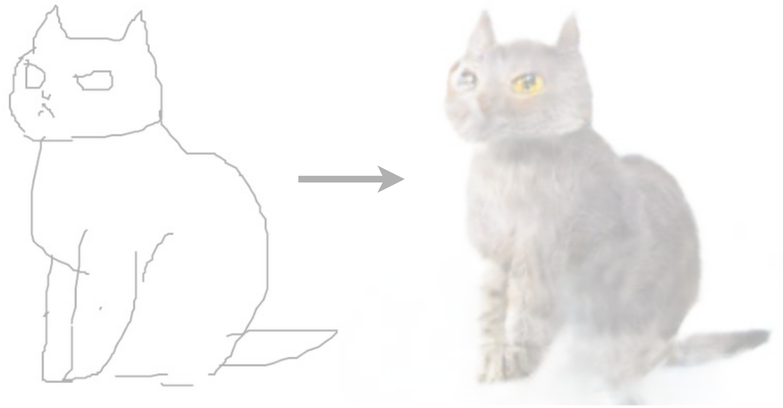
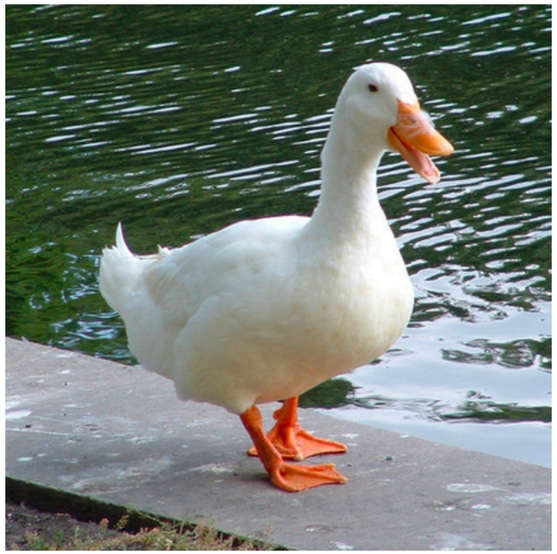
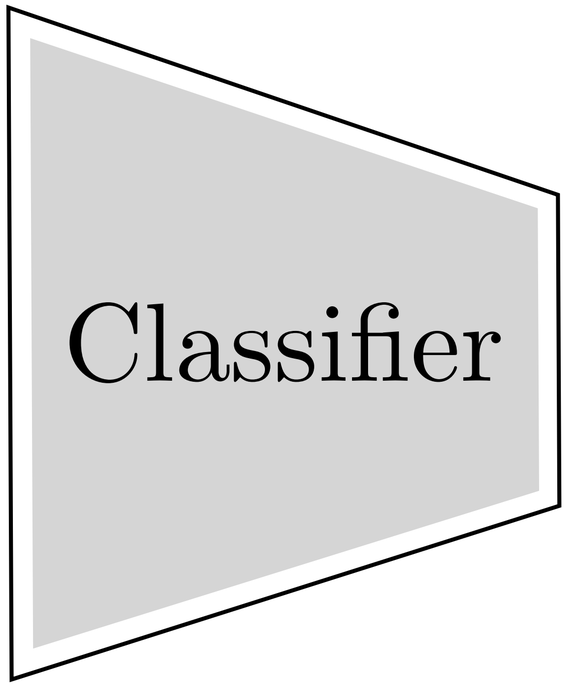


Lecture 20

Image Synthesis

Analysis



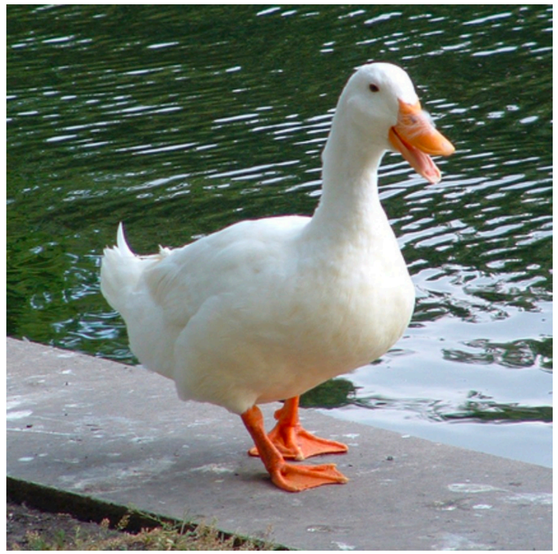


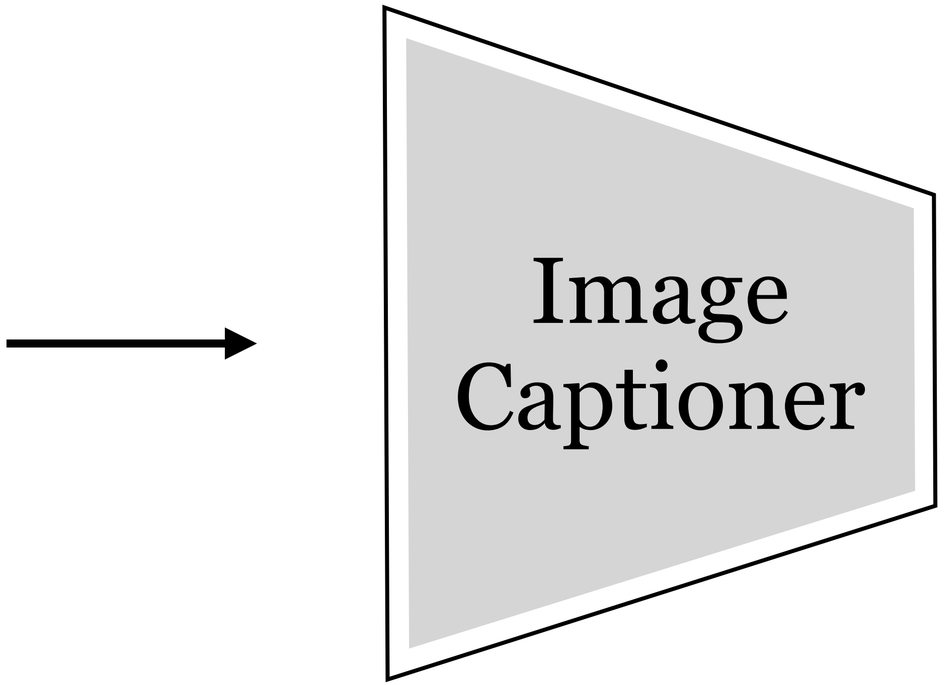
“Duck”

image **x**

label y

Analysis



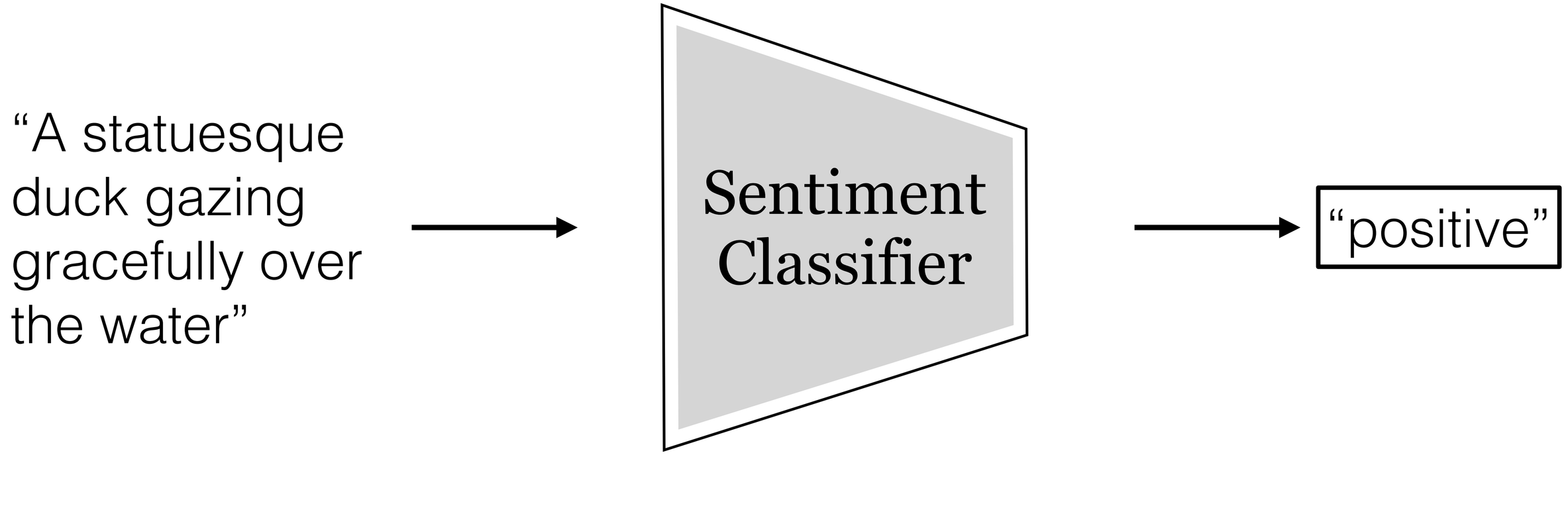


"A large duck standing by the river"

image **x**

caption **y**

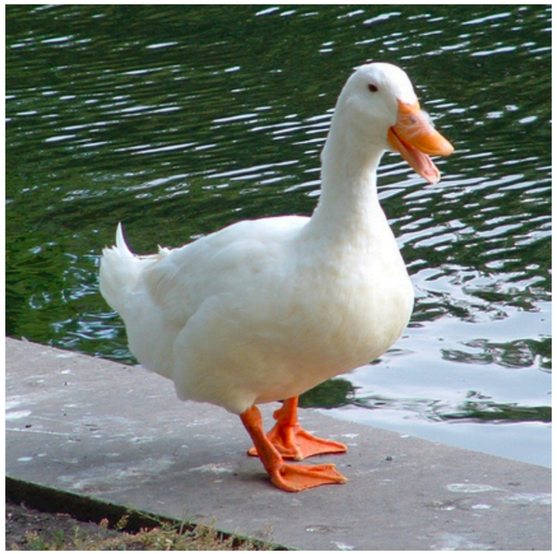
Analysis

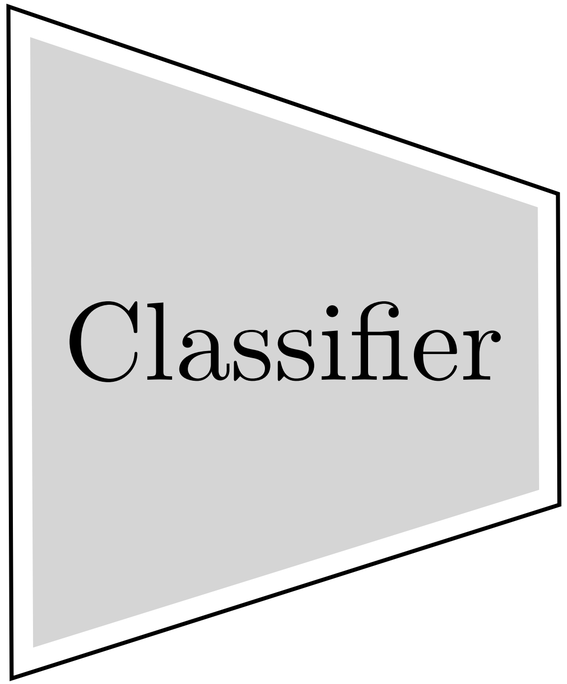


sentence **x**

sentiment y

Analysis





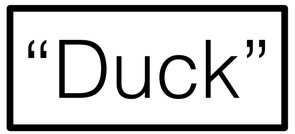
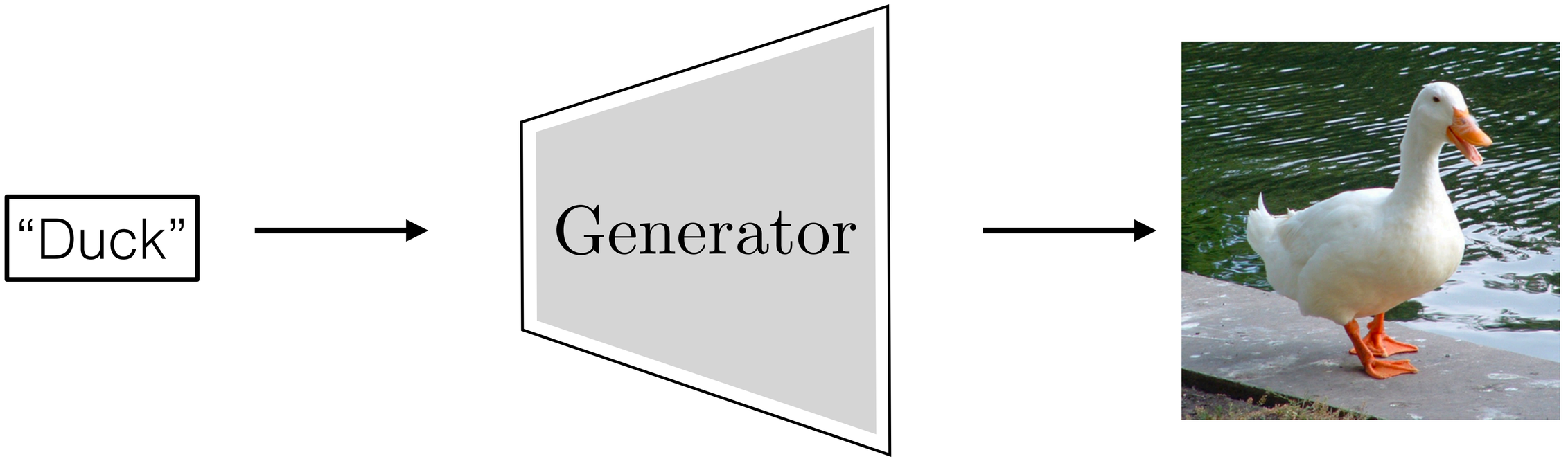


image **x**

label y

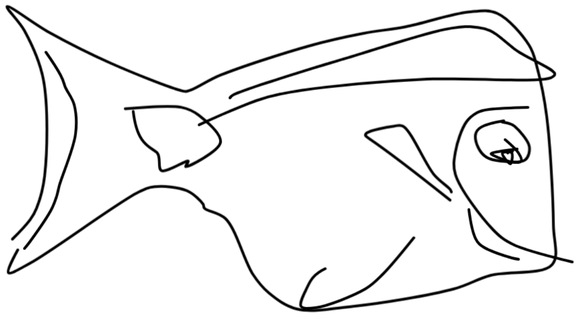
Synthesis

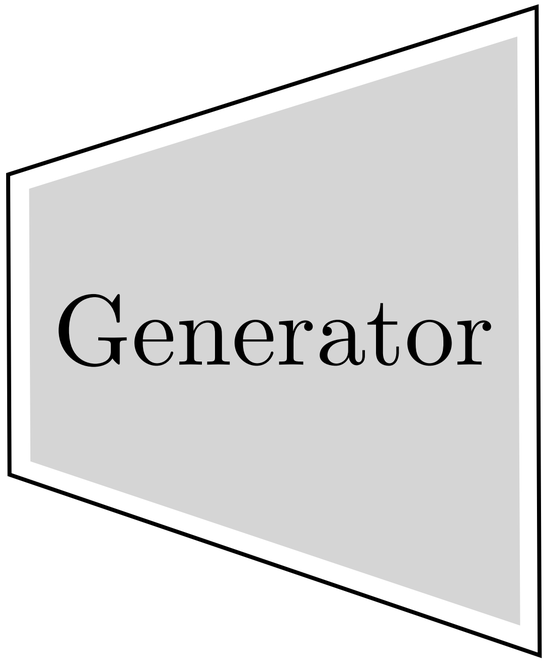


label y

image **x**

Synthesis



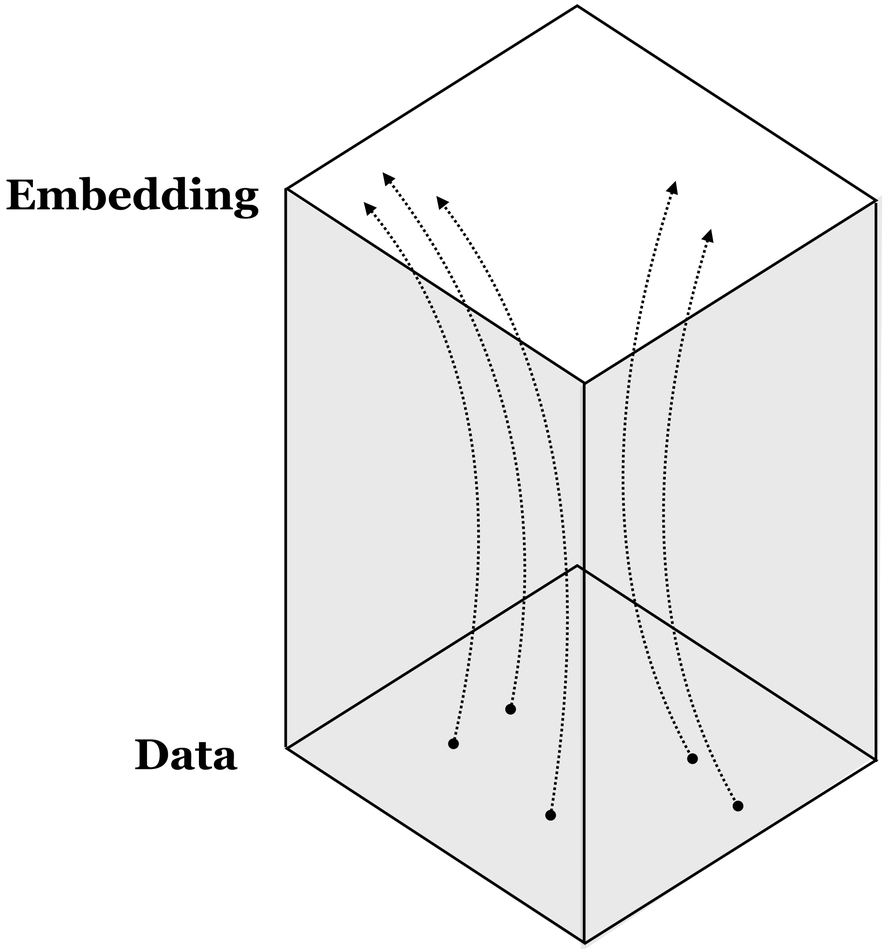


User sketch

Photo

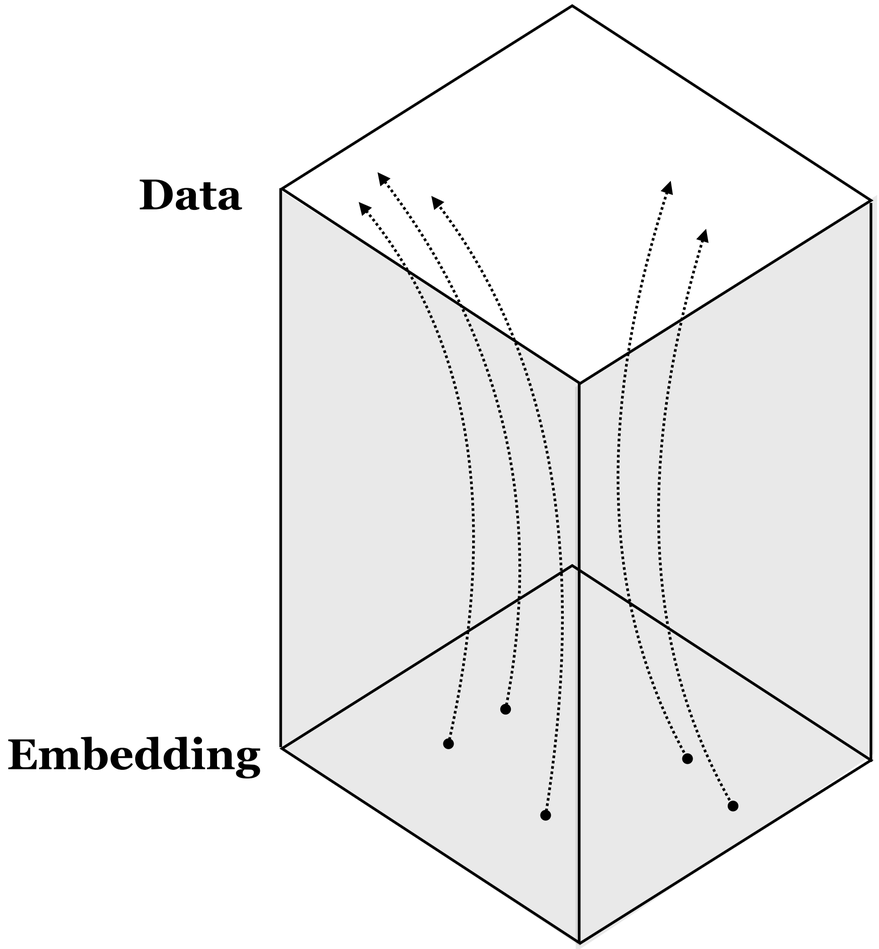
Deep nets are data transformers

- Deep nets transform datapoints, layer by layer
- Each layer is a different *representation* of the data



Deep nets are data transformers

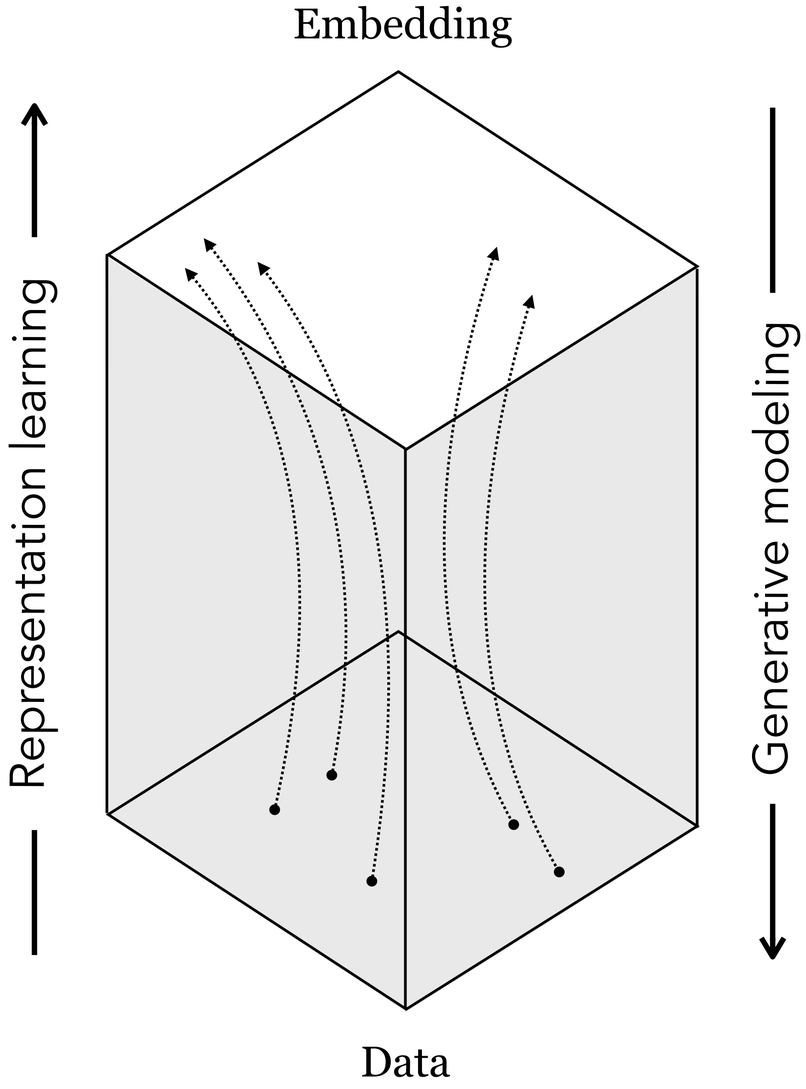
- Deep nets transform datapoints, layer by layer
- Each layer is a different *representation* of the data



Generative modeling vs Representation learning

Representation learning:
mapping data to abstract representations (analysis)

Generative modeling:
mapping abstract representations to data (*synthesis*)



What can you do with generative models?

1. Image synthesis
2. Representation learning
3. Data translation

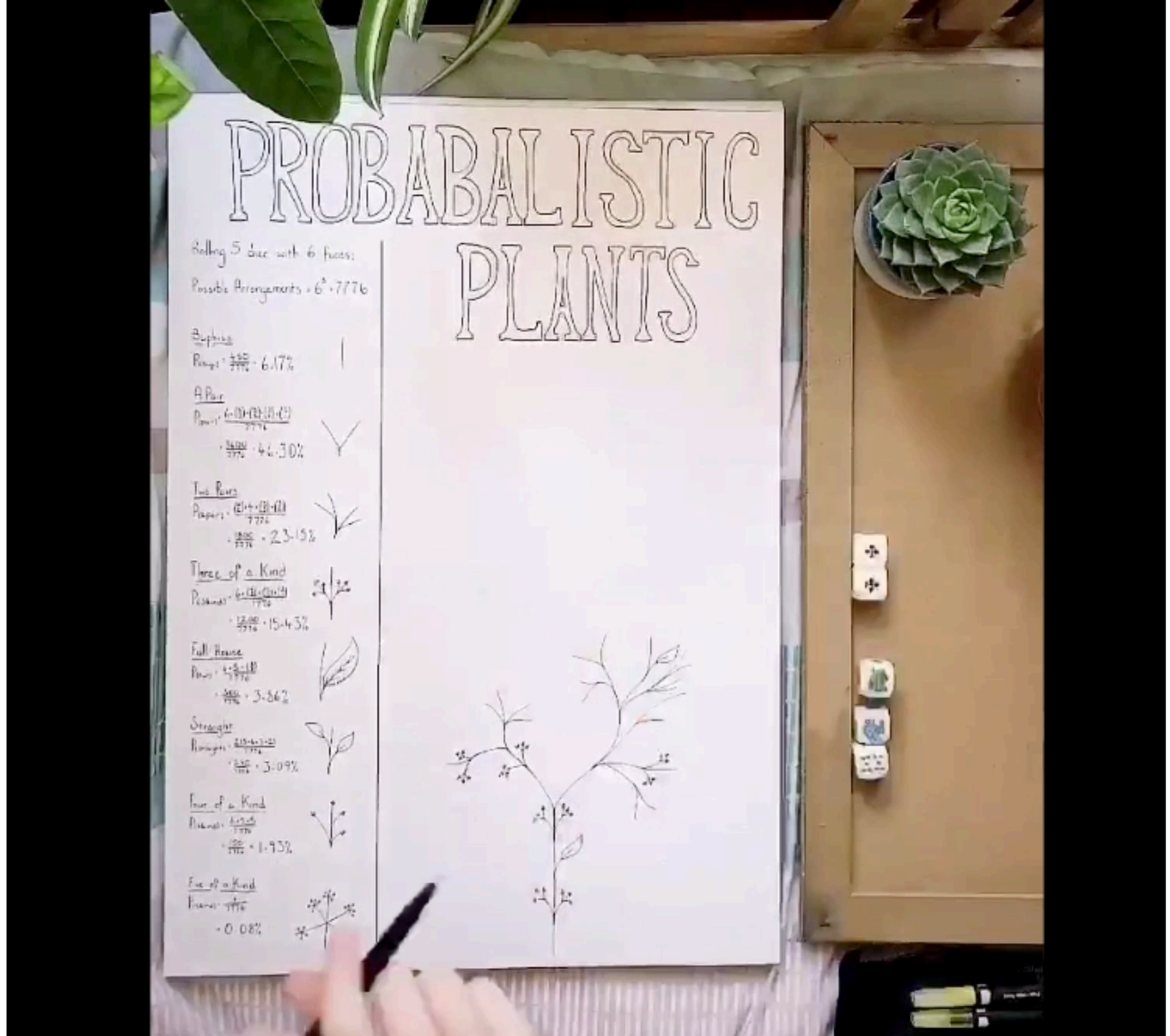
1. **Image synthesis**
2. Representation learning
3. Data translation

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

Image synthesis

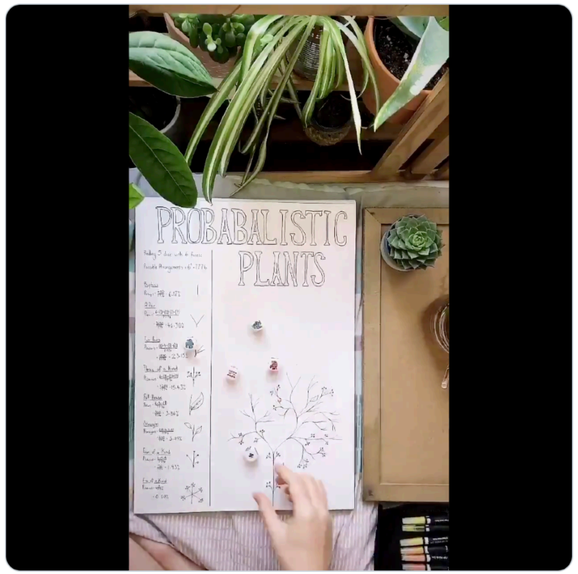
Procedural graphics

[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! 🌱🍀🌿

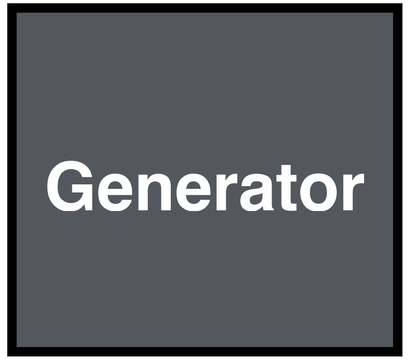


52 1.1K 4.5K

Image synthesis from “noise”

G

$z \sim p(z)$



$x = G(z)$

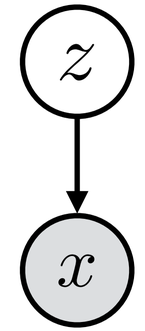
Sampler

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

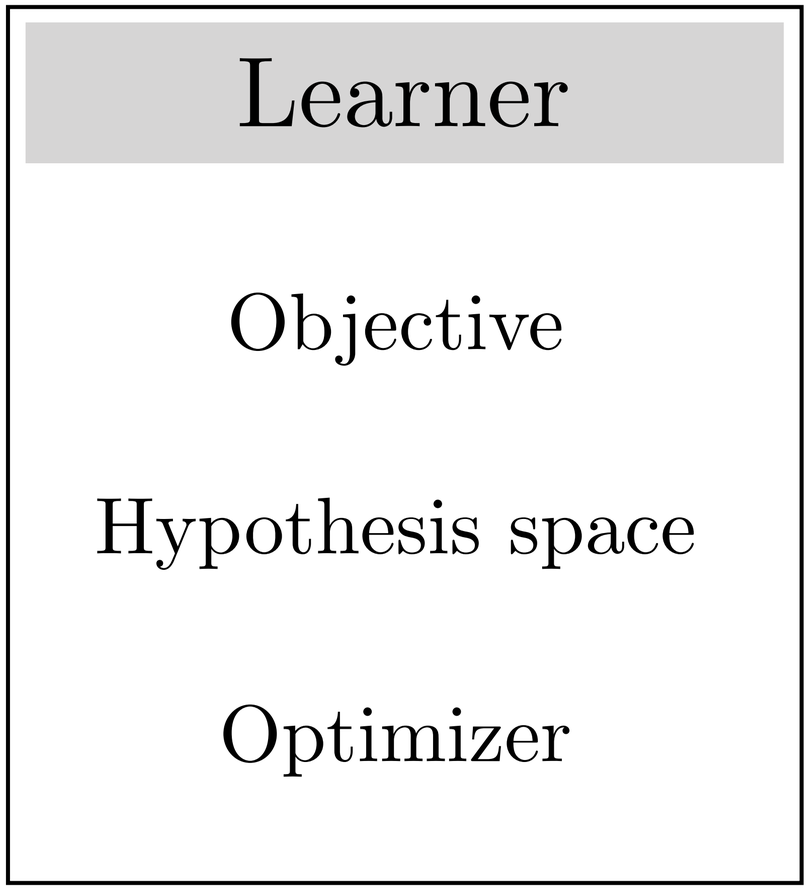
$$z \sim p(z)$$

$$x = G(z)$$

Learning a generative model

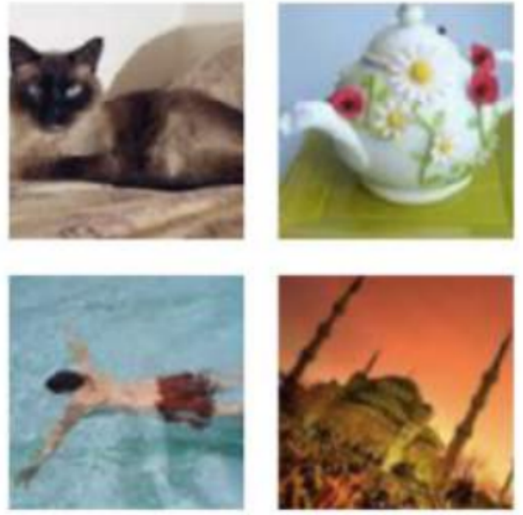


Input samples



latent variables

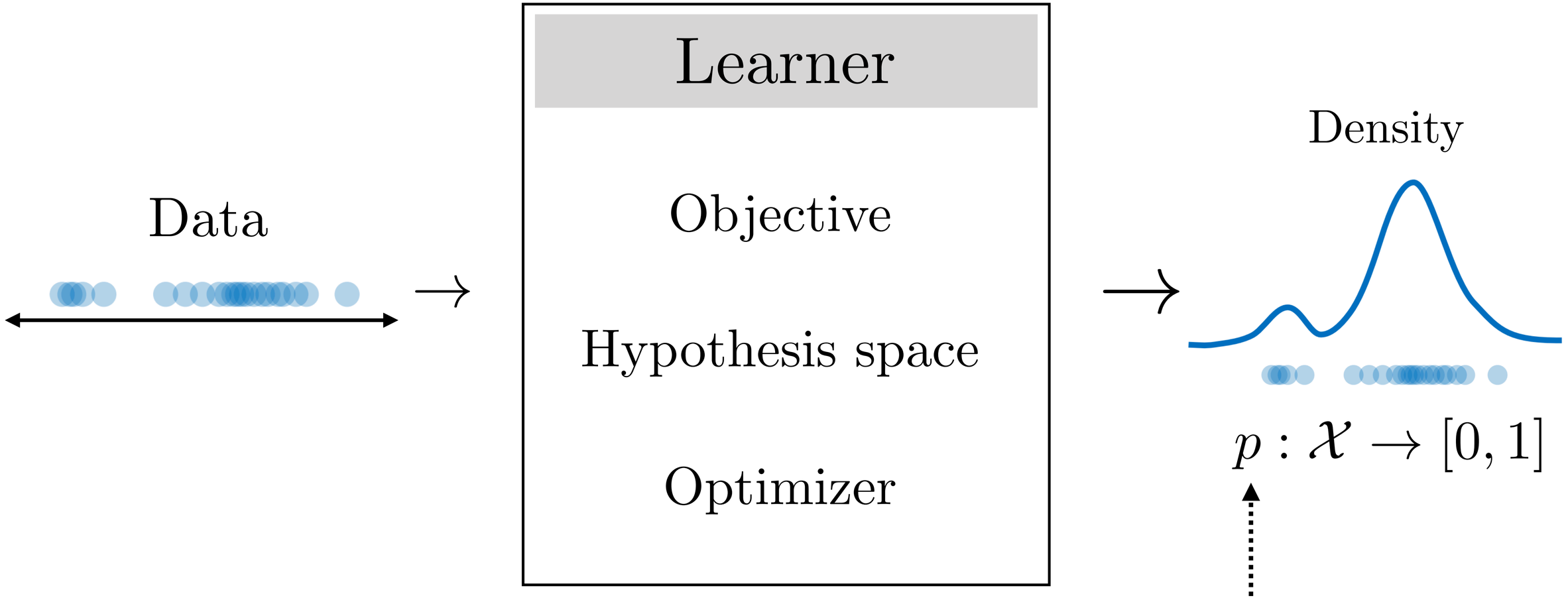
z



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

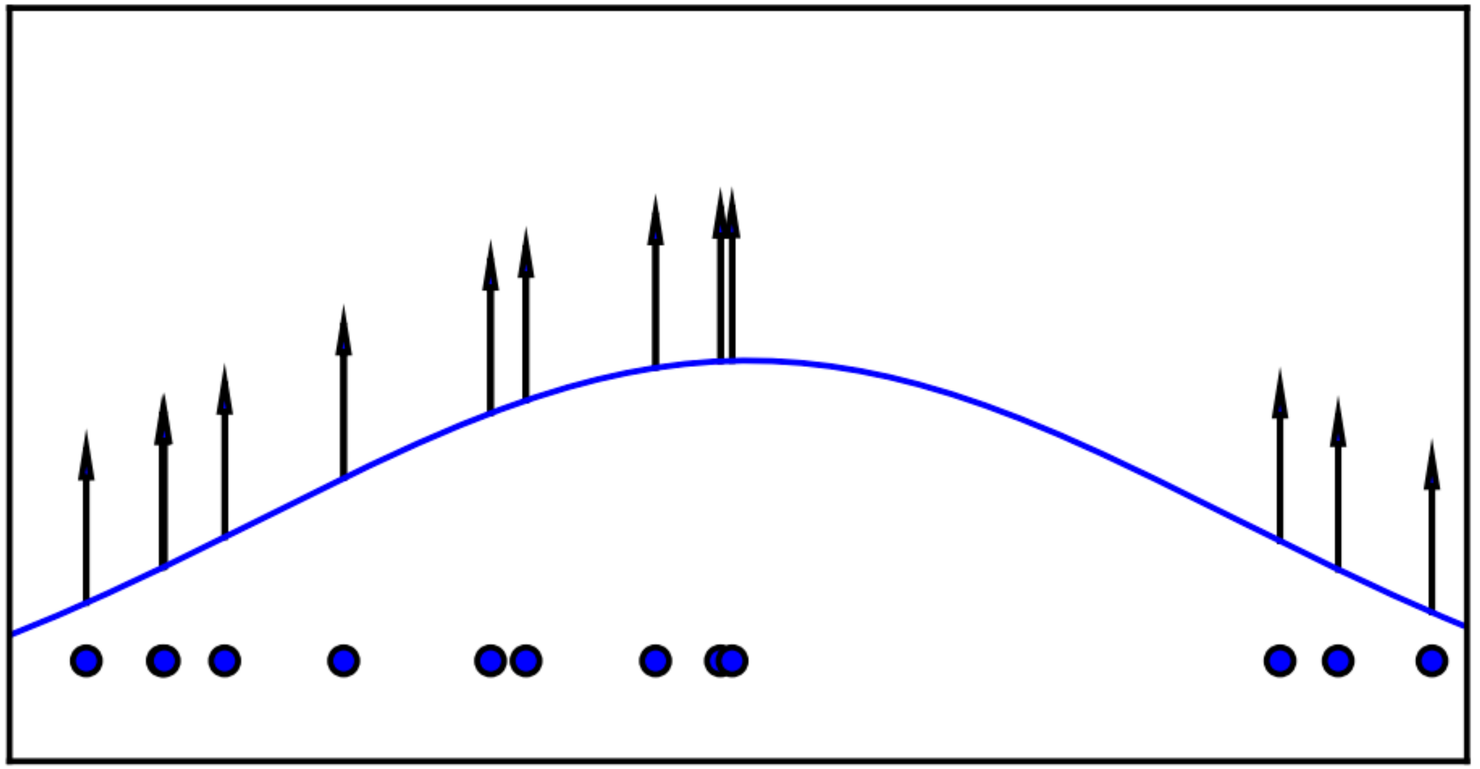


fig from [Goodfellow, 2016]

Max likelihood objective

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

Considering only Gaussian fits

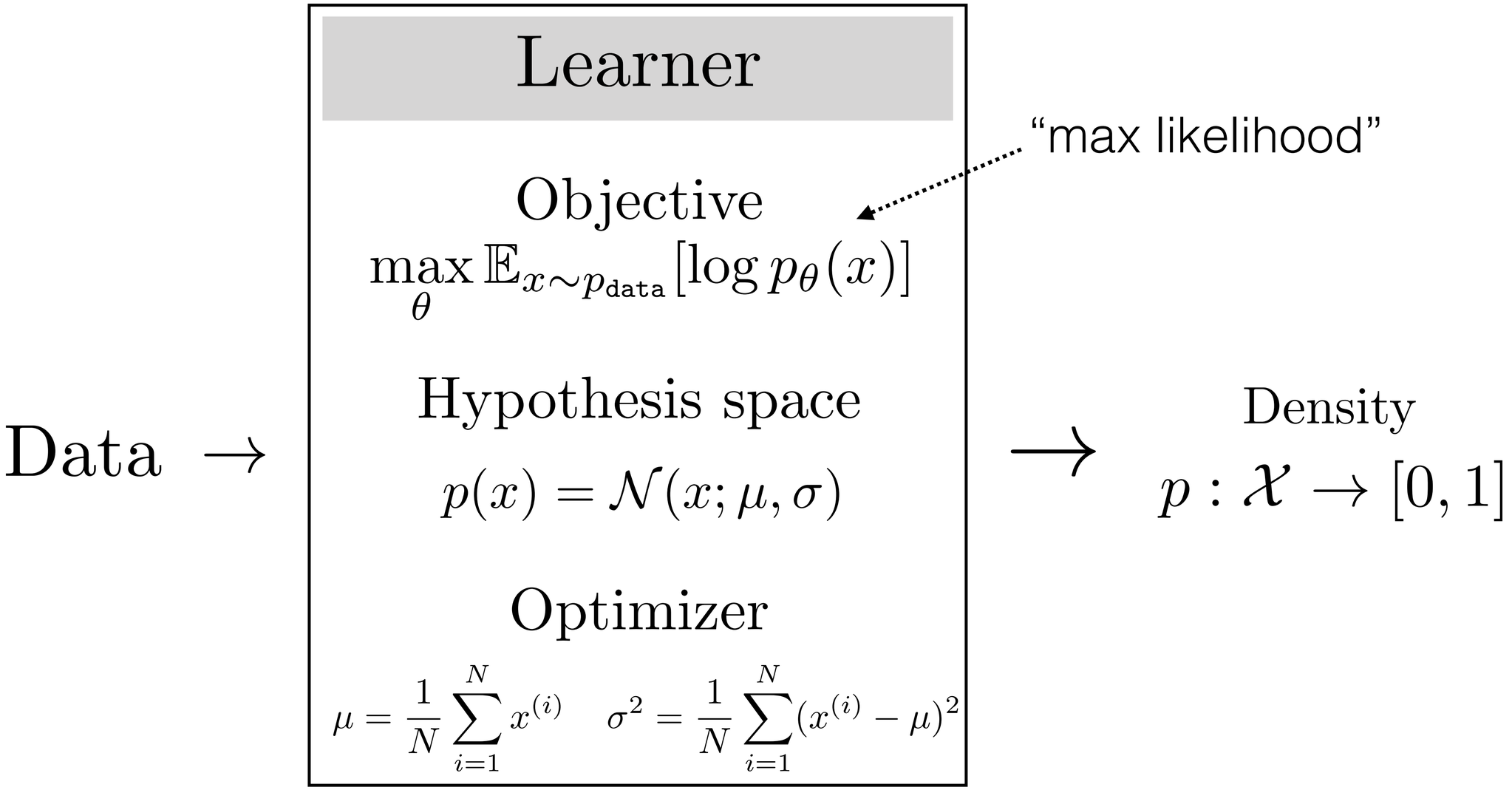
$$p_\theta(x) = \mathcal{N}(x; \mu, \sigma)$$

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

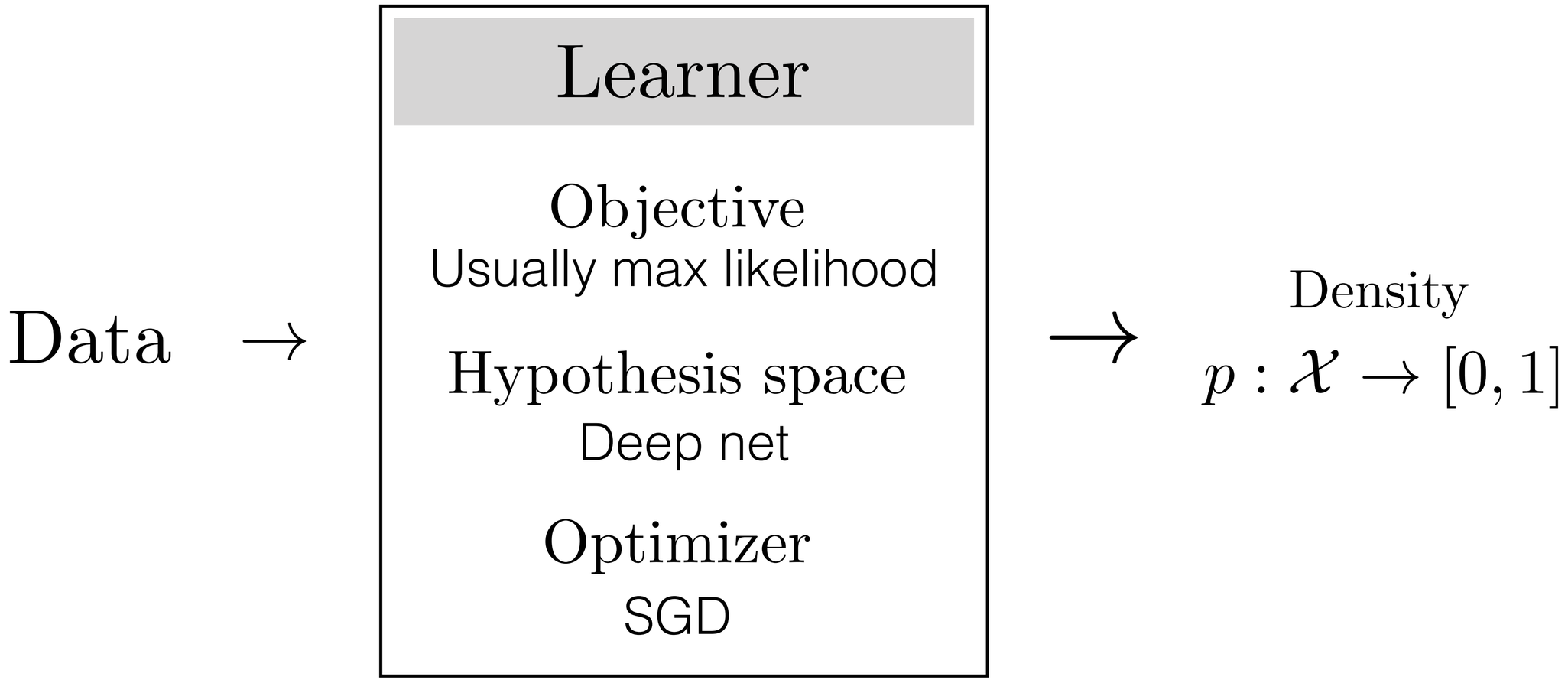
Closed form optimum:

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

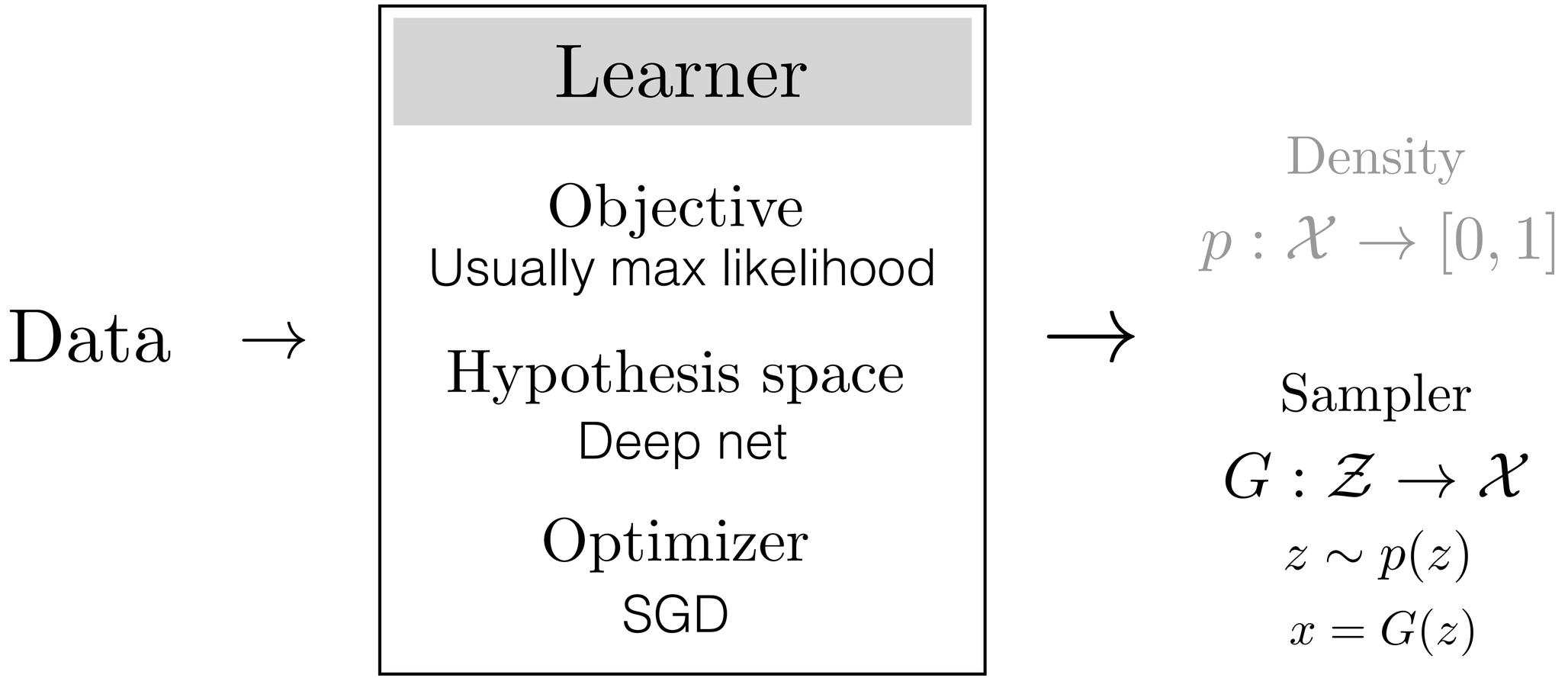
Case study #1: Fitting a Gaussian to data



Case study #2: learning a deep generative model

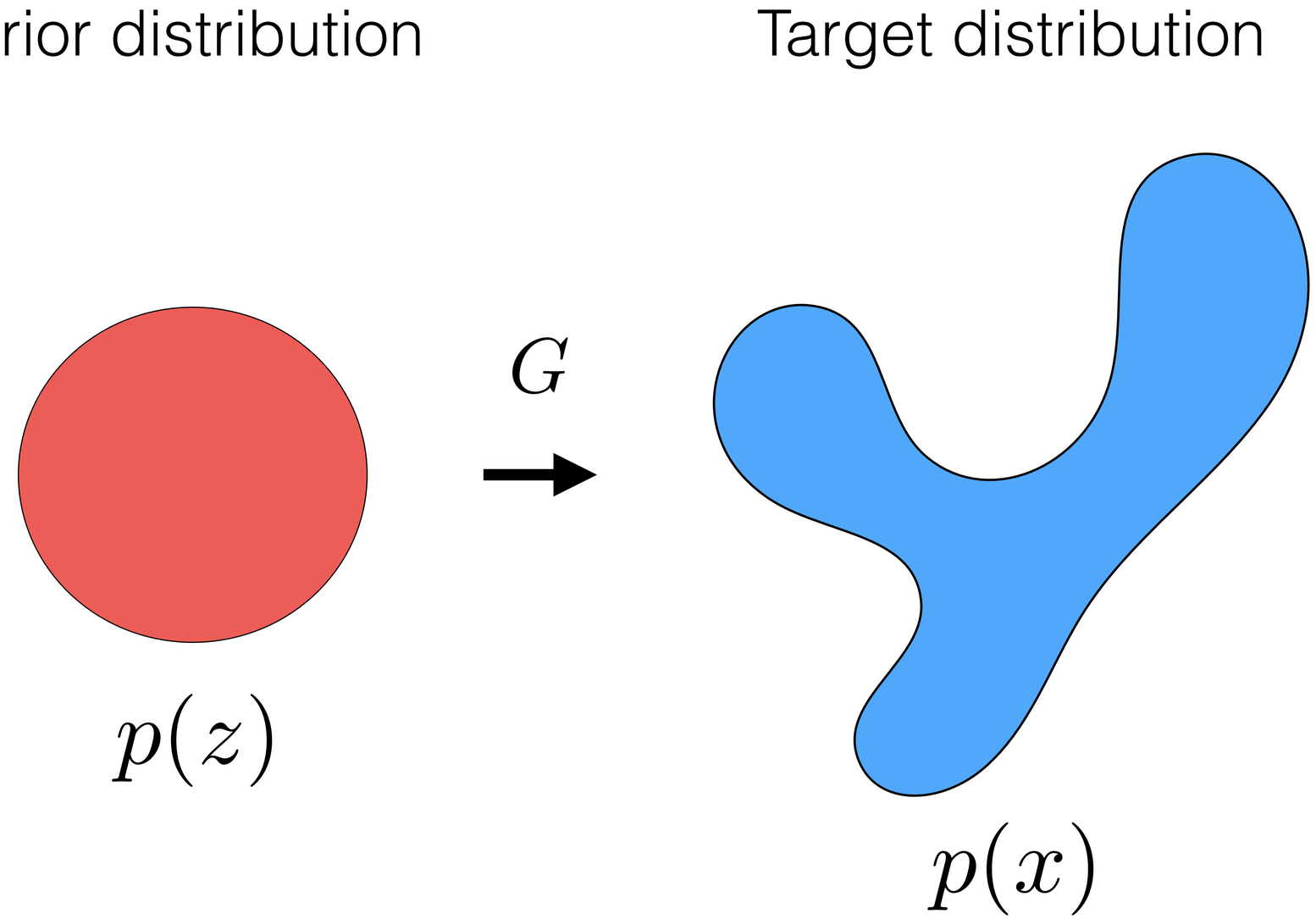


Case study #2: learning a deep generative model

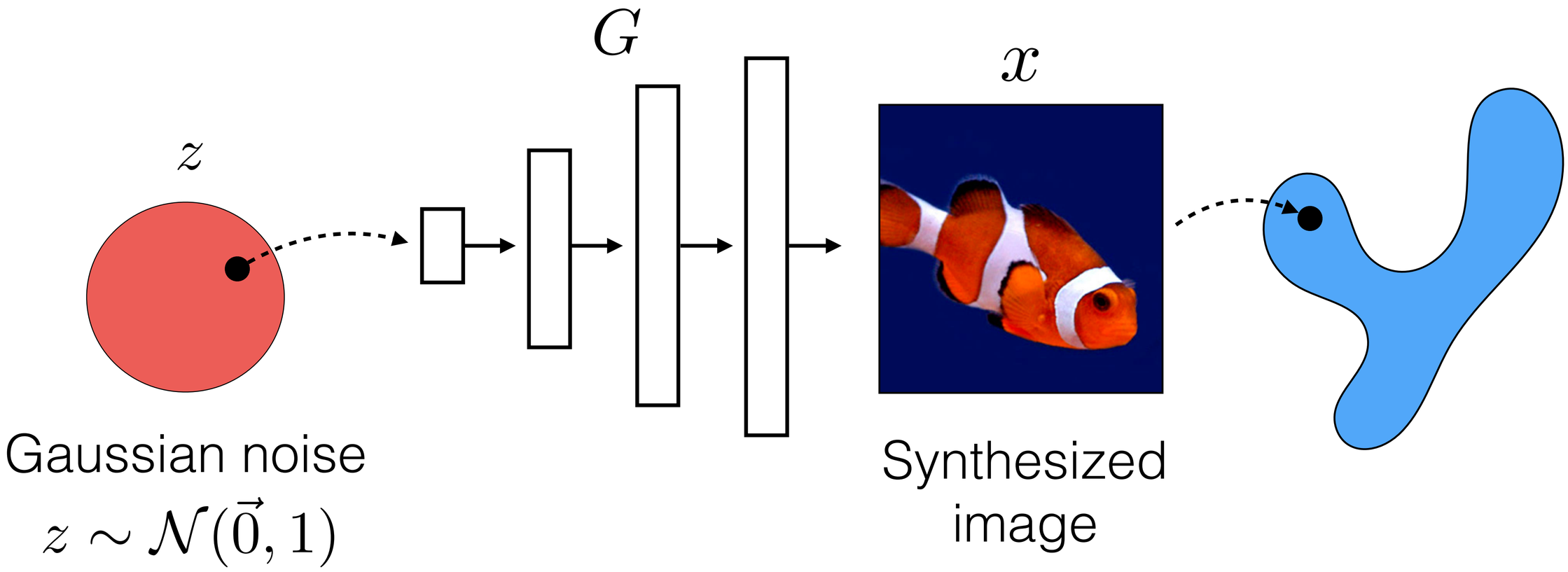


Models that provide a sampler but no density are called **implicit generative models**

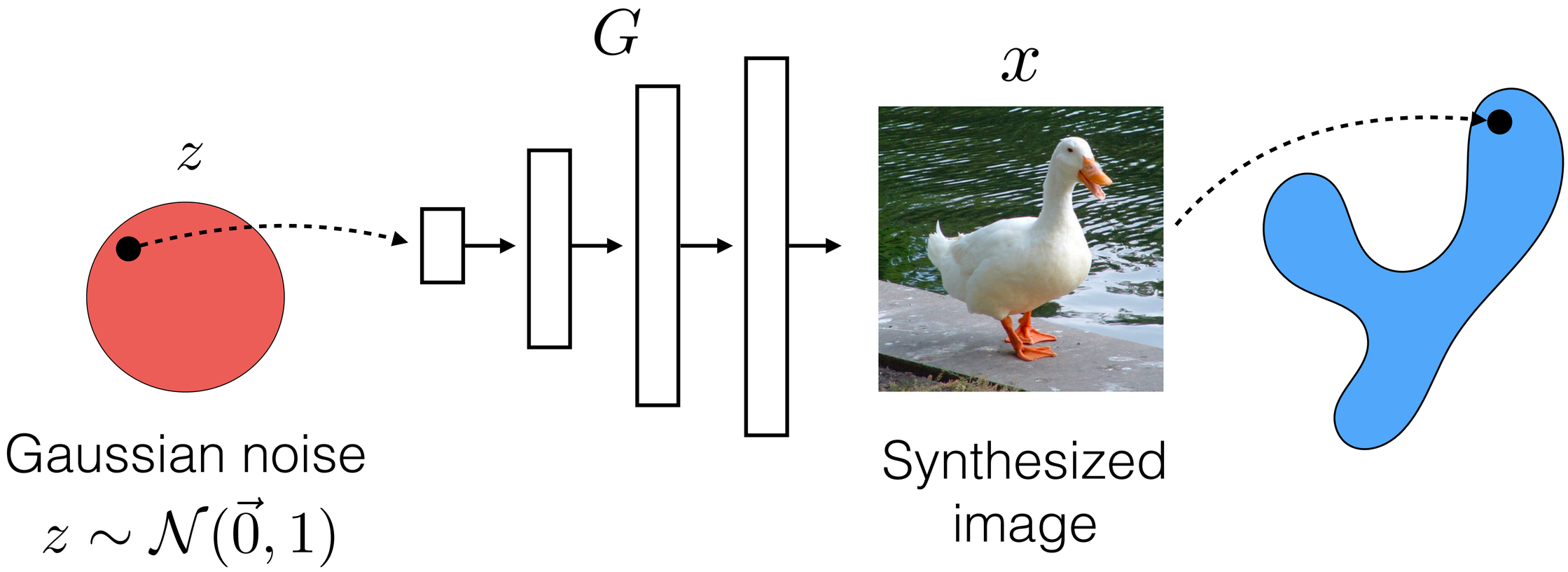
Deep generative models are distribution transformers



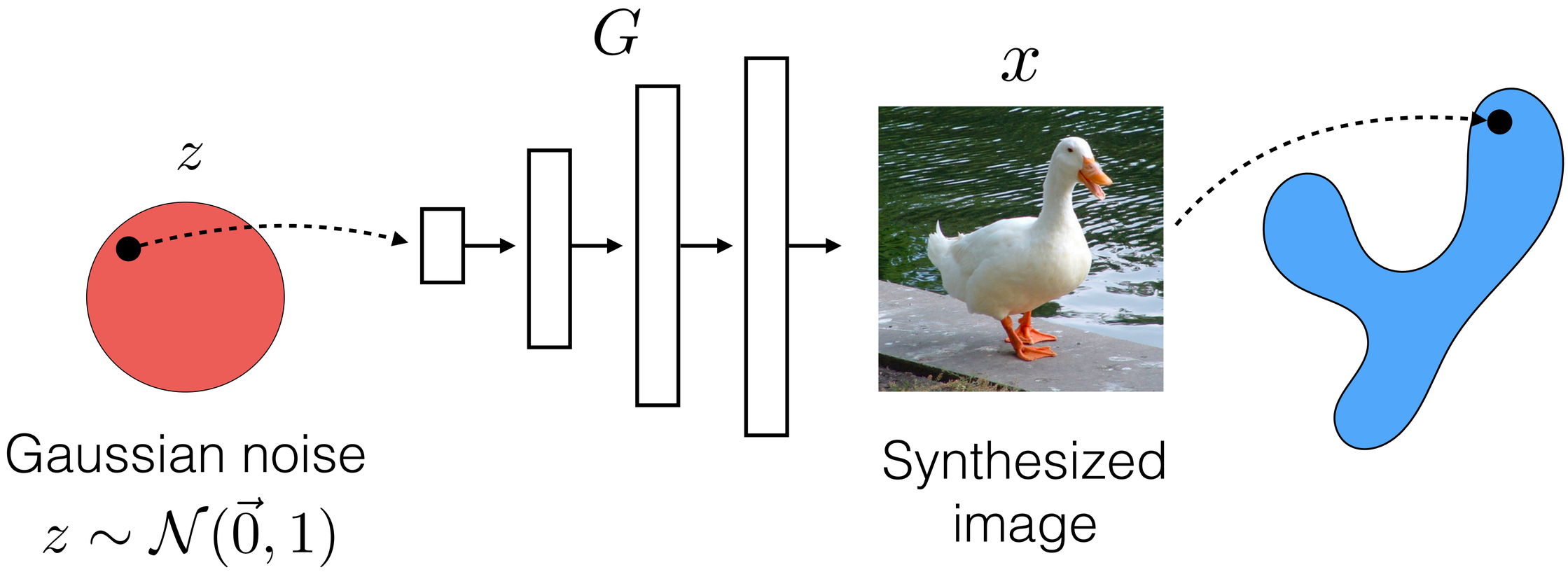
Deep generative models are distribution transformers

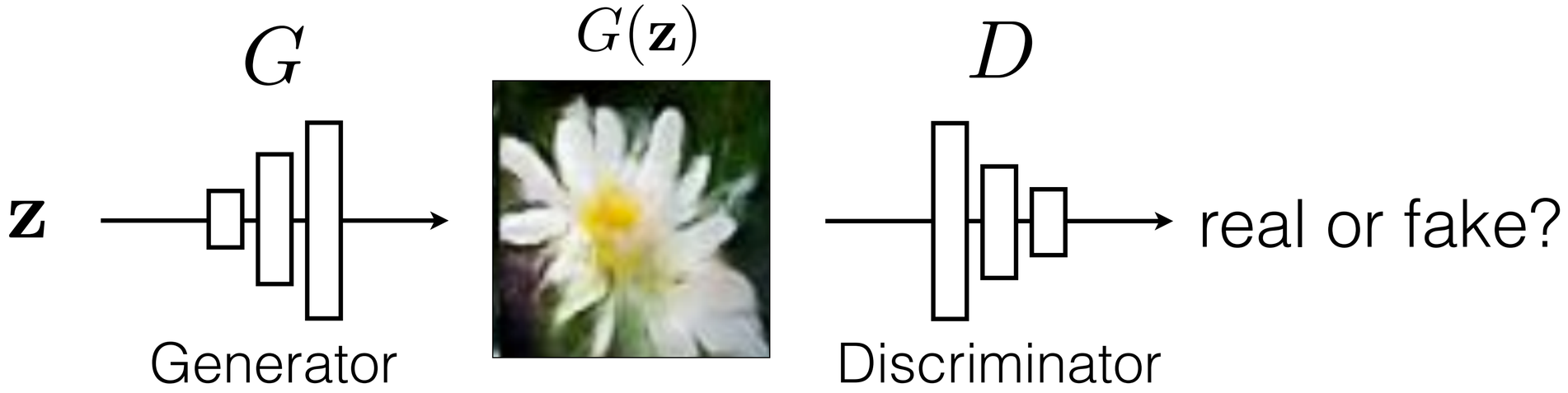


Deep generative models are distribution transformers



Generative Adversarial Networks (GANs)

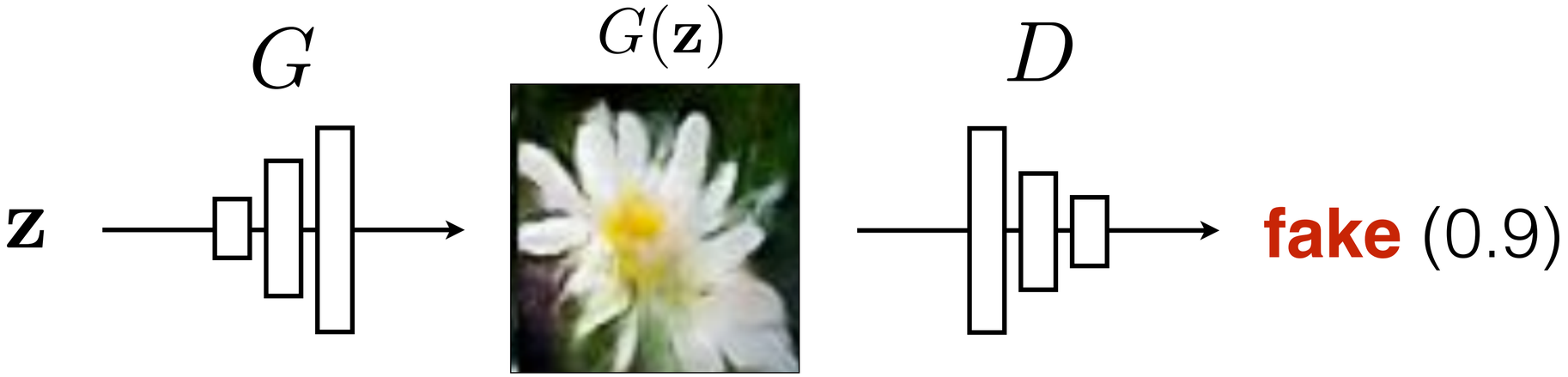


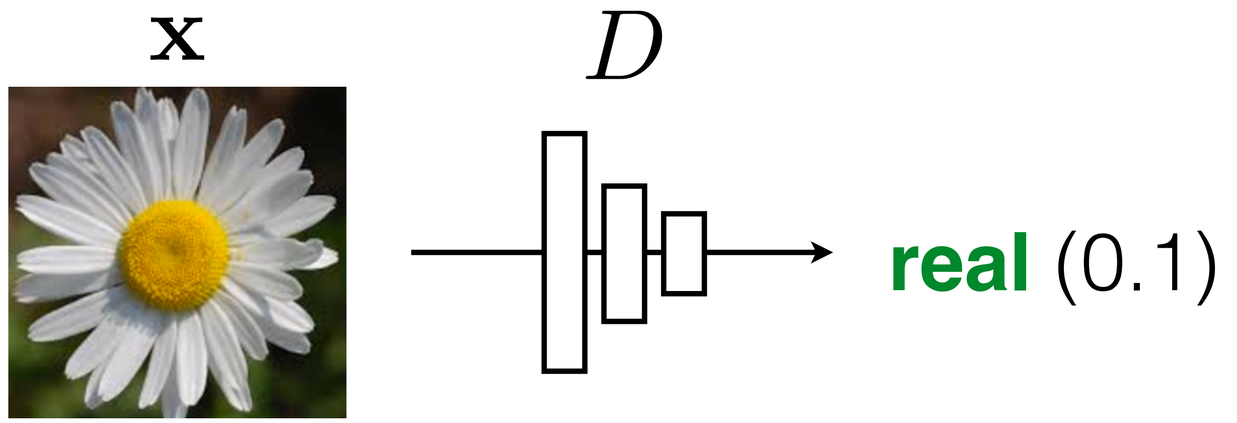


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

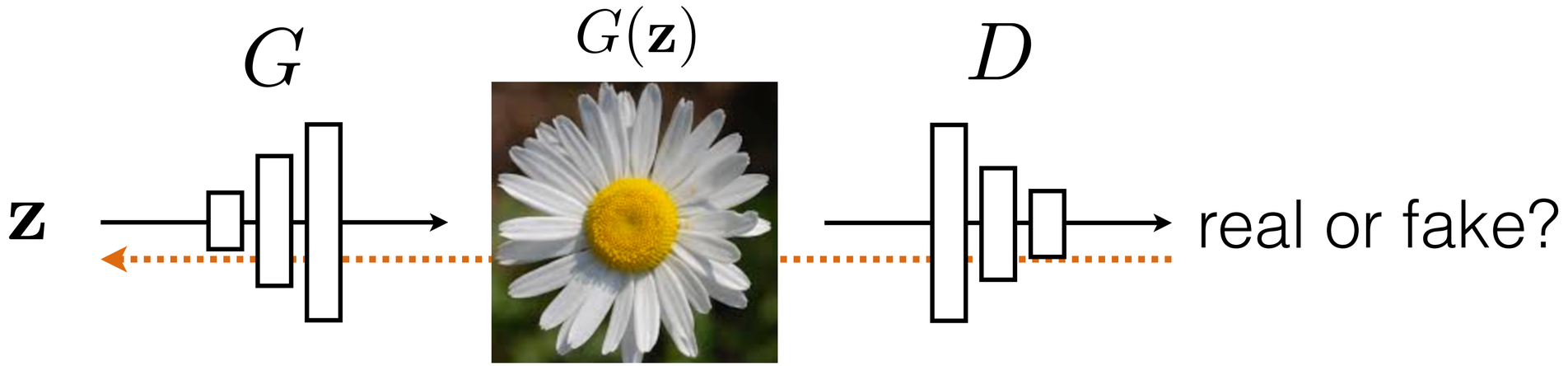
[Goodfellow et al., 2014]





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

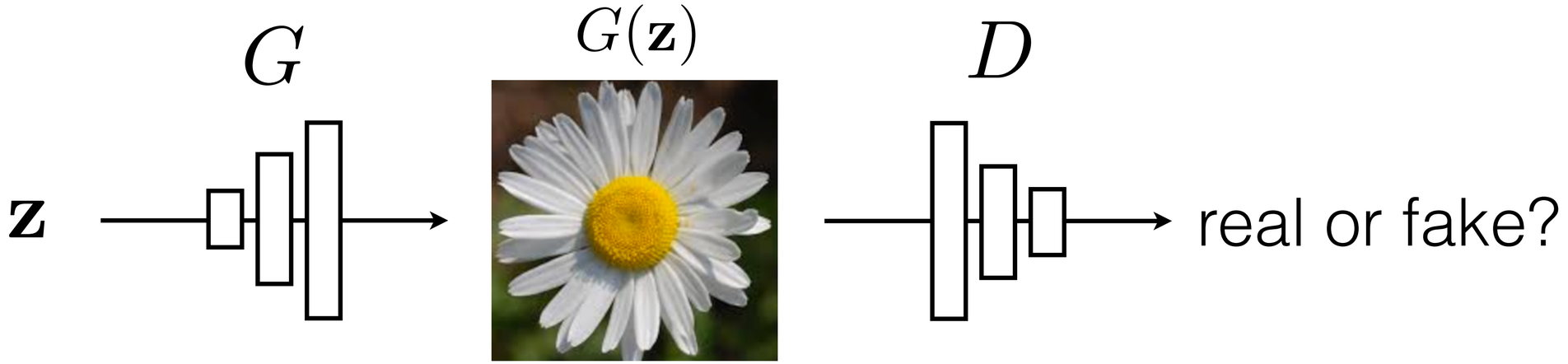
[Goodfellow et al., 2014]



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]

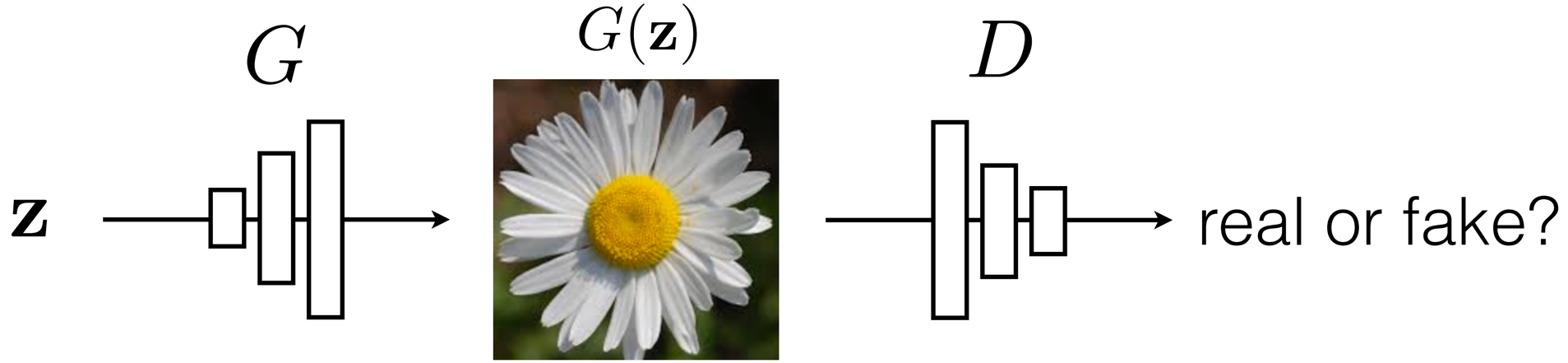


G tries to synthesize fake images that ***fool*** the ***best*** **D**:

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

[Goodfellow et al., 2014]

Training



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

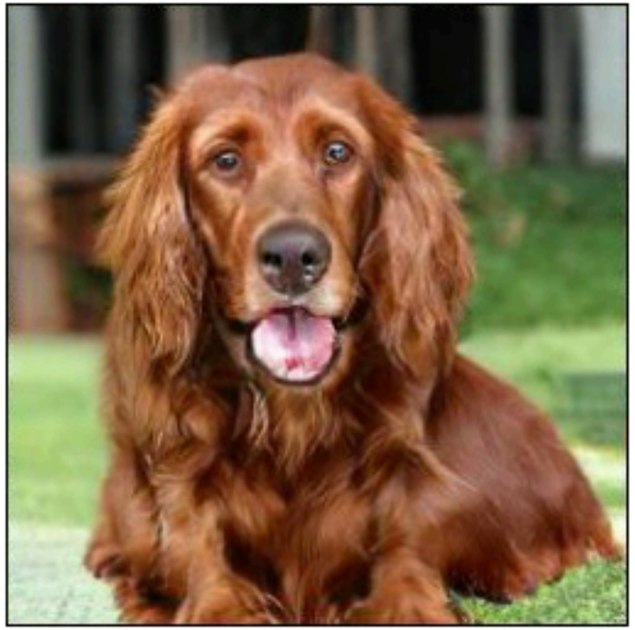
D tries to identify the fakes

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

[Goodfellow et al., 2014]

Samples from BigGAN

[Brock et al. 2018]




Generative Adversarial Network

Data $\rightarrow$

Learner

Objective

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

Hypothesis space

Deep nets G and D

Optimizer

Alternating SGD on G and D

$\rightarrow$

Critic

$$D : \mathcal{X} \rightarrow [0, 1]$$

Sampler

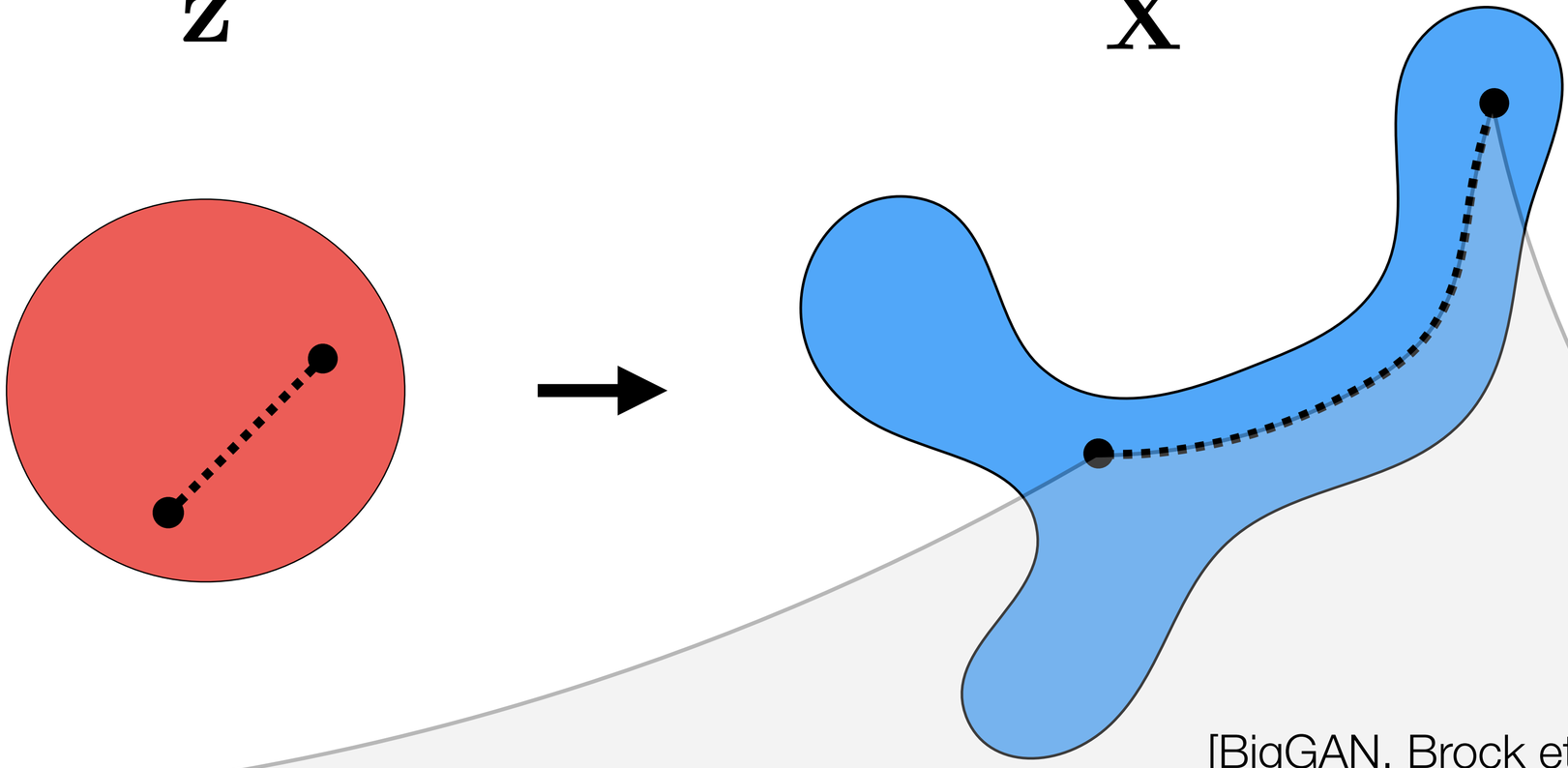
$$G : \mathcal{Z} \rightarrow \mathcal{X}$$

Latent space
(Gaussian)

$\mathbf{z}$

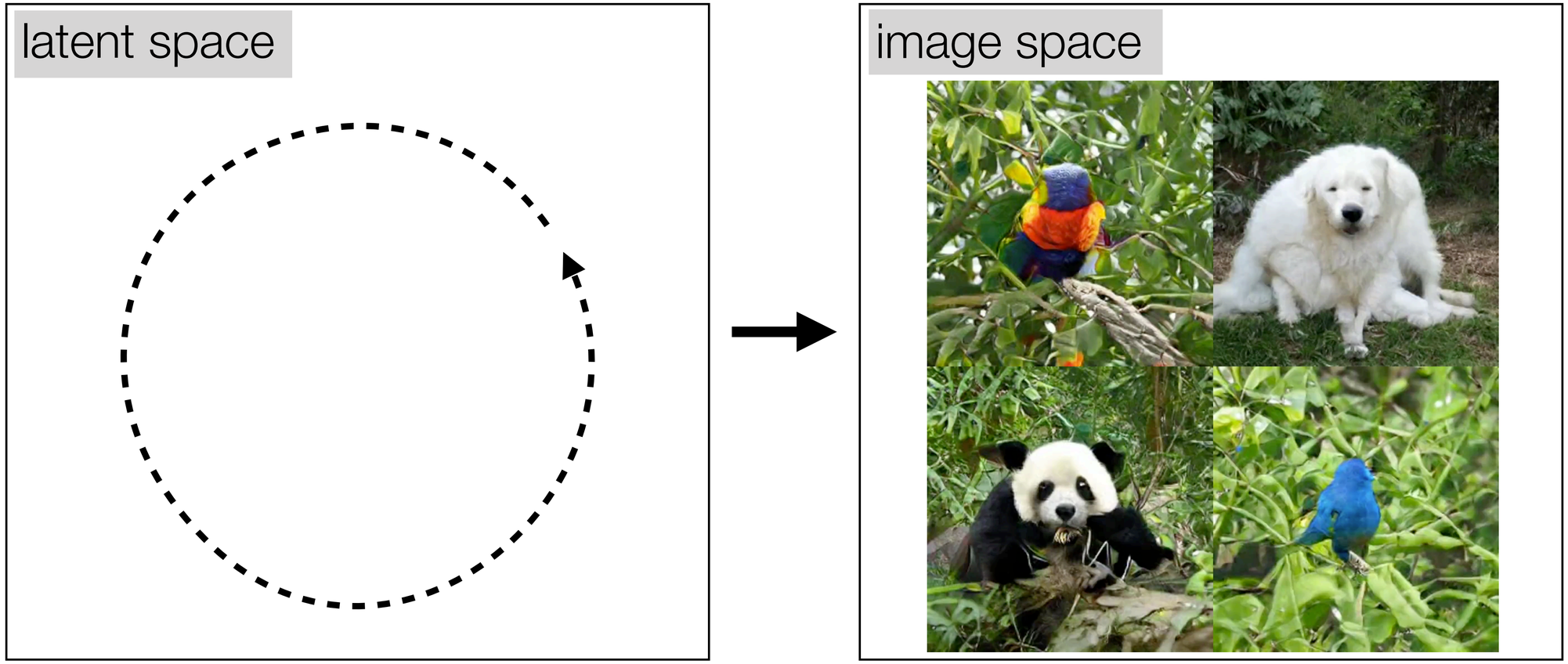
Data space
(Natural image manifold)

$\mathbf{x}$

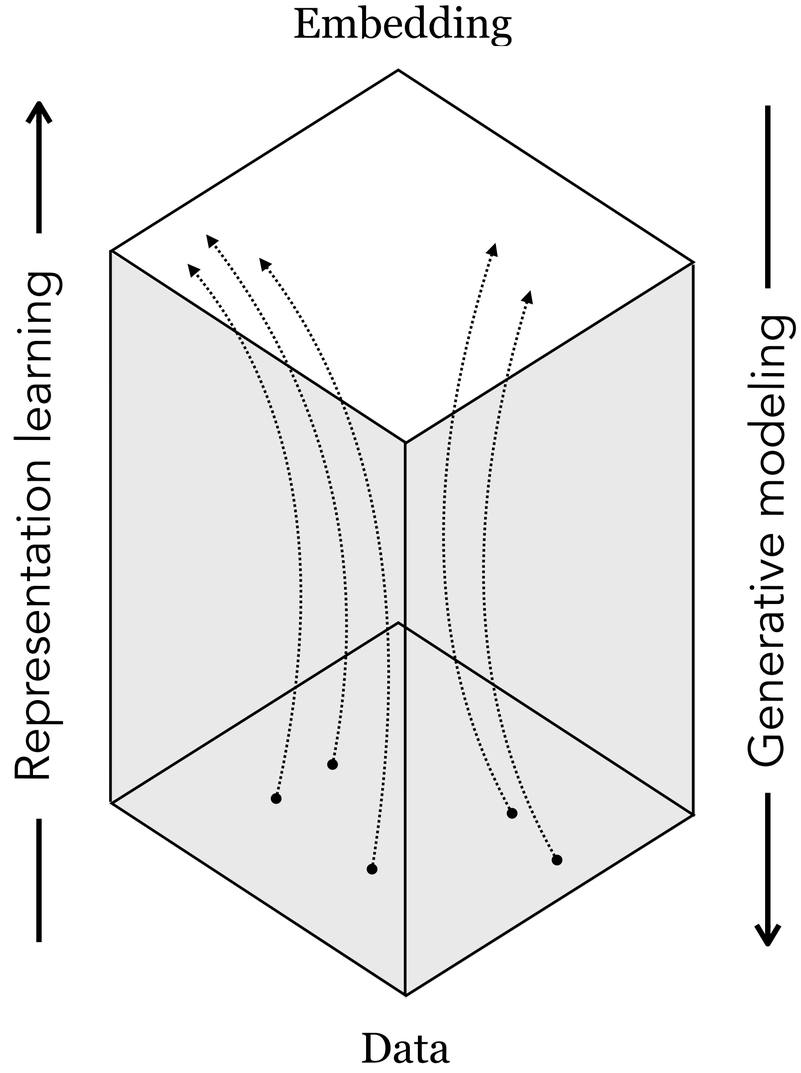


[BigGAN, Brock et al. 2018]

Generative models organize the manifold of natural images

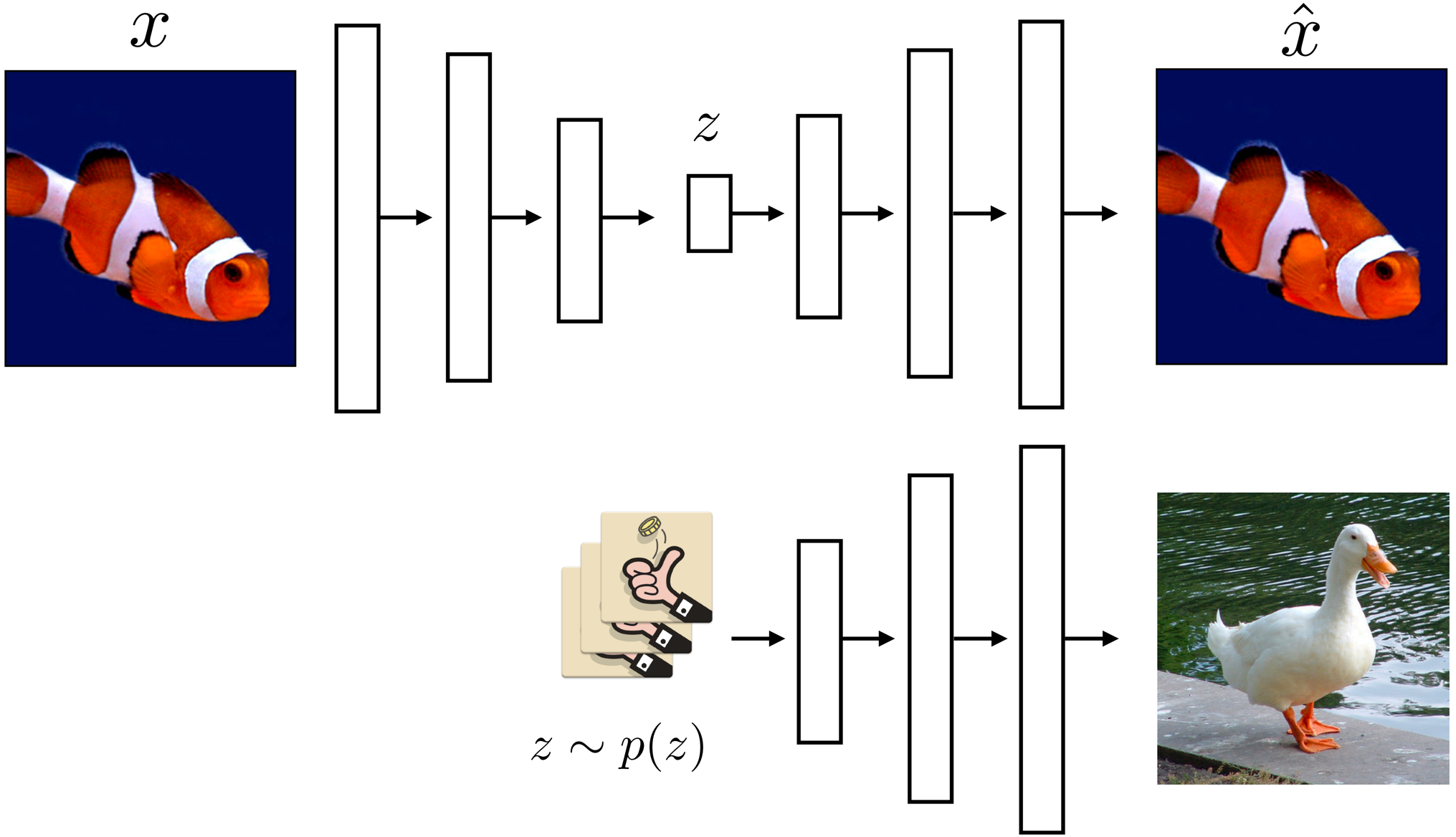


1. Image synthesis
2. **Representation learning**
3. Data translation



Representation learning

Autoencoder —> Generative model

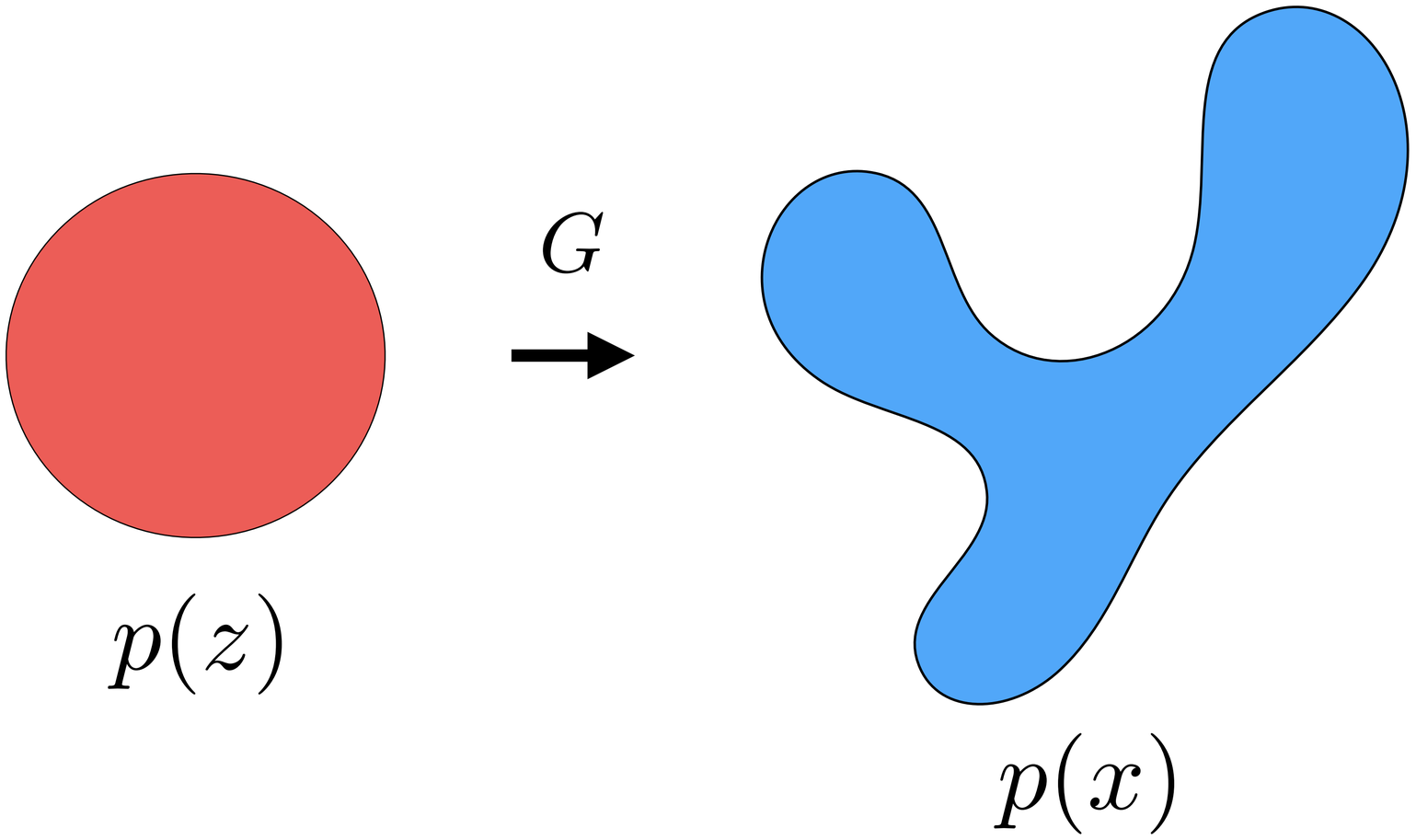


Variational Autoencoders (VAEs)

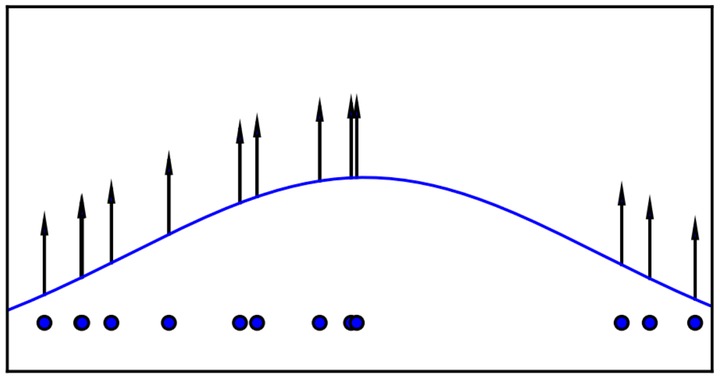
[Kingma & Welling, 2014; Rezende, Mohamed, Wierstra 2014]

Prior distribution

Target distribution

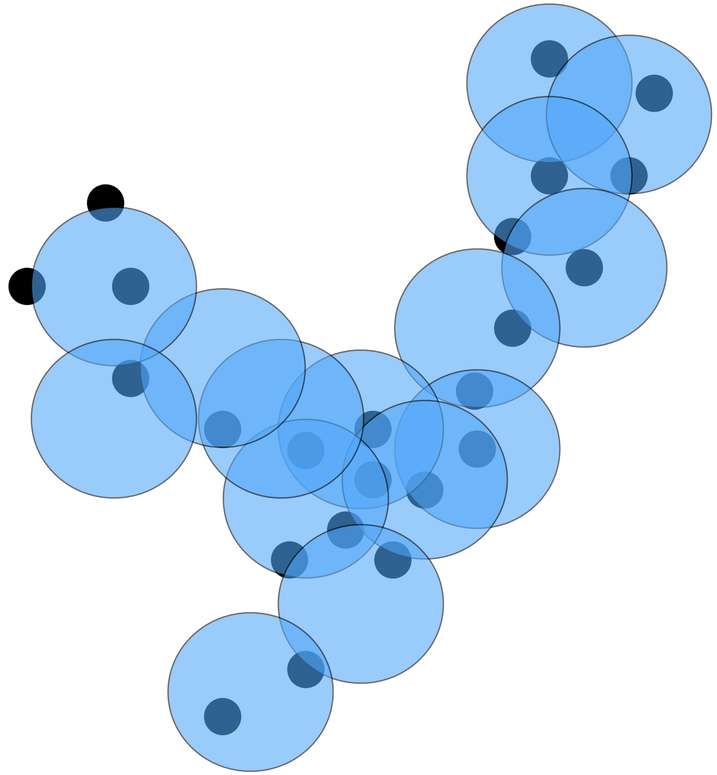


Mixture of Gaussians



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

Target distribution



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

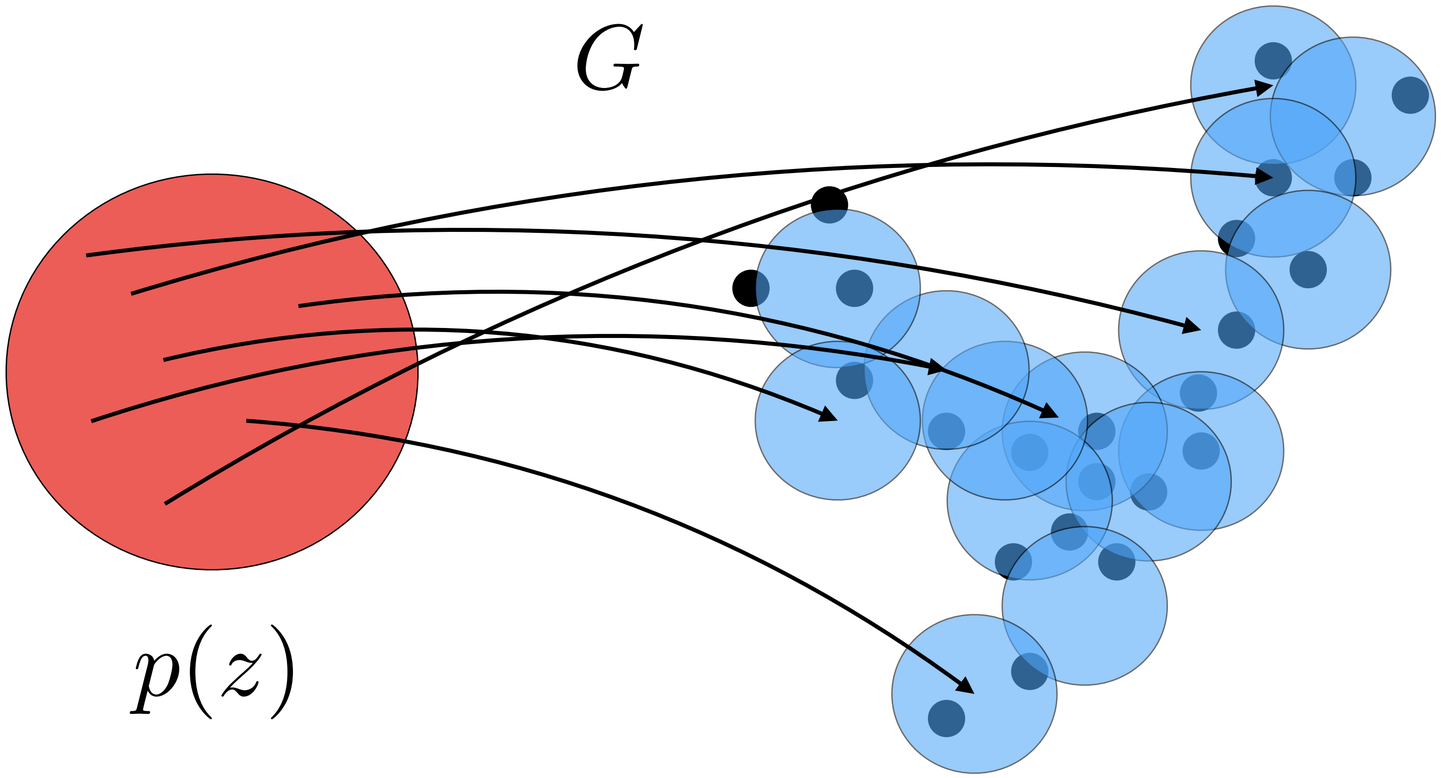
$$p_\theta(x)$$

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

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

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

$\longrightarrow$

Density

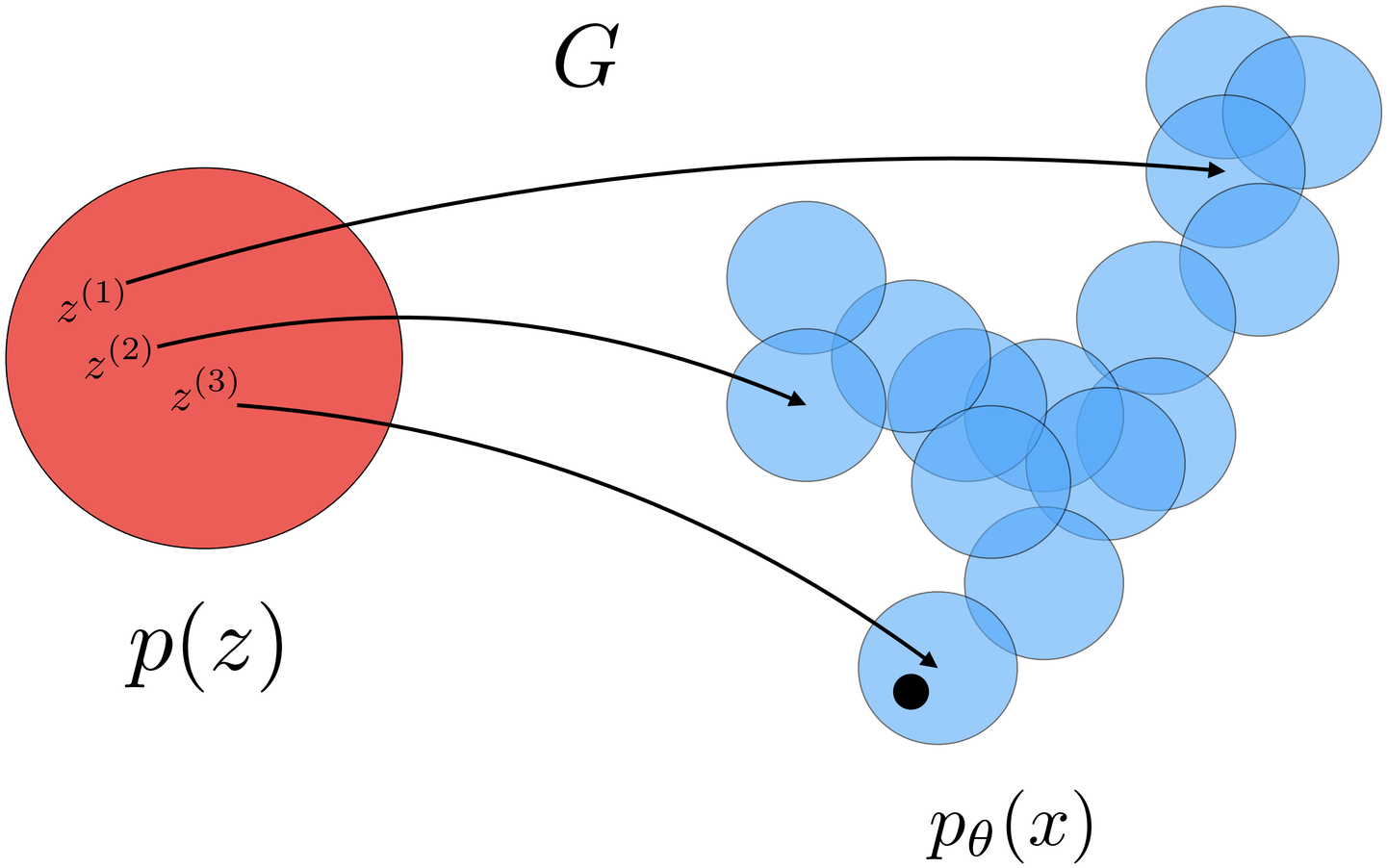
$$p_\theta : \mathcal{X} \rightarrow [0, 1]$$

Sampler

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

Prior distribution

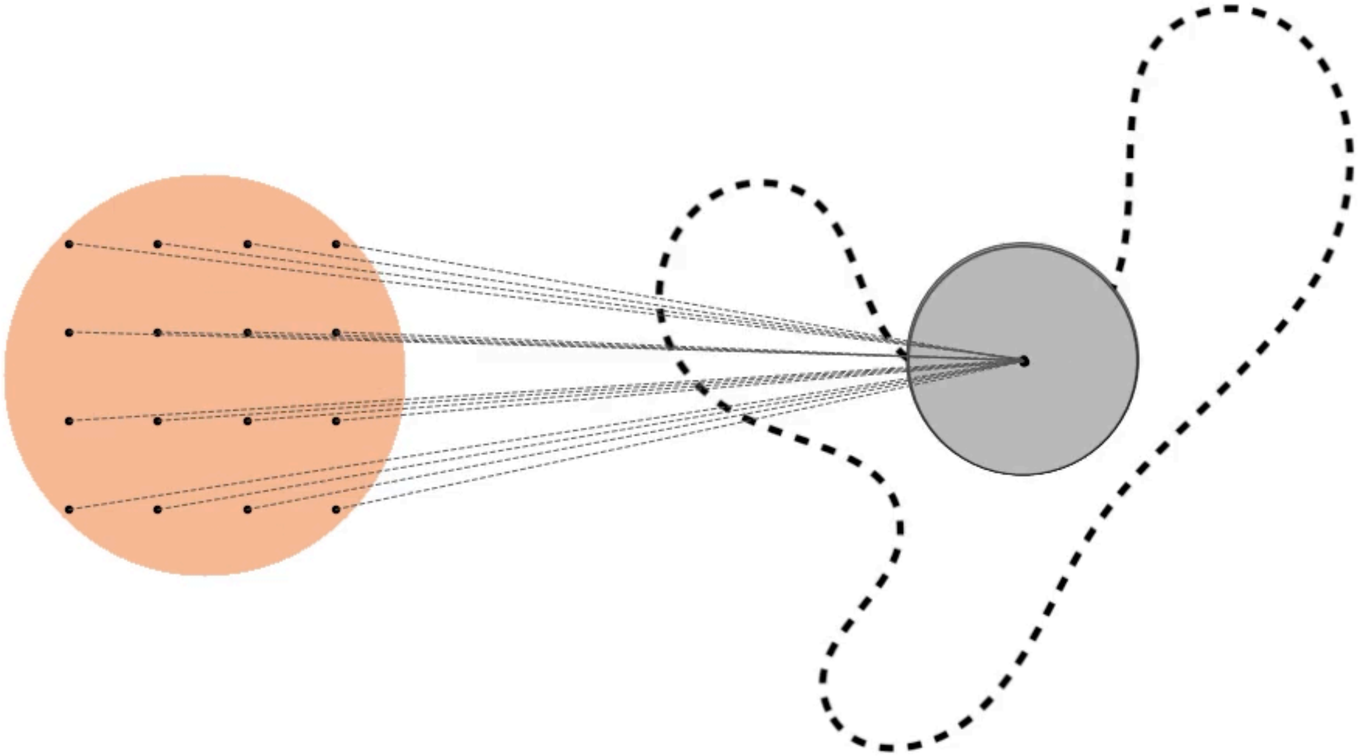
Current model of target distribution



In order to optimize our model, we need to measure the likelihood it assigns to each datapoint x

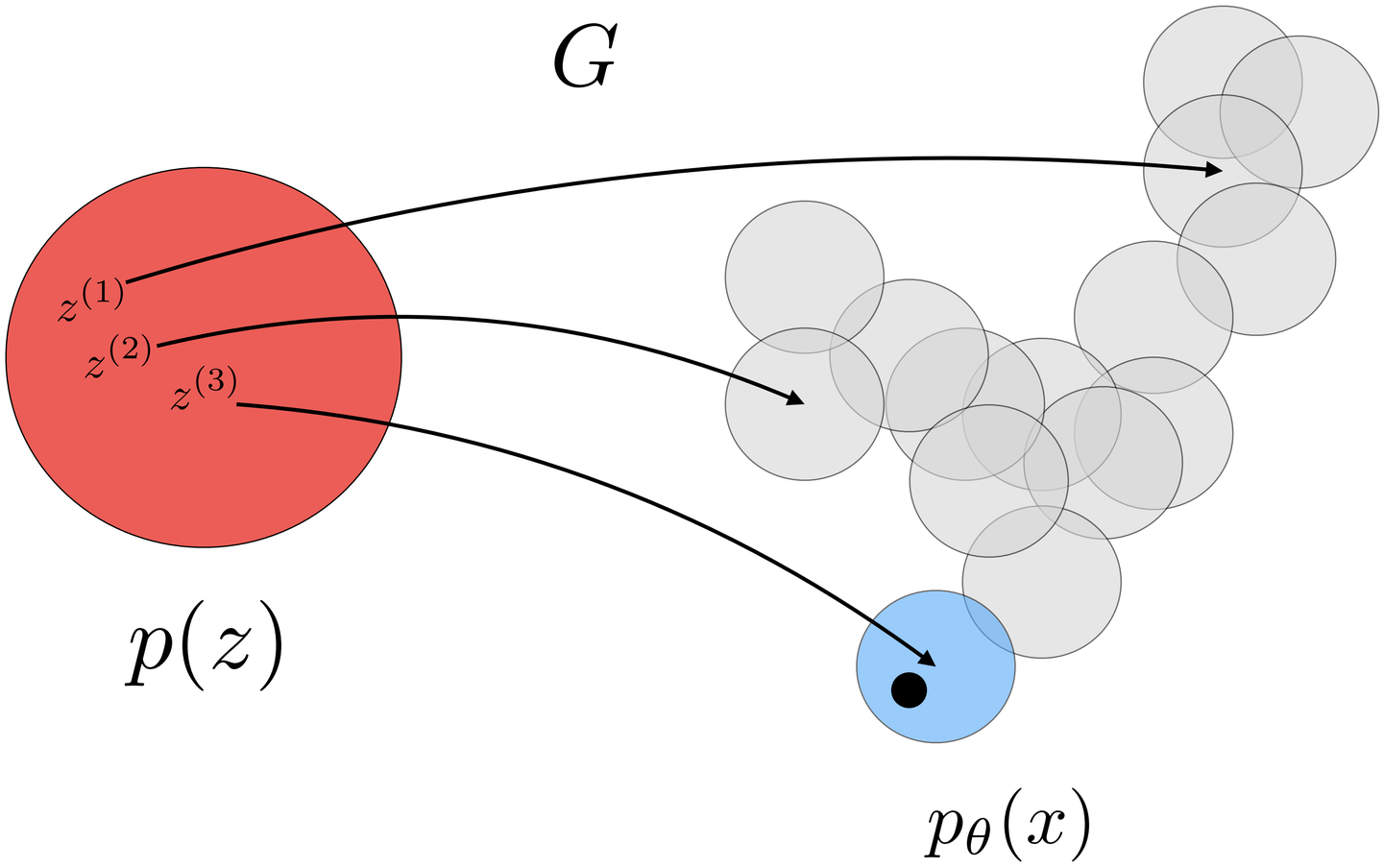
$$p_\theta(x) = \int p(x|z;\theta)p(z)dz$$

$$= p(x|z^{(1)})p(z^{(1)})dz +$$
$$p(x|z^{(2)})p(z^{(2)})dz +$$
$$p(x|z^{(3)})p(z^{(3)})dz + \ldots$$



Prior distribution

Current model of target distribution



In order to optimize our model, we need to measure the likelihood it assigns to each datapoint x

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

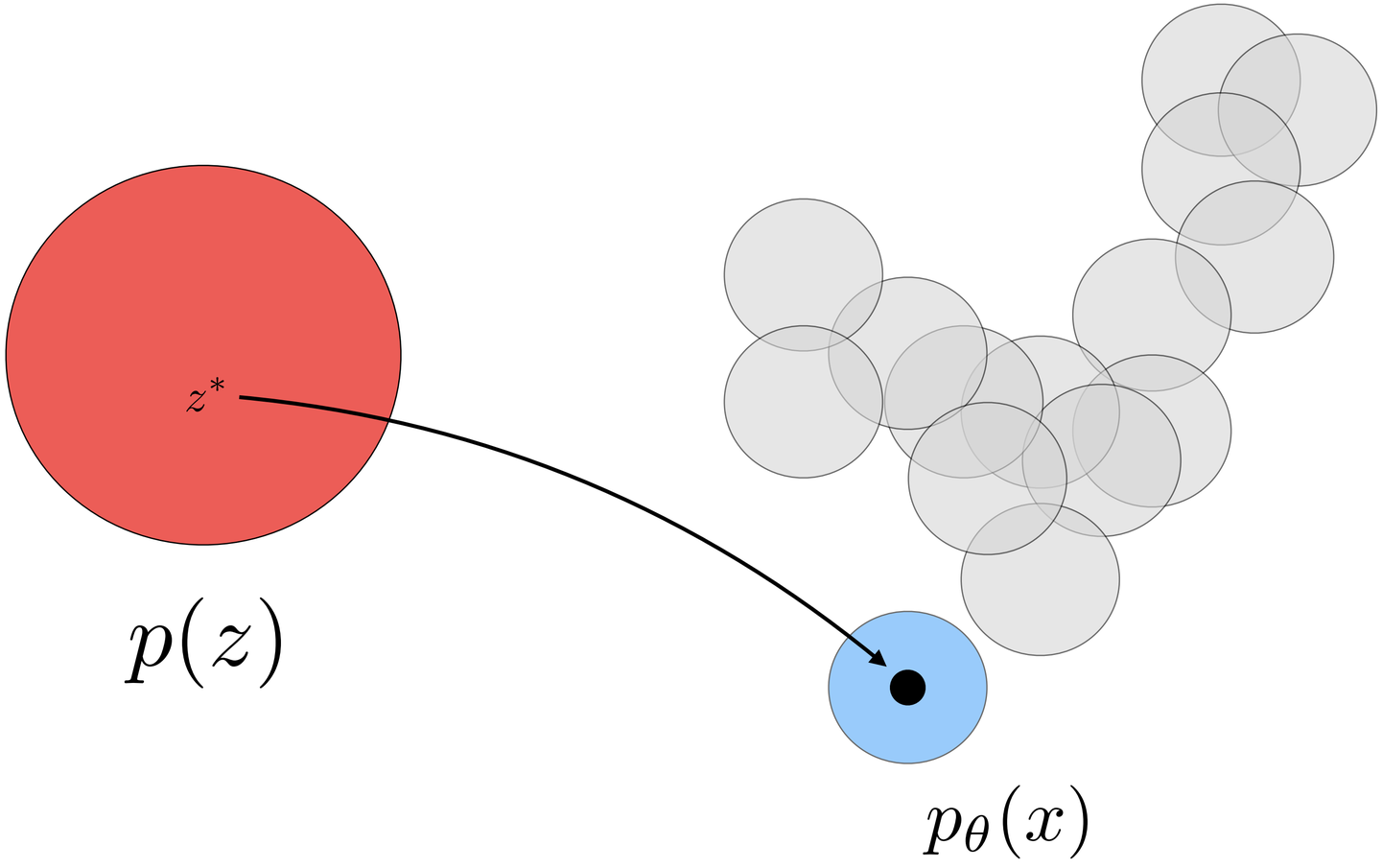
$$= \sim 0+$$

$$\sim 0+$$

$$p(x|z^{(3)})p(z^{(3)})dz + ...$$

Prior distribution

Current model of target distribution

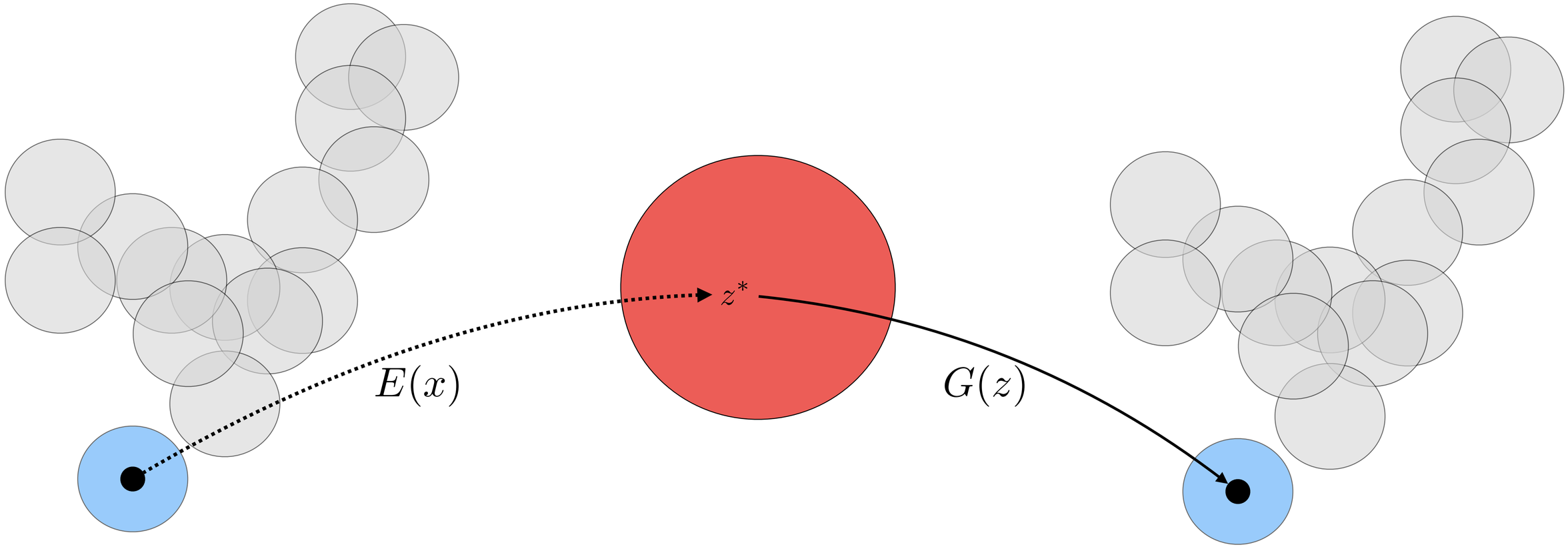


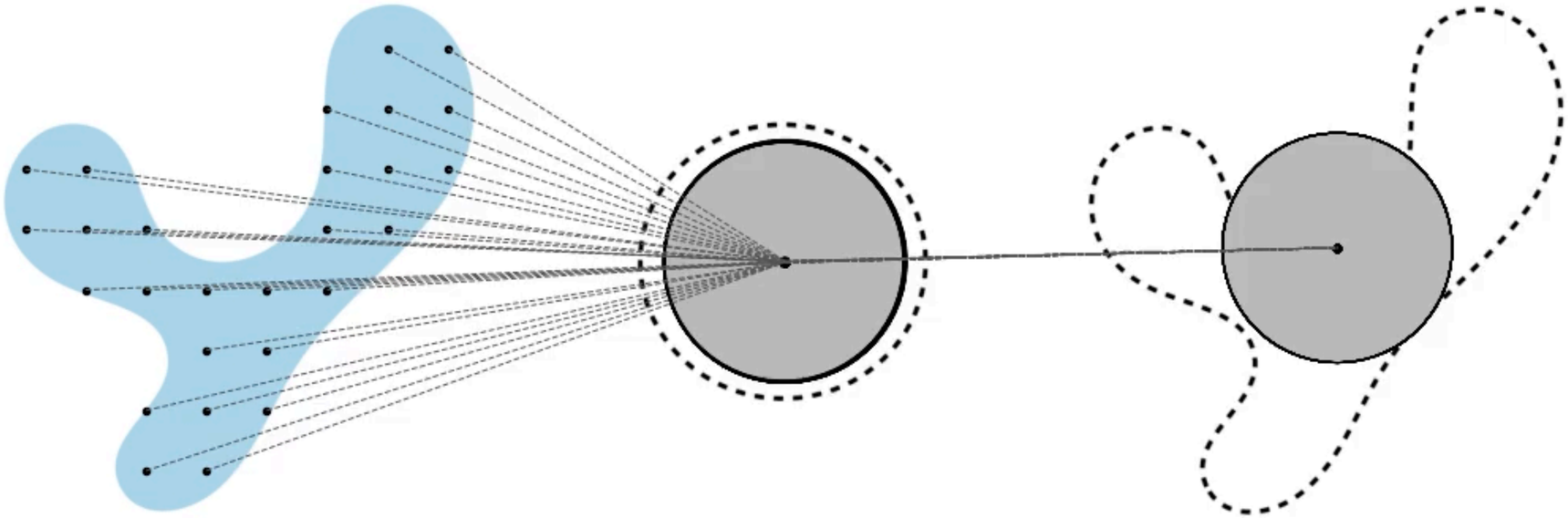
If only we knew z*, we wouldn't need the integral…

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

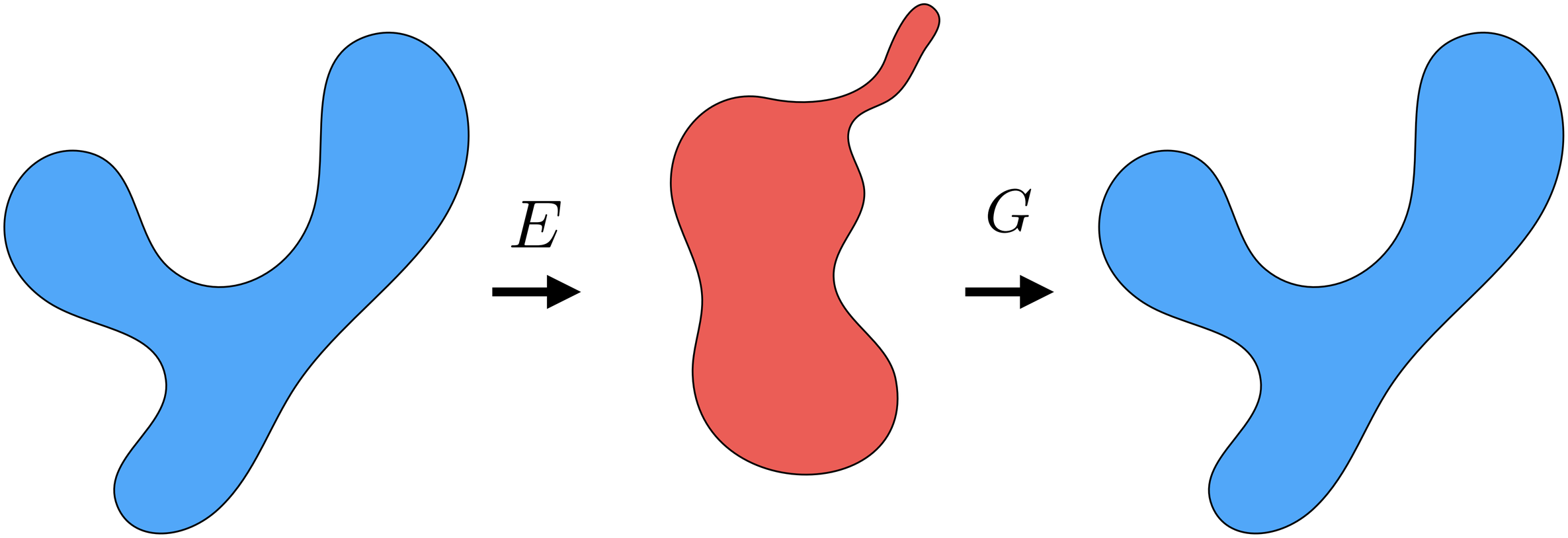
$$\approx p(x|z^*;\theta)p(z^*)$$

Autoencoder!



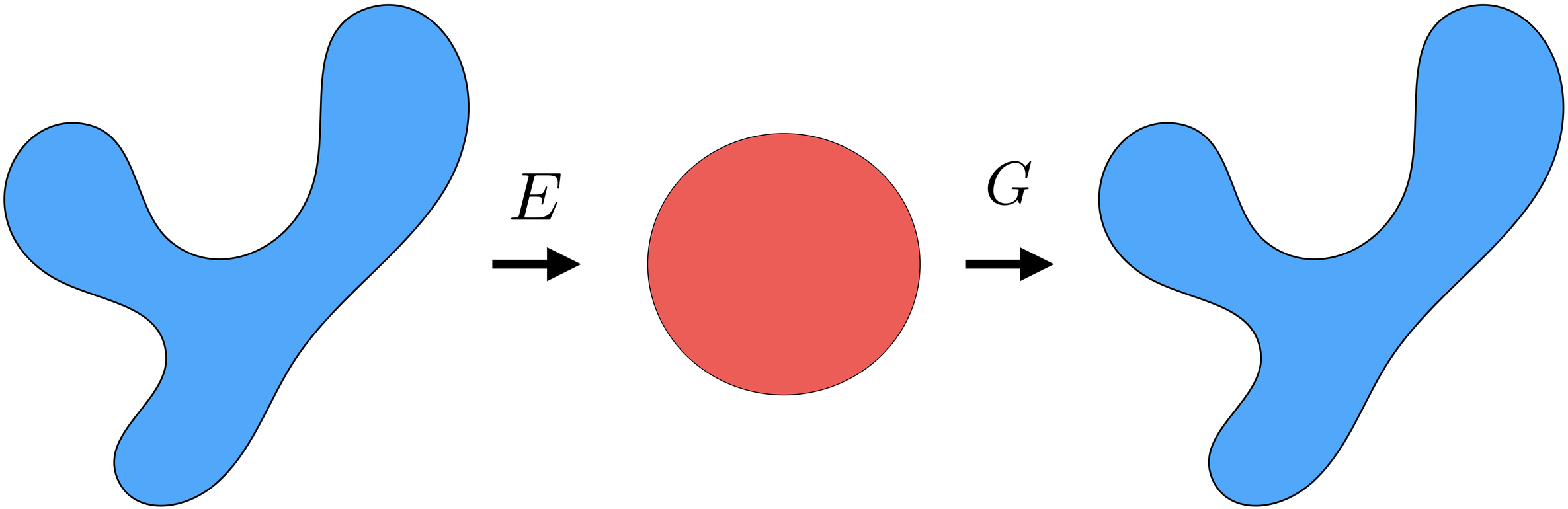


Classical Autoencoder



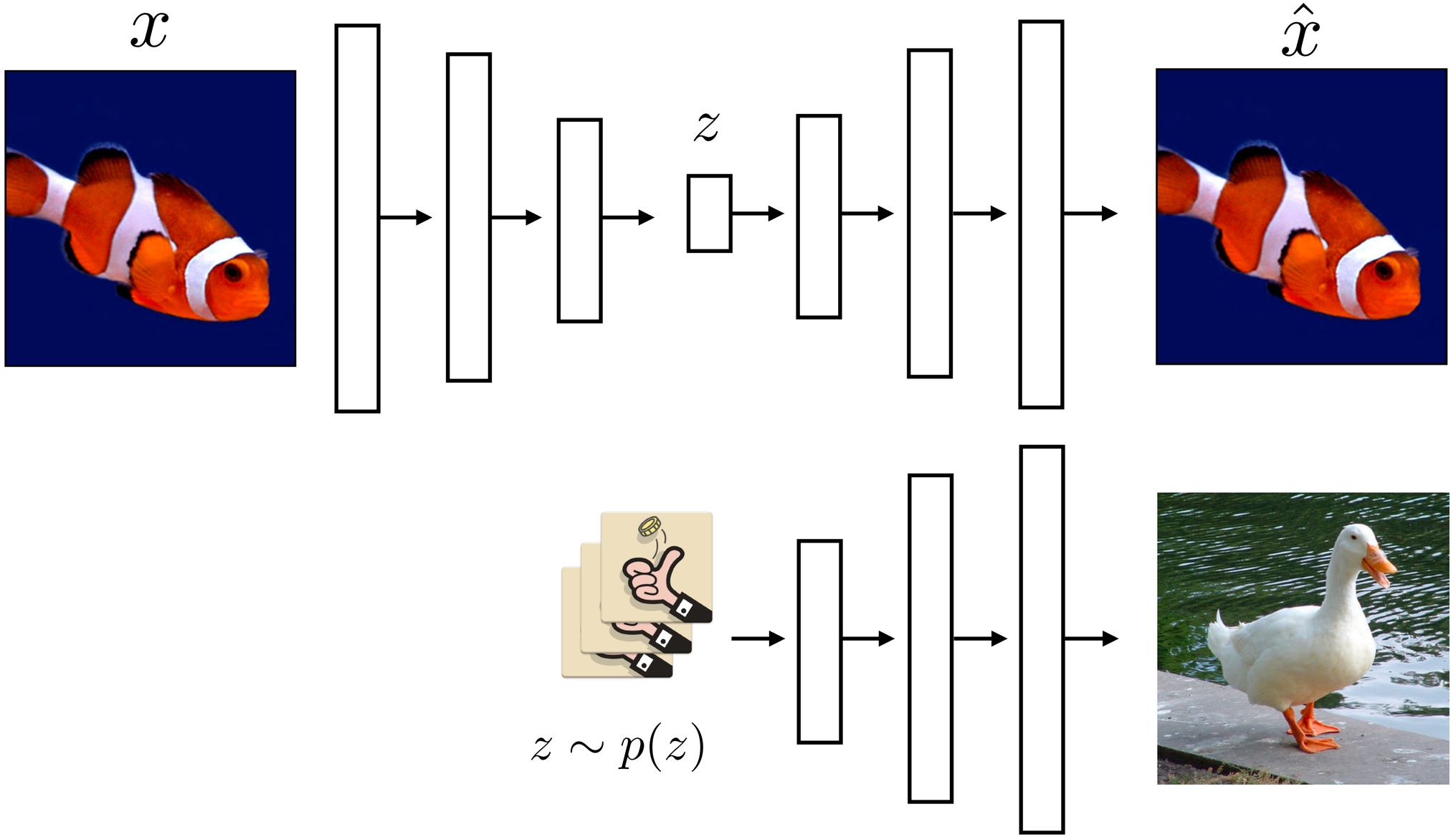
$$\arg\min_{G,E} \mathbb{E}_x[\|G(E(x)) - x\|_2^2]$$

Variational Autoencoder

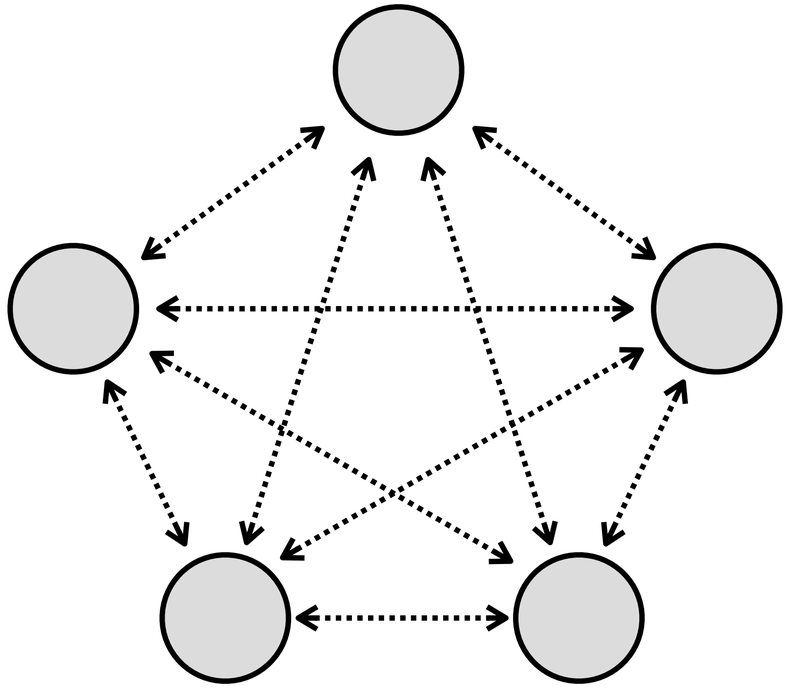


$$\arg\min_{G,E} \mathbb{E}_{x,\epsilon}[\|G(E(x+\epsilon)) - x\|_2^2 + \|E(x+\epsilon)\|_2^2]$$

Variational Autoencoder



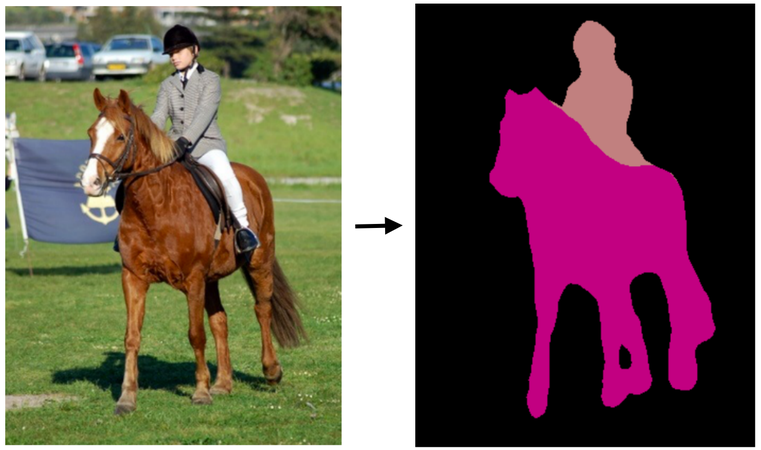
1. Image synthesis
2. Representation learning
3. **Data translation**



Data Translation

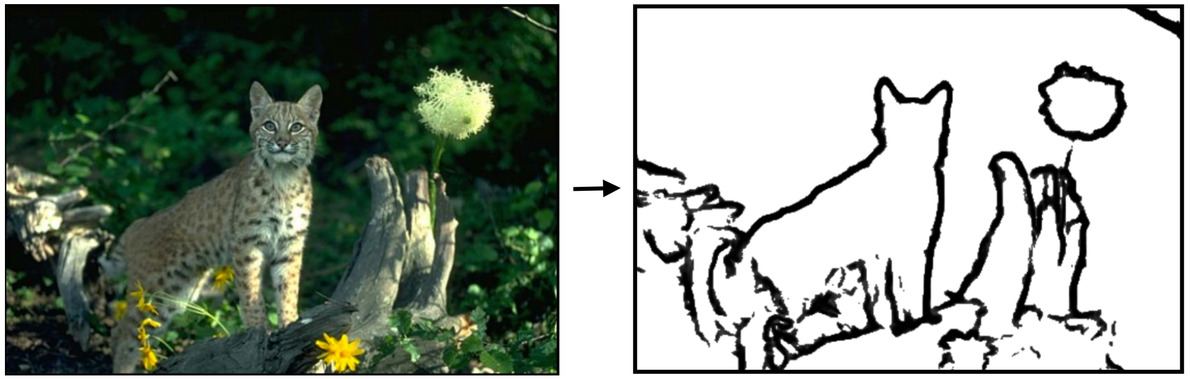
Data translation problems ("structured prediction")

Semantic segmentation



[Long et al. 2015, …]

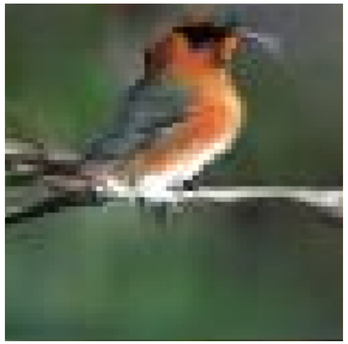
Edge detection



[Xie et al. 2015, …]

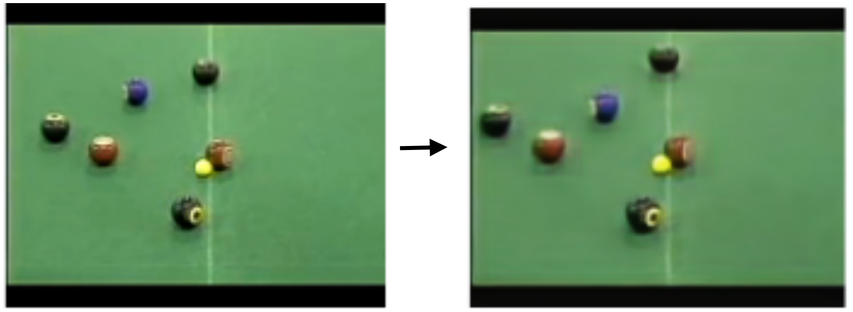
Text-to-photo

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



[Reed et al. 2014, …]

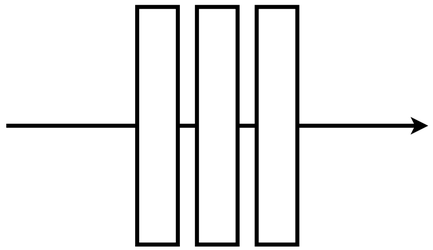
Future frame prediction



[Mathieu et al. 2016, …]

$\mathbf{x}$

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

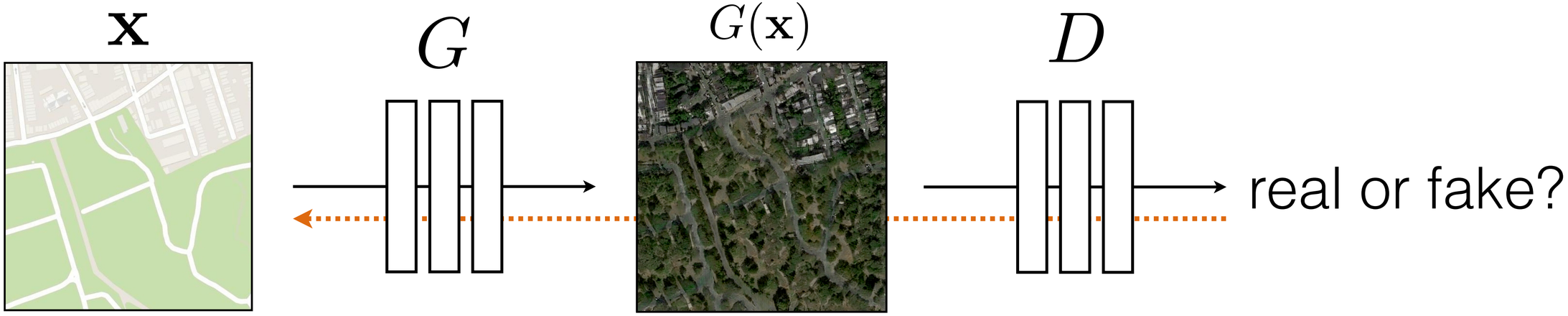
fake (0.9)

$\mathbf{y}$

D

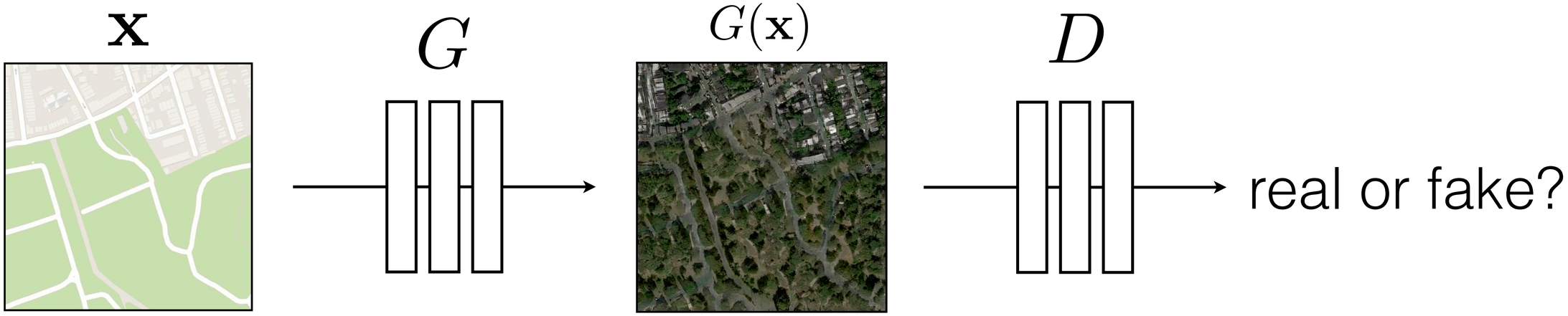
real (0.1)

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



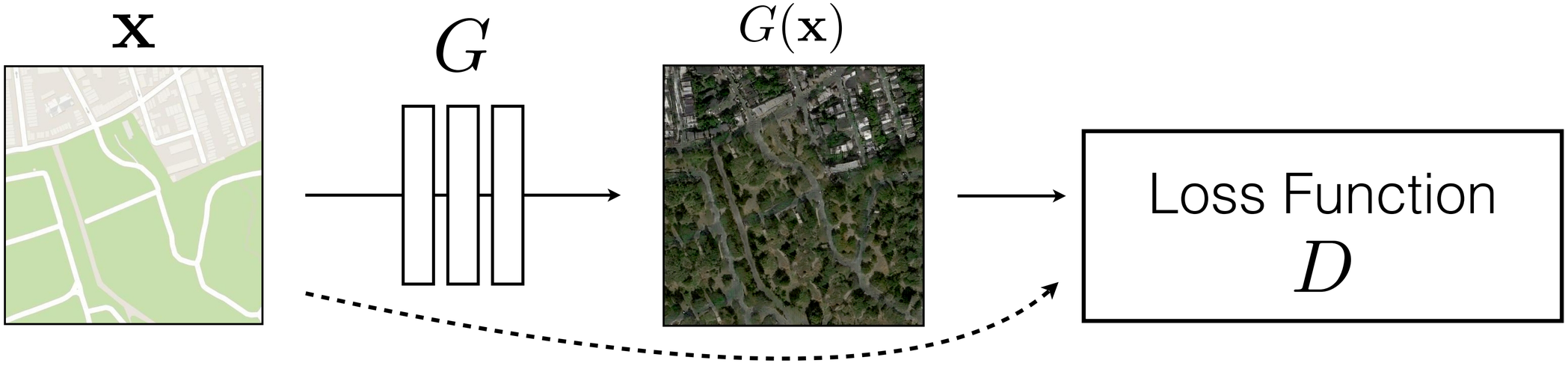
G tries to synthesize fake images that ***fool*** **D**:

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



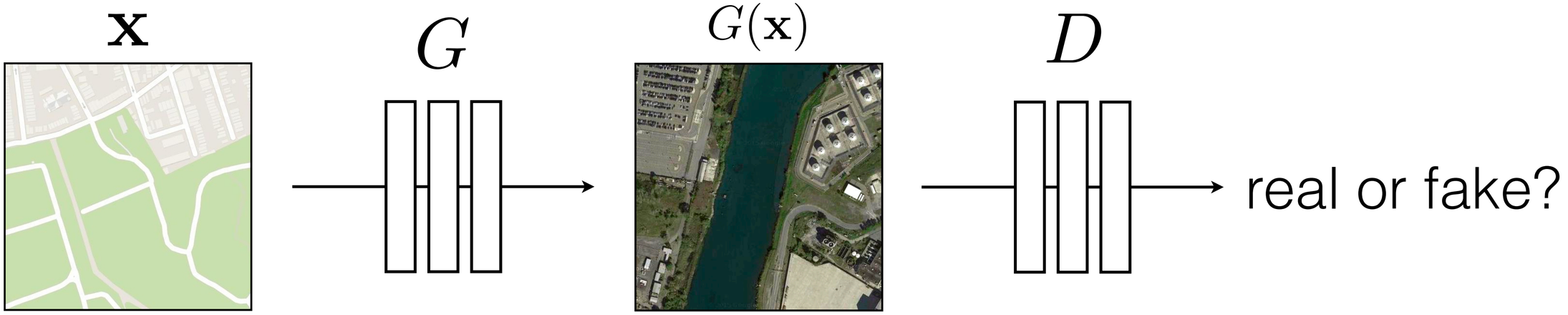
G tries to synthesize fake images that ***fool*** the ***best*** **D**:

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

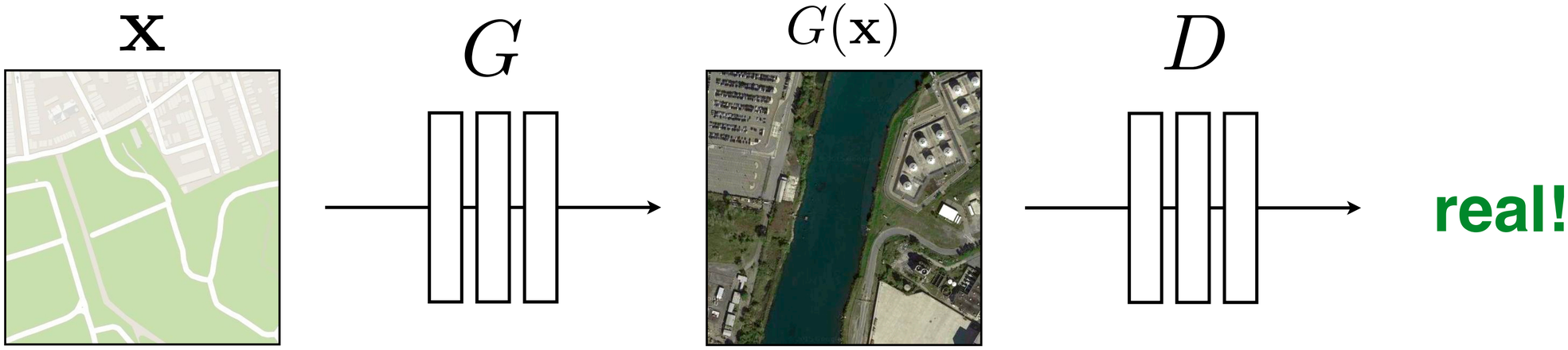


G's perspective: **D** is a loss function.

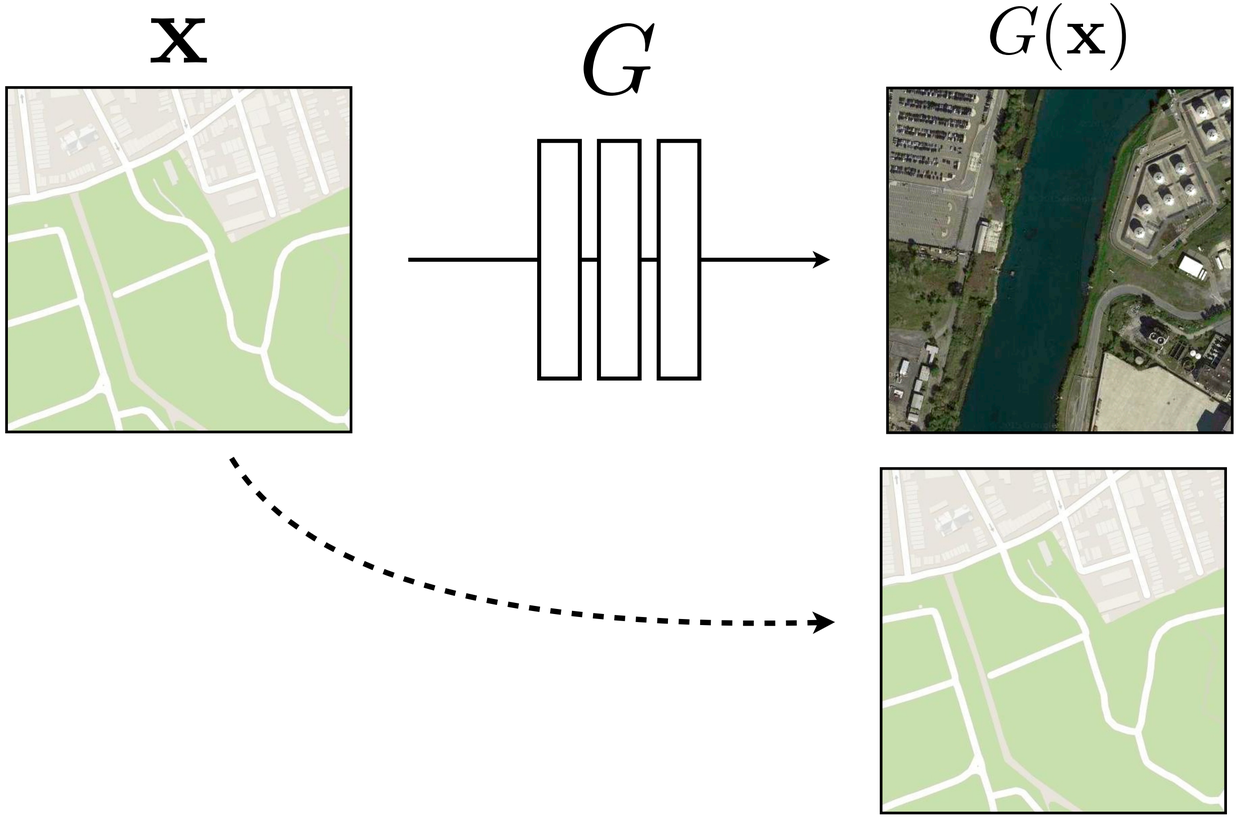
Rather than being hand-designed, it is *learned* and *highly structured*.



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



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



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

$\mathbf{x}$

G

$G(\mathbf{x})$

D

real or fake *pair* ?

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

$\mathbf{x}$

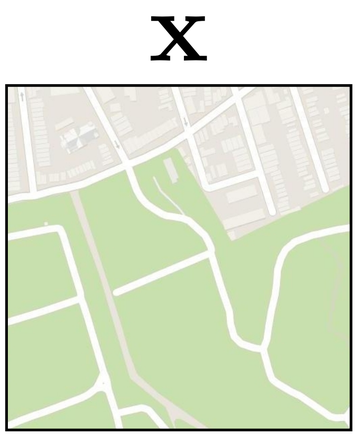
G

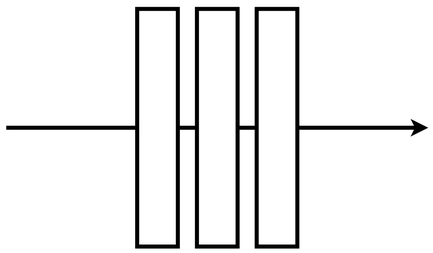
$G(\mathbf{x})$

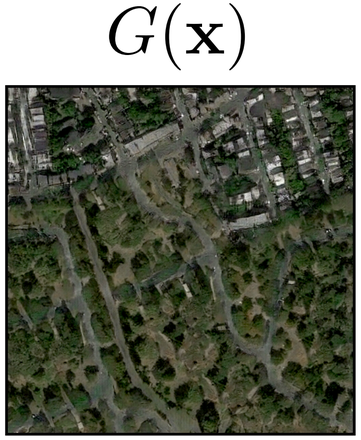
D

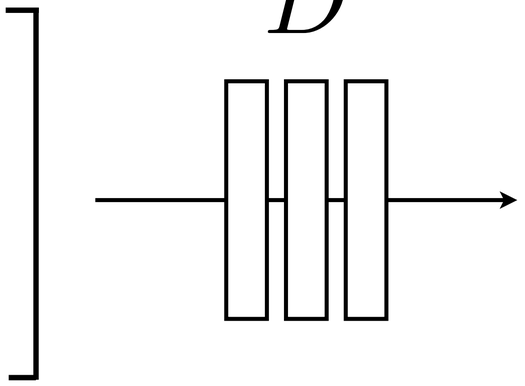
fake *pair*

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



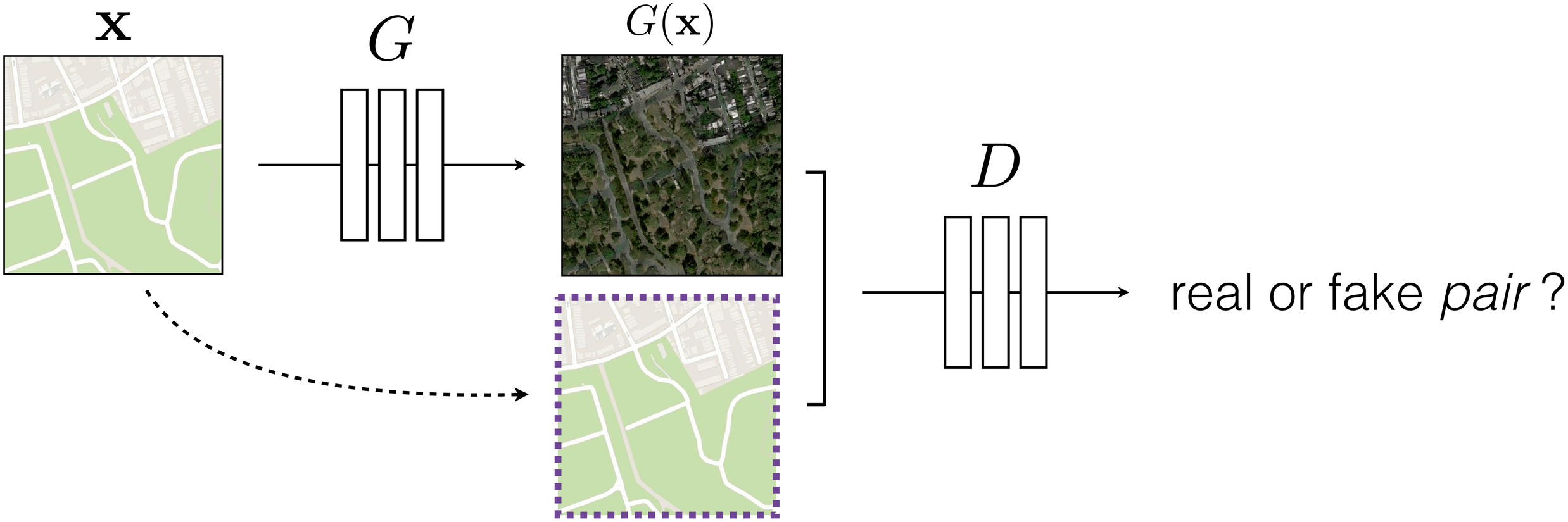






real *pair*

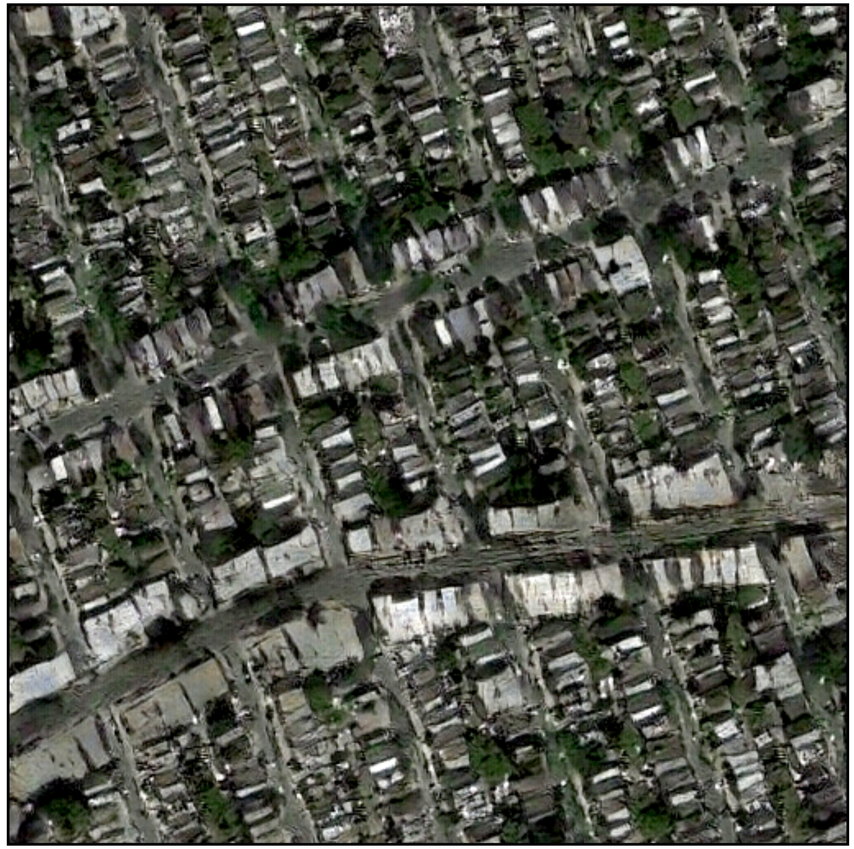
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$



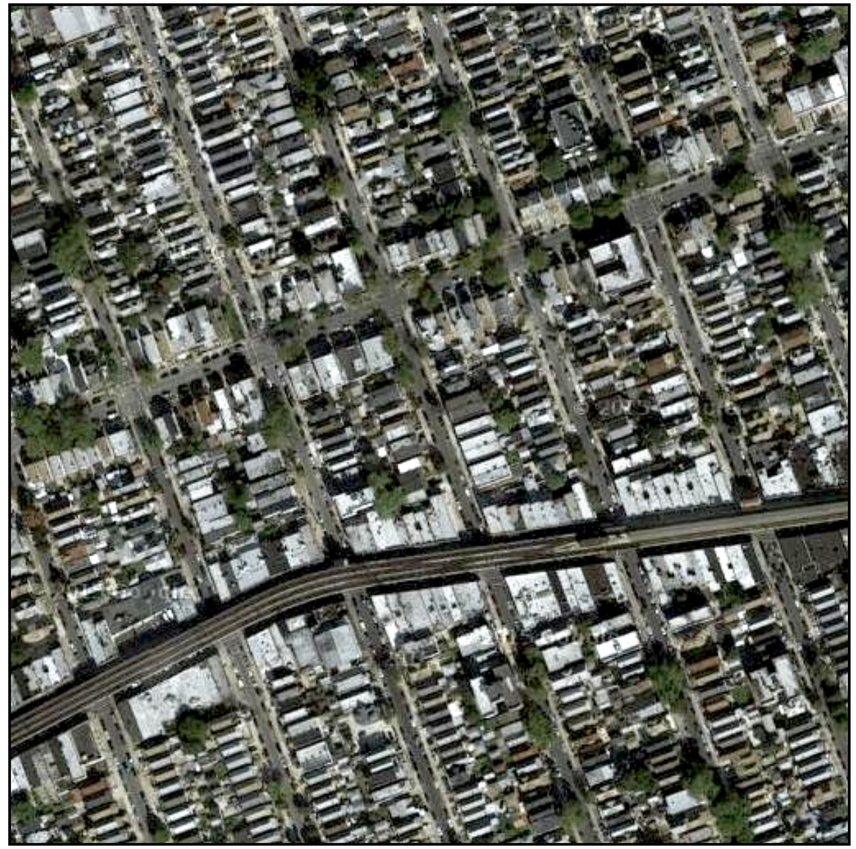
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\;\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;\;]$$

Input

Output



Groundtruth



Data from
[maps.google.com]

Input

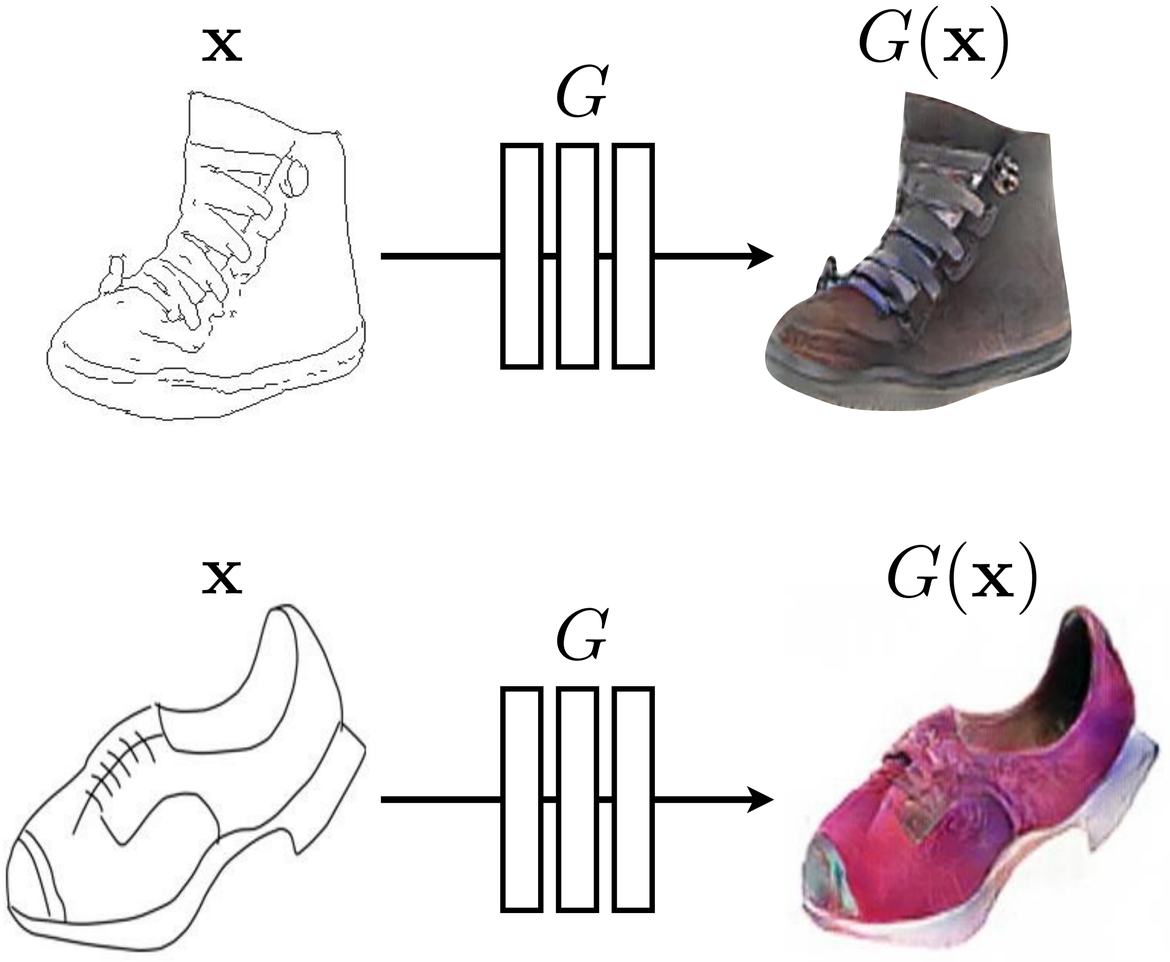
Output

Groundtruth

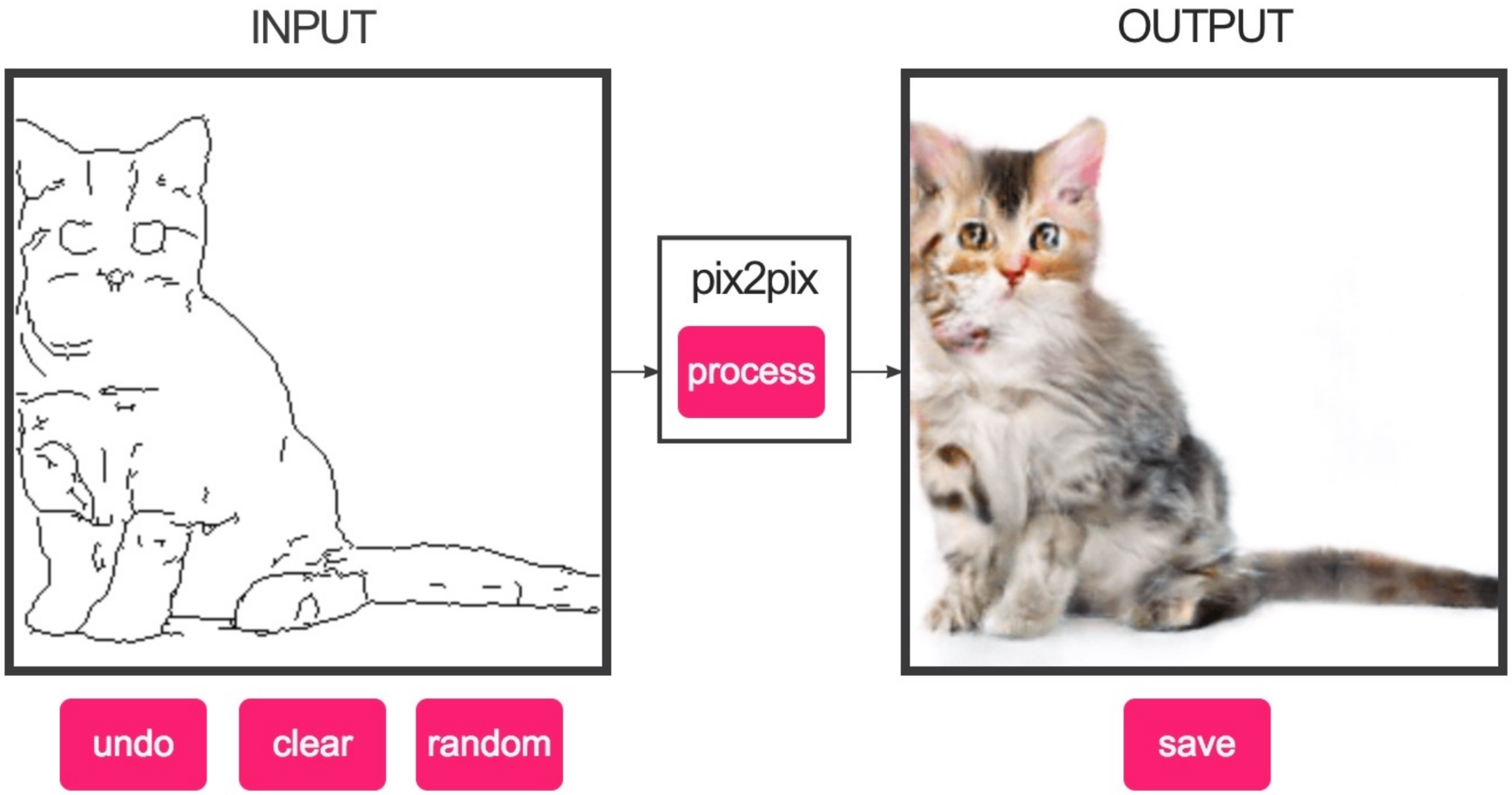
Data from [maps.google.com]

Training data

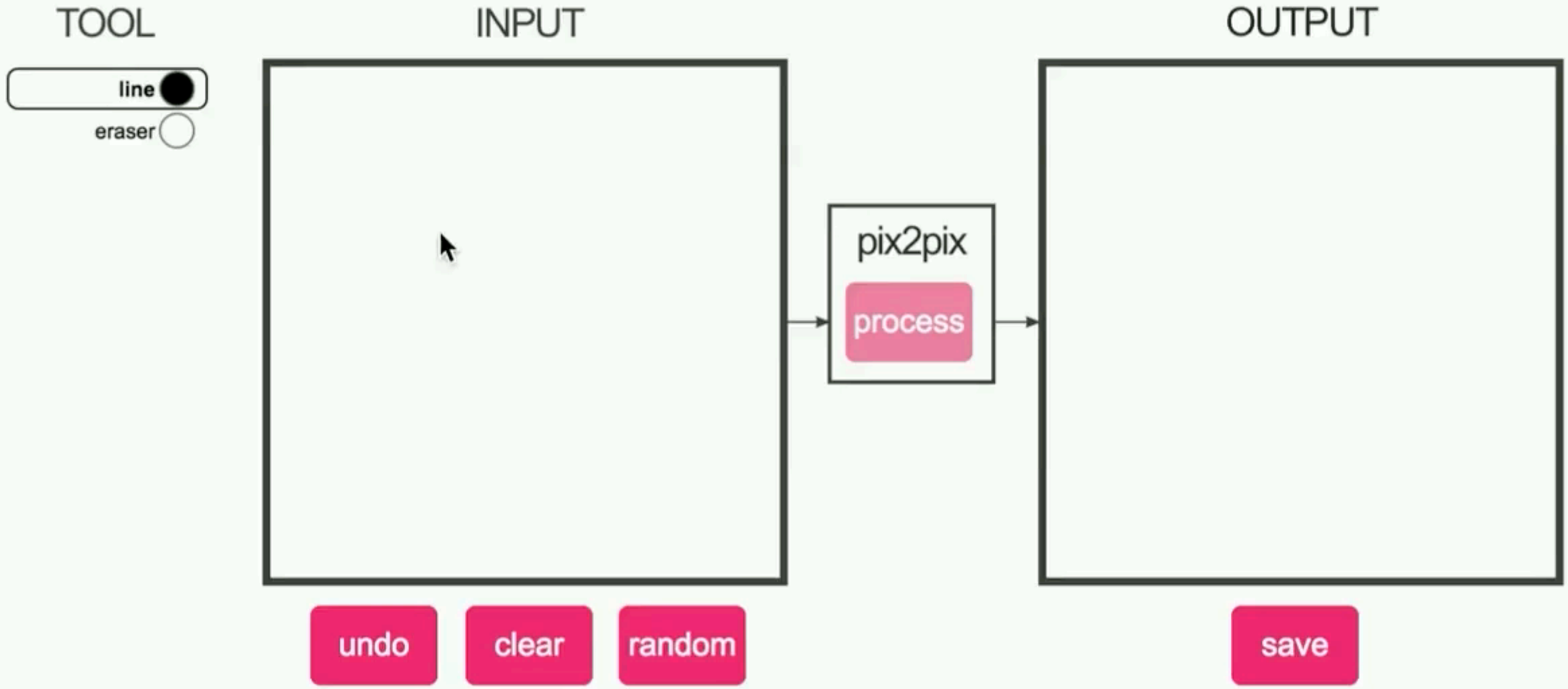
[HED, Xie & Tu, 2015]

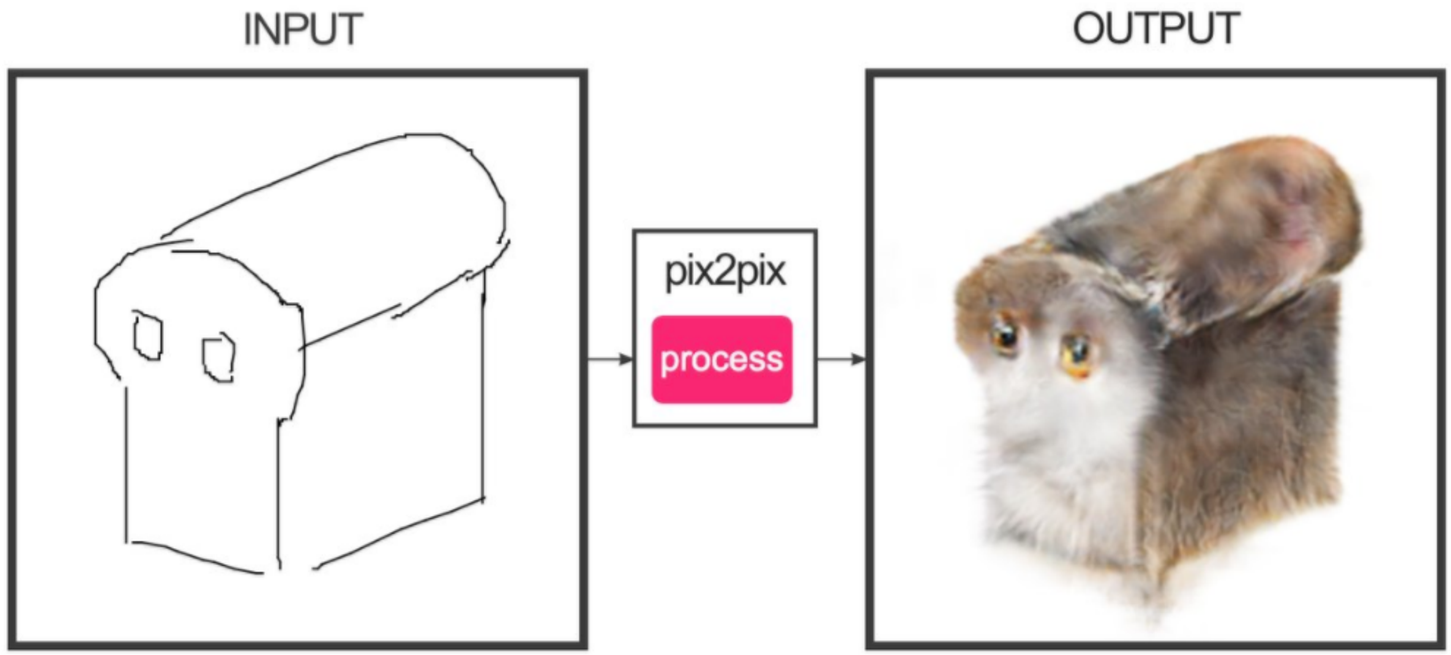


#edges2cats [Chris Hesse]



edges2cats


TOOL
line
eraser
INPUT
pix2pix
process
OUTPUT
undo
clear
random
save



Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

Leveraging pretrained models for efficient data translation

The point of deep learning is to enable learning with little data

Deep learning

Representations
(encoders)

Models
(decoders)

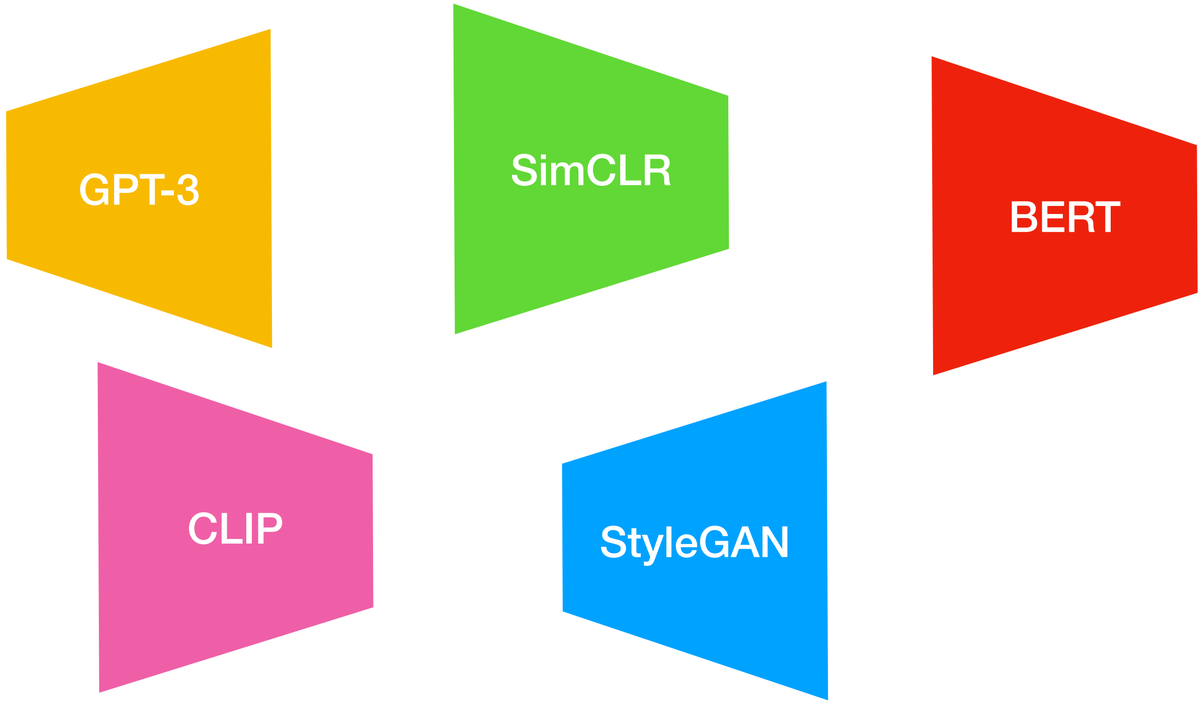
Foundation models

[Bommasani et al. 2021] https://arxiv.org/pdf/2108.07258.pdf

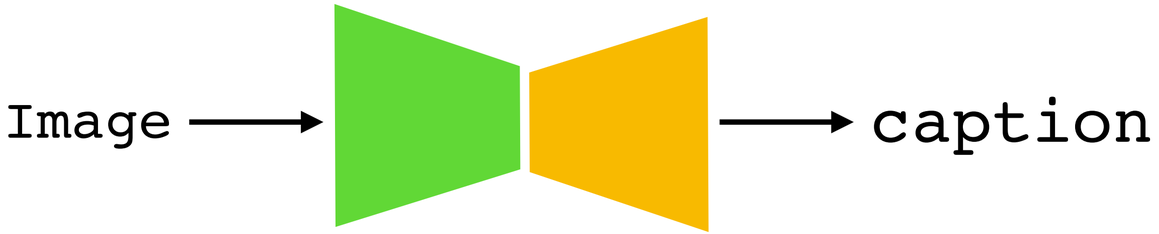
"If I have seen further it is by standing on the shoulders of Giants"
— Newton

[*Blind Orion Searching for the Rising Sun* by Nicolas Poussin, 1658]

1. Learn foundation model **encoders** and **decoders** for each domain

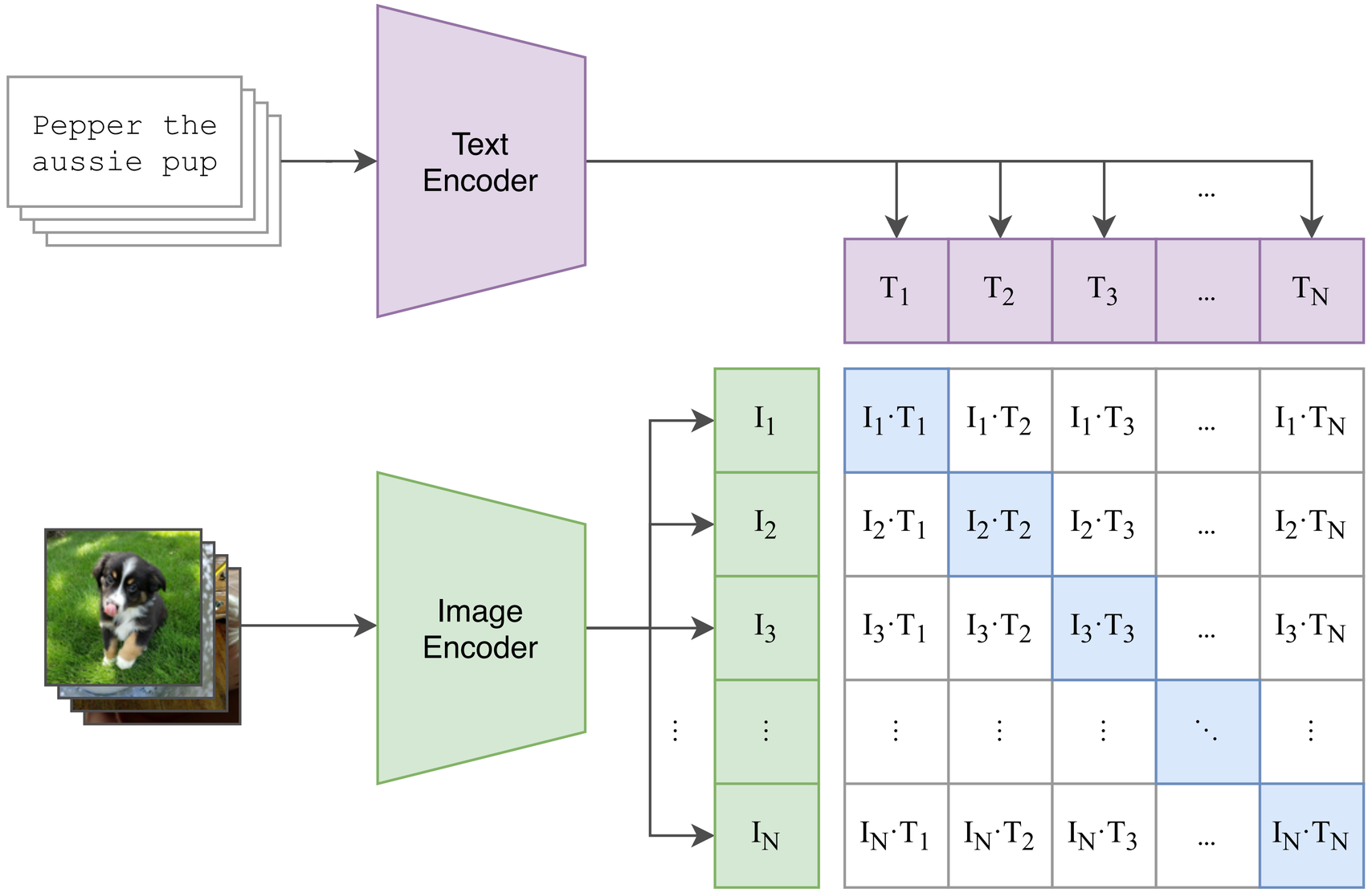


2. Plug them together to translate between modalities (may require finetuning)



CLIP

[Radford et al., 2021] https://arxiv.org/pdf/2103.00020.pdf

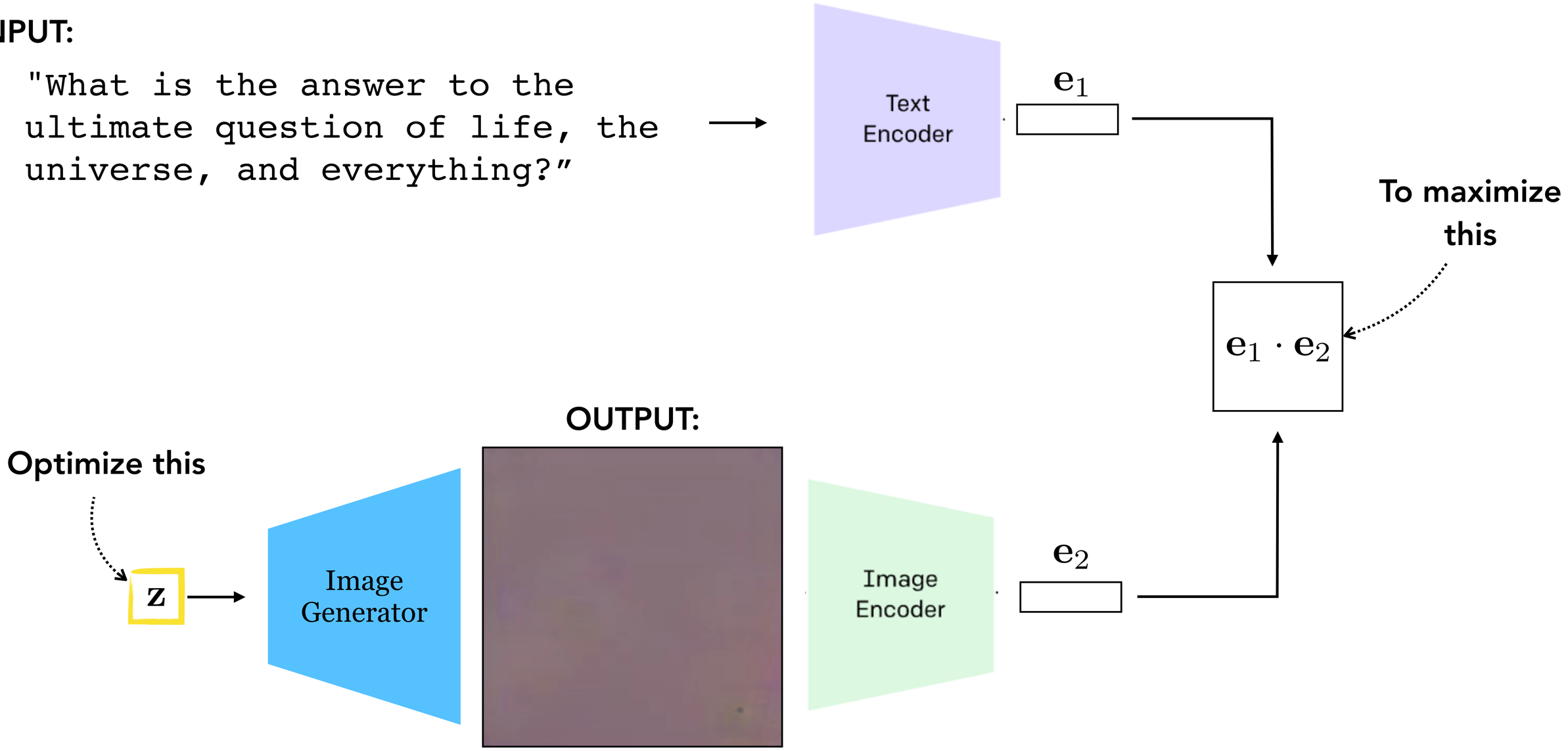


[https://openai.com/blog/clip/]

New capabilities by plugging pretrained models together: CLIP+GAN

INPUT:

"What is the answer to the ultimate question of life, the universe, and everything?"

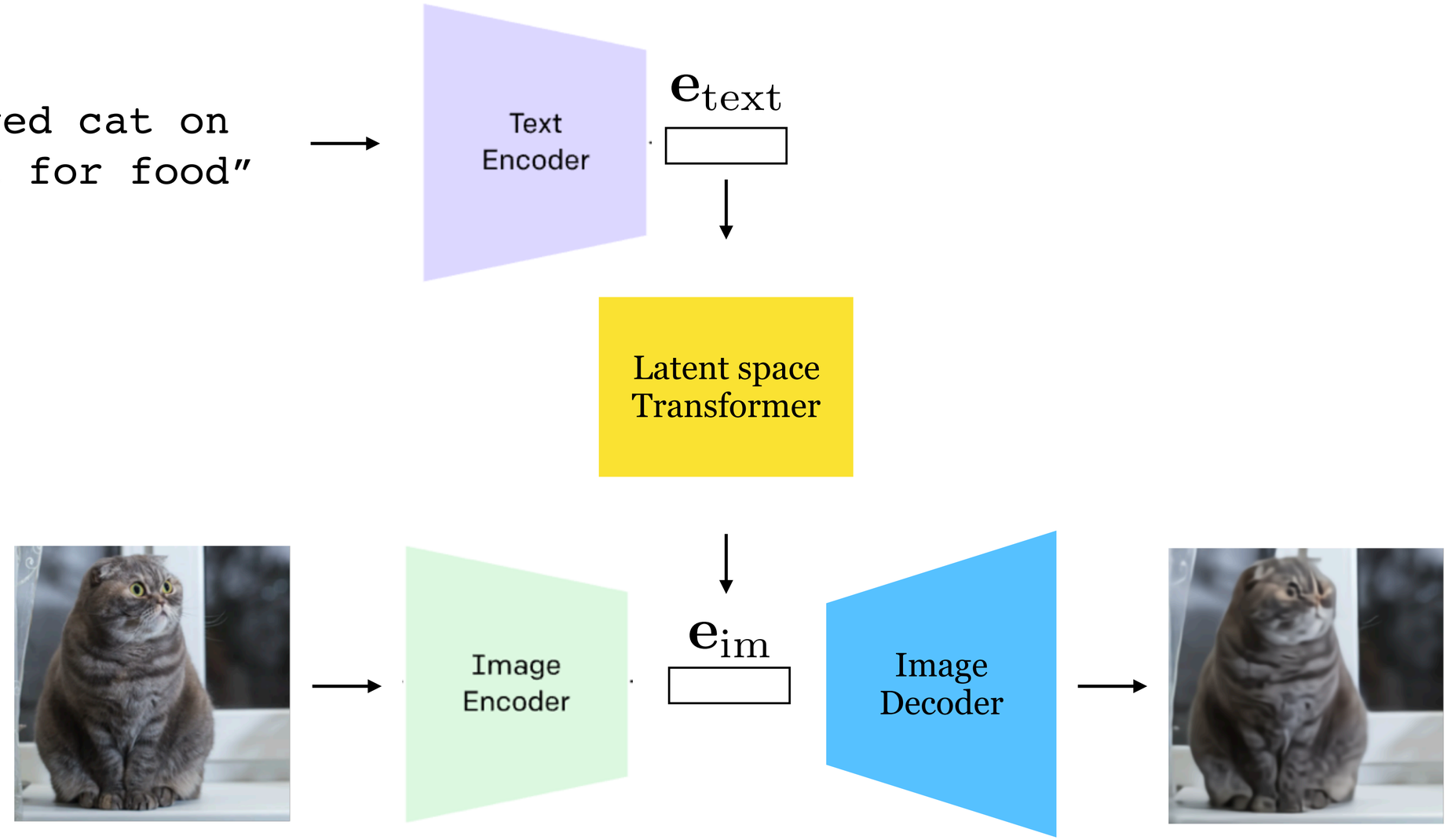


Code: https://colab.research.google.com/drive/1_4PQqzM_0KKytCzWtn-ZPi4cCa5bwK2F?usp=sharing

DALL-E [Ramesh et al. 2021] https://arxiv.org/pdf/2102.12092.pdf https://openai.com/blog/dall-e/

INPUT:

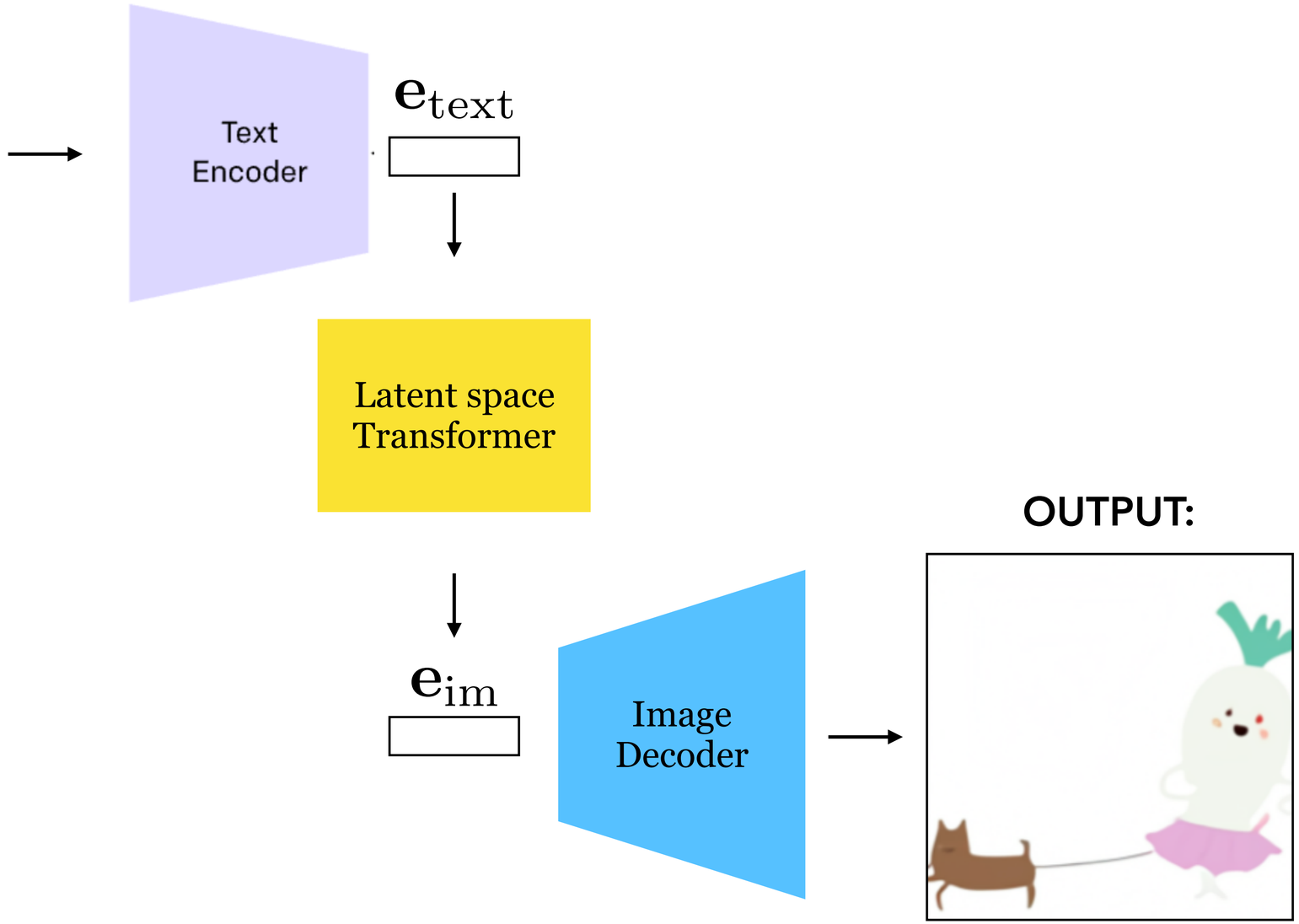
"A wide-eyed cat on the lookout for food"



Text-to-image translation

INPUT:

"An illustration of a baby daikon radish in a tutu walking a dog"



New capabilities by just asking: product design

TEXT PROMPT

an armchair in the shape of an avocado. an armchair imitating an avocado.

AI-GENERATED IMAGES

New capabilities by just asking: image translation

TEXT PROMPT

the exact same cat on the top as a sketch on the bottom

AI-GENERATED IMAGES

AI-GENERATED IMAGES

TEXT PROMPT a photo of a phone from the ...

AI-GENERATED IMAGES

TEXT PROMPT a photo of a computer from the ...

AI-GENERATED IMAGES

DALL-E 2

a painting of water lilies in a new art style no human has ever seen before

Report issue