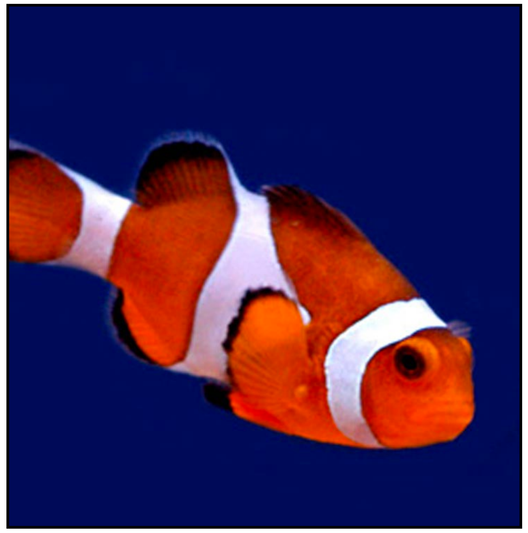
Synthesizing Images with Generative Adversarial Networks

Phillip Isola
OpenAI/Berkeley
10/26/17

CS194: Image Manipulation & Computational Photography
Alexei Efros, UC Berkeley, Fall 2017

Image classification



Learner

“Fish”

image X

label Y

Image classification

image X

Learner

“Fish”

label Y

Image classification

Learner

“Fish”

image X

label Y

Image classification

Learner

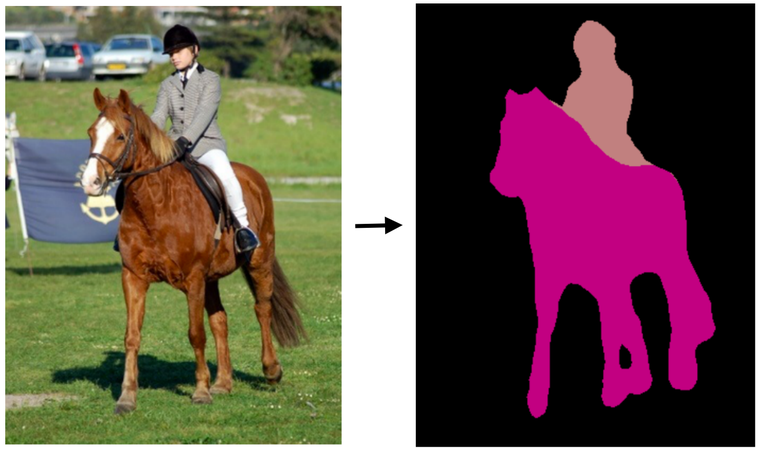
“Fish”

image X

label Y

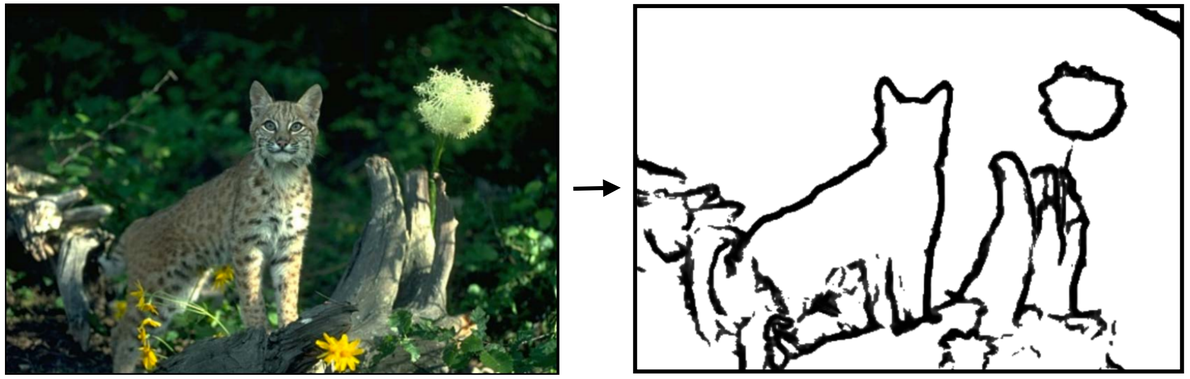
Image prediction ("structured prediction")

Object labeling



[Long et al. 2015, …]

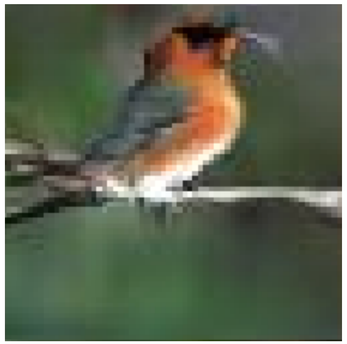
Edge Detection



[Xie et al. 2015, …]

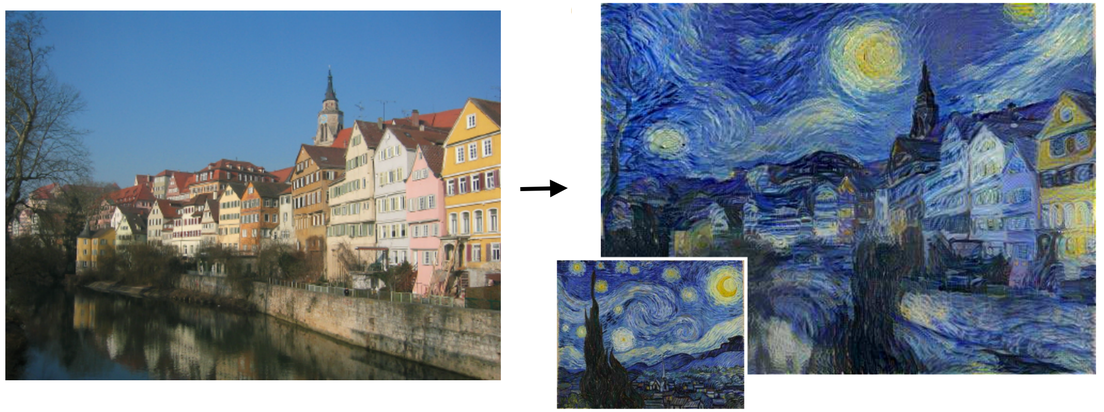
Text-to-photo

"this small bird has a pink breast and crown…"



[Reed et al. 2014, …]

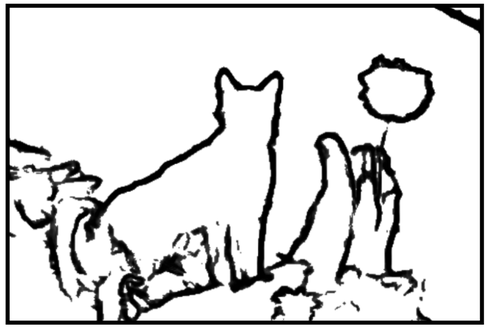
Style transfer



[Gatys et al. 2016, …]

Challenges

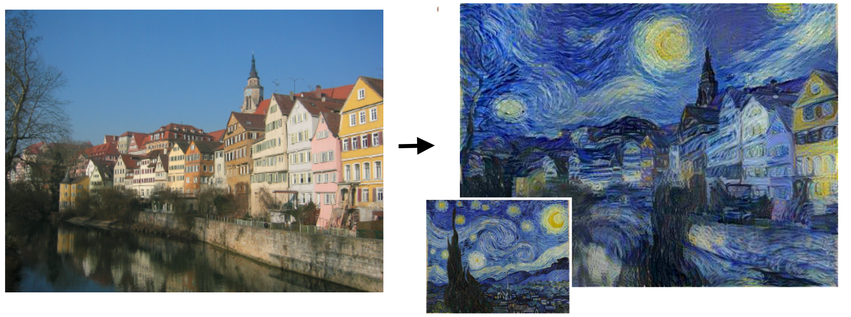
1. Output is high-dimensional, structured object



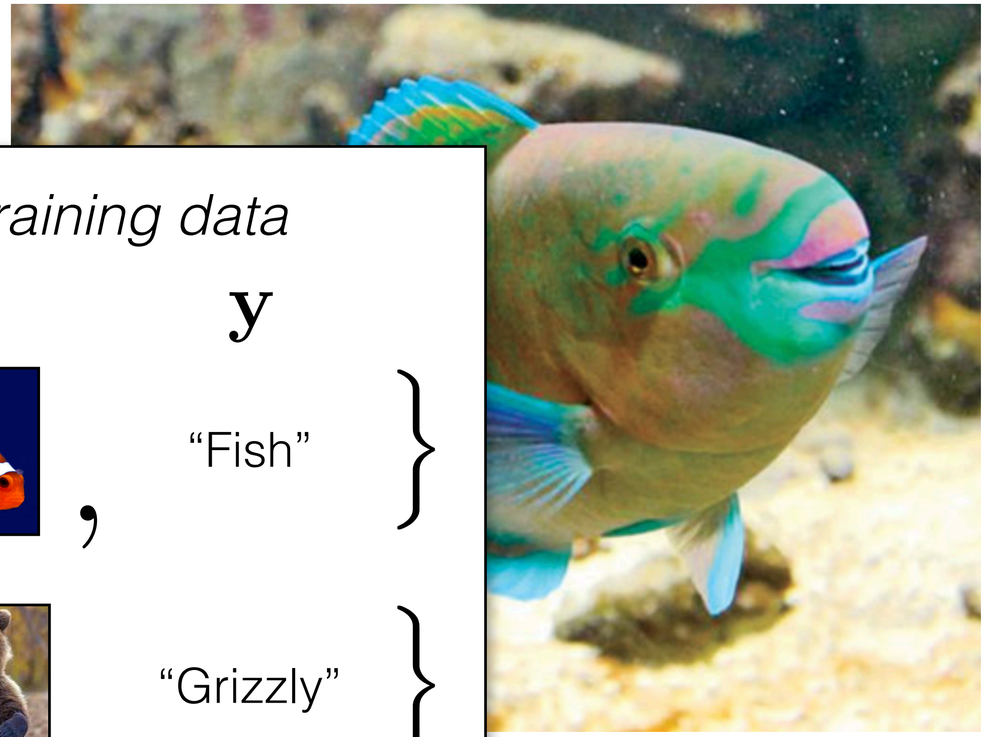
2. Uncertainty in mapping; many plausible outputs

“this small bird has a pink breast and crown…”

3. Lack of supervised training data



$\mathbf{x}$



Training data

$\mathbf{x}$	$\mathbf{y}$
	“Fish”
	“Grizzly”
	“Chameleon”
⋮	

$\mathbf{y}$

$\mathcal{F}$ → “Fish”

$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[L(\mathcal{F}(\mathbf{x}), \mathbf{y})]$$

Objective function (loss)

Neural Network

$\mathbf{x}$

$\mathcal{F}$

$\mathbf{y}$

"Fish"

$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[L(\mathcal{F}(\mathbf{x}), \mathbf{y})]$$

"**What** should I do"

"**How** should I do it?"

Basic loss functions

Prediction: $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{x})$ Truth: $\mathbf{y}$

Classification (cross-entropy):

$$L(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_i \hat{\mathbf{y}}_i \log \mathbf{y}_i$$

How many extra bits it takes to correct the predictions

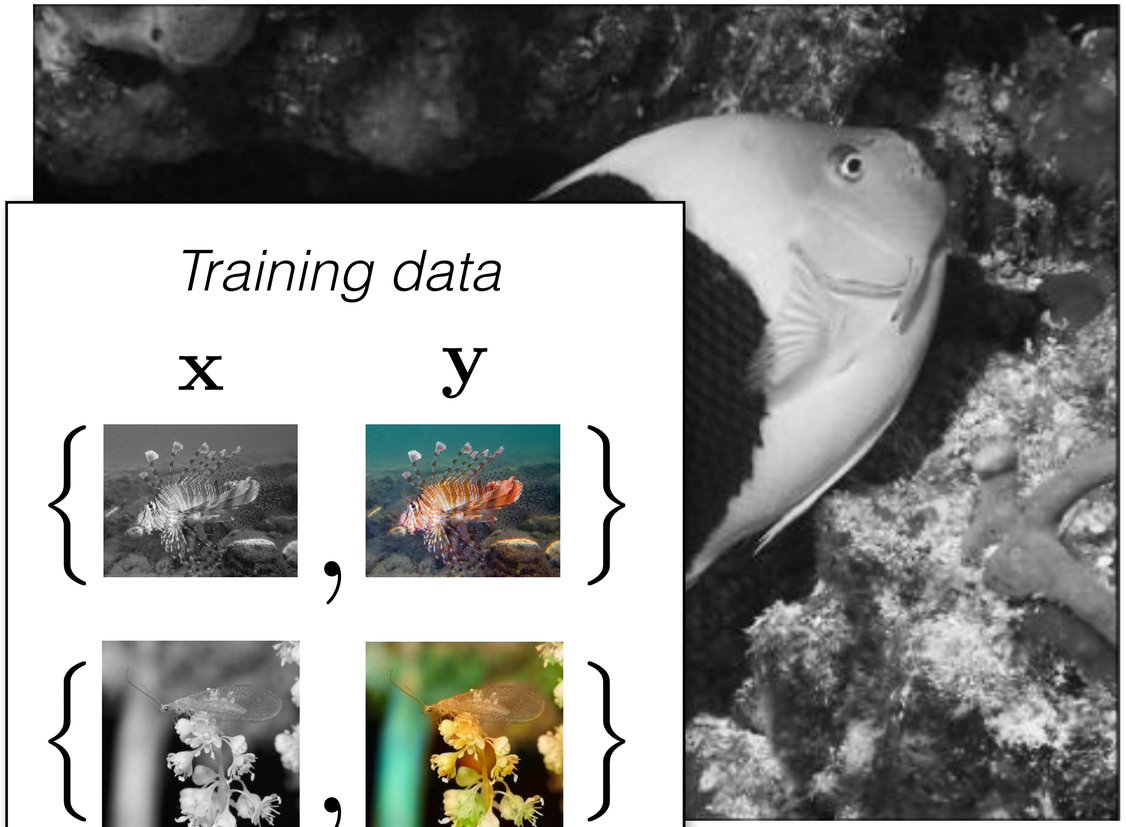
Least-squares regression:

$$L(\hat{\mathbf{y}}, \mathbf{y}) = \|\hat{\mathbf{y}} - \mathbf{y}\|_2$$

How far off we are in Euclidean distance

$\mathbf{x}$

$\mathbf{y}$



Training data

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

$\{$, $\}$

$\{$, $\}$

$\{$  ,  $\}$

$\vdots$

$\mathcal{F}$

L channel

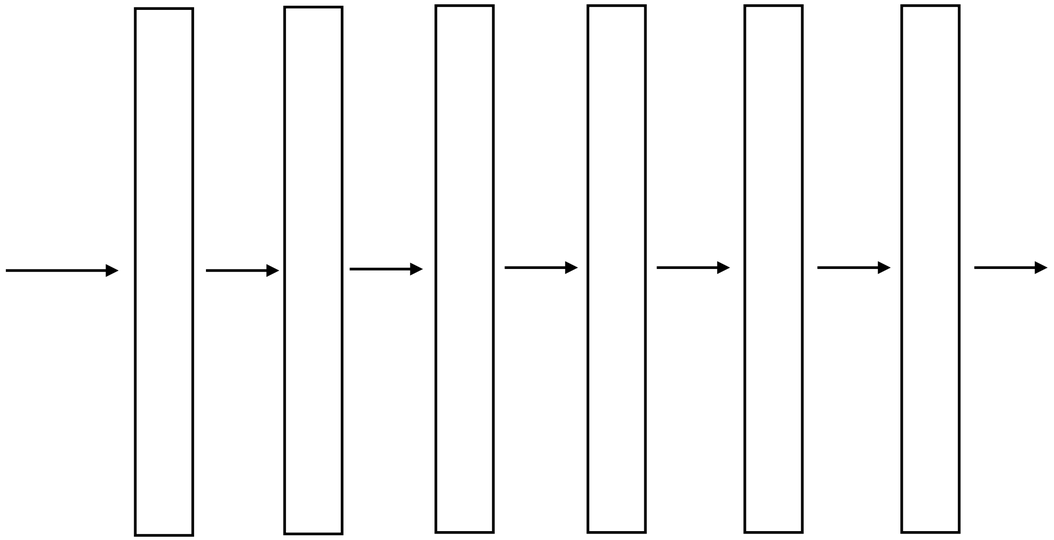
Color information: ab channels

$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[L(\mathcal{F}(\mathbf{x}), \mathbf{y})]$$

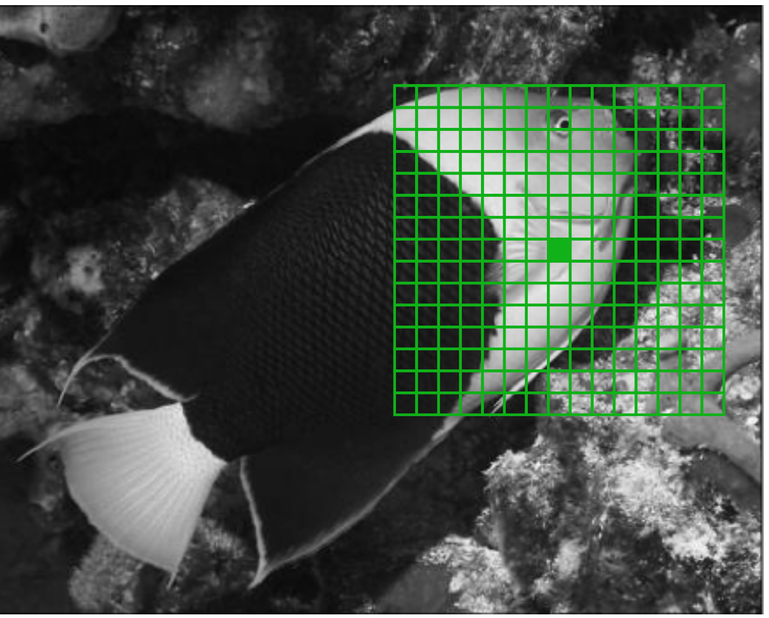
Objective function (loss)

Neural Network

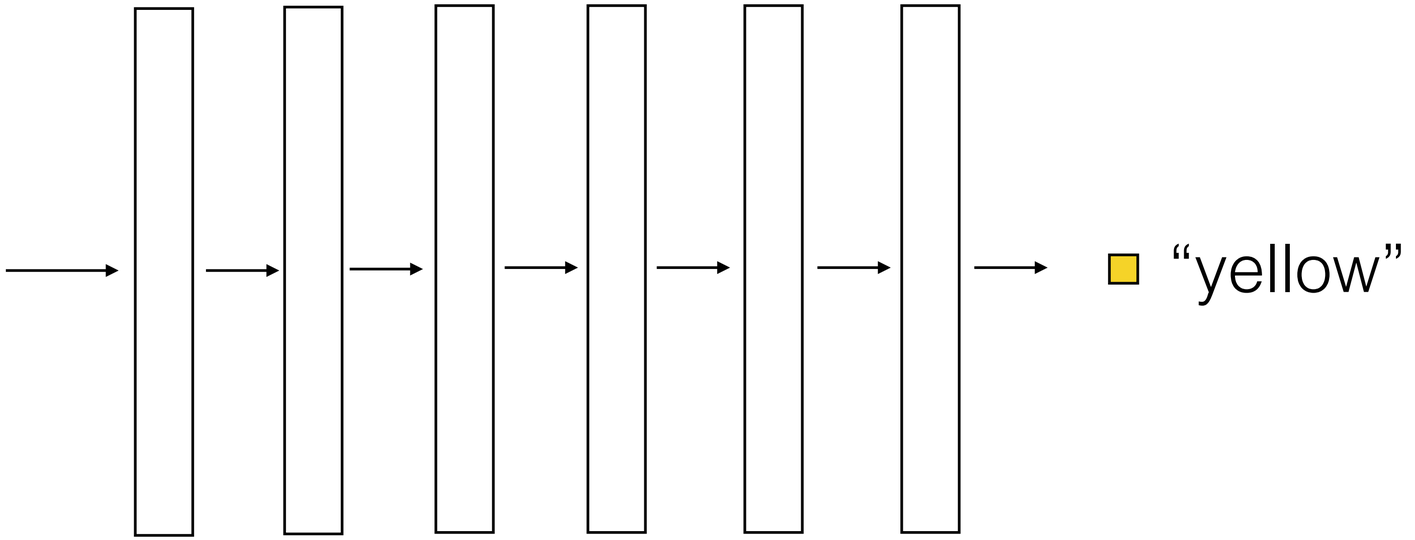
...



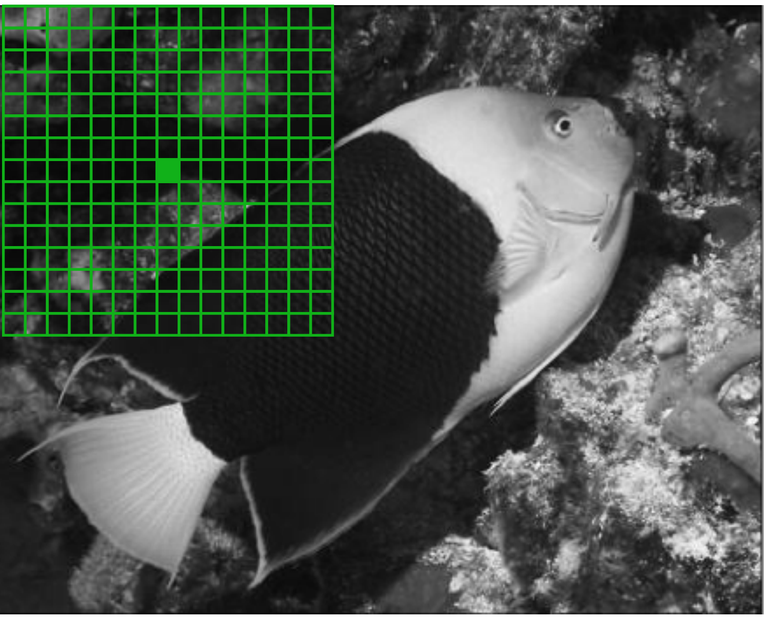
“rockfish”

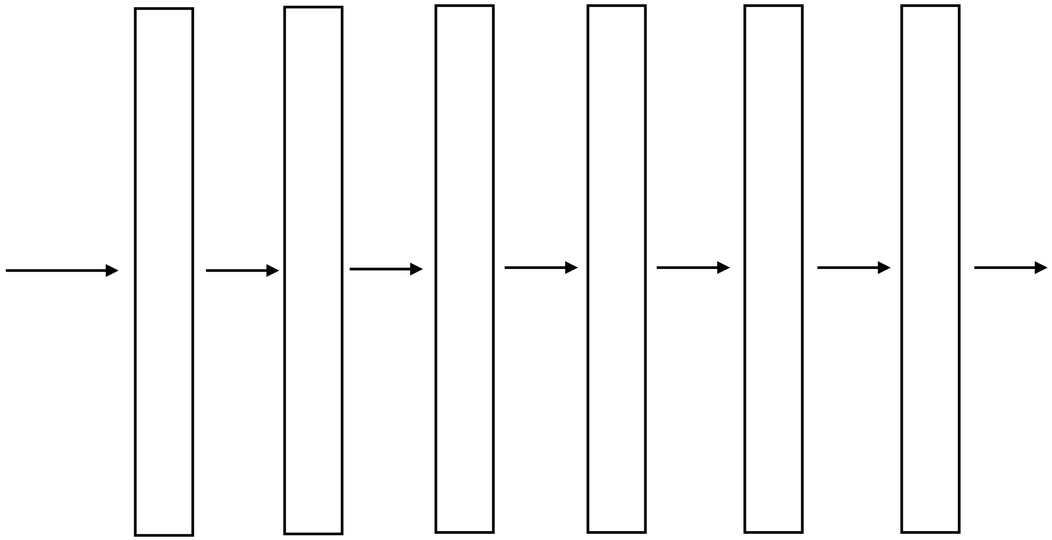


...



“yellow”





...

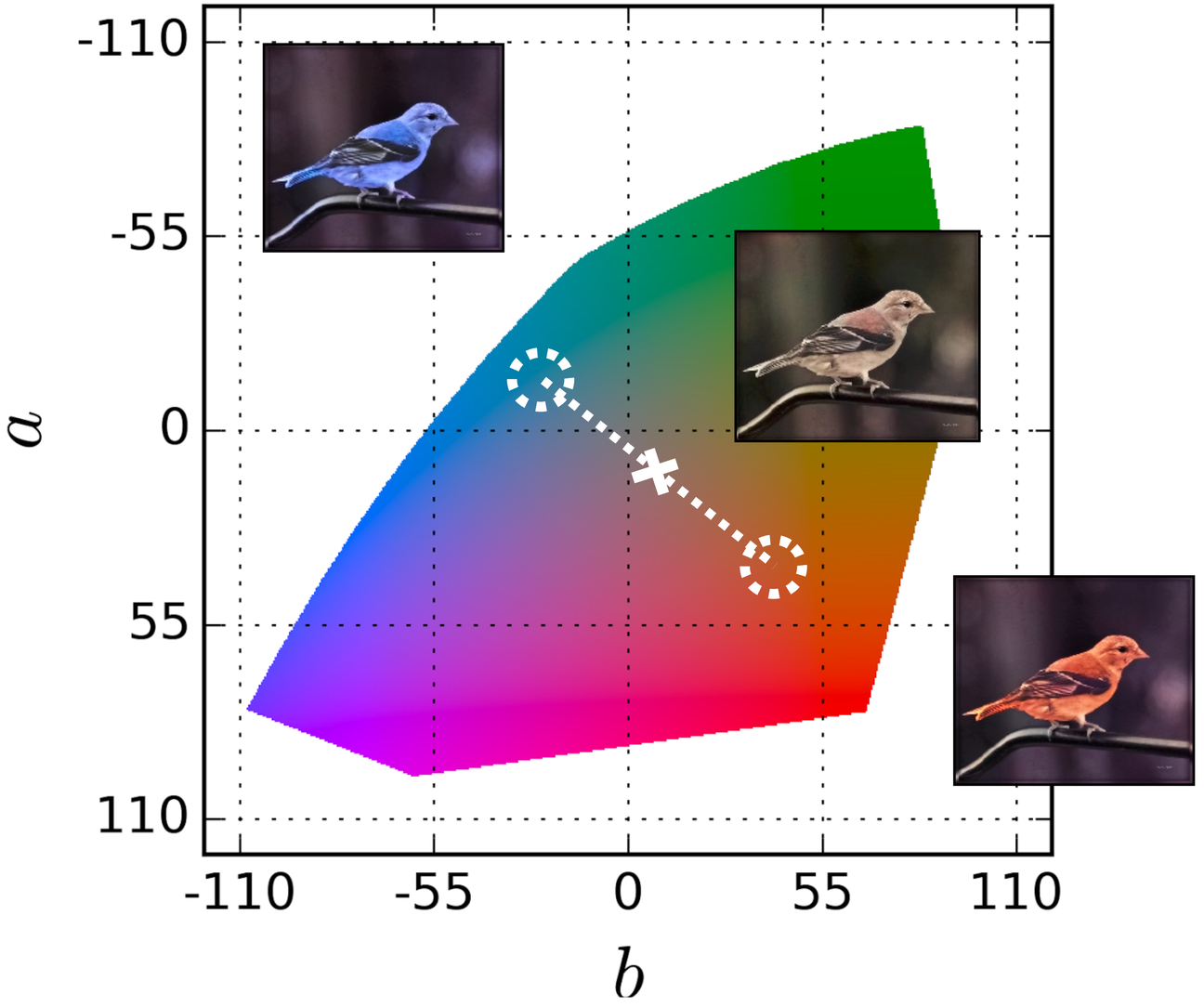
Designing loss functions

Input

Output

Ground truth

$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$



$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

Designing loss functions

Input

Zhang et al. 2016

Ground truth

Color distribution cross-entropy loss with colorfulness enhancing term.

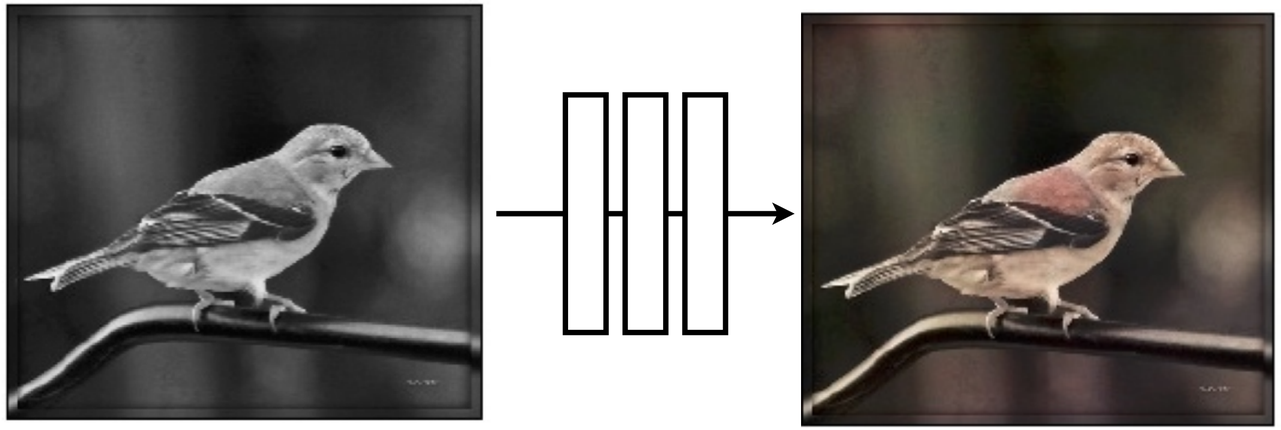
[Zhang, Isola, Efros, ECCV 2016]

Designing loss functions

Be careful what you wish for!

Designing loss functions

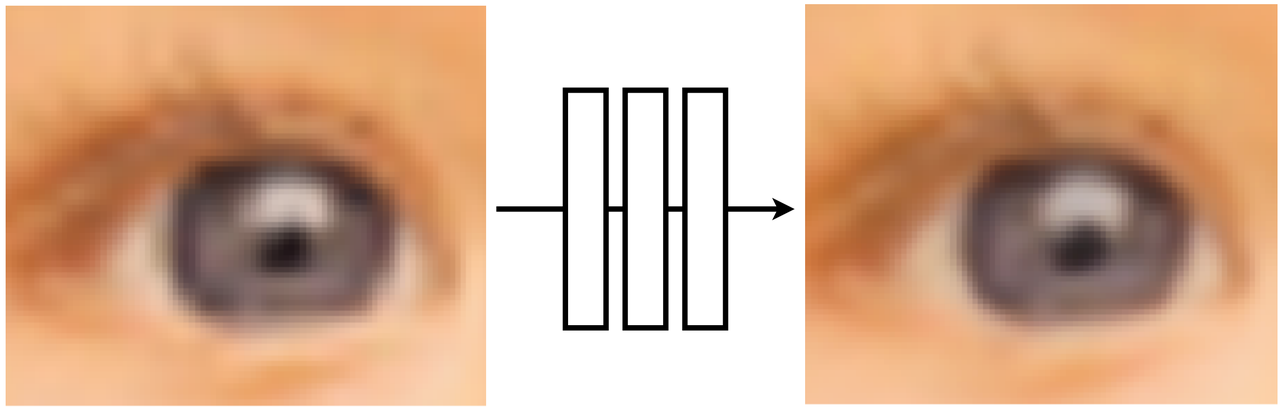
Image colorization



[Zhang, Isola, Efros, ECCV 2016]

L2 regression

Super-resolution

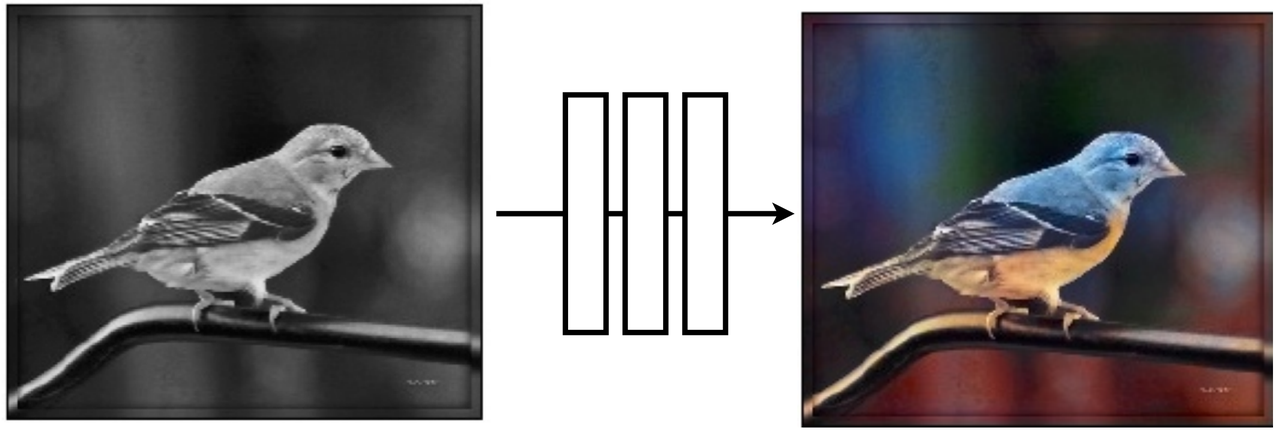


[Johnson, Alahi, Li, ECCV 2016]

L2 regression

Designing loss functions

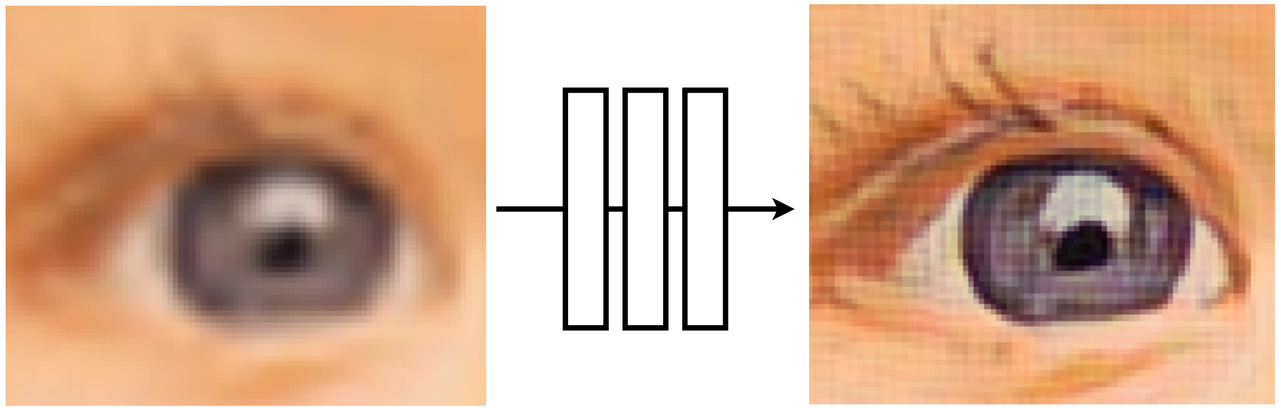
Image colorization



[Zhang, Isola, Efros, ECCV 2016]

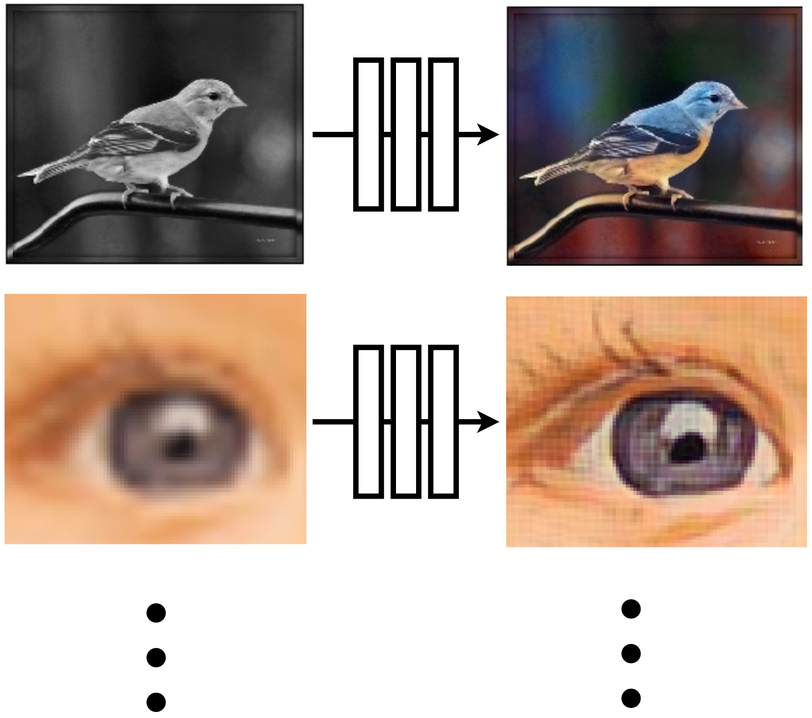
Cross entropy objective,
with colorfulness term

Super-resolution



[Johnson, Alahi, Li, ECCV 2016]

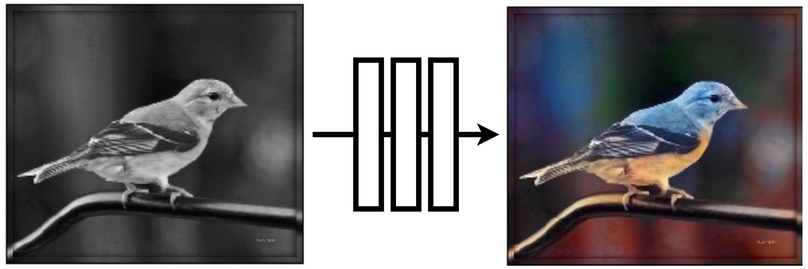
Deep feature covariance
matching objective

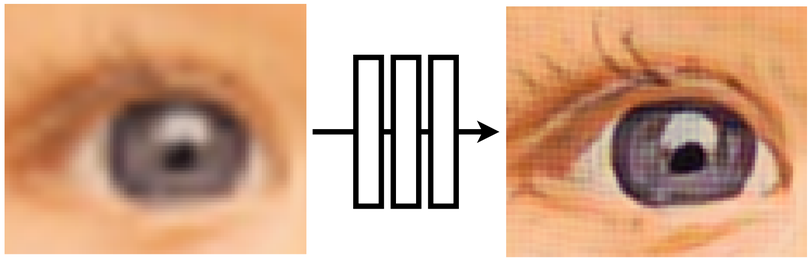


Universal loss?

“Generative Adversarial Network” (GANs)

Generated images





Generated vs Real (classifier)

Real photos

…

[Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, Bengio 2014]

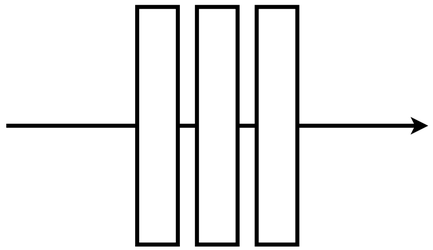
Conditional GANs

[Goodfellow et al., 2014]

[Isola et al., 2017]

$\mathbf{x}$

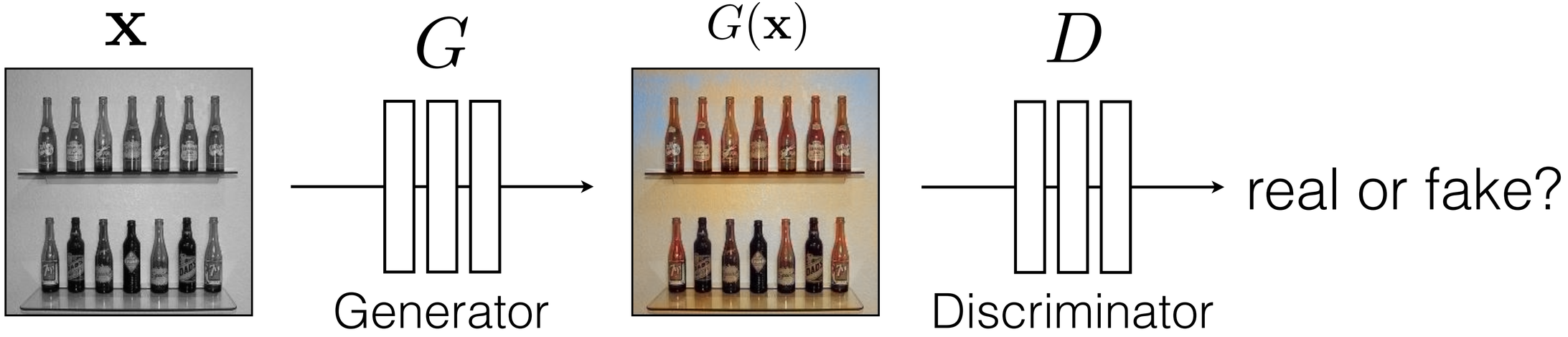
G



Generator

$G(\mathbf{x})$

[Goodfellow et al., 2014]



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

D tries to identify the fakes

[Goodfellow et al., 2014]

$\mathbf{x}$

G

$G(\mathbf{x})$

D

fake (0.9)

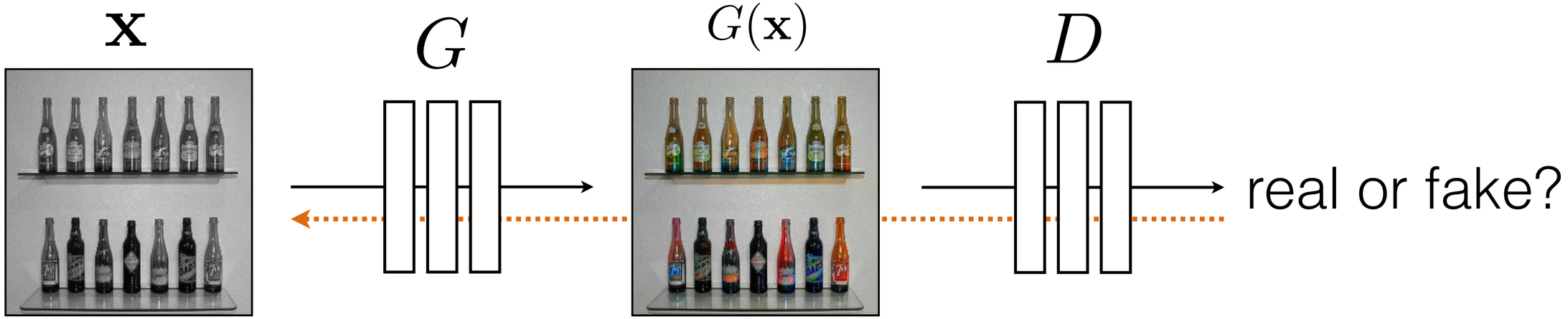
$\mathbf{y}$

D

real (0.1)

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

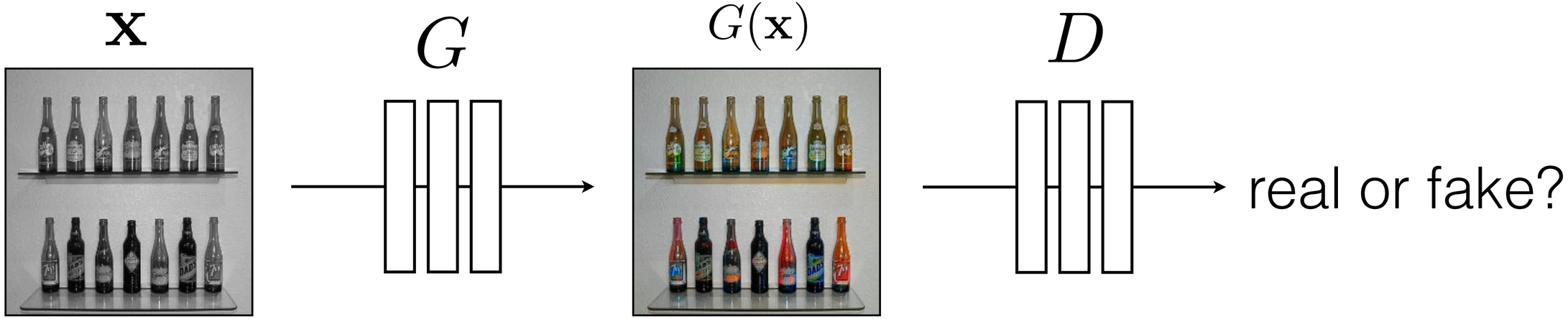
[Goodfellow et al., 2014]



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

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

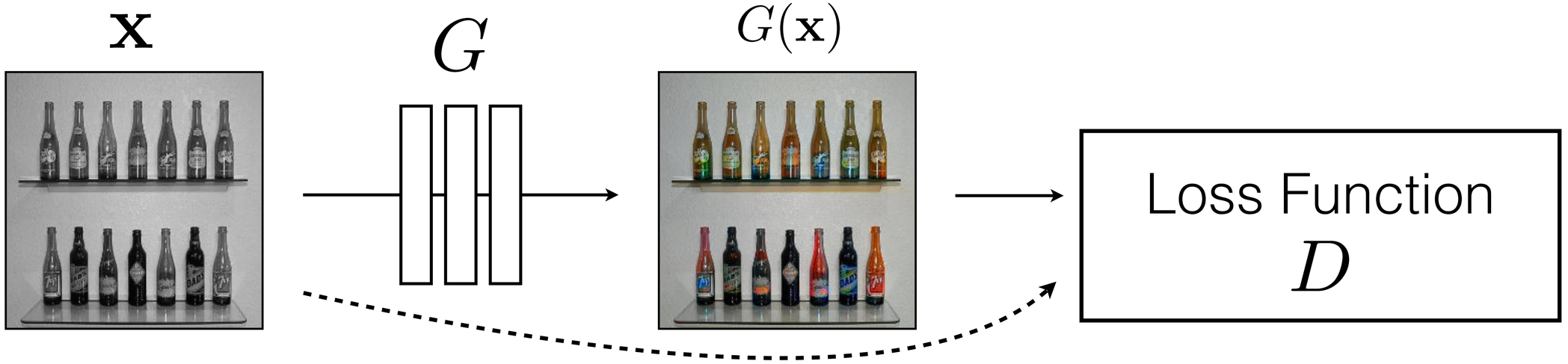
[Goodfellow et al., 2014]



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

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

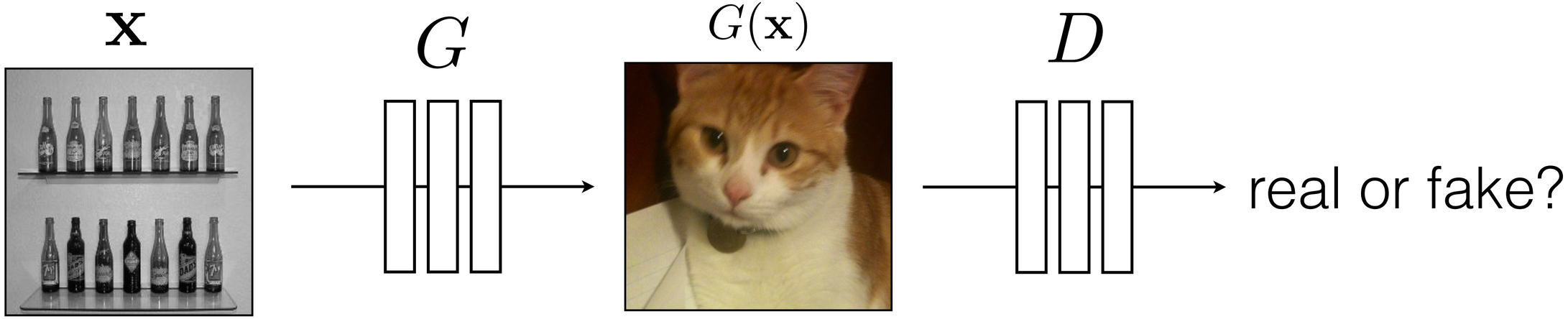
[Goodfellow et al., 2014]



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

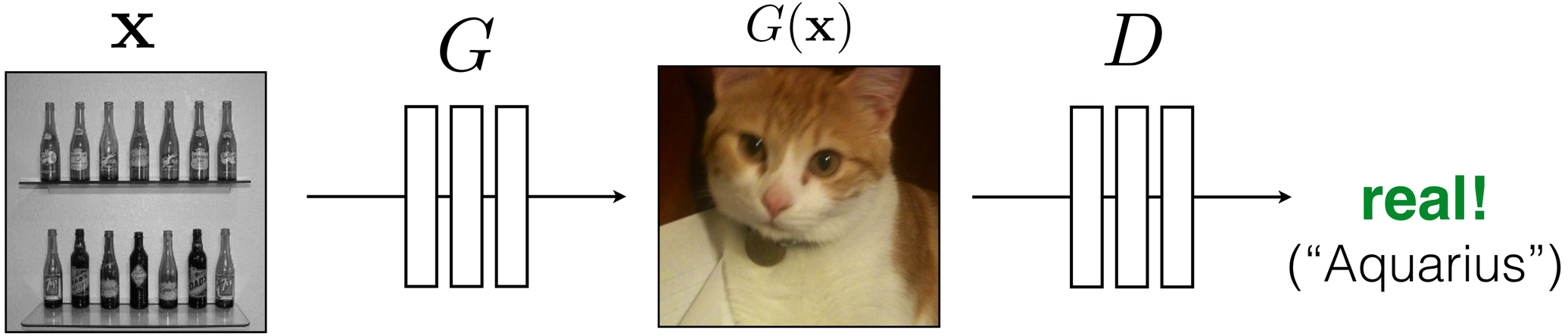
Rather than being hand-designed, it is *learned*.

[Goodfellow et al., 2014]
[Isola et al., 2017]



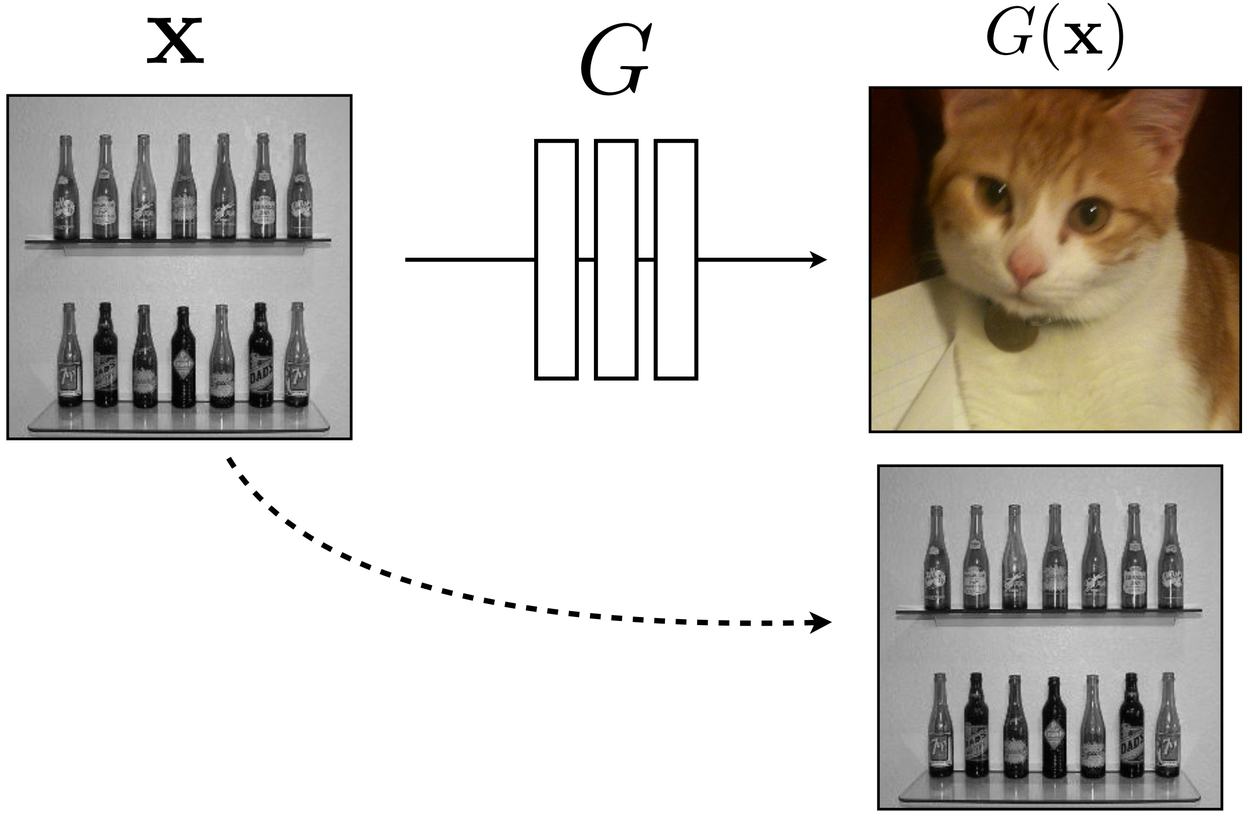
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y})) \;\;]$$

[Goodfellow et al., 2014]



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

[Goodfellow et al., 2014]



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

[Goodfellow et al., 2014]

[Isola et al., 2017]

$\mathbf{x}$

G

$G(\mathbf{x})$

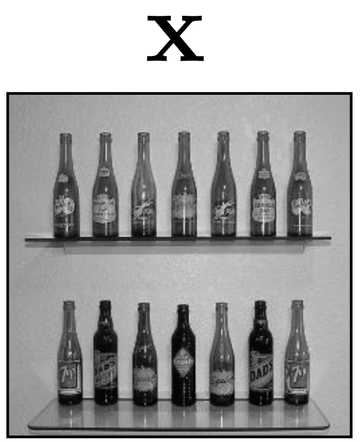
D

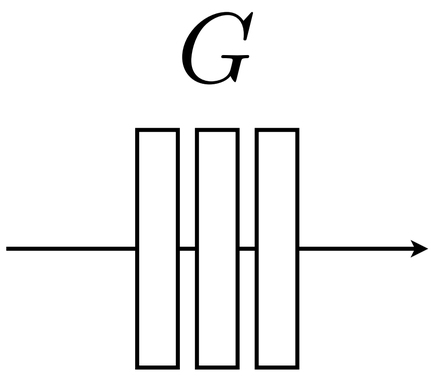
real or fake *pair* ?

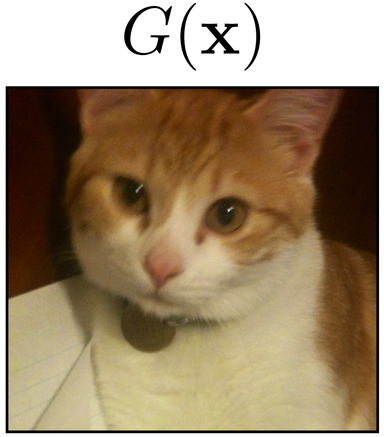
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$

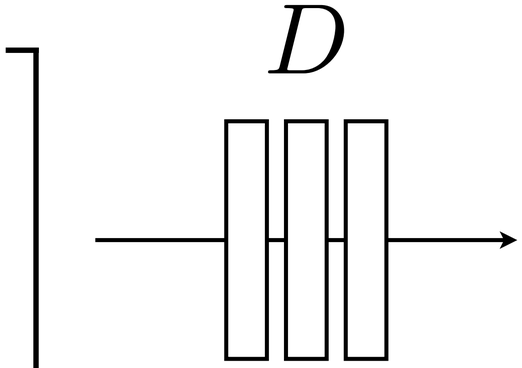
[Goodfellow et al., 2014]

[Isola et al., 2017]









fake *pair*

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

[Goodfellow et al., 2014]

[Isola et al., 2017]

$\mathbf{x}$

G

$G(\mathbf{x})$

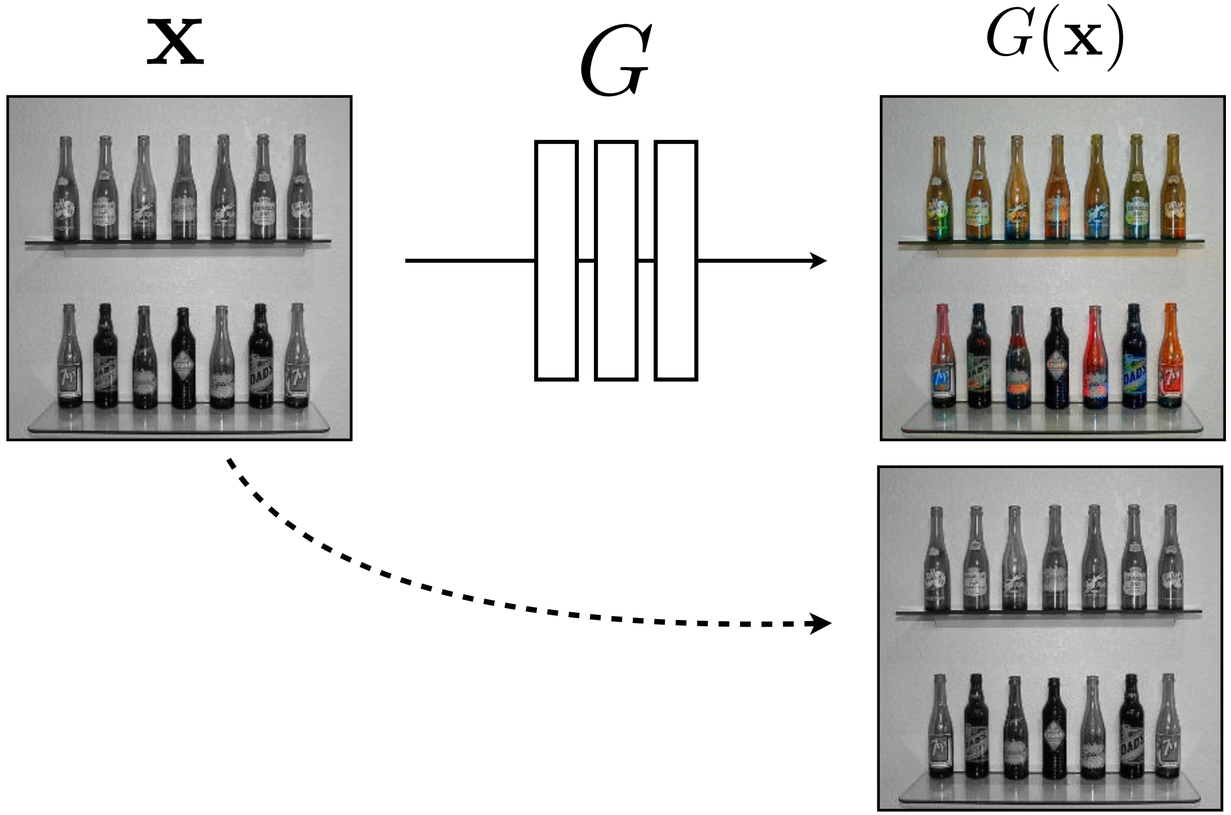
D

real *pair*

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

[Goodfellow et al., 2014]

[Isola et al., 2017]



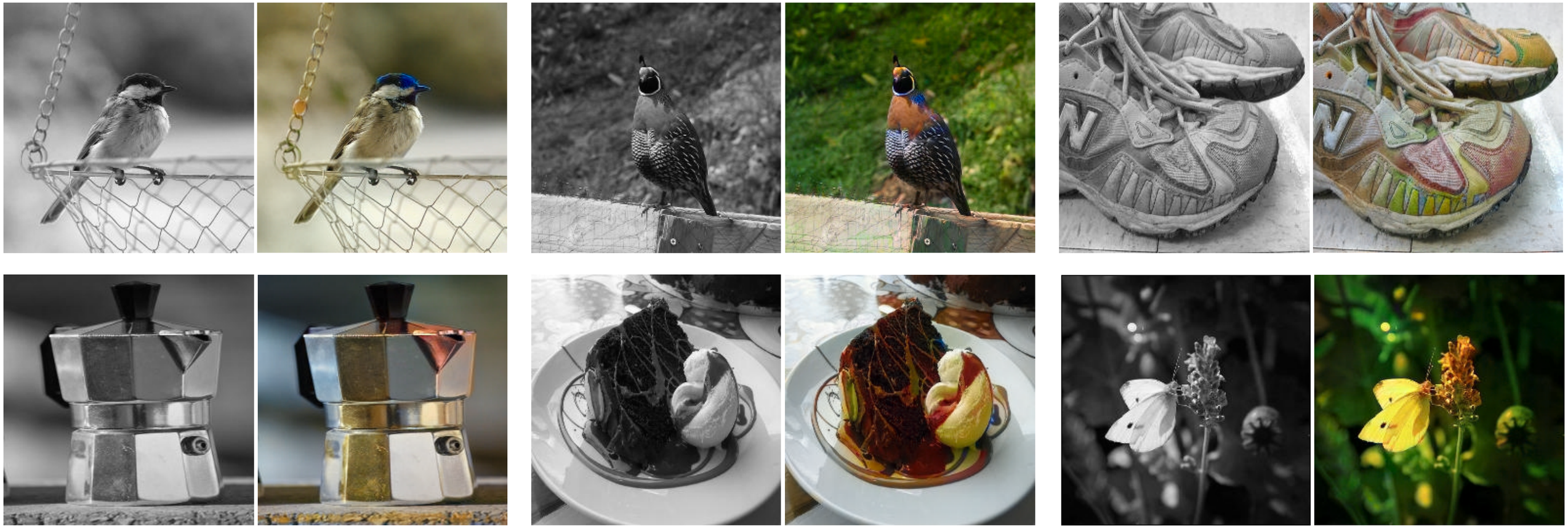
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;\;]$$

[Goodfellow et al., 2014]

[Isola et al., 2017]

BW → Color

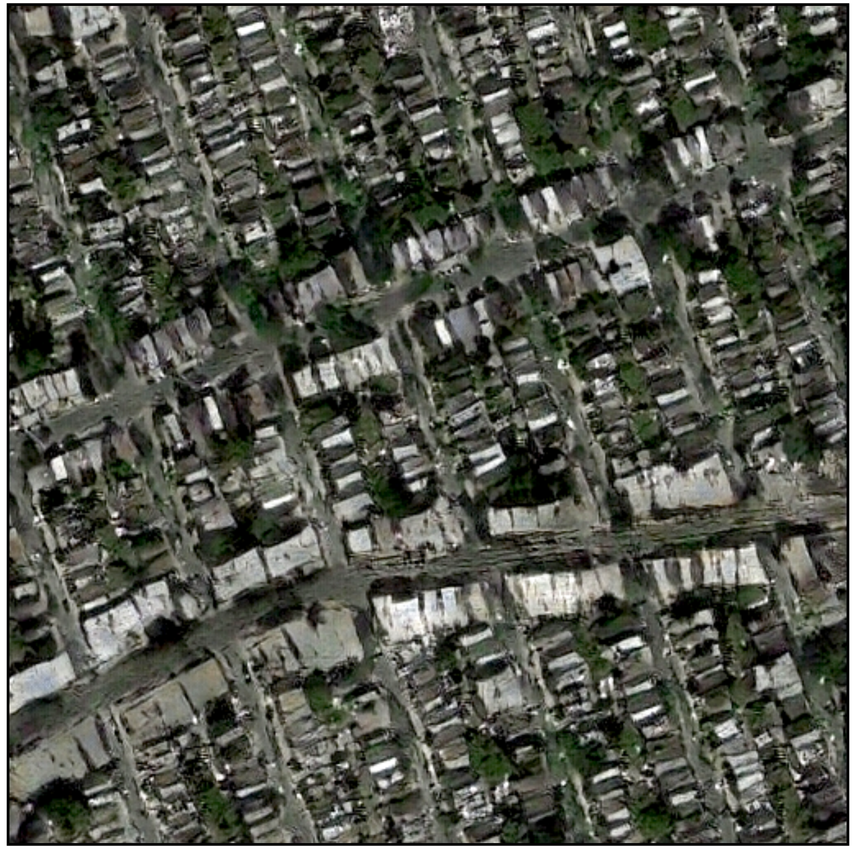
Input Output Input Output Input Output



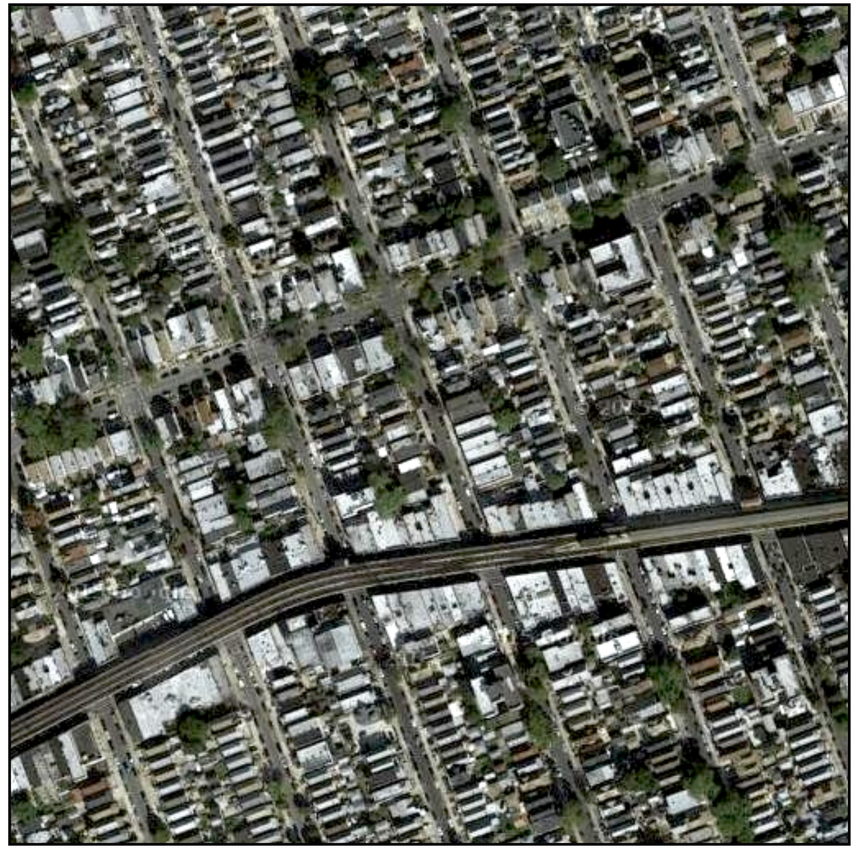
Data from [Russakovsky et al. 2015]

Input

Output



Groundtruth



Data from
[maps.google.com]

Input

Output

Groundtruth

Data from [maps.google.com]

Edges → Images

Input | Output | Input | Output | Input | Output

Edges from [Xie & Tu, 2015]

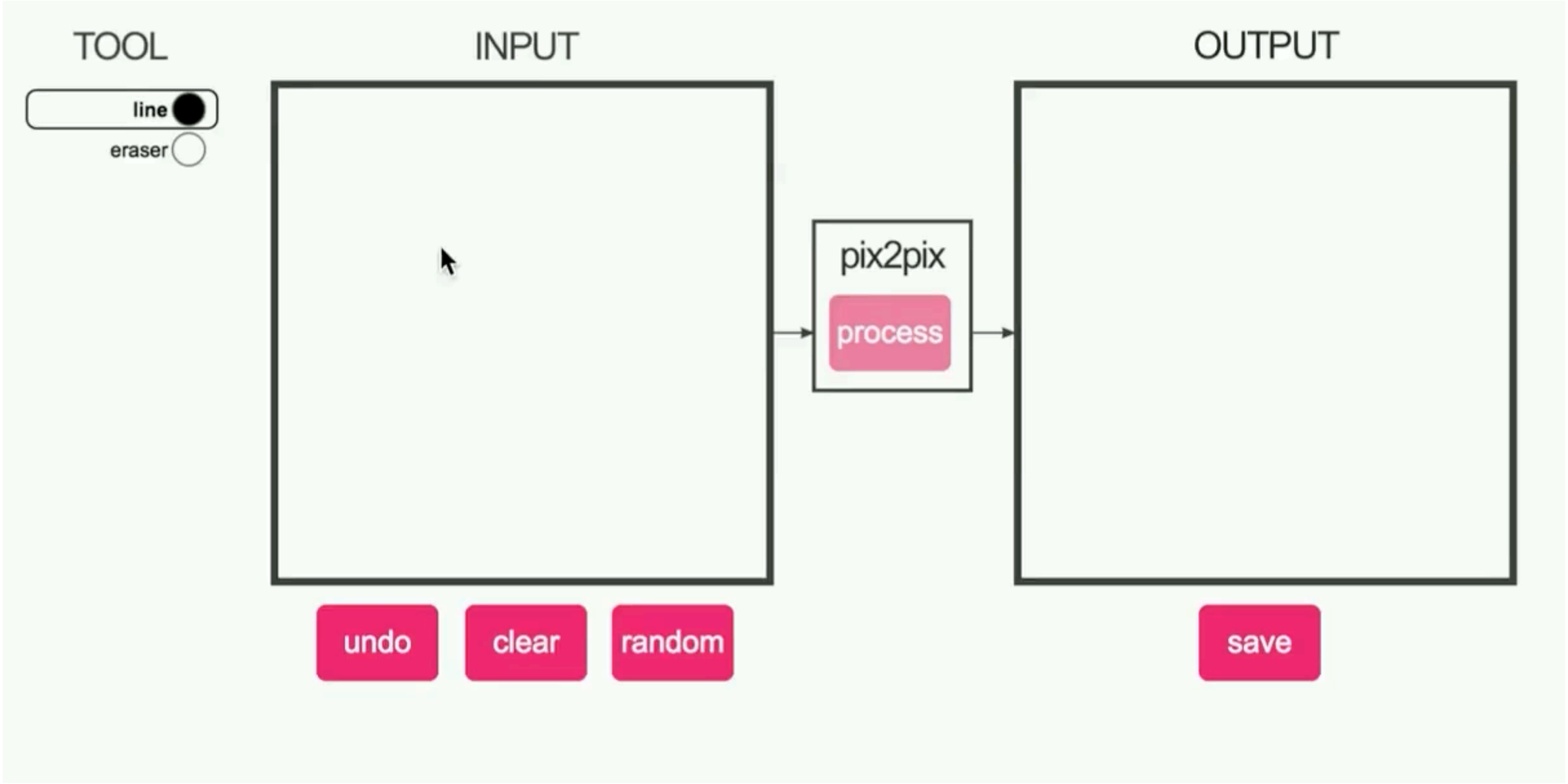
Sketches → Images

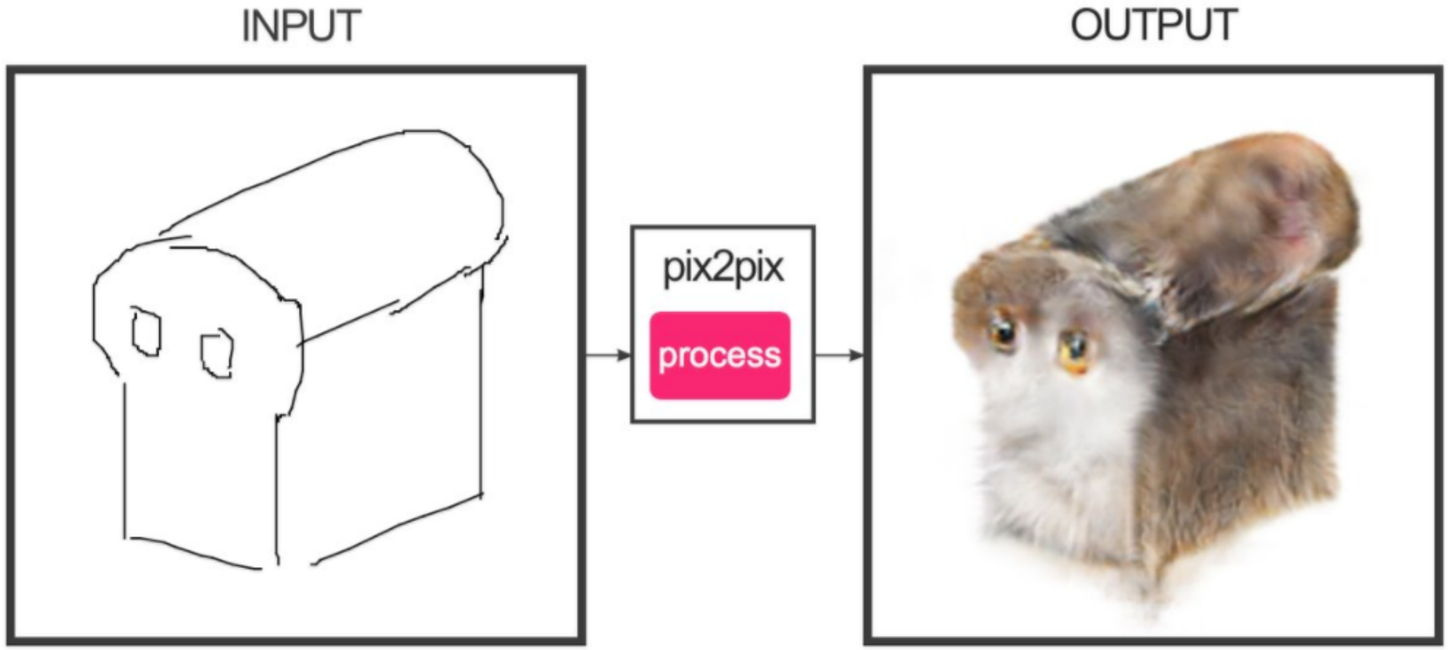
Input Output Input Output Input Output

Trained on Edges → Images

Data from [Eitz, Hays, Alexa, 2012]

#edges2cats [Chris Hesse]



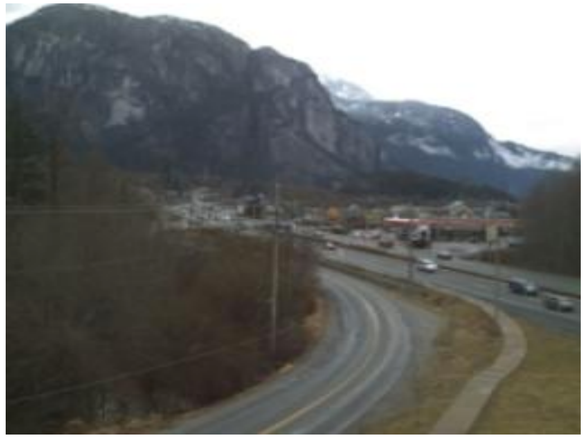


Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

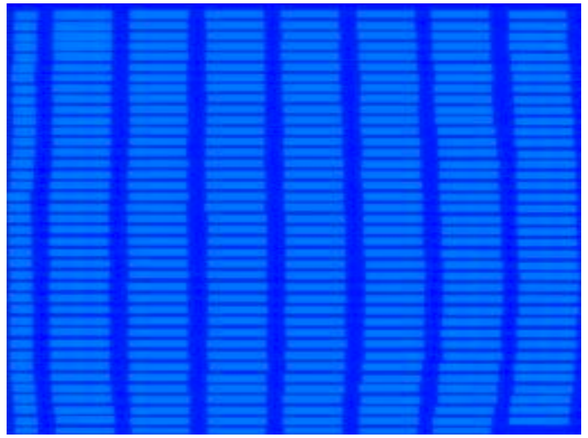
Hallucinations

Input

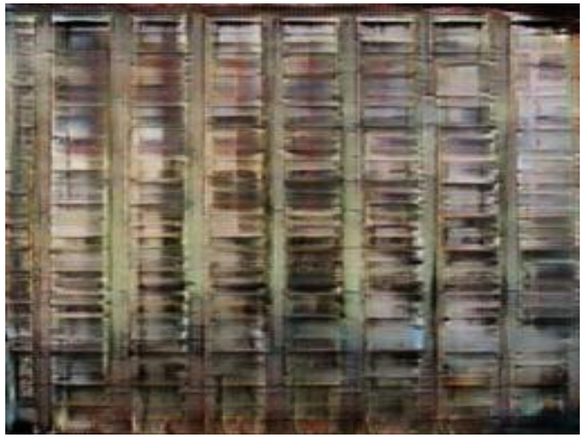


Output

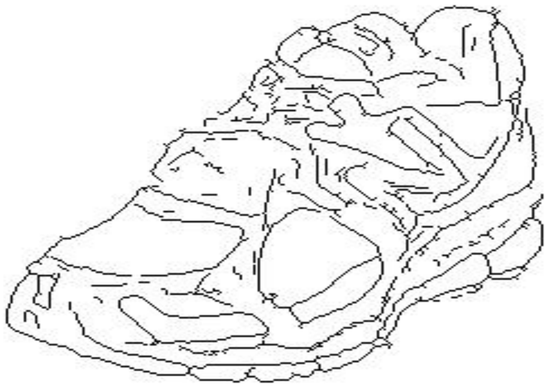
Input



Output

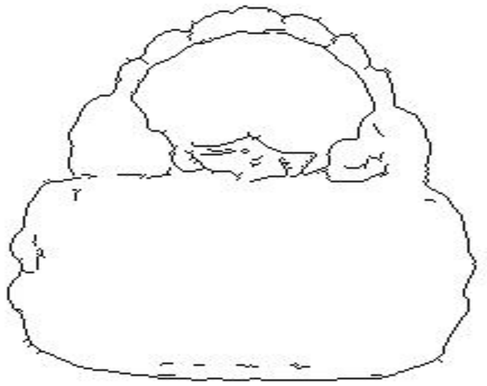


Input



Output

Input

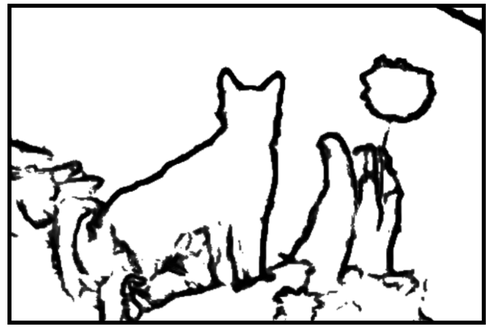


Output

Challenges —> Solutions

1. Output is high-dimensional, structured object

   —> Use a deep net, D, to analyze output!

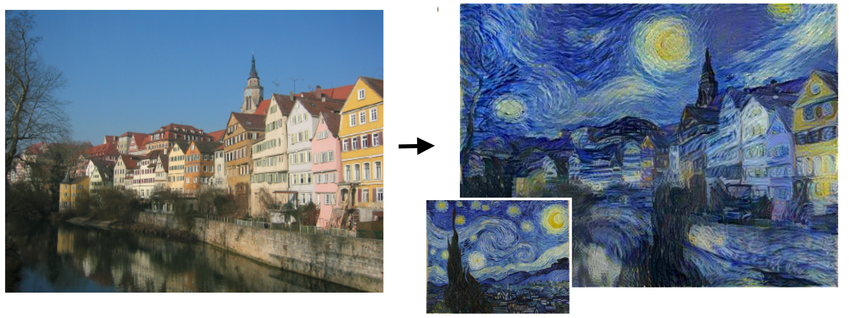


2. Uncertainty in mapping; many plausible outputs

   —> D only cares about “plausibility”, doesn’t hedge

“this small bird has a pink breast and crown…”

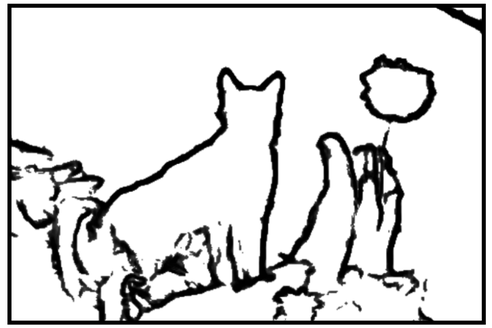
3. Lack of supervised training data



Challenges —> Solutions

1. Output is high-dimensional, structured object

   —> Use a deep net, D, to analyze output!

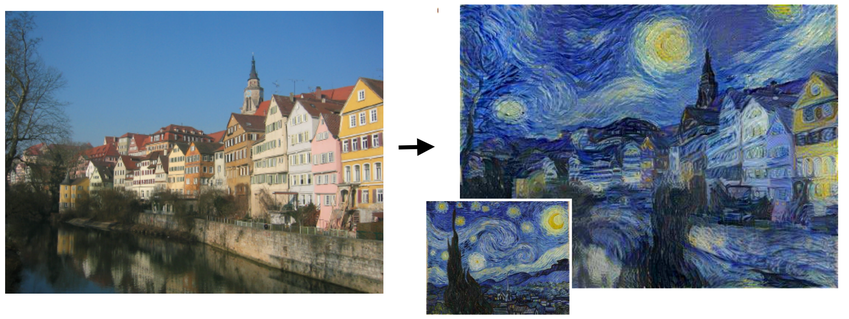


2. Uncertainty in mapping; many plausible outputs

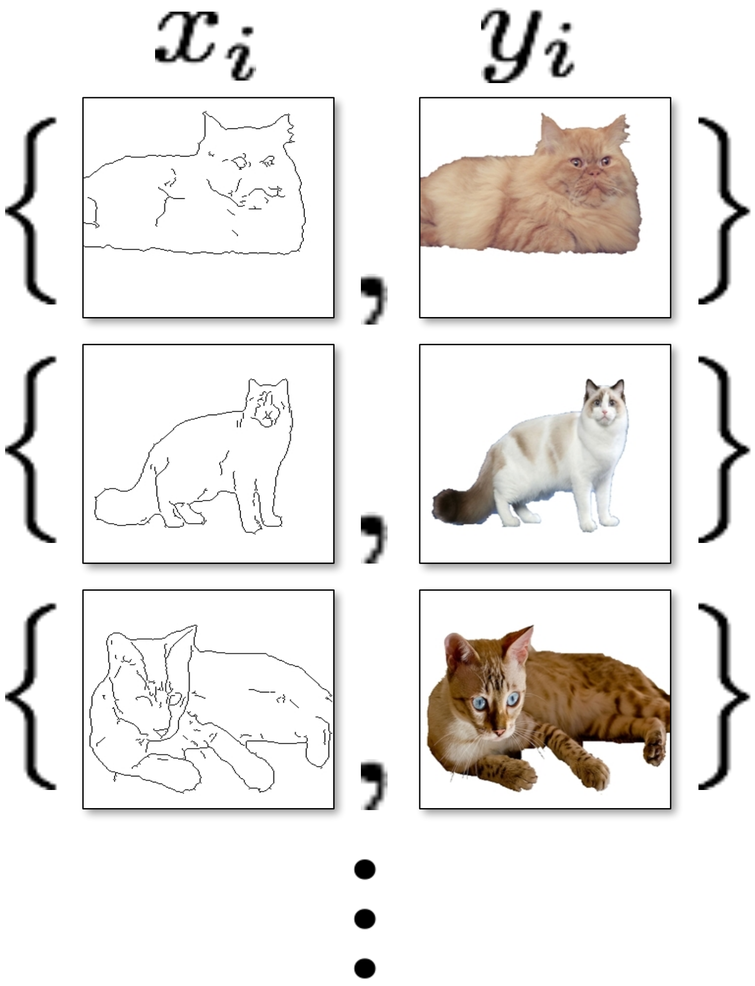
   —> D only cares about "plausibility", doesn't hedge

"this small bird has a pink breast and crown…"

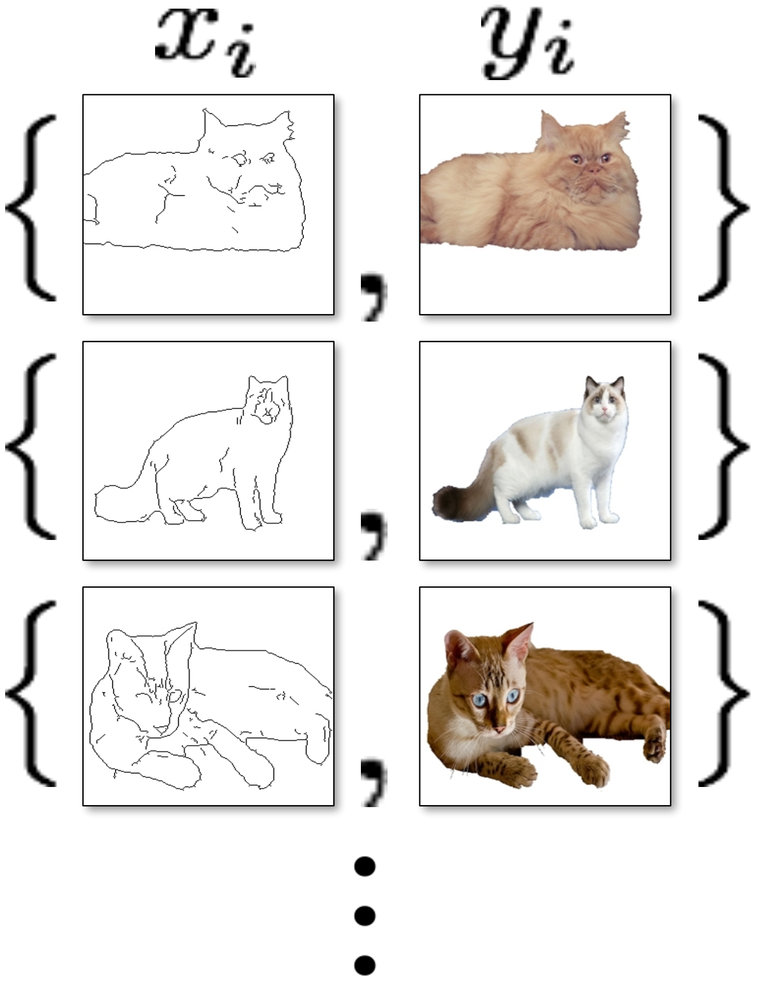
3. **Lack of supervised training data**



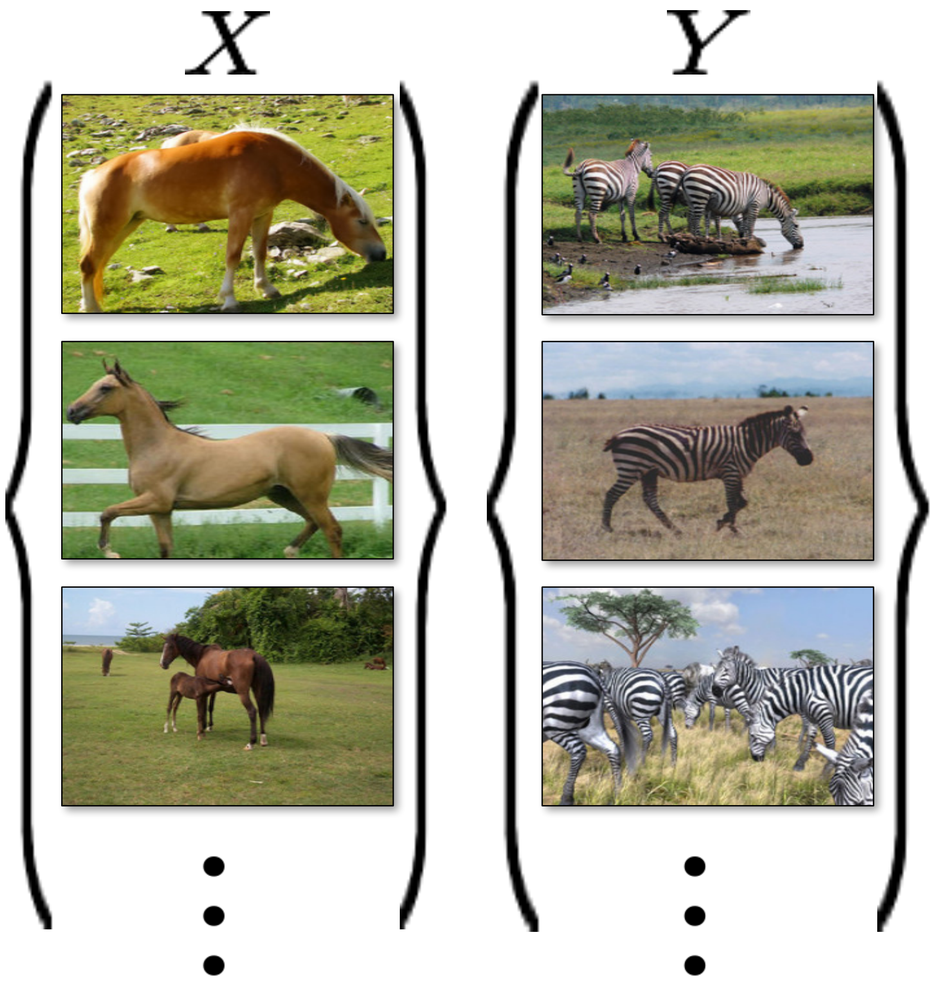
Paired data

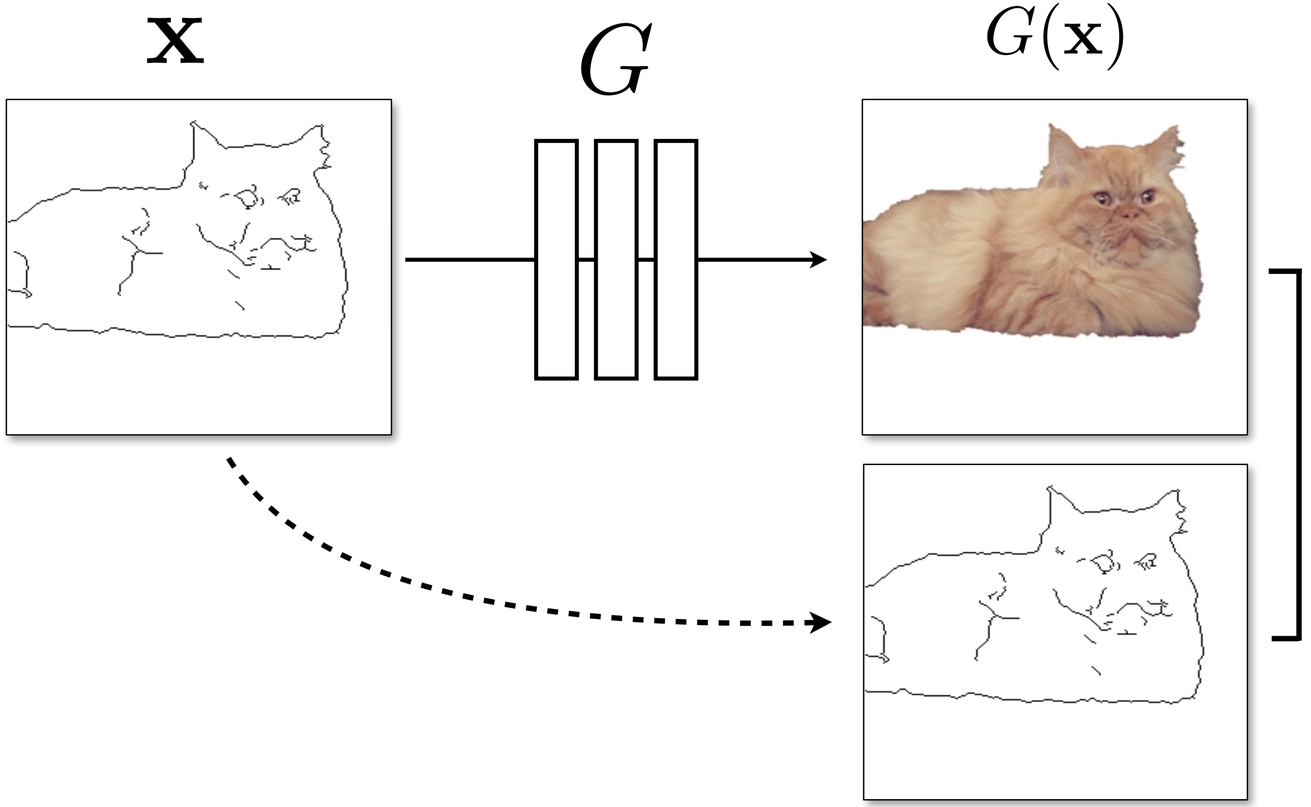


Paired data

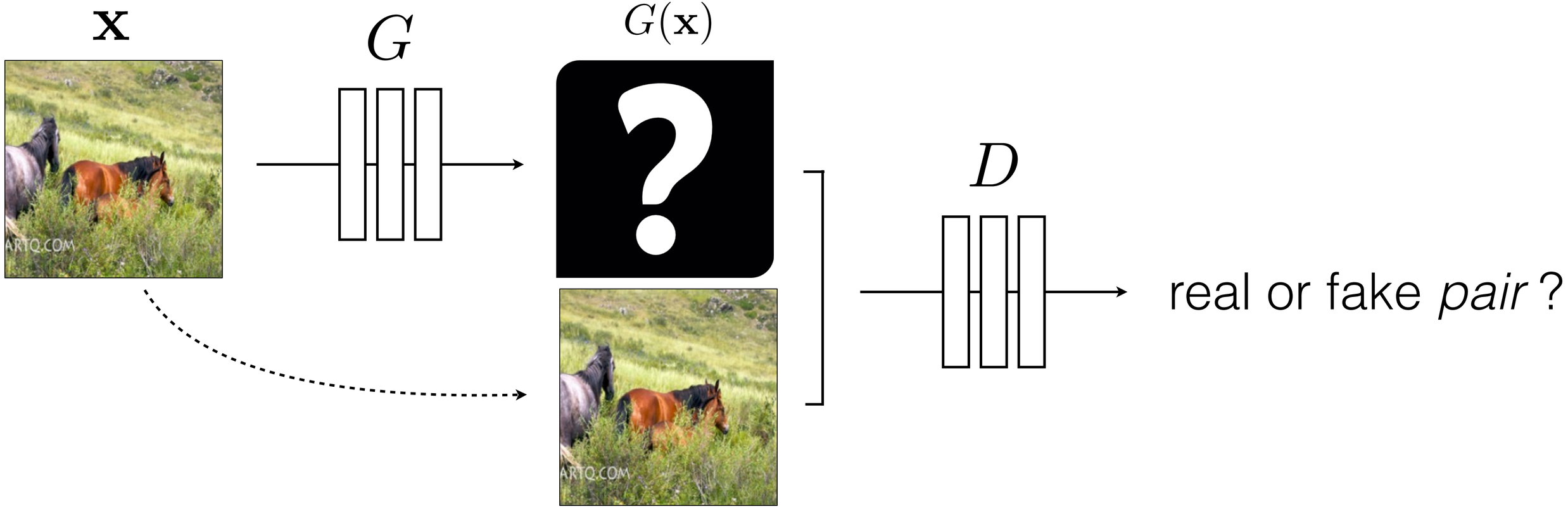


Unpaired data



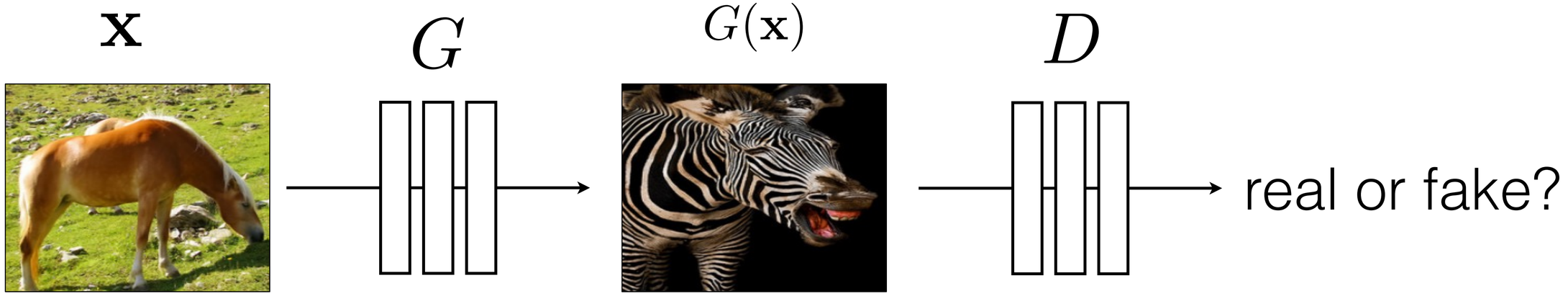


$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;\;]$$



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

No input-output pairs!



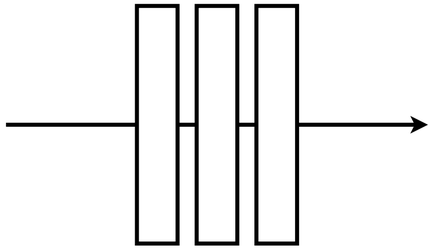
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\;]$$

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

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

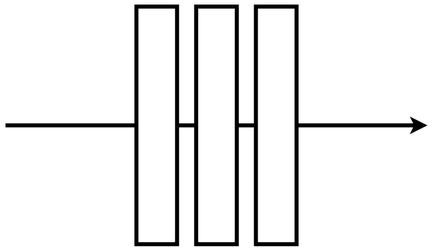
$\mathbf{x}$

G

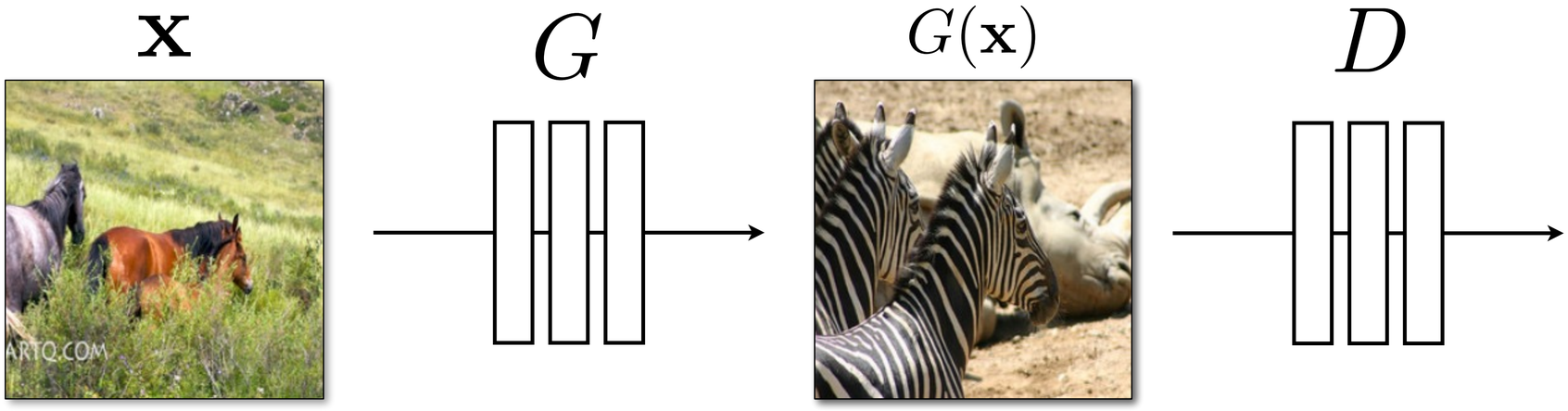


$G(\mathbf{x})$

D

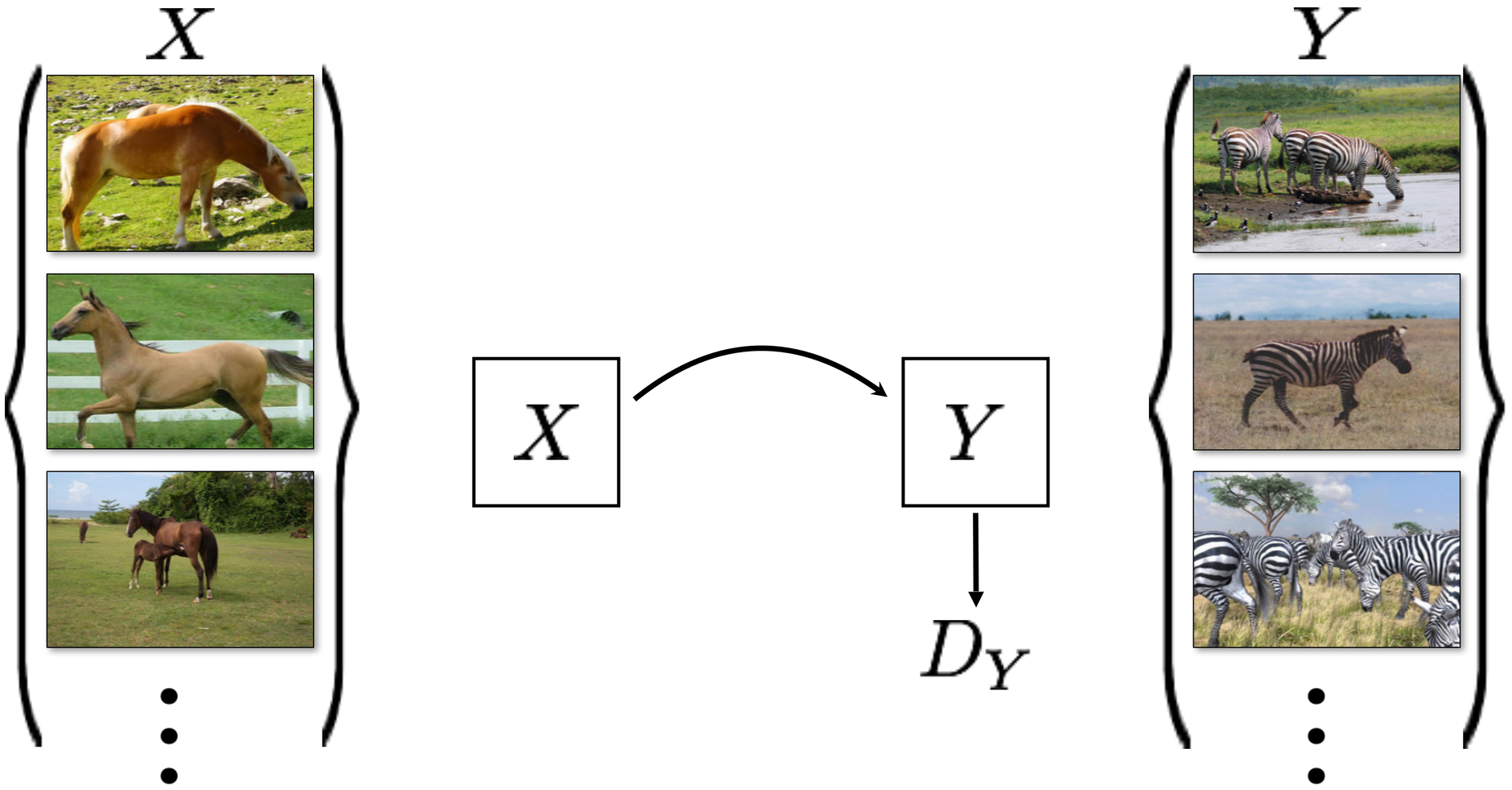


Real!



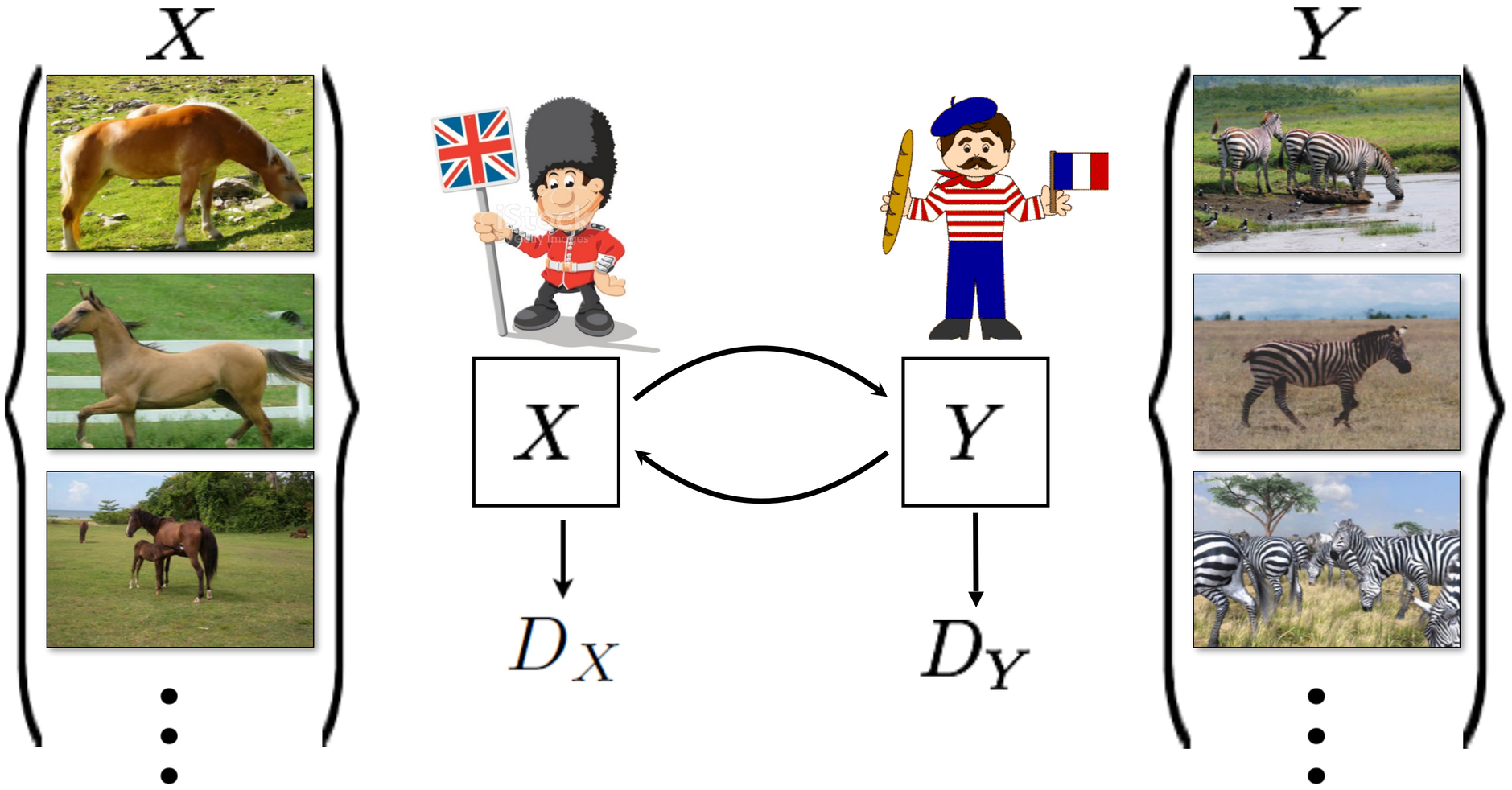
Nothing to force output to correspond to input

Cycle-Consistent Adversarial Networks

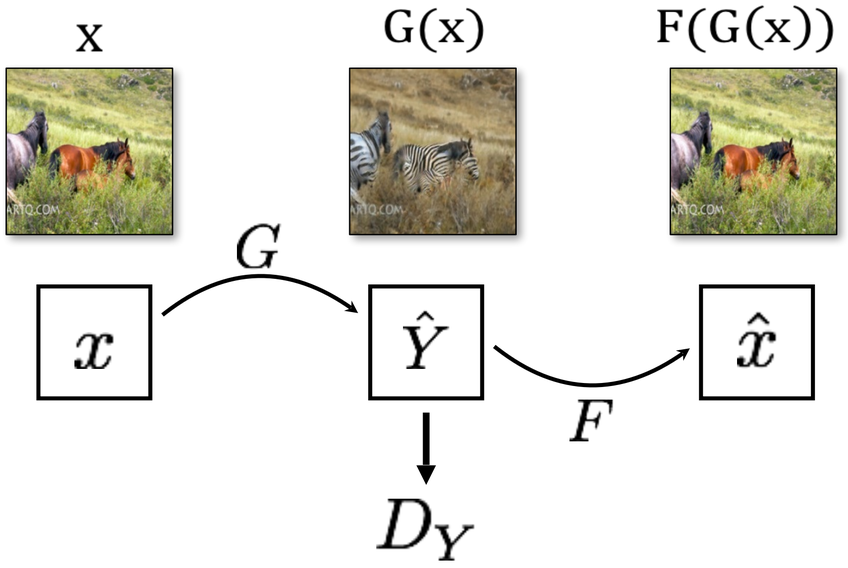


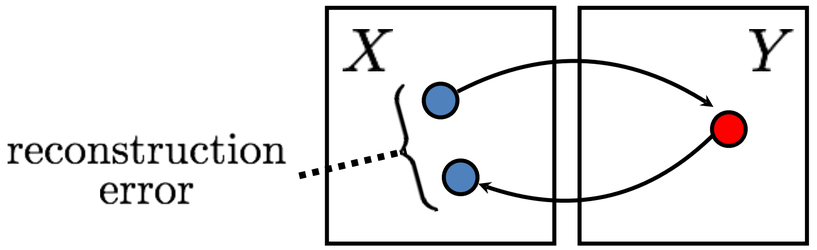
[Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

Cycle-Consistent Adversarial Networks



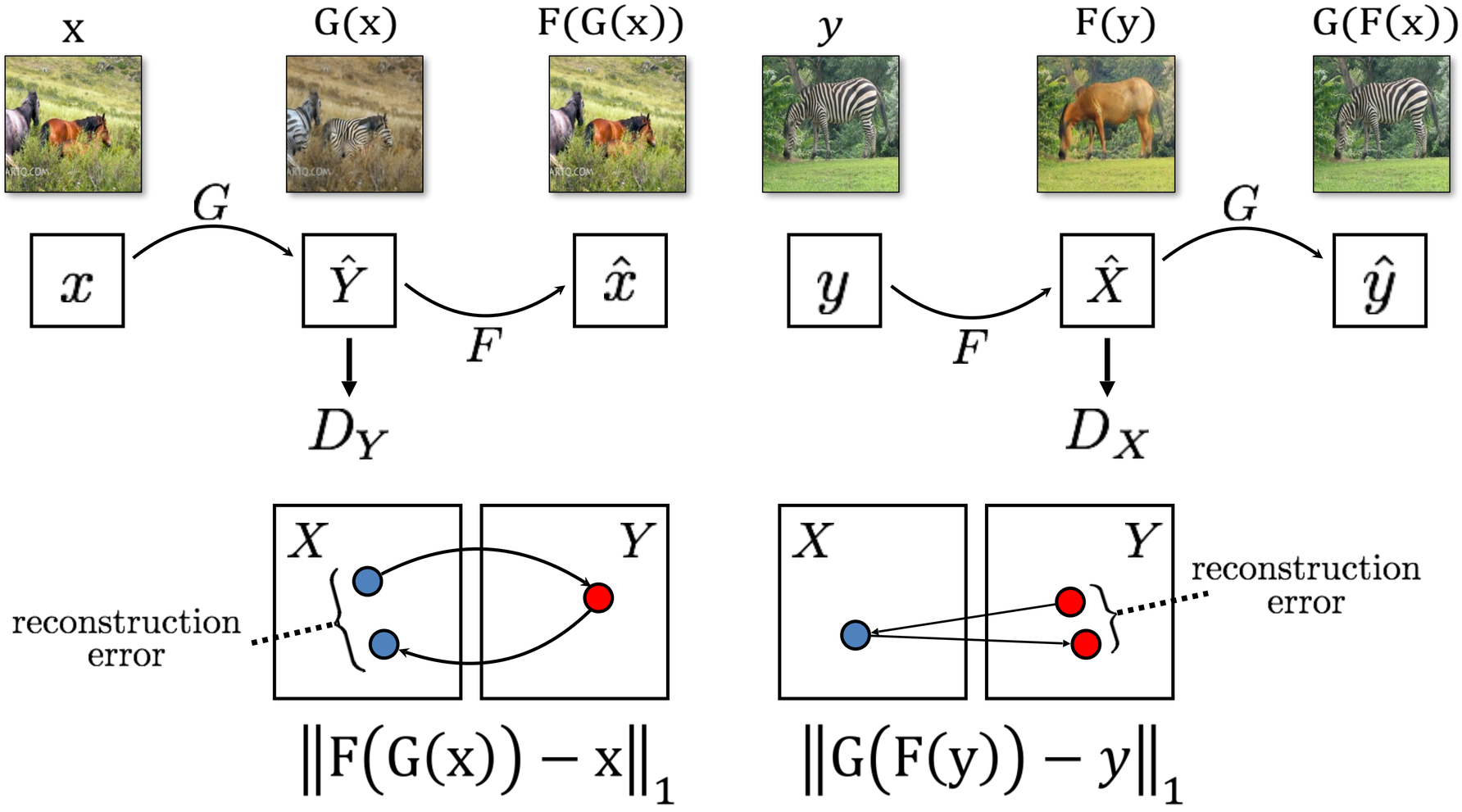
Cycle Consistency Loss

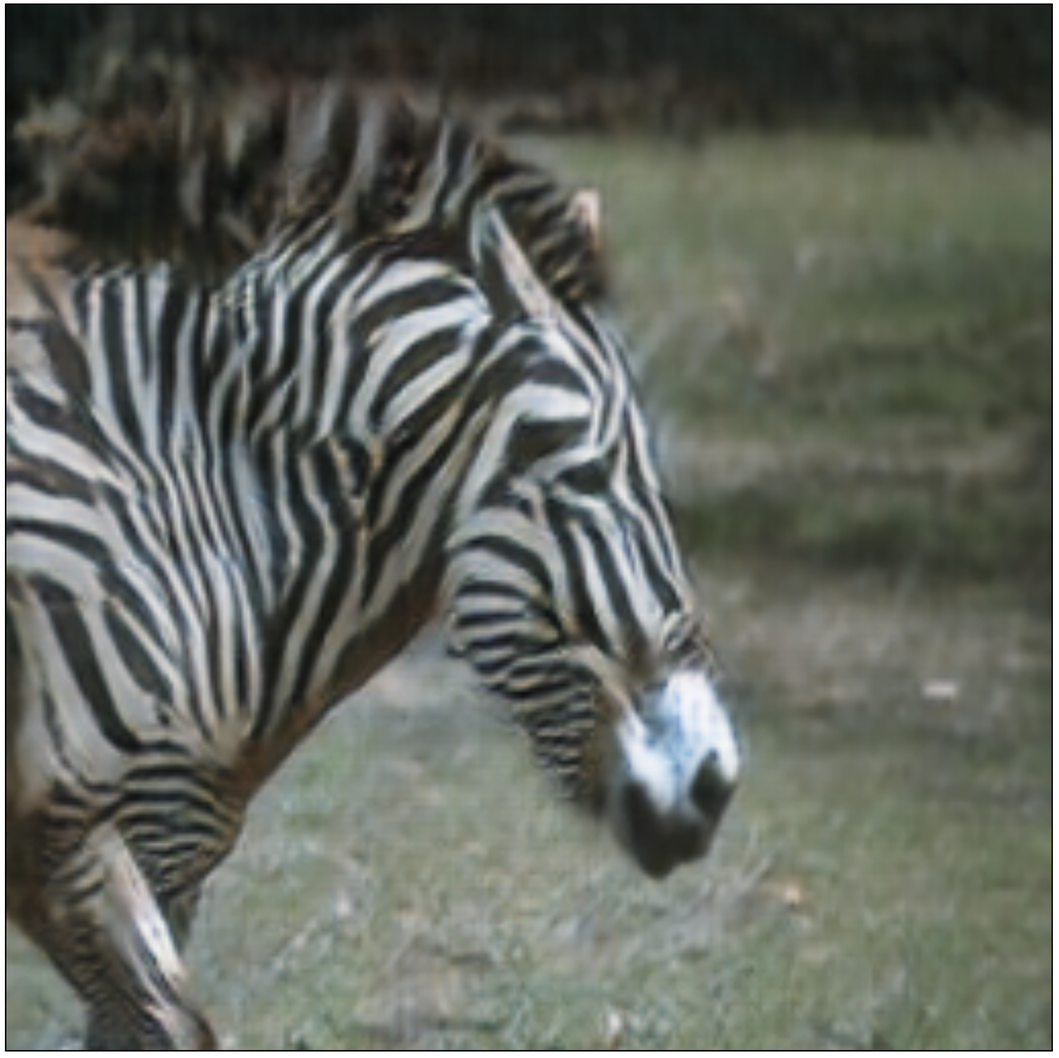




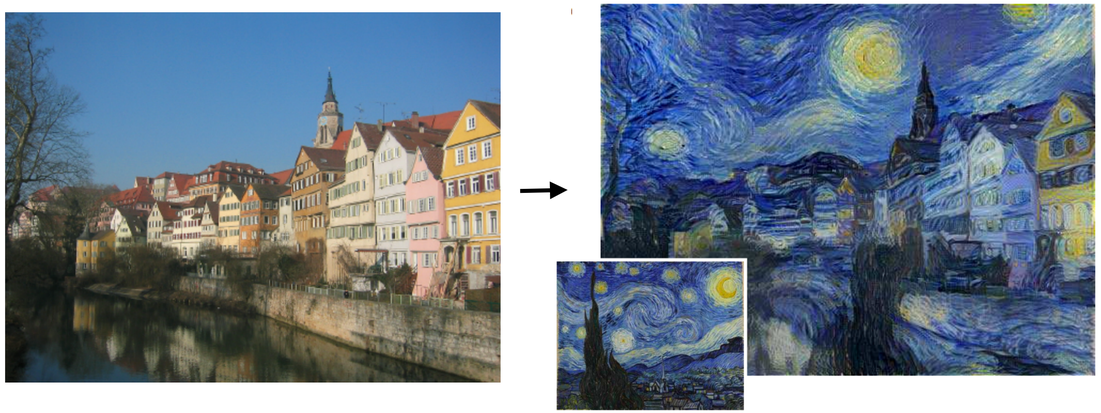
$$\|F(G(x)) - x\|_1$$

Cycle Consistency Loss

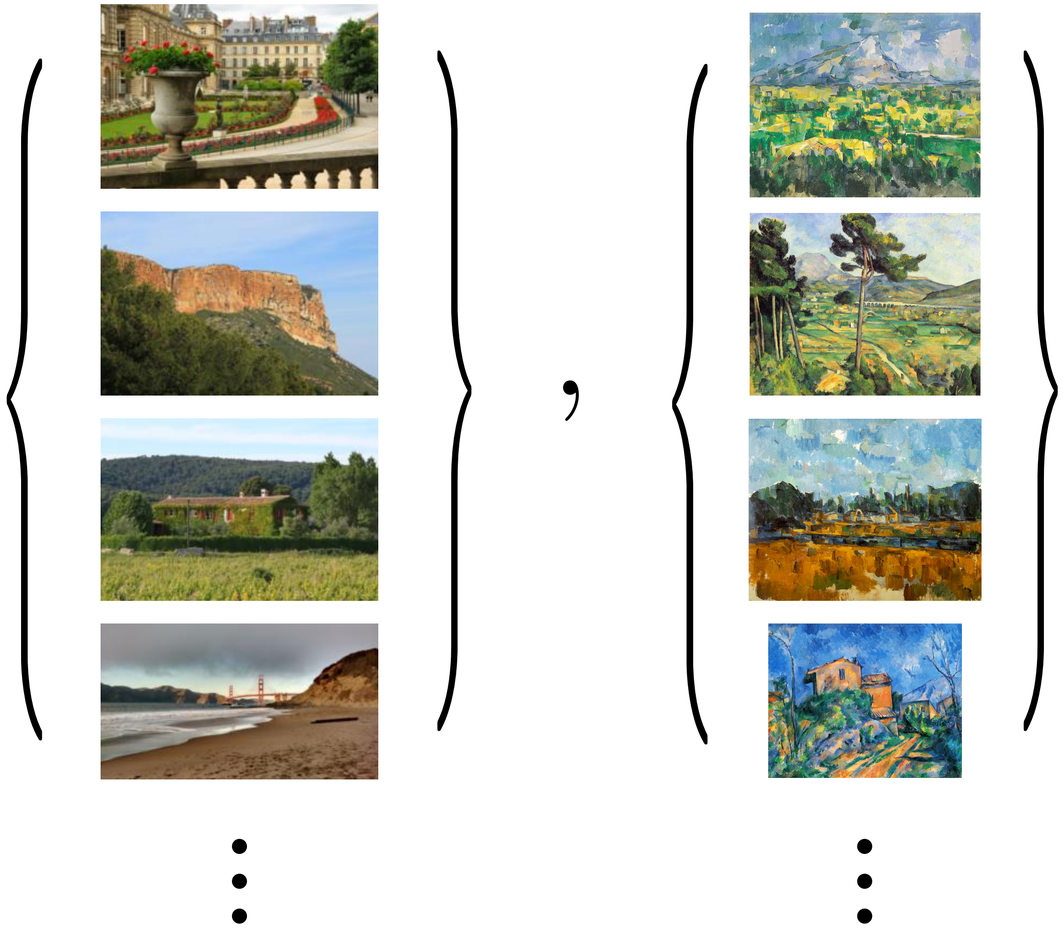




Artistic style transfer



[Gatys et al. 2016, …]



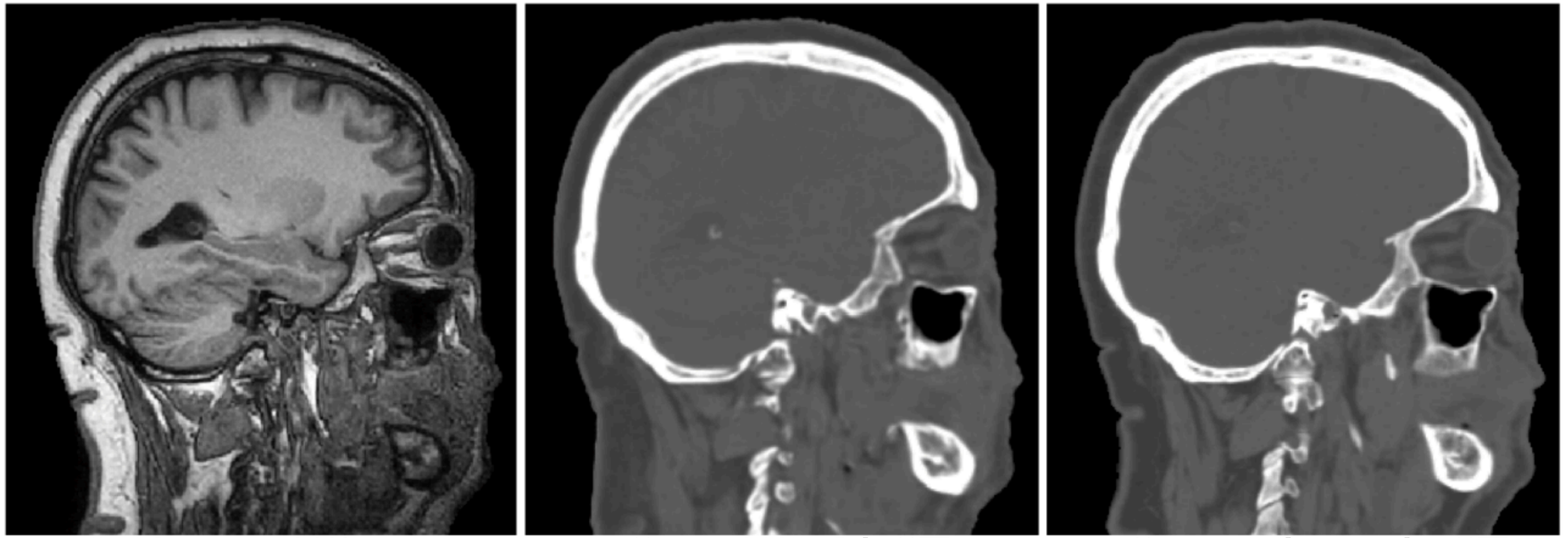

Input
Monet
Van Gogh
Cezanne
Ukiyo-e

Failure case

Failure case

Applications of CycleGAN

MR → CT [Wolterink et al] arxiv: 1708.01155



Input MR | Generated CT | Ground truth CT

- **MRI reconstruction** [Quan et al.] arxiv:1709.00753
- **Cardiac MR images from CT** [Chartsias et al. 2017]

Latest from #CycleGAN

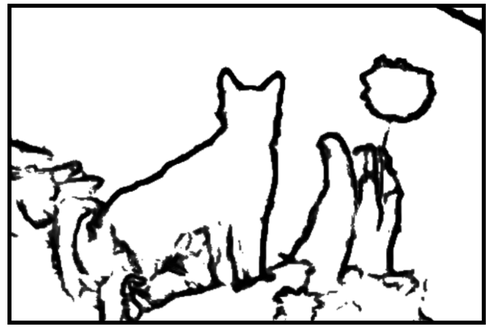
CycleGAN with architectural modifications, by itok_msi

https://qiita.com/itok_msi/items/b6b615bc28b1a720afd7

Challenges —> Solutions

1. Output is high-dimensional, structured object

   —> Use a deep net, D, to analyze output!



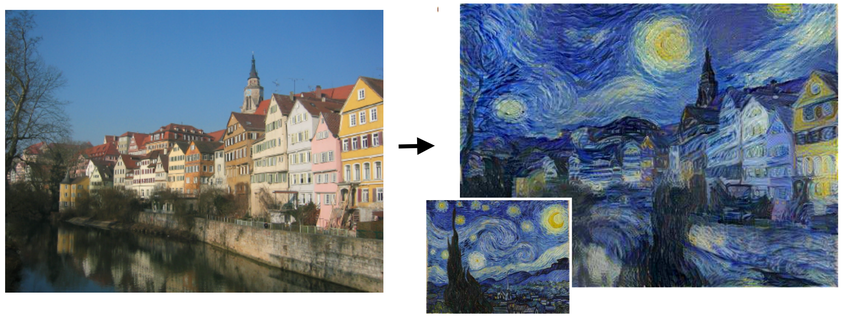
2. Uncertainty in mapping; many plausible outputs

   —> D only cares about "plausibility", doesn't hedge

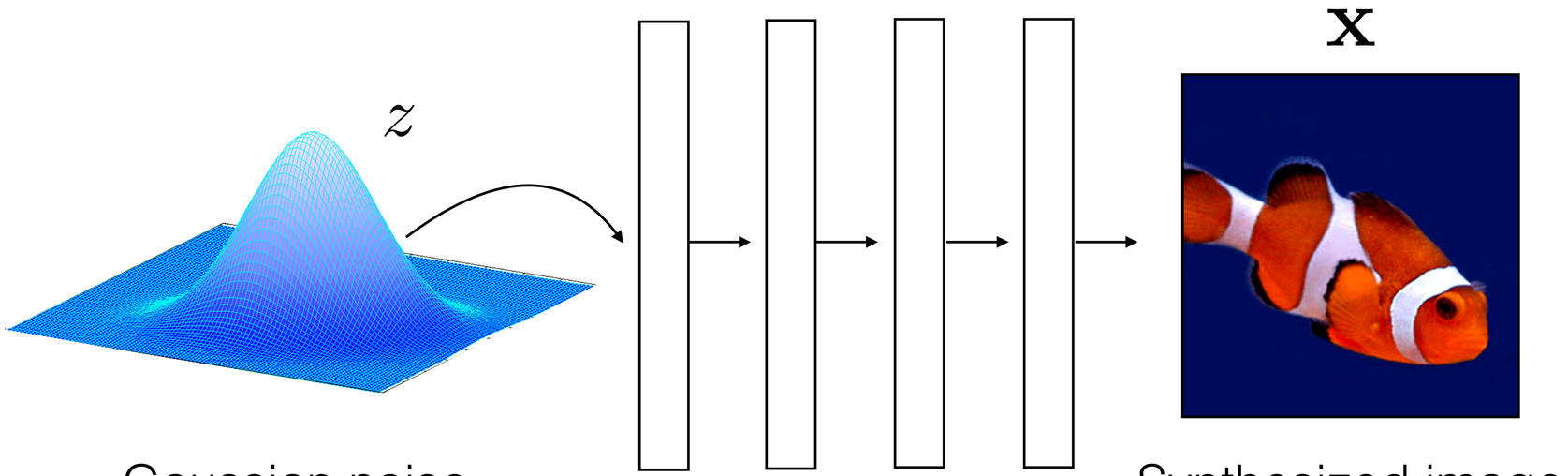
"this small bird has a pink breast and crown…"

3. Lack of supervised training data

   —> D doesn't require paired training instances



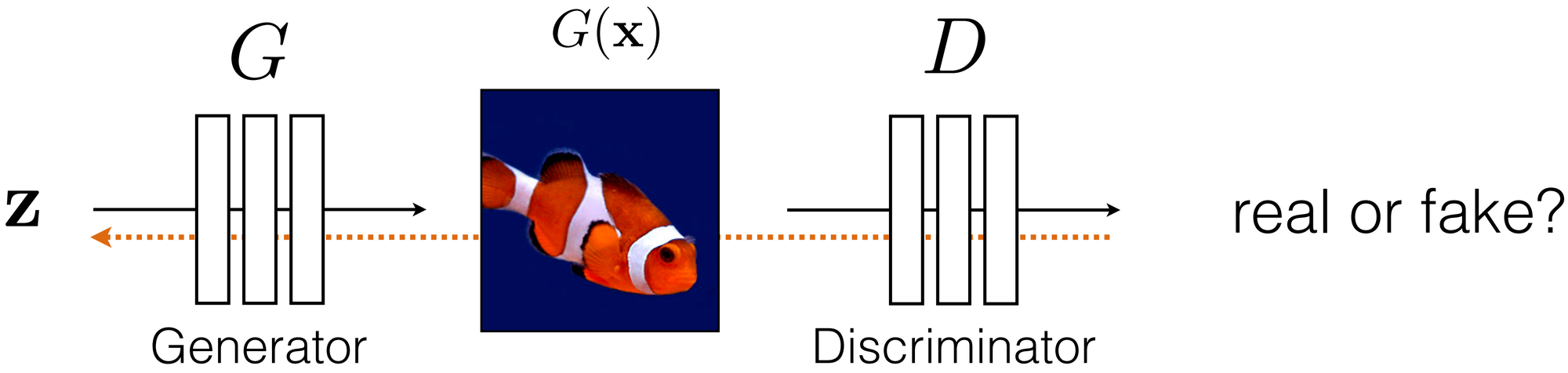
Can we *generate* images from scratch?



Gaussian noise

Synthesized image

$$z \sim \mathcal{N}(\vec{0}, 1)$$



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

D tries to identify the fakes

[Goodfellow et al., 2014]

GANs are implicit generative models

$p(\mathbf{x})$ ⟵ "generative model" of the data $\mathbf{x}$

Noise distribution

$\mathbf{z} \sim \mathcal{N}(0, 1)$ ⟶ GAN ⟶ $\mathbf{x} \sim p(\mathbf{x})$

Data distribution

$G(\mathbf{z}) \sim p(\mathbf{x})$ ⟵ Samples from GAN, at equilibrium, are samples from the data distribution

Randomly generated faces

[BEGAN: Berthelot et al., 2014]

Interpolation in z space

Input A — Interpolation from A to B — Input B

Code online: https://github.com/phillipi

pix2pix

Image-to-image translation with conditional adversarial nets

Lua ★ 3.5k 481

junyanz/**CycleGAN**

Software that can generate photos from paintings, turn horses into zebras, perform style transfer, and more.

Lua ★ 4.3k 468

junyanz/**pytorch-CycleGAN-and-pix2pix**

Image-to-image translation in PyTorch (e.g. horse2zebra, edges2cats, and more)

Python ★ 1.9k 253

Pix2pix: 144 lines
CycleGAN: 220 lines