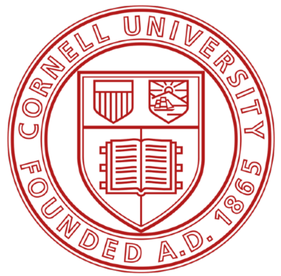
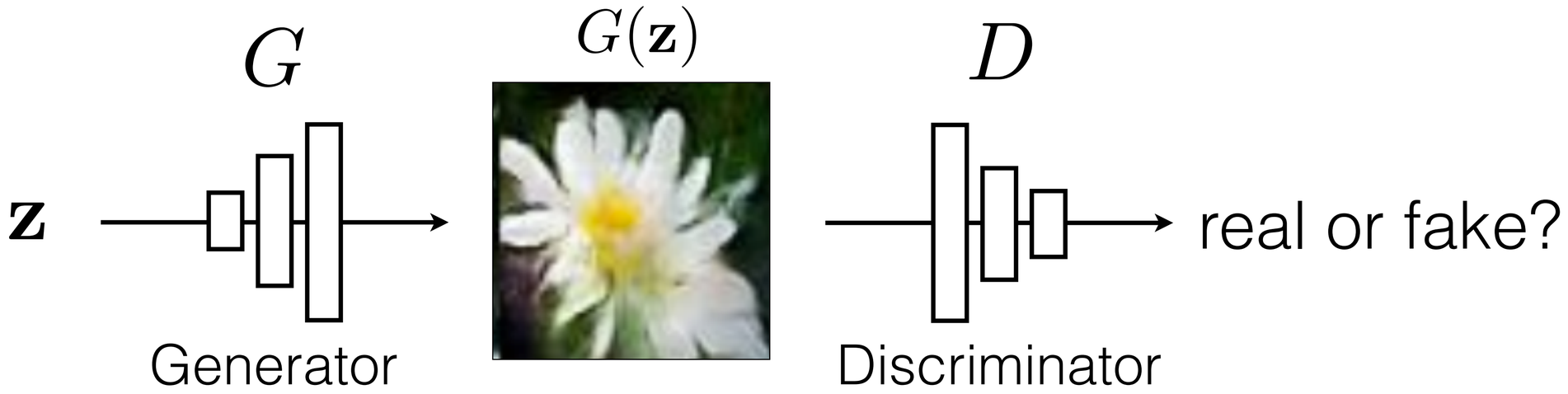
Lecture 8: GANs for conditional image manipulation

CS 5788: Introduction to Generative Models



Many slides from Phillip Isola

GAN review

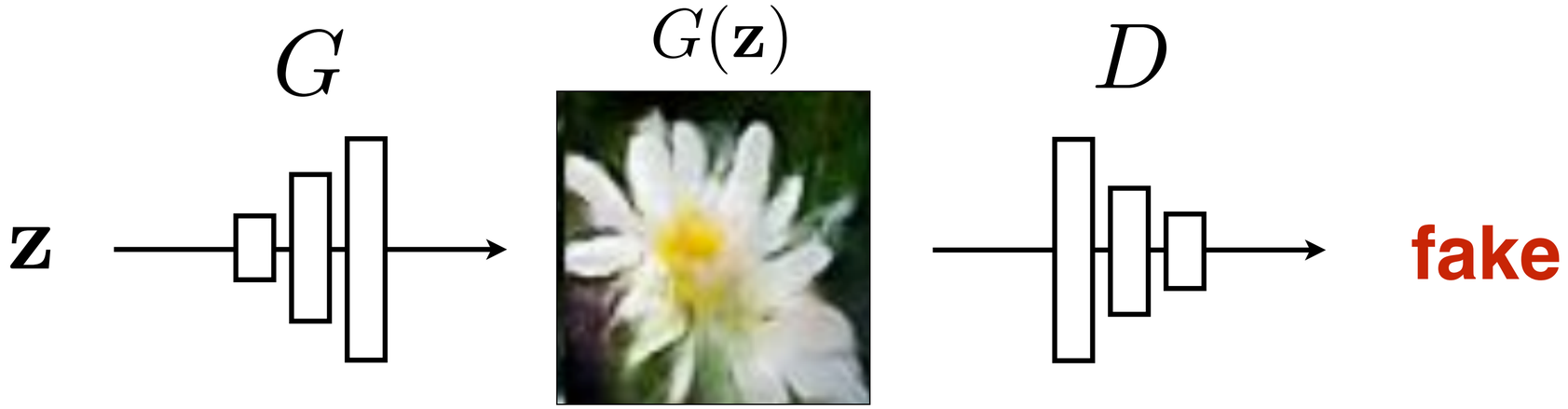


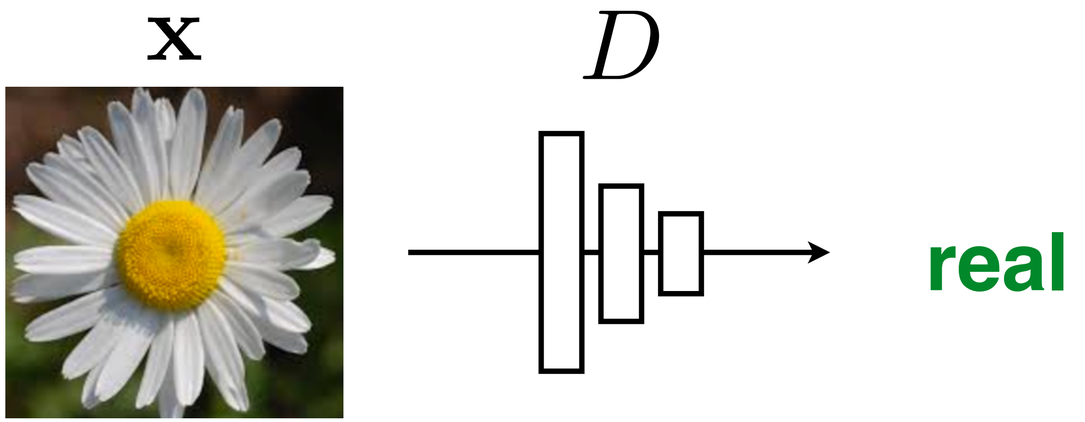
G tries to synthesize fake images that fool **D**

D tries to identify the fakes

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba



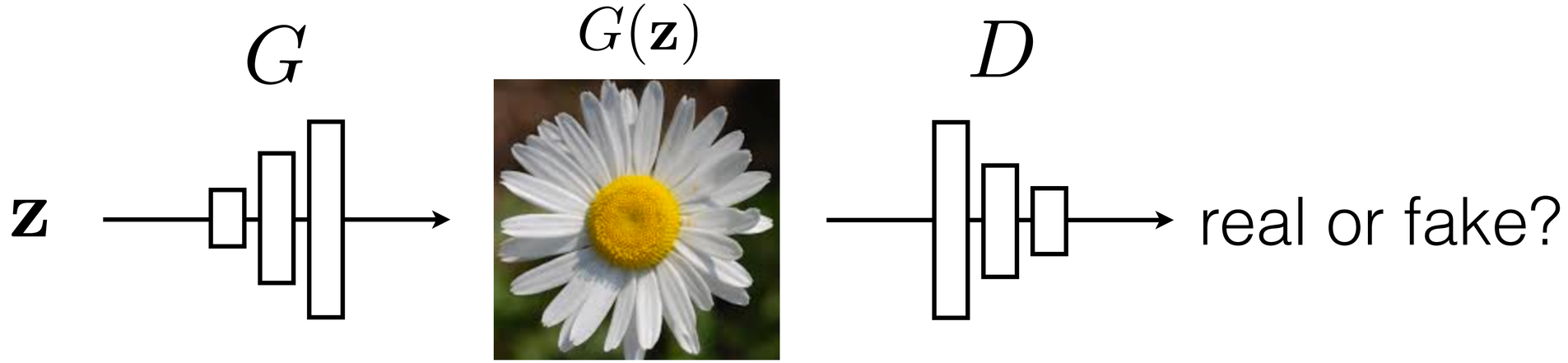


$$\arg\max_D \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \log D(G(\mathbf{z}))\ +\ \log\left(1 - D(\mathbf{x})\right)\ ]$$

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

Training



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

D tries to identify the fakes

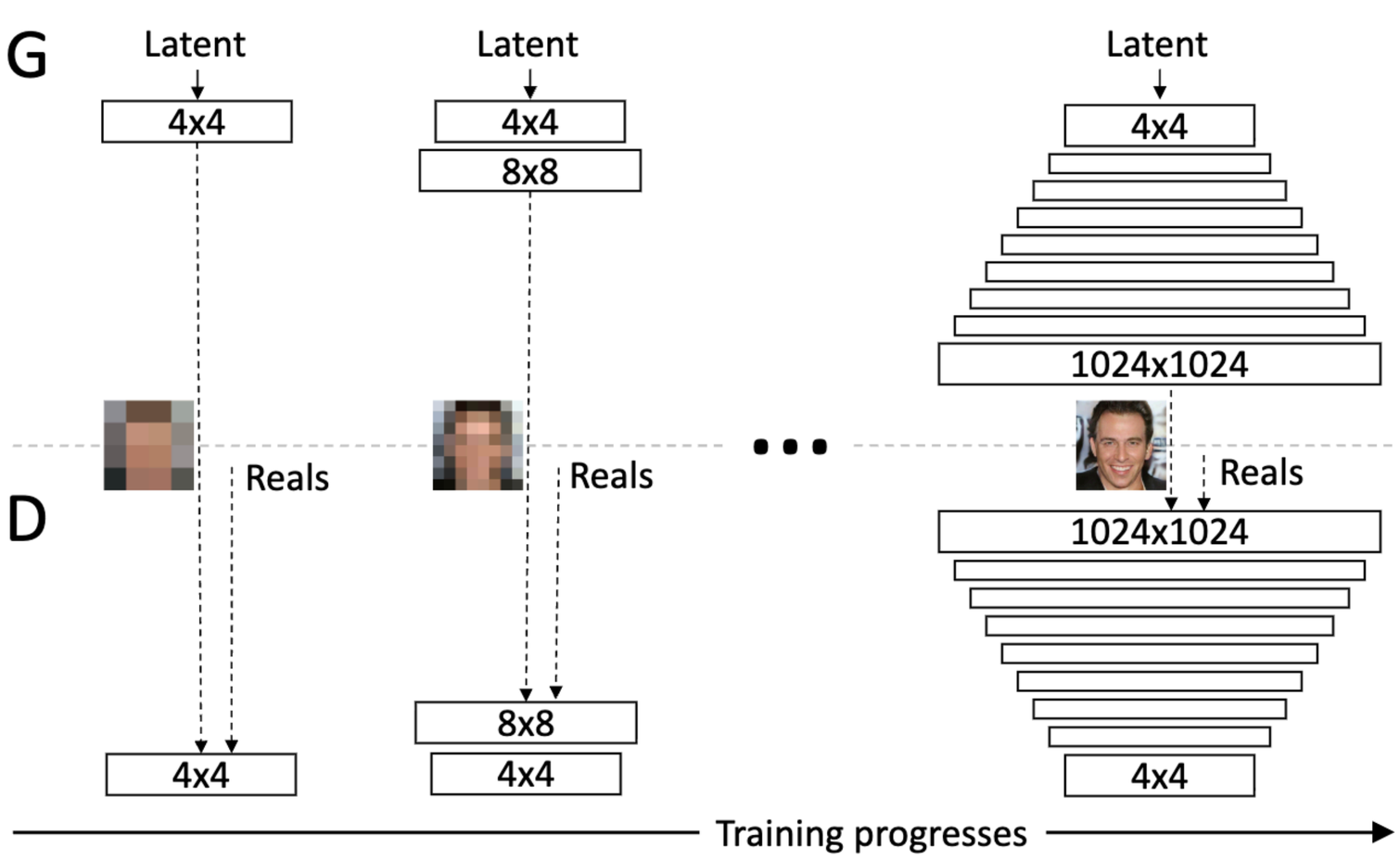
- Training: iterate between training D and G with backprop.
- Global optimum when G reproduces data distribution.

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

Image generation with GANs

Progressive generation

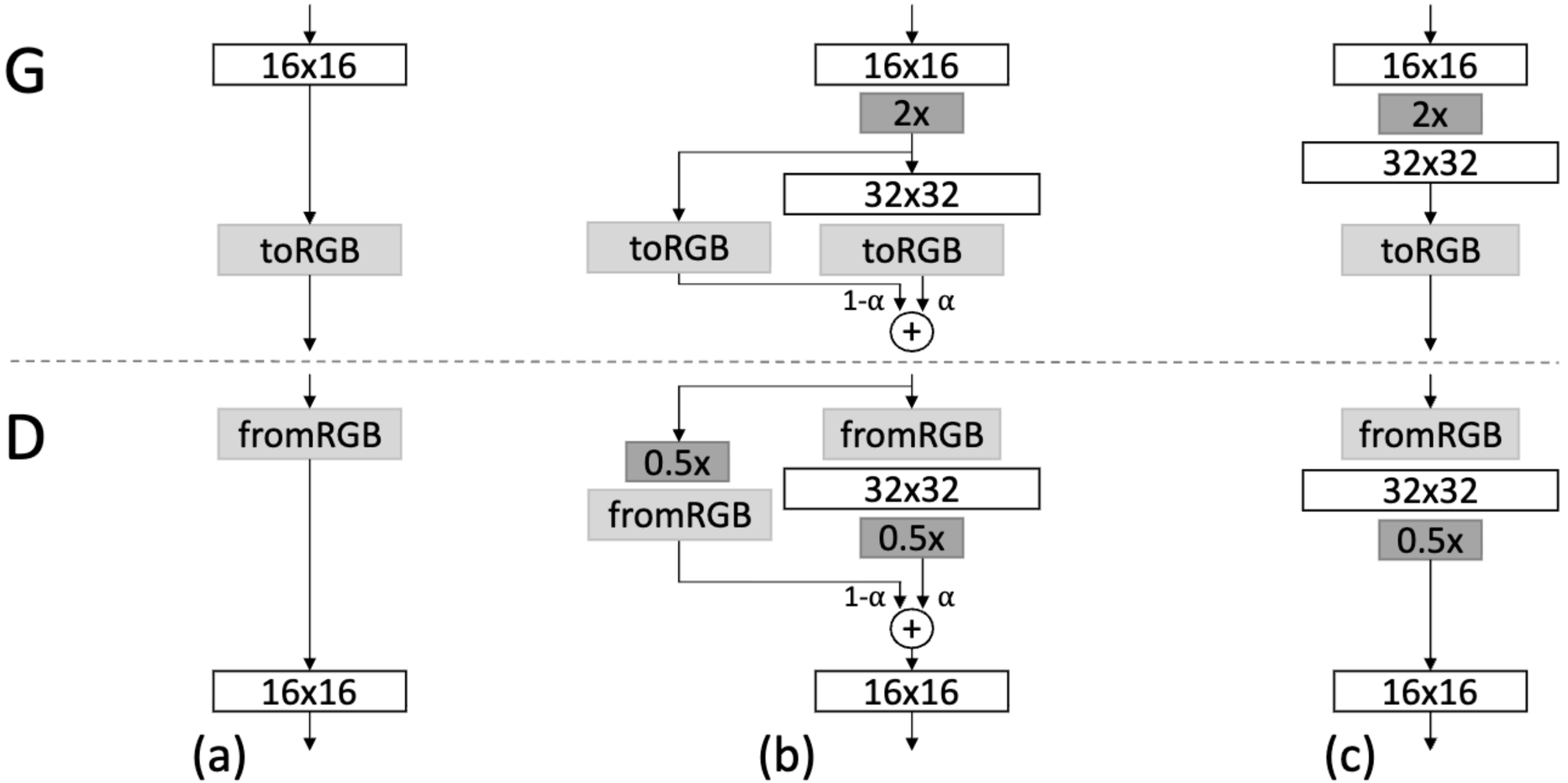


Train a small model, add more layers to G and D, repeat.

Speeds up training and deals with instabilities.

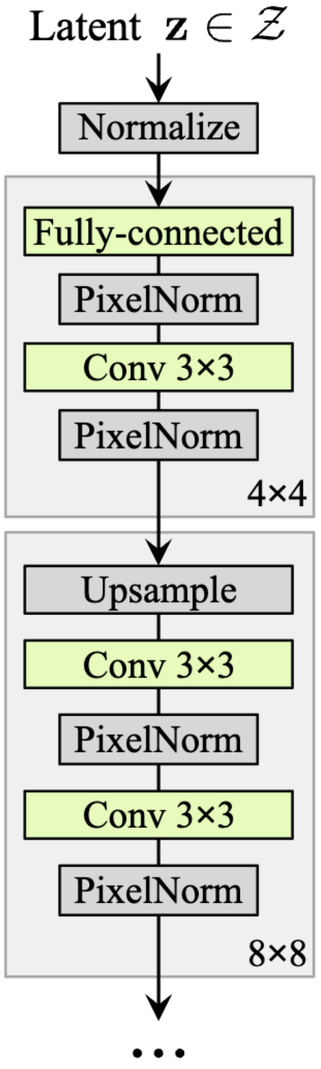
[Karras et al., "ProGAN", 2018]

Adding new layers

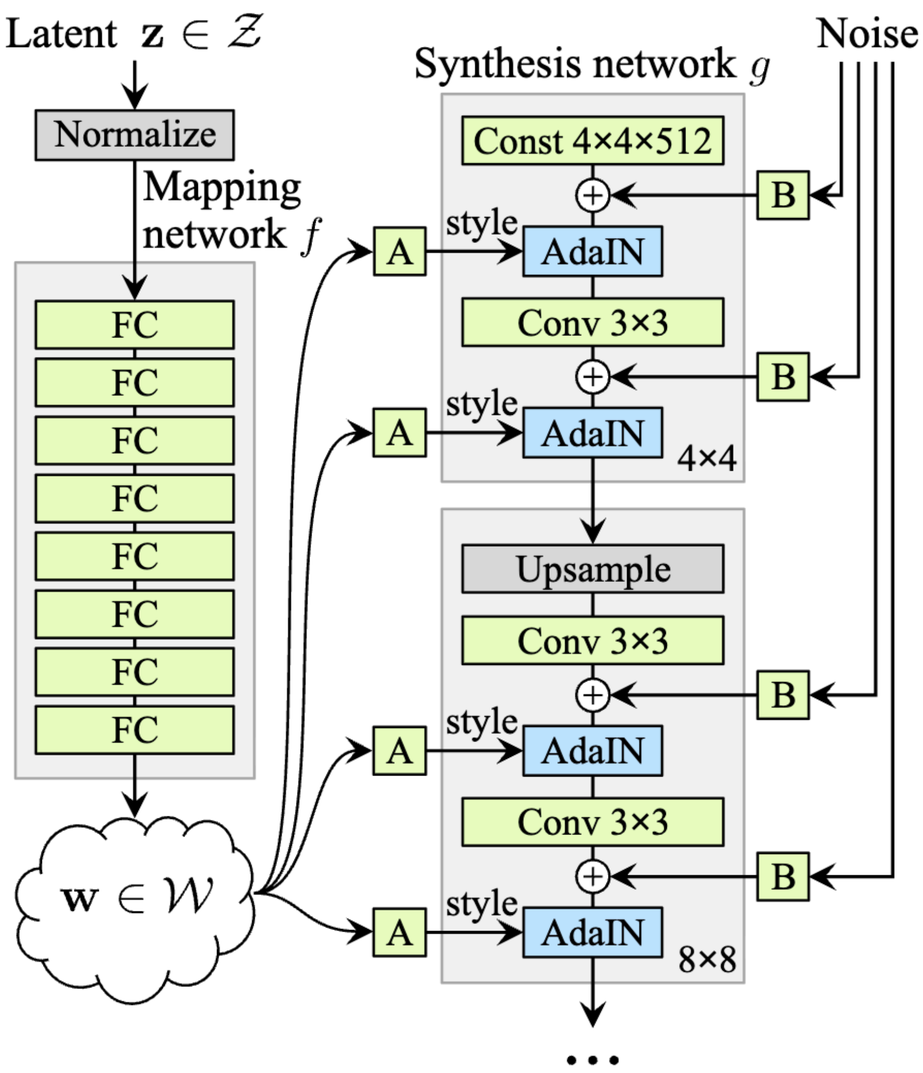


[Karras et al., "ProGAN", 2018]

Separating style and content



Traditional generator



Style-based generator

[Karras et al., "StyleGAN", 2019]

Copying coarse-scale style from source B to source A transfers pose, hair style, face shape.

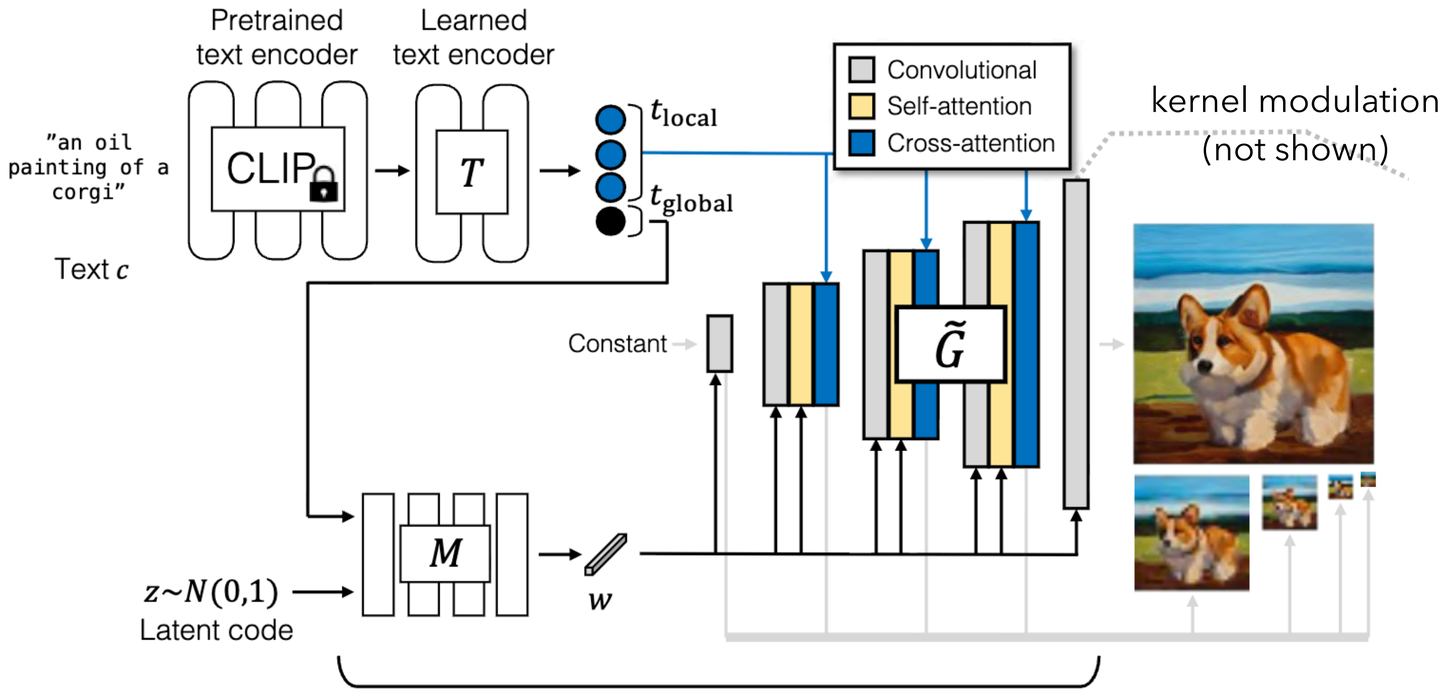
[Karras et al., “StyleGAN”, 2019]

Copying middle-scale style from source B to source A transfers fine-scale features.

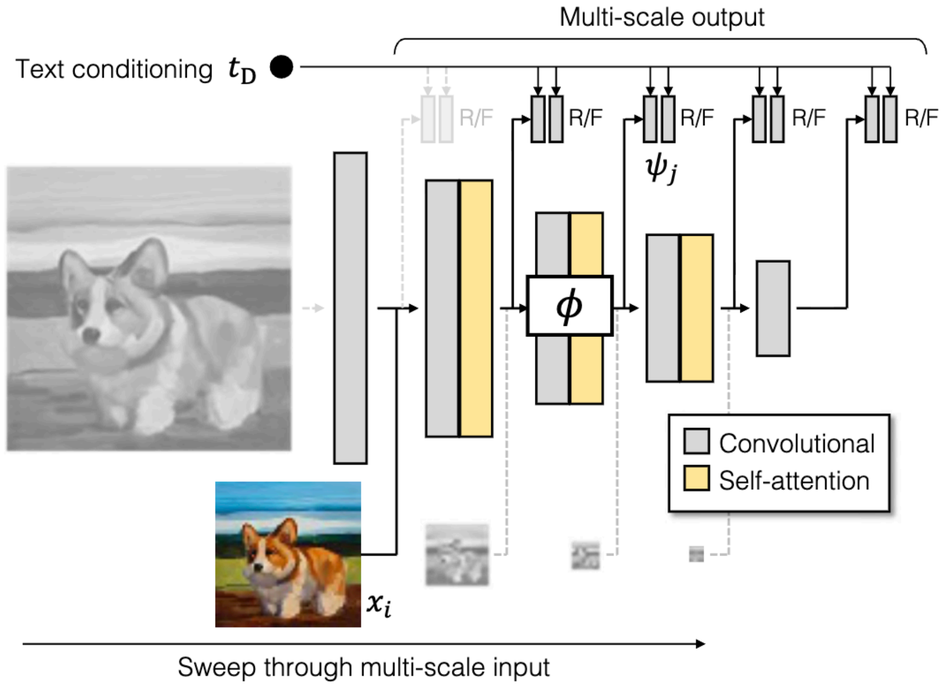
[Karras et al., “StyleGAN”, 2019]

Text-conditioned image generation

Generator



Discriminator



Multiscale loss: $$\mathcal{V}_{\text{MS-I/O}}(G, D) = \sum_{i=0}^{L-1} \sum_{j=i+1}^{L} \mathcal{V}_{\text{GAN}}(G_i, D_{ij}) + \mathcal{V}_{\text{match}}(G_i, D_{ij})$$

[Kang et al., "Scaling up GANs for Text-to-Image Synthesis", 2023]

Text-to-image GANs

A living room with a fireplace at a wood cabin. Interior design.

a blue Porsche 356 parked in front of a yellow brick wall.

Eiffel Tower, landscape photography

A painting of a majestic royal tall ship in Age of Discovery.

Isometric underwater Atlantis city with a Greek temple in a bubble.

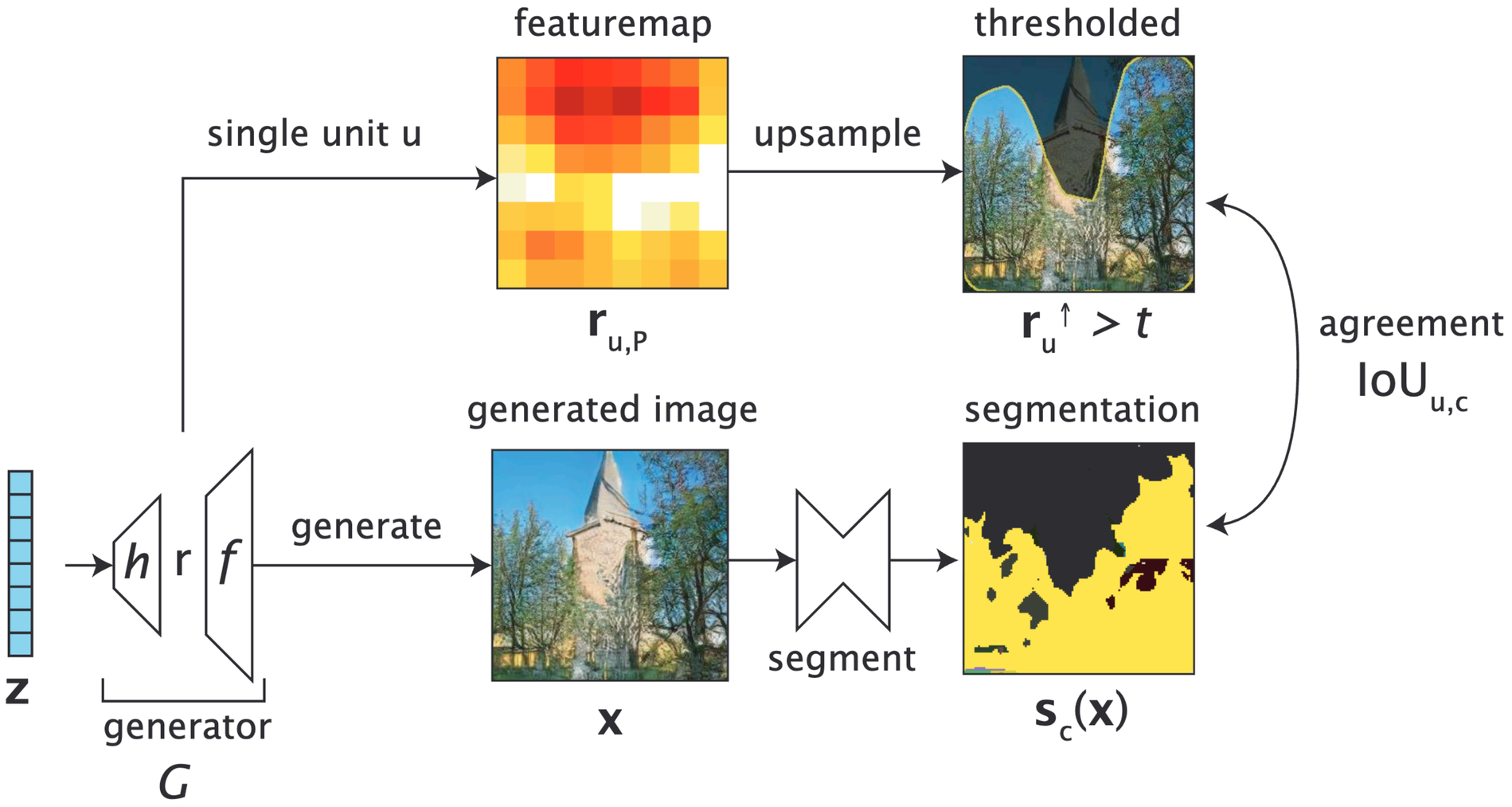
A hot air balloon in shape of a heart. Grand Canyon

low poly bunny with cute eyes

A cube made of denim on a wooden table

[Kang et al., “Scaling up GANs for Text-to-Image Synthesis”, 2023]

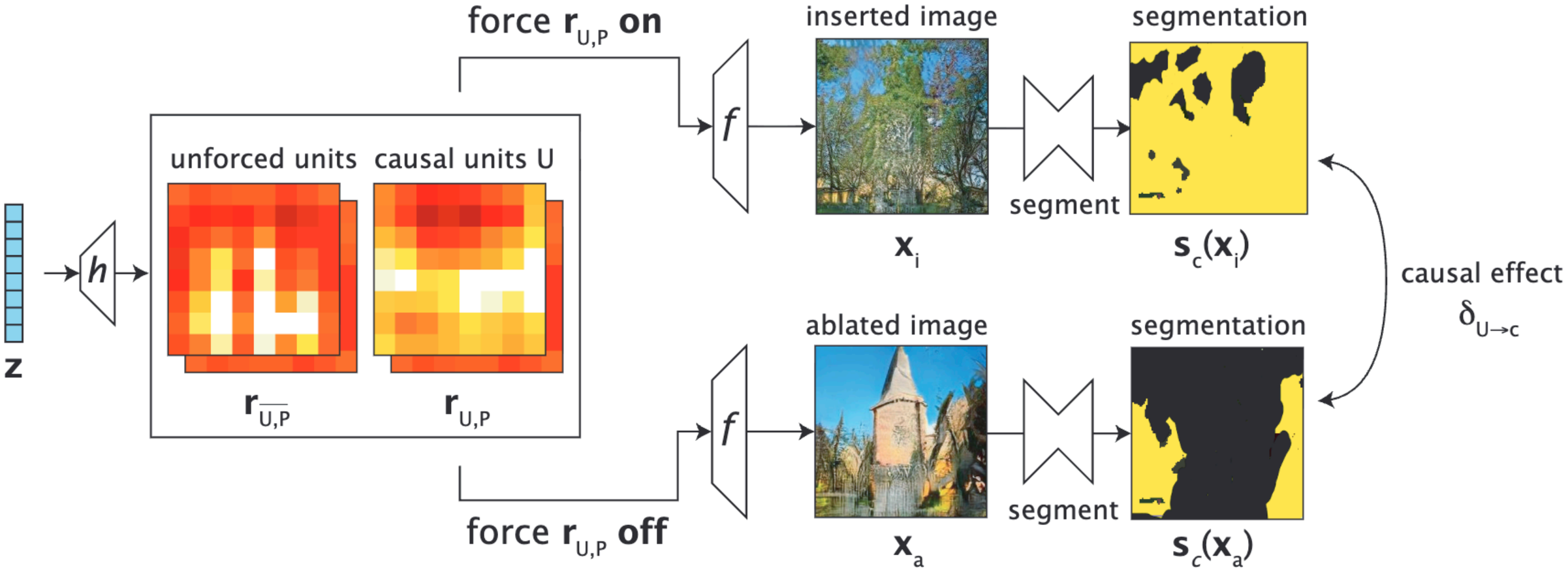
What are the features that a GAN learns?



There are CNN units whose activations correlate with object categories, like trees.

[Bau et al., "GAN Dissection", 2018]

Manipulating images by editing a network's feature representation



[Bau et al., "GAN Dissection", 2018]

Generate images of churches

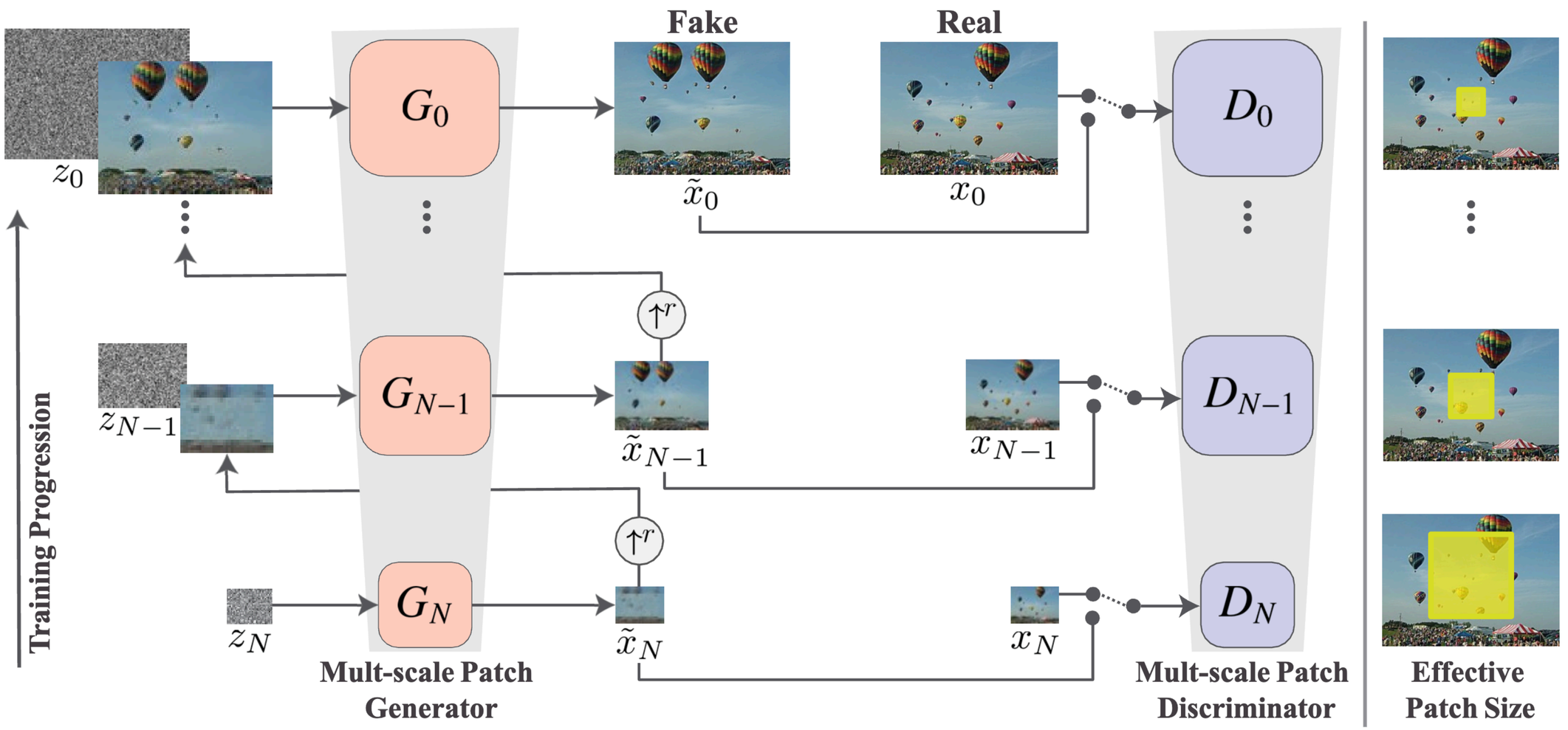
[Bau et al., “GAN Dissection”, 2018]

Interactive Editing with GANPaint

Do they “internal statistics” of an image give us useful information?

Figure source: [Zontak and Irani, “Internal Statistics of a Single Natural Image”, 2011]

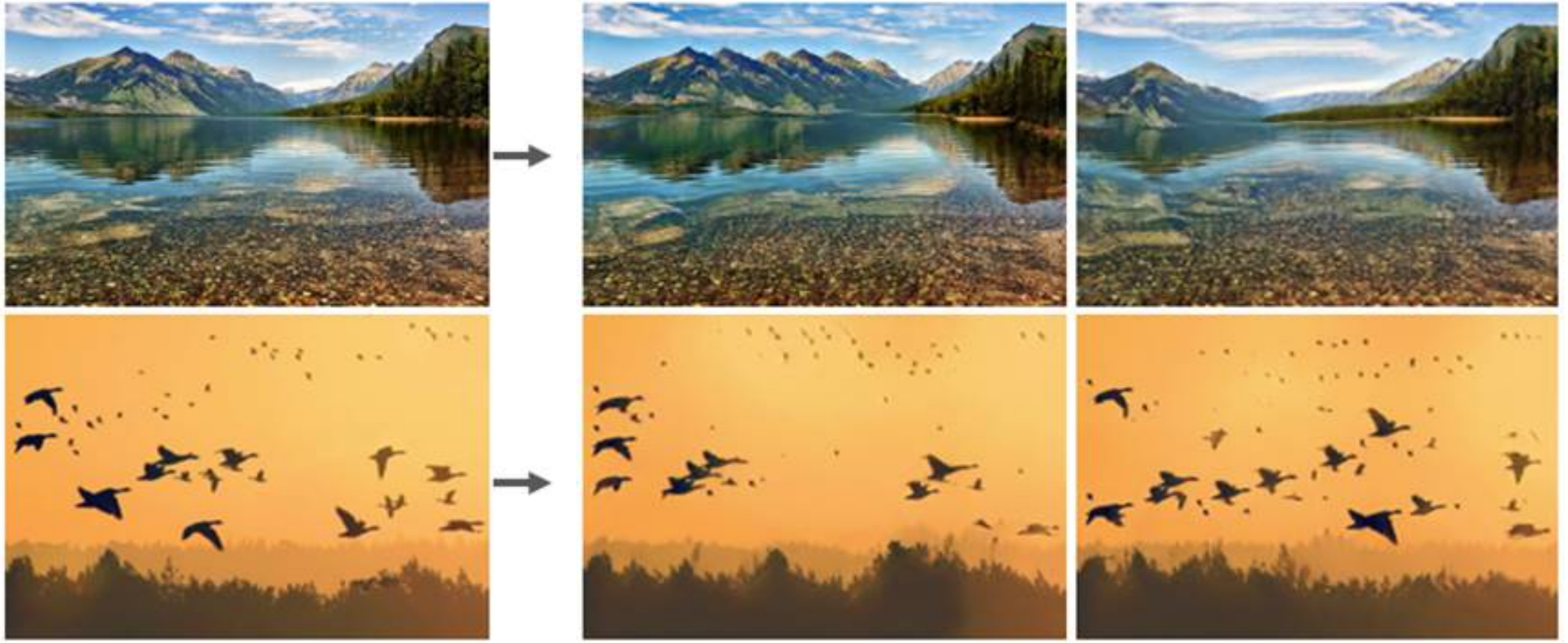
Single-image GAN



Do they "internal statistics" of an image give us useful information?

[Shaham et al., "SinGAN", 2019]

Single-image GAN



Training image

Random samples from GAN trained on single image

[Shaham et al., “SinGAN”, 2019]

Single-image GAN

Training image

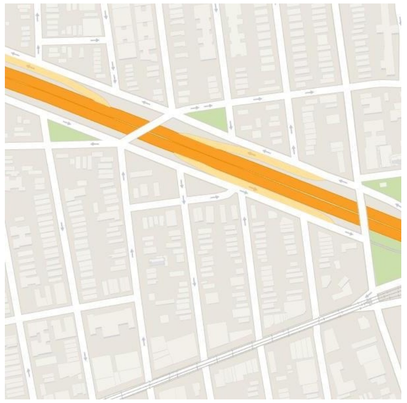
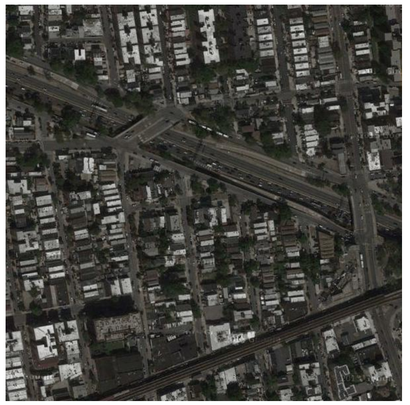
Random samples from GAN trained on single image (different aspect ratios)

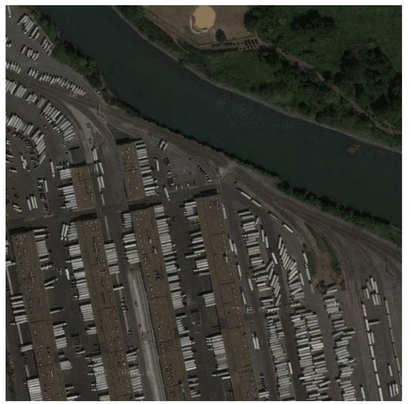
[Shaham et al., “SinGAN”, 2019]

Conditional generation

Map2Sat

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

$$\{ \text{  } , \text{  } \}$$

$$\{ \text{  } , \text{  } \}$$

$\mathbf{x}$

G

Generator

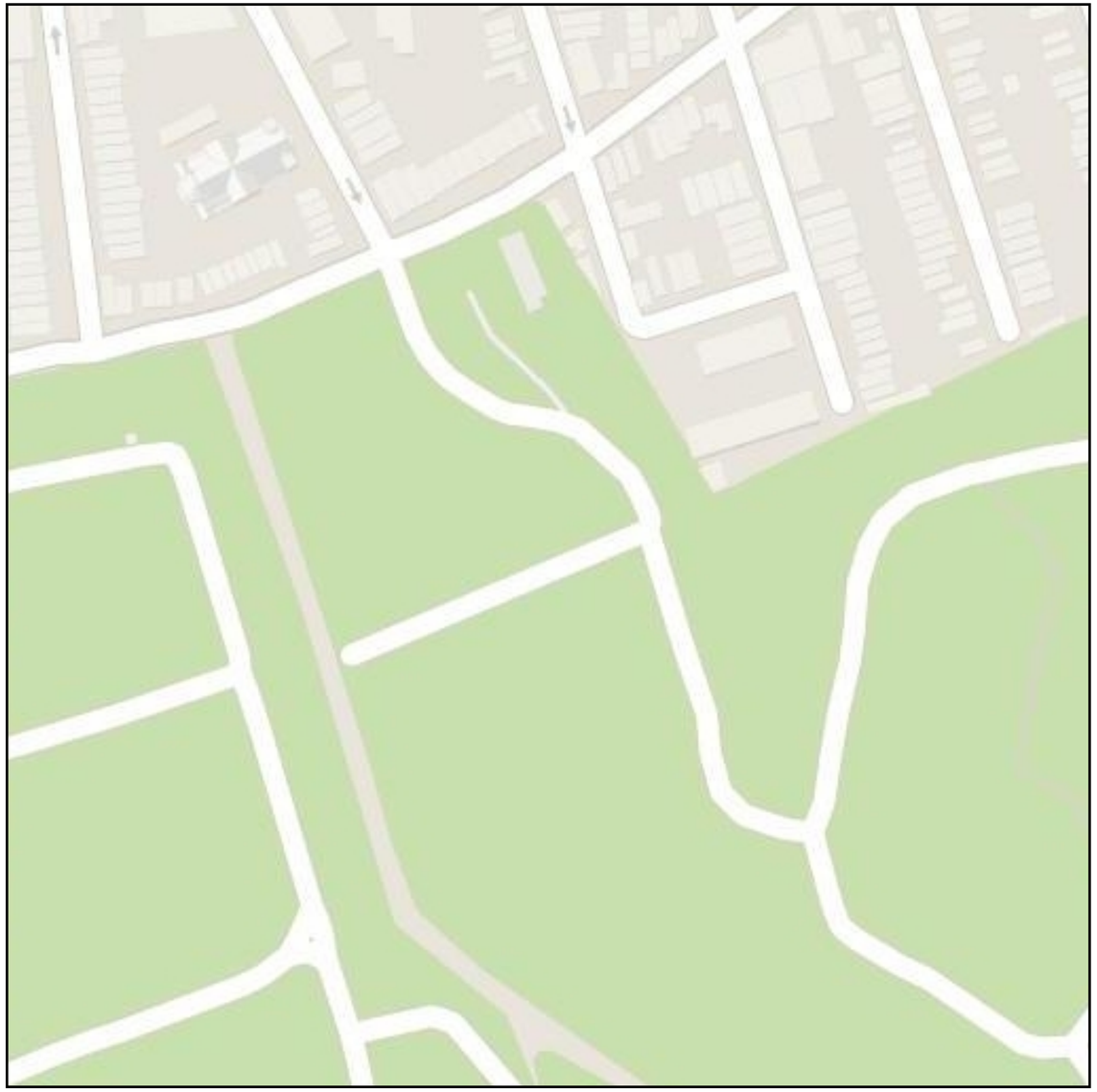
$G(\mathbf{x})$

Idea: L1 loss

$$\|G(\mathbf{x}) - \mathbf{y}\|_1$$

Source: Isola, Freeman, Torralba

Input

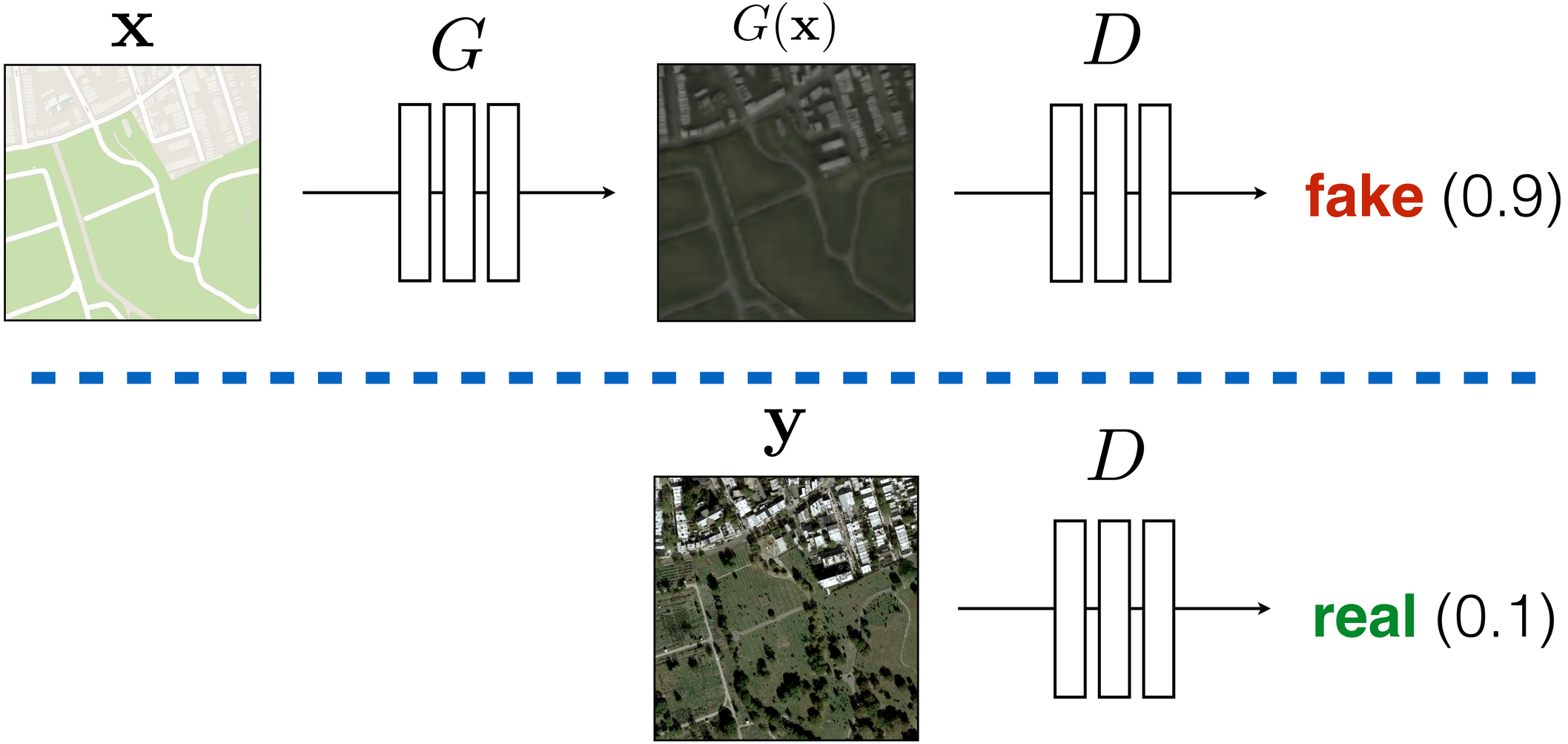


L1 loss

Source: Isola, Freeman, Torralba

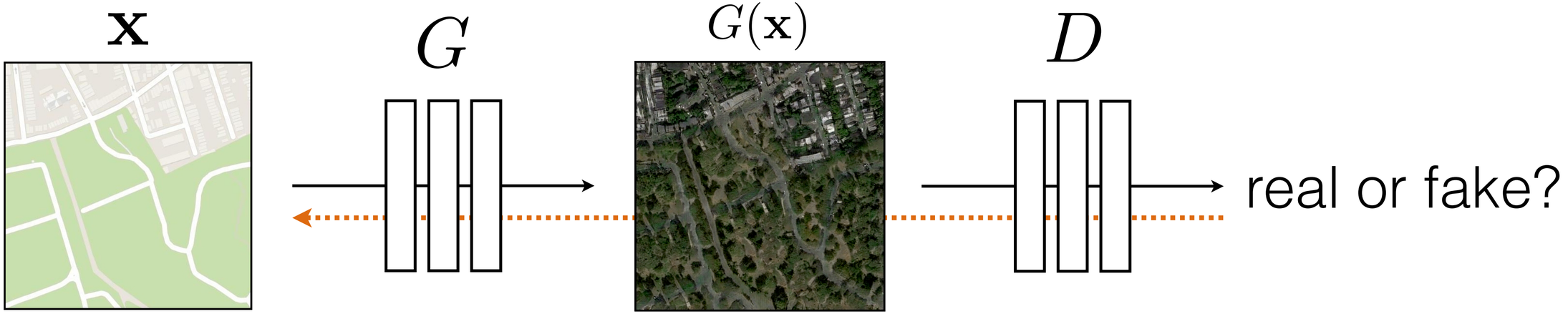
G tries to synthesize fake images that fool **D**

D tries to identify the fakes



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

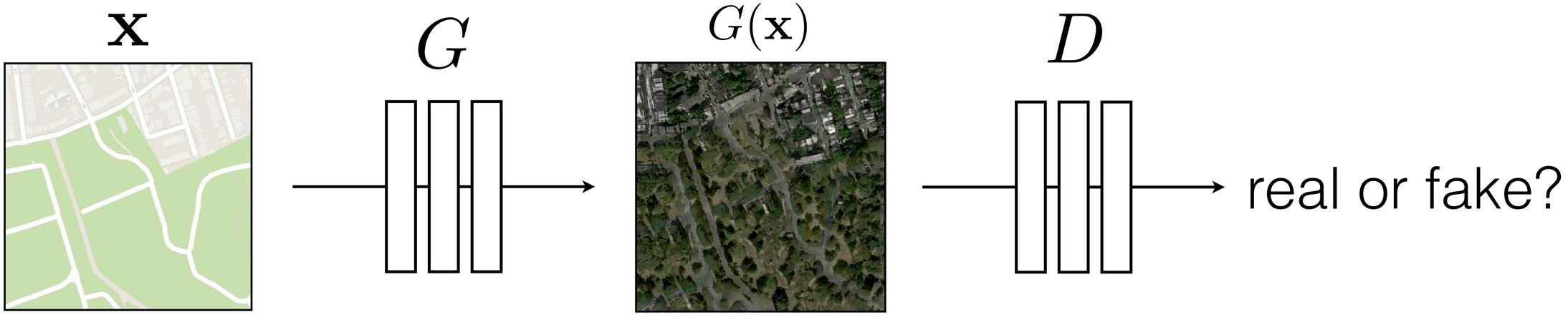
Source: Isola, Freeman, Torralba



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

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

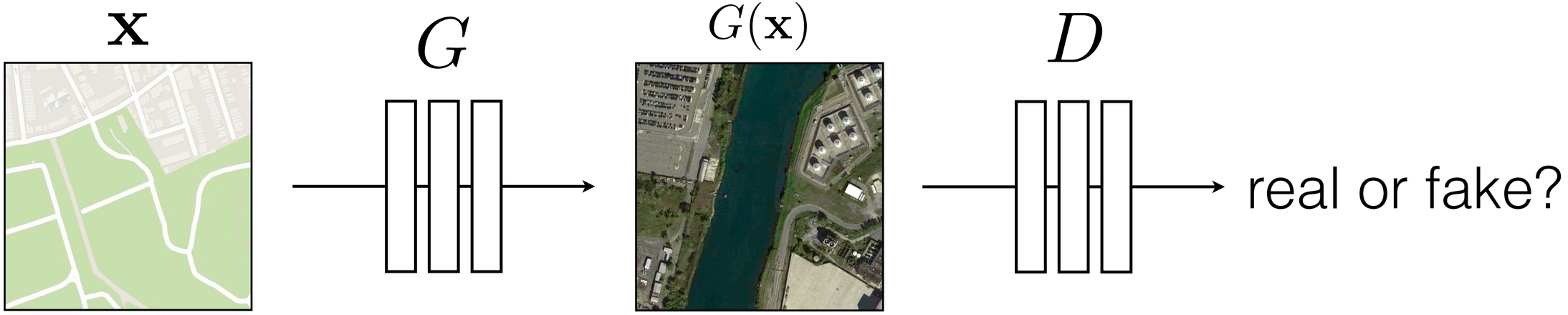
Source: Isola, Freeman, Torralba



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

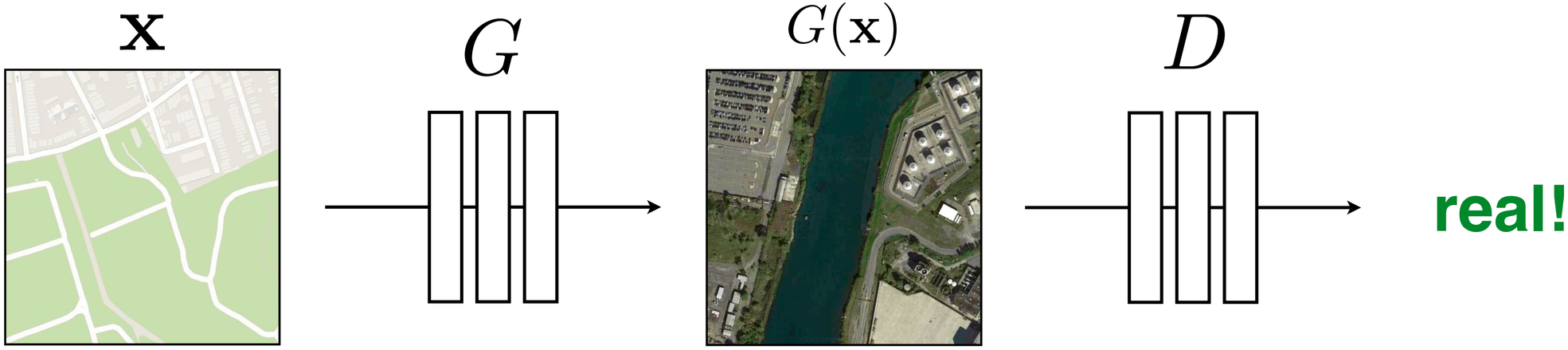
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ ]$$

Source: Isola, Freeman, Torralba



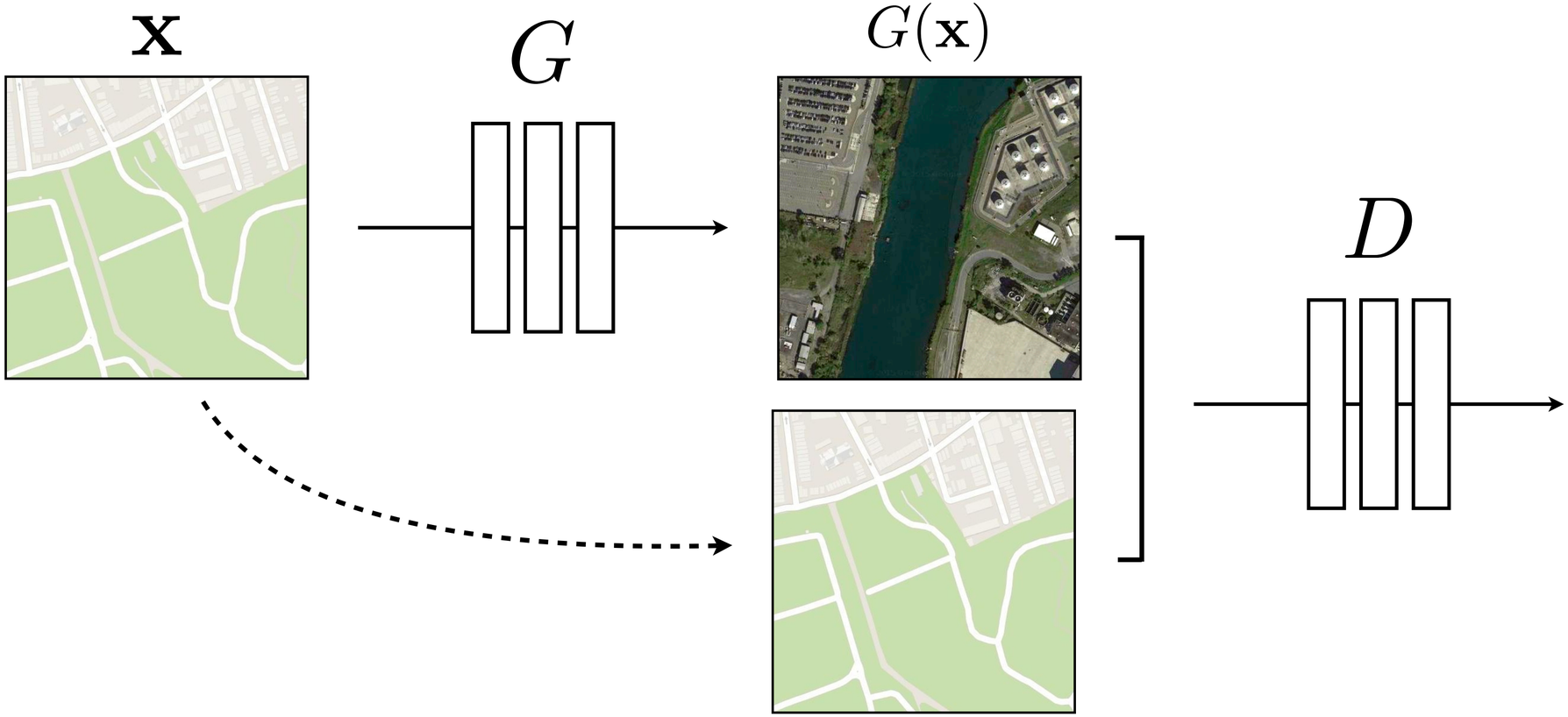
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y})) \;\;]$$

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



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

Source: Isola, Freeman, Torralba

$\mathbf{x}$

G

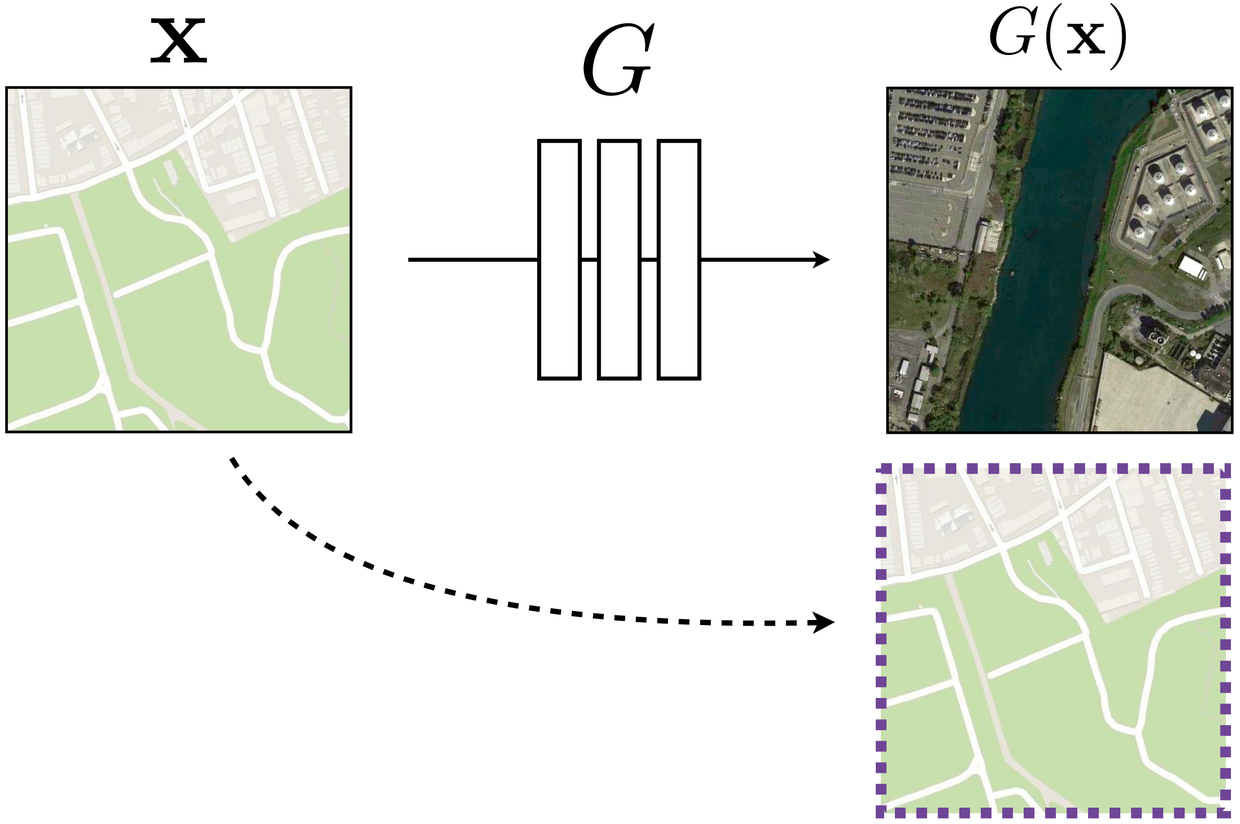
$G(\mathbf{x})$

D

real or fake *pair* ?

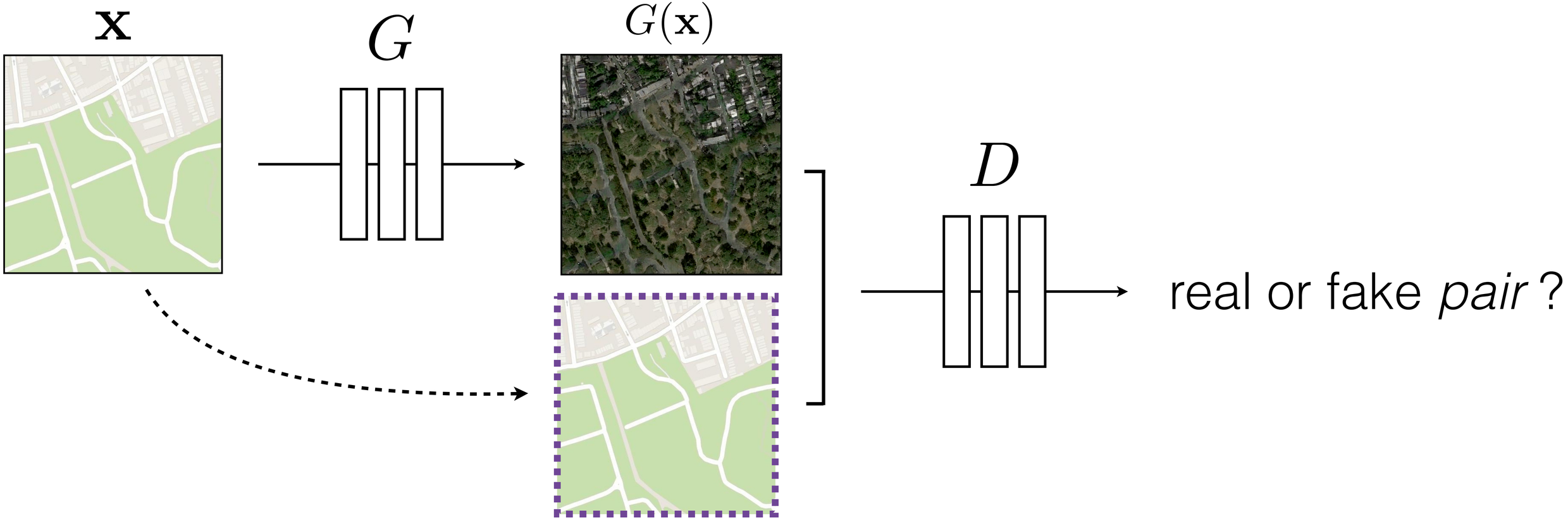
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$

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



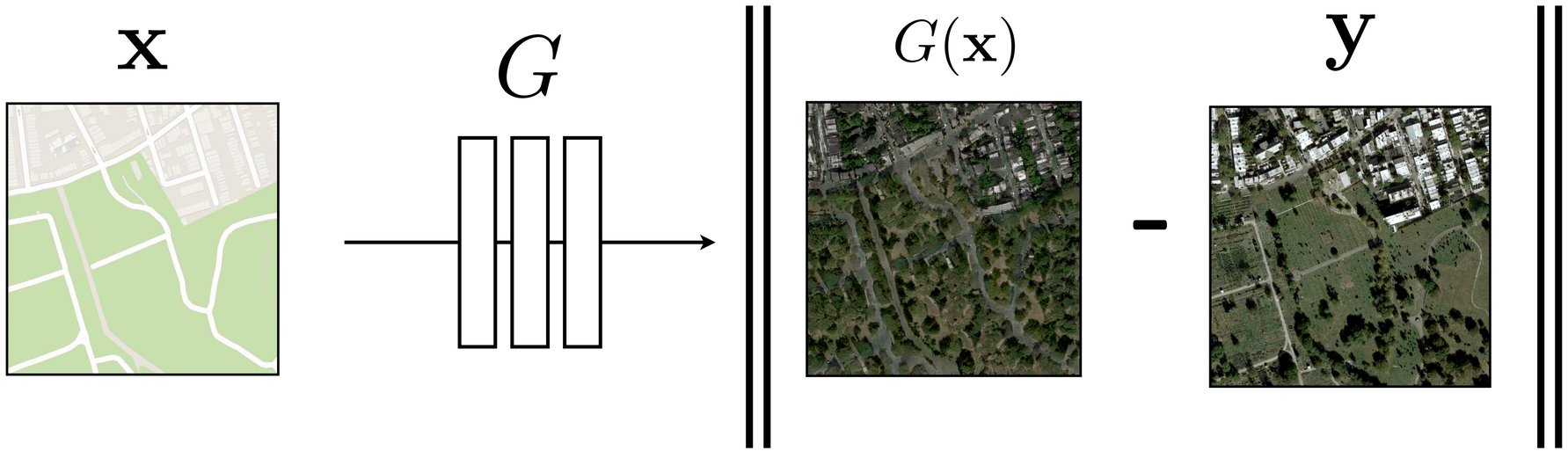
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y})) \;]$$

Source: Isola, Freeman, Torralba

Training Details: Loss function

Conditional GAN

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

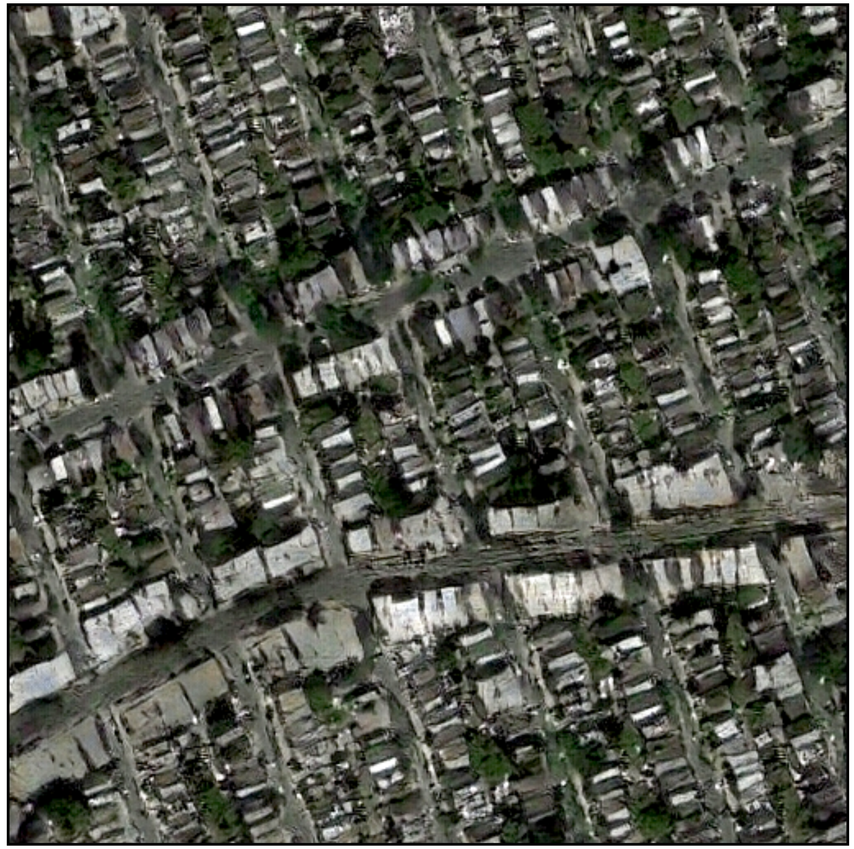


Pixel-wise loss helps stabilize training + faster convergence

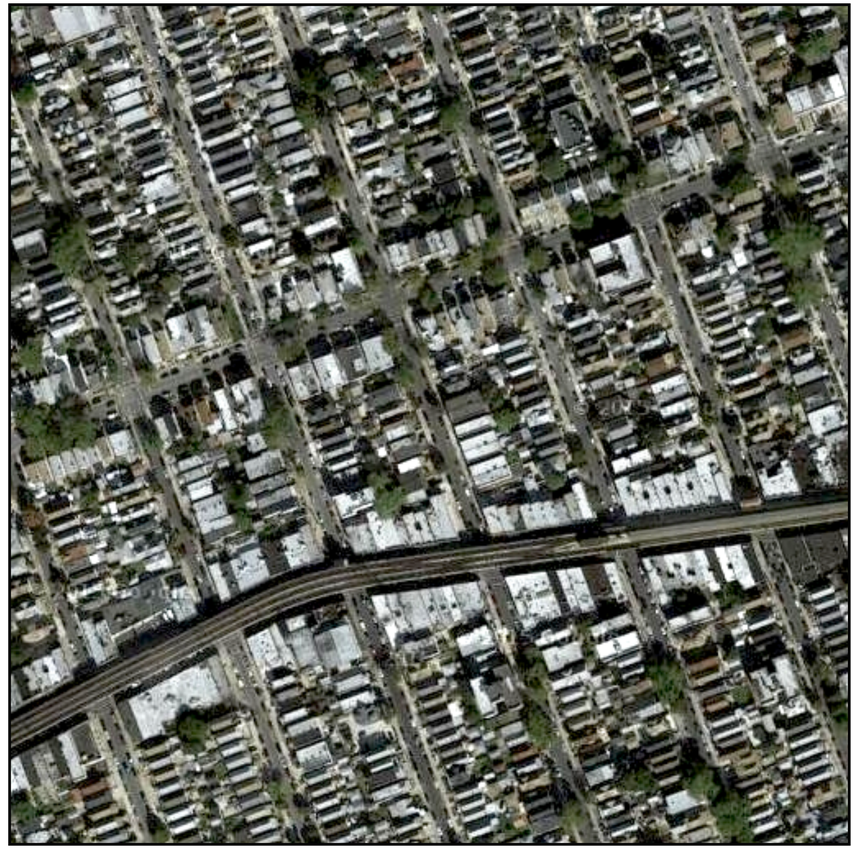
Source: Isola, Freeman, Torralba

Input

Output



Ground truth



Data from [maps.google.com]

Source: Isola, Freeman, Torralba

Input

Output

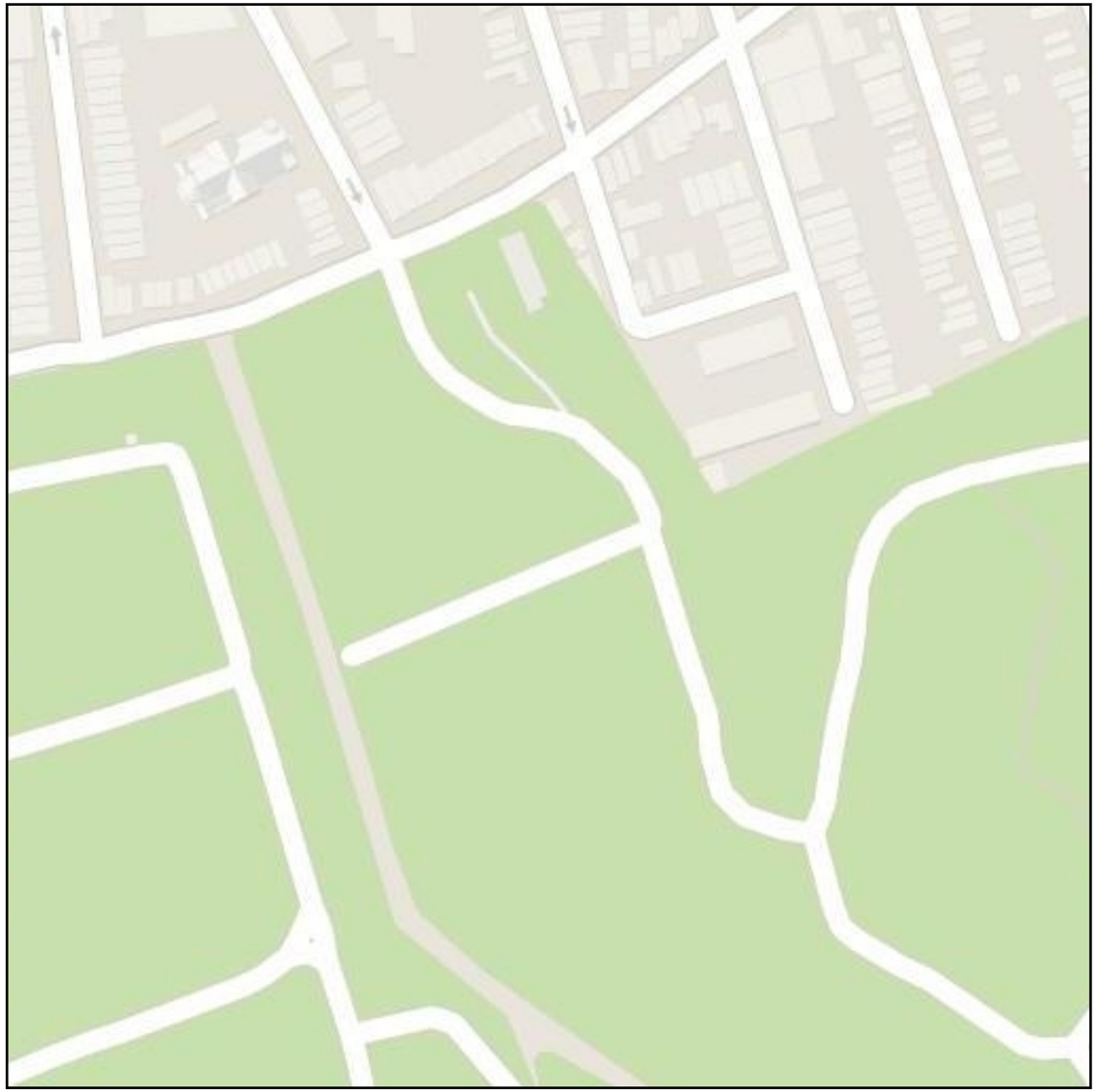
Ground truth

Data from [maps.google.com]

Source: Isola, Freeman, Torralba

Input

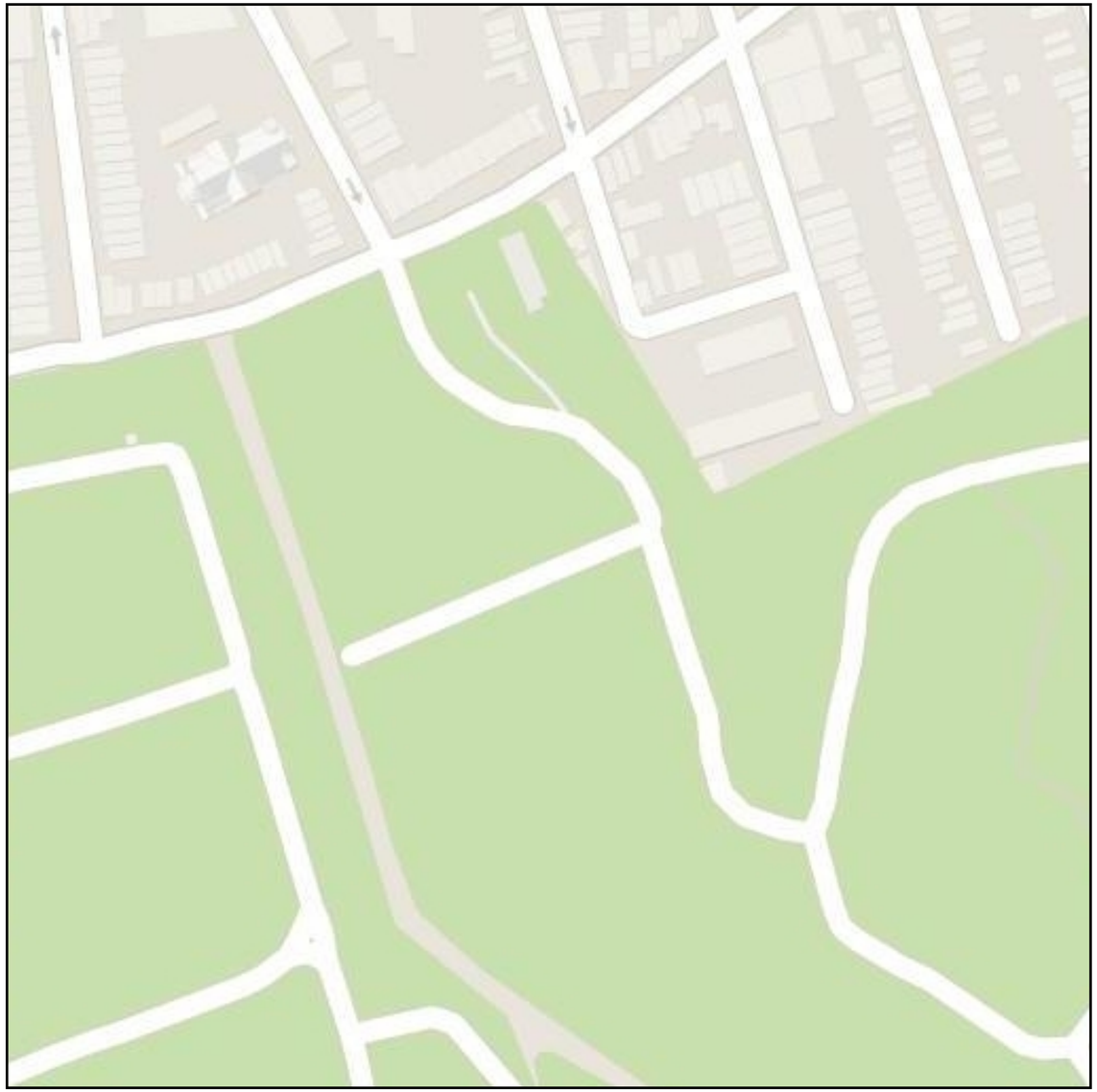
L1 loss only



Source: Isola, Freeman, Torralba

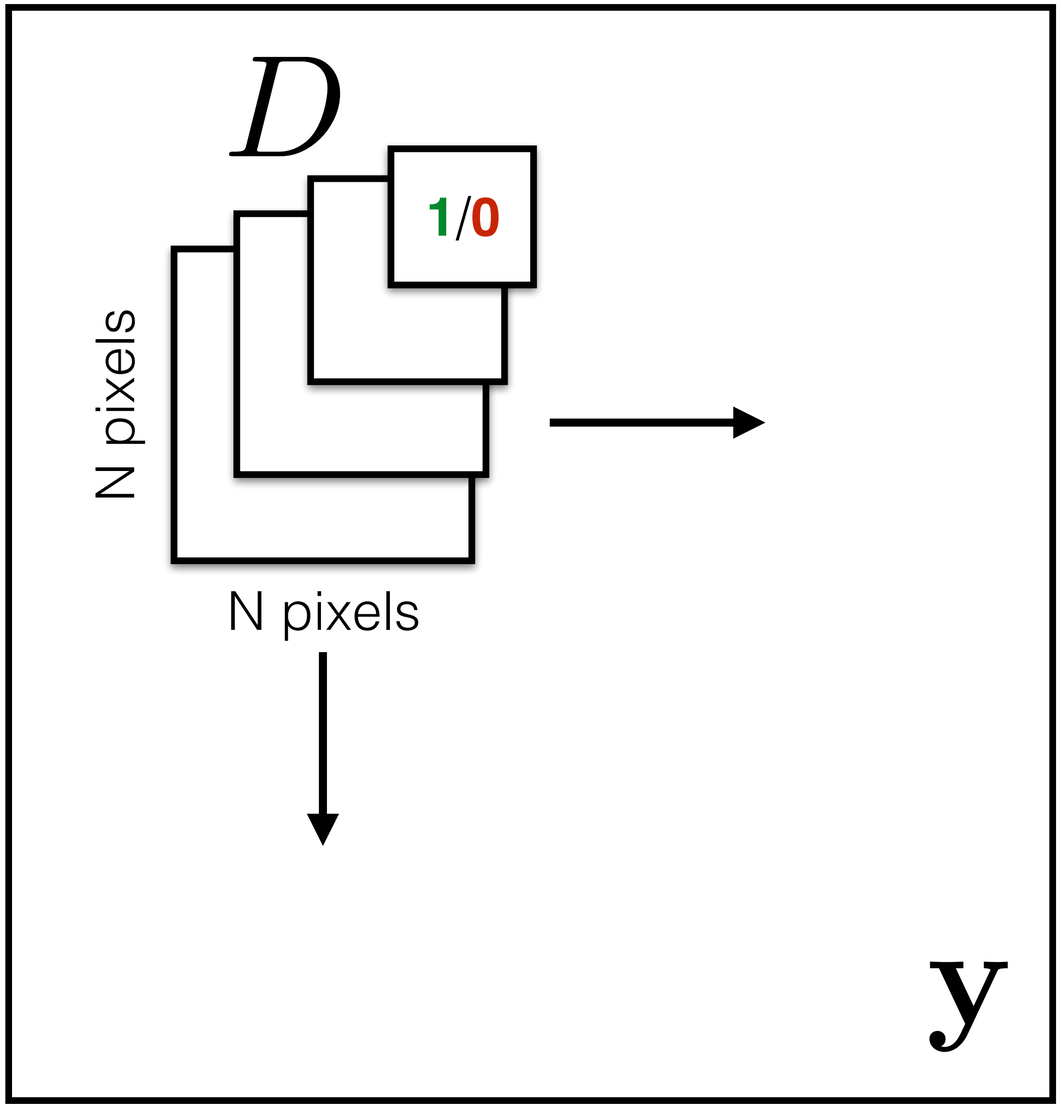
Input

L1 loss + discriminator



Source: Isola, Freeman, Torralba

Patch Discriminator



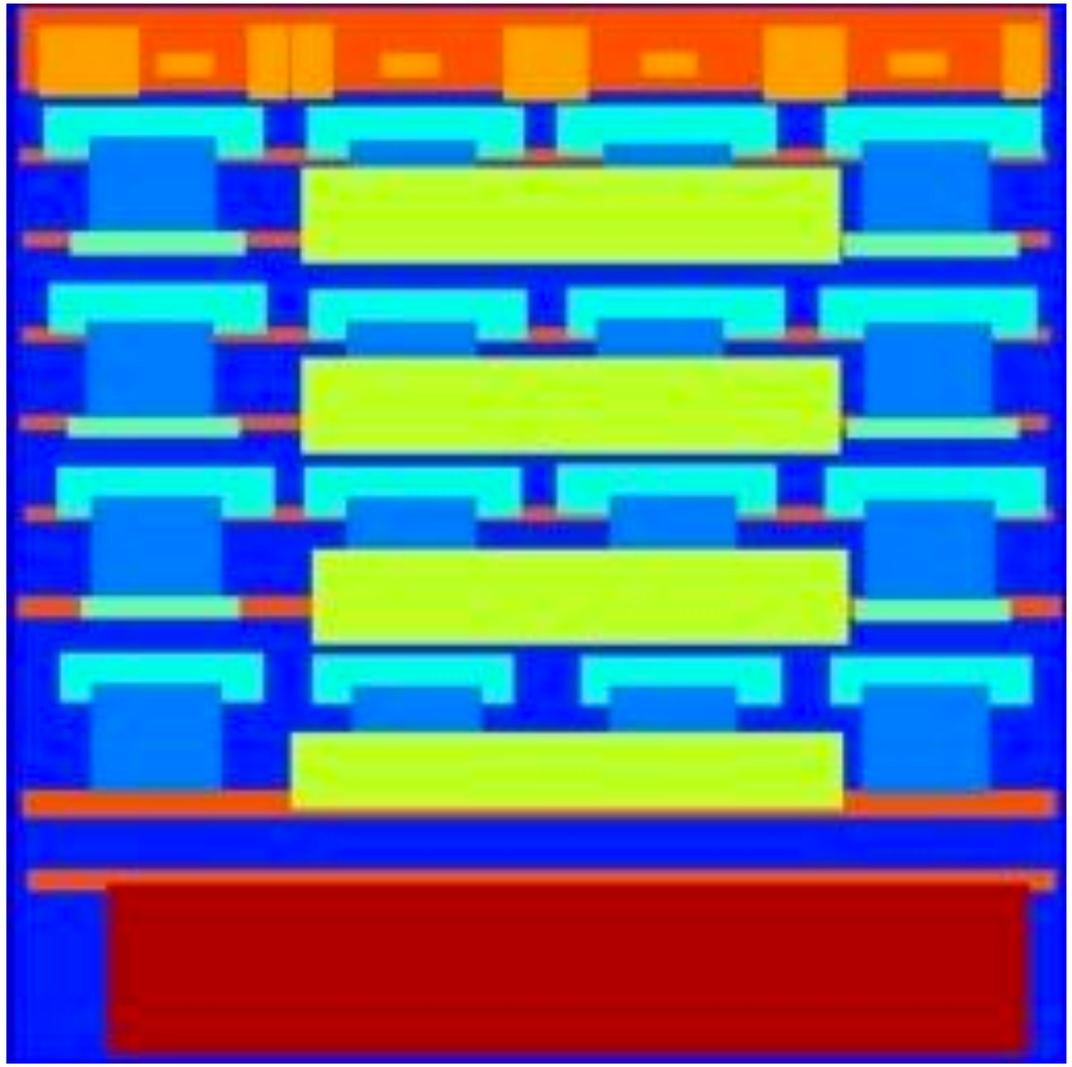
Rather than penalizing if output *image* looks fake, penalize if each overlapping *patch* in output looks fake

- Faster, fewer parameters
- More supervised observations
- Applies to arbitrarily large images

Source: Isola, Freeman, Torralba

Labels → Facades

Input



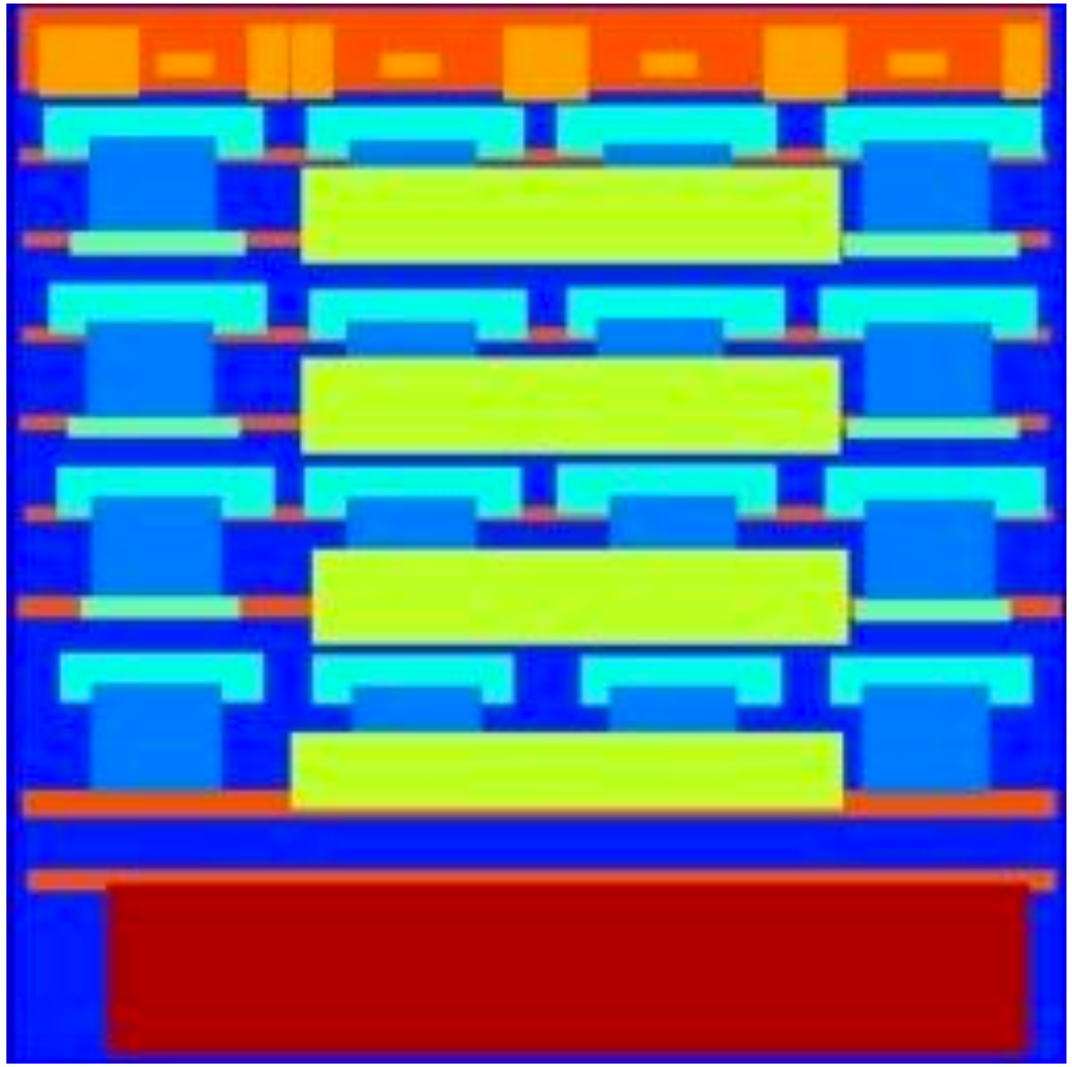
1x1 Discriminator

Source: Isola, Freeman, Torralba

Data from [Tylecek, 2013]

Labels → Facades

Input



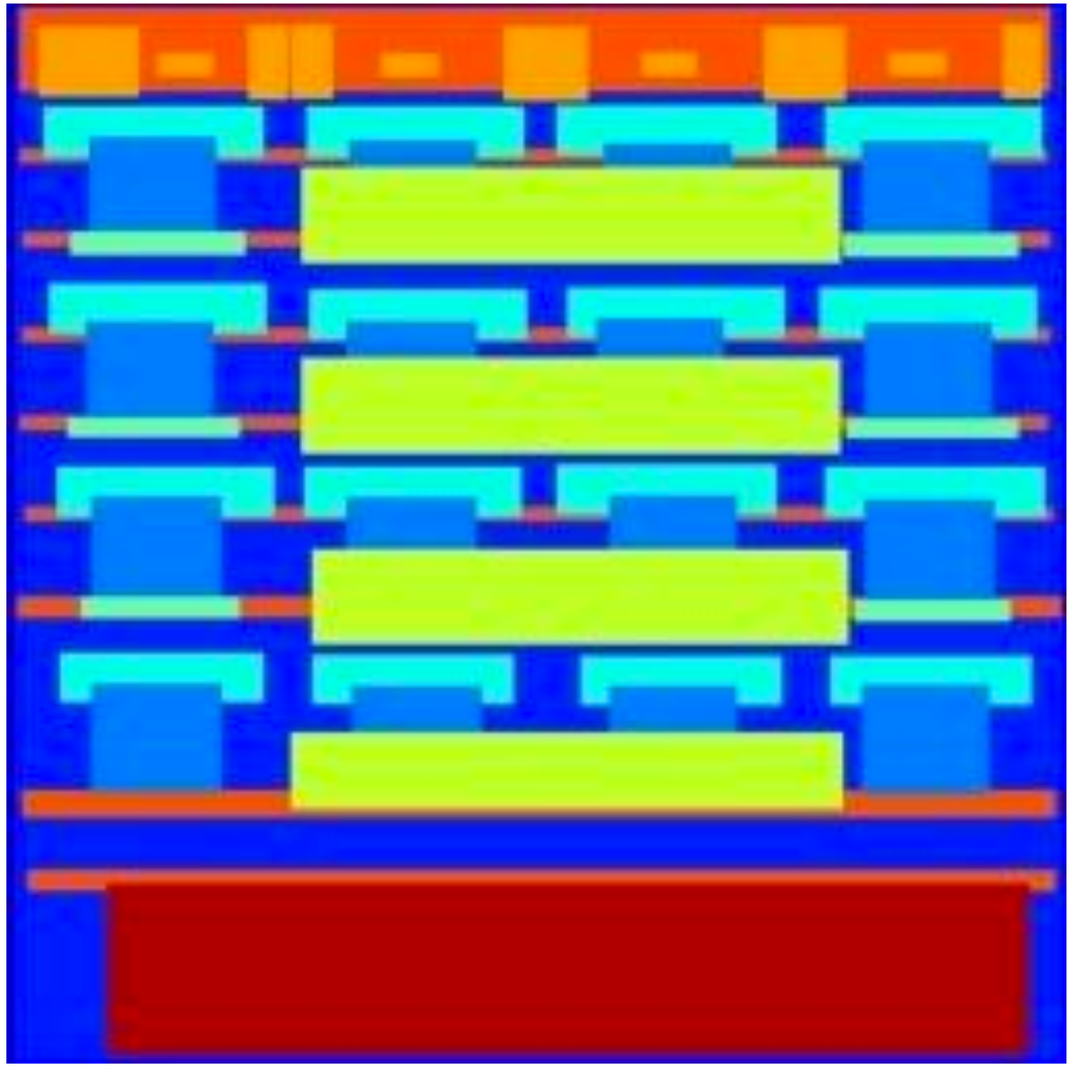
16x16 Discriminator

Source: Isola, Freeman, Torralba

Data from [Tylecek, 2013]

Labels → Facades

Input



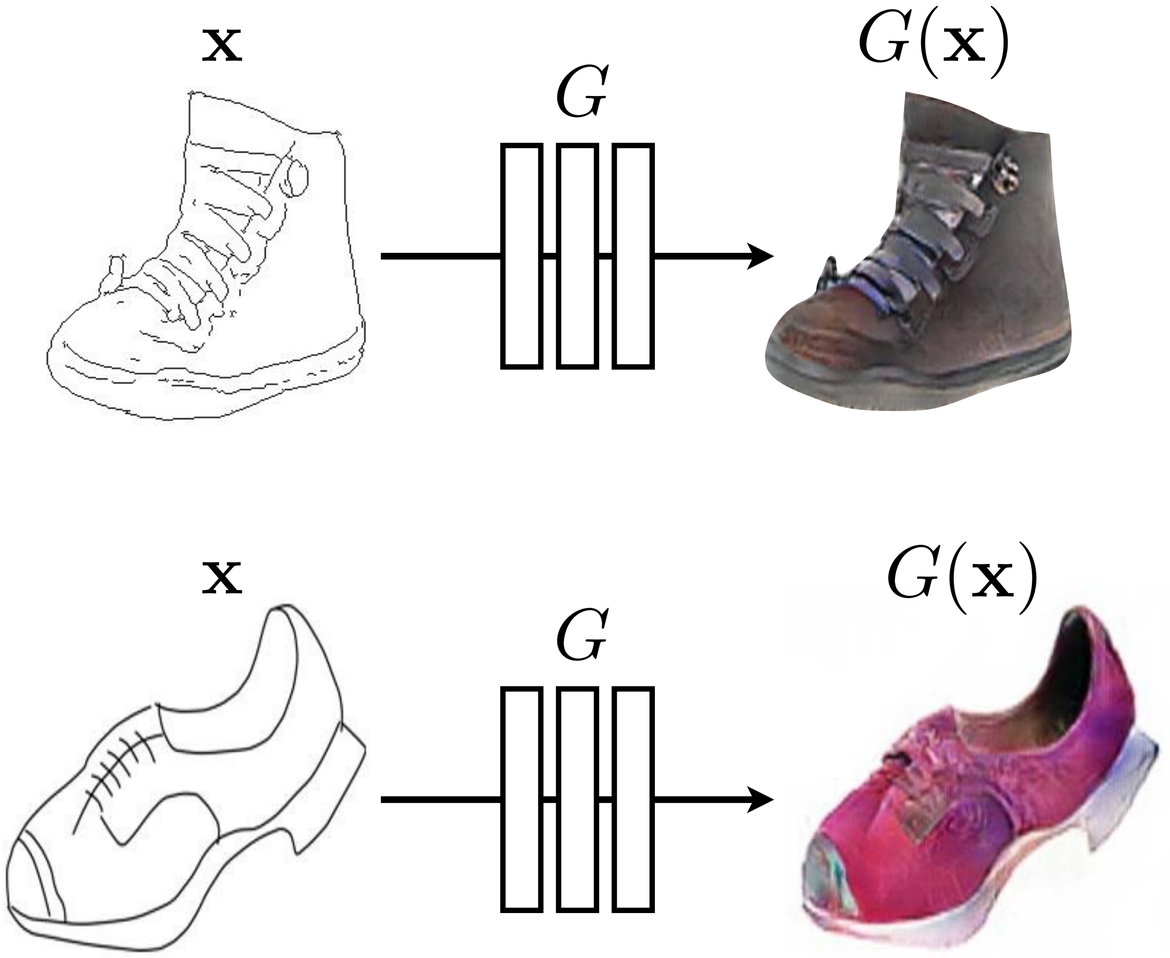
70x70 Discriminator

Source: Isola, Freeman, Torralba

Data from [Tylecek, 2013]

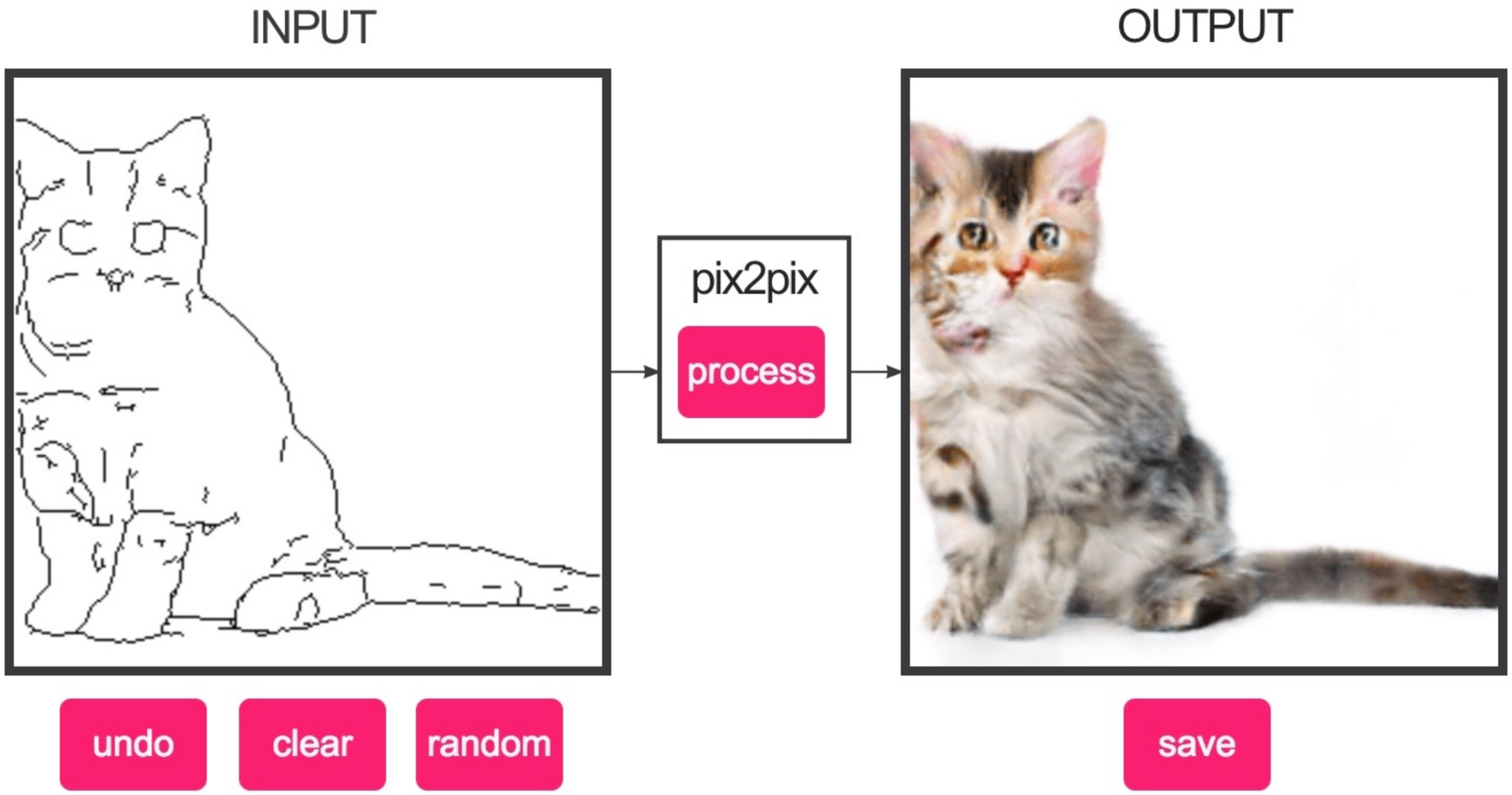
Training data

[HED, Xie & Tu, 2015]



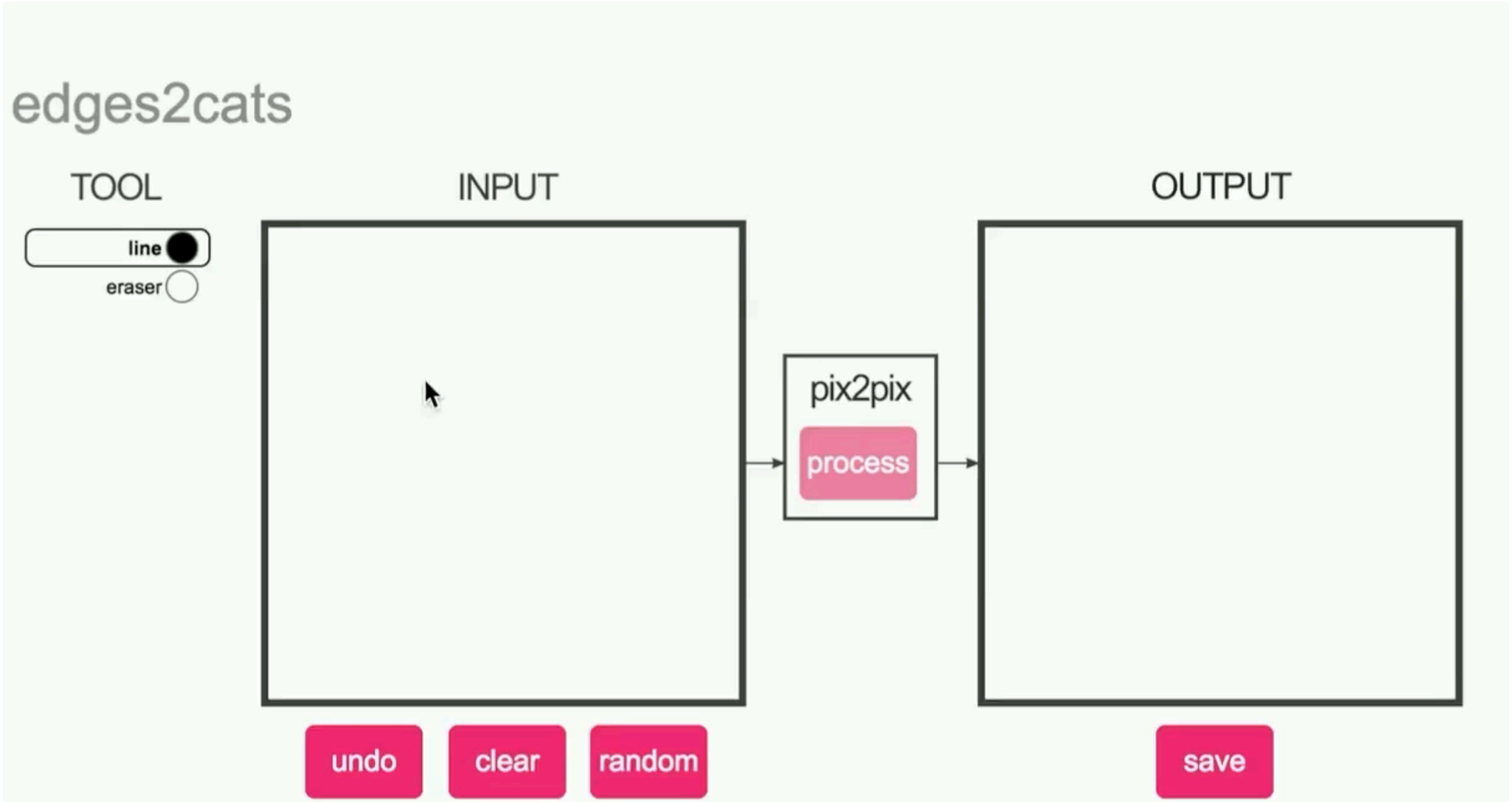
Source: Isola, Freeman, Torralba

edges2cats

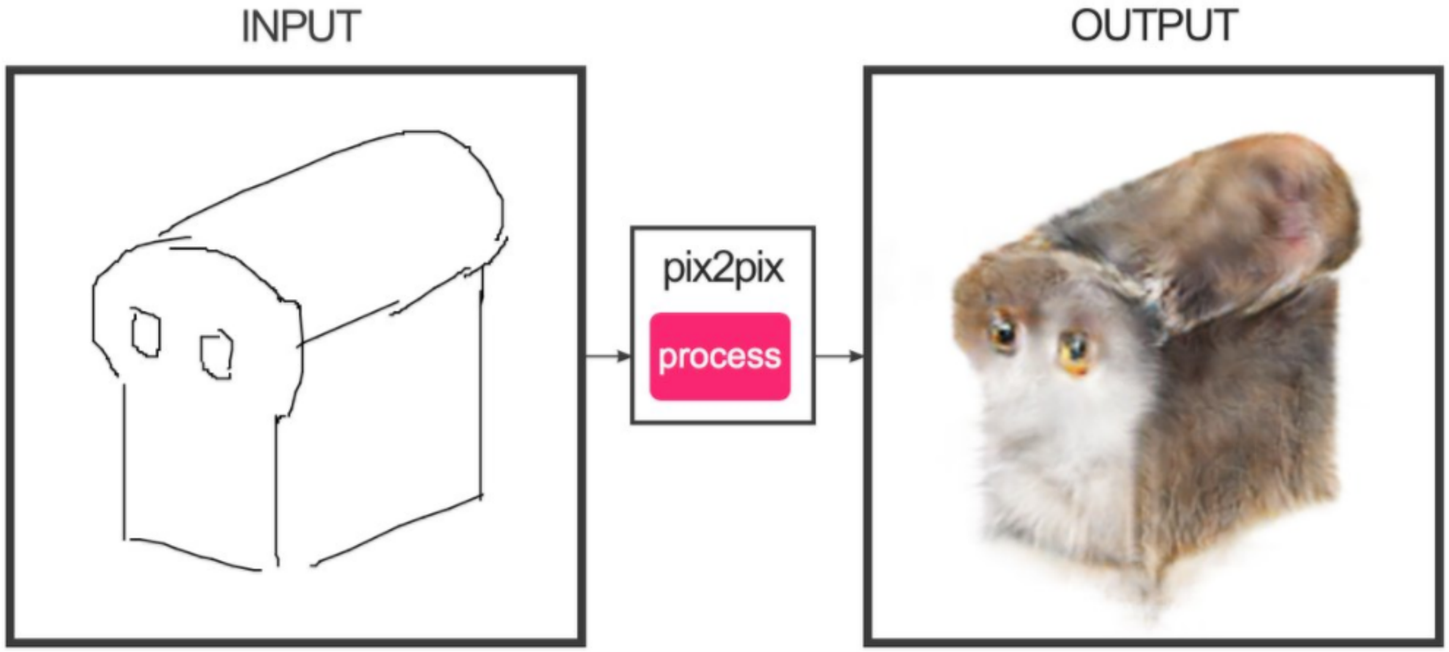


[Chris Hess, edges2cats]

Source: Isola, Freeman, Torralba



Source: Isola, Freeman, Torralba



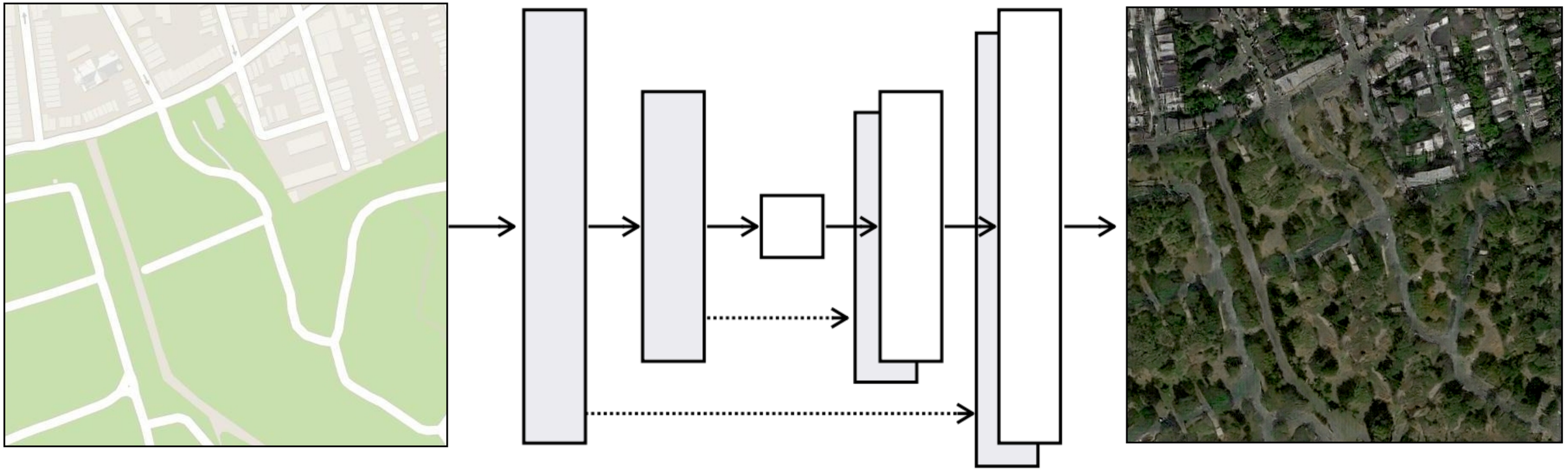
Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

Source: Isola, Freeman, Torralba

Architectures

Generator: U-Net

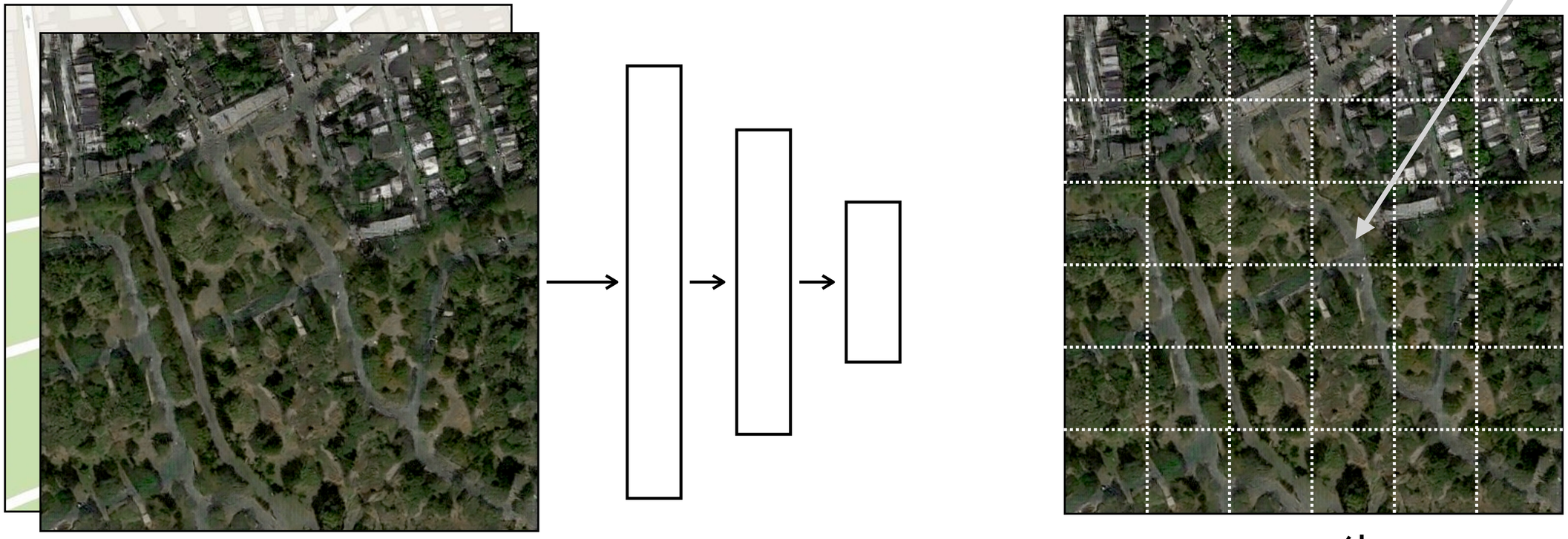


Skip connections between encoder and decoder layers

Figure from [Isola et al., “Image-to-Image Translation with Conditional Adversarial Networks”, 2017]

Architectures

Discriminator: fully convolutional network

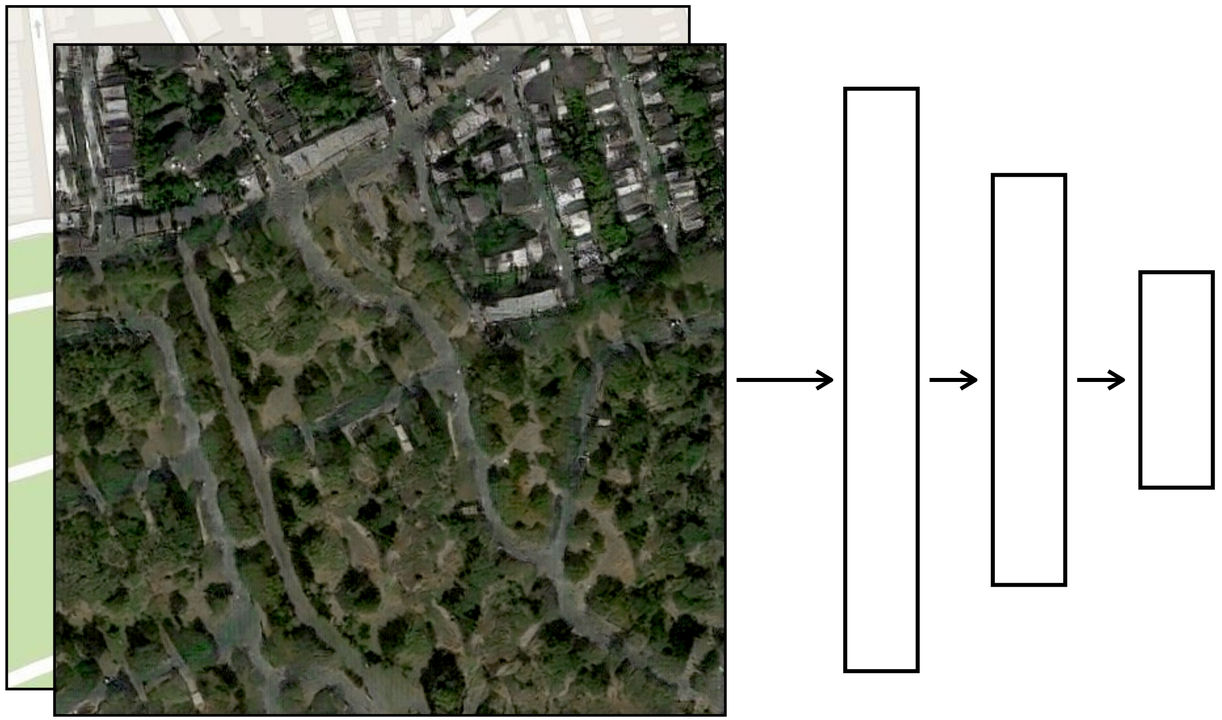


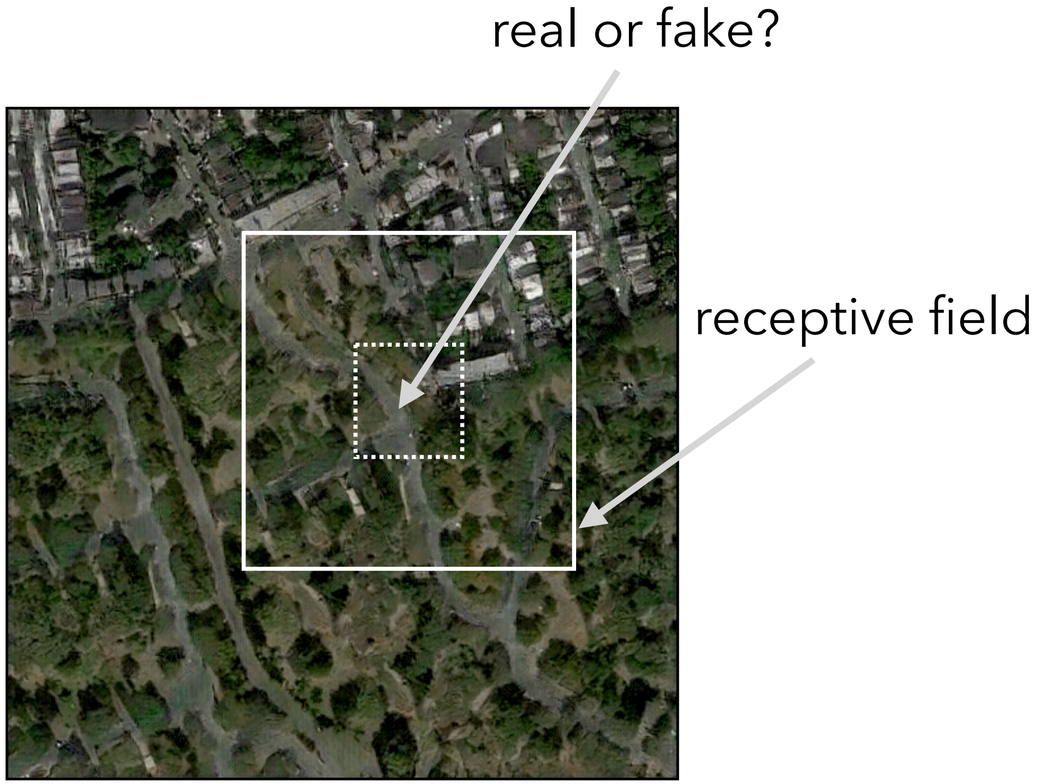
$n \times n$ output map (last conv. layer)

Sequence of strided convolutions

Architectures

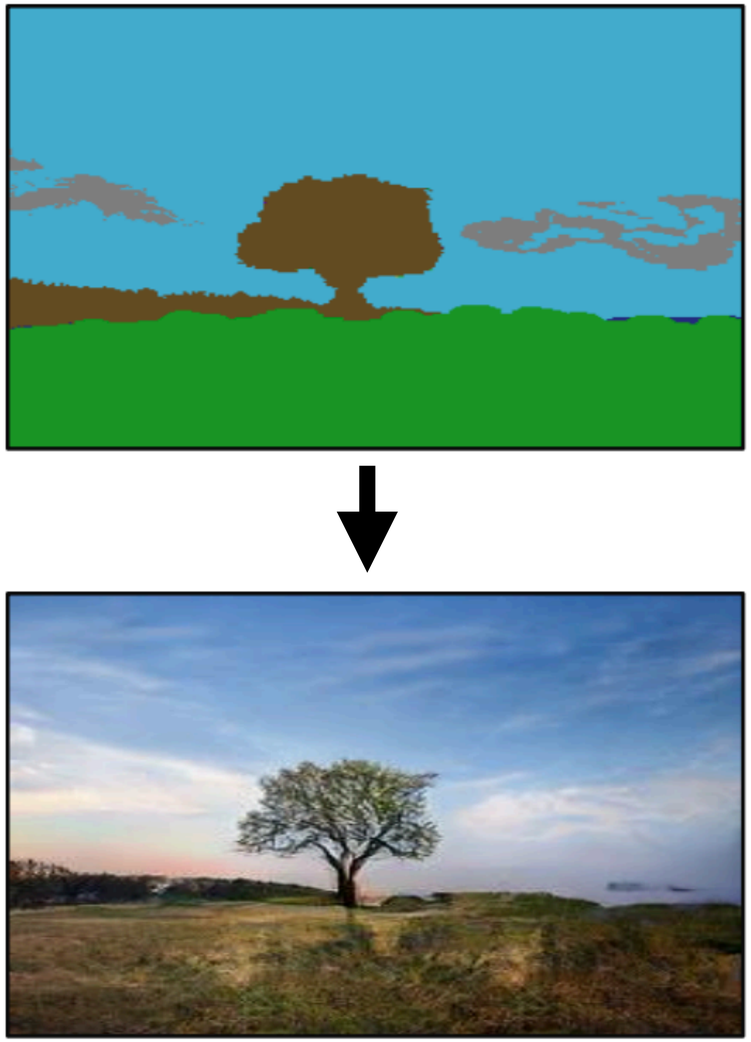
Discriminator: fully convolutional network



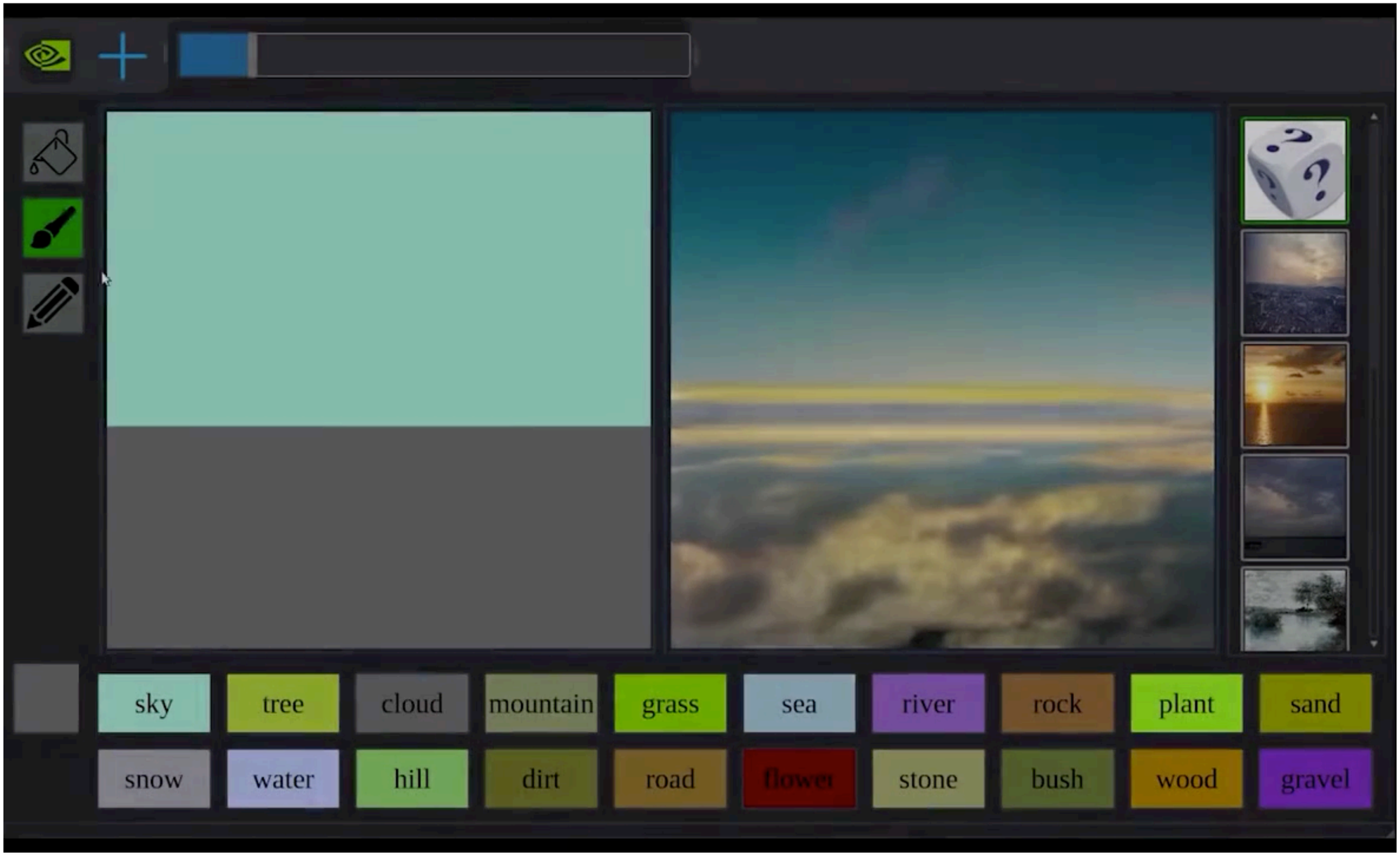


Sometimes called a Patch GAN, since it effectively only looks at patches

Larger models



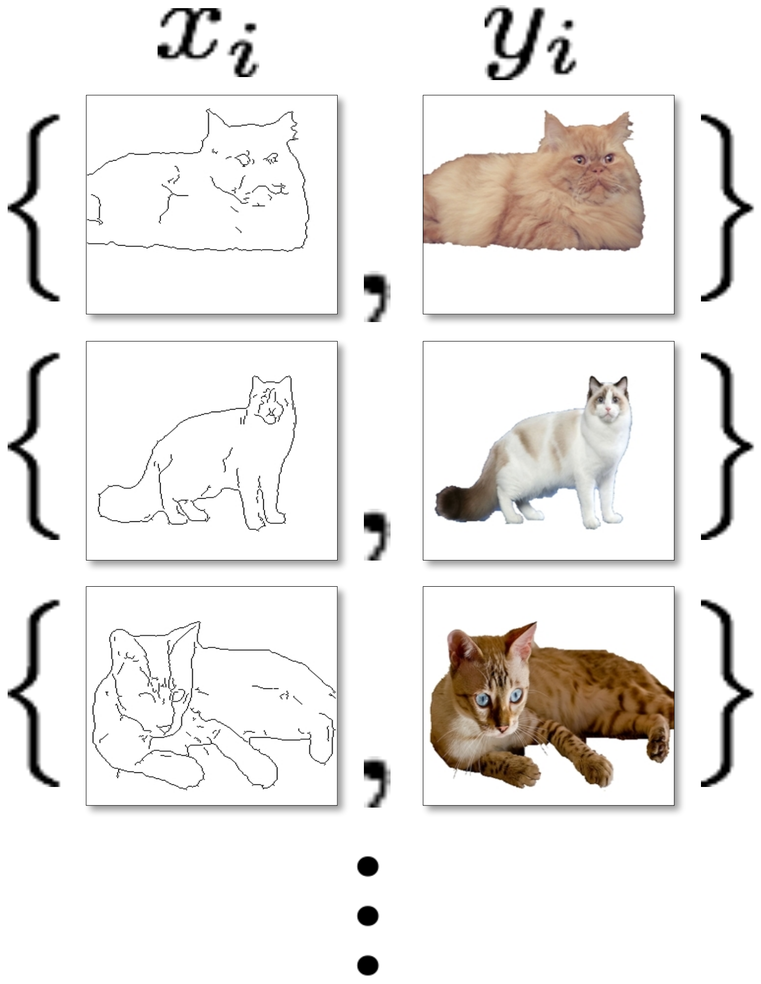
["GauGAN", Park et al., CVPR 2019]



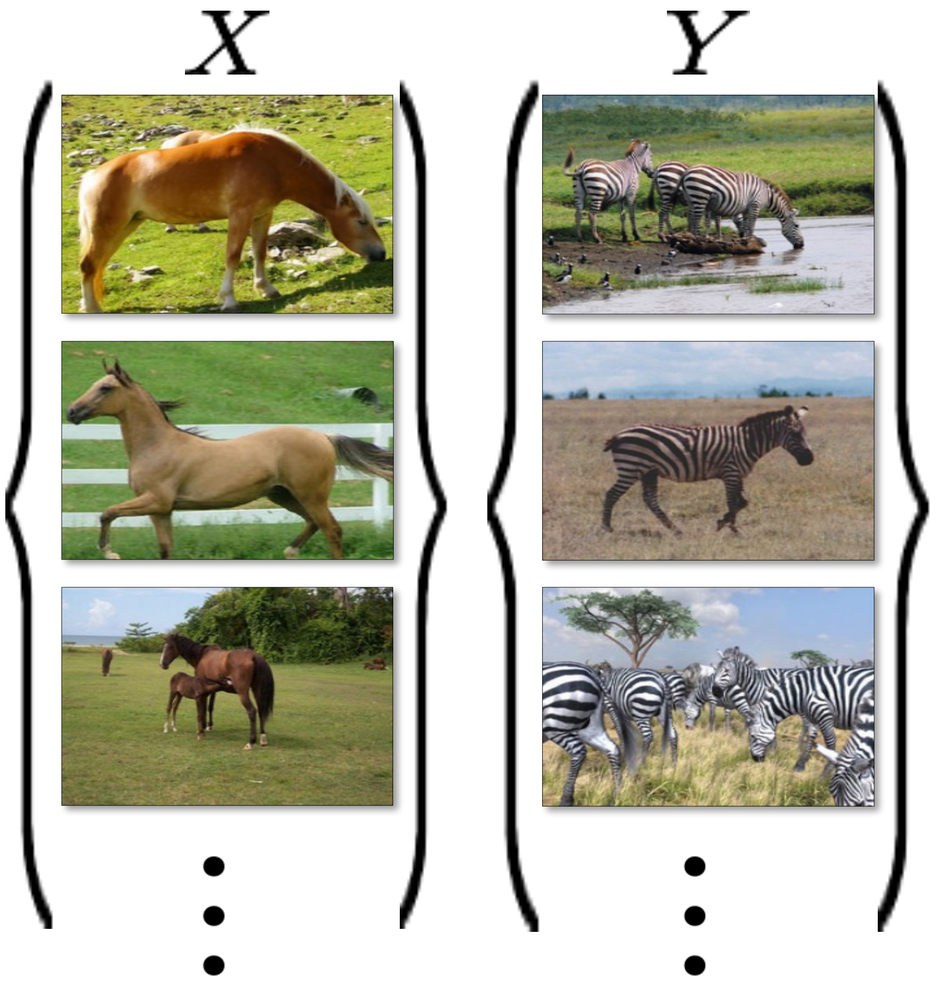
["GauGAN", Park et al., CVPR 2019]

Handling unpaired data

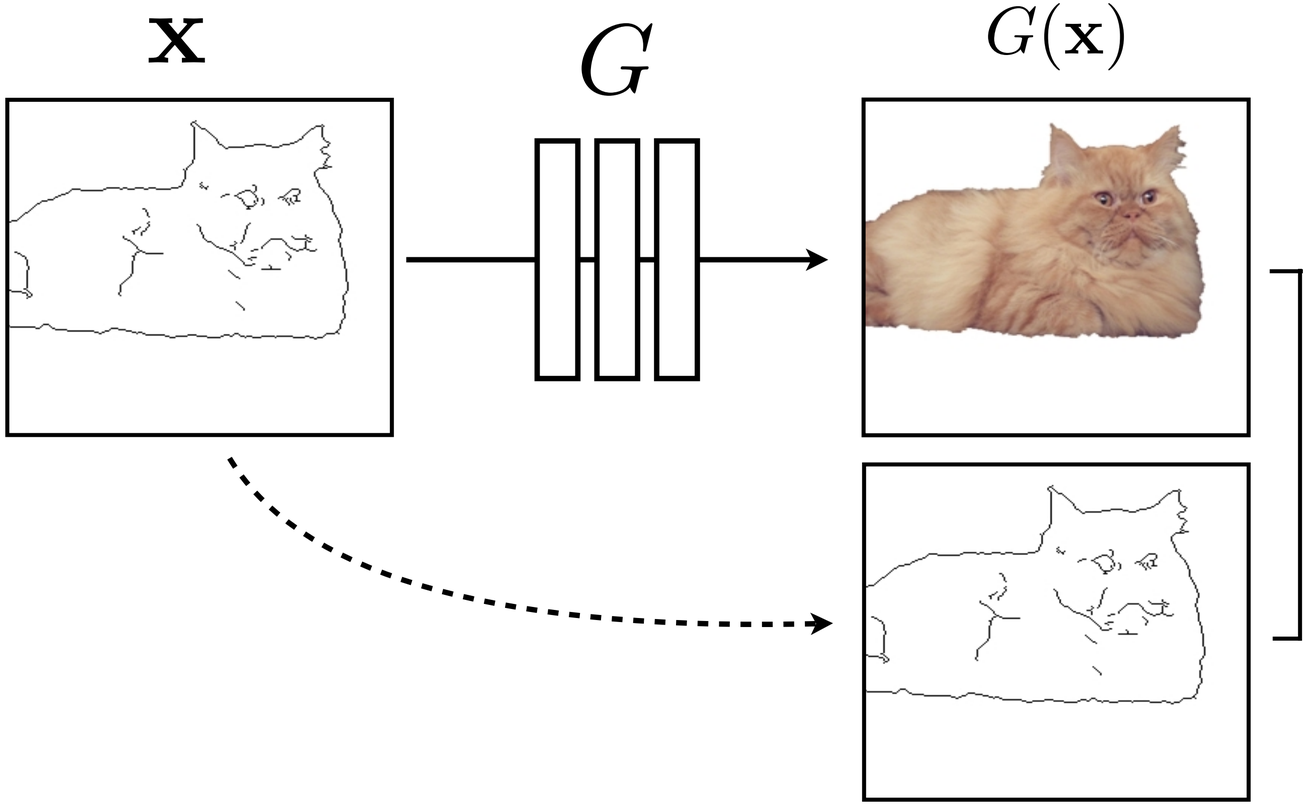
Paired data



Unpaired data

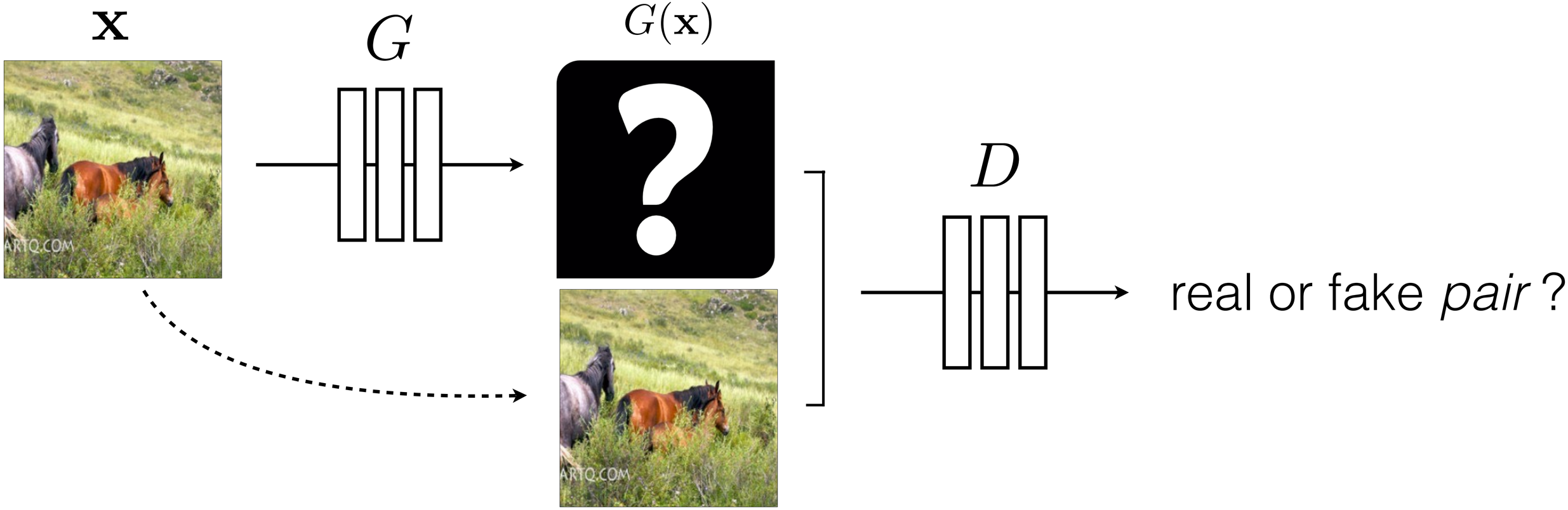


Source: Isola, Freeman, Torralba



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

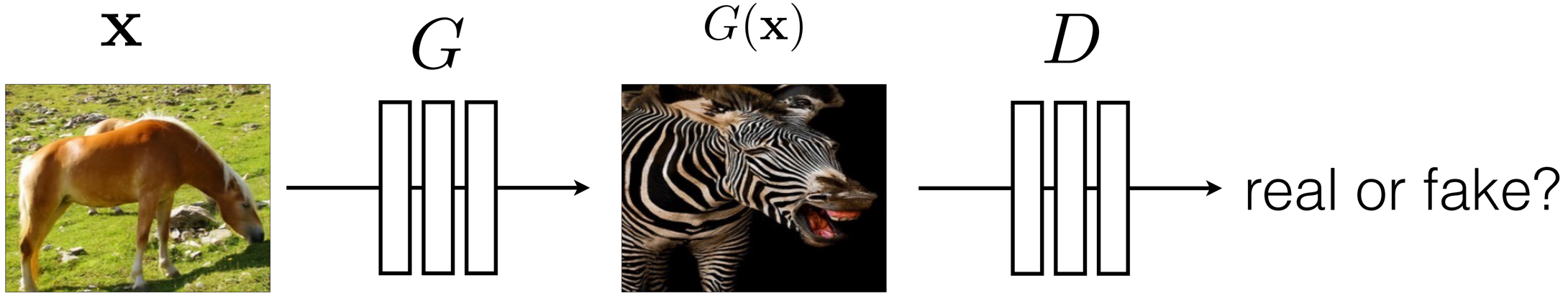
Source: Isola, Freeman, Torralba



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

No input-output pairs!

Source: Isola, Freeman, Torralba



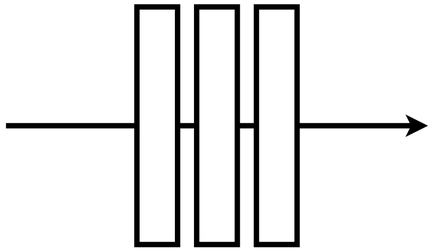
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\;]$$

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

Source: Isola, Freeman, Torralba

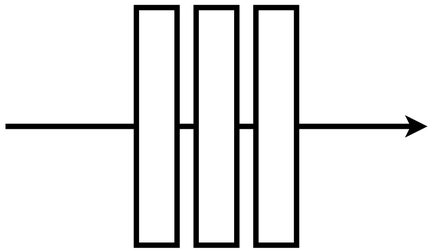
$\mathbf{x}$

G



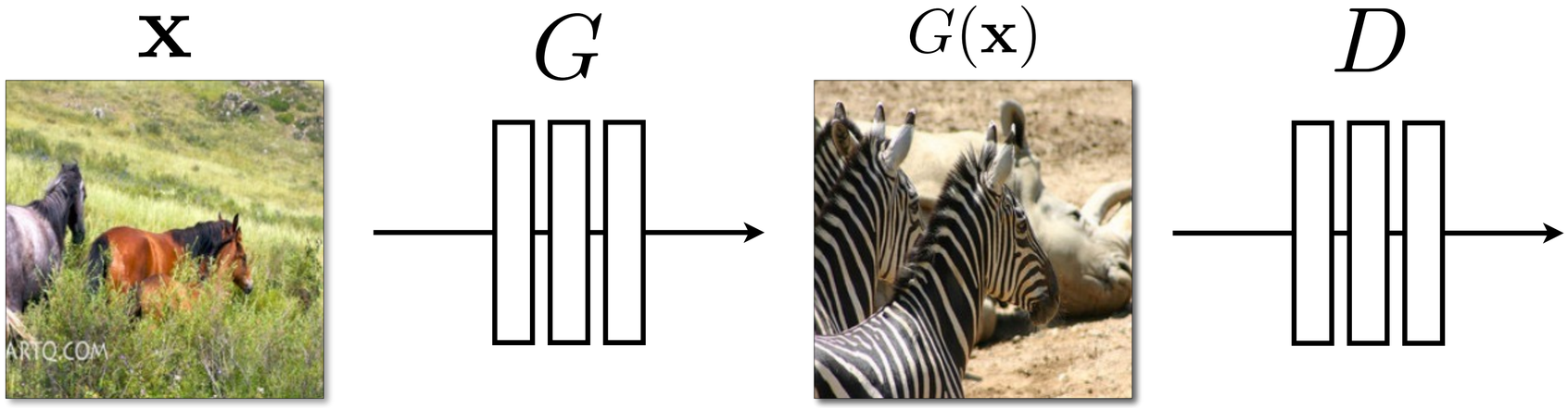
$G(\mathbf{x})$

D



Real!

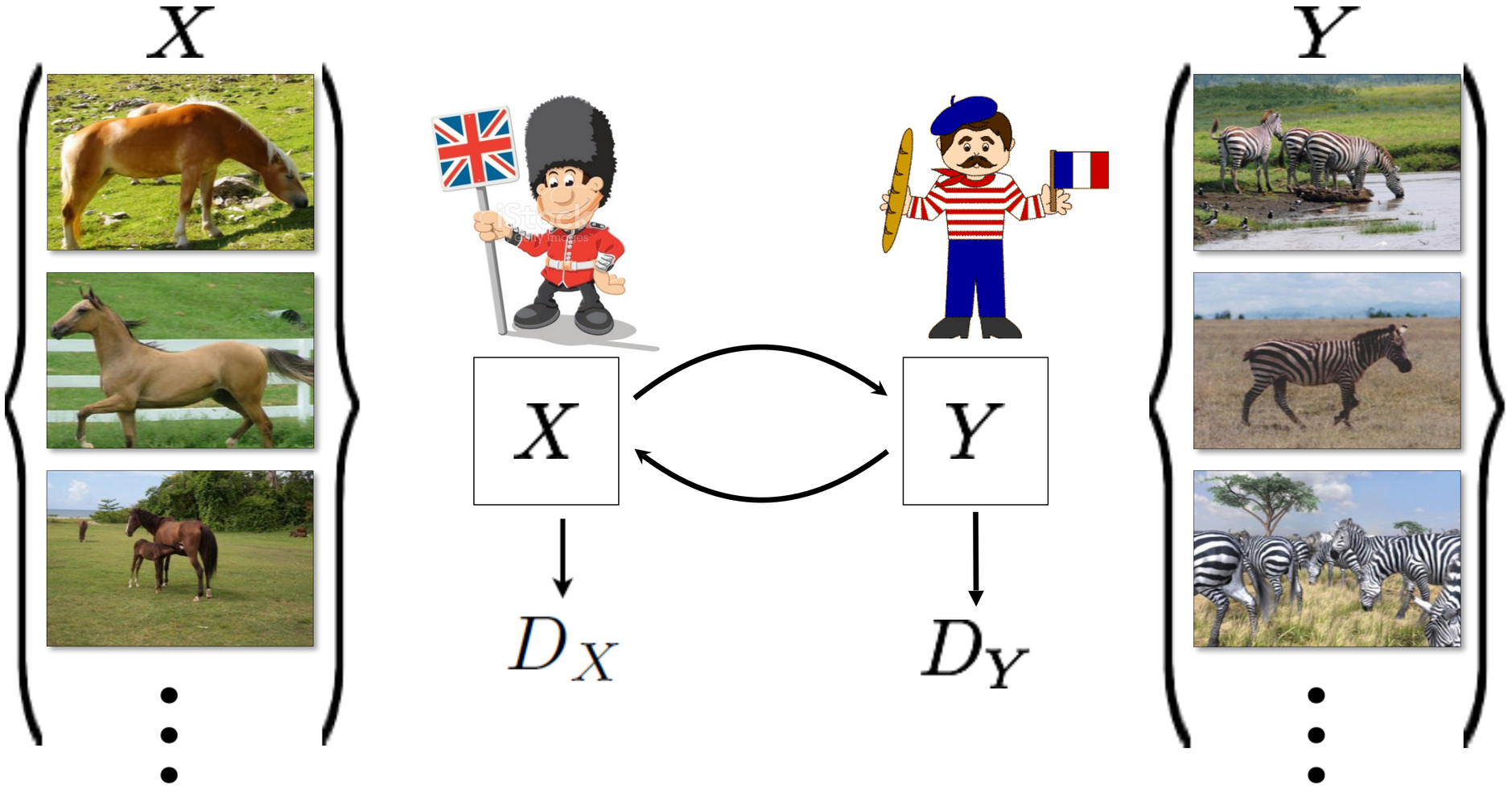
Source: Isola, Freeman, Torralba



Nothing to force output to correspond to input

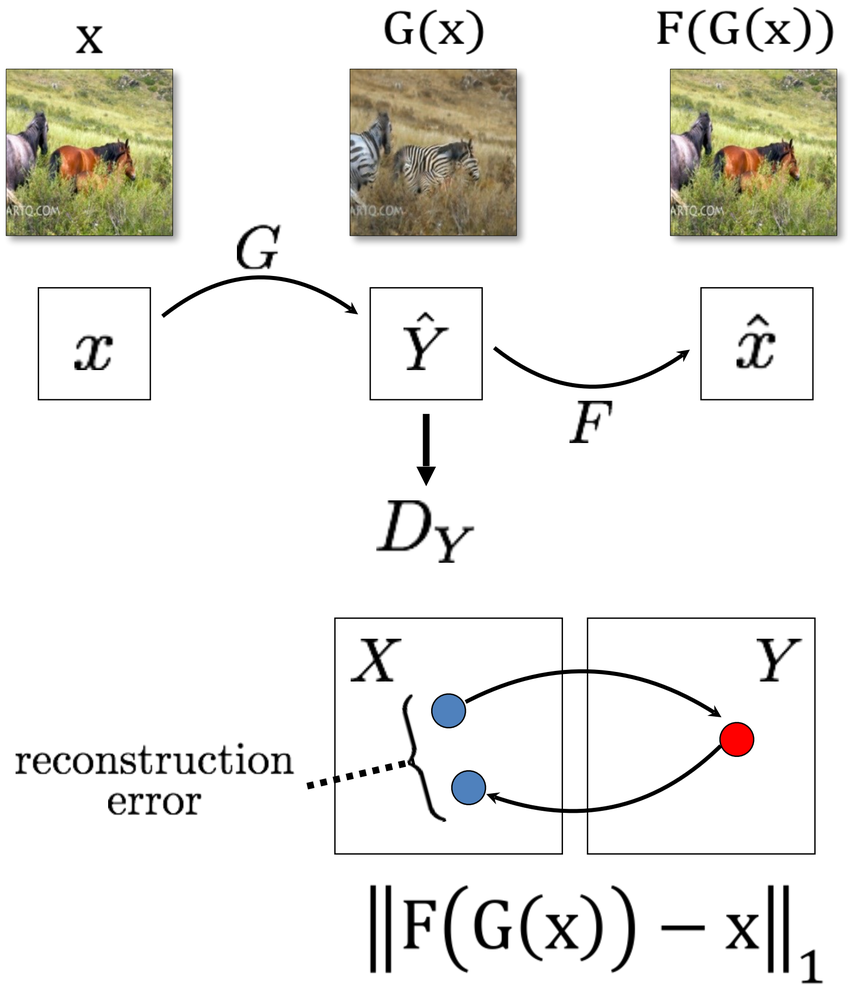
Source: Isola, Freeman, Torralba

CycleGAN



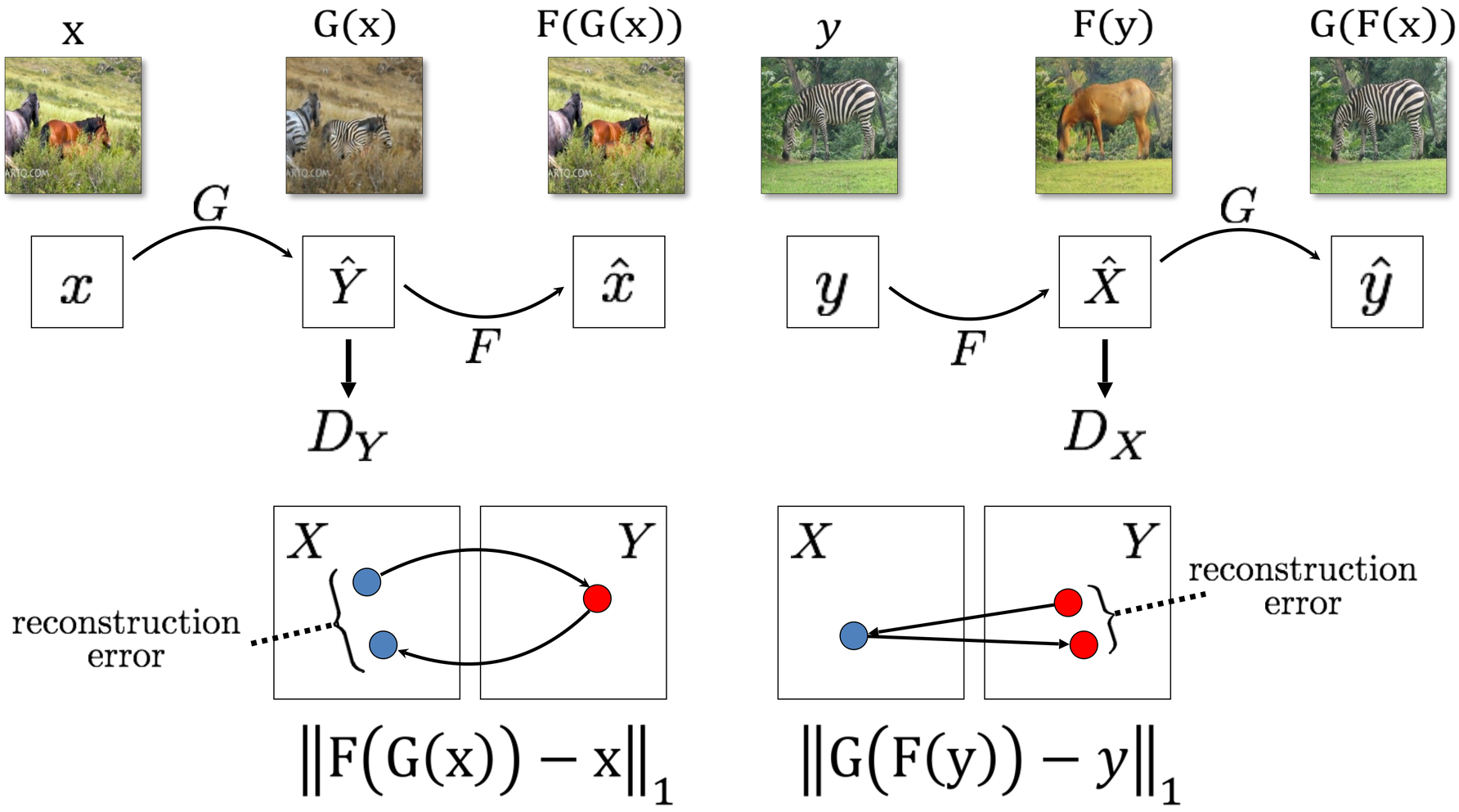
Source: Isola, Freeman, Torralba

Cycle Consistency Loss

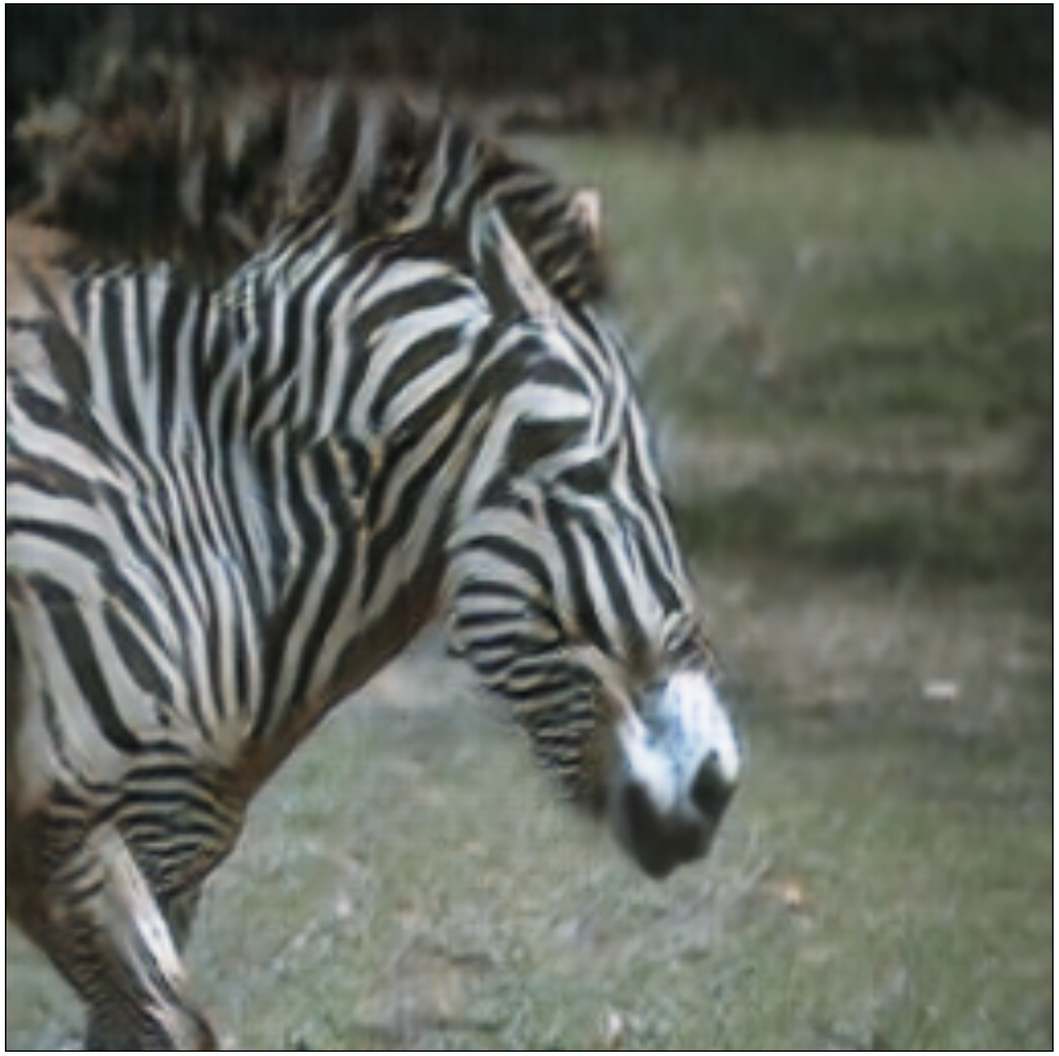


Source: Isola, Freeman, Torralba

Cycle Consistency Loss



Source: Isola, Freeman, Torralba



Collection Style Transfer

Photograph

Monet

Van Gogh

Cezanne

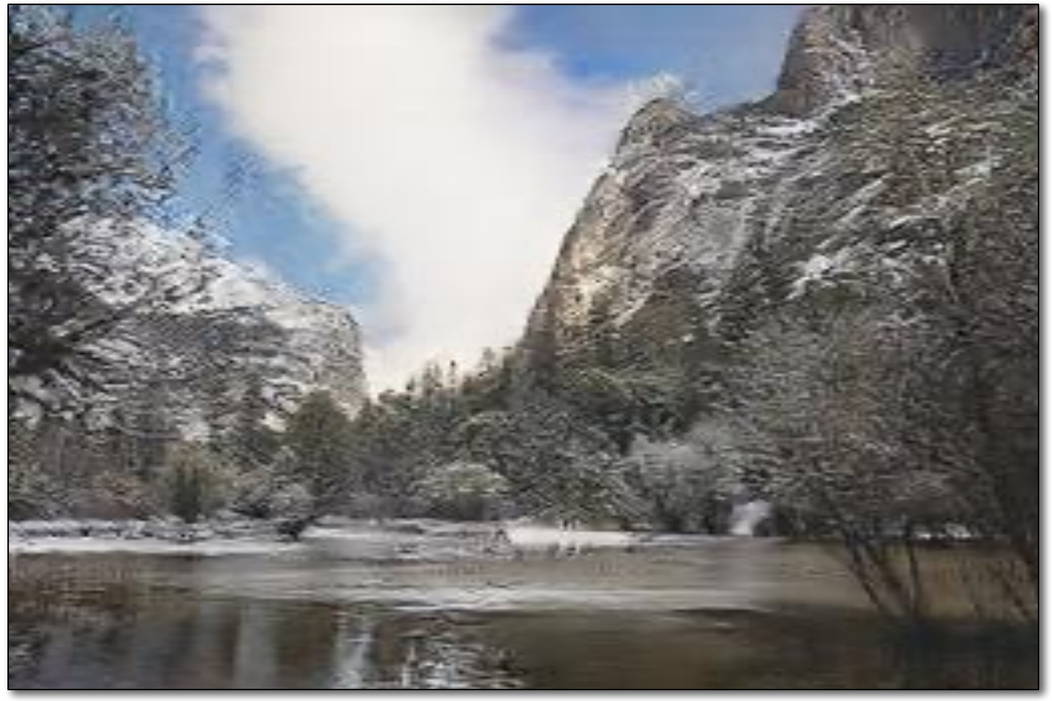
Ukiyo-e

Photo by Alexei Efros


Input
Monet
Van Gogh
Cezanne
Ukiyo-e

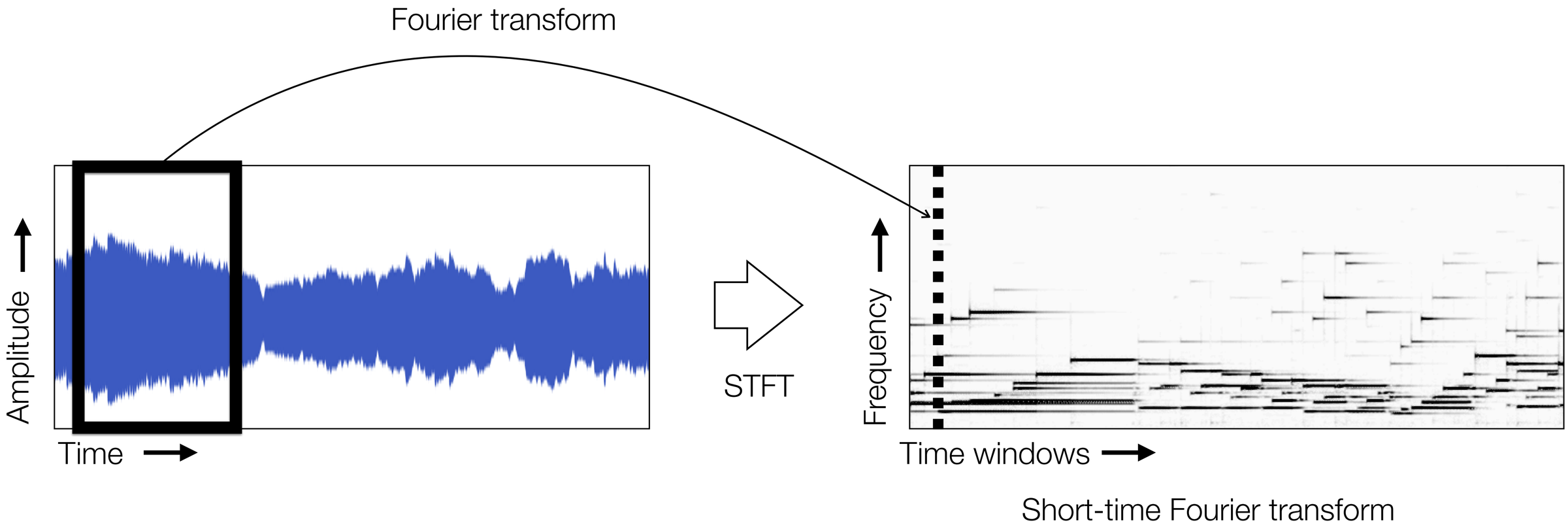
Monet’s paintings → photos

Monet’s paintings → photos

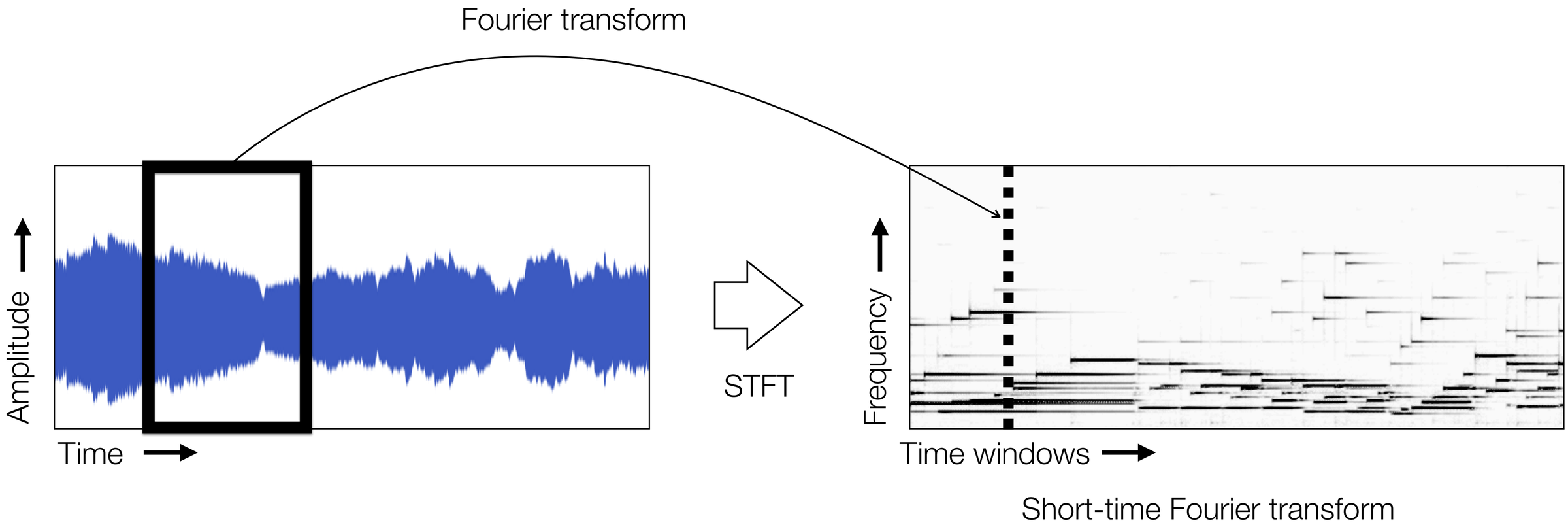


Other applications

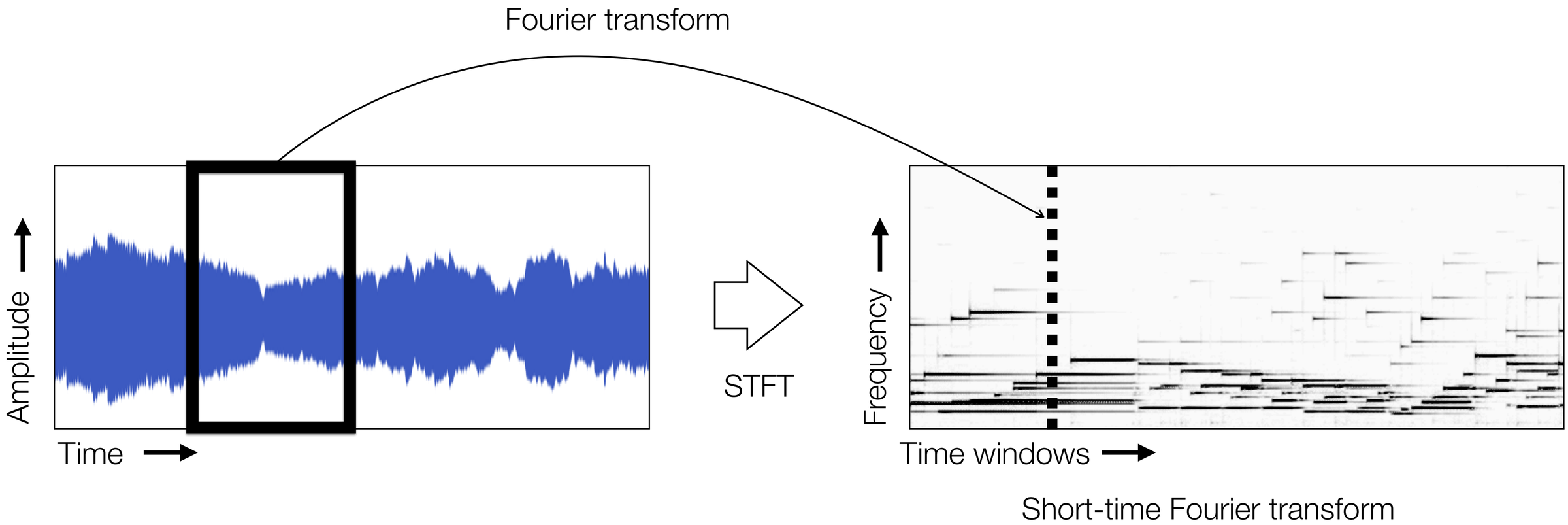
Spectrograms for audio



Spectrograms for audio

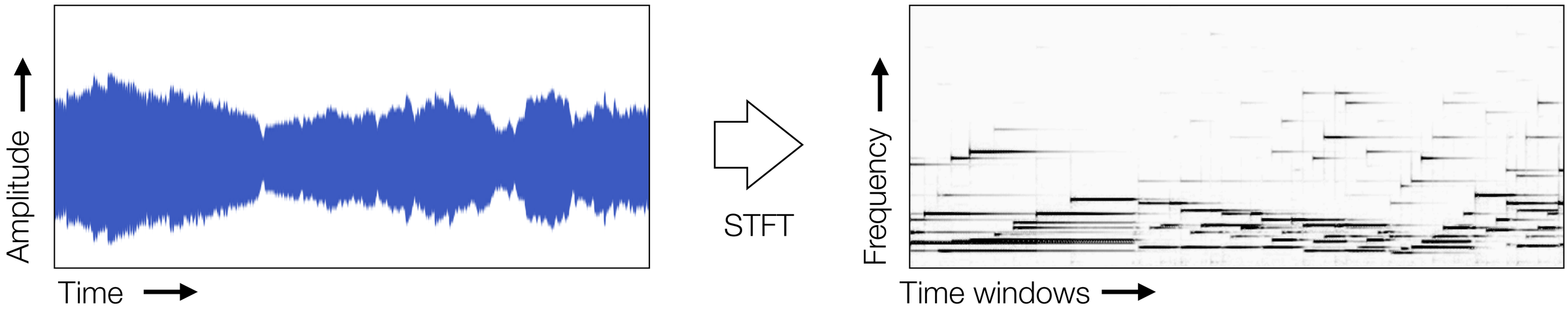


Spectrograms for audio



Spectrograms for audio

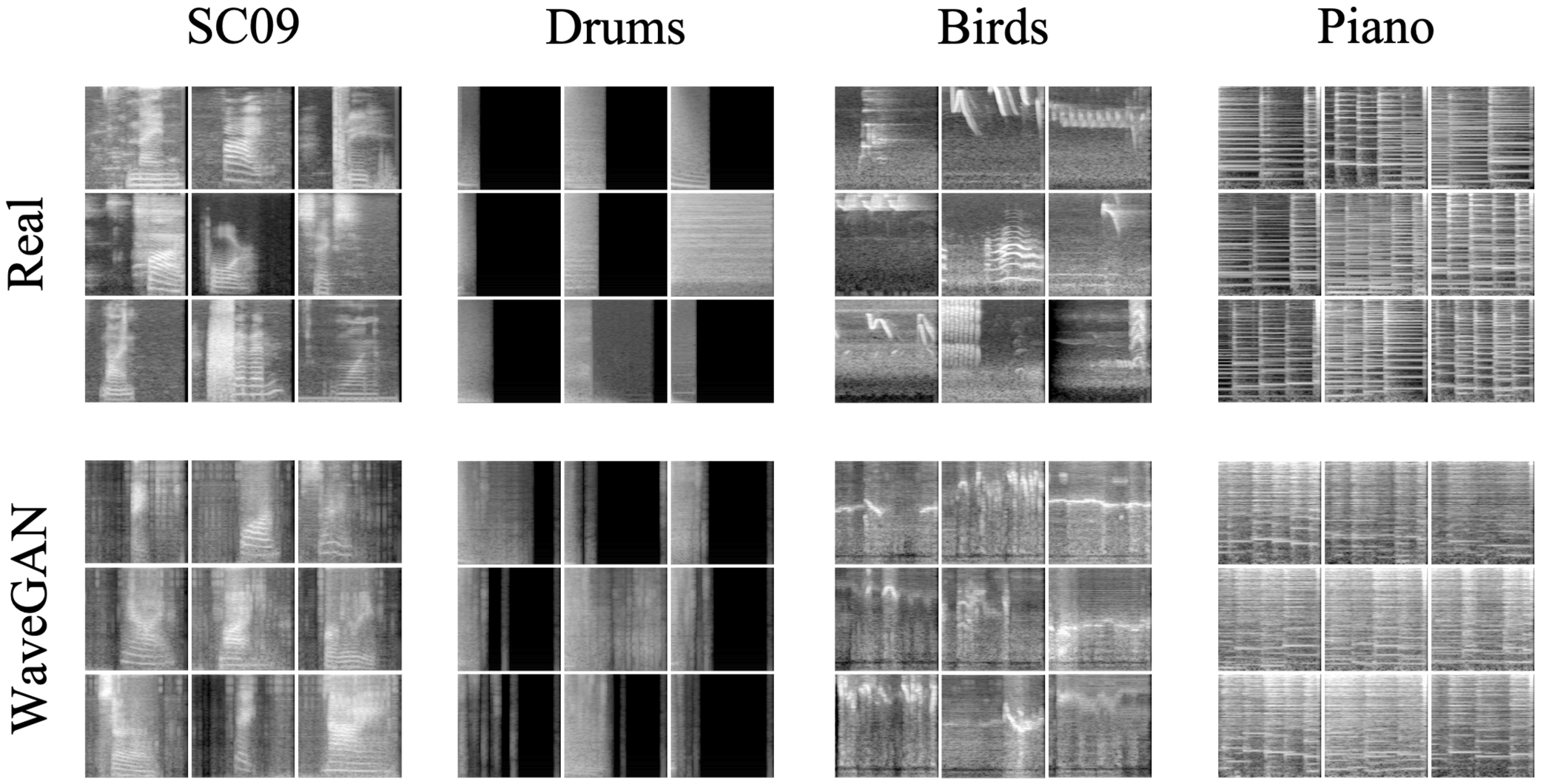
Treat it like an image and process with a CNN!



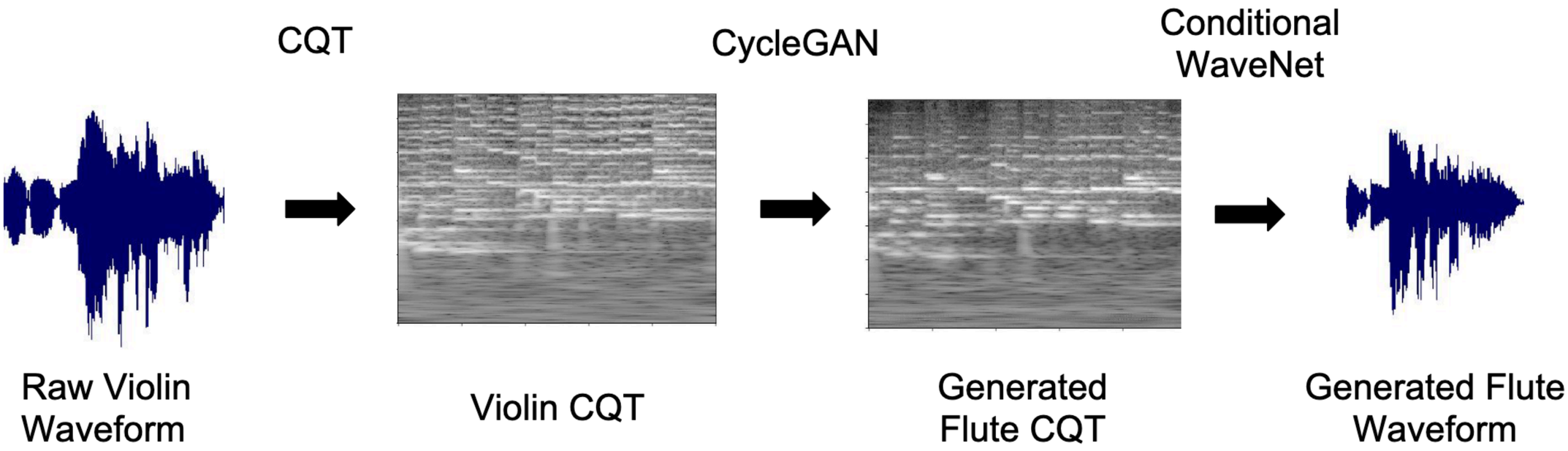
Short-time Fourier transform

Usually lower dimensional: e.g. 16,000 samples becomes 256 x 256 spectrogram

Audio generation with GANs



CycleGAN for audio



[Huang et al., “TimbreTron”, 2019]

CycleGAN for audio

[Huang et al., “TimbreTron”, 2019]

CycleGAN for audio

[Huang et al., “TimbreTron”, 2019]

CycleGAN for audio

[Huang et al., “TimbreTron”, 2019]

Domain adaptation

source domain

target domain

(where we actual use our model)

Train a model on $p_{\texttt{source}}$ test on $p_{\texttt{target}}$

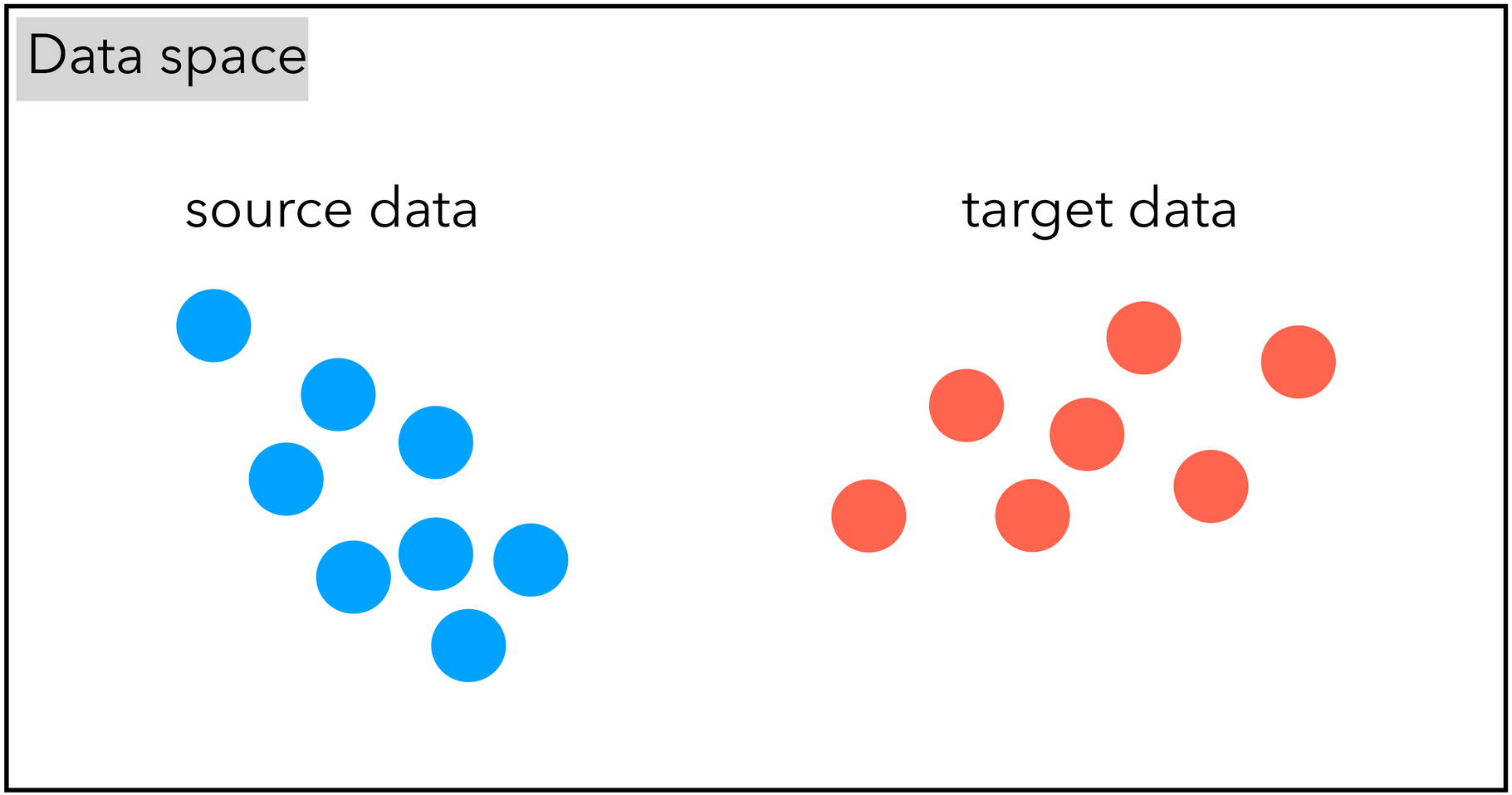
Space of images

Source data

Target data

Adapted from Isola, Torralba, Freeman

Domain adaptation



(Or vice versa)

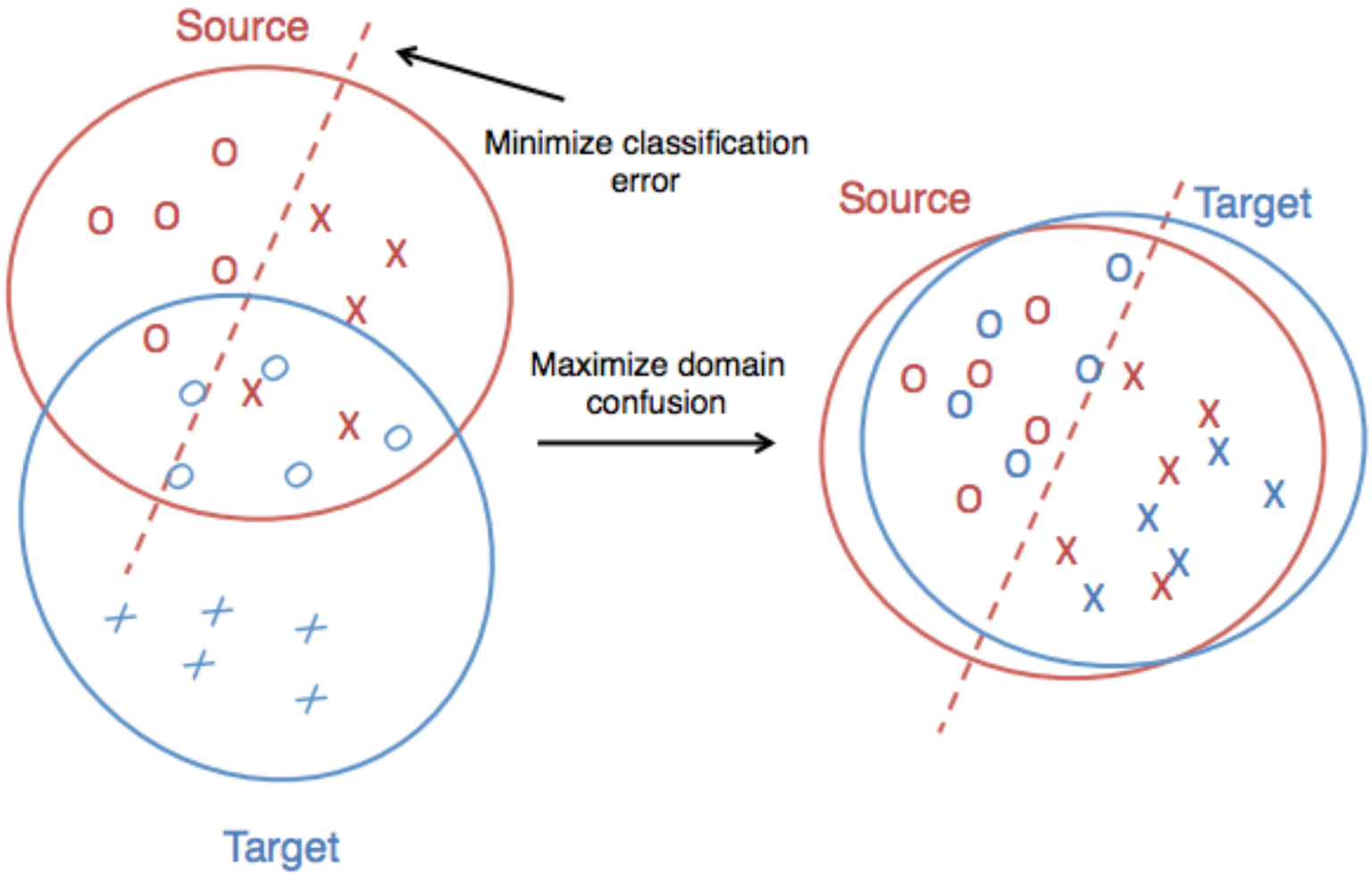
Source: Isola, Torralba, Freeman

Domain adaptation

- We have source domain pairs {$\mathbf{x}^{\mathbf{source}}$, $\mathbf{y}^{\mathbf{source}}$}
- Learn a mapping F: $\mathbf{x}^{\mathbf{source}} \rightarrow \mathbf{y}^{\mathbf{source}}$
- We want to apply F to target domain data $\mathbf{x}^{\mathbf{target}}$
- Find transformation T: $\mathbf{x}^{\mathbf{target}} \rightarrow \mathbf{x}^{\mathbf{source}}$
- Now apply F(T($\mathbf{x}^{\mathbf{target}}$)) to predict $\mathbf{y}^{\mathbf{target}}$

Source: Isola, Torralba, Freeman

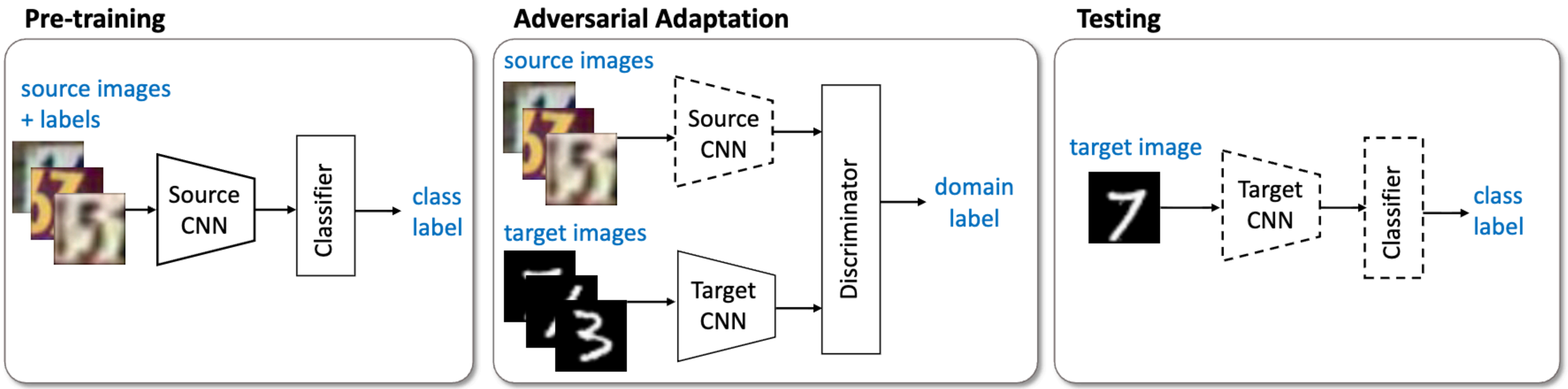
Domain Adaptation



[Tzeng et al. 2014]

Source: Isola, Torralba, Freeman

Domain adaptation

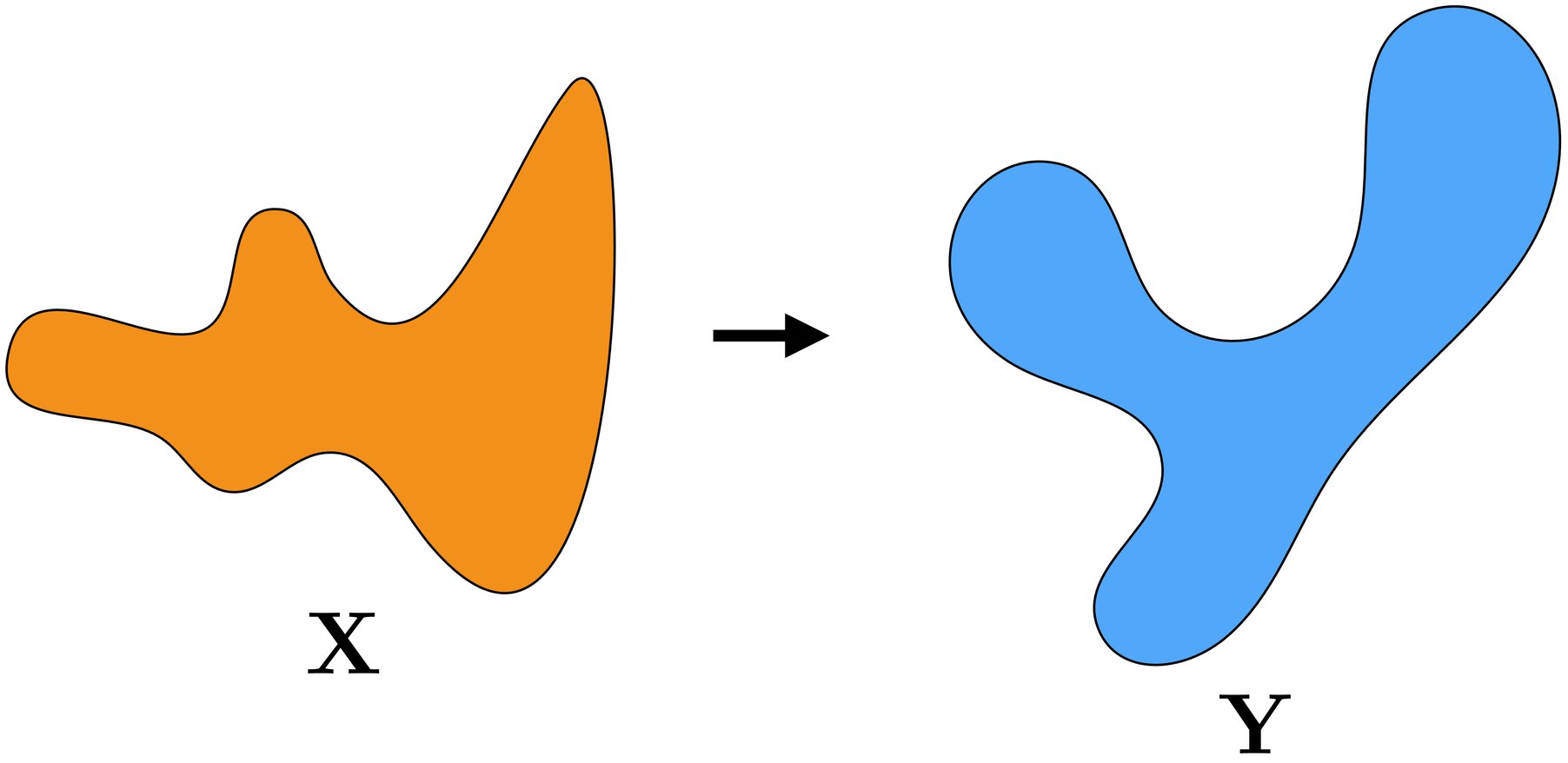


[Tzeng et al., "Adversarial Discriminative Domain Adaptation", CVPR 2017]

CycleGAN

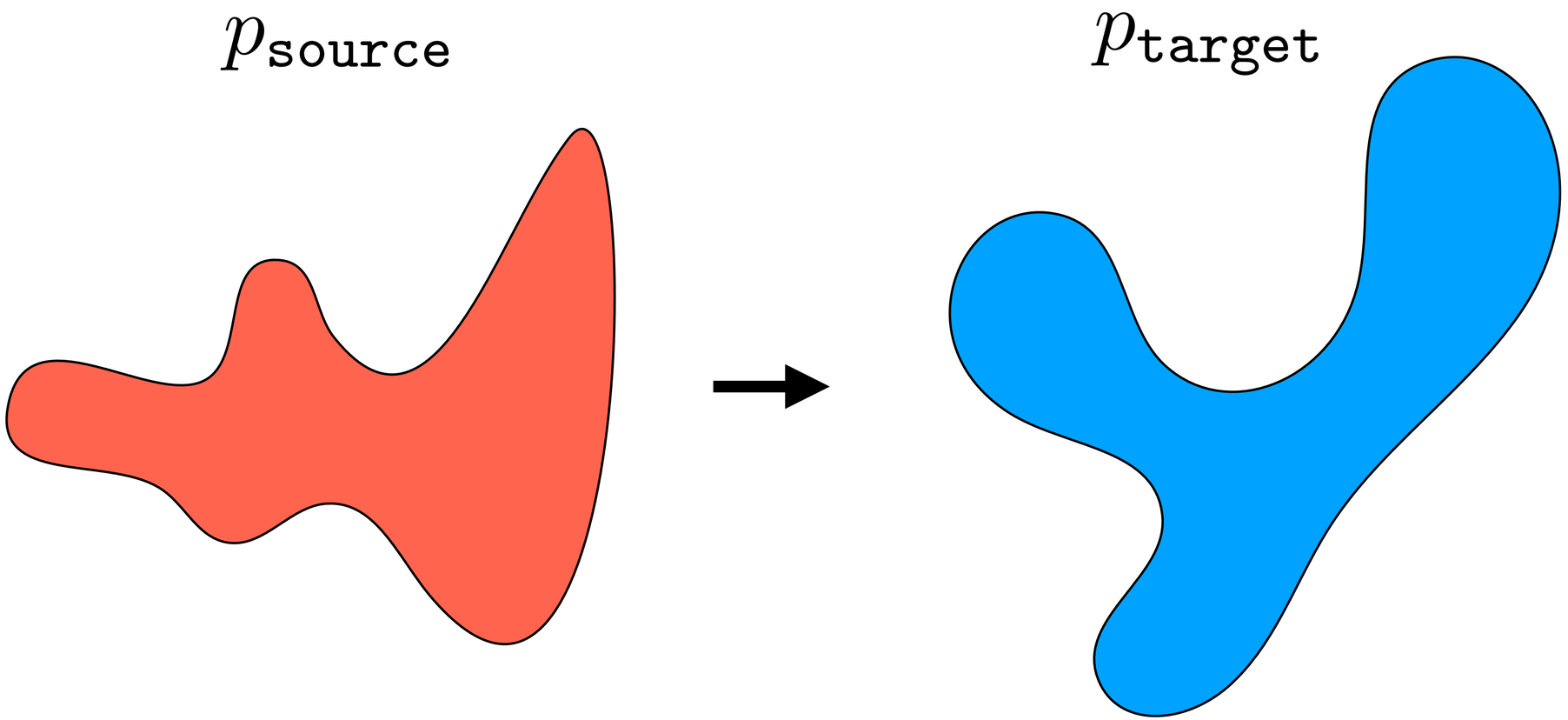
Horses

Zebras



Source: Isola, Torralba, Freeman

Domain adaptation

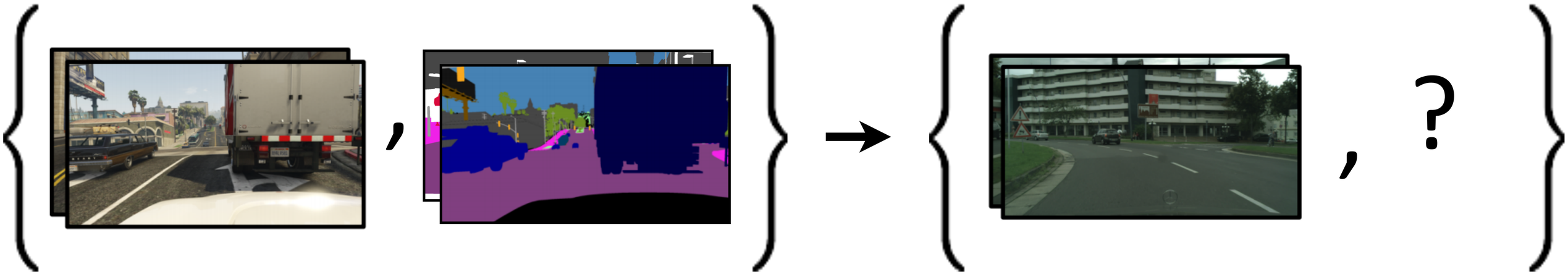


Source: Isola, Torralba, Freeman

Domain adaptation by image translation

Source domain

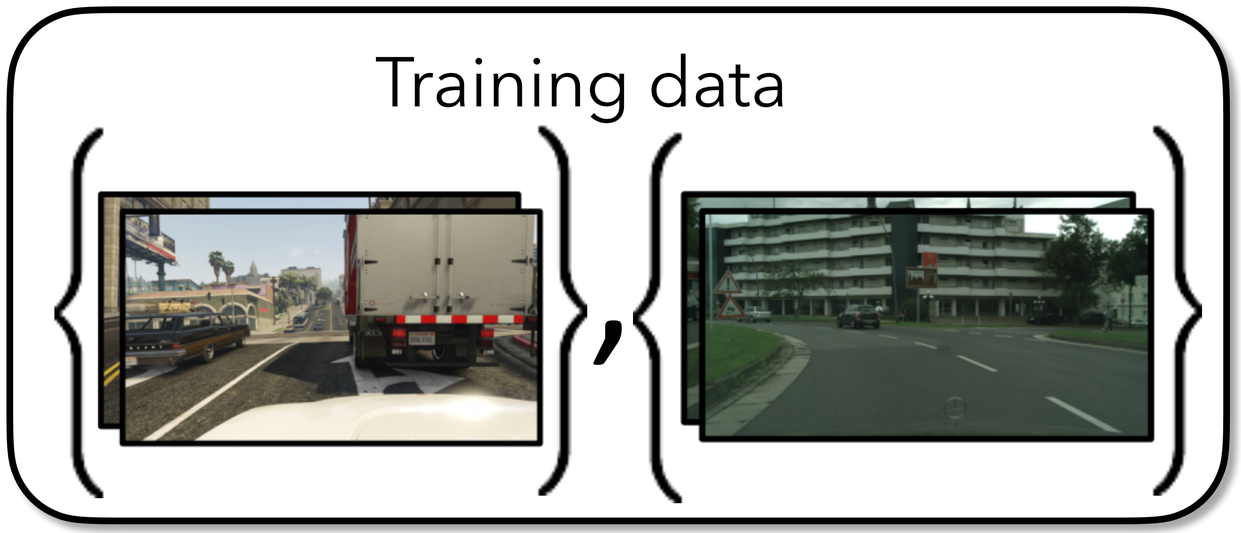
Target domain



[Hoffman, Tzeng, Park, Zhu, Isola, Saenko, Darrell, Efros, “CyCADA”, 2017]

Source: Isola, Torralba, Freeman

CycleGAN



Source: Isola, Torralba, Freeman

CycleGAN

Training data

,

Source: Isola, Torralba, Freeman

Image synthesis with GANs

- We can generate images, audio (and other continuous signals) with conditional and unconditional GANs.
- Translate between two domains without paired data using cycle consistency.
- GANs can be hard to train, and thus sometimes require bespoke architectures.
- No built-in encoder (unlike VAEs or normalizing flow)
- GANs learn useful internal representations that are useful for image manipulation.

Next class: energy-based models