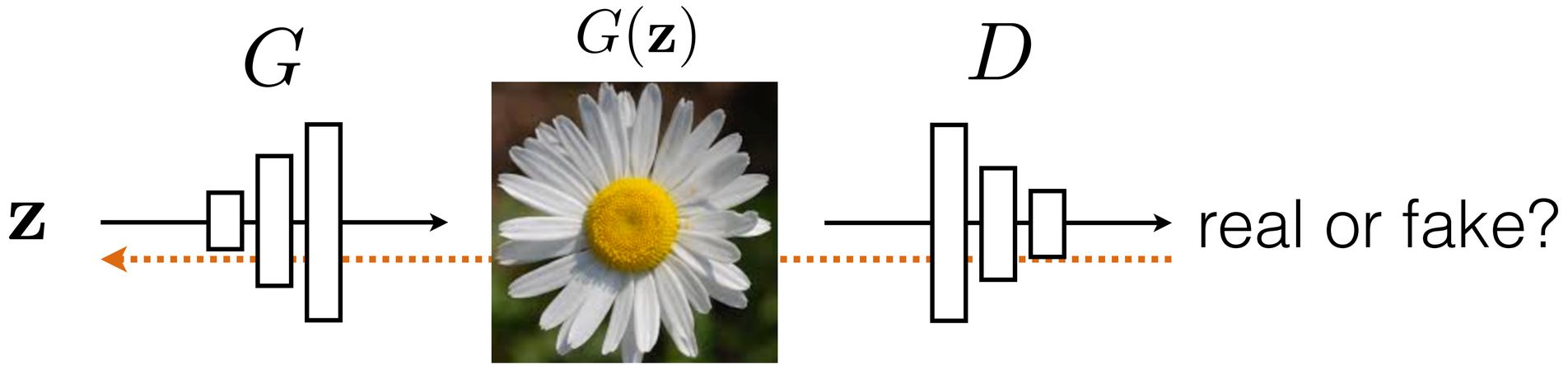




$$\arg\max_{D} \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \log D(G(\mathbf{z}))\ +\ \log\left(1 - D(\mathbf{x})\right)\ ]$$

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

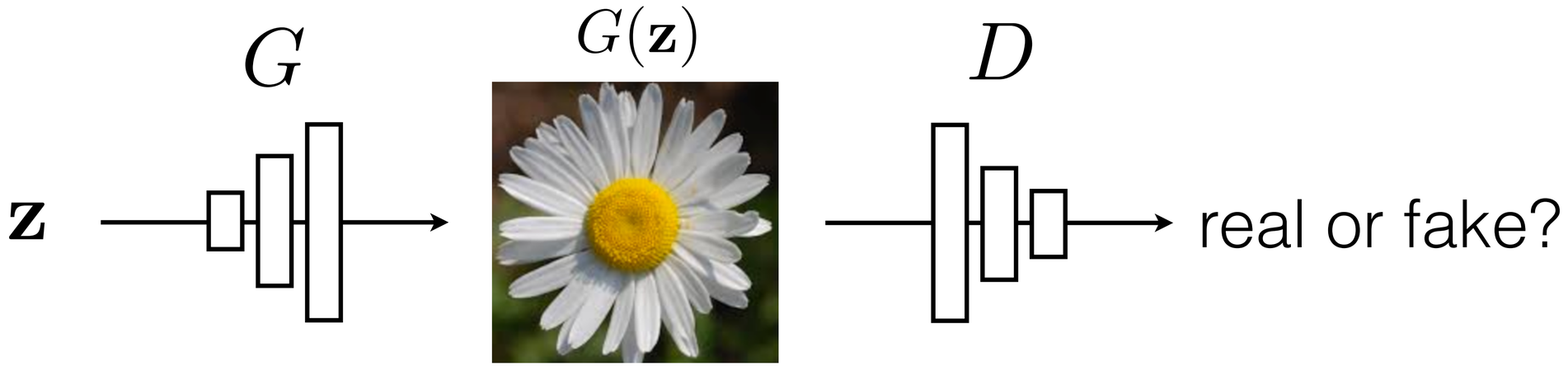


G tries to synthesize fake images that ***fool*** **D**:

$$\arg\min_{G} \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \log D(G(\mathbf{z}))\ +\ \log(1 - D(\mathbf{x}))\ ]$$

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba



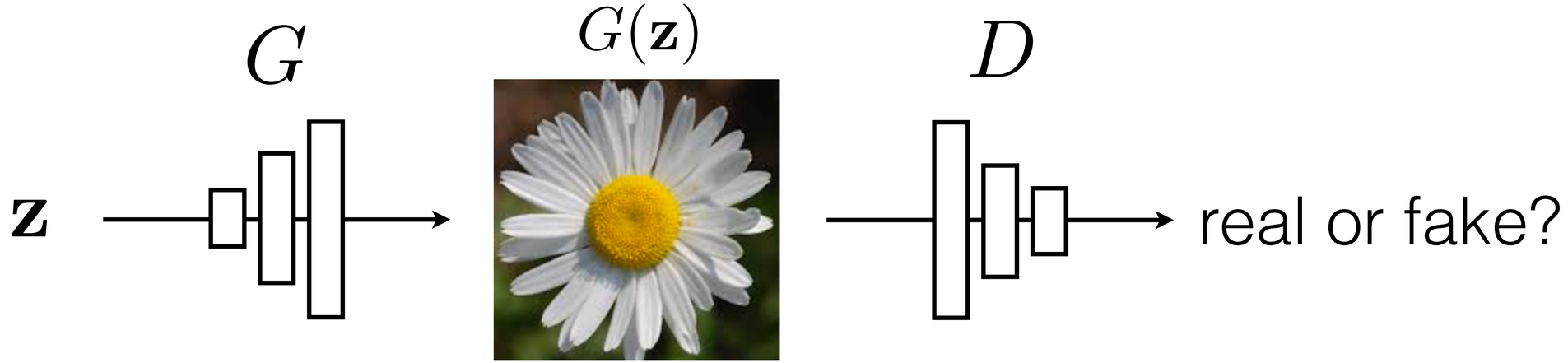
G tries to synthesize fake images that ***fool*** the ***best*** **D**:

$$\arg \min_G \max_D \; \mathbb{E}_{\mathbf{z},\mathbf{x}}[\;\; \log D(G(\mathbf{z})) \;\; + \;\; \log(1 - D(\mathbf{x})) \;\;\;]$$

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

Training



G tries to synthesize fake images that fool **D**

D tries to identify the fakes

- Training: iterate between training D and G with backprop.
- Global optimum when G reproduces data distribution.

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

$p_g = p_{data}$ is the unique global minimizer of the GAN objective.

Proof sketch:

Can show this is optimal D, given G

$$C(G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))]$$

$$= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]$$

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right)$$

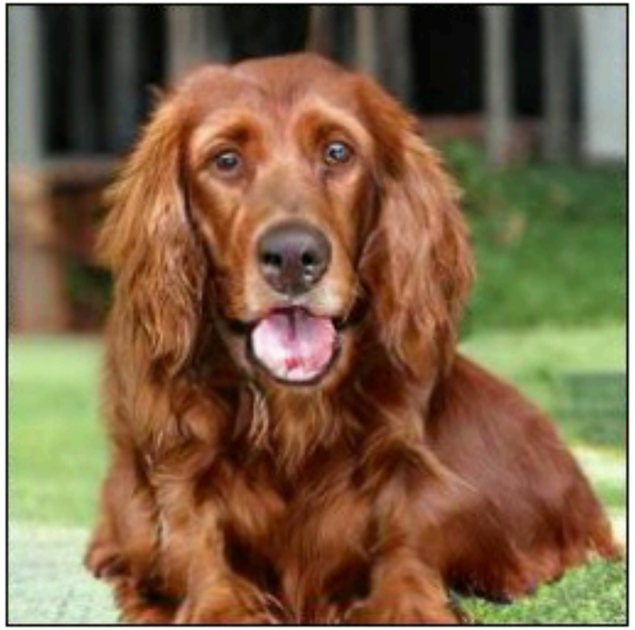
$$C(G) = -\log(4) + 2 \cdot \underbrace{JSD\left(p_{\text{data}} \| p_g\right)}_{\geq 0}$$

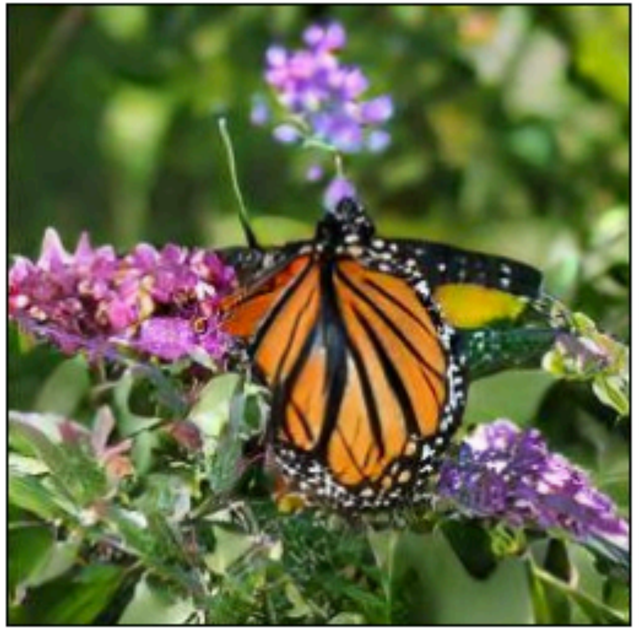
$$0 \overset{39}{\iff} p_g = p_{data} \quad \square$$

Source: Isola, Freeman, Torralba

Samples from BigGAN

[Brock et al. 2018]



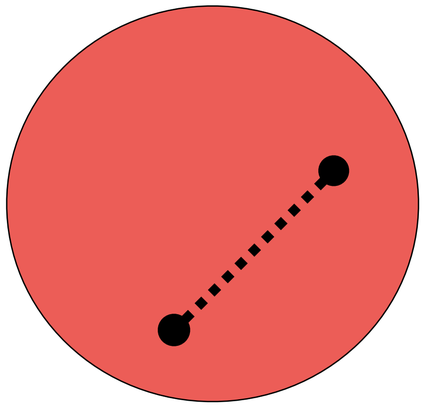



More here: https://arxiv.org/pdf/1809.11096.pdf

Source: Isola, Freeman, Torralba

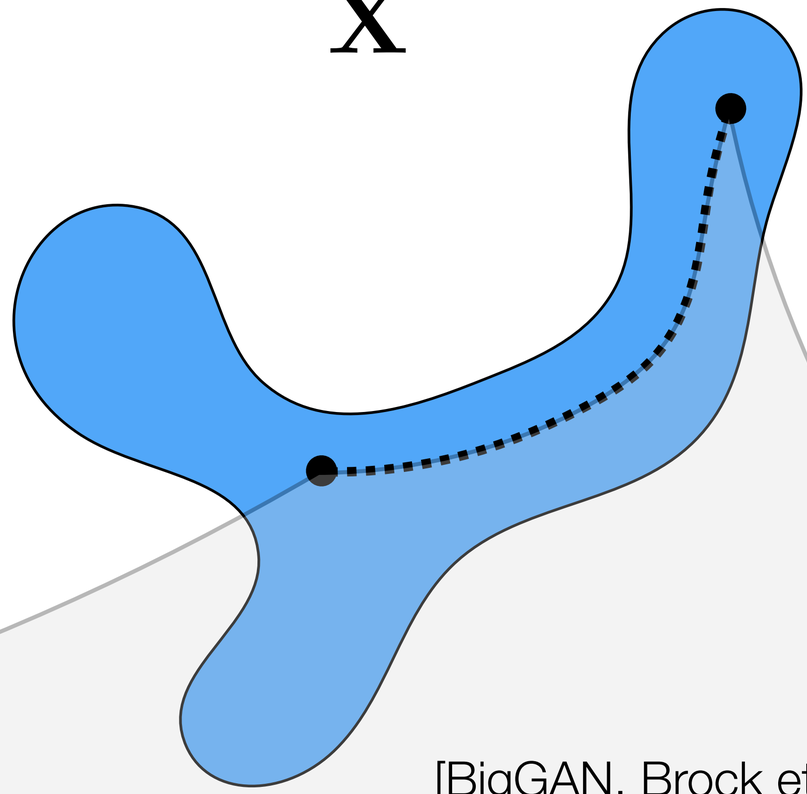
Latent space
(Gaussian)

$\mathbf{z}$



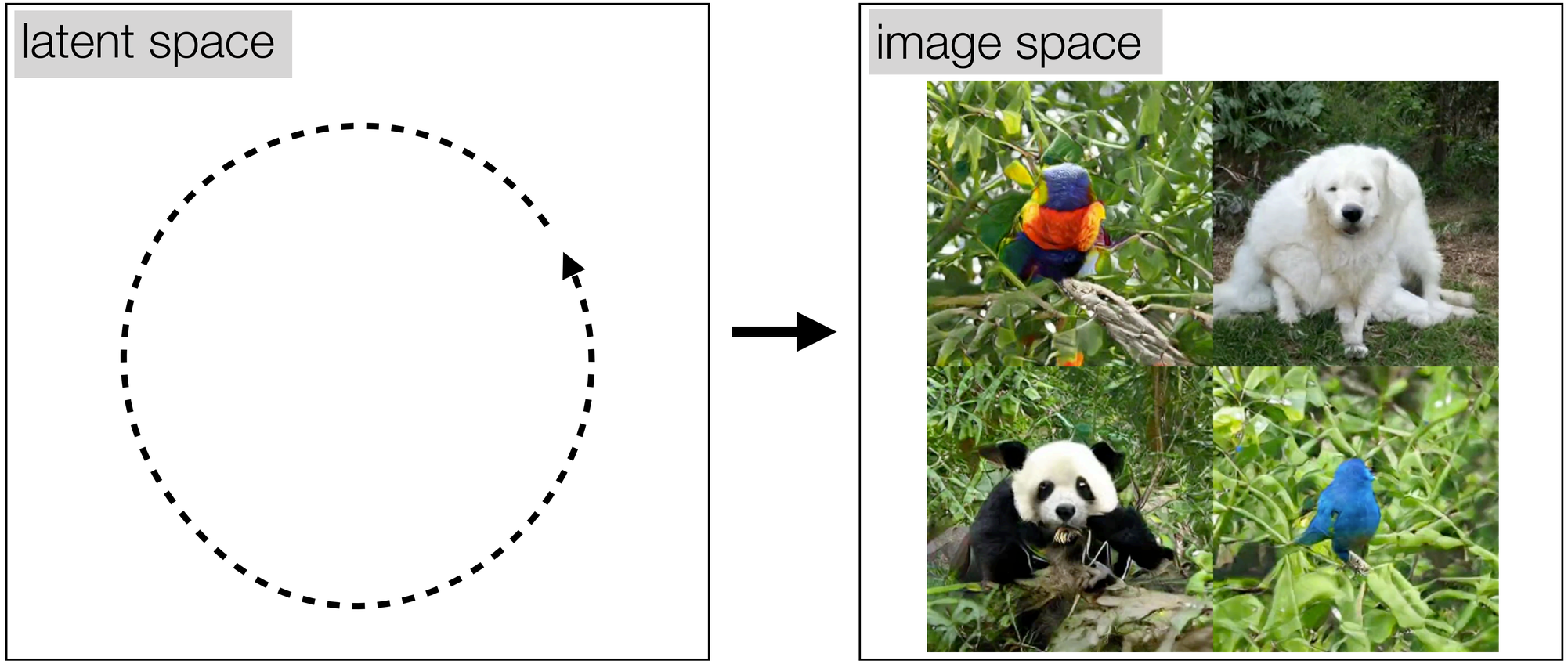
Data space
(Natural image manifold)

$\mathbf{x}$



[BigGAN, Brock et al. 2018]

Generative models organize the manifold of natural images



Source: Isola, Freeman, Torralba

What has driven GAN progress?

Ian Goodfellow @goodfellow_ian · Jan 14

4.5 years of **GAN progress** on face generation. arxiv.org/abs/1406.2661 arxiv.org/abs/1511.06434 arxiv.org/abs/1606.07536 arxiv.org/abs/1710.10196 arxiv.org/abs/1812.04948

Source: Isola, Freeman, Torralba

Better objectives? Optimization?

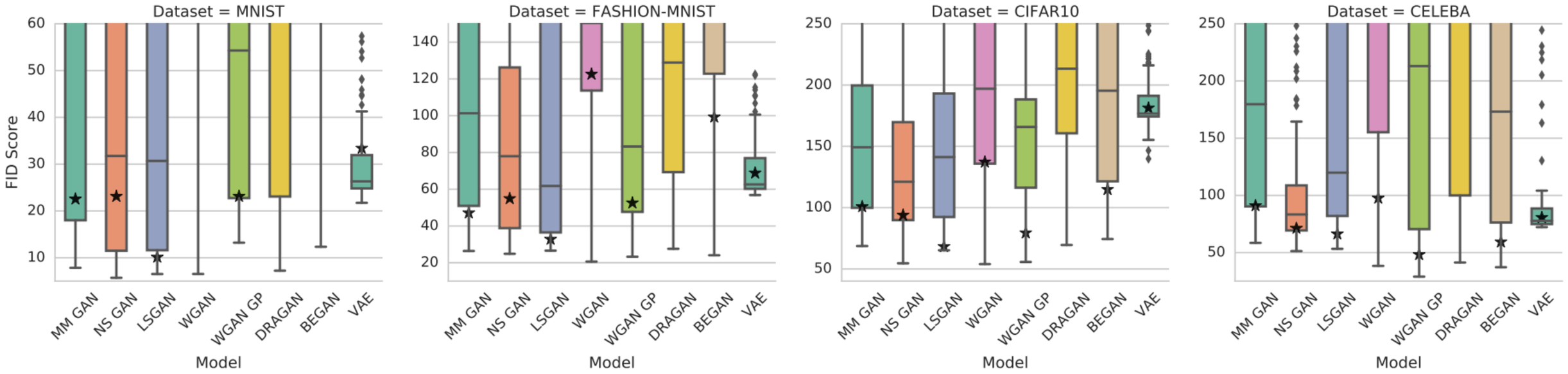


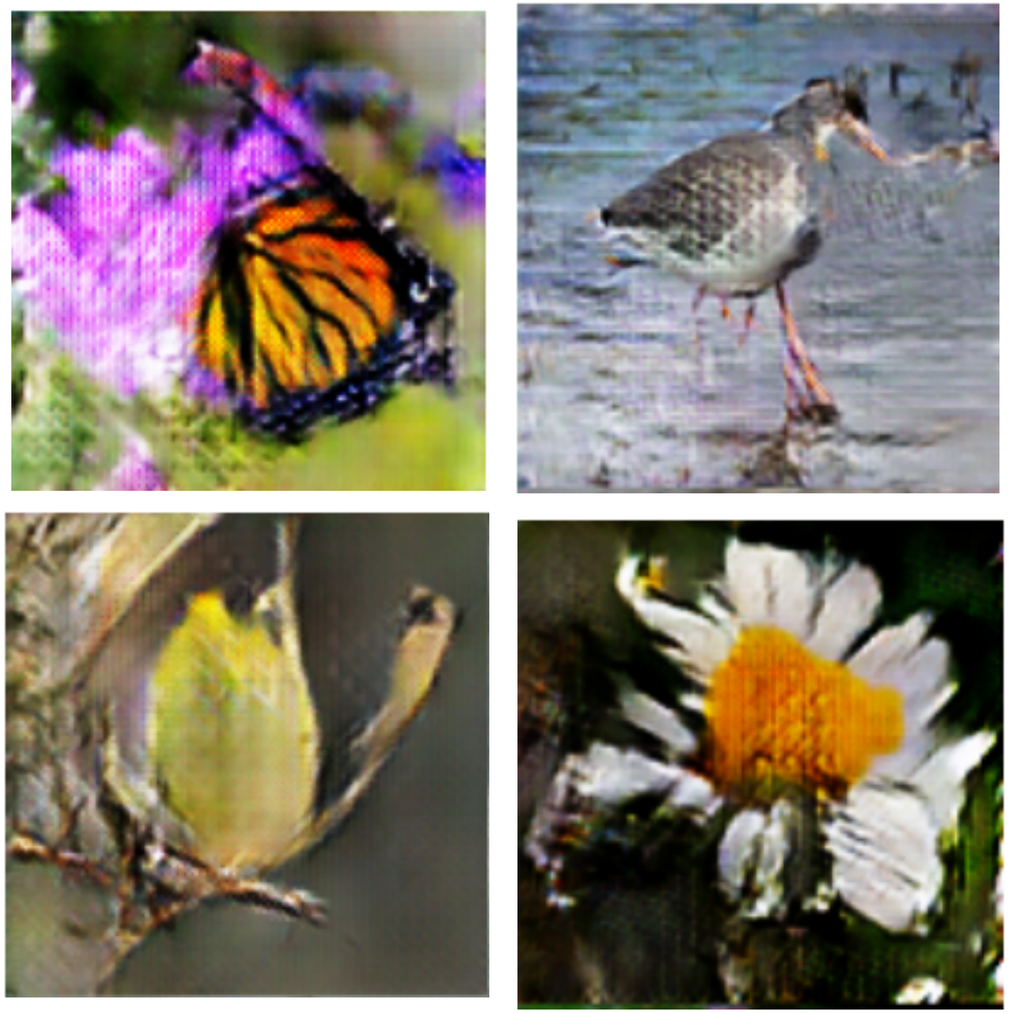
Figure 4: A *wide range* hyperparameter search (100 hyperparameter samples per model). Black stars indicate the performance of suggested hyperparameter settings. We observe that GAN training is extremely sensitive to hyperparameter settings and there is no model which is significantly more stable than others.

["Are all GANs Created Equal?", Lucic*, Kurach*, et al. 2018]

Source: Isola, Freeman, Torralba

More data?

ACGAN [Odena et al. 2016]



BigGAN [Brock et al. 2018]

Both trained on Imagenet

Source: Isola, Freeman, Torralba

Architectures

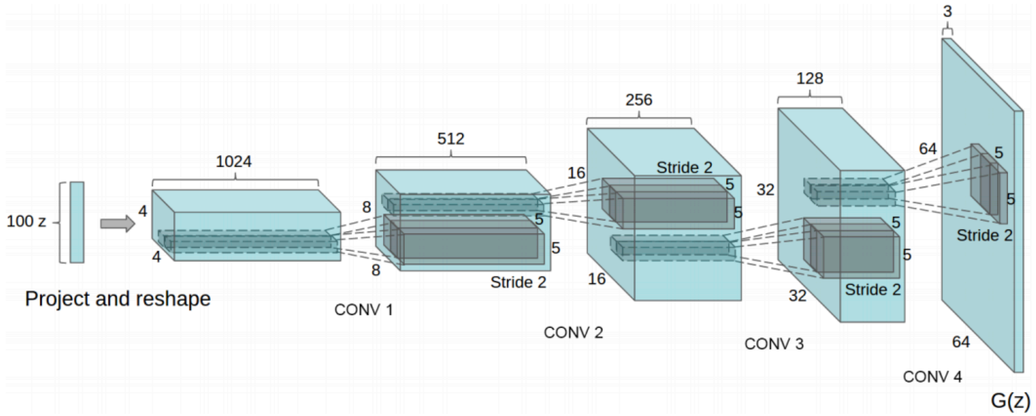
DCGAN
[Radford, Metz, Chintala 2016]

StyleGAN
[Karras, Laine, Aila 2019]

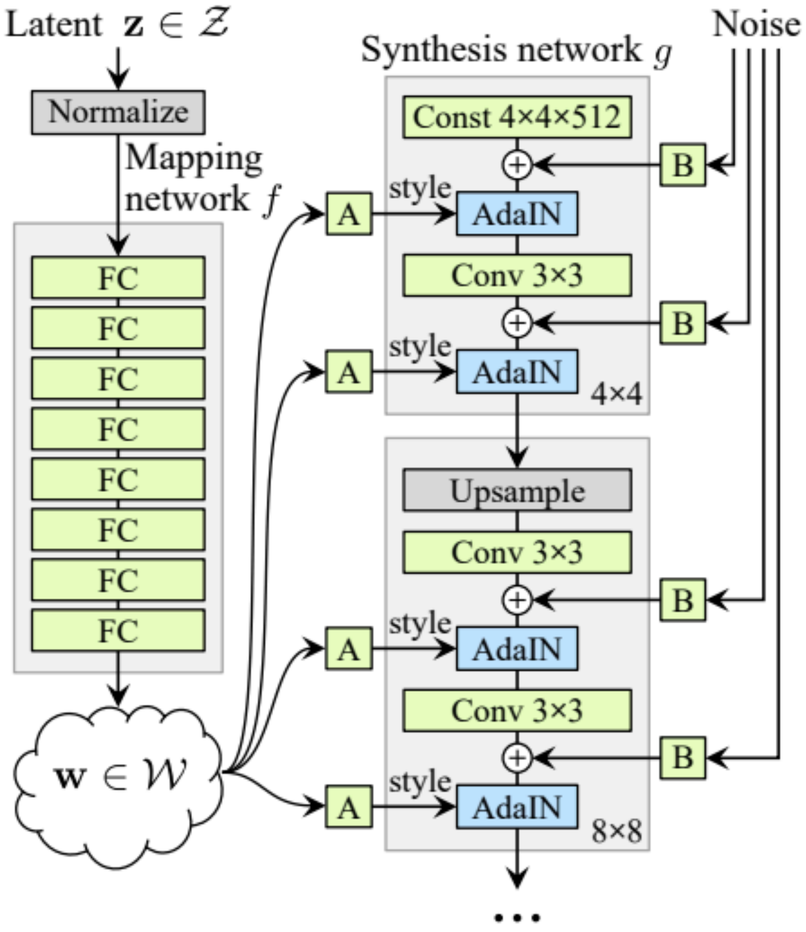
Source: Isola, Freeman, Torralba

Architectures

DCGAN
[Radford, Metz, Chintala 2016]



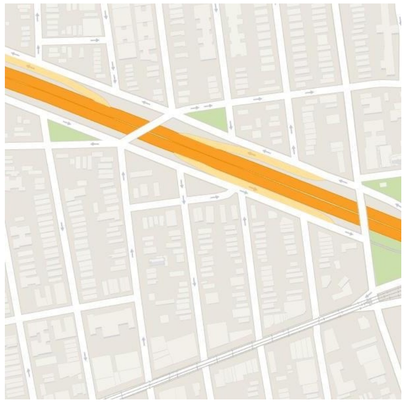
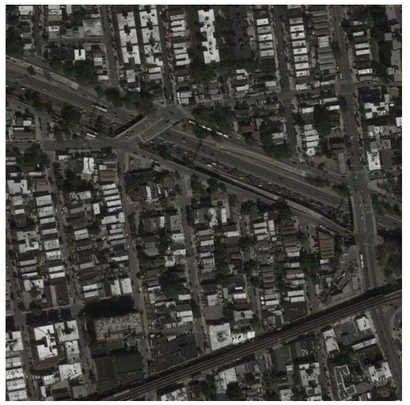
StyleGAN
[Karras, Laine, Aila 2019]

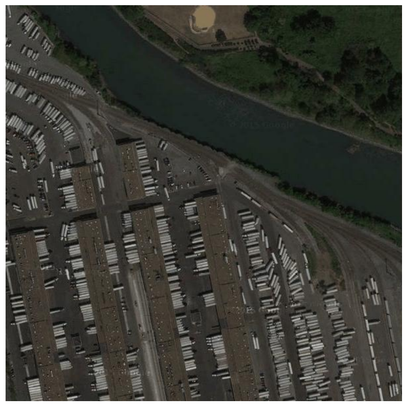


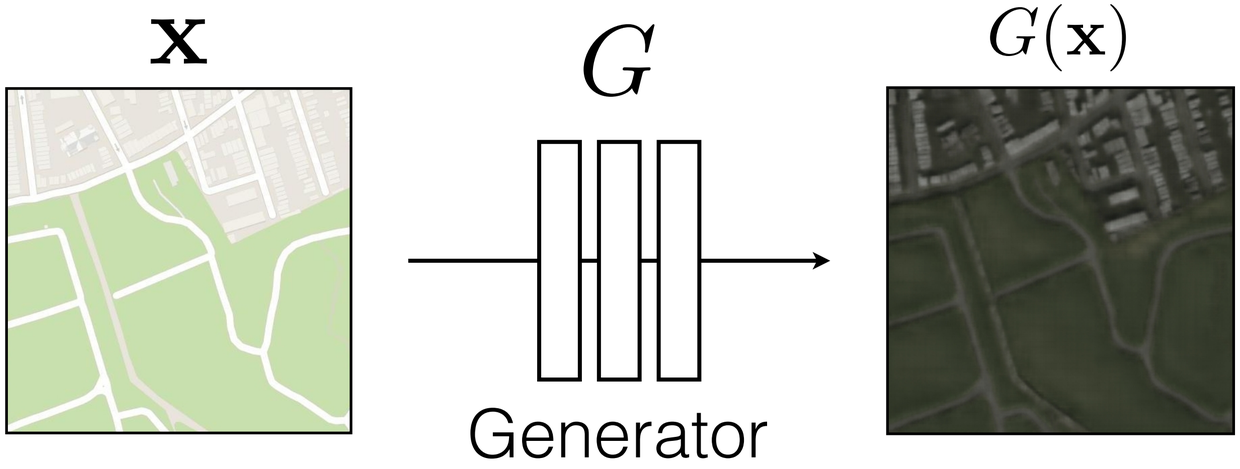
Source: Isola, Freeman, Torralba

Map2Sat

$\mathbf{x}$ $\mathbf{y}$

{  ,  }

{  ,  }



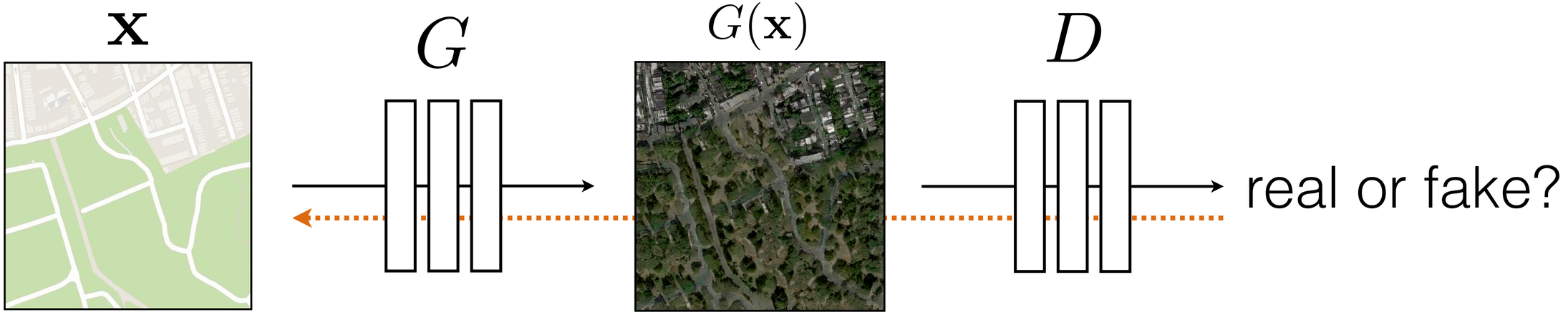
Source: Isola, Freeman, Torralba

G tries to synthesize fake images that fool **D**

D tries to identify the fakes

$$\arg\max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\, \log D(G(\mathbf{x})) \; + \; \log(1 - D(\mathbf{y}))\,]$$

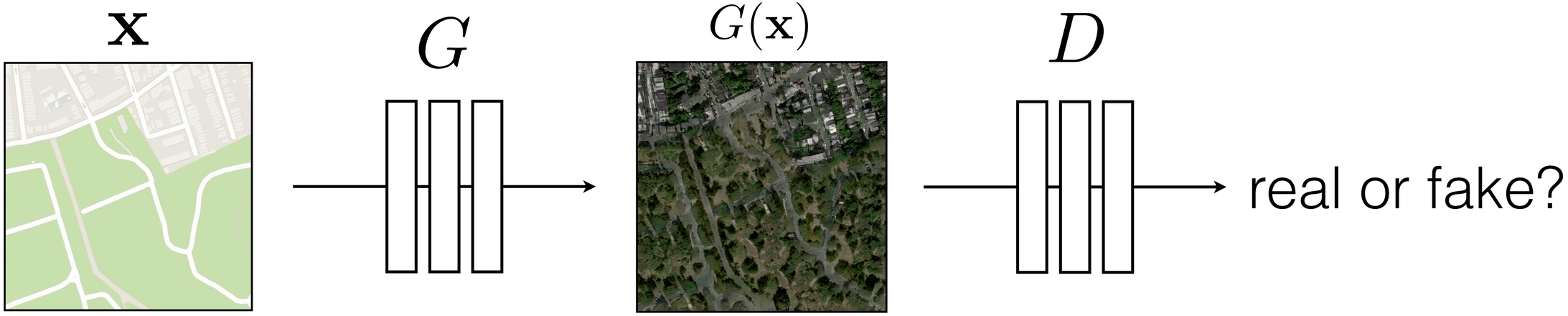
Source: Isola, Freeman, Torralba



G tries to synthesize fake images that ***fool*** **D**:

$$\arg \min_{G} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ ]$$

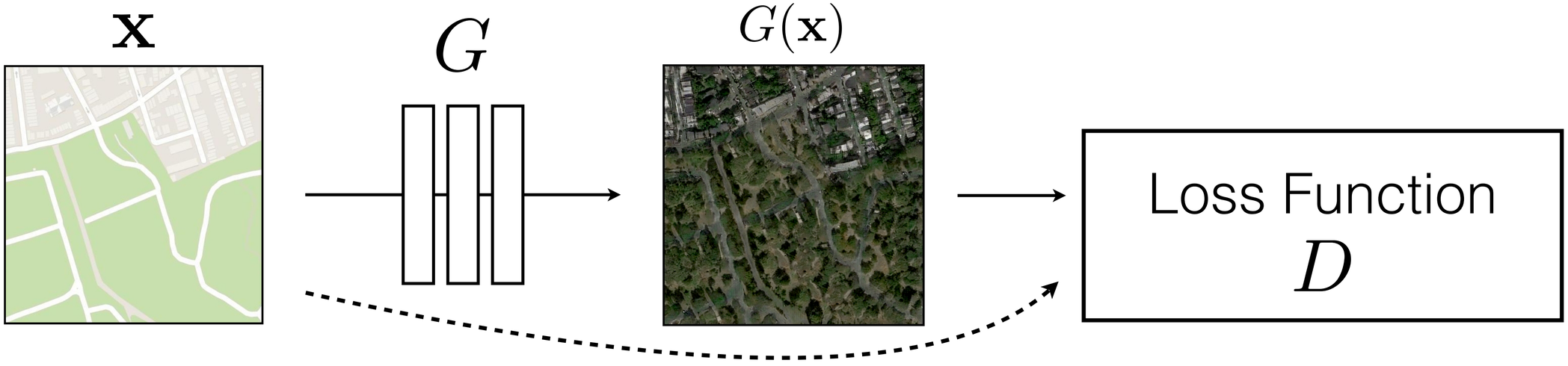
Source: Isola, Freeman, Torralba



G tries to synthesize fake images that ***fool*** the ***best*** **D**:

$$\arg \min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \;\; + \;\; \log(1 - D(\mathbf{y})) \;\;]$$

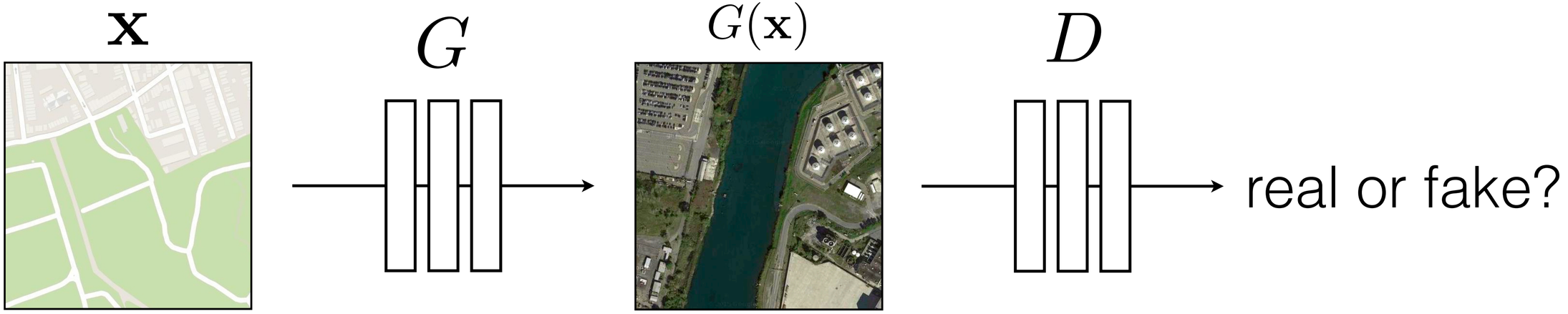
Source: Isola, Freeman, Torralba



G's perspective: **D** is a loss function.

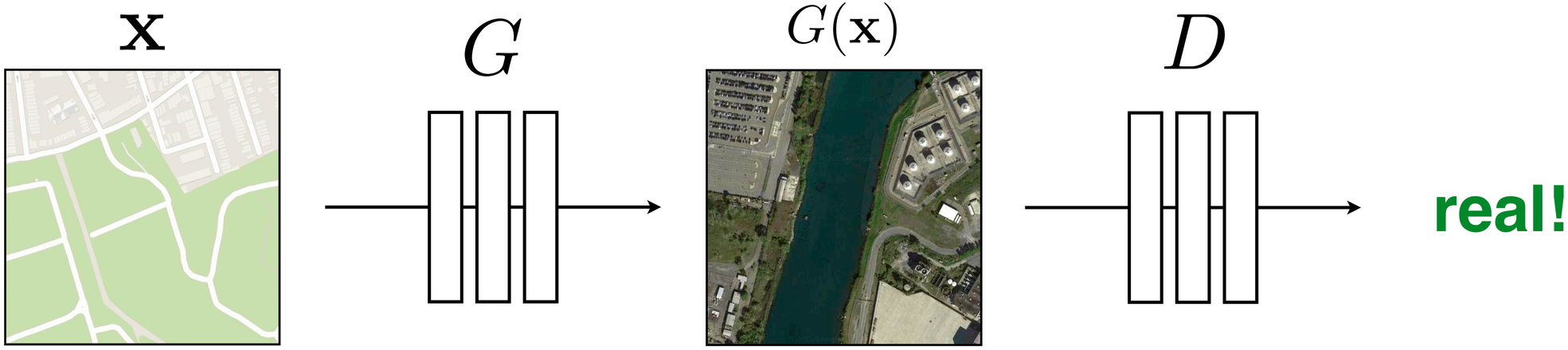
Rather than being hand-designed, it is *learned* and *highly structured*.

Source: Isola, Freeman, Torralba



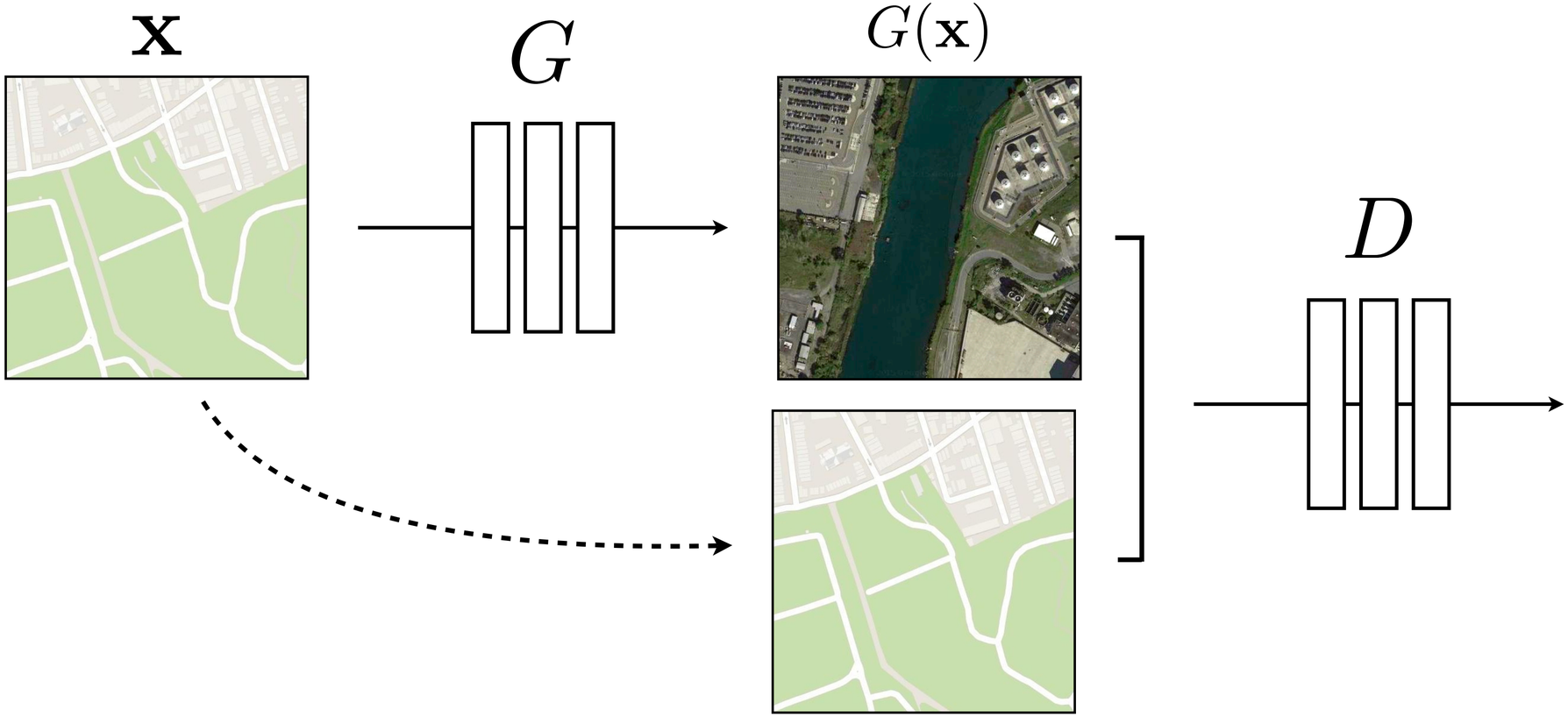
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\;]$$

Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y})) \;\;]$$

Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\;]$$

Source: Isola, Freeman, Torralba

$\mathbf{x}$

G

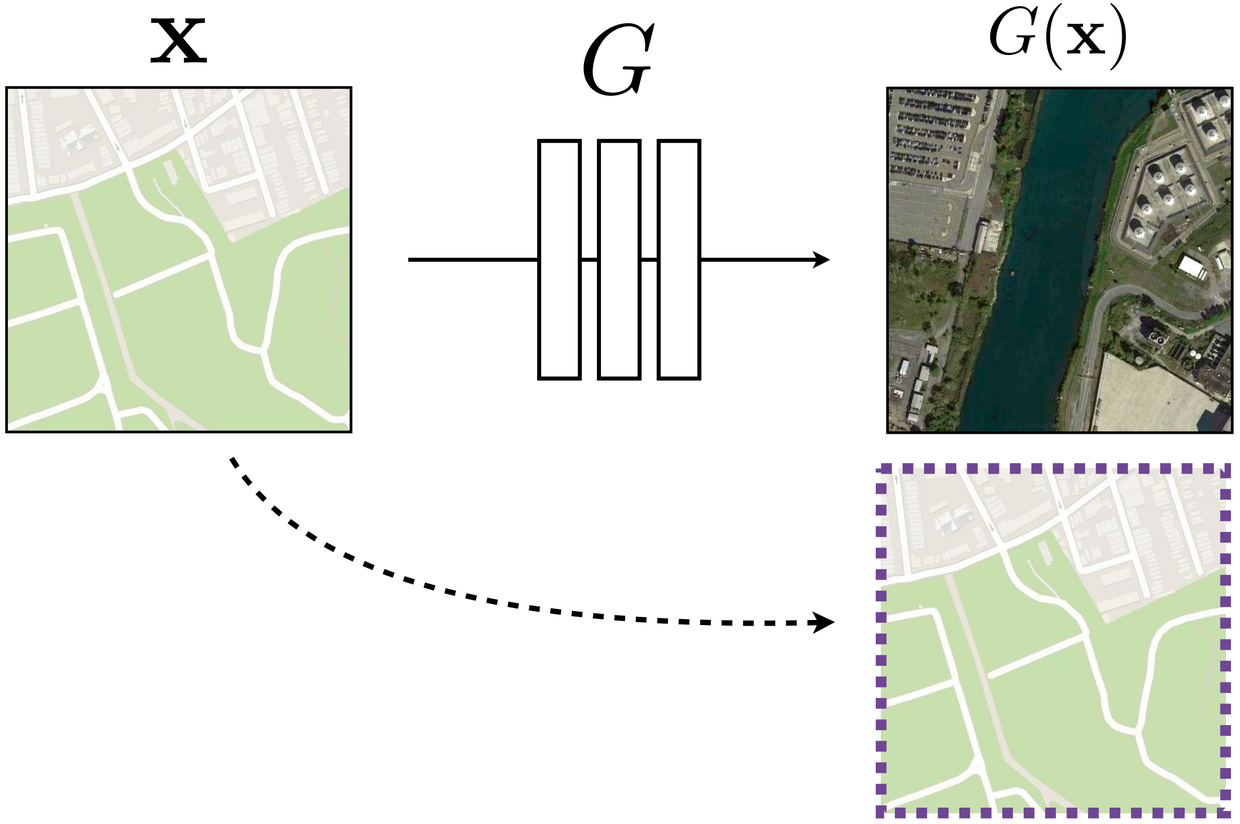
$G(\mathbf{x})$

D

real or fake *pair* ?

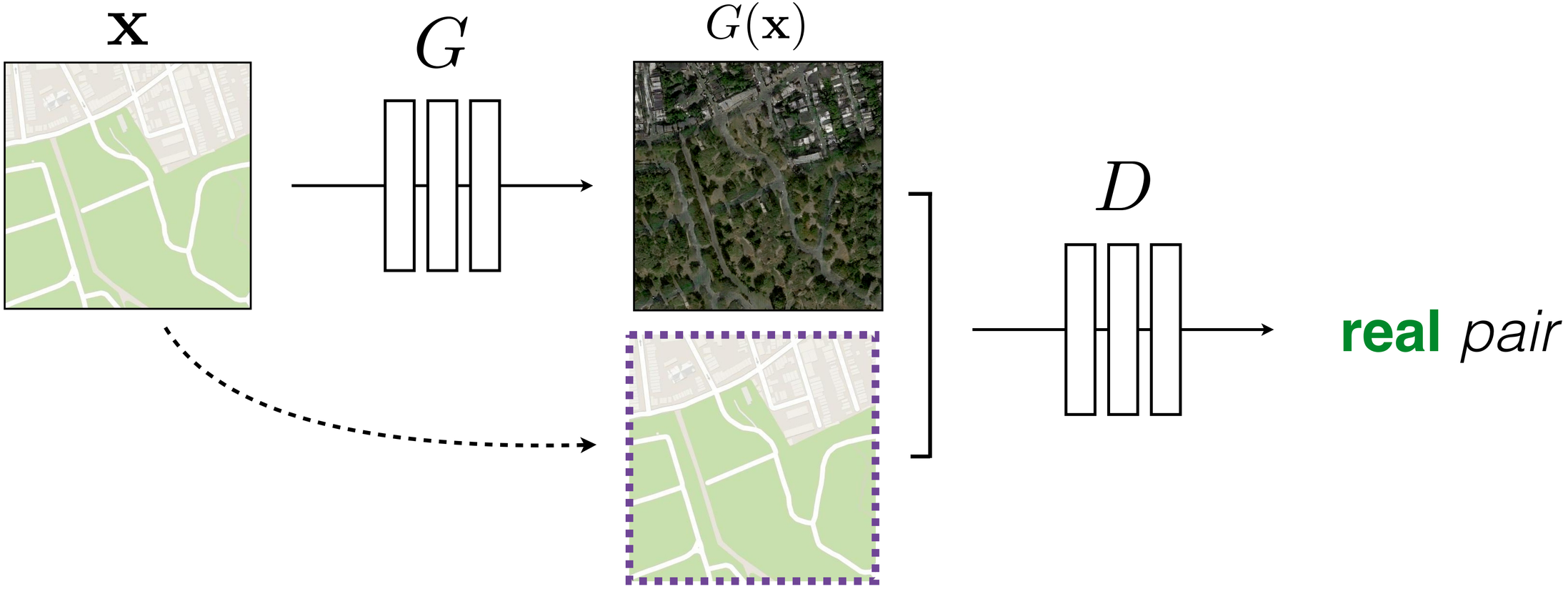
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$

Source: Isola, Freeman, Torralba



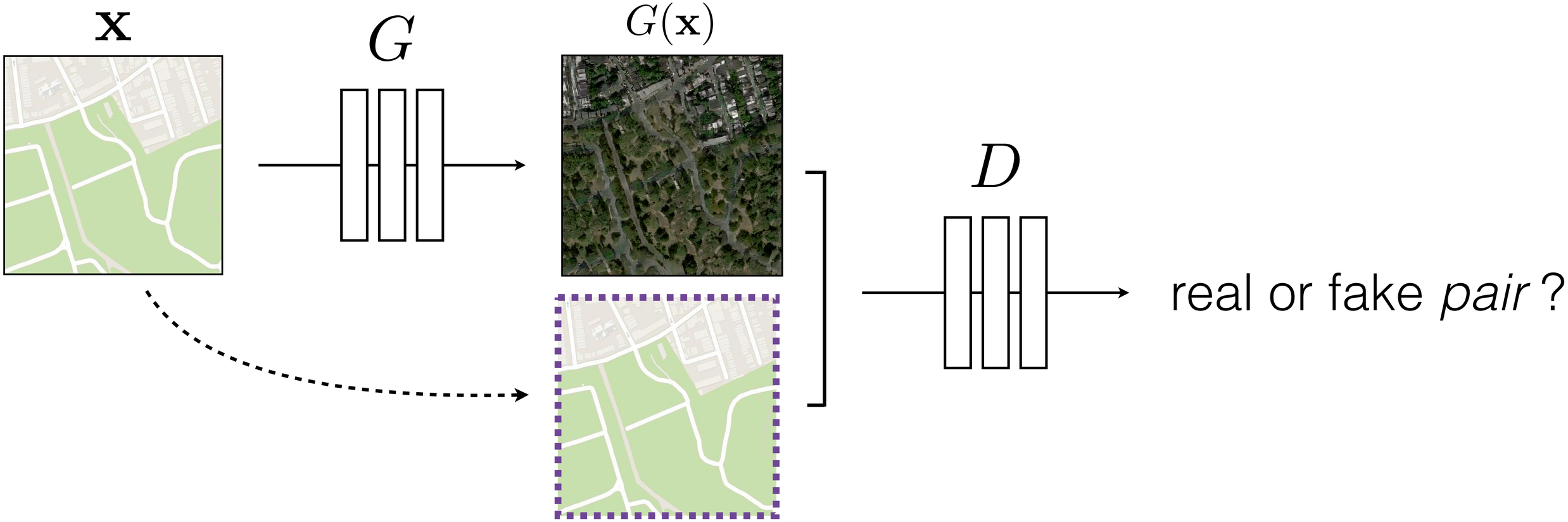
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$

Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$

Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;\;]$$

Source: Isola, Freeman, Torralba

Training Details: Loss function

Conditional GAN

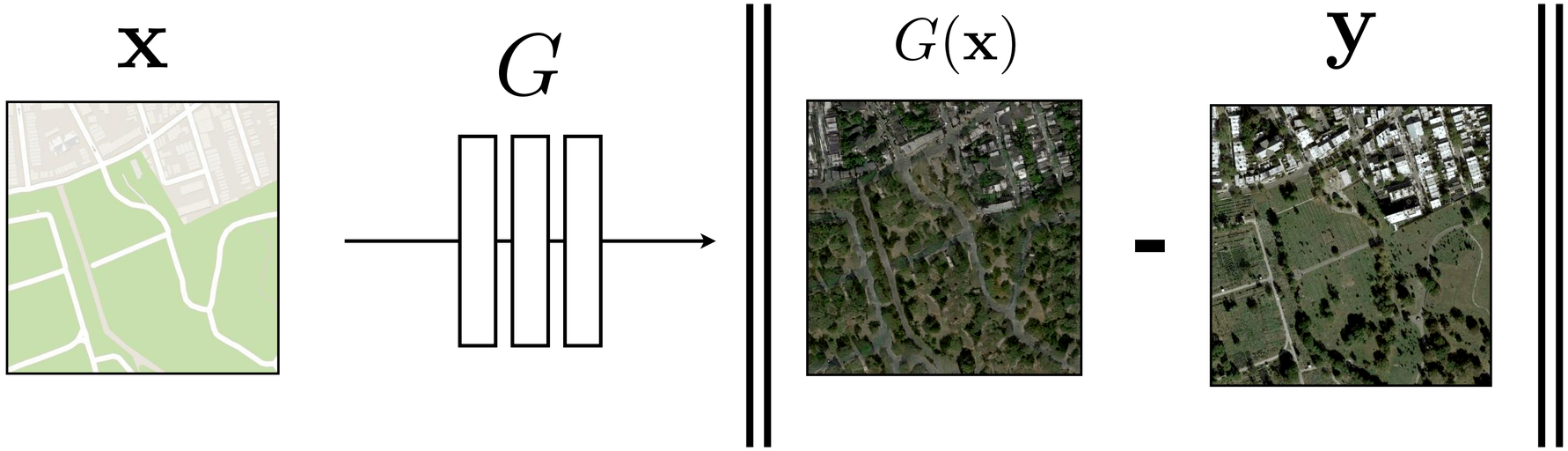
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

Source: Isola, Freeman, Torralba

Training Details: Loss function

Conditional GAN

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$



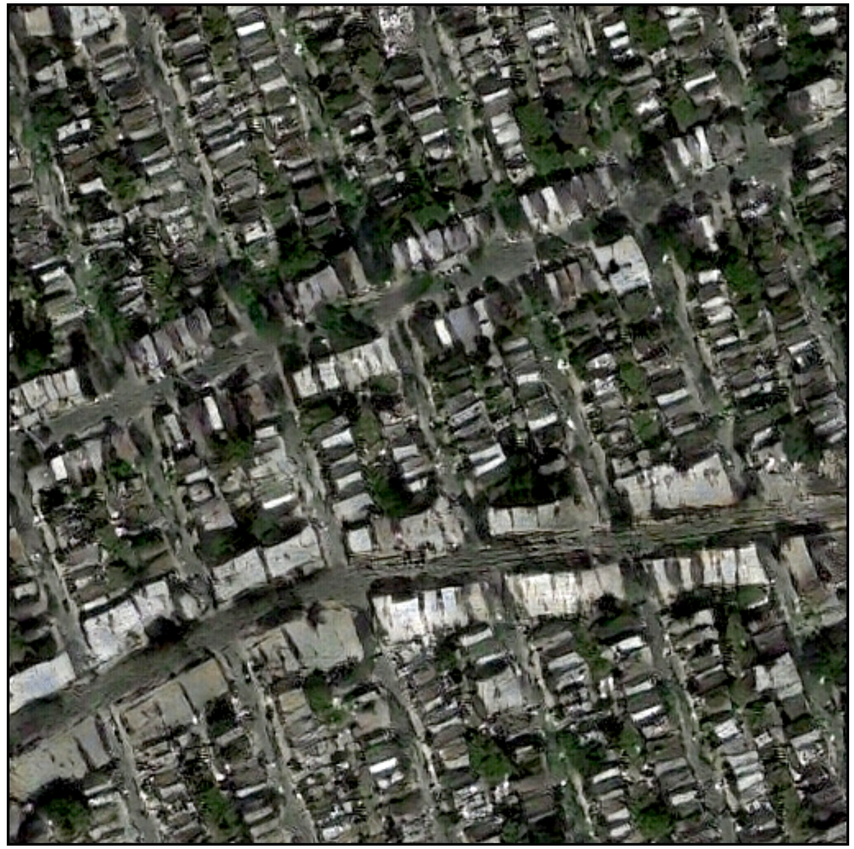
Stable training + fast convergence

[c.f. Pathak et al. CVPR 2016]

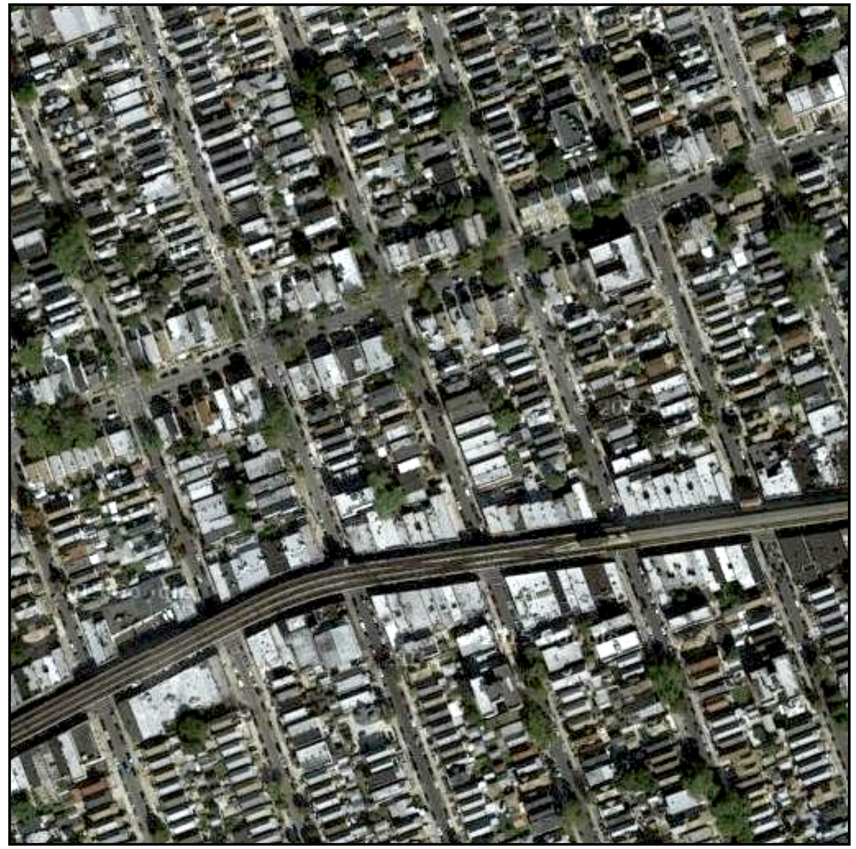
Source: Isola, Freeman, Torralba

Input

Output



Groundtruth



Data from
[maps.google.com]

Source: Isola, Freeman, Torralba

Input

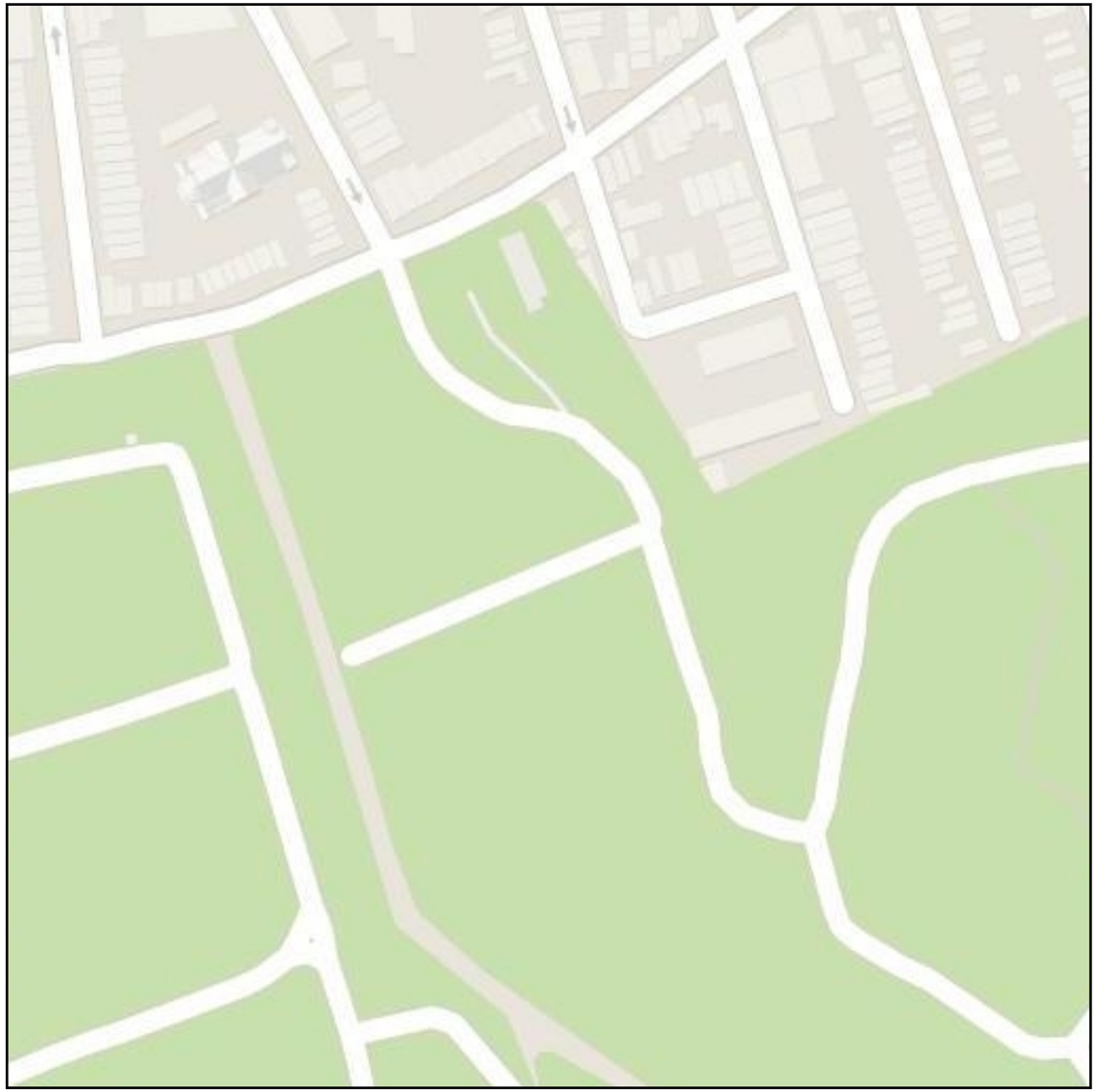
Output

Groundtruth

Data from [maps.google.com]

Source: Isola, Freeman, Torralba

Input

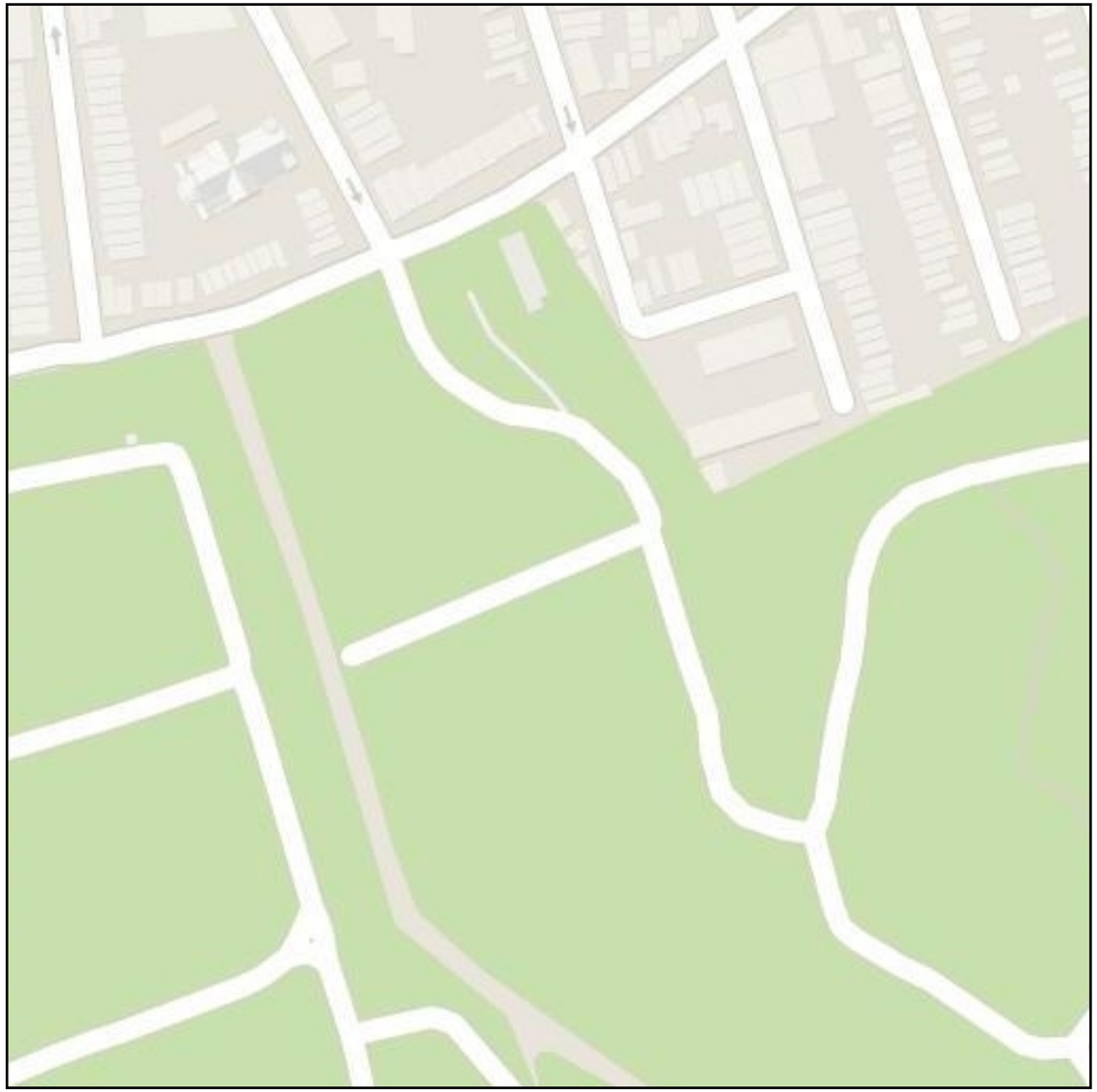


Unstructured prediction (L1)

Source: Isola, Freeman, Torralba

Input

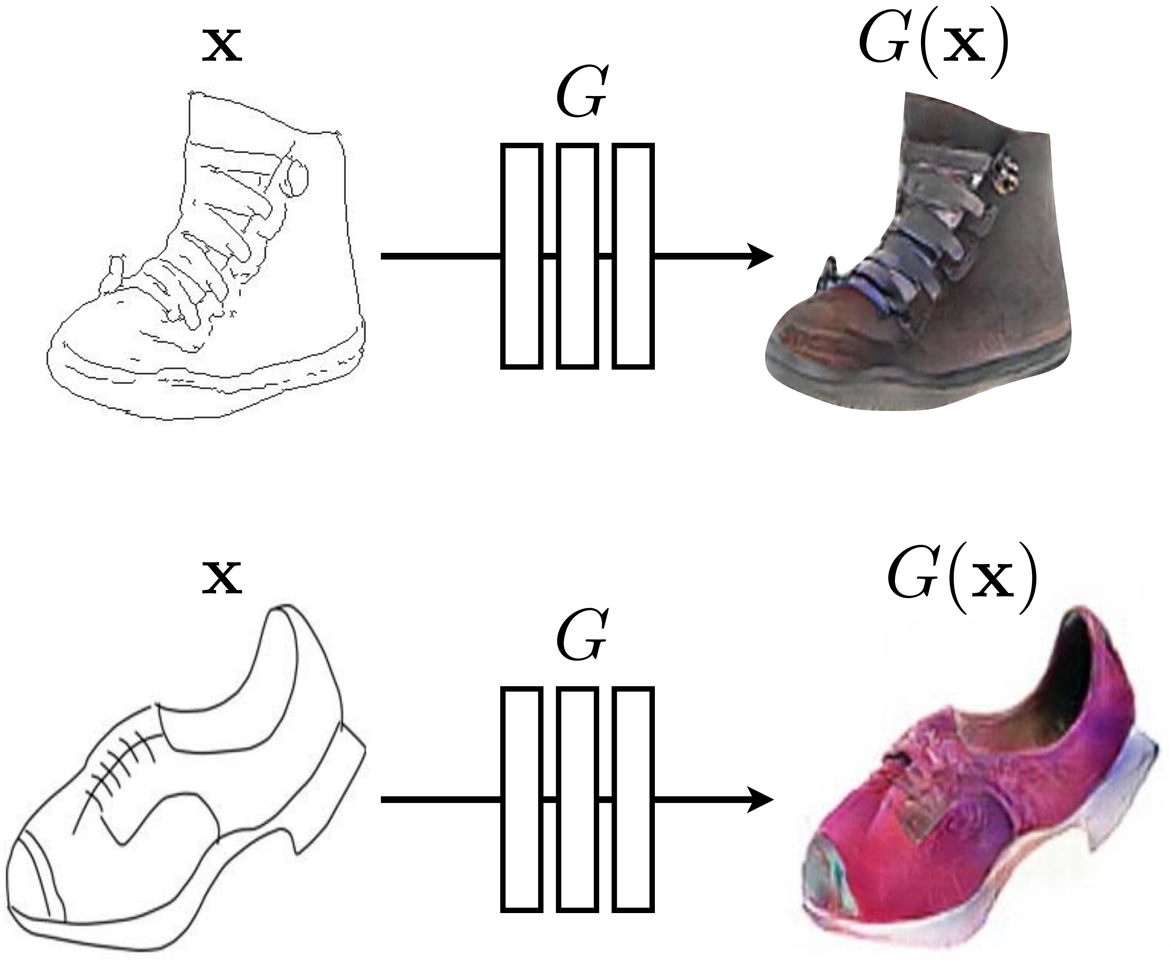
Structured Prediction (cGAN)



Source: Isola, Freeman, Torralba

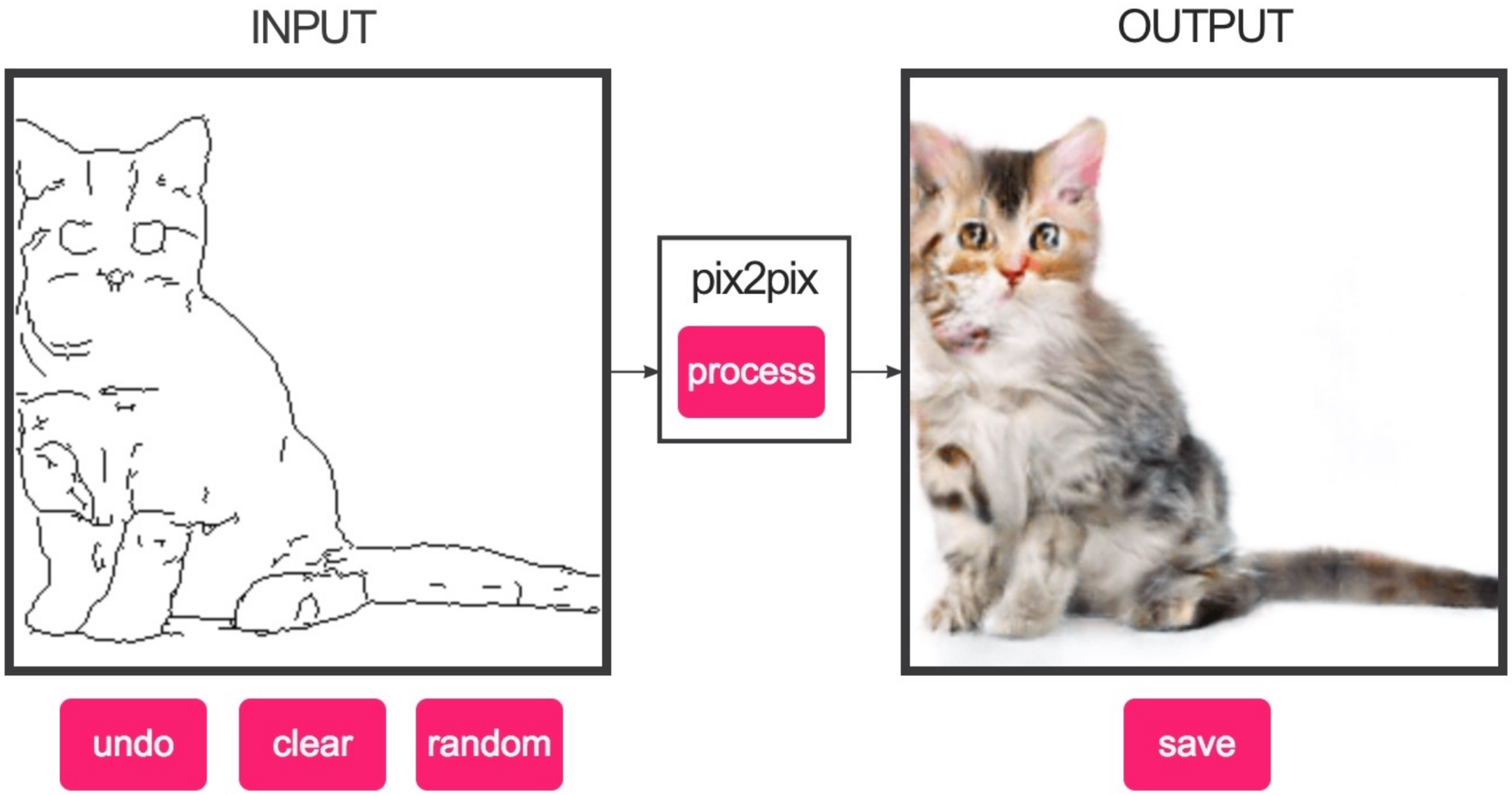
Training data

[HED, Xie & Tu, 2015]

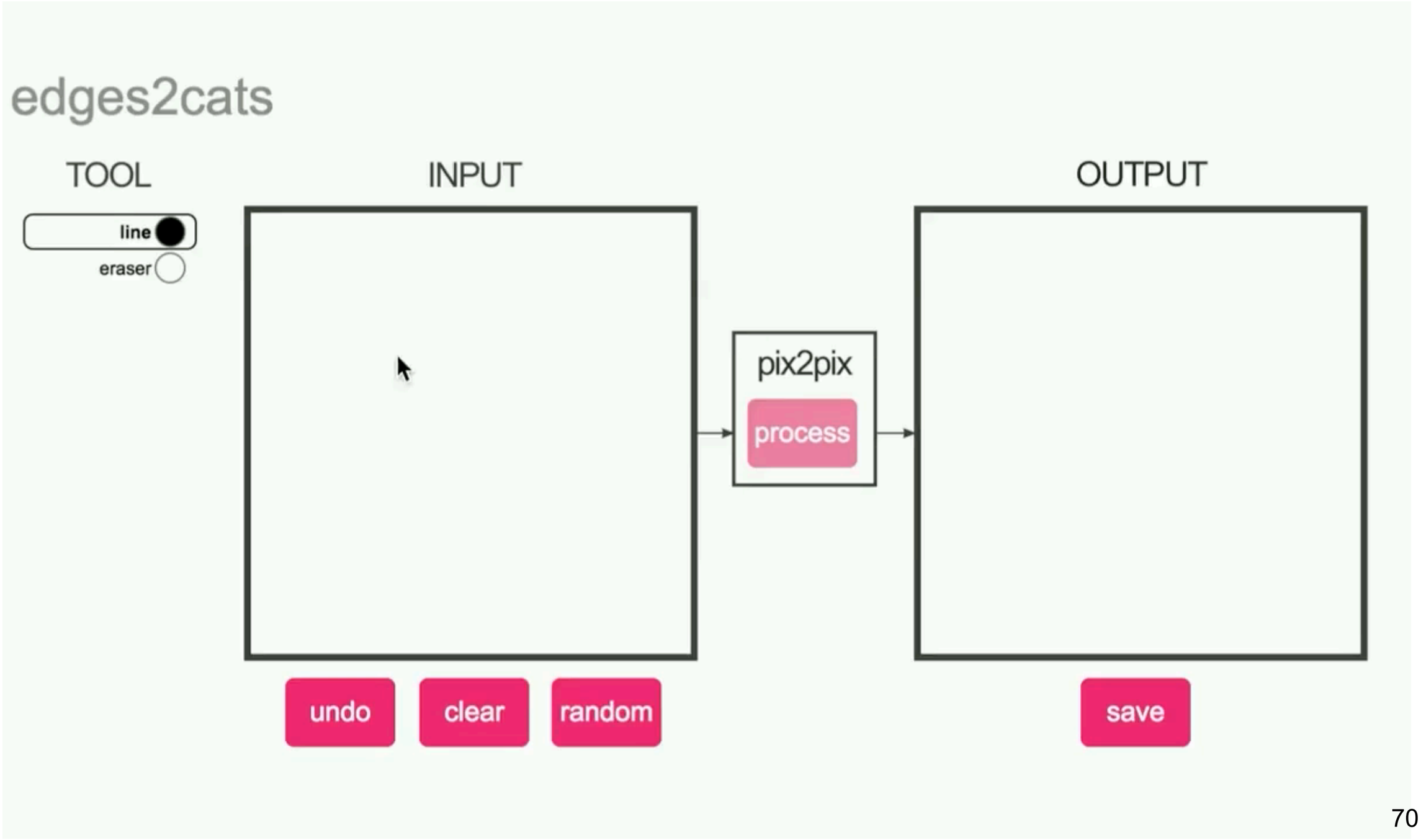


Source: Isola, Freeman, Torralba

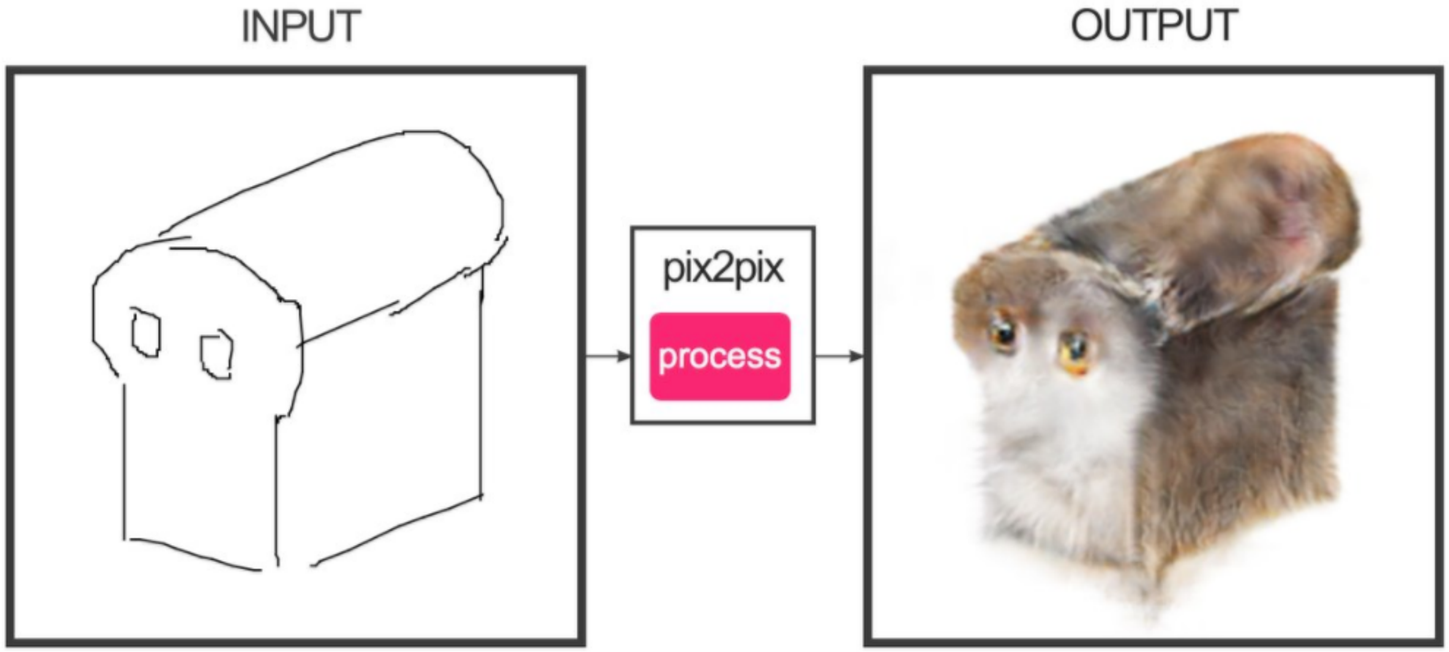
#edges2cats [Chris Hesse]



Source: Isola, Freeman, Torralba



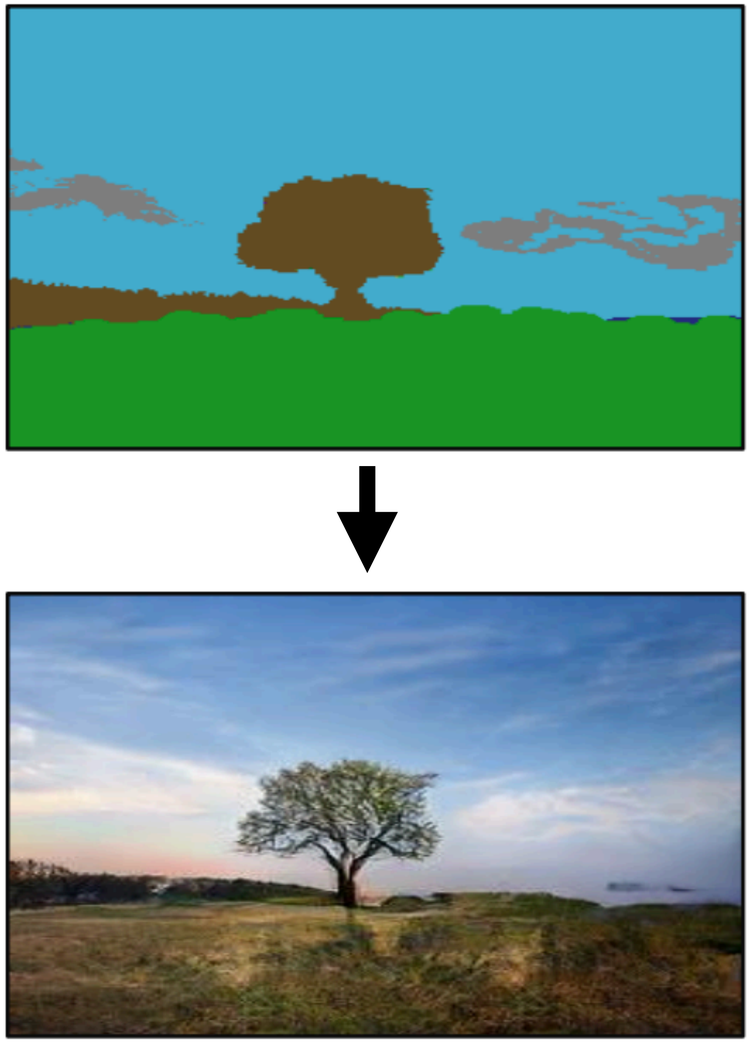
Source: Isola, Freeman, Torralba



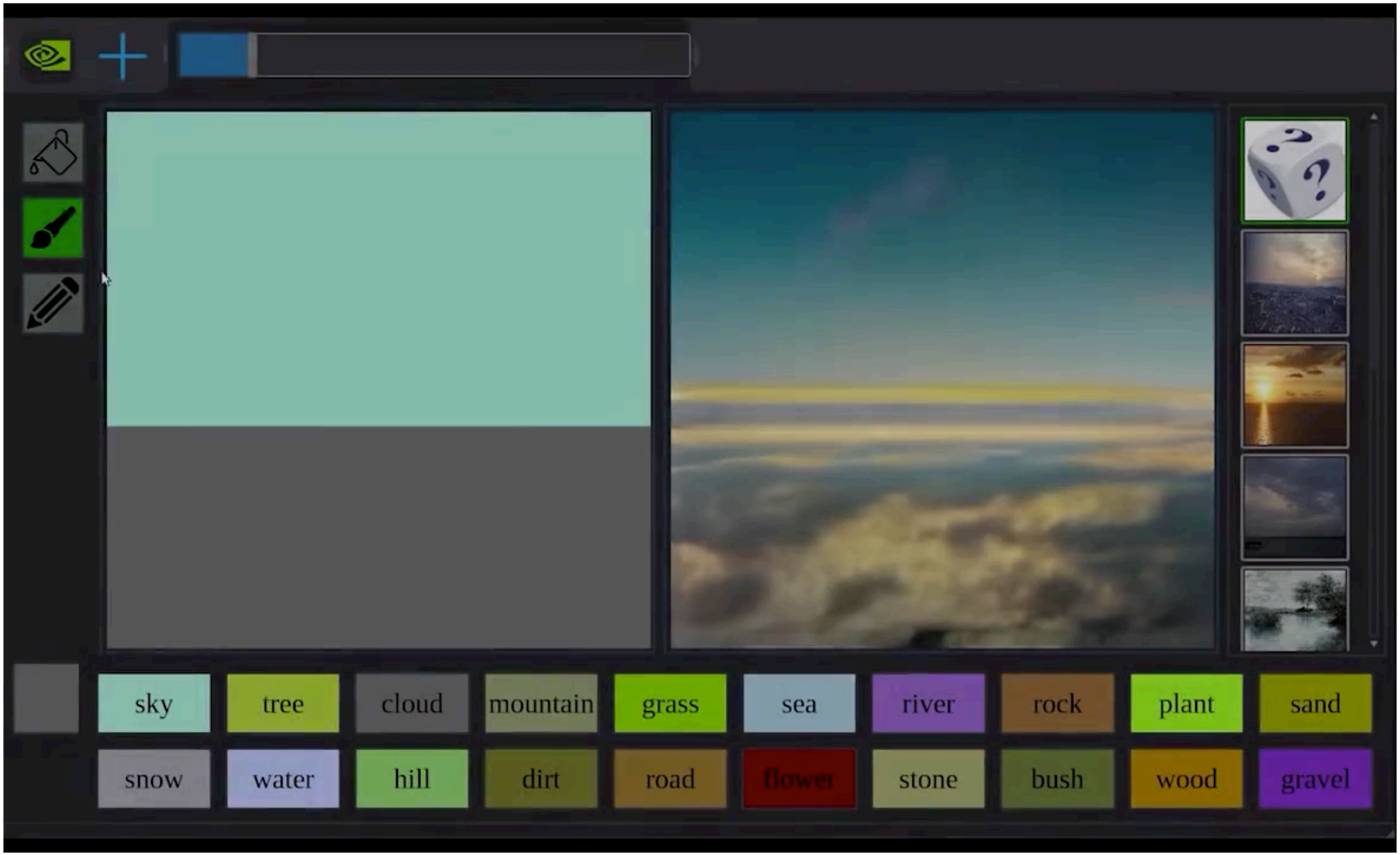
Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

Source: Isola, Freeman, Torralba



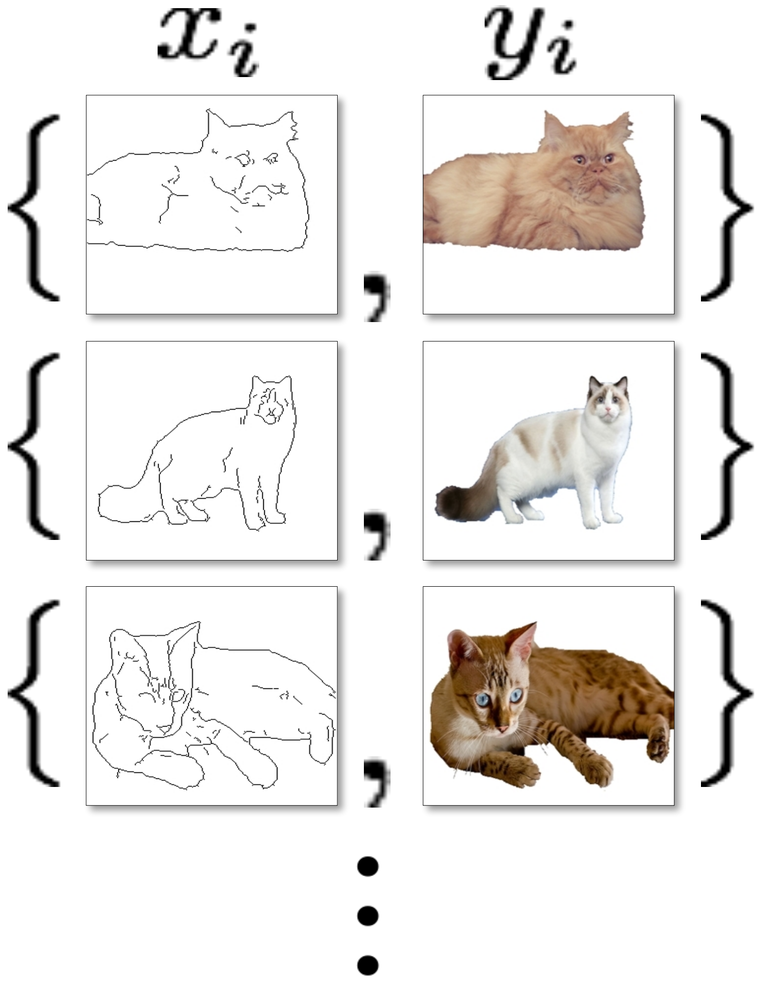
["GauGAN", Park et al., CVPR 2019]



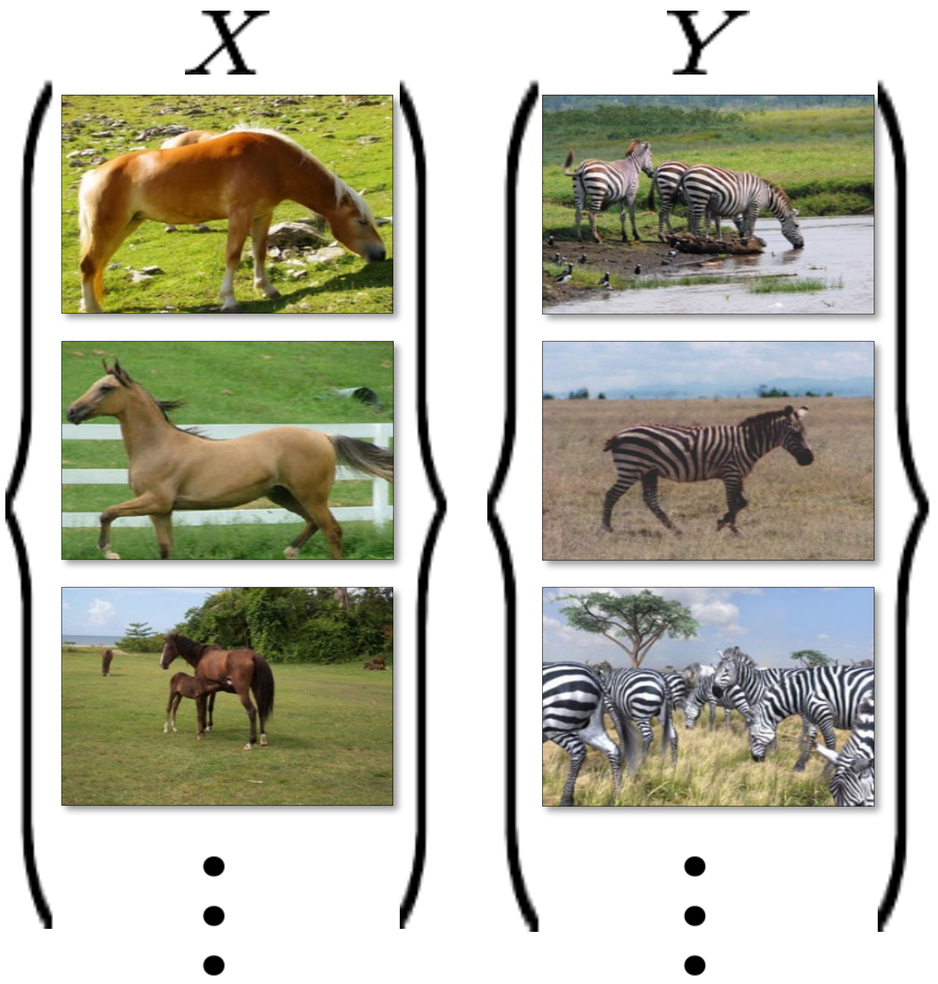
["GauGAN", Park et al., CVPR 2019]

Handling unpaired data

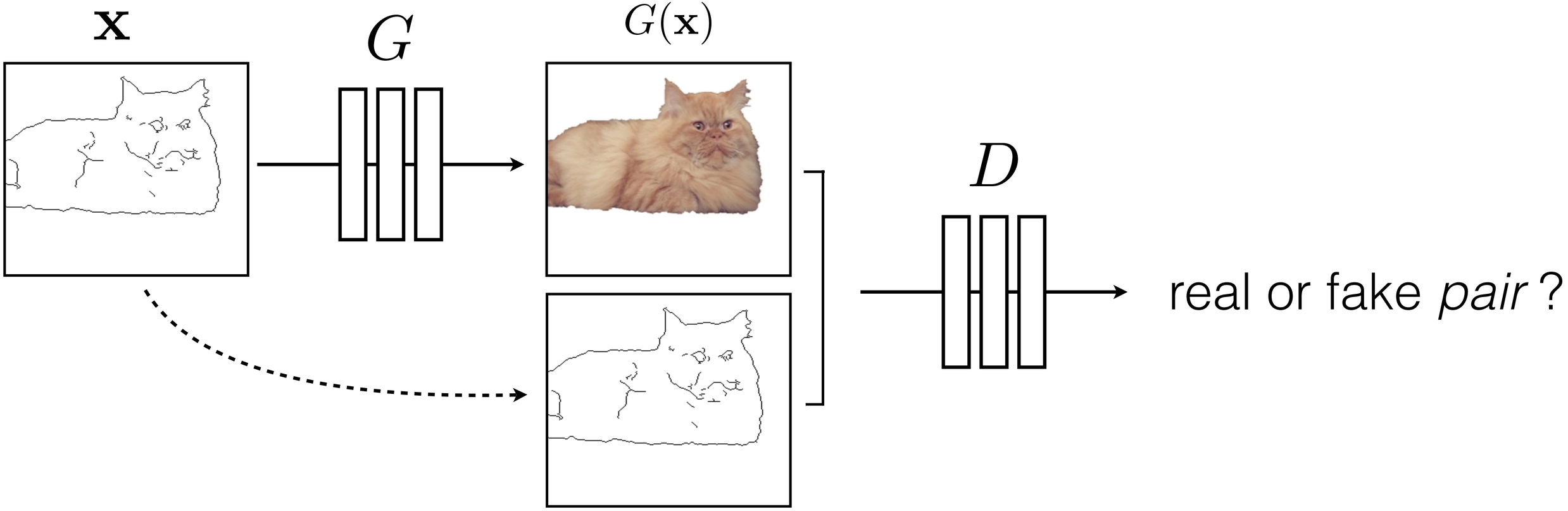
Paired data



Unpaired data

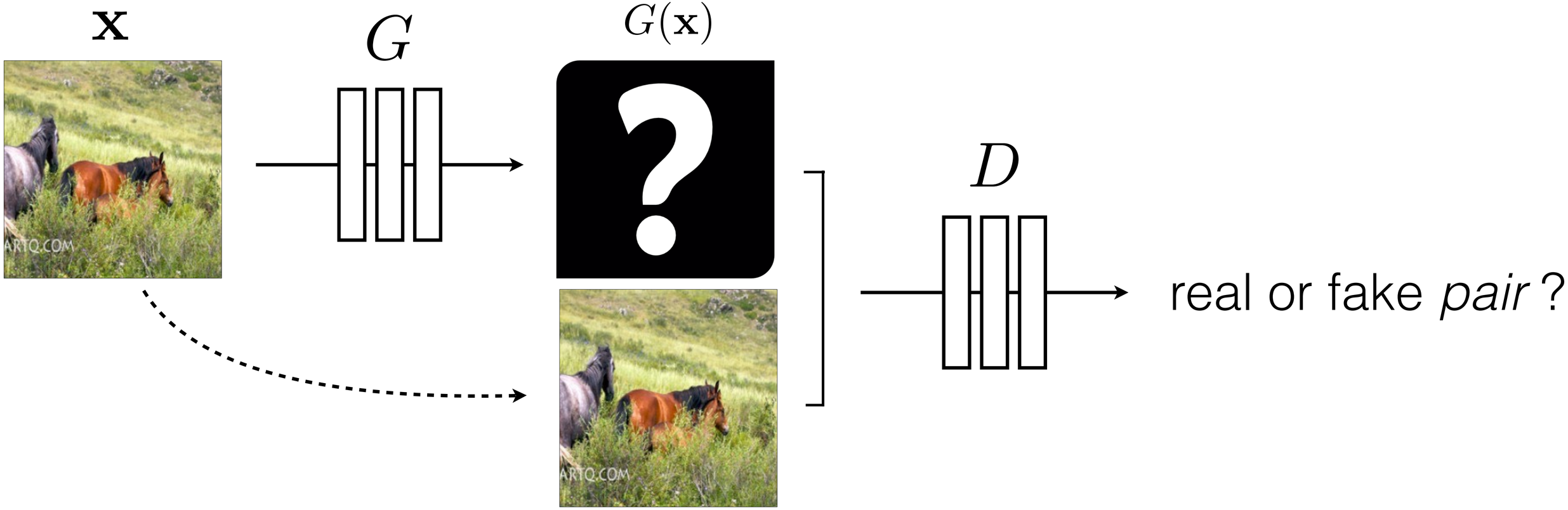


Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;\;]$$

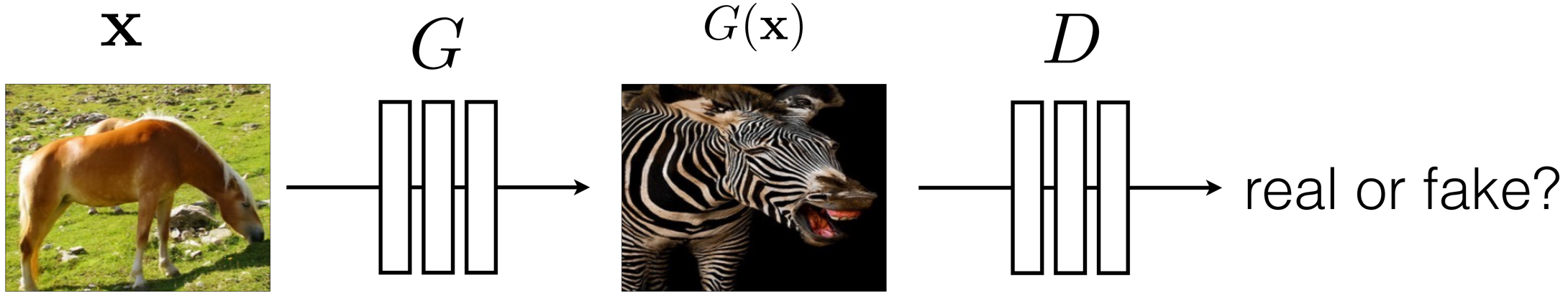
Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;\;]$$

No input-output pairs!

Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\;]$$

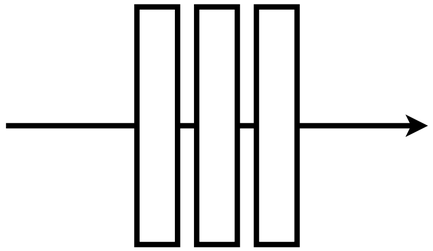
Usually loss functions check if output matches a target *instance*

GAN loss checks if output is part of an admissible *set*

Source: Isola, Freeman, Torralba

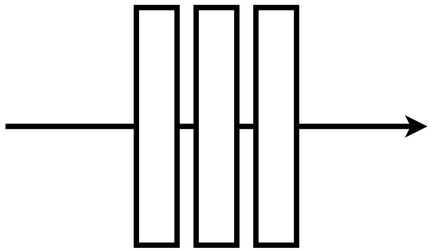
$\mathbf{x}$

G



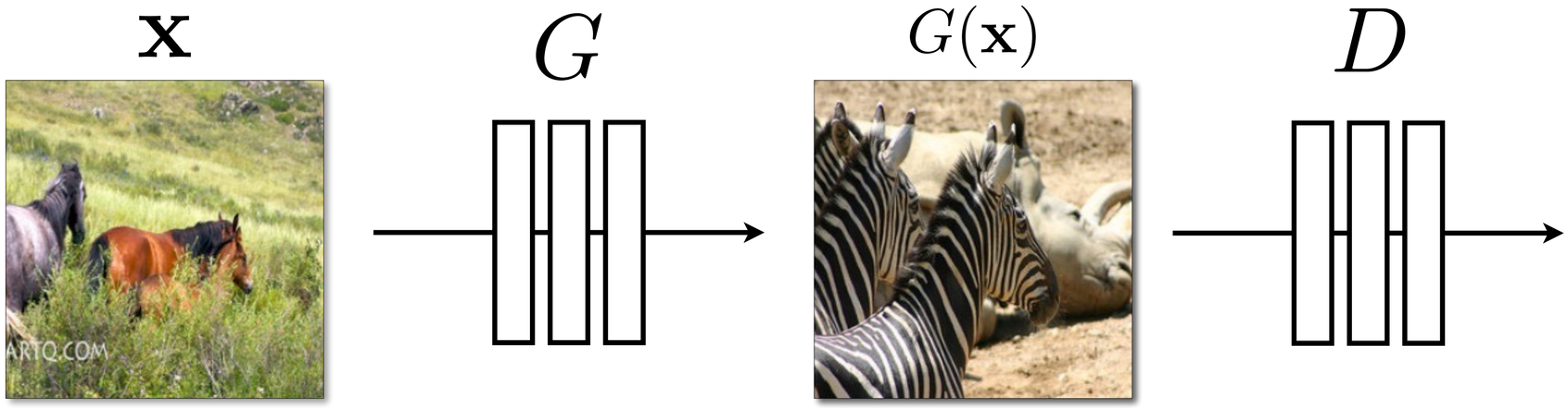
$G(\mathbf{x})$

D



Real!

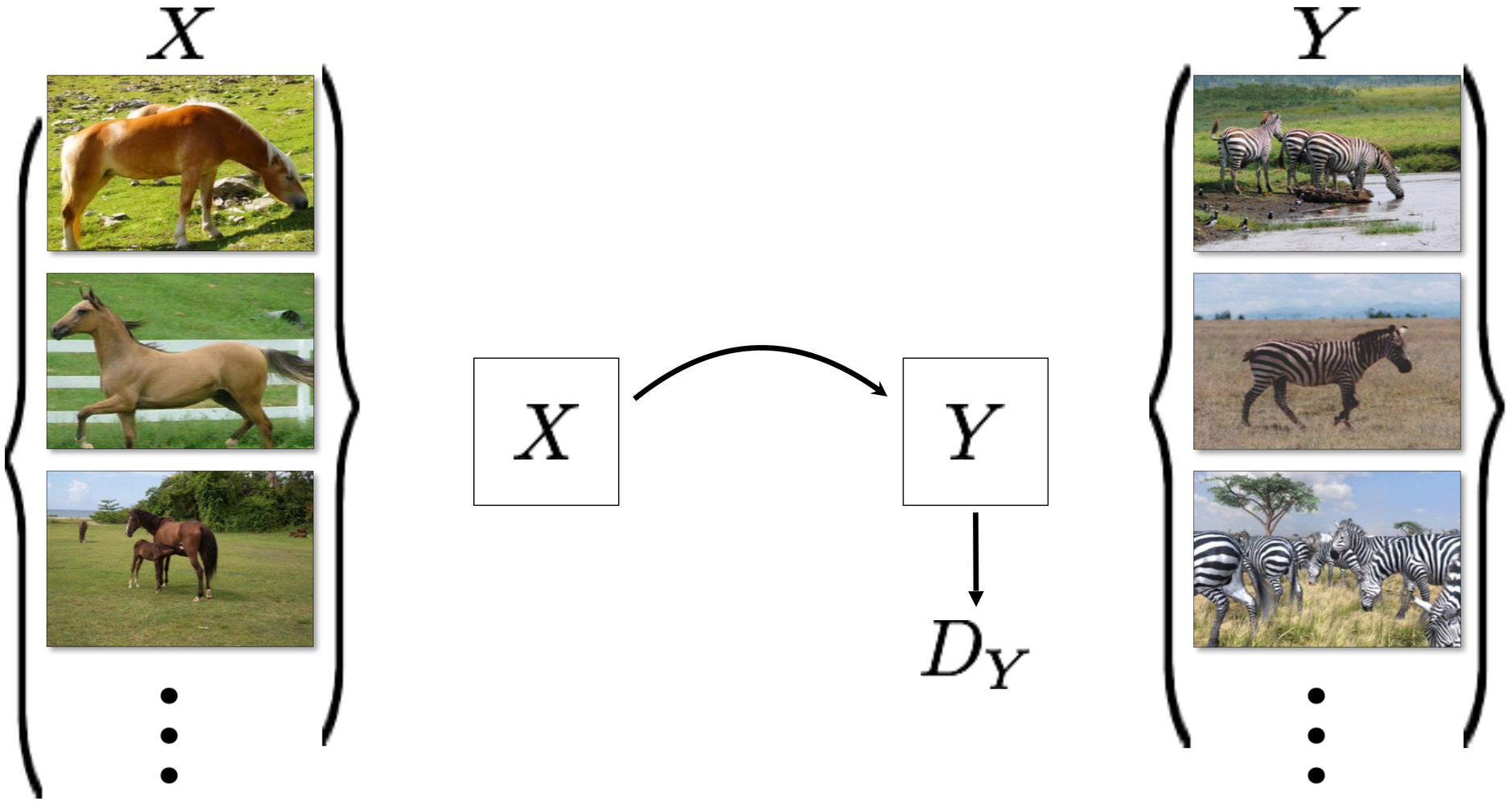
Source: Isola, Freeman, Torralba



Nothing to force output to correspond to input

Source: Isola, Freeman, Torralba

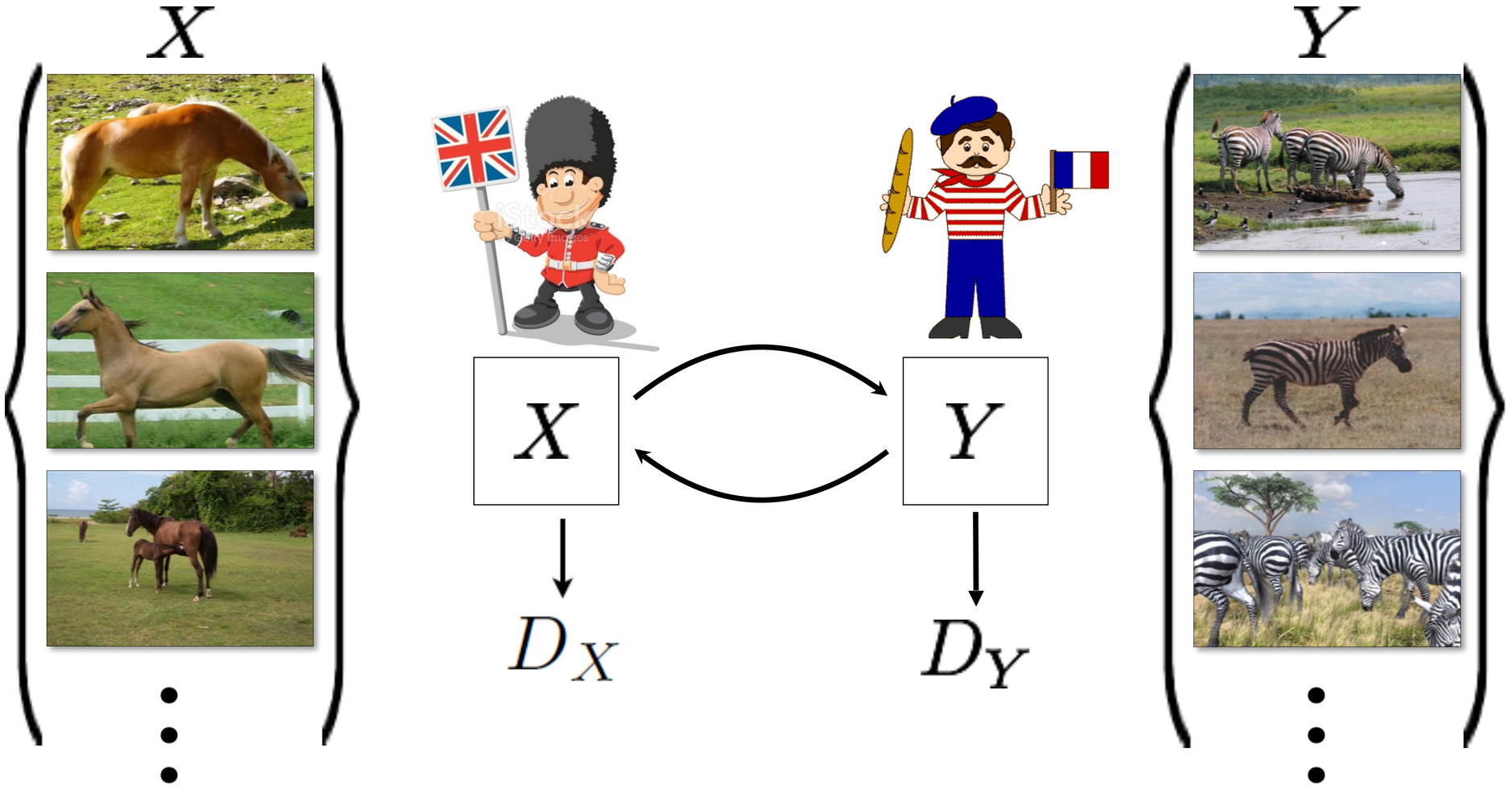
CycleGAN



[Zhu*, Park* et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

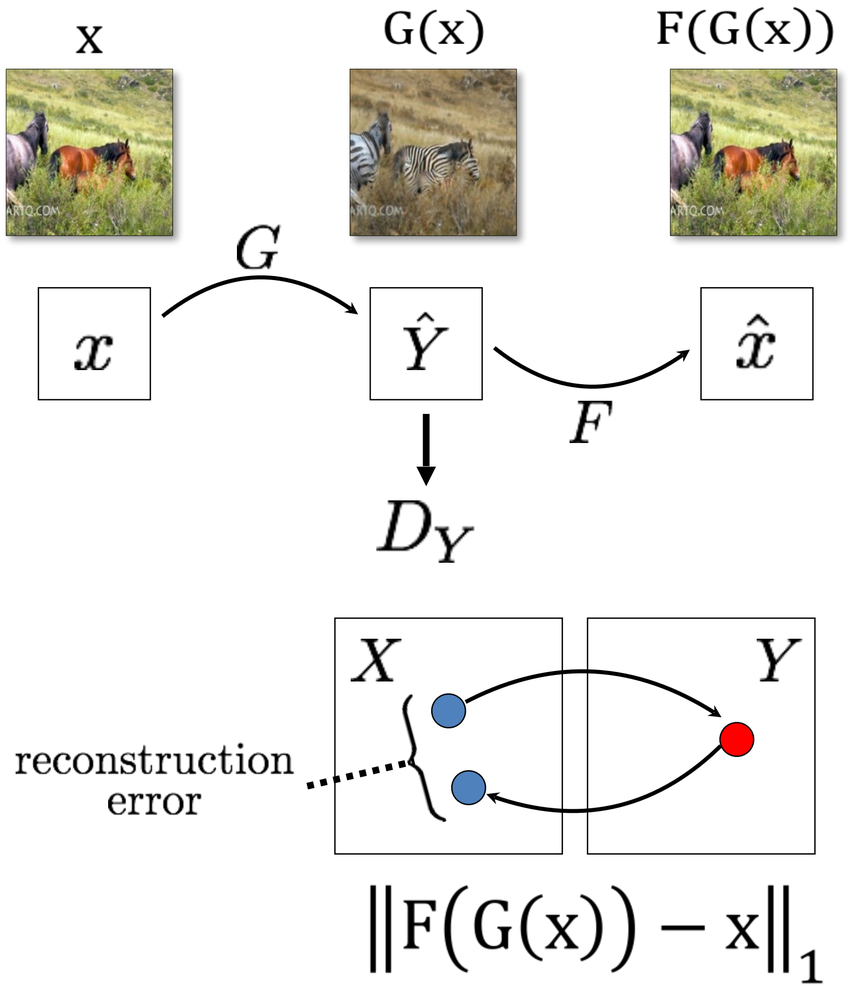
Source: Isola, Freeman, Torralba

CycleGAN



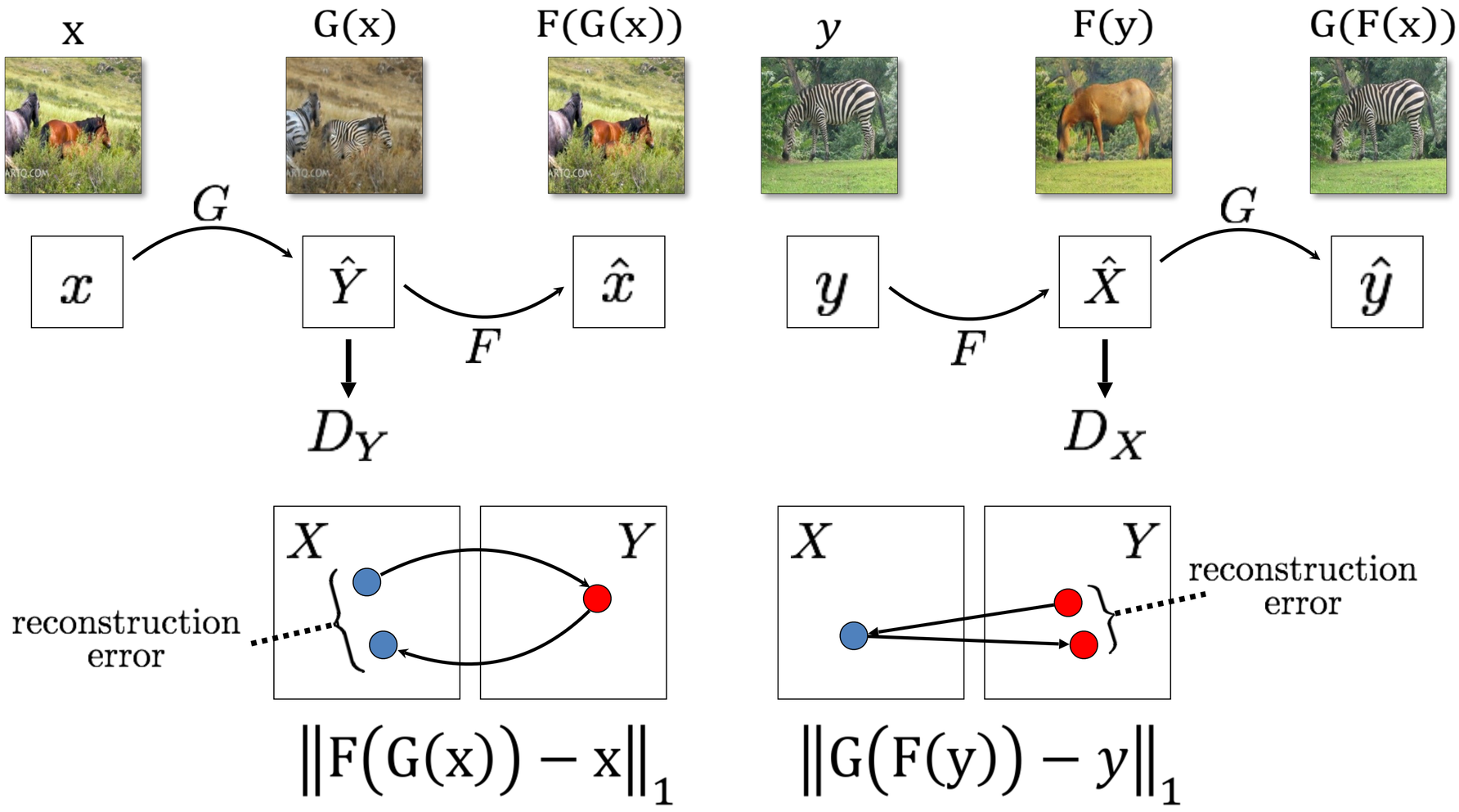
Source: Isola, Freeman, Torralba

Cycle Consistency Loss

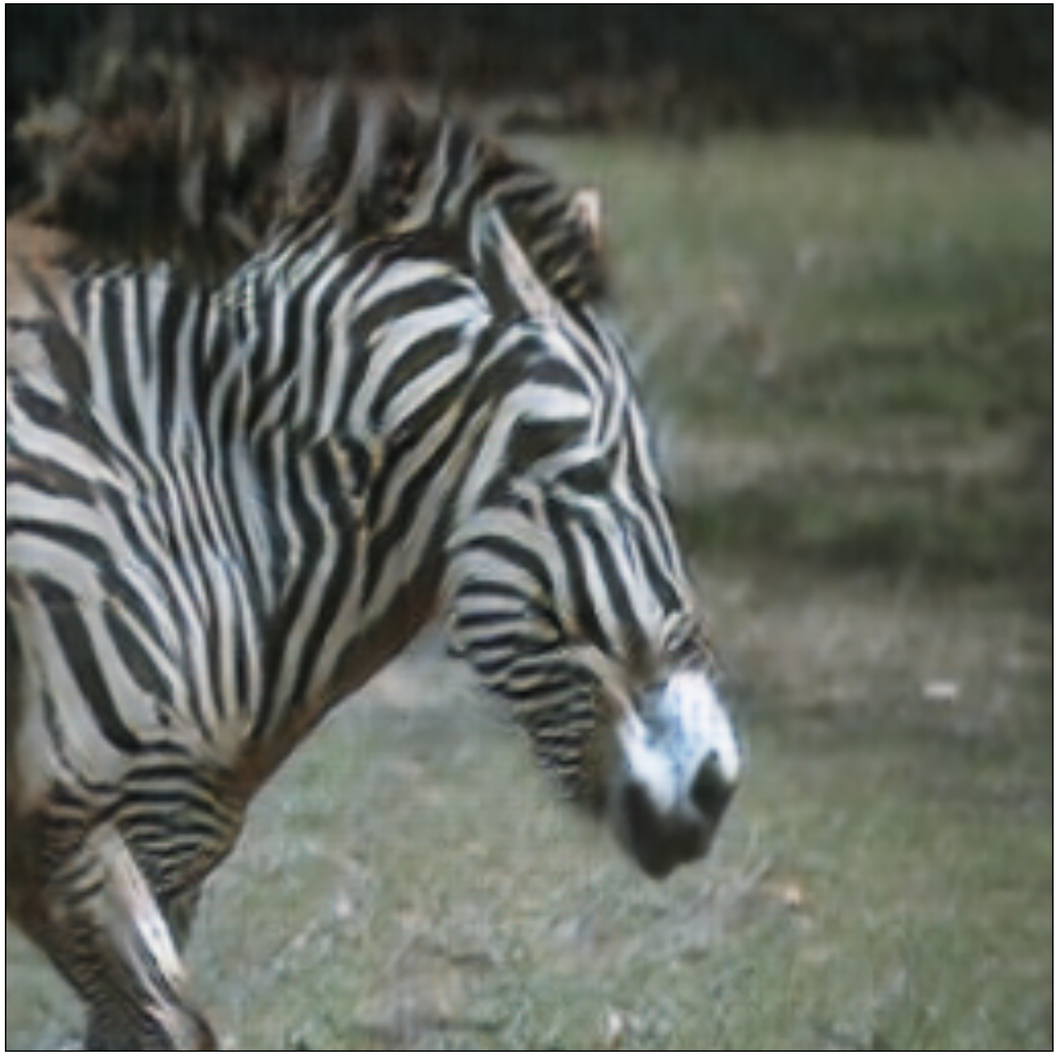


Source: Isola, Freeman, Torralba

Cycle Consistency Loss



Source: Isola, Freeman, Torralba



Source: Isola, Freeman, Torralba

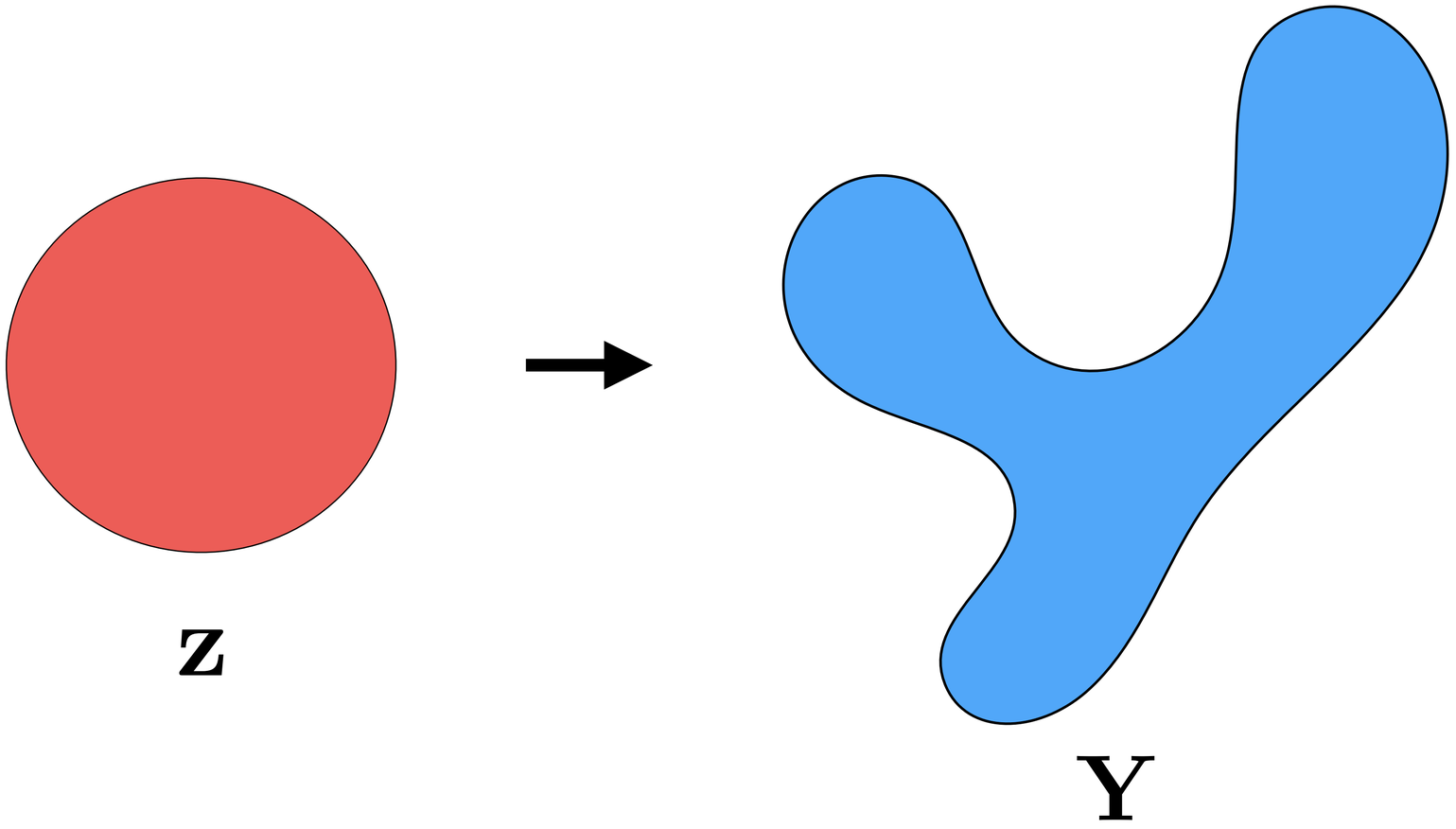
Source: Isola, Freeman, Torralba


Input
Monet
Van Gogh
Cezanne
Ukiyo-e

GANs

Gaussian

Target distribution

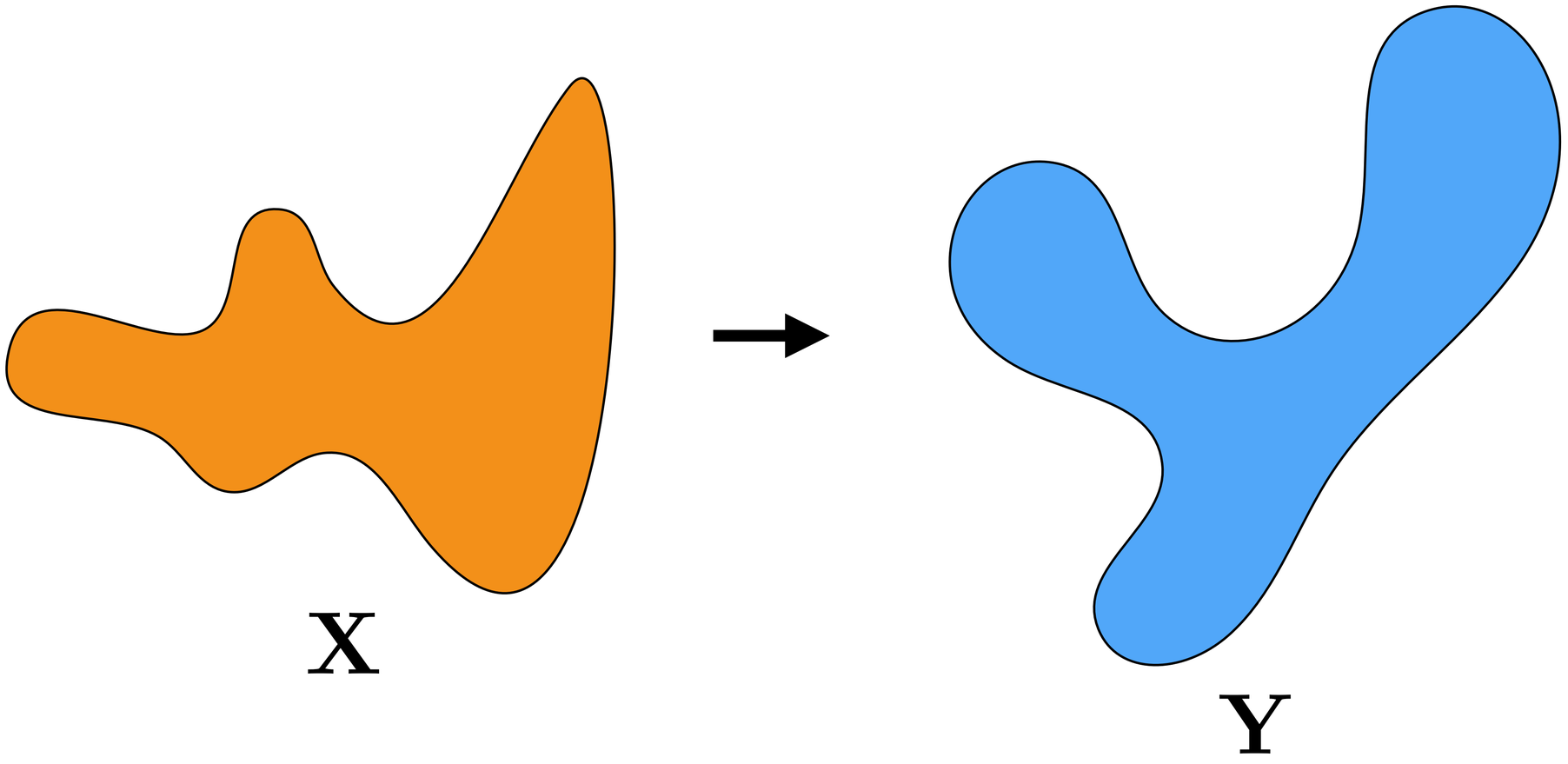


Source: Isola, Freeman, Torralba

CycleGAN

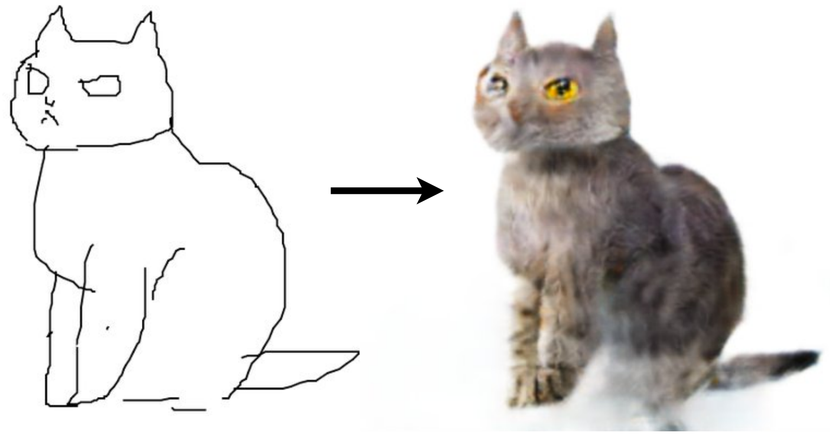
Horses

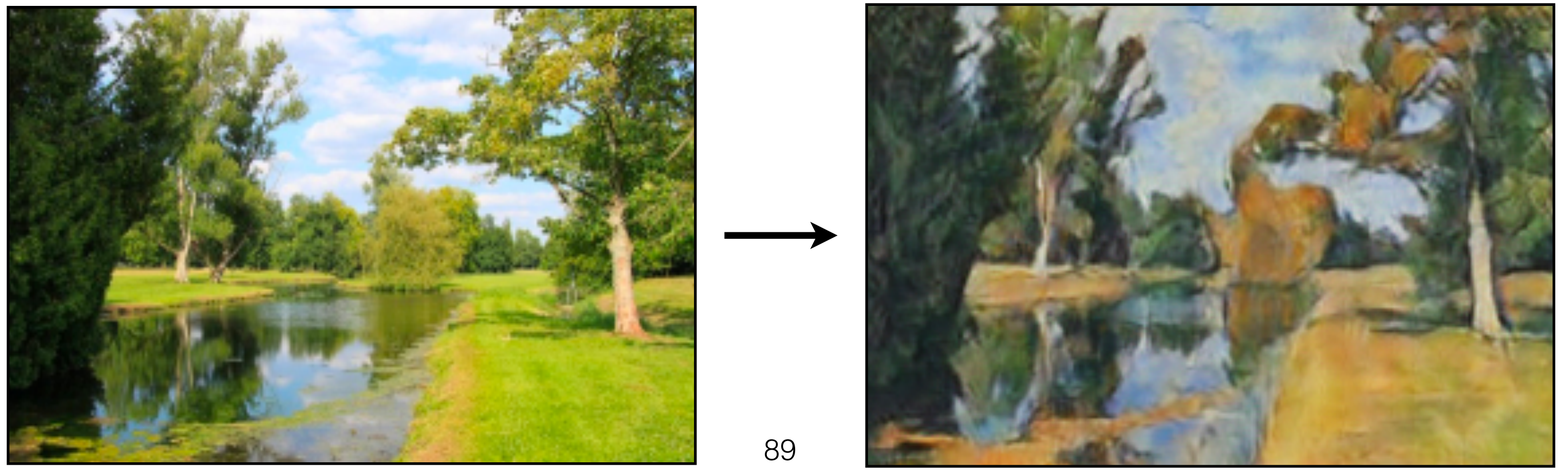
Zebras



Source: Isola, Freeman, Torralba

What would it look like if…?

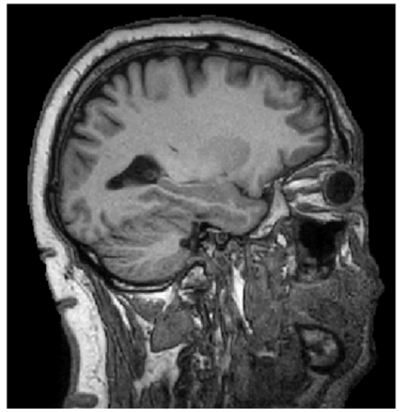




Source: Isola, Freeman, Torralba

What would it look like if…?

MRI



CT

[Wolterink et al, 2017]

Sim

“Real”

[Hoffman et al, 2018]

Source: Isola, Freeman, Torralba