

COMP527

COMPUTATIONAL IMAGING

Lecture #11 – Deep Generative Models

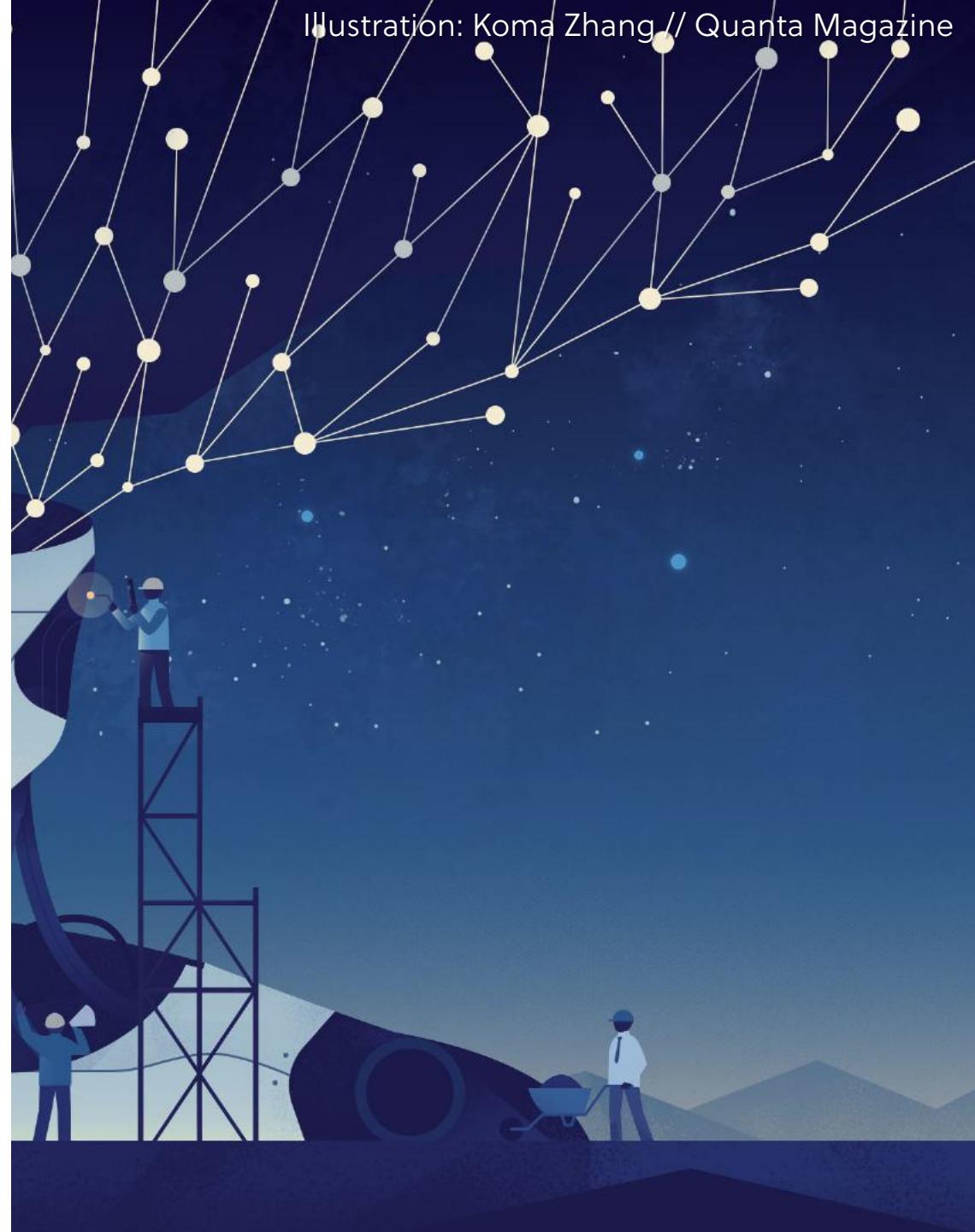


KOÇ
UNIVERSITY

Aykut Erdem // Koç University // Spring 2023

Previously on COMP527

- unreasonable effectiveness of data
- deep learning
- computation in a neural net
- optimization
- backpropagation
- convolutional neural networks
- applications in computational photography



Today's Lecture

- Discriminative vs. generative models
- Image synthesis
- Representation learning
- Data translation
- Applications in computational photography

Disclaimer: The material and slides for this lecture were borrowed from
—Bill Freeman, Antonio Torralba and Phillip Isola's MIT 6.869 class
—Philip Isola and Stefanie Jegelka's MIT 6.S898 class

Discriminative vs. Generative Models

Analysis

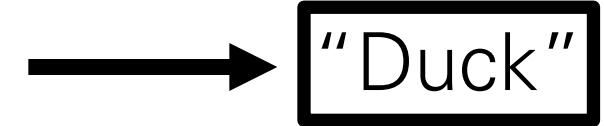
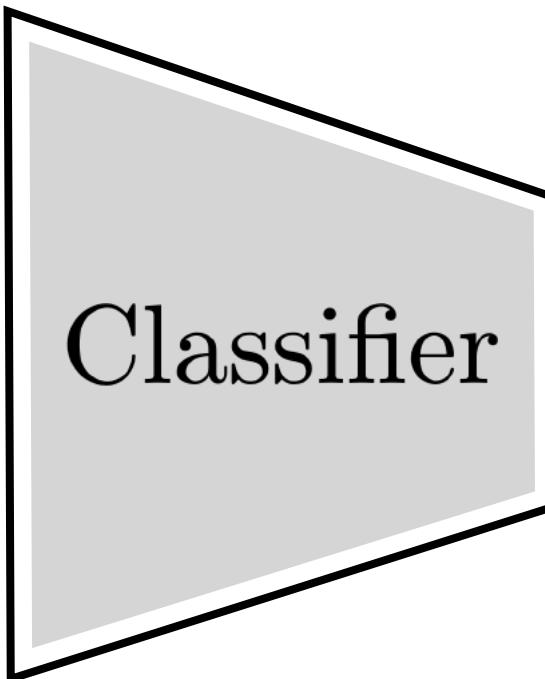
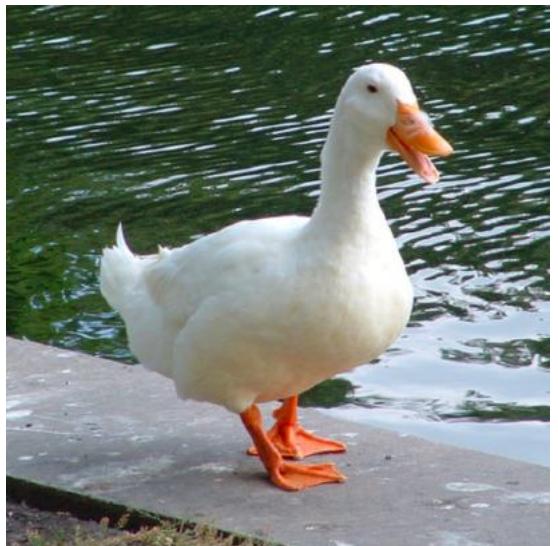
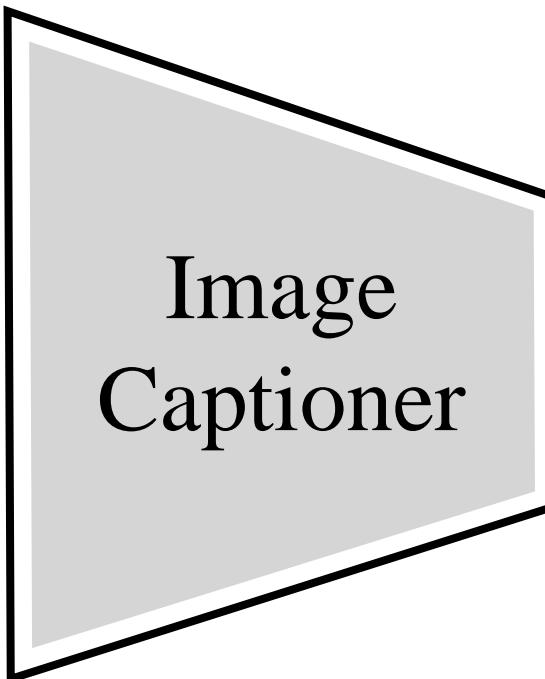
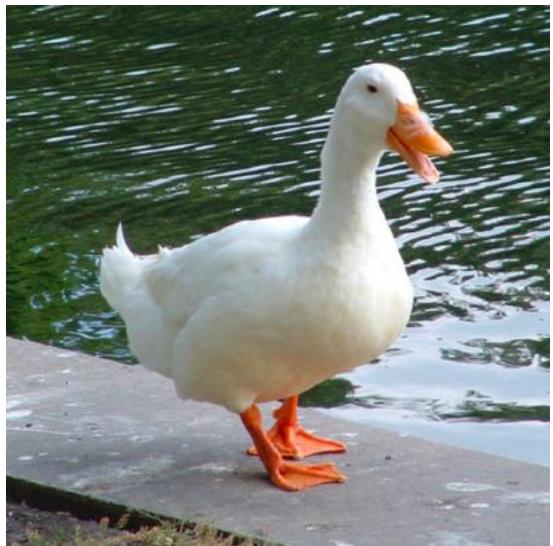


image x

label y

Analysis



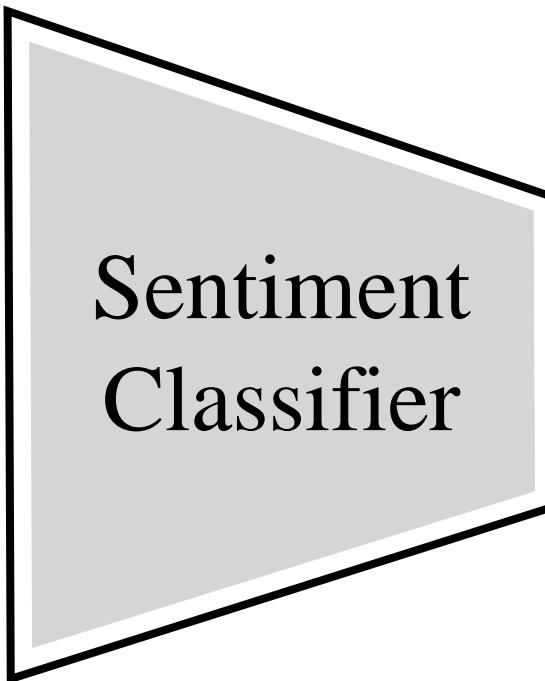
"A large duck
standing by
the river"

image x

caption y

Analysis

“A statuesque
duck gazing
gracefully over
the water”



sentence x

sentiment y

Analysis

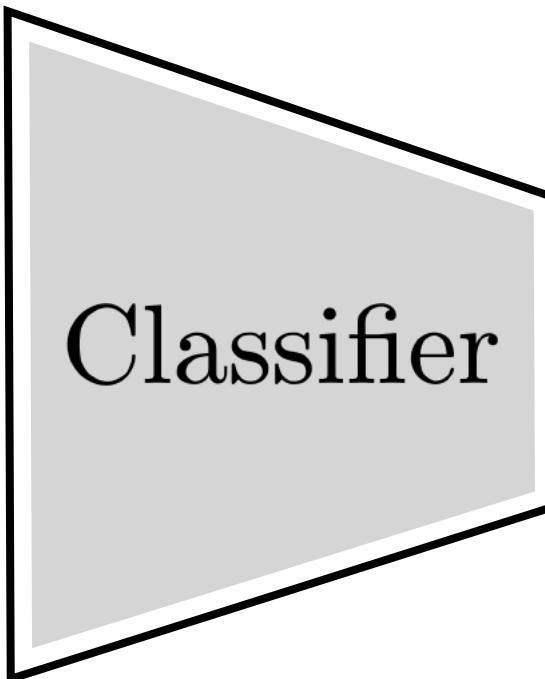
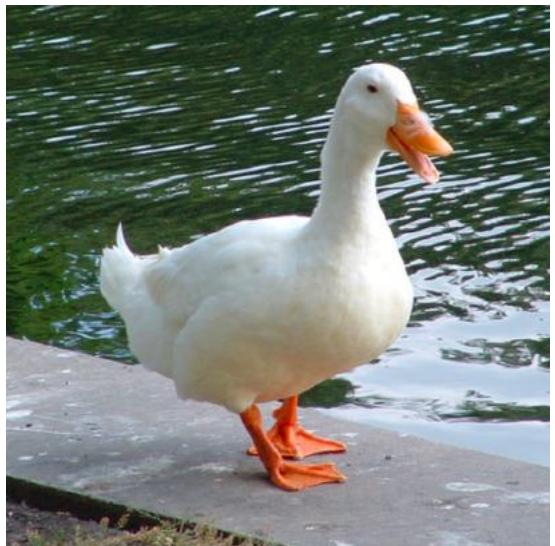


image x

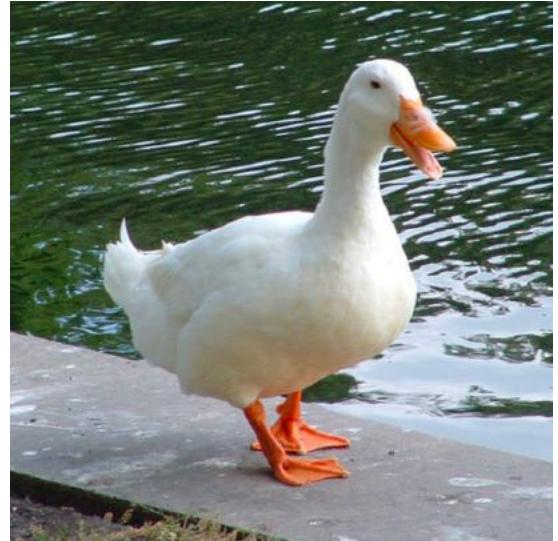
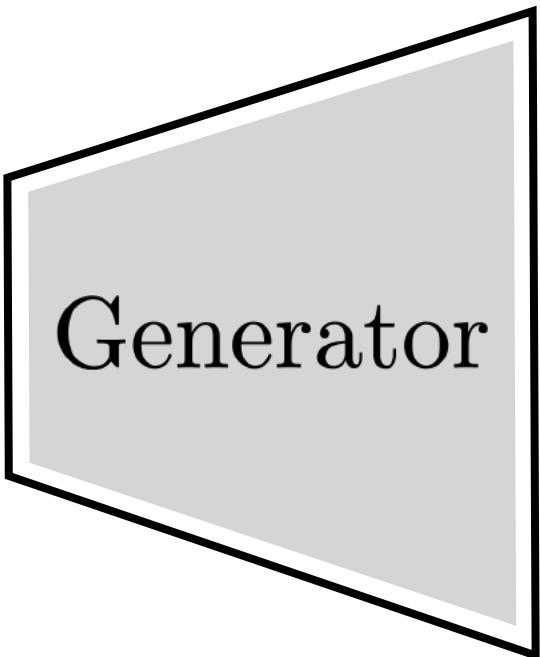


label y

"Duck"

Synthesis

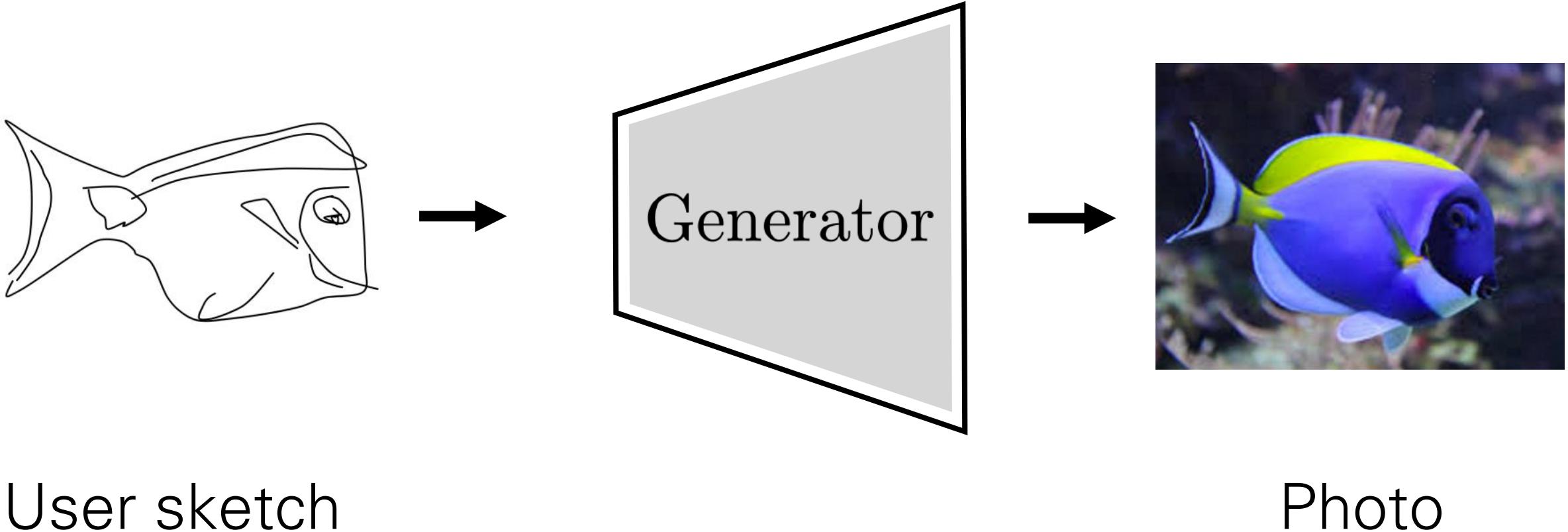
“Duck”



label y

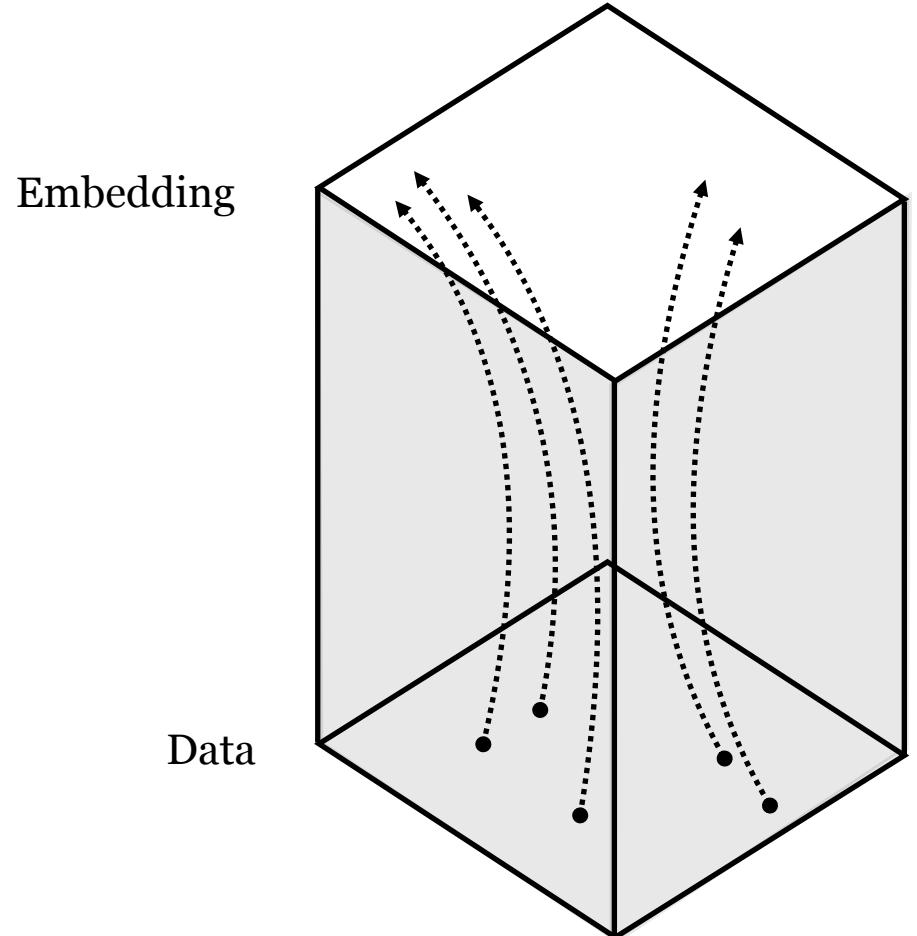
image x

Synthesis



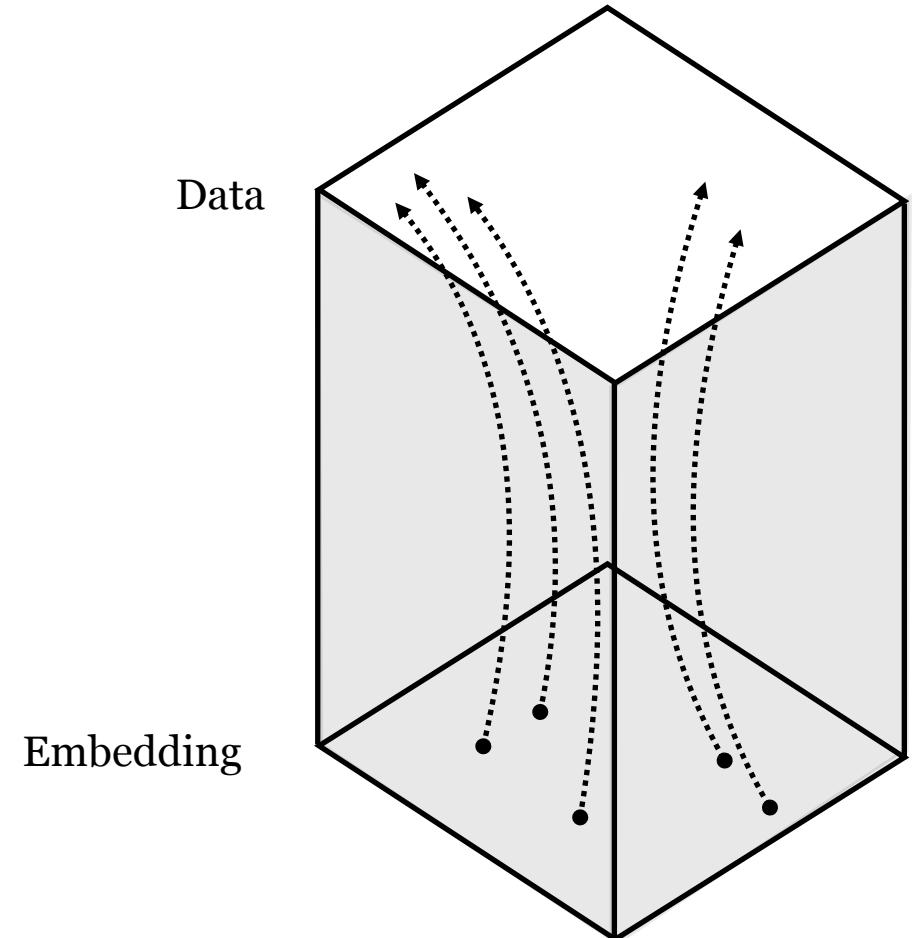
Deep nets are data transformers

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



Deep nets are data transformers

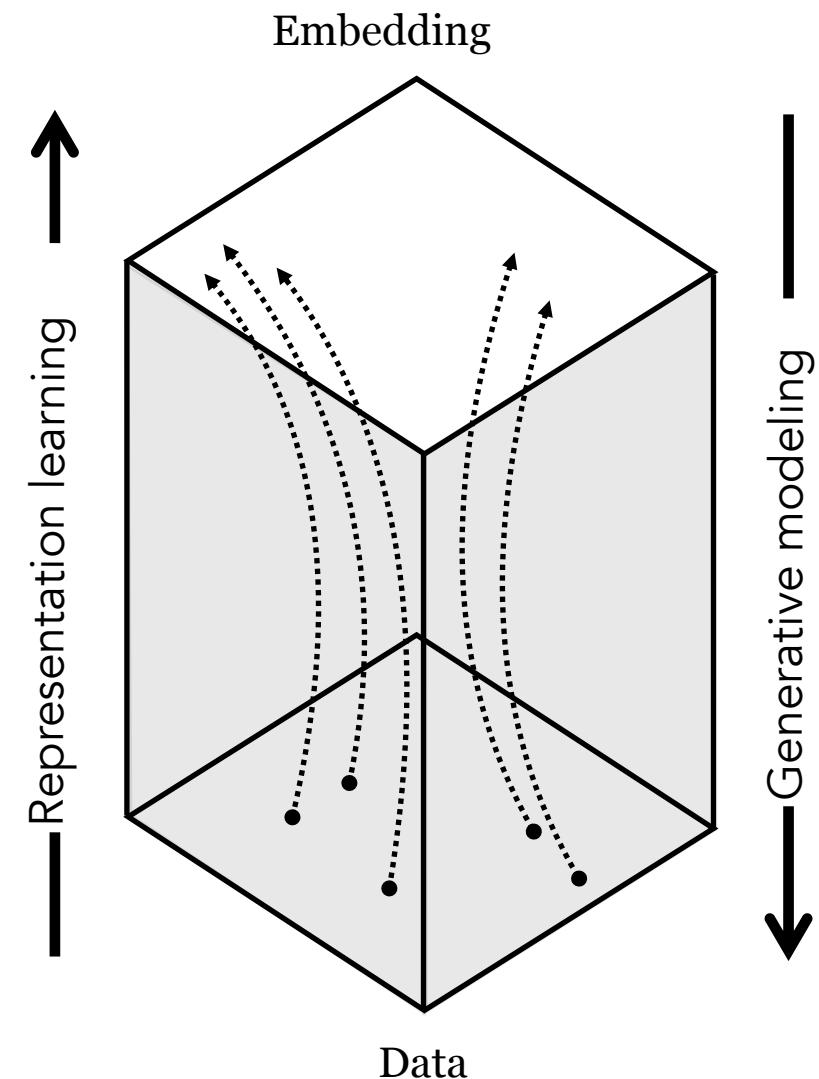
- Deep nets transform datapoints, layer by layer
- Each layer is a different representation of the data



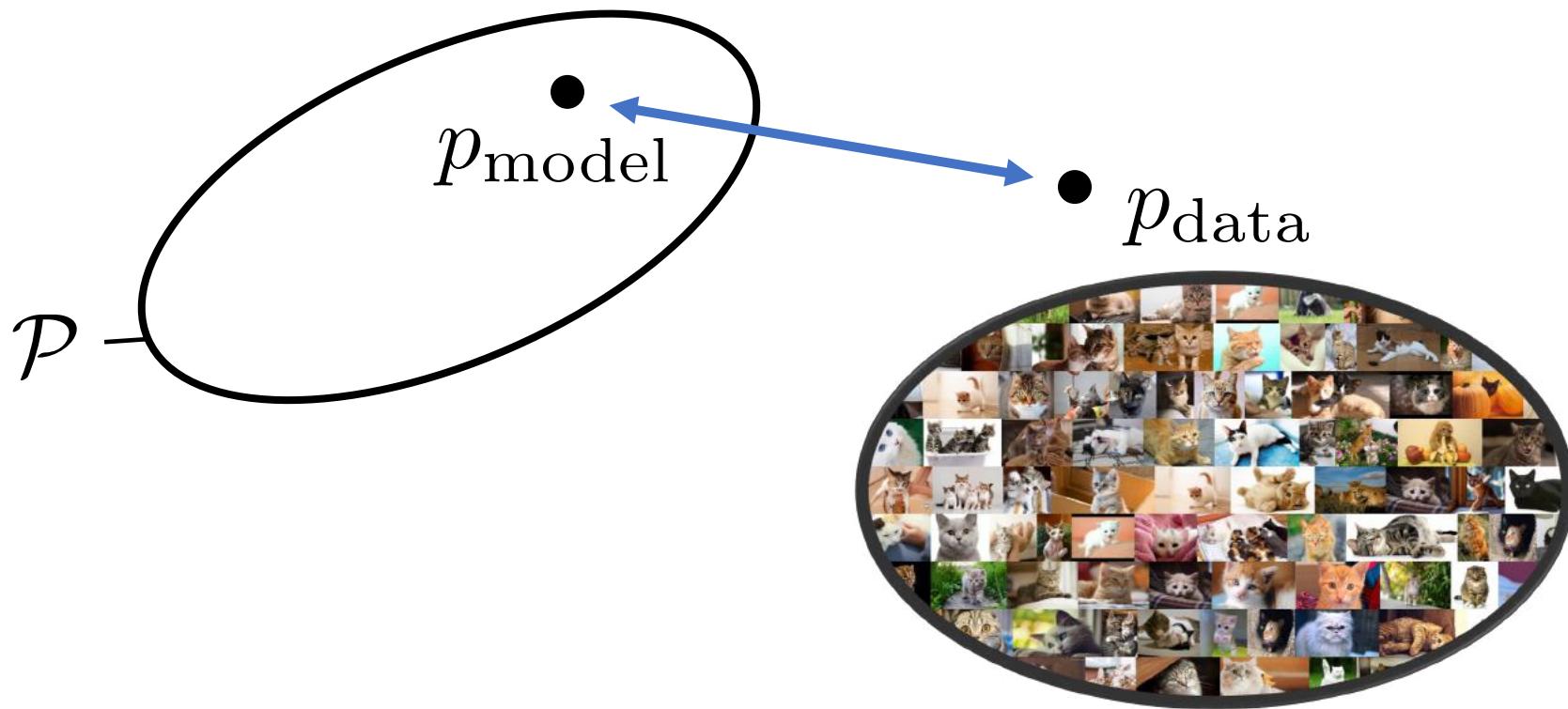
Generative modeling vs Representation learning

Representation learning:
mapping data to abstract representations
(analysis)

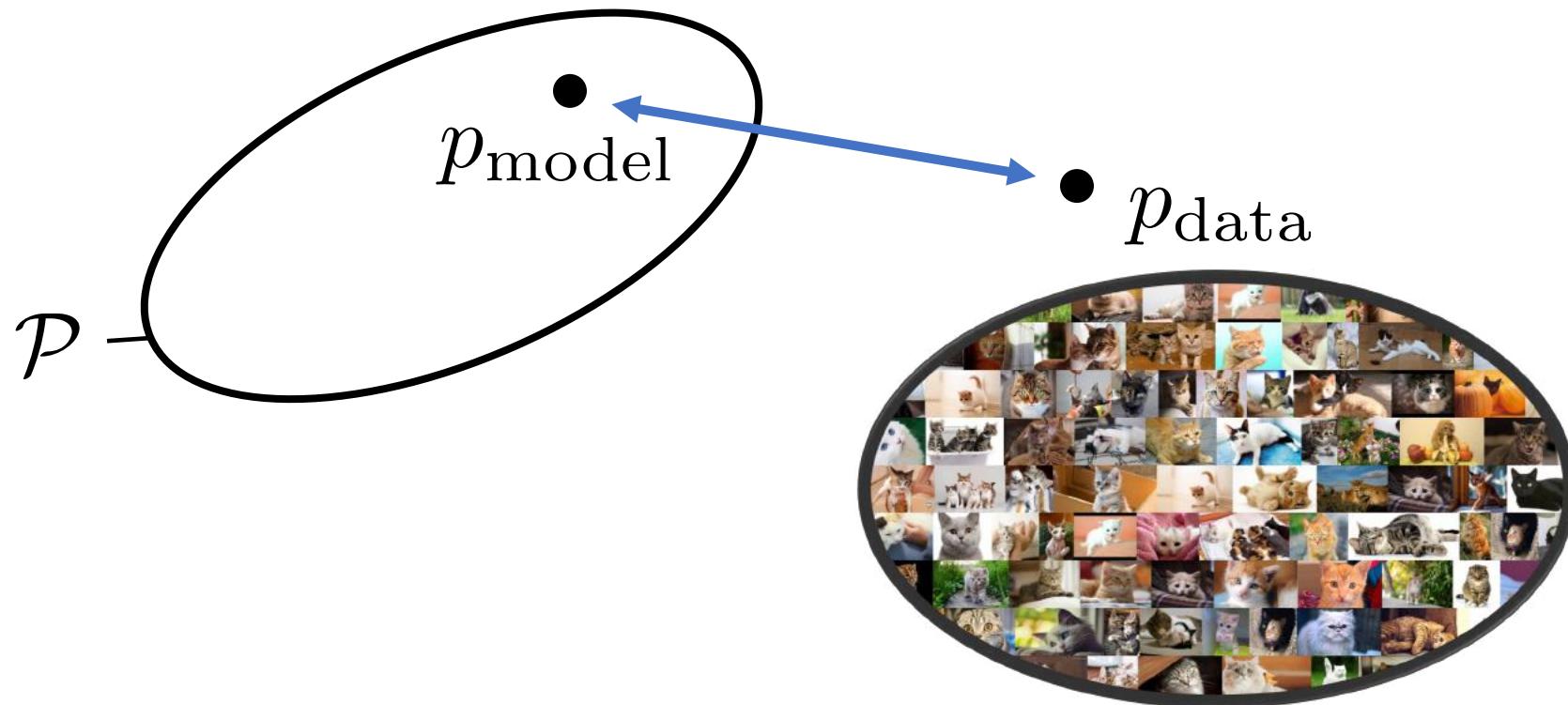
Generative modeling:
mapping abstract representations to data
(synthesis)



Generative Modeling

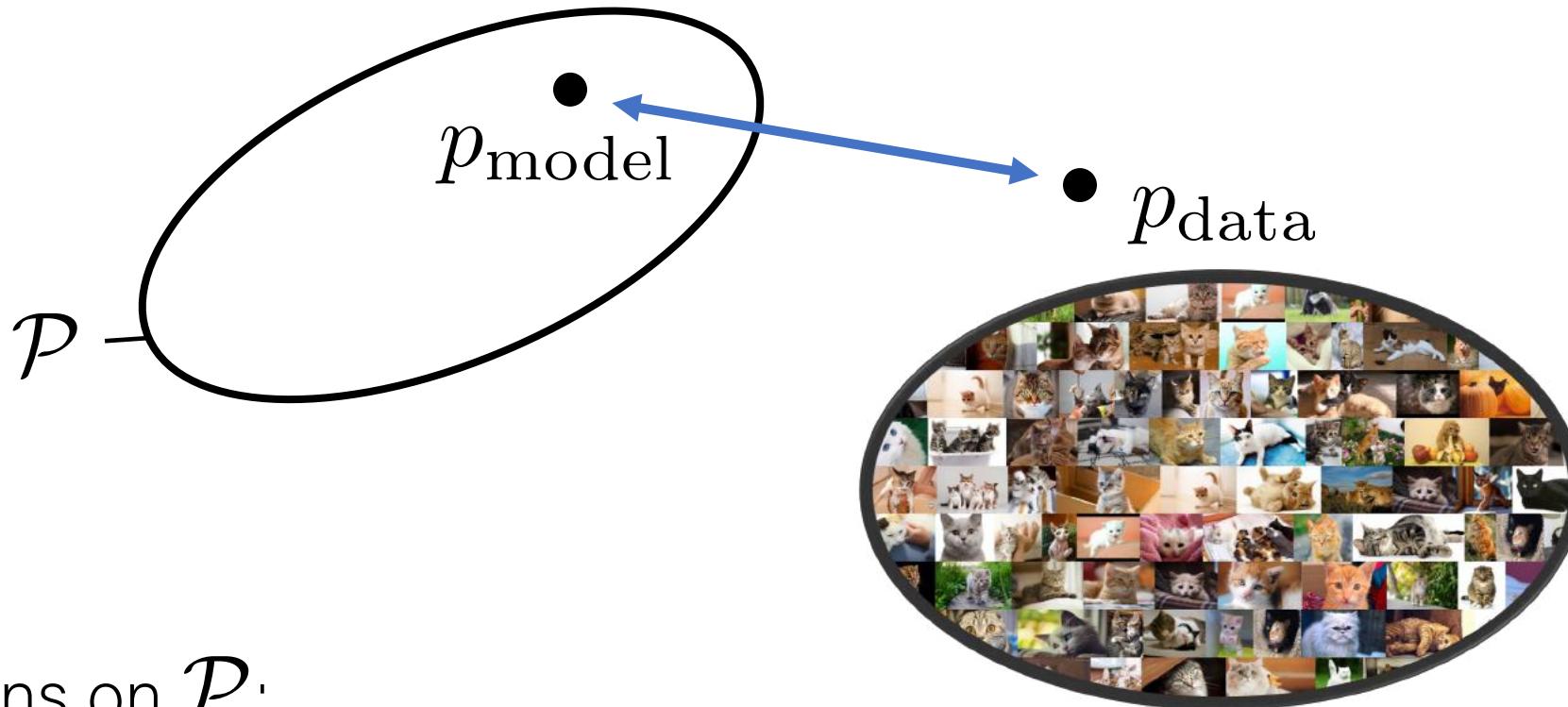


Generative Modeling



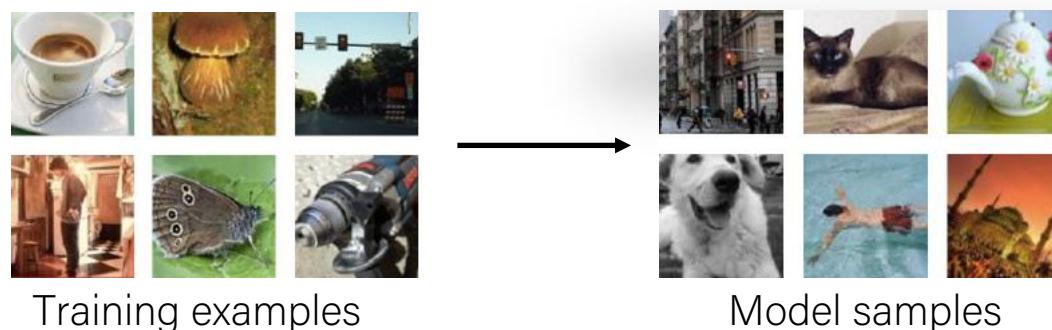
- Goal: Learn some underlying hidden structure of the training samples to generate novel samples from same data distribution

Generative Modeling

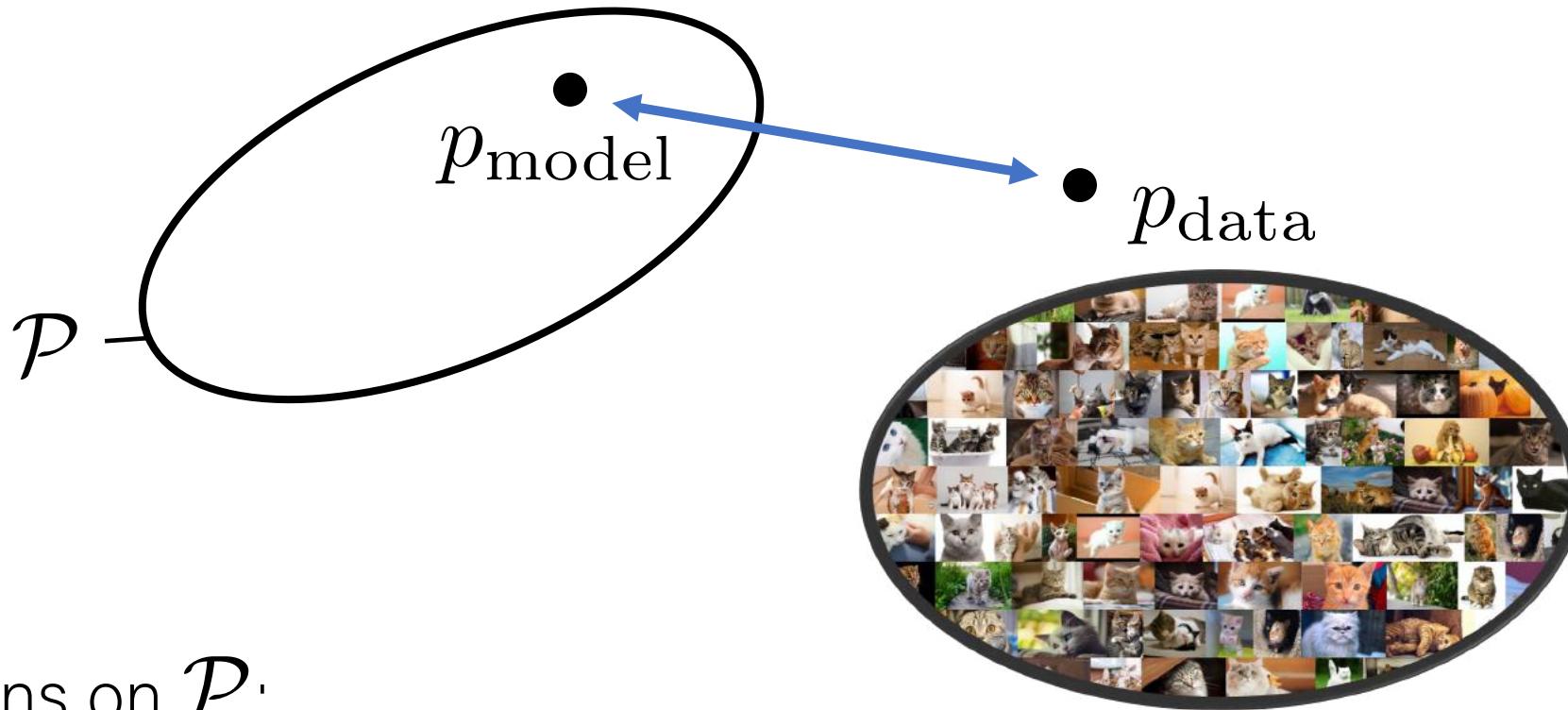


Assumptions on \mathcal{P} :

- tractable sampling

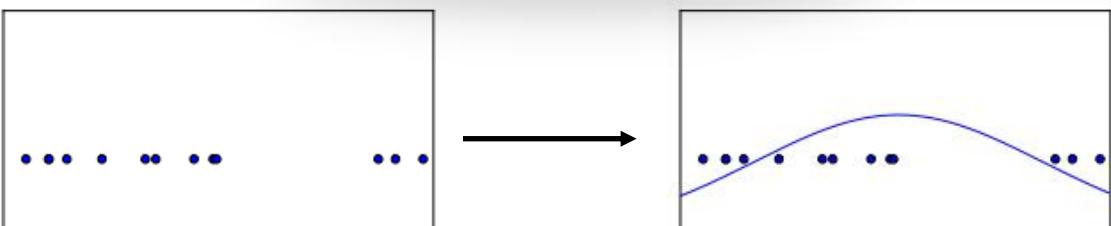


Generative Modeling



Assumptions on \mathcal{P} :

- tractable sampling
- tractable likelihood function





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

Image synthesis

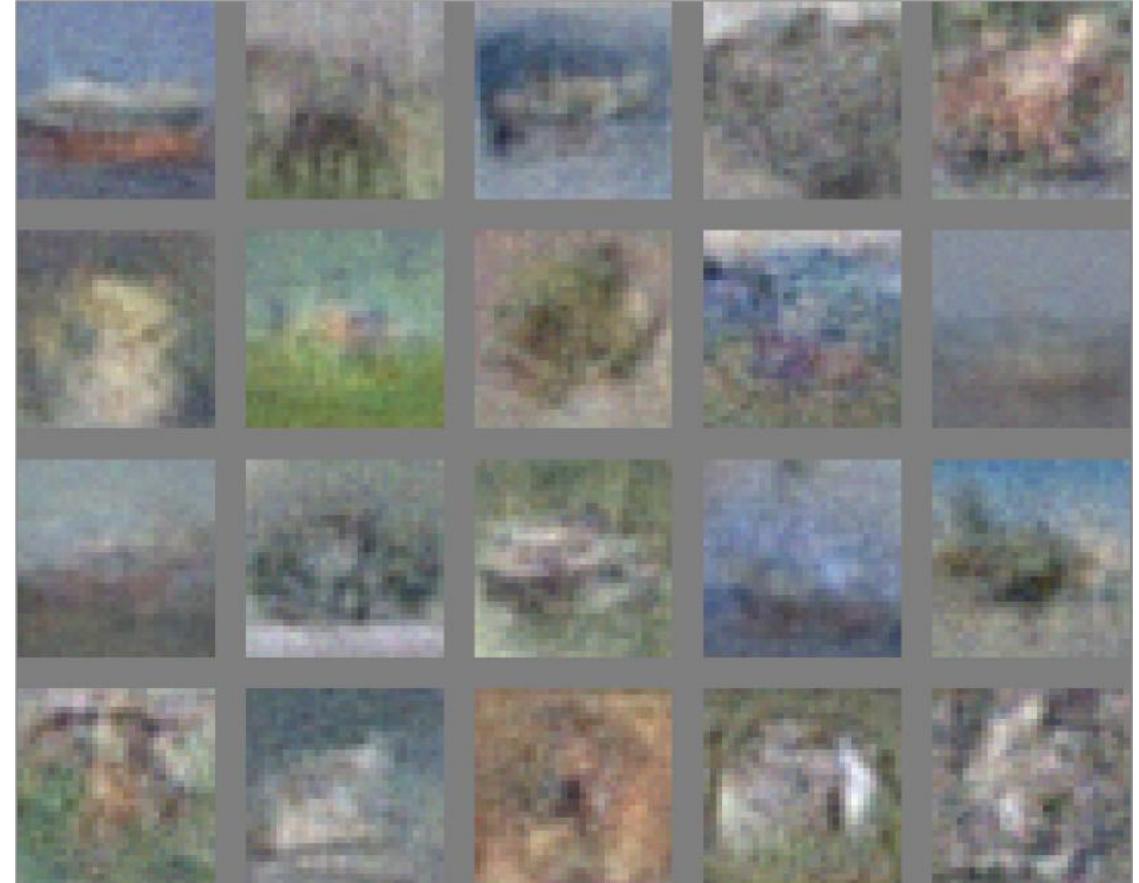
Generate Images



Generate Images

8 0 2 3 9 0 0
9 1 1 7 1 9 4
2 0 8 0 8 2 9
6 3 8 7 0 6 1
9 8 9 8 9 1 0
4 3 4 8 2 8 1

Generate Images



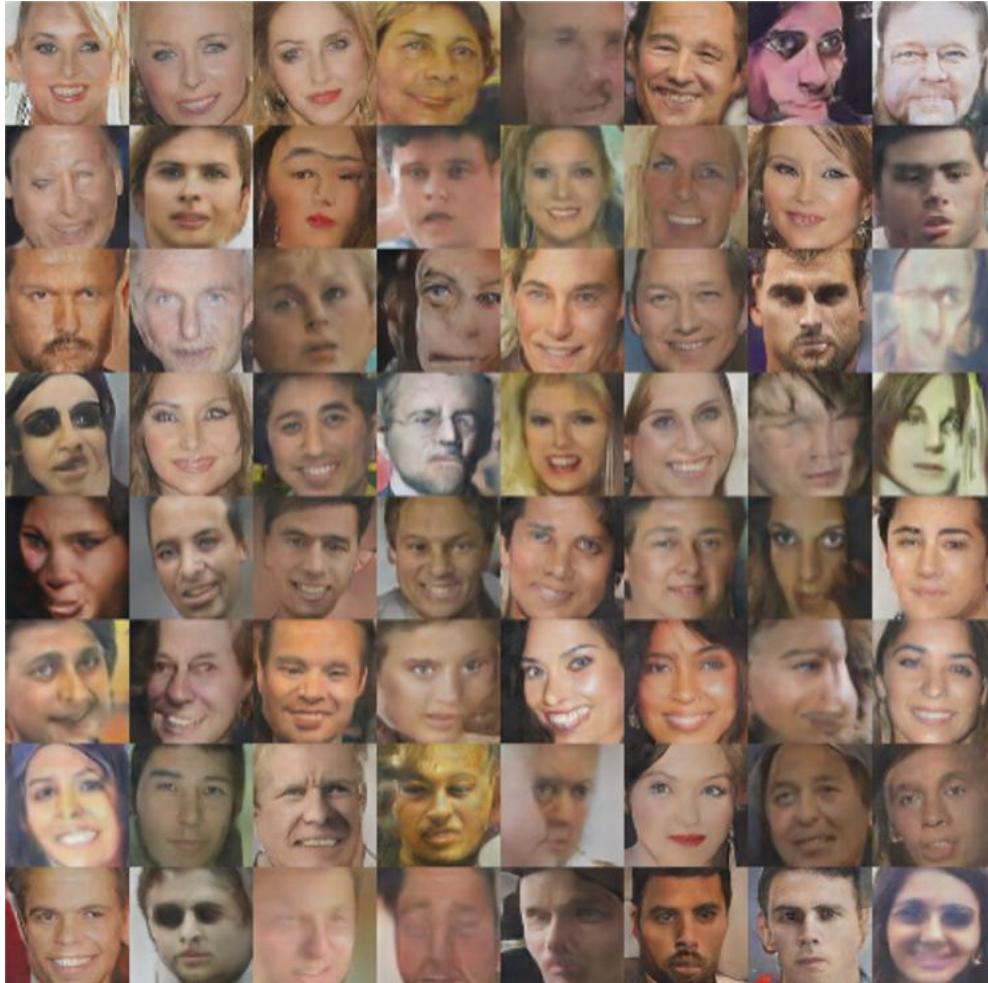
[GAN, Goodfellow et al. 2014]

Generate Images



[DCGAN, Radford, Metz, Chintala 2015]

Generate Images



[DCGAN, Radford, Metz, Chintala 2015]

Generate Images



Generate Images

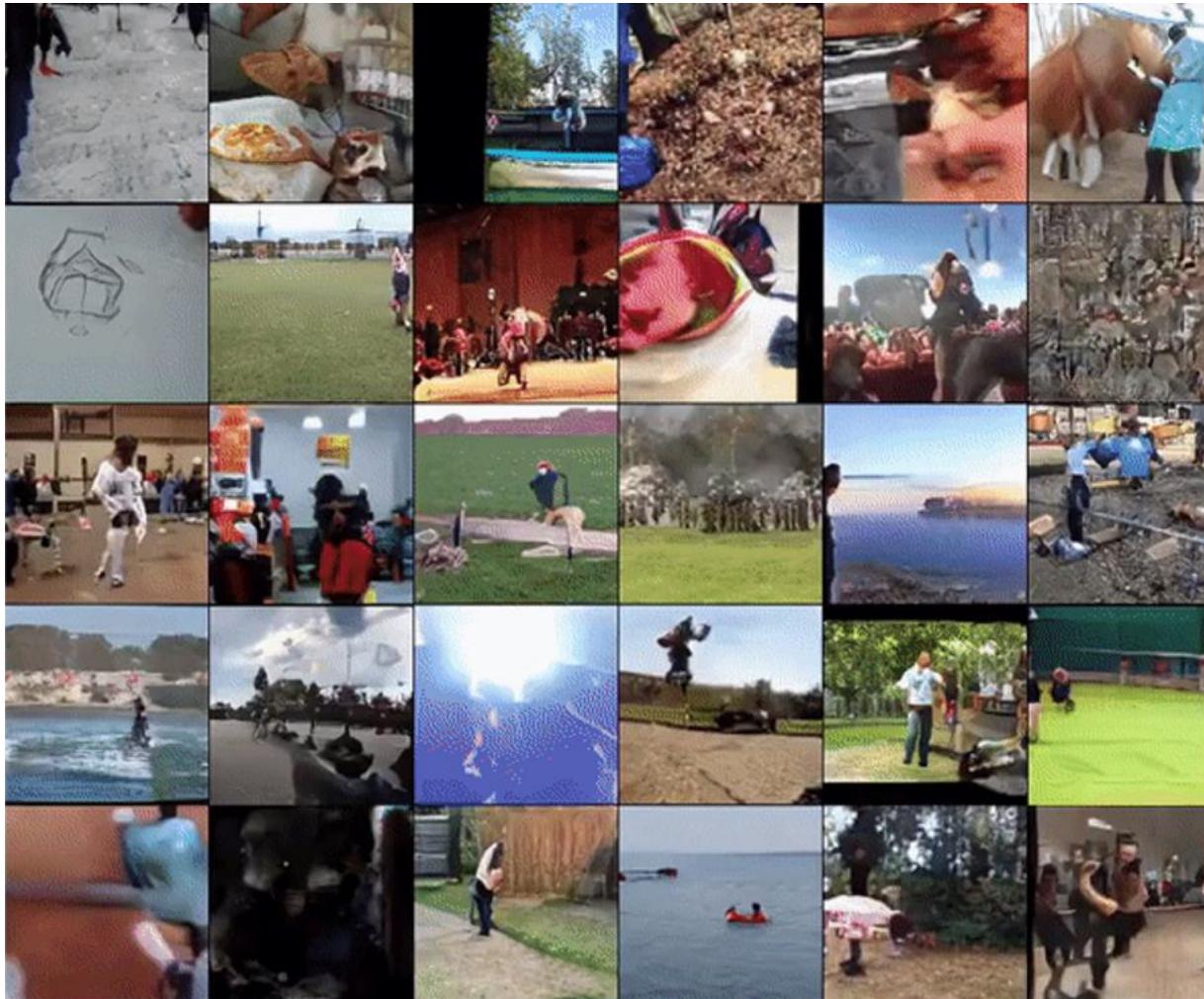


[BigGAN, Brock, Donahue, Simonyan, 2018]

Generate Images



Generate Video



DVD-GAN: Adversarial Video Generation on Complex Datasets, Clark,
Donahue, Simonyan, 2019

Procedural graphics





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! 🌱🍀

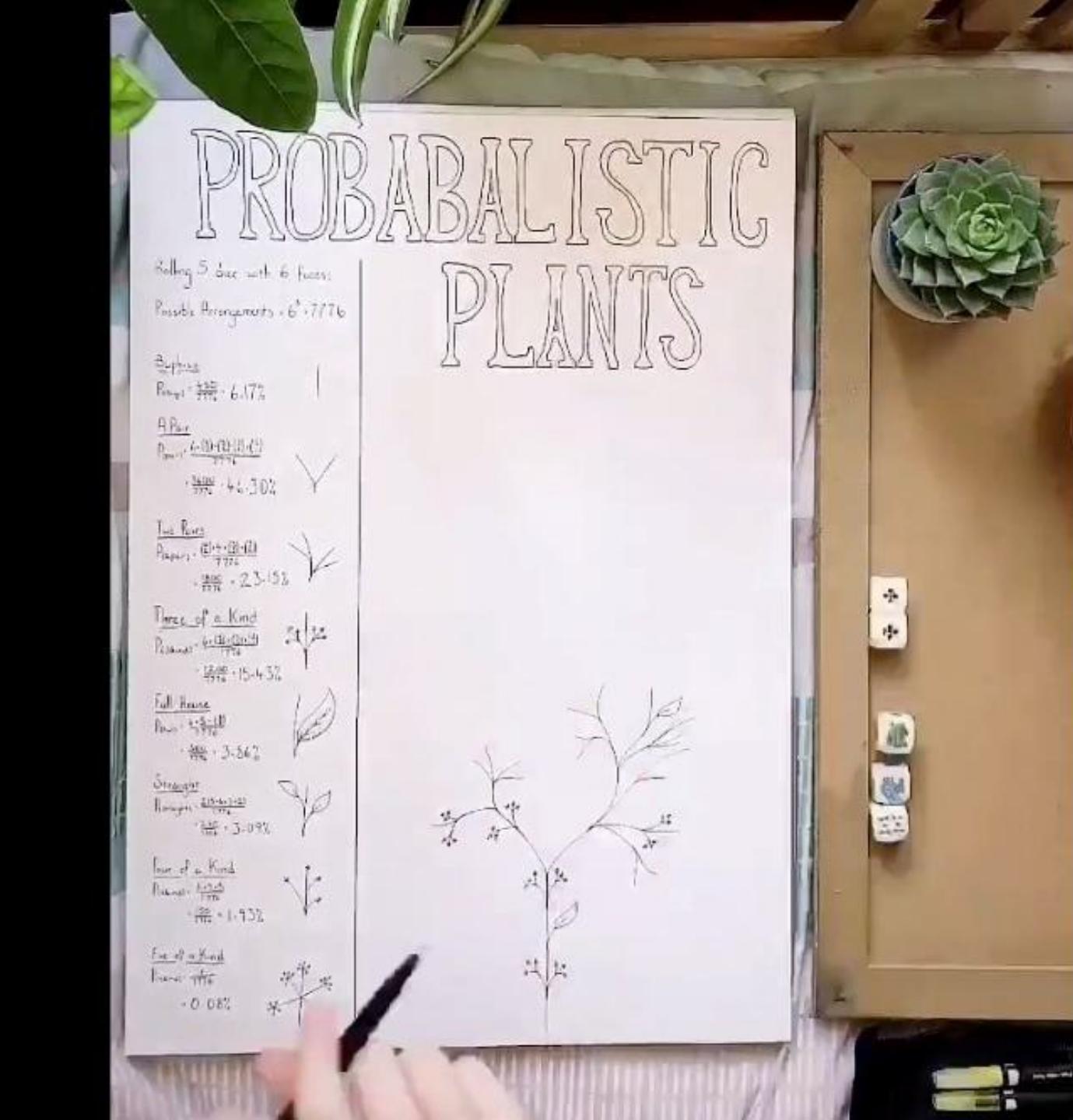
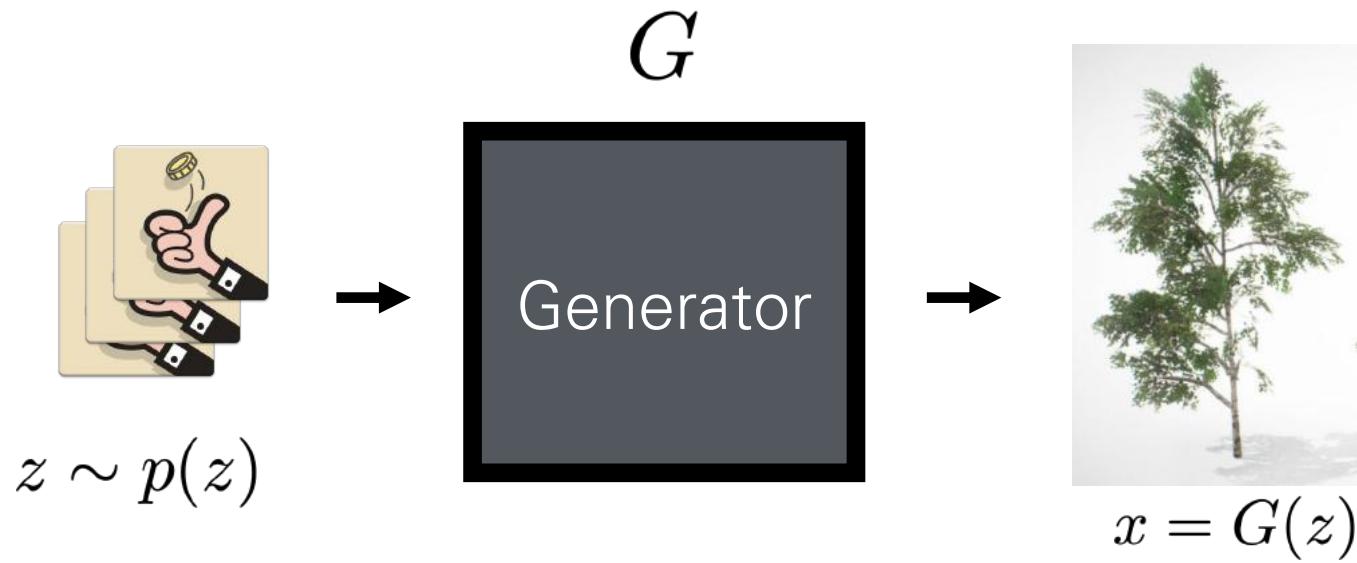


Image synthesis from “noise”



Sampler

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

$$z \sim p(z)$$

$$x = G(z)$$

$\mathbf{z} \sim \text{Bernoulli}(0.5)$

for $i = 1, \dots, N$ **do**

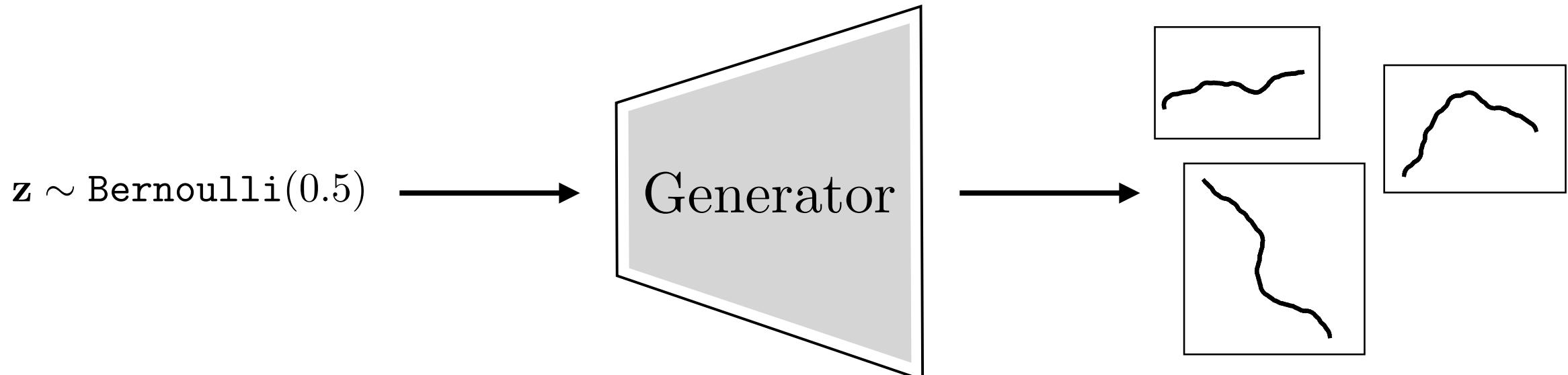
 extend line 1 unit in current heading direction

if $z_i == 1$ **then**

 rotate heading 10° to the right

else

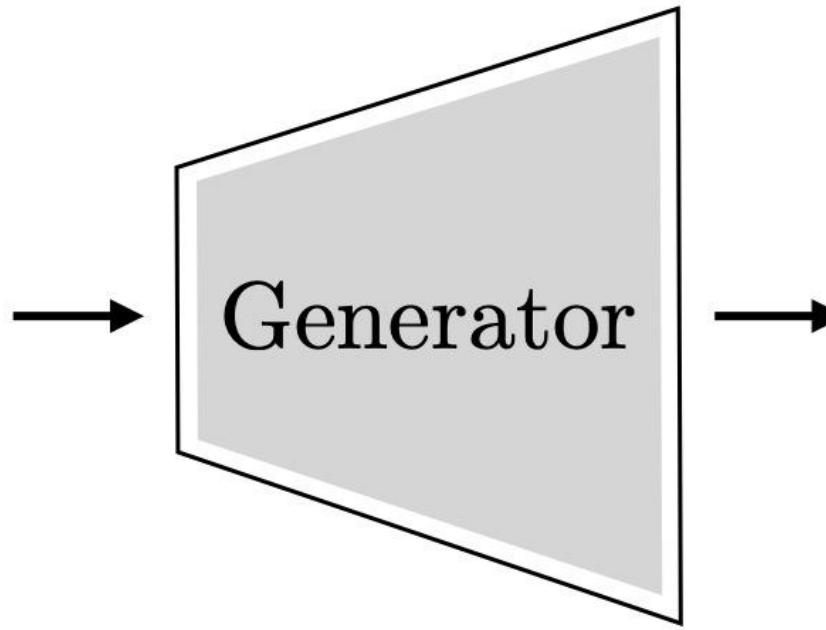
 rotate heading 10° to the left



“noise” “latent variables”

\downarrow \downarrow

$\mathbf{z}_1 \sim \text{Bernoulli}(0.5)$ (River turns)
 $\mathbf{z}_2 \sim \text{Normal}(\mu_1, \Sigma_1)$ (Grass color)
 $\mathbf{z}_3 \sim \text{Unif}(0, 10)$ (Number trees)
⋮



Concept #1: **noise is latent variables**

What's the goal of generative modeling?

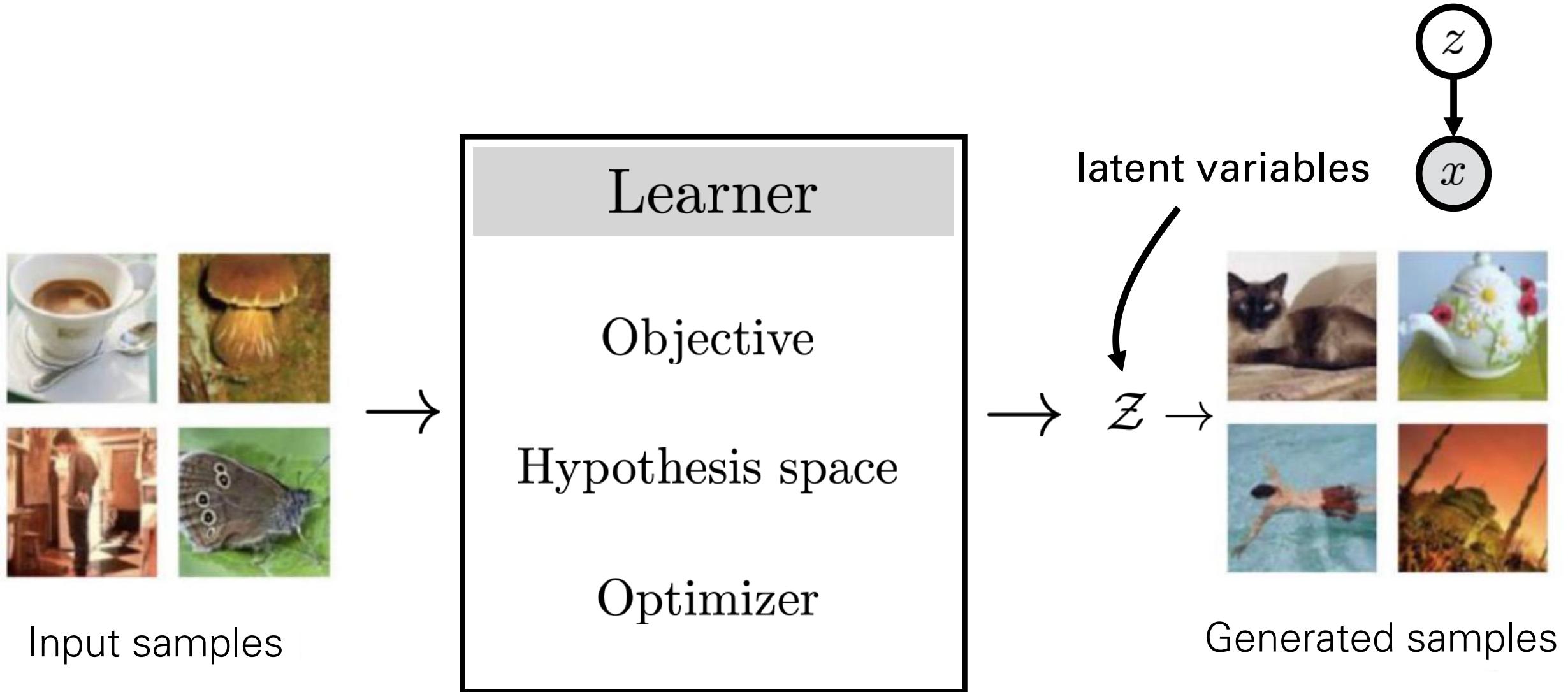
- Make synthetic data that “looks like” real data.
- How to measure “looks like”?
- The main answer in deep generative models is: “has high probability under a density model fit to real data.”

What's the goal of generative modeling?

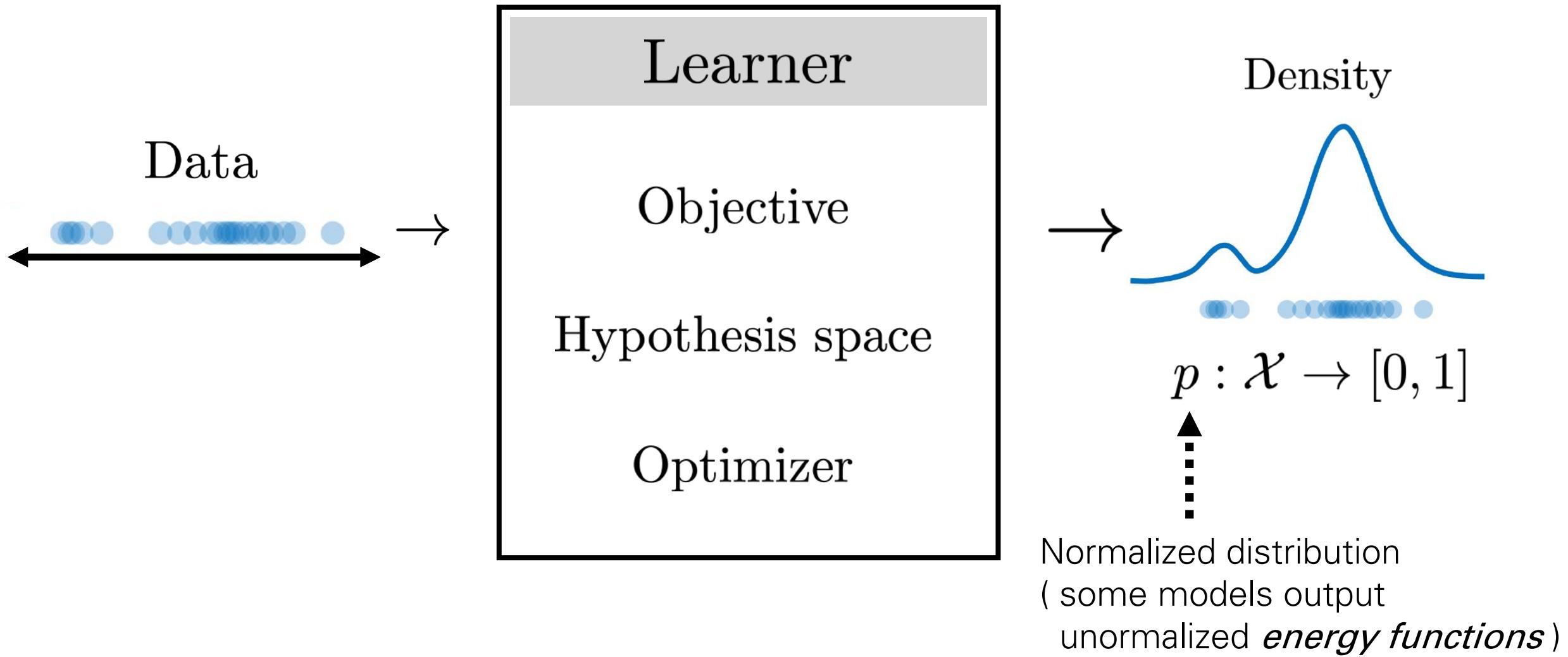
The goal is not to replicate the training data but to make **new** data that is **realistic** (captures the essential properties of real data)

(A model that memorizes the training data is overfit in exactly the same sense as a classifier can be overfit)

Learning a generative model



Learning a density model



Case study #1: Fitting a Gaussian to data

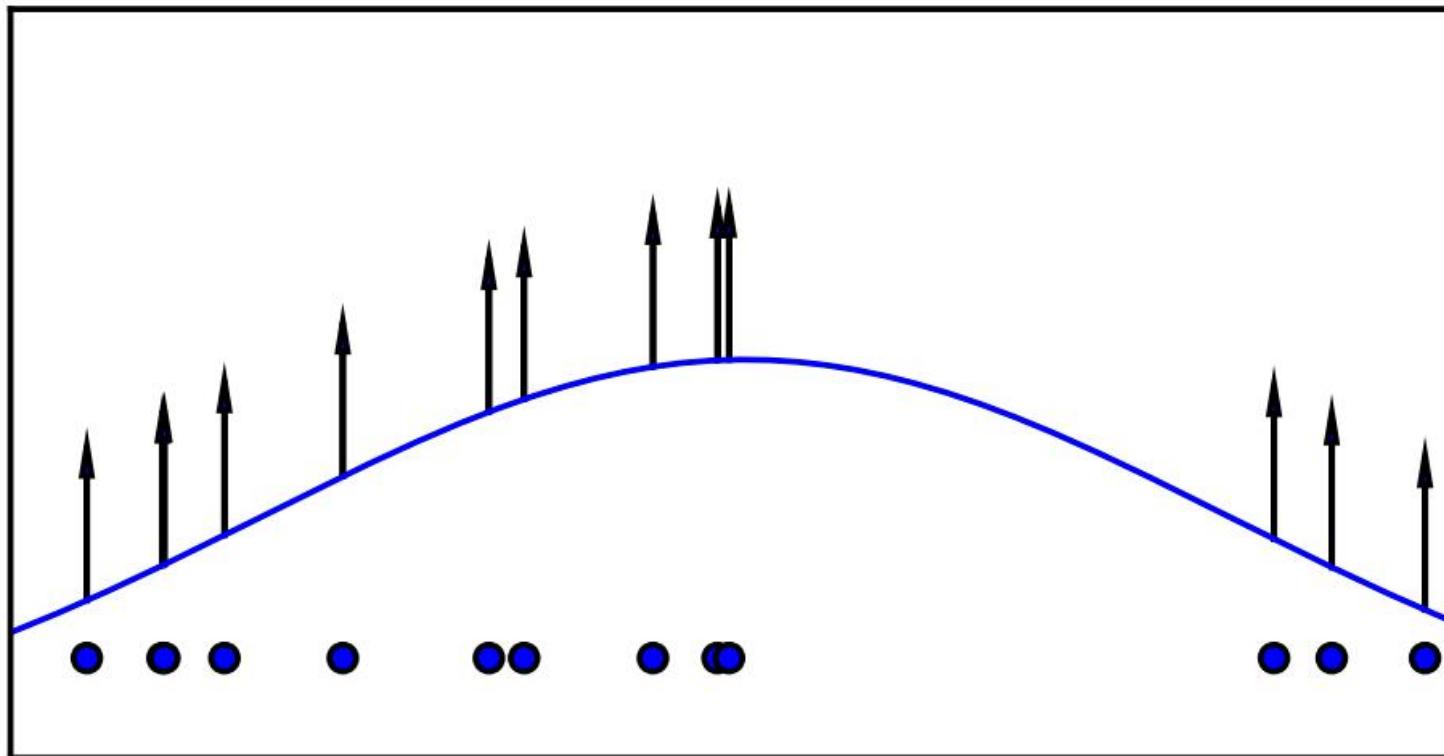


fig from [Goodfellow, 2016]

Max likelihood objective

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

Considering only Gaussian fits

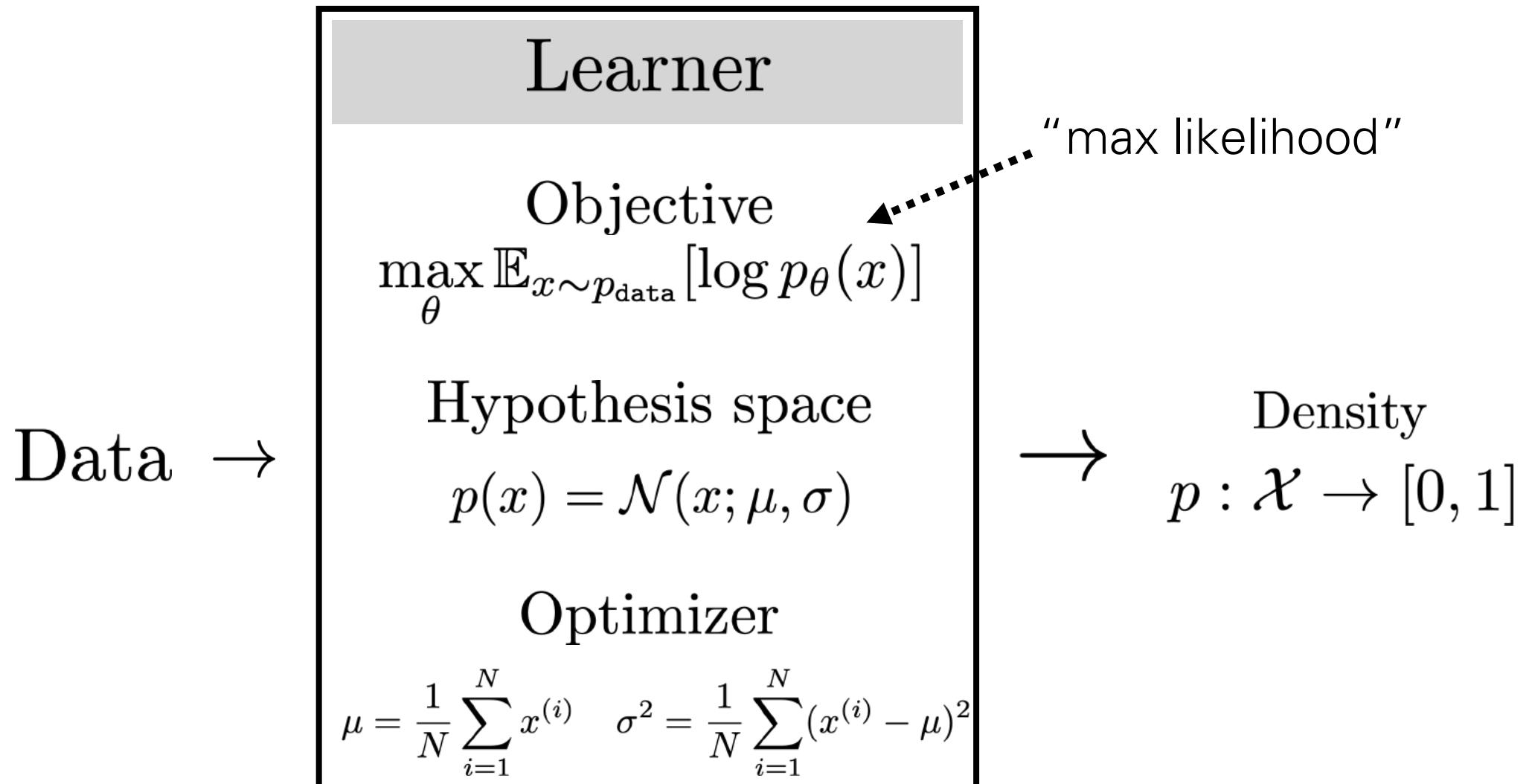
$$p_{\theta}(x) = \mathcal{N}(x; \mu, \sigma)$$

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

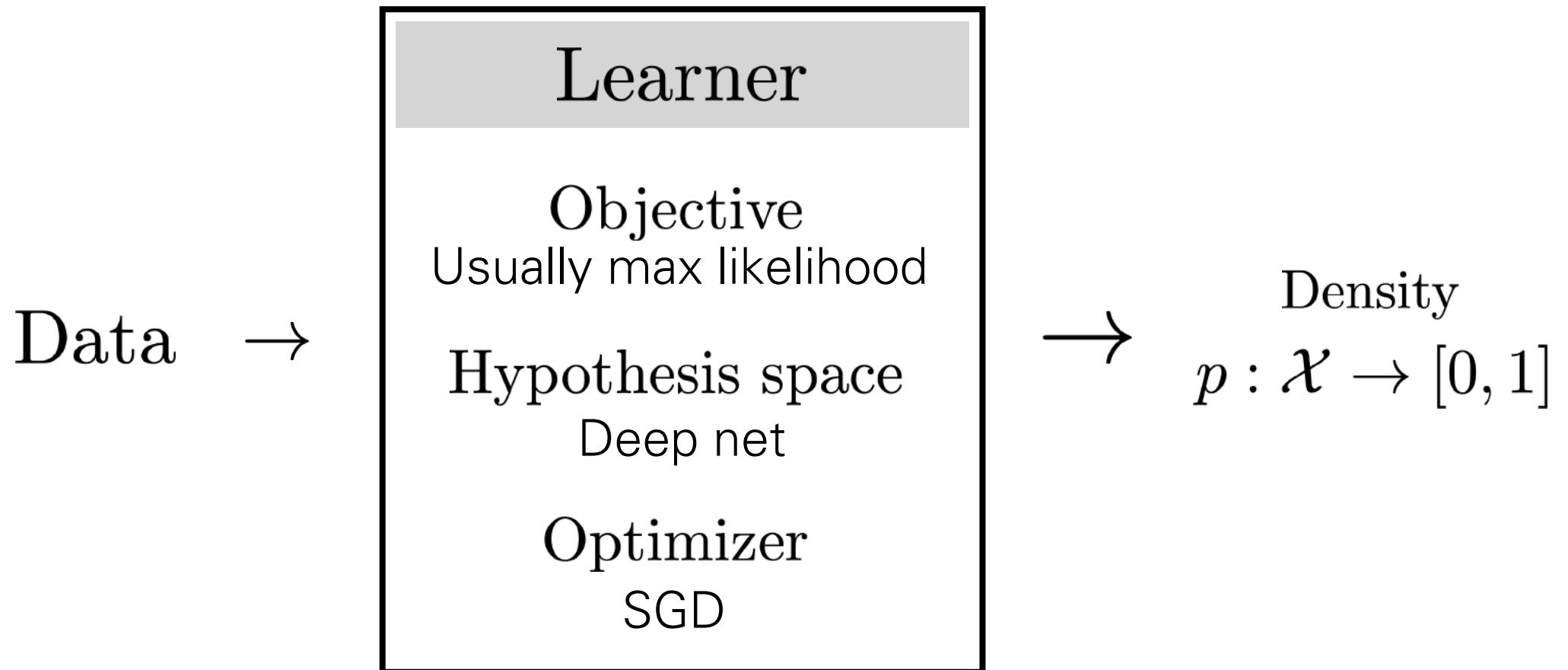
Closed form optimum:

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

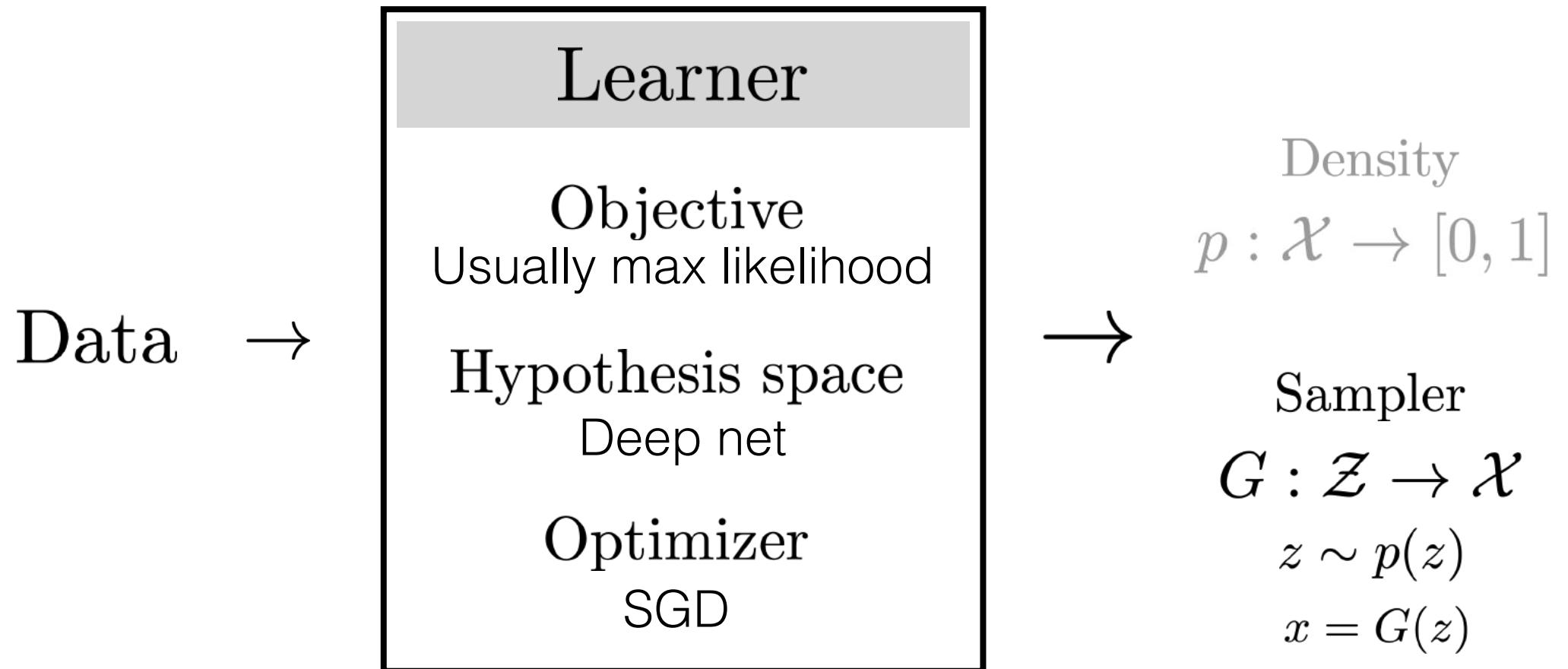
Case study #1: Fitting a Gaussian to data



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**

Learning data generator

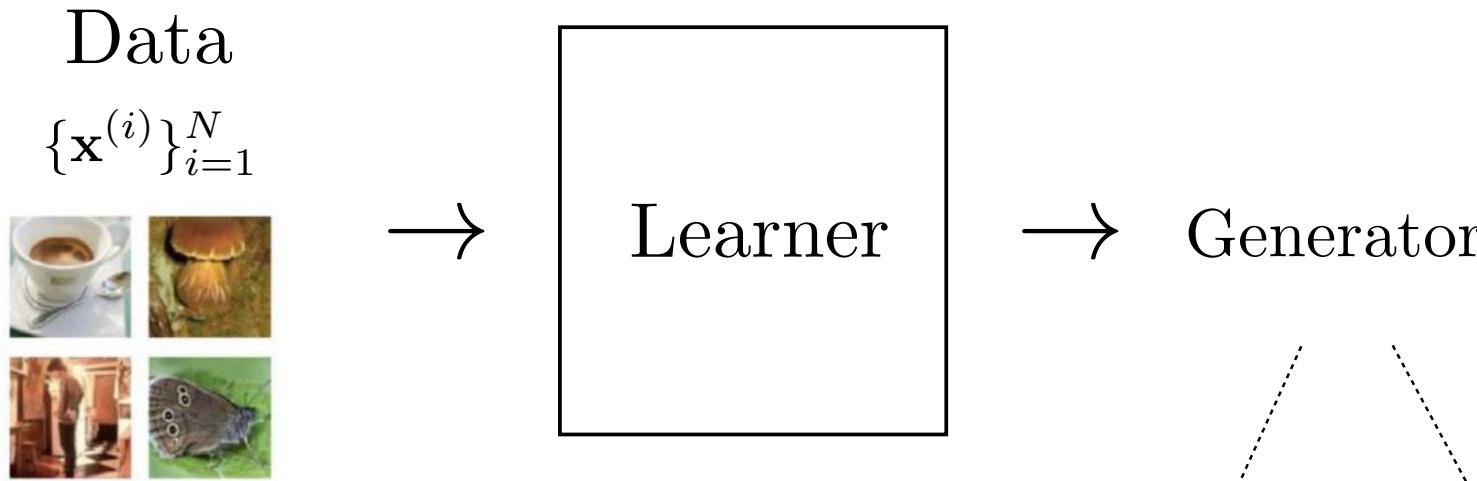
Two approaches:

confusingly, sometimes called an “implicit generative model”

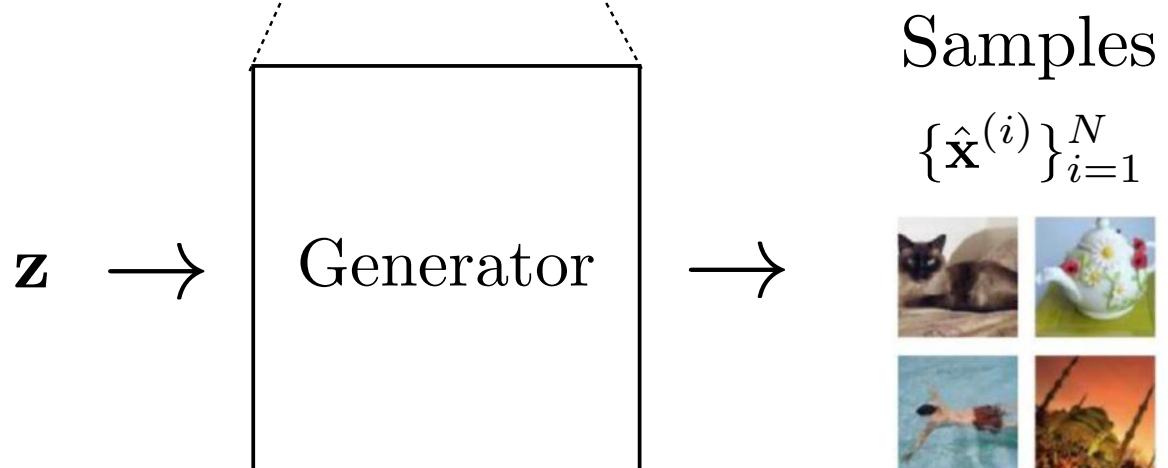
- 
- 1. Direct approach:** learn a function that generates data directly $G : \mathcal{Z} \rightarrow \mathcal{X}$
 - 2. Indirect approach:** learn a function that scores data; generate data by finding points that score highly under this function $E : \mathcal{X} \rightarrow \mathbb{R}$

Direct approach

Training

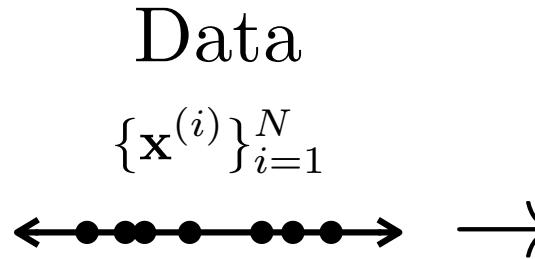


Sampling



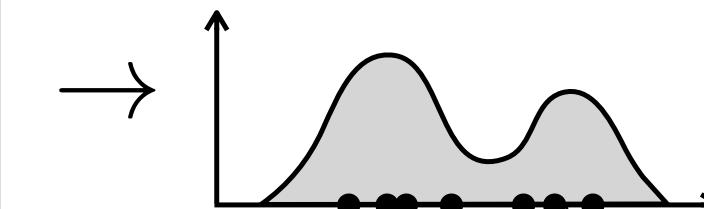
Indirect approach

Training



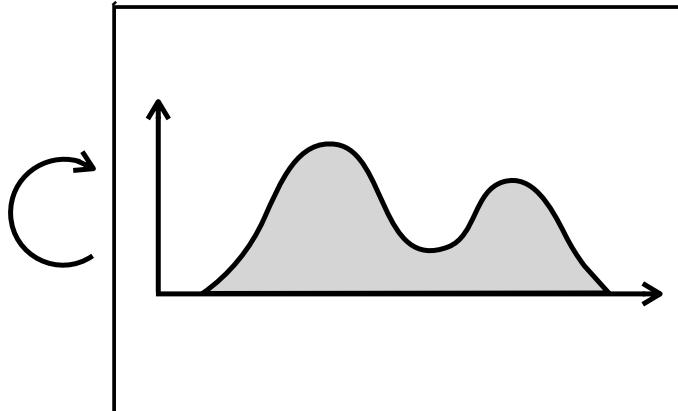
Scoring function

e.g., likelihood, energy,
"score function"



Sampling

Sampling algorithm
(e.g., MCMC)

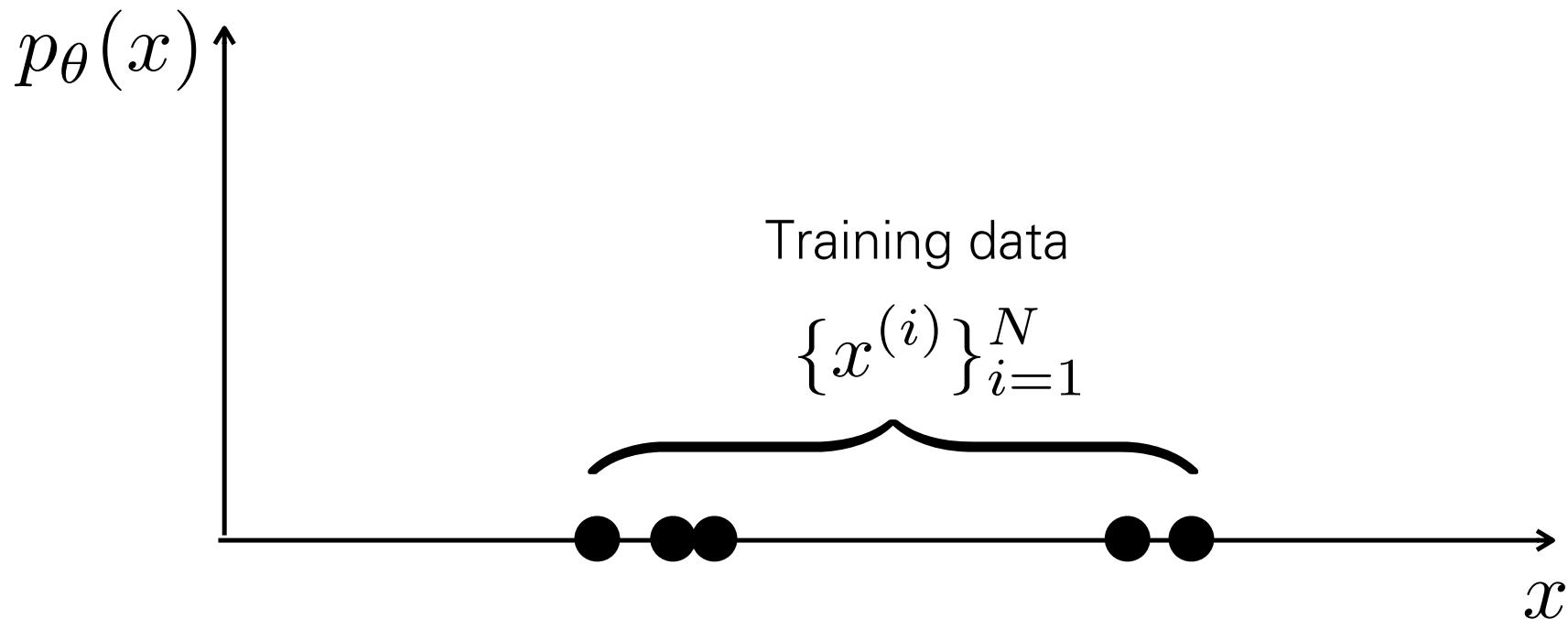


Samples
 $\{\hat{\mathbf{x}}^{(i)}\}_{i=1}^N$



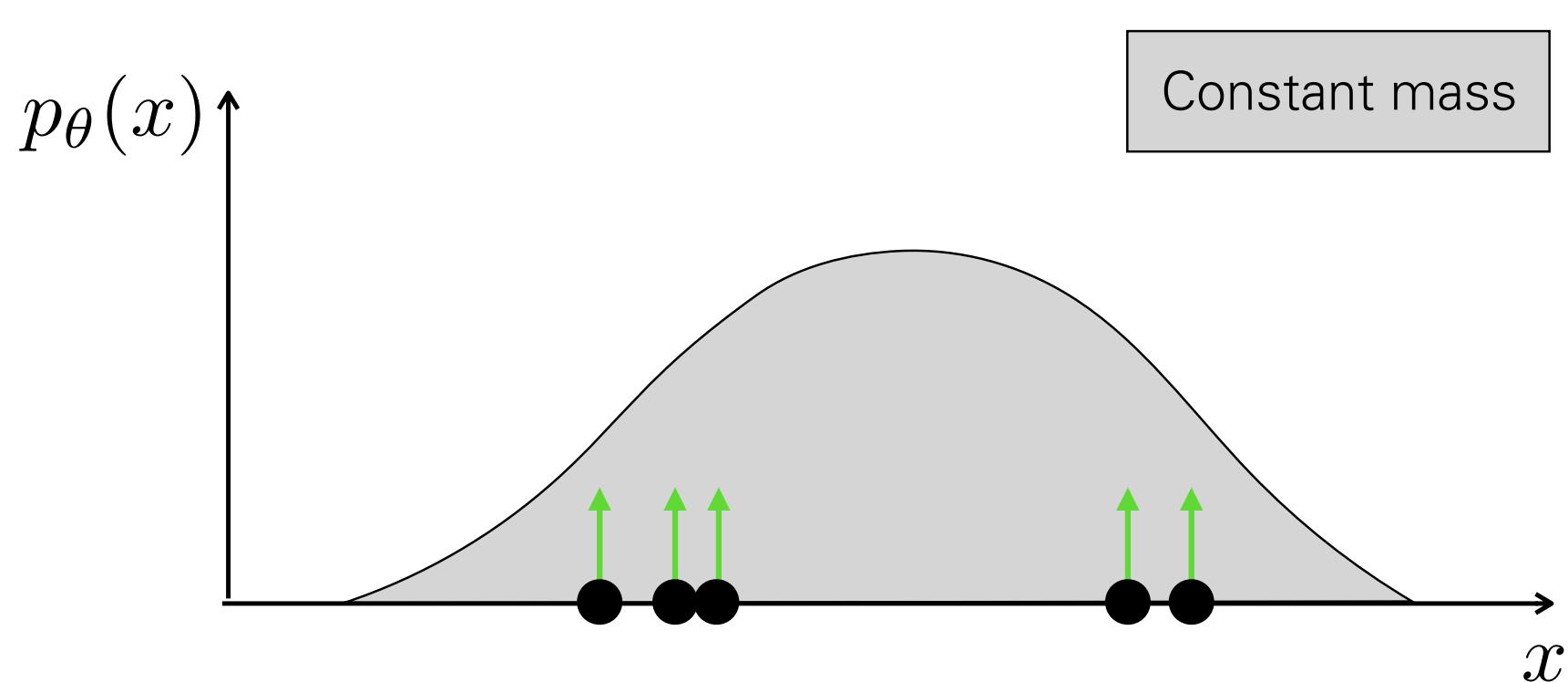
Density models

$$p_\theta : \mathcal{X} \rightarrow [0, \infty)$$



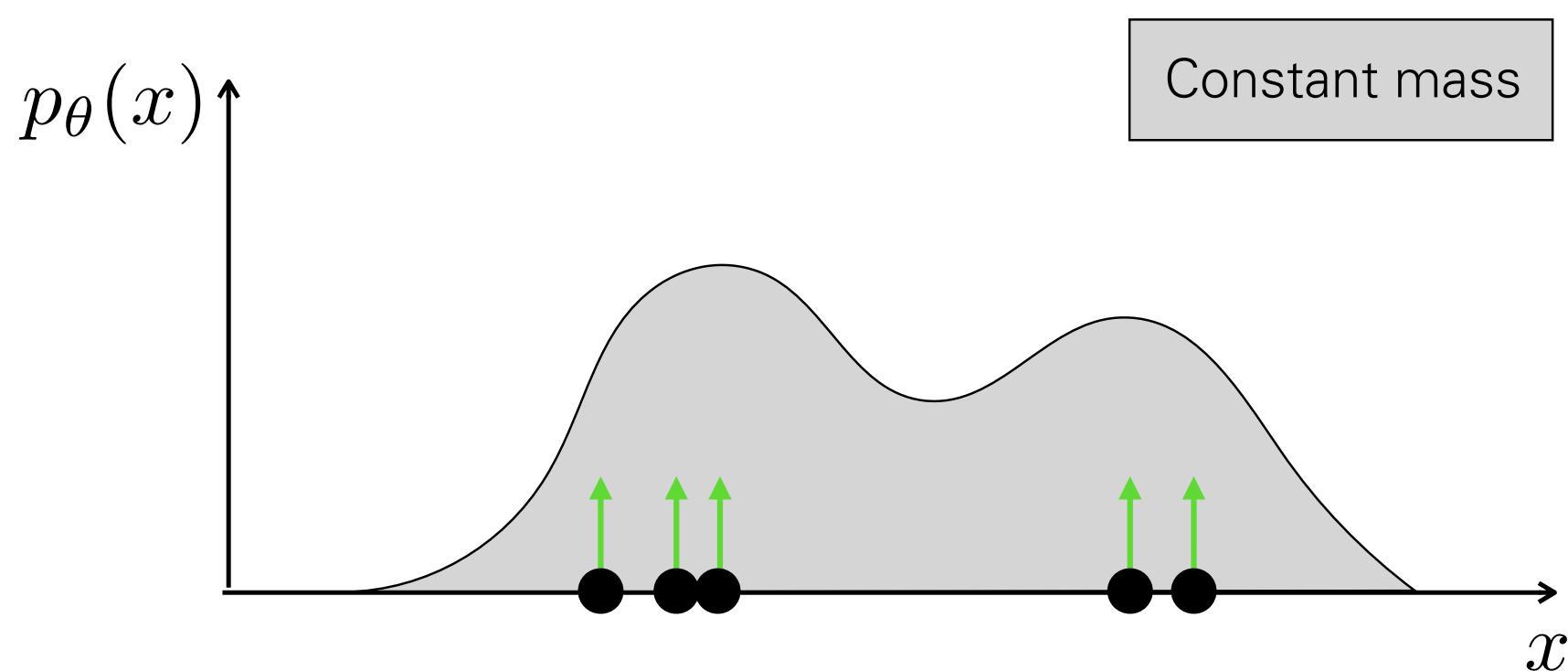
Density models

$$p_\theta : \mathcal{X} \rightarrow [0, \infty)$$

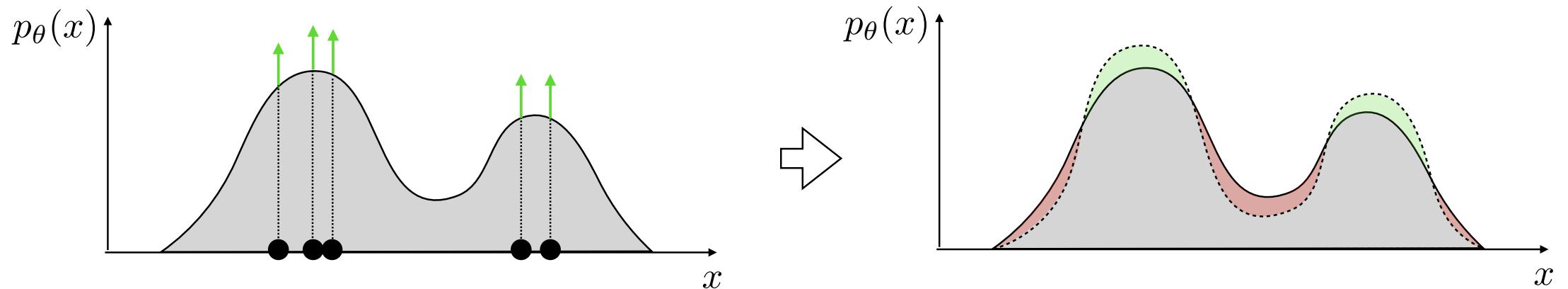


Density models

$$p_\theta : \mathcal{X} \rightarrow [0, \infty)$$



Density models



$$\begin{aligned} p_\theta^* &= \arg \min_{p_\theta} \text{KL}(p_{\text{data}}, p_\theta) \\ &= \arg \min_{p_\theta} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log \frac{p_\theta(\mathbf{x})}{p_{\text{data}}(\mathbf{x})}] \\ &= \arg \min_{p_\theta} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log p_\theta(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log p_{\text{data}}(\mathbf{x})] \\ &= \arg \min_{p_\theta} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log p_\theta(\mathbf{x})] \quad \triangleleft \text{dropped 2nd term since no dependence on } p_\theta \\ &\approx \arg \min_{p_\theta} \frac{1}{N} \sum_{i=1}^N \log p_\theta(\mathbf{x}^{(i)}) \end{aligned}$$

max likelihood

Energy-based models

→ i.e. unnormalized
probability models

$$\int_{\mathbf{x}} p_{\theta}(\mathbf{x}) d\mathbf{x} = 1$$

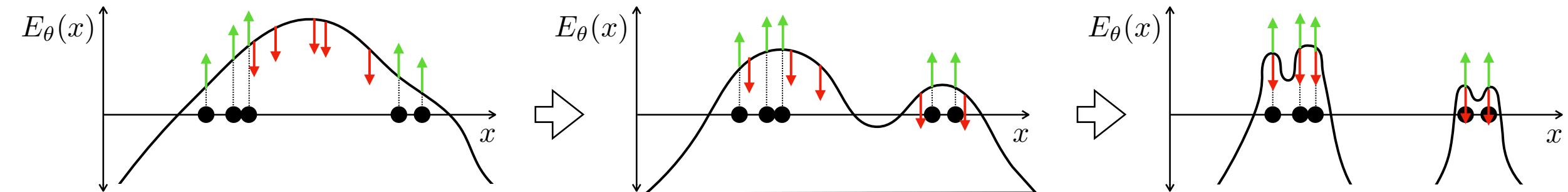
$$p_{\theta} = \frac{e^{-E_{\theta}}}{Z(\theta)} \quad Z(\theta) = \int_{\mathbf{x}} e^{-E_{\theta}(\mathbf{x})} d\mathbf{x}$$

$$\frac{p_{\theta}(\mathbf{x}_1)}{p_{\theta}(\mathbf{x}_2)} = \frac{e^{-E_{\theta}(\mathbf{x}_1)}/Z(\theta)}{e^{-E_{\theta}(\mathbf{x}_2)}/Z(\theta)} = \frac{e^{-E_{\theta}(\mathbf{x}_1)}}{e^{-E_{\theta}(\mathbf{x}_2)}}$$

← Relative
probabilities are often
all you need (e.g., for
sampling)

Energy-based models

At convergence, green (data) and red (model) samples are identical and model update (green-red) cancels out



Contrastive divergence

Energy-based models (learning model parameters)

$$\begin{aligned}\nabla_{\theta} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log p_{\theta}(\mathbf{x})] &= \nabla_{\theta} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log \frac{e^{-E_{\theta}(\mathbf{x})}}{Z(\theta)}] \\ &= -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\nabla_{\theta} E_{\theta}(\mathbf{x})] - \underbrace{\nabla_{\theta} \log Z(\theta)}_{\text{ }}\end{aligned}$$

How to measure this?

Energy-based models (learning model parameters)

$$\begin{aligned} -\nabla_{\theta} \log Z(\theta) &= \frac{1}{Z(\theta)} \nabla_{\theta} Z(\theta) && \triangleleft \quad \nabla_x \log f(x) = \frac{1}{f(x)} \nabla_x f(x) \\ &= \frac{1}{Z(\theta)} \nabla_{\theta} \int_x e^{-E_{\theta}(\mathbf{x})} d\mathbf{x} && \triangleleft \quad \text{definition of } Z \\ &= \frac{1}{Z(\theta)} \int_x \nabla_{\theta} e^{-E_{\theta}(\mathbf{x})} d\mathbf{x} && \triangleleft \quad \text{exchange sum and grad} \\ &= \frac{1}{Z(\theta)} - \int_x e^{-E_{\theta}(\mathbf{x})} \nabla_{\theta} E_{\theta}(\mathbf{x}) d\mathbf{x} \\ &= - \int_x \frac{e^{-E_{\theta}(\mathbf{x})}}{Z(\theta)} \nabla_{\theta} E_{\theta}(\mathbf{x}) d\mathbf{x} \\ &= - \int_x p_{\theta}(\mathbf{x}) \nabla_{\theta} E_{\theta}(\mathbf{x}) d\mathbf{x} && \triangleleft \quad \text{definition of } p_{\theta} \\ &= -\mathbb{E}_{\mathbf{x} \sim p_{\theta}} [\nabla_{\theta} E_{\theta}(\mathbf{x})] && \triangleleft \quad \text{definition of expectation} \end{aligned}$$

Energy-based models (learning model parameters)

$$\nabla_{\theta} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log p_{\theta}(\mathbf{x})] = \nabla_{\theta} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log \frac{e^{-E_{\theta}(\mathbf{x})}}{Z(\theta)}]$$

$$= -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\nabla_{\theta} E_{\theta}(\mathbf{x})] - \boxed{\nabla_{\theta} \log Z(\theta)}$$

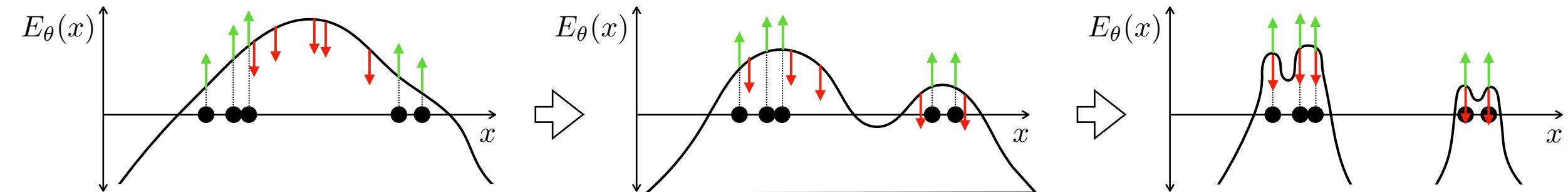
$$= -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\nabla_{\theta} E_{\theta}(\mathbf{x})] + \boxed{\mathbb{E}_{\mathbf{x} \sim p_{\theta}} [\nabla_{\theta} E_{\theta}(\mathbf{x})]}$$

$$\approx -\frac{1}{N} \sum_{i=1}^N \nabla_{\theta} E_{\theta}(\mathbf{x}^{(i)}) + \boxed{\frac{1}{N} \sum_{i=1}^N \nabla_{\theta} E_{\theta}(\hat{\mathbf{x}}^{(i)})}$$

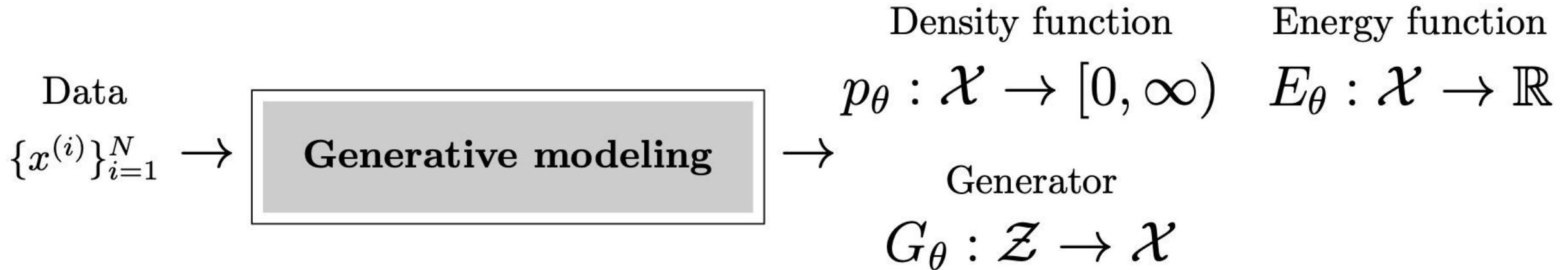
$$\mathbf{x}^{(i)} \sim p_{\text{data}} \quad \hat{\mathbf{x}}^{(i)} \sim p_{\theta}$$

Energy-based models

At convergence, green (data) and red (model) samples are identical and model update (green-red) cancels out

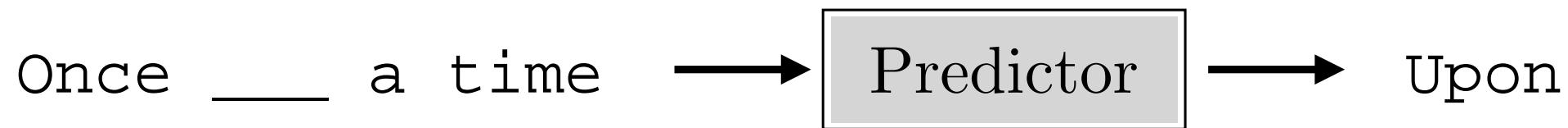
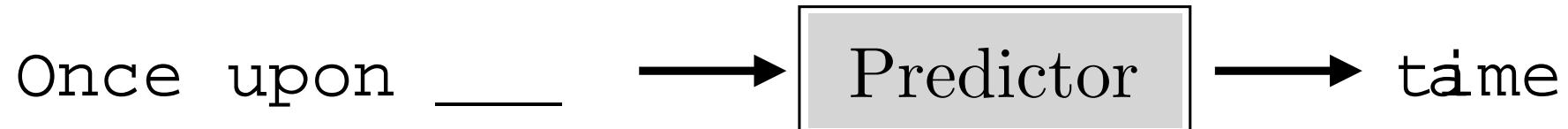


Contrastive divergence



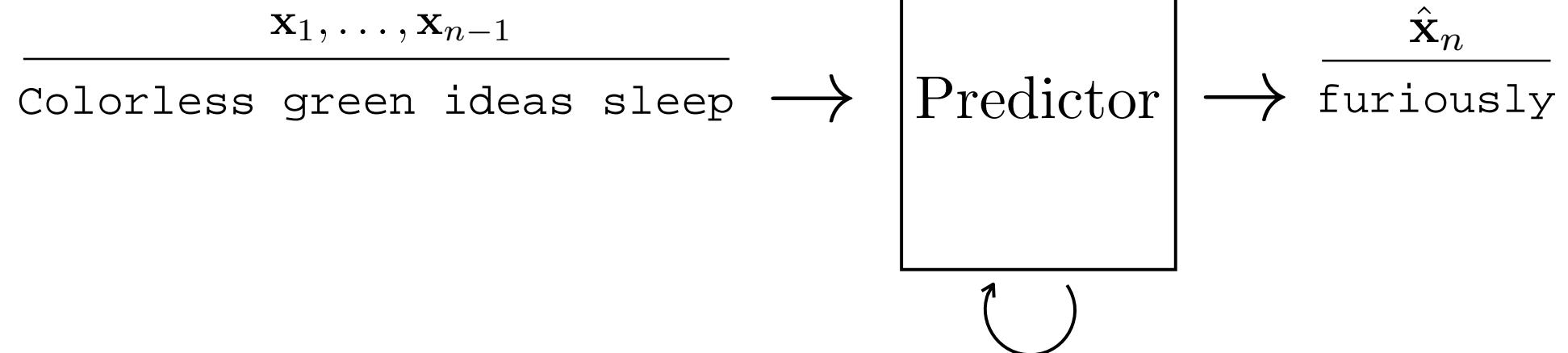
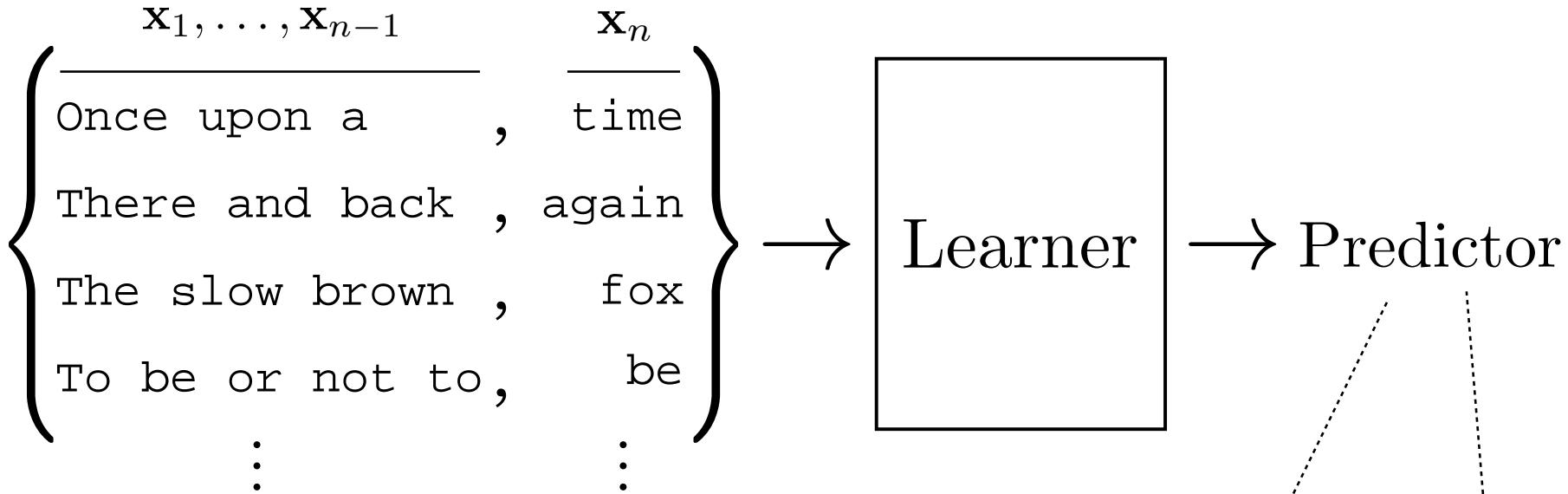
Concept #2: you can represent the data generating process directly or indirectly

Autoregressive models



Sampling

Training



Autoregressive probability models

$$p(\mathbf{X}) = p(\mathbf{x}_n | \mathbf{x}_1, \dots, \mathbf{x}_{n-1}) p(\mathbf{x}_{n-1} | \mathbf{x}_1, \dots, \mathbf{x}_{n-2}) \quad \dots \quad p(\mathbf{x}_2 | \mathbf{x}_1) p(\mathbf{x}_1)$$

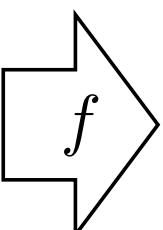
$$p(\mathbf{X}) = \prod_{i=1}^n p(\mathbf{x}_i | \mathbf{x}_1, \dots, \mathbf{x}_{i-1})$$

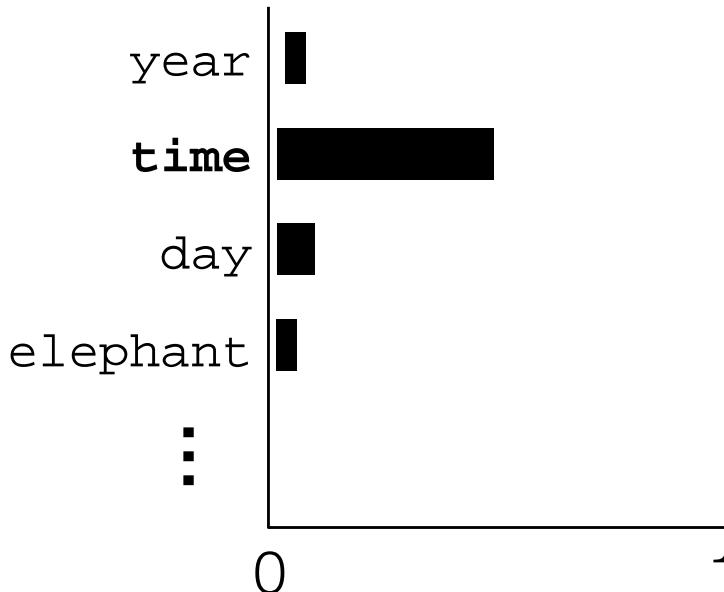
$p(\text{time} | \text{Once, upon, a})$
 $p(\text{a} | \text{Once, upon})$
 $p(\underbrace{\text{Once } \text{upon } \text{a}}_{p(\text{Once})} \text{ time})$
 $p(\text{upon} | \text{Once})$

Modeling a sequence of words

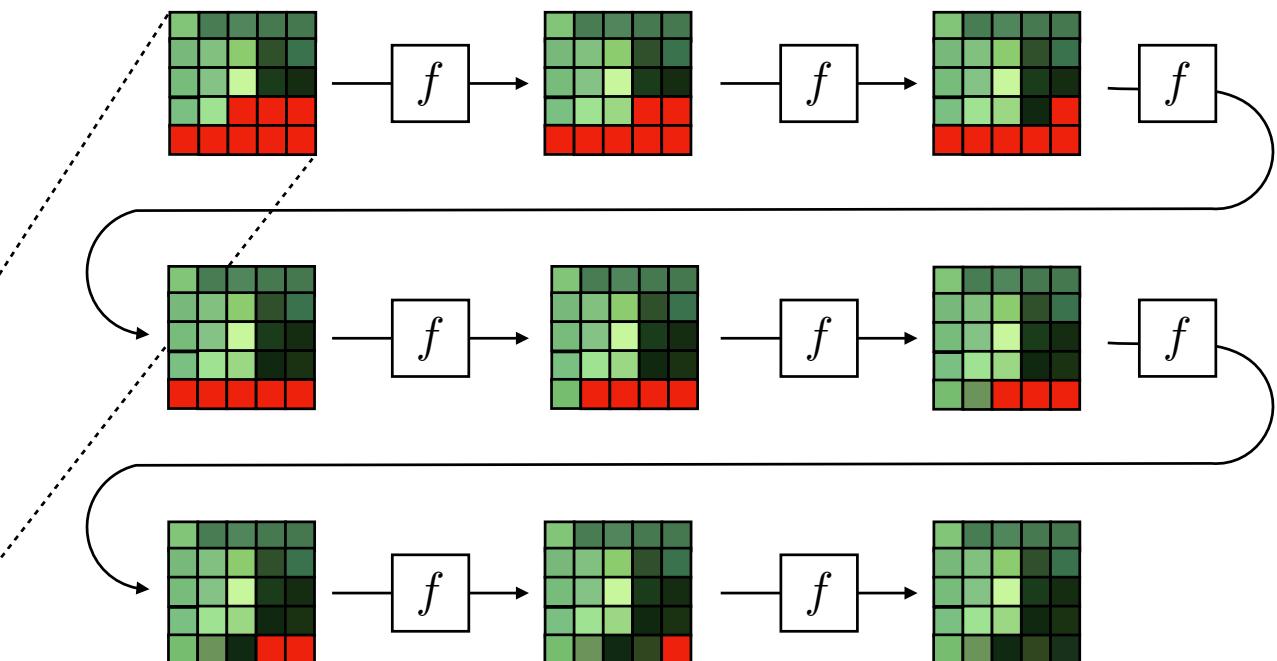
How to model $p(\text{time}|\text{Once, upon, a})$?

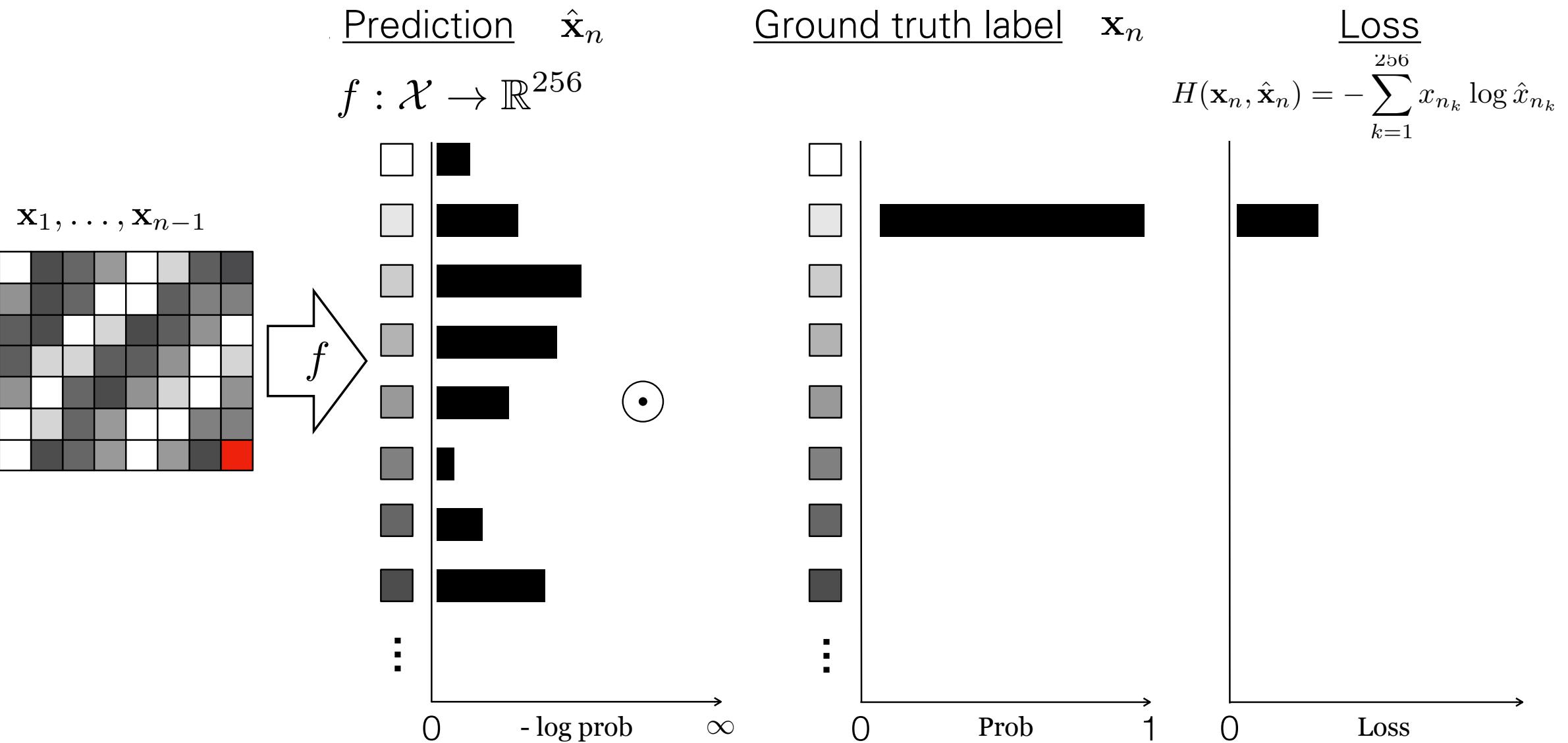
Just treat it as a next word classifier!

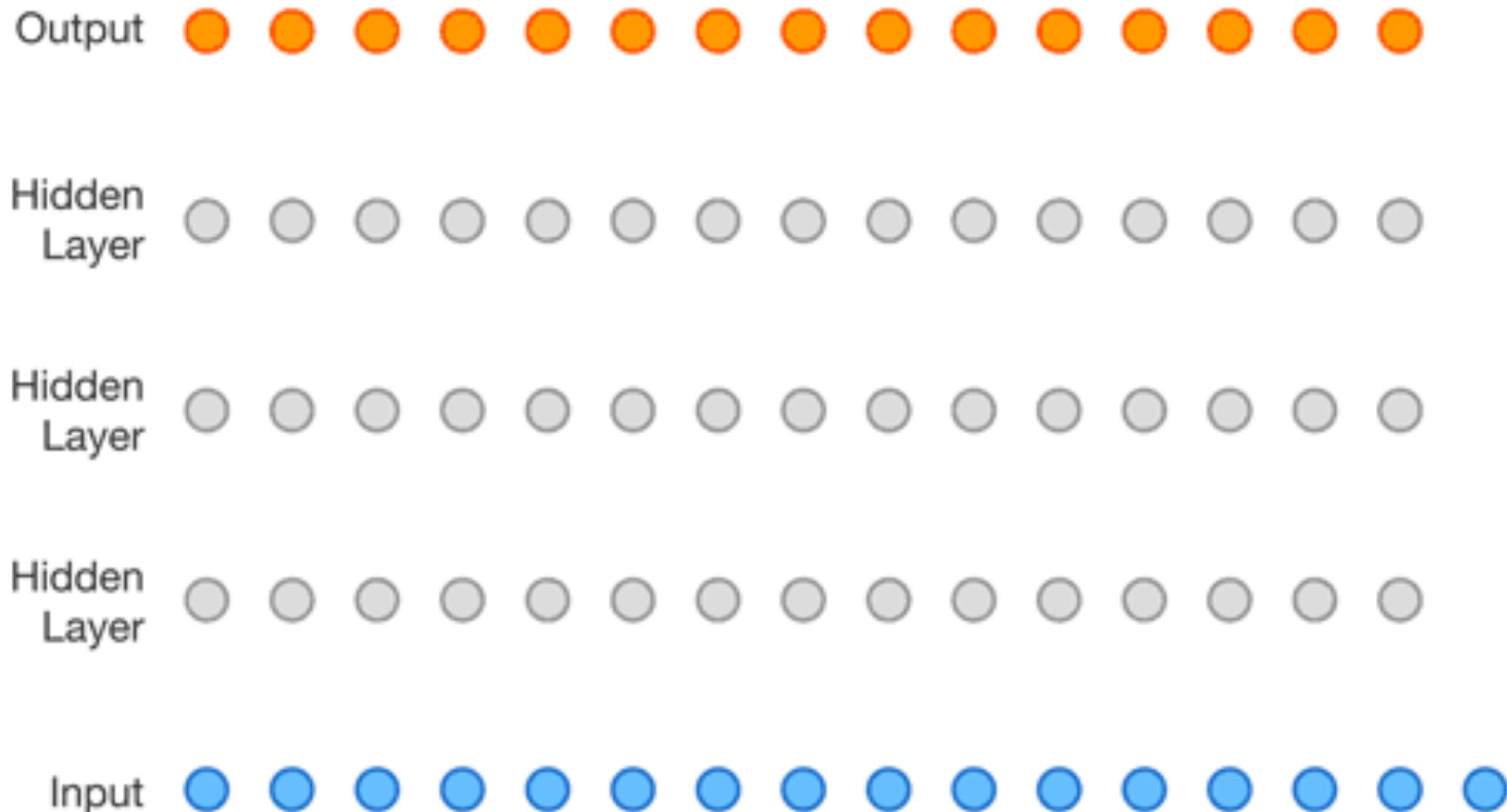
Once upon a 



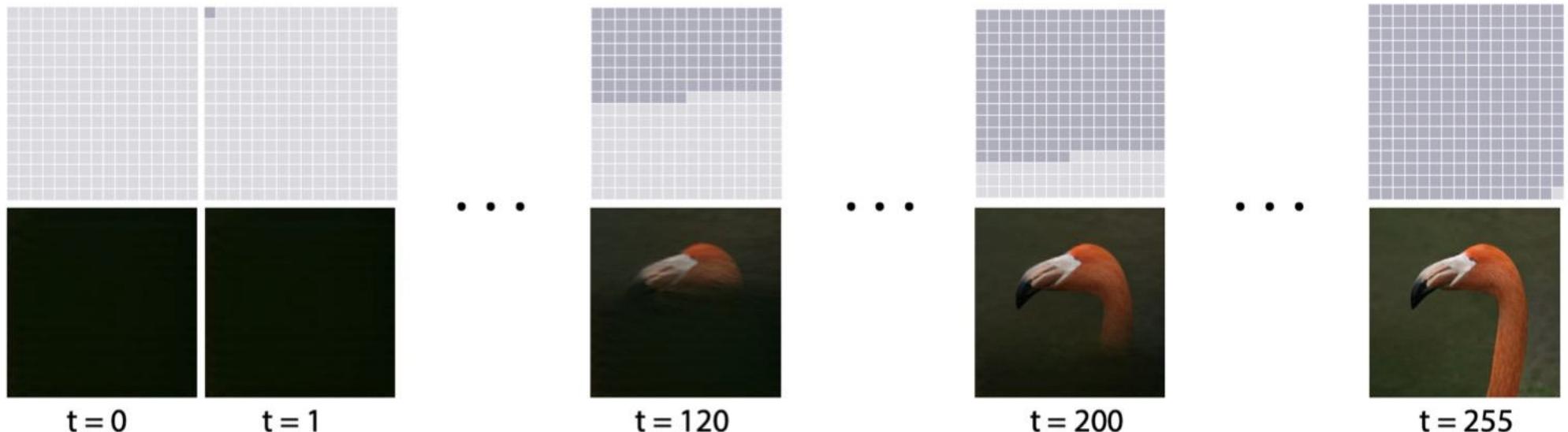
Autoregressive model of pixels



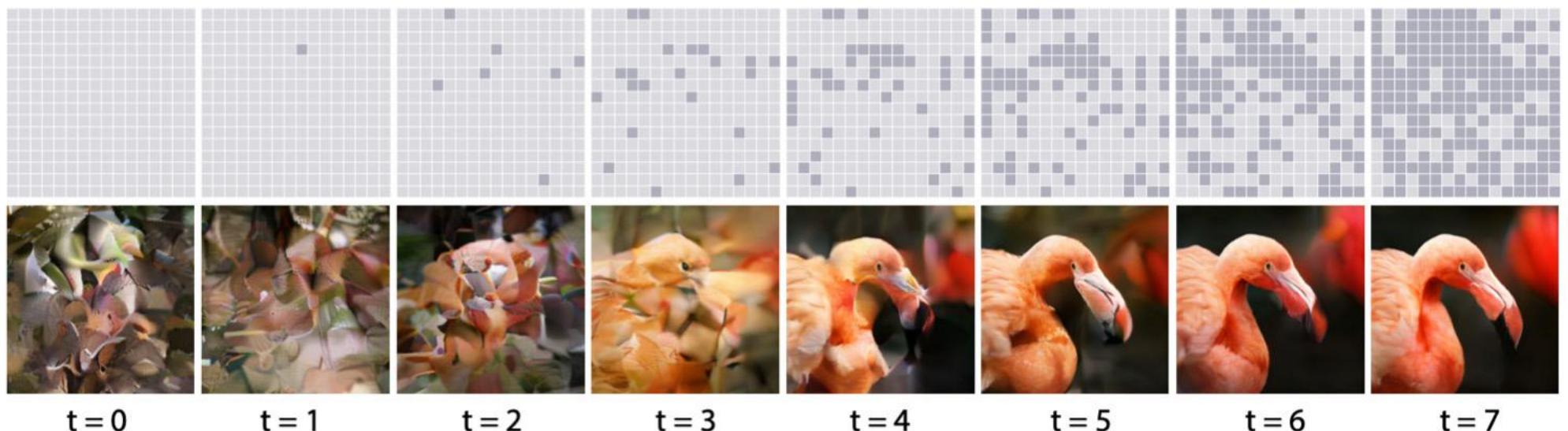




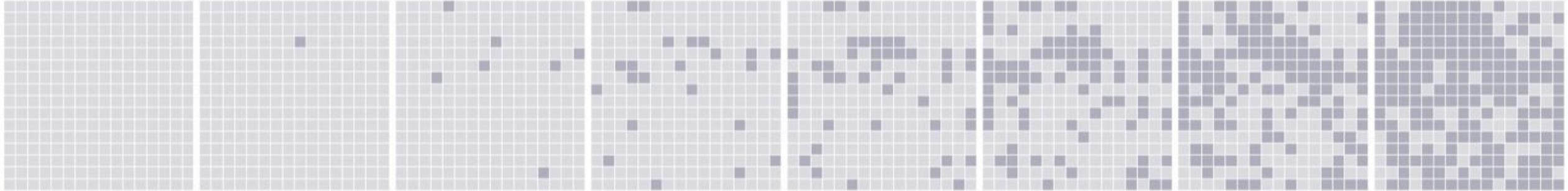
Sequential Decoding with Autoregressive Transformers



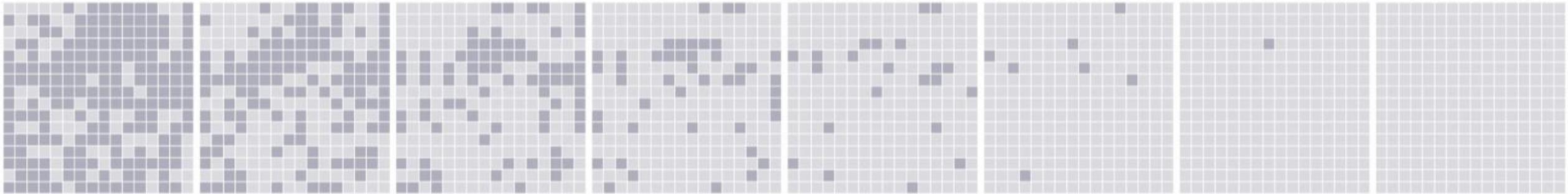
Scheduled Parallel Decoding with MaskGIT



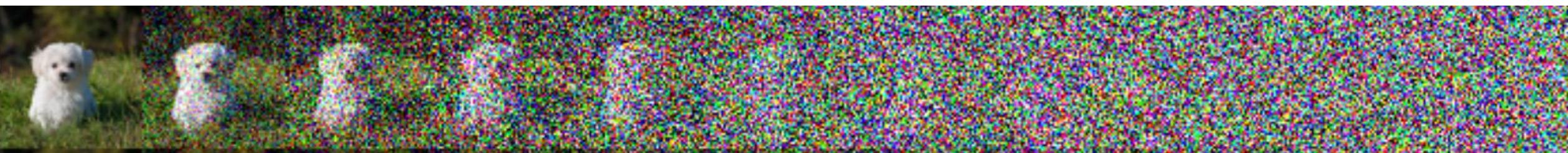
Diffusion models



Diffusion models



Corrupting the input →

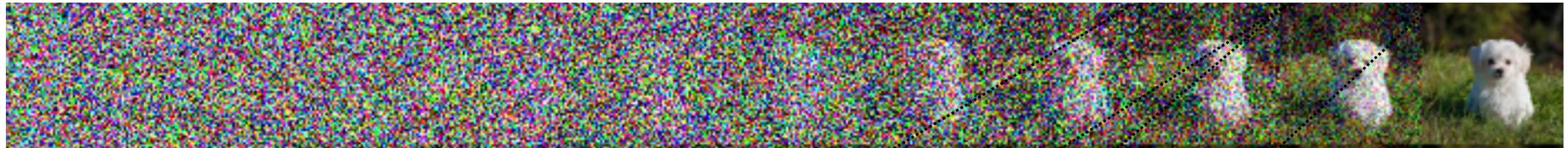


Corrupting the input →

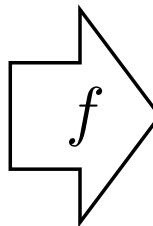
Diffusion models

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

\mathbf{x}



Denoising



\mathbf{x}_t

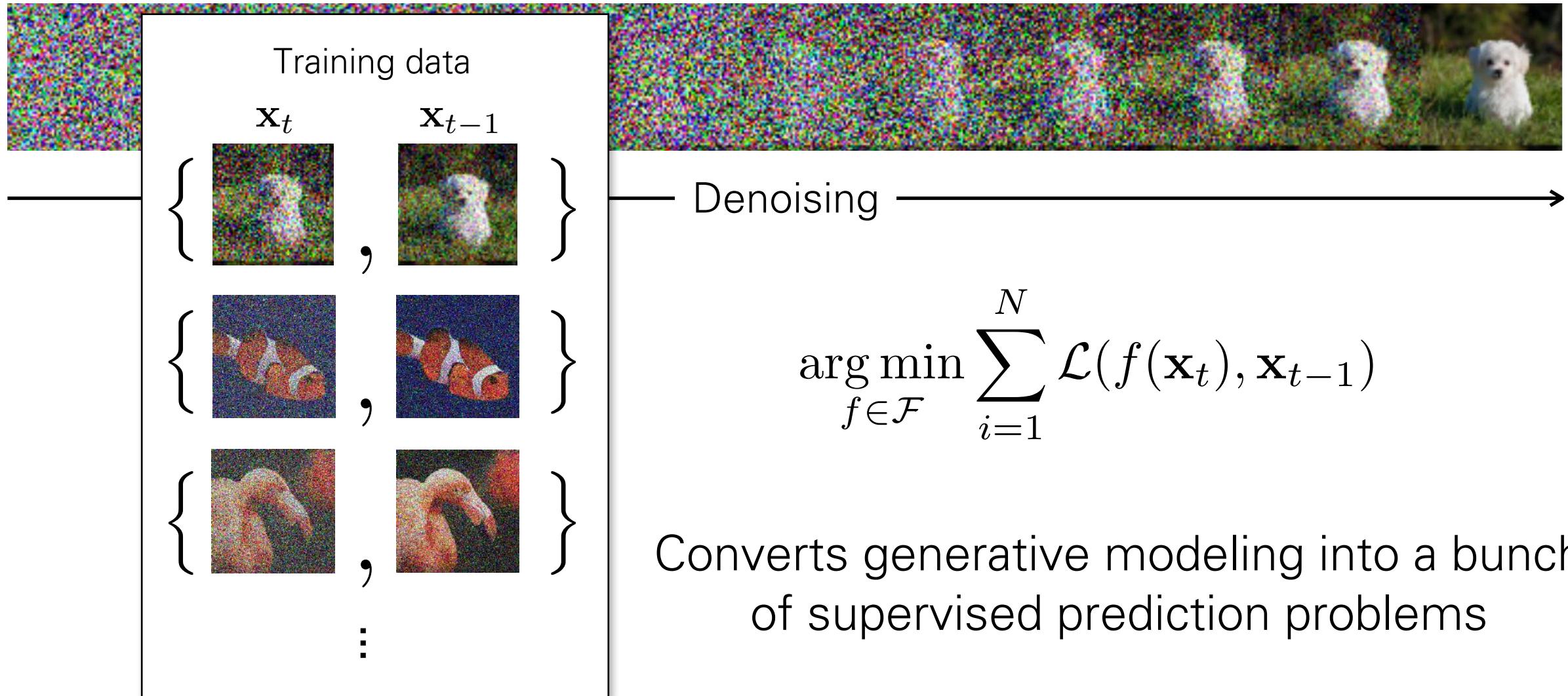


\mathbf{x}_{t-1}

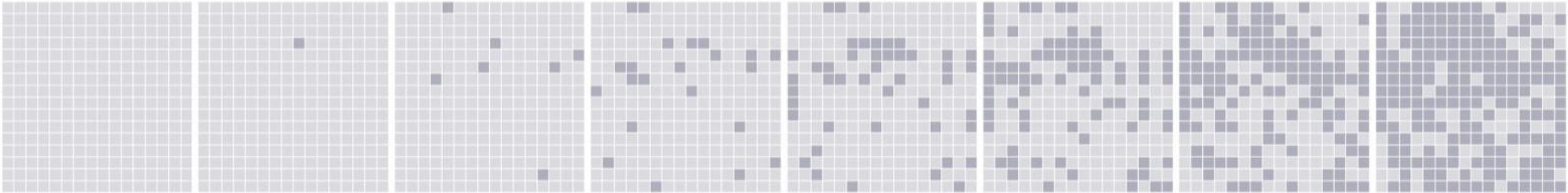
Diffusion models

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

\mathbf{x}



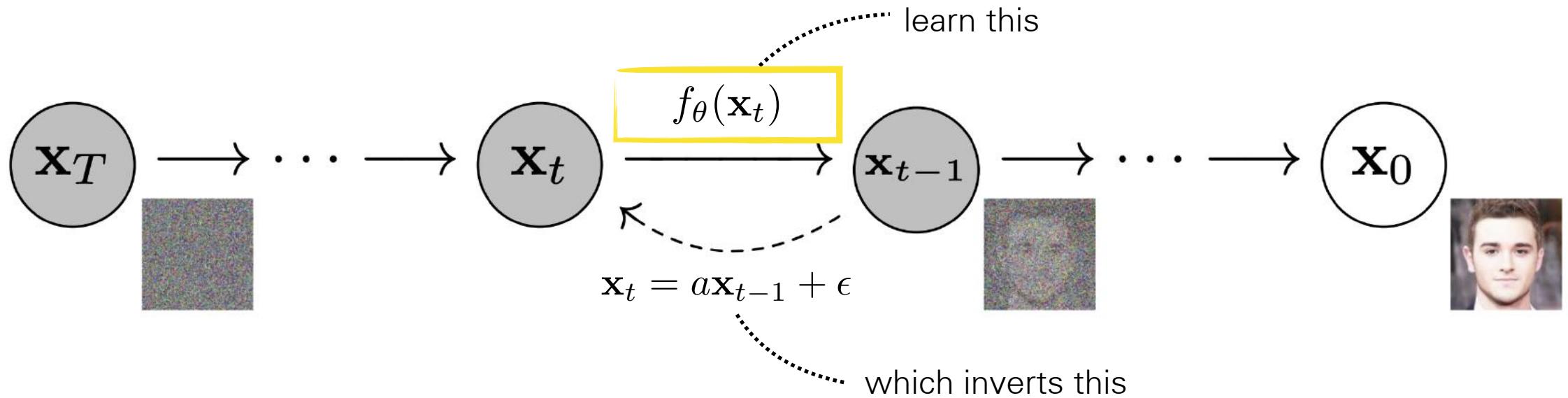
Diffusion models



Concept #3: **A common strategy is to turn generative modeling into a sequence of supervised learning problems**



Gaussian diffusion models



Forward process:

$$\epsilon \sim \mathcal{N}(0, b)$$

$$\mathbf{x}_t = a\mathbf{x}_{t-1} + \epsilon$$

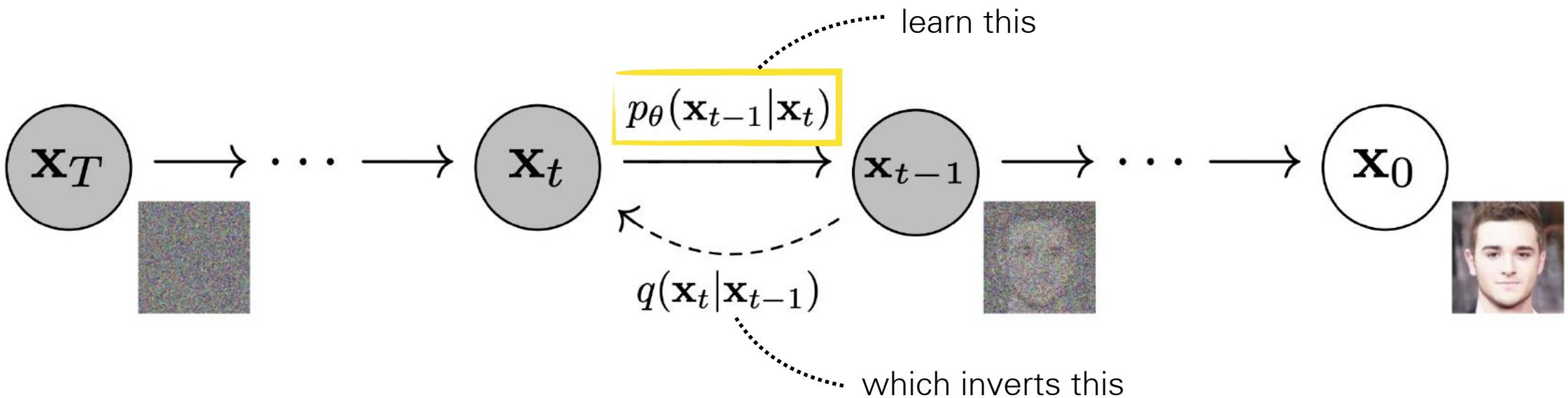
Reverse process:

$$\mu = f_\theta(\mathbf{x}_t)$$

$$\mathbf{x}_{t-1} \sim \mathcal{N}(\mu, c)$$

a, b, and c are hyperparameters
(see Ho, Jain, Abbeel for one
way to set them)

Gaussian diffusion models



Forward process:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(a\mathbf{x}_{t-1}, b)$$

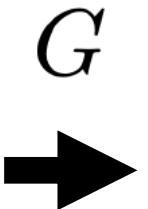
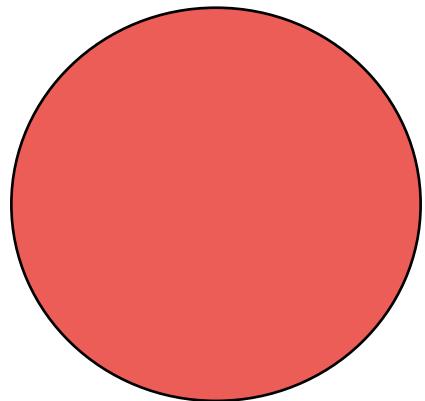
Reverse process:

$$p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(f_{\theta}(\mathbf{x}_t), c)$$

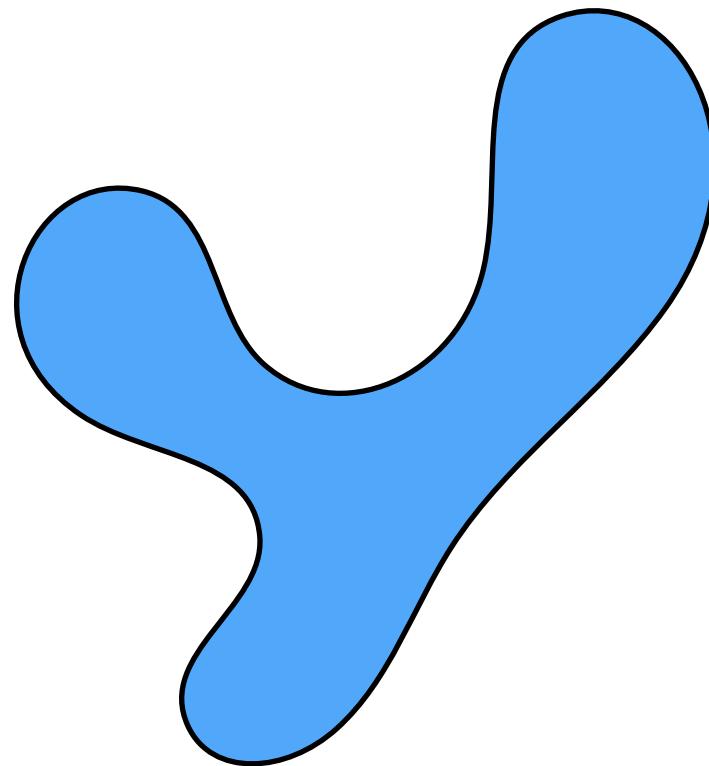
a, b, and c are hyperparameters
(see Ho, Jain, Abbeel for one
way to set them)

Deep generative models are distribution transformers

Prior distribution



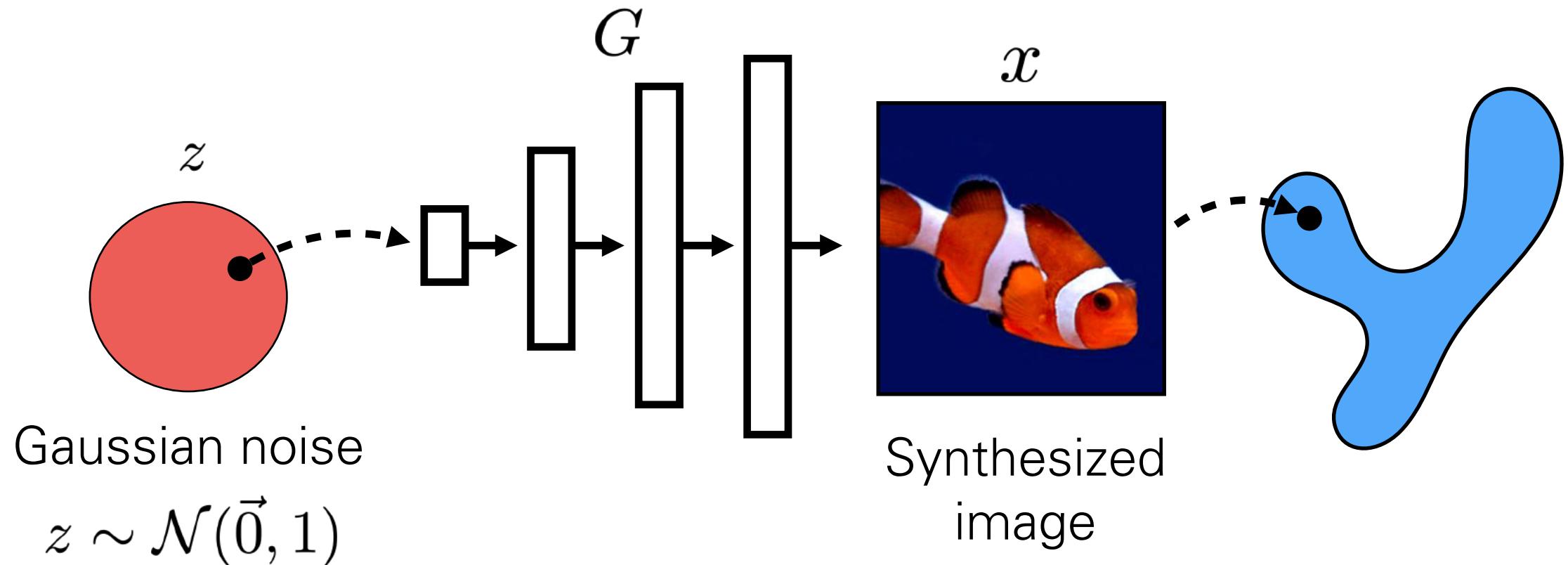
Target distribution



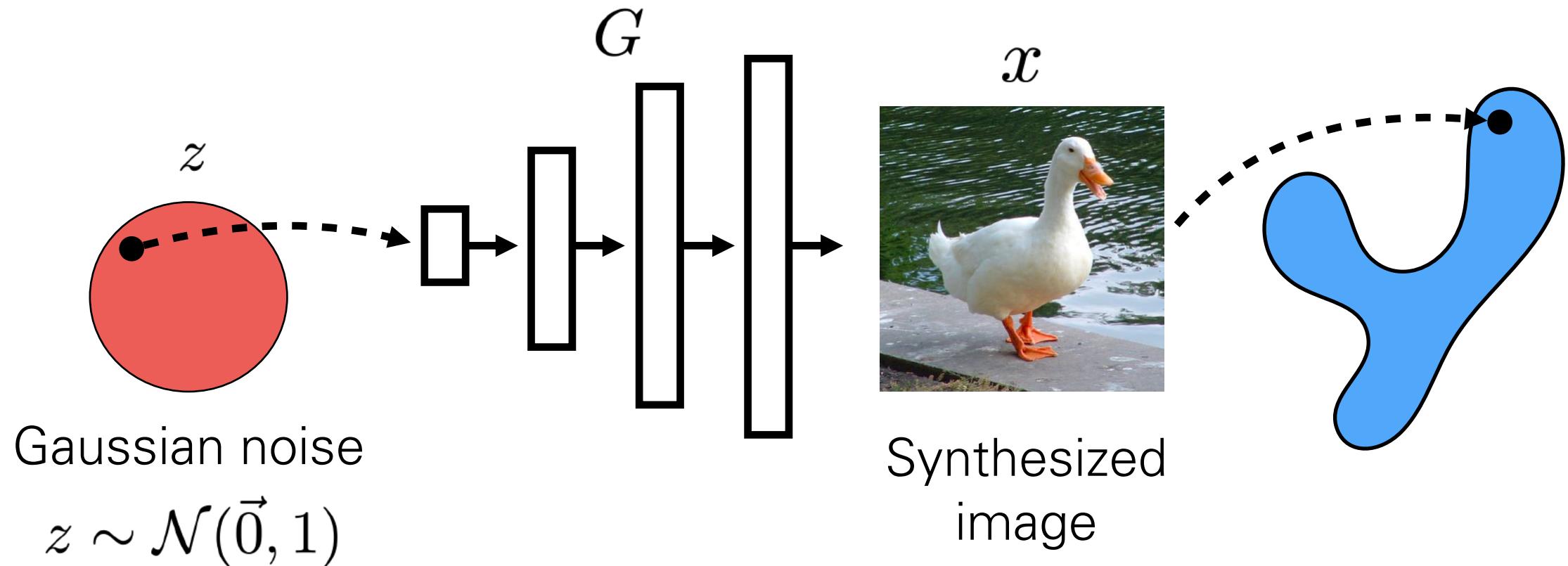
$p(z)$

$p(x)$

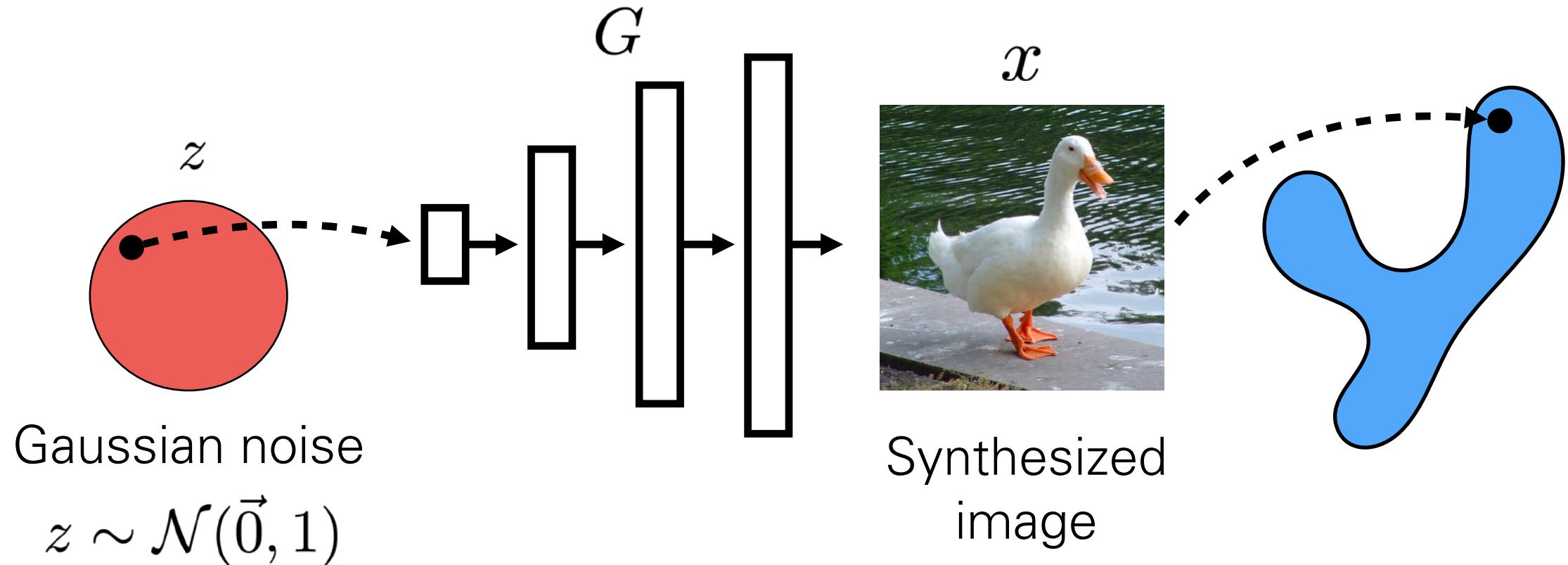
Deep generative models are distribution transformers

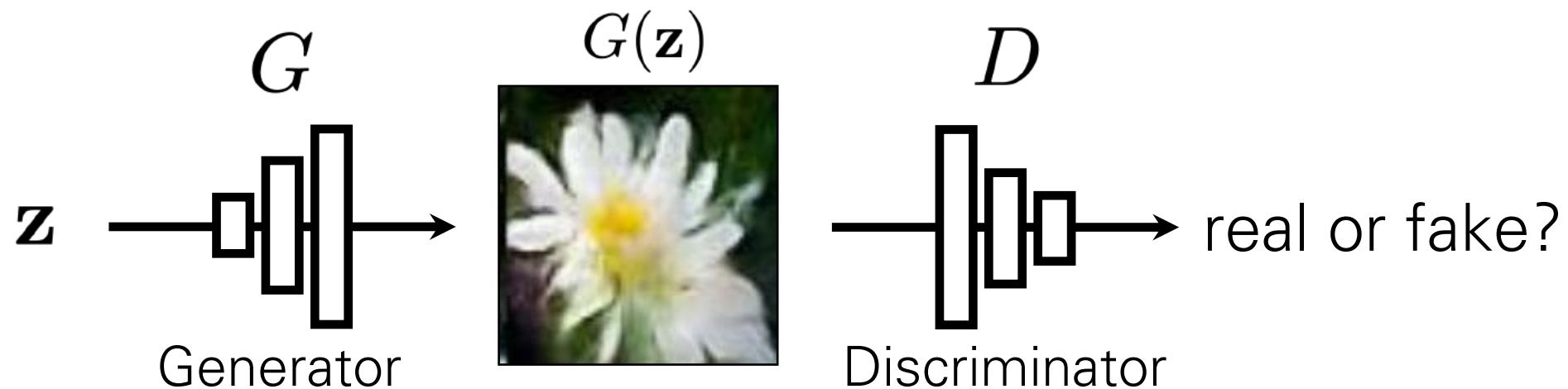


Deep generative models are distribution transformers



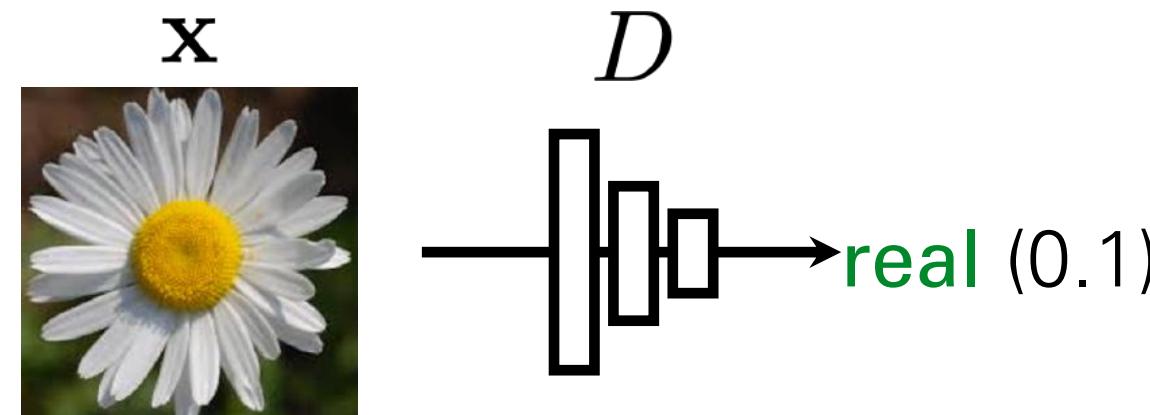
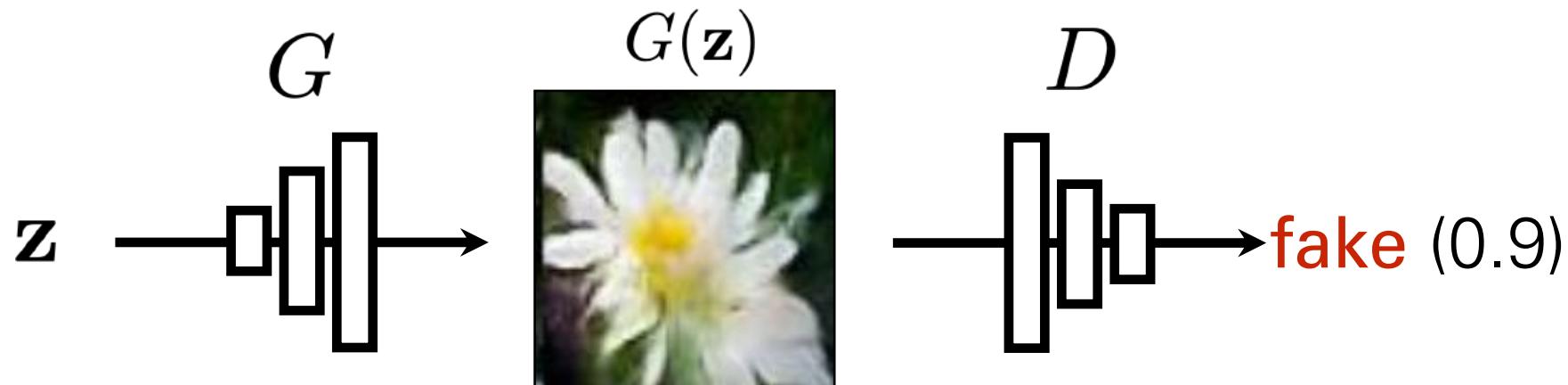
Generative Adversarial Networks (GANs)



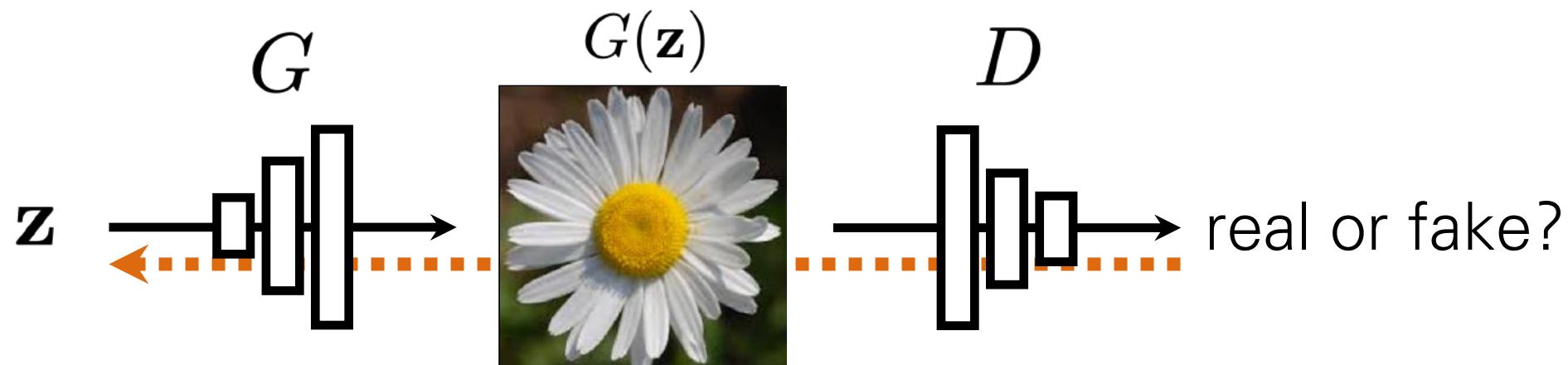


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

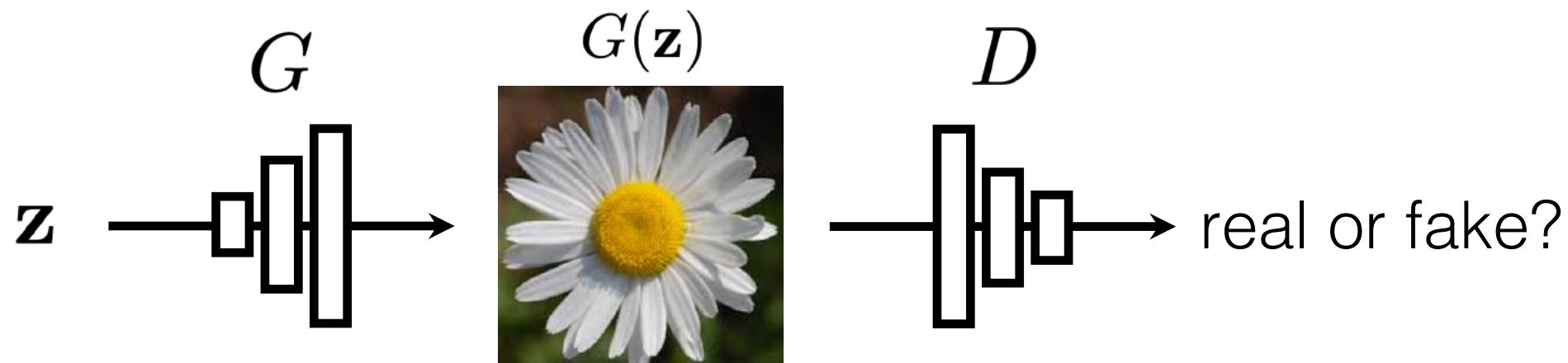


$$\arg \max_D \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\boxed{\log D(G(\mathbf{z}))} + \boxed{\log (1 - D(\mathbf{x}))}]$$



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

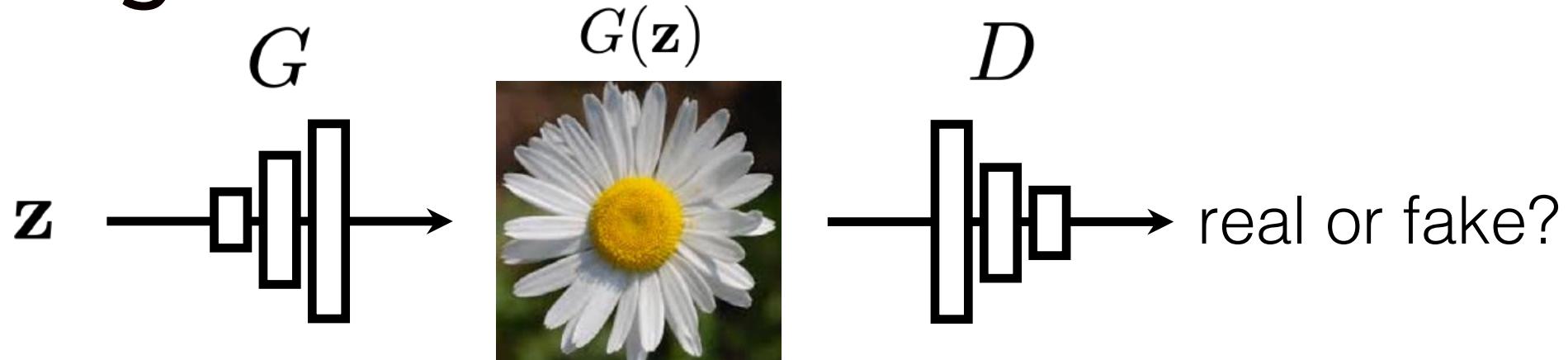
$$\arg \min_G \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\log D(G(\mathbf{z})) + \log (1 - D(\mathbf{x}))]$$



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

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

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.

GAN Training: Minimax Game

(Goodfellow et al., 2014)

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D_{\omega}(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} [\log (1 - D_{\omega}(G_{\theta}(\mathbf{z})))]$$



Real data



Noise vector used to generate data

$$J^{(D)} = -\frac{1}{2} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log D(\mathbf{x}) - \frac{1}{2} \mathbb{E}_{\mathbf{z}} \log (1 - D(G(\mathbf{z})))$$

Cross-entropy loss for binary classification

$$J^{(G)} = -\frac{1}{2} \mathbb{E}_{\mathbf{z}} \log D(G(\mathbf{z}))$$

Generator maximizes the log-probability of the discriminator being mistaken

- Equilibrium of the game
- Minimizes the Jensen-Shannon divergence

GAN Training: Minimax Game

(Goodfellow et al., 2014)

$$\min_{\theta} \max_{\omega} \mathbb{E}_{x \sim p_{\text{data}}} [\log D_{\omega}(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D_{\omega}(G_{\theta}(z)))]$$



Real data



Noise vector used to

$$J^{(D)} = -\frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}} \log D_{\omega}(x)$$

$$J^{(G)} = -\frac{1}{2} \mathbb{E}_z \log (1 - D_{\omega}(G_{\theta}(z)))$$

**Important question is
“Does this converge??”**

of the discriminator being mistaken

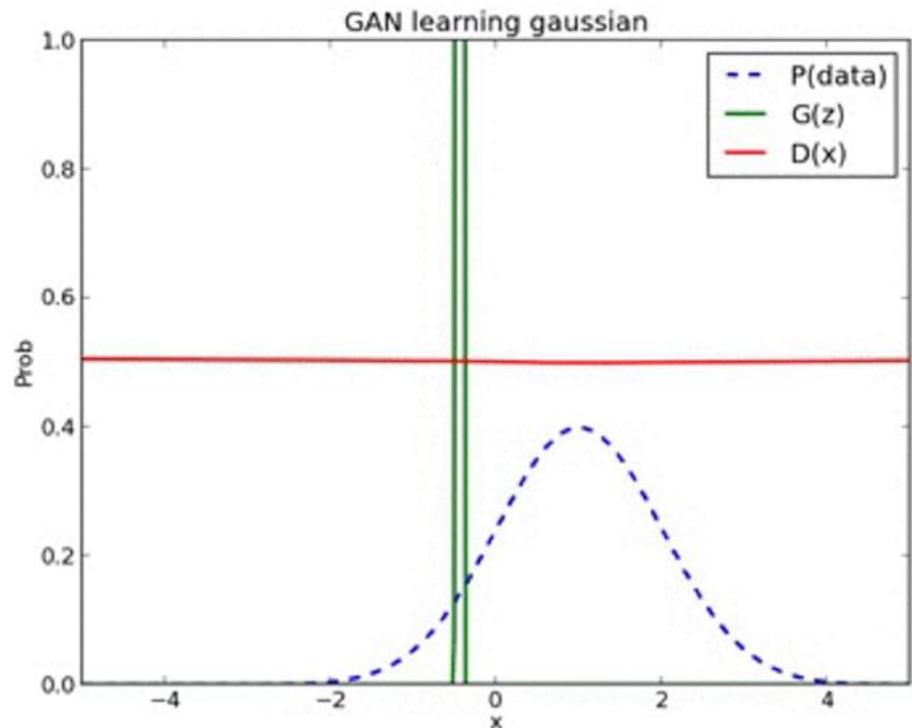
ross-entropy
ss for binary
assification

probability

- Equilibrium of the game
- Minimizes the Jensen-Shannon divergence

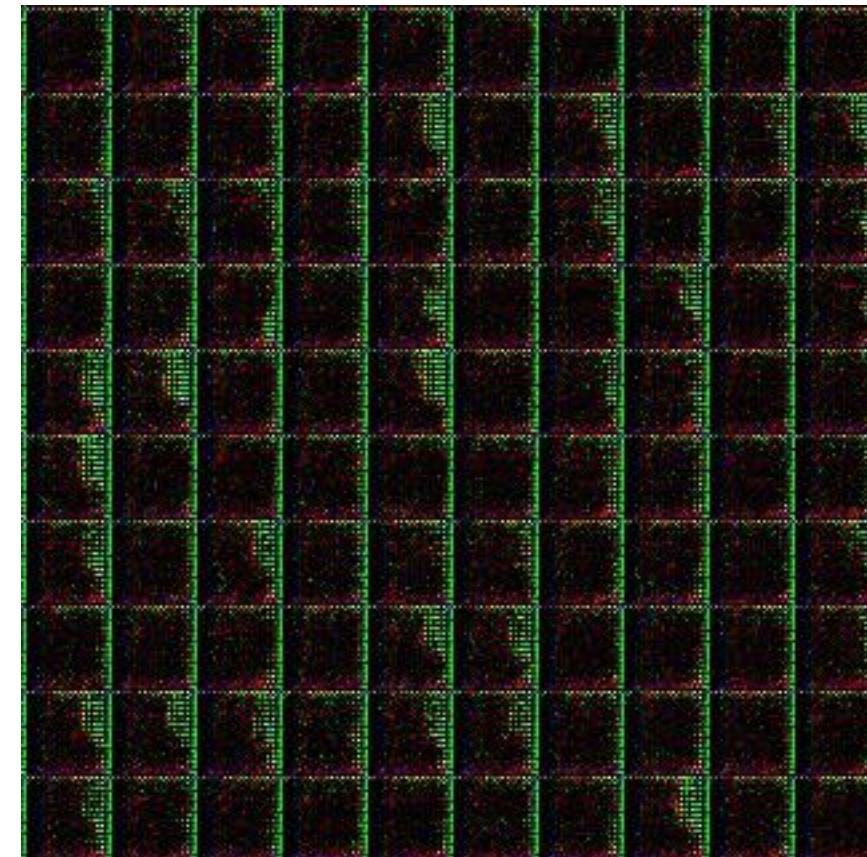
Training Procedure

(Goodfellow et al., 2014)



Source: Alec Radford

Generating 1D points



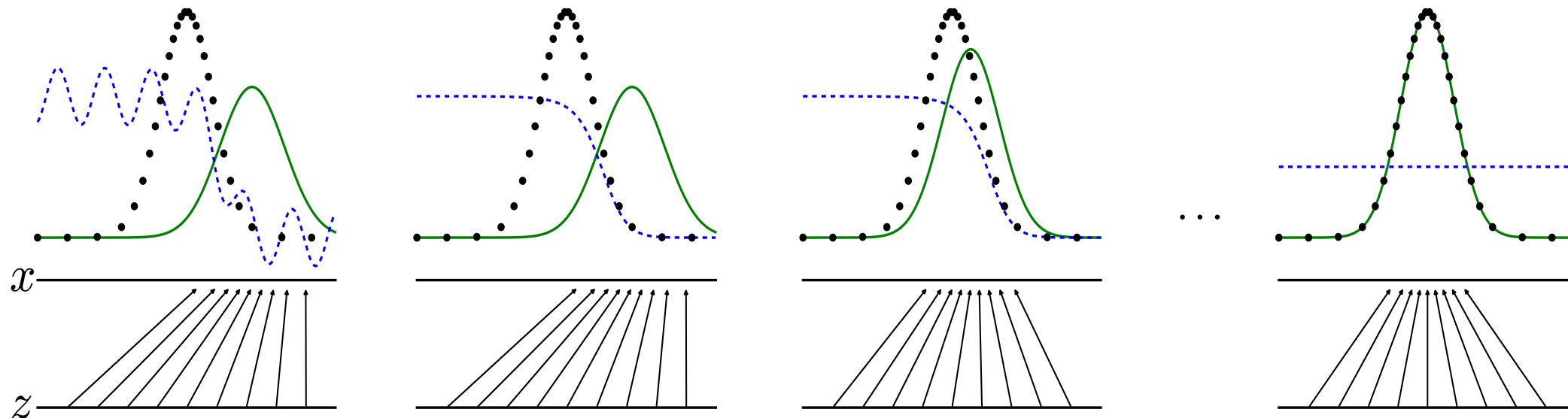
Source: OpenAI blog

Generating images

Training Procedure

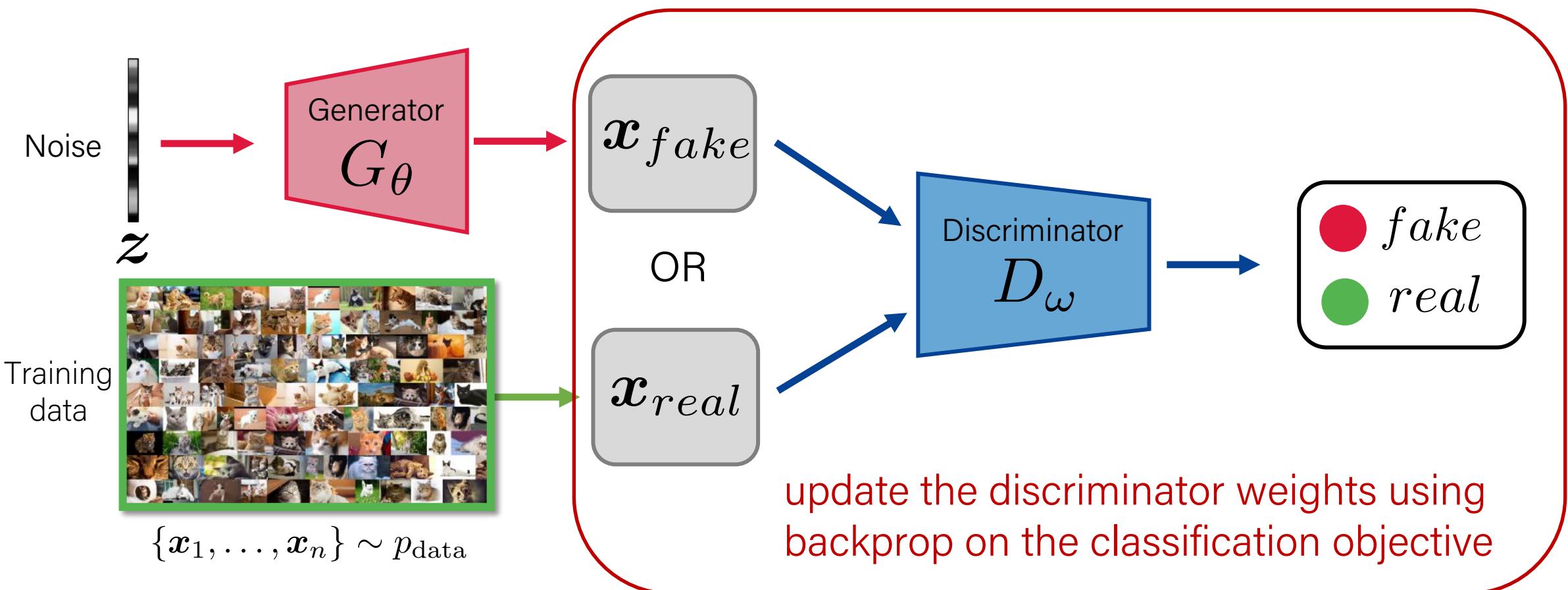
(Goodfellow et al., 2014)

- Use SGD on two minibatches simultaneously:
 - A minibatch of training examples
 - A minibatch of generated samples



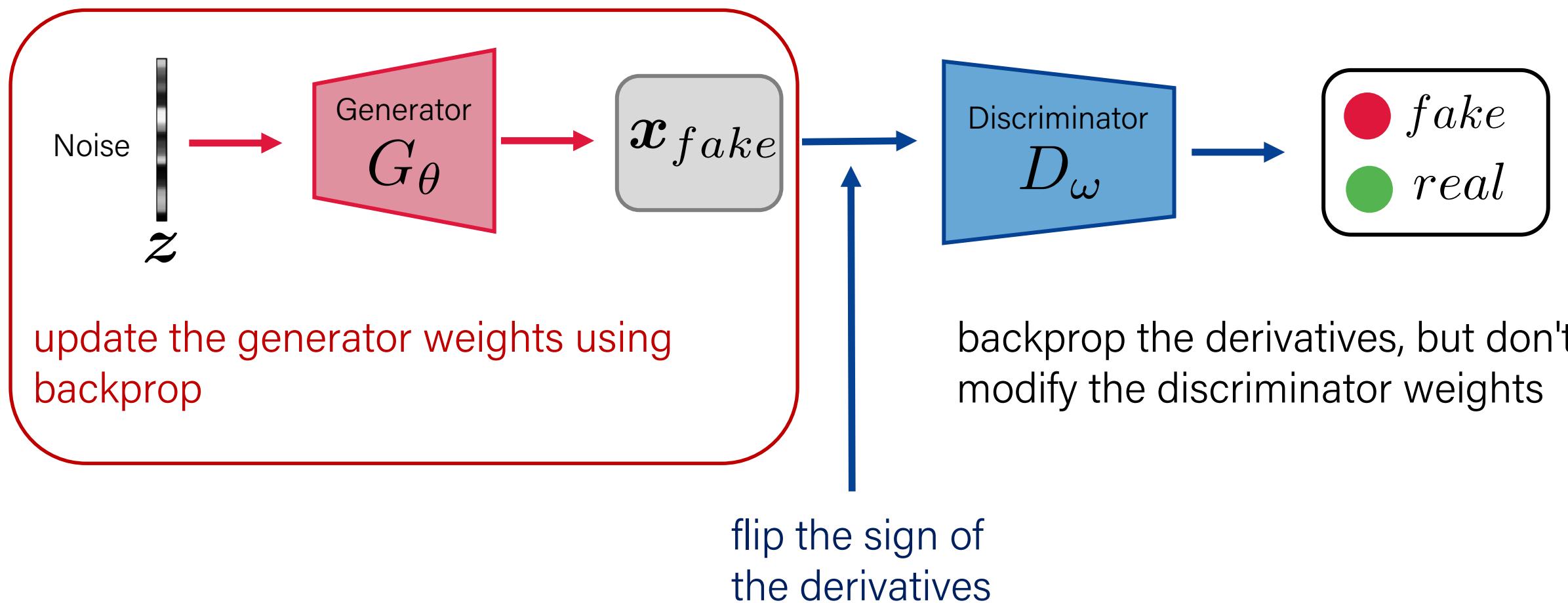
Training Procedure

- Updating the discriminator:



Training Procedure

- Updating the generator:



Early Results

(Goodfellow et al., 2014)

- The generator uses a mixture of rectifier linear activations and/or sigmoid activations
- The discriminator net used maxout activations.



MNIST samples



TFD samples



CIFAR10 samples
(fully-connected model)



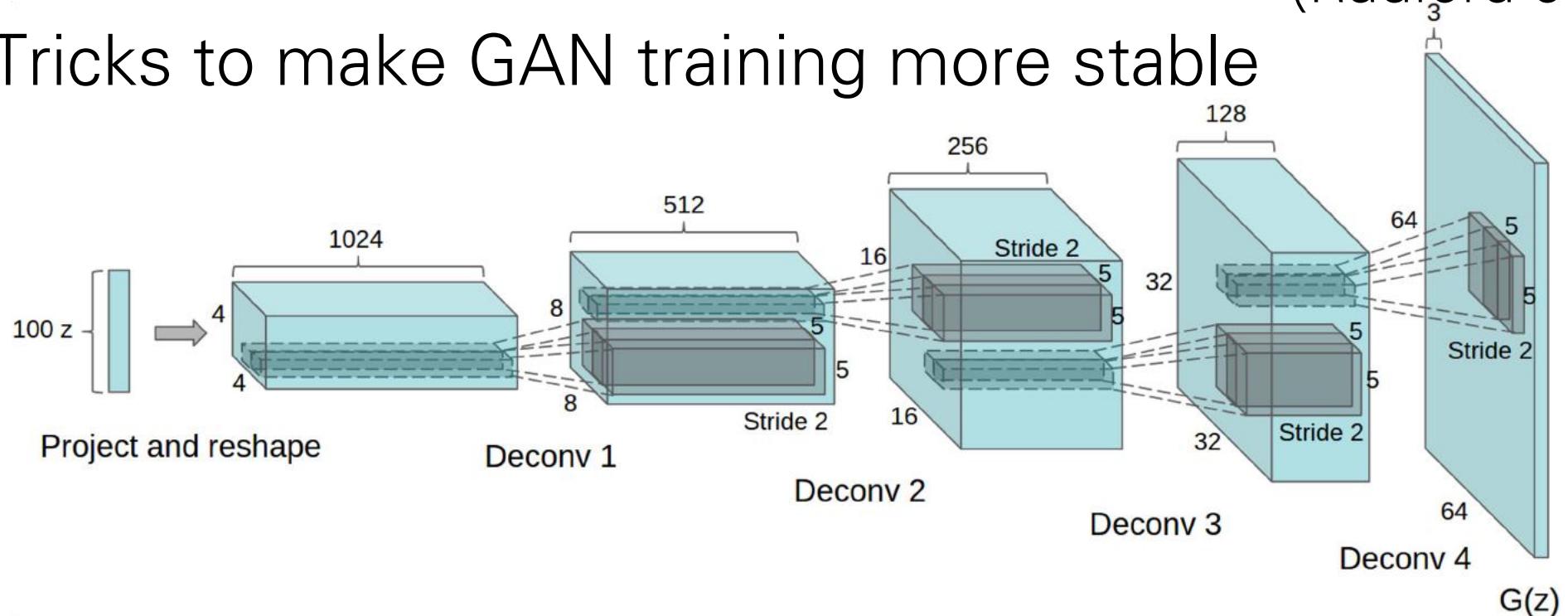
CIFAR10 samples
(convolutional discriminator,
deconvolutional generator)

Deep Convolutional GANs (DCGAN)



(Radford et al., 2015)

- Idea: Tricks to make GAN training more stable



- No fully connected layers
- Batch Normalization
(Ioffe and Szegedy, 2015)
- Leaky Rectifier in D
- Use Adam (Kingma and Ba, 2015)
- Tweak Adam hyperparameters a bit
(lr=0.0002, b1=0.5)

DCGAN for LSUN Bedrooms

64×64 pixels

~3M images

(Radford et al., 2015)



Walking over the latent space

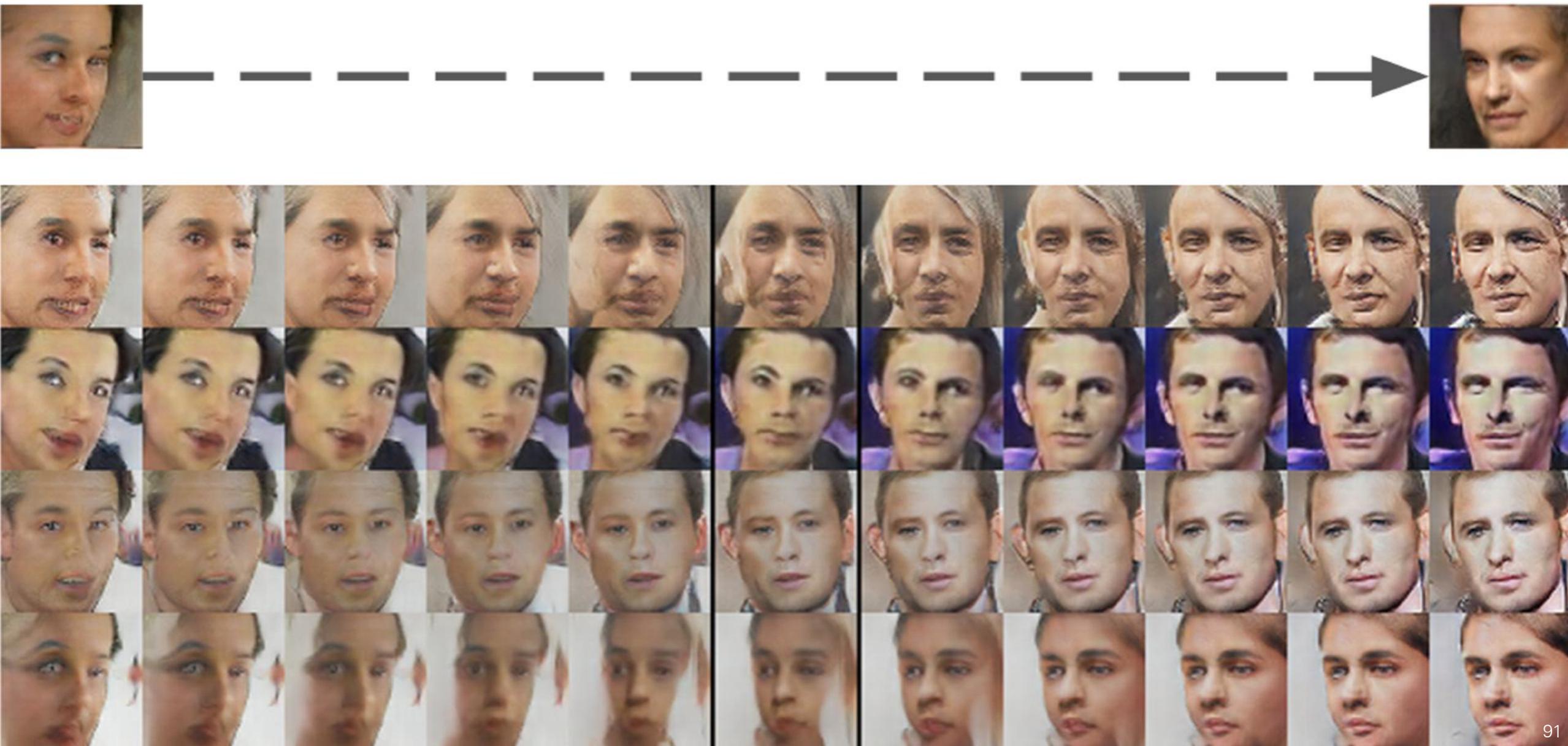
(Radford et al., 2015)

- Interpolation suggests non-overfitting behavior



Walking over the latent space

(Radford et al., 2015)



Vector Space Arithmetic

(Radford et al., 2015)



man
with glasses



man
without glasses



woman
without glasses



woman with glasses

Vector Space Arithmetic

(Radford et al., 2015)



smiling woman



neutral
woman



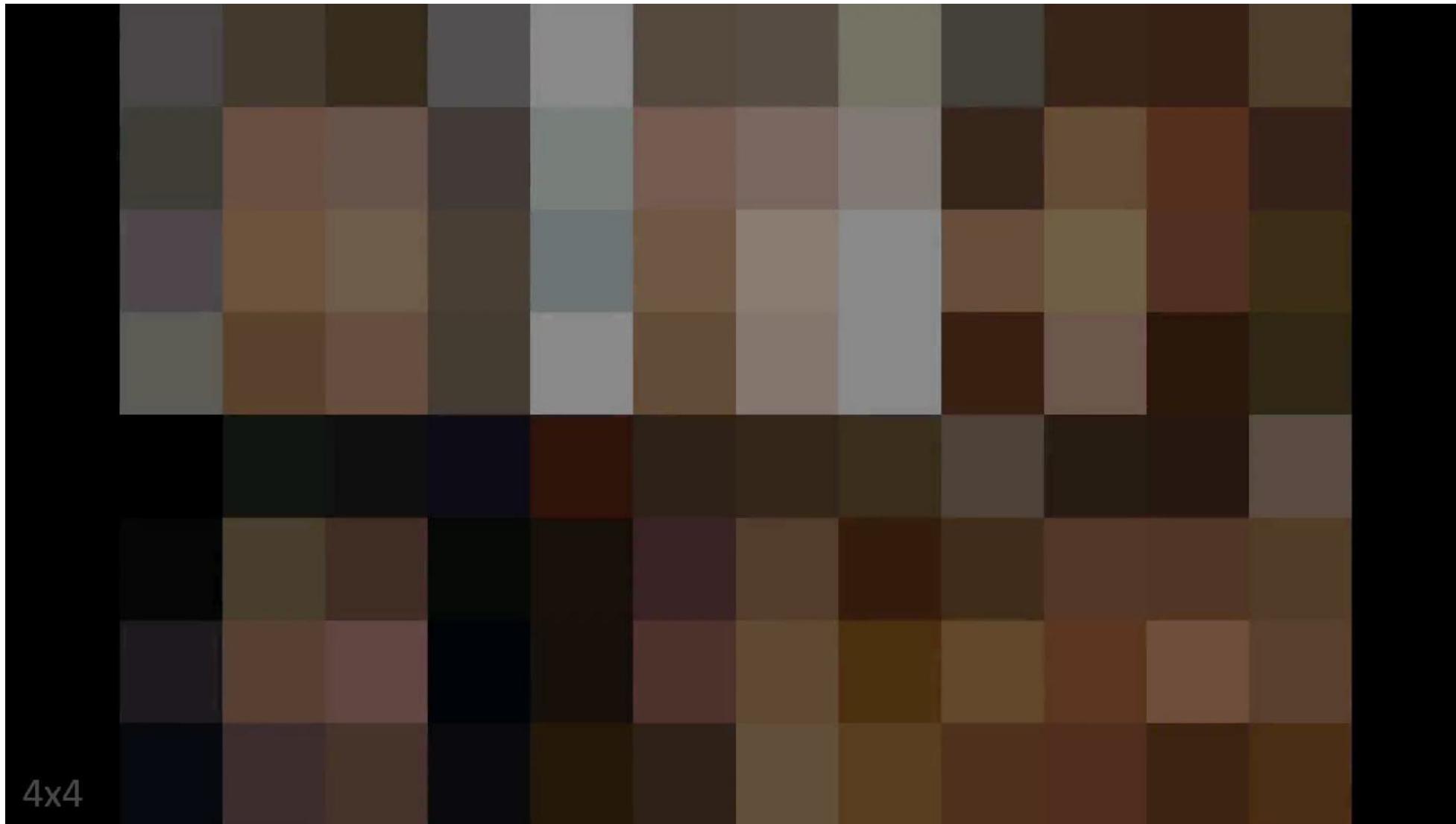
neutral
man



smiling man

Progressive GANs

(Karras et al., 2018)



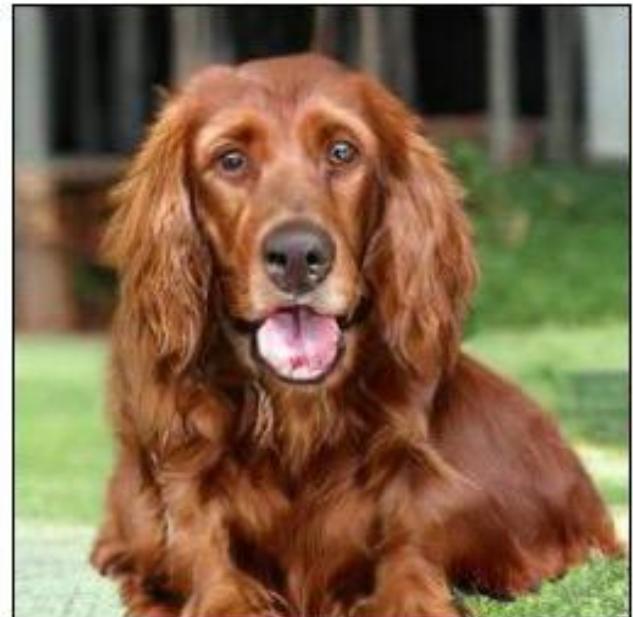
Progressive GANs (Karras et al., 2018)

CelebA-HQ
random interpolations



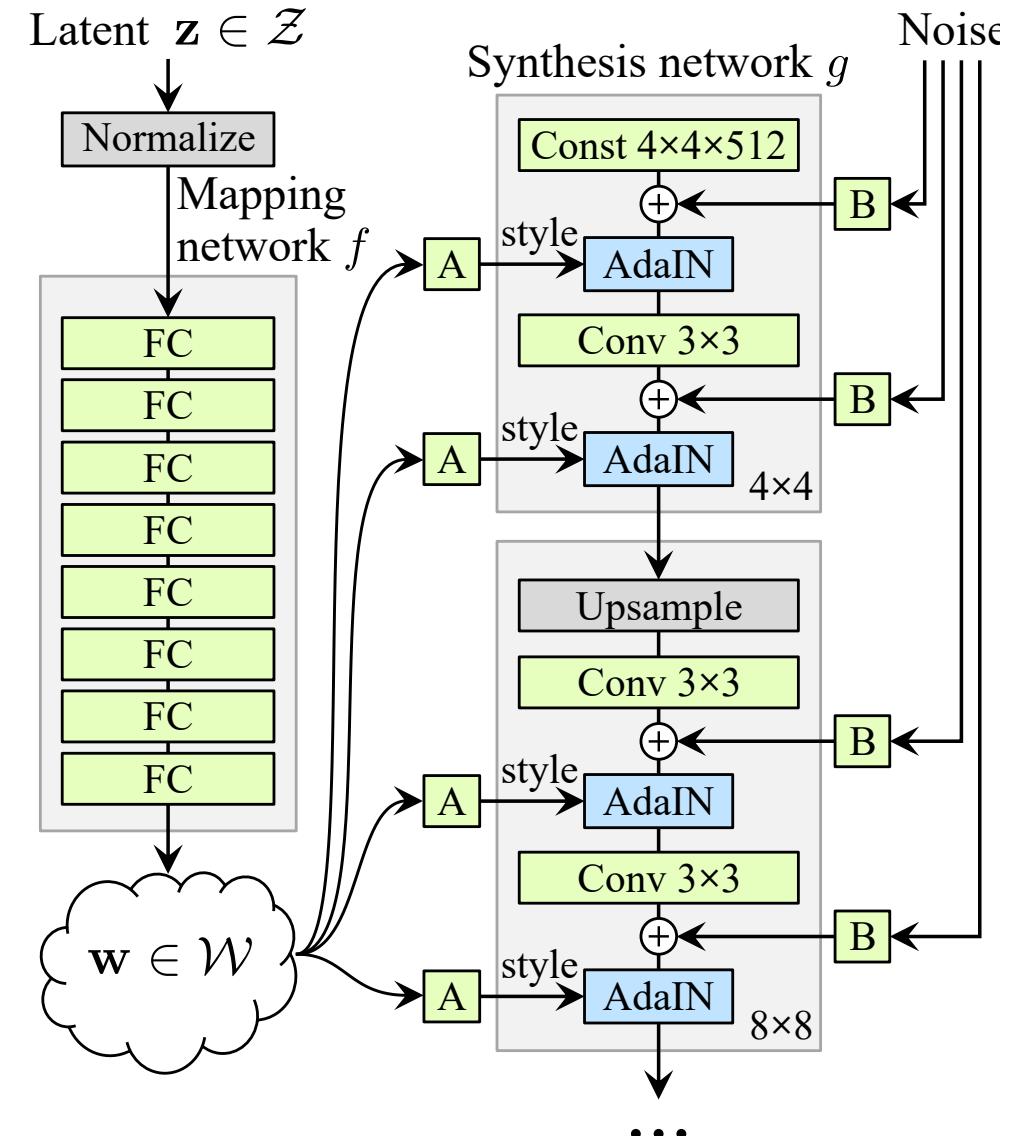
Samples from BigGAN

[Brock et al. 2018]



StyleGANs (Karras et al., 2019)

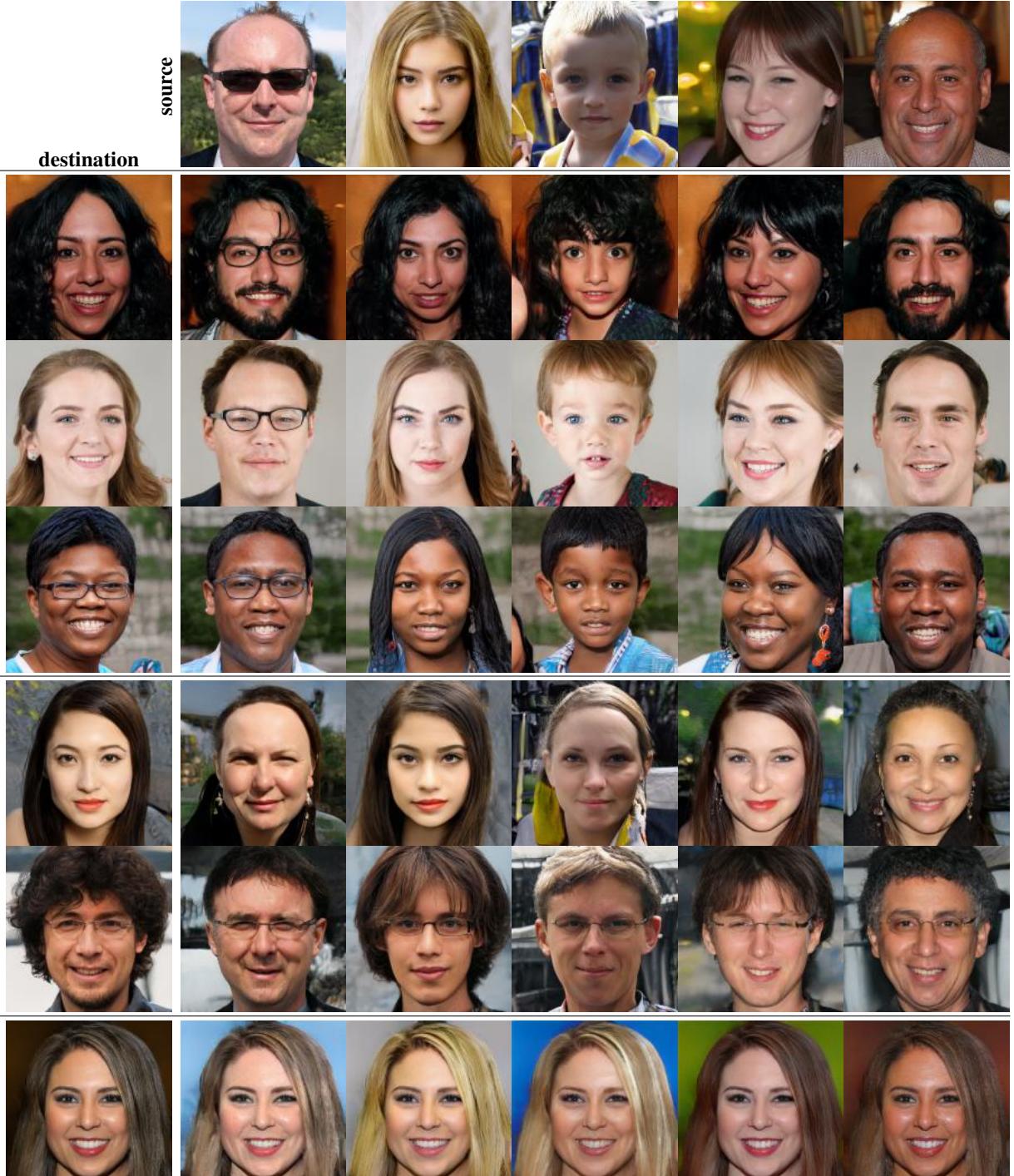
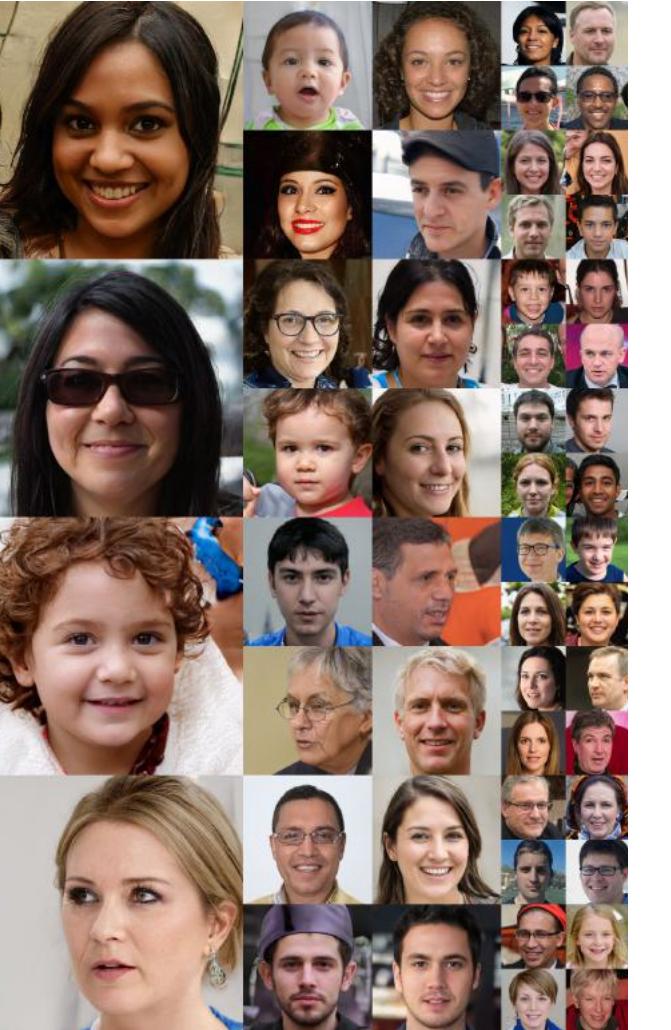
- An architecture motivated by the style transfer networks
- allows unsupervised separation of high-level attributes and stochastic variation in the generated images



[“StyleGAN”, Karras, Laine, Aila, 2018]

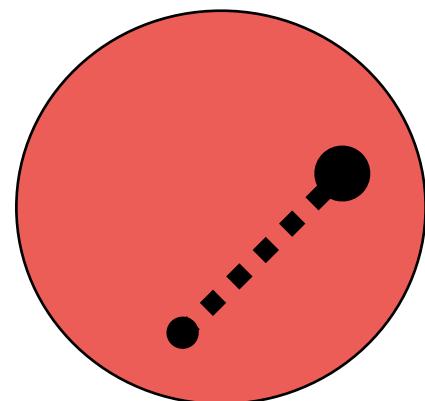
StyleGANs

(Karras et al., 2018)



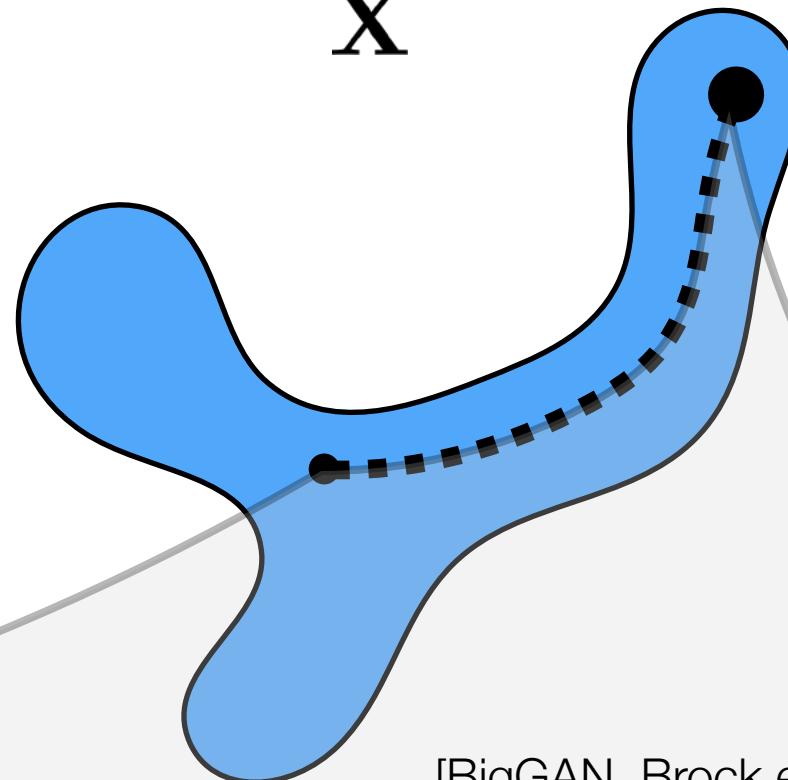
Latent space
(Gaussian)

z



Data space
(Natural image manifold)

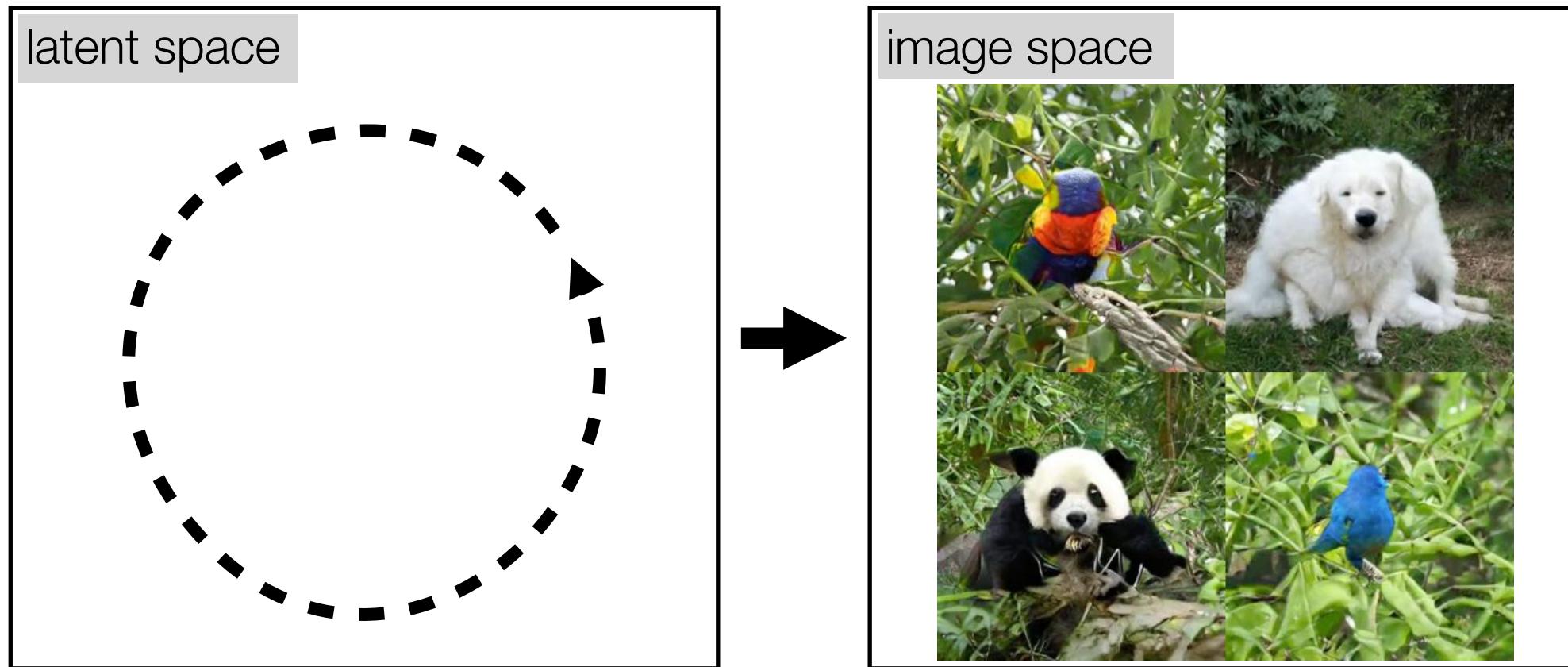
x



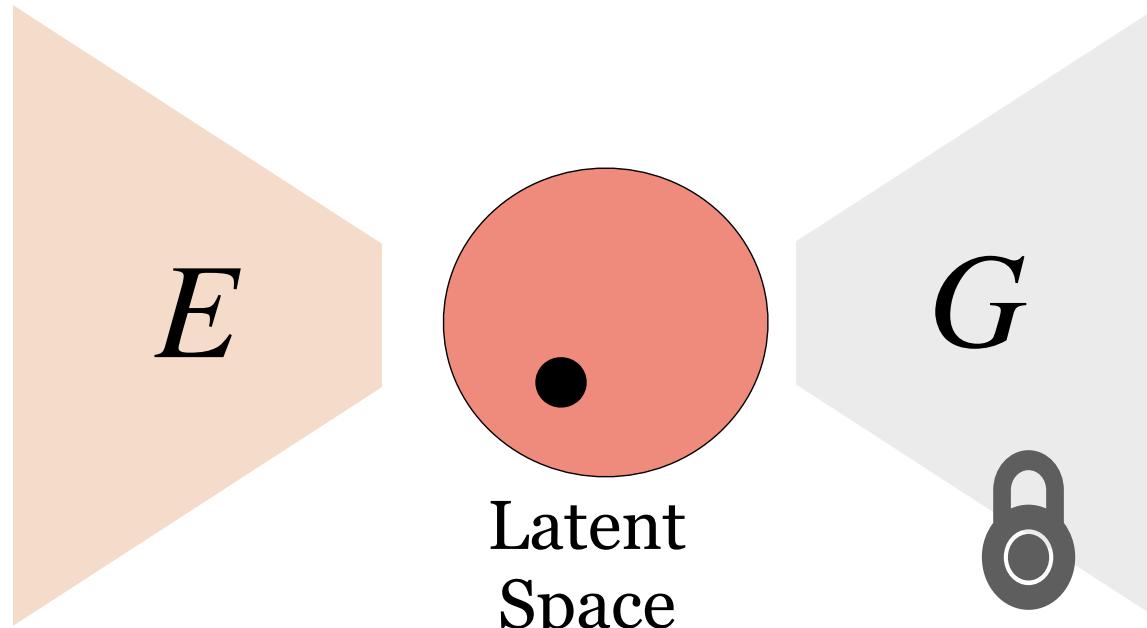
[BigGAN, Brock et al. 2018]



Generative models organize the manifold of natural images

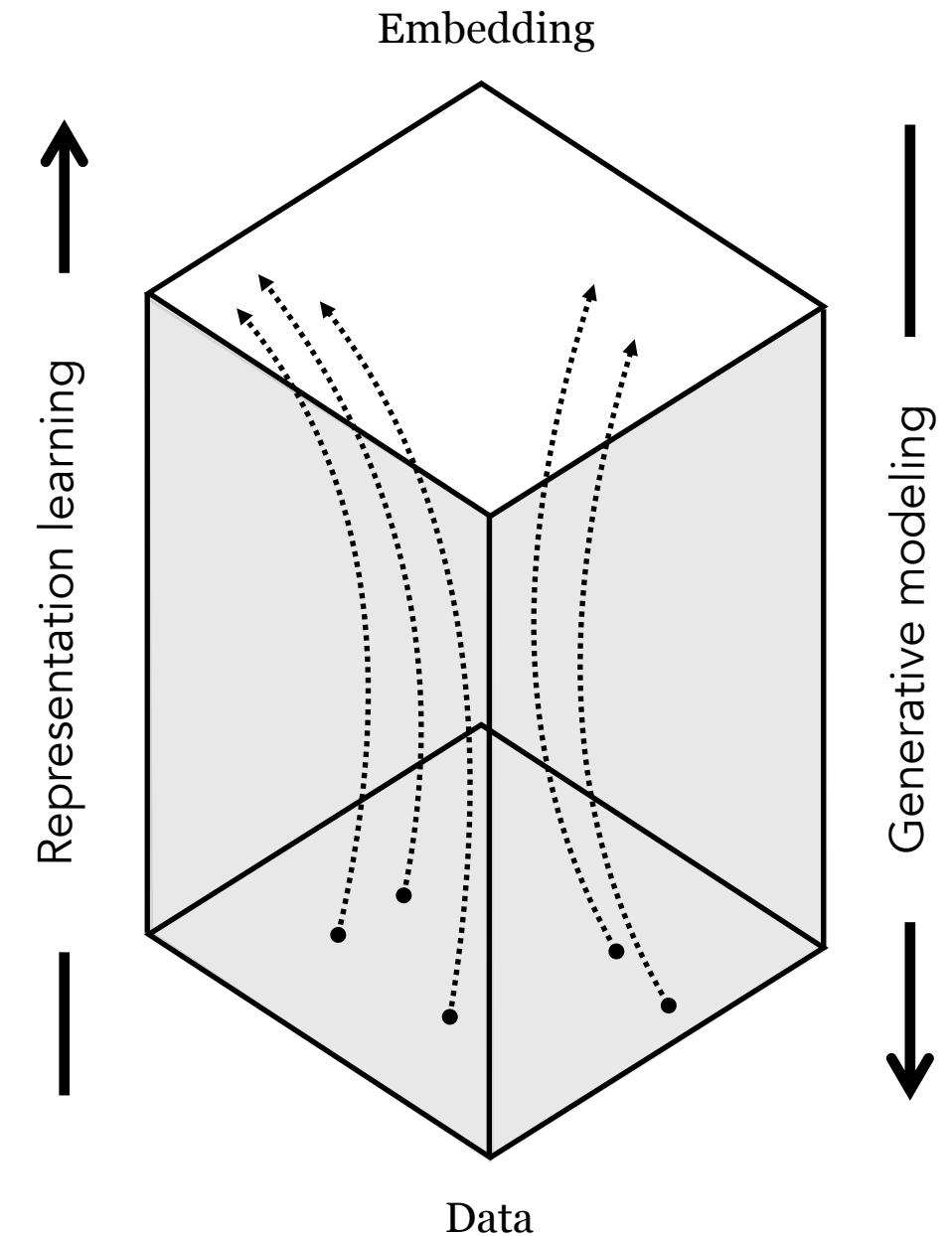


Projecting images into GAN latent space



$$w^* = \arg \min_w L_{\text{img}}(x, G(w)) + \lambda L_{\text{latent}}(w, E(x))$$

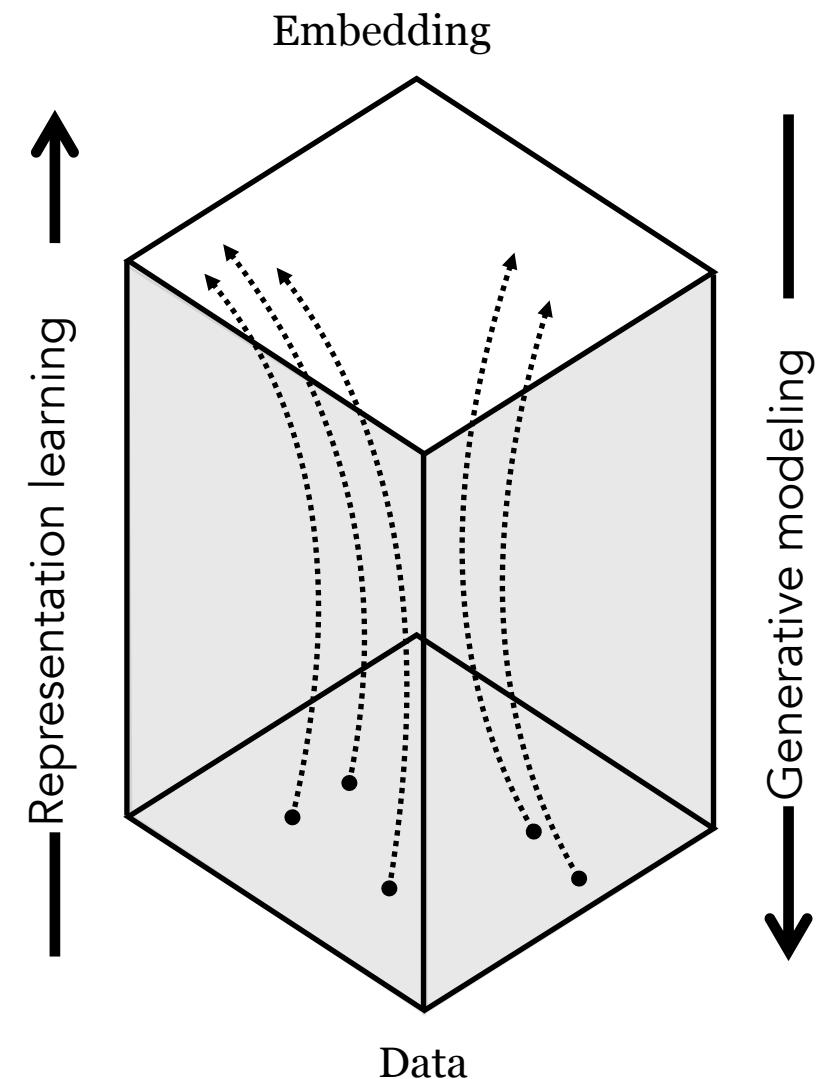
Representation Learning



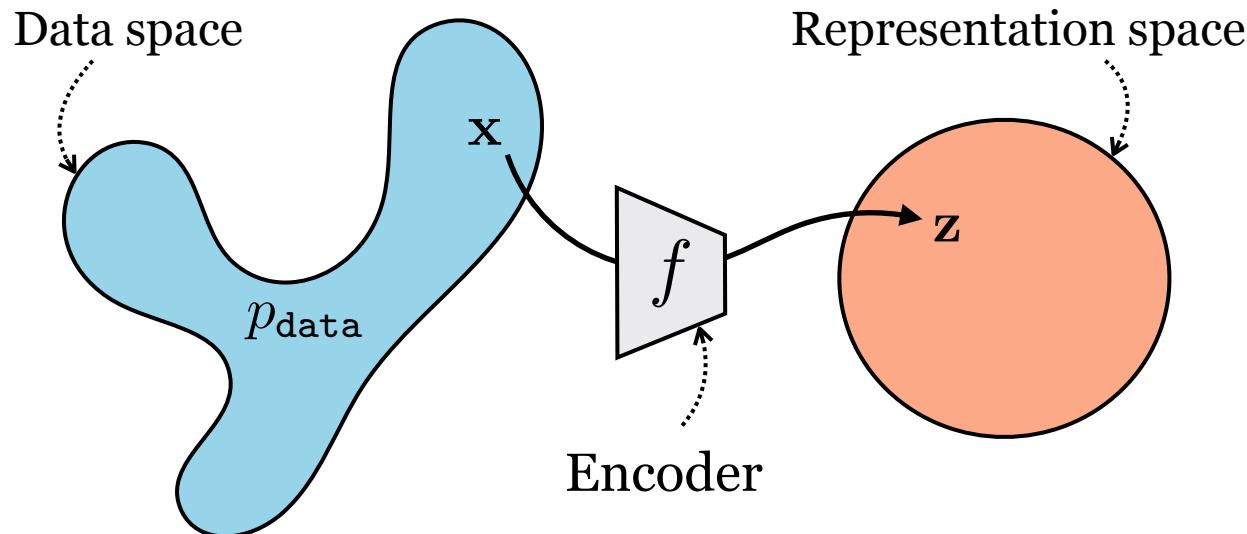
Generative modeling vs Representation learning

Representation learning:
mapping data to abstract representations
(analysis)

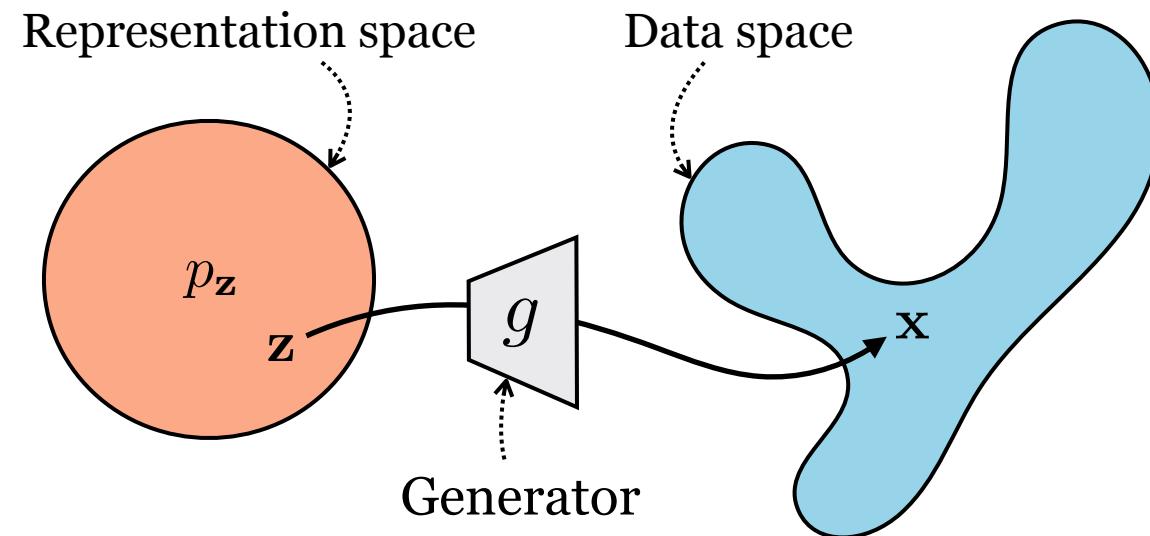
Generative modeling:
mapping abstract representations to data
(synthesis)



Representation learning



Generative modeling



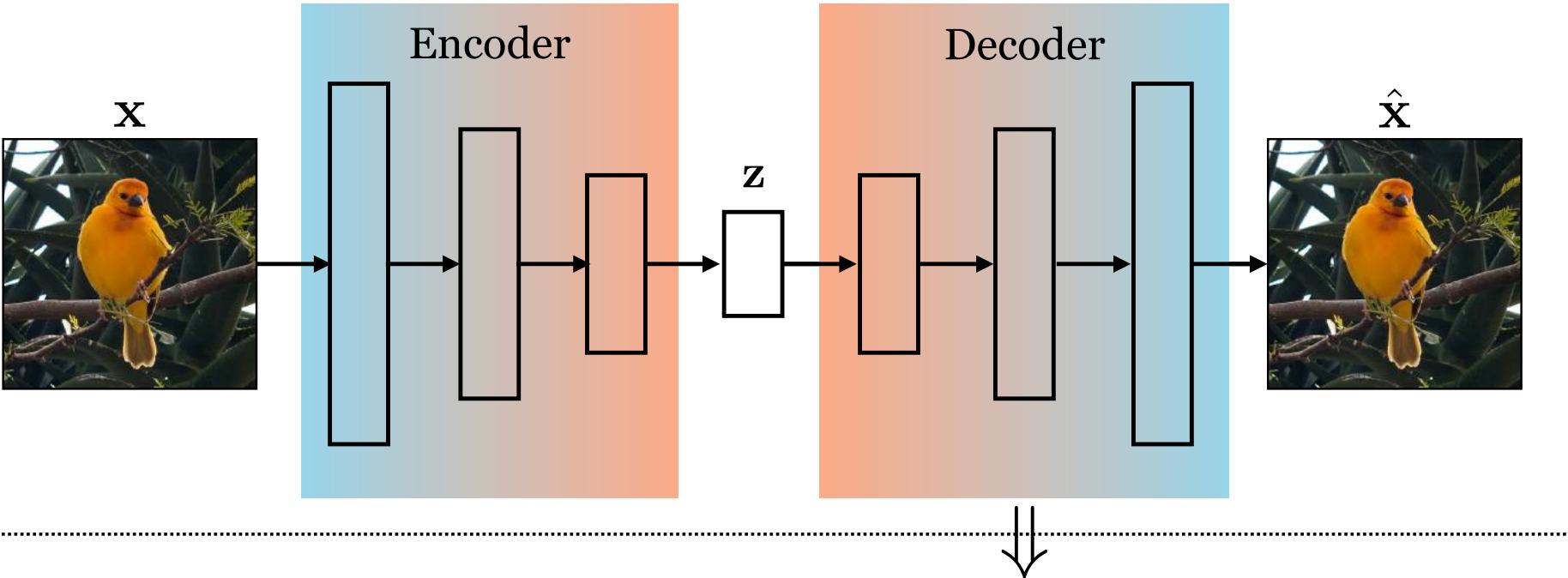
Generative model

$$z \sim p_z \rightarrow \boxed{z}$$

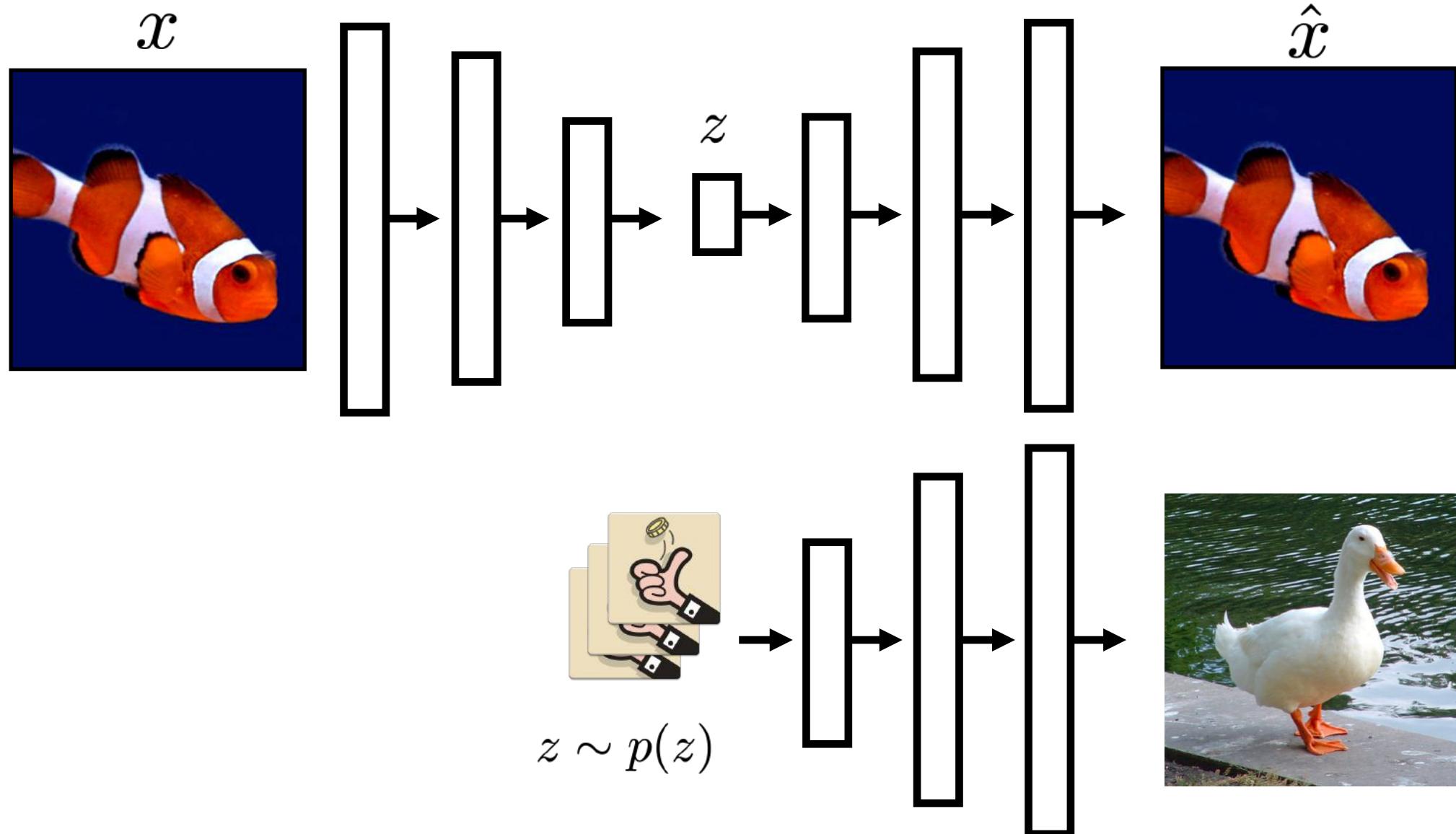


⋮

Autoencoder



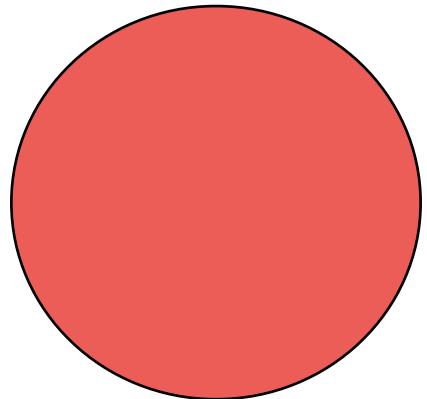
Autoencoder → Generative model



Variational Autoencoder (VAE)

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

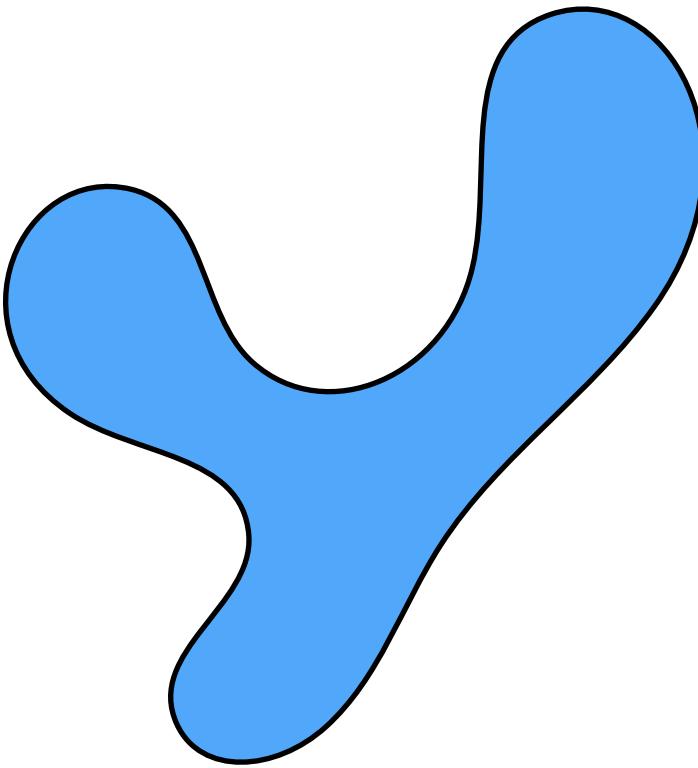
Prior distribution



$$p(z)$$

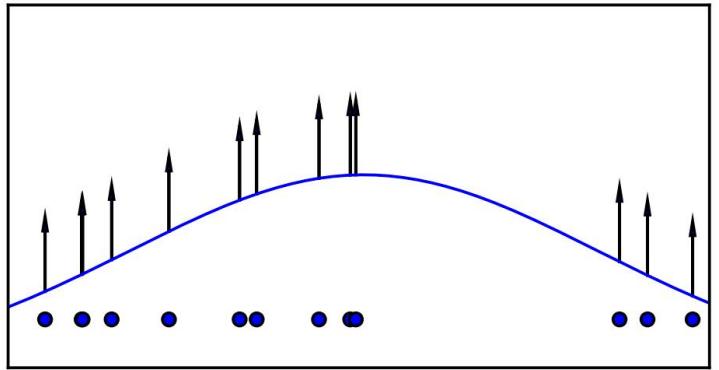


Target distribution



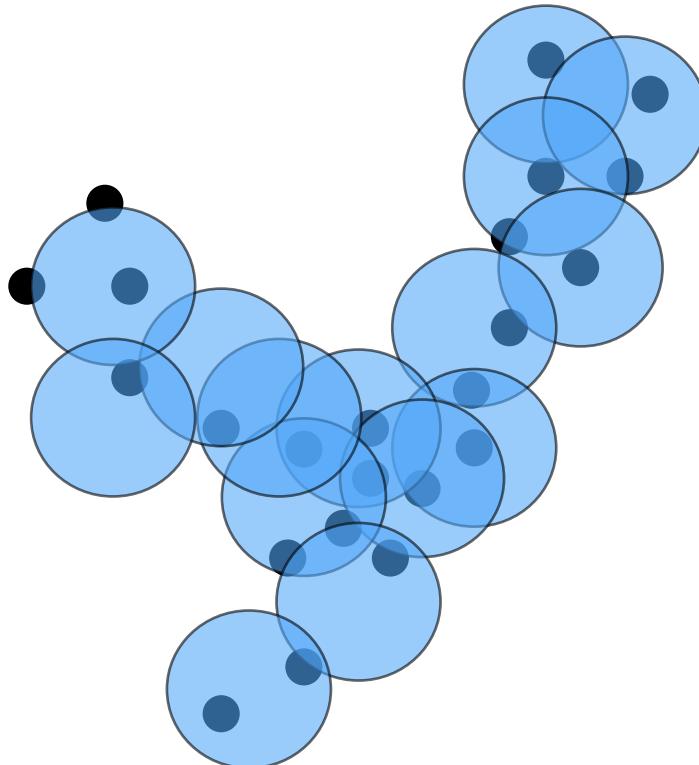
$$p(x)$$

Mixture of Gaussians



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

Target distribution



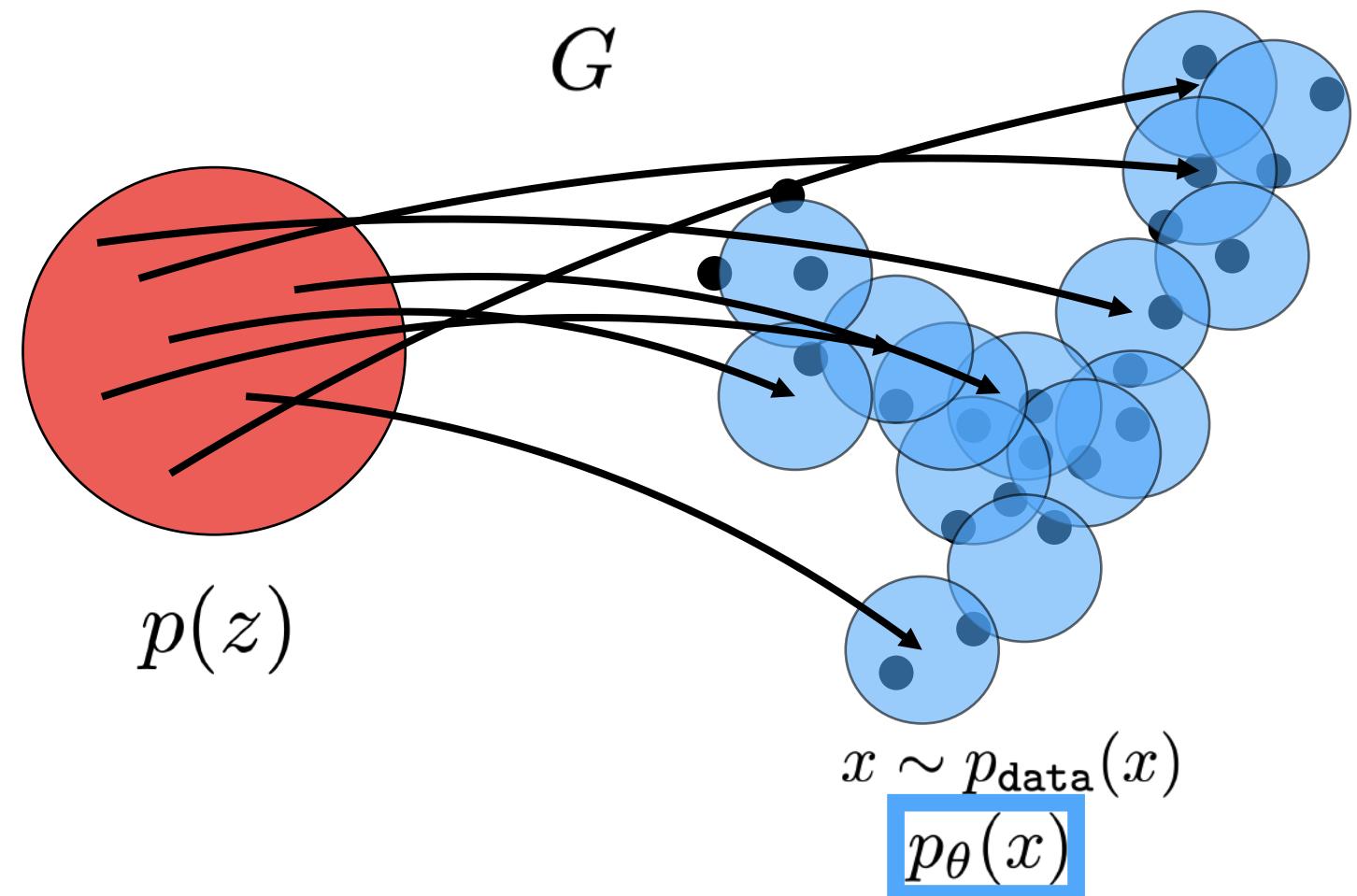
$$x \sim p_{\text{data}}(x)$$

$$p_{\theta}(x)$$

Variational Autoencoder (VAE)

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

Prior distribution



Target distribution

Density model:

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

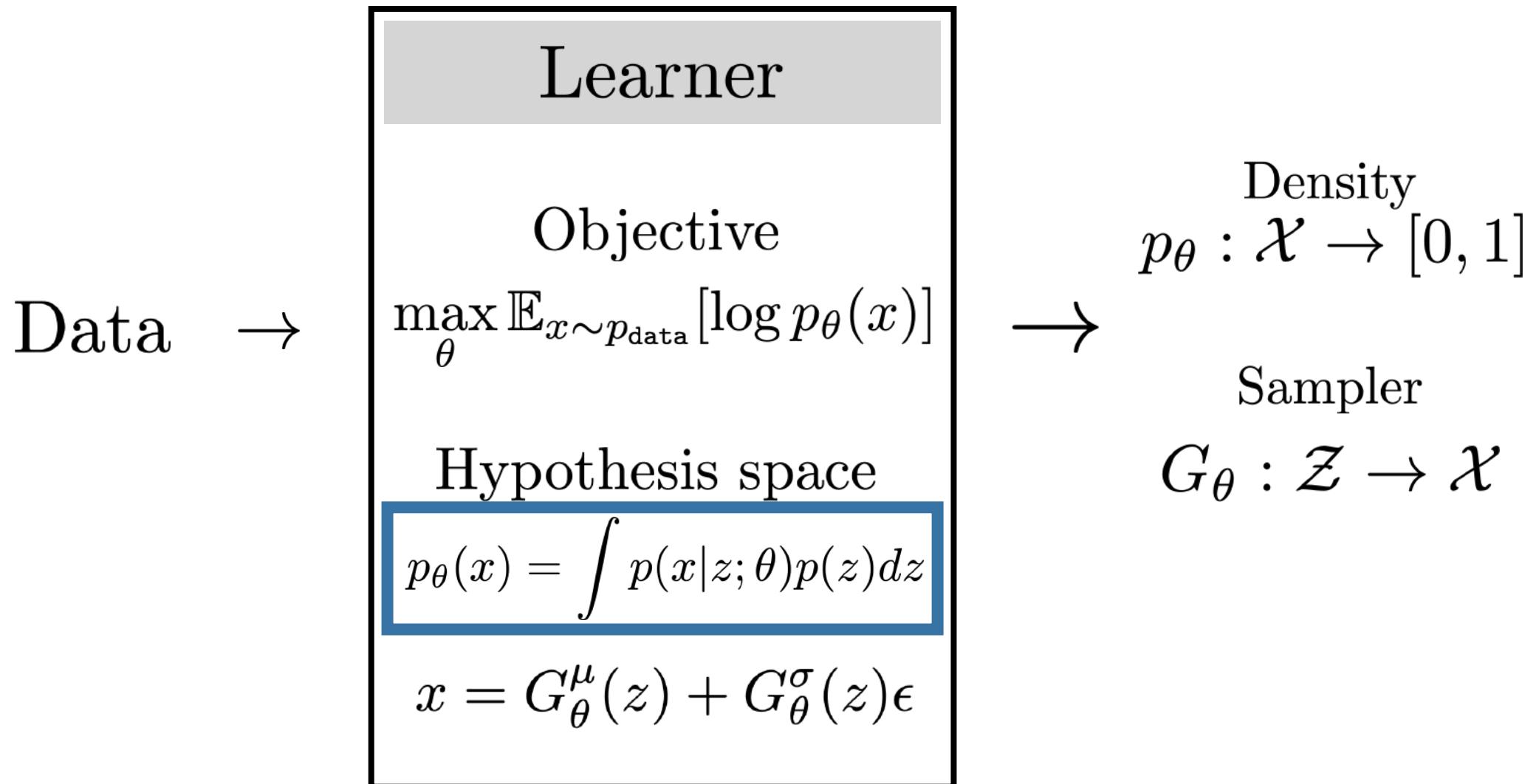
$$p(x|z; \theta) \sim \mathcal{N}(x; G_{\theta}^{\mu}(z), G_{\theta}^{\sigma}(z))$$

Sampling:

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

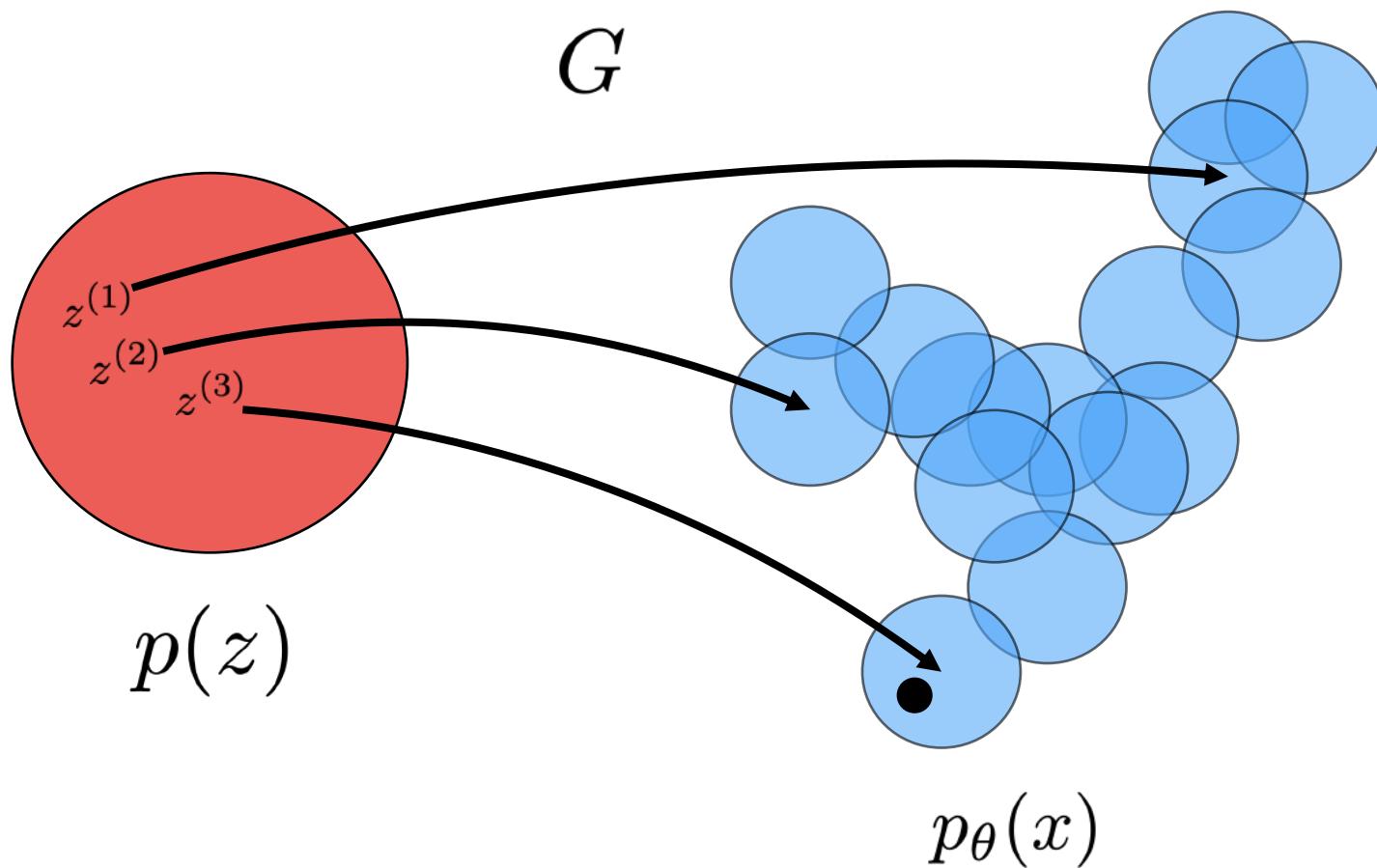
$$x = G_{\theta}^{\mu}(z) + G_{\theta}^{\sigma}(z)\epsilon$$

Variational Autoencoder (VAE)



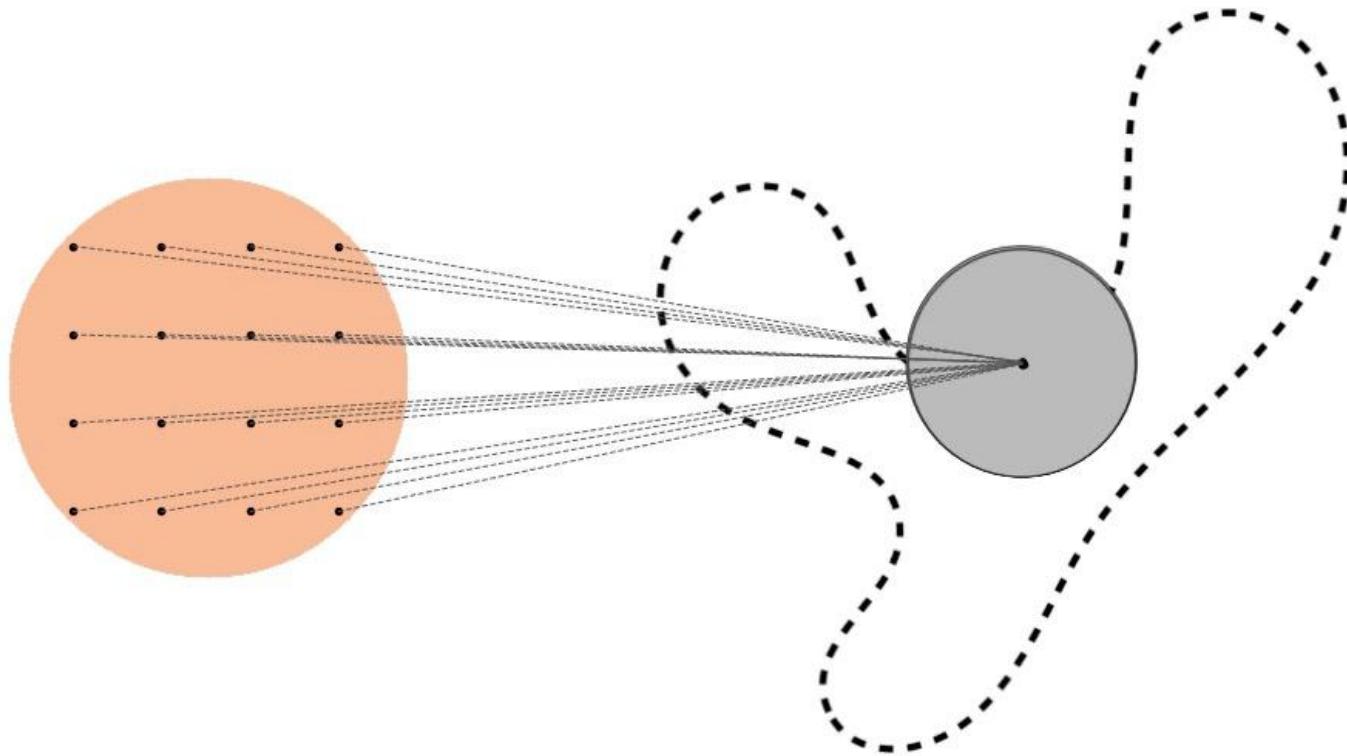
Prior distribution

Current model of target distribution



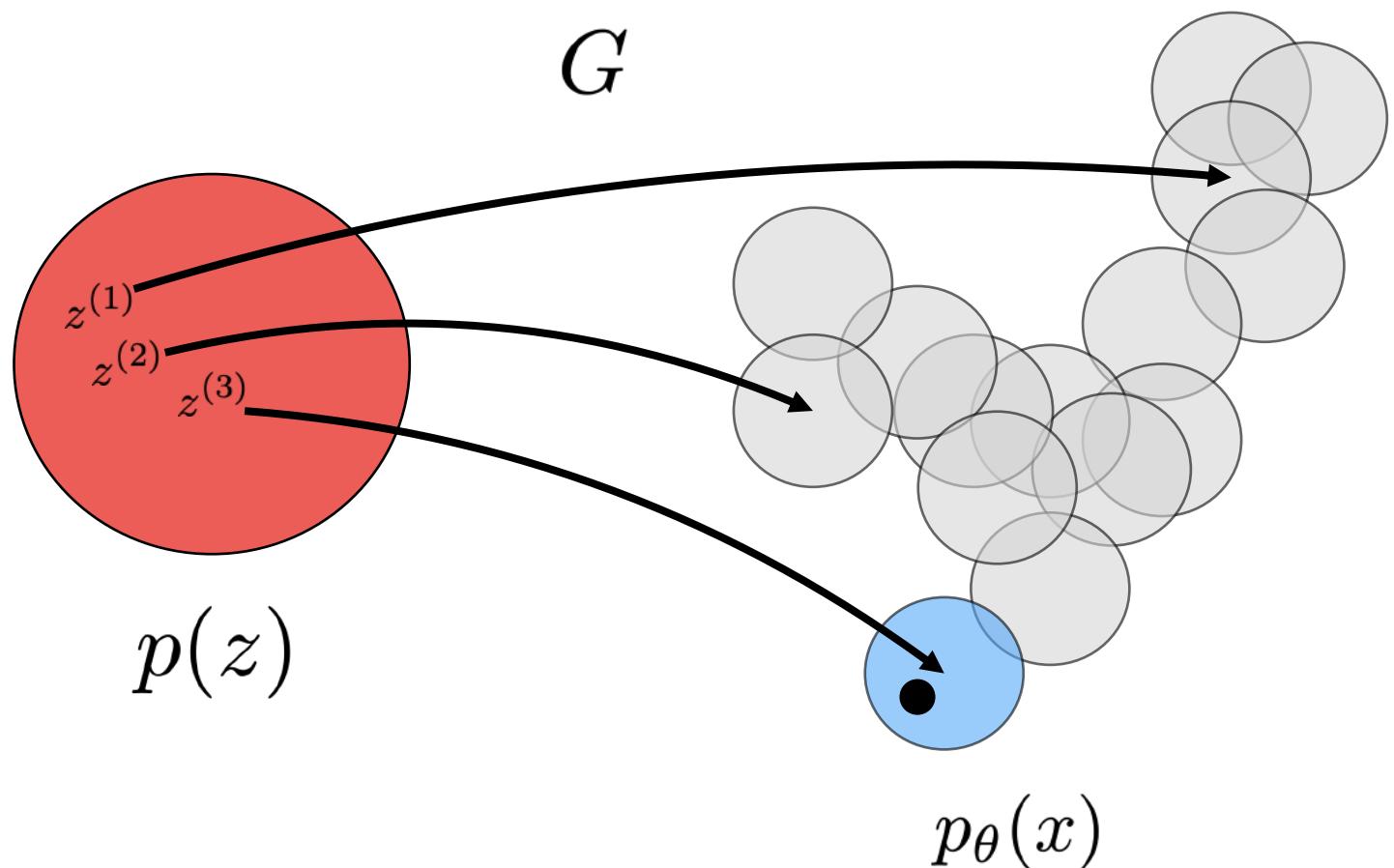
In order to optimize our model, we need to measure the likelihood it assigns to each datapoint x

$$\begin{aligned} p_{\theta}(x) &= \int p(x|z; \theta)p(z)dz \\ &= p(x|z^{(1)})p(z^{(1)})dz + \\ &\quad p(x|z^{(2)})p(z^{(2)})dz + \\ &\quad p(x|z^{(3)})p(z^{(3)})dz + \dots \end{aligned}$$



Prior distribution

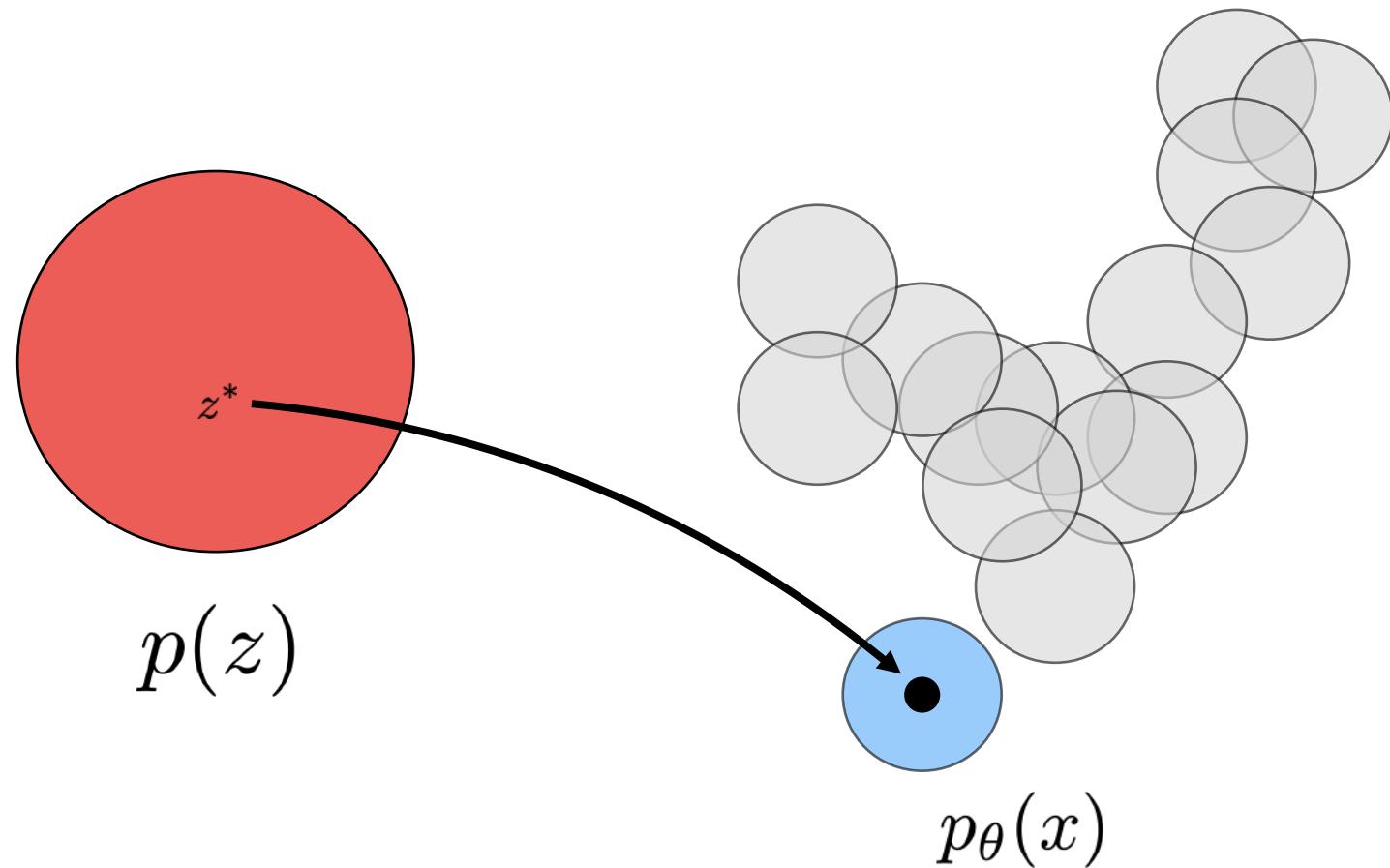
Current model of target distribution



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

$$\begin{aligned} p_\theta(x) &= \int p(x|z; \theta)p(z)dz \\ &= \sim 0+ \\ &\quad \sim 0+ \\ &\quad p(x|z^{(3)})p(z^{(3)})dz + \dots \end{aligned}$$

Prior distribution

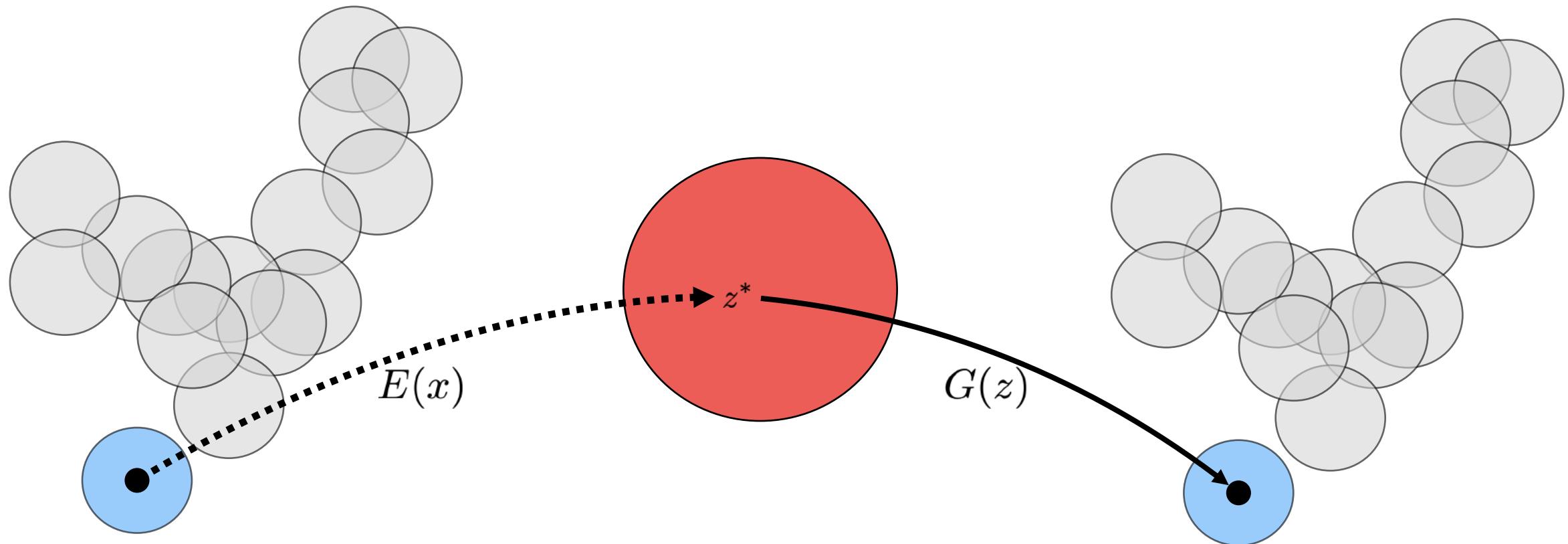


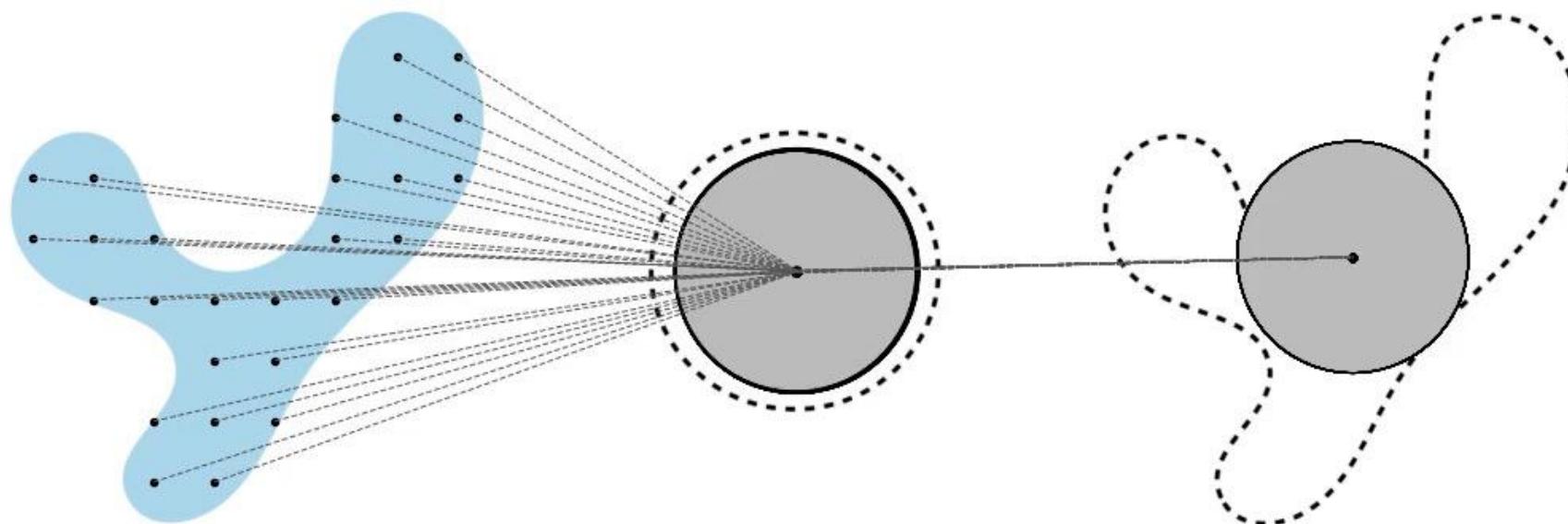
Current model of
target distribution

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

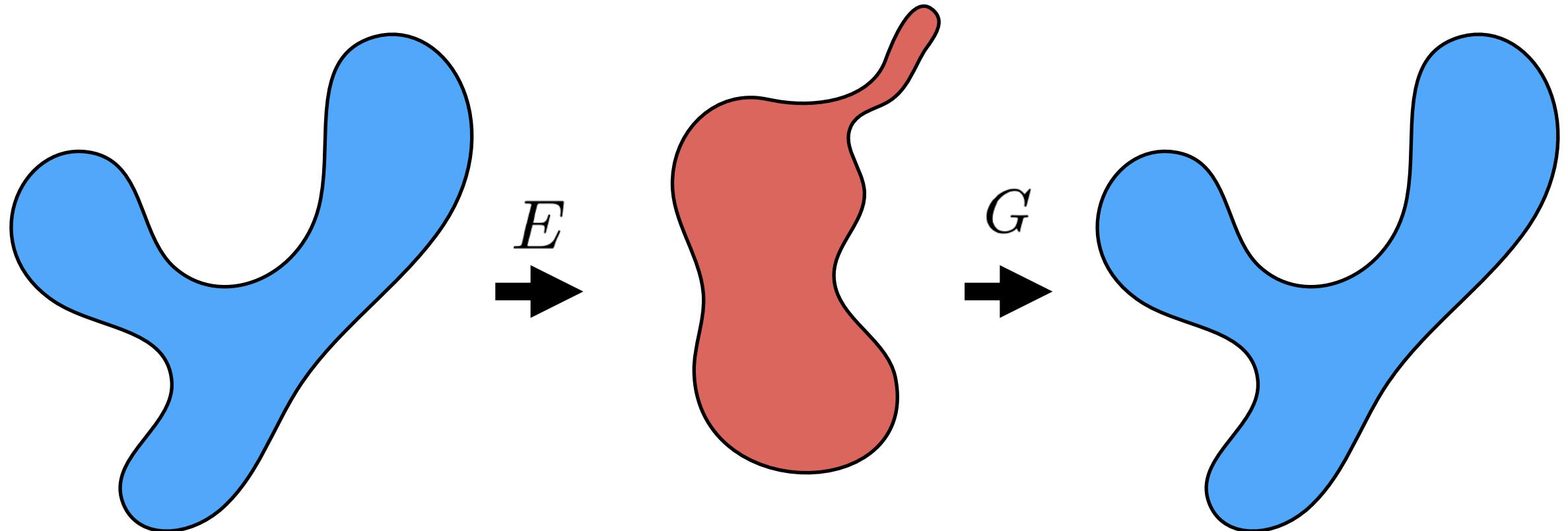
$$\begin{aligned} p_\theta(x) &= \int p(x|z; \theta)p(z)dz \\ &\approx p(x|z^*; \theta)p(z^*) \end{aligned}$$

Autoencoder!



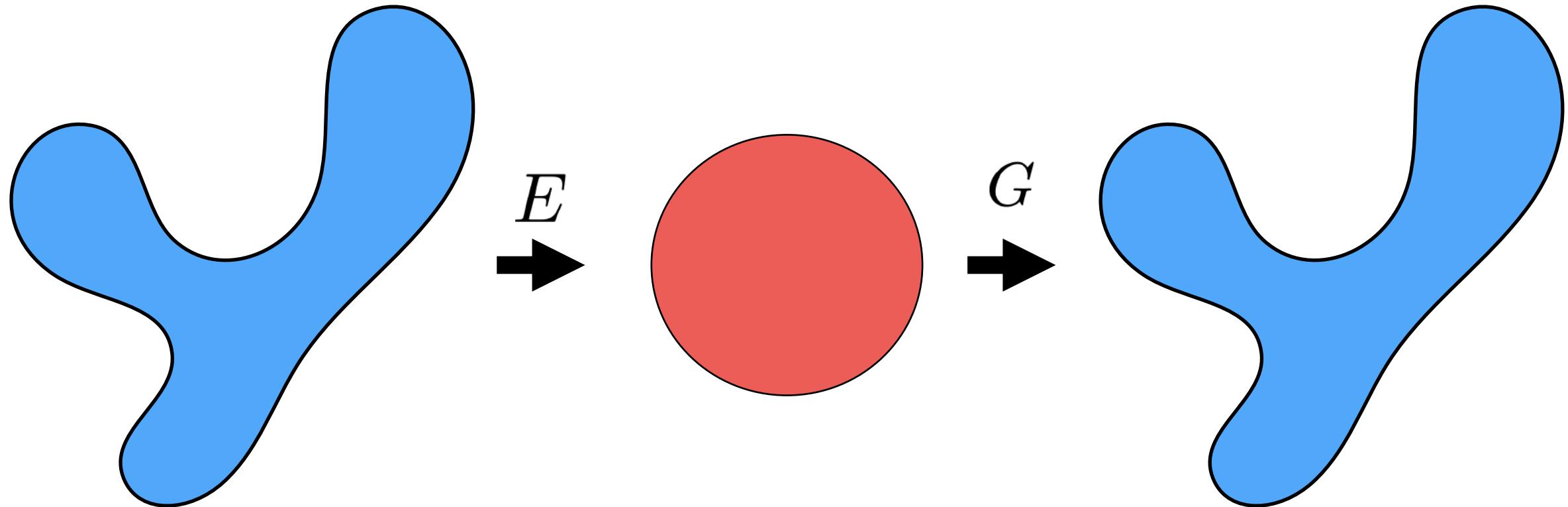


Classical Autoencoder



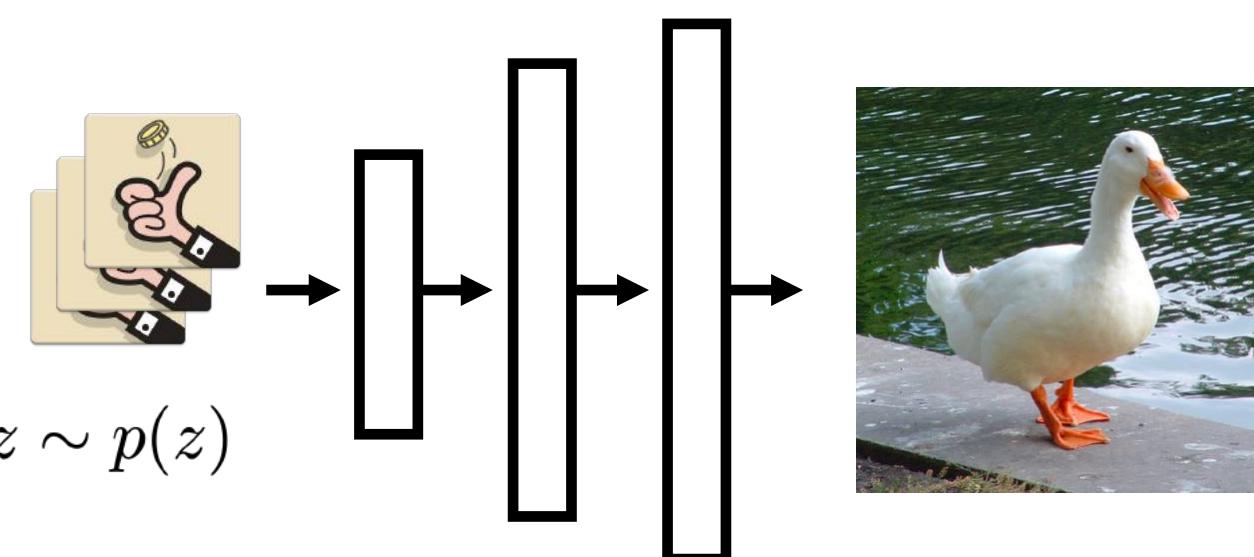
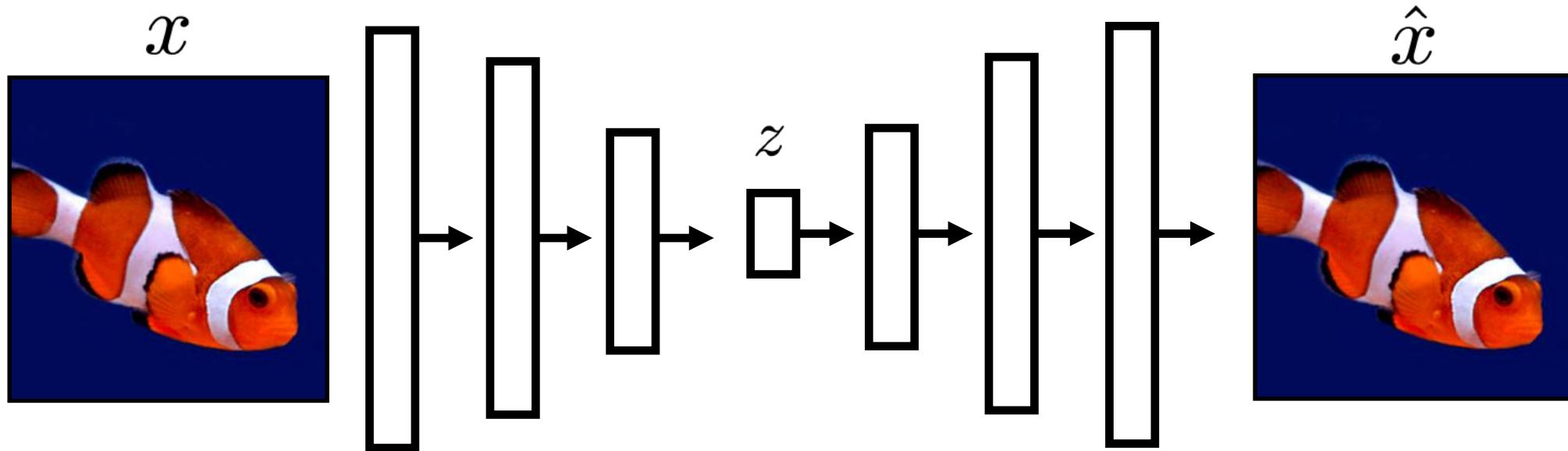
$$\arg \min_{G, E} \mathbb{E}_x [\|G(E(x)) - x\|_2^2]$$

Variational Autoencoder



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

Variational Autoencoder



Gaussian VAEs 2013

Sample $z \sim \mathcal{N}(0, I)$ and compute $y_\Phi(z)$



[Alec Radford]

Vector Quantized VAEs (VQ-VAE) 2019



VQ-VAE-2, Razavi et al., NeurIPS 2019

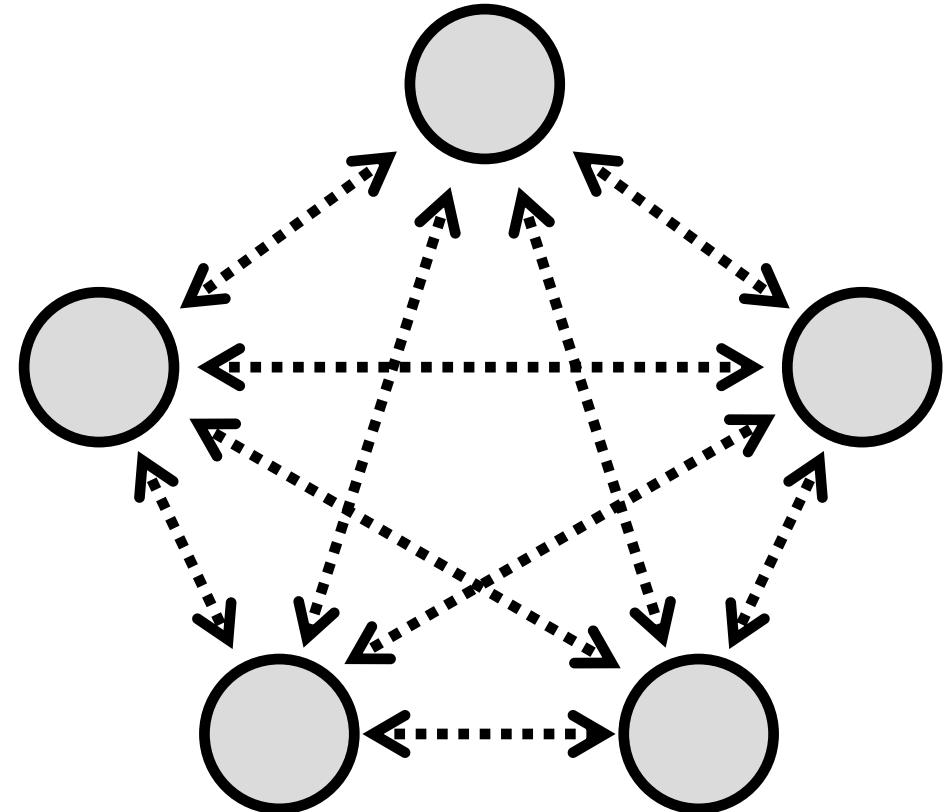
Vector Quantized VAEs (VQ-VAE) 2019



Figure 1: Class-conditional 256x256 image samples from a two-level model trained on ImageNet.

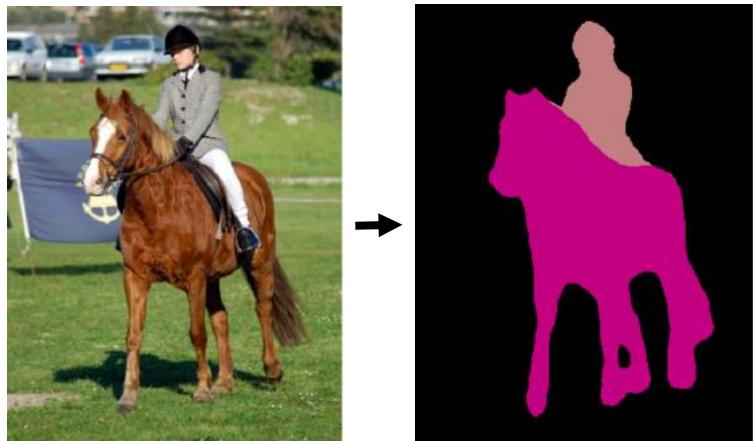
VQ-VAE-2, Razavi et al., NeurIPS 2019

Data Translation



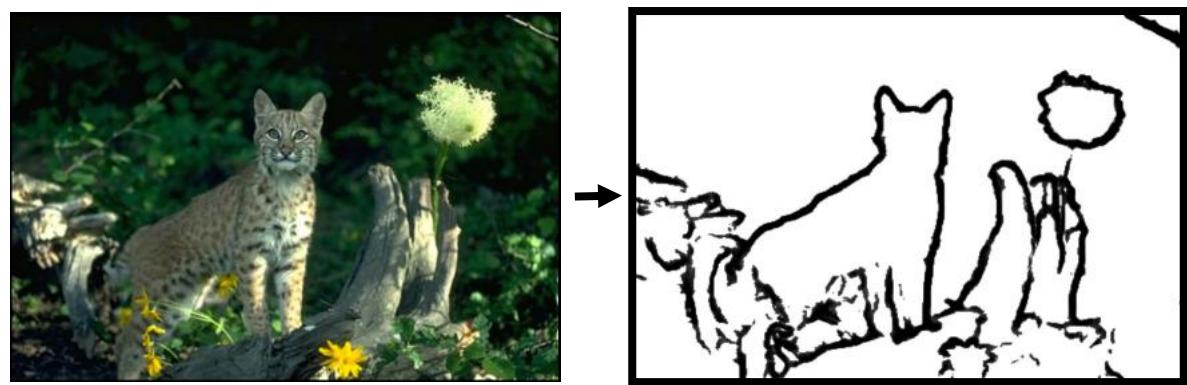
Data translation problems (“structured prediction”)

Semantic segmentation



[Long et al. 2015, ...]

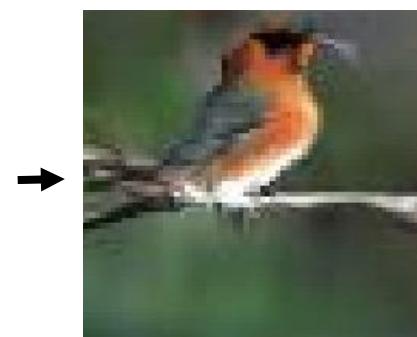
Edge detection



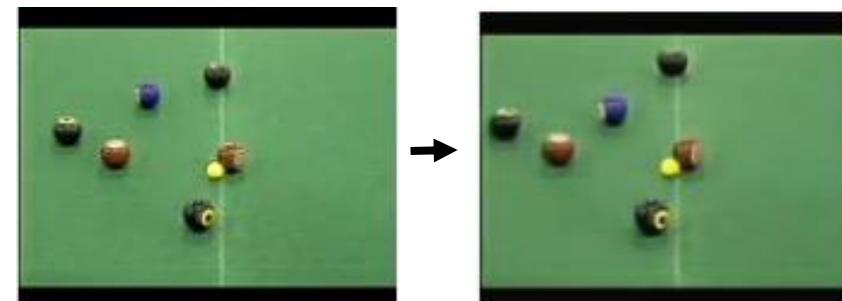
[Xie et al. 2015, ...]

Text-to-image

“this small bird has a pink
breast and crown...”
[Reed et al. 2014, ...]

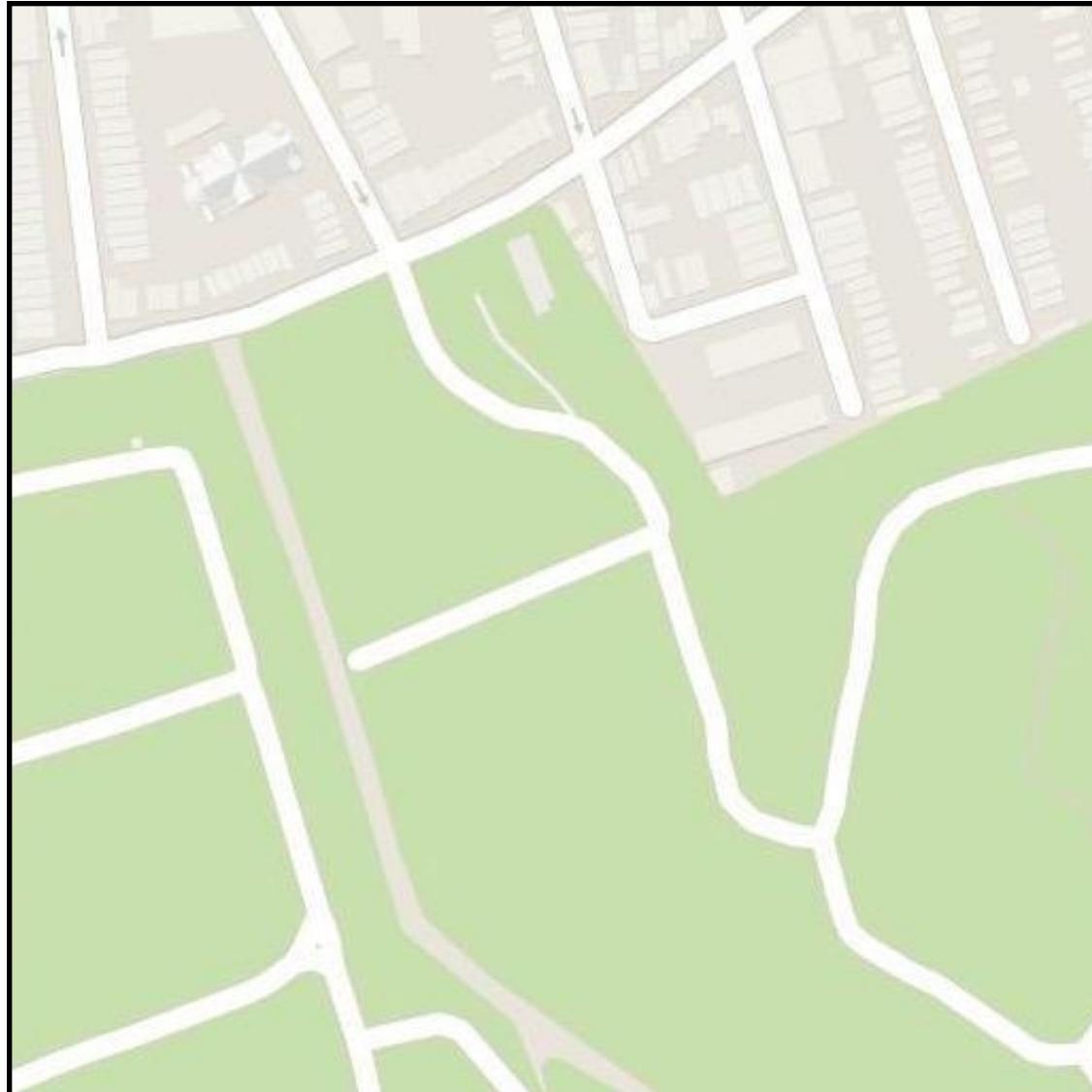


Future frame prediction



[Mathieu et al. 2016, ...]

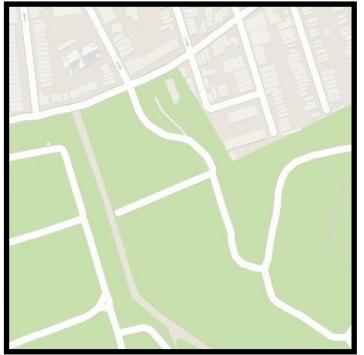
Input



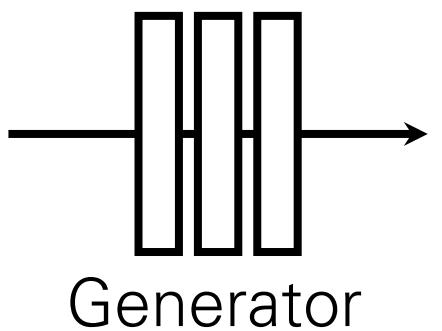
Deep net output



x

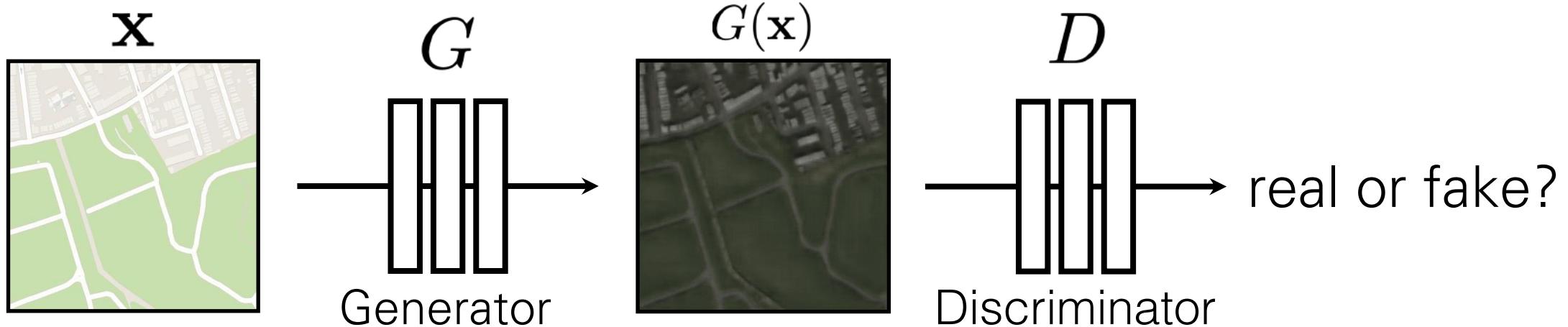


G



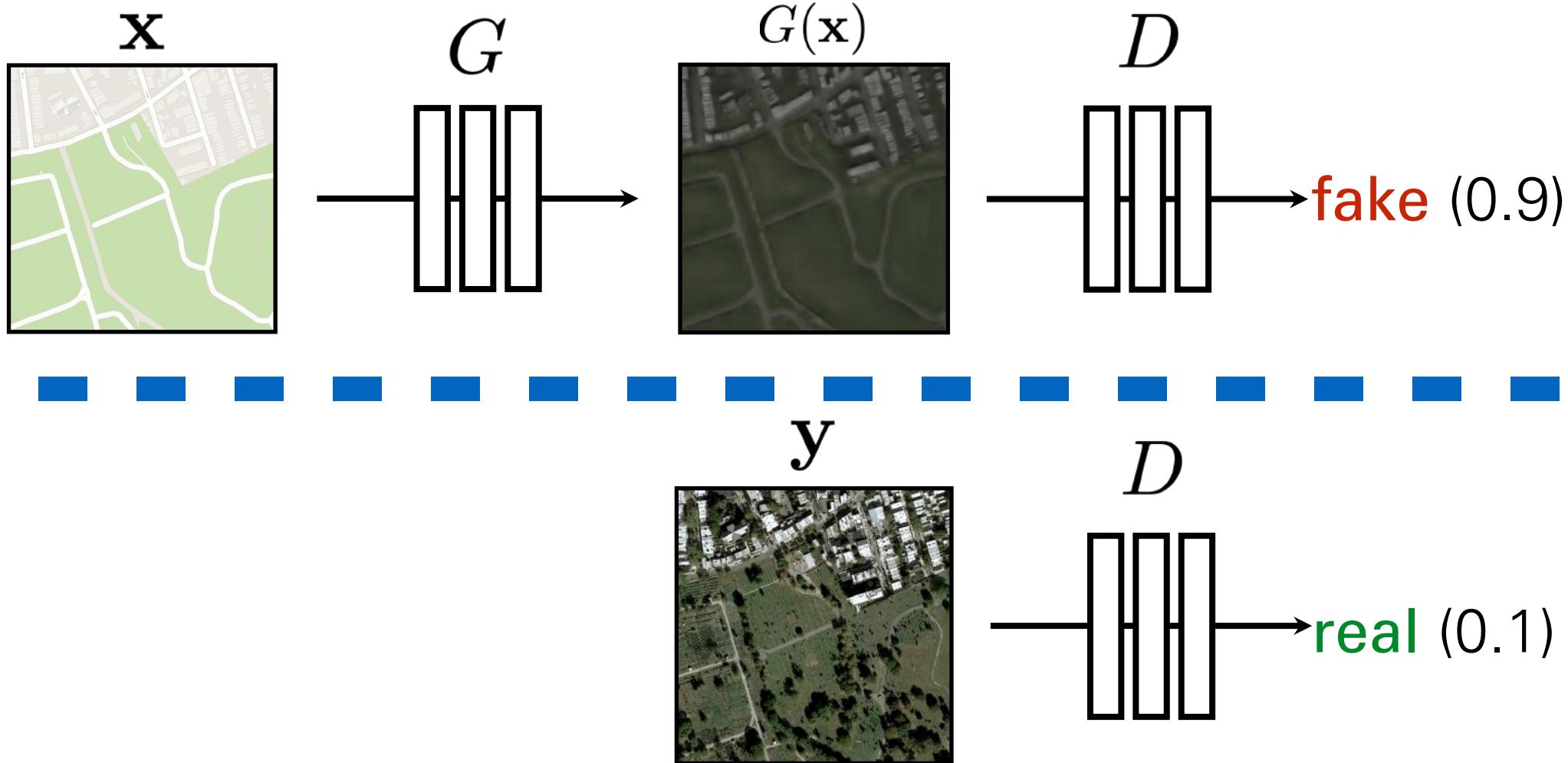
G(x)



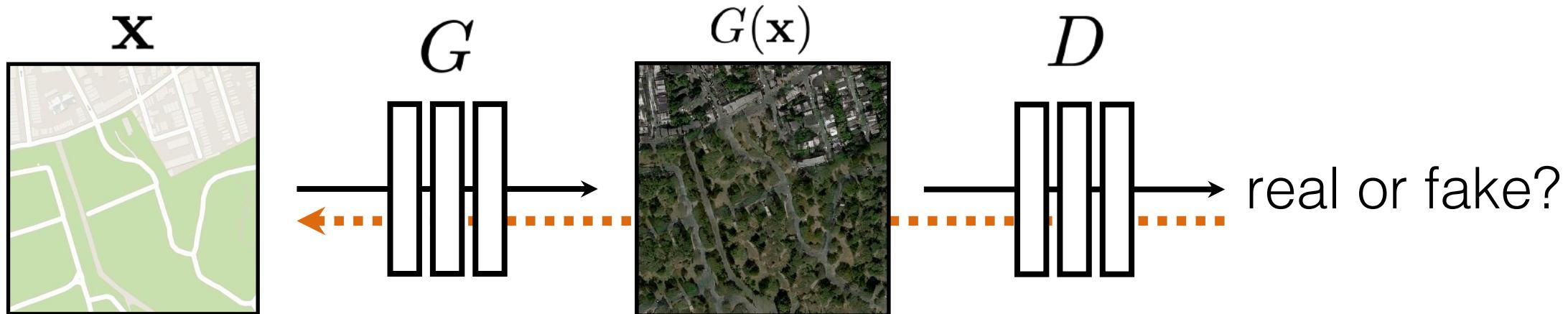


G tries to synthesize fake images that fool **D**

D tries to identify the fakes



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



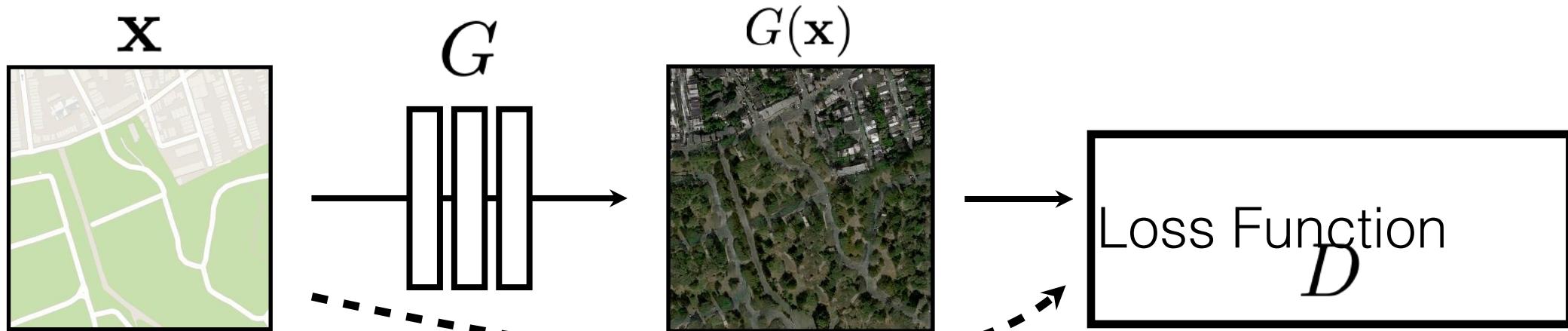
G tries to synthesize fake images that *fool* D:

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



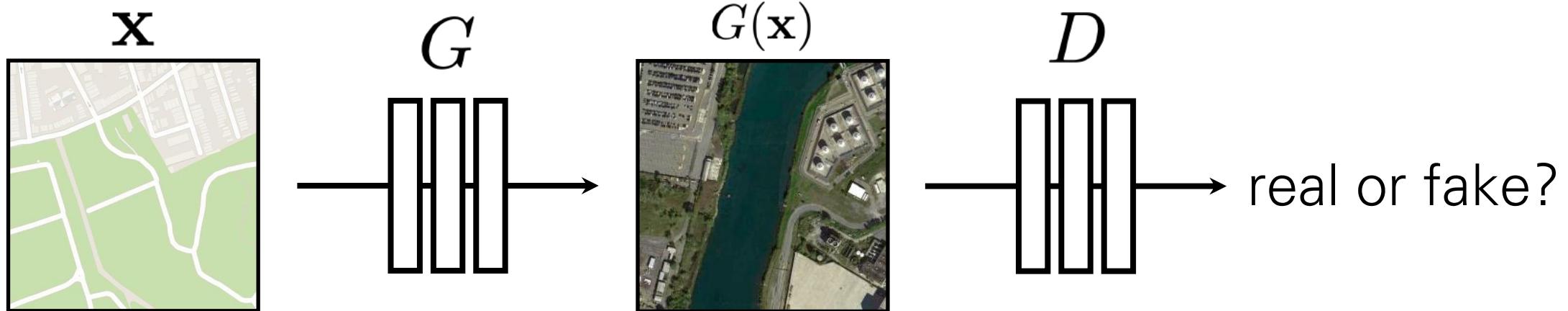
\mathbf{G} tries to synthesize fake images that *fool* the *best* \mathbf{D} :

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

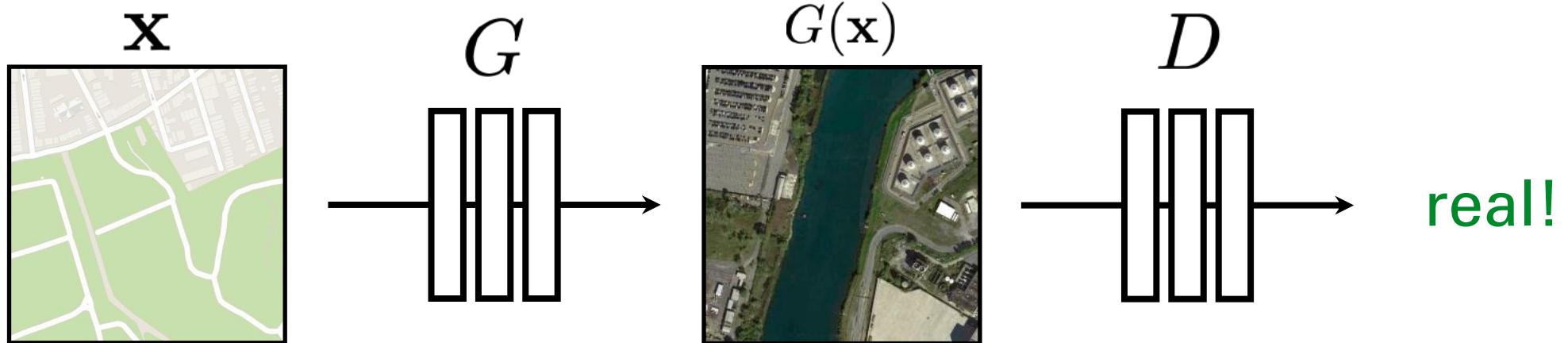


G's perspective: **D** is a loss function.

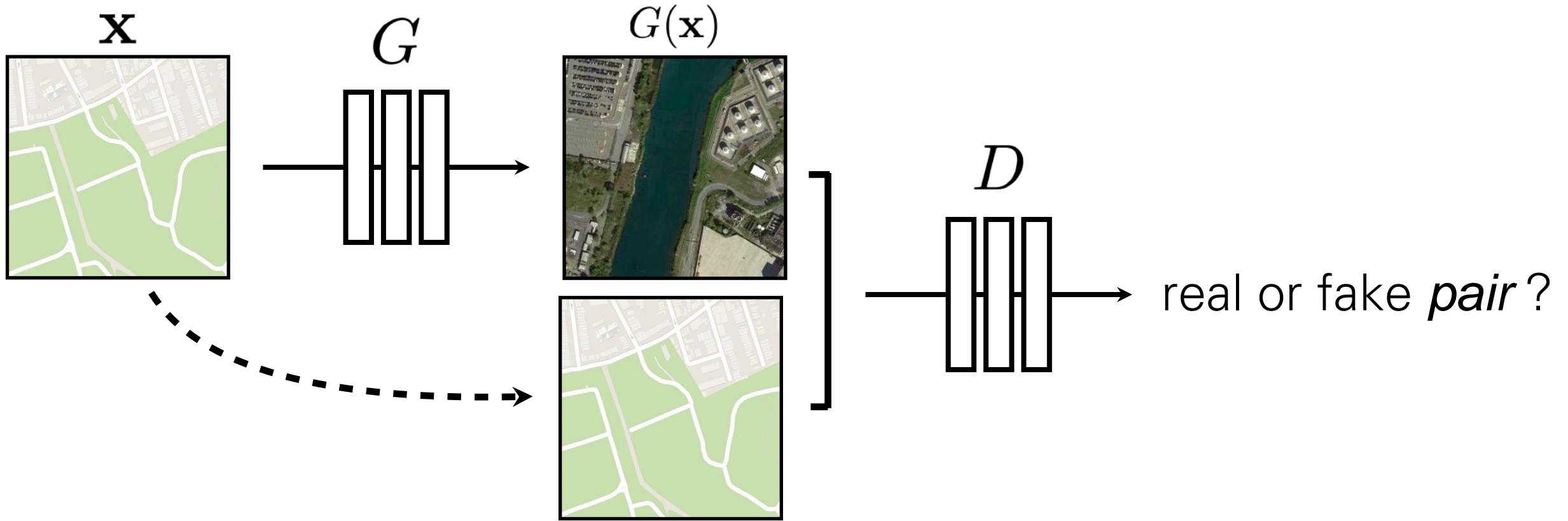
Rather than being hand-designed, it is *learned* and *highly structured*.



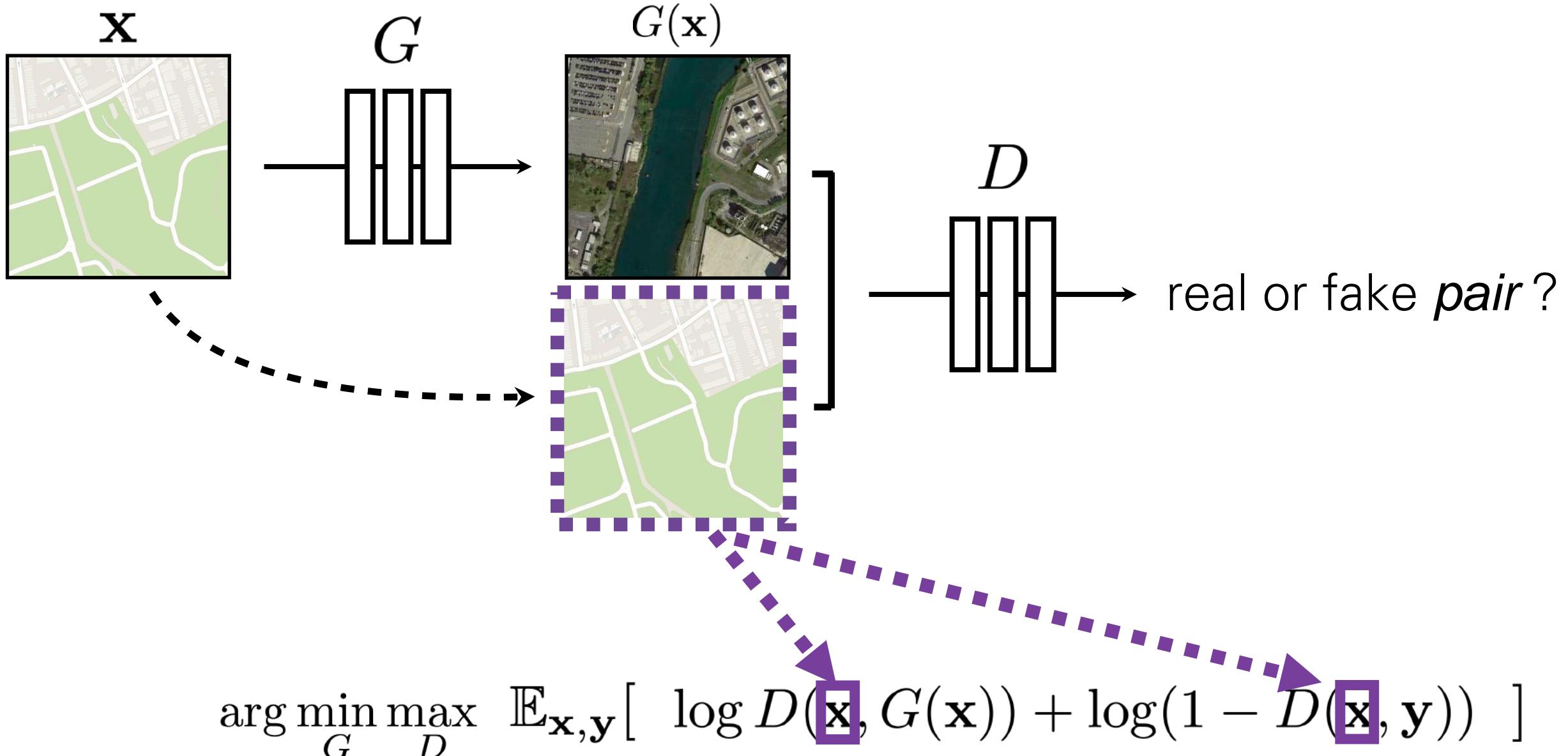
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$

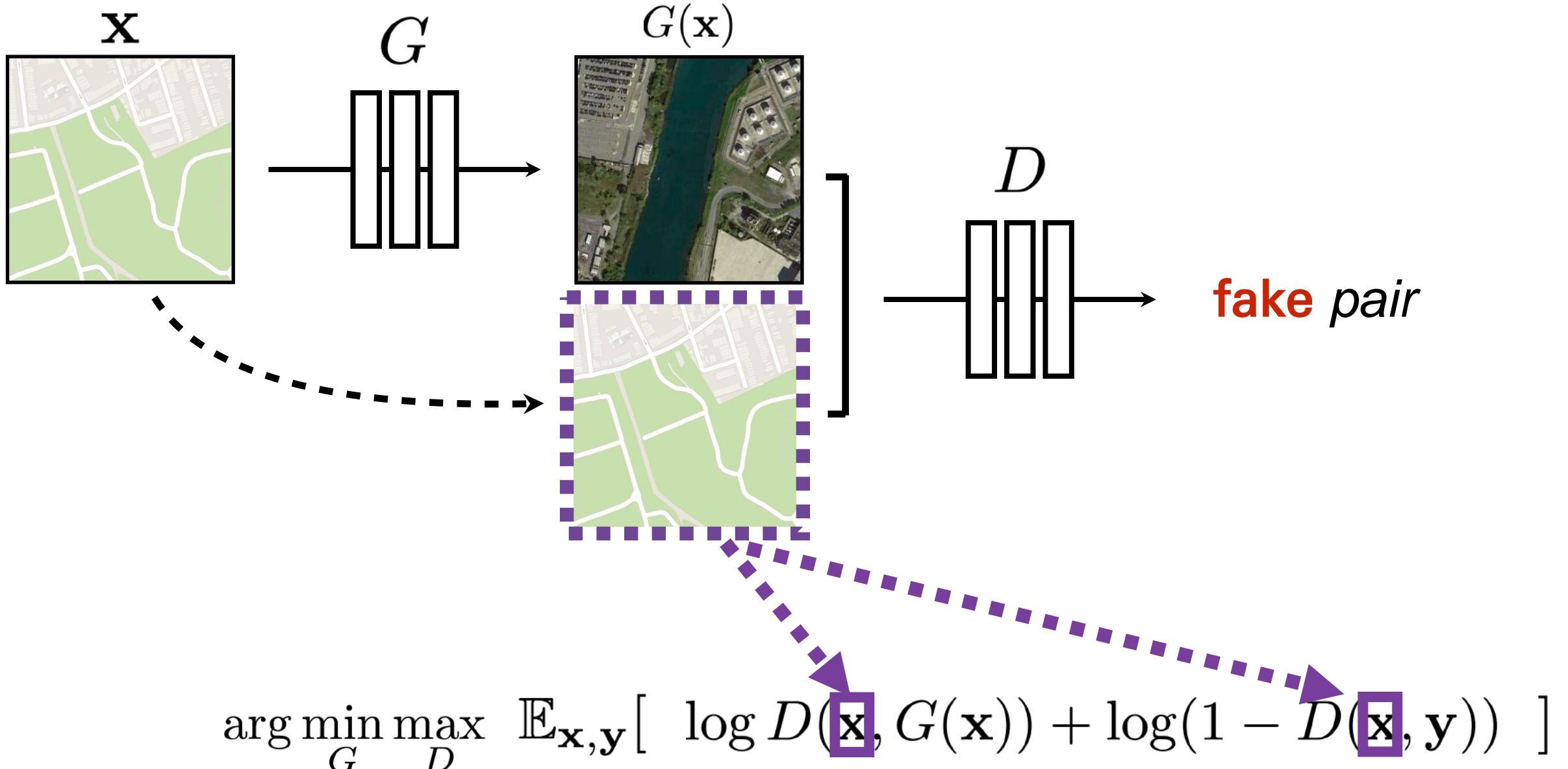


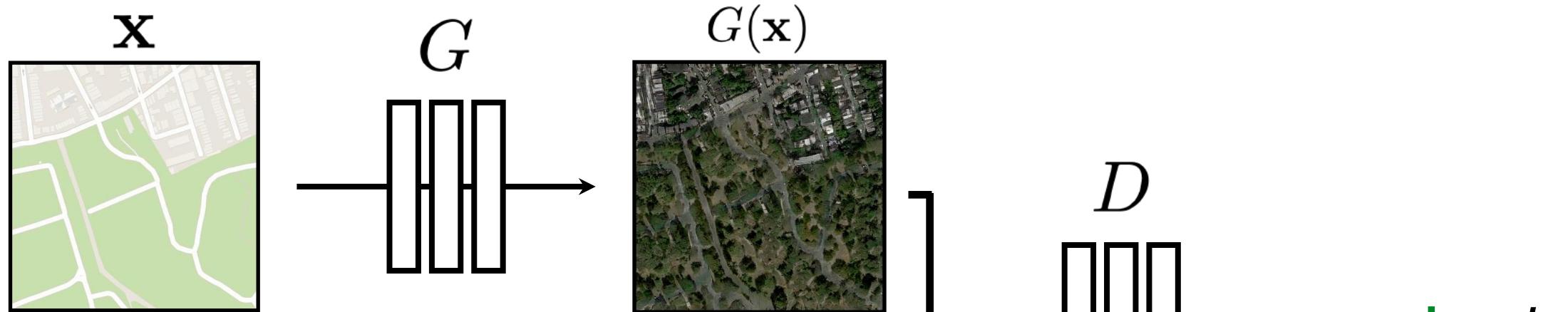
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$



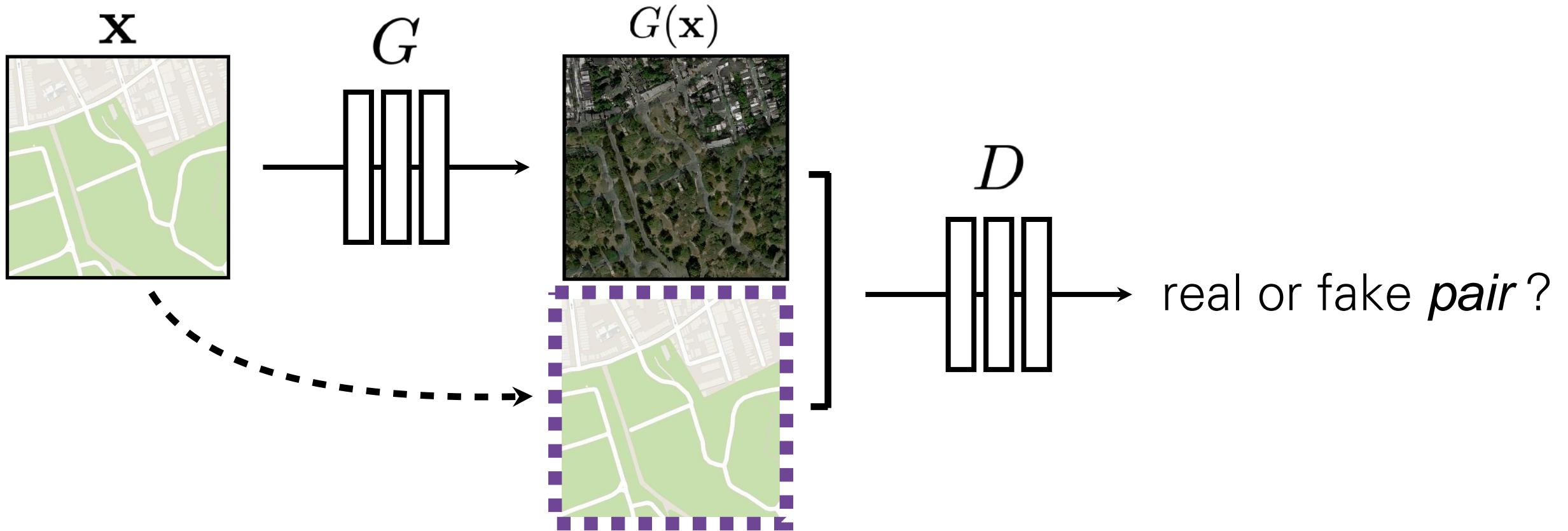
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$







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



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

Training Details: Loss function

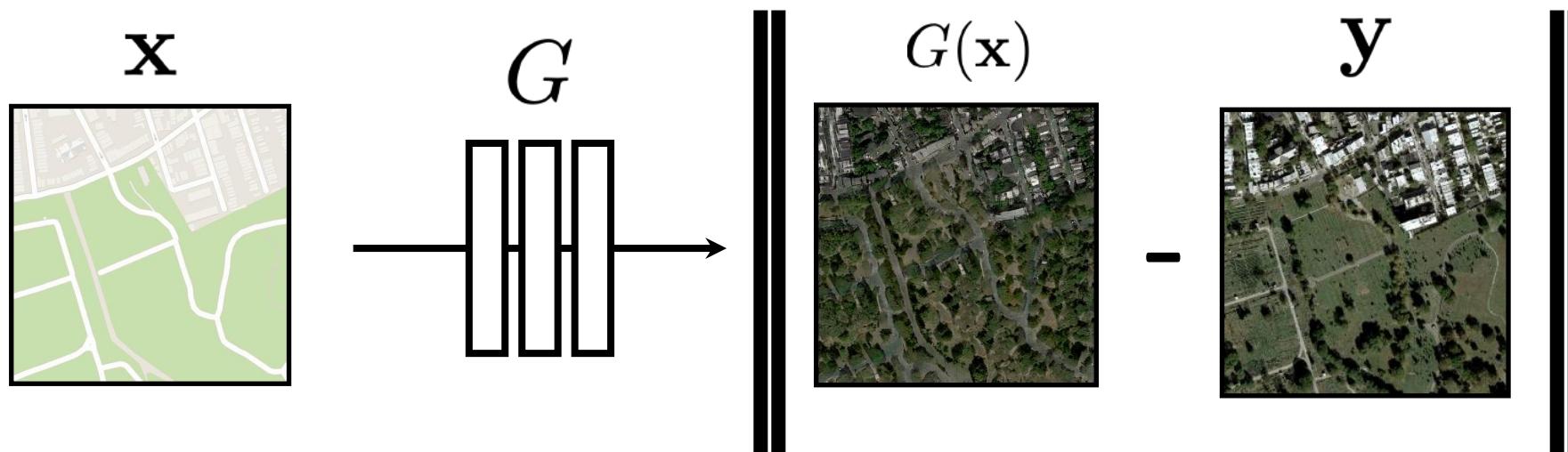
Conditional GAN

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

Training Details: Loss function

Conditional GAN

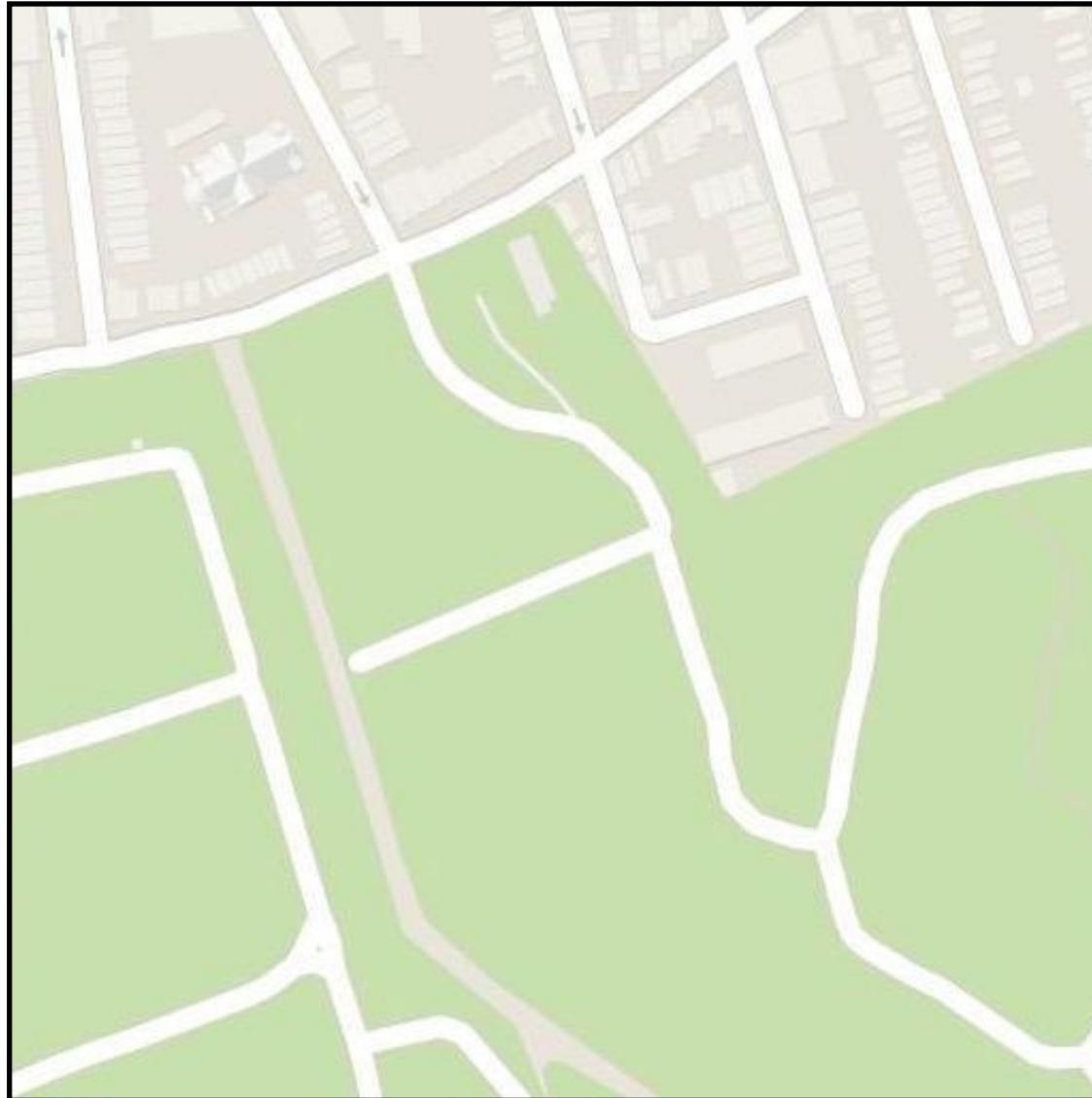
$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$



Stable training + fast convergence

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

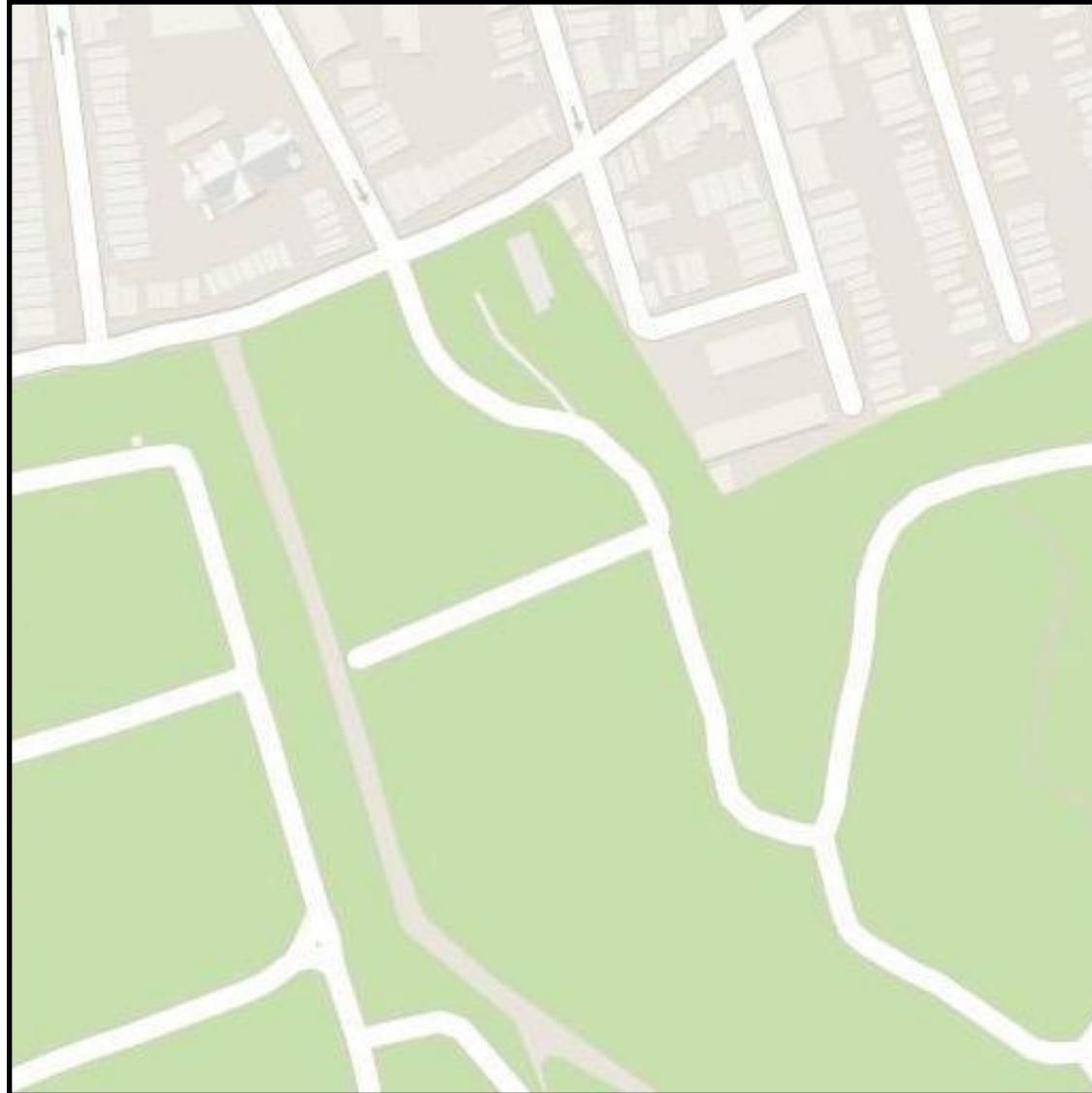
Input



Unstructured prediction (L1)



Input



Structured Prediction (cGAN)



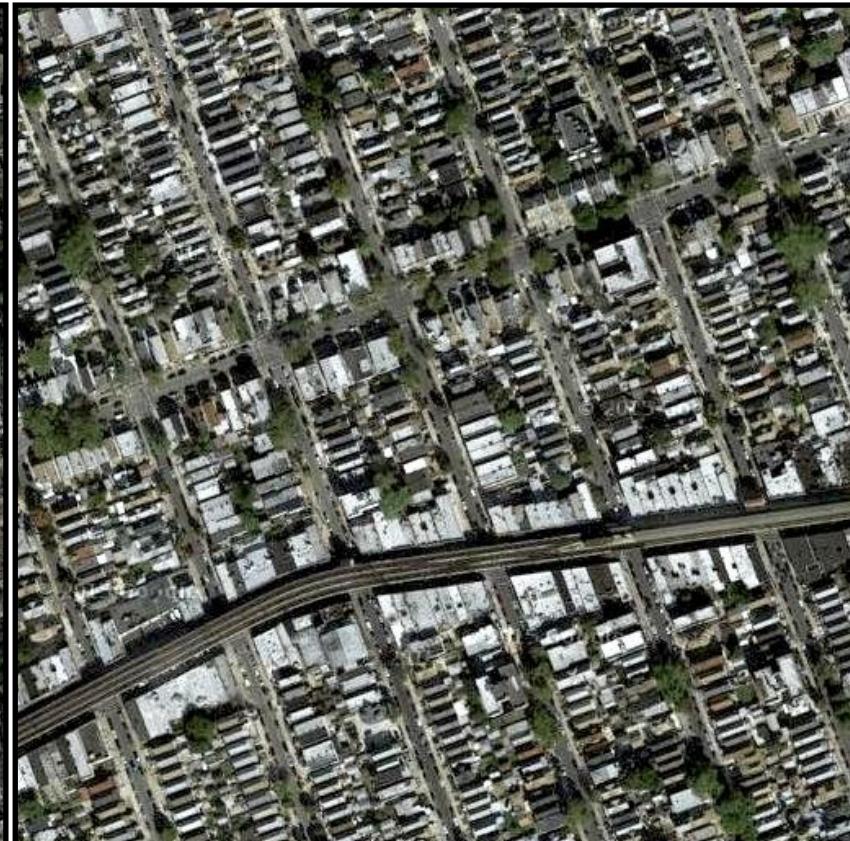
Input



Output



Groundtruth



Data from
[\[maps.google.com\]](https://maps.google.com)



Input

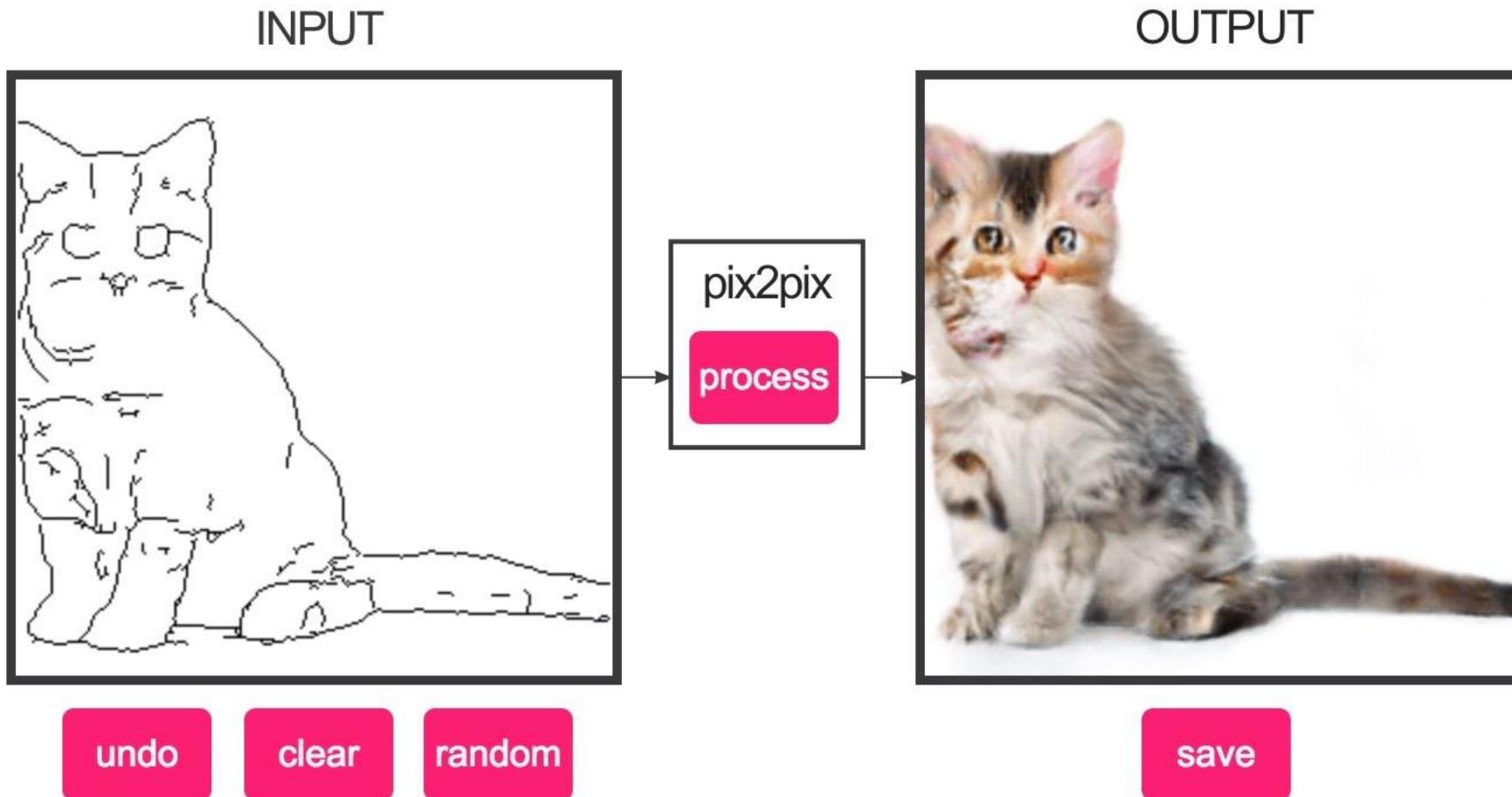
Output

Groundtruth



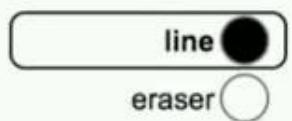
Data from [maps.google.com]

#edges2cats [Chris Hesse]

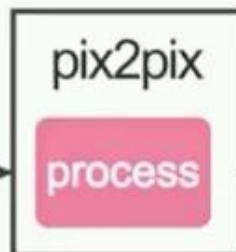
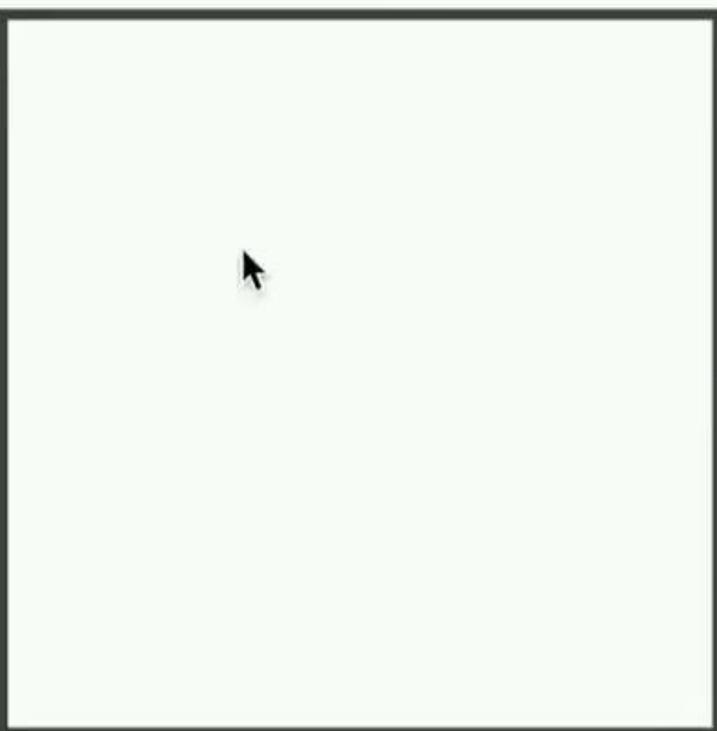


edges2cats

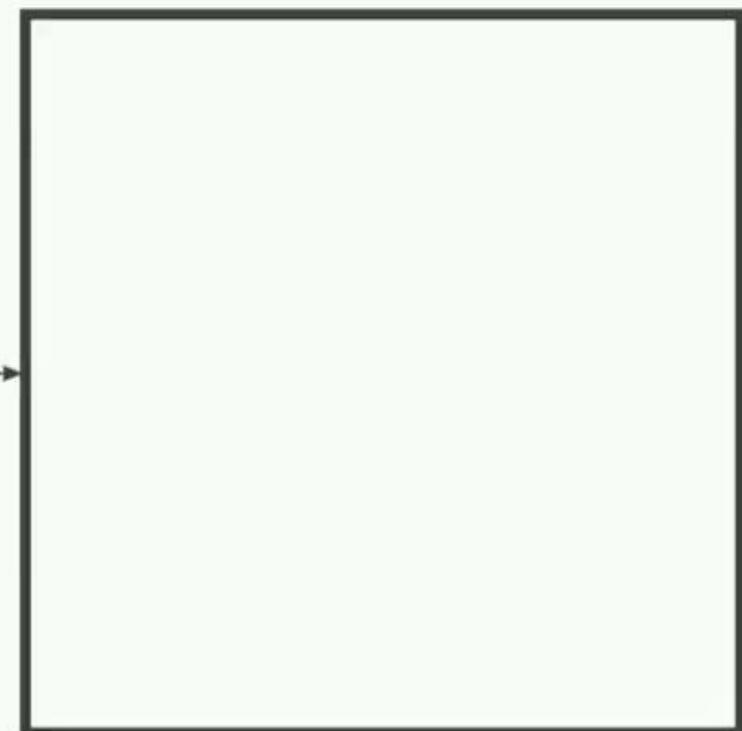
TOOL



INPUT



OUTPUT



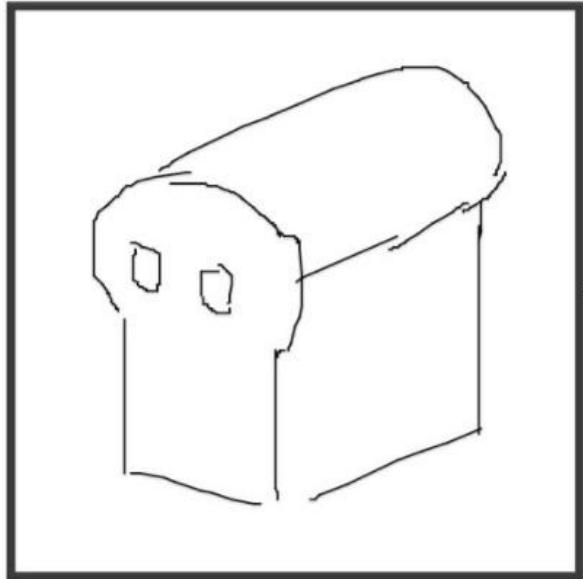
undo

clear

random

save

INPUT



OUTPUT

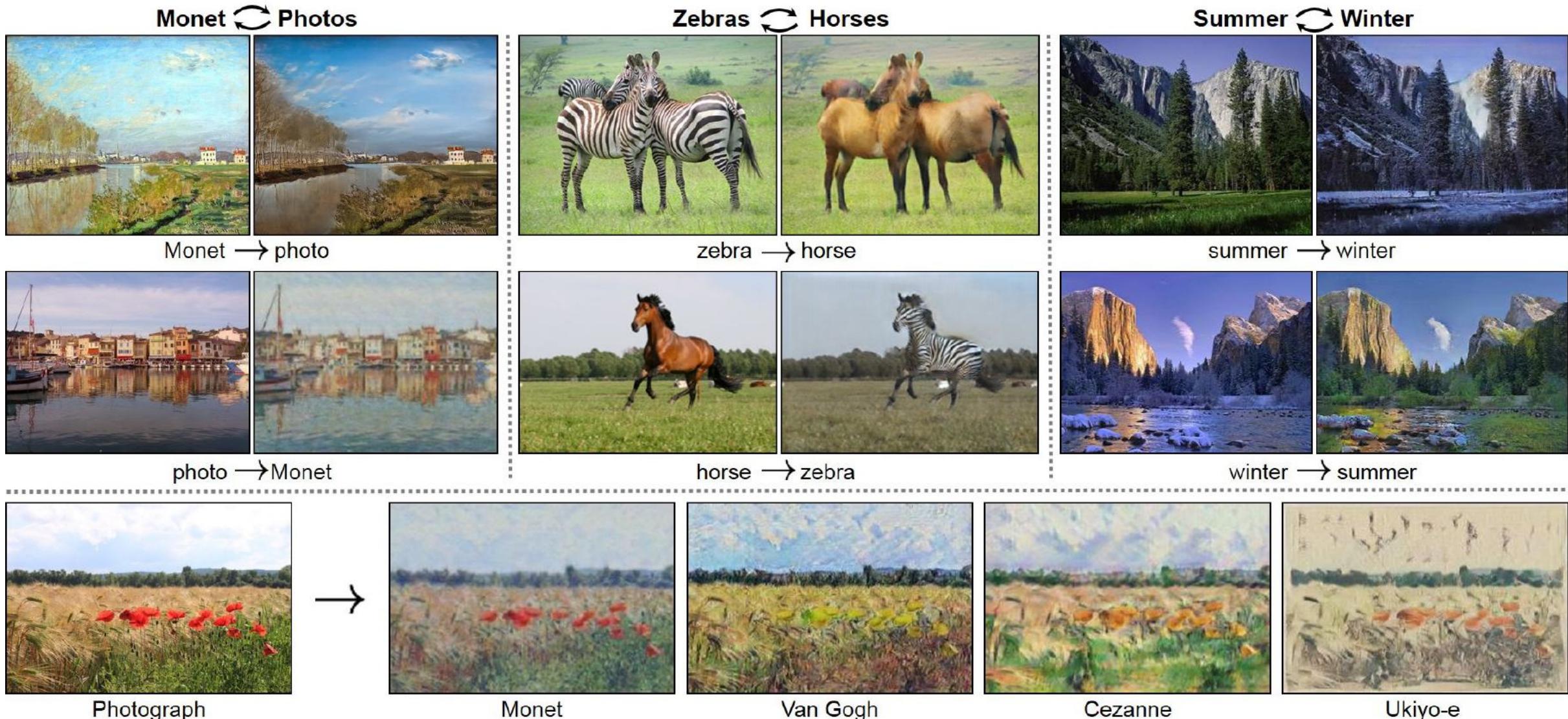


Ivy Tasi @ivymyt



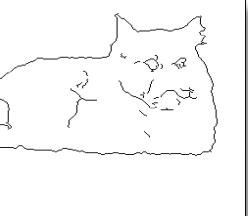
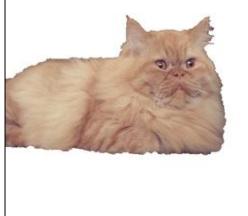
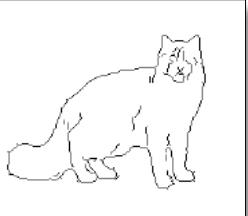
Vitaly Vidmirov @vvid

CycleGAN: Pix2Pix w/o input-output pairs

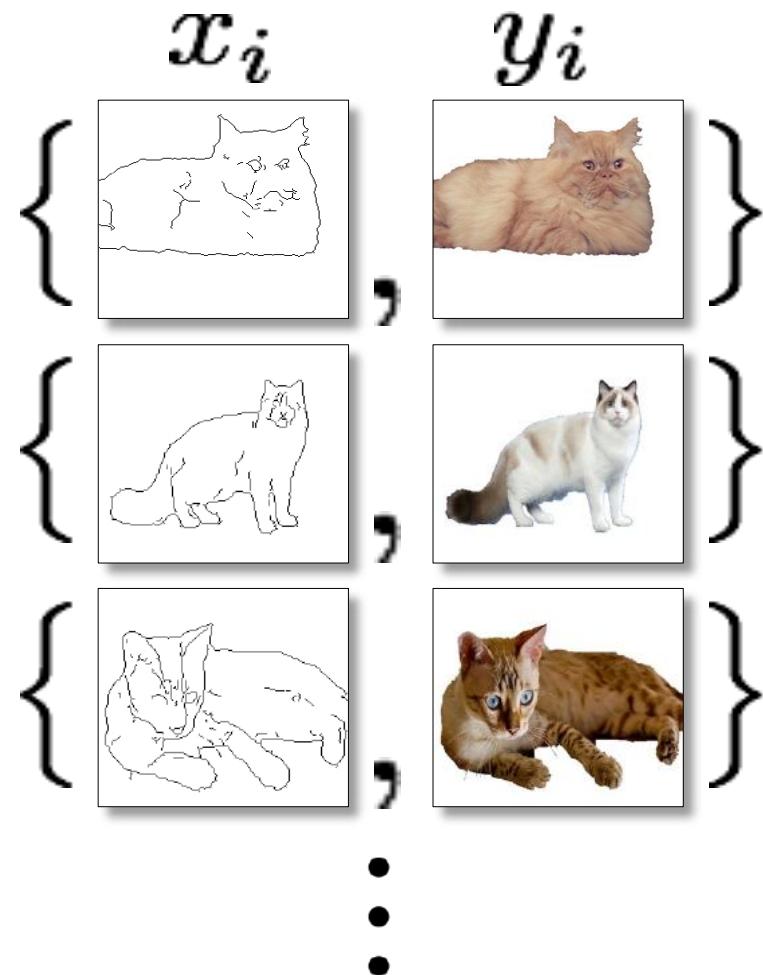


(Zhu et al. 2017)

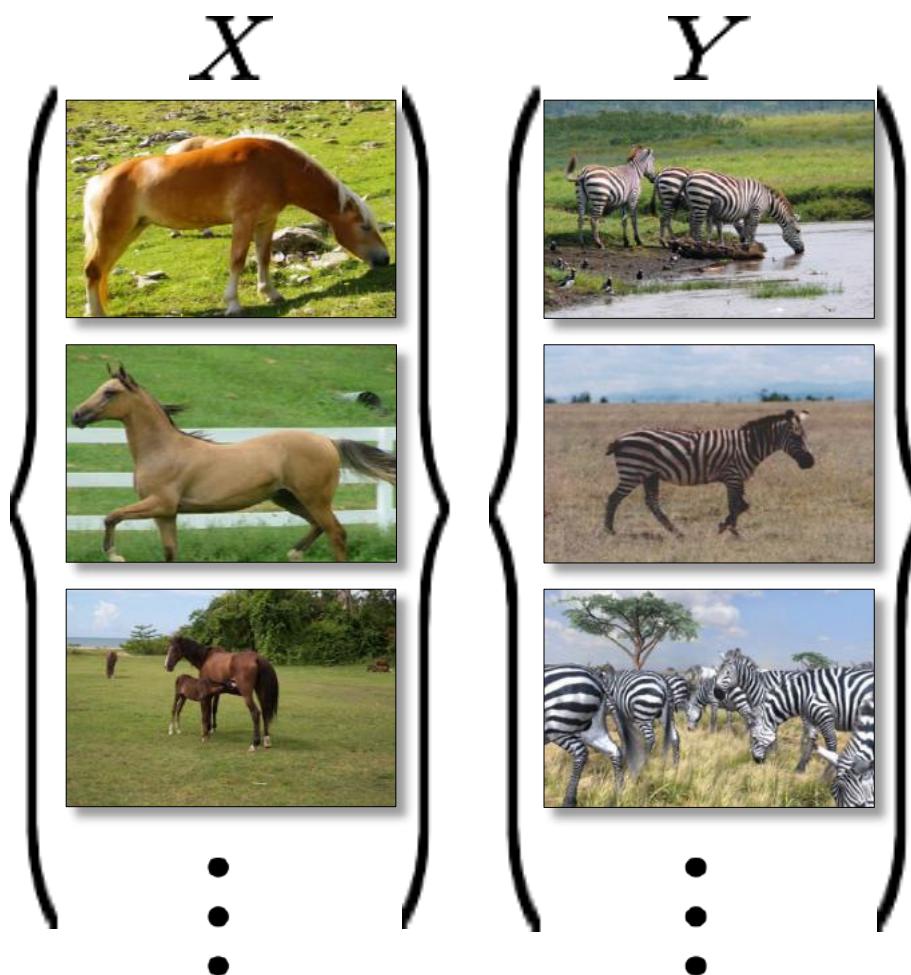
Paired data

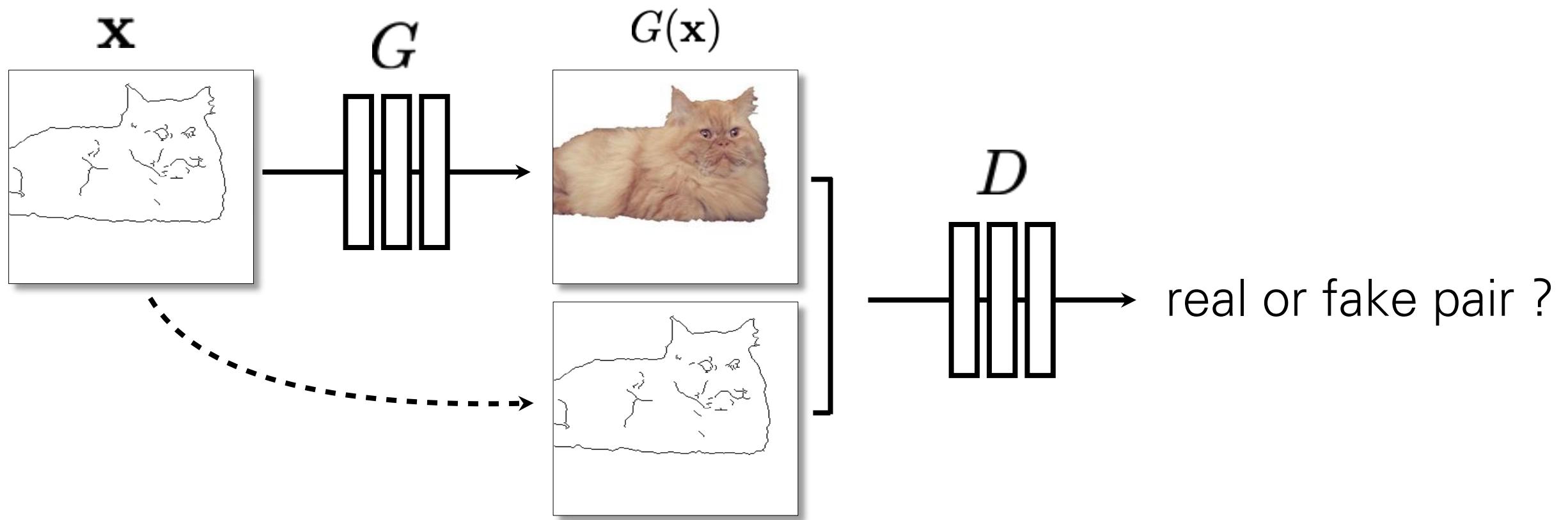
| x_i | y_i |
|--|---|
| {  , } | {  } |
| {  , } | {  } |
| {  , } | {  } |
| ⋮ | |

Paired data

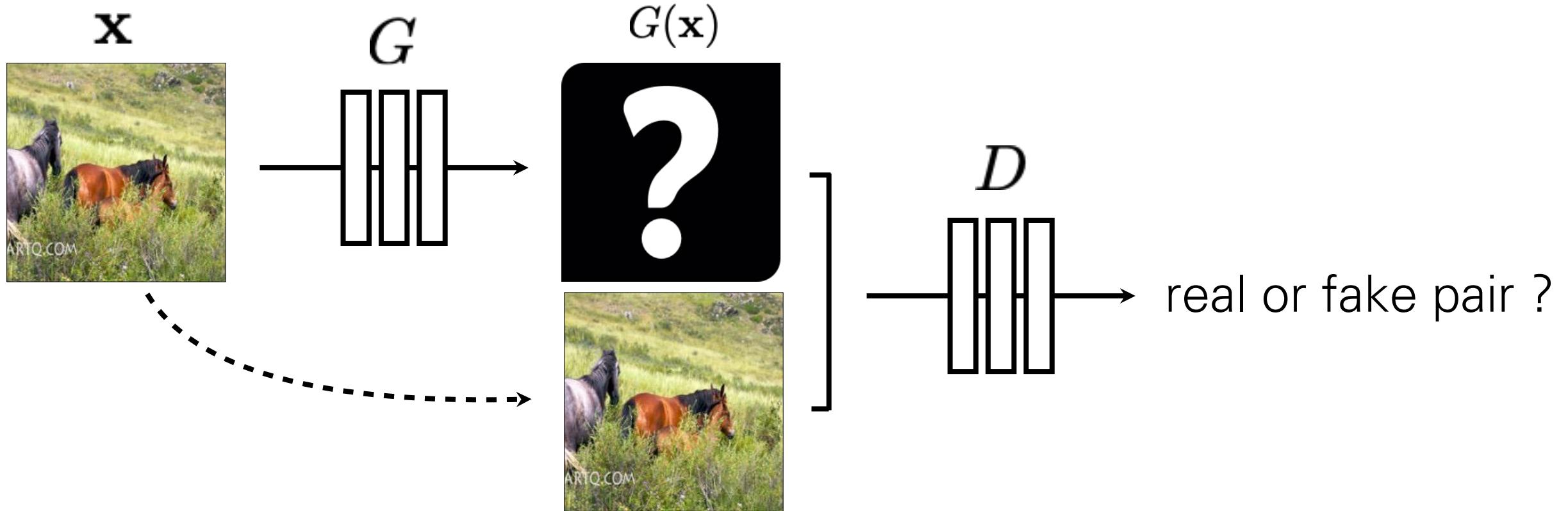


Unpaired data



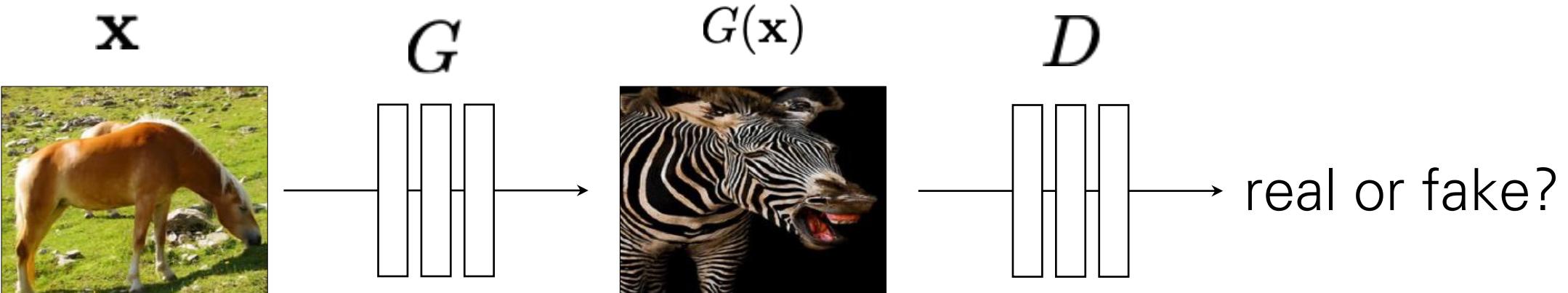


$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))]$$



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

No input-output pairs!



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

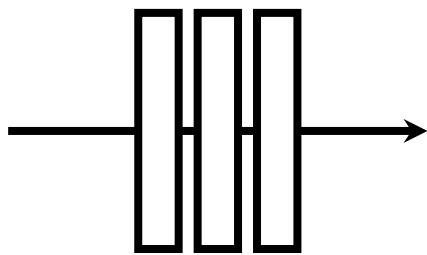
Usually loss functions check if output matches a target instance

GAN loss checks if output is part of an admissible set

x



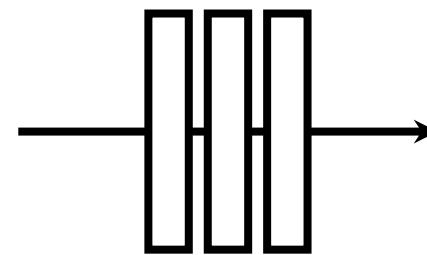
G



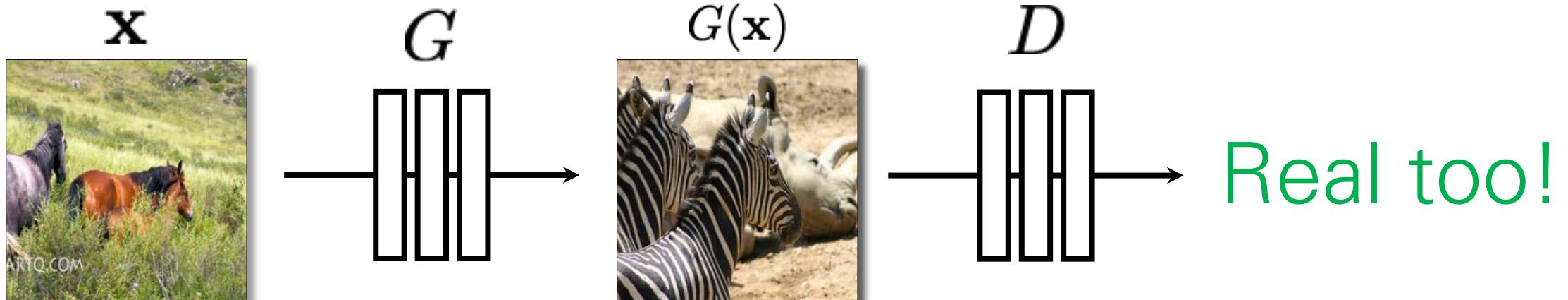
G(x)



D

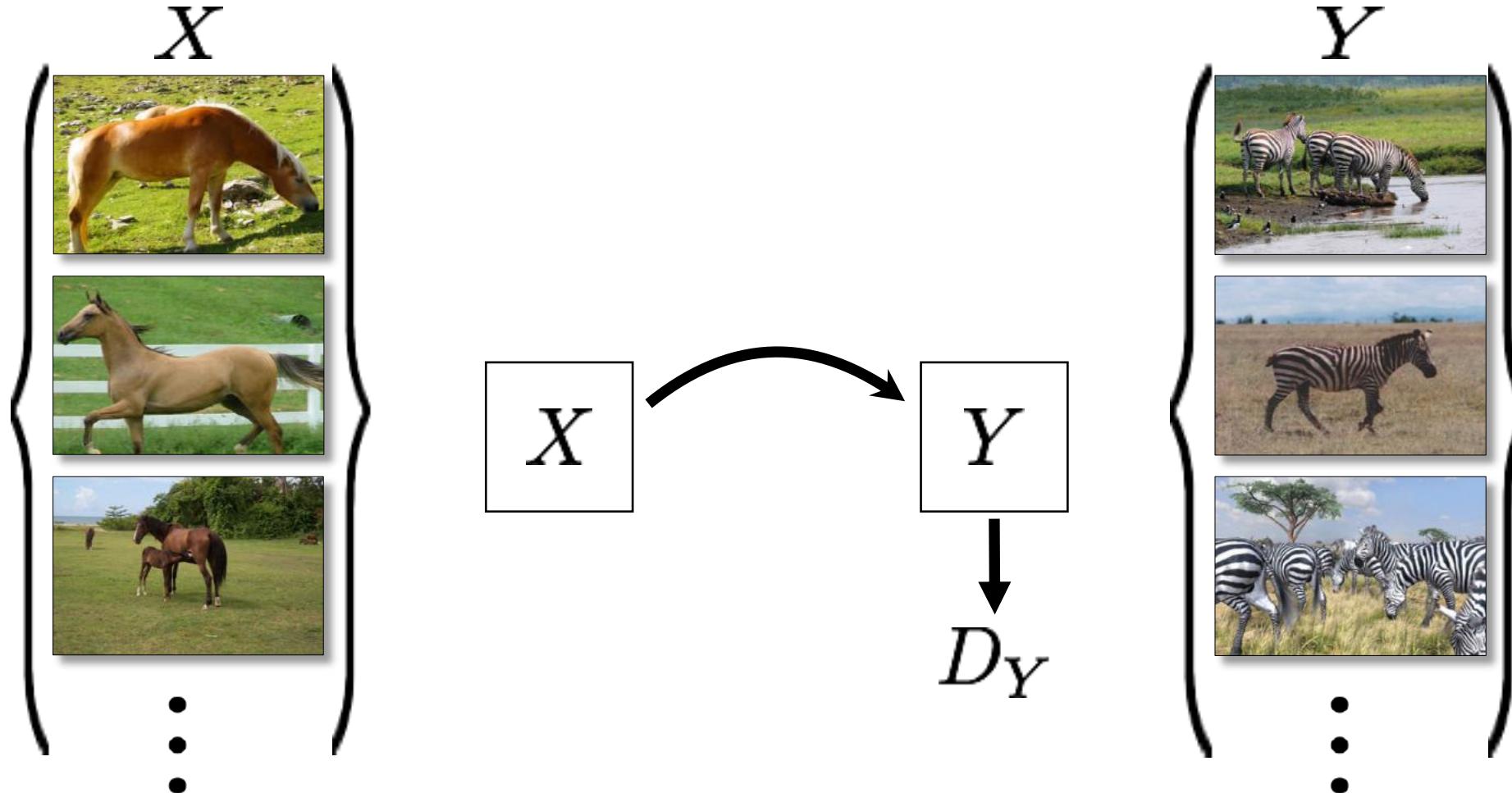


Real!



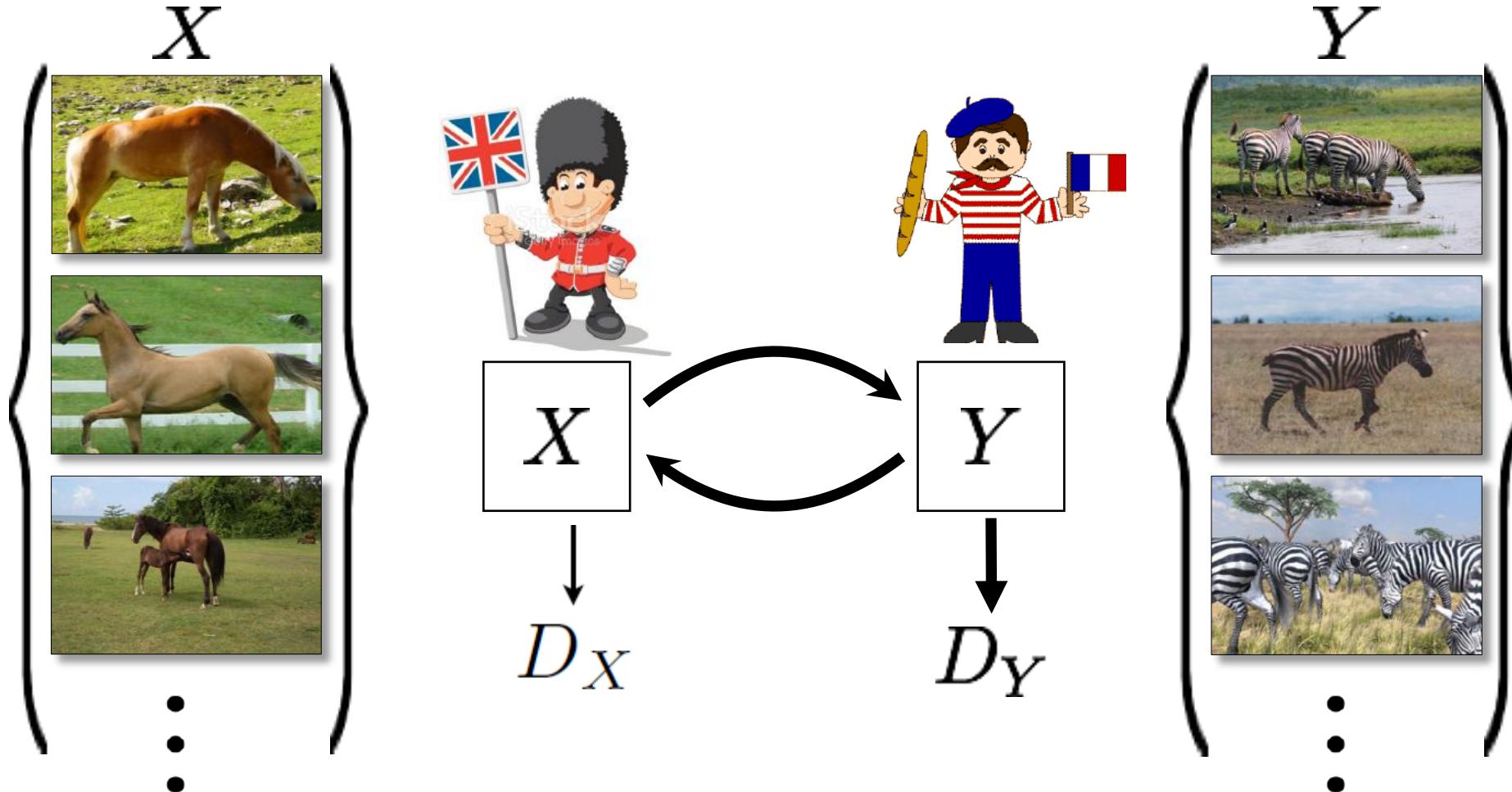
Nothing to force output to correspond to input

Cycle-Consistent Adversarial Networks

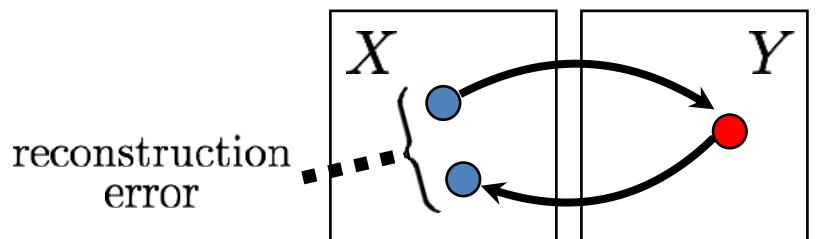
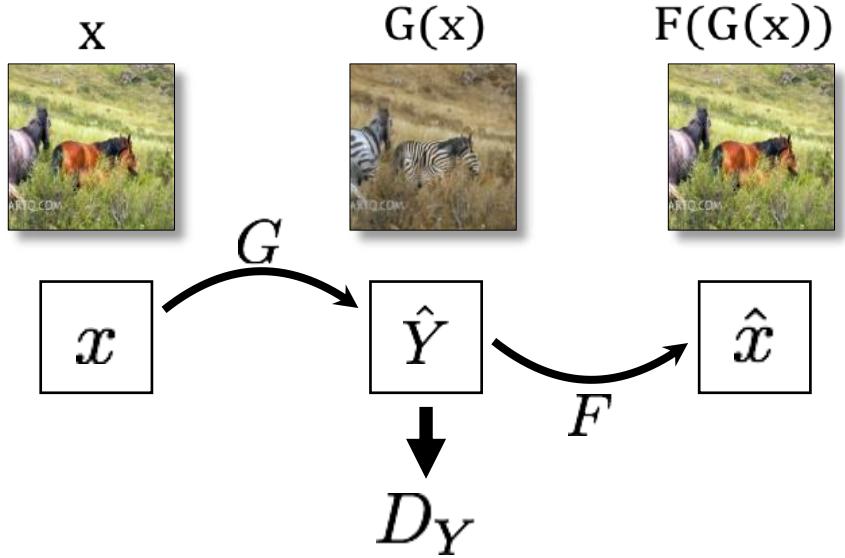


[Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

Cycle-Consistent Adversarial Networks

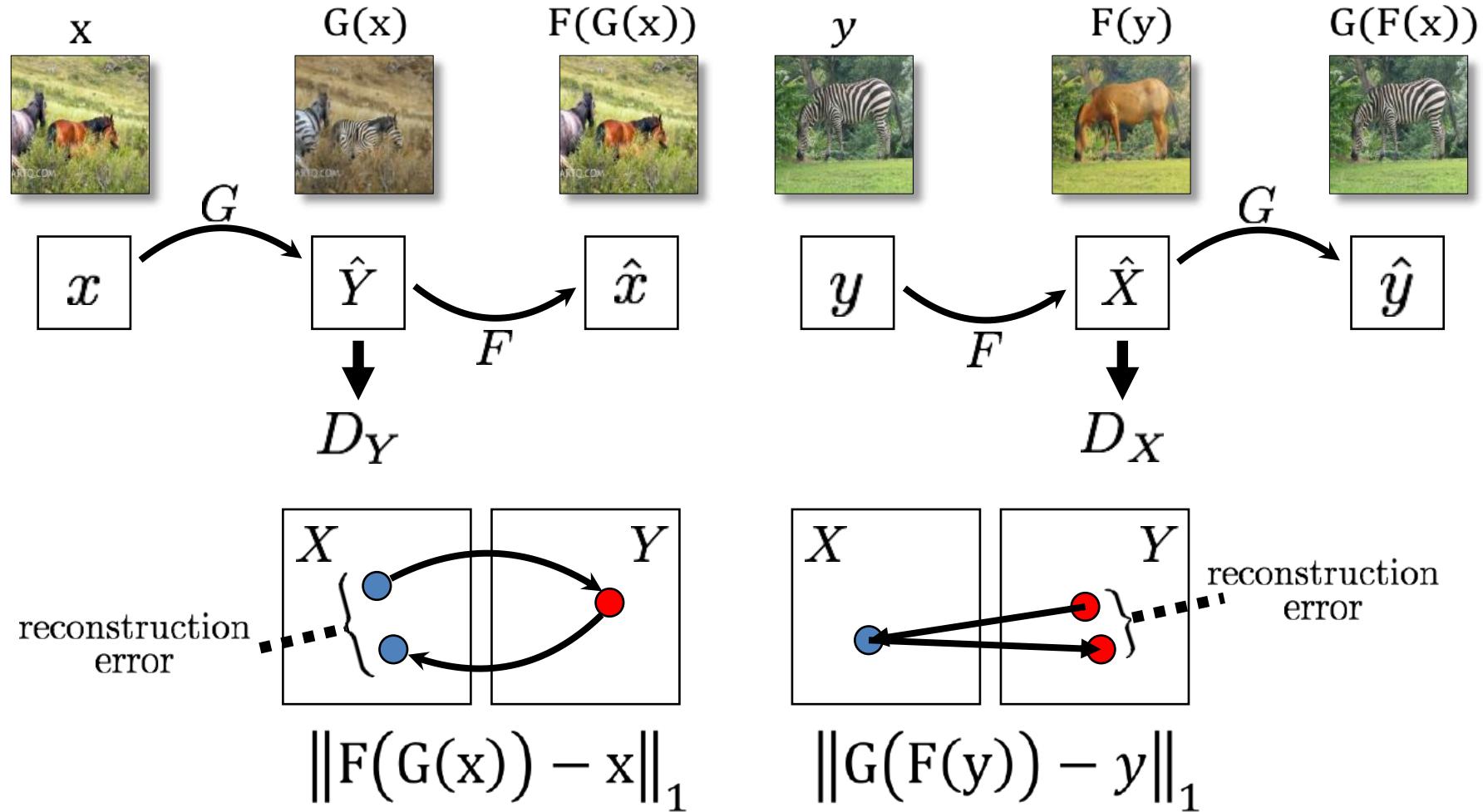


Cycle Consistency Loss



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

Cycle Consistency Loss





Input



Monet



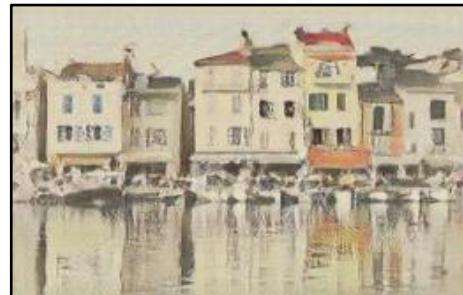
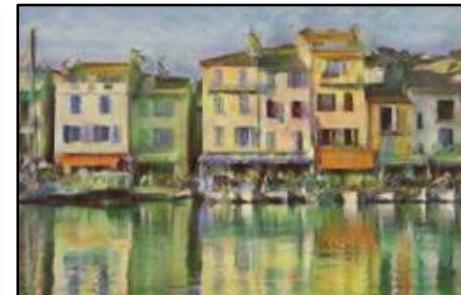
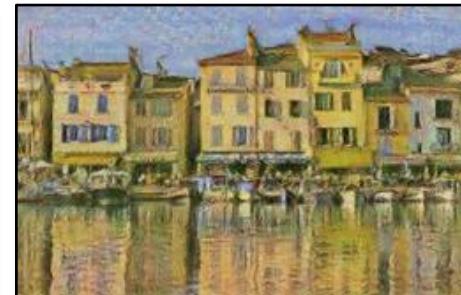
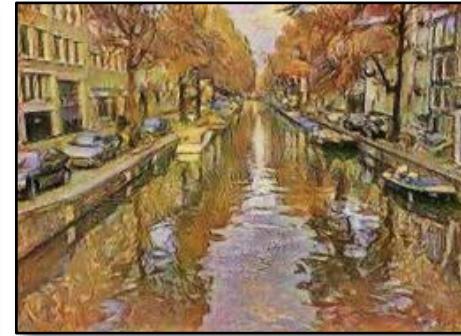
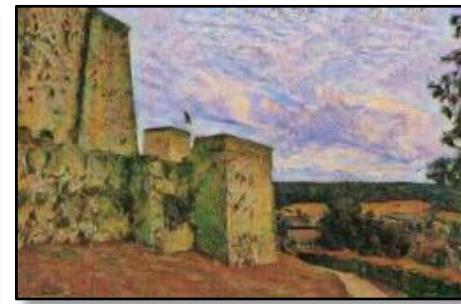
Van Gogh



Cezanne



Ukiyo-e



Monet's paintings → photos

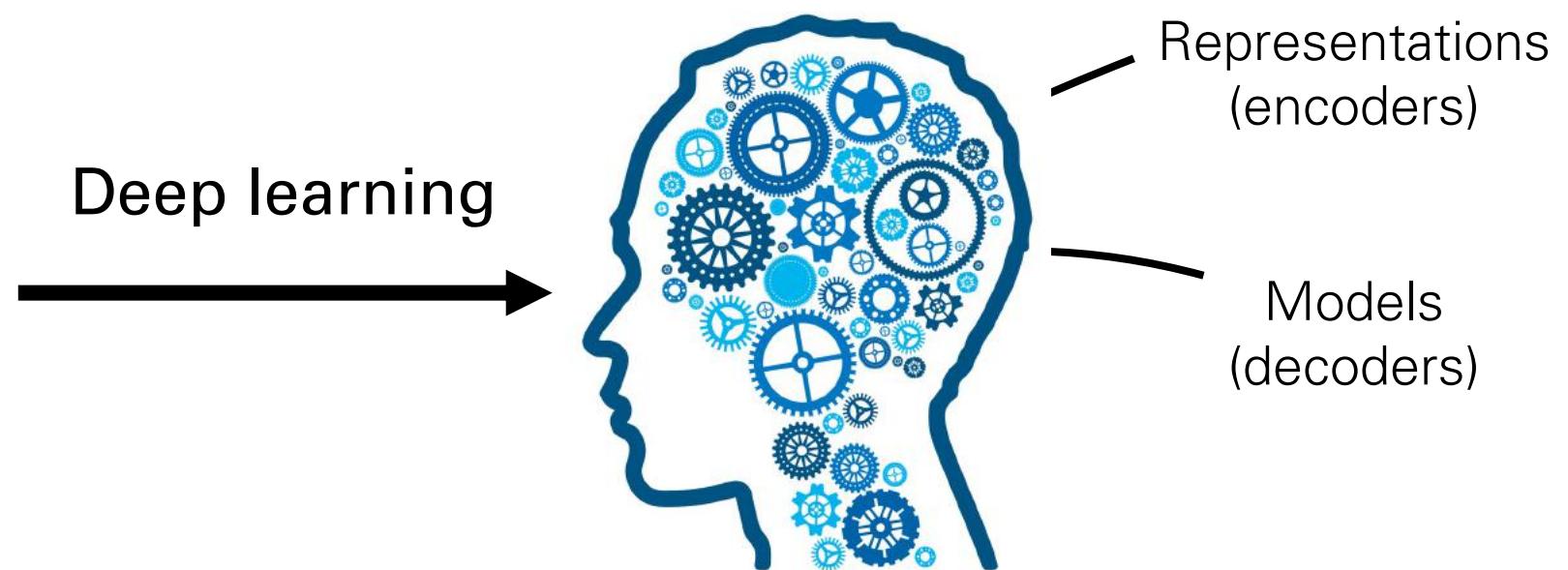


Monet's paintings → photos



Leveraging pretrained
models for efficient
data translation

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



Foundation models

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

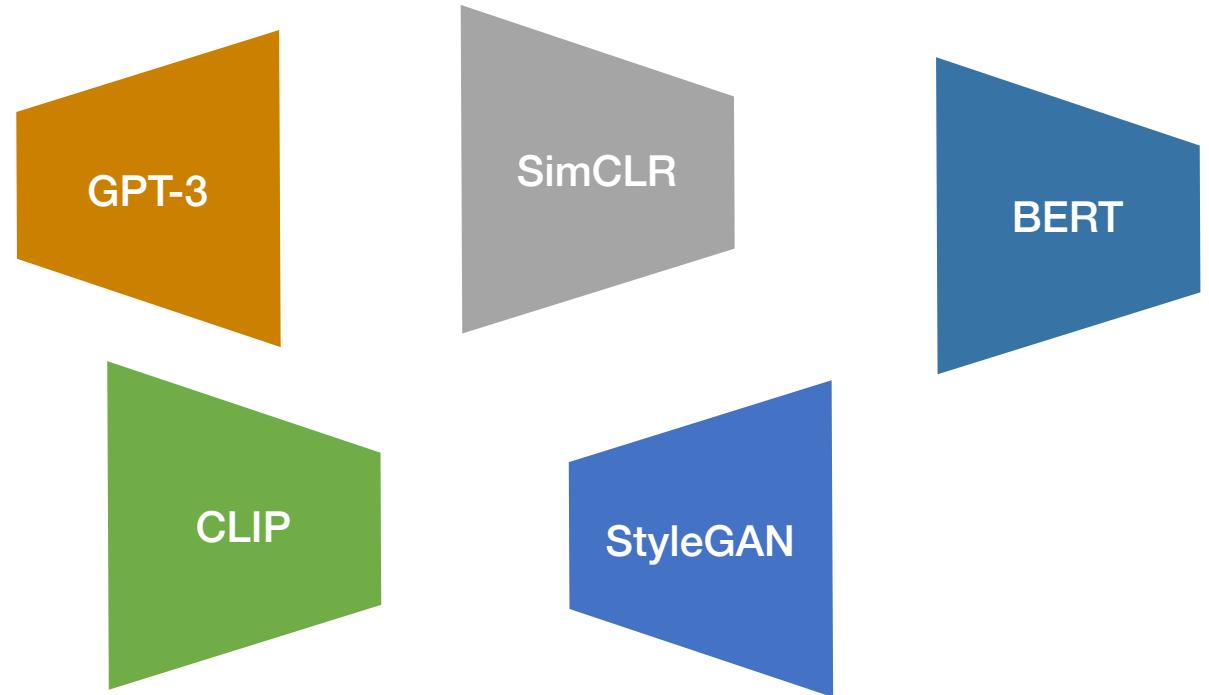
“If I have seen further
it is by standing on the
shoulders of Giants”

— Newton

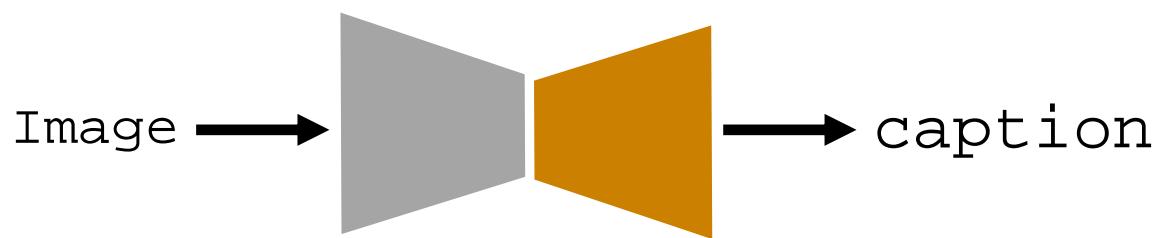


[Blind Orion Searching for the Rising Sun by Nicolas Poussin, 1658]

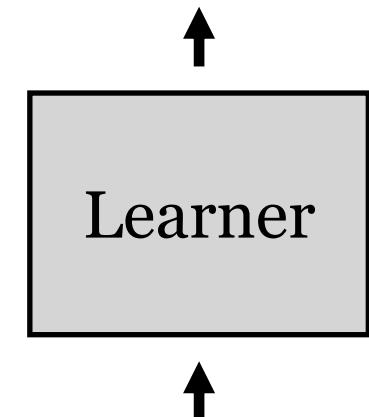
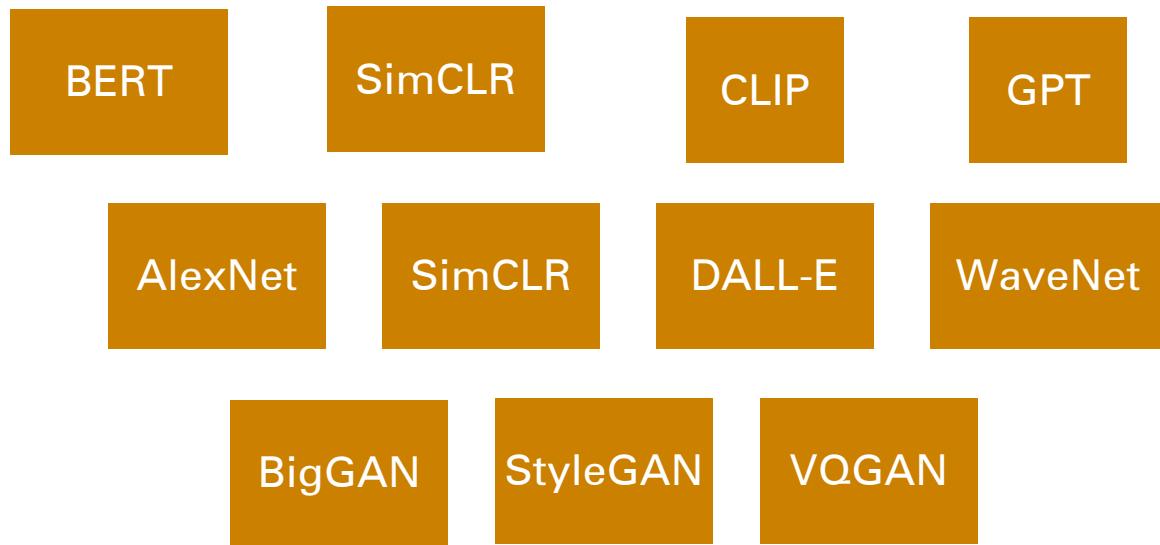
1. Learn foundation model encoders and decoders for each domain



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

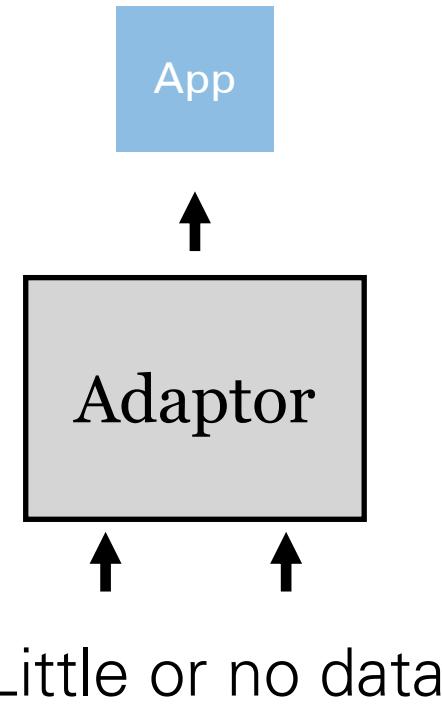


Learn foundation models



Tons of data

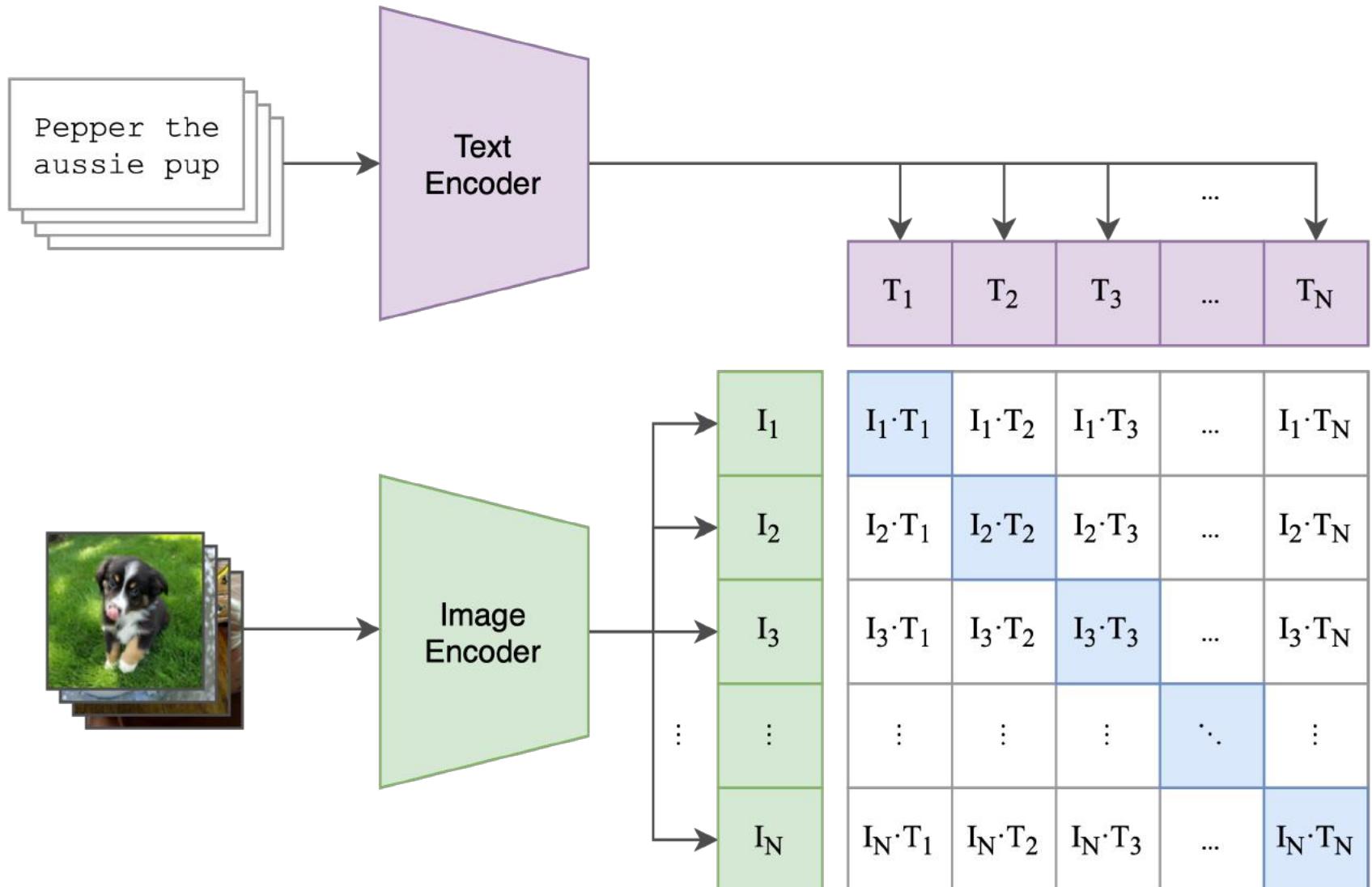
Use/adapt foundations to solve new problems



Little or no data

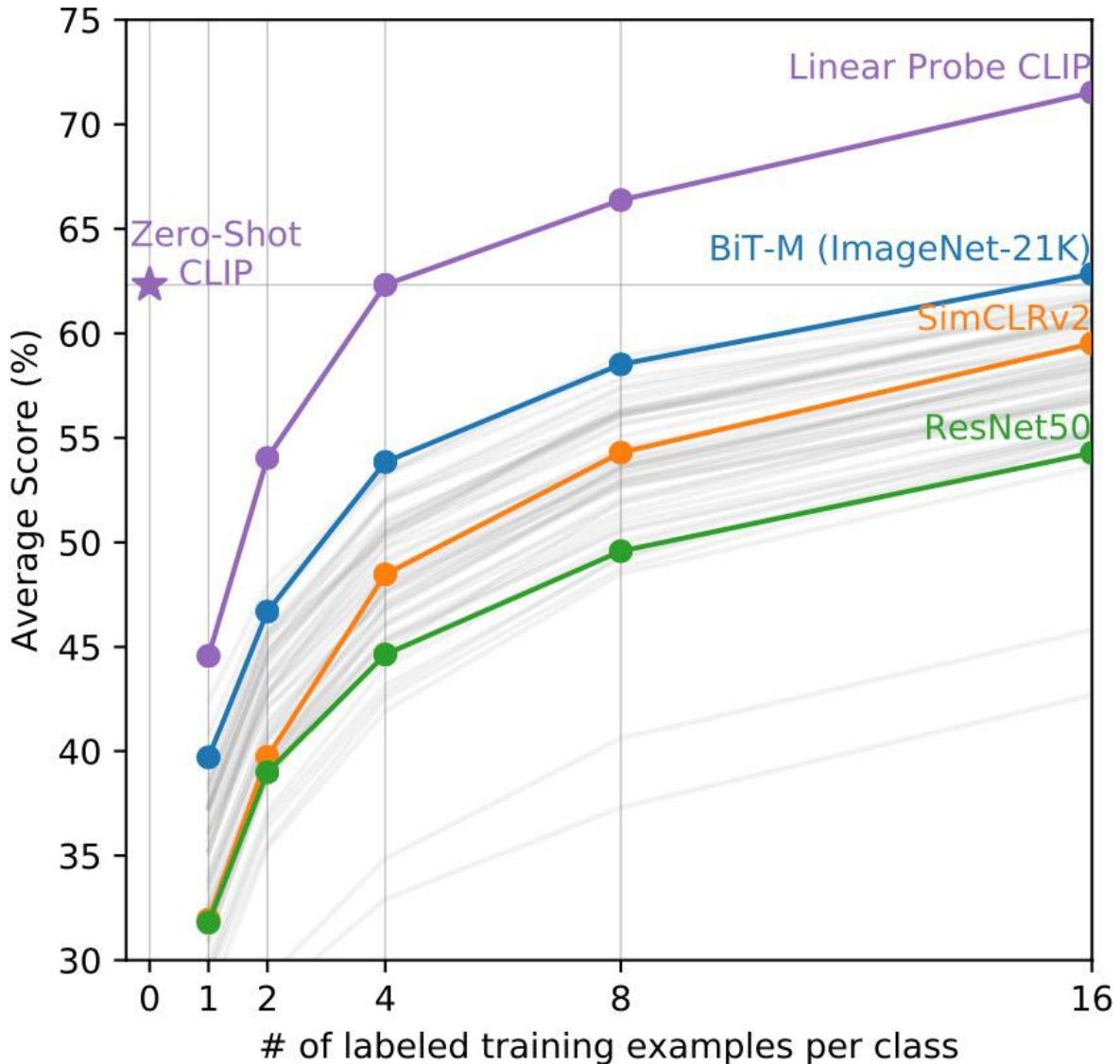
CLIP

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



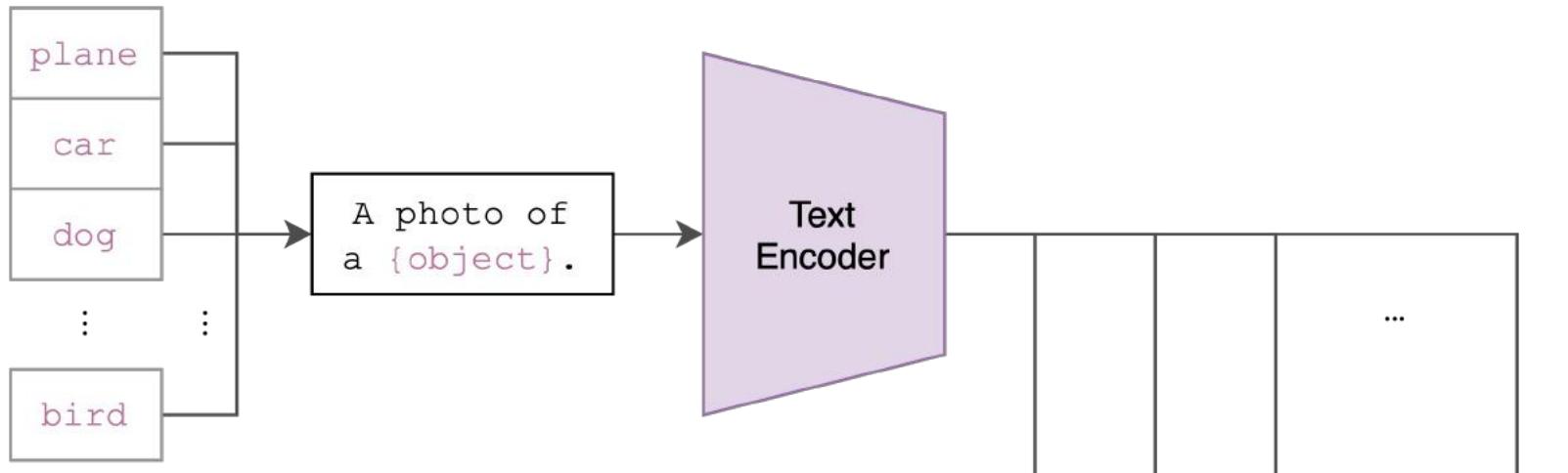
<https://openai.com/blog/clip/>

2. Adaptor: Linear classifier on top of image encodings

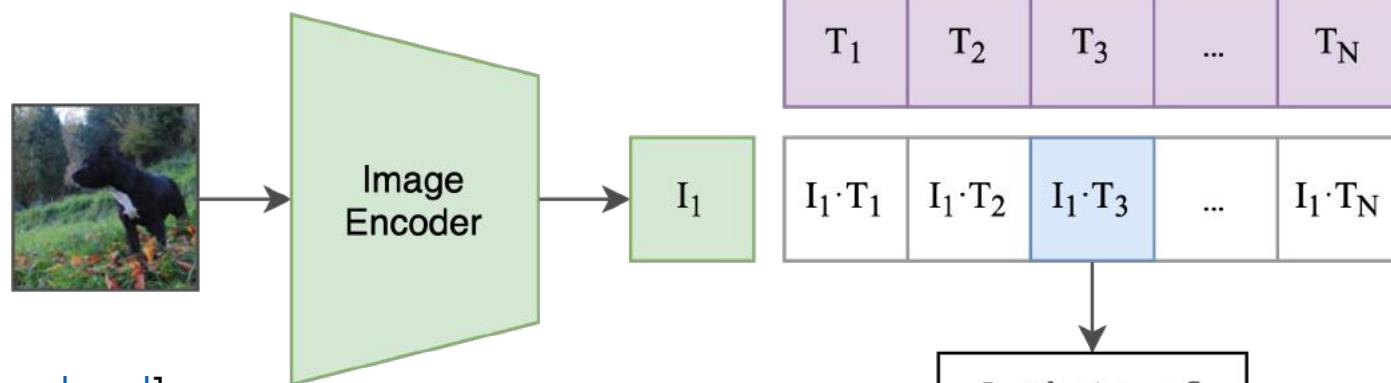


[<https://openai.com/blog/clip/>]

(2) Create dataset classifier from label text



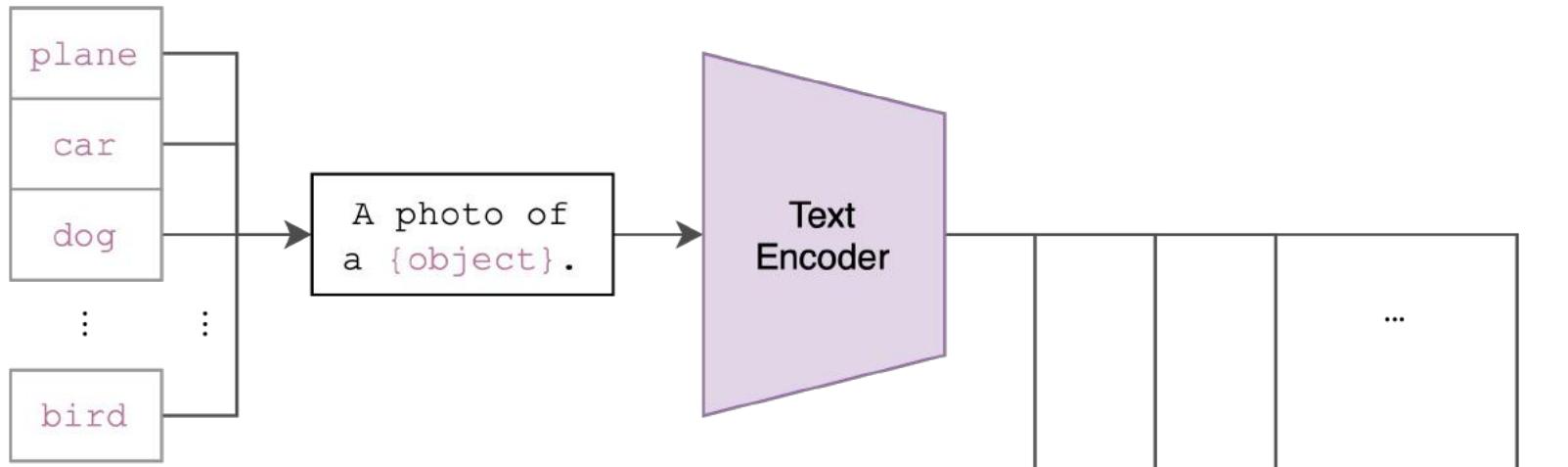
(3) Use for zero-shot prediction

2. Adaptor:
Just ask

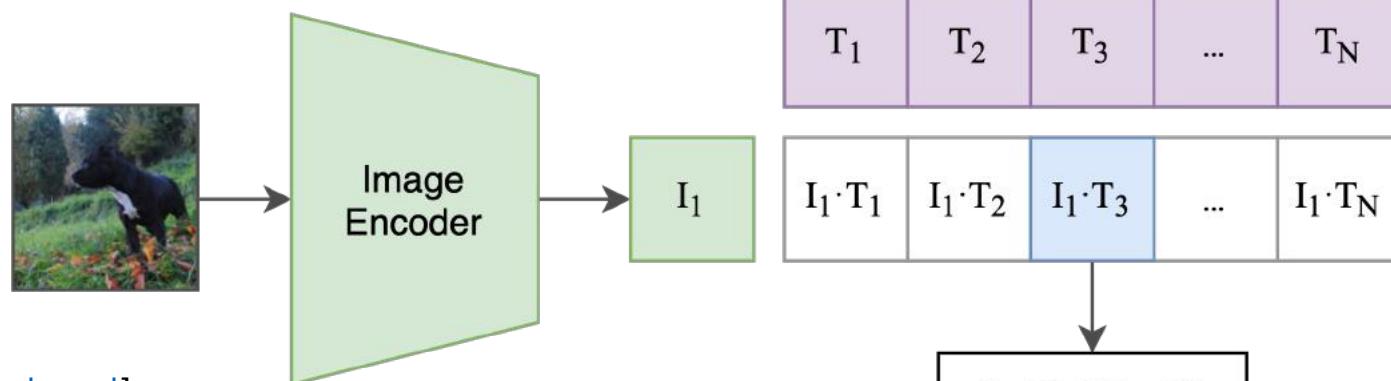
[<https://evjang.com/2021/10/23/generalization.html>]

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

(2) Create dataset classifier from label text



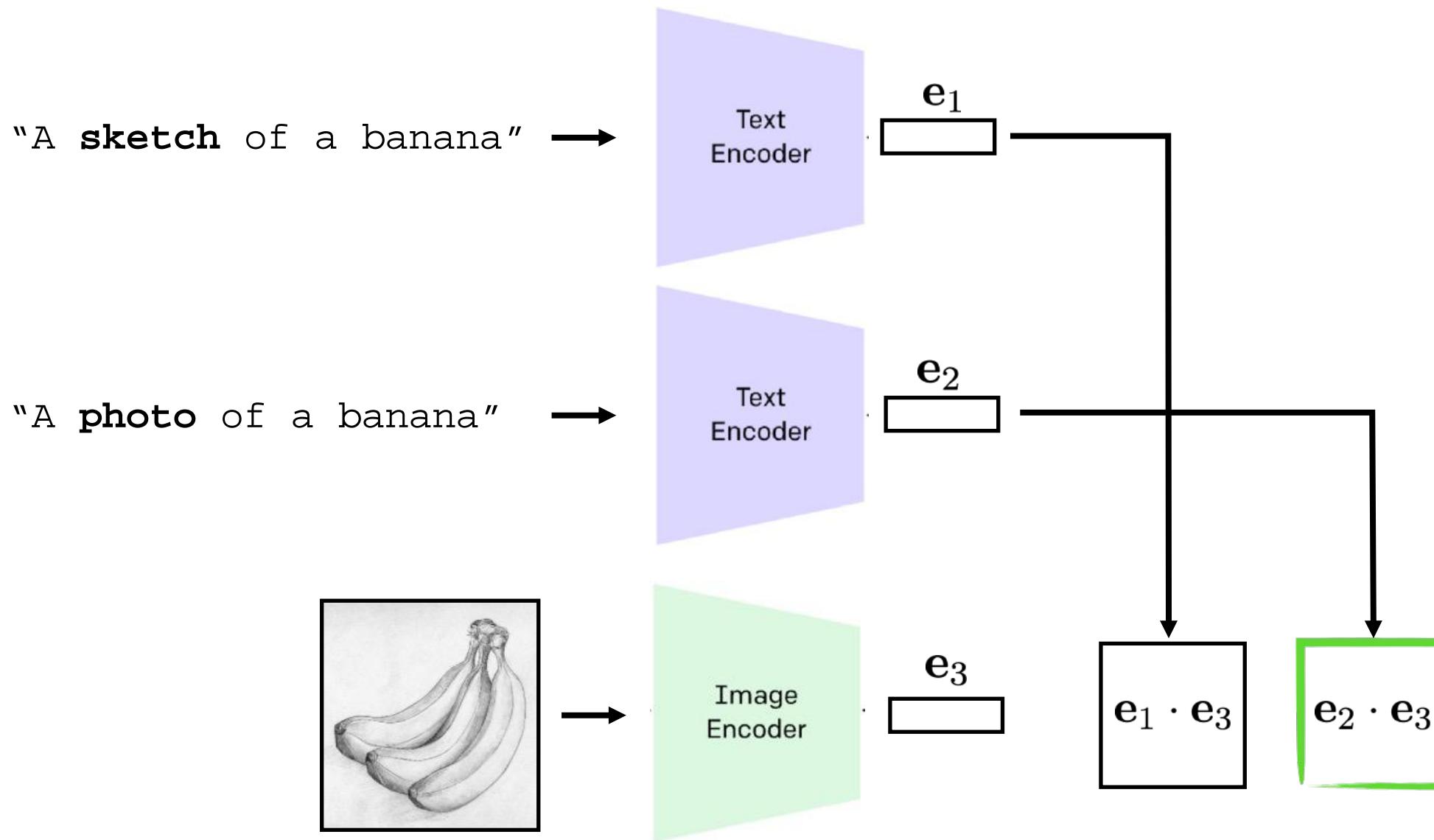
(3) Use for zero-shot prediction

2. Adaptor:
Just ask

[<https://evjang.com/2021/10/23/generalization.html>]

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

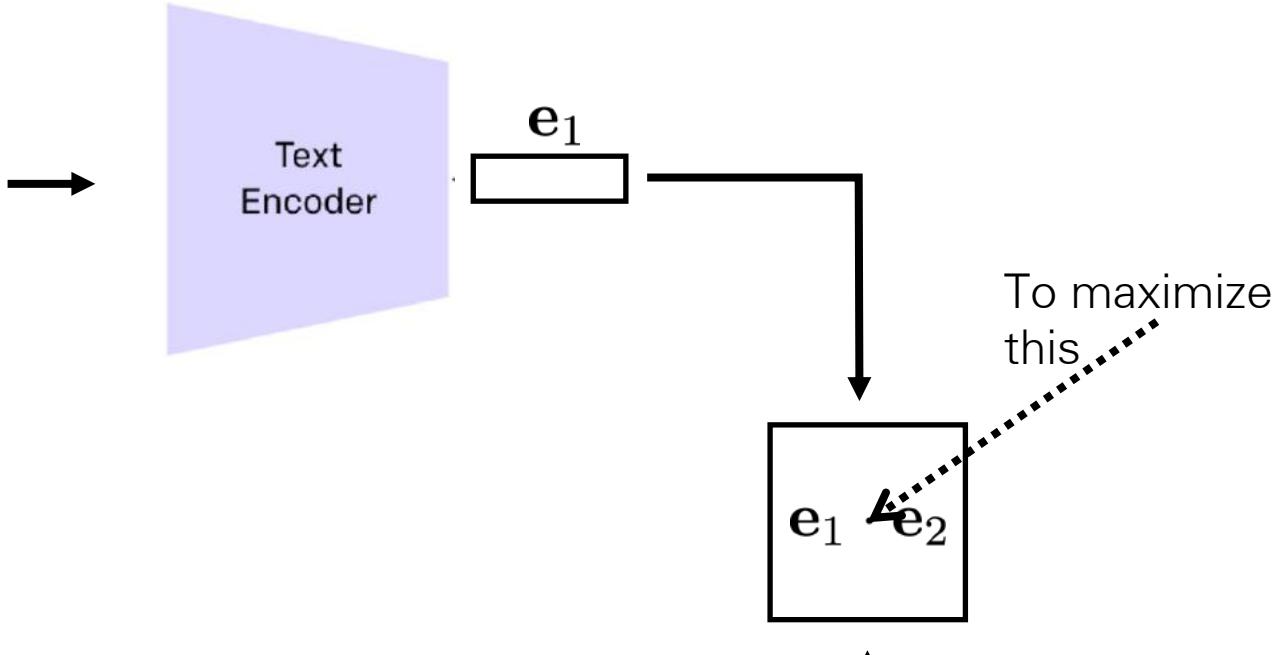
New capabilities by just asking



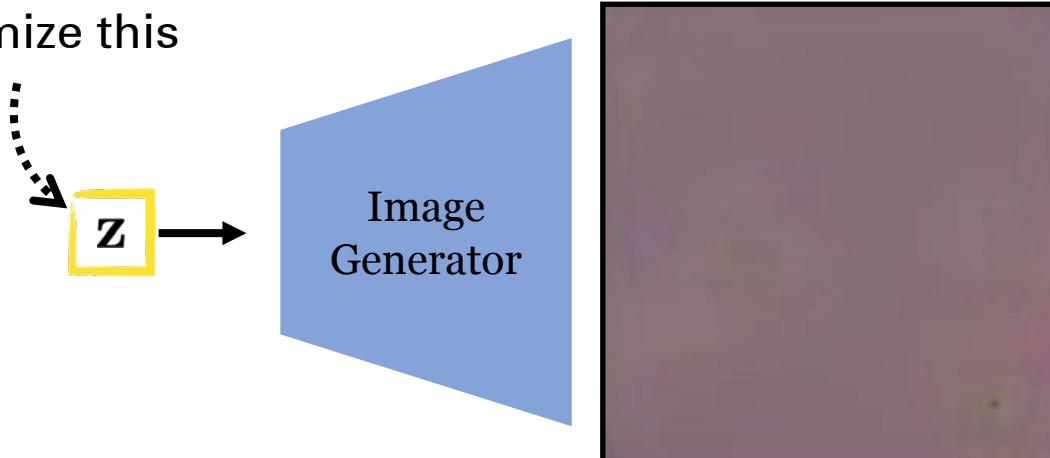
New capabilities by plugging pretrained models together: CLIP+GAN

INPUT:

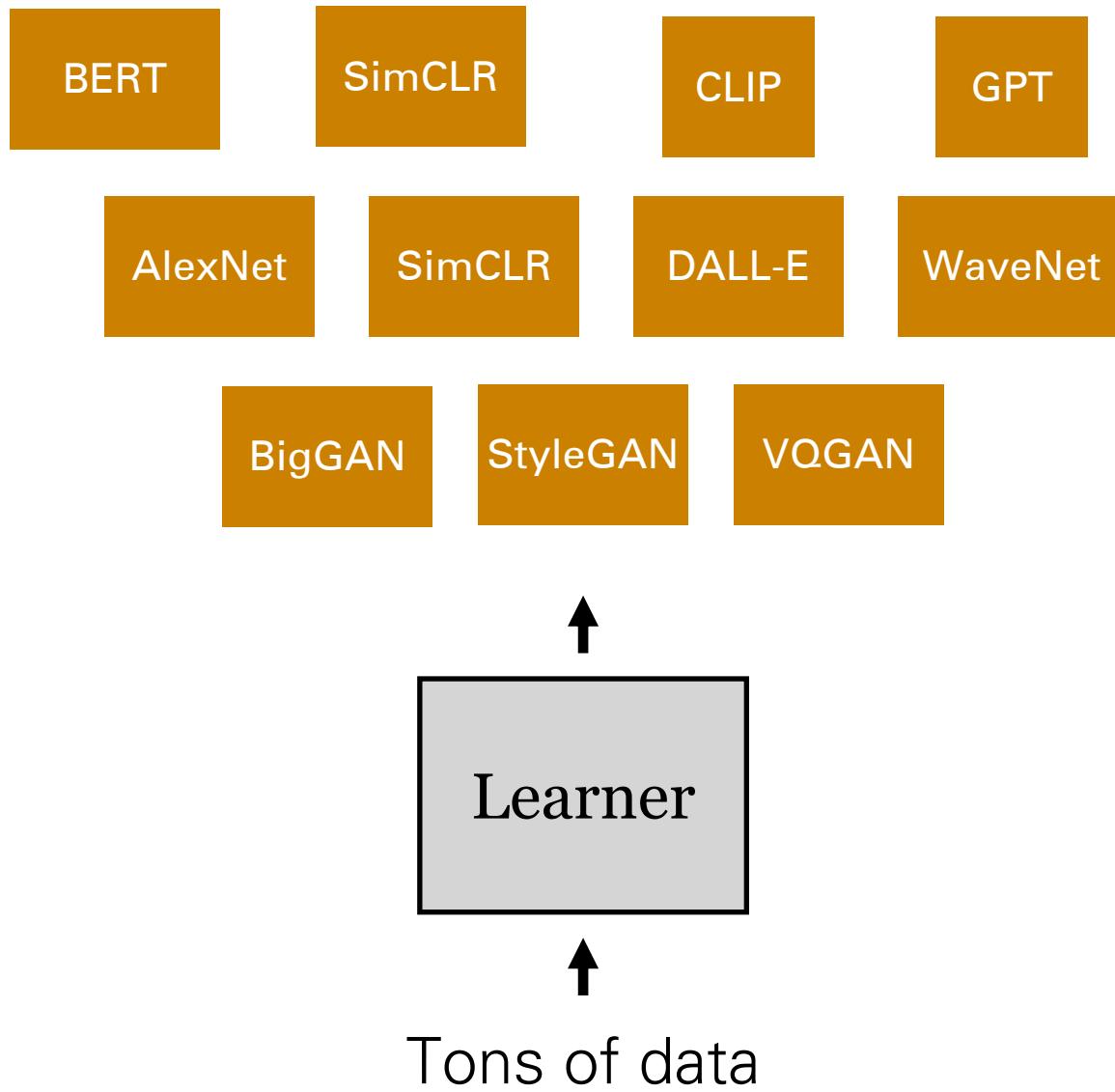
"What is the answer to the ultimate question of life, the universe, and everything?"



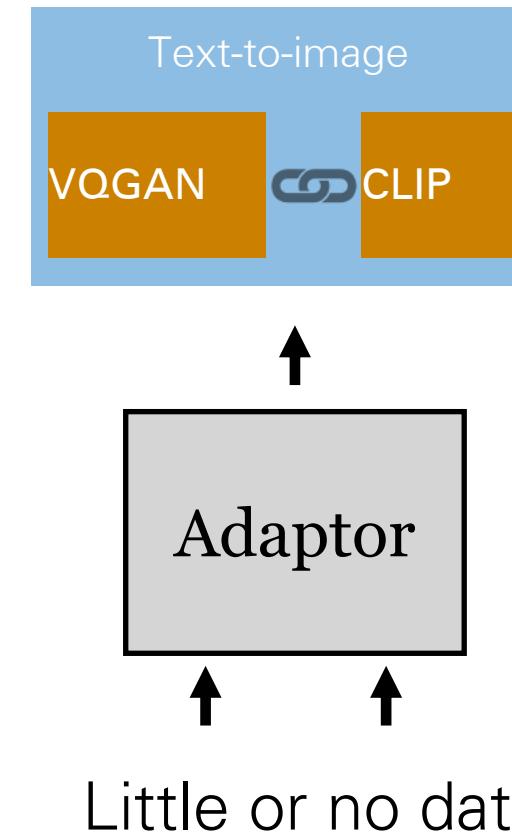
Optimize this



Learn foundation models



Use/adapt foundations to solve new problems



DALL-E

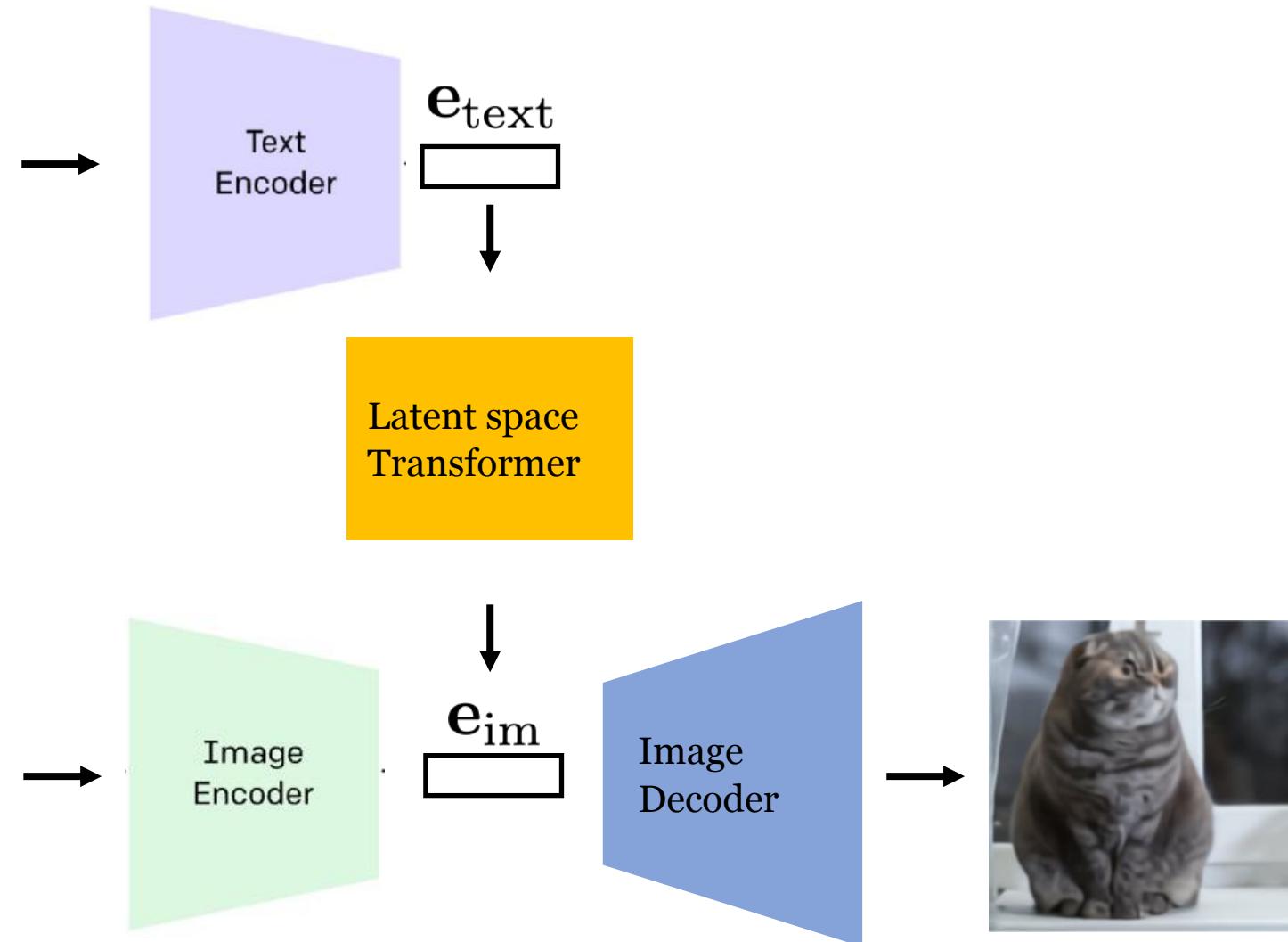
[Ramesh et al. 2021]

<https://arxiv.org/pdf/2102.12092.pdf>

<https://openai.com/blog/dall-e/>

INPUT:

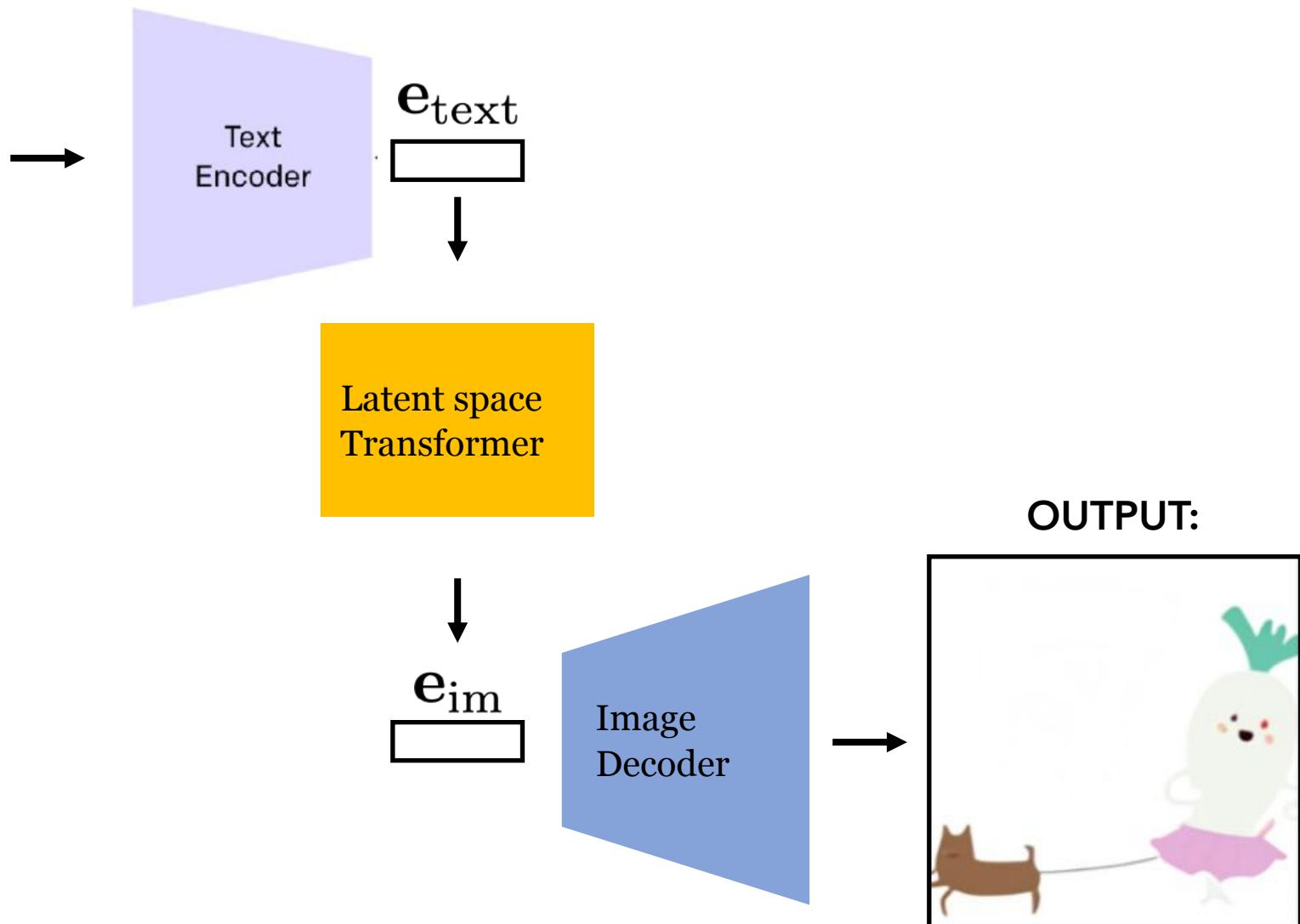
"A wide-eyed cat on
the lookout for food"



Text-to-image translation

INPUT:

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



OUTPUT:



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

1900



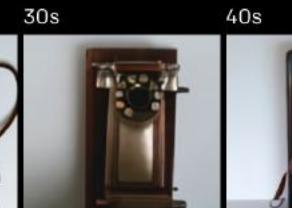
1910s



20s



30s



40s



50s



60s



70s



80s



90s



two thousands



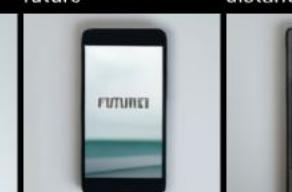
twenty tens



today



future

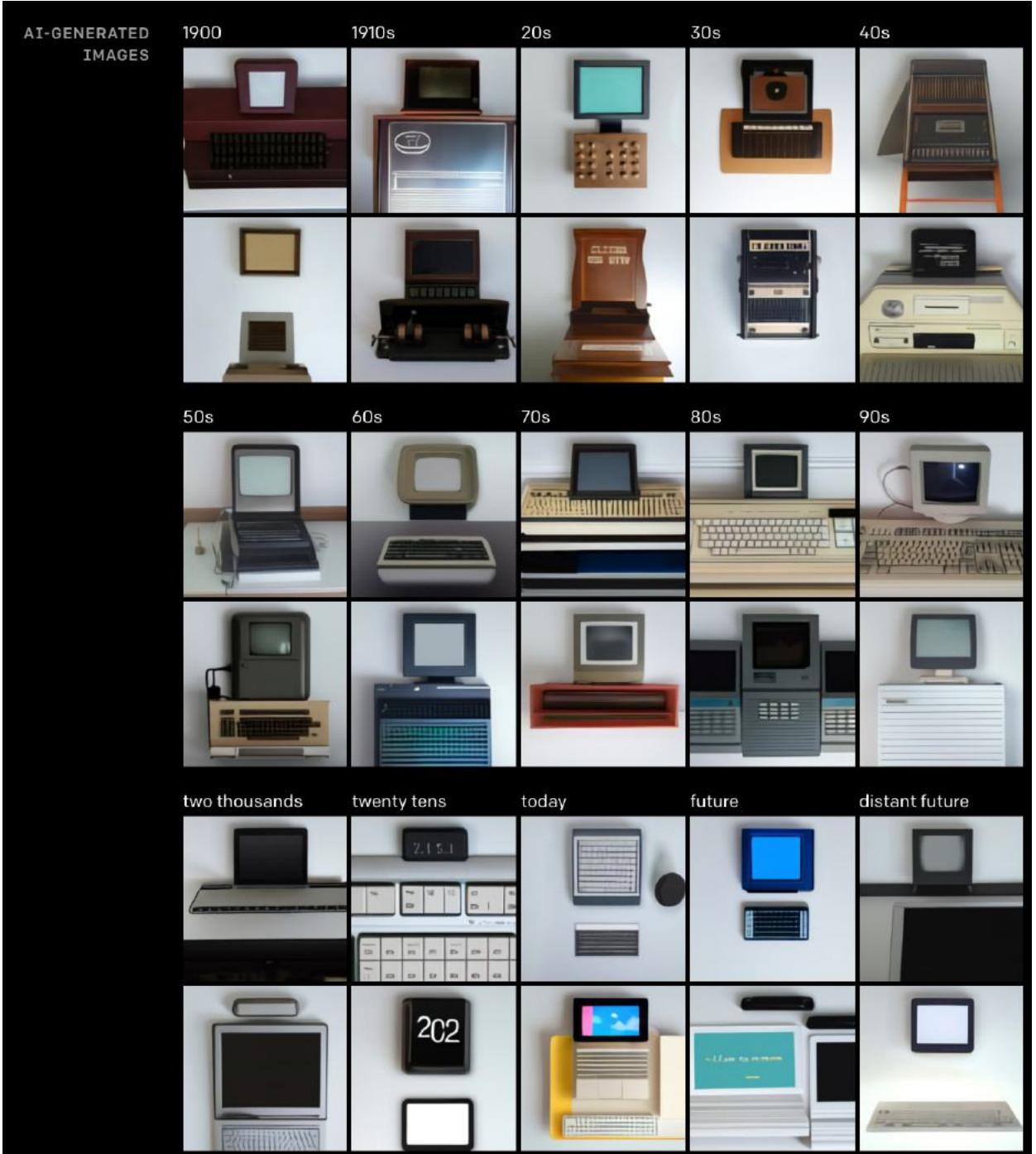


distant future



TEXT PROMPT

a photo of a computer from the ...



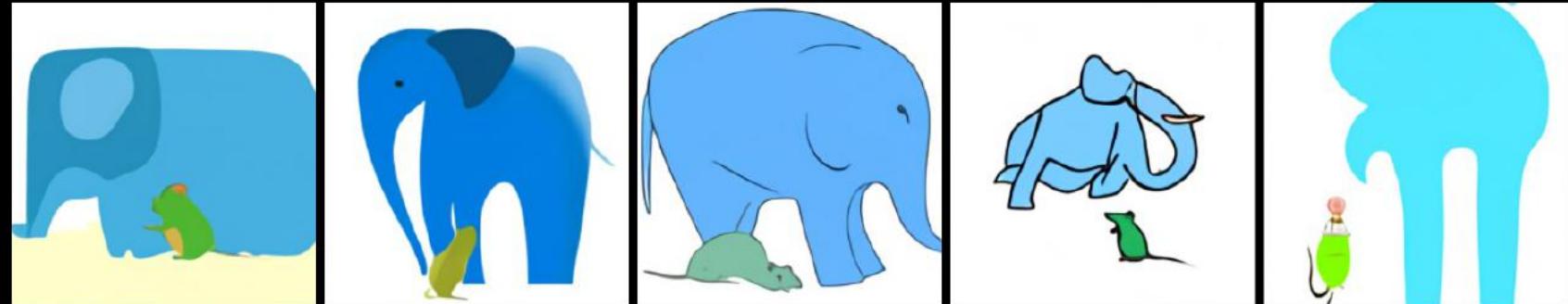
DALL-E

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

TEXT PROMPT

an illustration of a small green mouse sitting below a large blue elephant

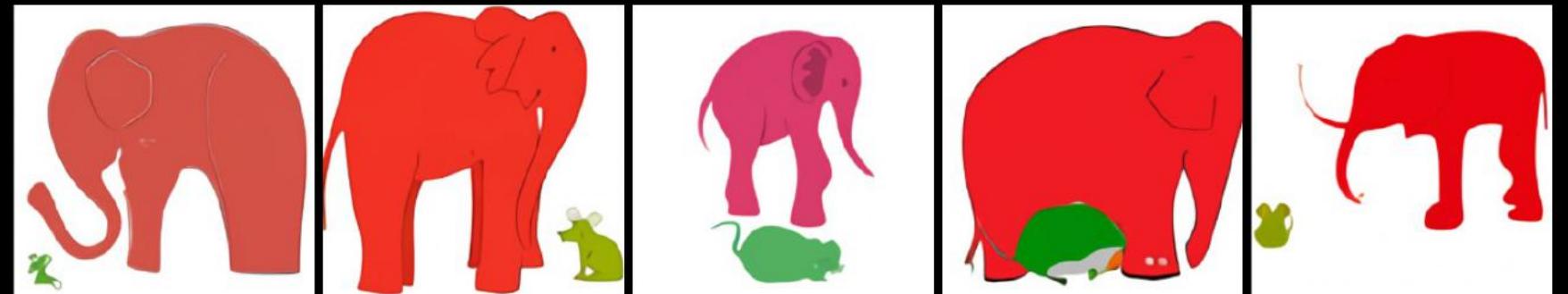
AI-GENERATED
IMAGES



TEXT PROMPT

an illustration of a small green mouse sitting below a large red elephant

AI-GENERATED
IMAGES



DALL-E 2

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



Report issue



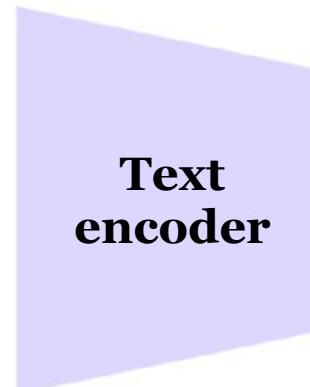
DALL-E 2

[Ramesh et al. 2022]

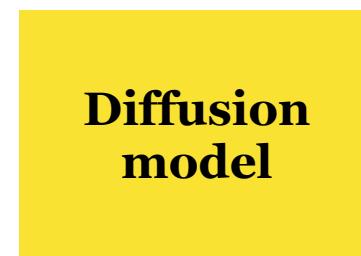
<https://cdn.openai.com/papers/dall-e-2.pdf>

INPUT:

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



e_{text}

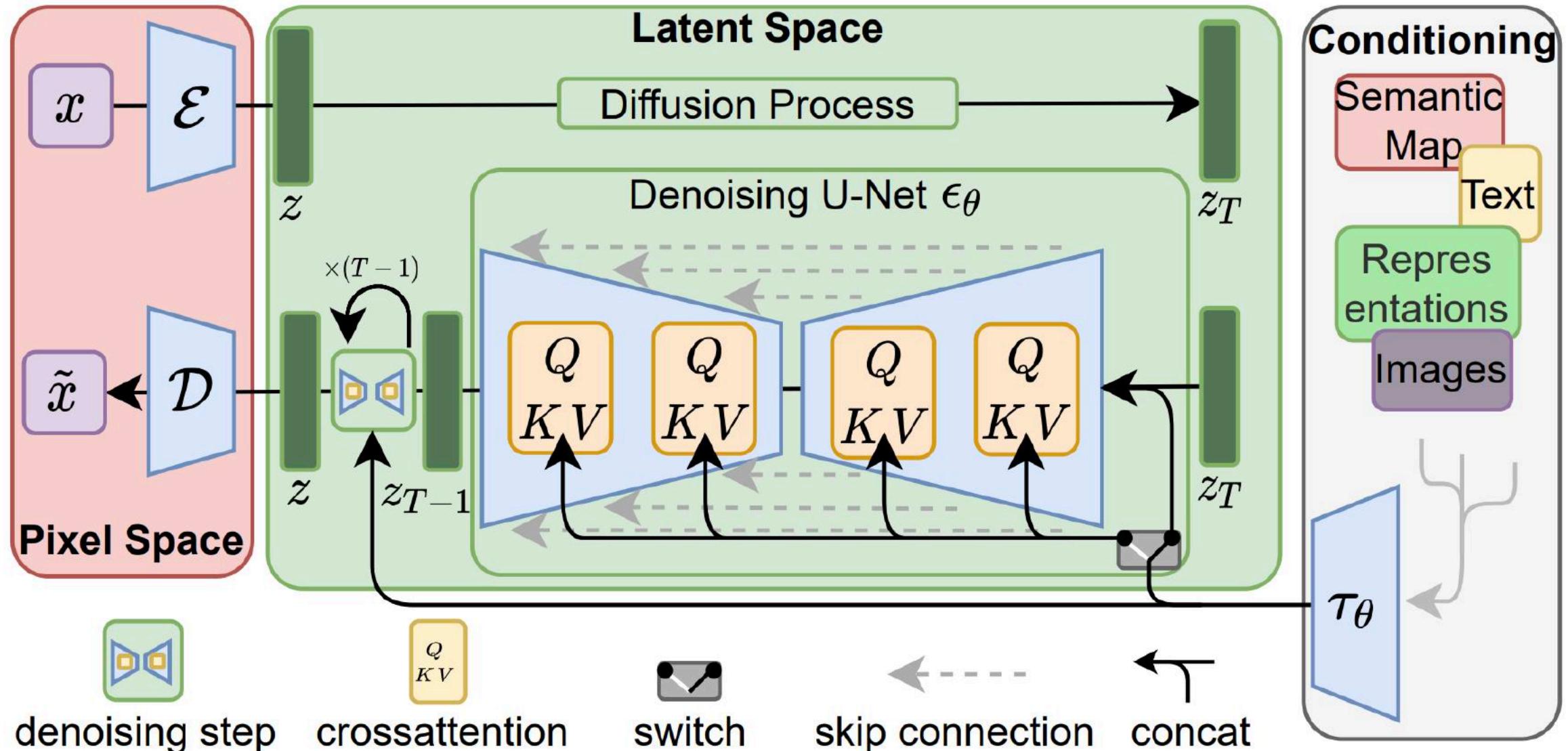


OUTPUT:



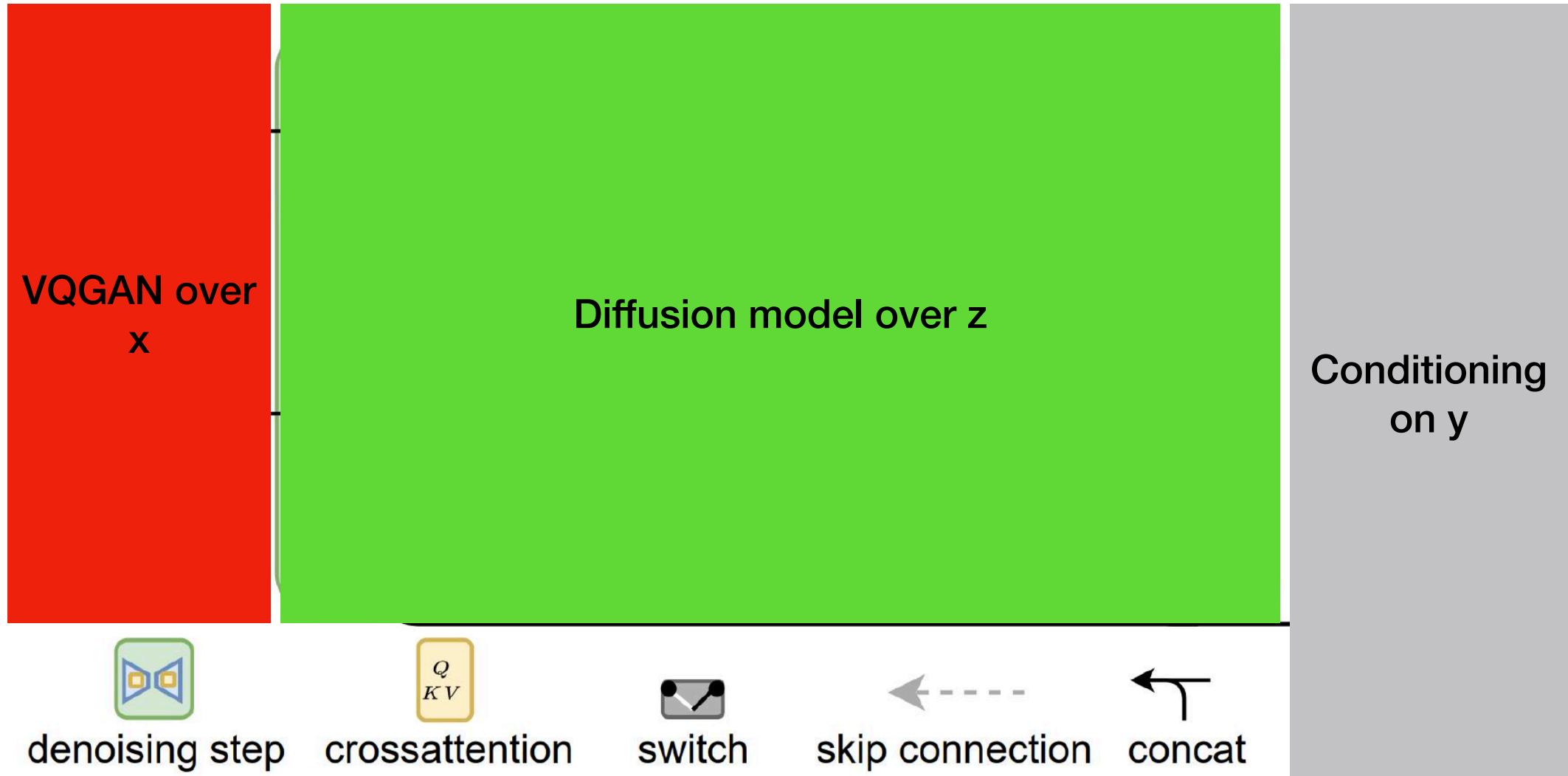
Latent diffusion

[Rombach*, Blattman* et al. 2022] <https://arxiv.org/abs/2112.10752>



Latent diffusion

[Rombach*, Blattman* et al. 2022] <https://arxiv.org/abs/2112.10752>



Stable Diffusion

[Rombach*, Blattman* et al. 2022] <https://arxiv.org/abs/2112.10752>



Stable Diffusion Applications: Twitter Mega Thread

https://twitter.com/daniel_eckler/status/1572210382944538624

IMAGEN

[Saharia*, Chan*, Saxena† et al. 2022] <https://arxiv.org/pdf/2205.11487.pdf>



Sprouts in the shape of text 'Imagen' coming out of a fairytale book.



A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.



A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

IMAGEN Video

[Ho et al., "Imagen Video", 2022]

<https://arxiv.org/abs/2210.02303>



A teddy bear
running in New York City



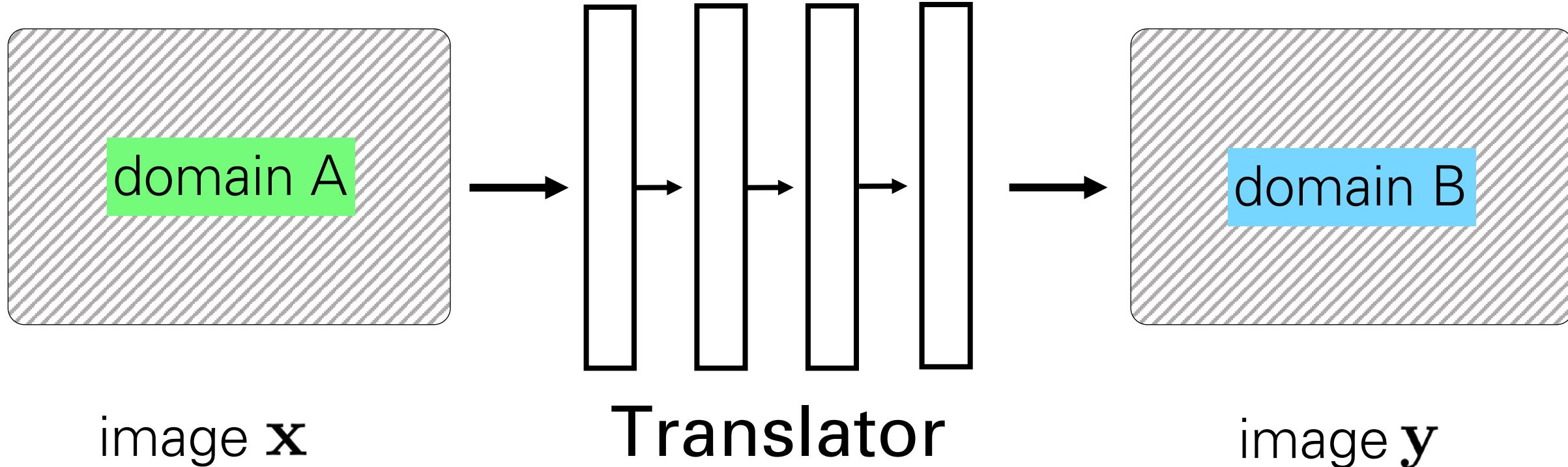
A british shorthair
jumping over a coach



A swarm of bees
flying around their hive

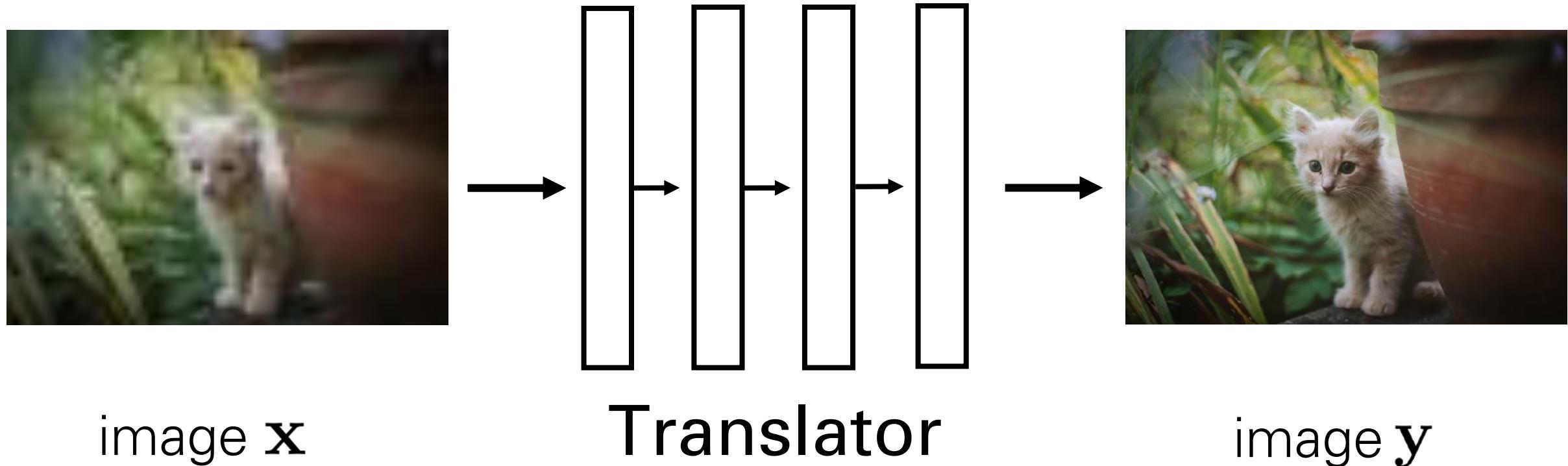
Applications in Computational Photography

Image-to-Image Translation



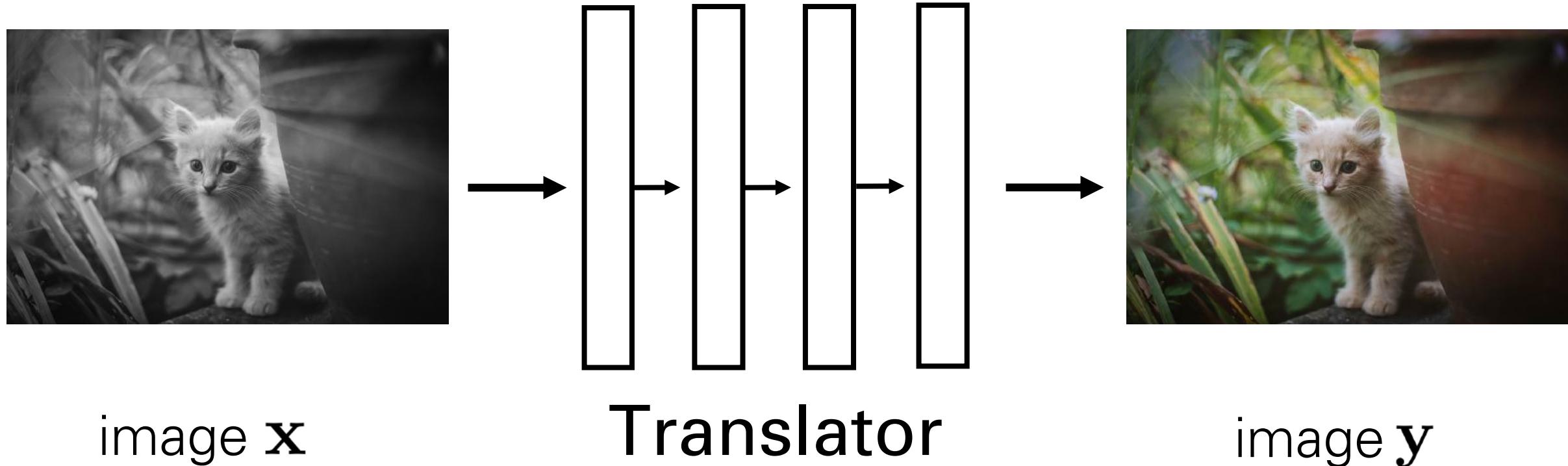
- Learning to map images from one domain into another
- A general framework for both low and high-level image processing

Super-resolution



- Input: low-res image
 - Output: high-res image

Colorization



- Input: black & white image
 - Output: color image

BW → Color

Input



Output



Input



Output



Input



Output

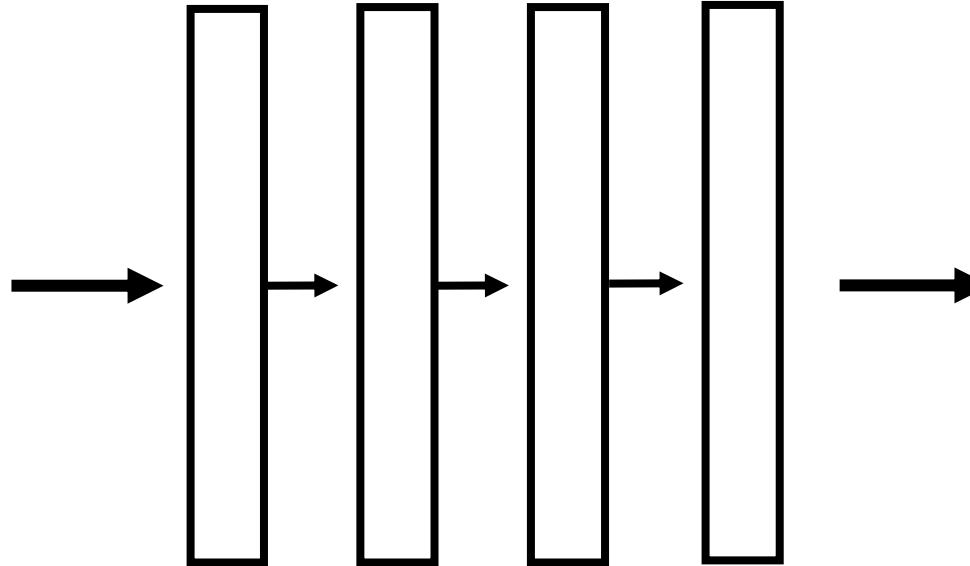


Data from [Russakovsky et al. 2015]

Day to Night



image **x**



Translator



image **y**

- Input: day image
- Output: night image

Photo Style Transfer



image **x**

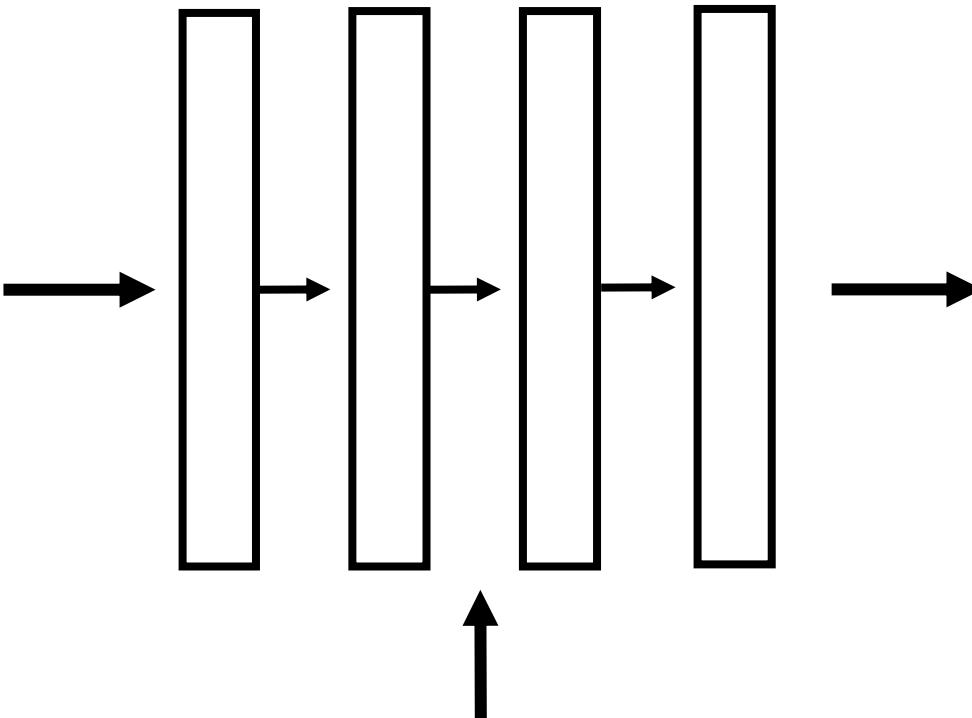


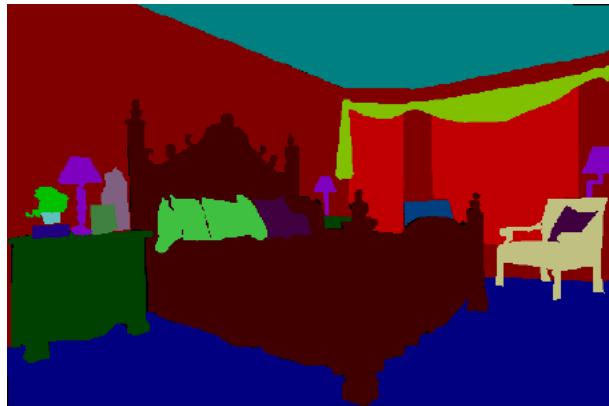
image **y**

- Input: input image
+ target style image
- Output: synthesized image

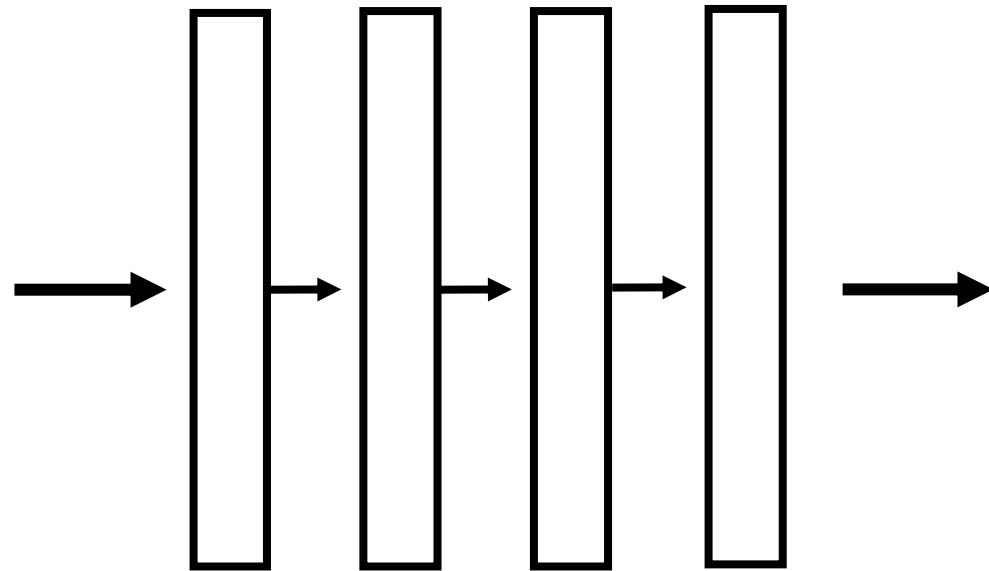


Target style
image **t**

Semantic Image Synthesis



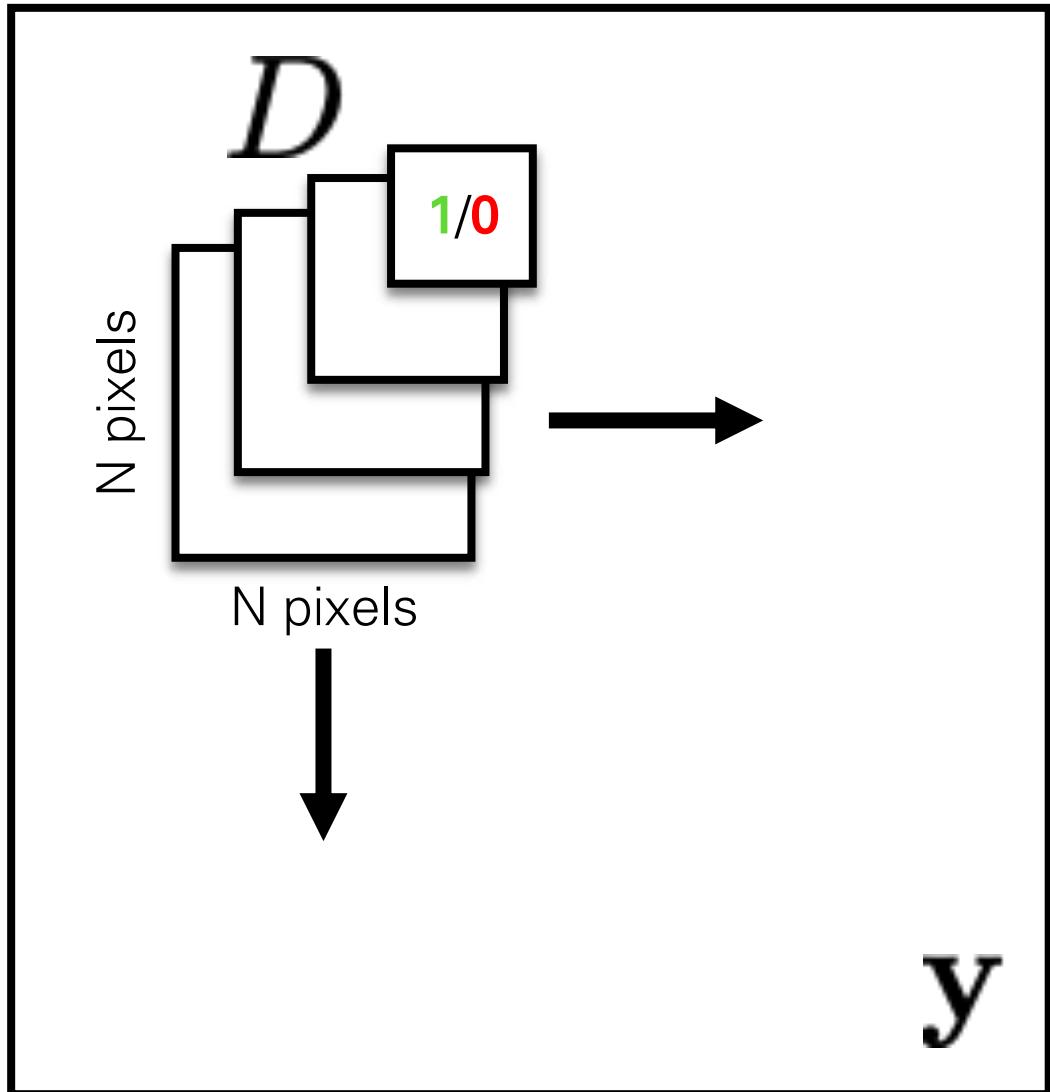
semantic
layout \mathbf{x}



synthesized
image \mathbf{y}

- Input: input layout
- Output: synthesized image

Shrinking the capacity: 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

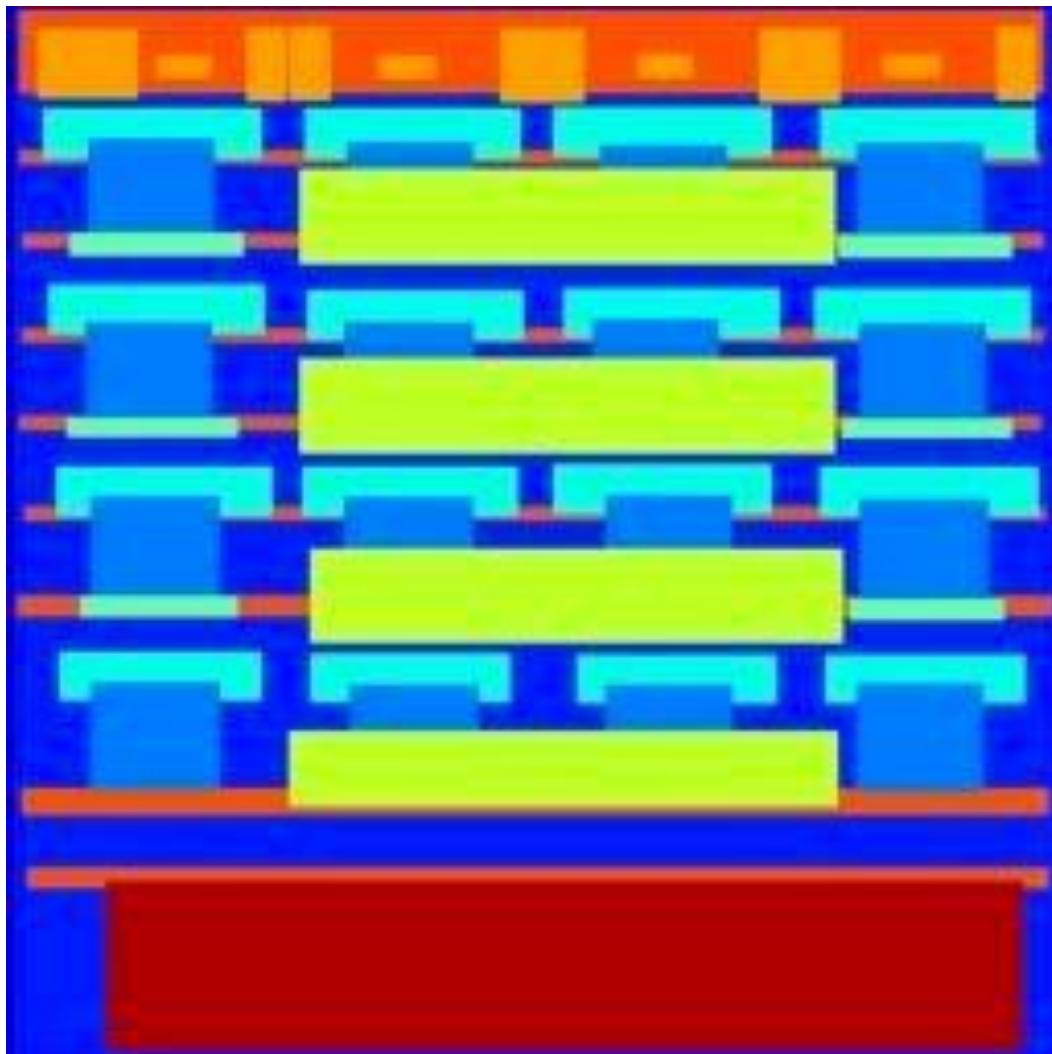
[Li & Wand 2016]

[Shrivastava et al. 2017]

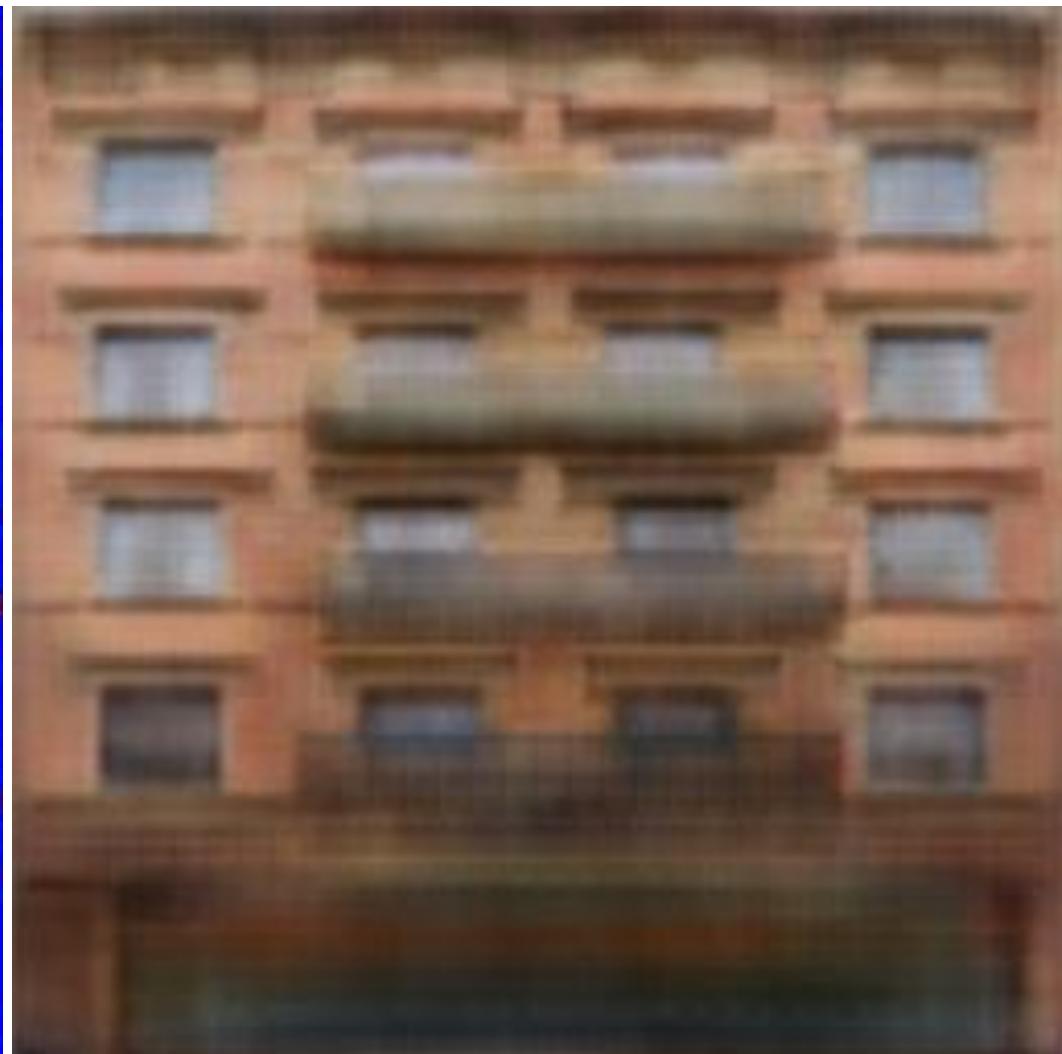
[Isola et al. 2017]

Labels → Facades

Input



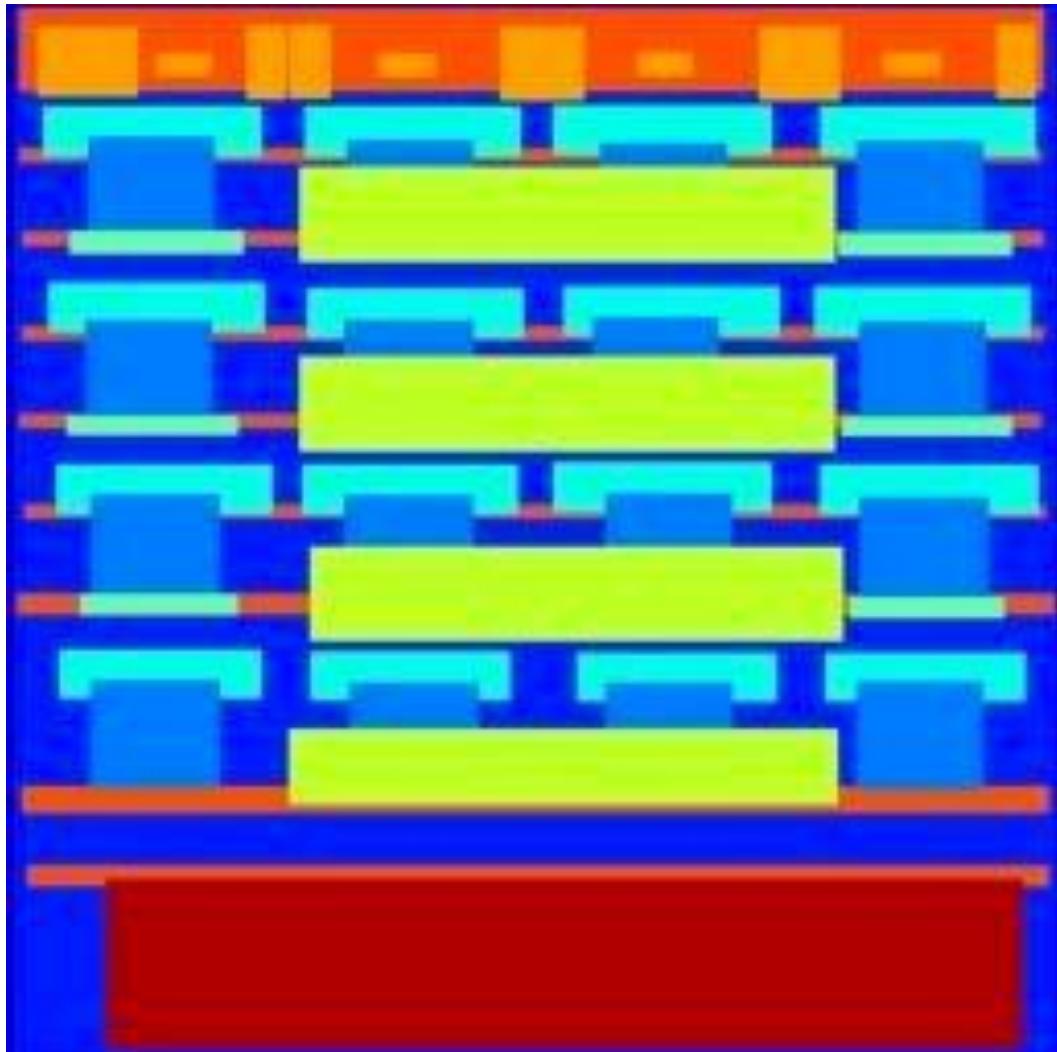
1x1 Discriminator



Data from [Tylecek, 2013]

Labels → Facades

Input



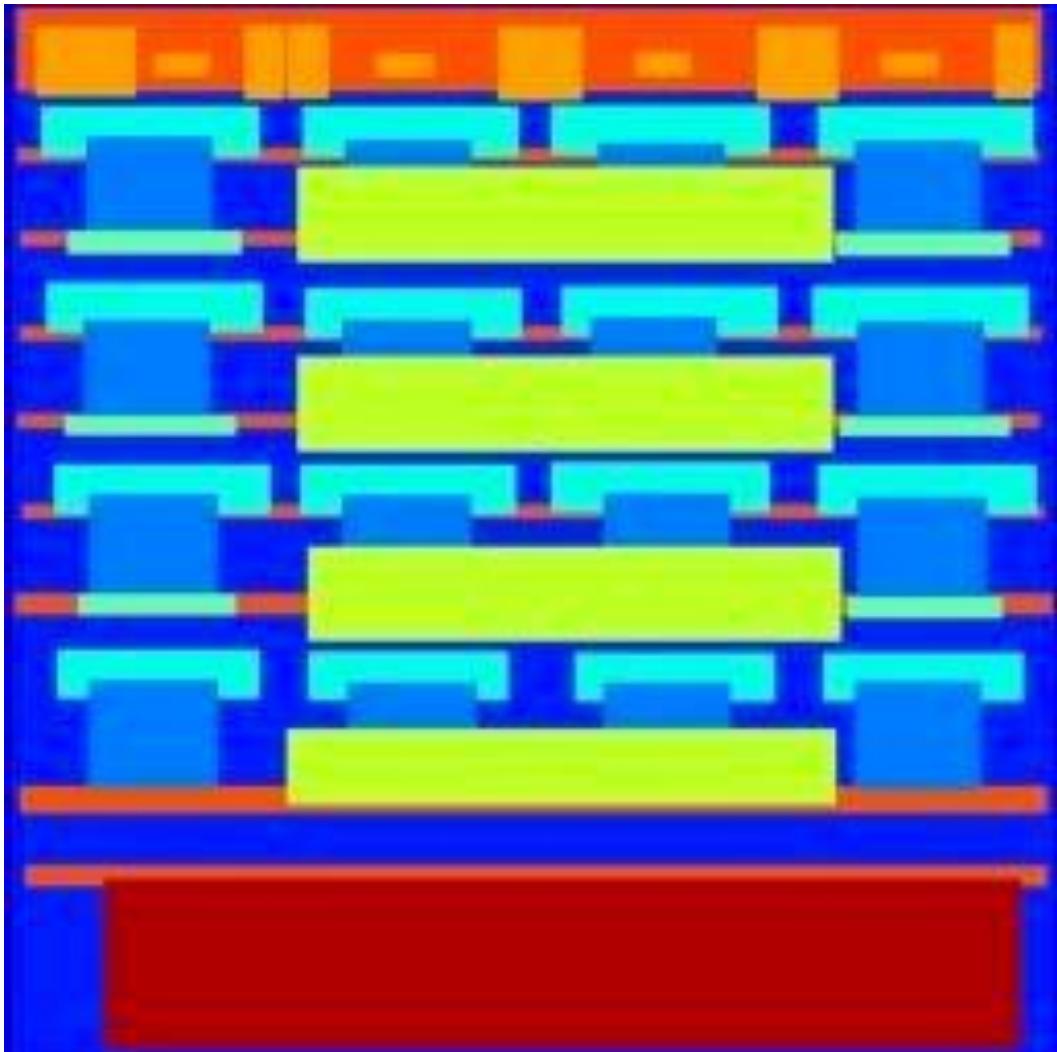
16x16 Discriminator



Data from [Tylecek, 2013]

Labels → Facades

Input



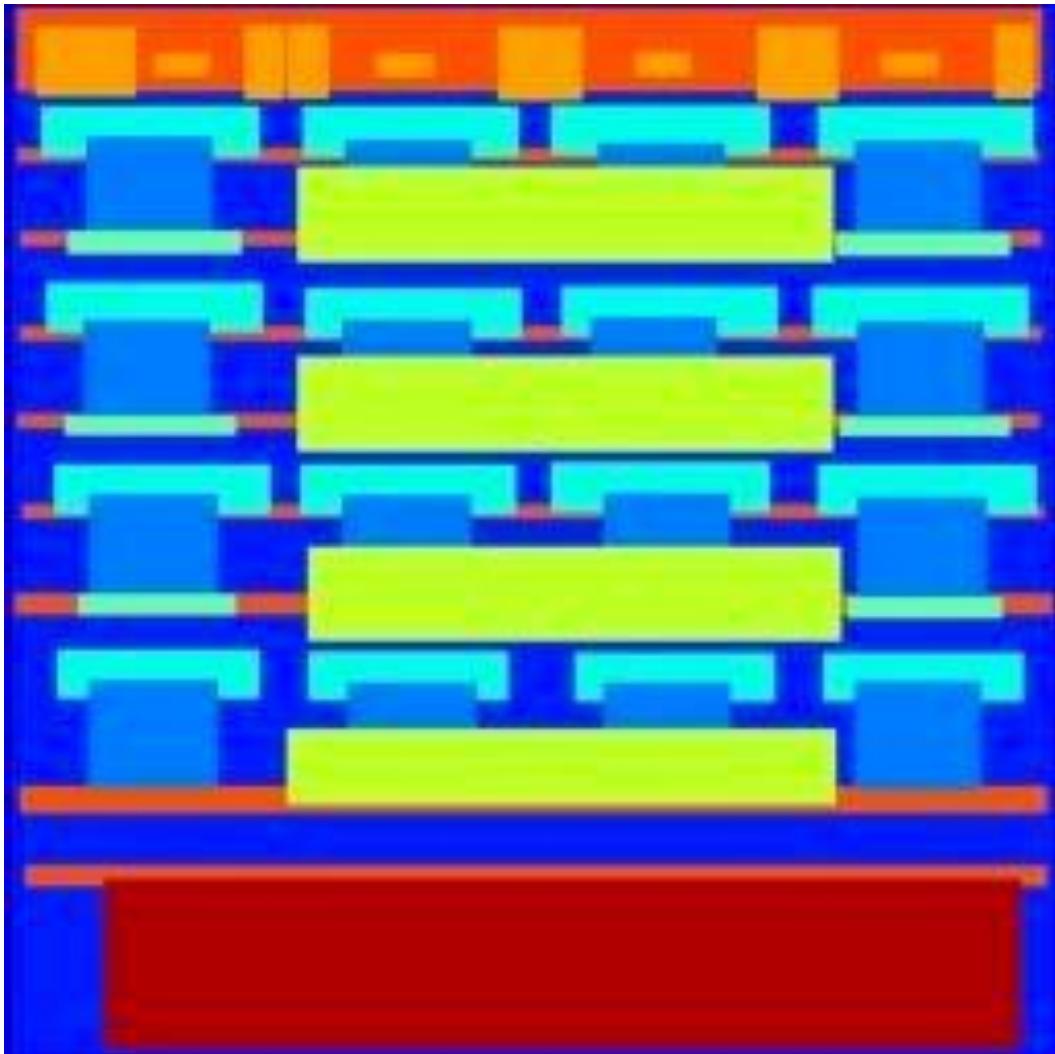
70x70 Discriminator



Data from [Tylecek, 2013]

Labels → Facades

Input



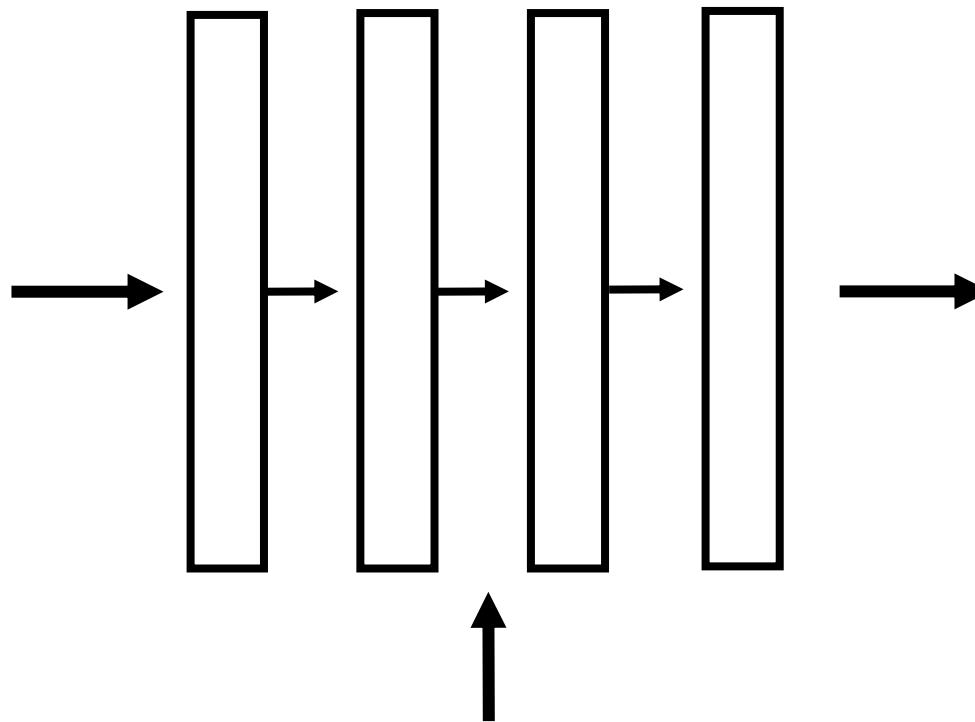
Full image Discriminator



Semantic Image Synthesis



semantic
layout \mathbf{x}



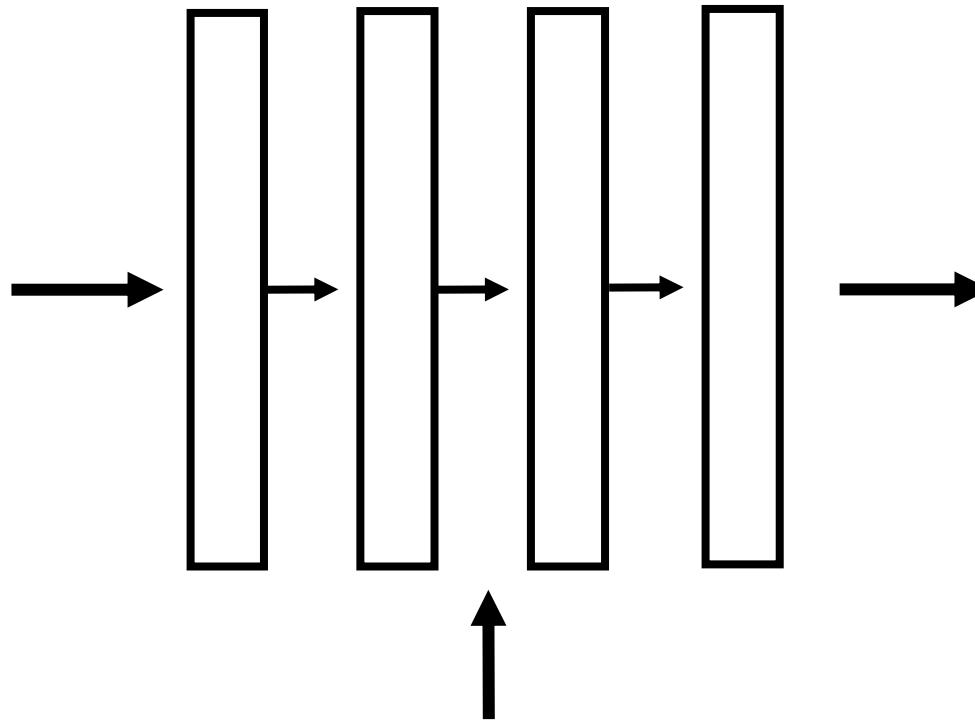
- Input: input layout
+ target style image
- Output: synthesized image

Target style
image \mathbf{t}

Semantic Image Editing



image **x**



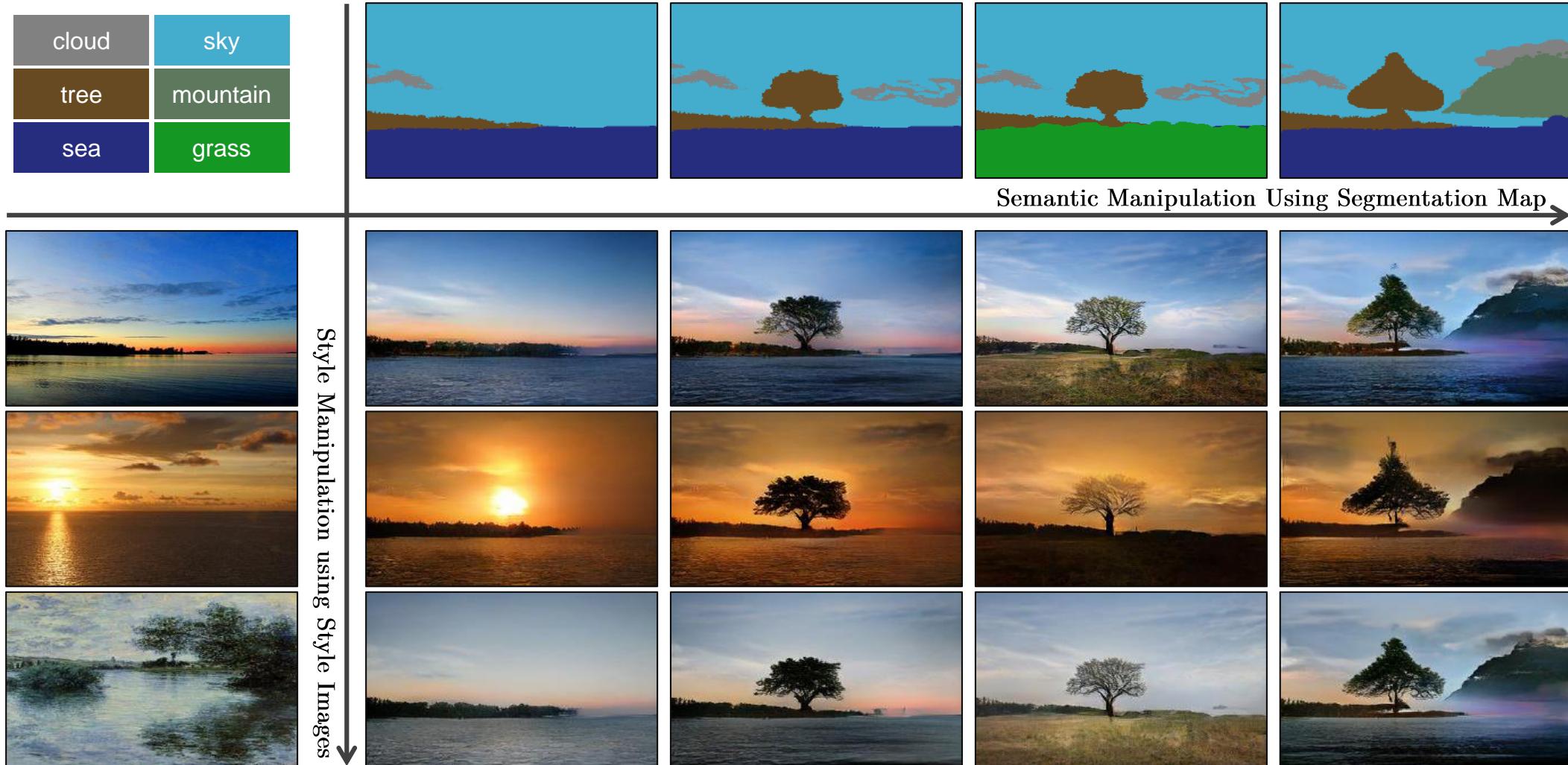
manipulated
image **y**

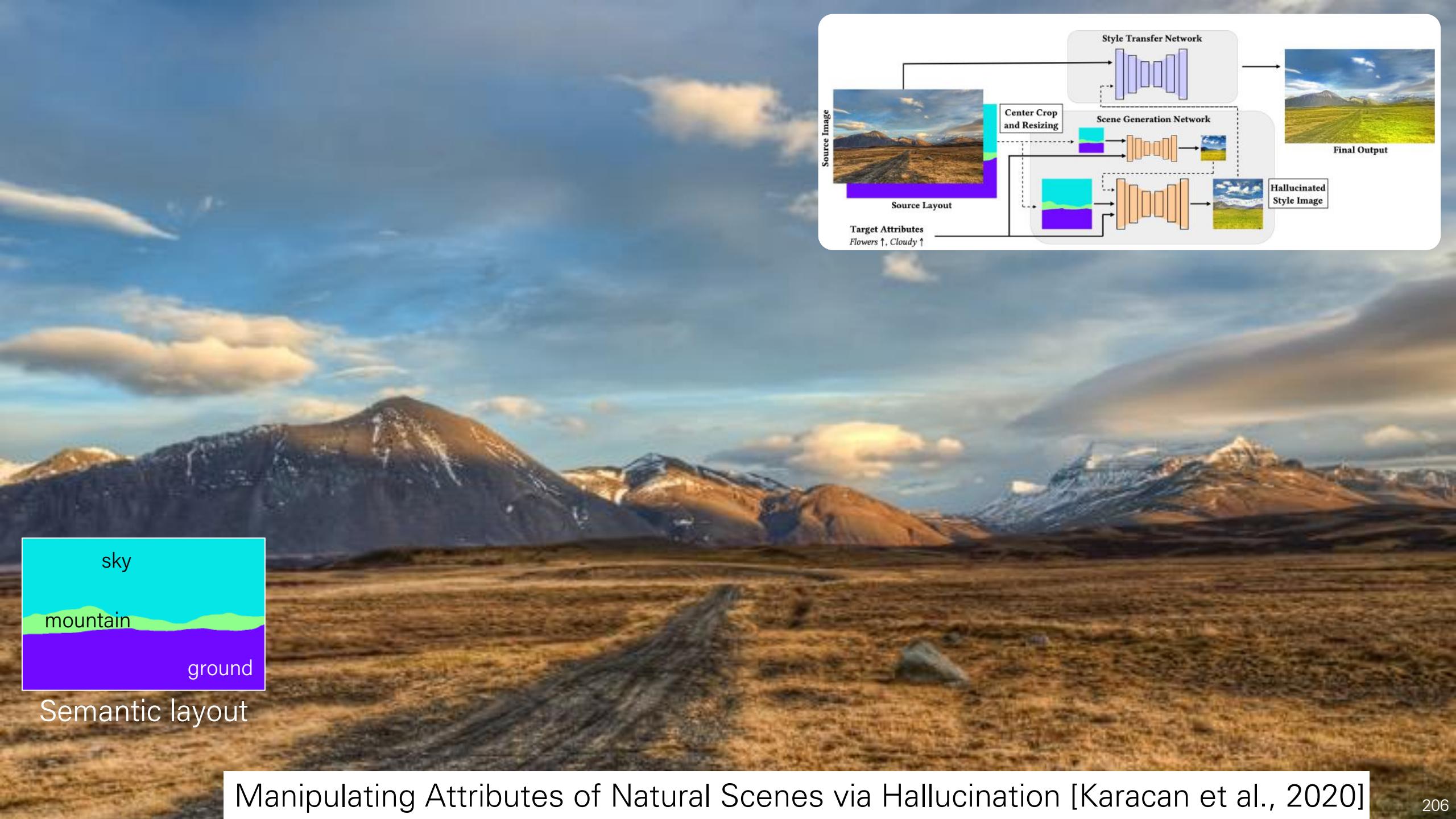
- **Input:** input image
(+ semantic layout)
+ target attribute
- Output:** manipulated image

“more flowers”
“more cloudy”
Target scene
attribute(s) **t**

Semantic Image Synthesis (SPADE) (Park et al., 2019)

- Image generation conditioned on semantic layouts





Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]



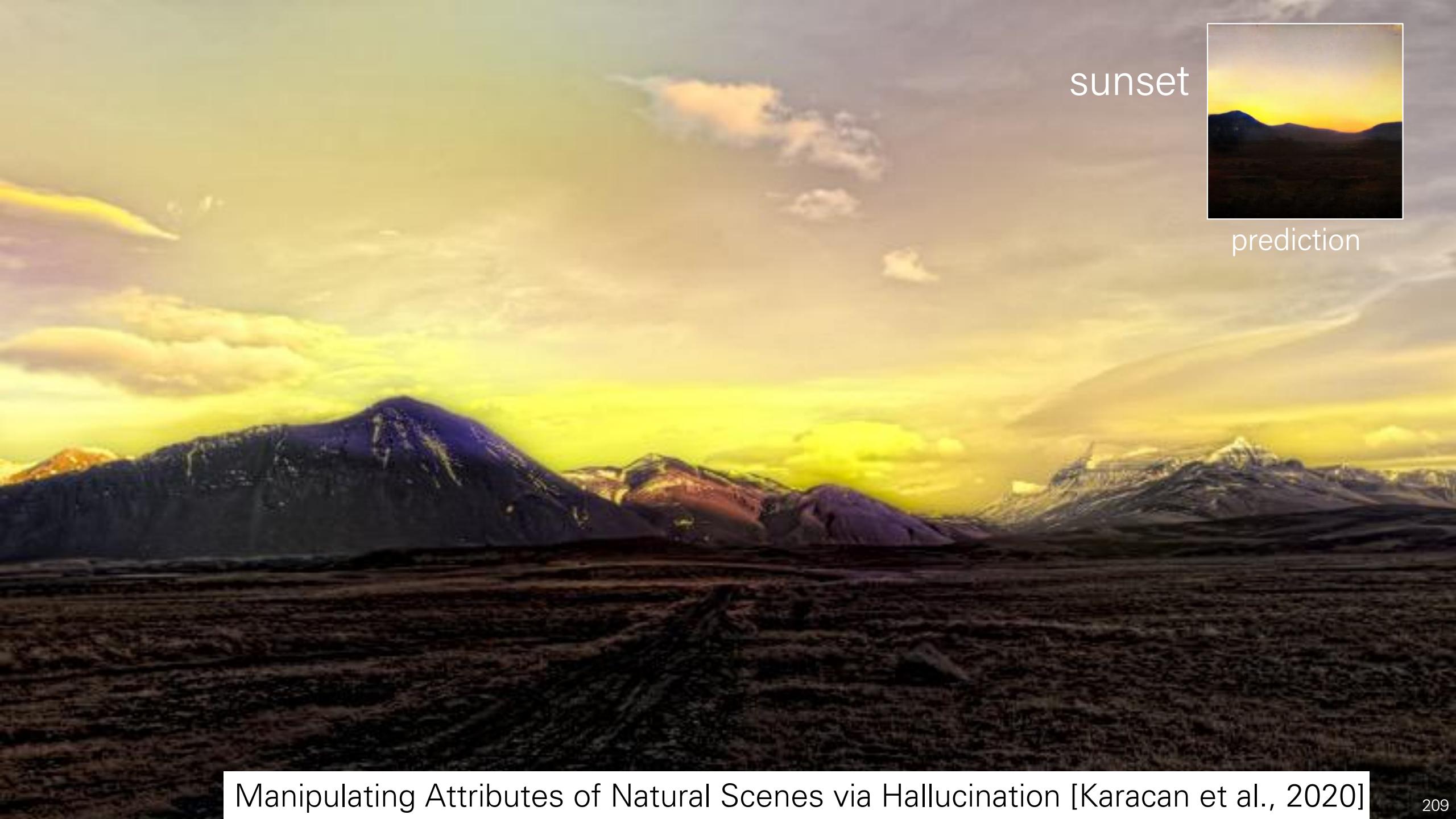
Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]



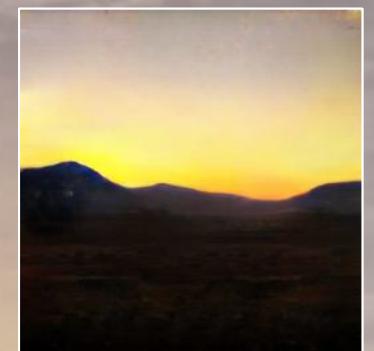
night



prediction

A wide-angle photograph of a sunset over a mountain range. The sky is filled with warm, orange and yellow hues, with scattered clouds. In the foreground, there is a dark, flat landscape, possibly a plain or a large body of water. The mountains in the background are silhouetted against the bright sky.

sunset





snow



prediction



winter



prediction

Spring and clouds



prediction



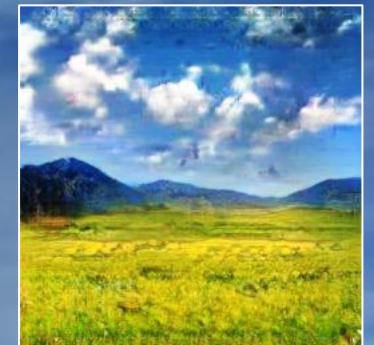
Moist, rain and fog



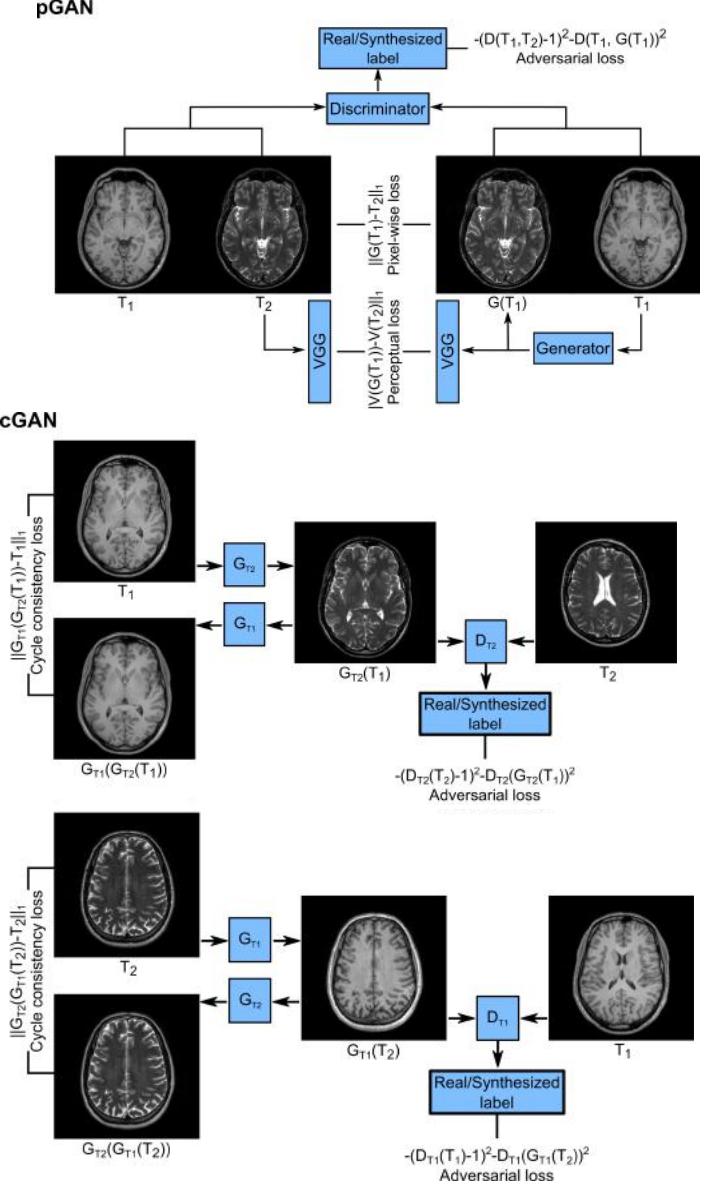
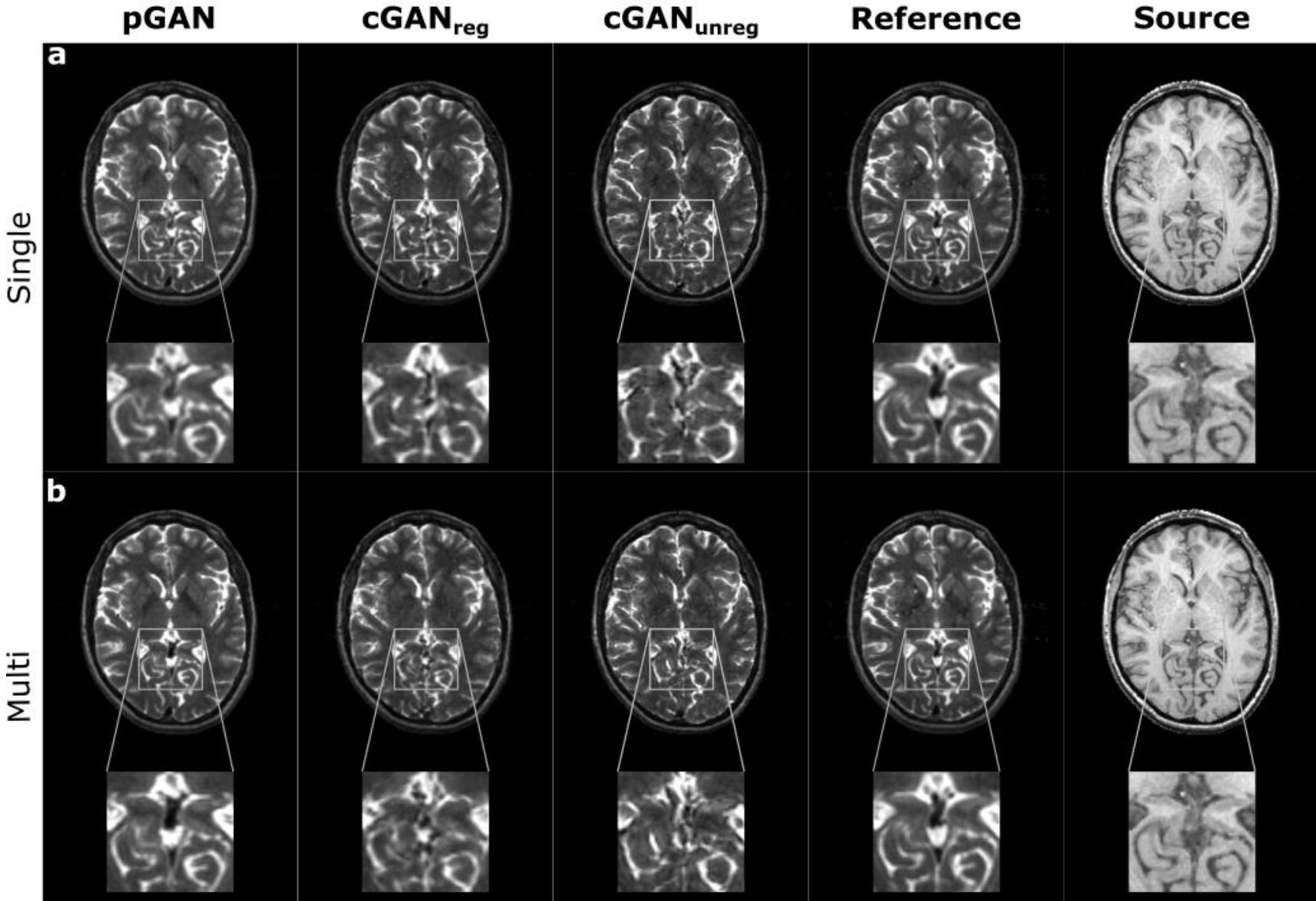
prediction



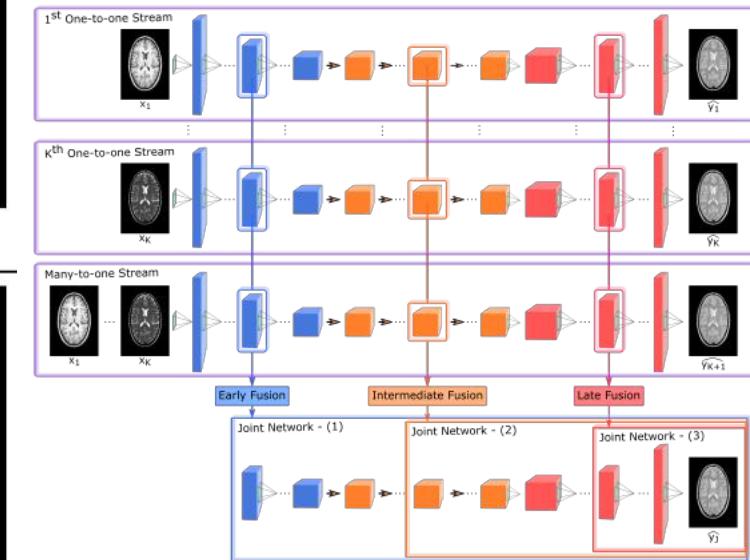
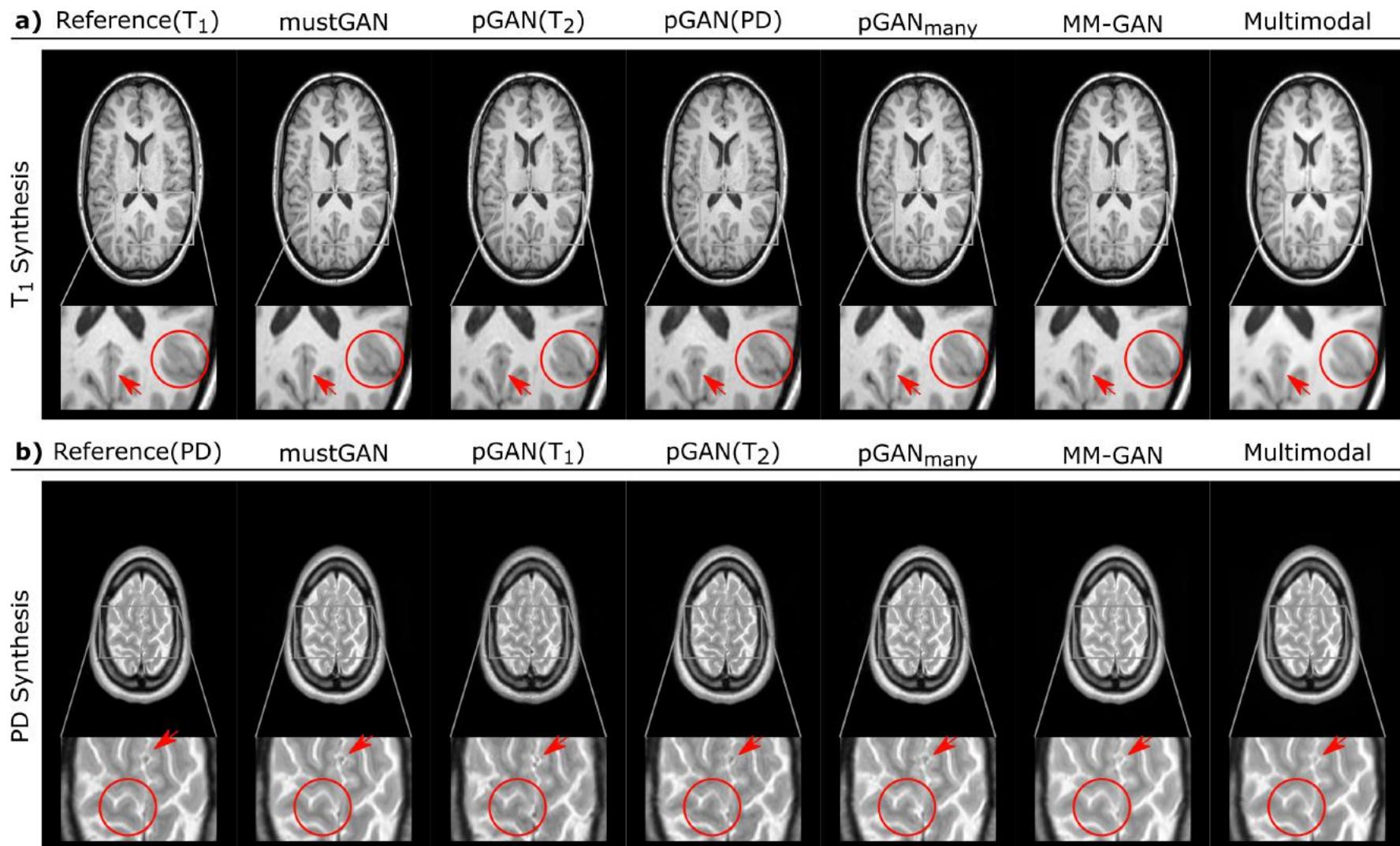
flowers



prediction



- Image Synthesis in Multi-Contrast MRI [UI Hassan Dar et al. 2019]



- Image Synthesis in Multi-Contrast MRI [Mahmut Yurt et al. 2021]

Single Image Super-Resolution

bicubic



SRResNet



SRGAN



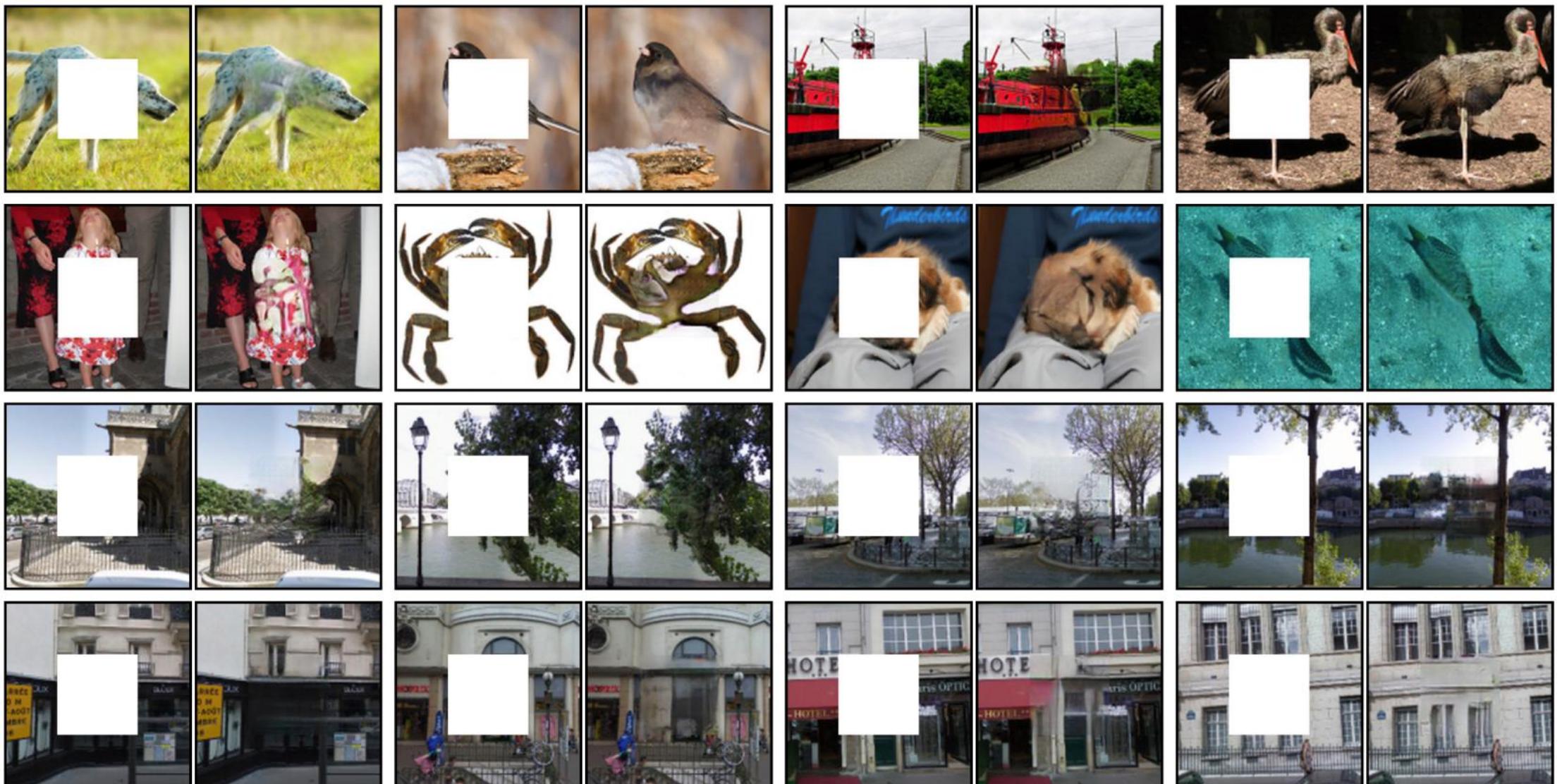
original



- Key idea: Combine content loss with adversarial loss

Image Inpainting

(Pathak et al., 2016)



- Key idea: Combine content loss with adversarial loss

Image Deblurring

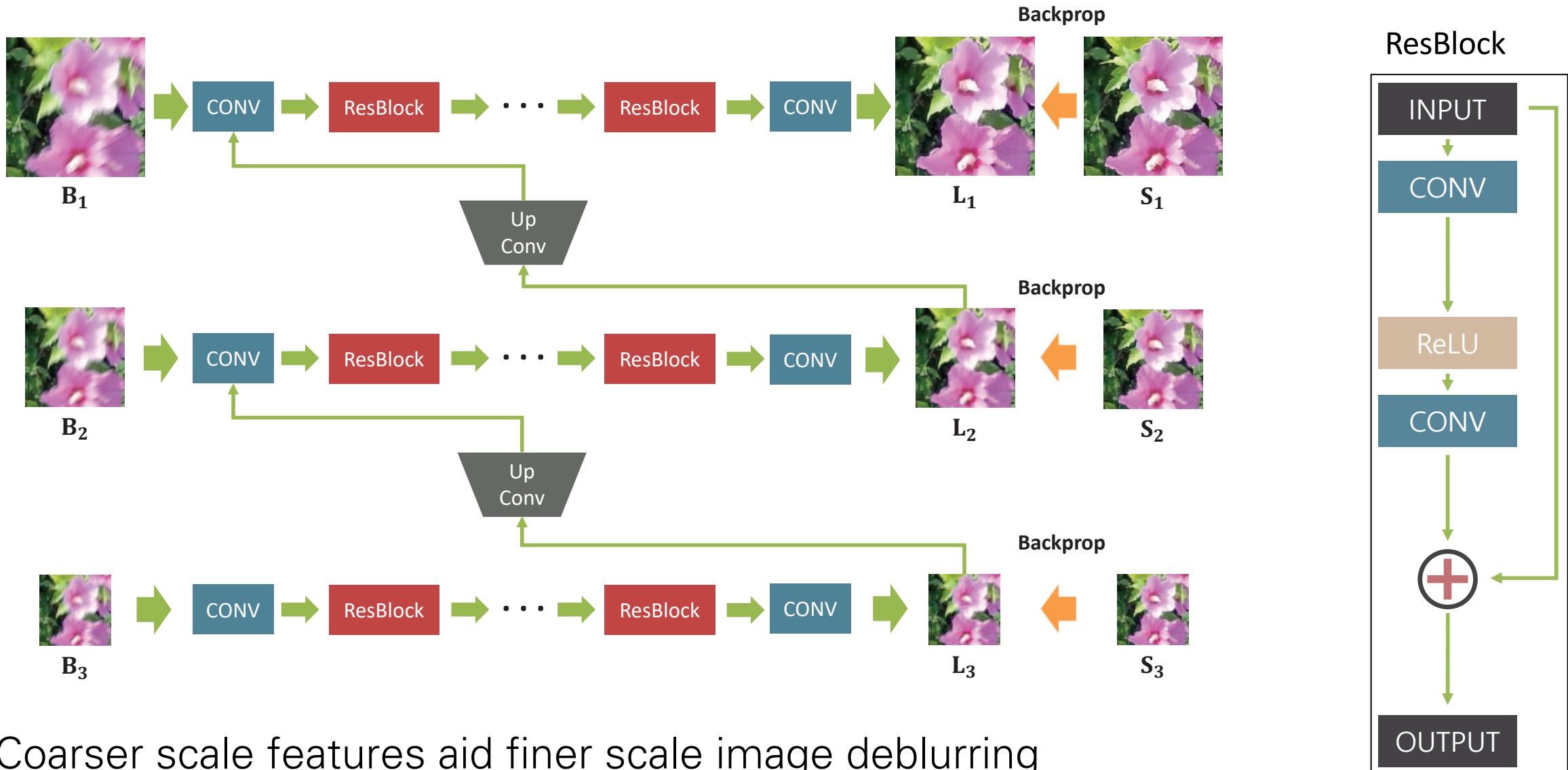
- non-uniform motion blur from a single blurry image.
- **Key idea:** Use multiscale CNNs to restore sharp images in an end-to-end manner

$$\mathcal{L}_{cont} = \frac{1}{2K} \sum_{k=1}^K \frac{1}{c_k w_k h_k} \|L_k - S_k\|^2$$

- can be interpreted as a kind of image to image translation 219
- An additional adversarial loss

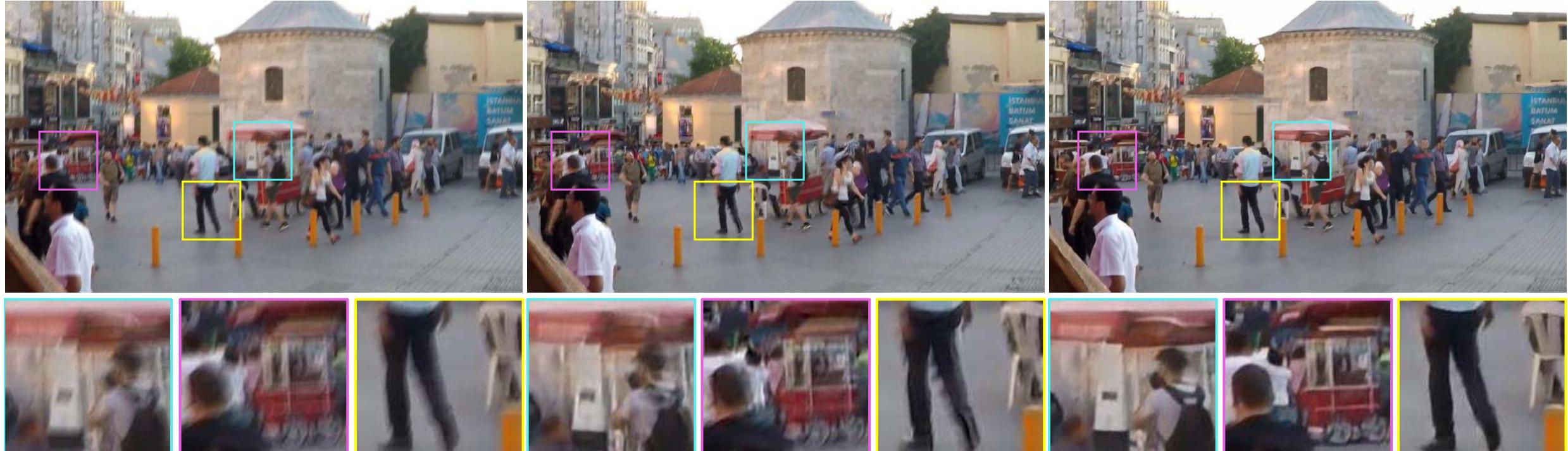
$$\begin{aligned}\mathcal{L}_{adv} = & \mathbb{E}_{S \sim p_{sharp}(S)} [\log D(S)] + \\ & \mathbb{E}_{B \sim p_{blurry}(B)} [\log(1 - D(G(B)))]\end{aligned}$$

Image Deblurring



- Coarser scale features aid finer scale image deblurring

Image Deblurring



Blurred images

Sun et al., CVPR 2015

Nah et al., CVPR 2017

Image Deblurring

- non-uniform motion blur from a single blurry image.
- **Key idea:** Use a conditional GAN and content loss

$$\mathcal{L} = \underbrace{\mathcal{L}_{GAN}}_{\substack{adv\ loss \\ total\ loss}} + \underbrace{\lambda \cdot \mathcal{L}_X}_{content\ loss} \quad \mathcal{L}_{GAN} = \sum_{n=1}^N -D_{\theta_D}(G_{\theta_G}(I^B)) \quad \mathcal{L}_X = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^S)_{x,y} - \phi_{i,j}(G_{\theta_G}(I^B))_{x,y})^2$$

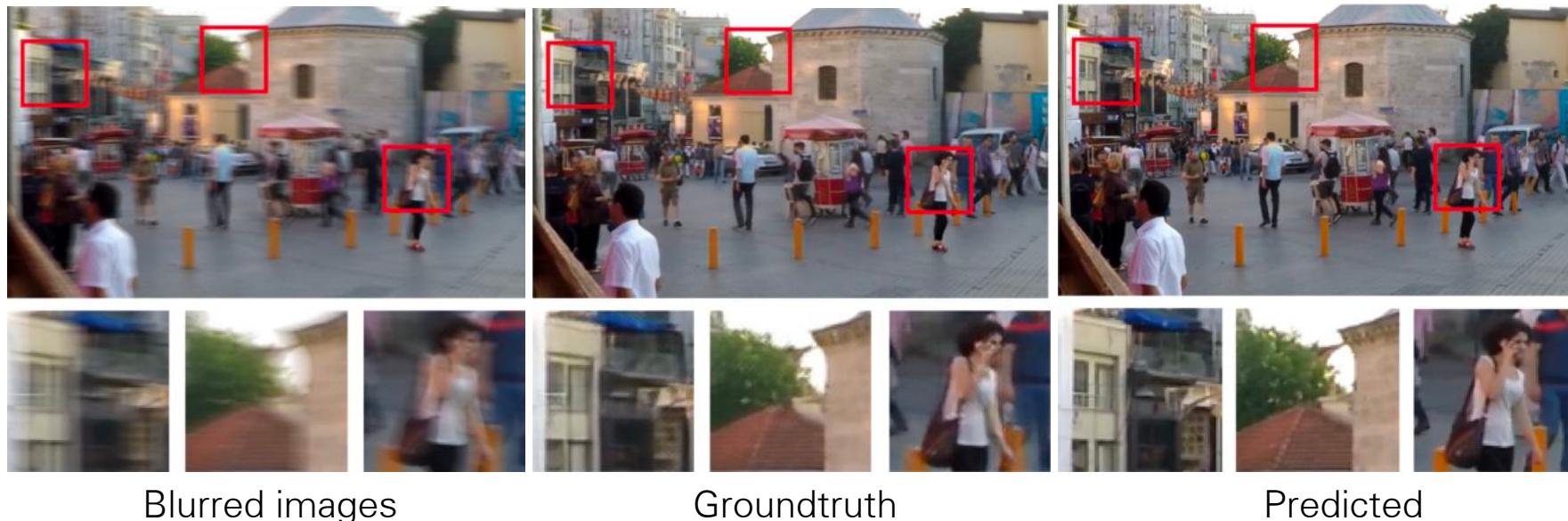
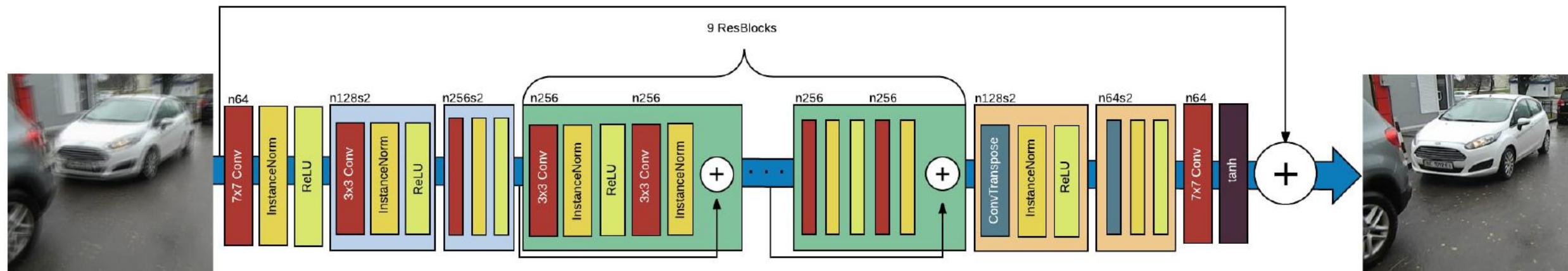


Image Deblurring

- Key idea: An image to image translation model that learns the residual to sharpen the blurred image



223

Image Deblurring

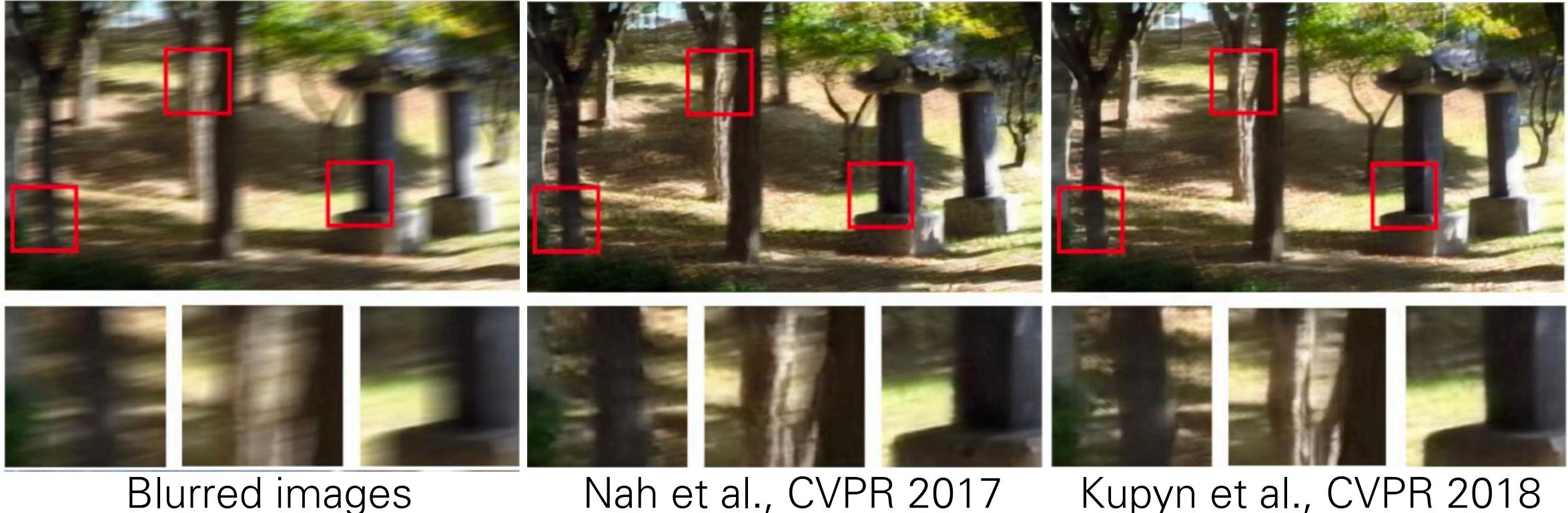


Image Deblurring



Blurred images

Nah et al., CVPR 2017

Kupyn et al., CVPR 2018

Image Deblurring

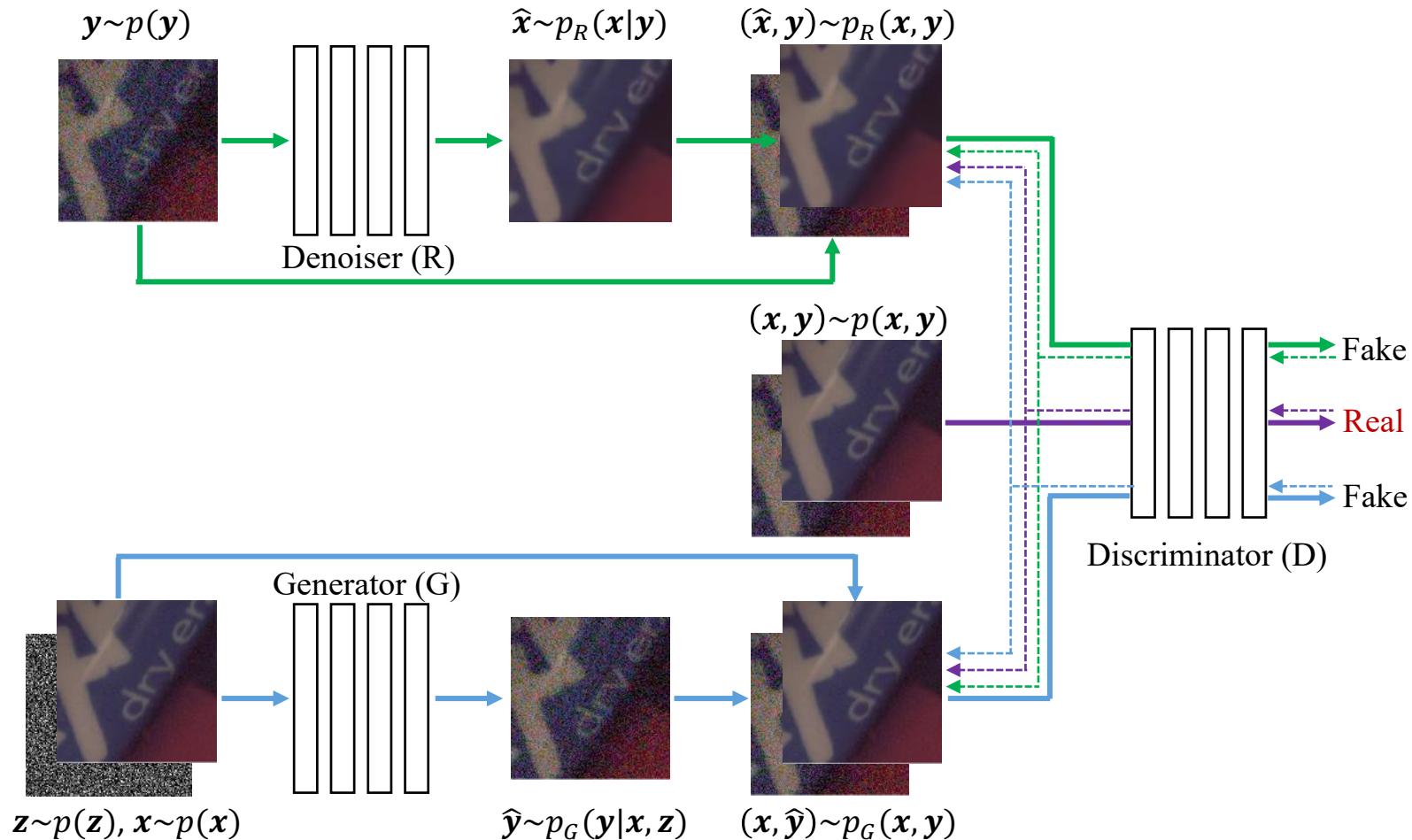


Blurred images

Nah et al., CVPR 2017

Kupyn et al., CVPR 2018

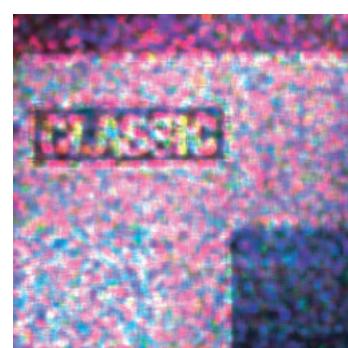
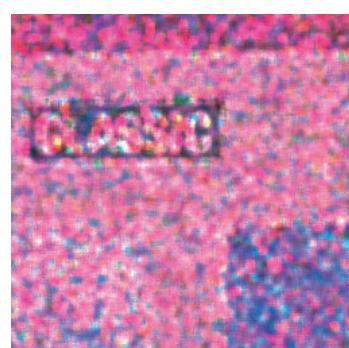
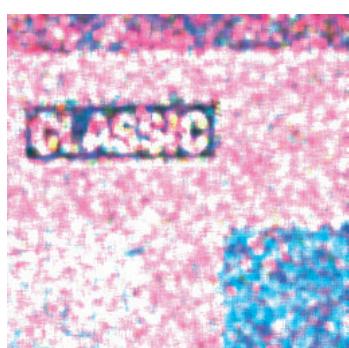
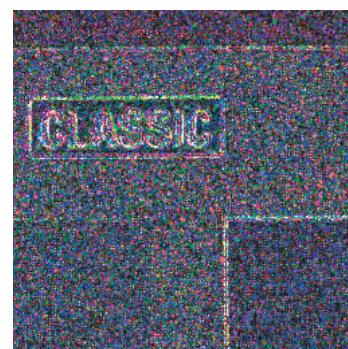
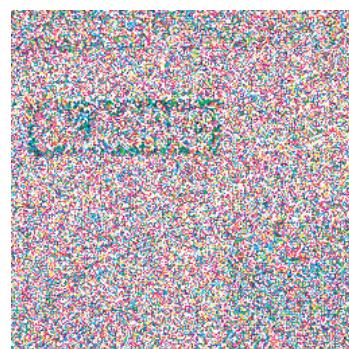
Image Denoising



- Key idea: simultaneously deal with the noise removal and noise generation tasks.

Image Denoising

Generated Noisy Images



(a) Real

(b) CBDNet

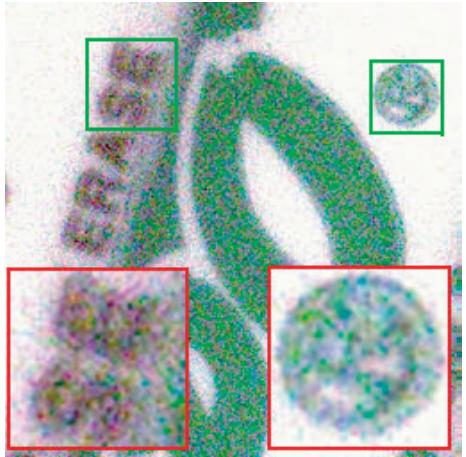
(c) ULRD

(d) GRDN

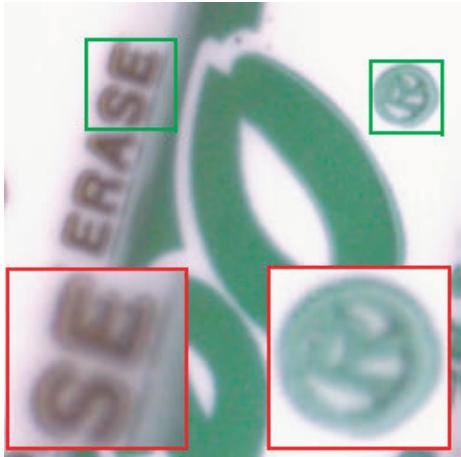
(e) DANet

Image Denoising

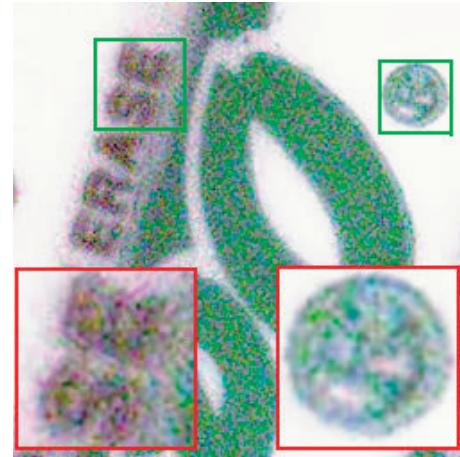
Denoising results



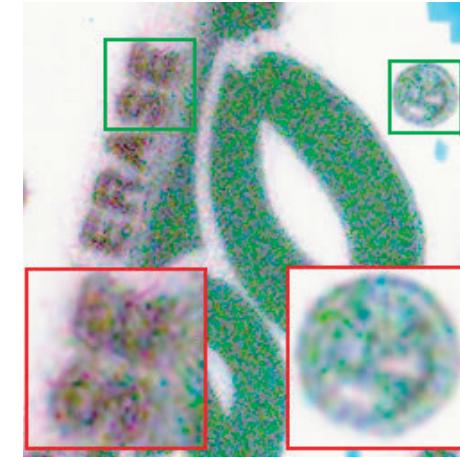
(a): Noisy



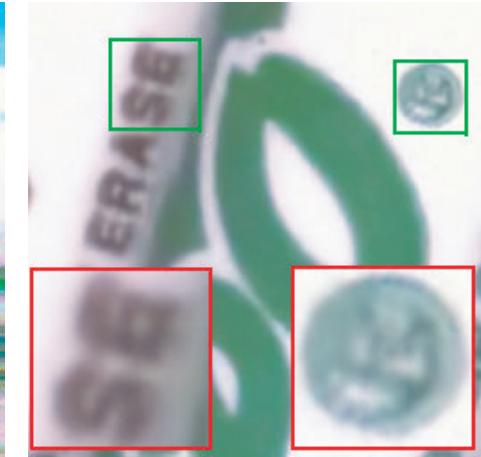
(b): GroundTruth



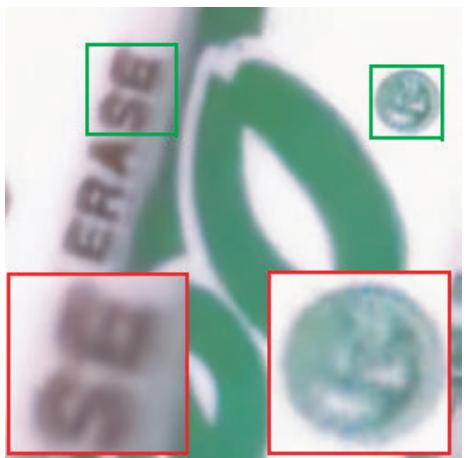
(c): BM3D



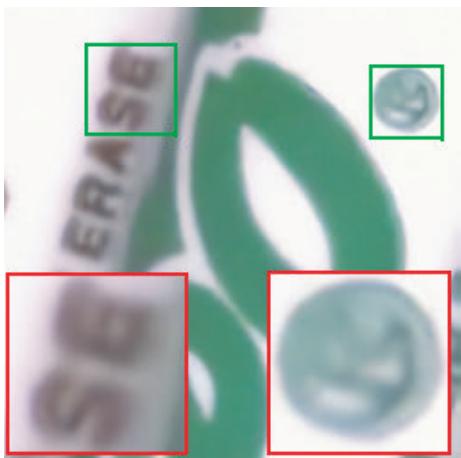
(d): WNNM



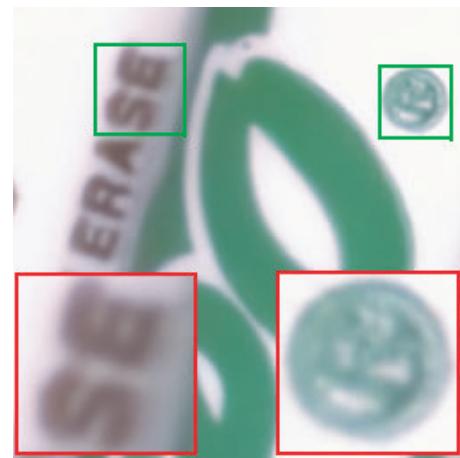
(e): DnCNN



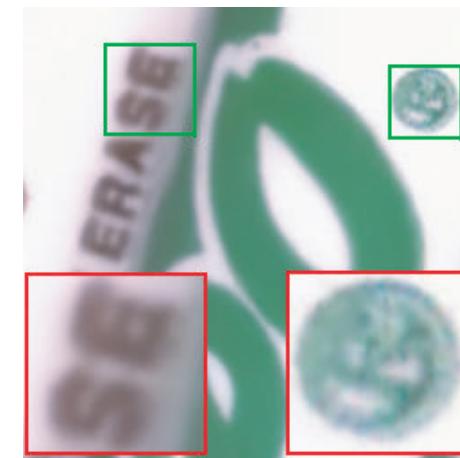
(f): CBDNet



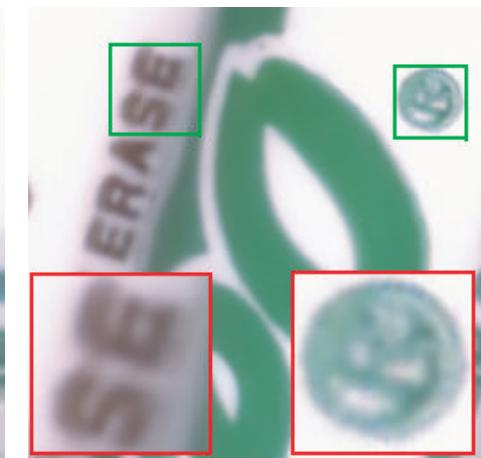
(g): RIDNet



(h): VDN

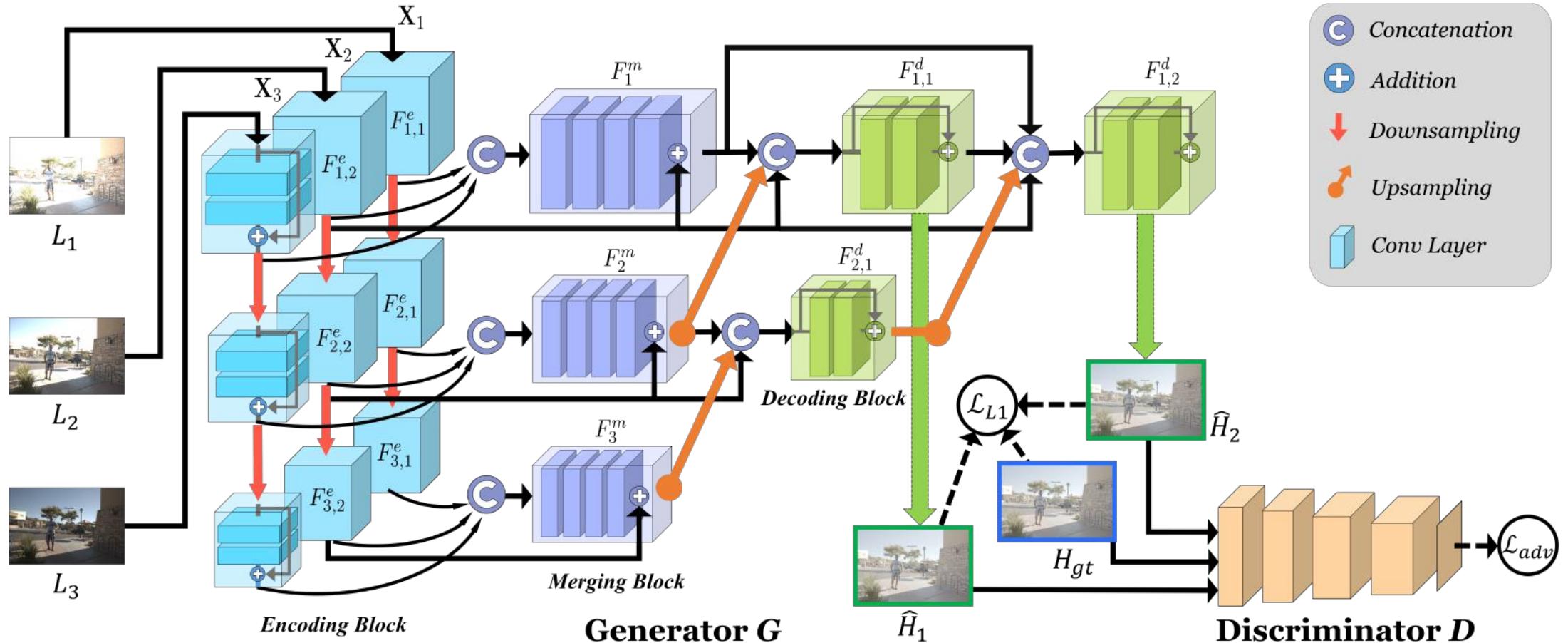


(i): DANet



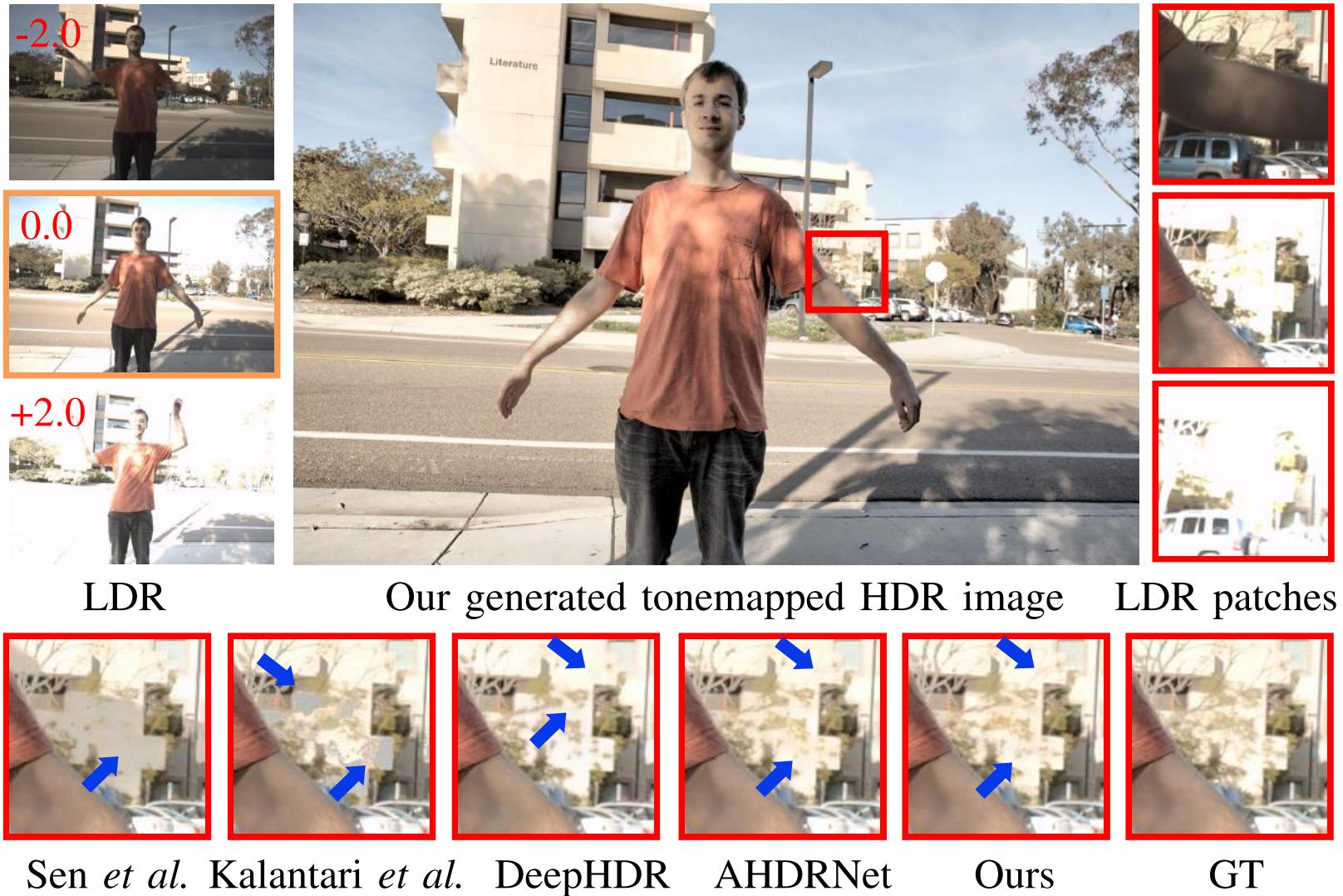
(j): DANet+

High Dynamic Range Imaging



- Key idea: Incorporate adversarial learning to be able to produce faithful information in the regions with missing content.

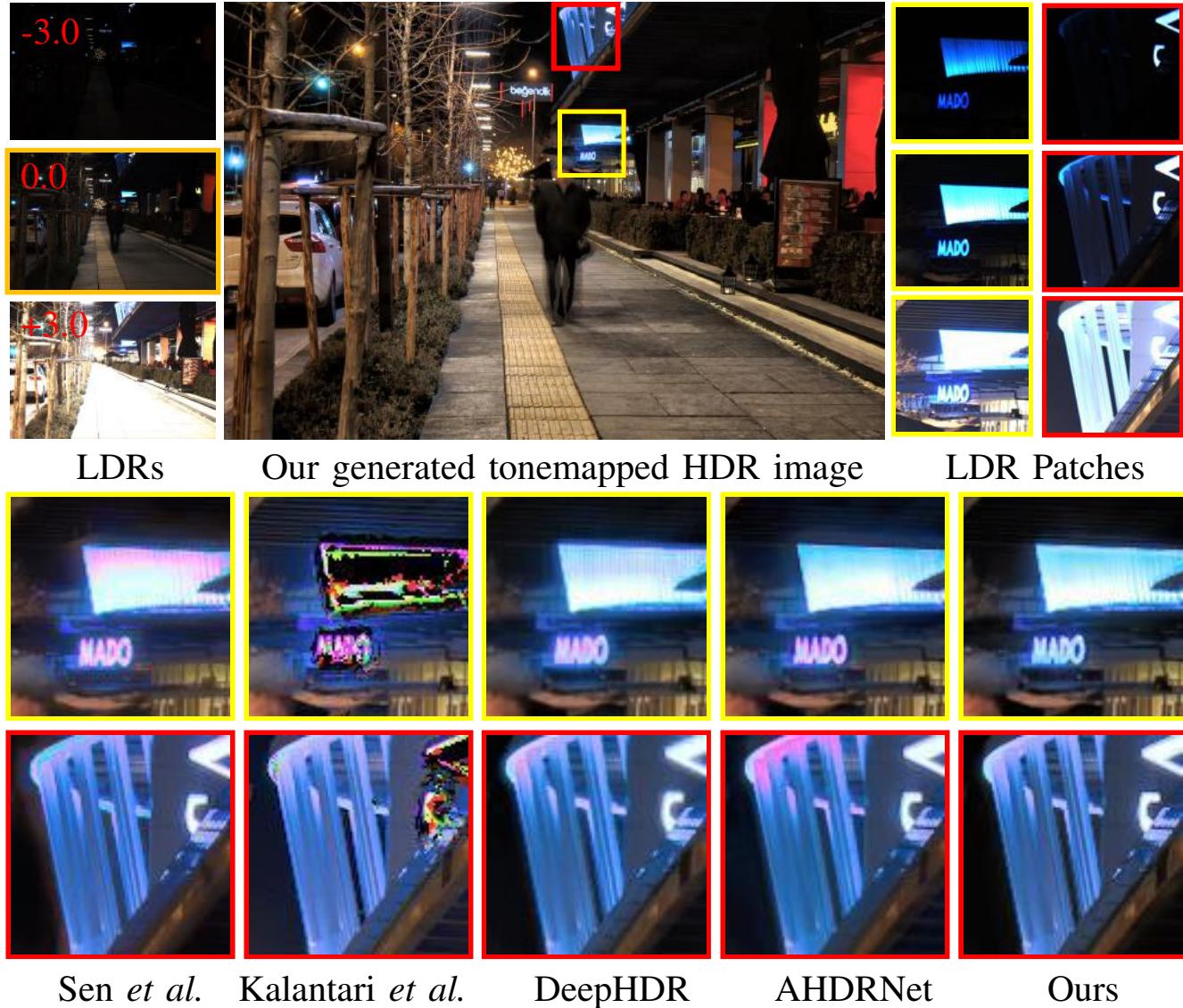
High Dynamic Range Imaging



High Dynamic Range Imaging



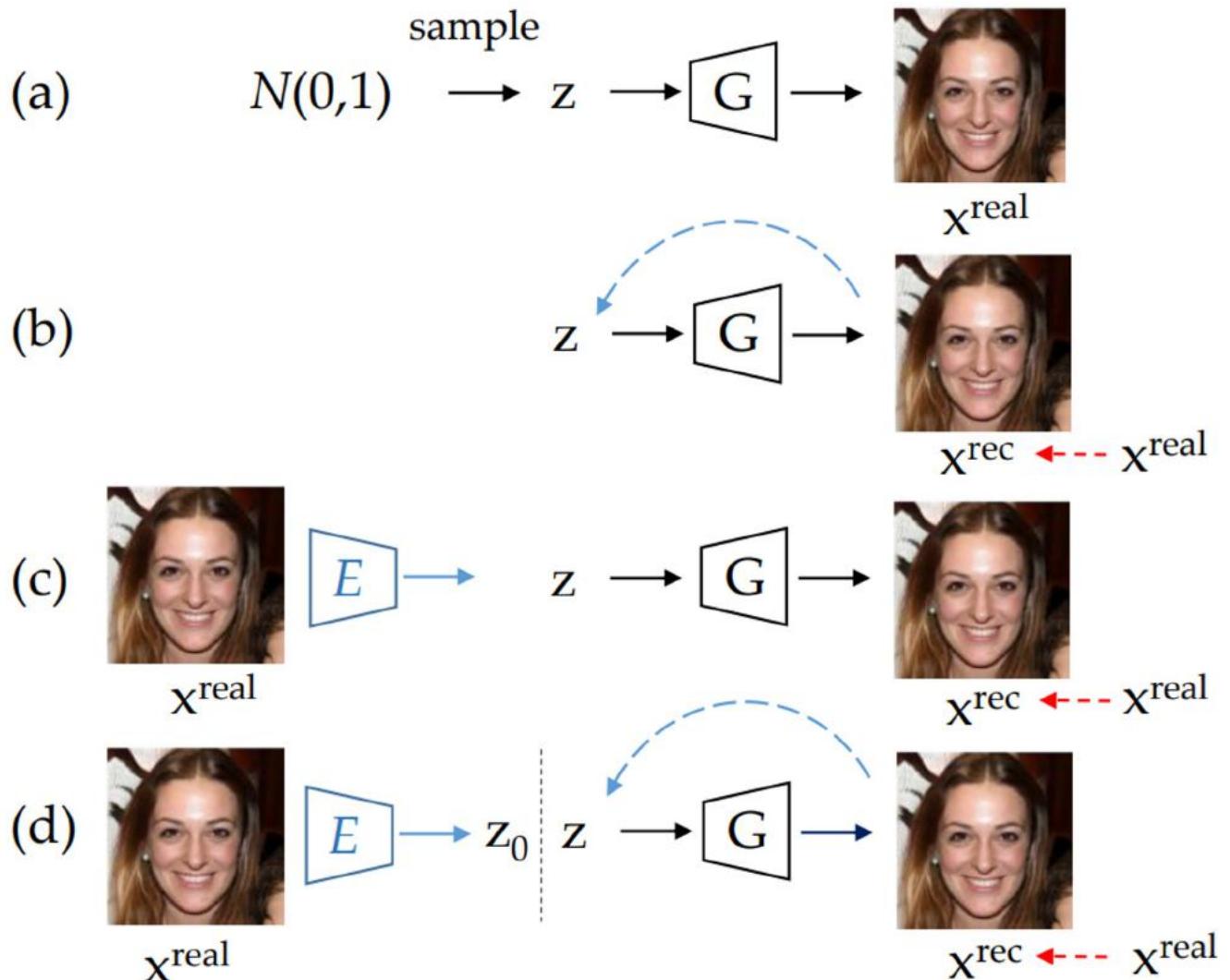
High Dynamic Range Imaging



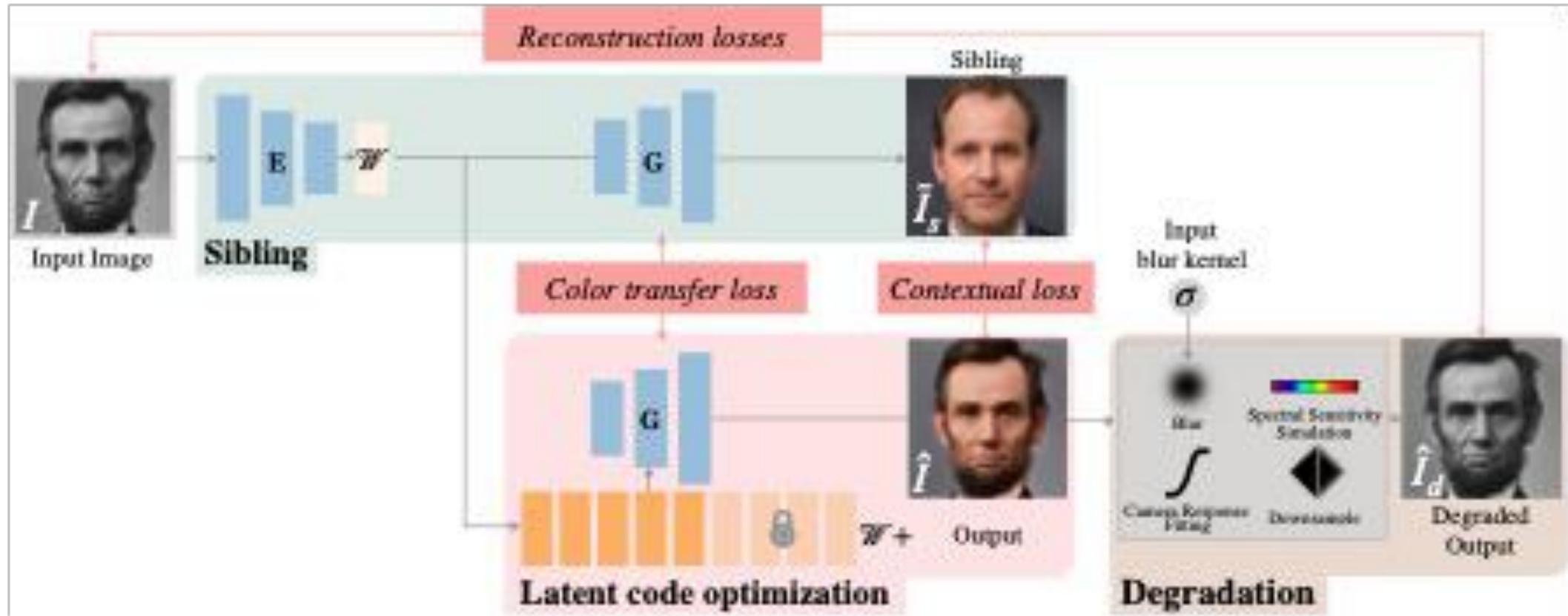
GAN Inversion

3 methods of inversion:

- Optimization-based (b)
- Learning-based (c)
- Hybrid (d)

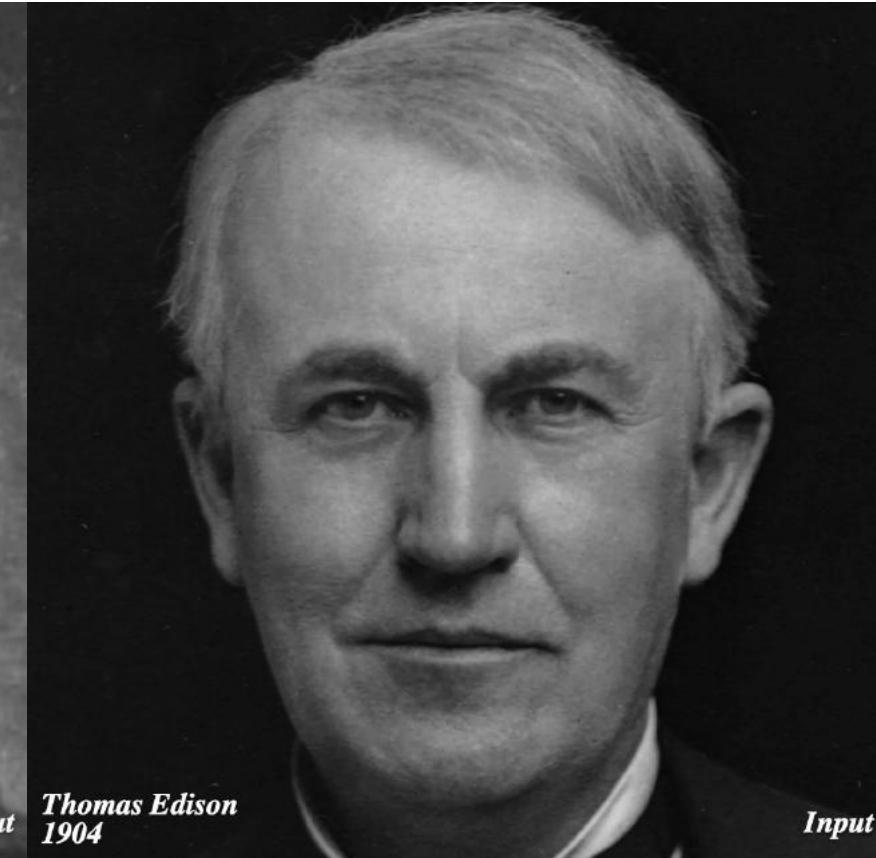


Time-travel Rephotography

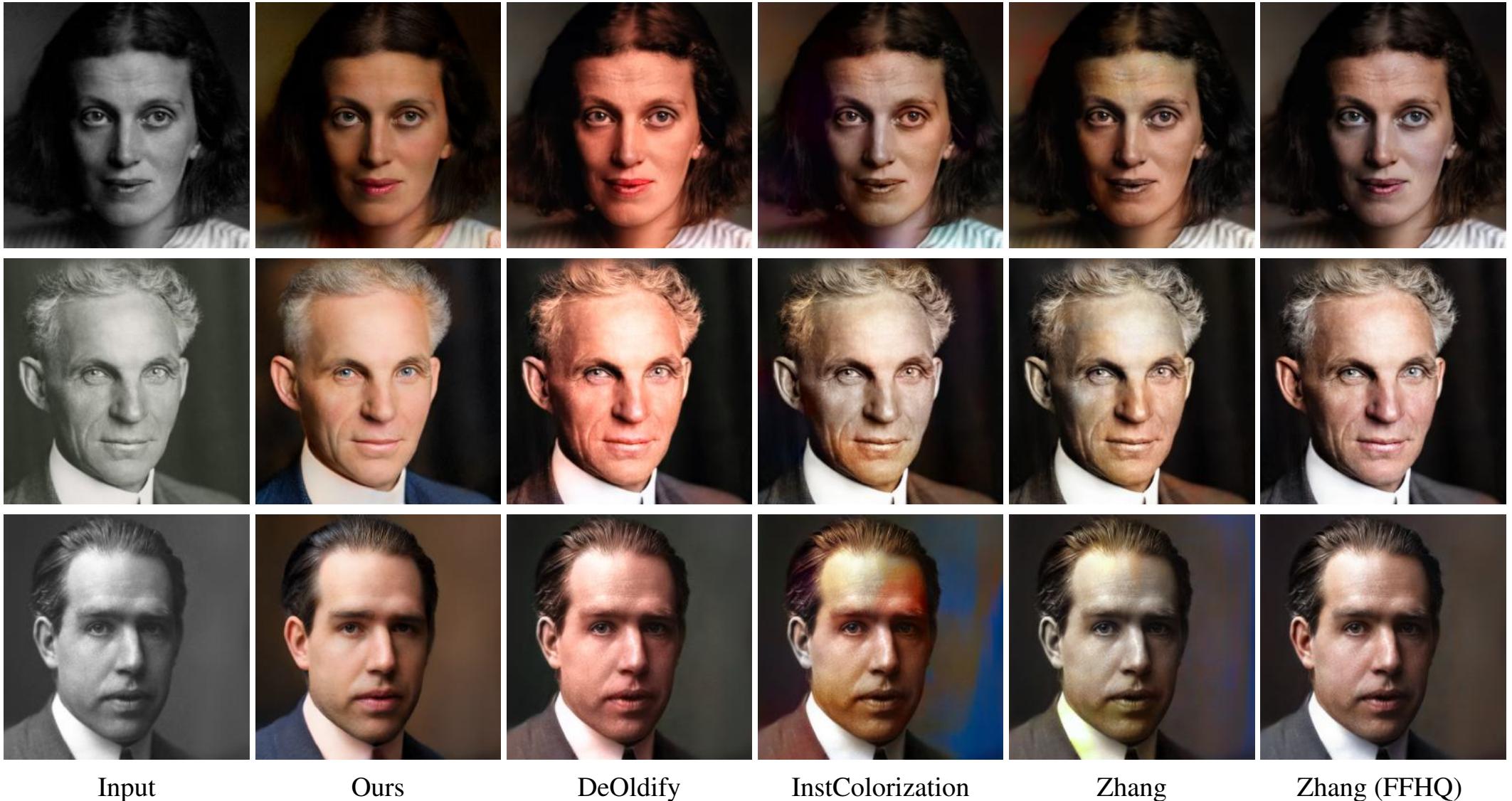


- Key idea: Use the StyleGAN2 framework to project old photos into the space of modern high-resolution photos for enhancing their quality.

Time-travel Rephotography

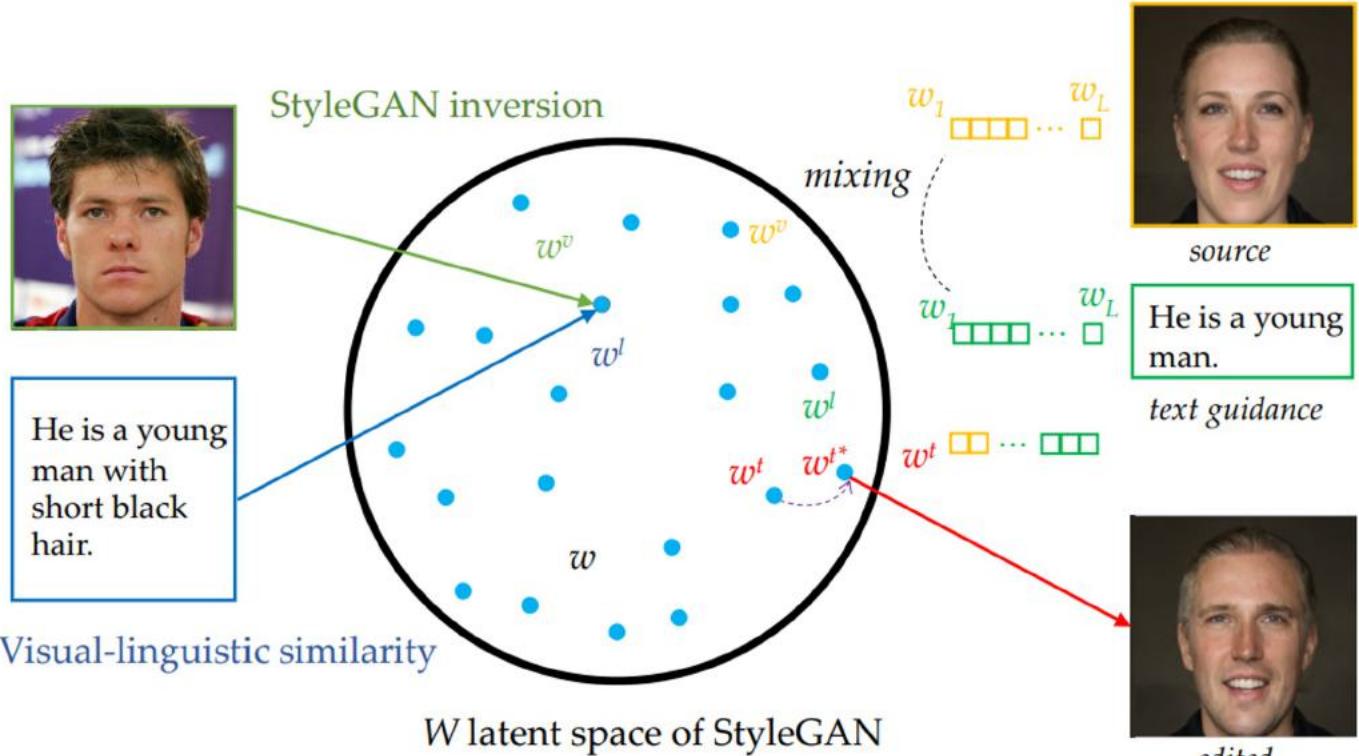


Time-travel Rephotography



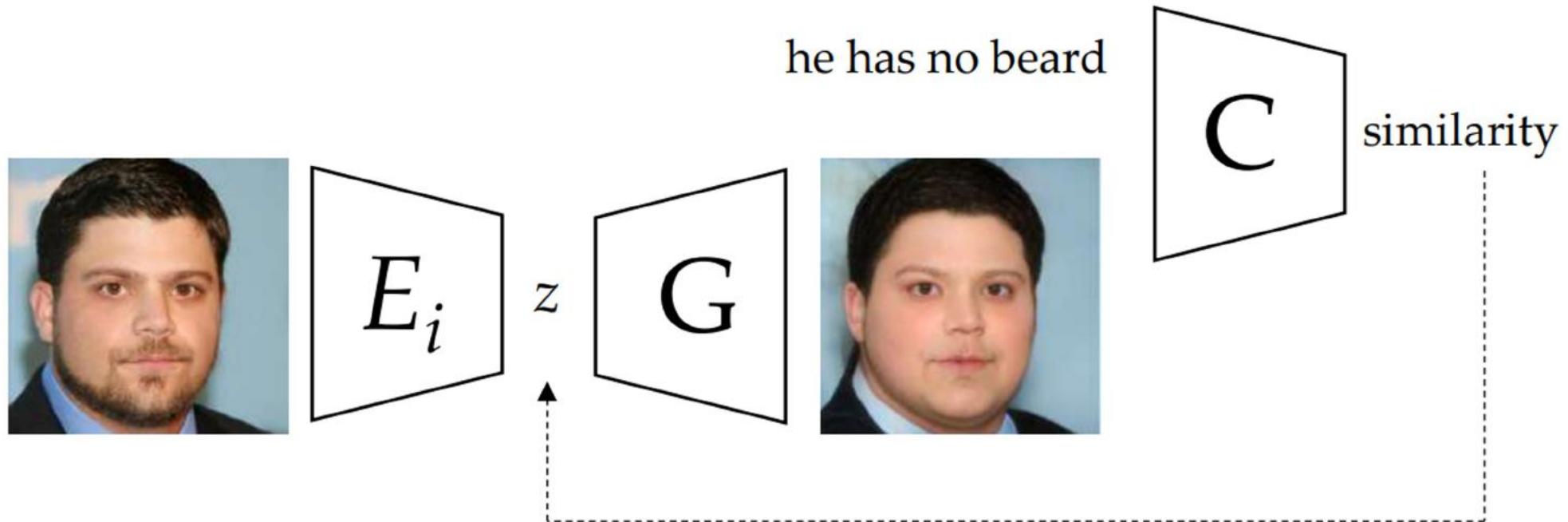
TediGAN

- A recent inversion based manipulation model.
- They proposed two different approaches.
- In their first approach, they learn a common visual-linguistic semantic space.
- They train a text encoder such that both modalities are embedded to the same space



TediGAN

- Their second method is optimization based.
- They use CLIP to provide a signal to directly optimize the latent code



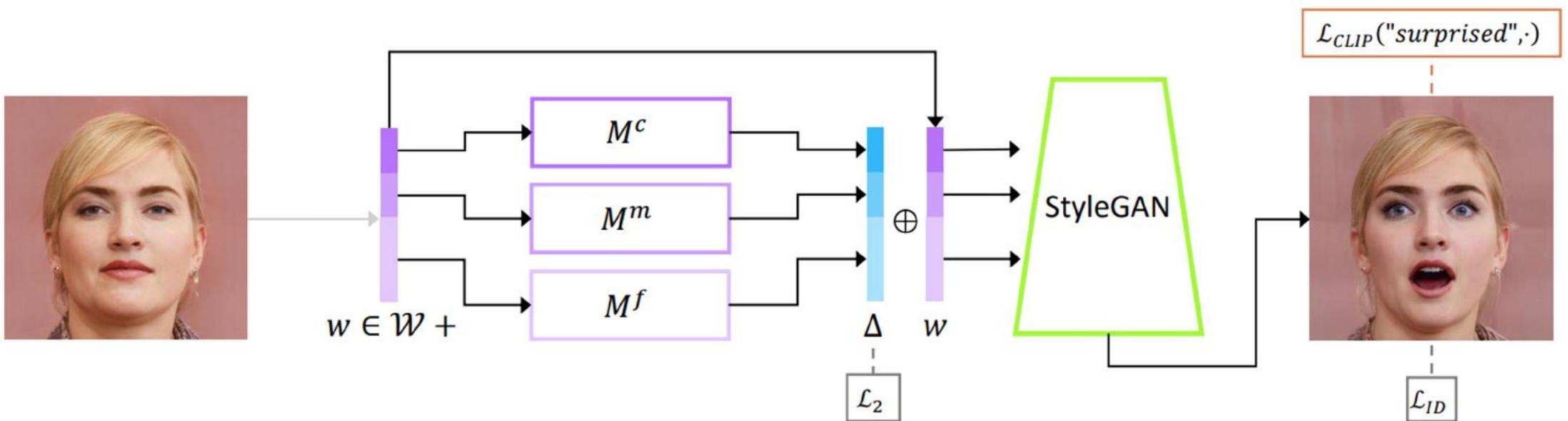
StyleCLIP

- StyleCLIP is another inversion based manipulation model
- Direct optimization
- Similar to TediGAN, they use CLIP to optimize the latent code directly
- They include a term to preserve identity

$$\arg \min_{w \in \mathcal{W}^+} D_{\text{CLIP}}(G(w), t) + \lambda_{\text{L2}} \|w - w_s\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w)$$

StyleCLIP

- Images are inverted to latent code and residuals (delta in the figure).
- Textual input is not used during inversion, it is only used in the loss
- Required to train different mappers for different text prompts.



Our work in progress

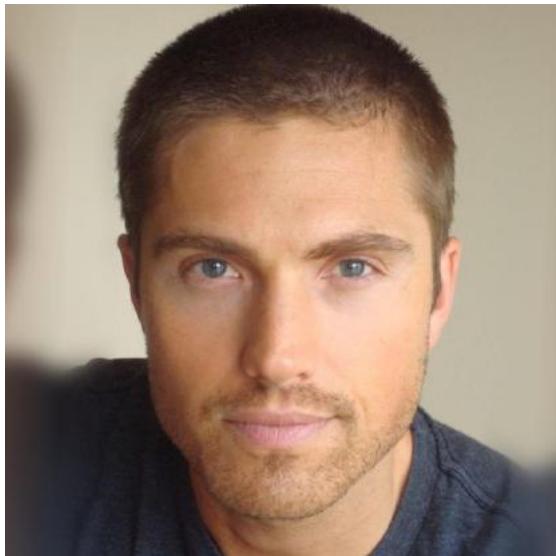
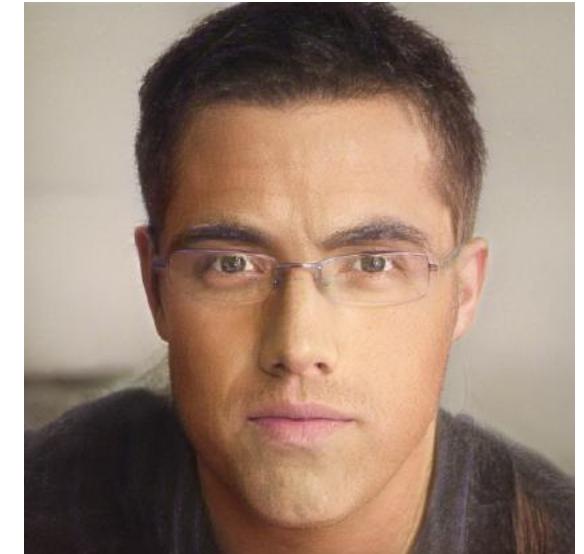
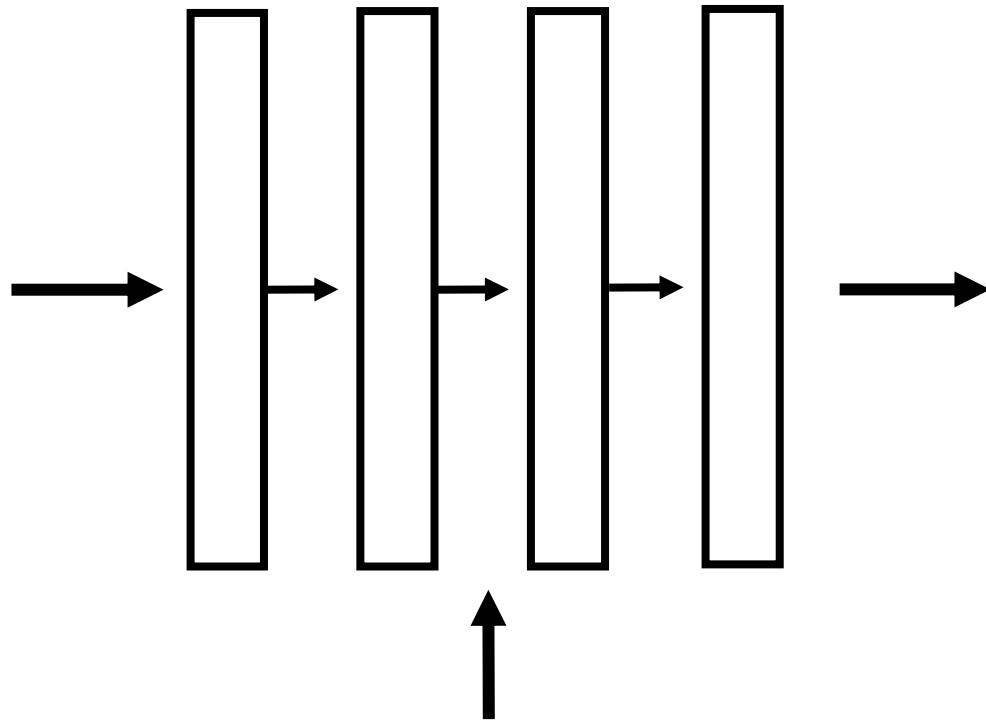


image **x**



manipulated
image **y**

- **Input:** input image + target description
- Output: manipulated image

“he is tanned and has eye glasses.”

Target description **t**

Our work in progress

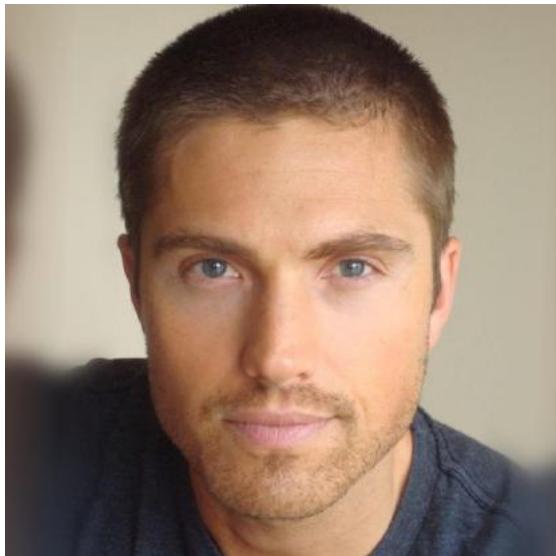
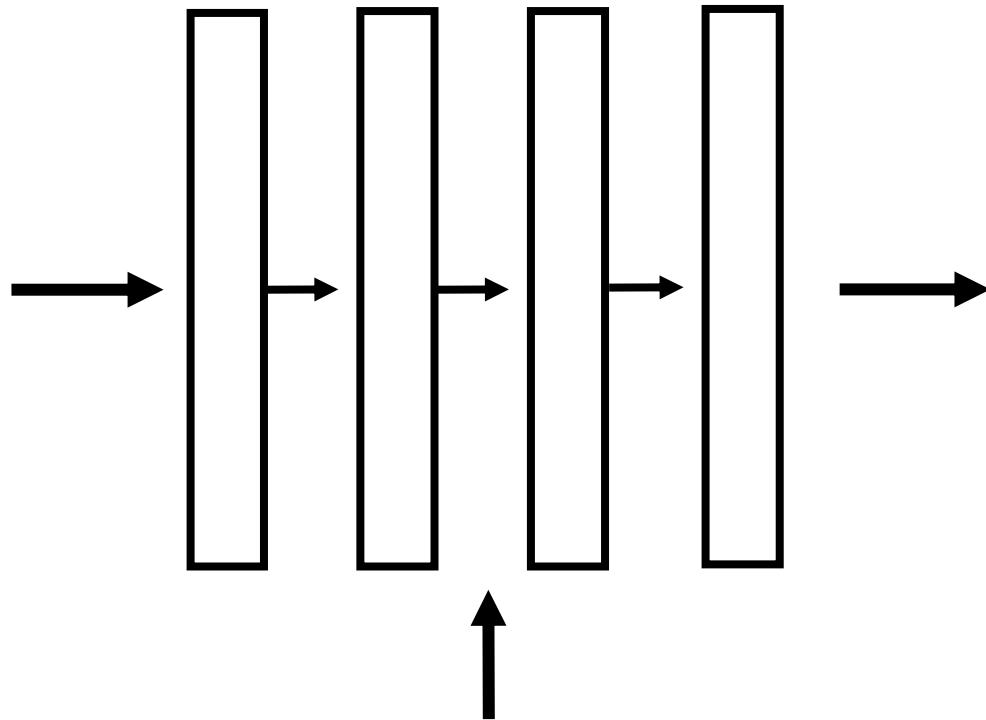


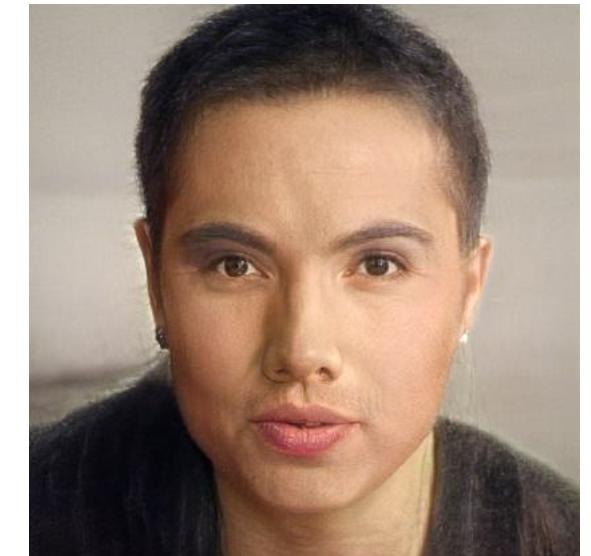
image **x**



“an Asian woman”

Target description **t**

- **Input:** input image
+ target description
Output: manipulated image

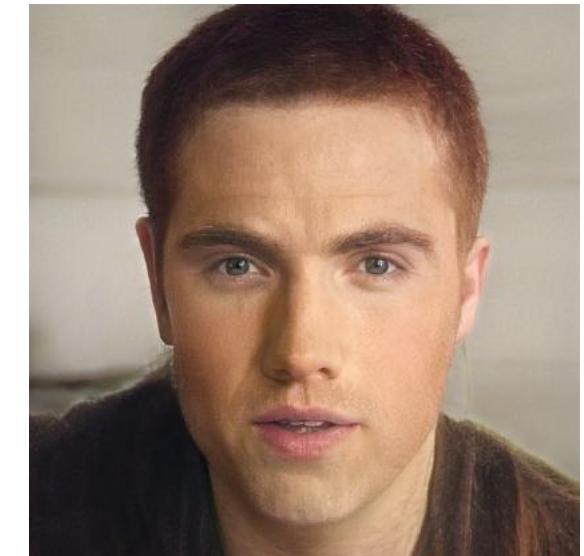
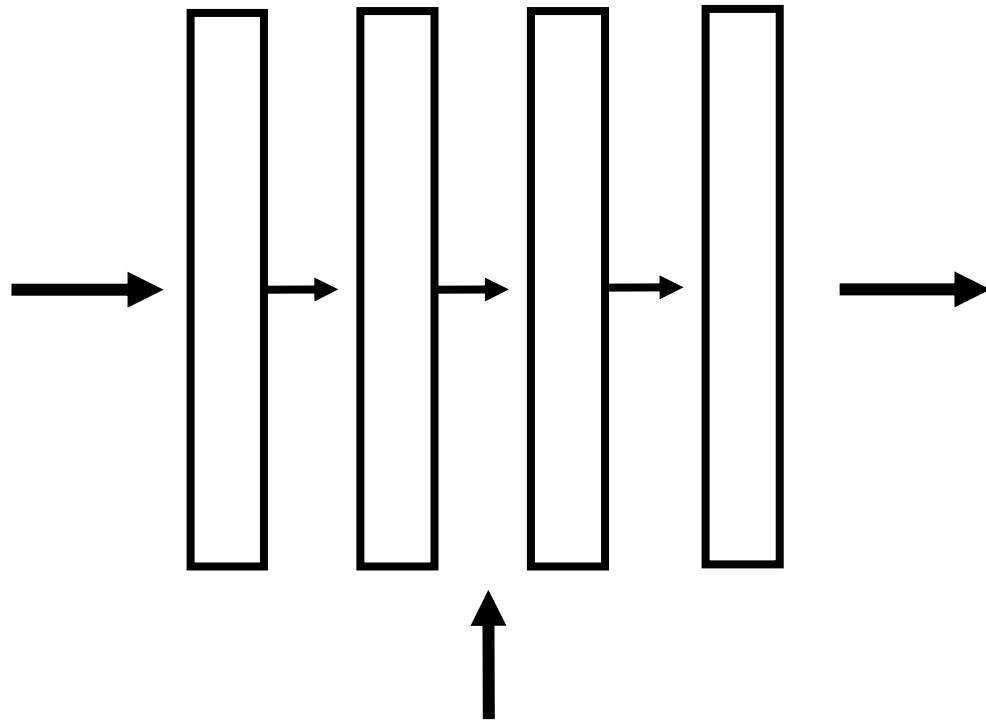


manipulated
image **y**

Our work in progress



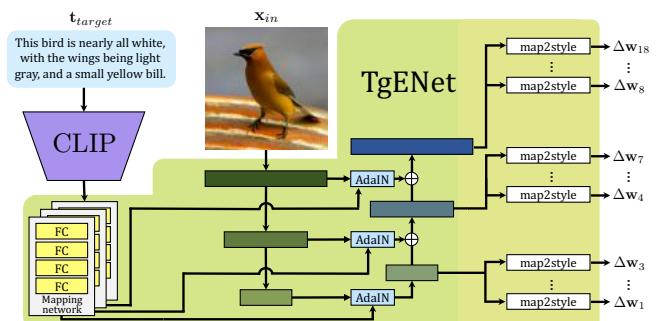
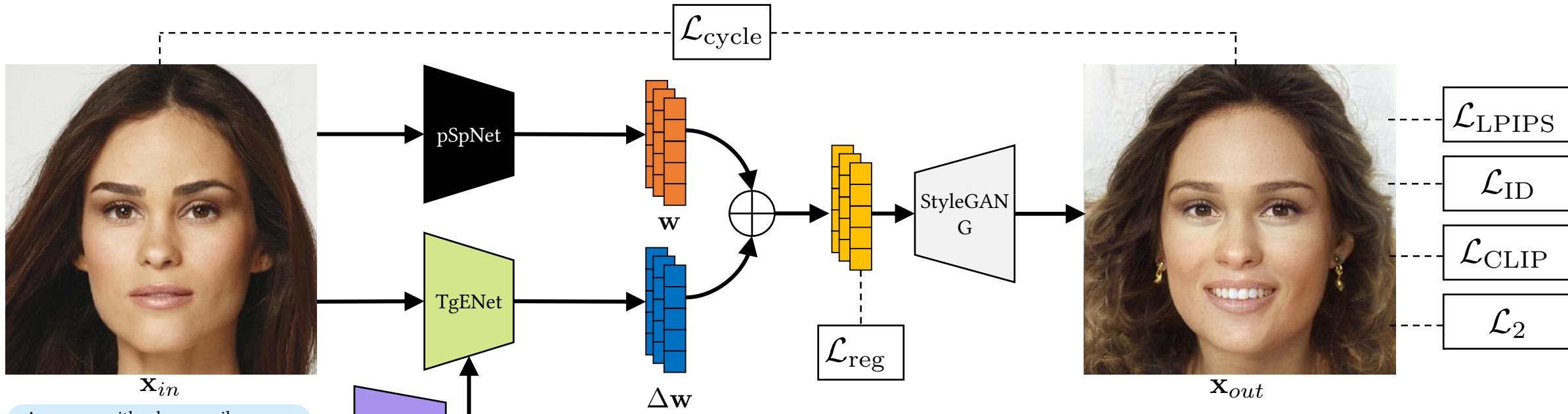
image **x**



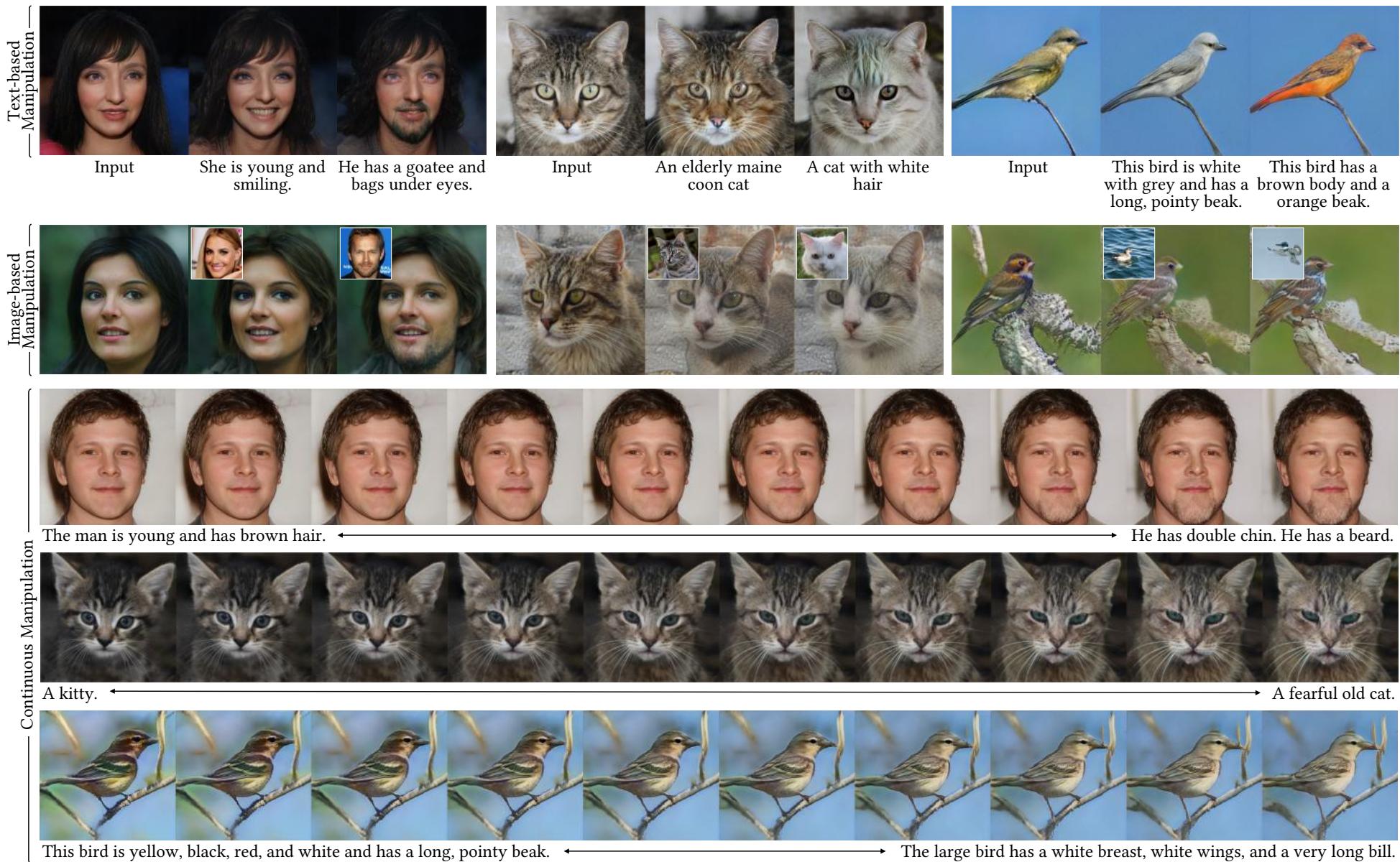
manipulated
image **y**

- **Input:** input image + target description
- Output: manipulated image
- “he has read hair”
- Target description **t**

Our work in progress



Our work in progress



Our work in progress



The person has brown hair. She wears lipstick.



A grumpy cat with ginger hair.



This bird has a blue crown with a blue belly and blue coverts.



The man has pointy nose, goatee, and sideburns.



A scottish fold cat with grey hair.



This bird is multi colored in color, with a stubby beak.



She wears lipstick. She has bags under eyes, and big lips.
She is young.



An elderly persian cat.



A small bird with a black throat and yellow orange crown
and similar color in the breast and side region.

Our work in progress

Original



Smiling



Smiling+Chubby



Smiling+Chubby+Beard



Original



Curly Hair



Silver Hair



Facial Hair



Our work in progress



This old woman has big lips, pale skin, and gray hair.



She wears eyeglasses.



She is smiling. She has blonde hair.



He has a beard. He wears necktie.

Our work in progress



This young woman has pointy nose, and big lips. ← → This person has bags under eyes and wears heavy makeup. She is smiling.



He is attractive and has bushy eyebrows, black hair, and big nose. ← → She is wearing lipstick. She is smiling, and young and has brown hair.



A egyptian kitty with brown hair. ← → A fearful elderly british shorthair cat.



An old cat. ← → An old cat with ginger hair.

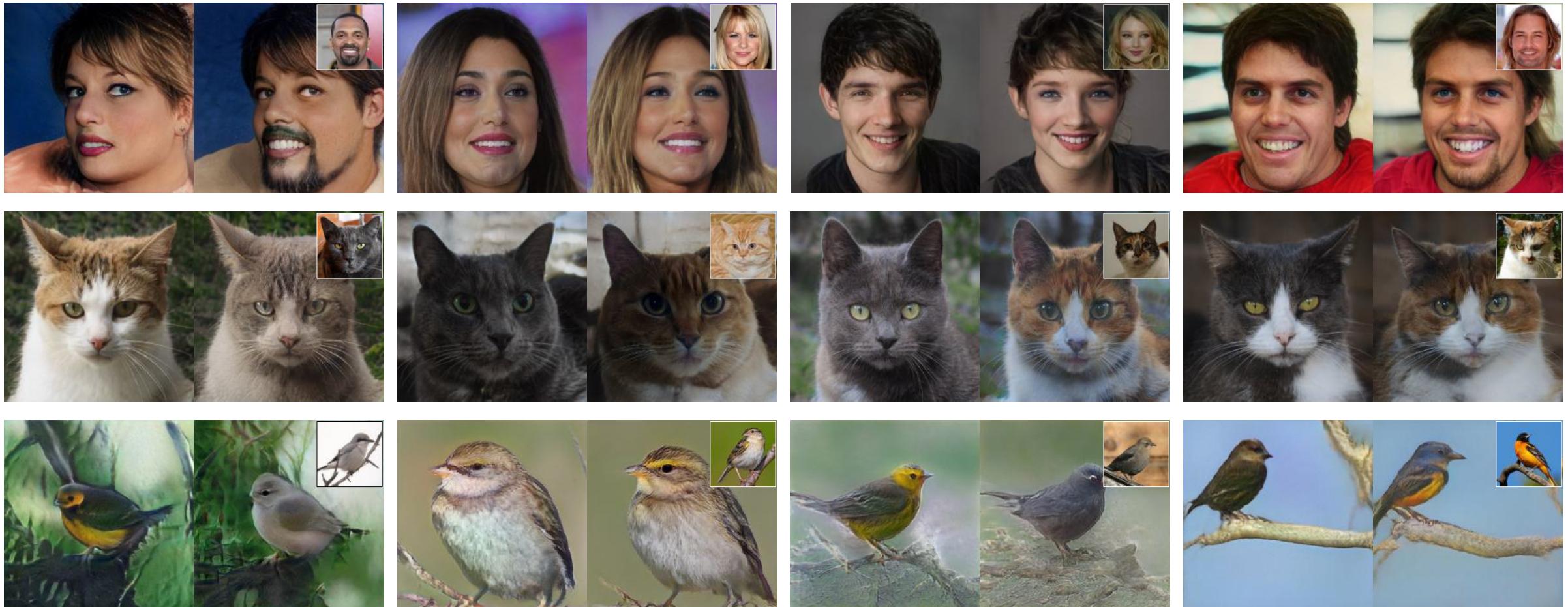


This bird has wings that are grey and white and has a yellow belly ← → This bird has red and black wings with a red crown.

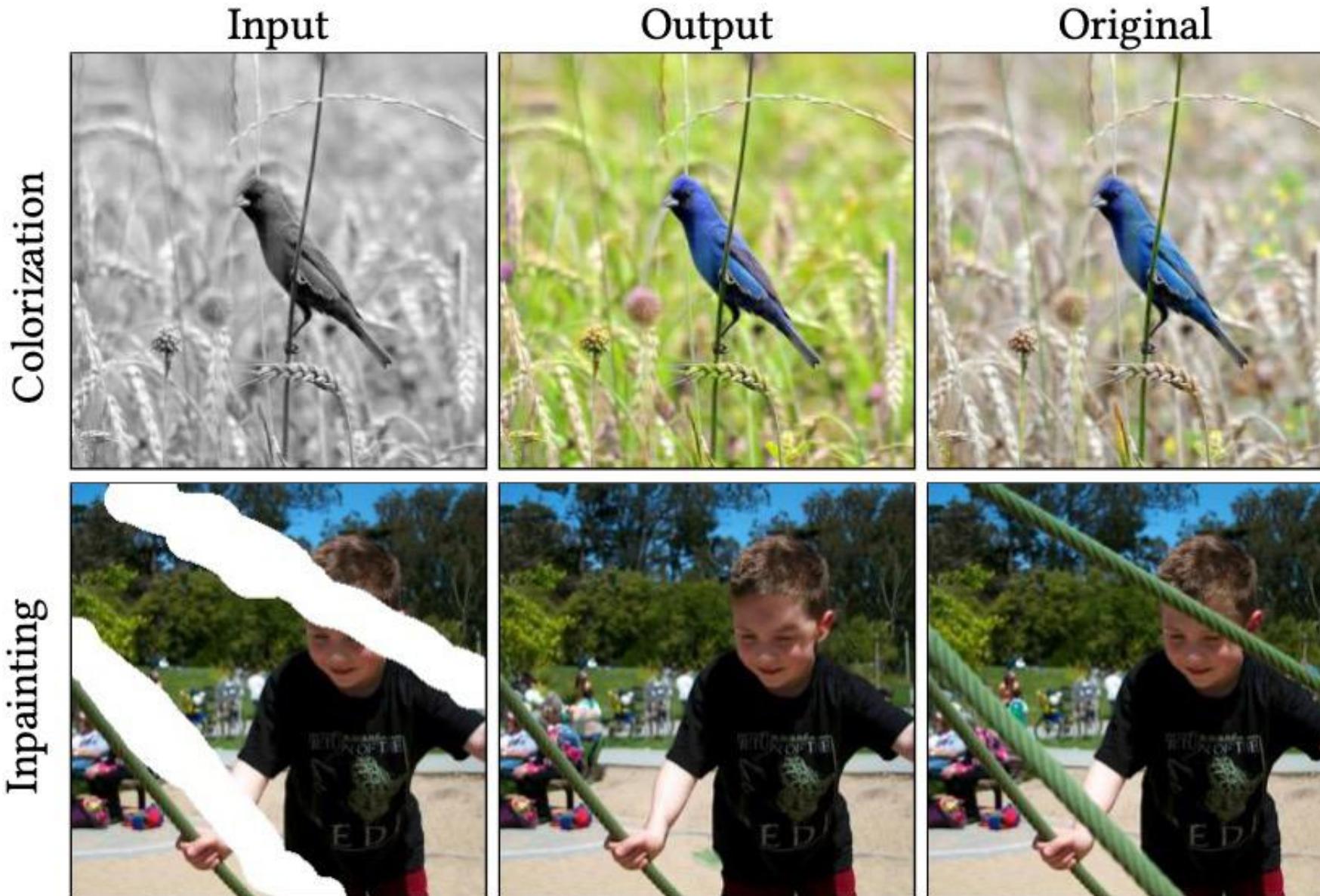


This bird has a brown crown, white forehead, grey coverts and secondaries and a white chest. ← → This bird has wings that are black and has a yellow belly.

Our work in progress

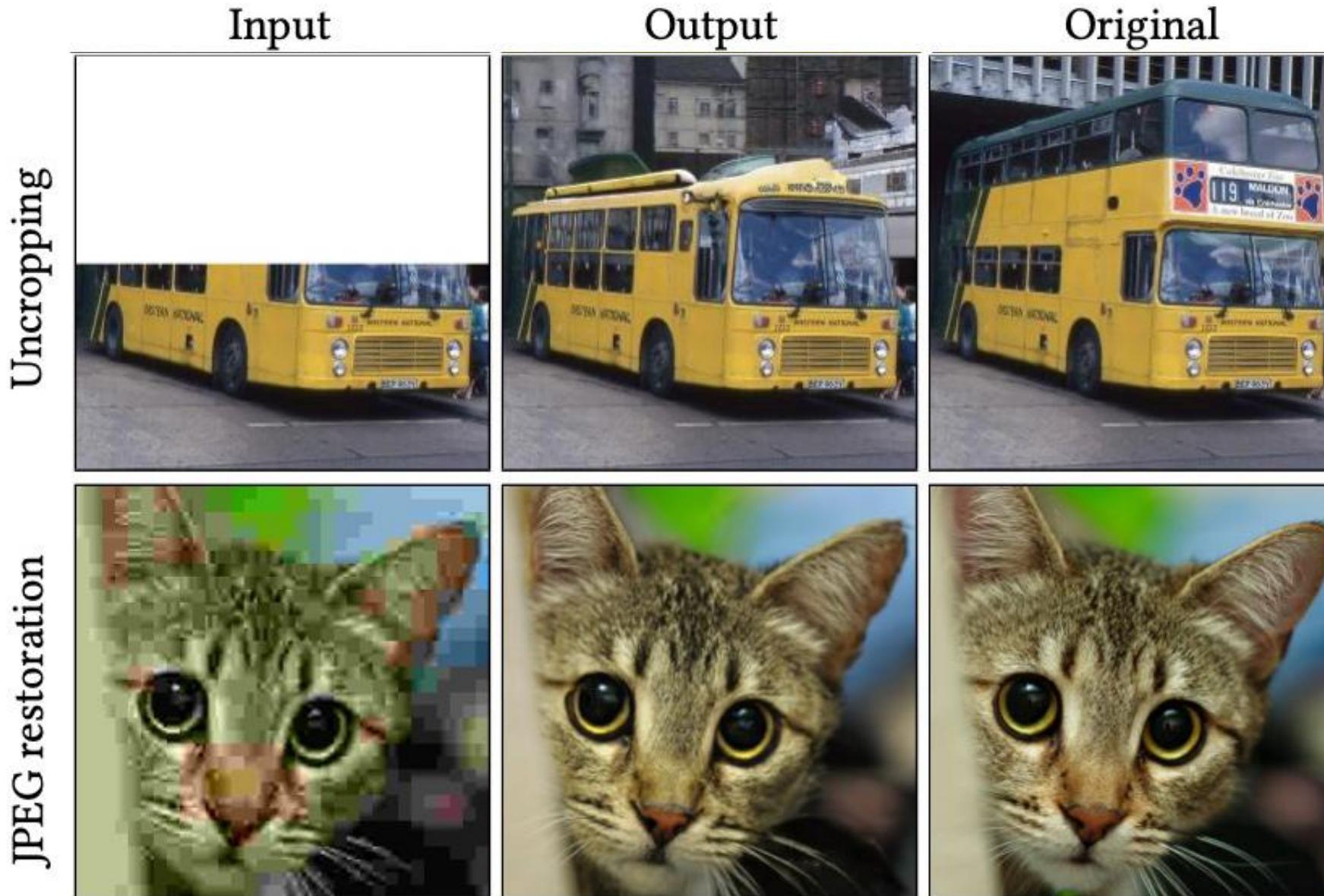


Palette: Image-to-Image Diffusion Models



[Saharia et al. 2022]

Palette: Image-to-Image Diffusion Models



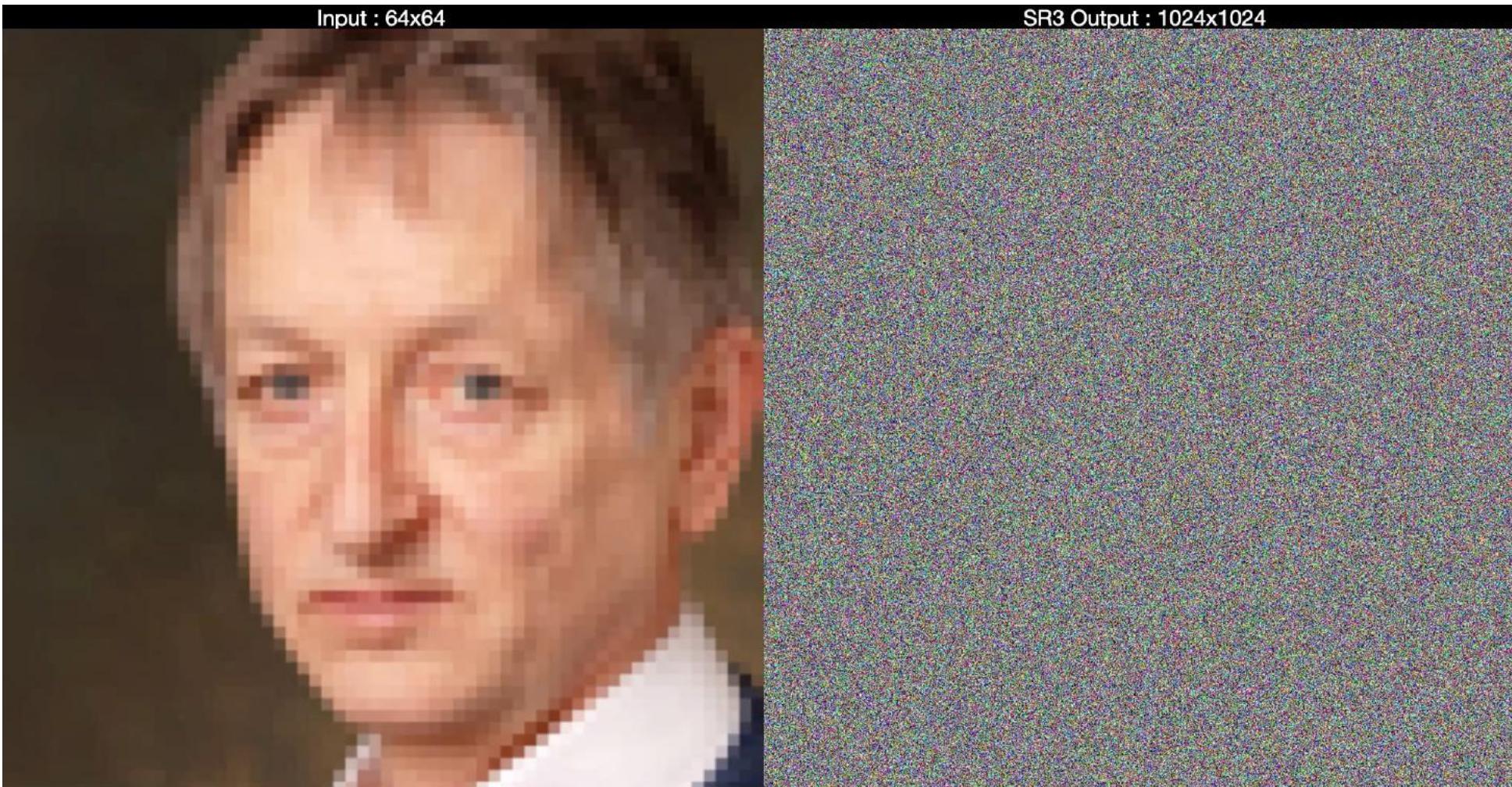
[Saharia et al. 2022]

Palette: Image-to-Image Diffusion Models

Panorama Generation



Super Resolution



Results of a SR3 model ($64 \times 64 \rightarrow 512 \times 512$), trained on FFHQ, and applied to images outside of the training set.

Latent Diffusion

Autoencoder with KL or VQ regularization

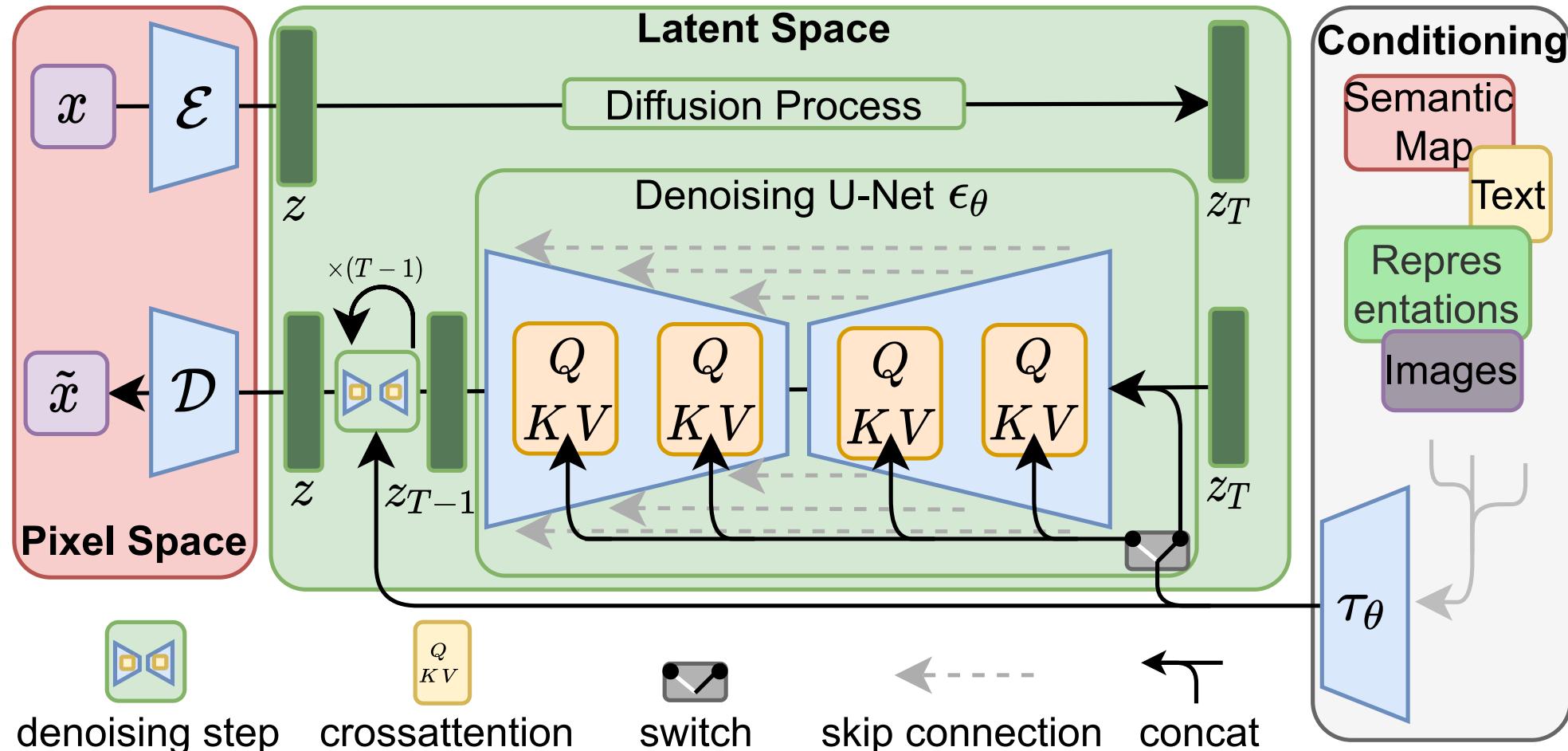
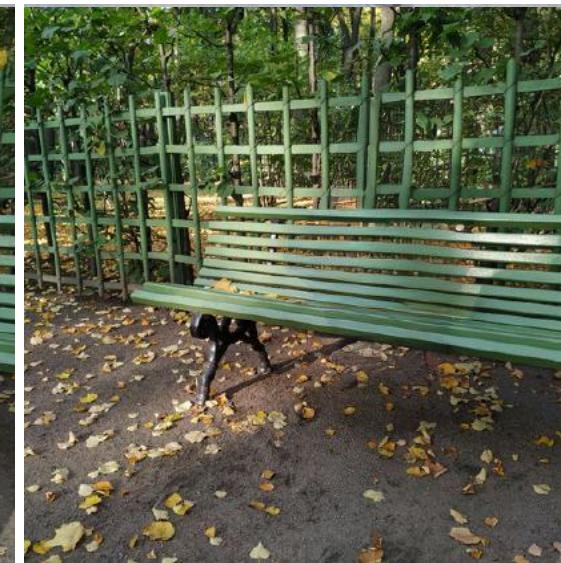
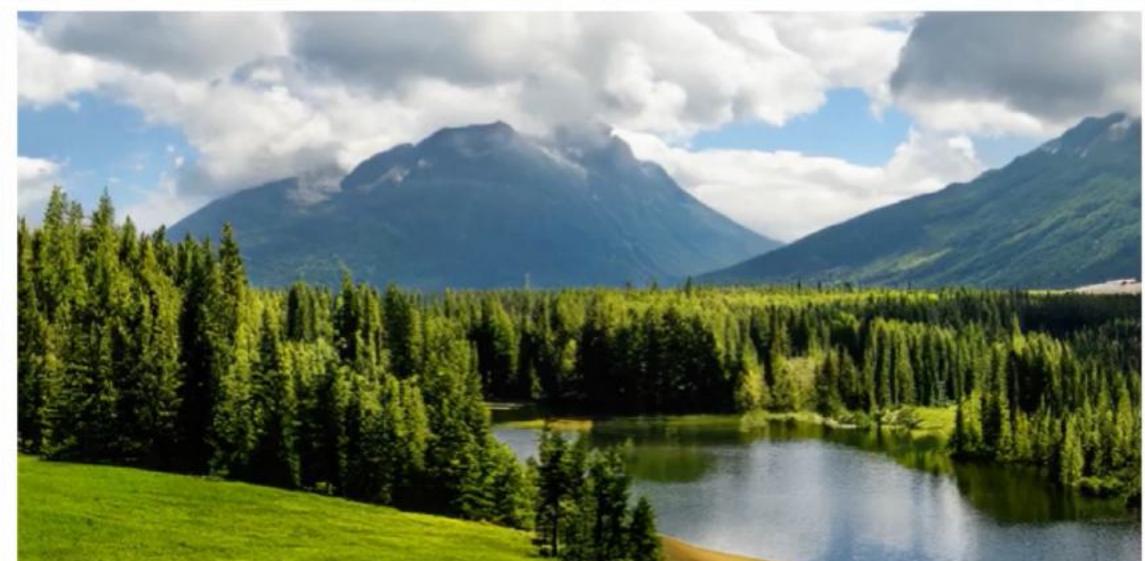
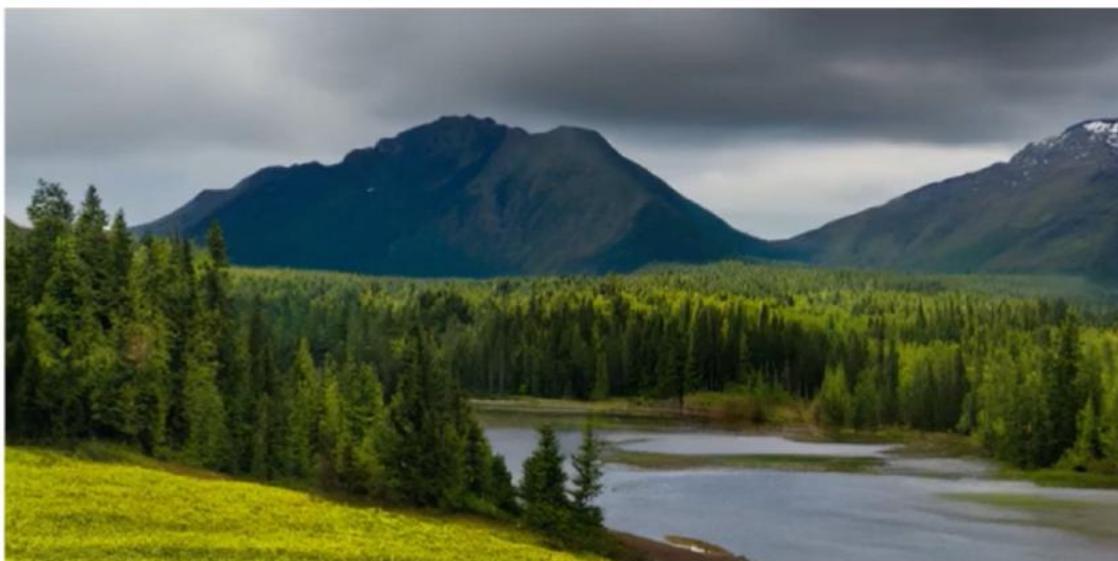


Image Inpainting



Semantic Image Synthesis



Generating Images from Text



"A sunset over a mountain, vector image"



"A sunset over a mountain, oil on canvas"



[“LDM”, Rombach et al., 2022] 259

Next Lecture:

Visual Quality Assessment