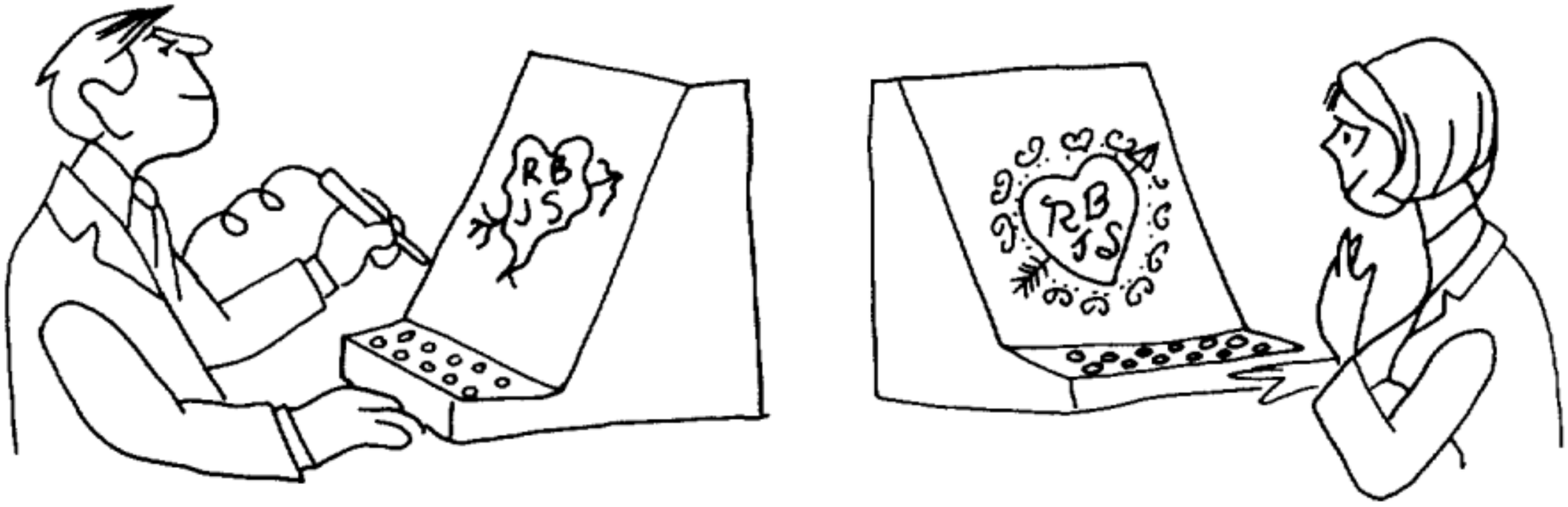
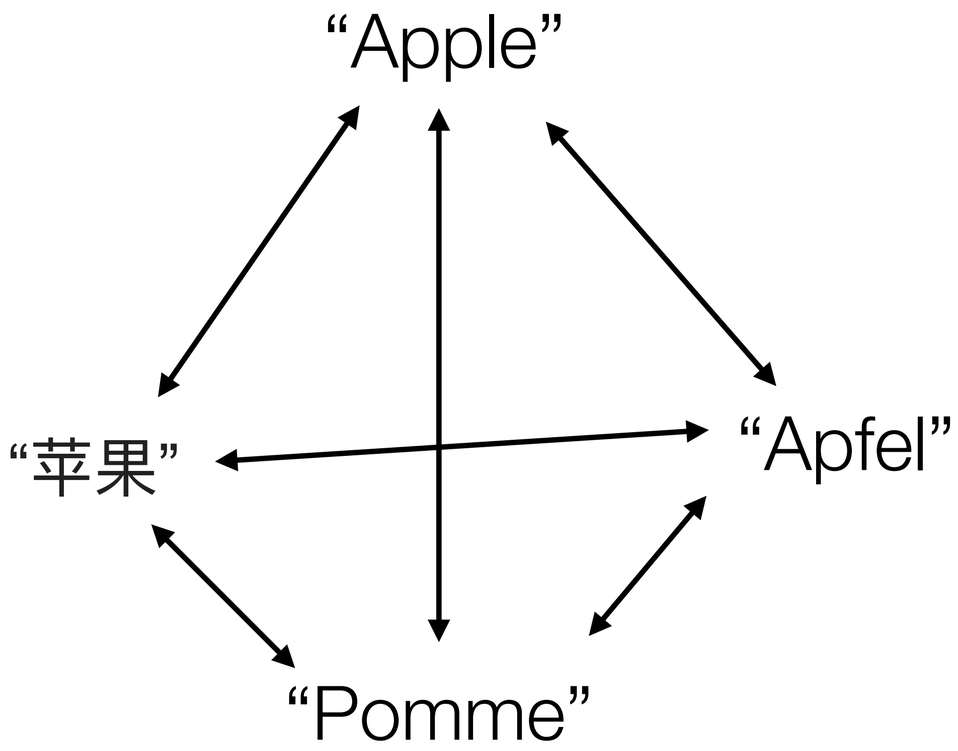


Paired and unpaired image translation with GANs

Phillip Isola, MIT
6/16/19

[Cartoon: The Computer as a Communication Device, Licklider & Taylor 1968]

Image-to-Image Translation



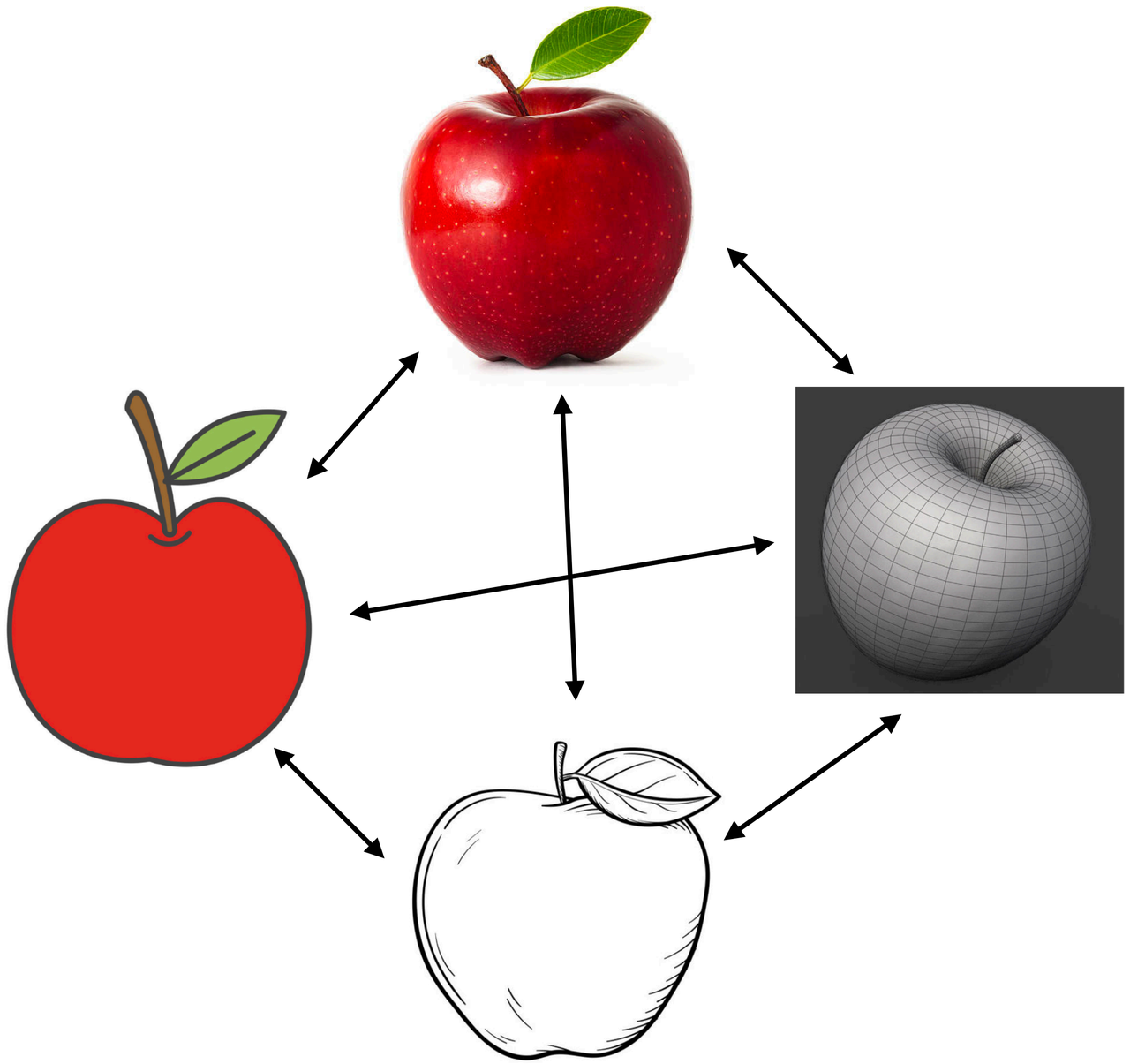
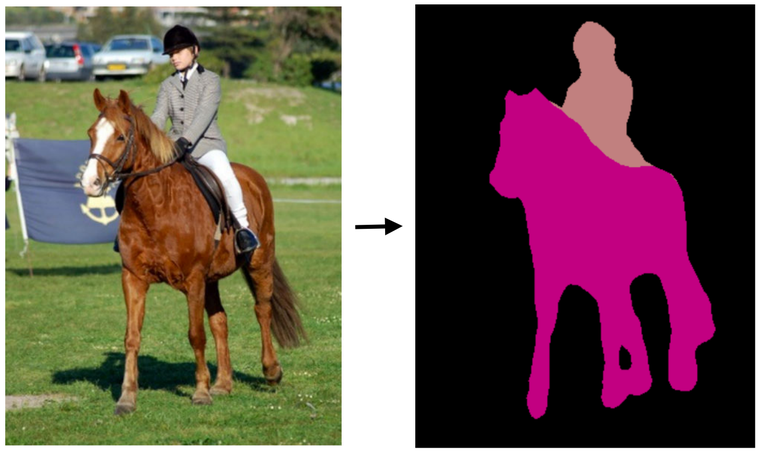


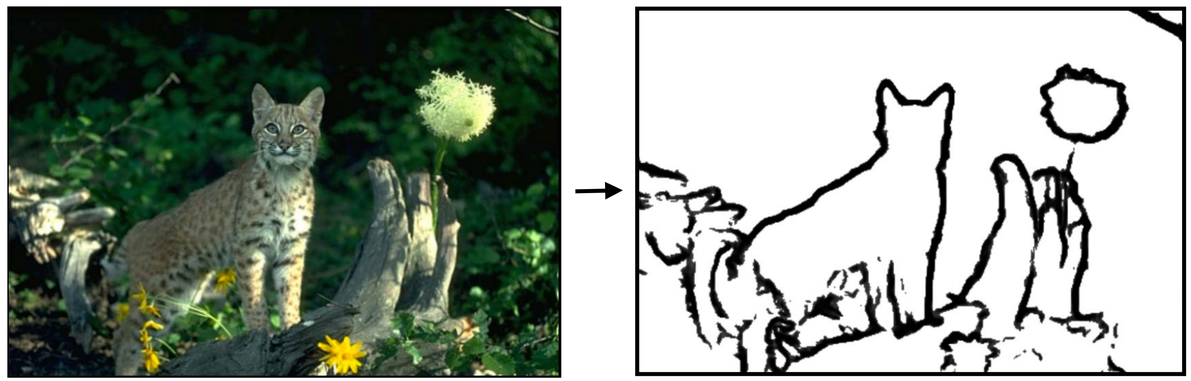
Image-to-Image Translation

Object labeling



[Long et al. 2015]

Edge Detection

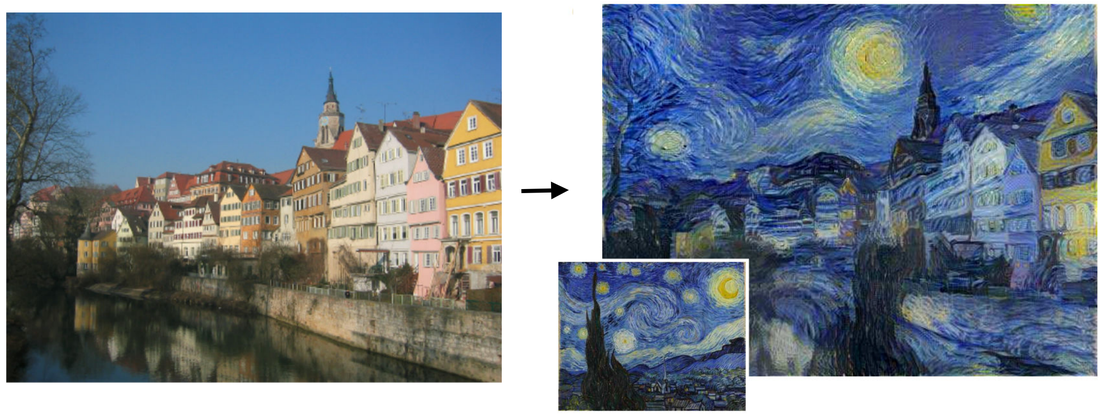


[Xie et al. 2015]

Season change

[Laffont et al. 2014]

Artistic style transfer



[Gatys et al. 2016]

Paired Image-to-Image Translation

Input $\mathbf{x}$

Output $\mathbf{y}$

$\mathcal{F}$

Training data

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

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

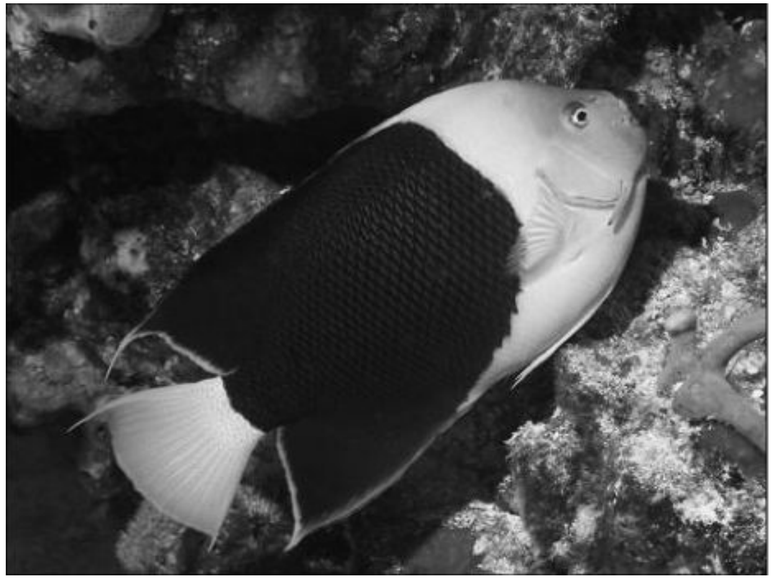
Objective function (loss)

Neural Network

[Zhang, Isola, Efros, ECCV 2016]

Paired Image-to-Image Translation

Input $\mathbf{x}$



$\mathcal{F}$

Output $\mathbf{y}$

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

"**What** should I do"

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

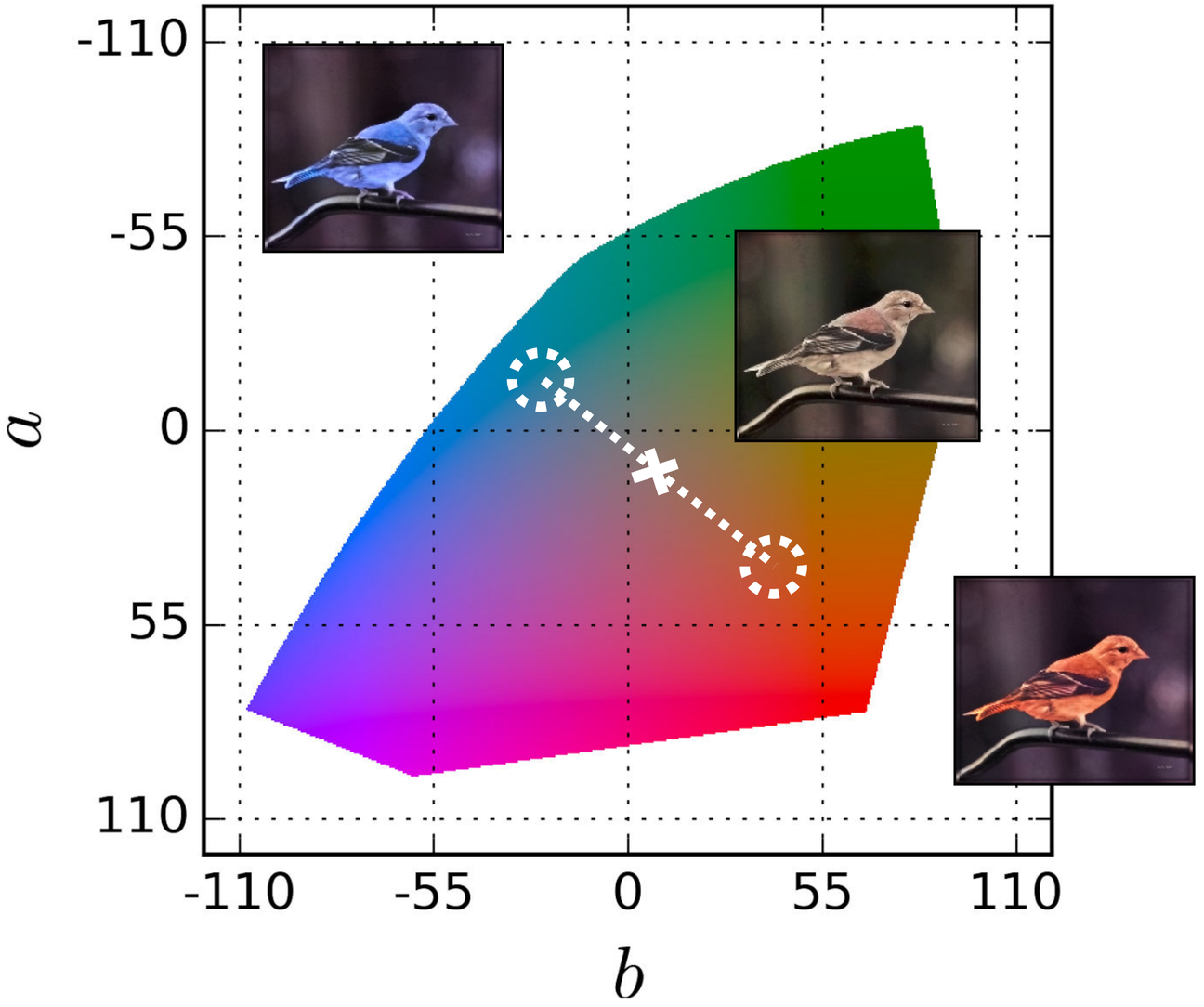
Designing loss functions

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]

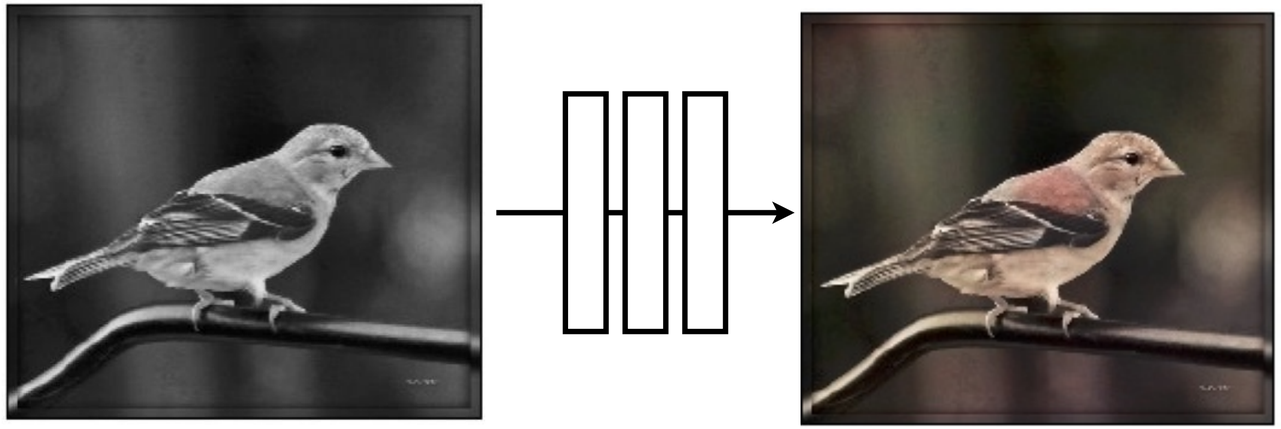

DADS

Designing loss functions

Be careful what you wish for!

Designing loss functions

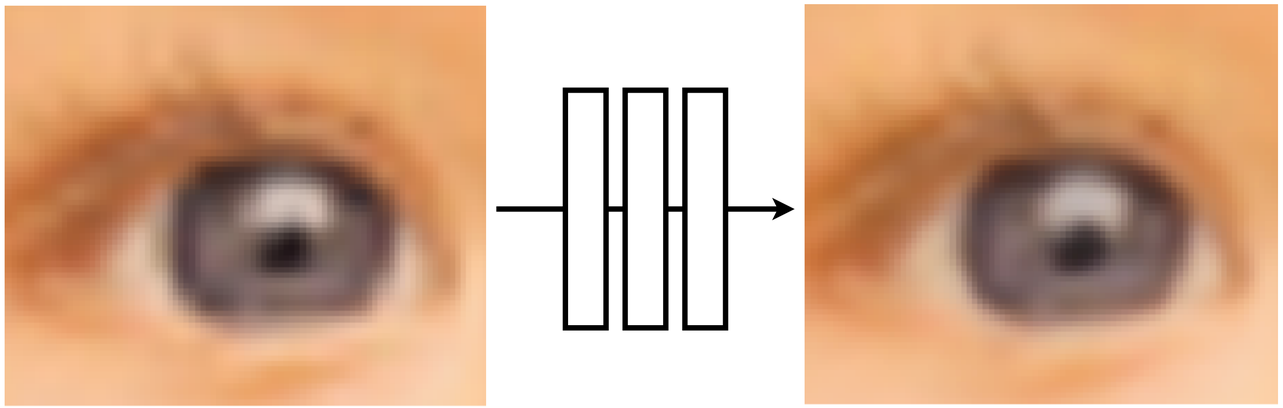
Image colorization



[Zhang, Isola, Efros, ECCV 2016]

L2 regression

Super-resolution

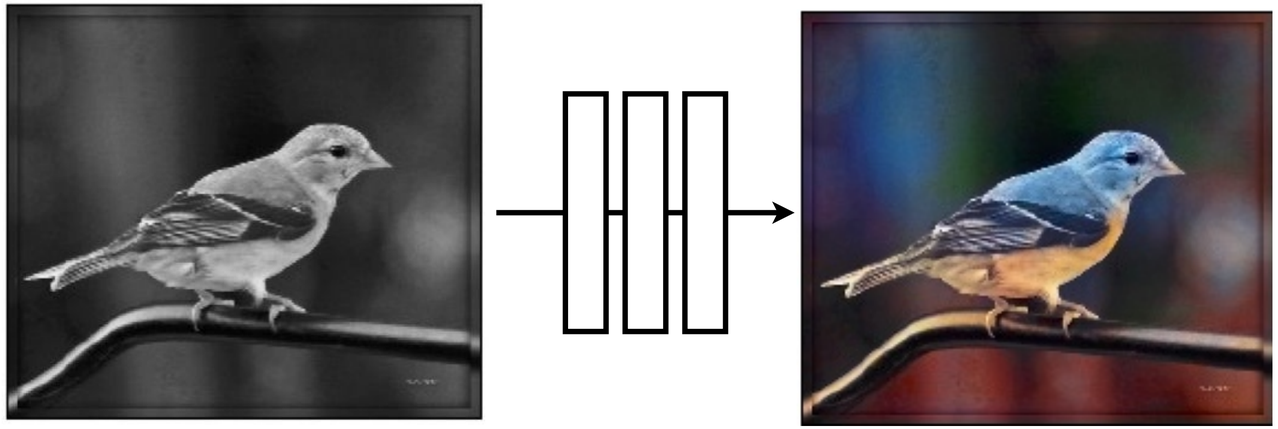


[Johnson, Alahi, Li, ECCV 2016]

L2 regression

Designing loss functions

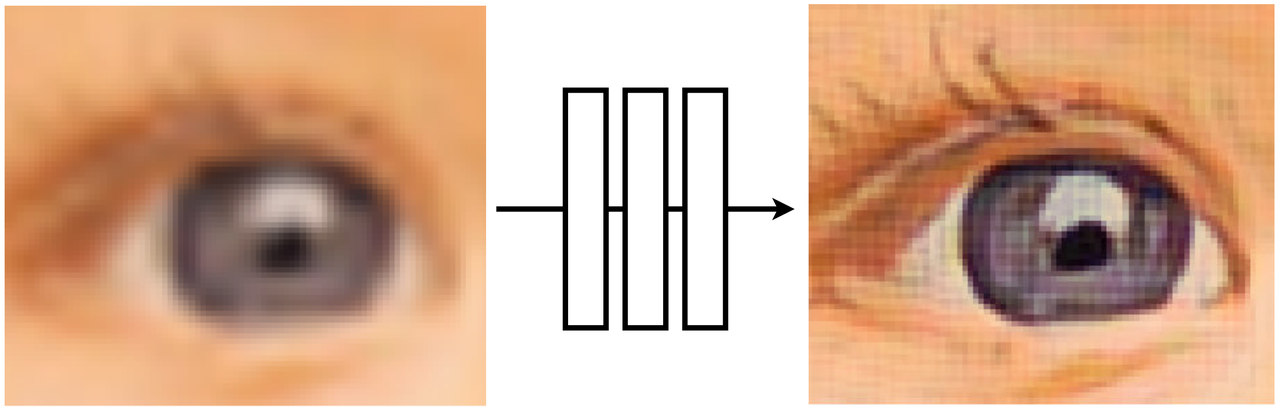
Image colorization



[Zhang, Isola, Efros, ECCV 2016]

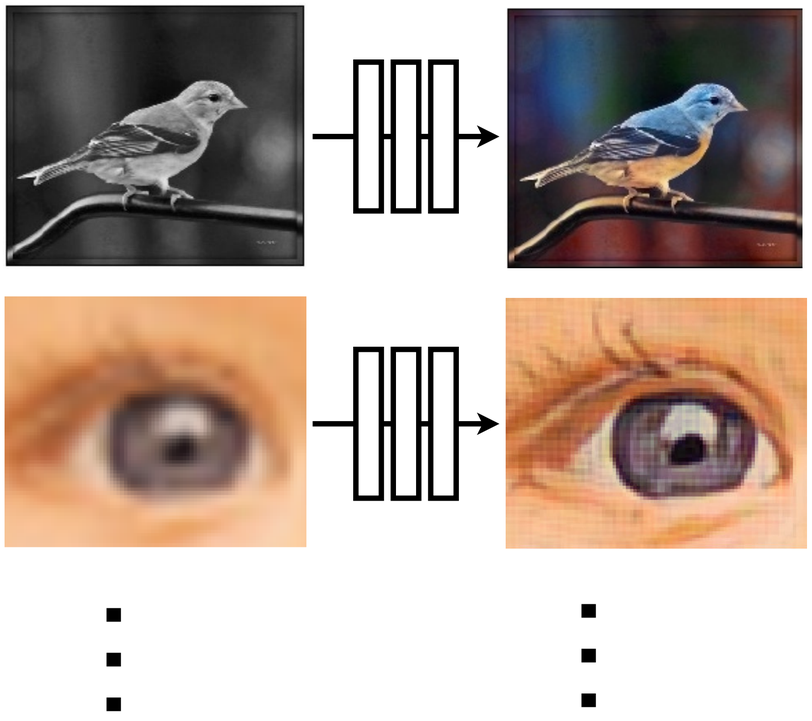
Cross entropy objective,
with colorfulness term

Super-resolution



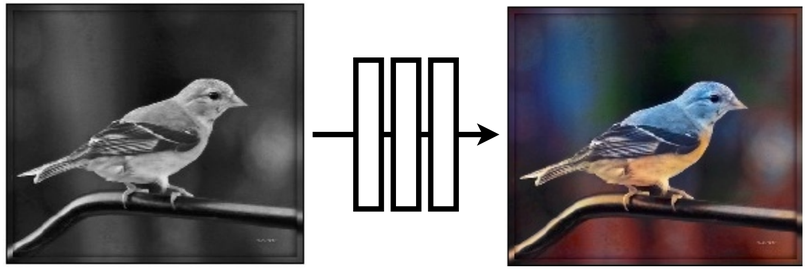
[Johnson, Alahi, Li, ECCV 2016]

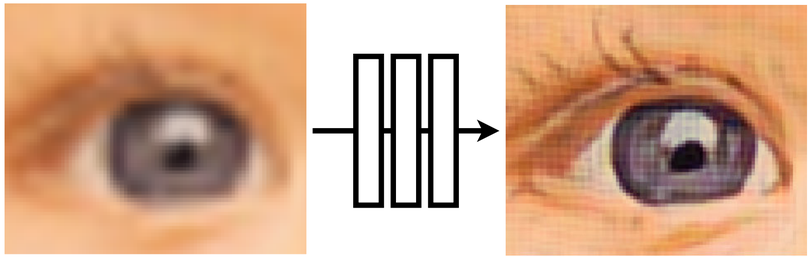
Deep feature covariance
matching objective



Universal loss?

Generated images





“Generative Adversarial Network” (GANs)

Generated vs Real (classifier)

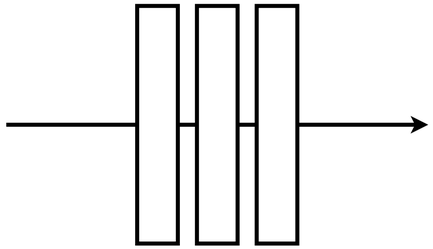
Real photos

…

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

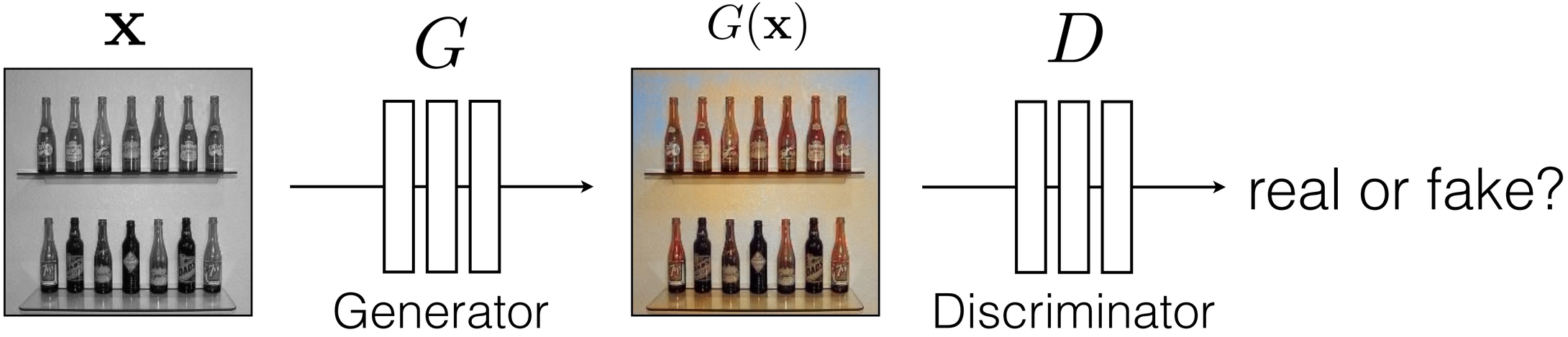
$\mathbf{x}$

G



Generator

$G(\mathbf{x})$



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

D tries to identify the fakes

$\mathbf{x}$

G

$G(\mathbf{x})$

D

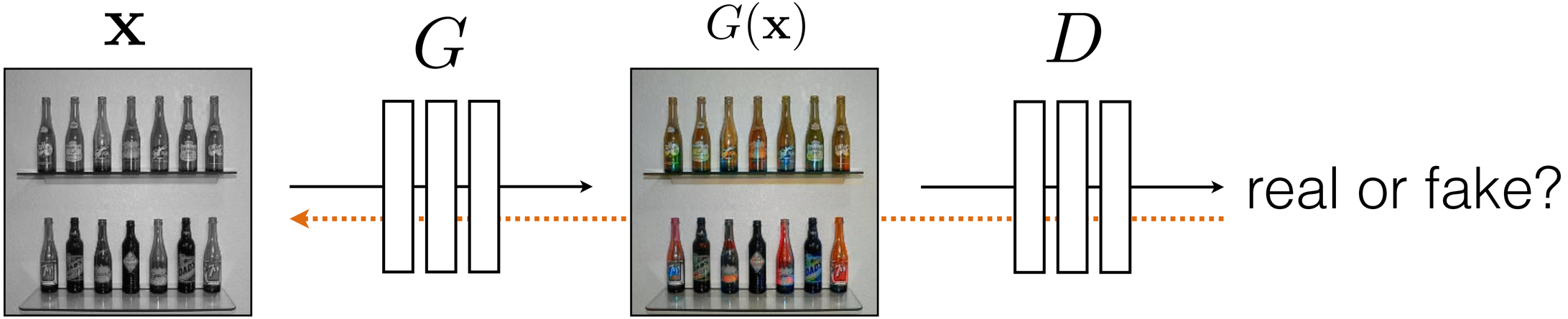
fake (0.9)

$\mathbf{y}$

D

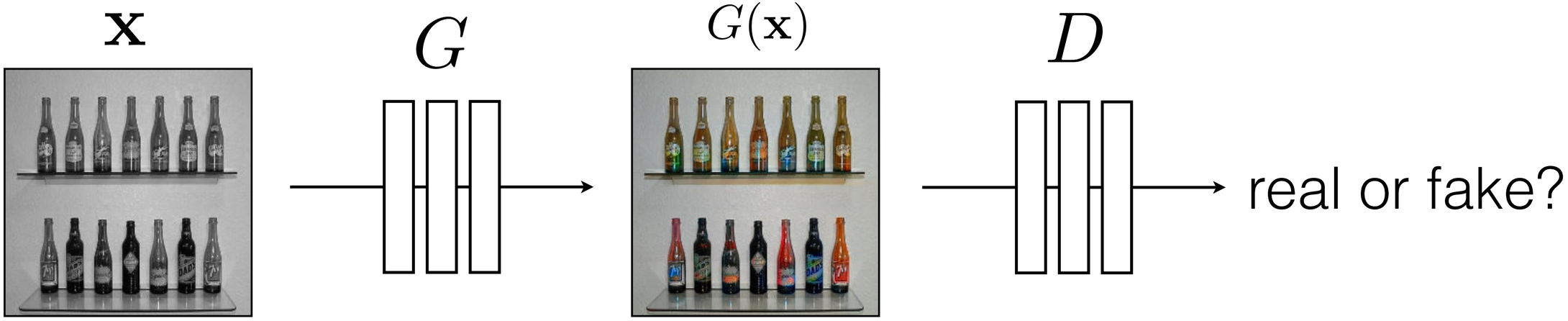
real (0.1)

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



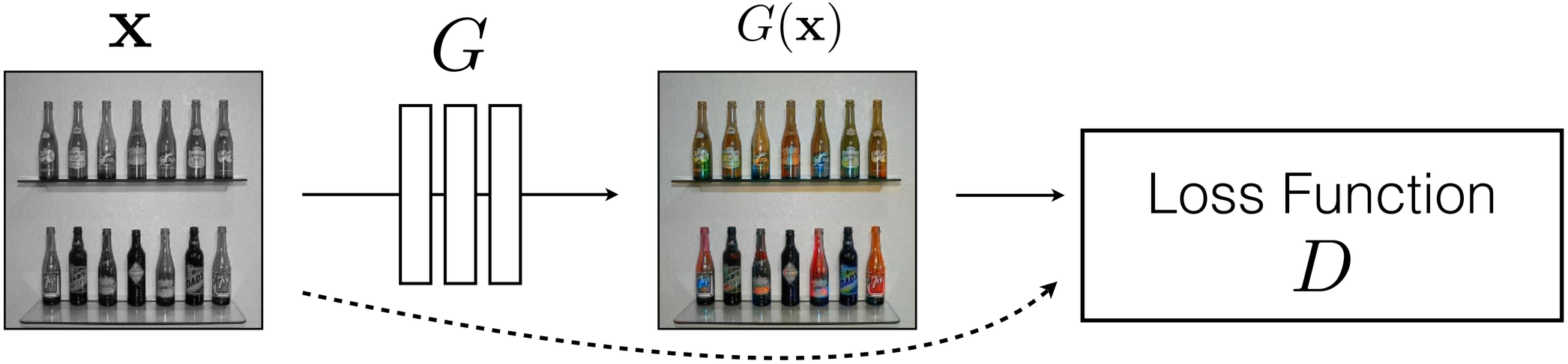
G tries to synthesize fake images that ***fool*** **D**:

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



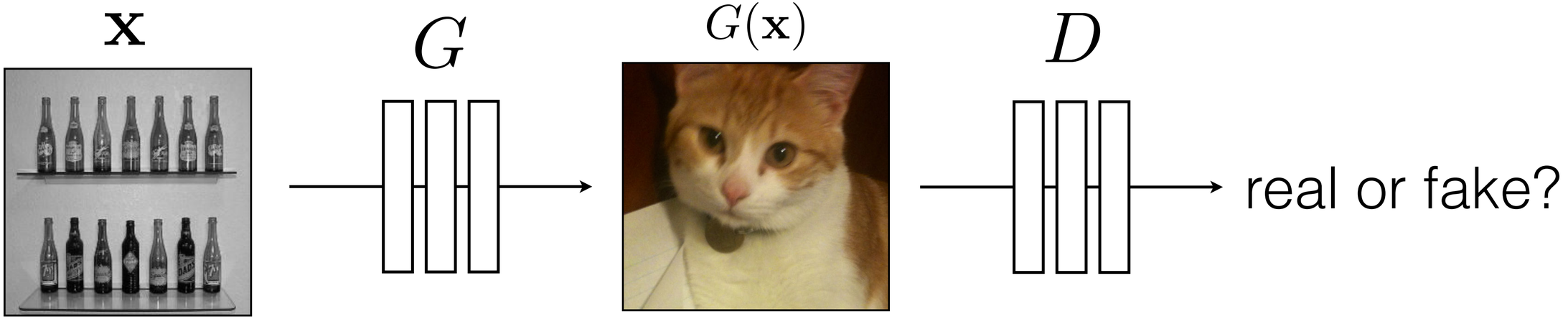
G tries to synthesize fake images that ***fool*** the ***best*** **D**:

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

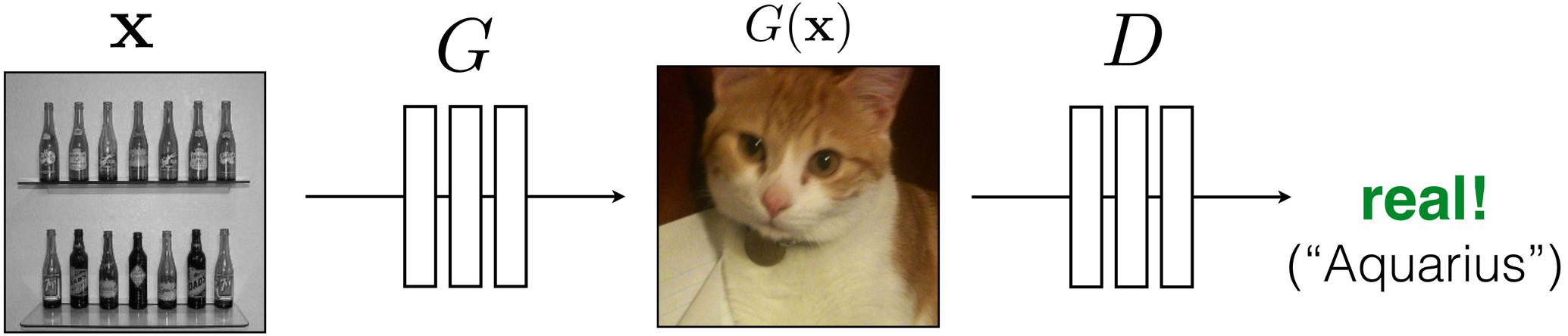


G's perspective: **D** is a loss function.

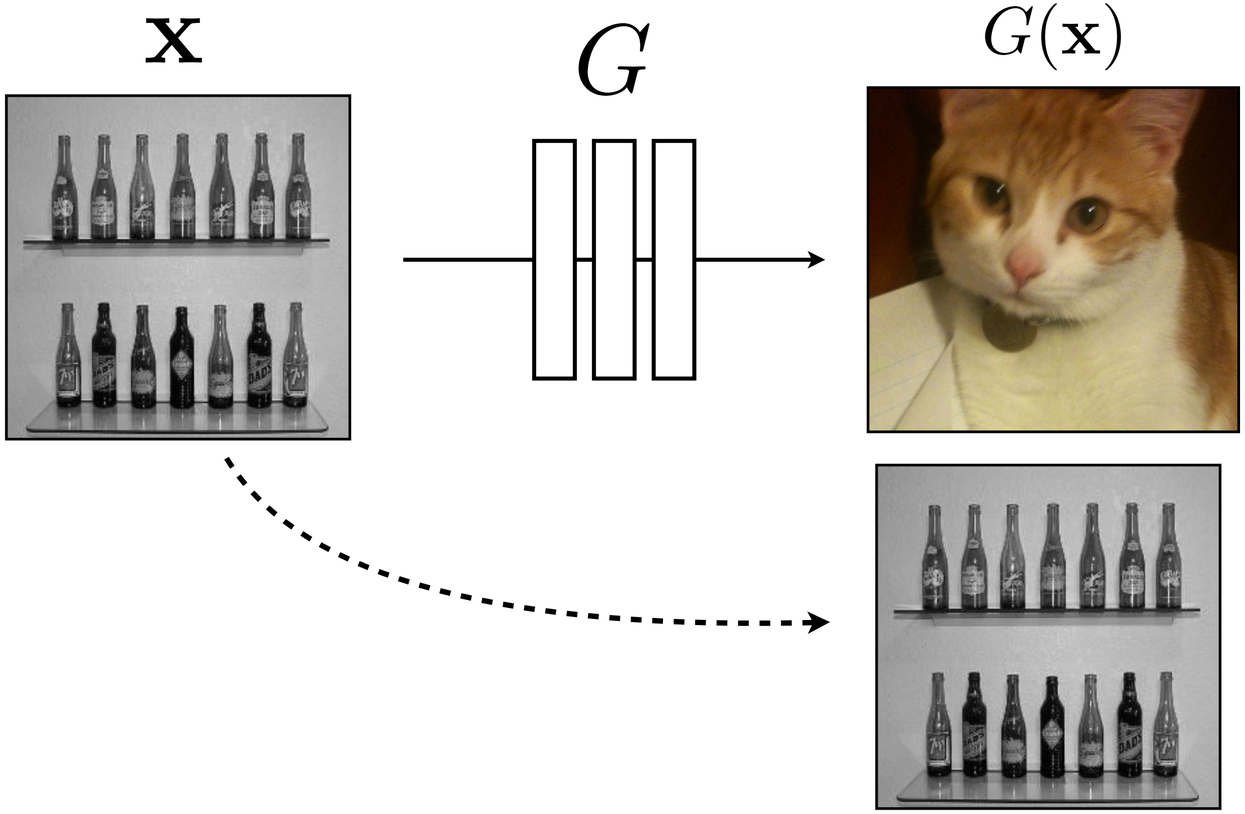
Rather than being hand-designed, it is *learned*.



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



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



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

$\mathbf{x}$

G

$G(\mathbf{x})$

D

real or fake *pair* ?

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

$\mathbf{x}$

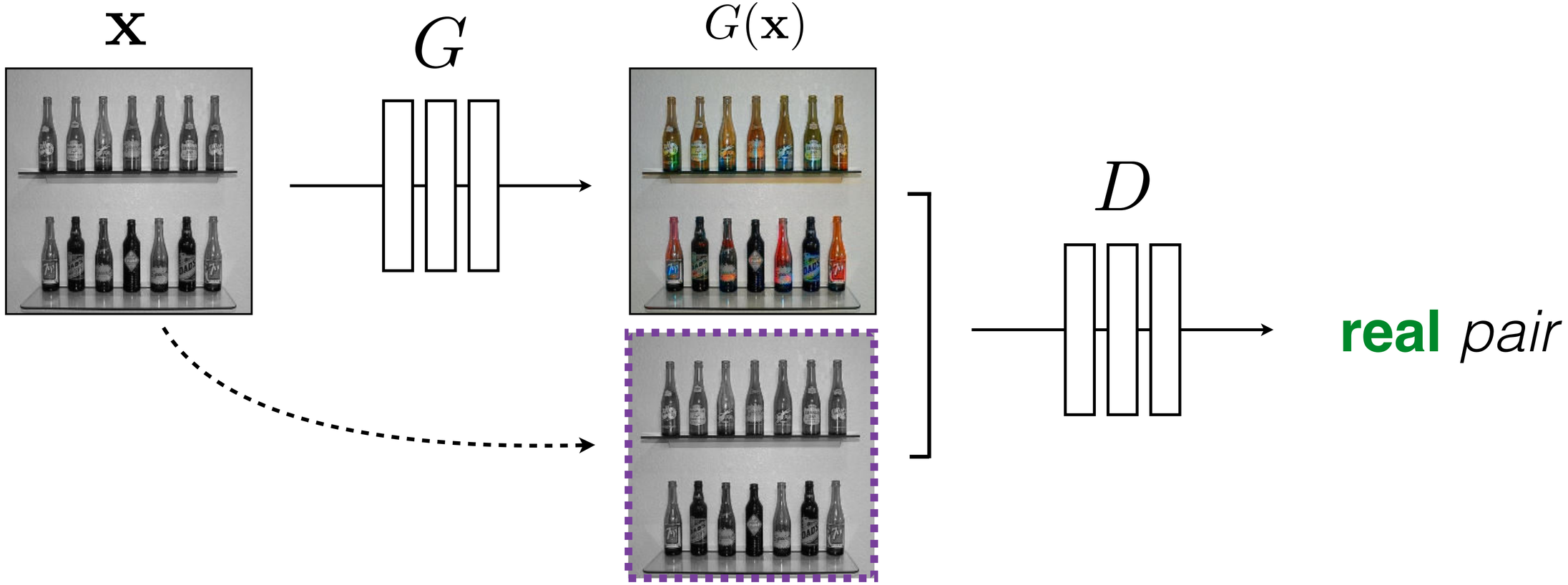
G

$G(\mathbf{x})$

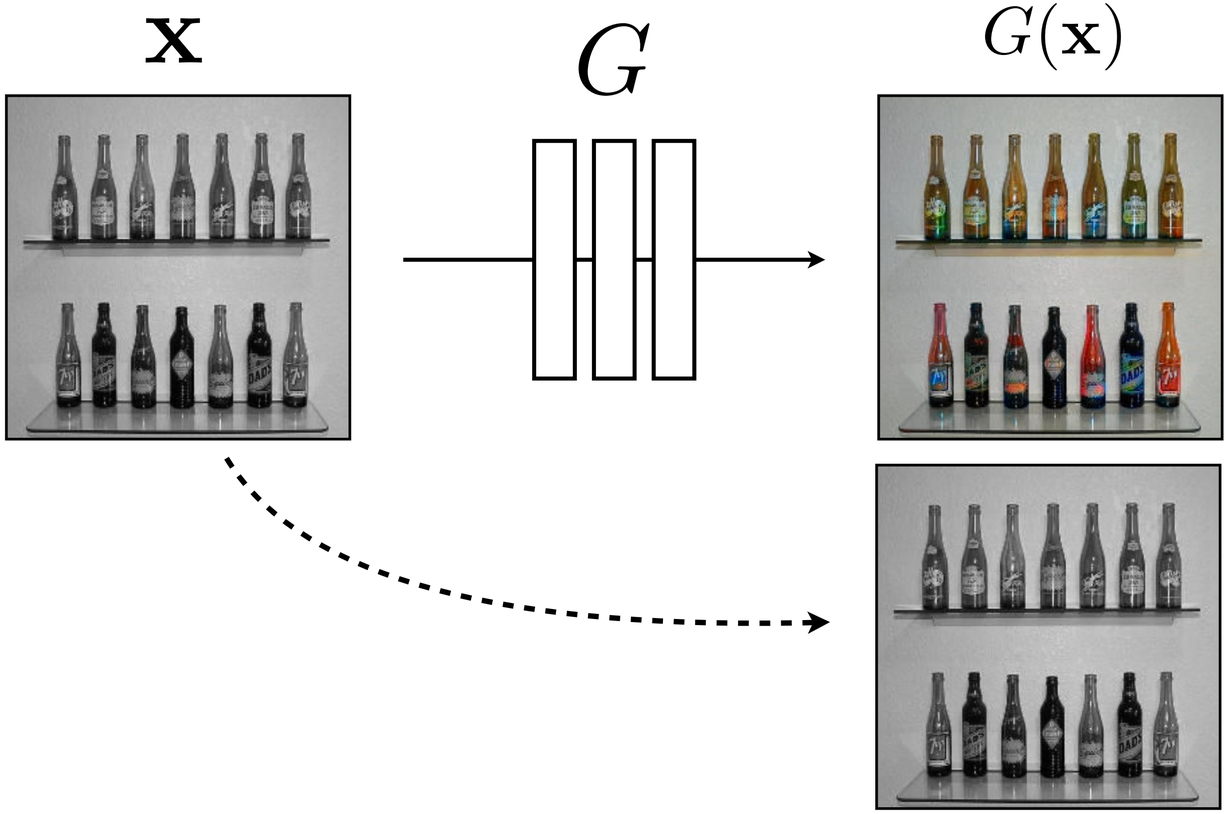
D

fake *pair*

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



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



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

Training Details: Loss function

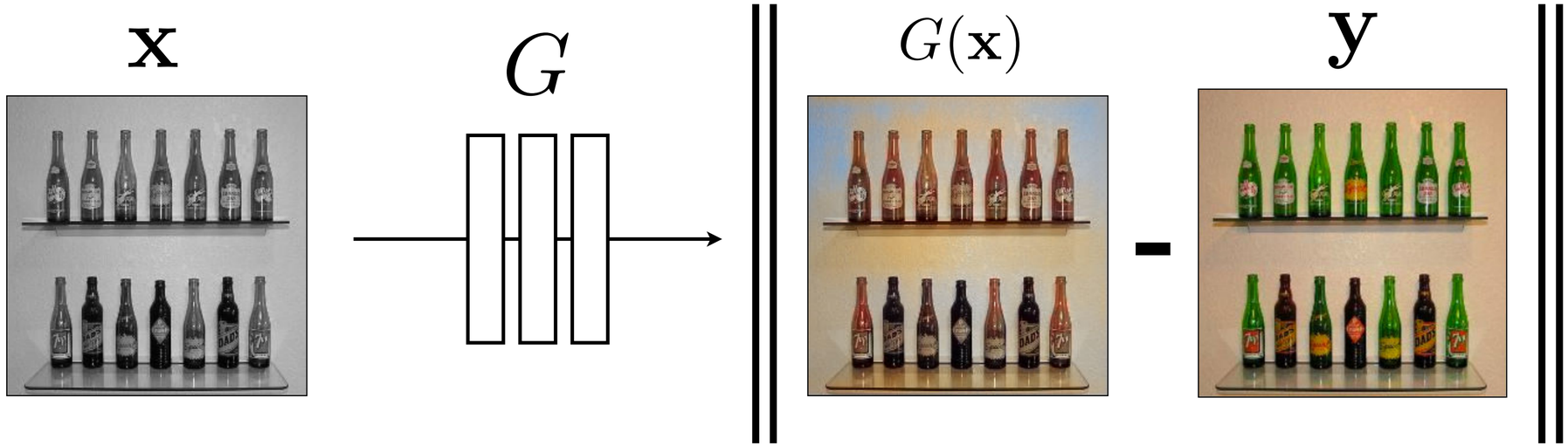
Conditional GAN

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

Training Details: Loss function

Conditional GAN

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

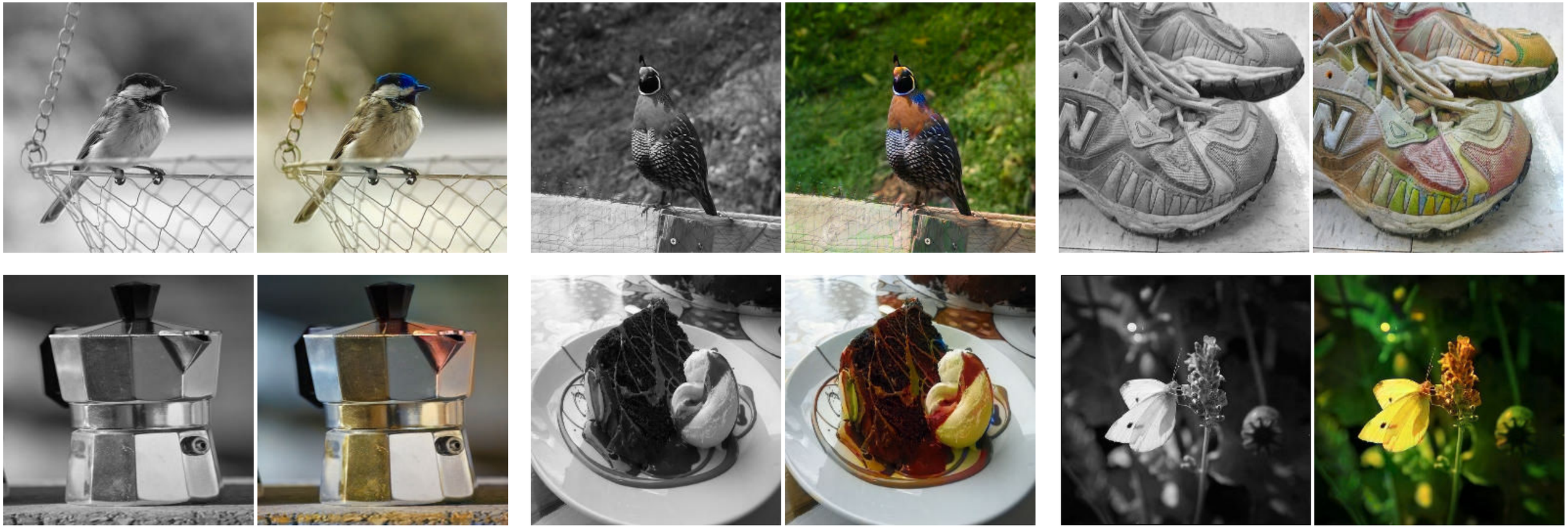


Stable training + fast convergence

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

BW → Color

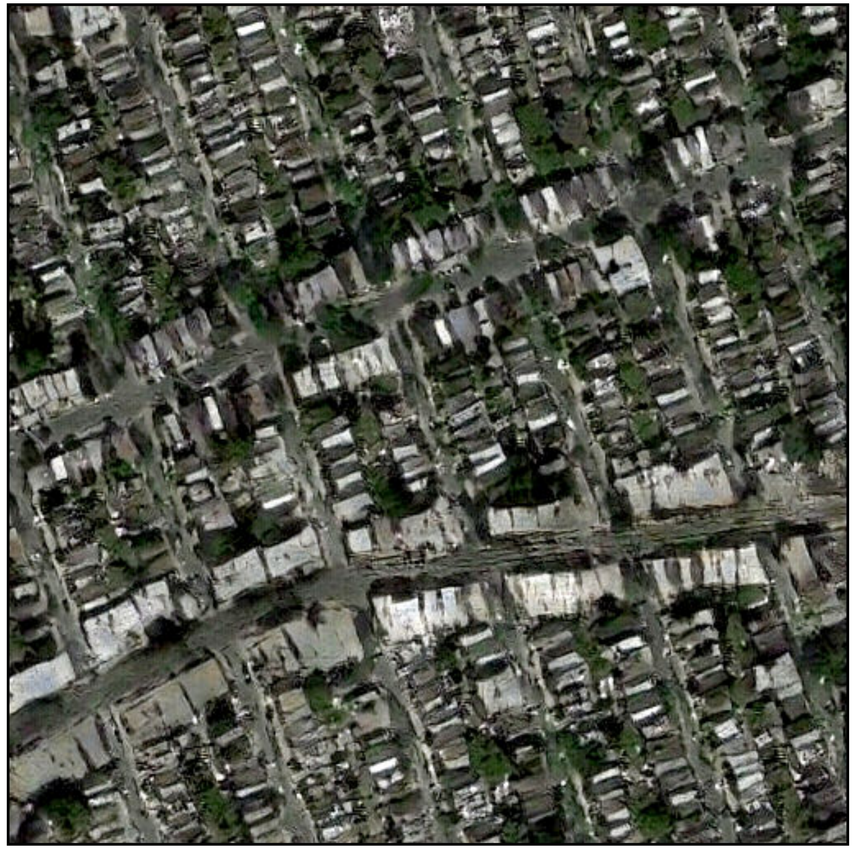
Input Output Input Output Input Output



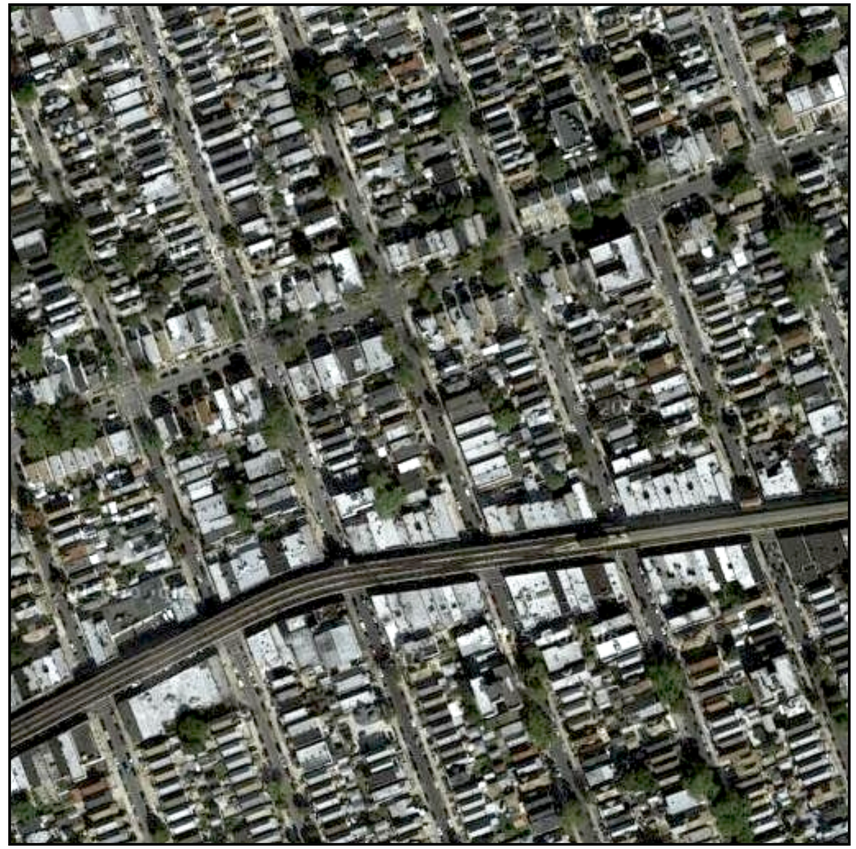
Data from [Russakovsky et al. 2015]

Input

Output



Groundtruth



Data from
[maps.google.com]

Input

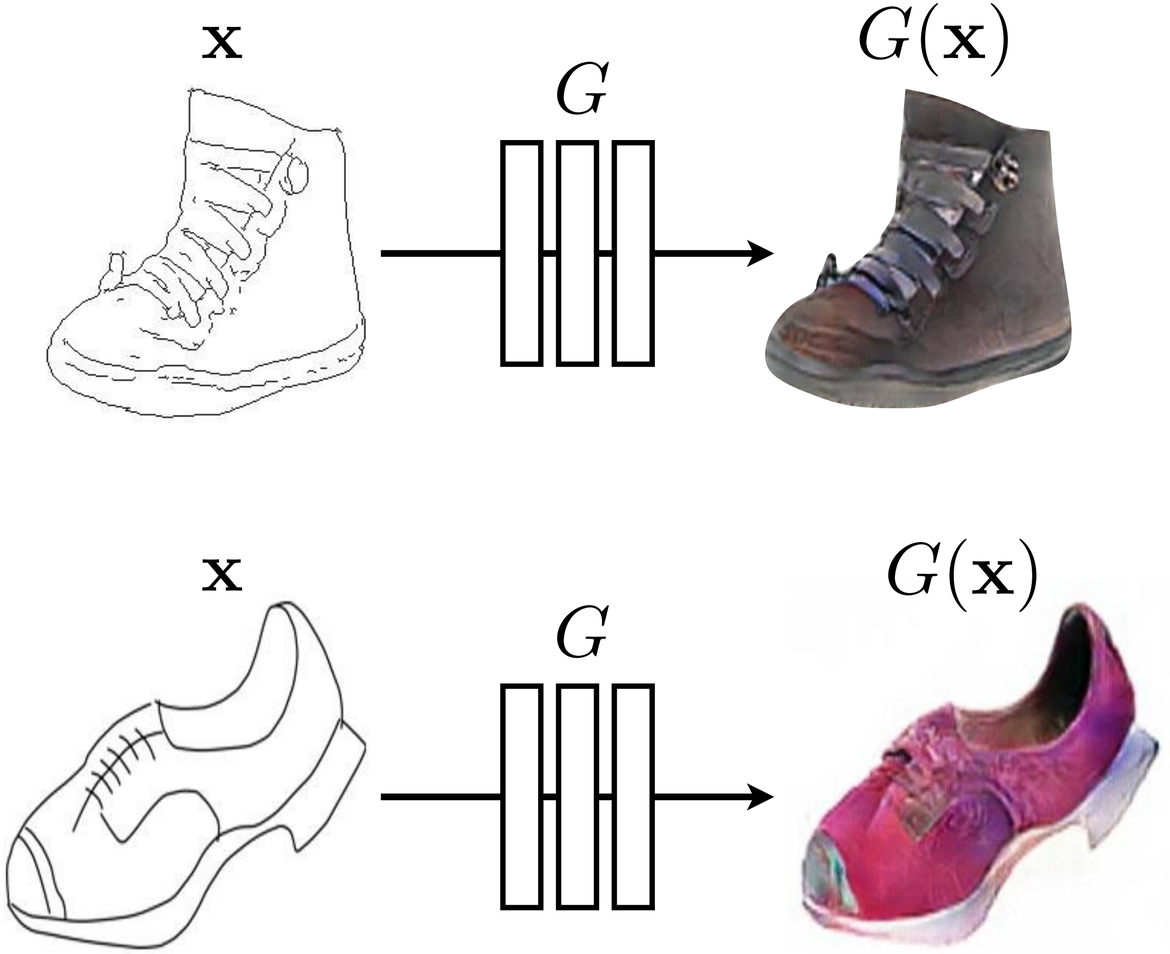
Output

Groundtruth

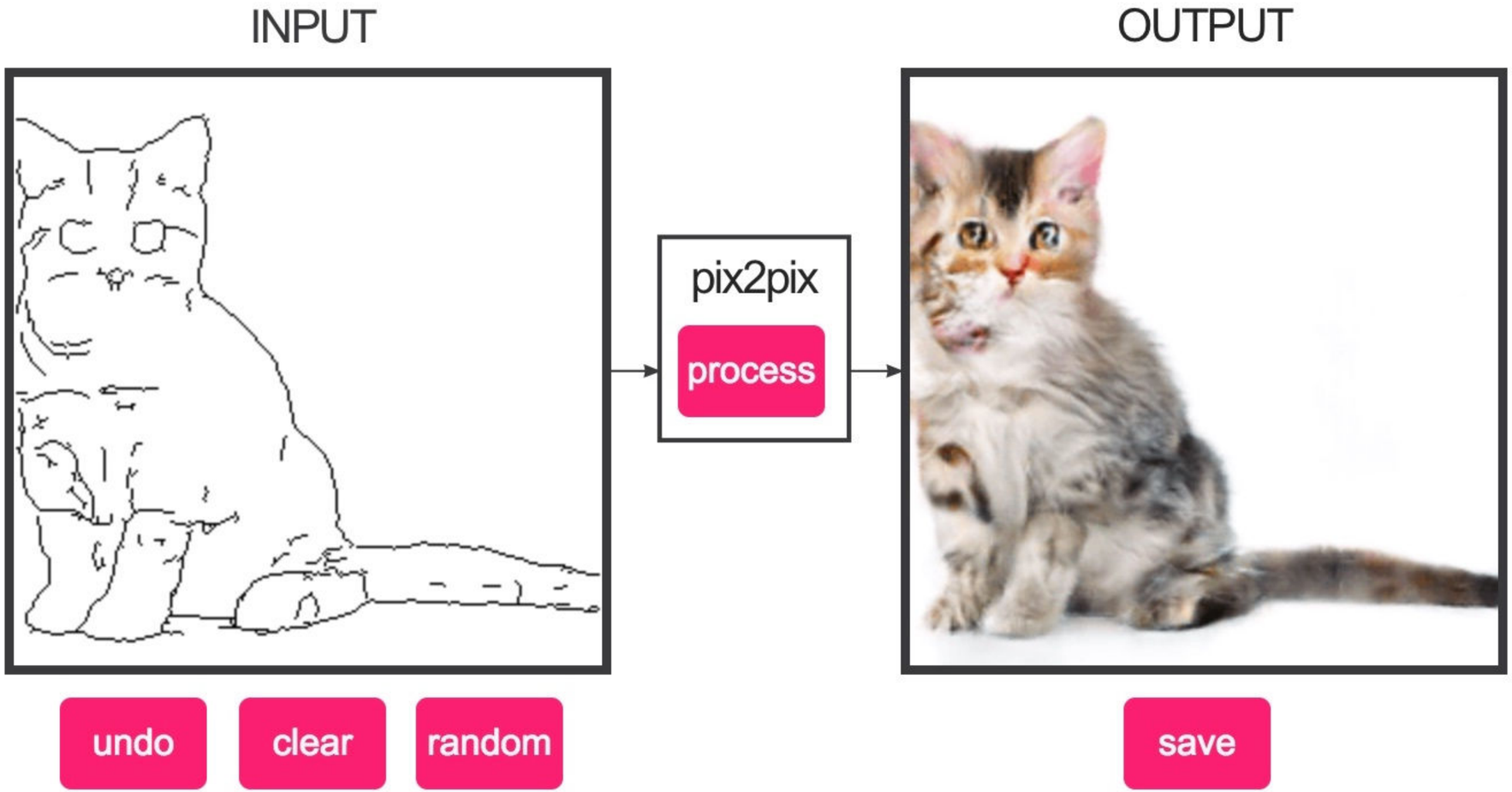
Data from [maps.google.com]

Training data

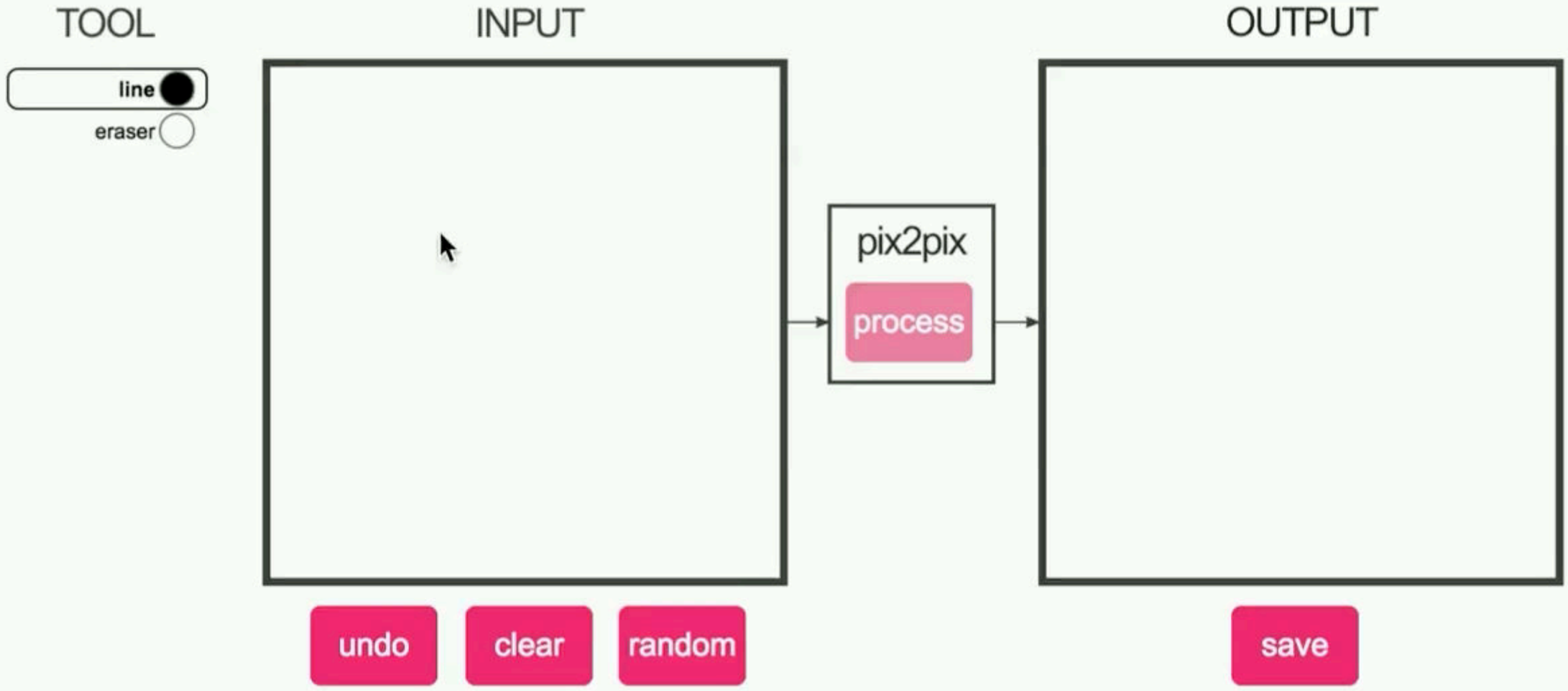
[HED, Xie & Tu, 2015]

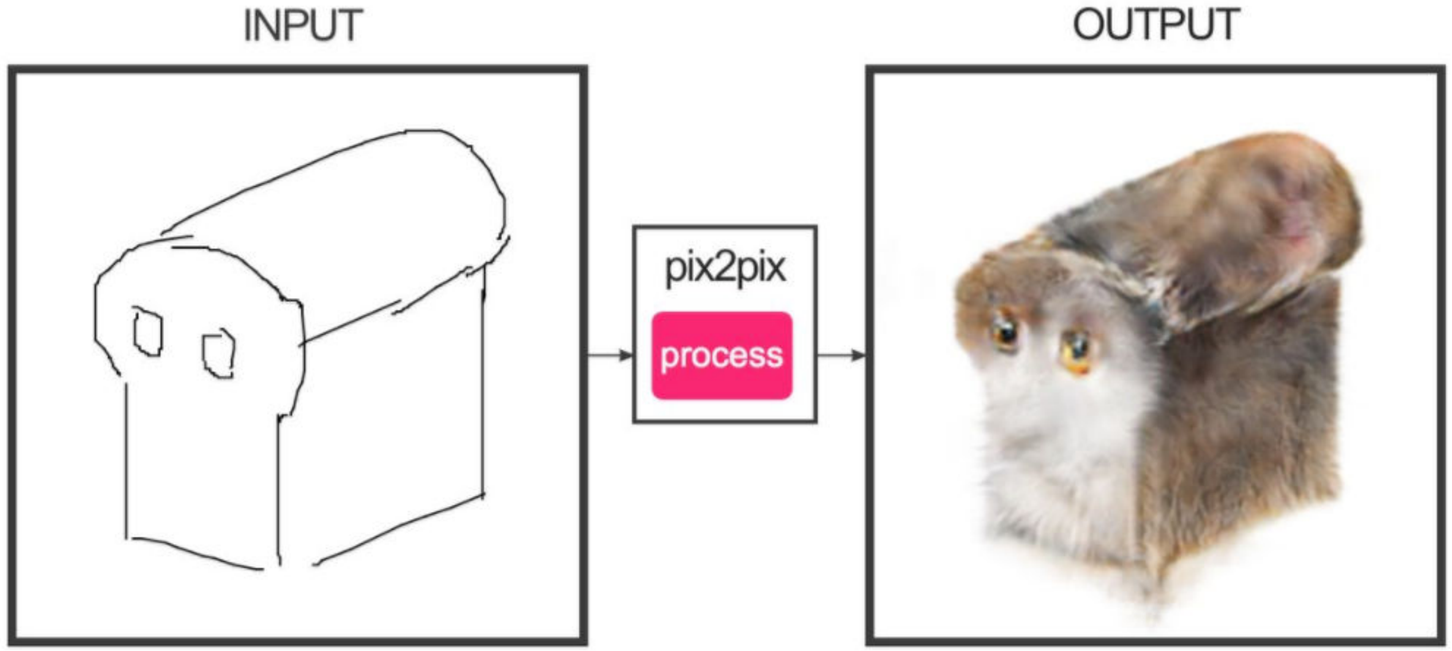


#edges2cats [Chris Hesse]



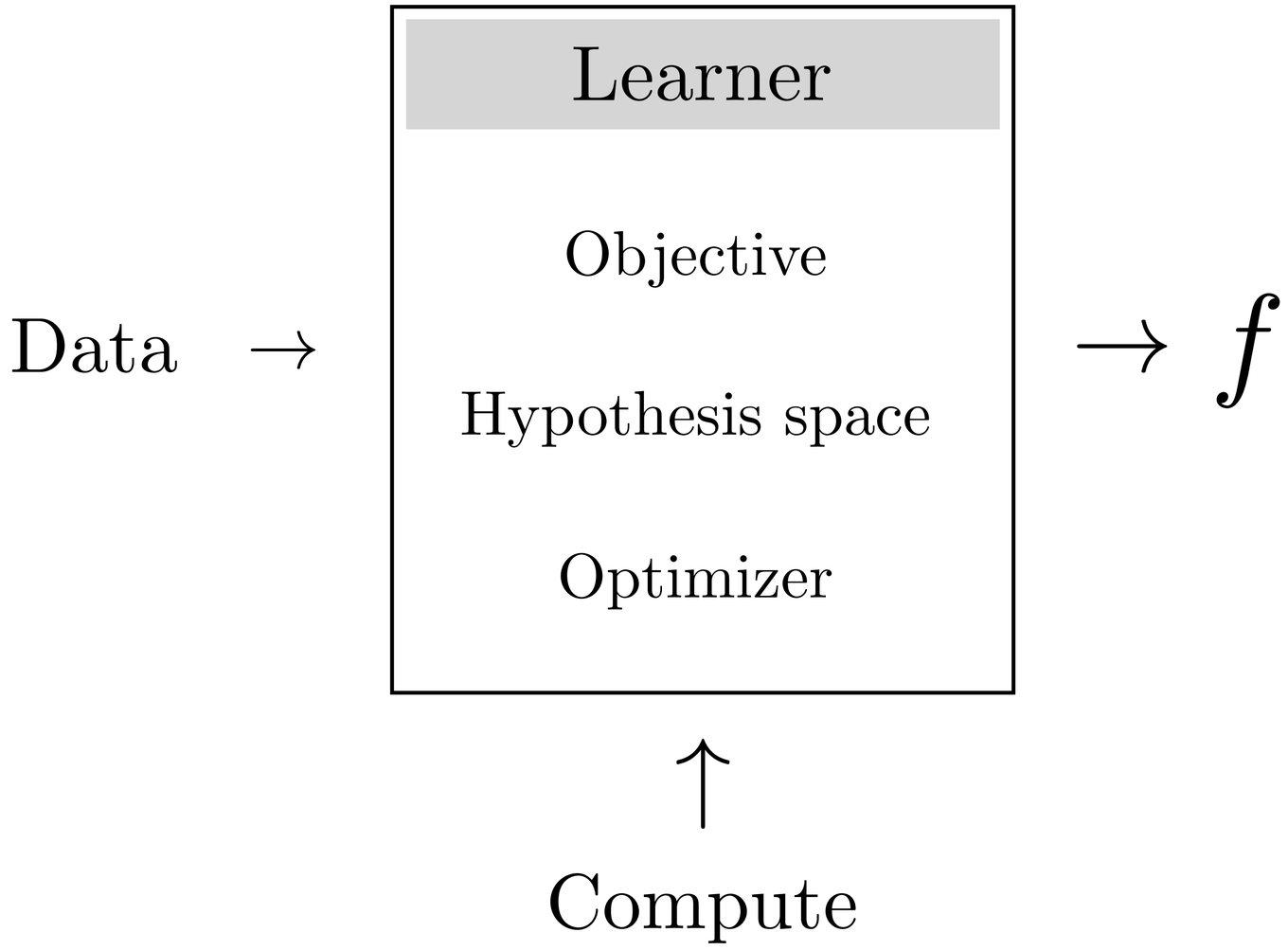
edges2cats


TOOL
line
eraser
INPUT
pix2pix
process
OUTPUT
undo
clear
random
save



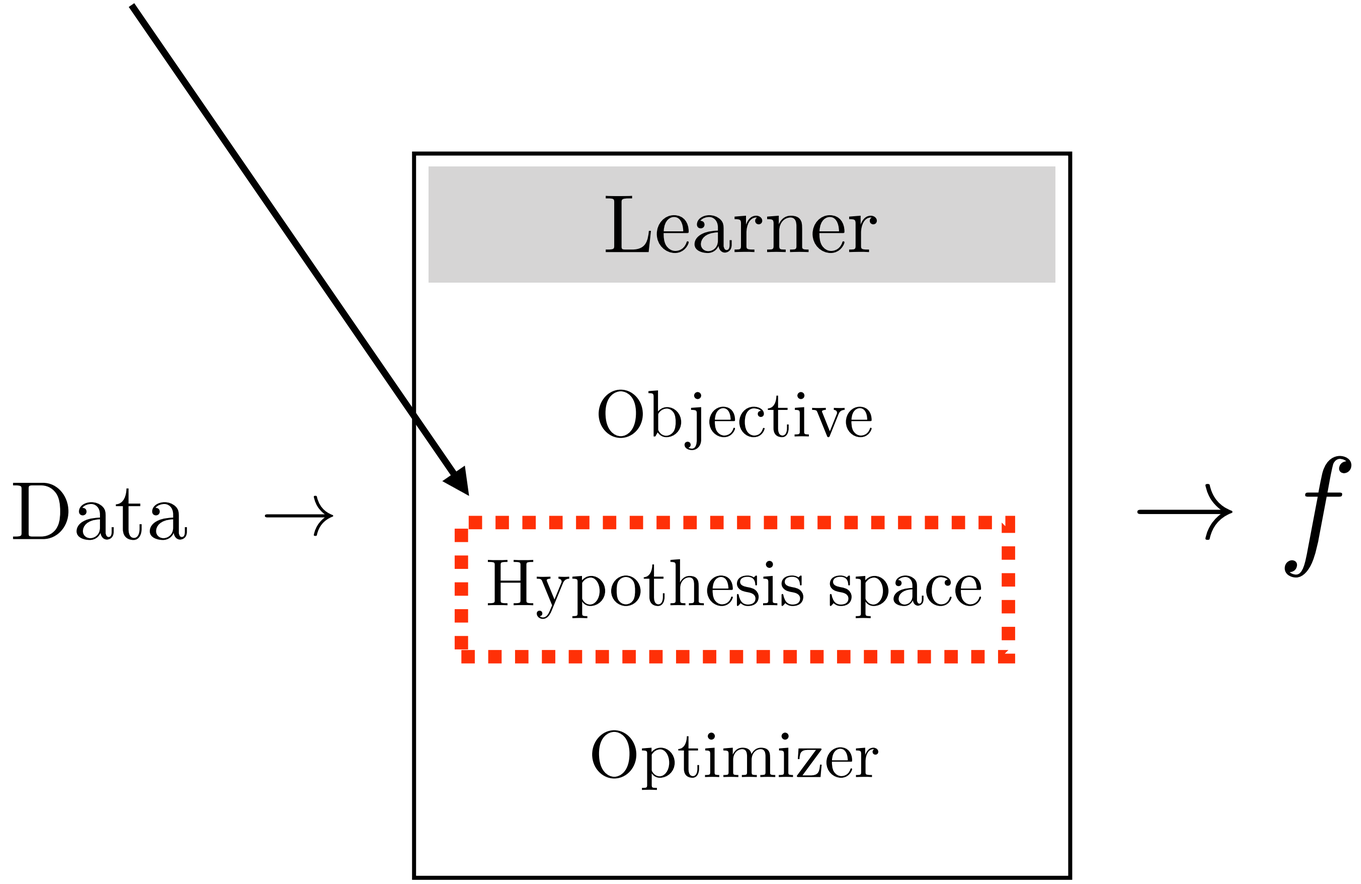
Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

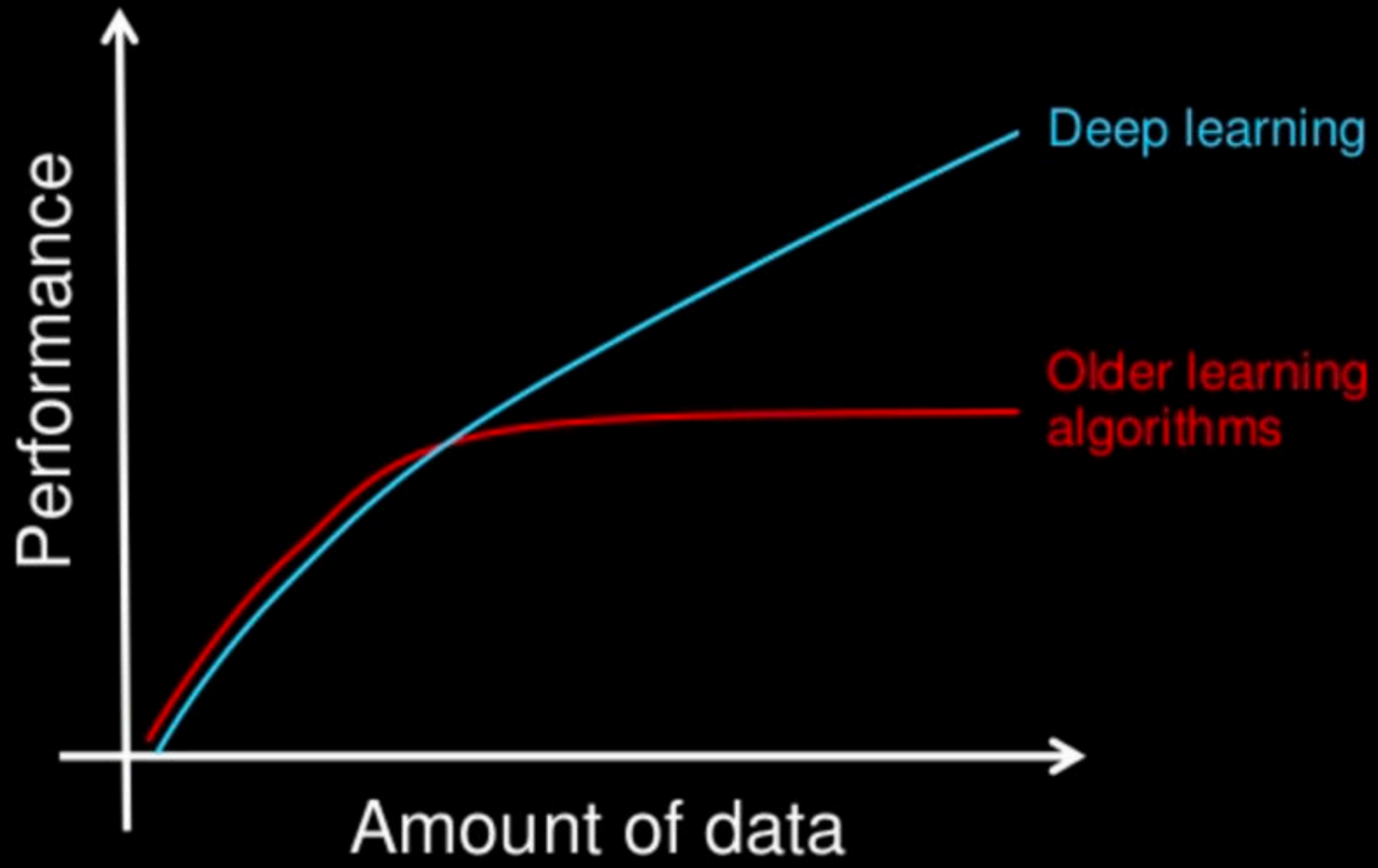

Learner
Objective
Hypothesis space
Optimizer
Data
$\rightarrow$
$\rightarrow f$
$\uparrow$
Compute

Deep learning in 2012

Use a **hypothesis space** that can model complex structure (e.g., a CNN)



Why deep learning



How do data science techniques scale with amount of data?

[Slide credit: Andrew Ng]

What's the object class of the center pixel?




f "Bush"

f "Bush"

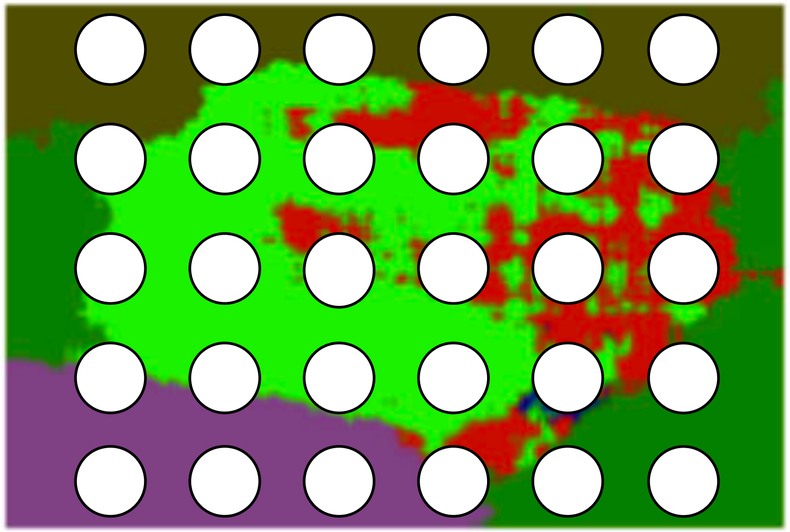
f "Bench"

f "Bench"

Fully-factored loss: $\psi(\hat{\mathbf{y}}, \mathbf{y}) = \sum_i \phi_i(\hat{\mathbf{y}}_i, \mathbf{y}_i)$

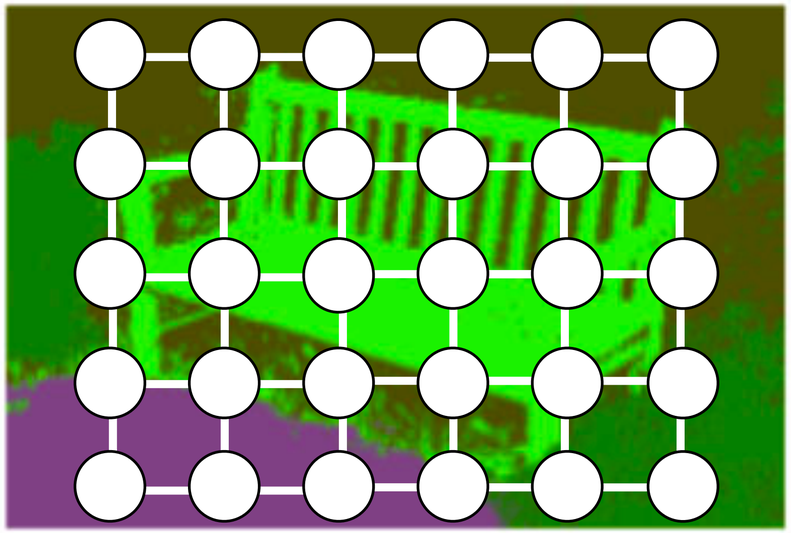
Input

Independent prediction per-pixel



$$\max \prod_i p(y_i|\mathbf{x})$$

Find a configuration of compatible labels

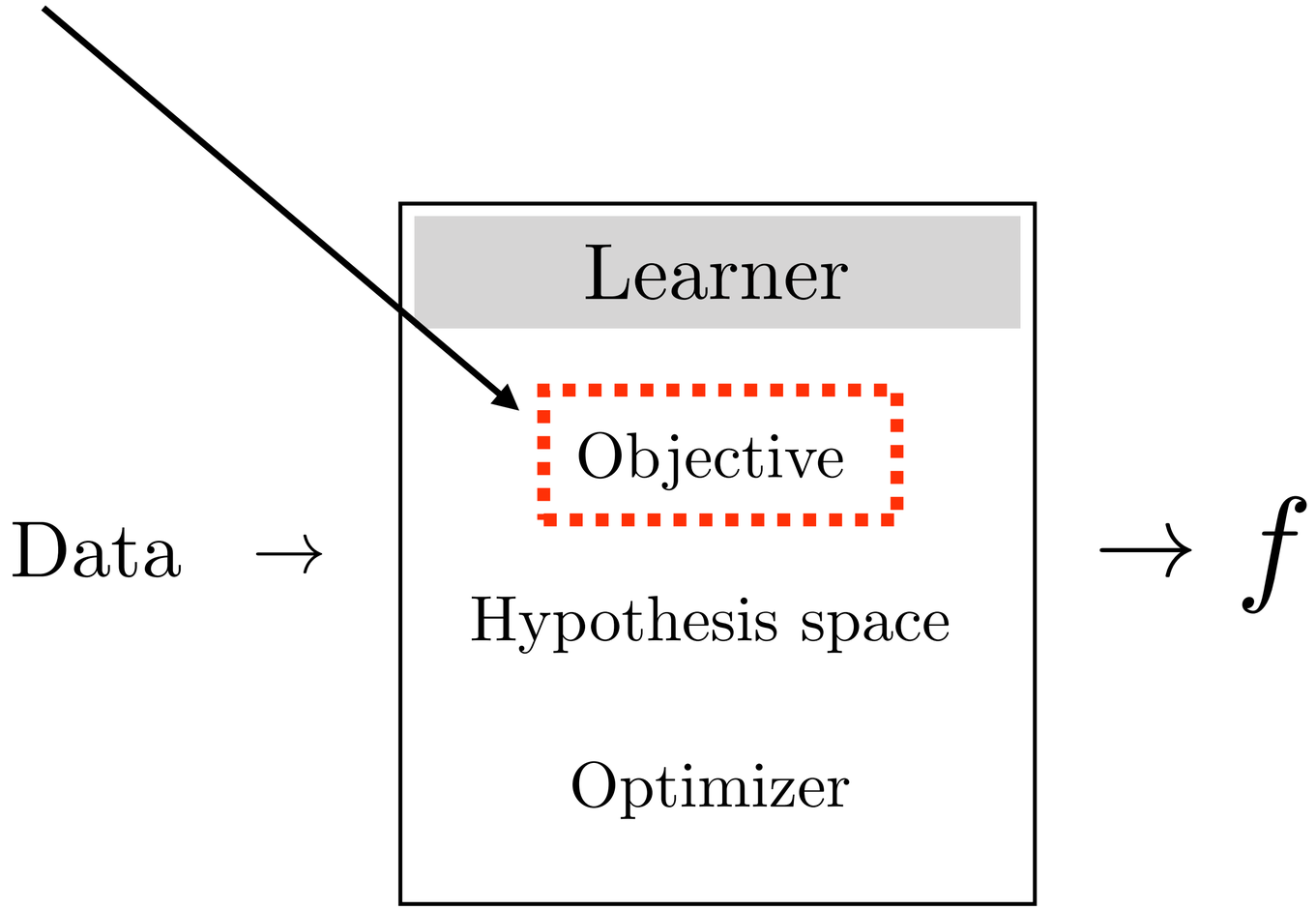


$$\max \frac{1}{Z} \prod_{i,j} p(y_i, y_j|\mathbf{x})$$

["Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials", Krahenbuhl and Koltun, NeurIPS 2011]

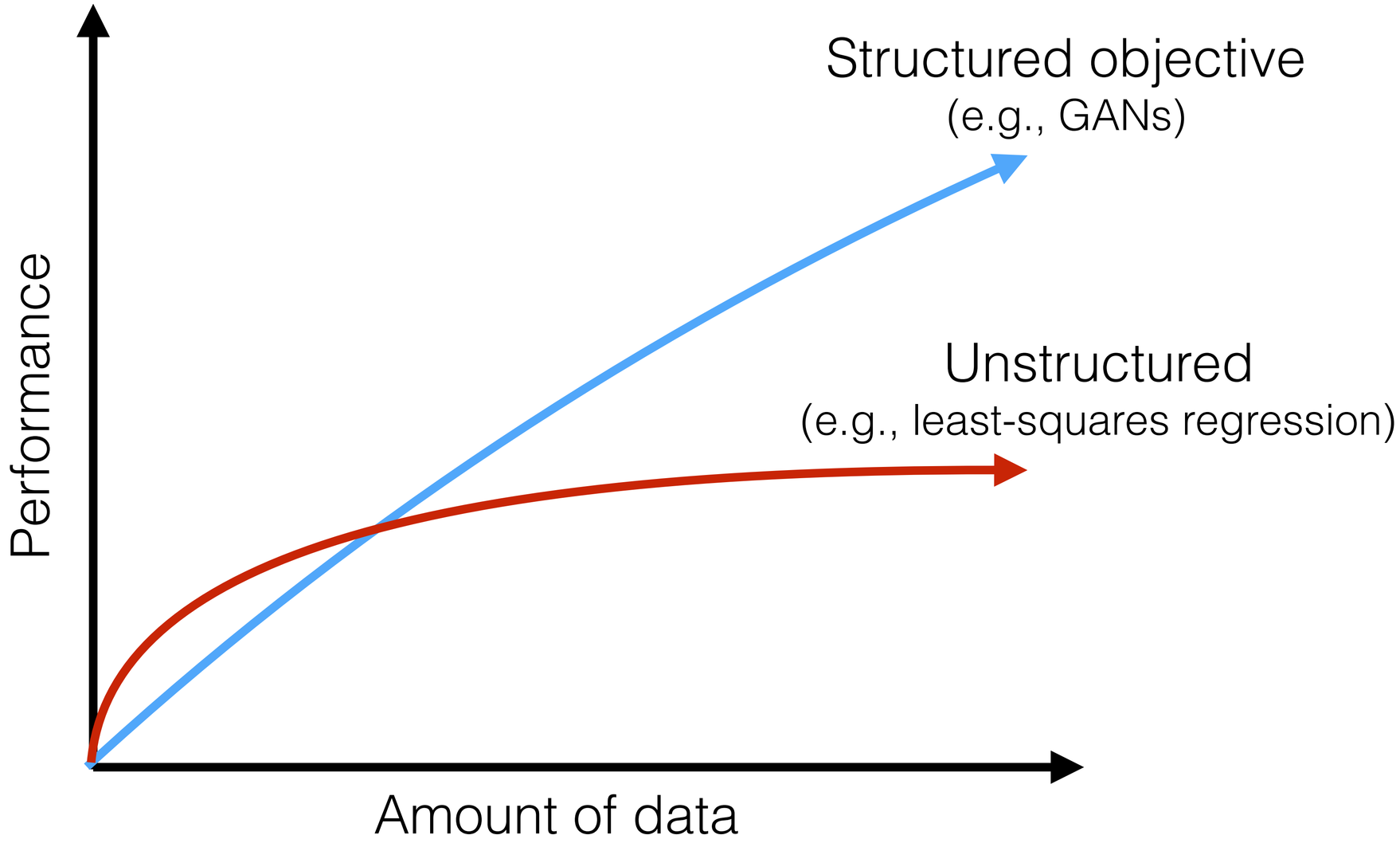
Structured prediction

Use an **objective** that can model structure! (e.g., a graphical model, a GAN, etc)

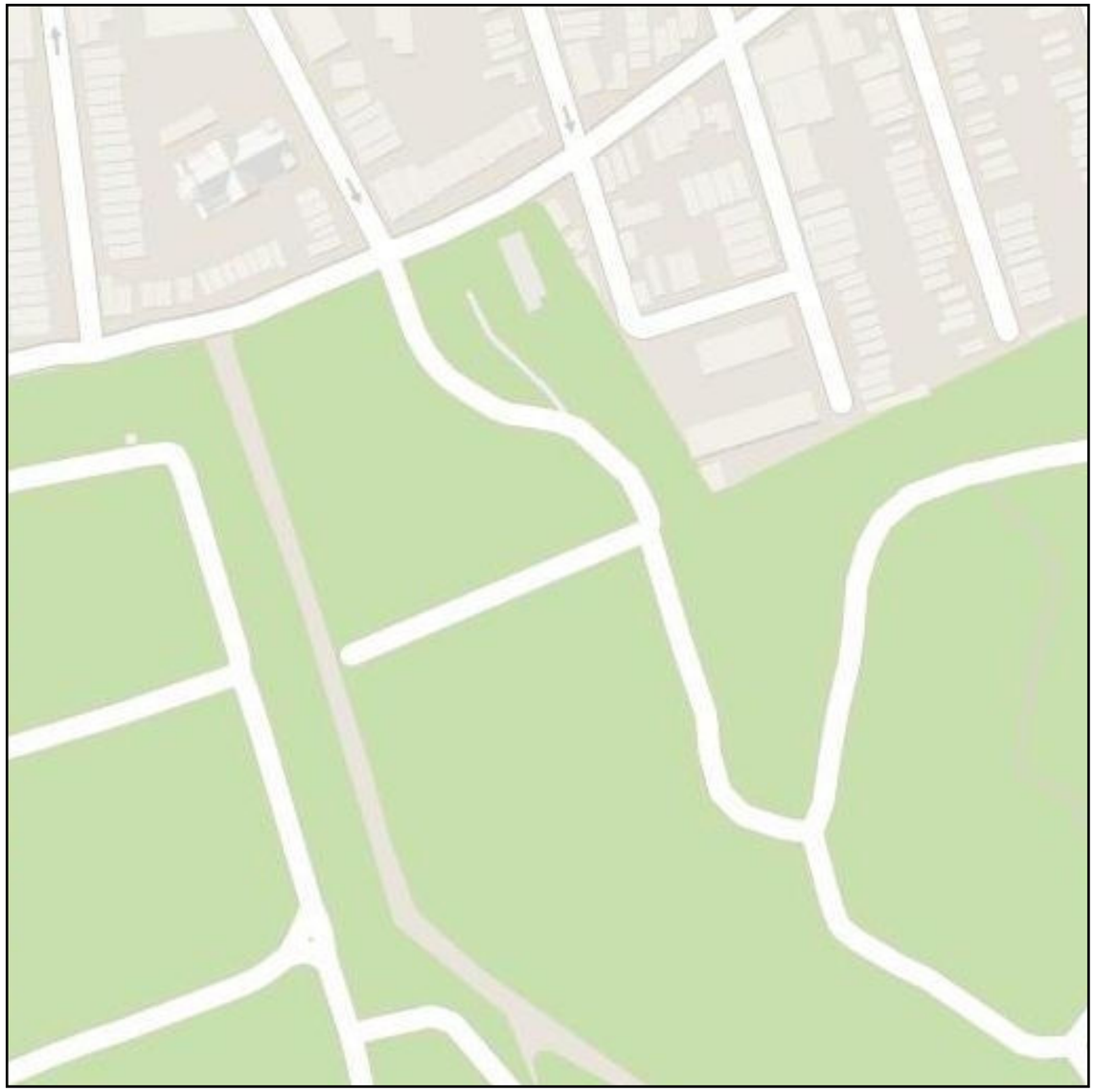


Why structured objectives

(cartoon)

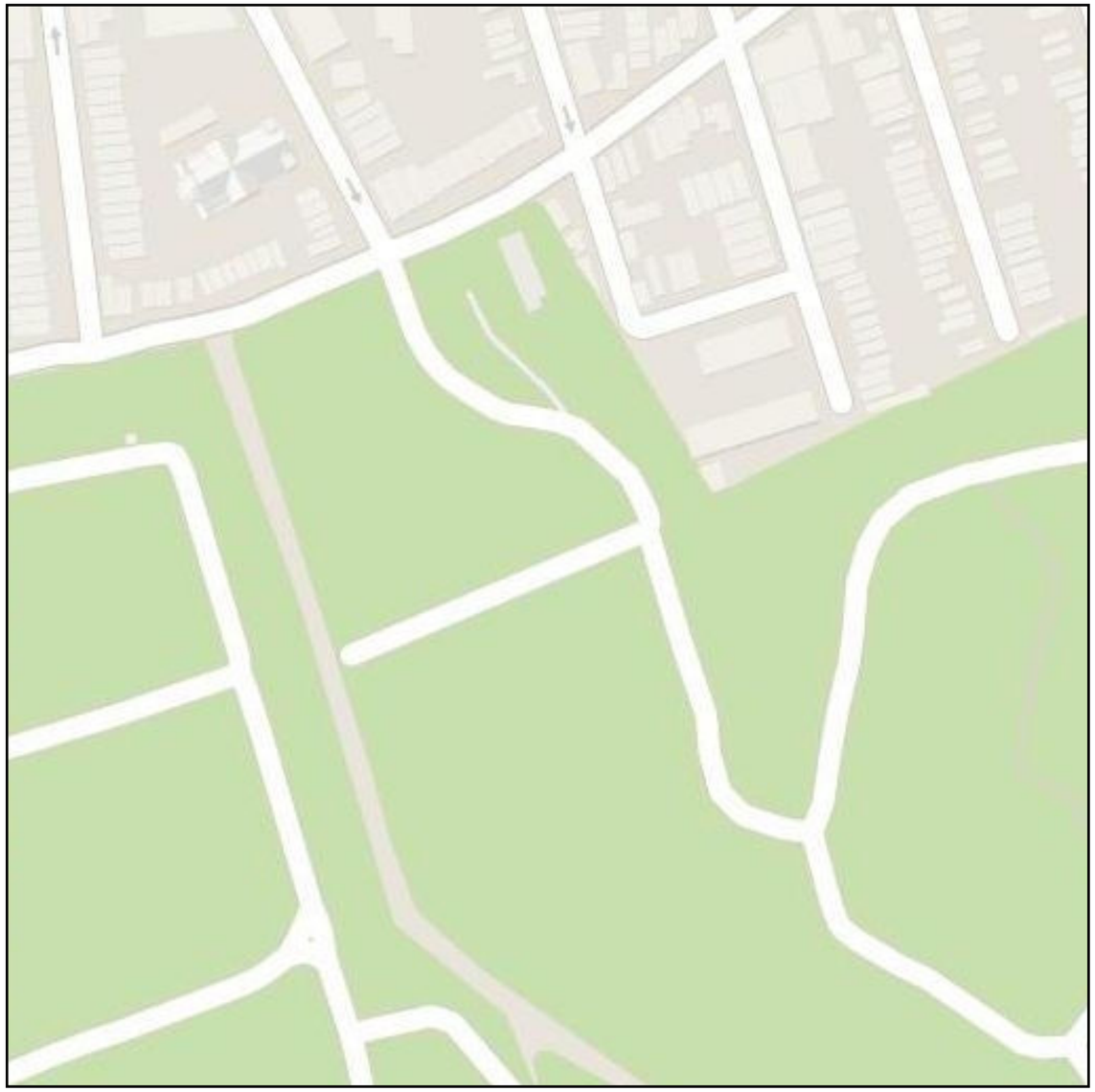


Input



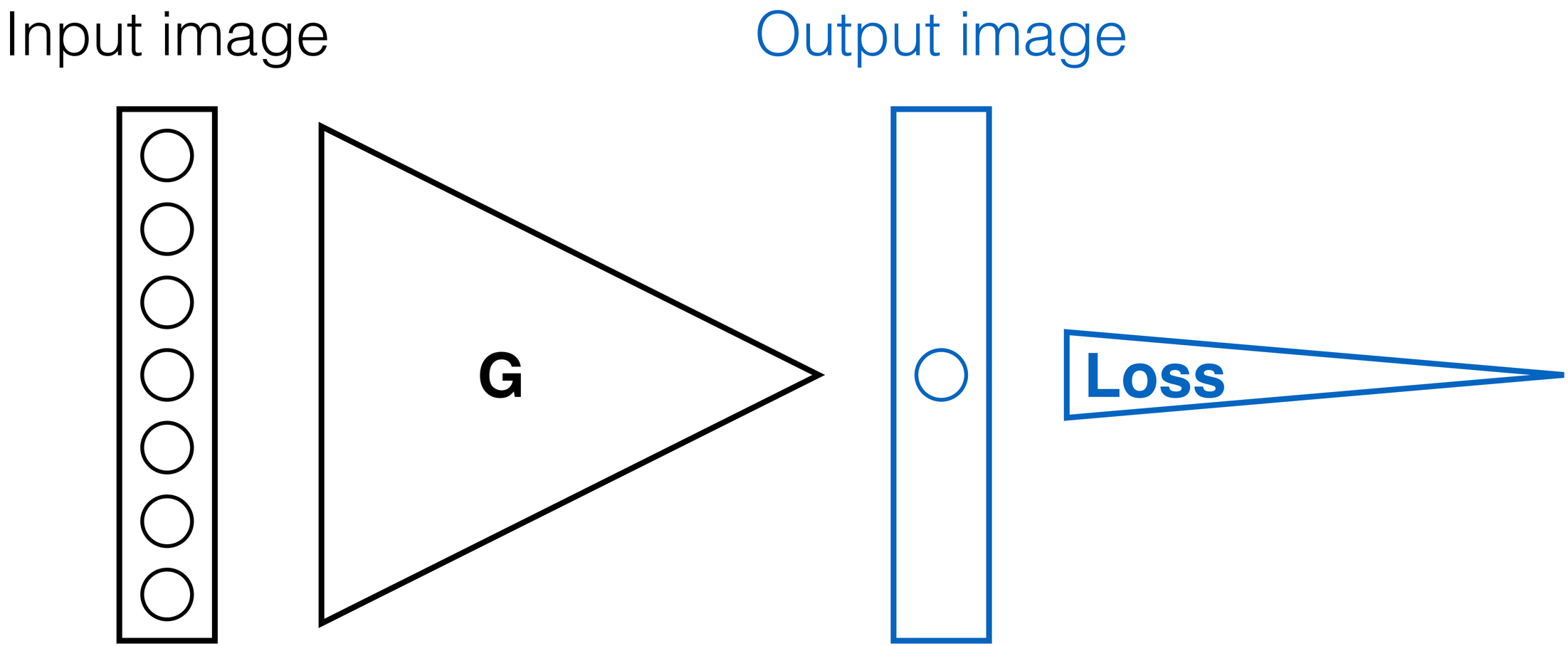
Unstructured prediction (L1)

Input

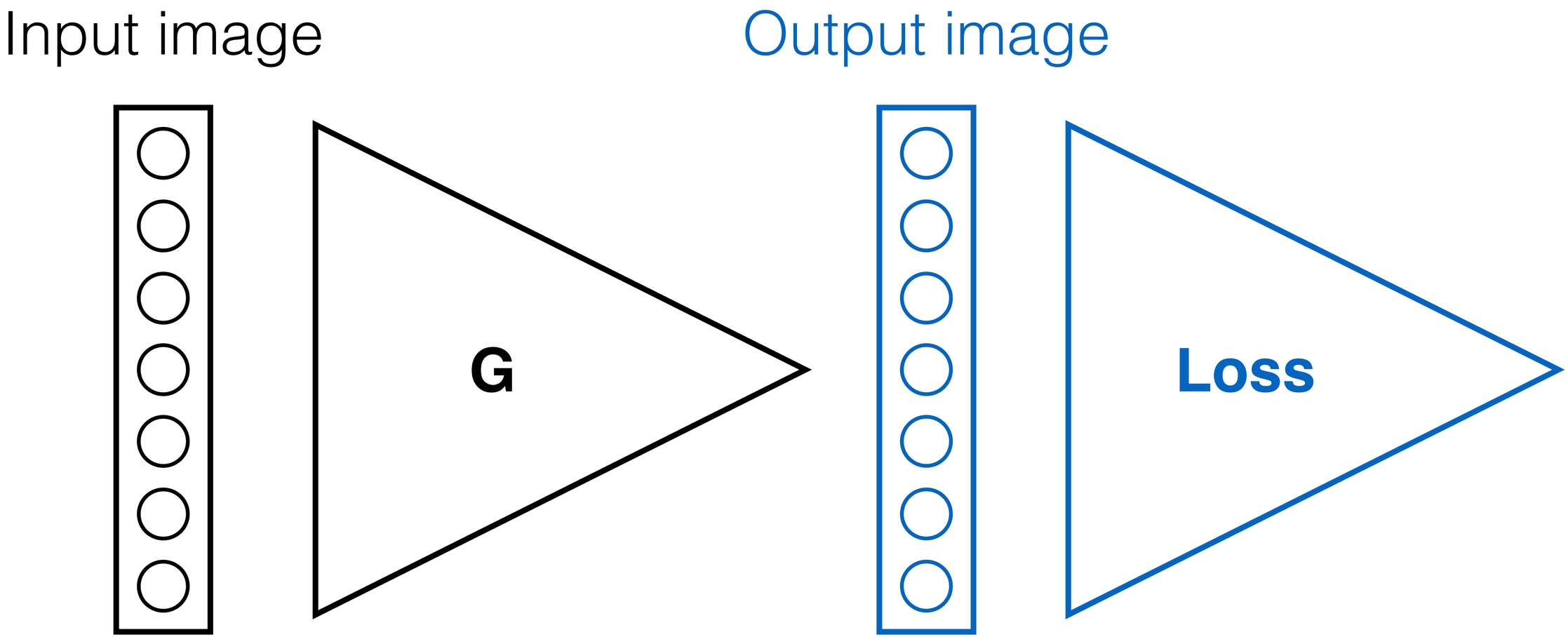


Structured Prediction (cGAN)

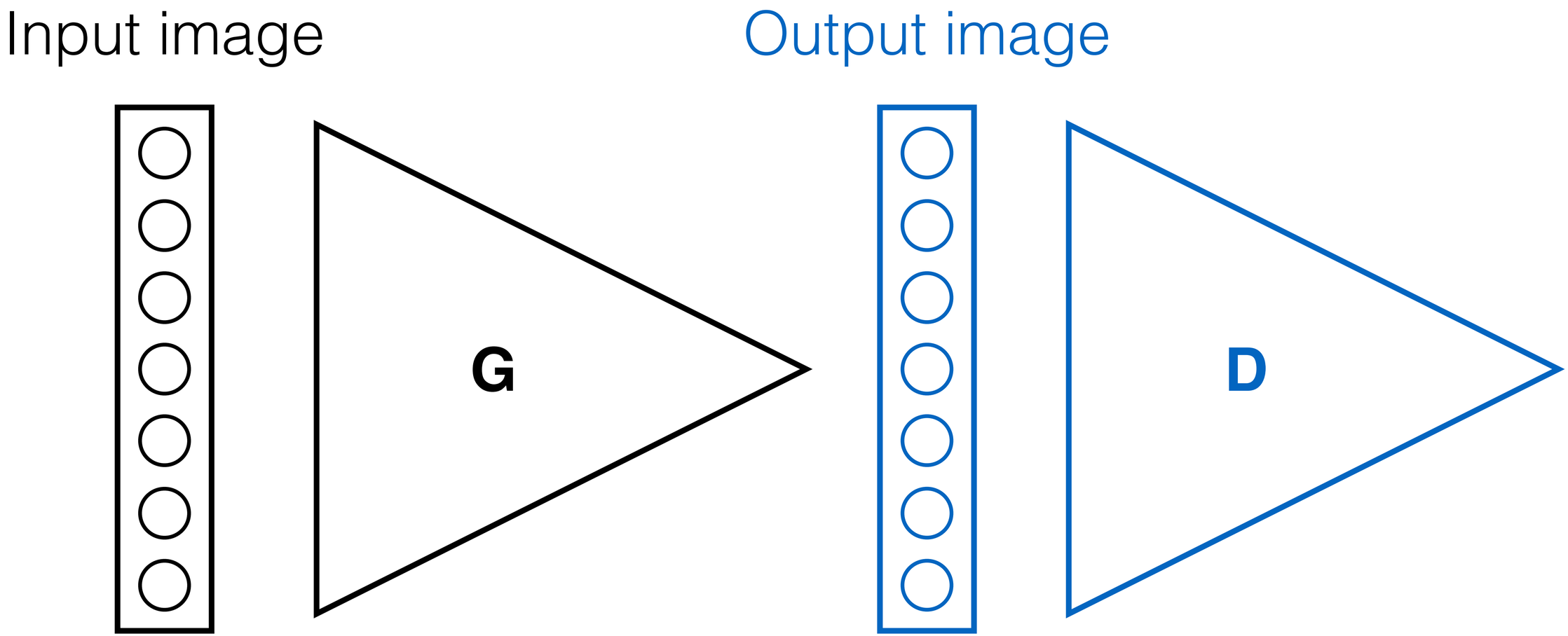
Unstructured prediction



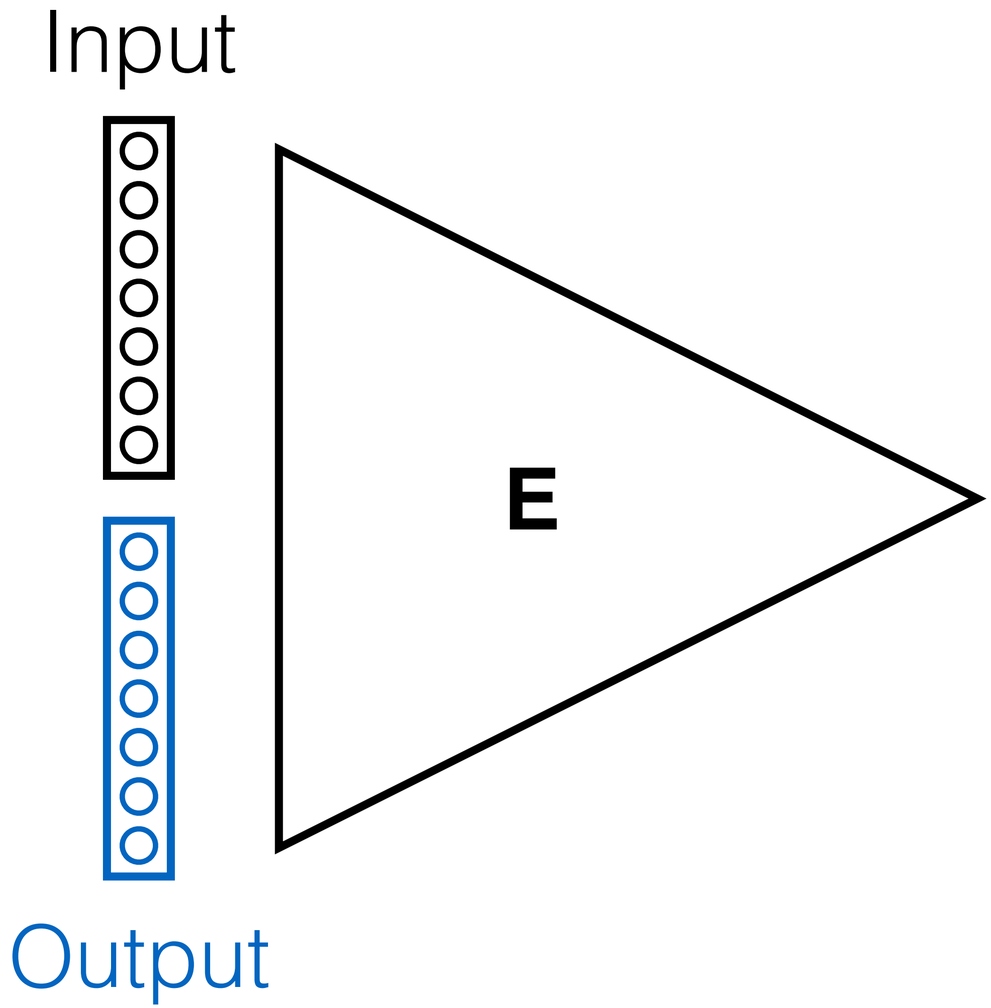
Structured prediction



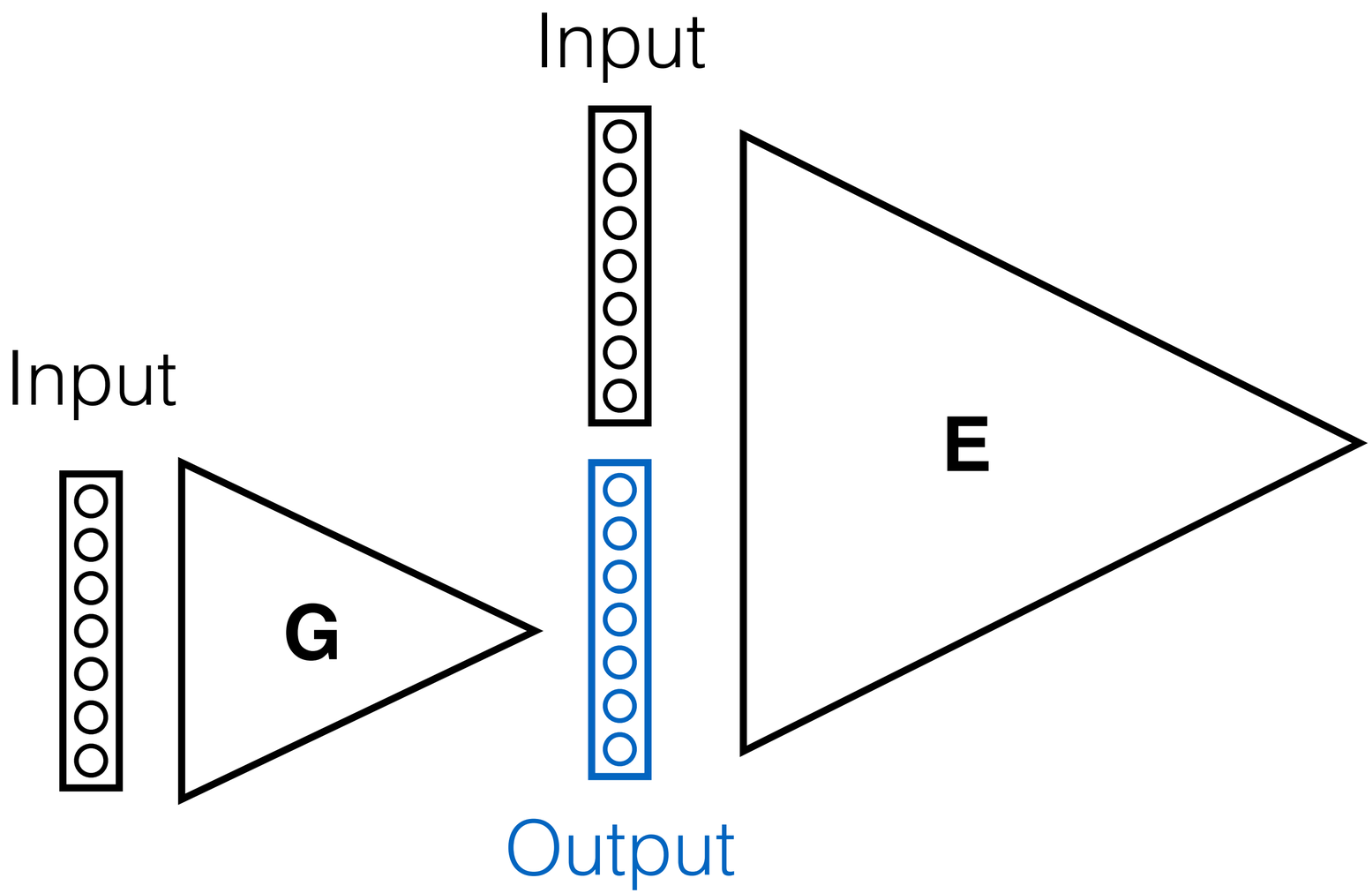
GANs



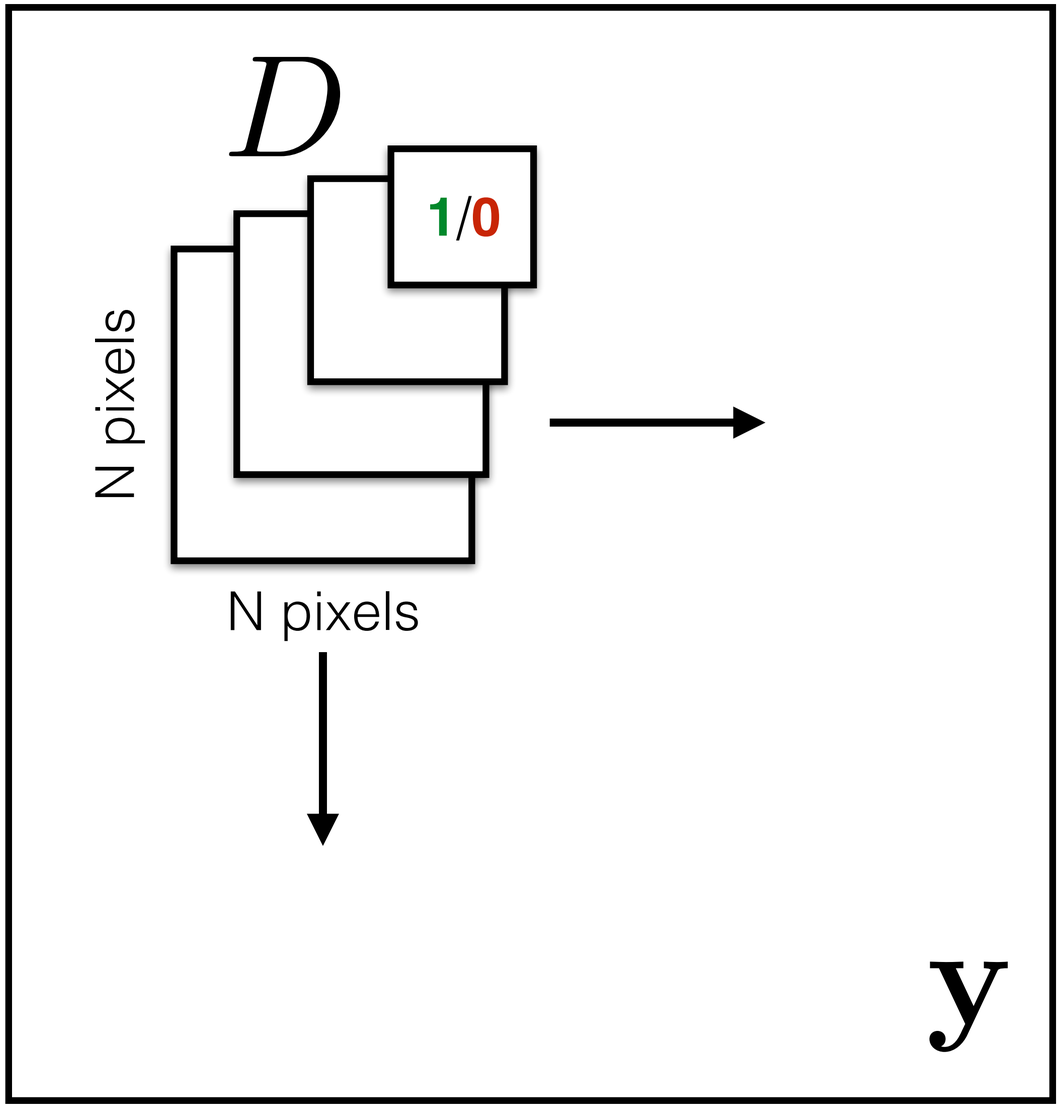
Energy-based models, Graphical models



Conditional GANs



Patch Discriminator

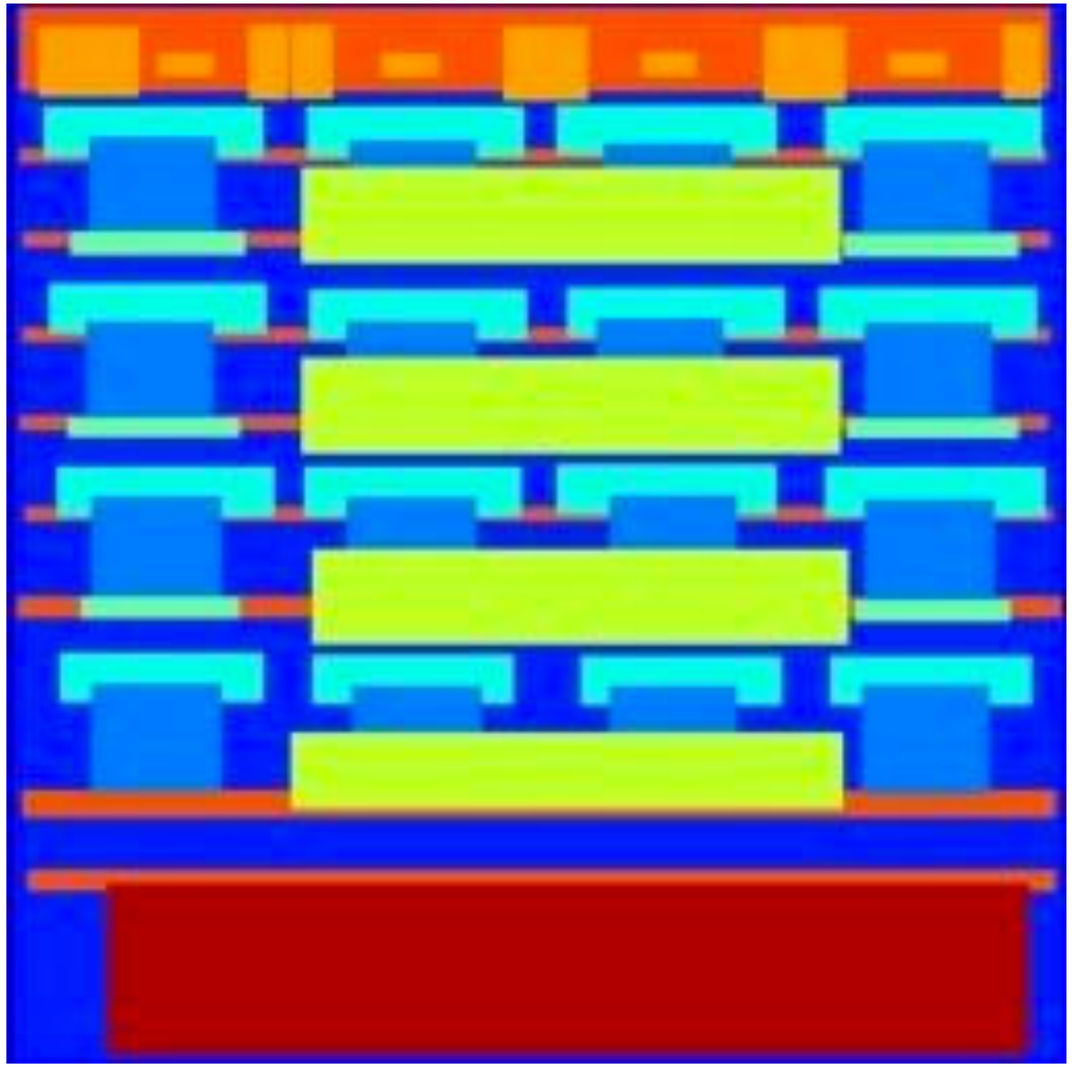


Rather than penalizing if output *image* looks fake, penalize if each overlapping *patch* in output looks fake

[Li & Wand 2016]
[Shrivastava et al. 2017]
[Isola et al. 2017]

Labels → Facades

Input

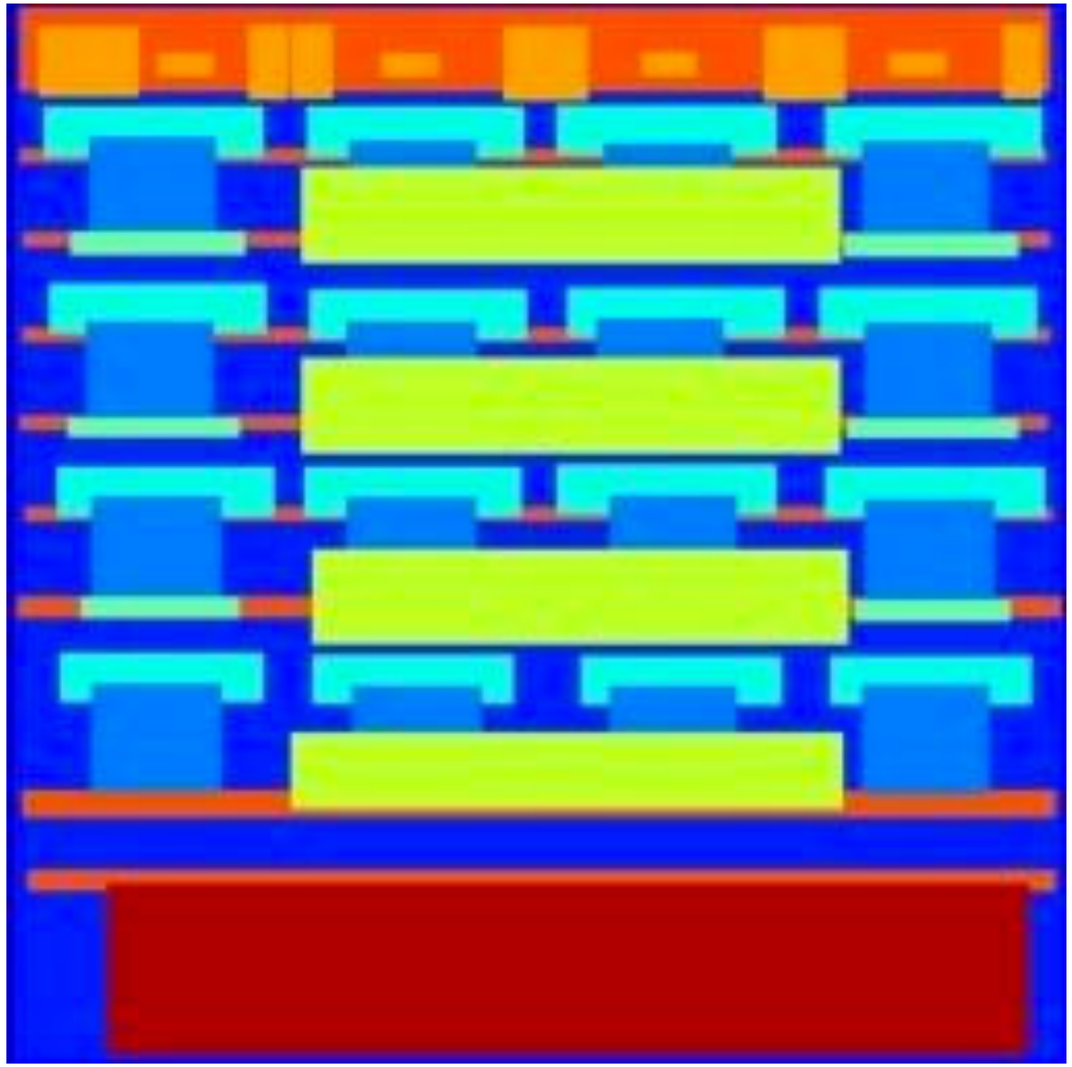


1x1 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

Input

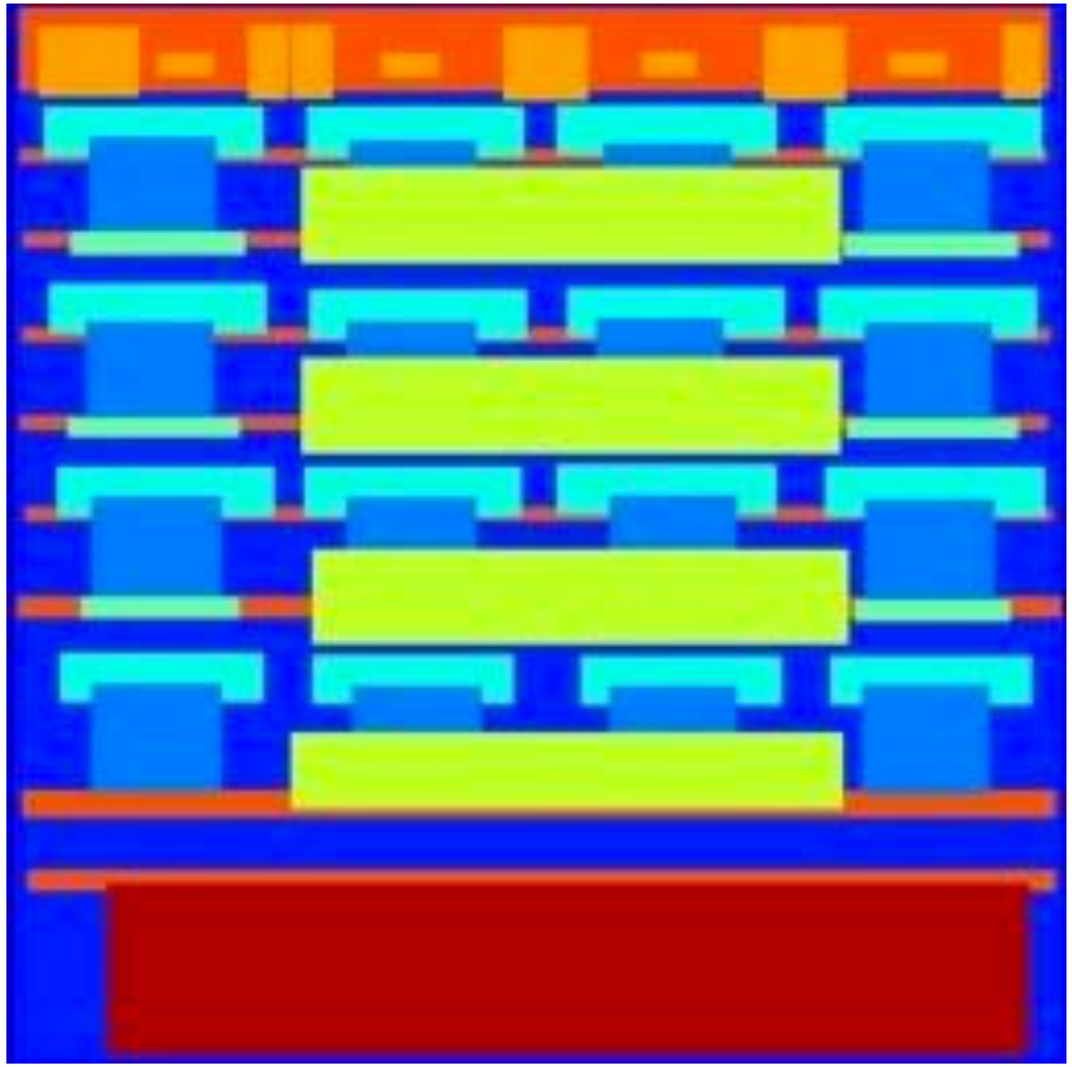


16x16 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

Input

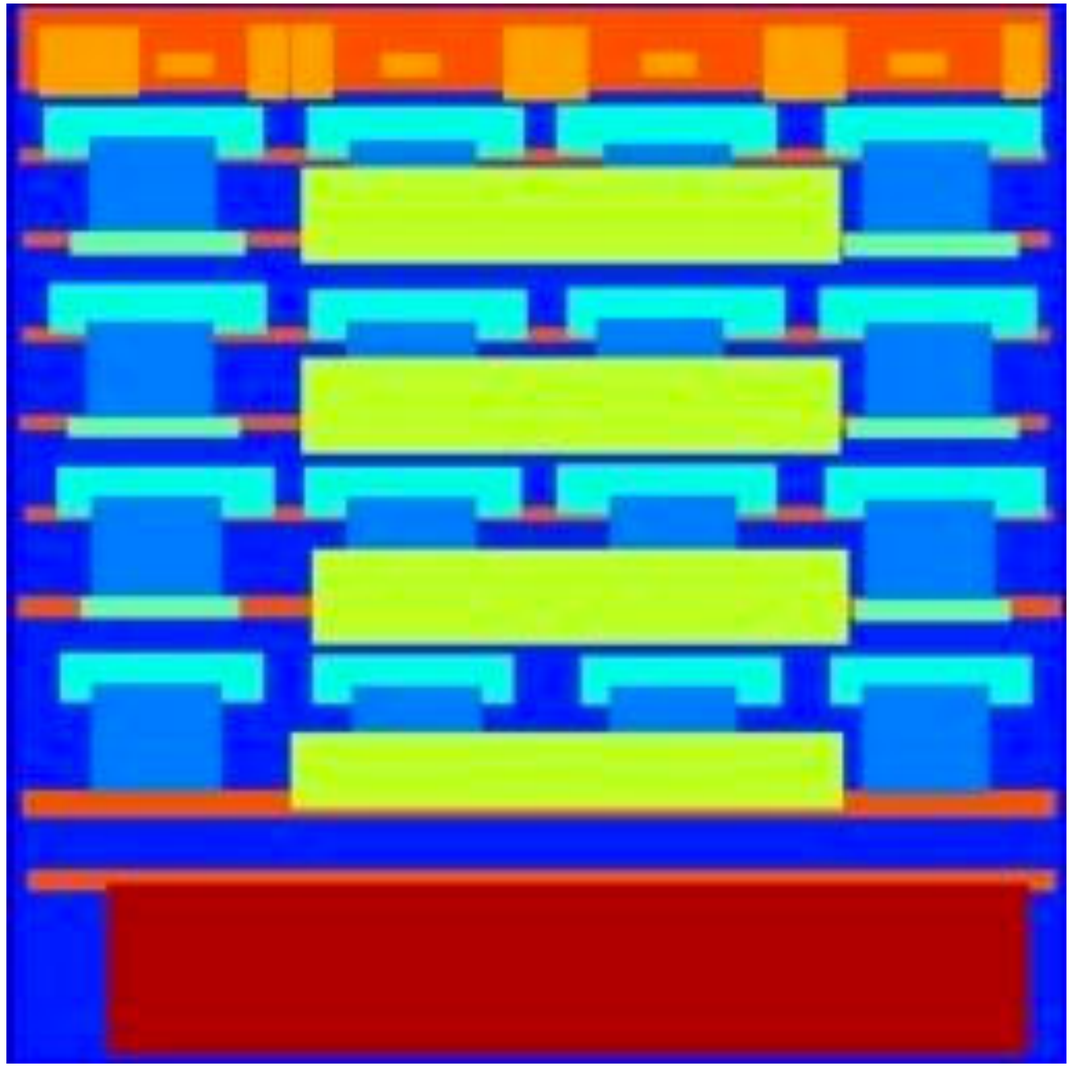


70x70 Discriminator

Data from [Tylecek, 2013]

Labels → Facades

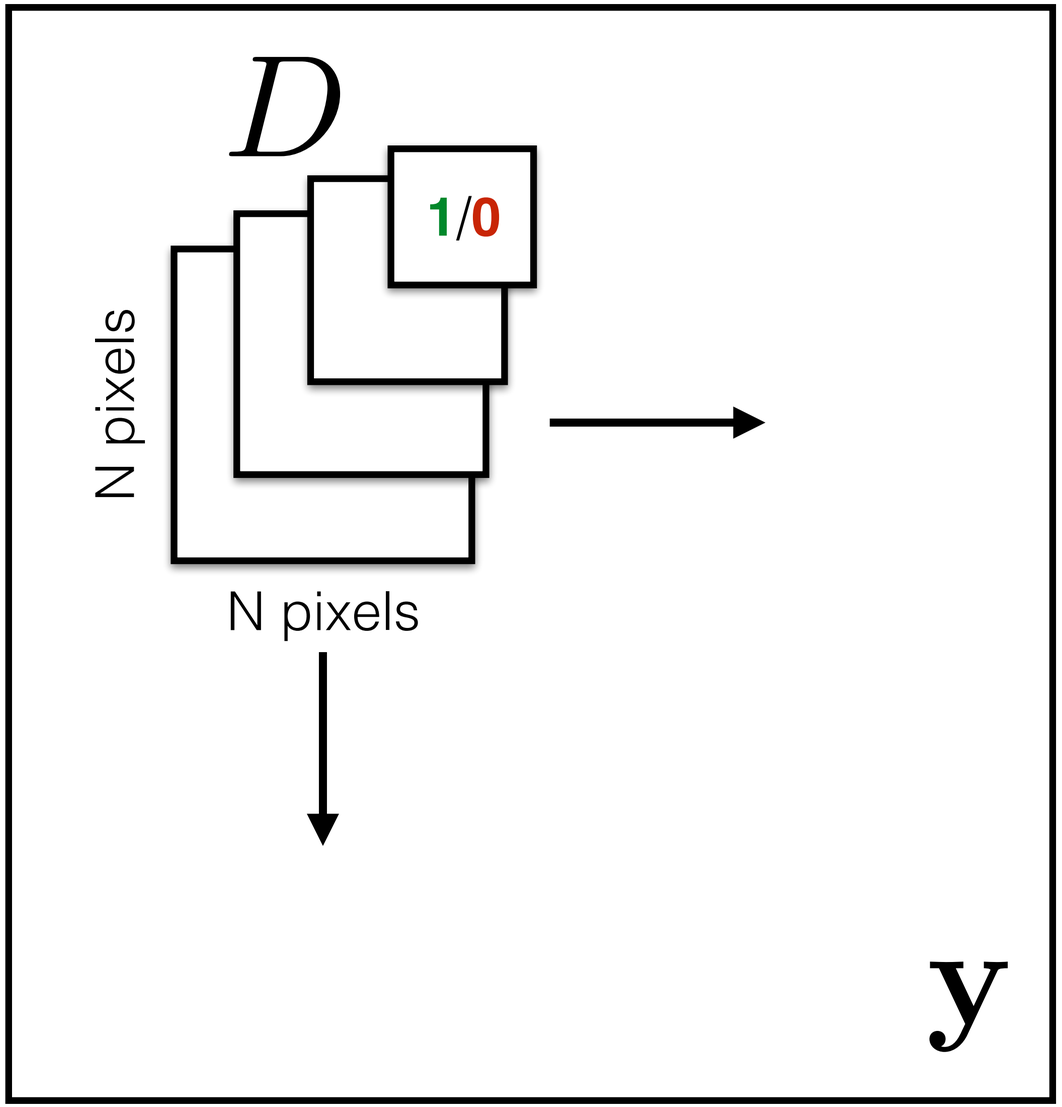
Input



Full image Discriminator

Data from [Tylecek, 2013]

Patch Discriminator

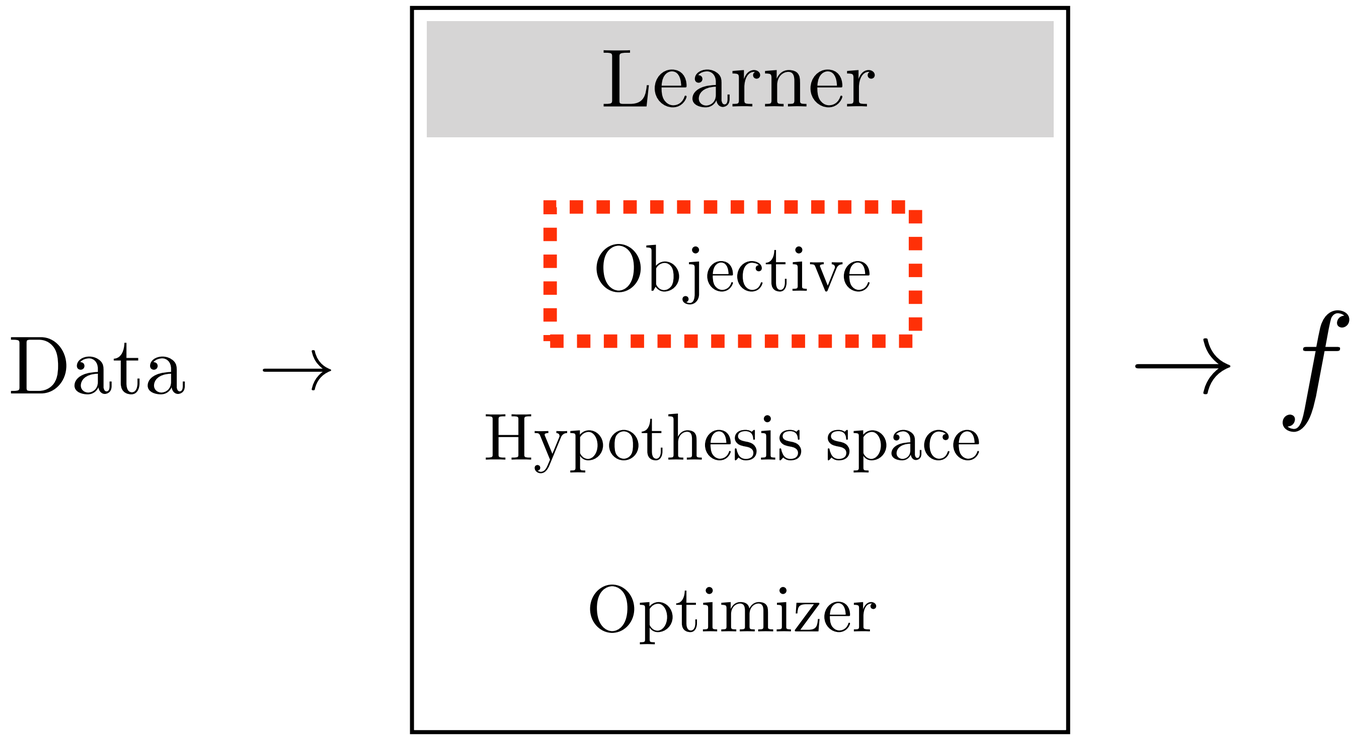


Rather than penalizing if output *image* looks fake, penalize if each overlapping *patch* in output looks fake

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

A structured objective is all you need

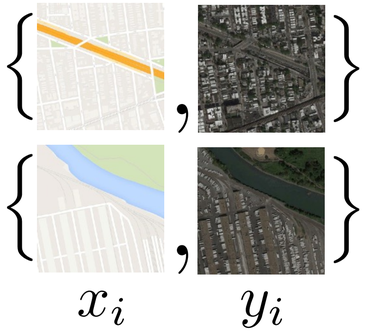
Marr, Poggio



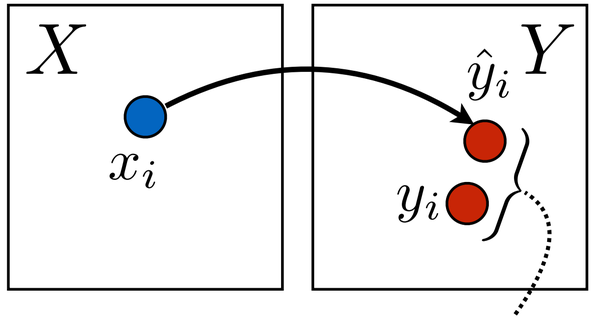
All the other considerations — data, compute, hypothesis space, optimizer — can simply be “scaled” ;)

Paired translation

Training data



Objective



regression error

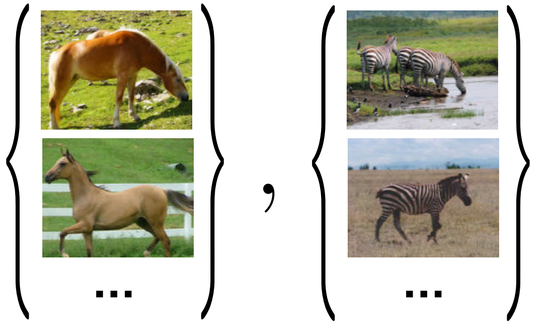
Input

Result

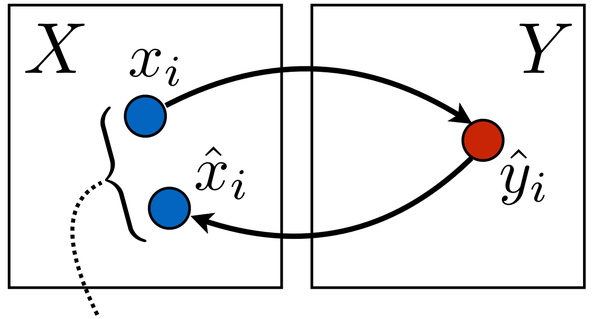
[“pix2pix”, Isola, Zhu, Zhou, Efros, 2017]

Unpaired translation

Training data



Objective



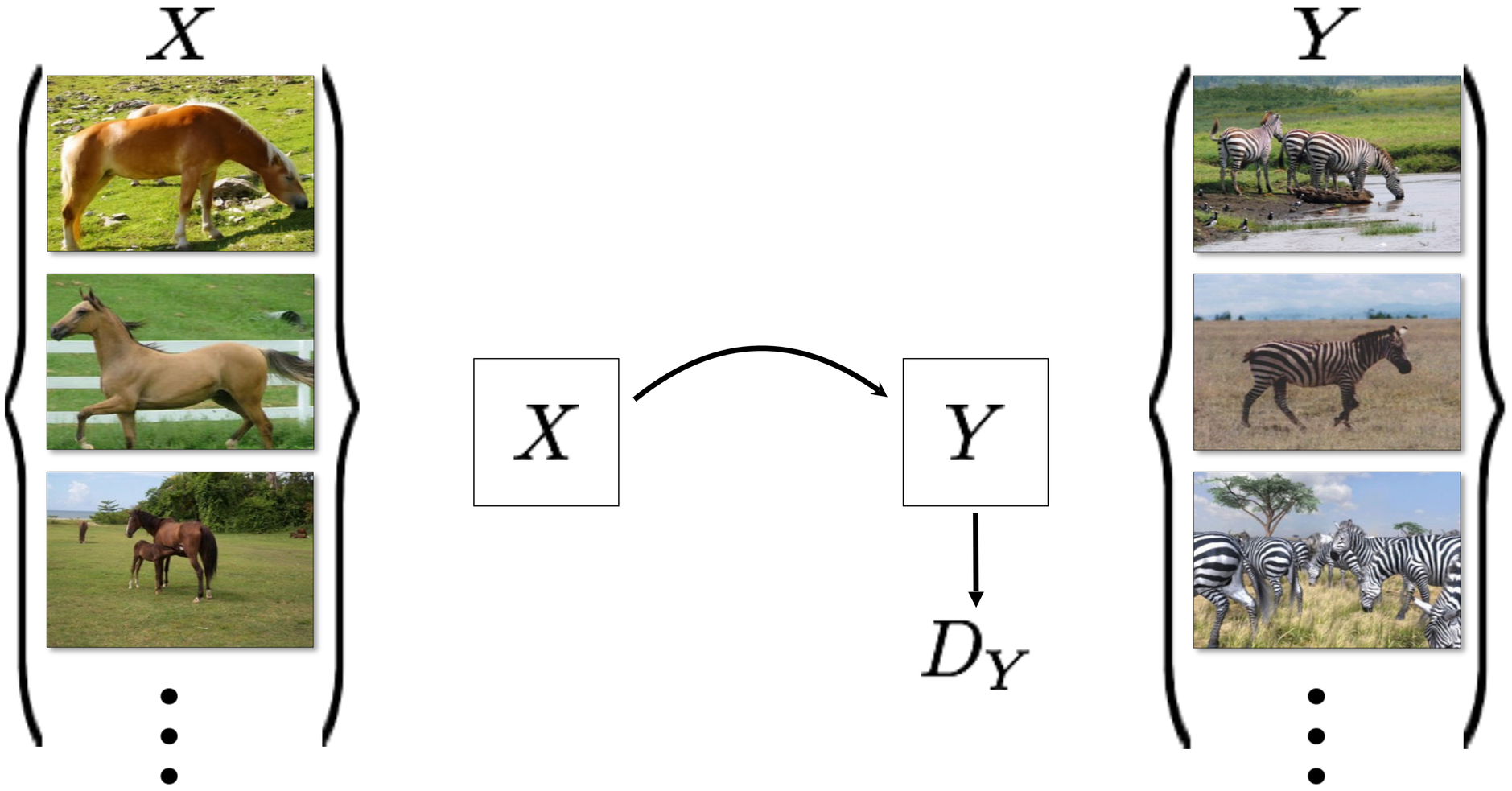
cycle-consistency error

Input

Result

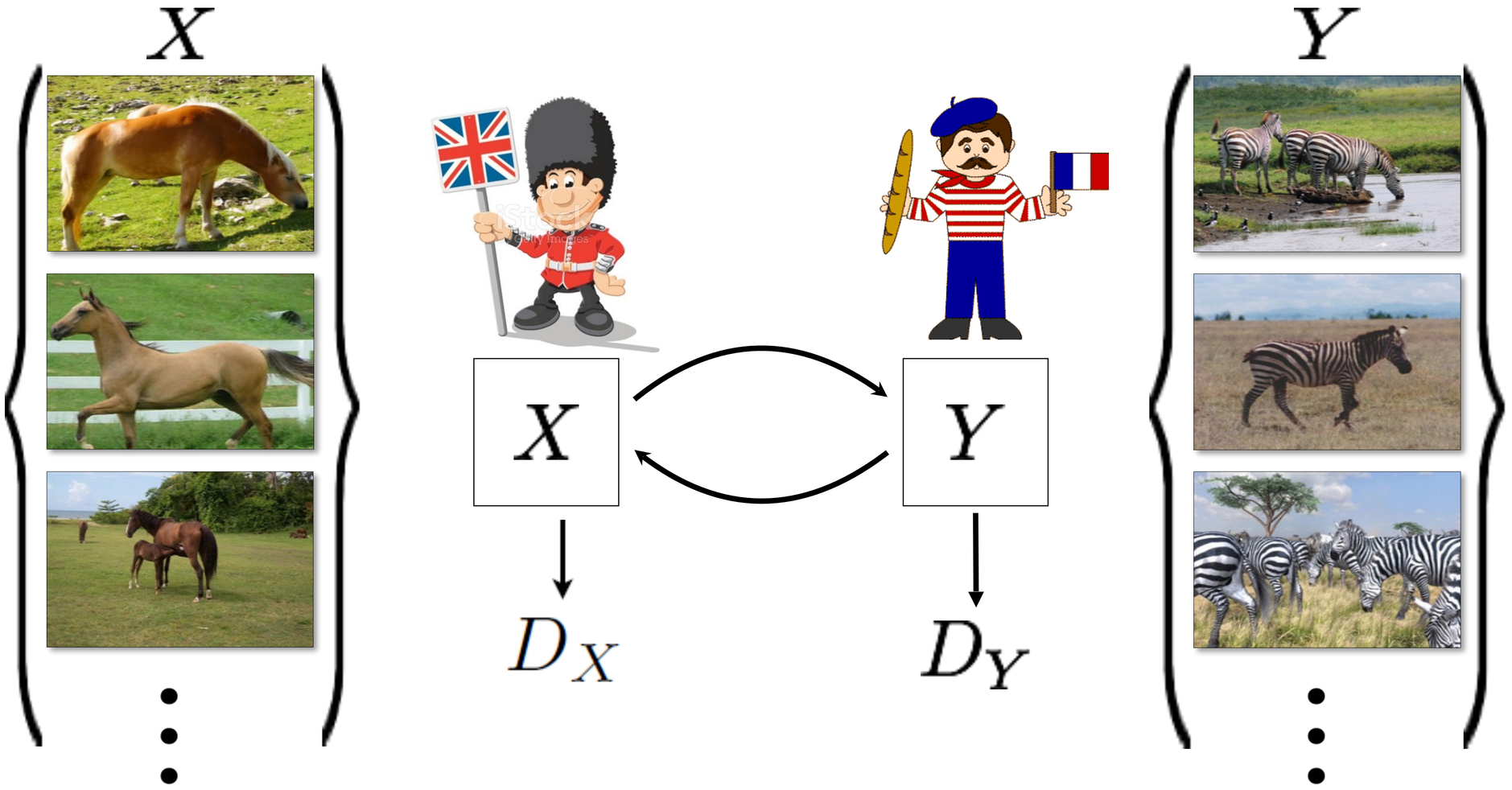
[“CycleGAN”, Zhu*, Park*, Isola, Efros, 2017]

Cycle-Consistent Adversarial Networks

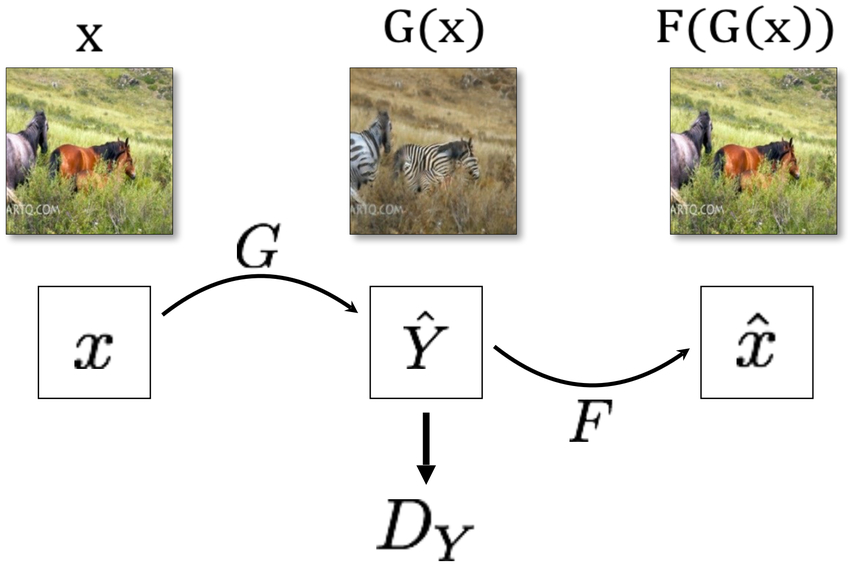


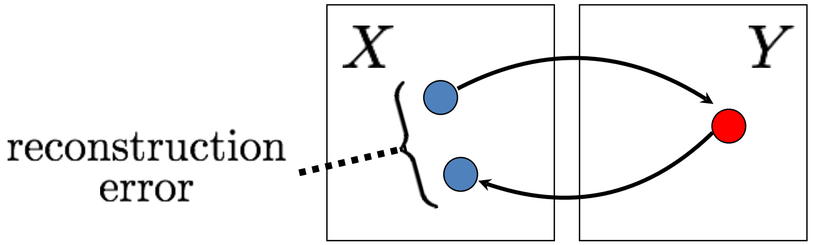
[Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

Cycle-Consistent Adversarial Networks



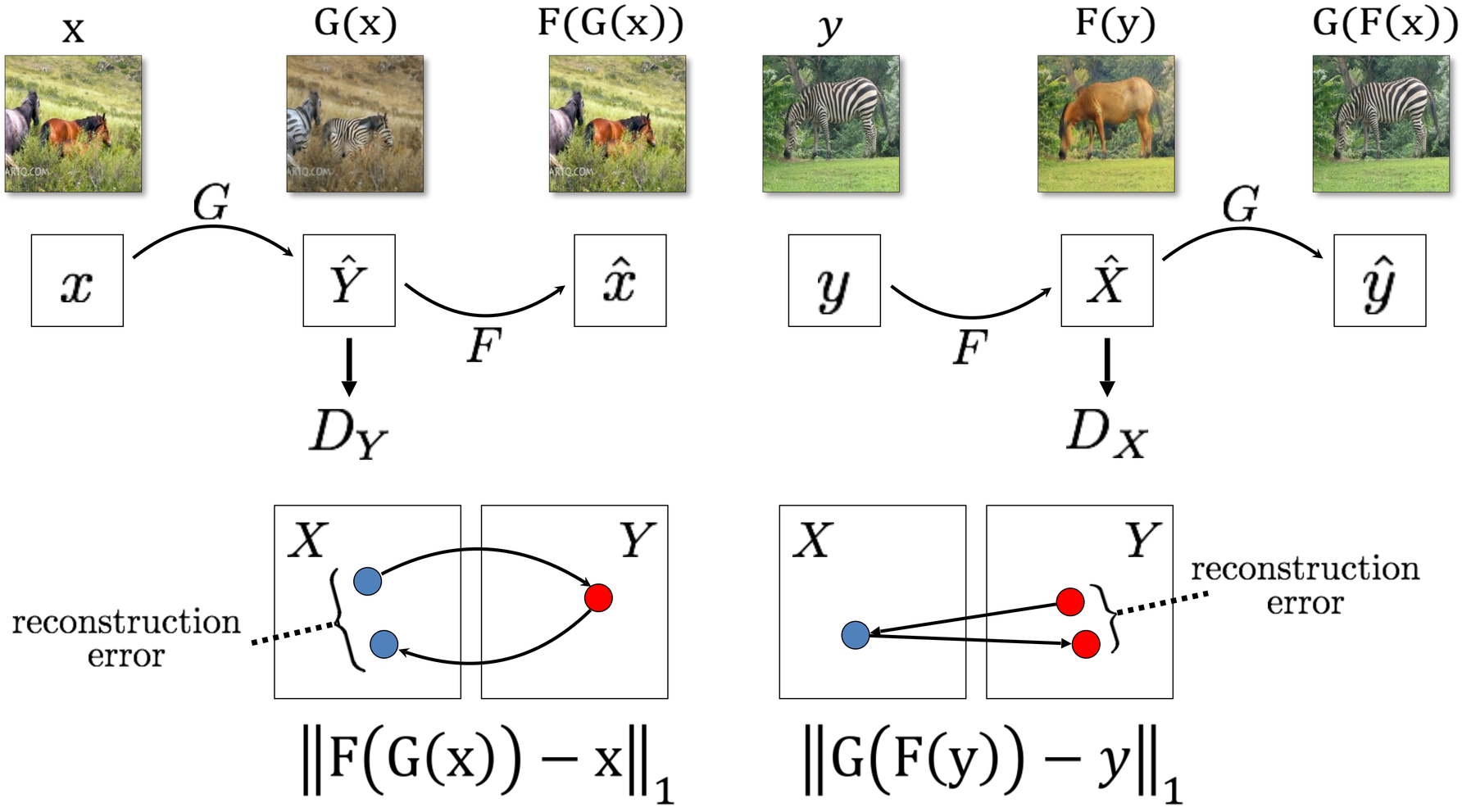
Cycle Consistency Loss





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

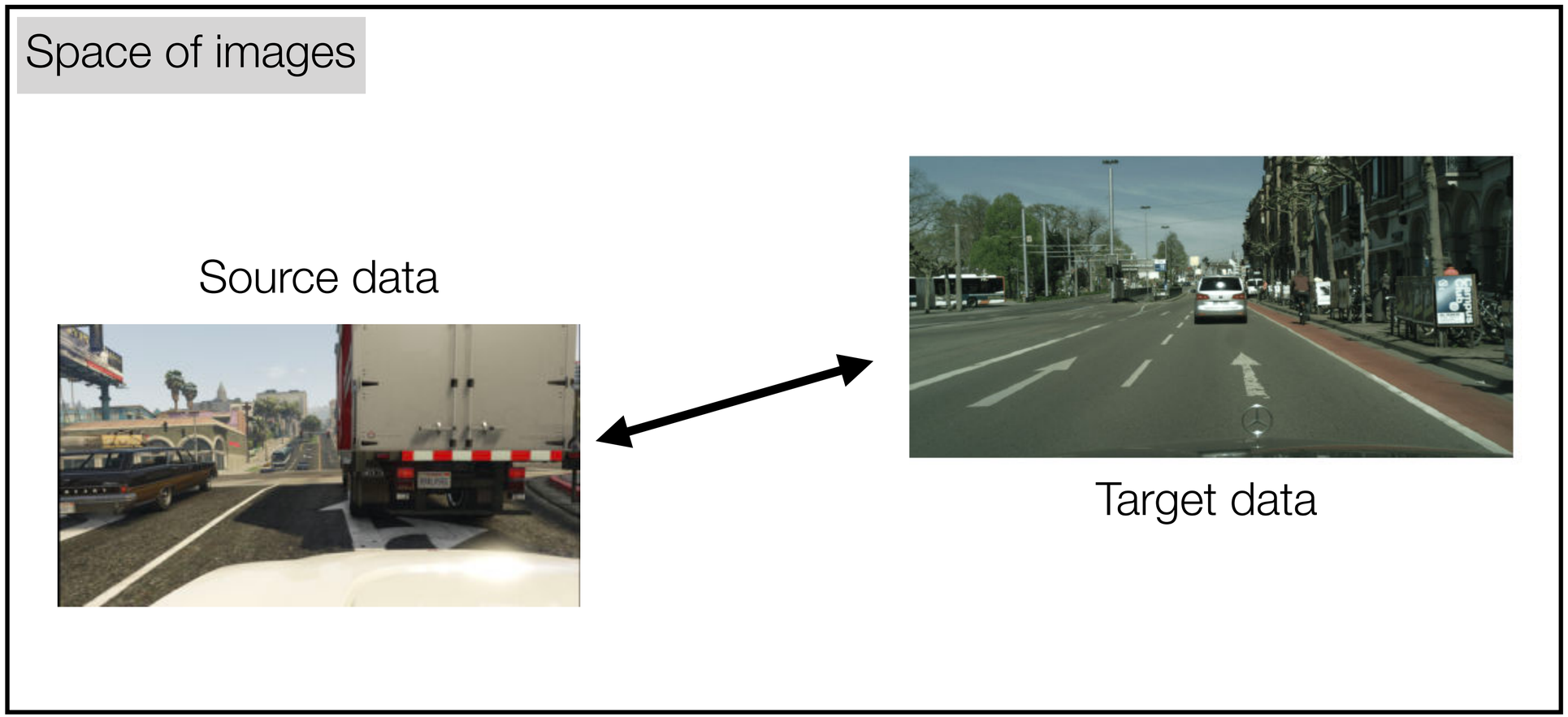
Cycle Consistency Loss



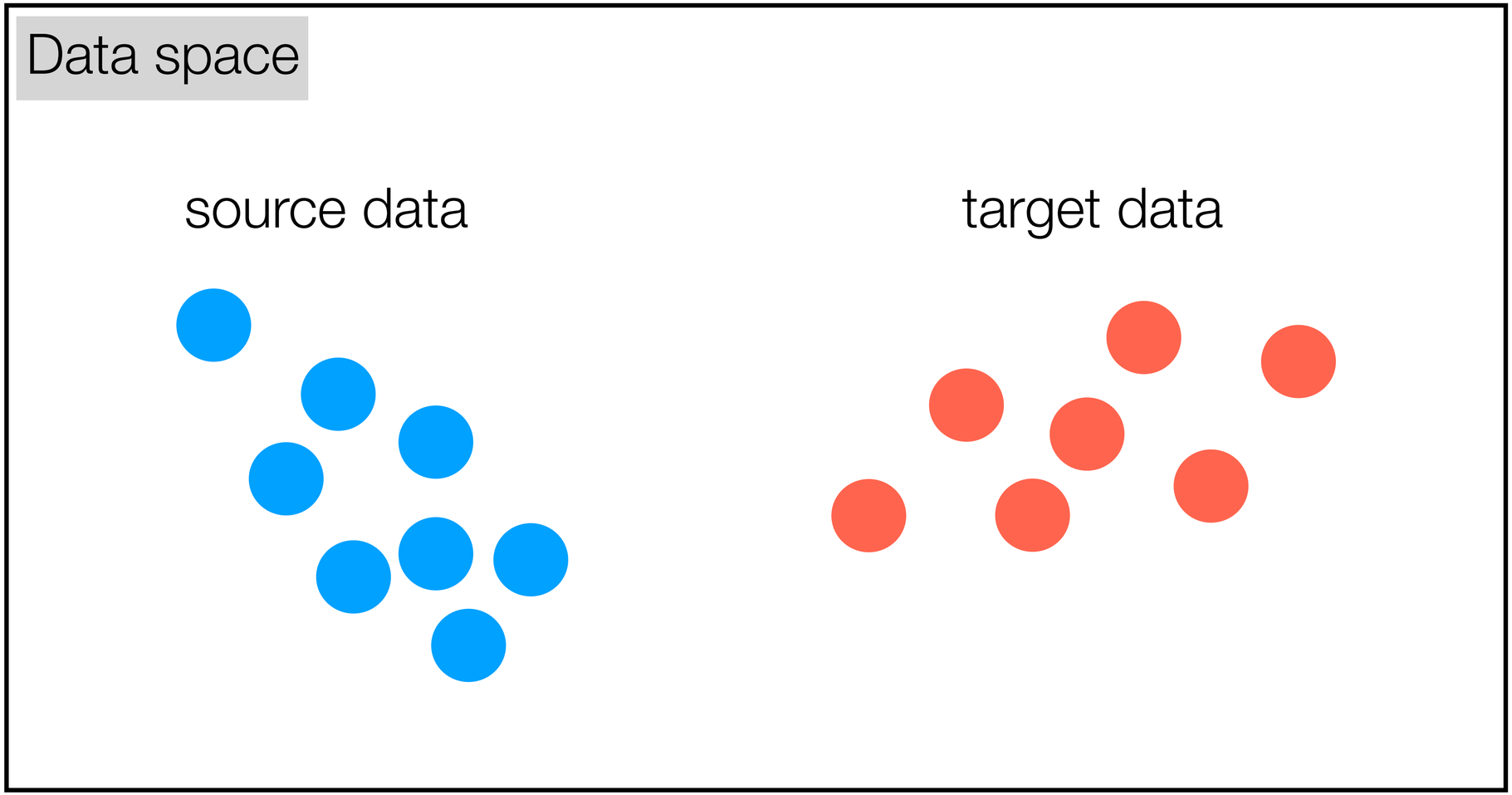
source domain

target domain
(where we actual use our model)

Domain gap between $p_{\texttt{source}}$ and $p_{\texttt{target}}$ will cause us to fail to generalize.



Idea #1: transform the source domain to look like the target domain



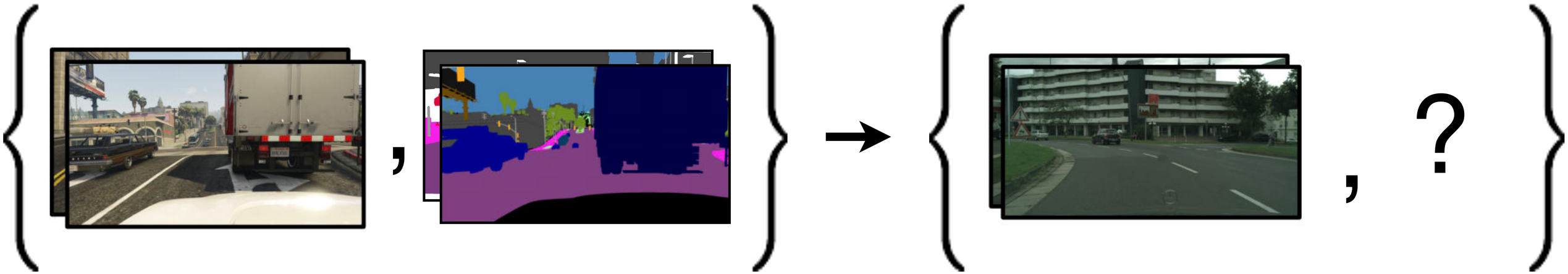
(Or vice versa)

This is called **domain adaptation** or **calibration**

CyCADA: Cycle-Consistent Adversarial Domain Adaptation

Source domain

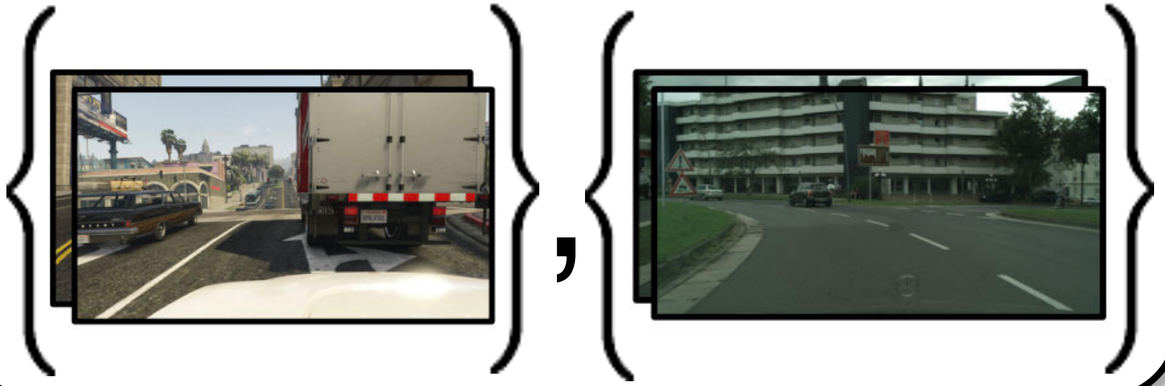
Target domain



[Hoffman, Tzeng, Park, Zhu, Isola, Saenko, Darrell, Efros, arXiv 2017]

CycleGAN

Training data



CycleGAN

Training data

,

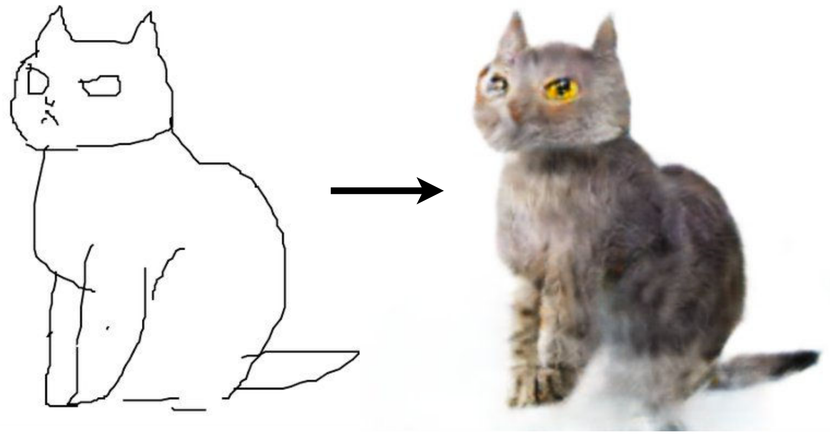
CycleGAN

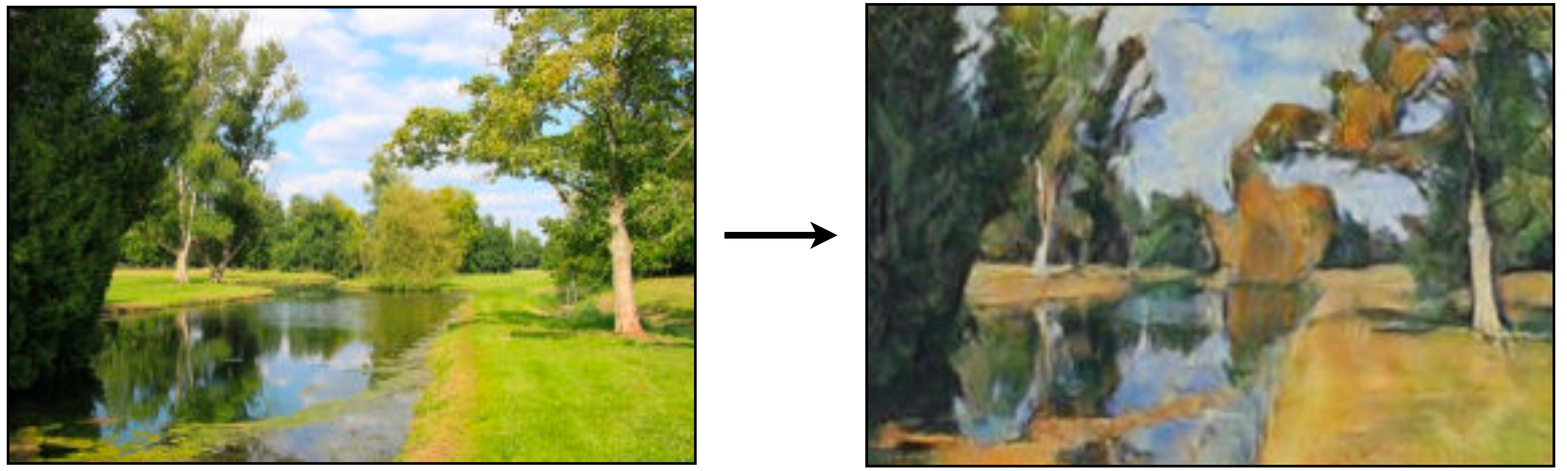
FCN

Training data

,

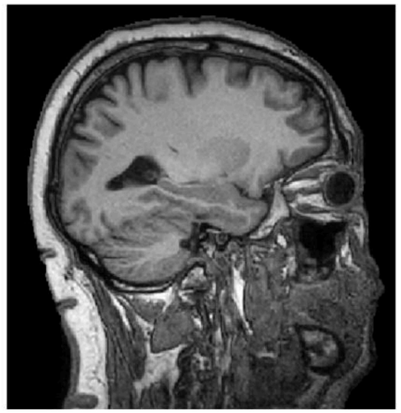
What would it look like if…?



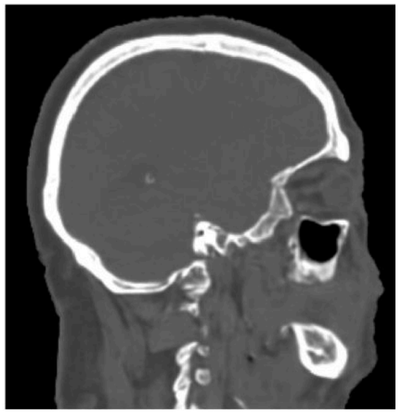


What would it look like if…?

MRI



CT



[Wolterink et al, 2017]

Sim

“Real”

[Hoffman et al, 2018]

Thank you!