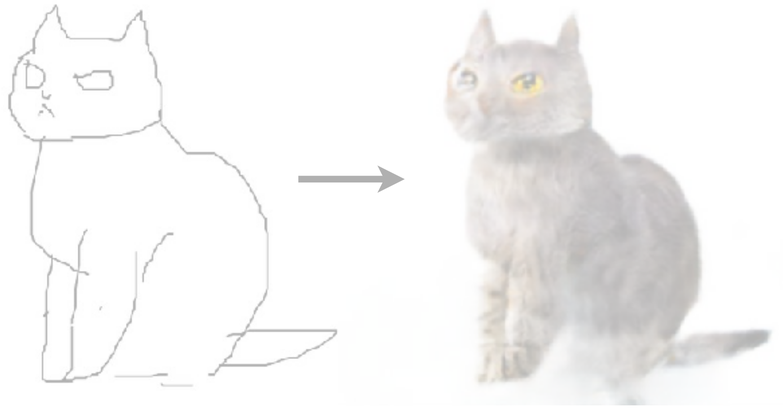


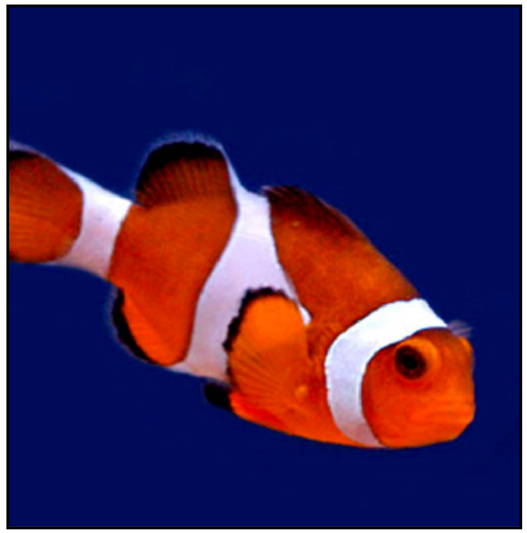
Lecture 22

Image Synthesis

22. Image Synthesis

- Image synthesis
  - Variational Autoencoders
  - Generative Adversarial Networks
- Structured prediction
  - Image-to-image GANs
- Domain mapping

Image classification



Classifier

“Fish”

image **x**

label y

Image classification

Classifier

“Fish”

image **x**

label y

Image classification

Classifier

“Fish”

image **x**

label y

Image classification

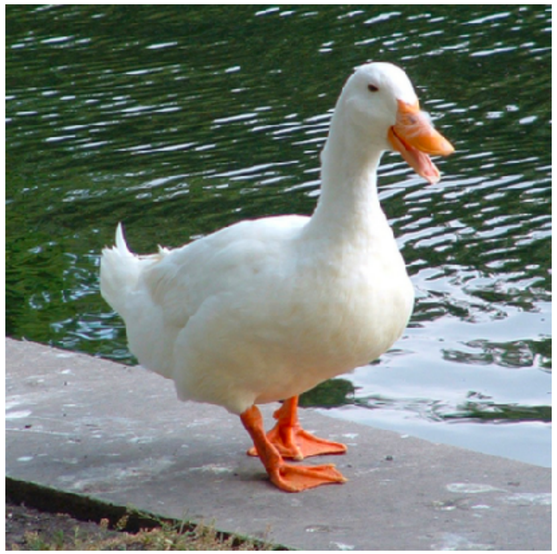


image **x**

Classifier

"Duck"

label y

Image synthesis

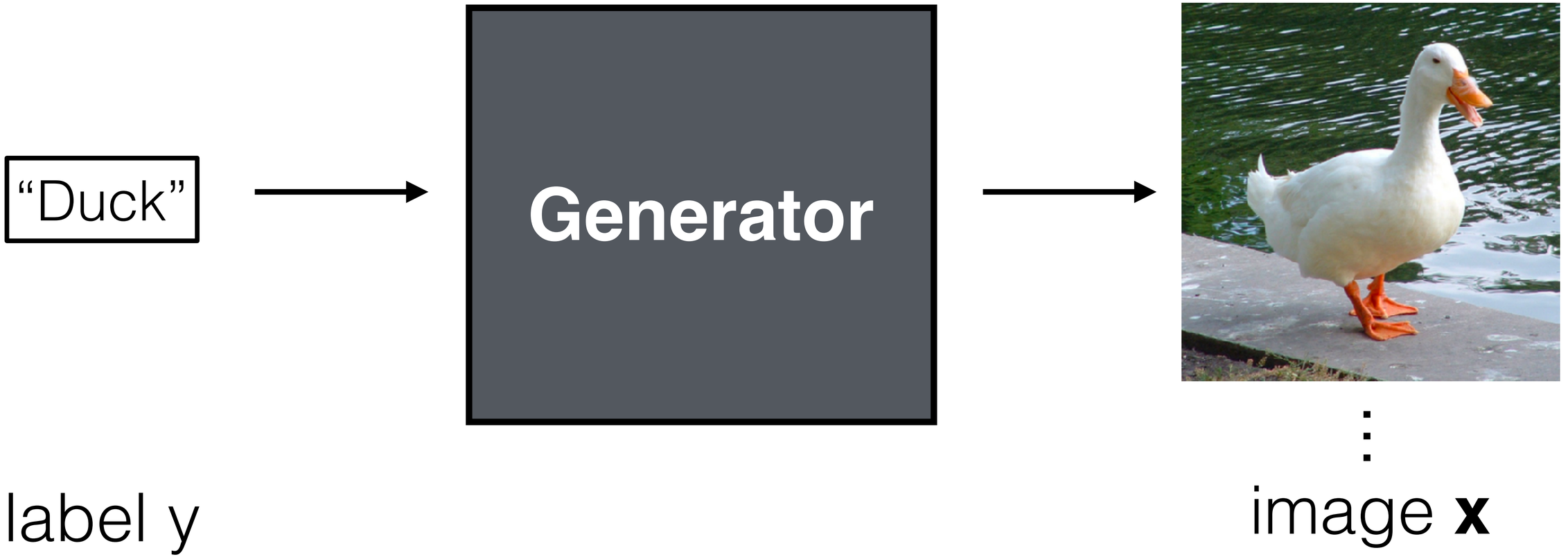
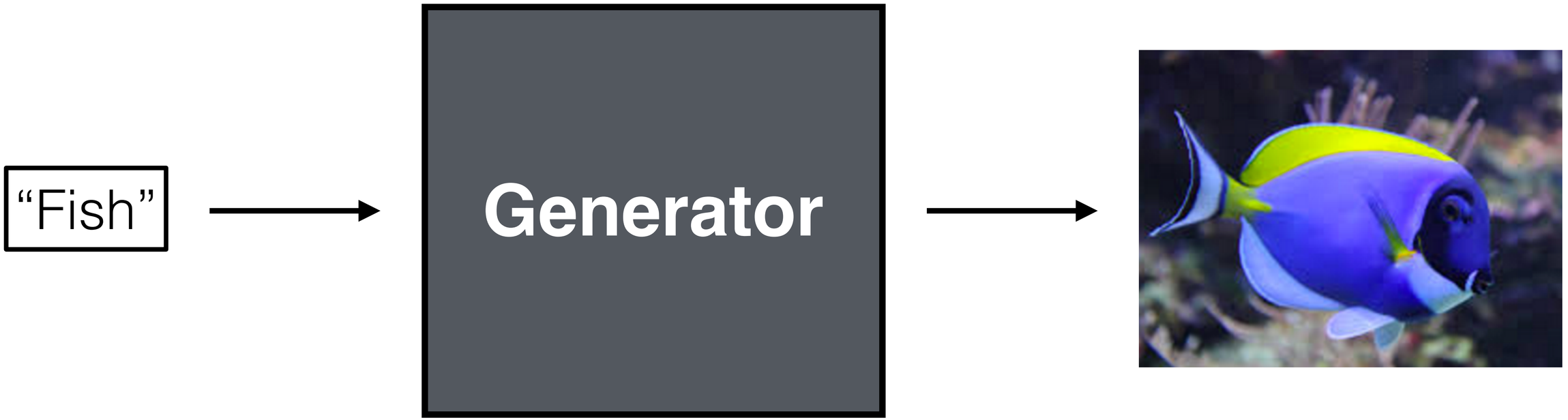


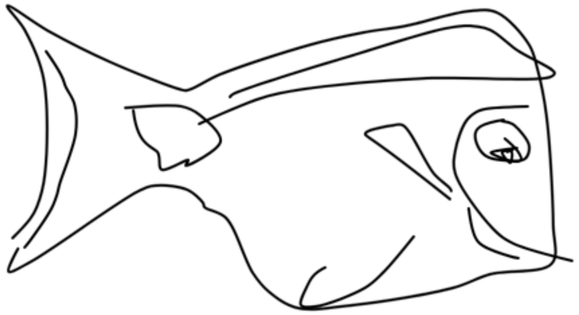
Image synthesis

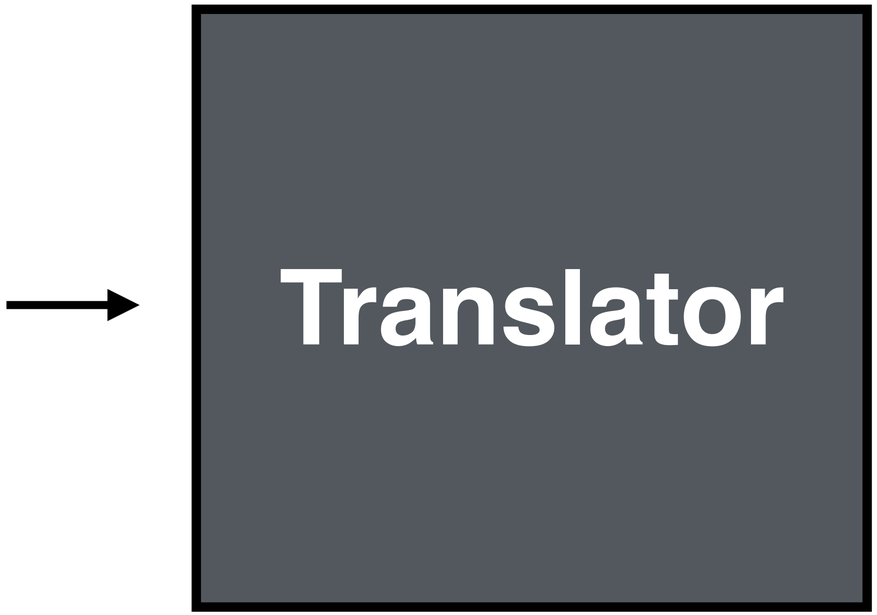


label y

image **x**

Image translation

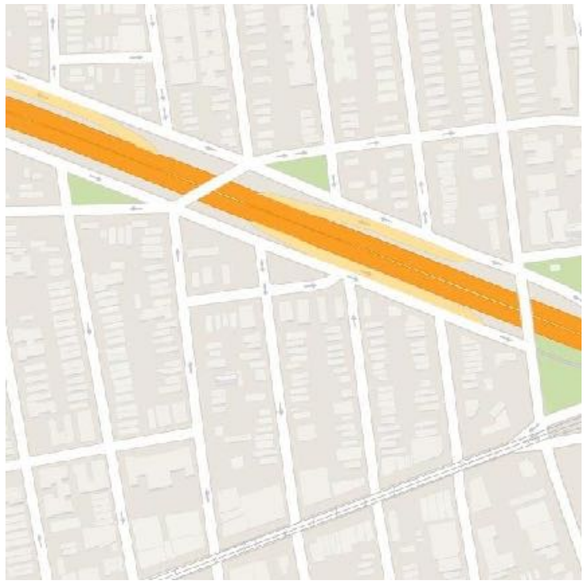




User sketch

Photo

Image translation



Translator

Google Map

Satellite photo

Image translation

BW image

Translator

Color image

Image synthesis via **generative modeling**

X is high-dimensional!

Model of high-dimensional structured data $P(\mathbf{X}|\mathbf{Y}=\mathbf{y})$

In vision, this is usually what we are interested in!

What can you do with generative models?

1. Image synthesis
2. Structured prediction
3. Domain mapping
4. (Representation learning)
5. (Model-based intelligence)

1. **Image synthesis**
2. Structured prediction
3. Domain mapping

[Images: https://ganbreeder.app/]

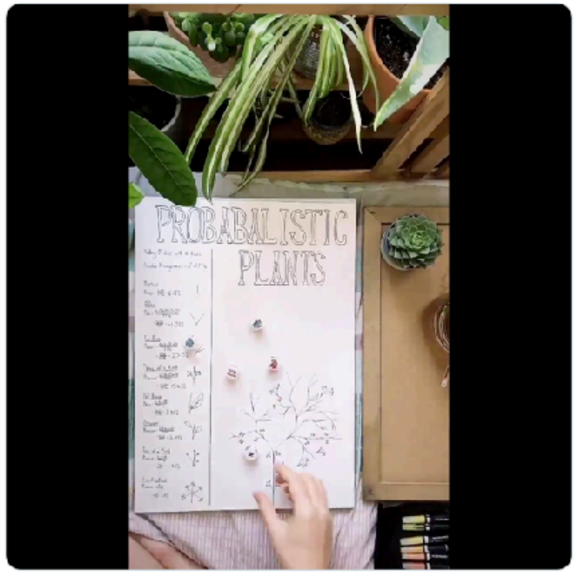
Image synthesis

Procedural graphics

[Anders Scheil]

Ayliean @Ayliean · Nov 17

Made up a set of rules and rolled some dice to decide how this plant would grow. I never did get that five of a kind, as expected, but I was still hopeful! 🌱🍀🌿

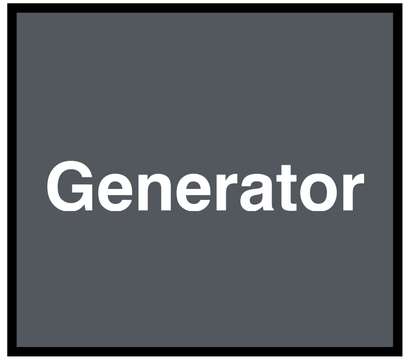


52 1.1K 4.5K

Image synthesis from “noise”

G

$z \sim p(z)$



$x = G(z)$

Sampler

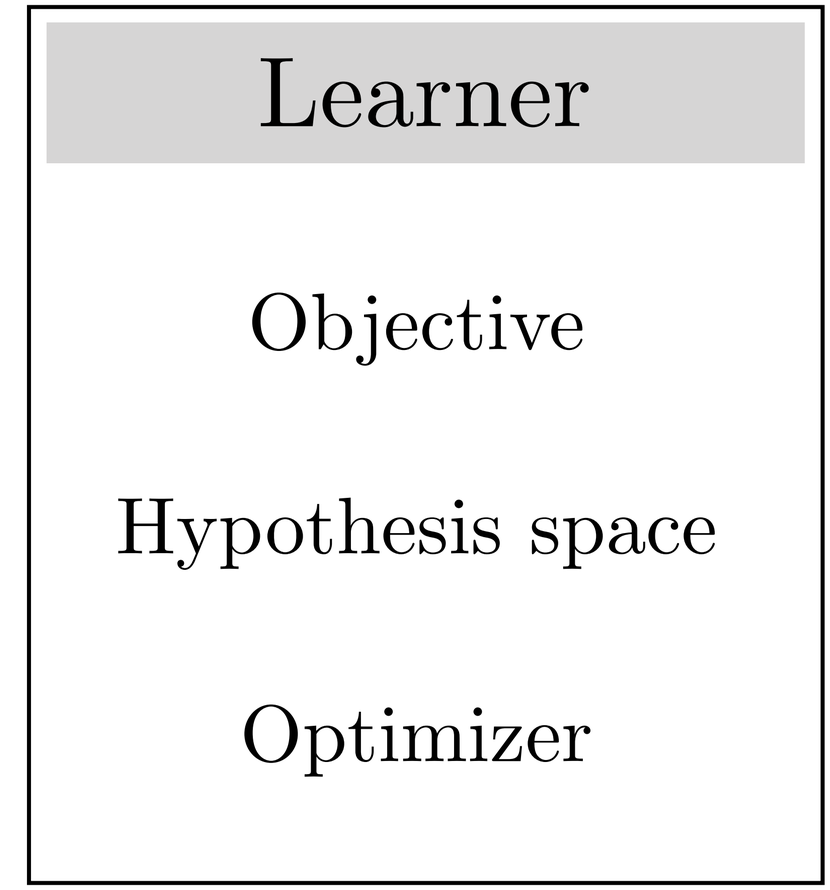
$$G : \mathcal{Z} \rightarrow \mathcal{X}$$

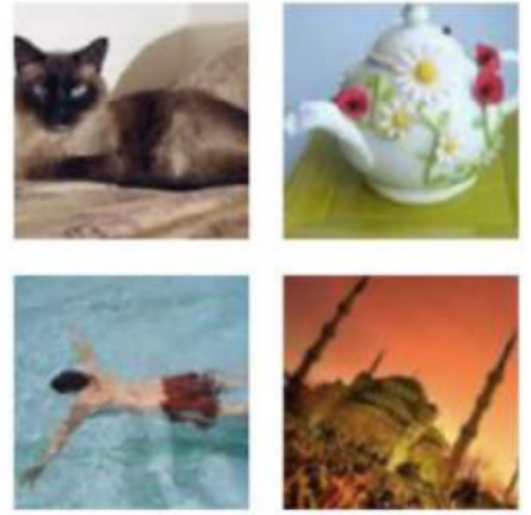
$$z \sim p(z)$$

$$x = G(z)$$

Learning a generative model

Input samples

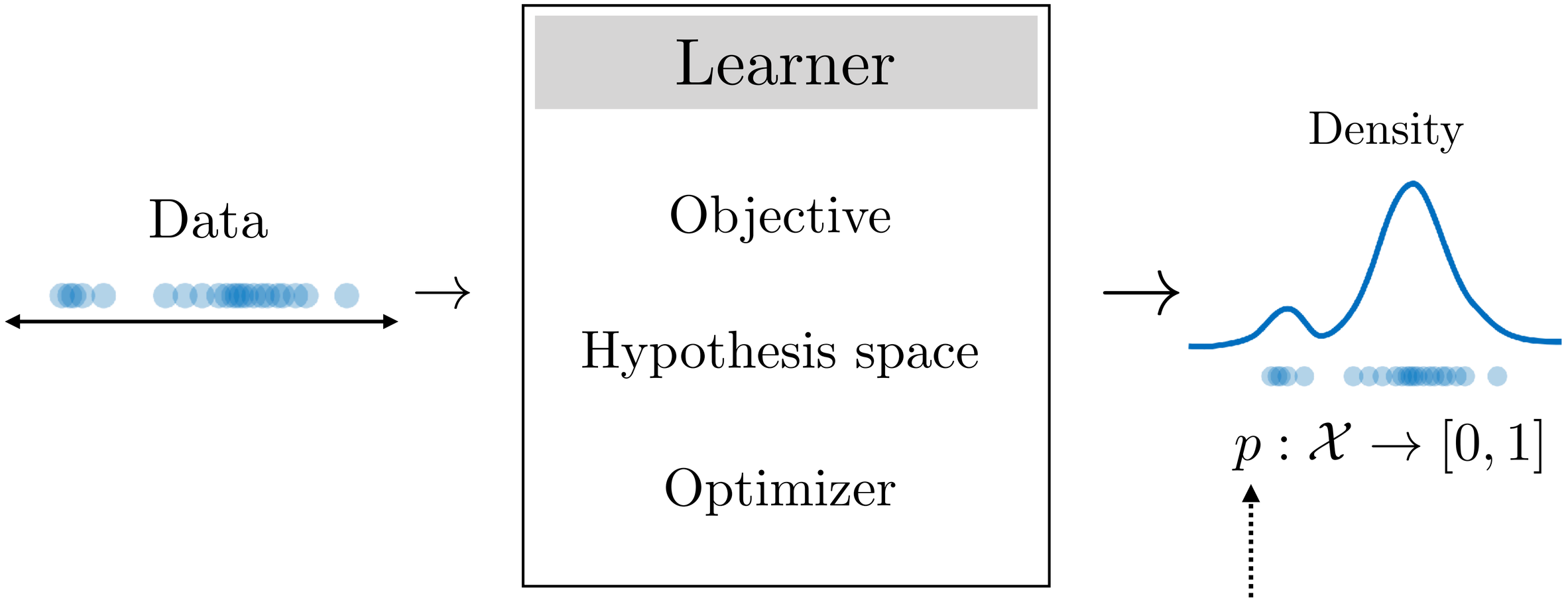




Generated samples

[figs modified from: http://introtodeeplearning.com/materials/2019_6S191_L4.pdf]

Learning a density model



[figs modified from: http://introtodeeplearning.com/materials/2019_6S191_L4.pdf]

Case study #1: Fitting a Gaussian to data

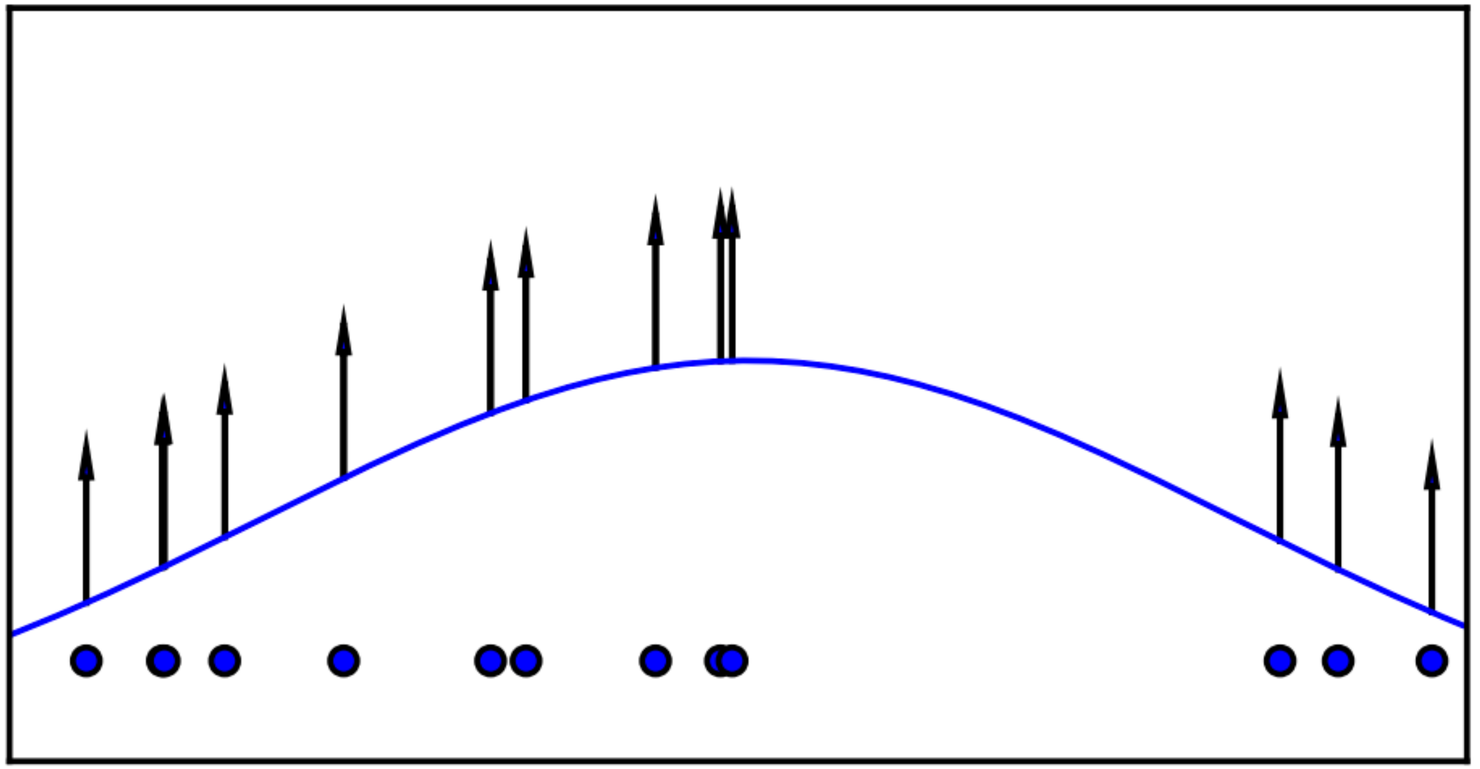


fig from [Goodfellow, 2016]

Max likelihood objective

$$\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} [\log p_\theta(x)]$$

Considering only Gaussian fits

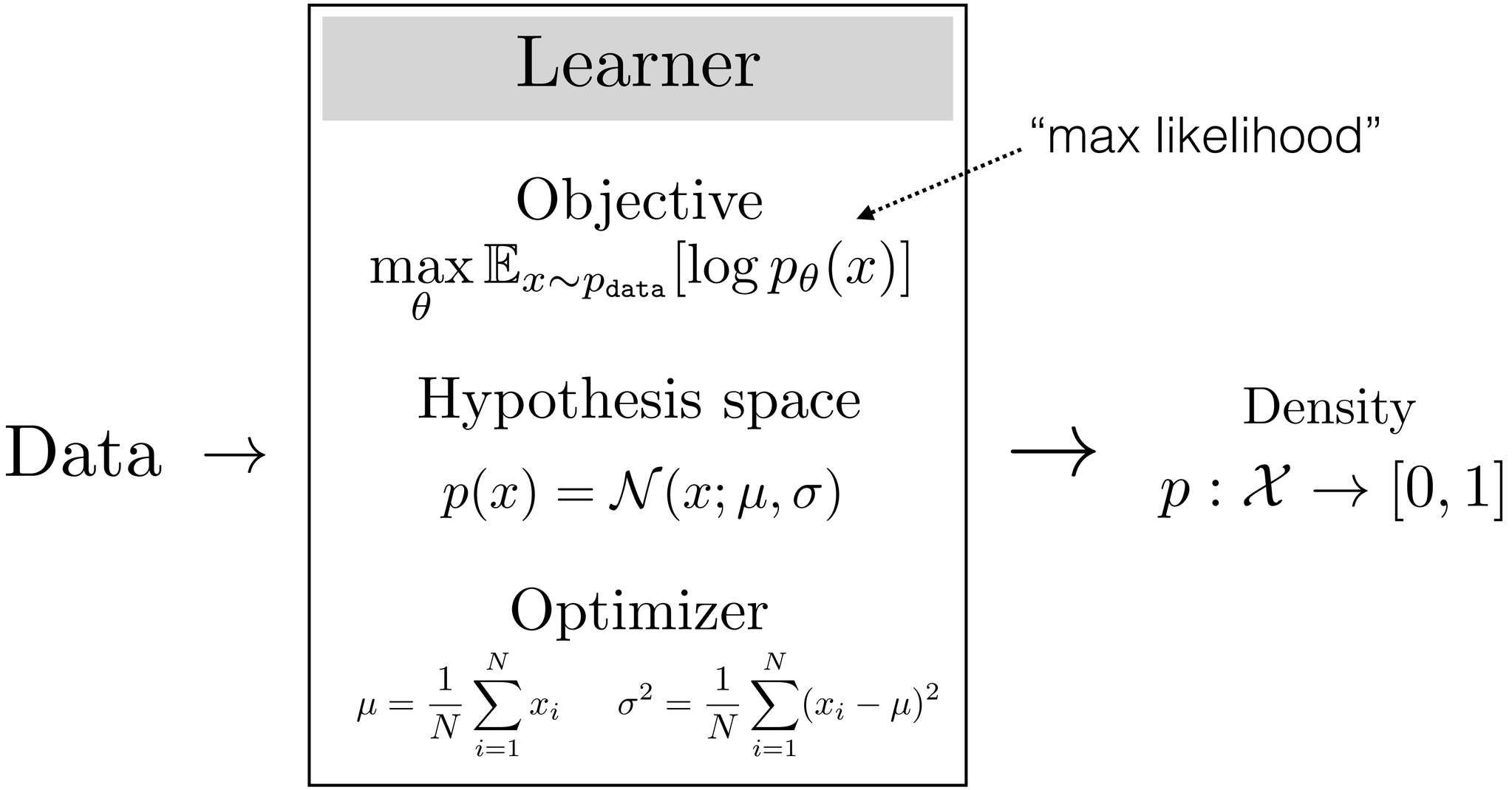
$$p_\theta(x) = \mathcal{N}(x; \mu, \sigma)$$

$$\theta = [\mu, \sigma]$$

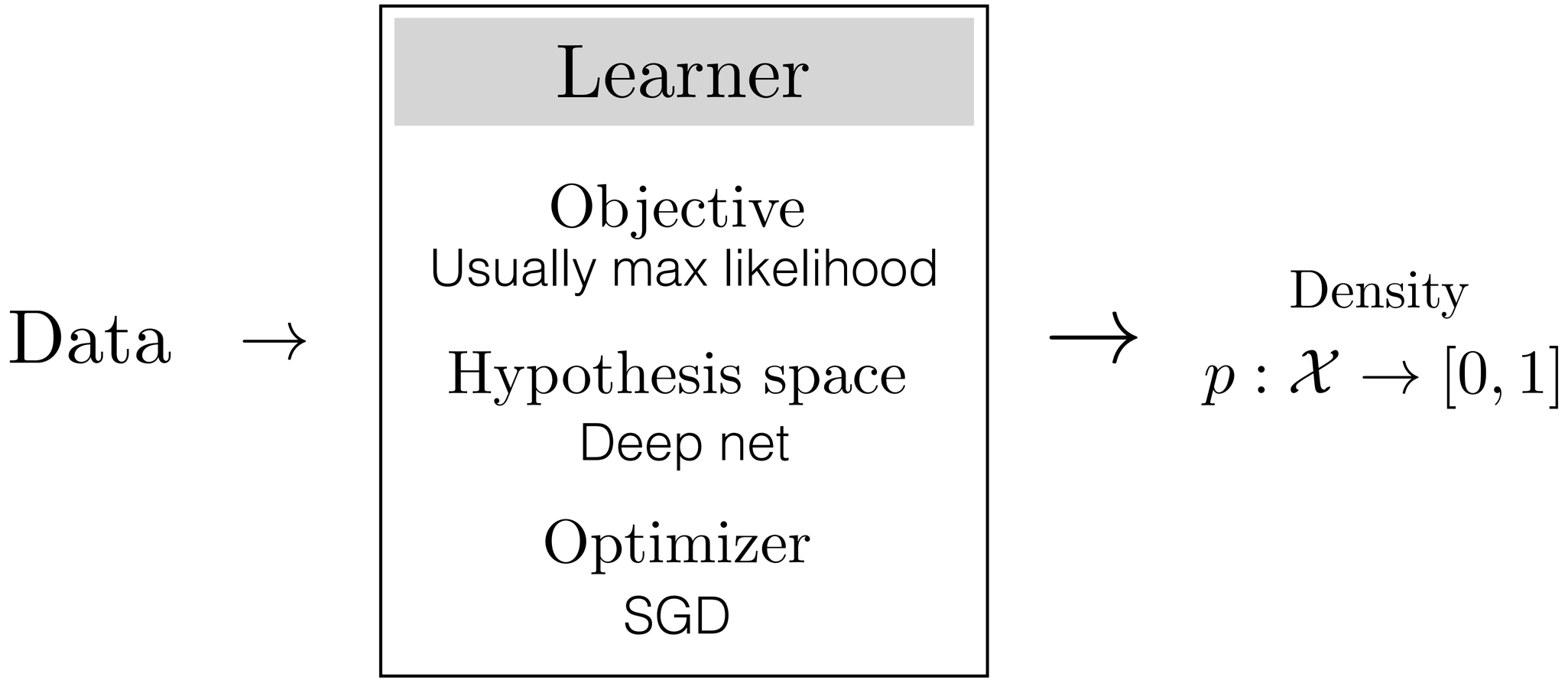
Closed form optimum:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2$$

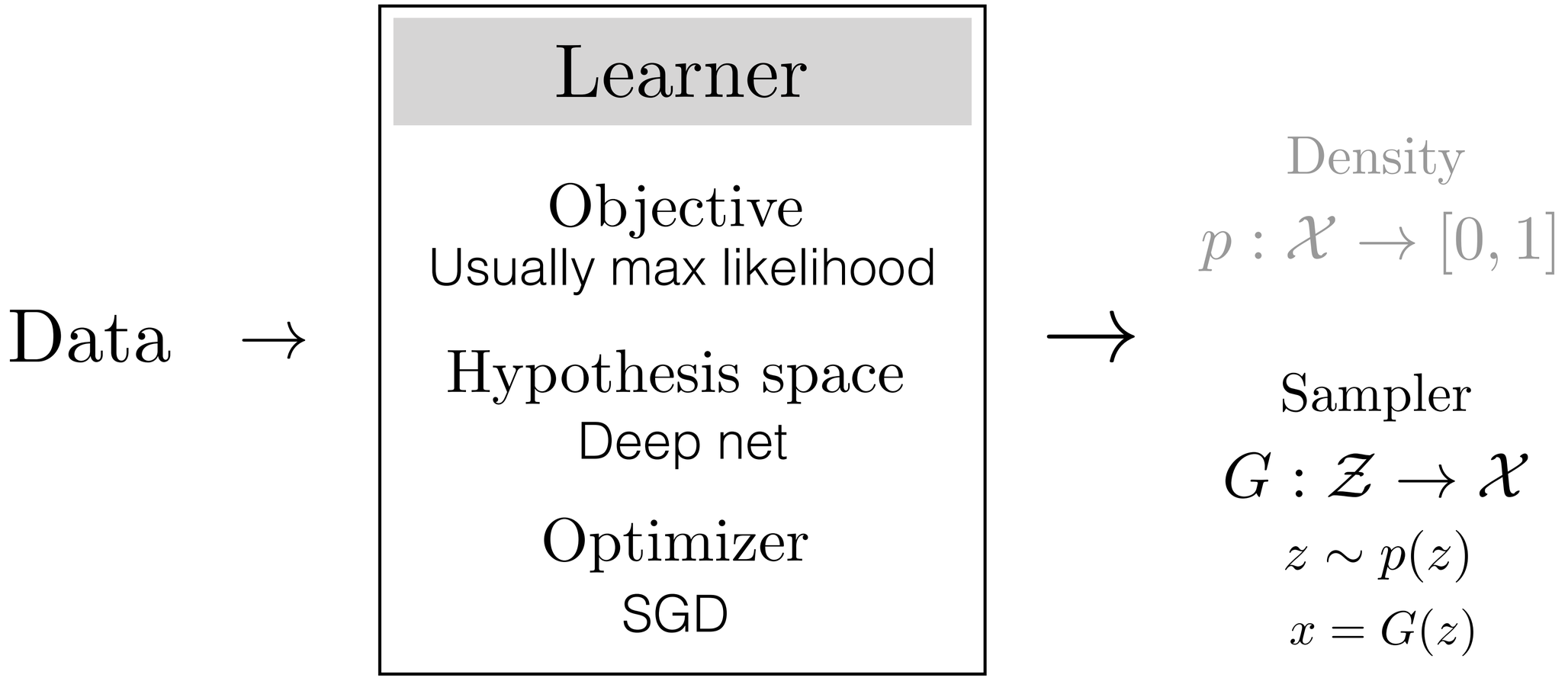
Case study #1: Fitting a Gaussian to data



Case study #2: learning a deep generative model

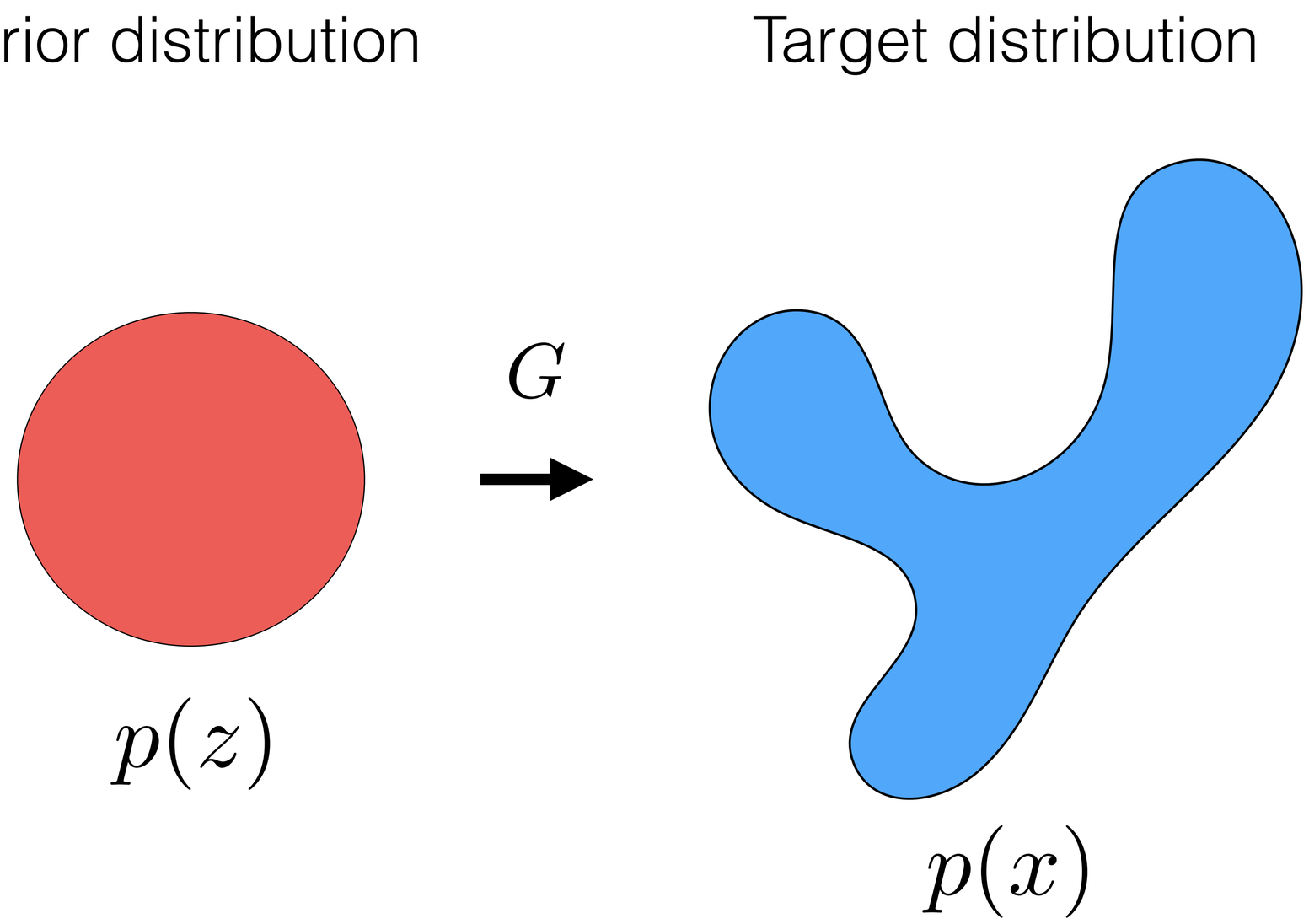


Case study #2: learning a deep generative model

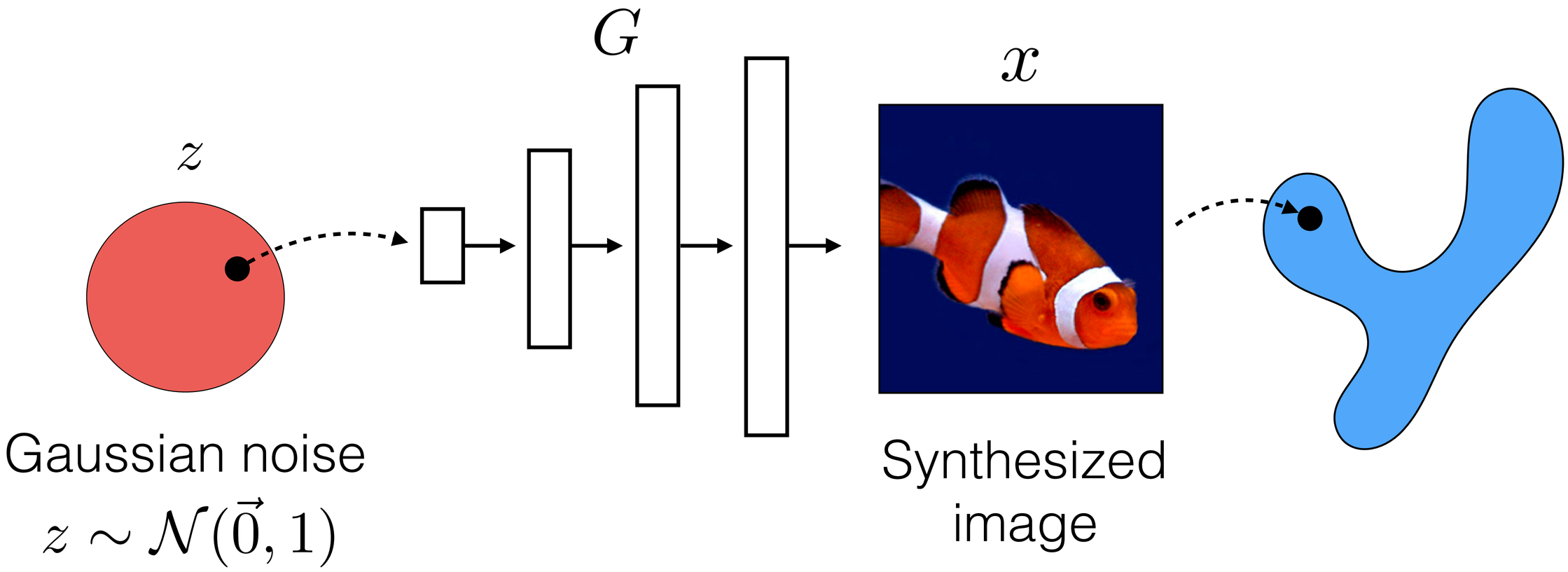


Models that provide a sampler but no density are called **implicit generative models**

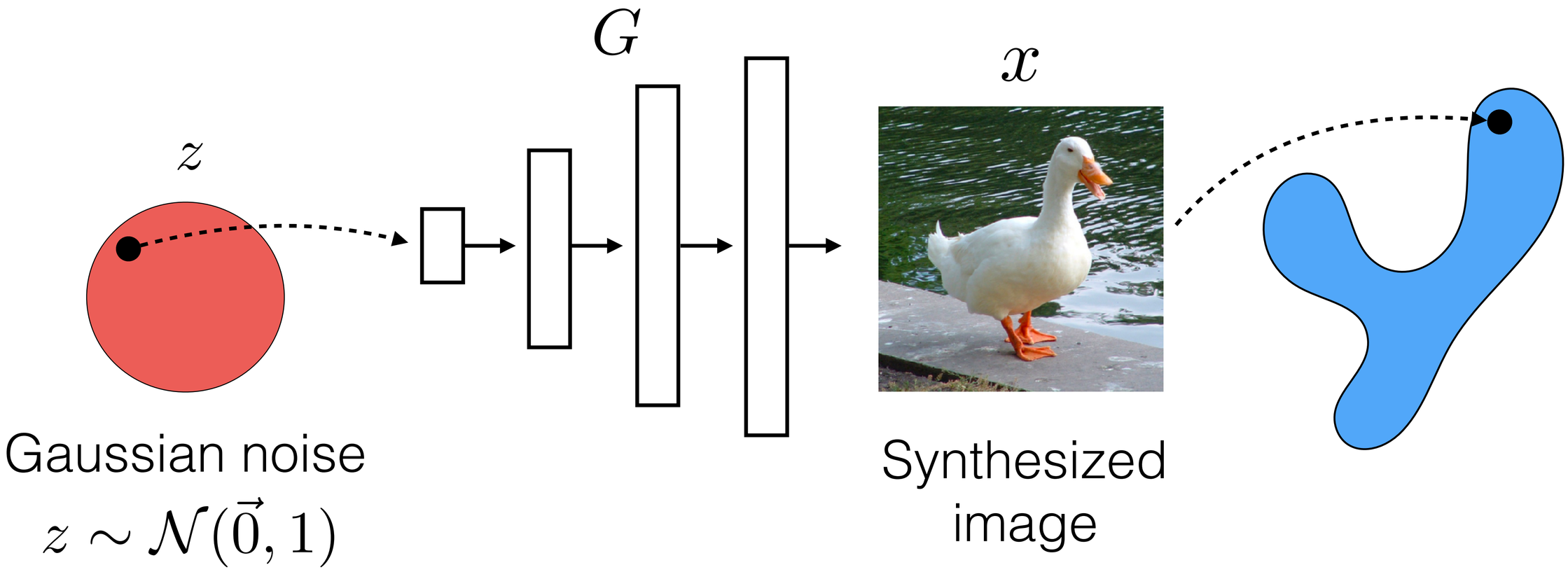
Deep generative models are distribution transformers



Deep generative models are distribution transformers



Deep generative models are distribution transformers

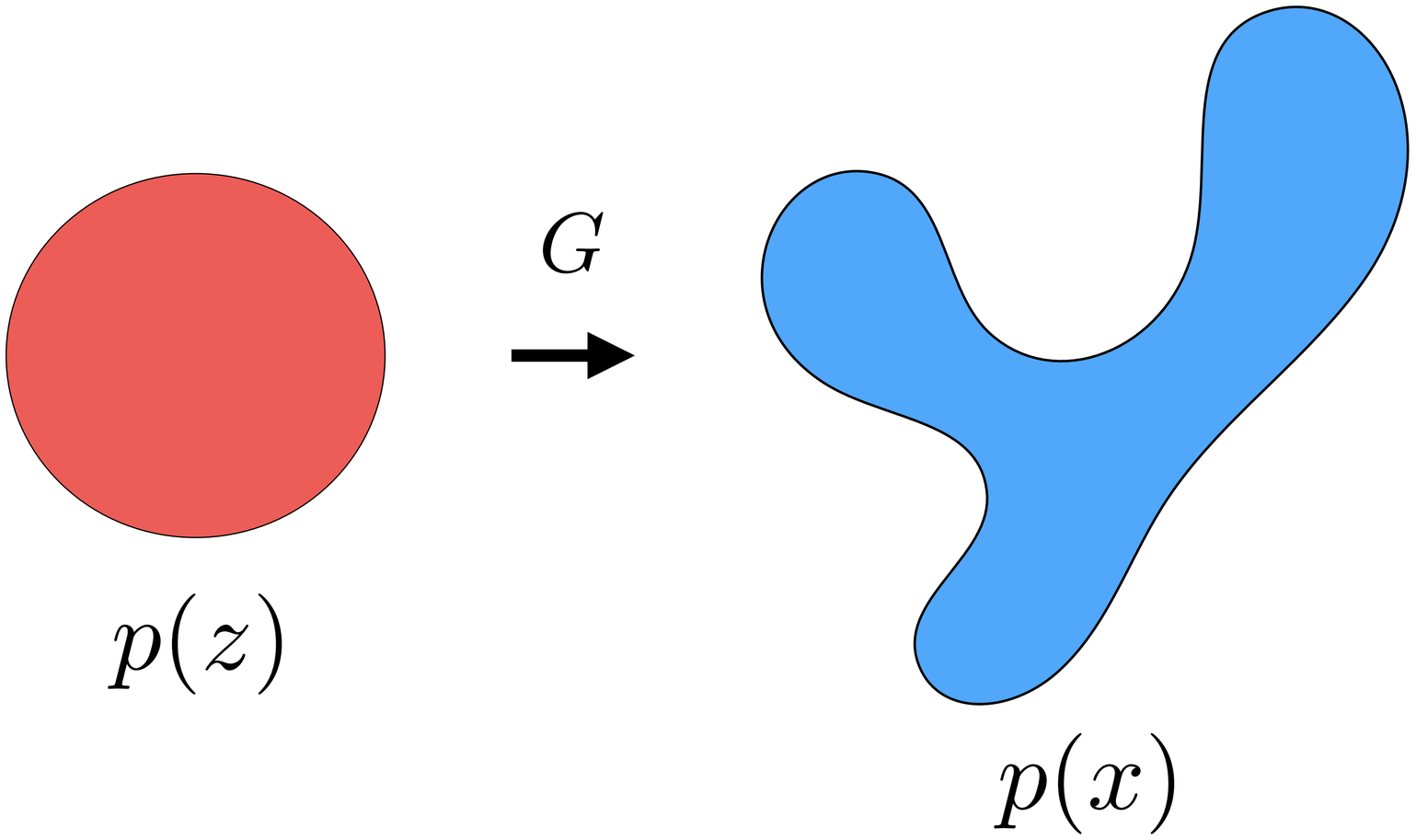


Variational Autoencoders (VAEs)

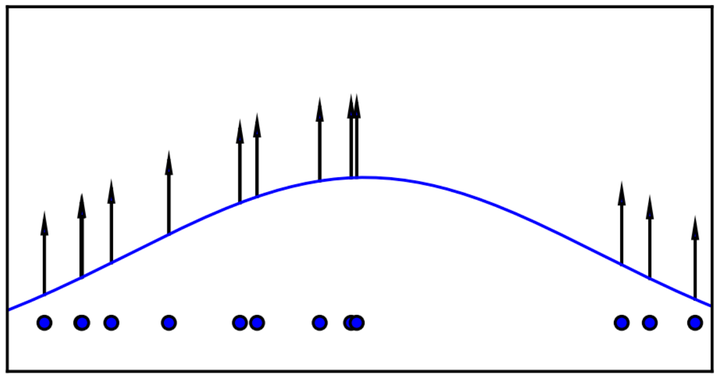
[Kingma & Welling, 2014; Rezende, Mohamed, Wierstra 2014]

Prior distribution

Target distribution

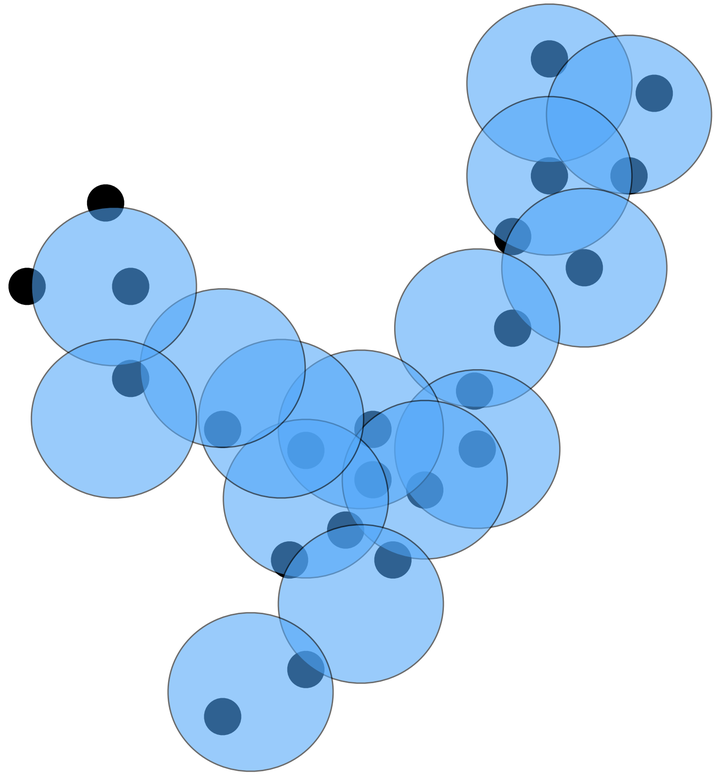


Mixture of Gaussians



$$p_\theta(x) = \sum_{i=1}^{k} w_i \mathcal{N}(x; u_i, \Sigma_i)$$

Target distribution



$$x \sim p_{\texttt{data}}(x)$$

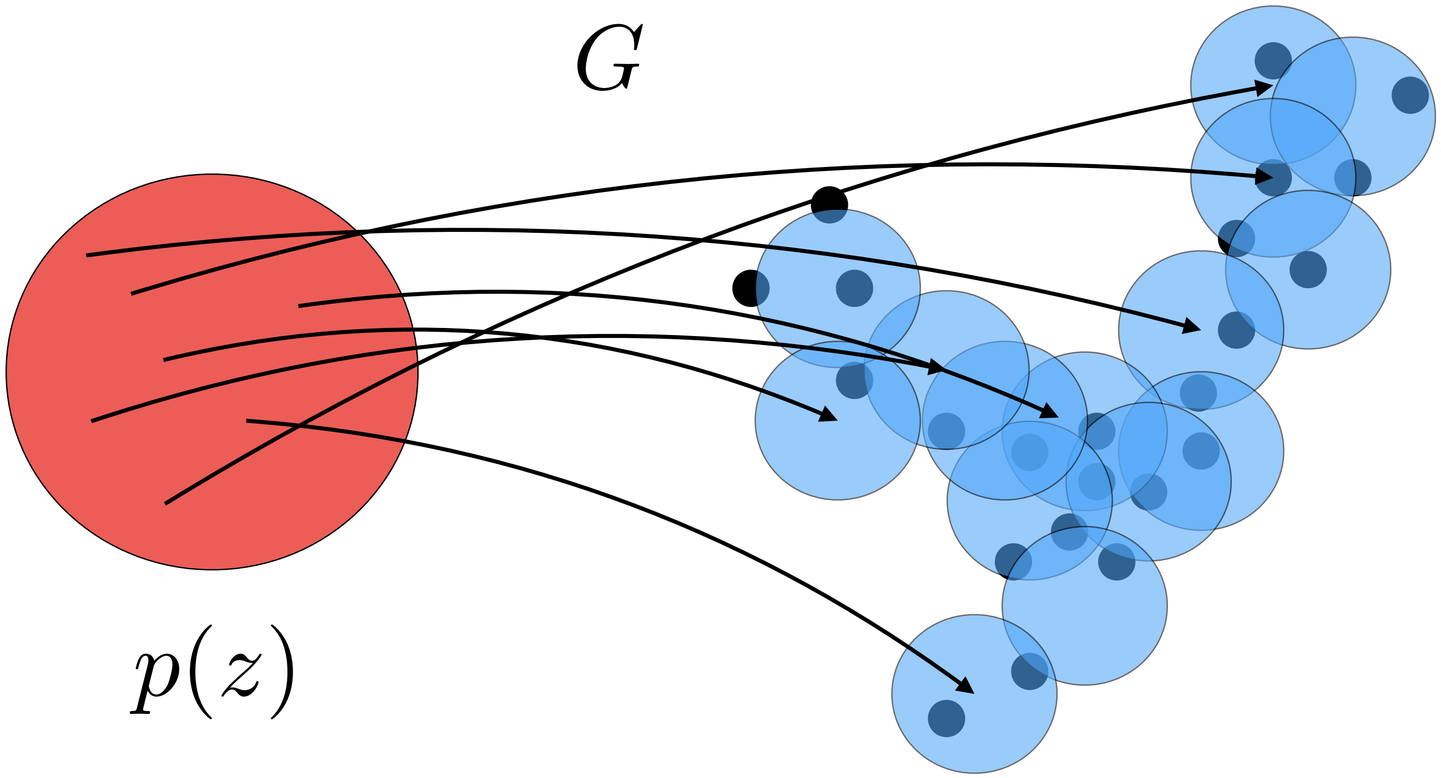
$$p_\theta(x)$$

Variational Autoencoders (VAEs)

[Kingma & Welling, 2014; Rezende, Mohamed, Wierstra 2014]

Prior distribution

Target distribution



Density model:

$$p_\theta(x) = \int p(x|z;\theta)p(z)dz$$

$$p(x|z;\theta) \sim \mathcal{N}(x; G^\mu_\theta(x), G^\sigma_\theta(x))$$

Sampling:

$$z \sim p(z) \quad \epsilon \sim \mathcal{N}(0,1)$$

$$x = G^\mu_\theta(z) + G^\sigma_\theta(z)\epsilon$$

Variational Autoencoder (VAE)

Data $\rightarrow$

Learner

Objective

$$\max_\theta \mathbb{E}_{x \sim p_{\text{data}}}[\log p_\theta(x)]$$

Hypothesis space

~~$p_\theta(x) = \int p(x|z;\theta)p(z)dz$~~

$$x = G_\theta^\mu(z) + G_\theta^\sigma(z)\epsilon$$

$\rightarrow$

~~Density~~

~~$p_\theta : \mathcal{X} \rightarrow [0,1]$~~

Sampler

$$G_\theta : \mathcal{Z} \rightarrow \mathcal{X}$$

Variational Autoencoders (VAEs) — Training

Fitting a model to data requires computing $p_\theta(x)$

How to compute $p_\theta(x)$ efficiently?

$$p_\theta(x) = \int p(x|z;\theta)p(z)dz$$ ⟵ almost all terms are near zero

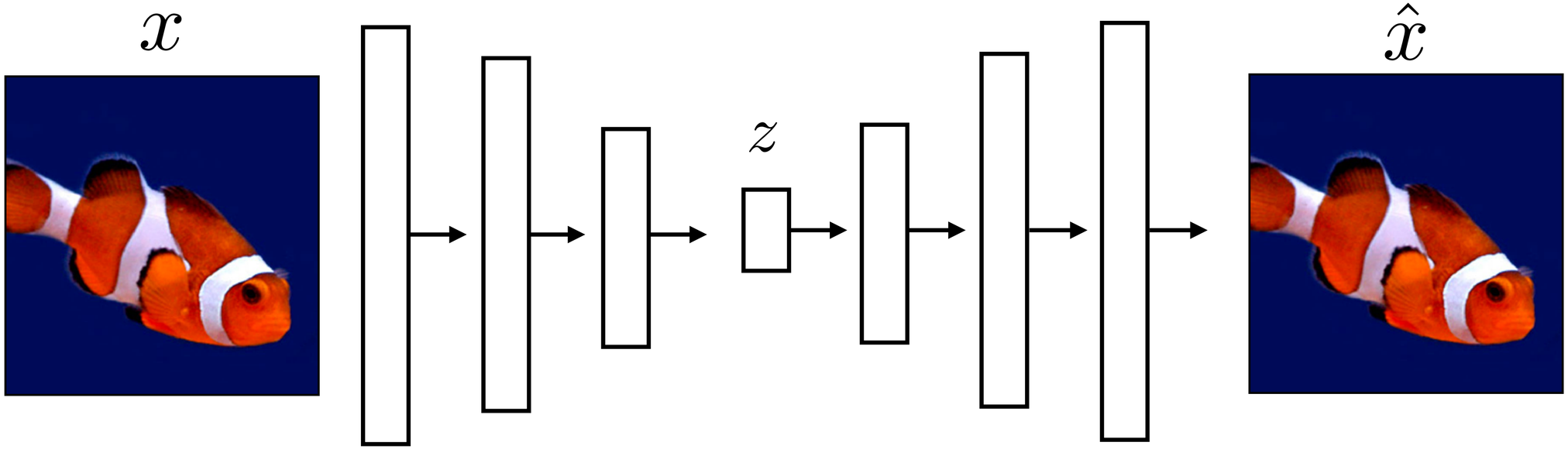
Train "inference network" $q_\psi(z|x)$
to give distribution over the z's that are likely to produce x

Approximate $p_\theta(x)$ with $\mathbb{E}_{q_\psi(z|x)}[p_\theta(x|z)]$

[Doersch, Tutorial on VAEs, 2016]

Variational Autoencoders (VAEs) — Optimization

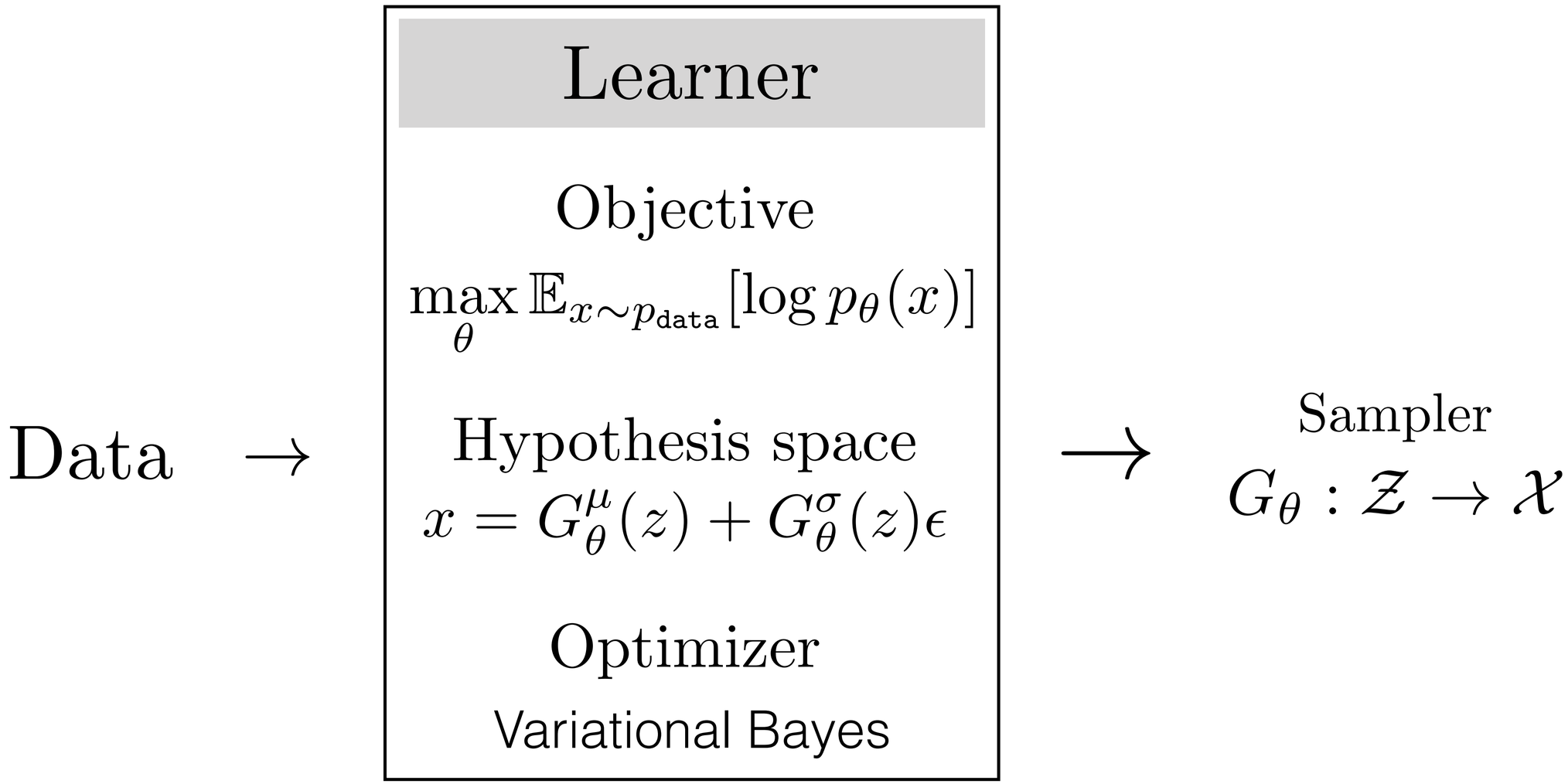
(aka training)



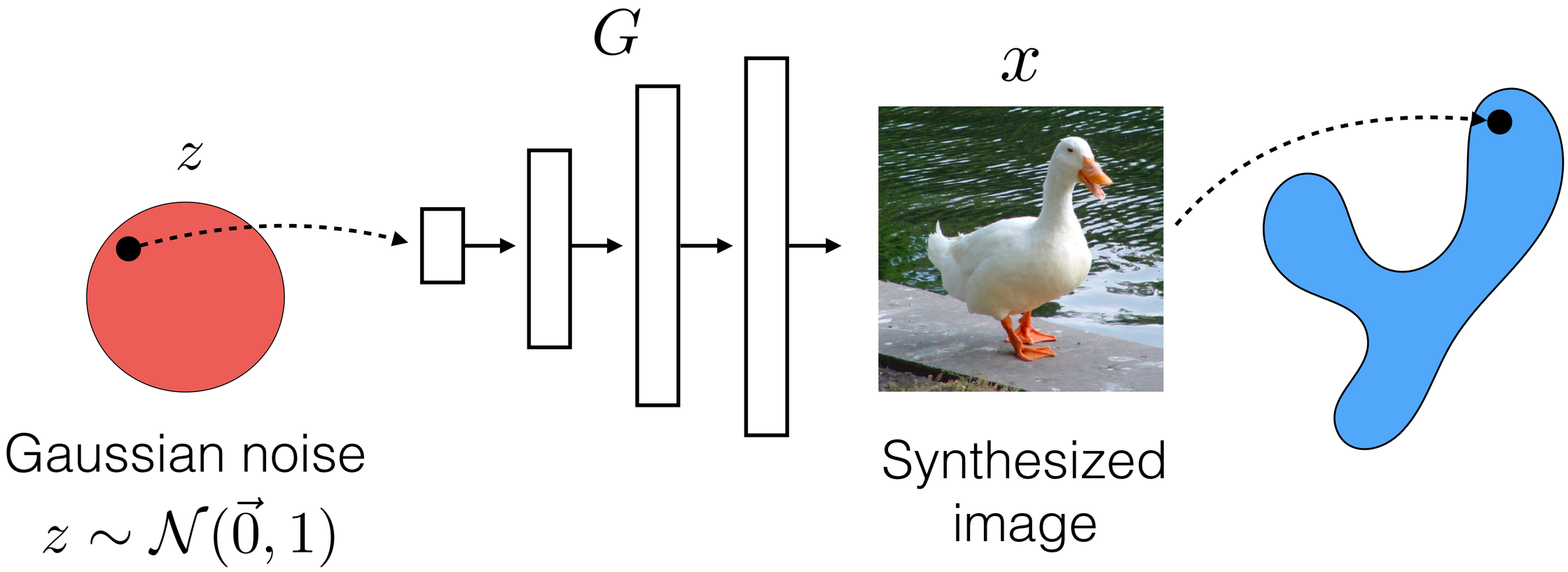
$$\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} [\log p_\theta(x)]$$

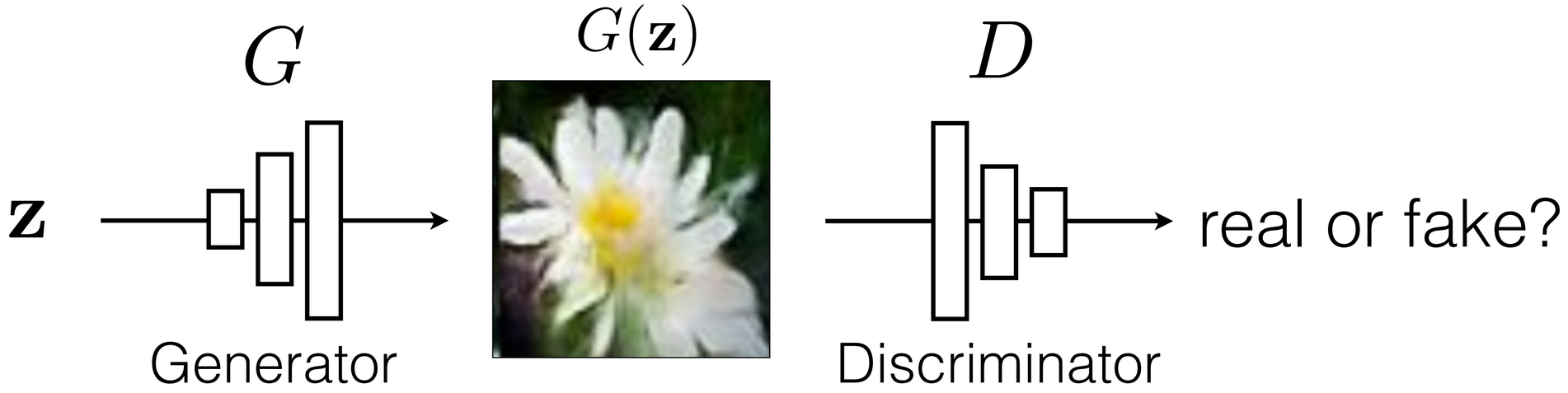
$$\geq \max_{\theta, \psi} \mathbb{E}_{x_i \sim p_{\text{data}}} [\mathbb{E}_{q_\psi(z|x_i)} [p_\theta(x|z)] - \text{KL}(q_\psi(z|x_i)||p(z))]$$

Variational Autoencoder (VAE)



Generative Adversarial Networks (GANs)

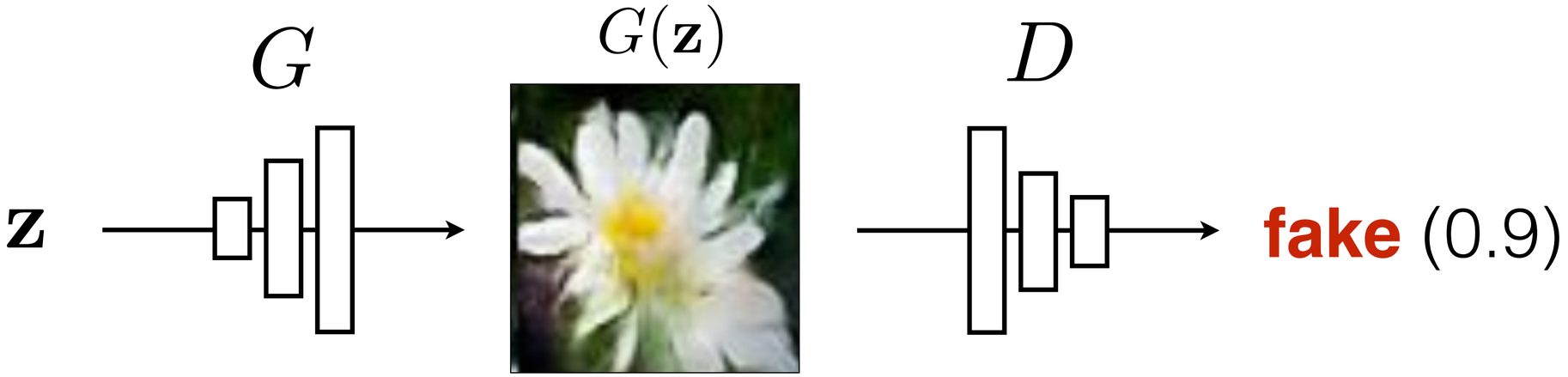


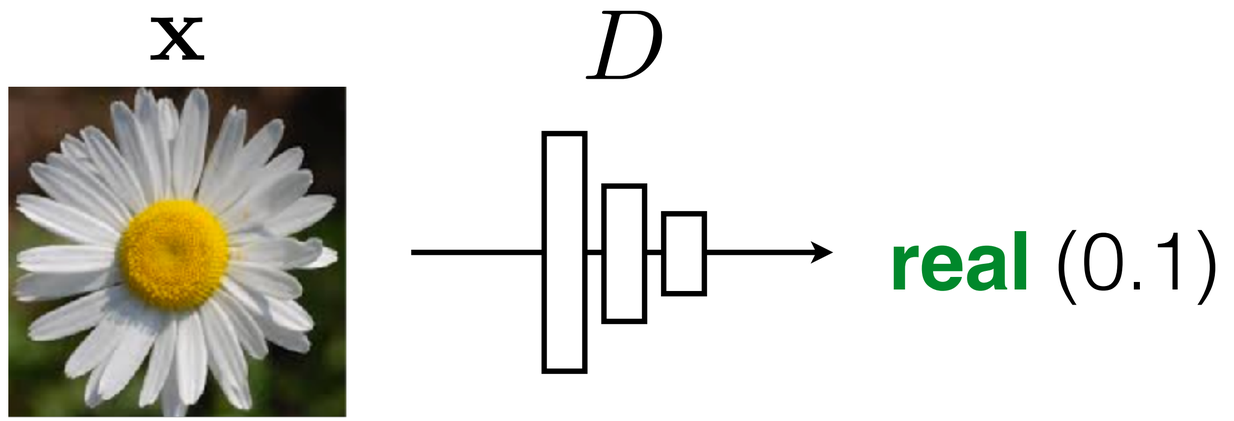


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

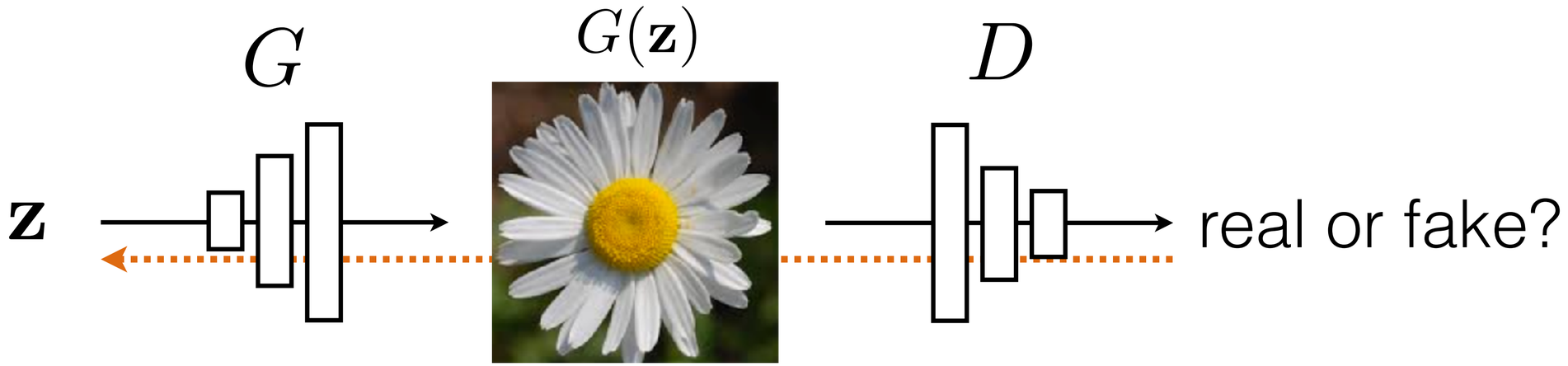
[Goodfellow et al., 2014]





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

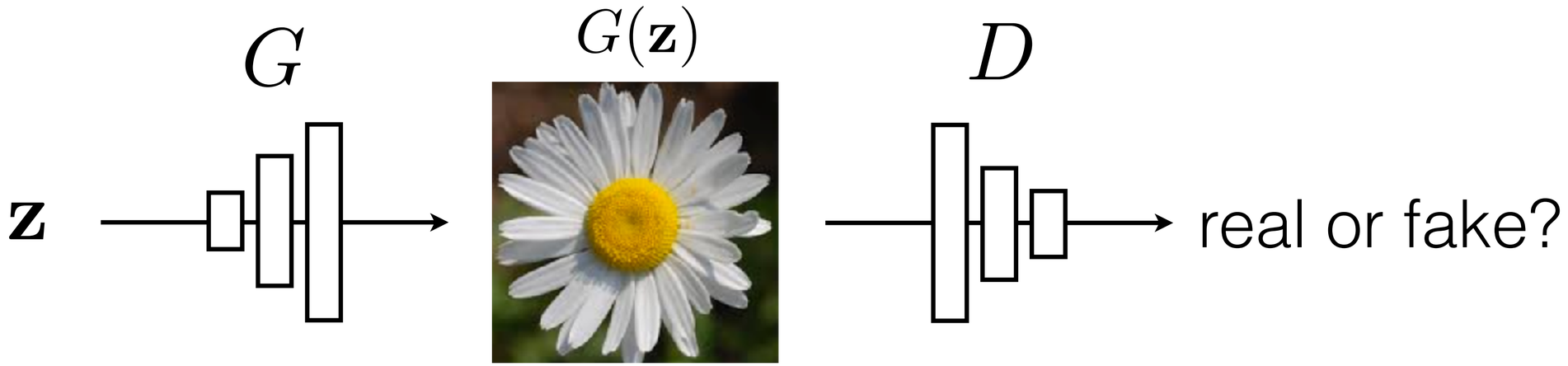
[Goodfellow et al., 2014]



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]

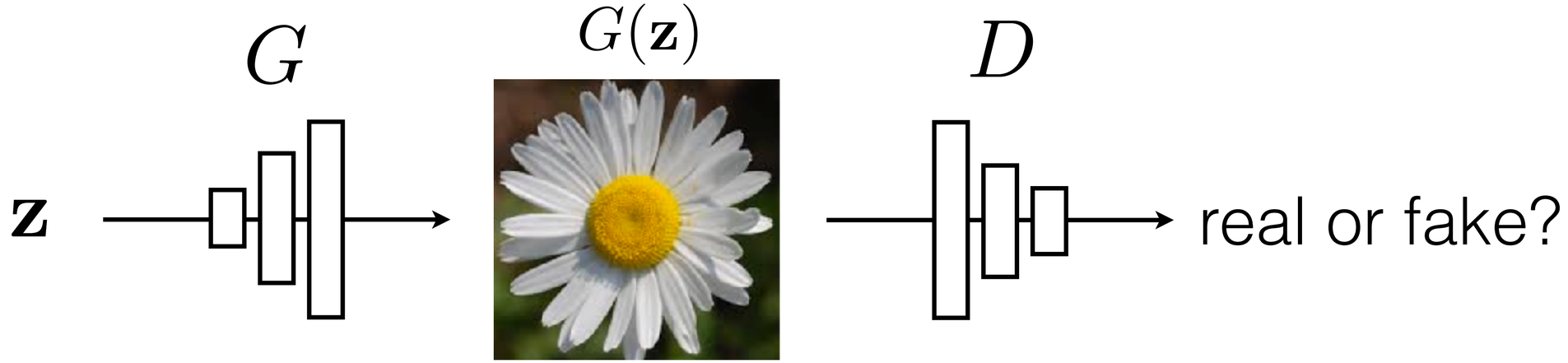


G tries to synthesize fake images that ***fool*** the ***best*** **D**:

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

[Goodfellow et al., 2014]

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]

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

Proof

$$
\begin{aligned}
C(G) =& \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
=& \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

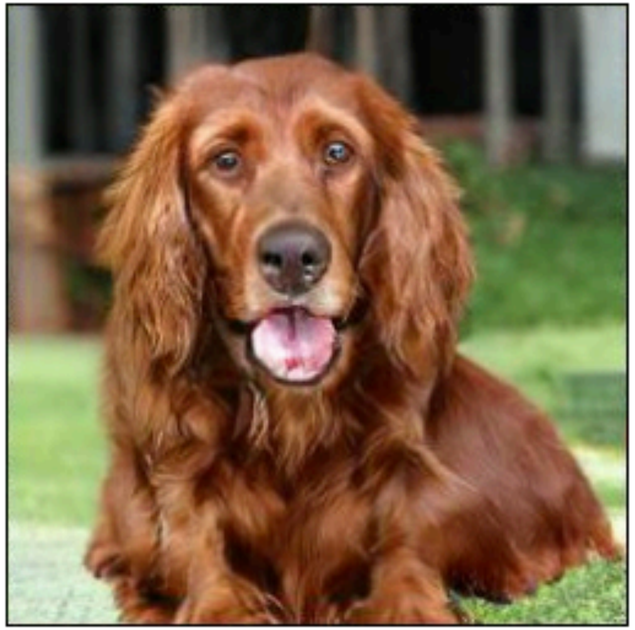
$$
C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right)
$$

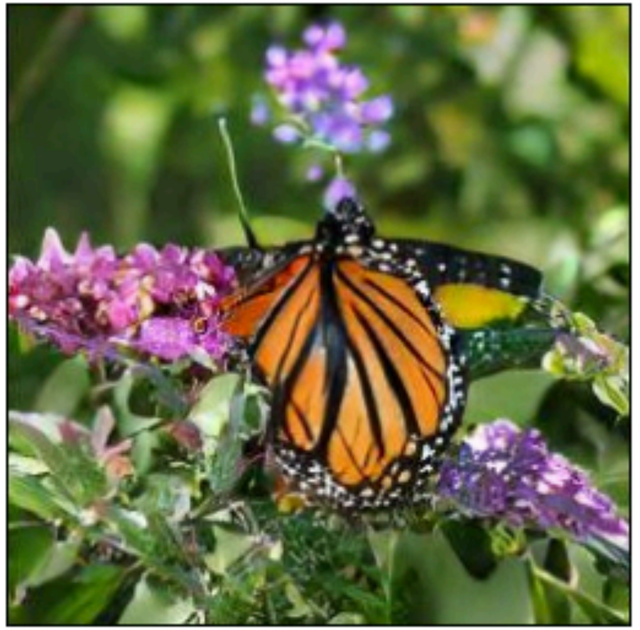
$$
C(G) = -\log(4) + 2 \cdot \underbrace{JSD\left(p_{\text{data}} \| p_g\right)}_{\geq 0}
$$

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

Samples from BigGAN

[Brock et al. 2018]



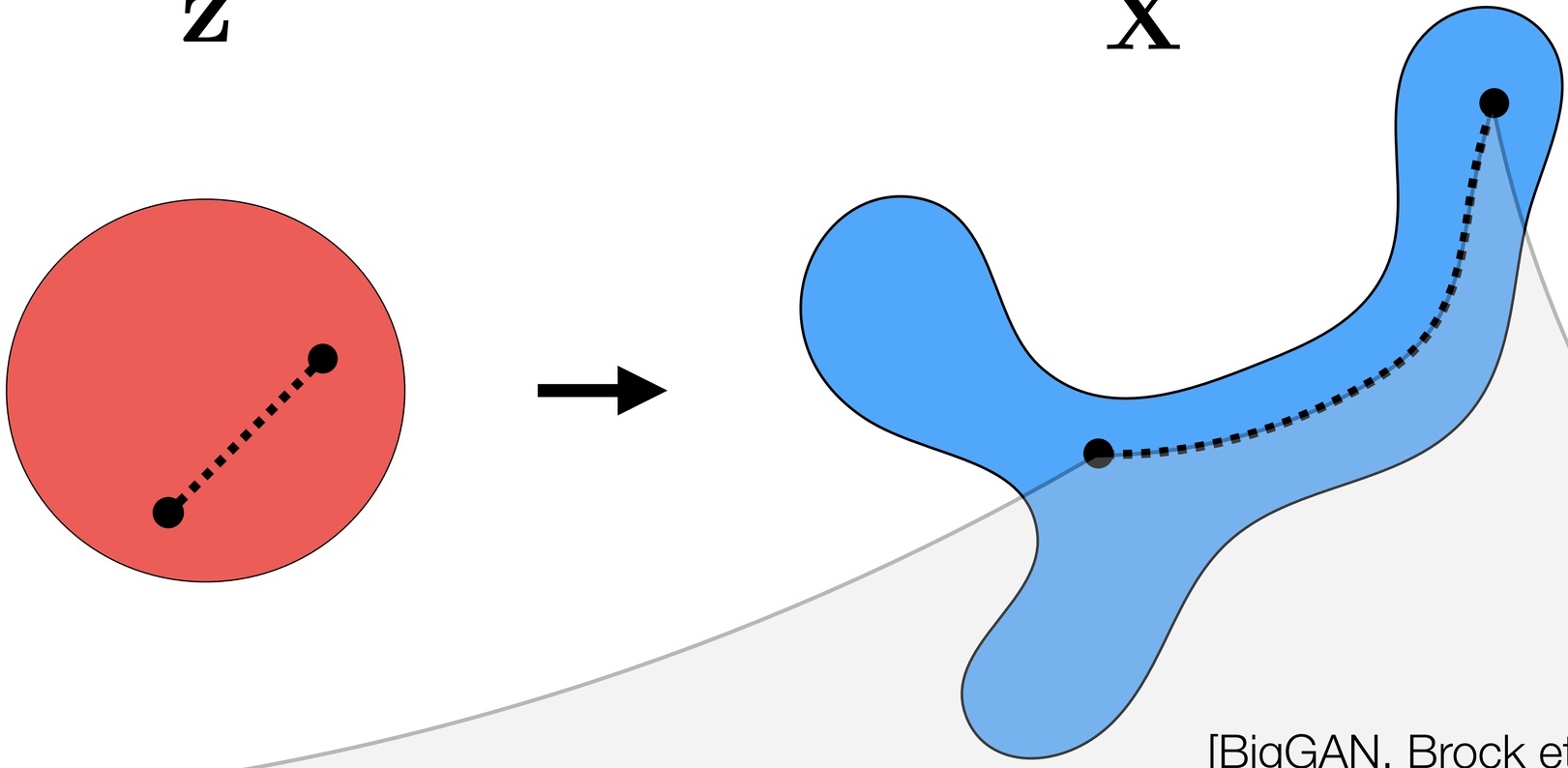



Latent space
(Gaussian)

$\mathbf{z}$

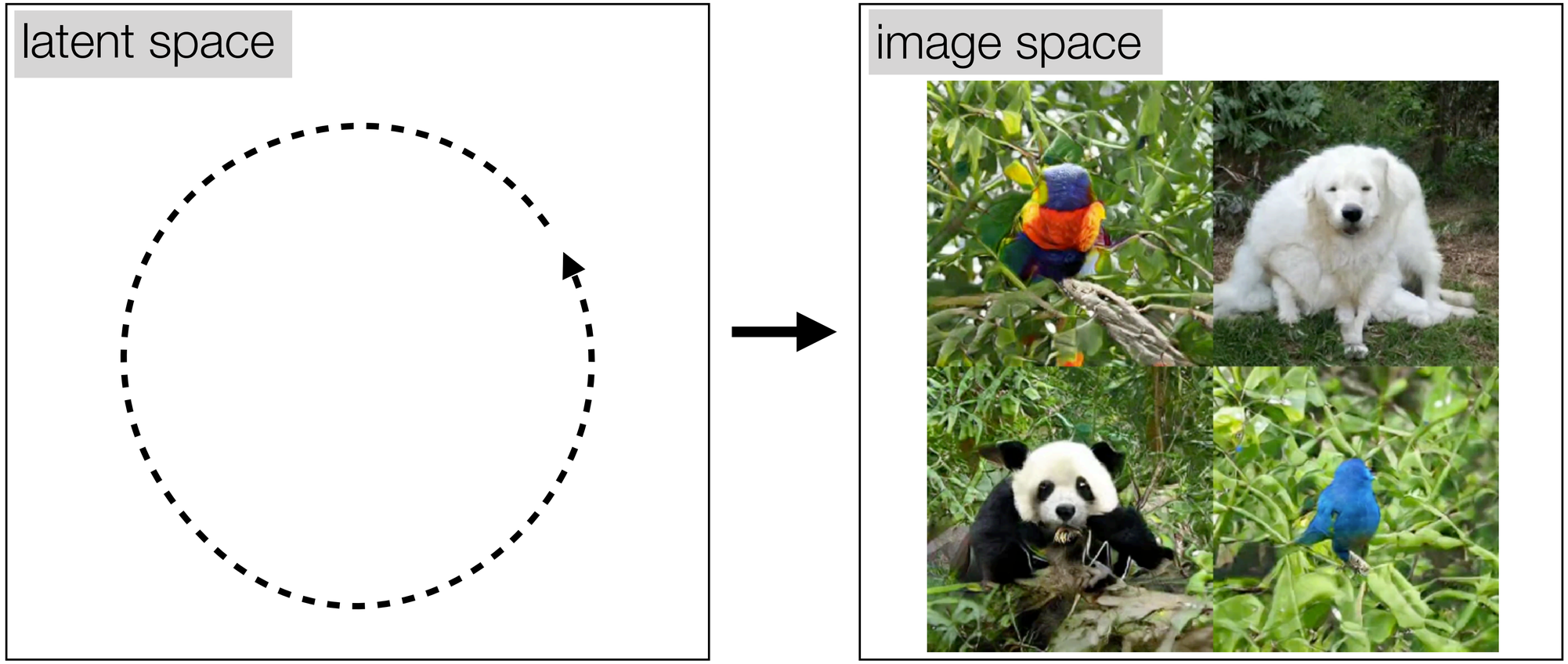
Data space
(Natural image manifold)

$\mathbf{x}$

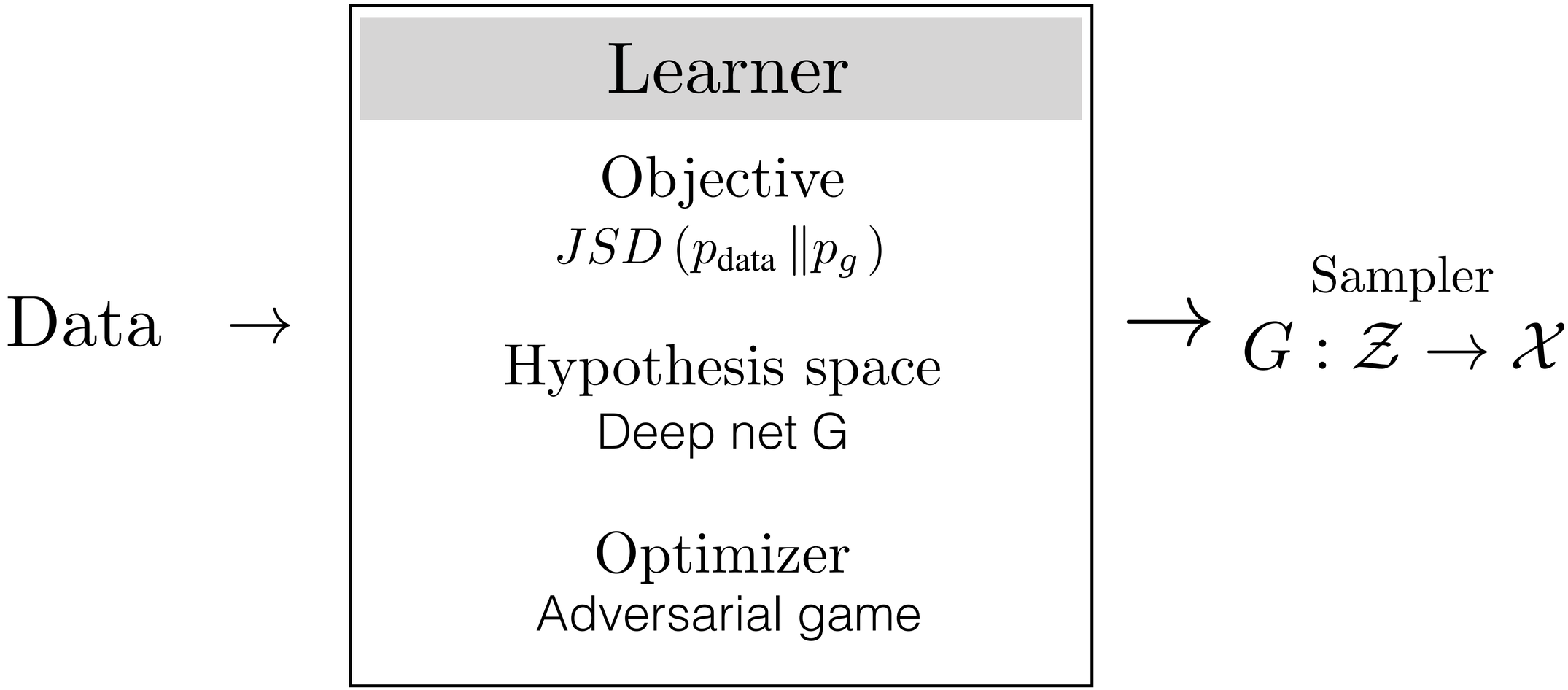


[BigGAN, Brock et al. 2018]

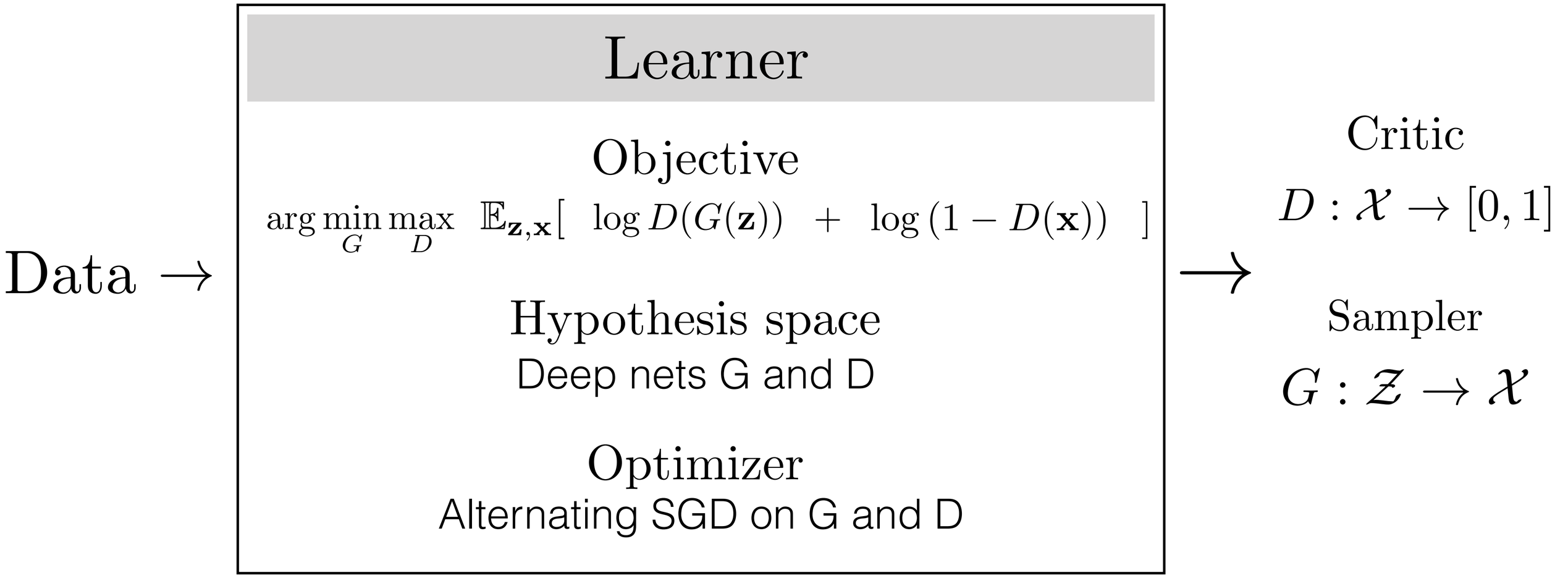
Generative models organize the manifold of natural images



Generative Adversarial Network



Generative Adversarial Network



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

Better objectives? optimizers?

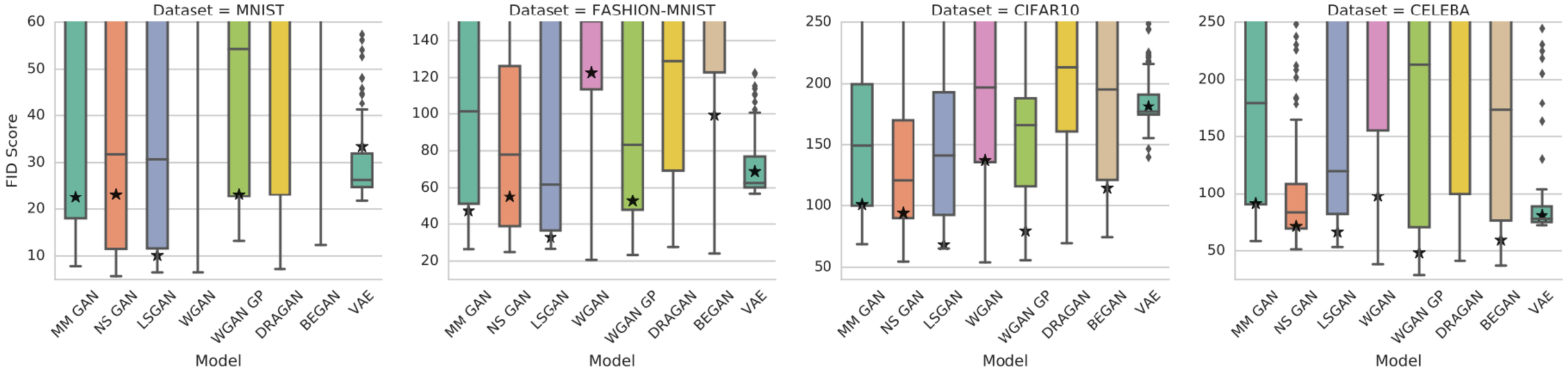
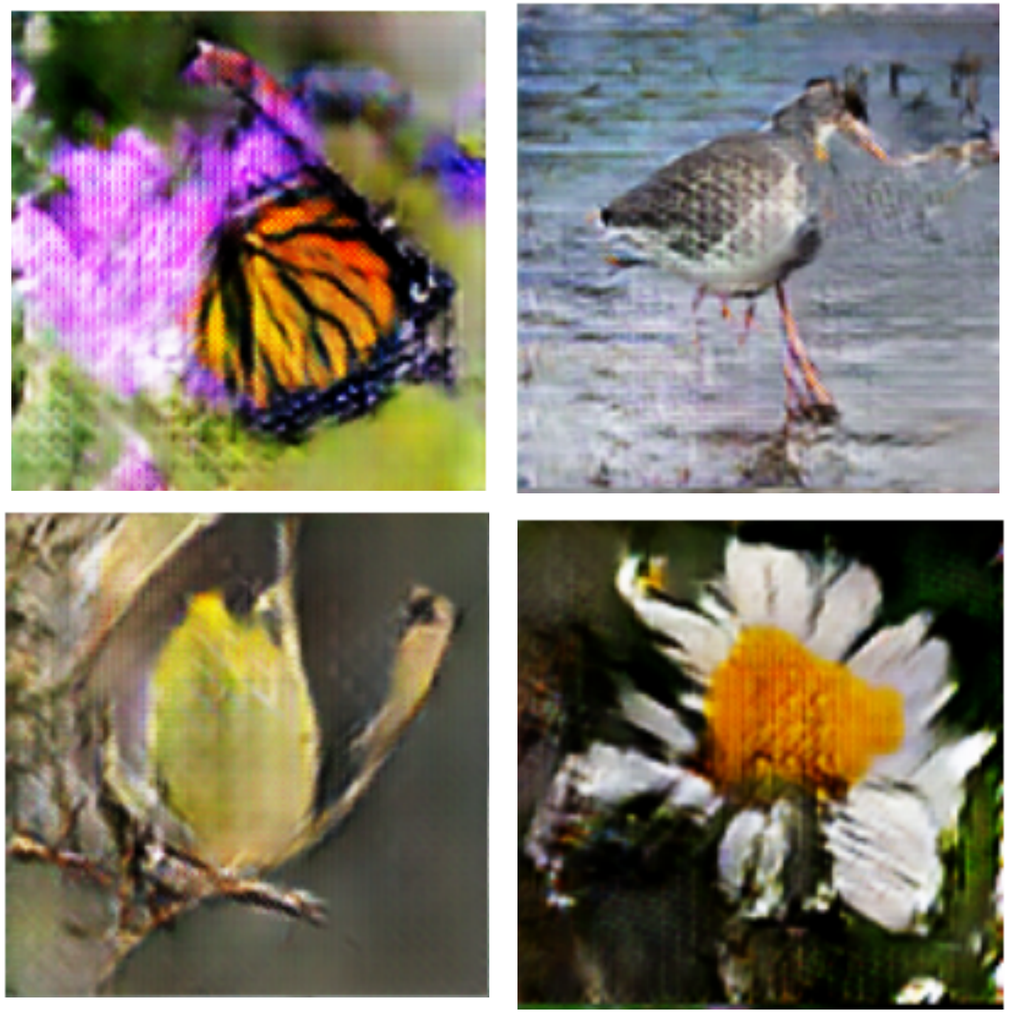


Figure 4: A *wide range* hyperparameter search (100 hyperparameter samples per model). Black stars indicate the performance of suggested hyperparameter settings. We observe that GAN training is extremely sensitive to hyperparameter settings and there is no model which is significantly more stable than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

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

More data?

ACGAN [Odena et al. 2016]



BigGAN [Brock et al. 2018]

Both trained on Imagenet

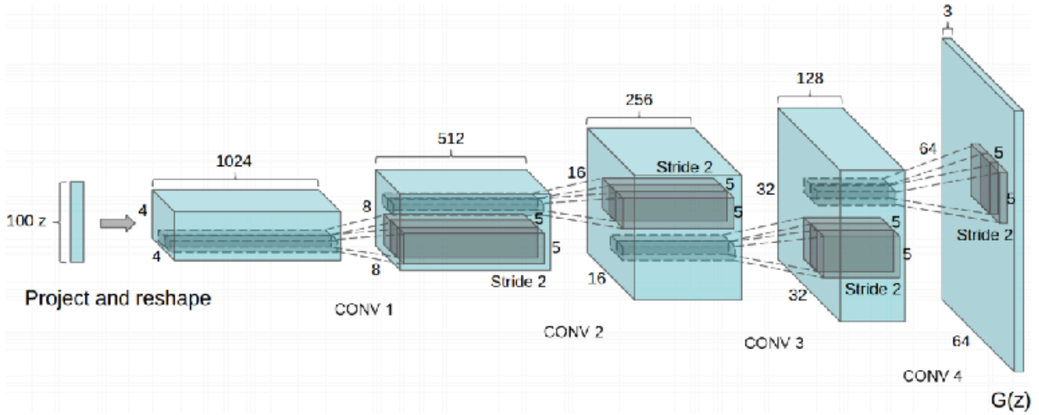
Architectures

DCGAN
[Radford, Metz, Chintala 2016]

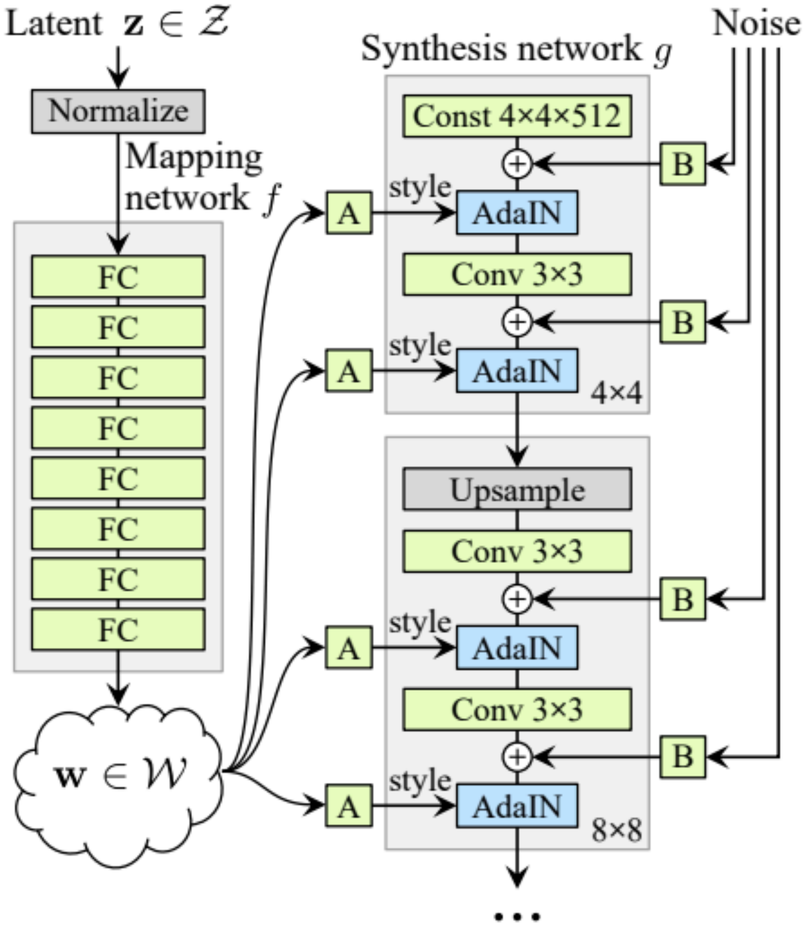
StyleGAN
[Karras, Laine, Aila 2019]

Architectures

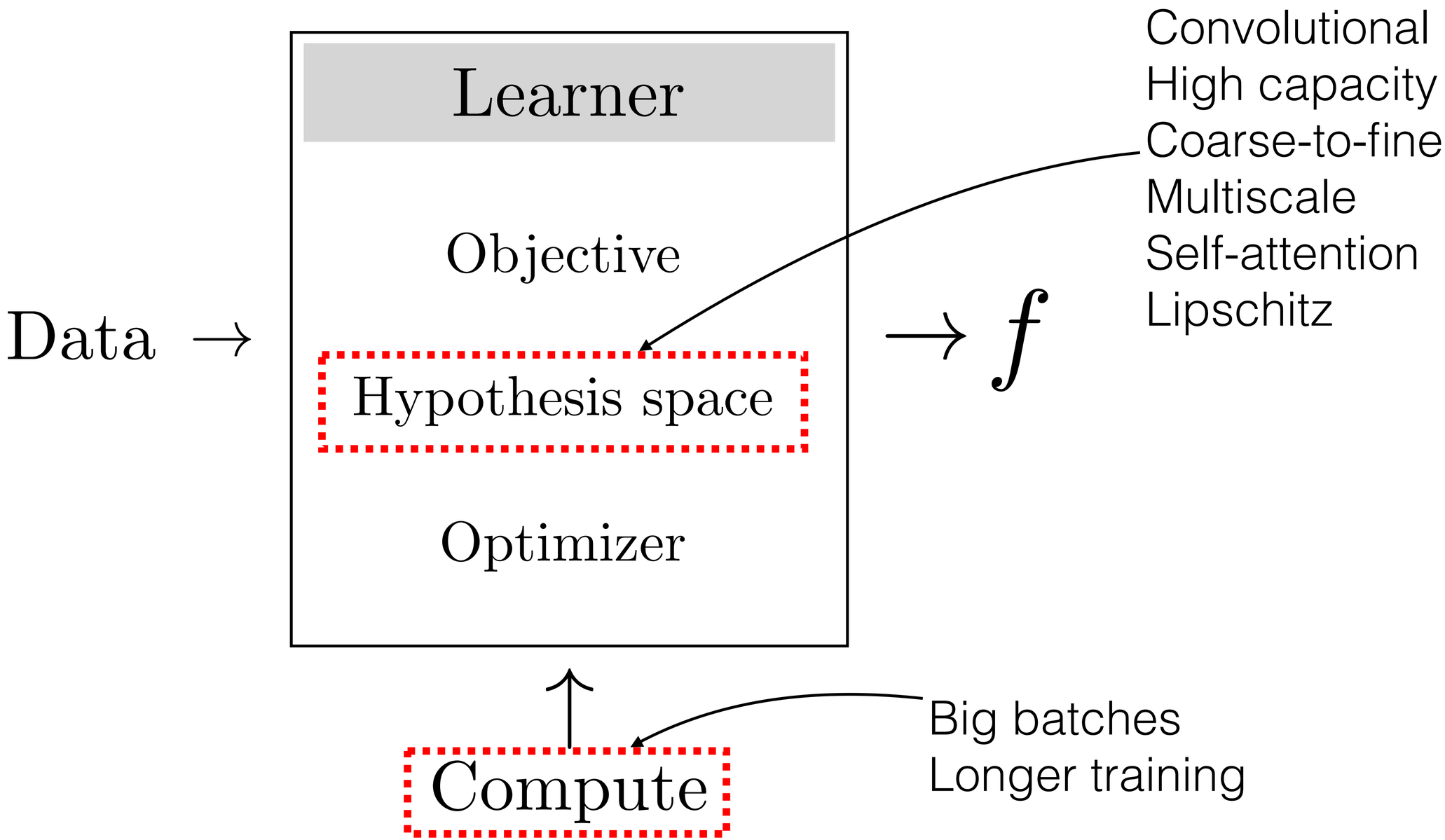
DCGAN
[Radford, Metz, Chintala 2016]



StyleGAN
[Karras, Laine, Aila 2019]



What has driven GAN progress?



VAEs

Pros: Cheap to sample, good coverage

Cons: Blurry samples (in practice)

GANs

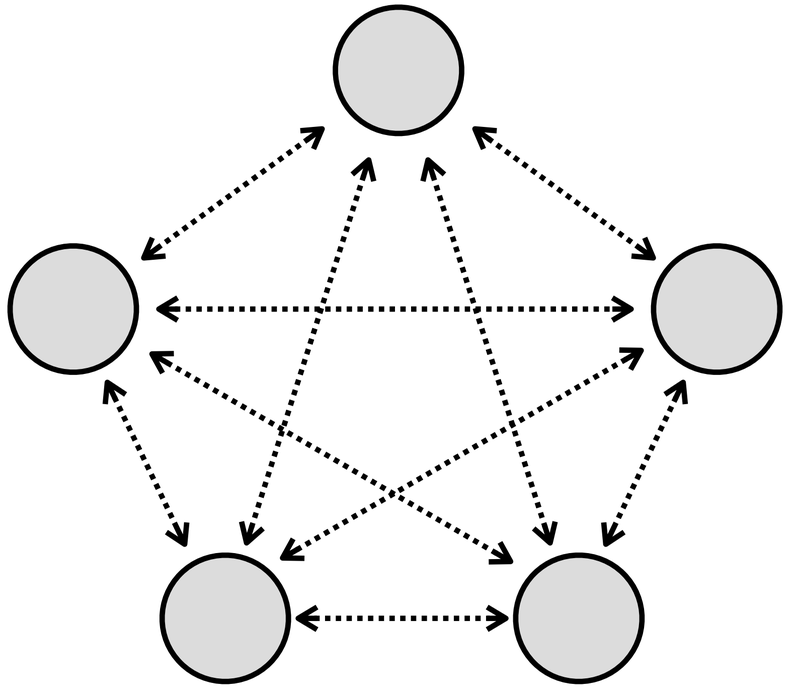
Pros: Cheap to sample, fast to train, require little data

Cons: No likelihoods, bad coverage (mode collapse), finicky to train (minimax)

Other deep generative models:
Autoregressive models, Normalizing flows, **Energy-based models**

[adapted from slide by David Duvenaud]

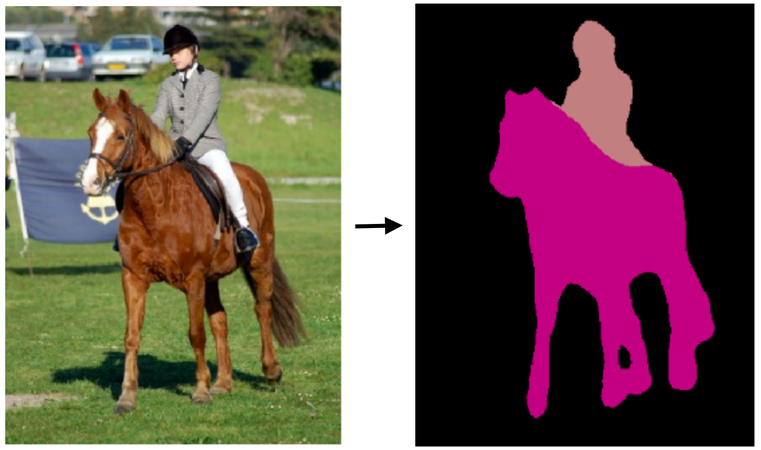
1. Image synthesis
2. **Structured prediction**
3. Domain mapping



Strutured Prediction

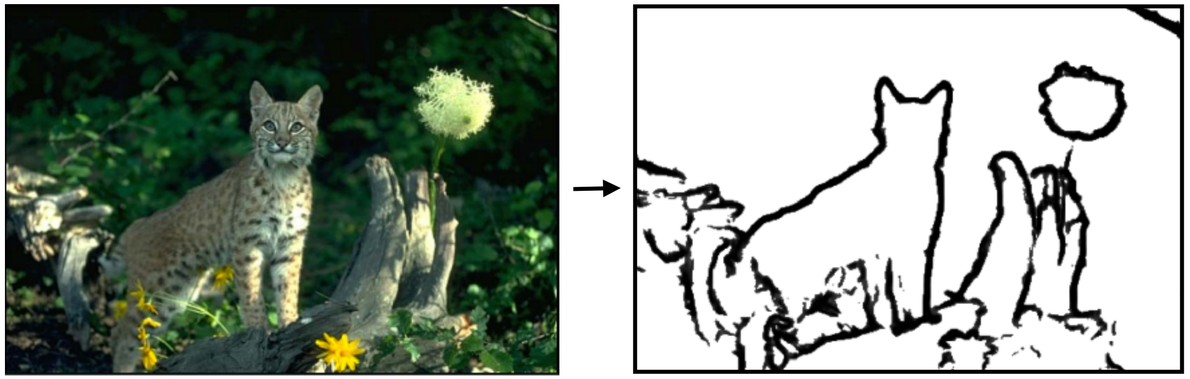
Data prediction problems ("structured prediction")

Semantic segmentation



[Long et al. 2015, …]

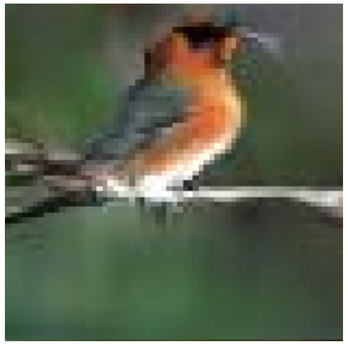
Edge detection



[Xie et al. 2015, …]

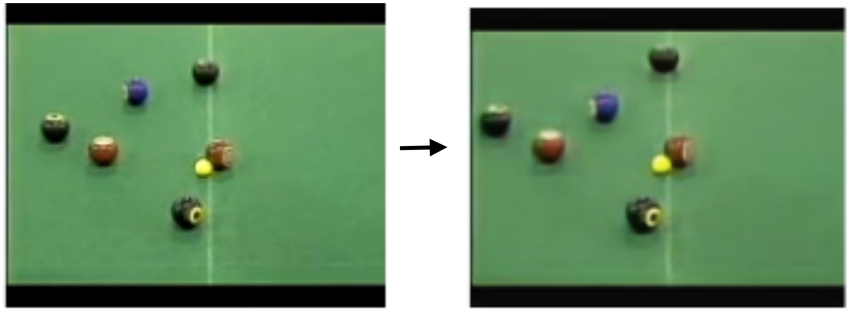
Text-to-photo

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



[Reed et al. 2014, …]

Future frame prediction



[Mathieu et al. 2016, …]

Structured prediction

X is high-dimensional

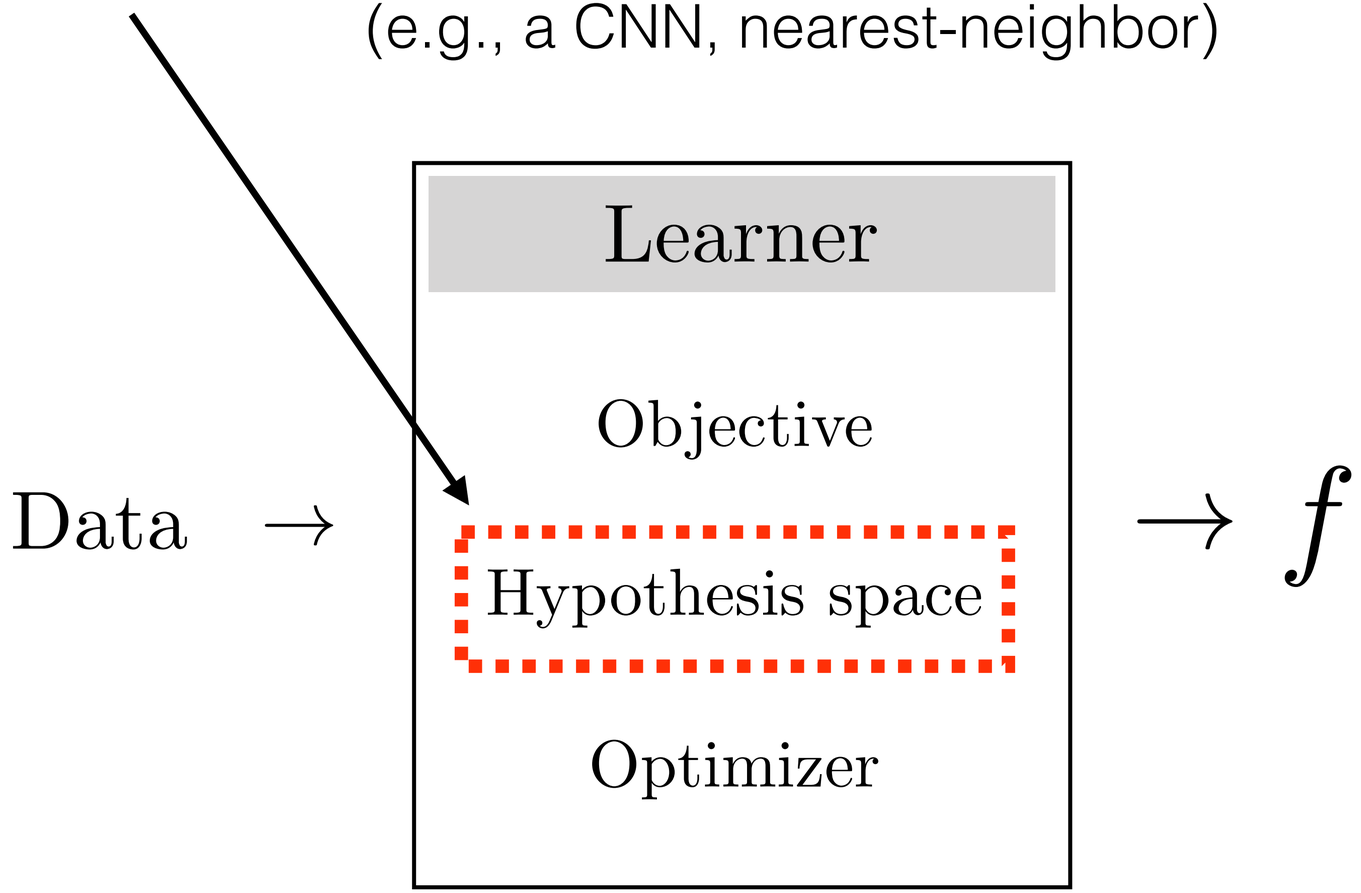
Model *joint* distribution of high-dimensional data $P(\mathbf{X}|\mathbf{Y}=\mathbf{y})$

In vision this is usually what we are interested in

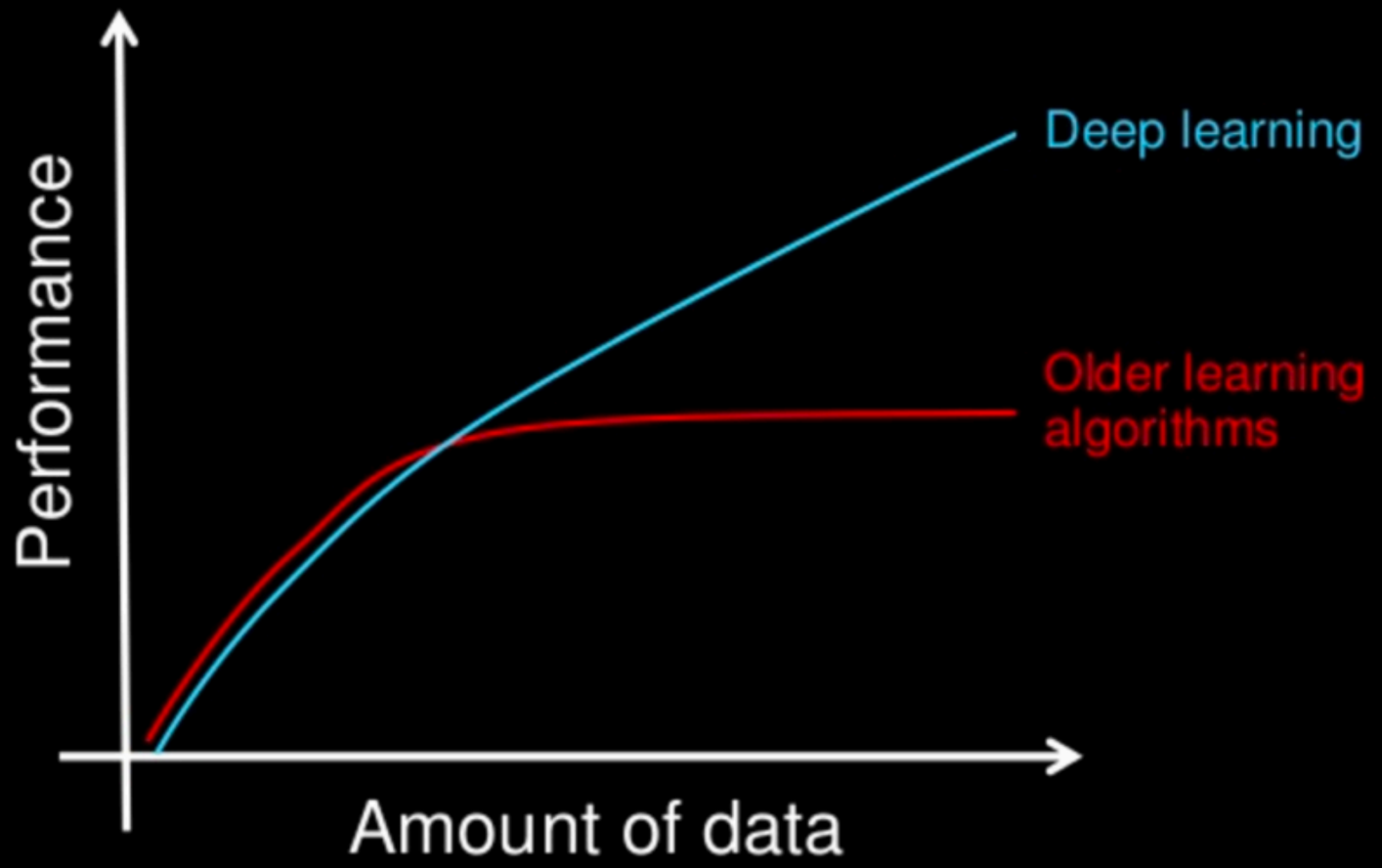
Unstructured: $\prod_i p(X_i|\mathbf{Y}=\mathbf{y})$

Deep learning in 2012

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

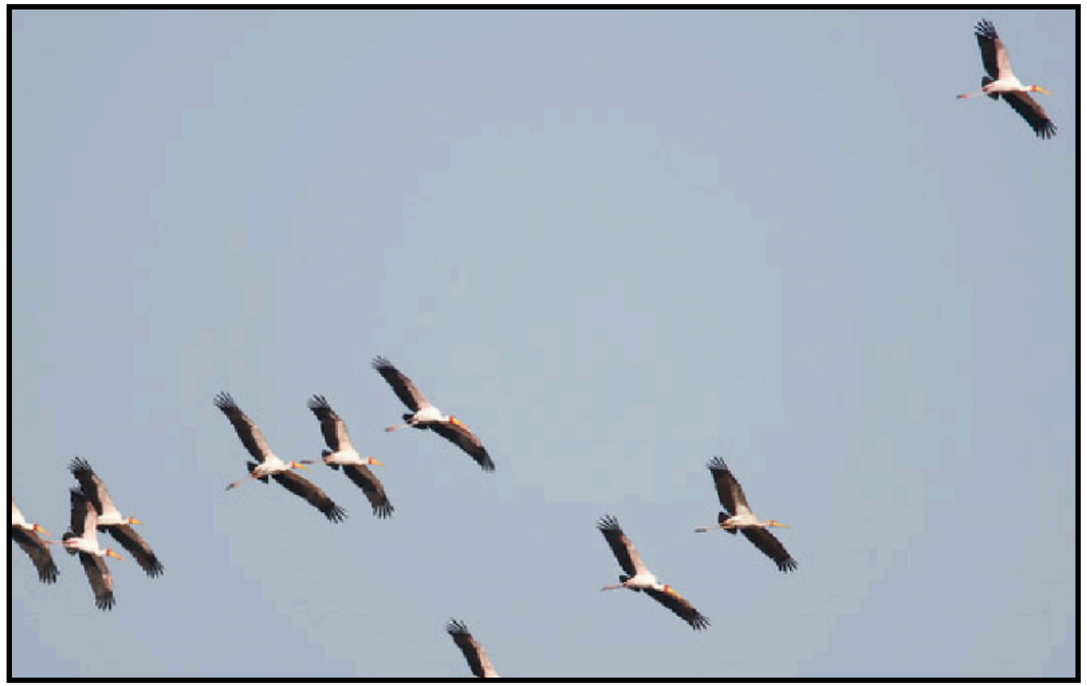


Why deep learning

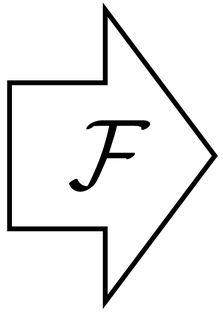


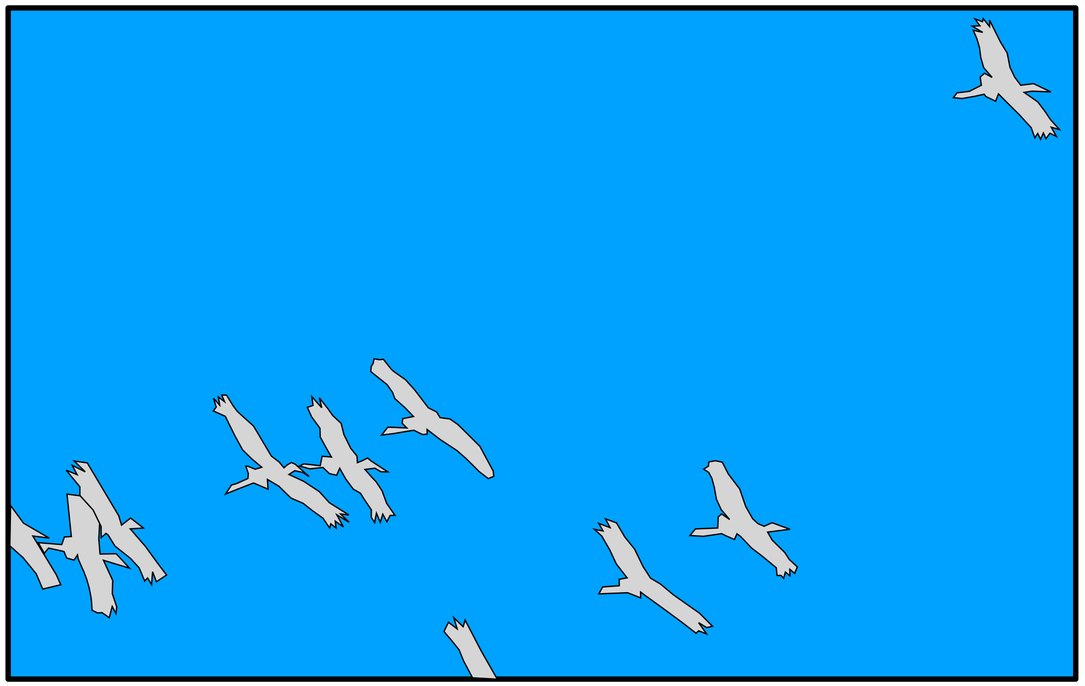
How do data science techniques scale with amount of data?

[Slide credit: Andrew Ng]



[Photo credit: Fredo Durand]



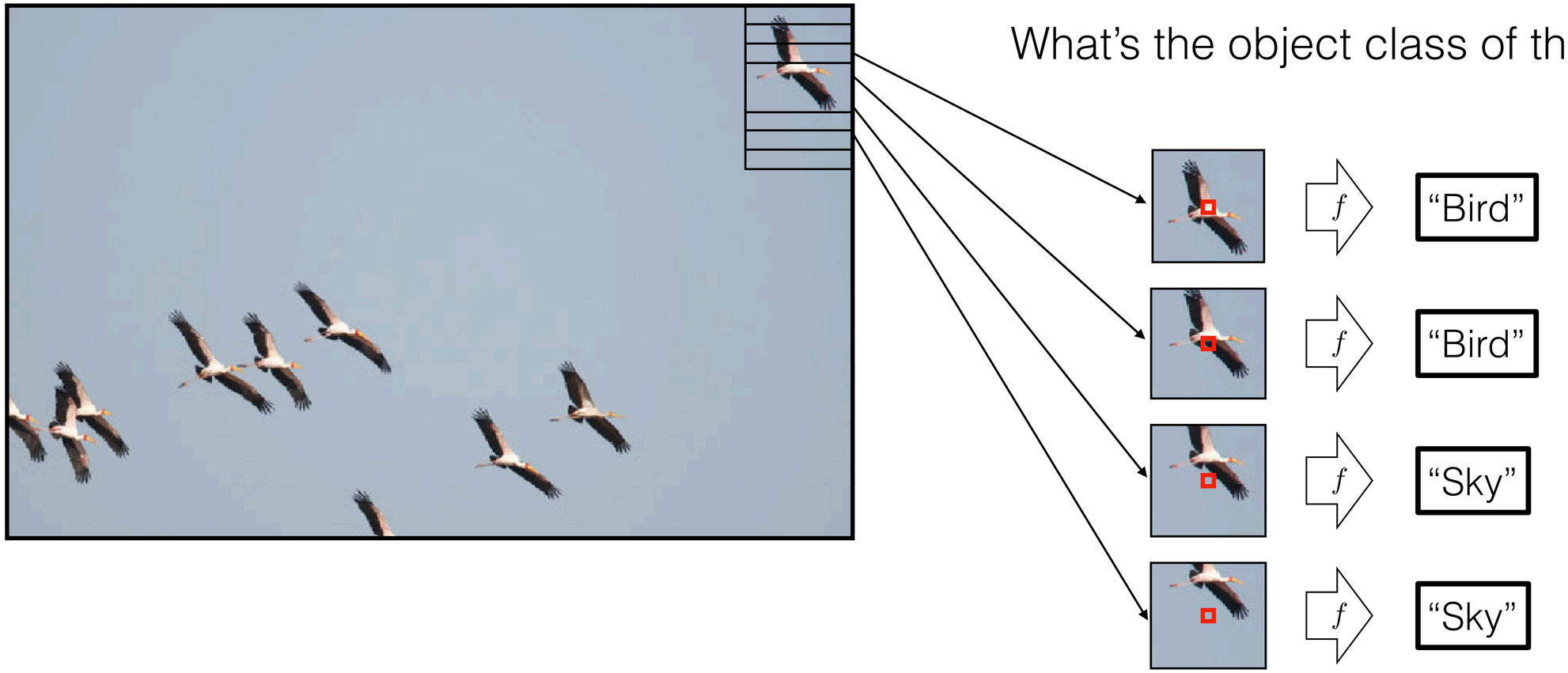


(Colors represent one-hot codes)

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

Hypothesis space

Objective function
(loss)

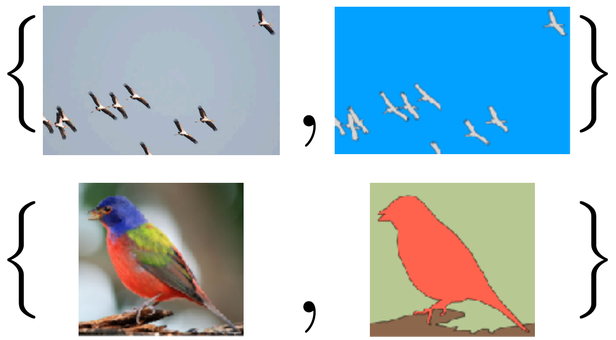


Fully-factored loss: $L(\hat{\mathbf{y}}, \mathbf{y}) = \sum_i \phi_i(\hat{\mathbf{y}}_i, \mathbf{y}_i)$

Semantic Segmentation

Data

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



$\vdots$

$\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$

$\mathbf{y} \in \mathbb{R}^{H \times W \times K}$

$\rightarrow$

Learner

Objective

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} H(\mathbf{y}_i, \hat{\mathbf{y}}_i)$$

Hypothesis space

Convolutional neural net

Optimizer

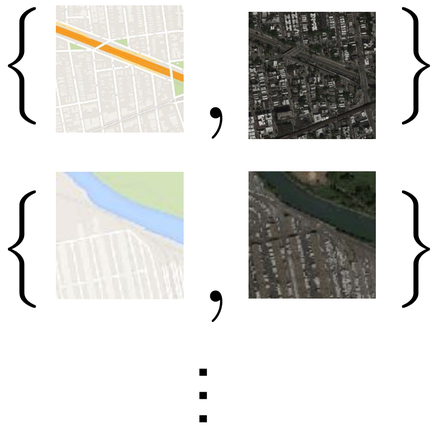
Stochastic gradient descent

$\rightarrow$ f

Sat2Map

Data

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



$\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$

$\mathbf{y} \in \mathbb{R}^{H \times W \times 3}$

$\rightarrow$

Learner

Objective

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} (f_\theta(\mathbf{x})_i - y_i)^2$$

Hypothesis space

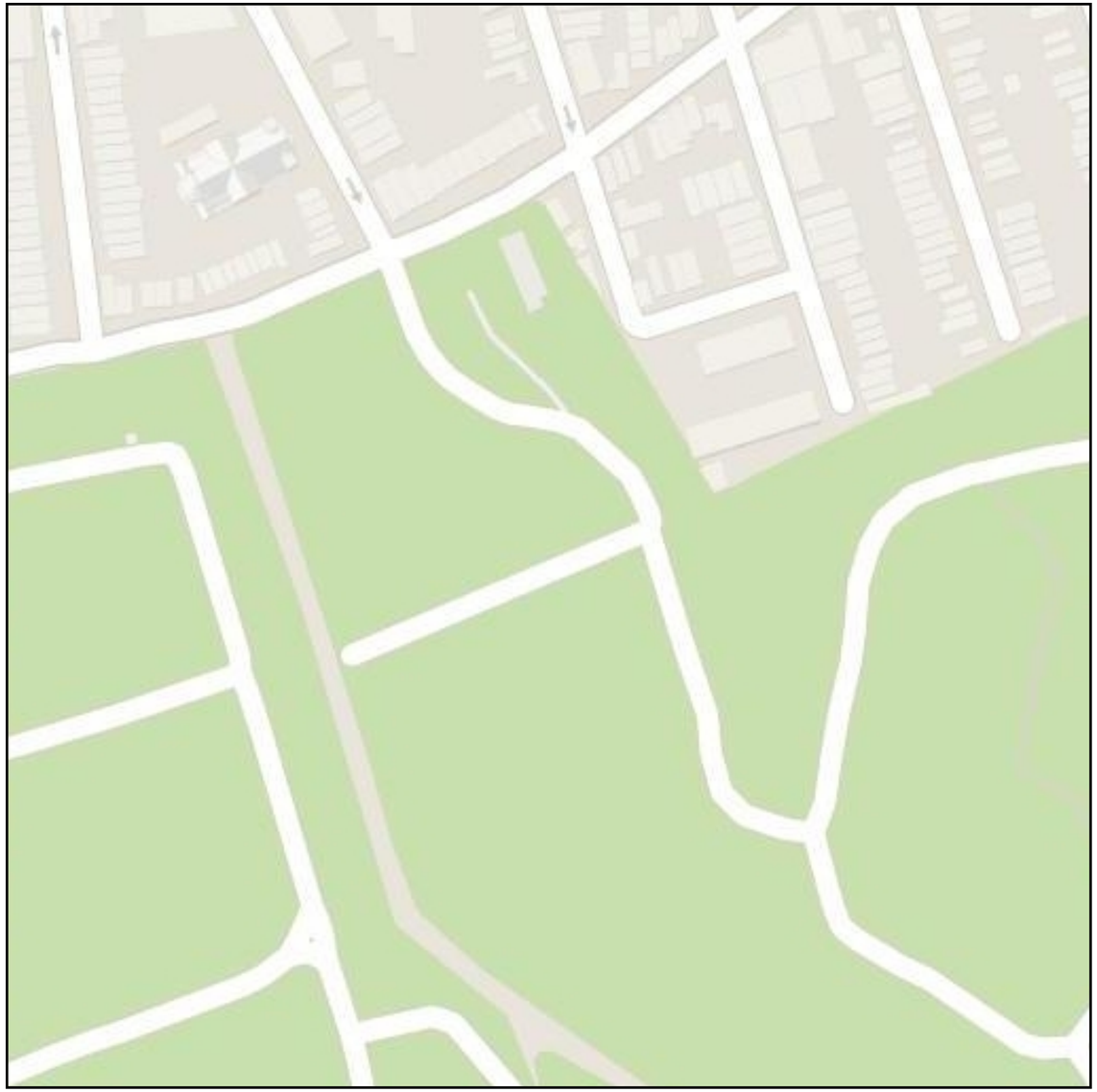
Convolutional neural net

Optimizer

Stochastic gradient descent

$\rightarrow$ f

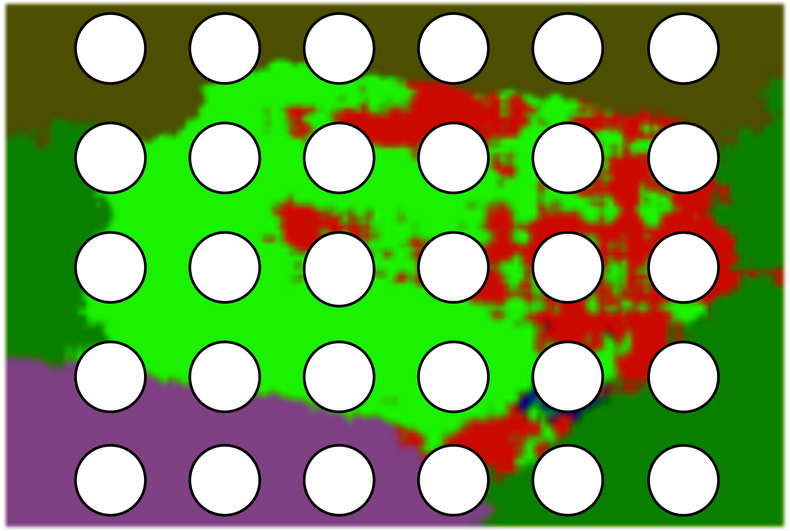
Input



Deep net output

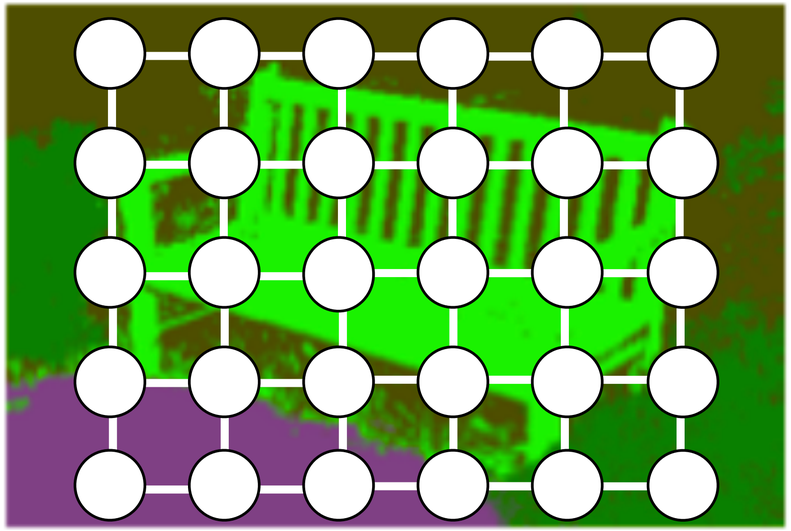
Input

Independent prediction per-pixel



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

Find a configuration of compatible labels

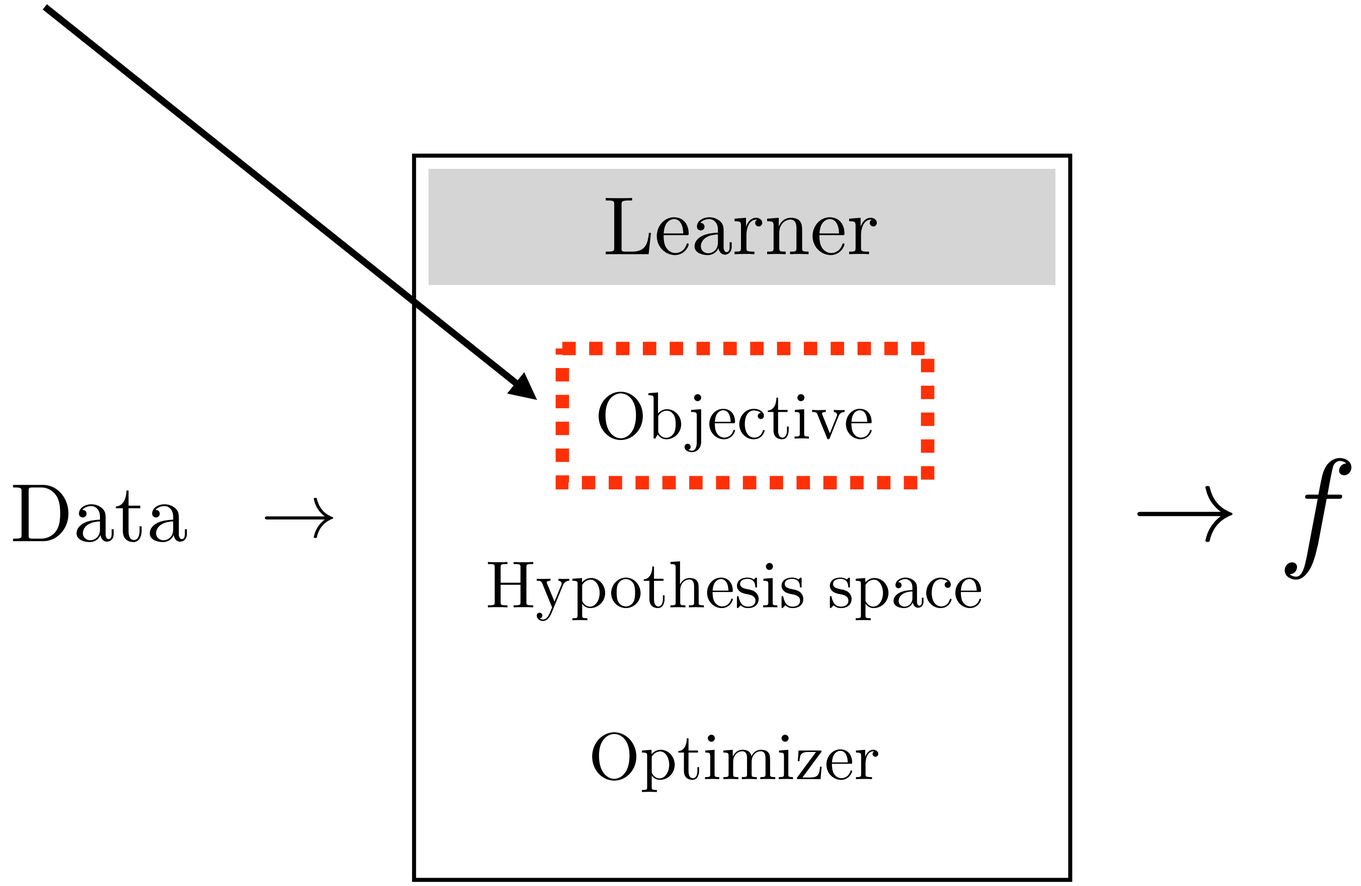


$$\max \frac{1}{Z} \prod_{i,j} p(y_i, y_j|\mathbf{x})$$

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

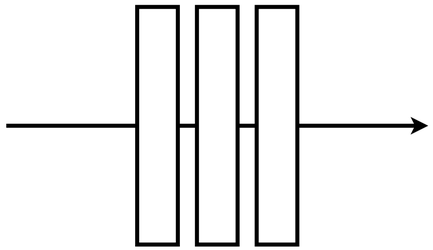
Structured prediction

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



$\mathbf{x}$

G



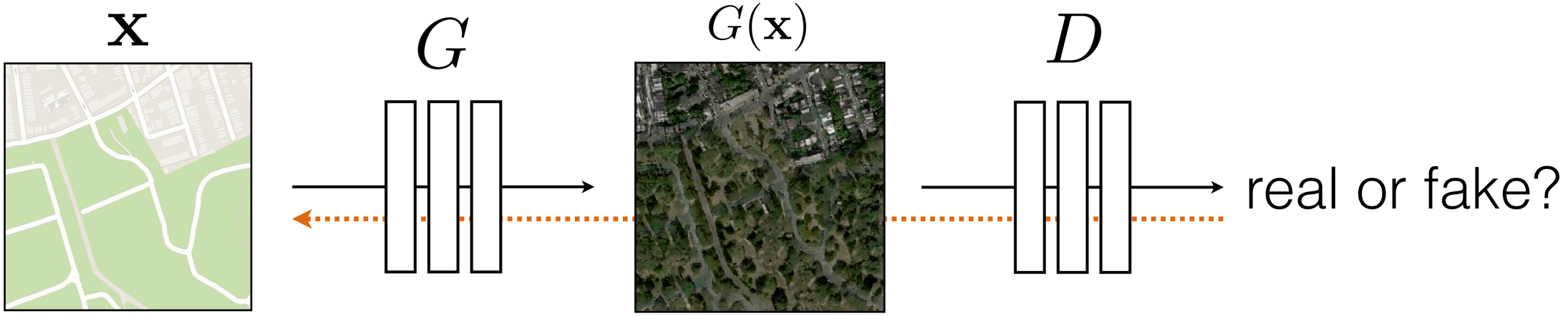
Generator

$G(\mathbf{x})$

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

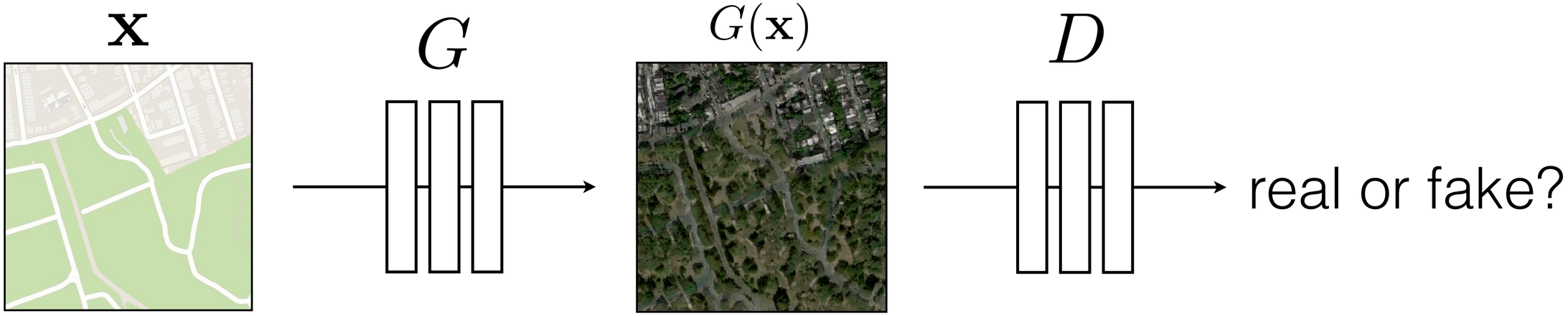
D tries to identify the fakes

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



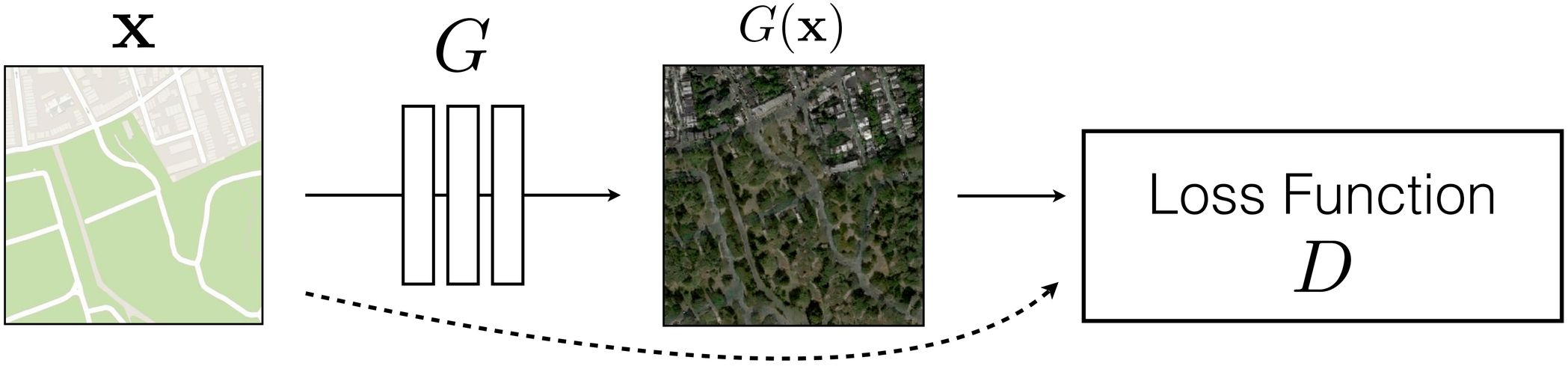
G tries to synthesize fake images that ***fool*** **D**:

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



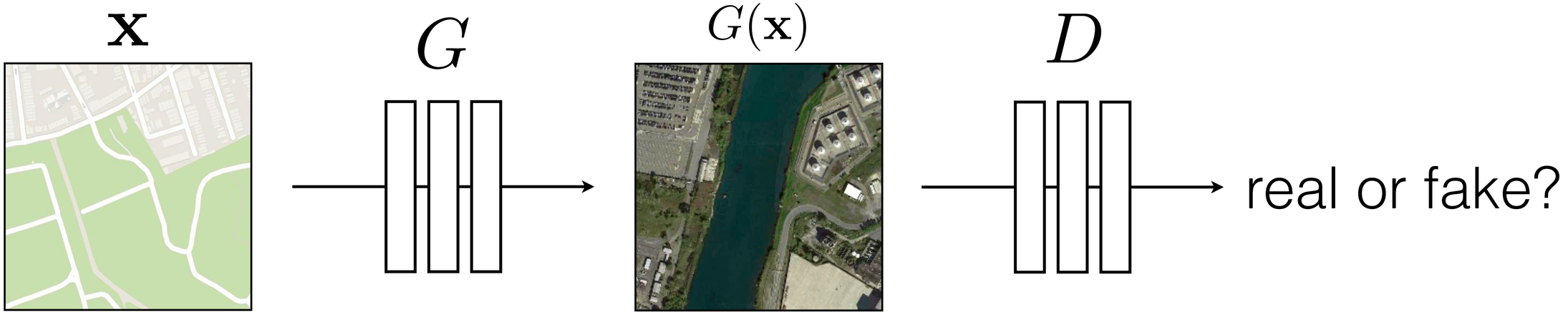
G tries to synthesize fake images that ***fool*** the ***best*** **D**:

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

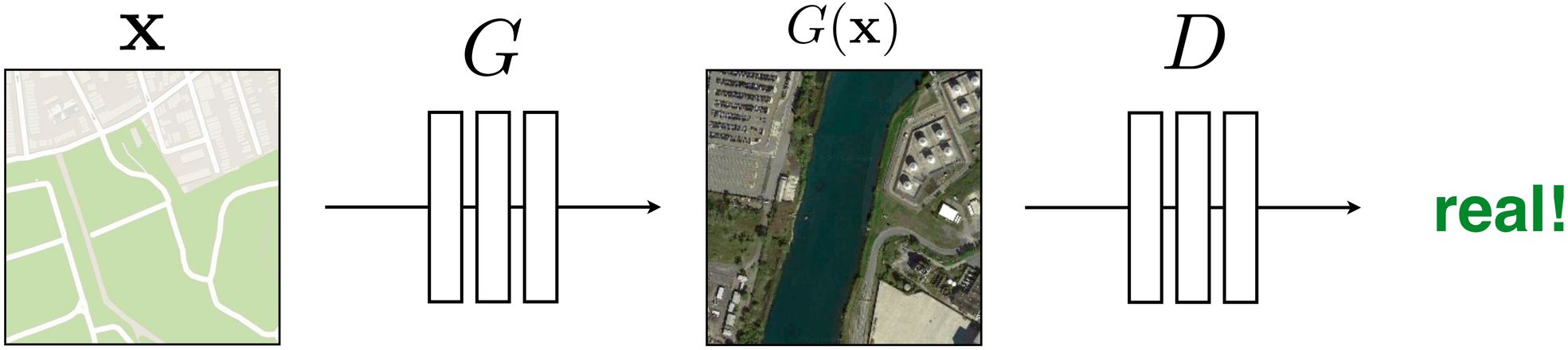


G's perspective: **D** is a loss function.

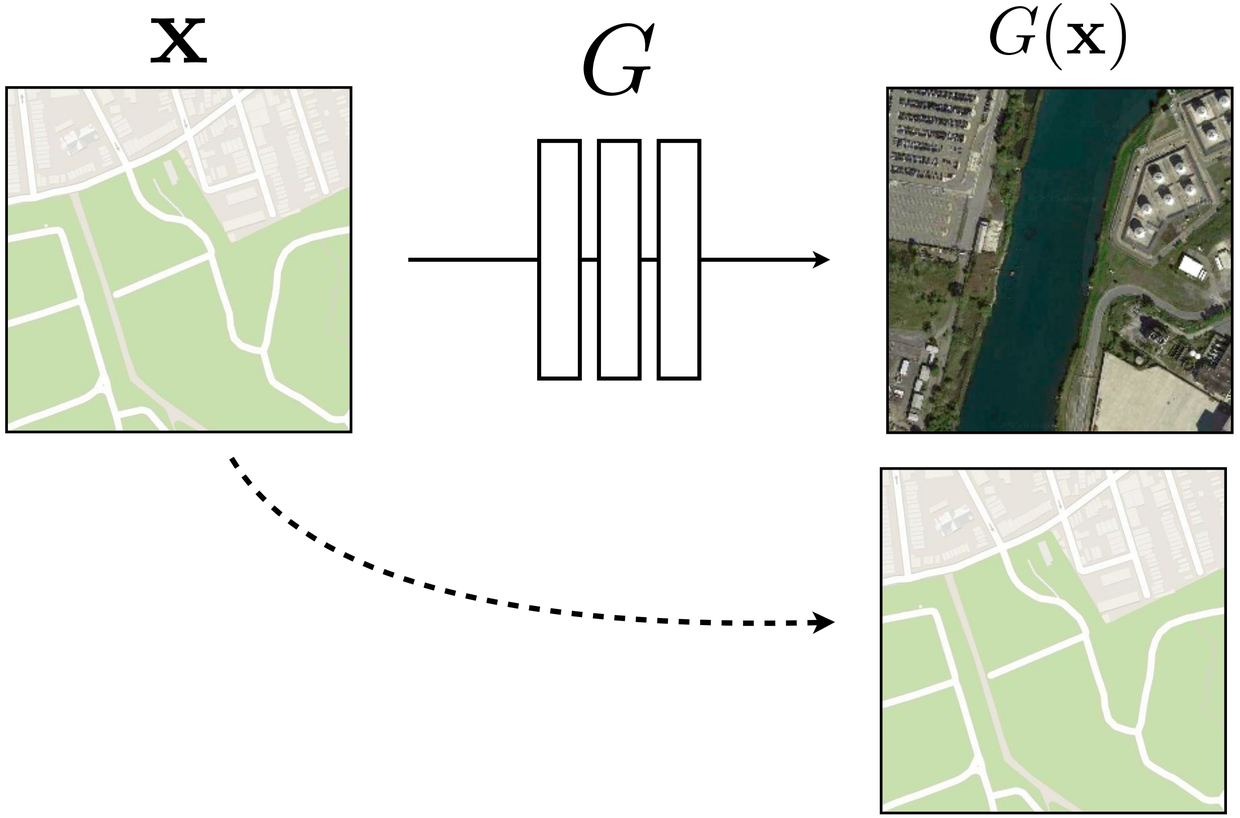
Rather than being hand-designed, it is *learned* and *highly structured*.



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



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



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

$\mathbf{x}$

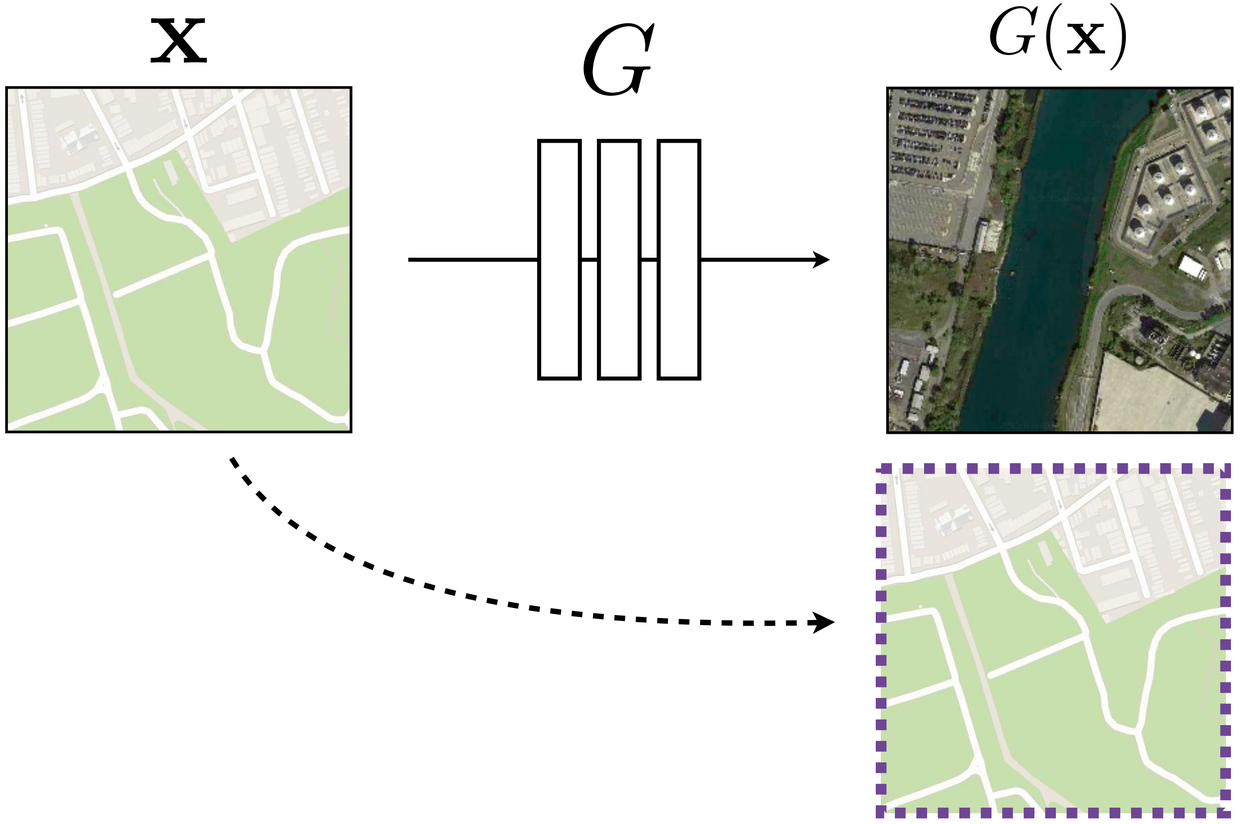
G

$G(\mathbf{x})$

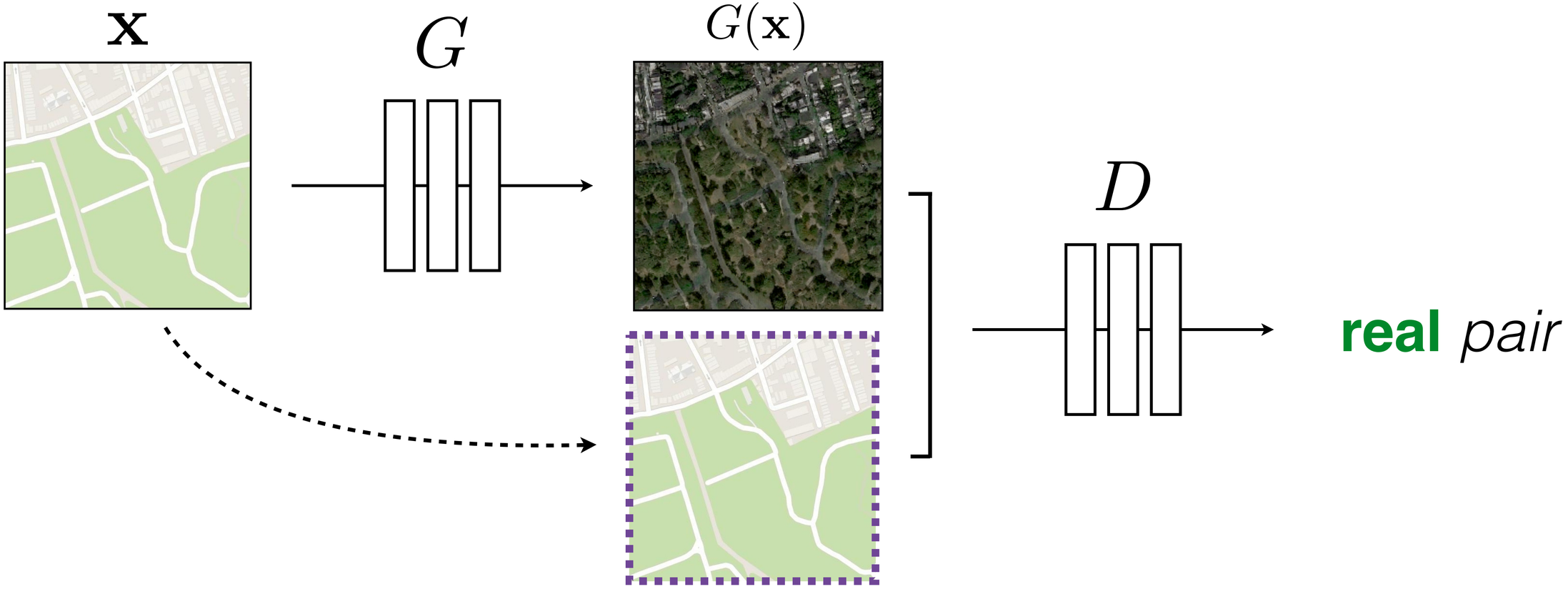
D

real or fake *pair* ?

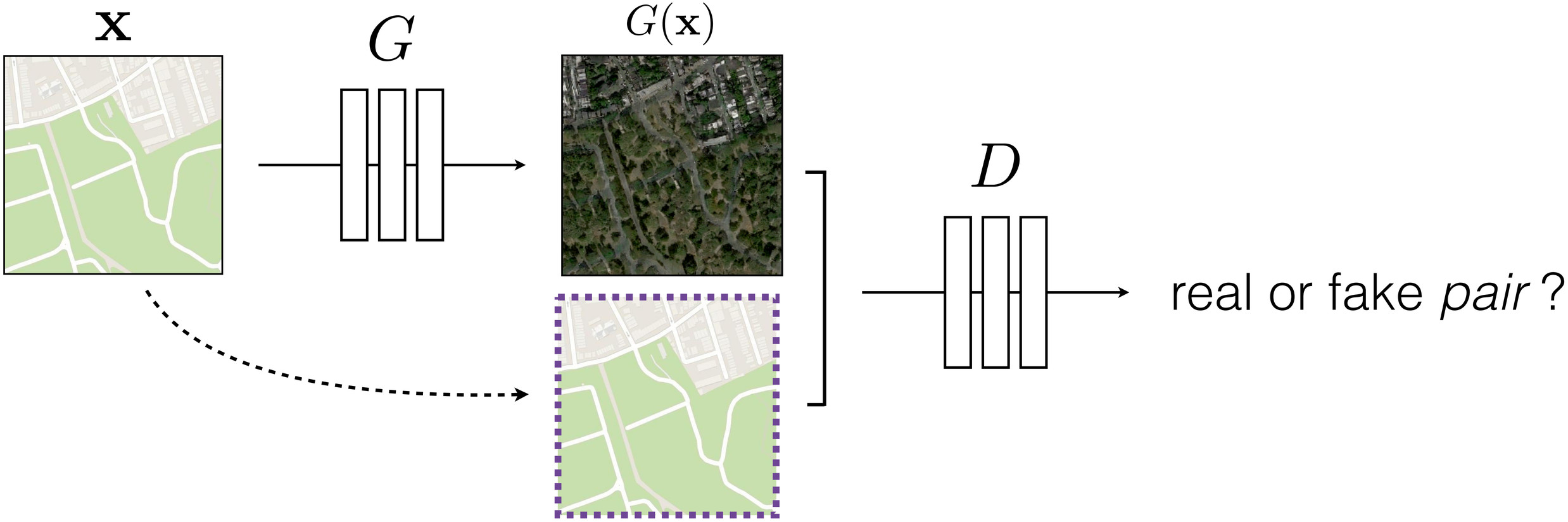
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$



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



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



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

Training Details: Loss function

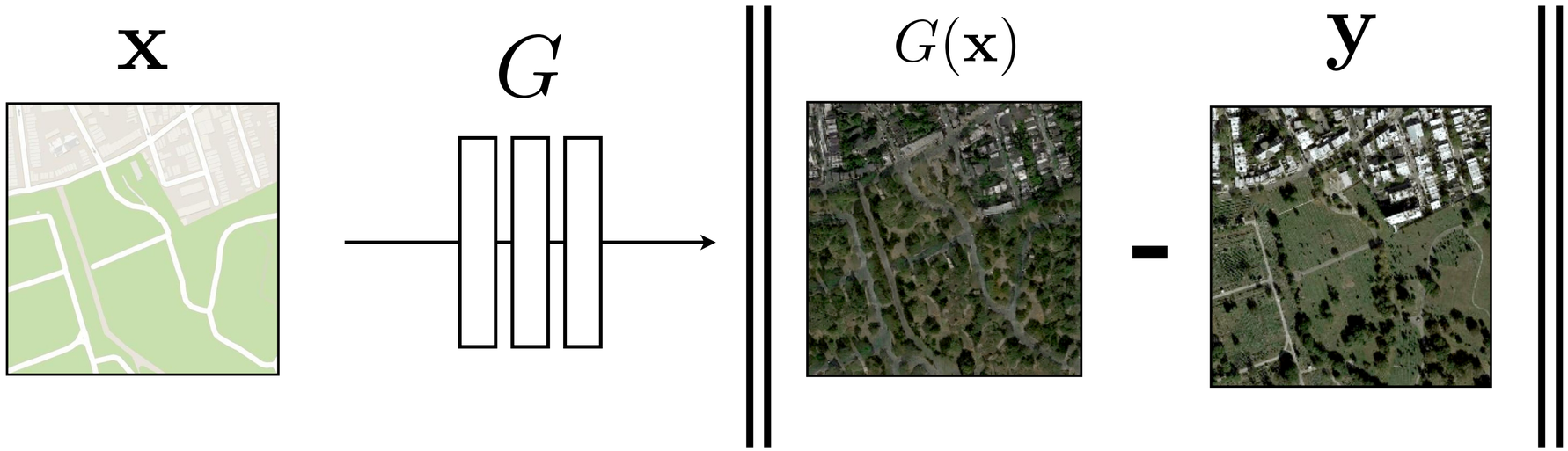
Conditional GAN

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

Training Details: Loss function

Conditional GAN

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

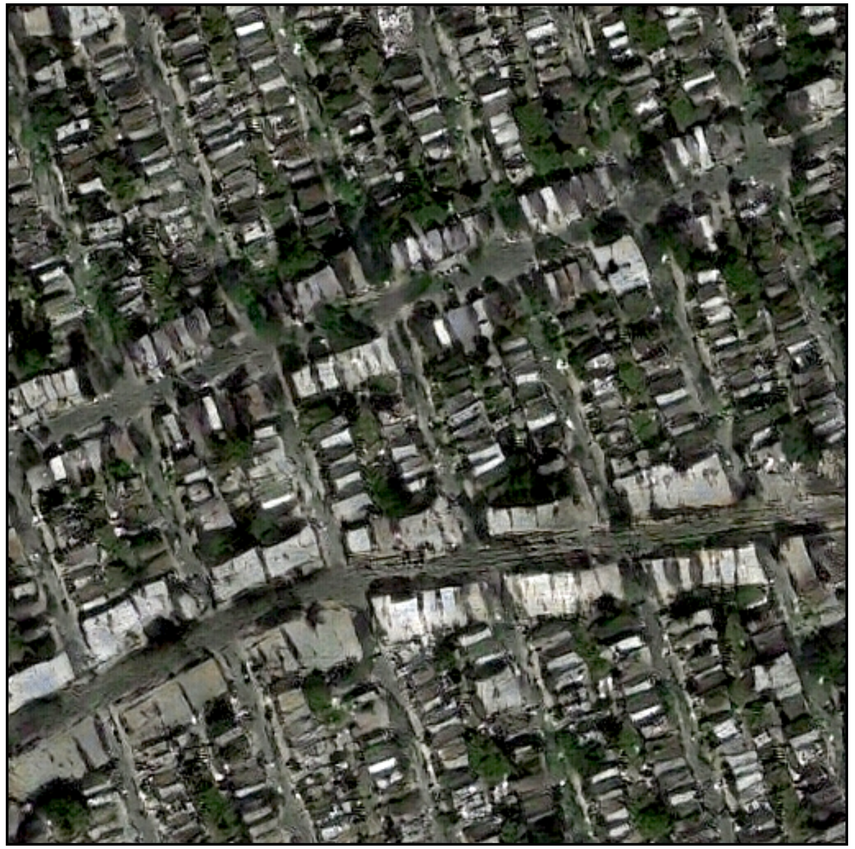


Stable training + fast convergence

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

Input

Output



Groundtruth

Data from
[maps.google.com]

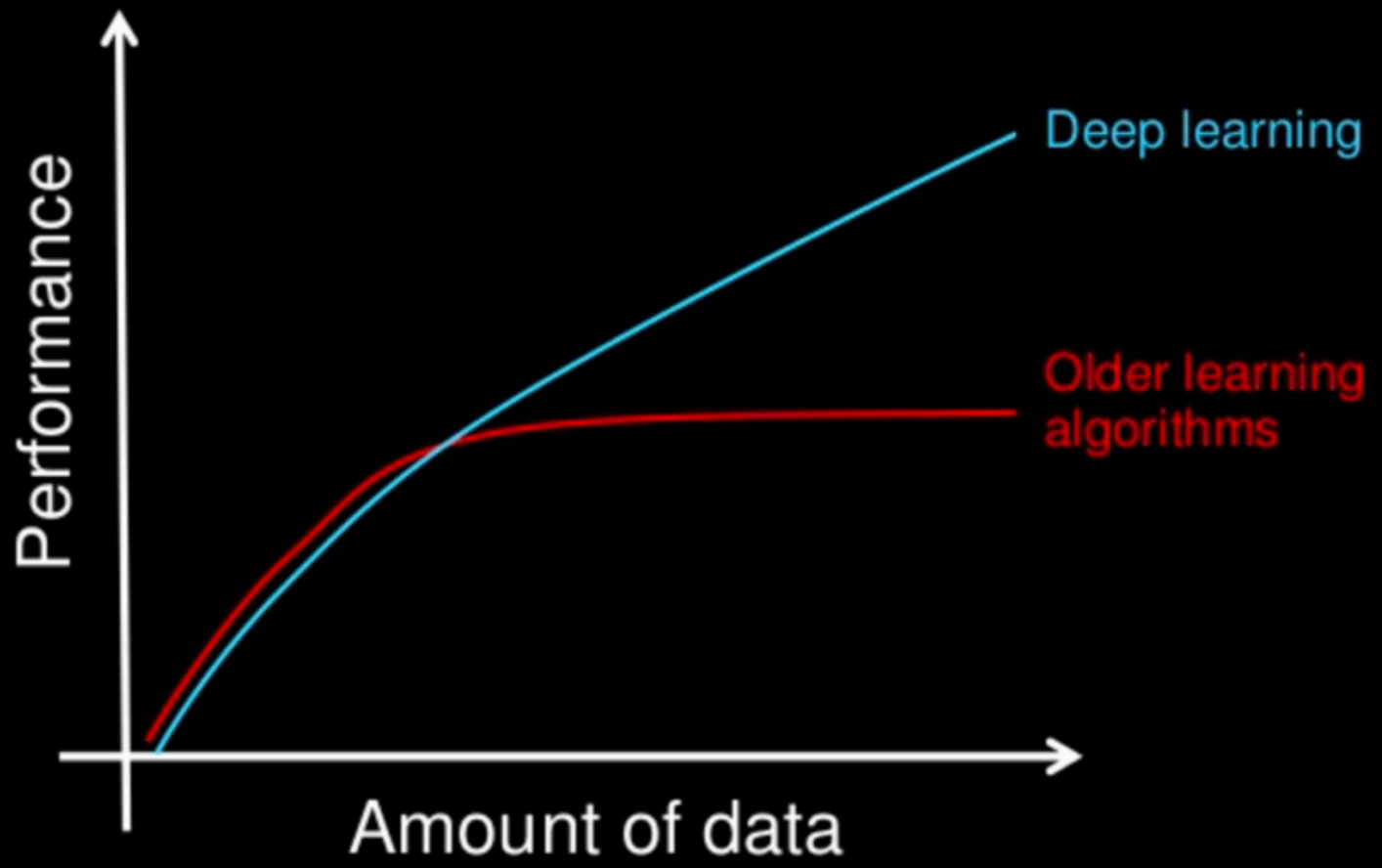
Input

Output

Groundtruth

Data from [maps.google.com]

Why deep learning

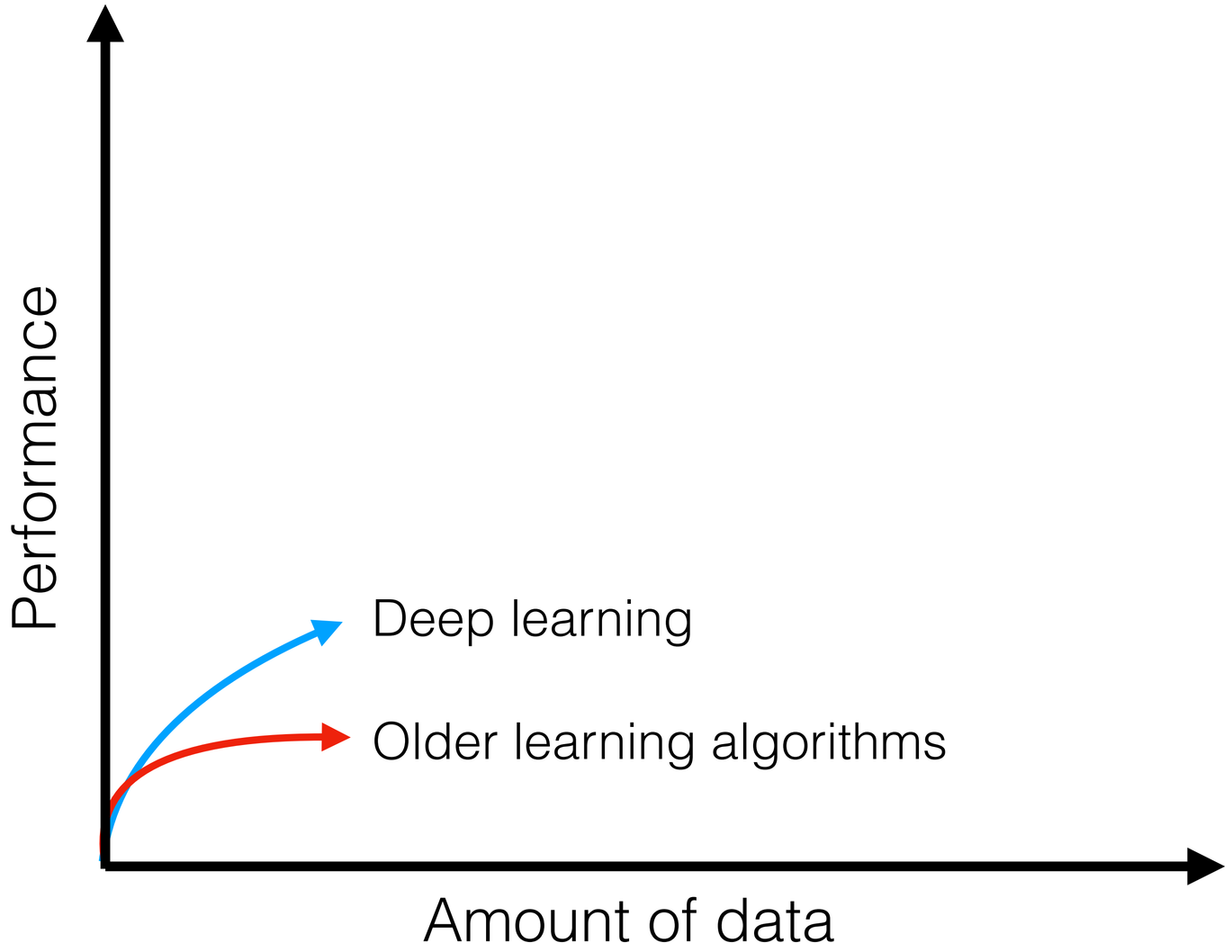


How do data science techniques scale with amount of data?

[Slide credit: Andrew Ng]

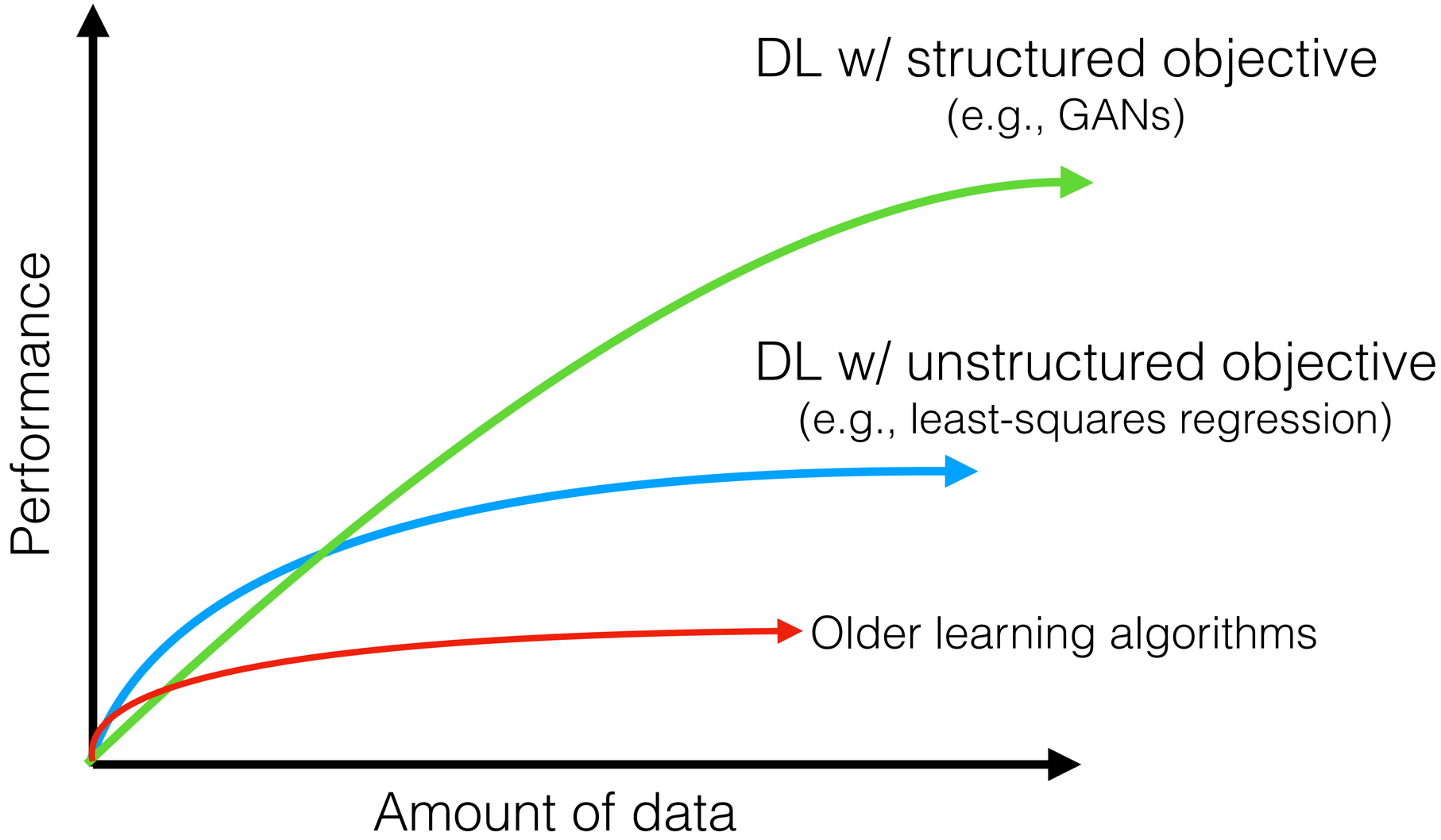
Why structured objectives

(cartoon)

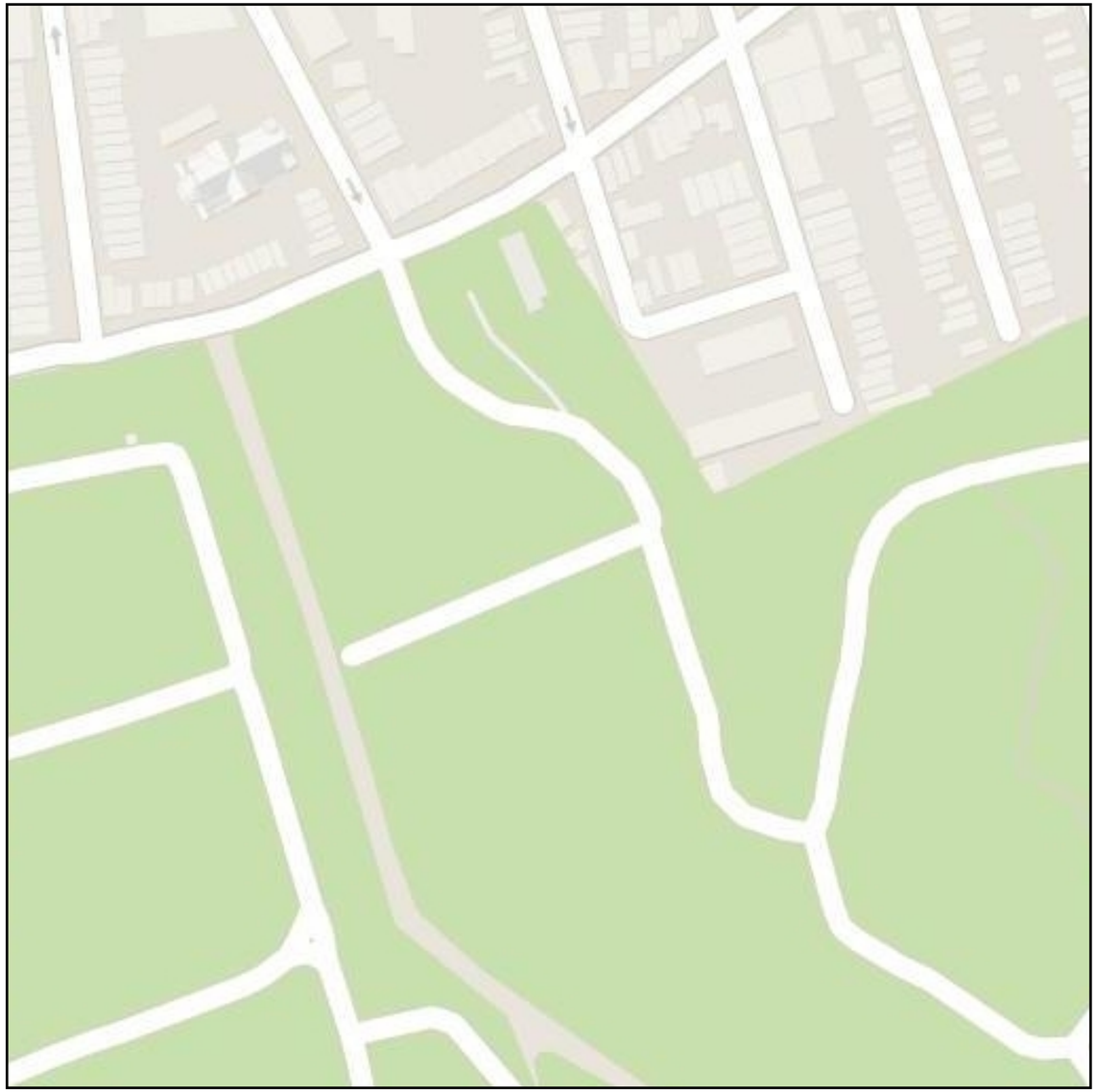


Why structured objectives

(cartoon)



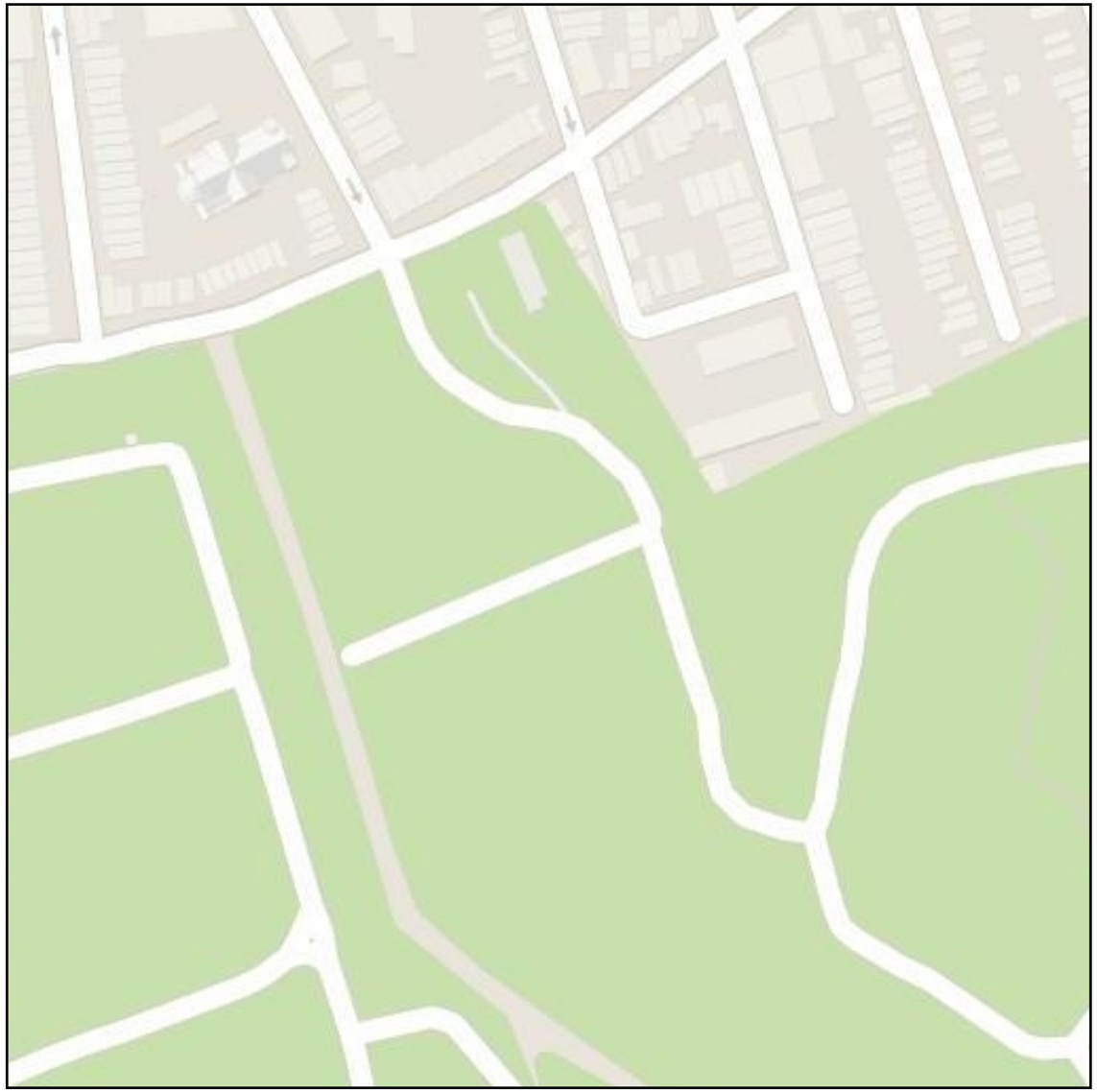
Input



Unstructured prediction (L1)

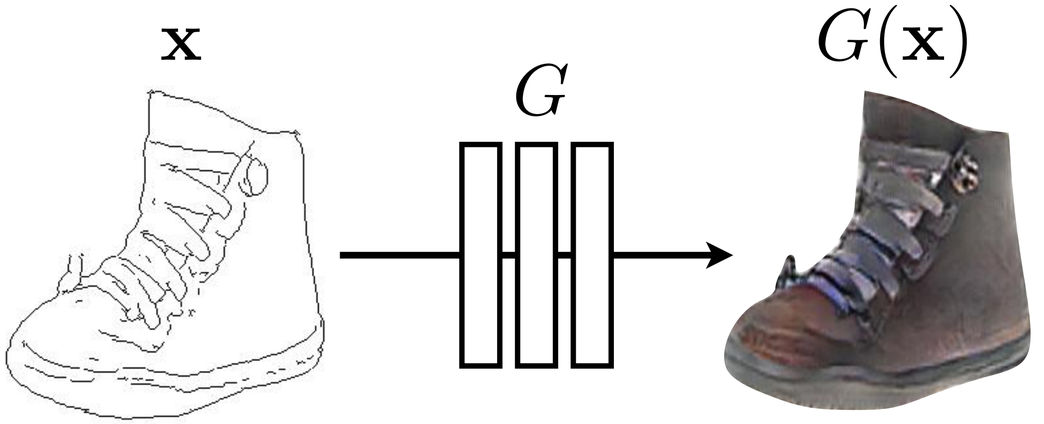
Input

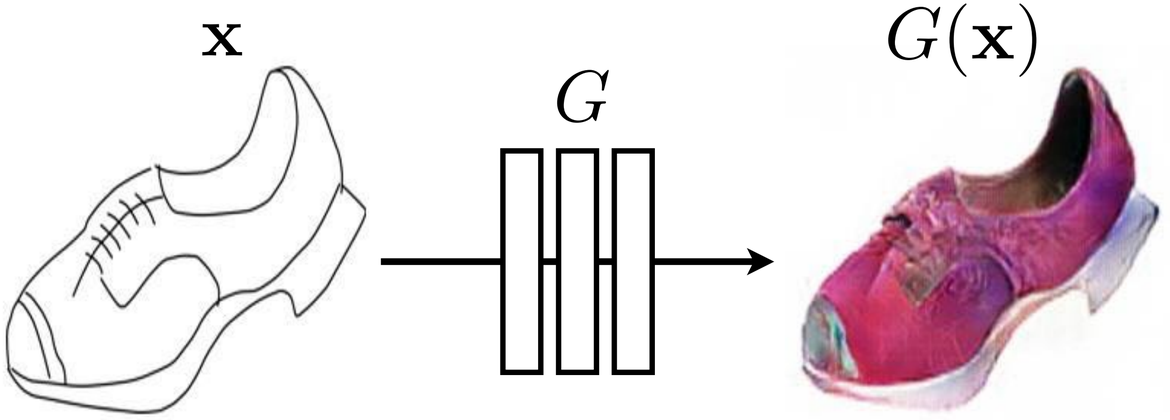
Structured Prediction (cGAN)



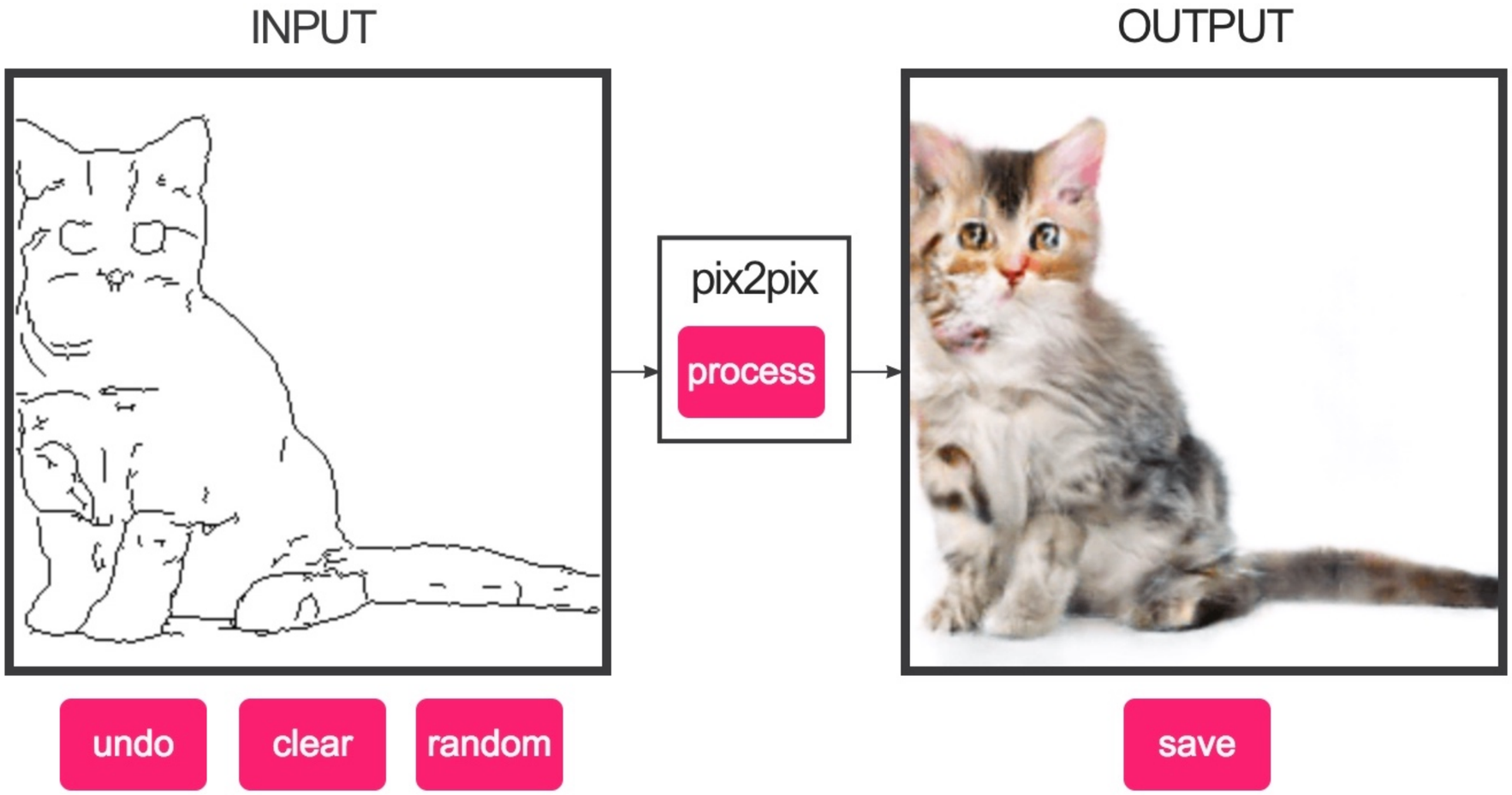
Training data

[HED, Xie & Tu, 2015]

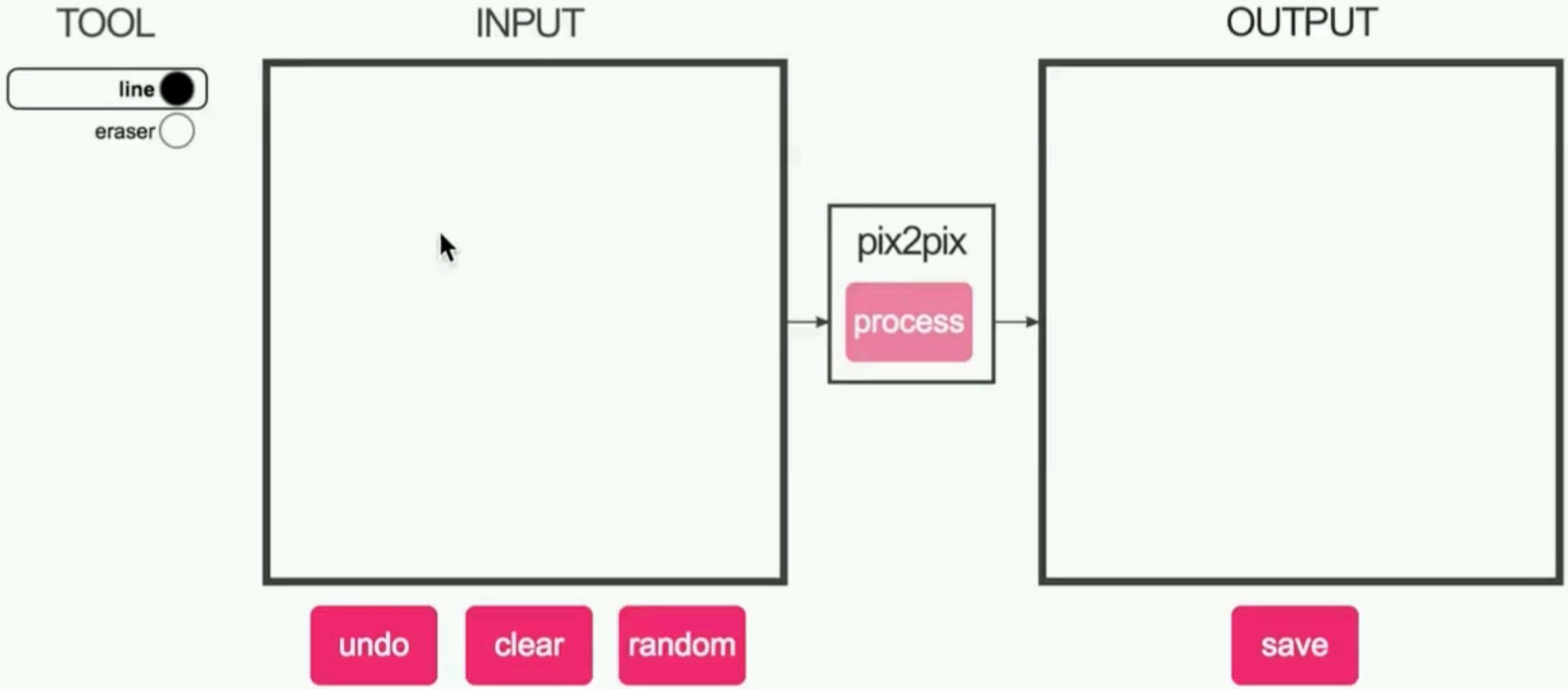


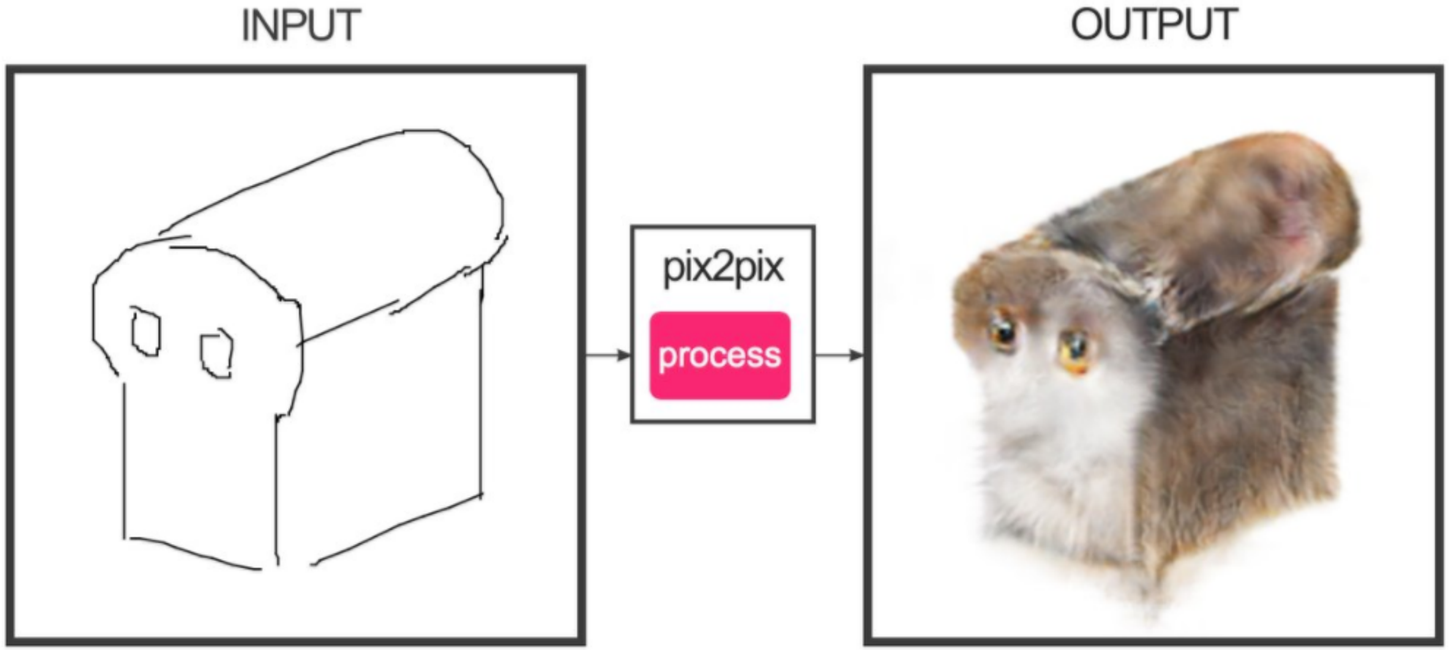


#edges2cats [Chris Hesse]



edges2cats

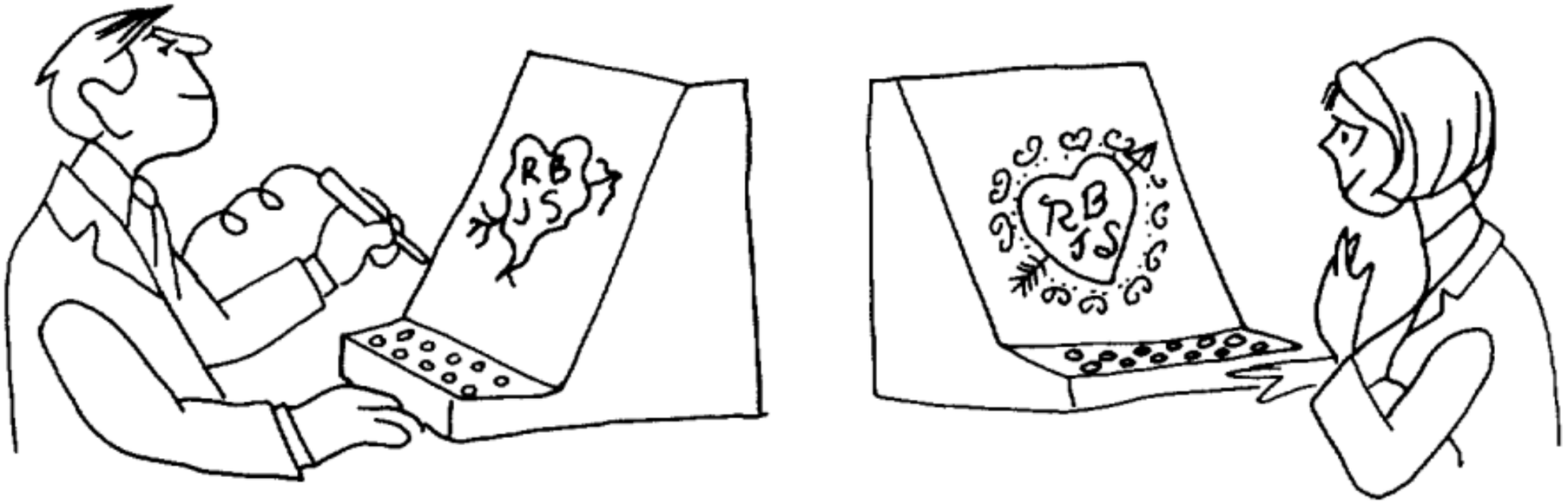

TOOL
line
eraser
INPUT
pix2pix
process
OUTPUT
undo
clear
random
save



Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

1. Image synthesis
2. Structured prediction
3. **Domain mapping**

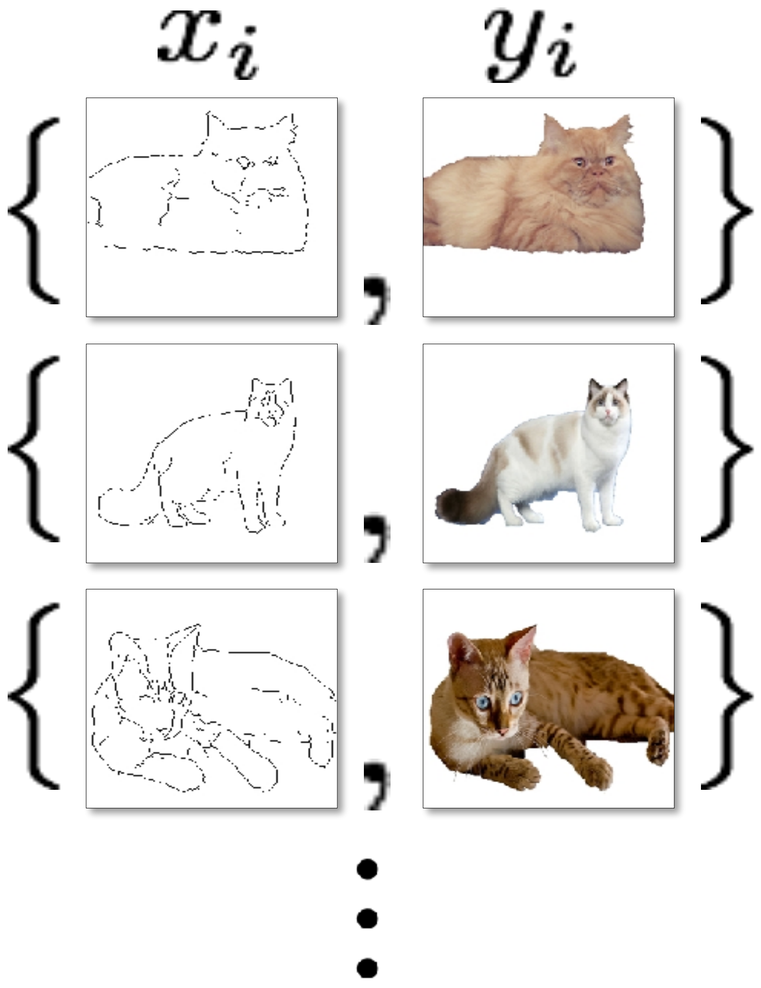


Domain mapping

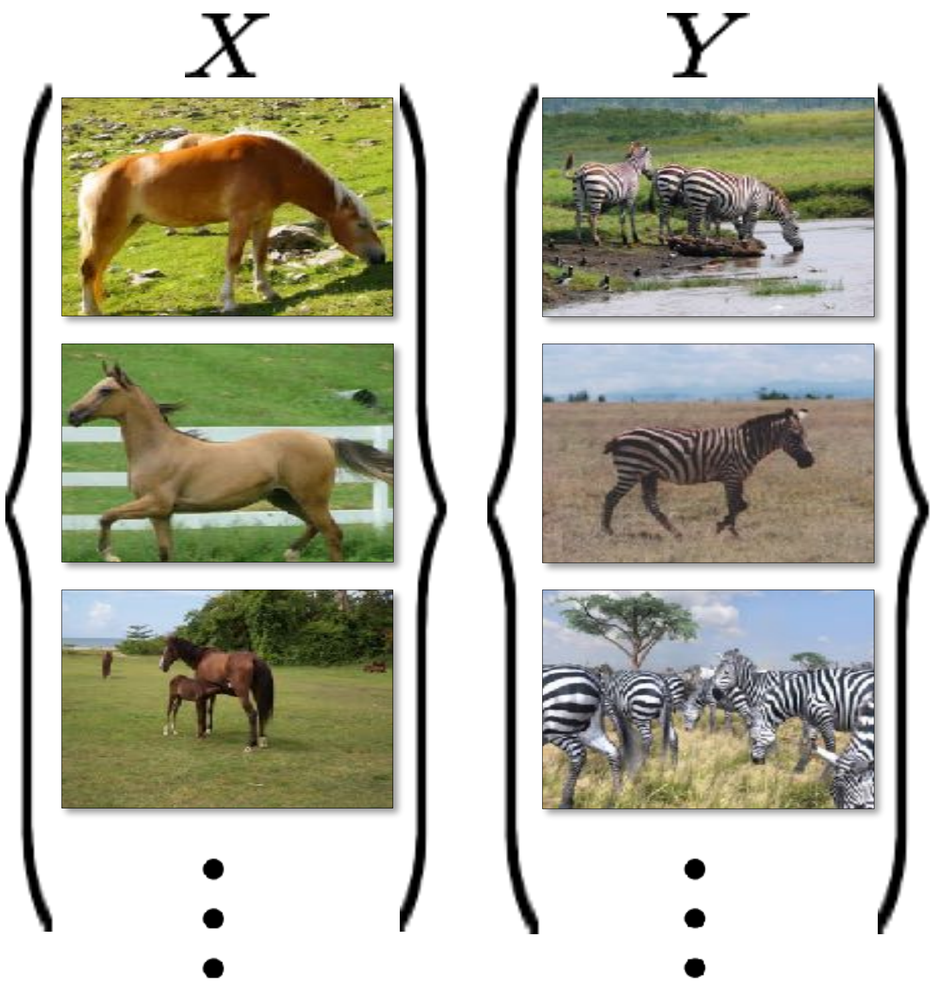
[Includes slides from Jun-Yan Zhu, Taesung Park]

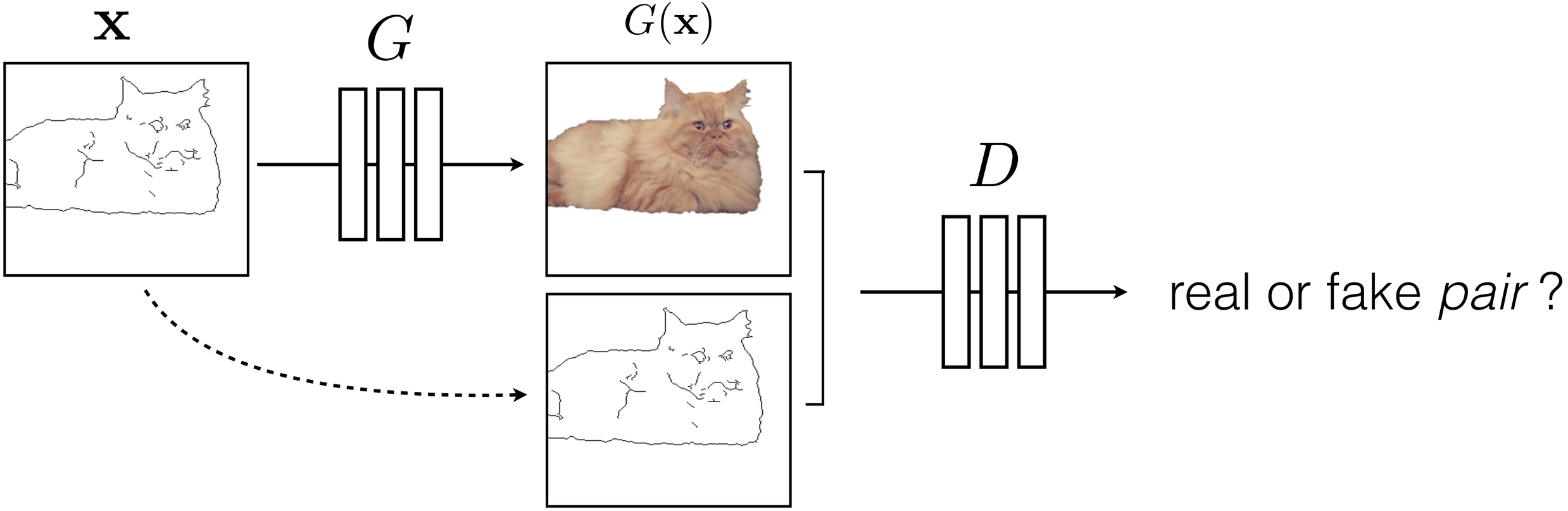
[Cartoon: The Computer as a Communication Device, Licklider & Taylor 1968]

Paired data

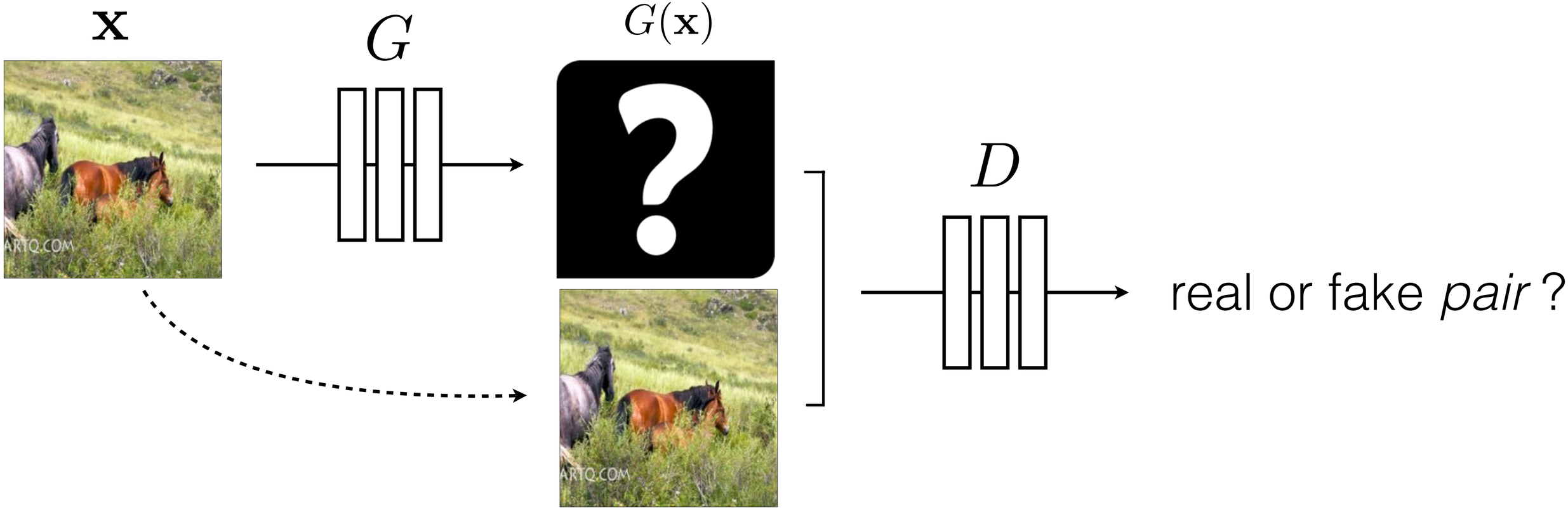


Unpaired data



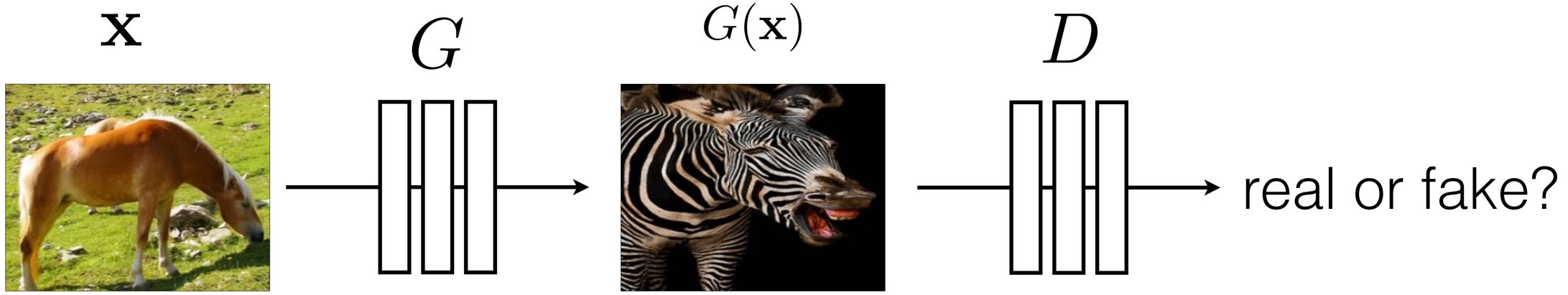


$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;\;]$$



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

No input-output pairs!



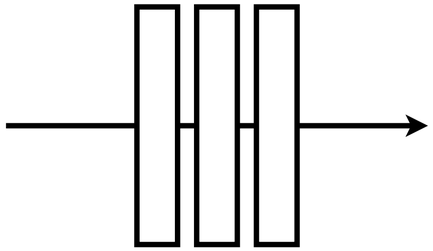
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\;]$$

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

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

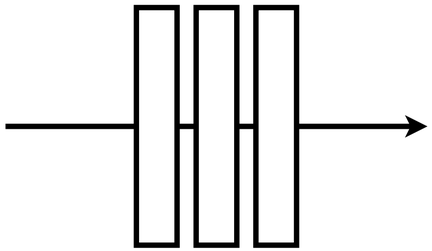
$\mathbf{x}$

G

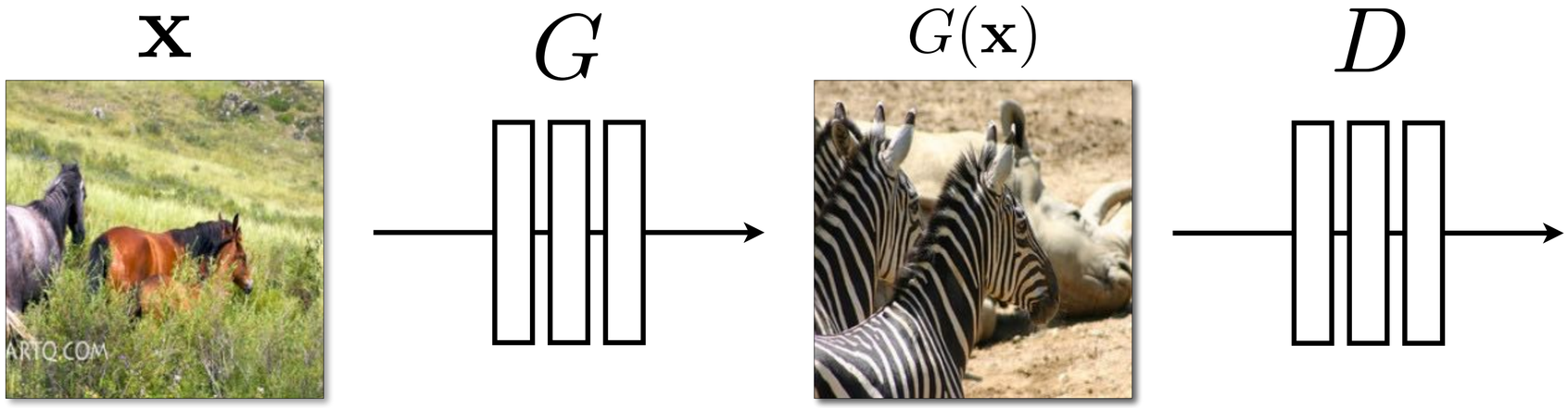


$G(\mathbf{x})$

D

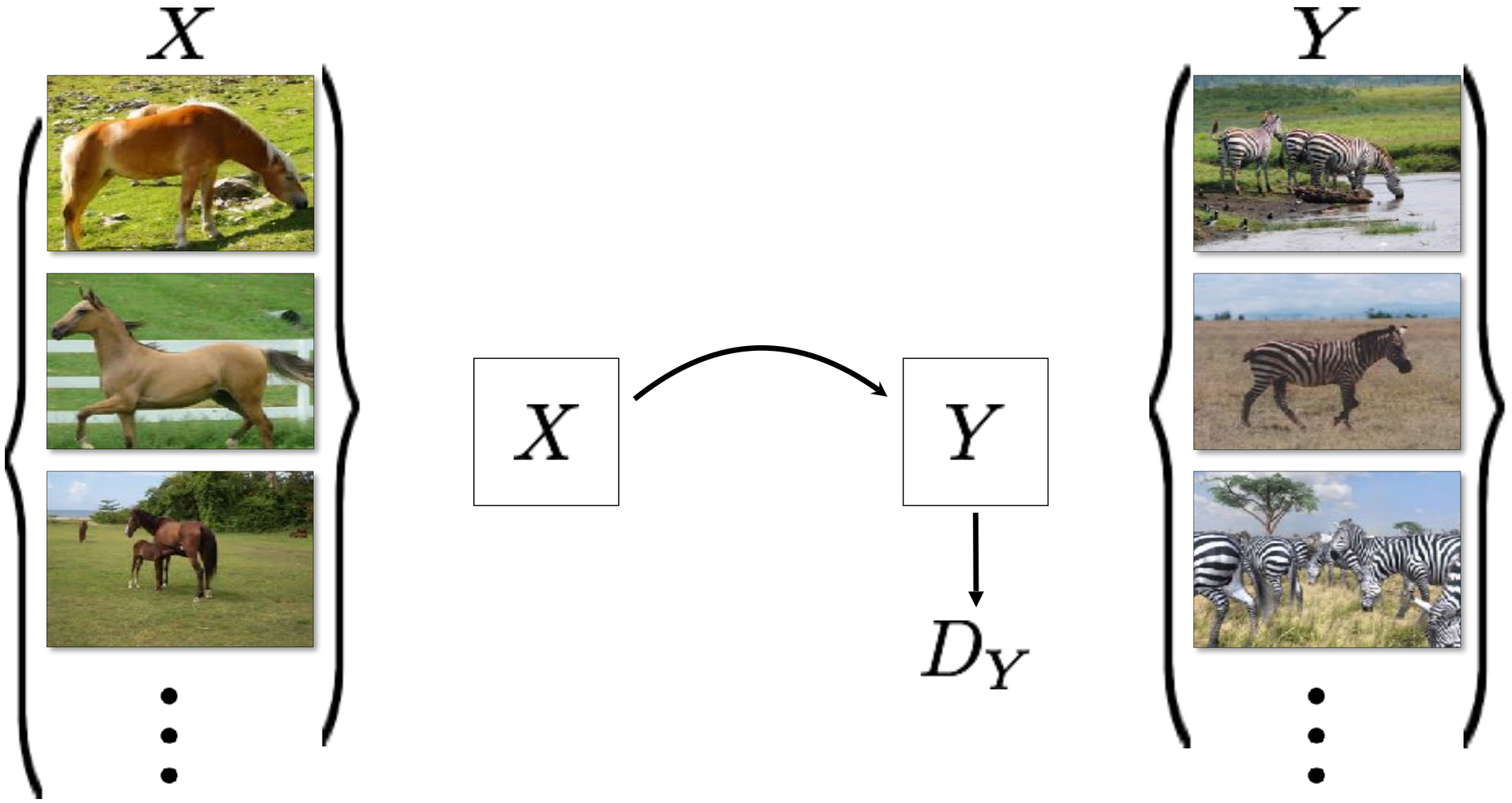


Real!



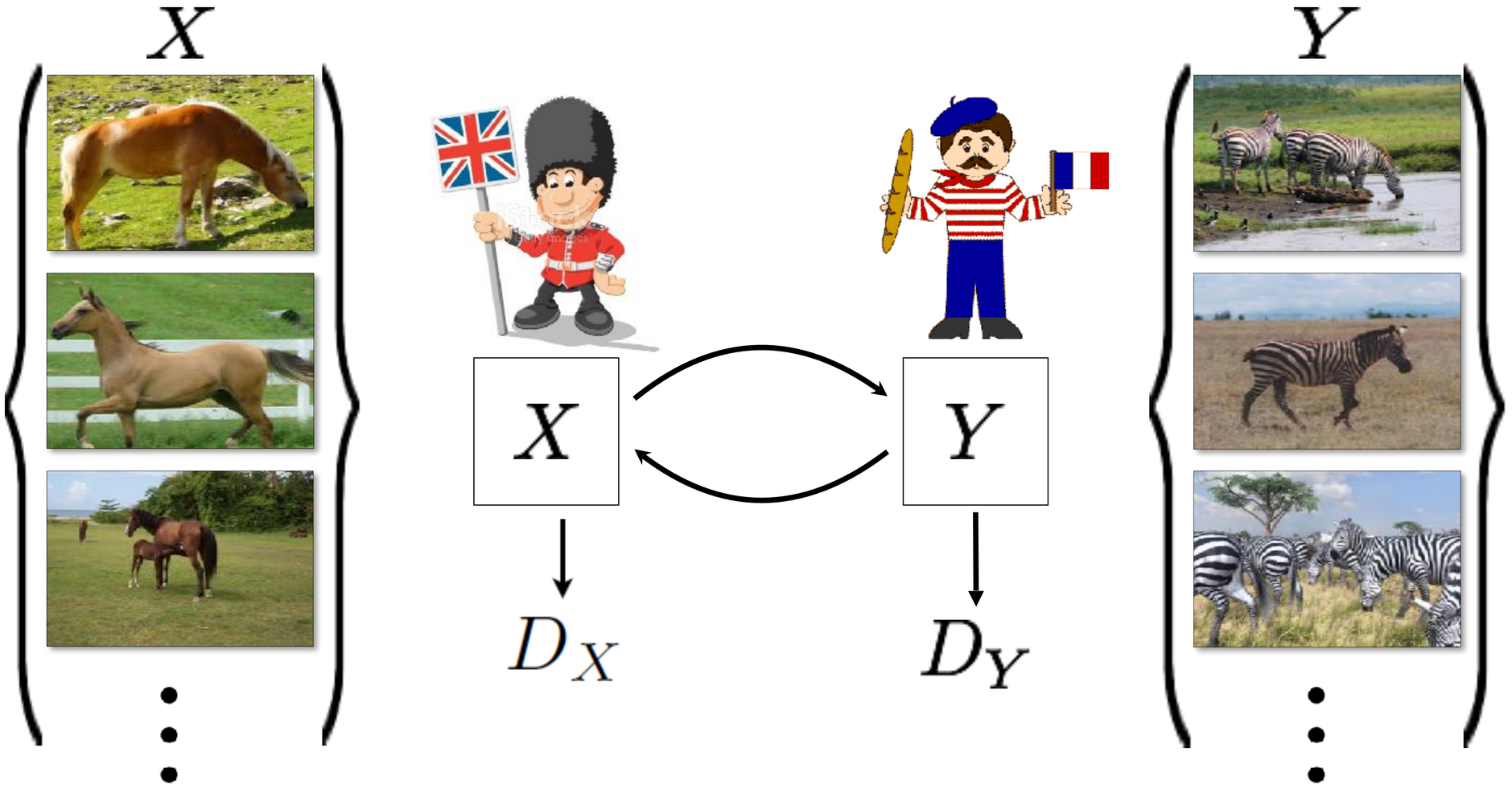
Nothing to force output to correspond to input

CycleGAN, or there and back aGAN

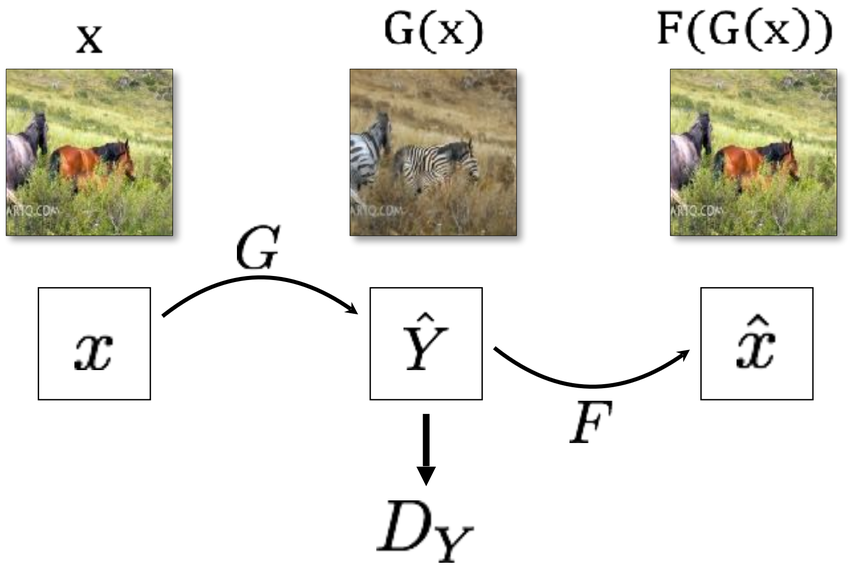


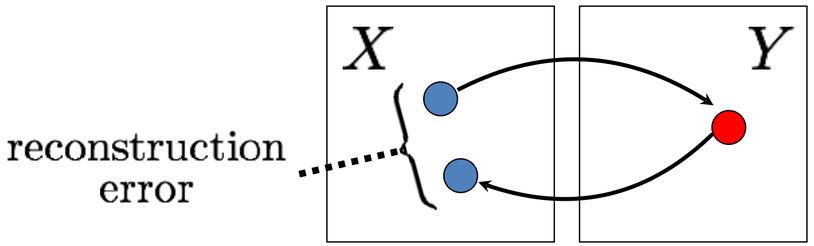
[Zhu*, Park* et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

CycleGAN, or there and back aGAN



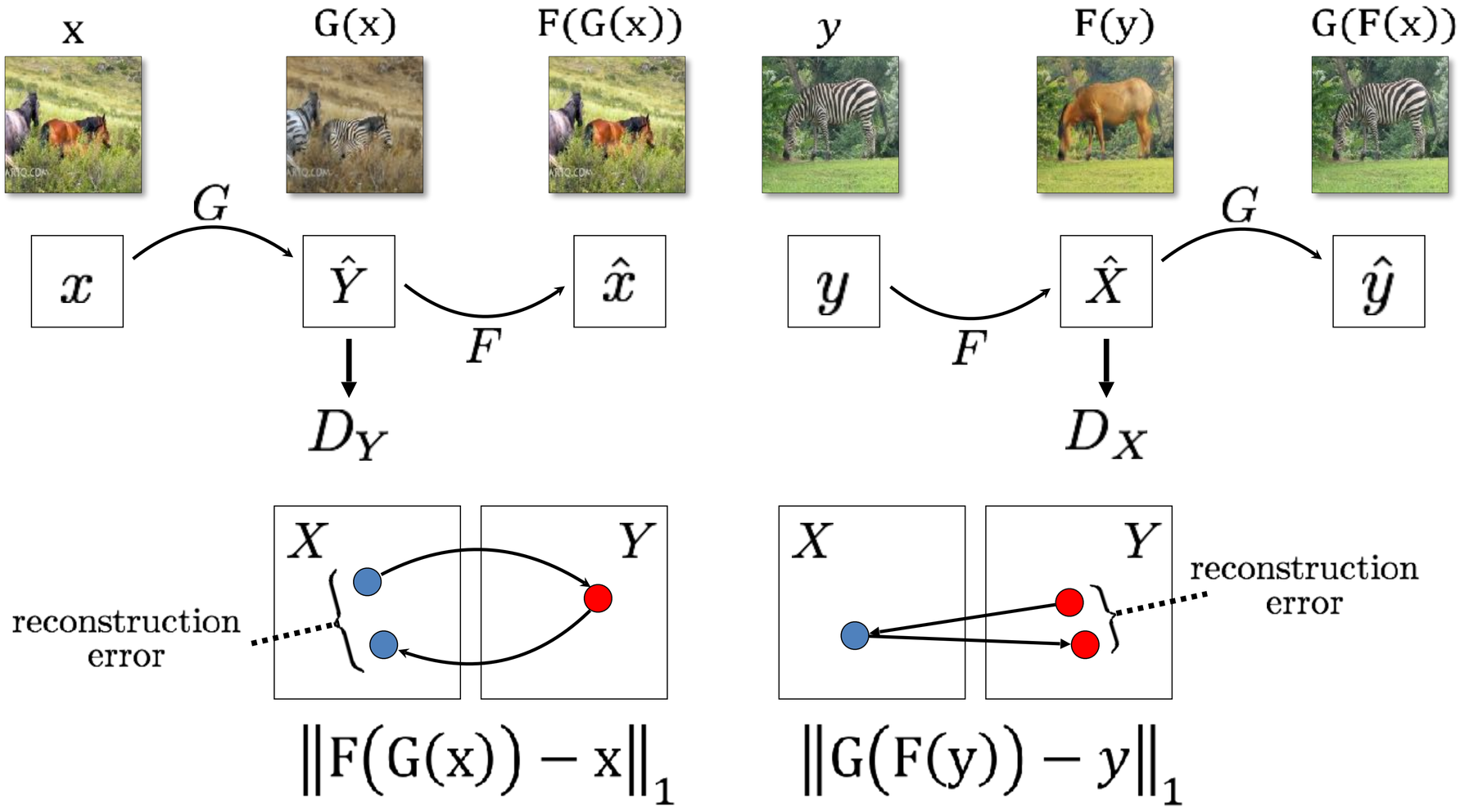
Cycle Consistency Loss





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

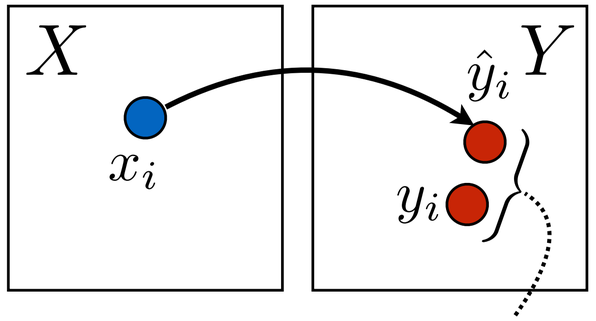
Cycle Consistency Loss



Paired translation

Training data

Objective



regression error

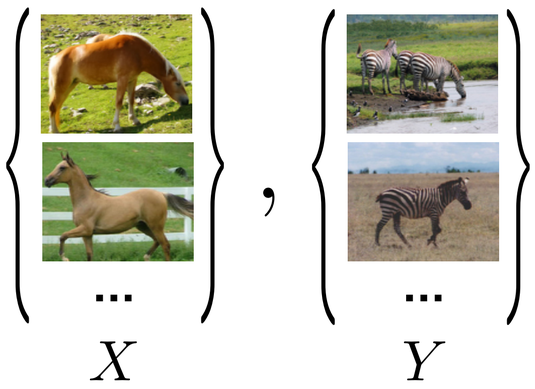
Input

Result

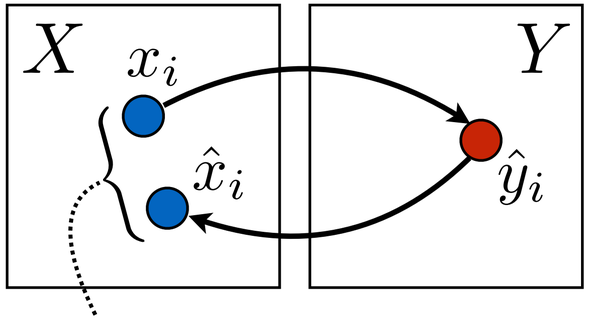
[“pix2pix”, Isola, Zhu, Zhou, Efros, 2017]

Unpaired translation

Training data



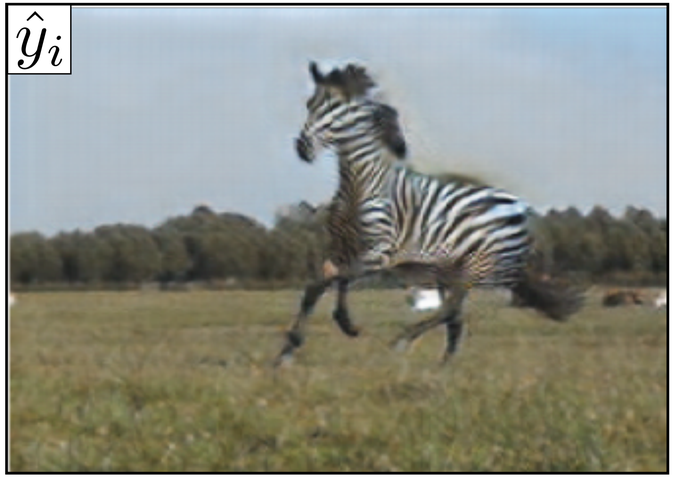
Objective



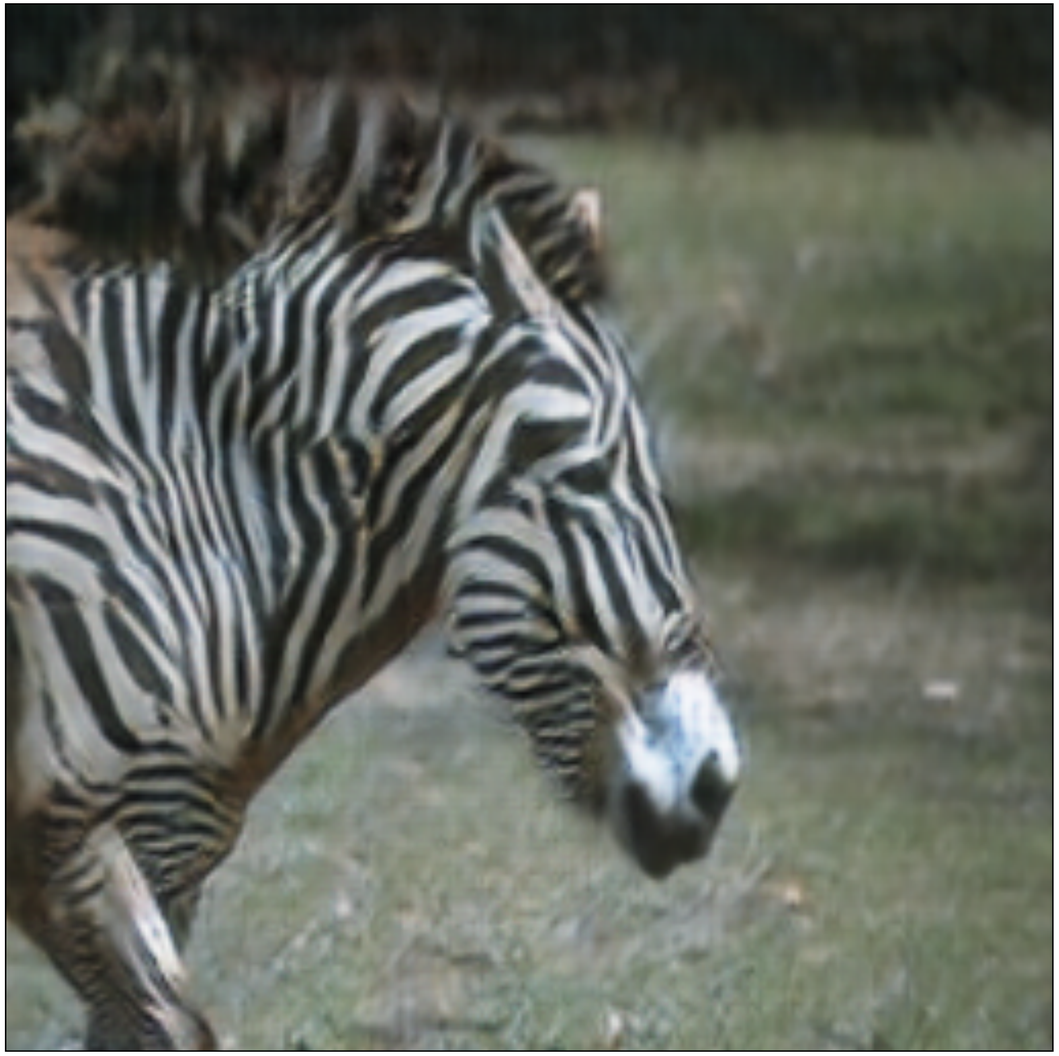
cycle-consistency error

Input

Result



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

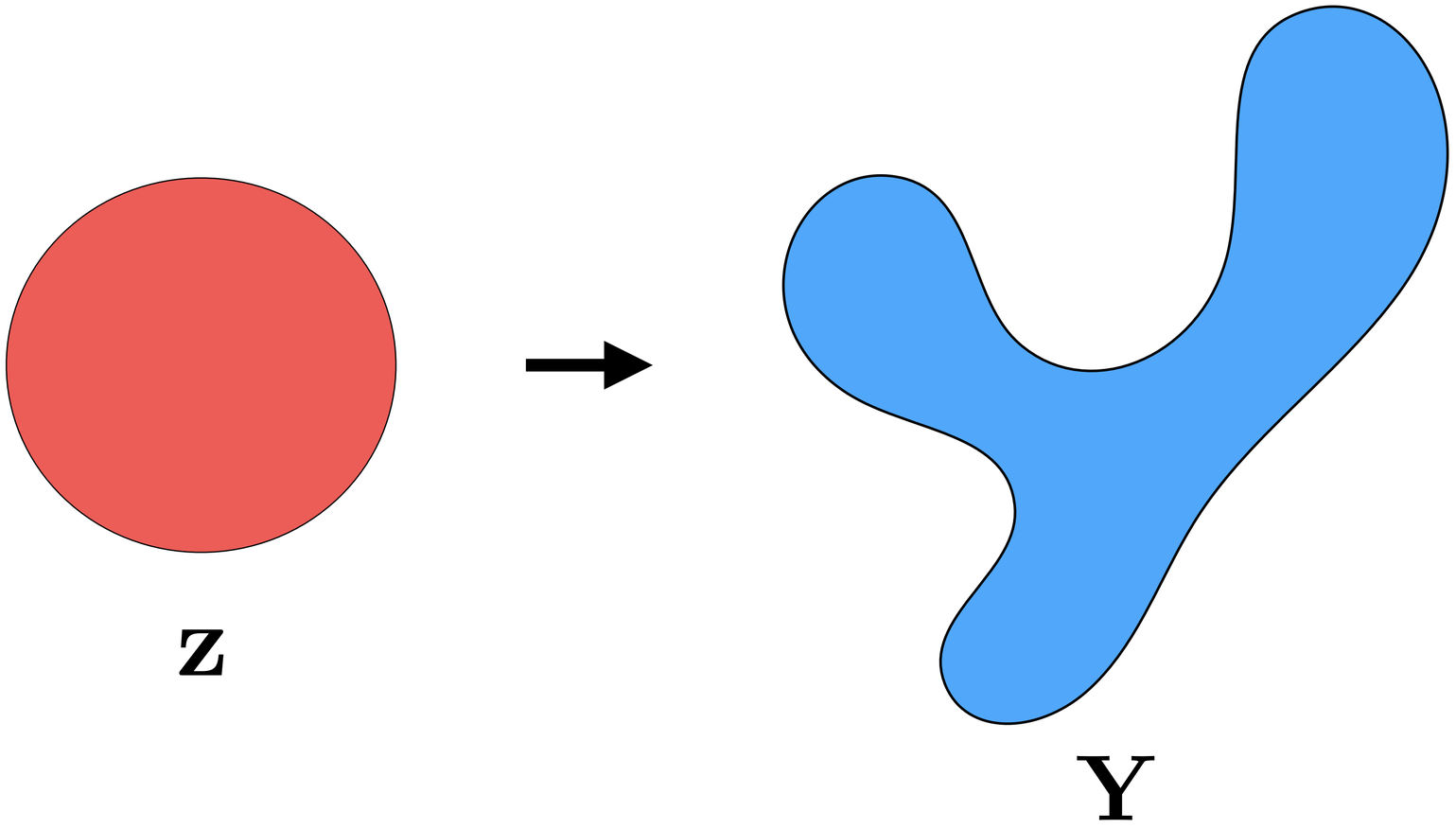



Input
Monet
Van Gogh
Cezanne
Ukiyo-e

GANs

Gaussian

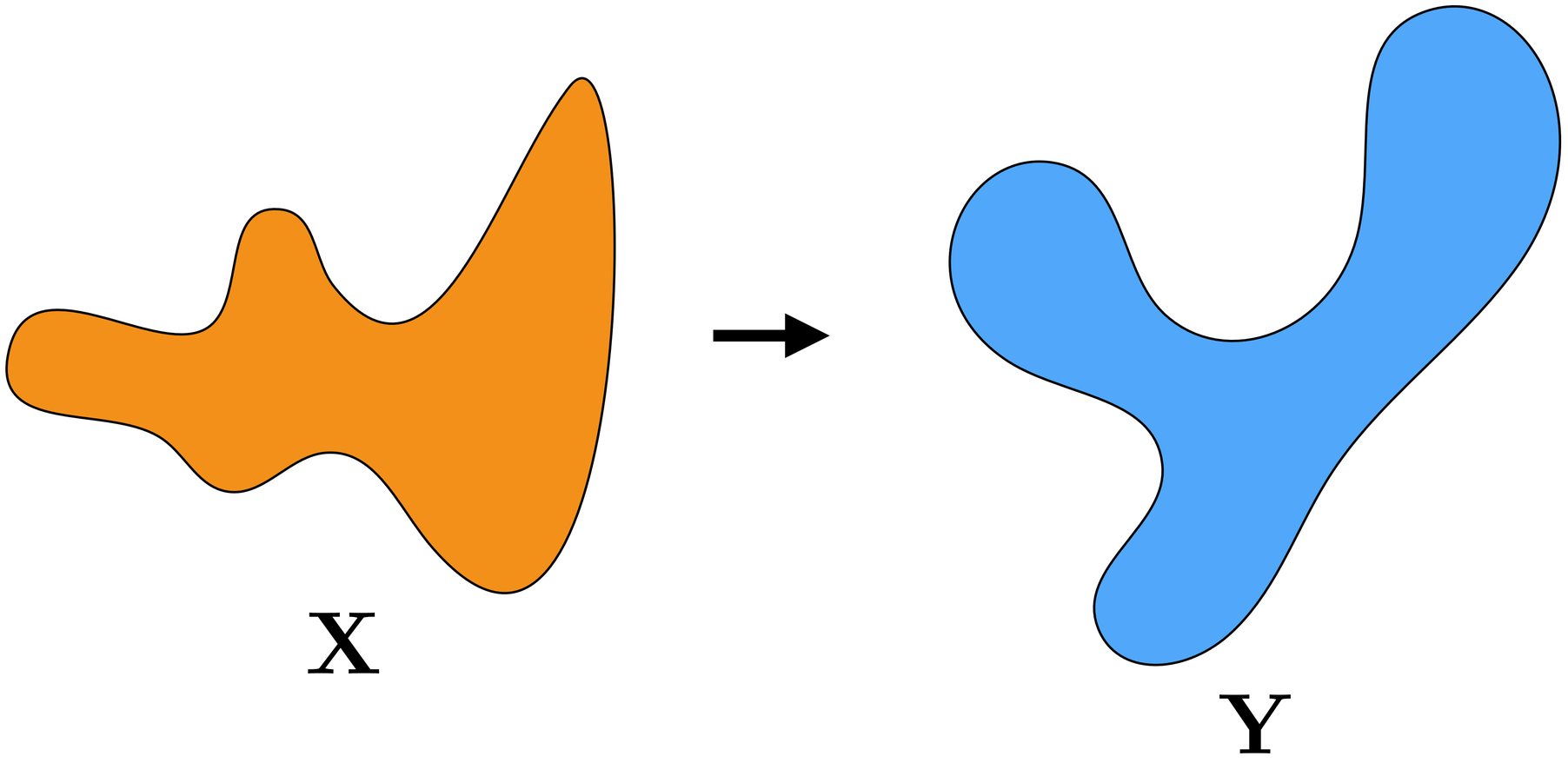
Target distribution



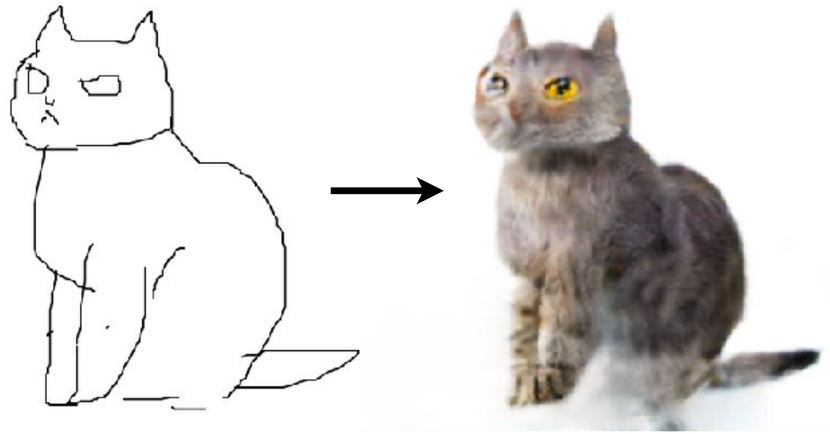
CycleGAN

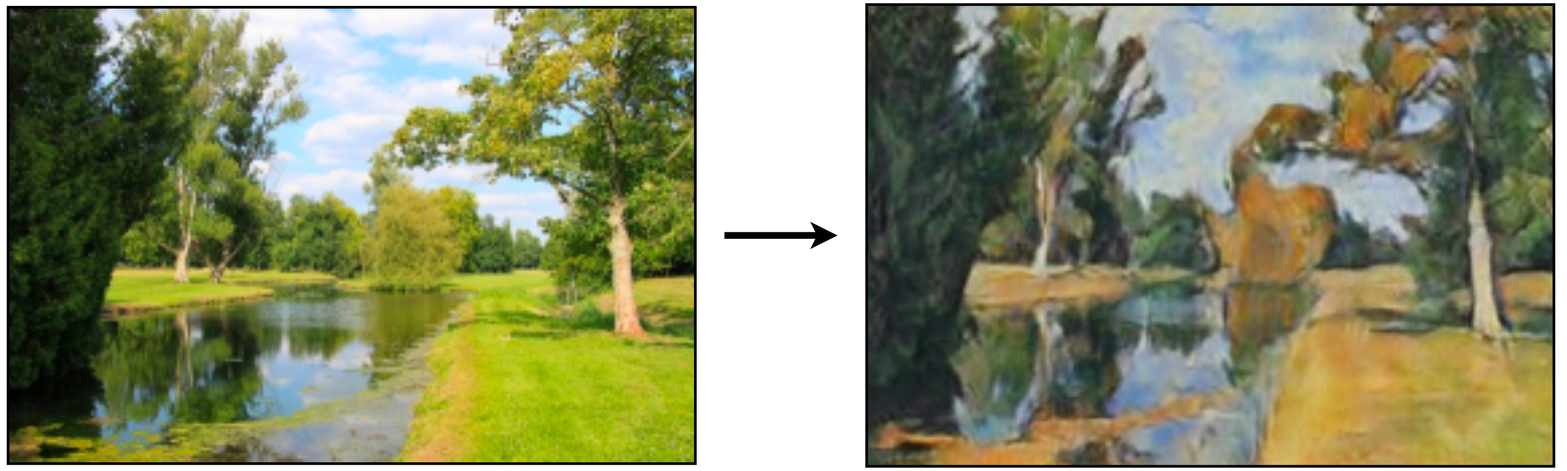
Horses

Zebras



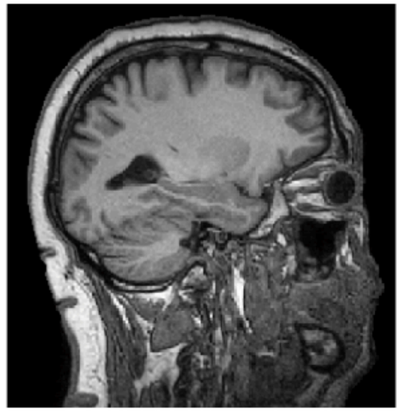
What would it look like if…?





What would it look like if…?

MRI



CT

[Wolterink et al, 2017]

Sim

“Real”

[Hoffman et al, 2018]