

BBM444

FUNDAMENTALS OF COMPUTATIONAL PHOTOGRAPHY

Lecture #10 – Deep Generative Models



HACETTEPE
UNIVERSITY
COMPUTER
VISION LAB

Erkut Erdem // Hacettepe University // Spring 2026

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

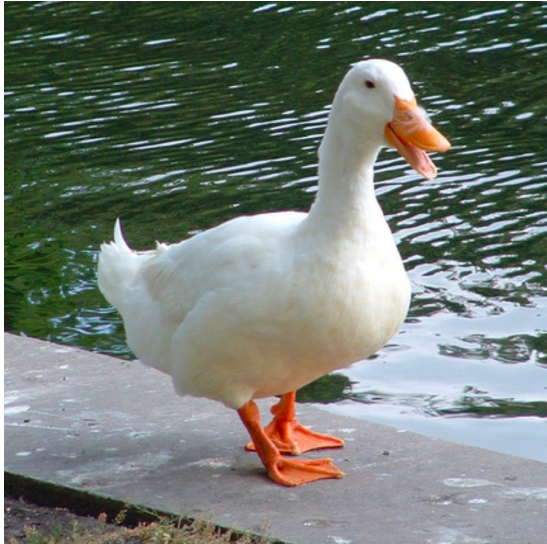
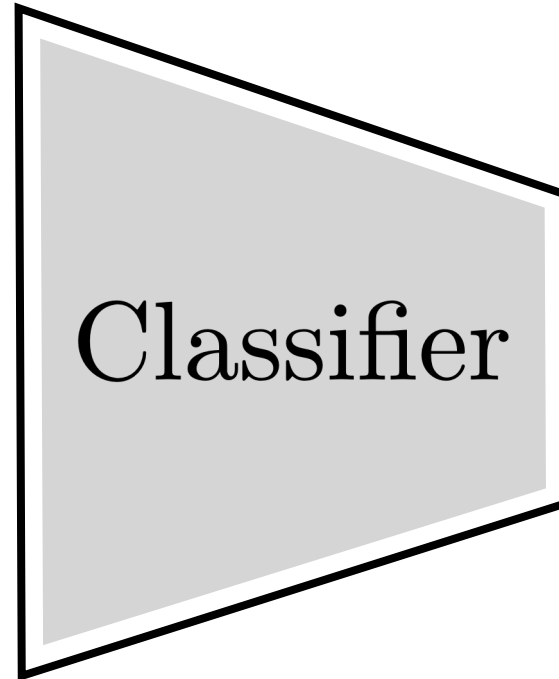


image x



"Duck"

label y

Analysis

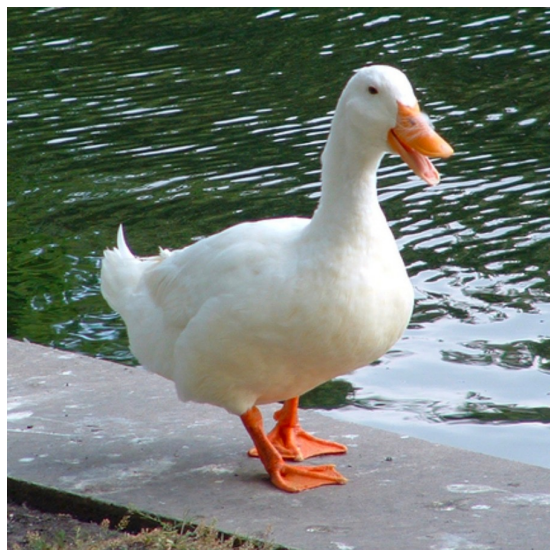
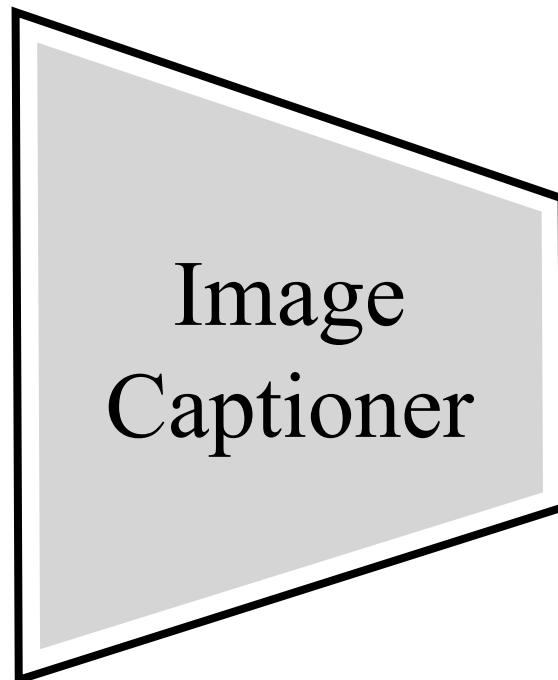


image x



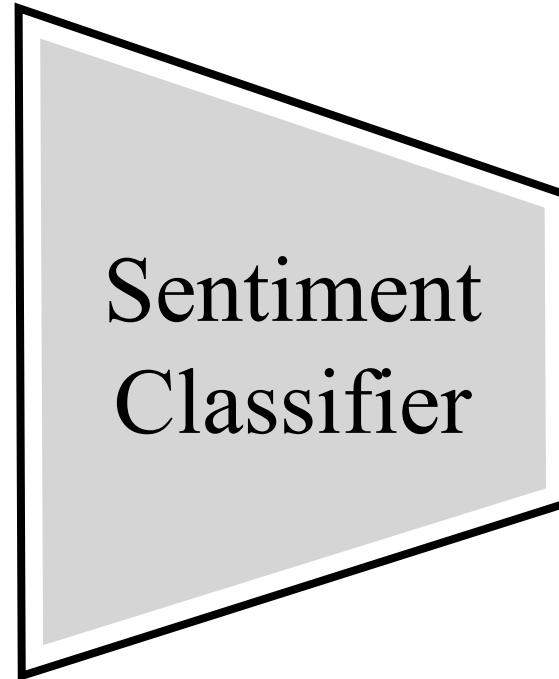
"A large duck standing by the river"

caption y

Analysis

"A statuesque
duck gazing
gracefully over
the water"

sentence **x**



"positive"

sentiment **y**

Analysis

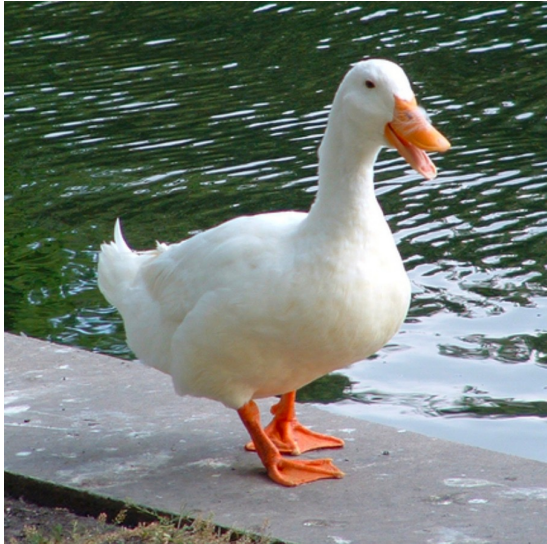
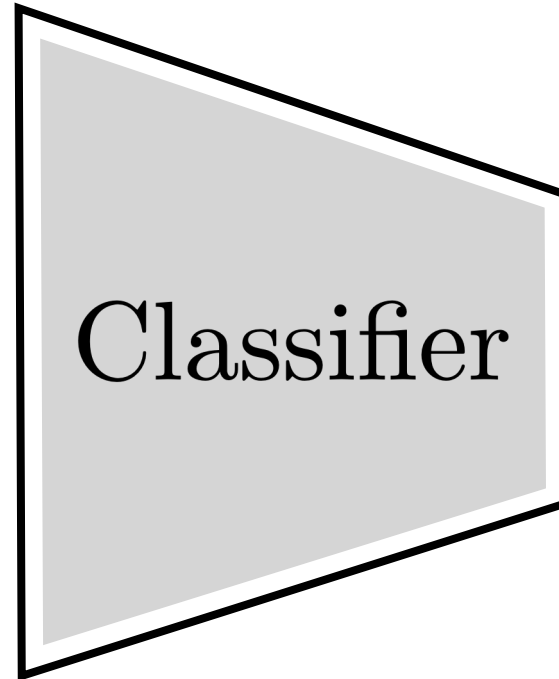


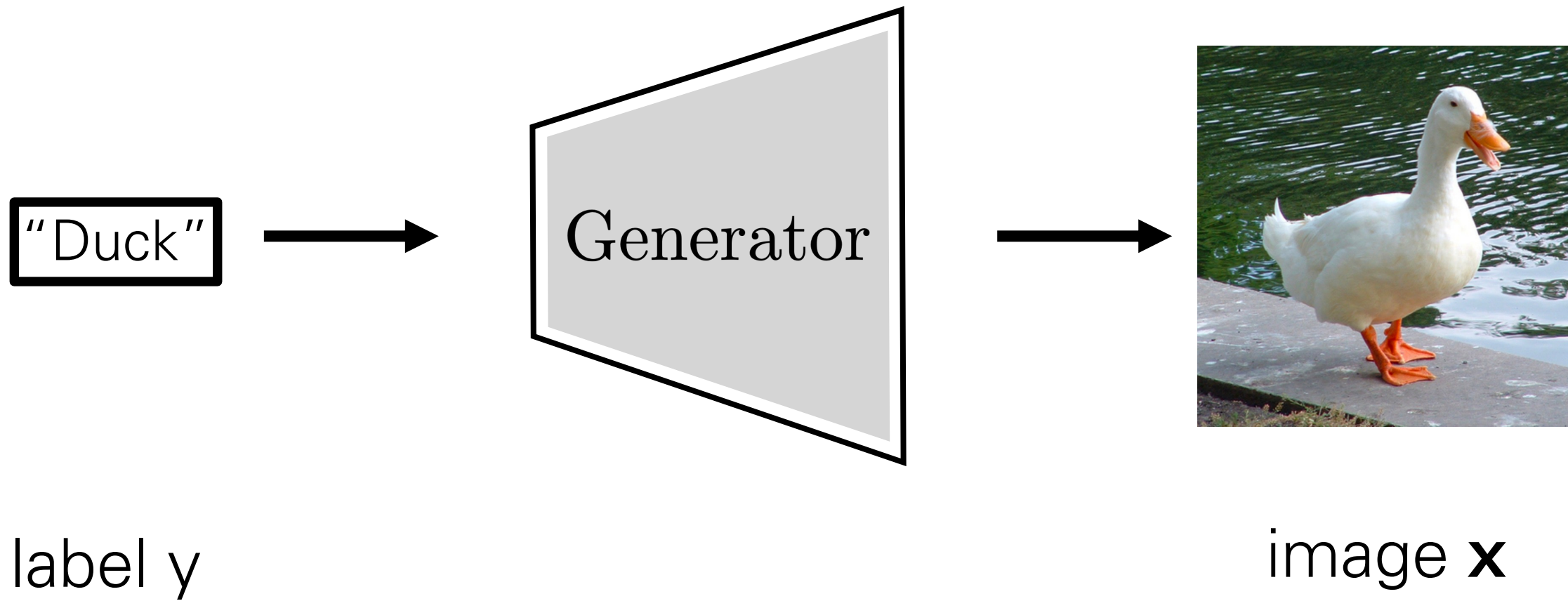
image x



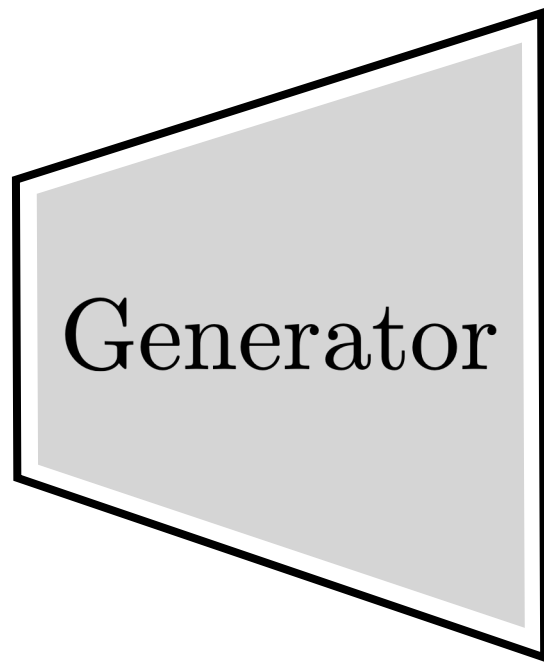
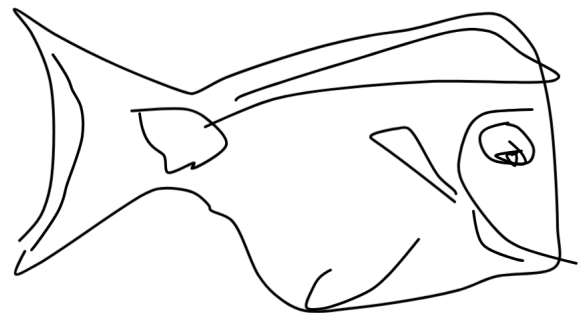
"Duck"

label y

Synthesis



Synthesis

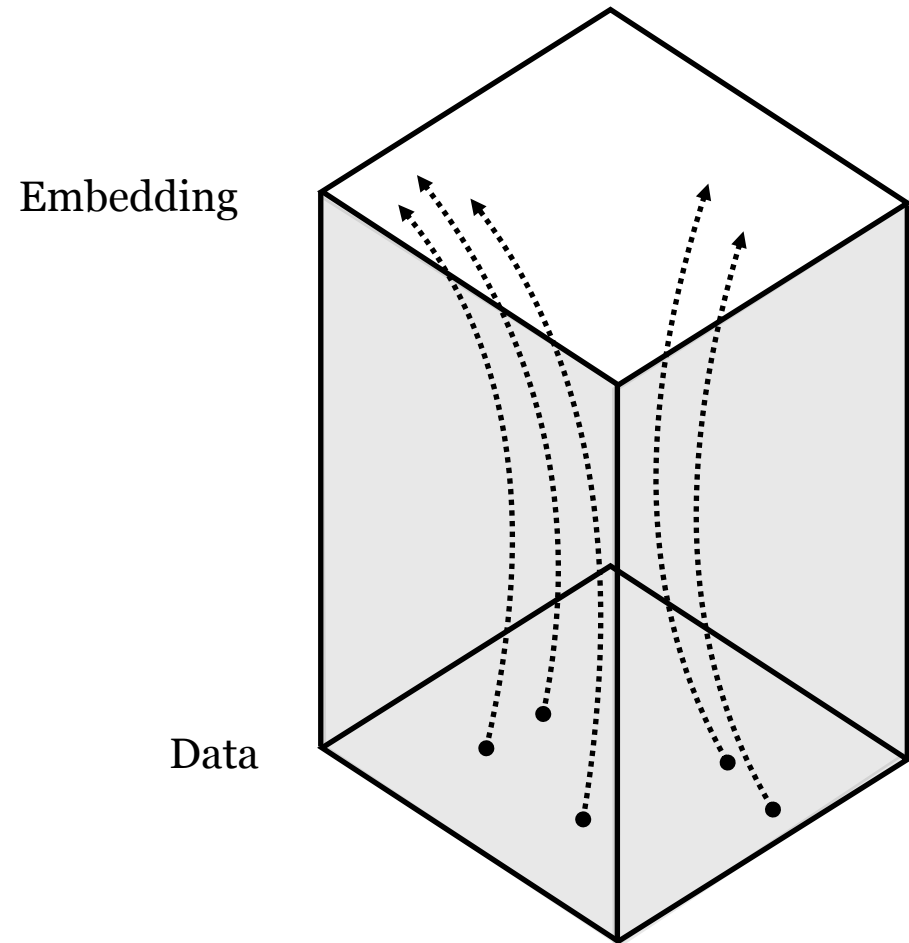


User sketch

Photo

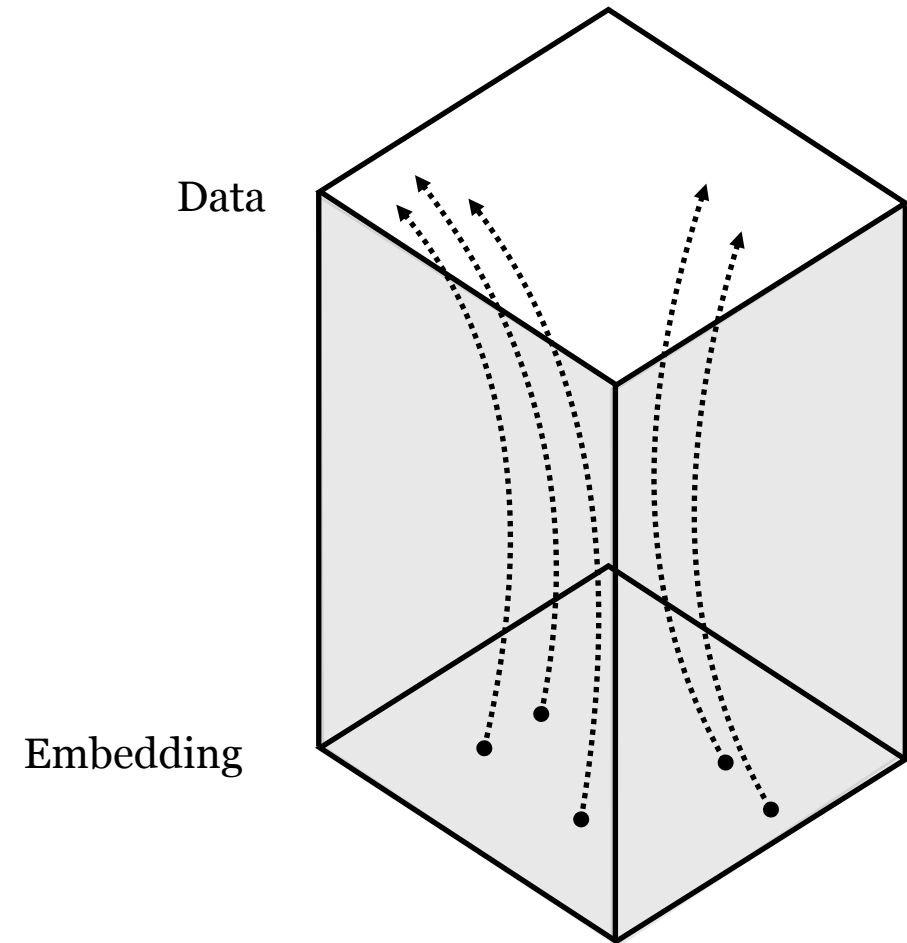
Deep nets are data transformers

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



Deep nets are data transformers

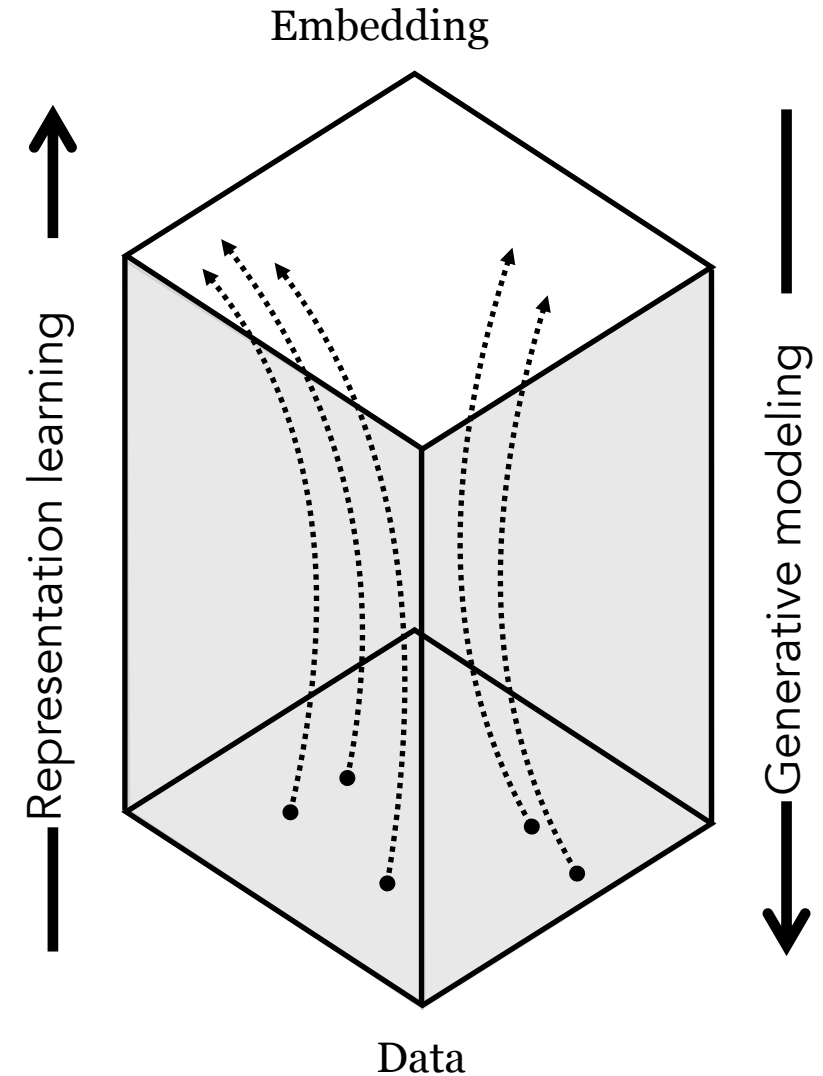
- Deep nets transform datapoints, layer by layer
- Each layer is a different representation of the data



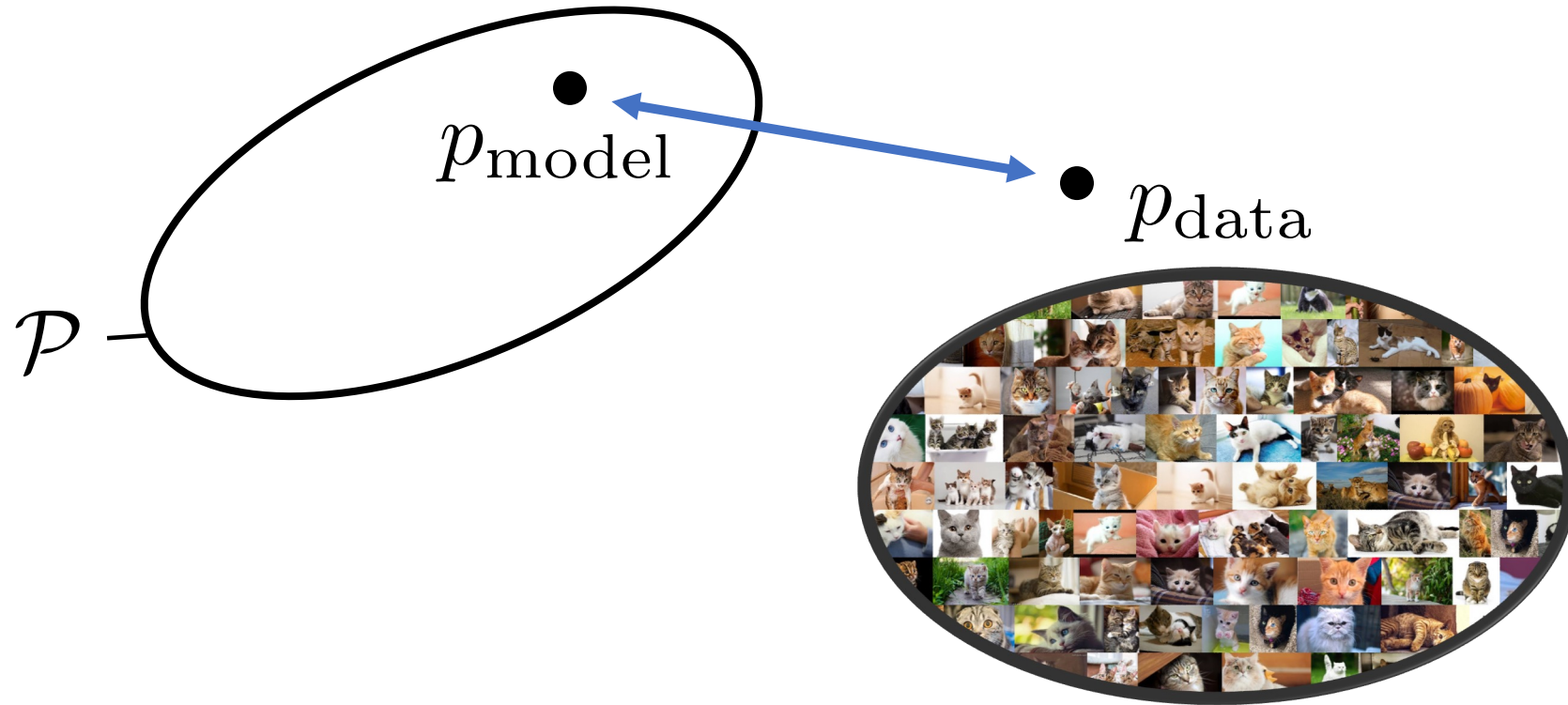
Generative modeling vs Representation learning

Representation learning:
mapping data to abstract representations
(analysis)

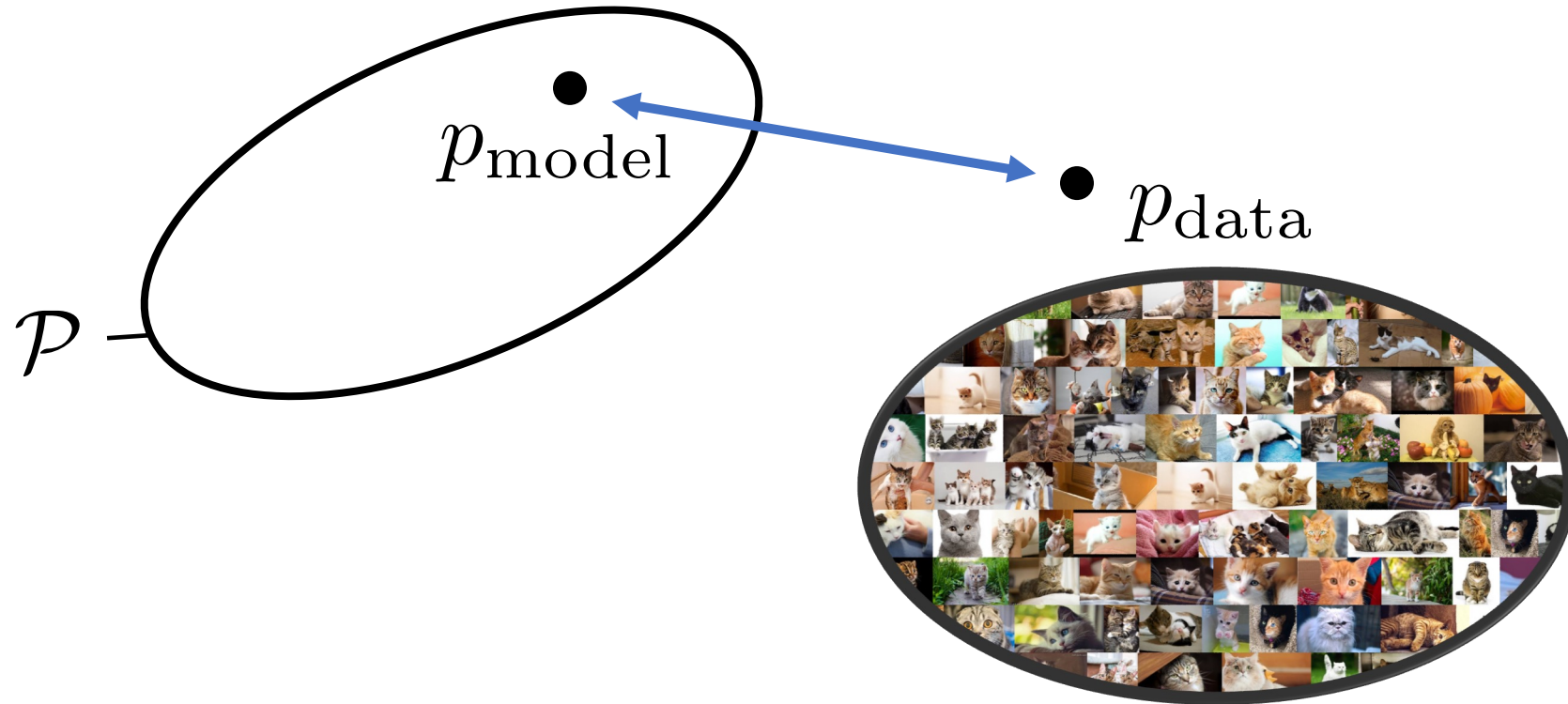
Generative modeling:
mapping abstract representations to data
(synthesis)



Generative Modeling

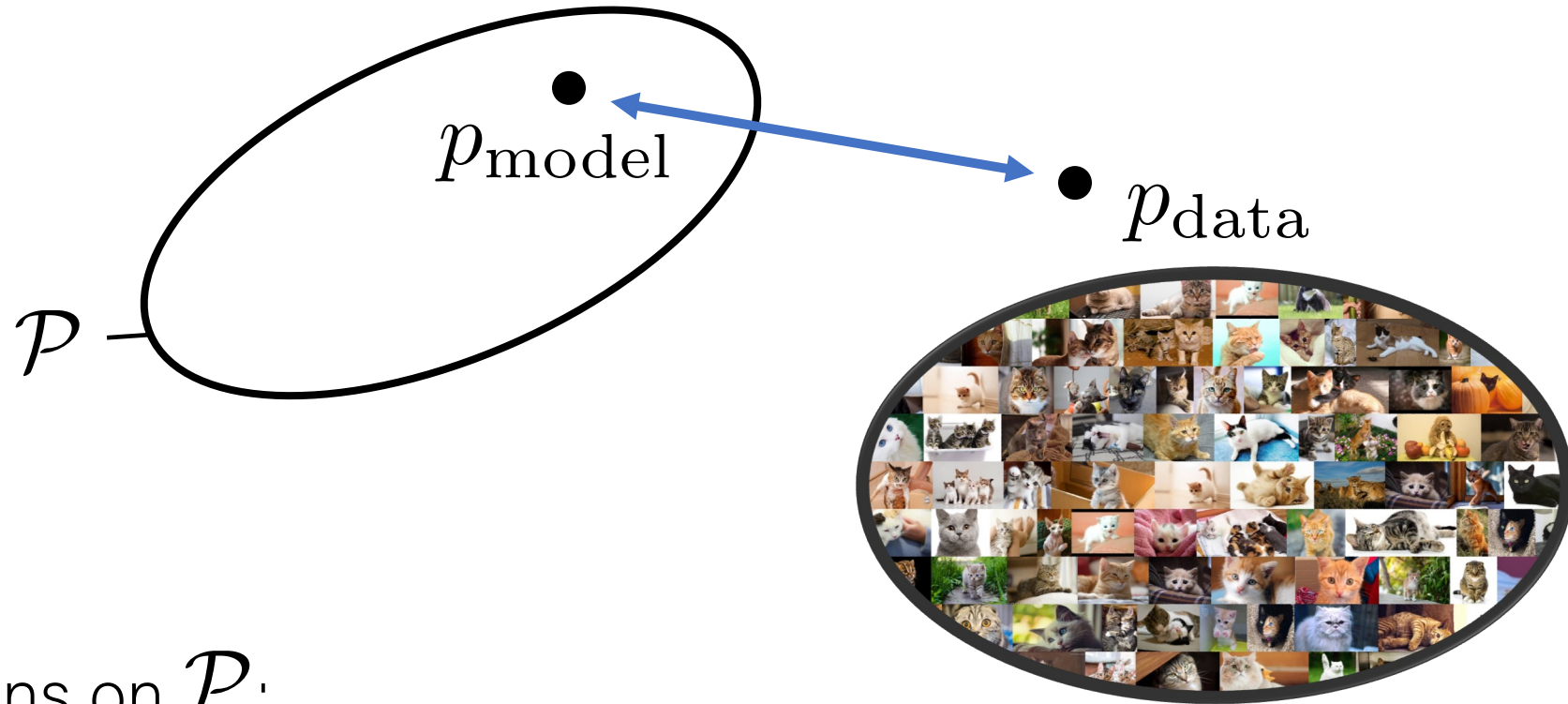


Generative Modeling



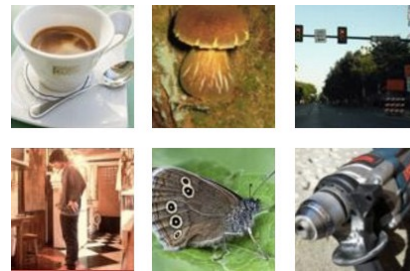
- **Goal:** Learn some underlying hidden structure of the training samples to generate novel samples from same data distribution

Generative Modeling

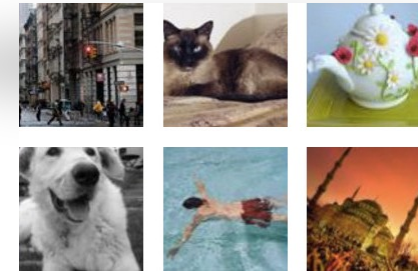


Assumptions on \mathcal{P} :

- tractable sampling

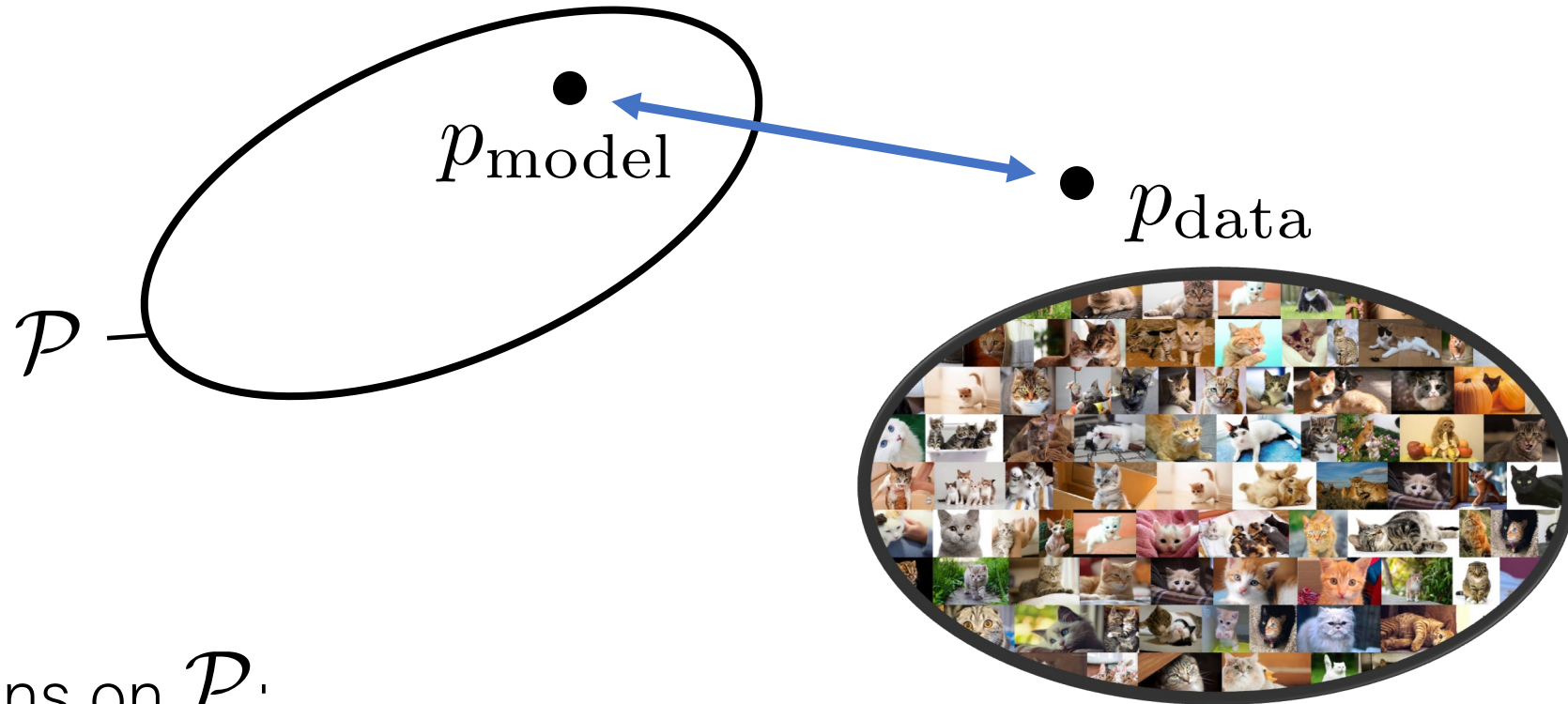


Training examples



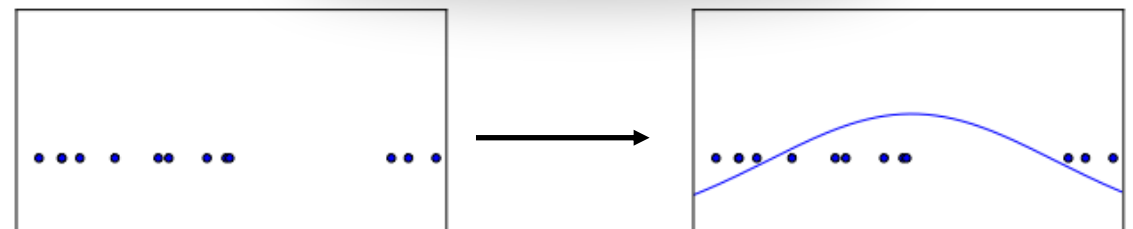
Model samples

Generative Modeling



Assumptions on \mathcal{P} :

- tractable sampling
- tractable likelihood function





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

Image synthesis

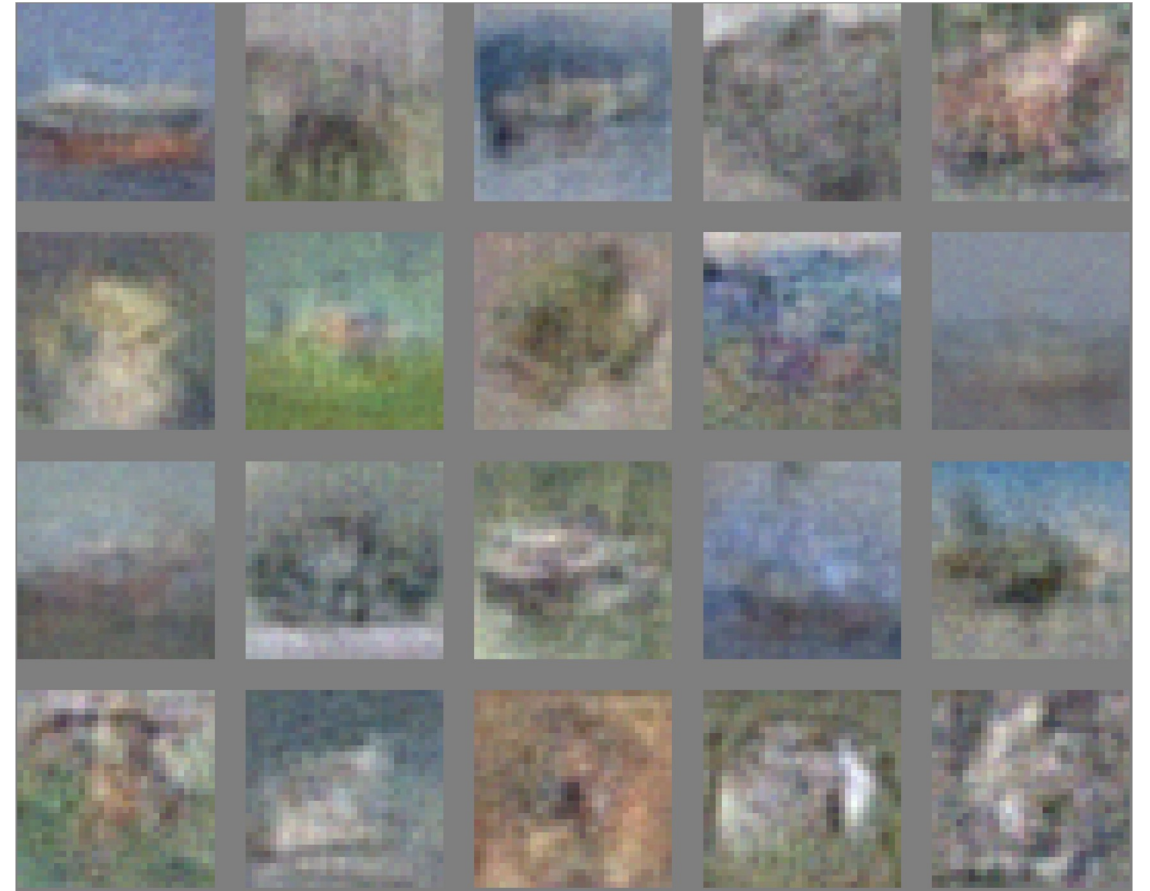
Generate Images



Generate Images



Generate Images



[GAN, Goodfellow et al. 2014]

Generate Images



[DCGAN, Radford, Metz, Chintala 2015]

Generate Images



Generate Images



Generate Images



Generate Images



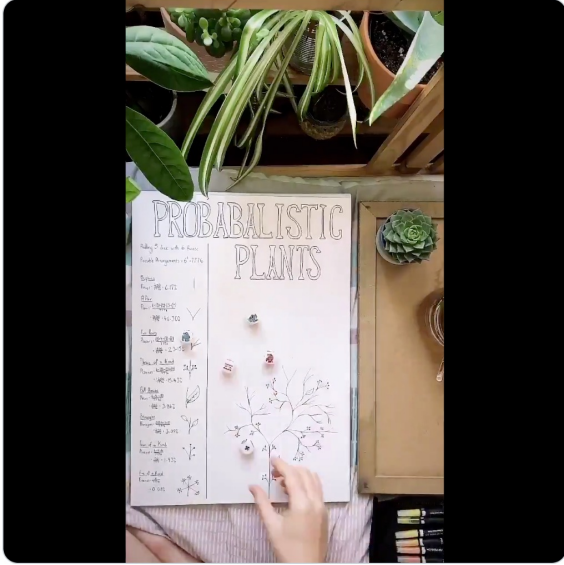
Procedural graphics





Aylian @Aylian · Nov 17

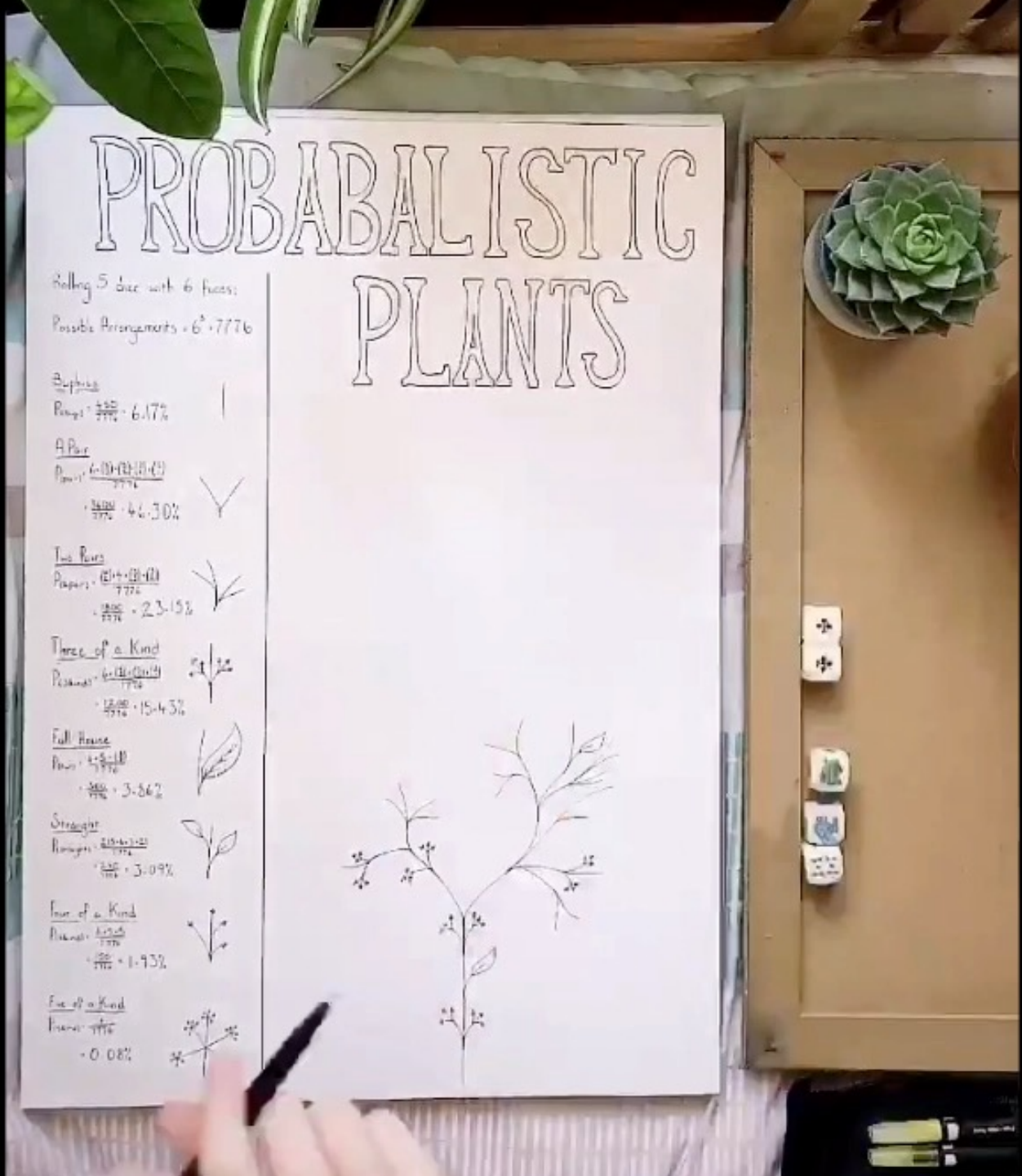
Made up a set of rules and rolled some dice to decide how this plant would grow. I never did get that five of a kind, as expected, but I was still hopeful! 🍀🍀🍀



52

1.1K

4.5K



PROBABALISTIC PLANTS

Rolling 5 dice with 6 faces:

Possible Arrangements = $6^5 = 7776$

Supplies

Range: $\frac{3600}{7776} = 6.17\%$

A Pair

Range: $\frac{4 \cdot (1+2+3+4+5)}{7776} = 4.30\%$

Two Pairs

Range: $\frac{6 \cdot (1+2+3)}{7776} = 2.31\%$

Three of a Kind

Range: $\frac{6 \cdot (1+2+3+4+5)}{7776} = 15.43\%$

Full House

Range: $\frac{4 \cdot (1+2+3+4+5)}{7776} = 3.24\%$

Straight

Range: $\frac{2 \cdot (1+2+3+4+5)}{7776} = 3.09\%$

Four of a Kind

Range: $\frac{4 \cdot (1+2+3+4+5)}{7776} = 1.93\%$

Five of a Kind

Range: $\frac{5}{7776} = 0.08\%$

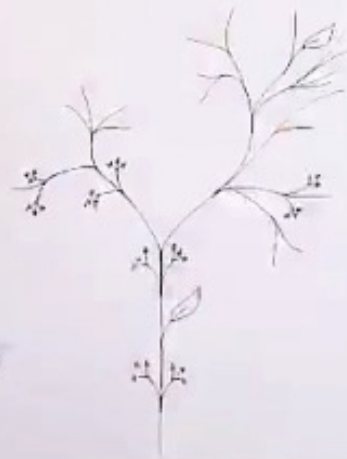
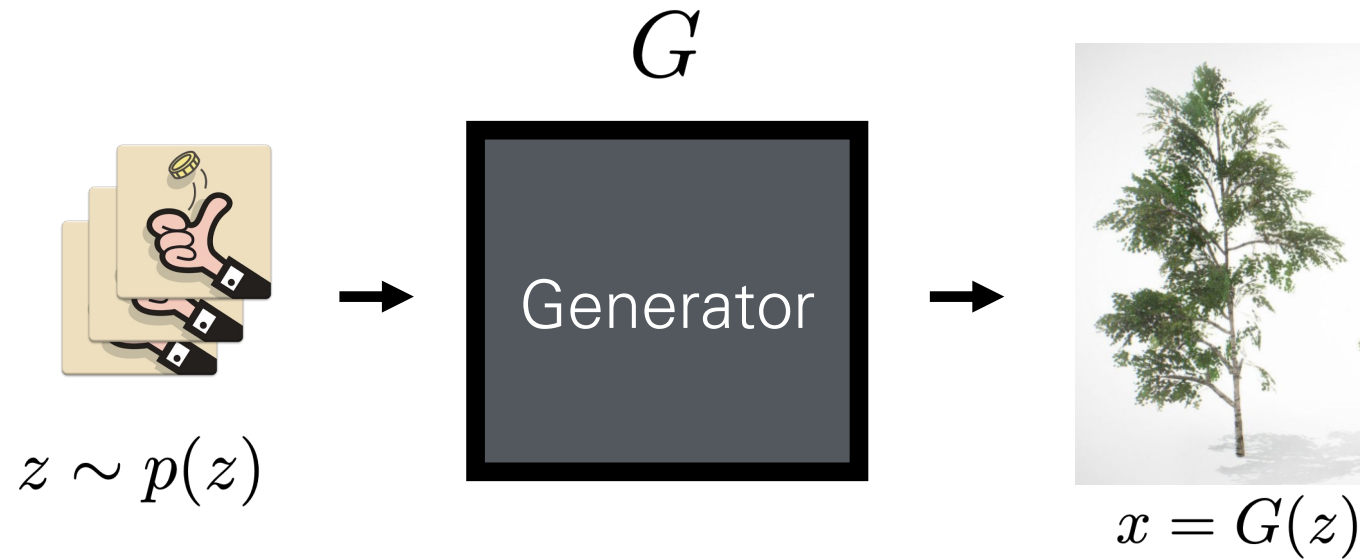


Image synthesis from "noise"



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

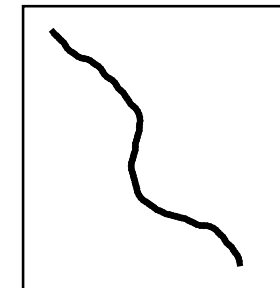
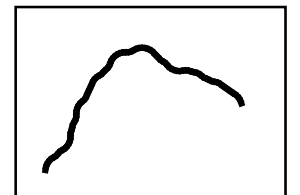
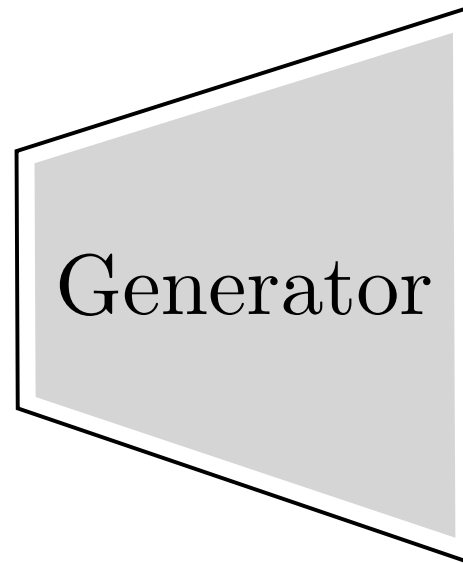
if $z_i == 1$ **then**

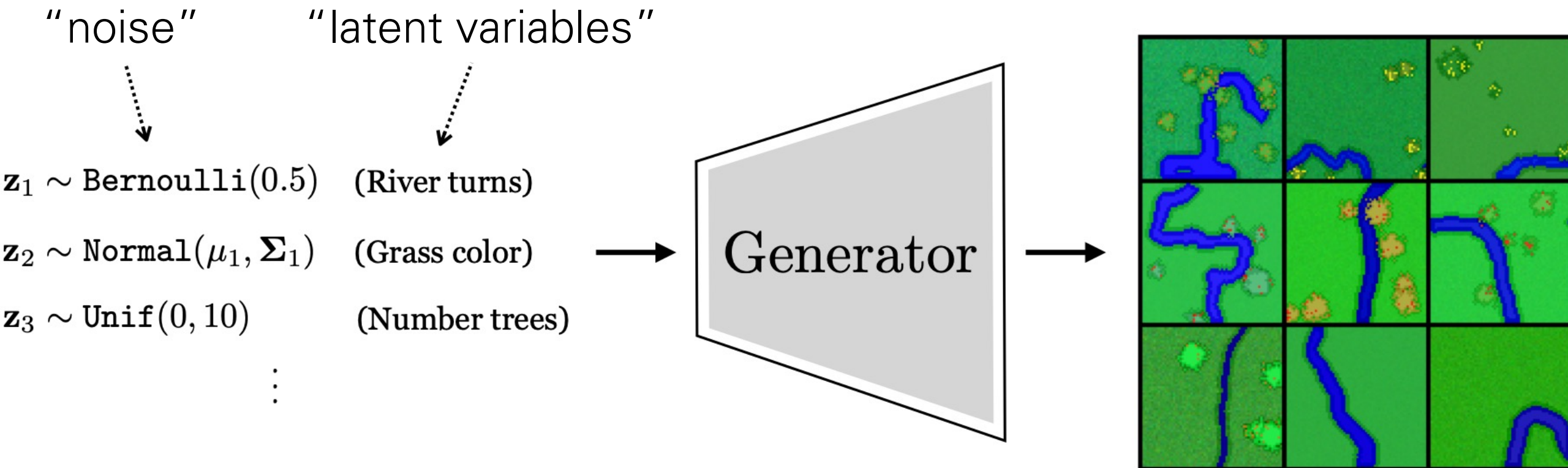
 rotate heading 10° to the right

else

 rotate heading 10° to the left

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





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

What's the goal of generative modeling?

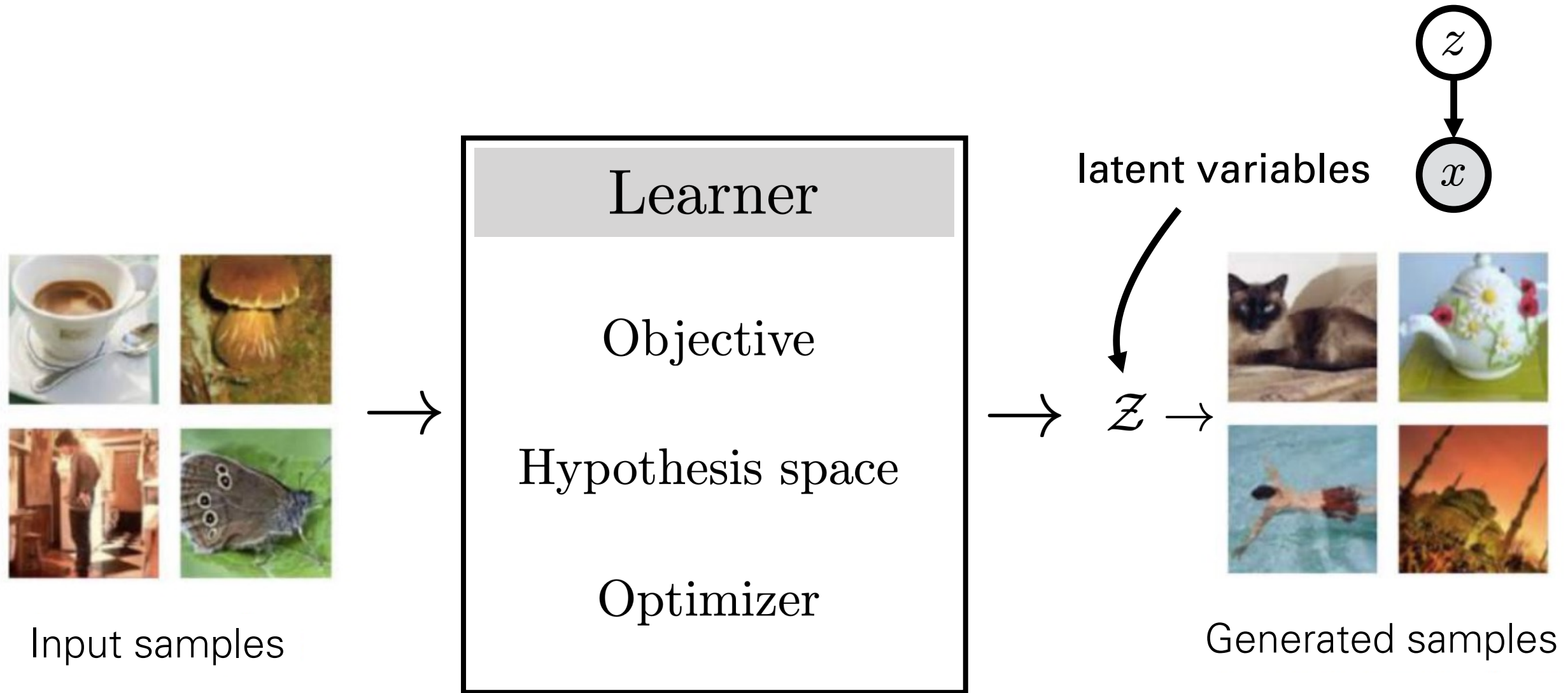
- Make synthetic data that “looks like” real data.
- How to measure “looks like”?
- The main answer in deep generative models is: “has high probability under a density model fit to real data.”

What's the goal of generative modeling?

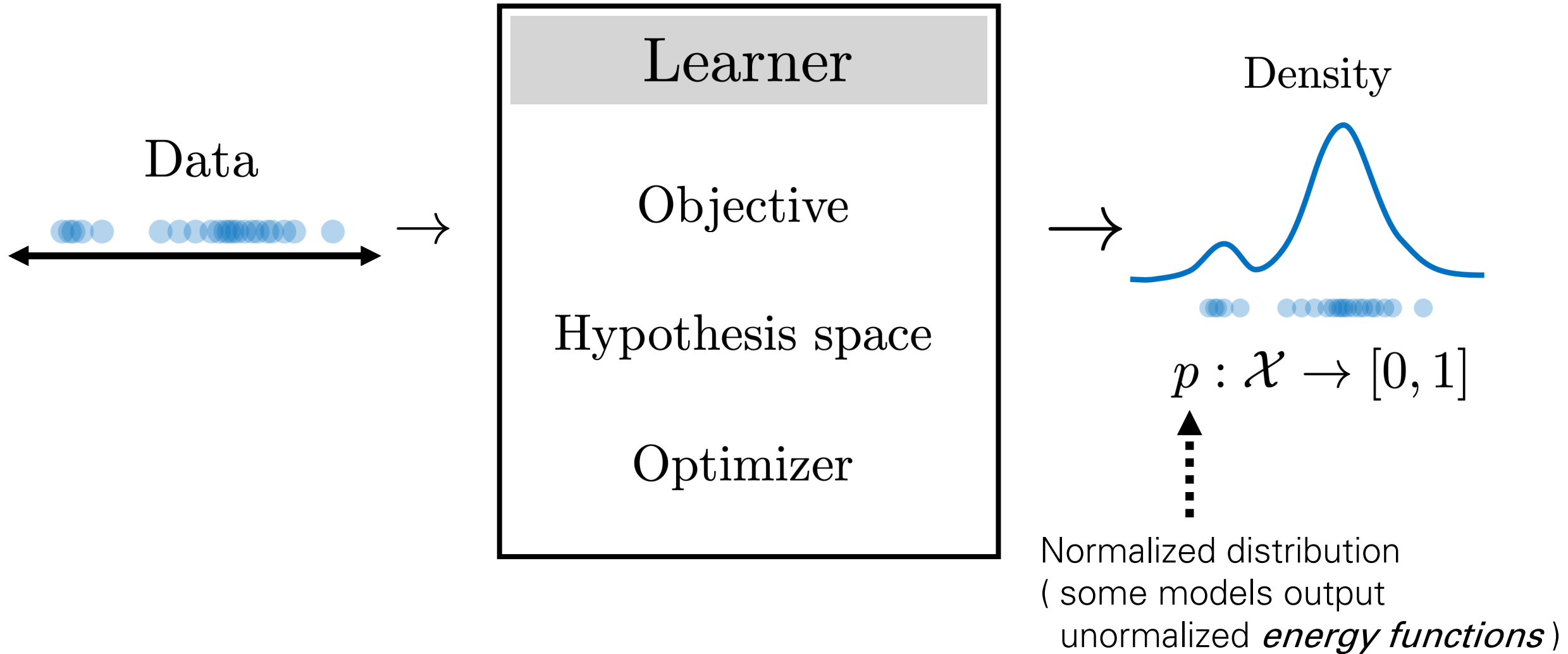
The goal is not to replicate the training data but to make **new** data that is **realistic** (captures the essential properties of real data)

(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

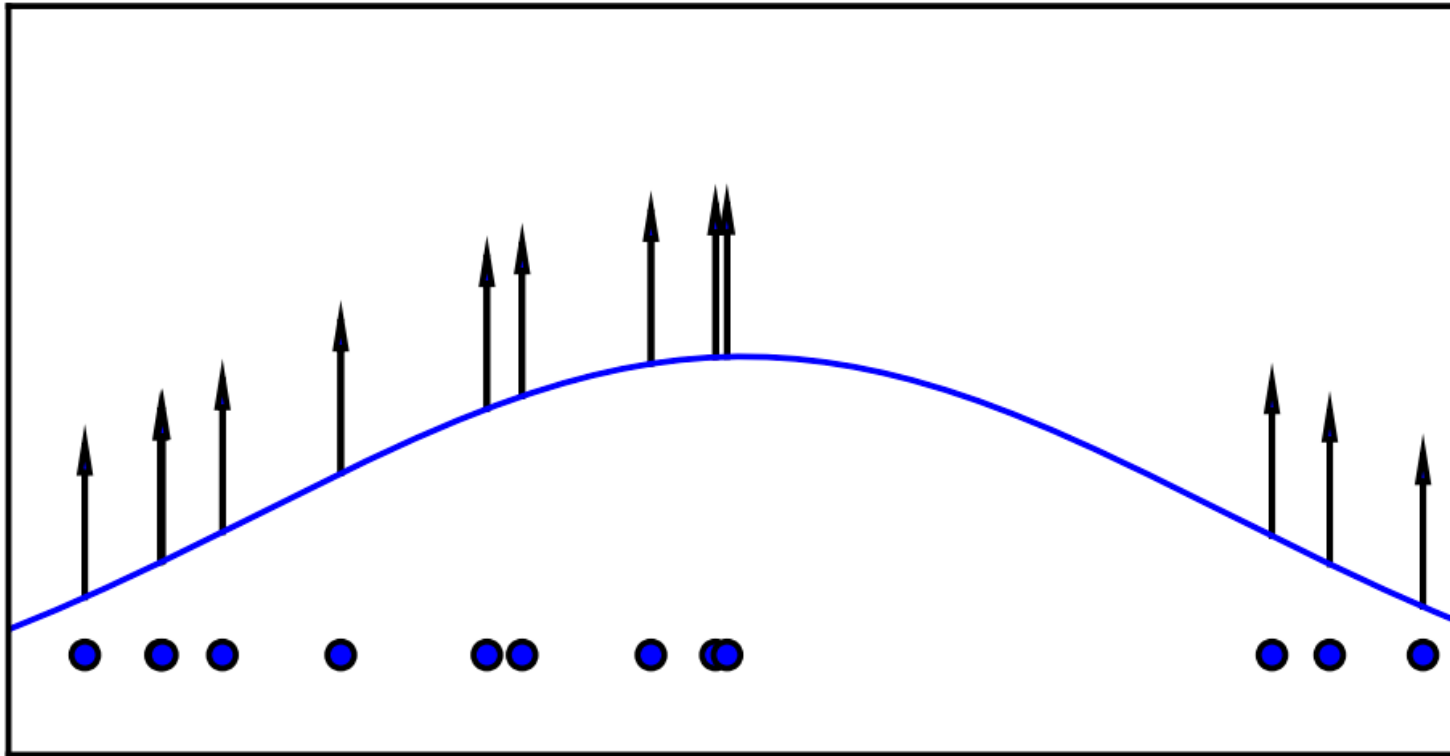


fig from [Goodfellow, 2016]

Max likelihood objective

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

Considering only Gaussian fits

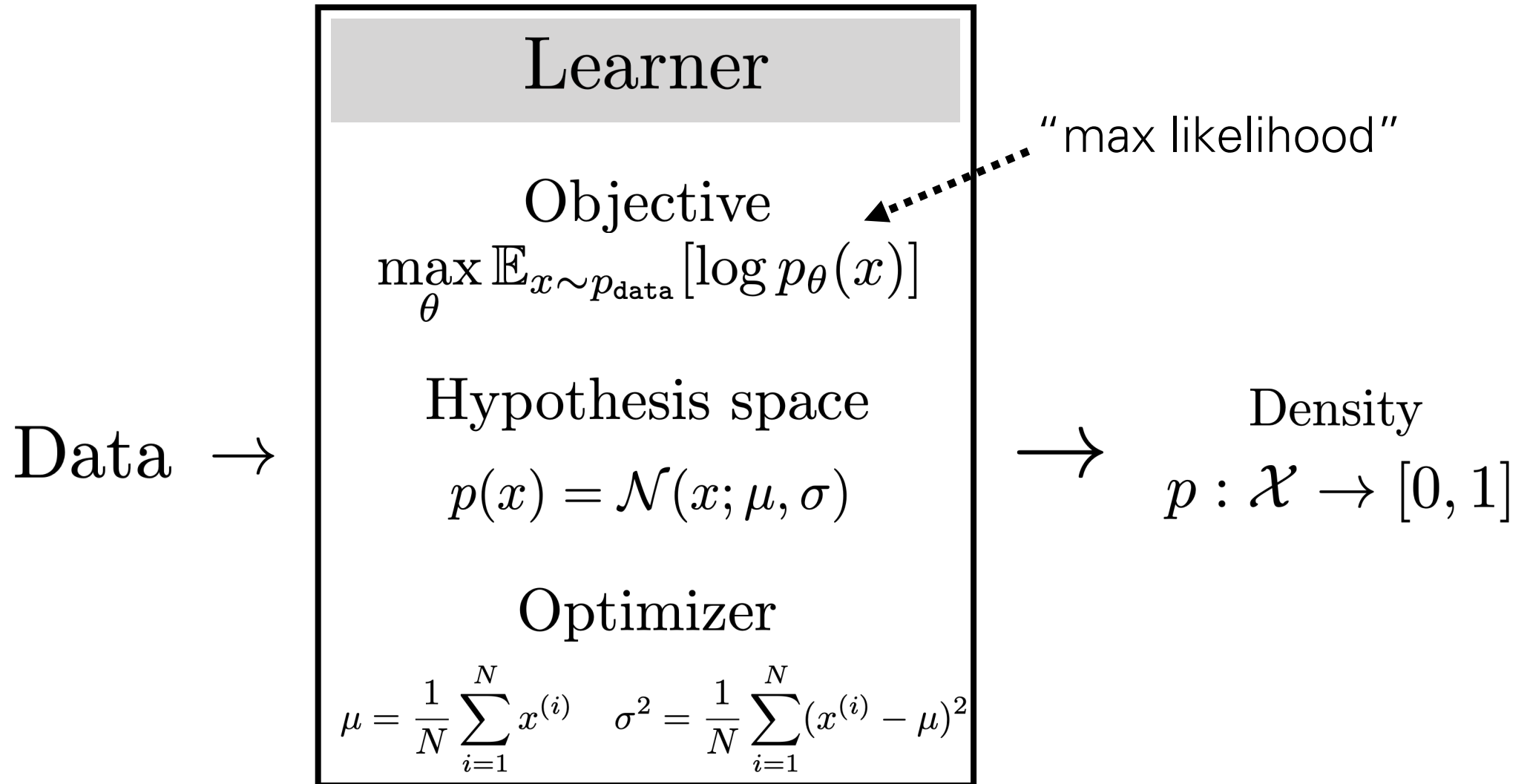
$$p_{\theta}(x) = \mathcal{N}(x; \mu, \sigma)$$

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

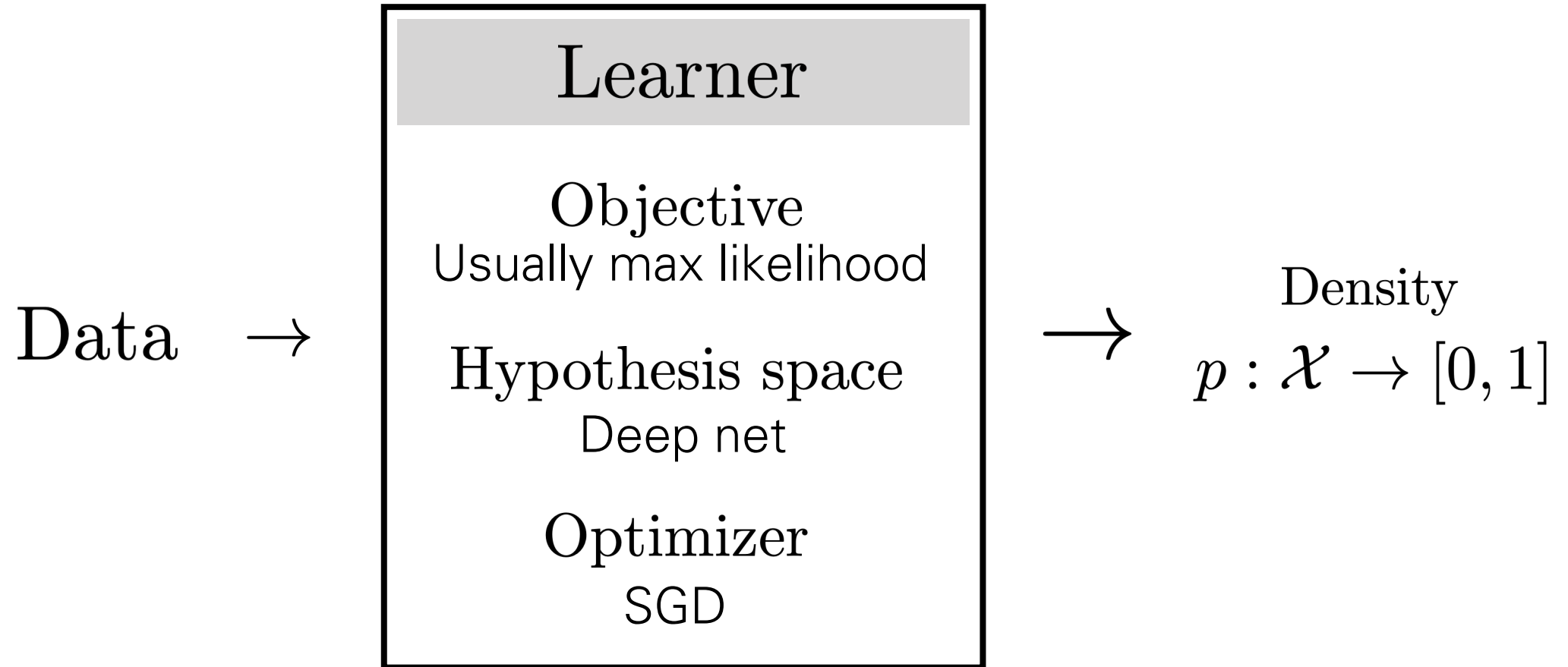
Closed form optimum:

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

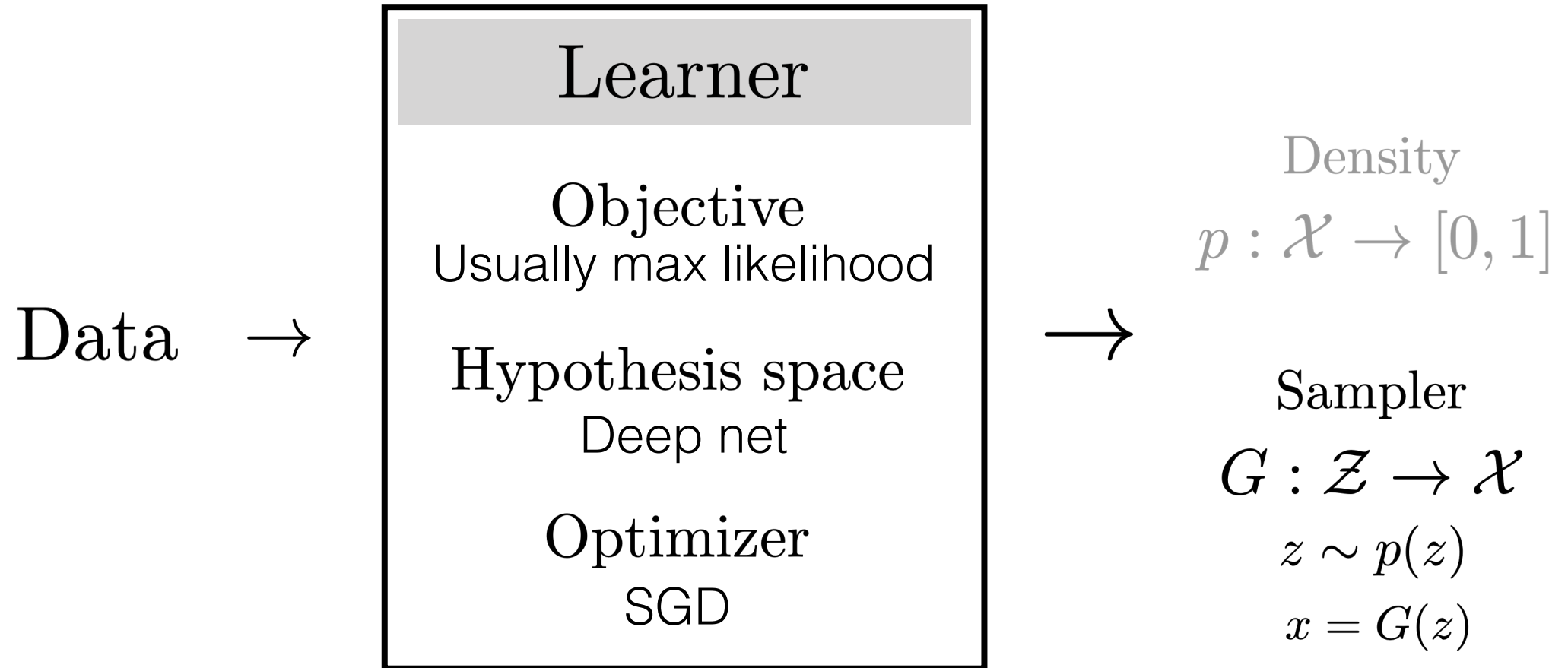
Case study #1: Fitting a Gaussian to data



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:

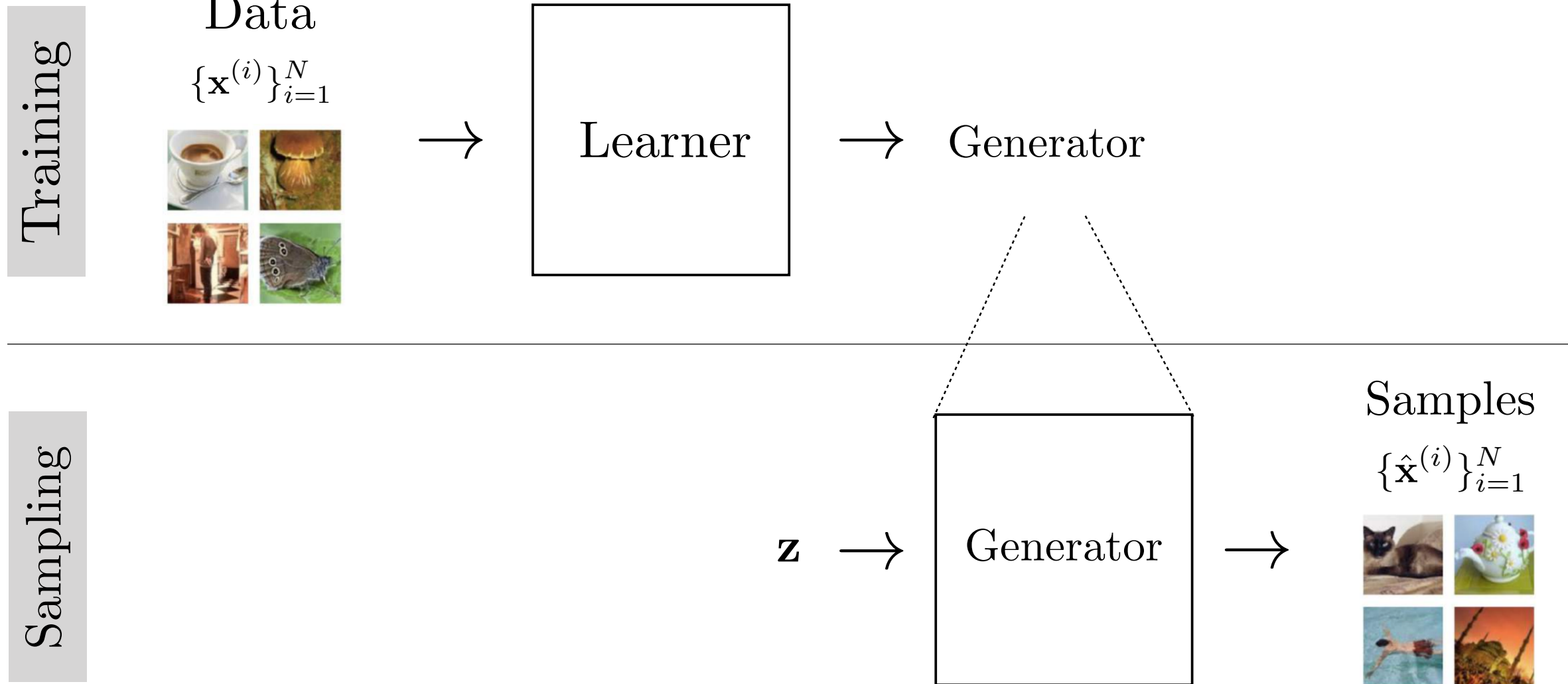
confusingly, sometimes called an “implicit generative model”



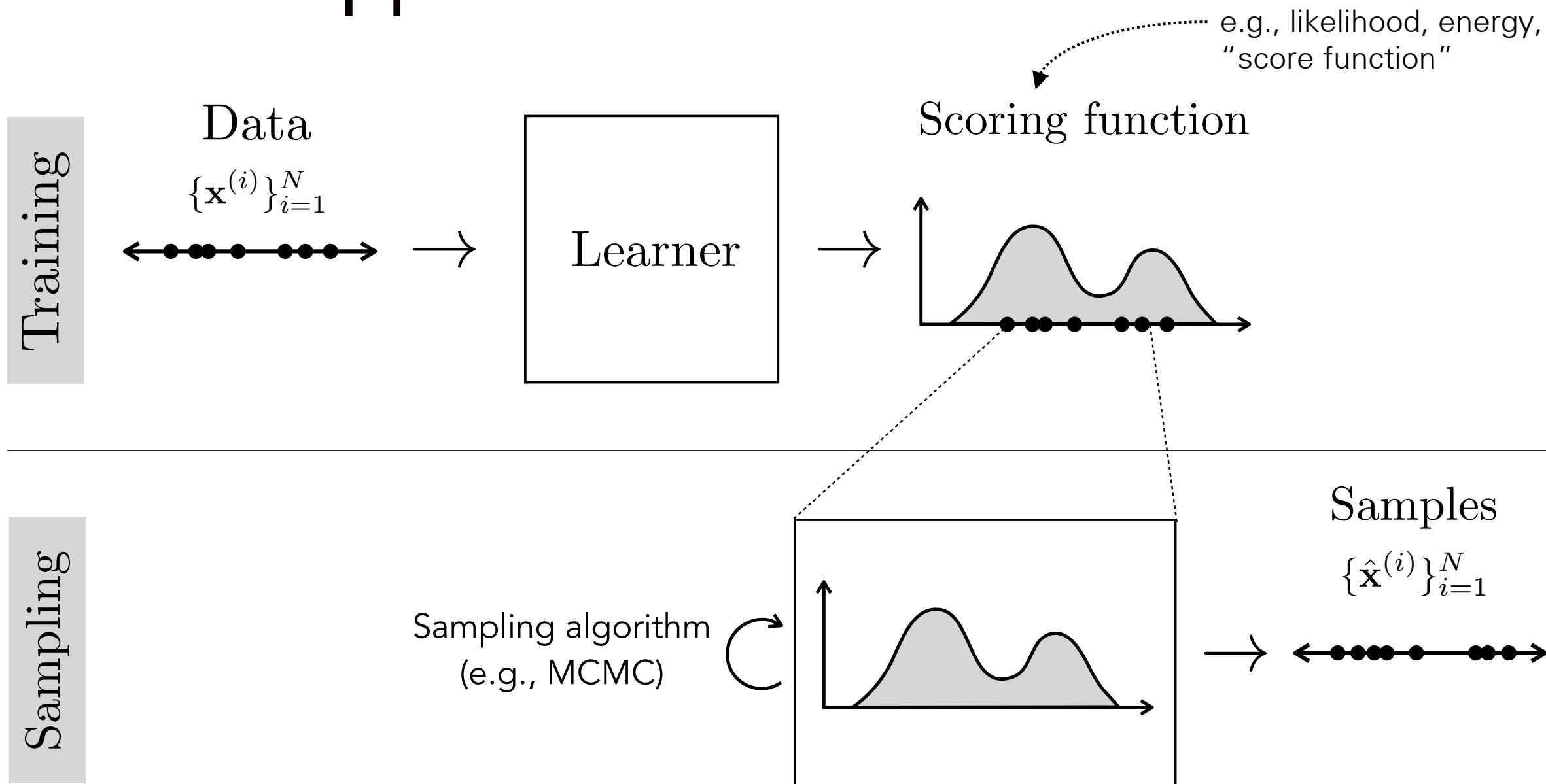
1. Direct approach: learn a function that generates data directly $G : \mathcal{Z} \rightarrow \mathcal{X}$

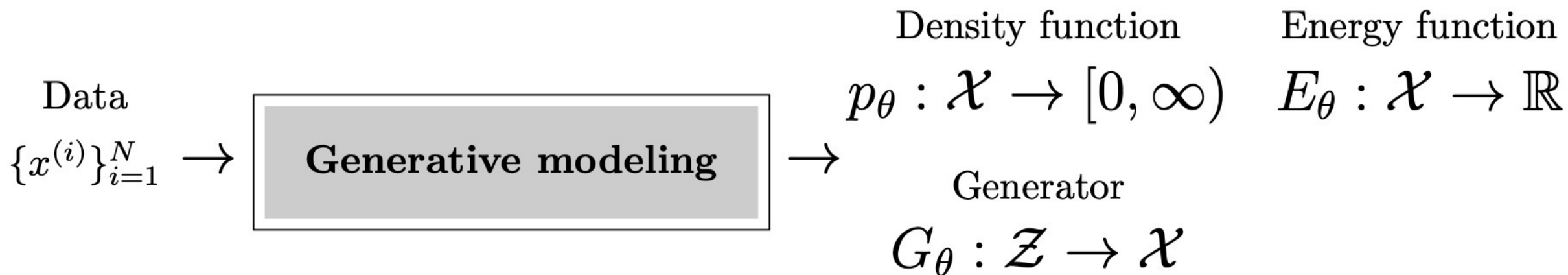
2. Indirect approach: learn a function that scores data; generate data by finding points that score highly under this function $E : \mathcal{X} \rightarrow \mathbb{R}$

Direct approach



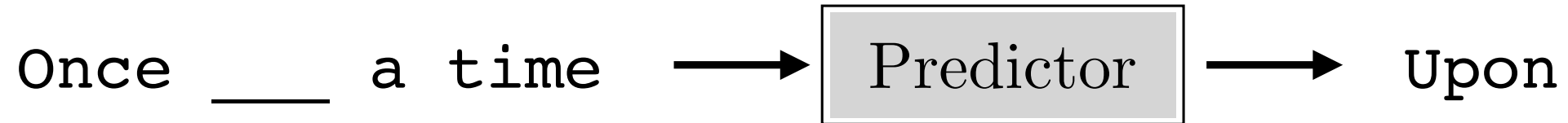
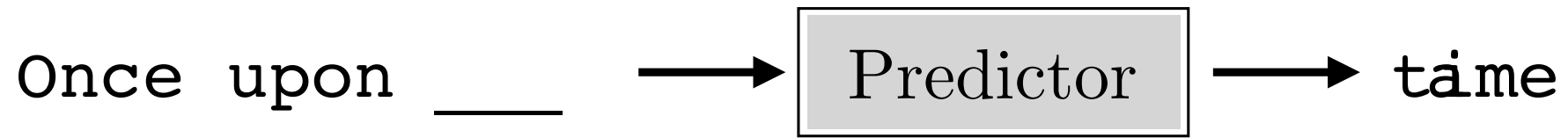
Indirect approach





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

Autoregressive models



Autoregressive probability models

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

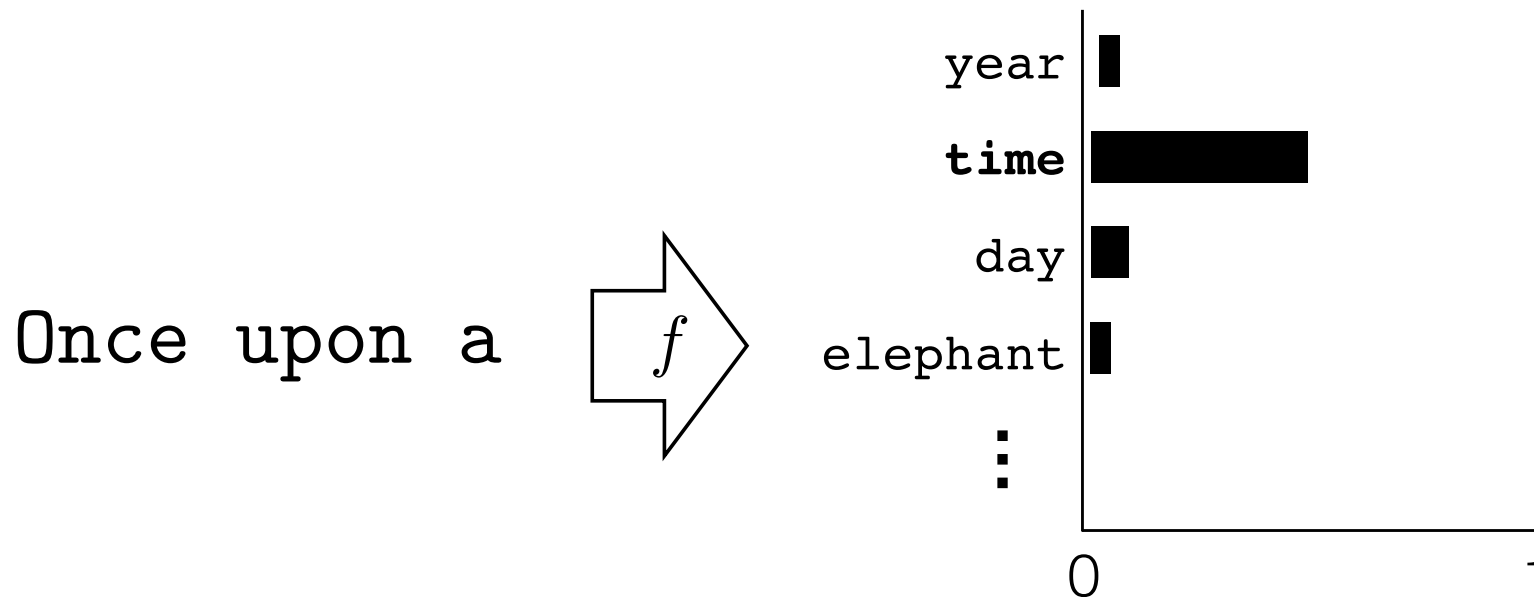
$$p(\mathbf{X}) = \prod_{i=1}^n p(\mathbf{x}_i | \mathbf{x}_1, \dots, \mathbf{x}_{i-1})$$

$$\begin{array}{c} p(\text{time} | \text{Once, upon, a}) \\ \underbrace{\hspace{10em}} \\ p(\text{a} | \text{Once, upon}) \\ \underbrace{\hspace{10em}} \\ p(\text{Once upon a time}) \\ \underbrace{\hspace{2em}} \\ p(\text{Once}) \\ \underbrace{\hspace{10em}} \\ p(\text{upon} | \text{Once}) \end{array}$$

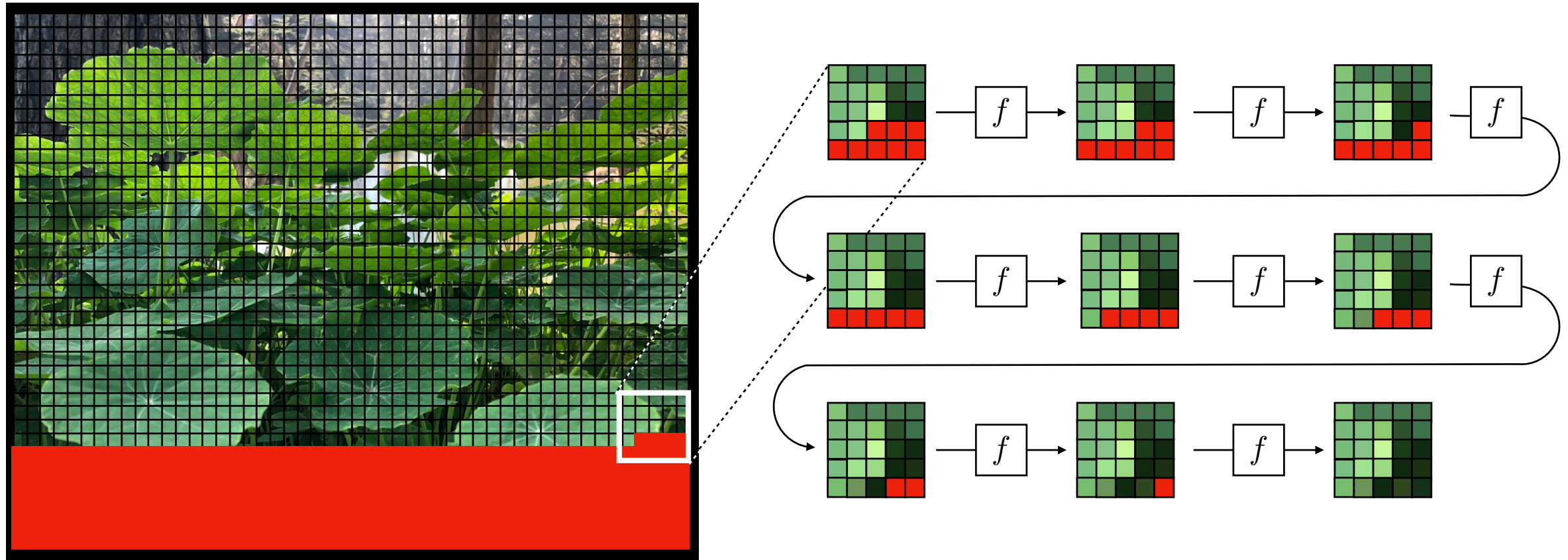
Modeling a sequence of words

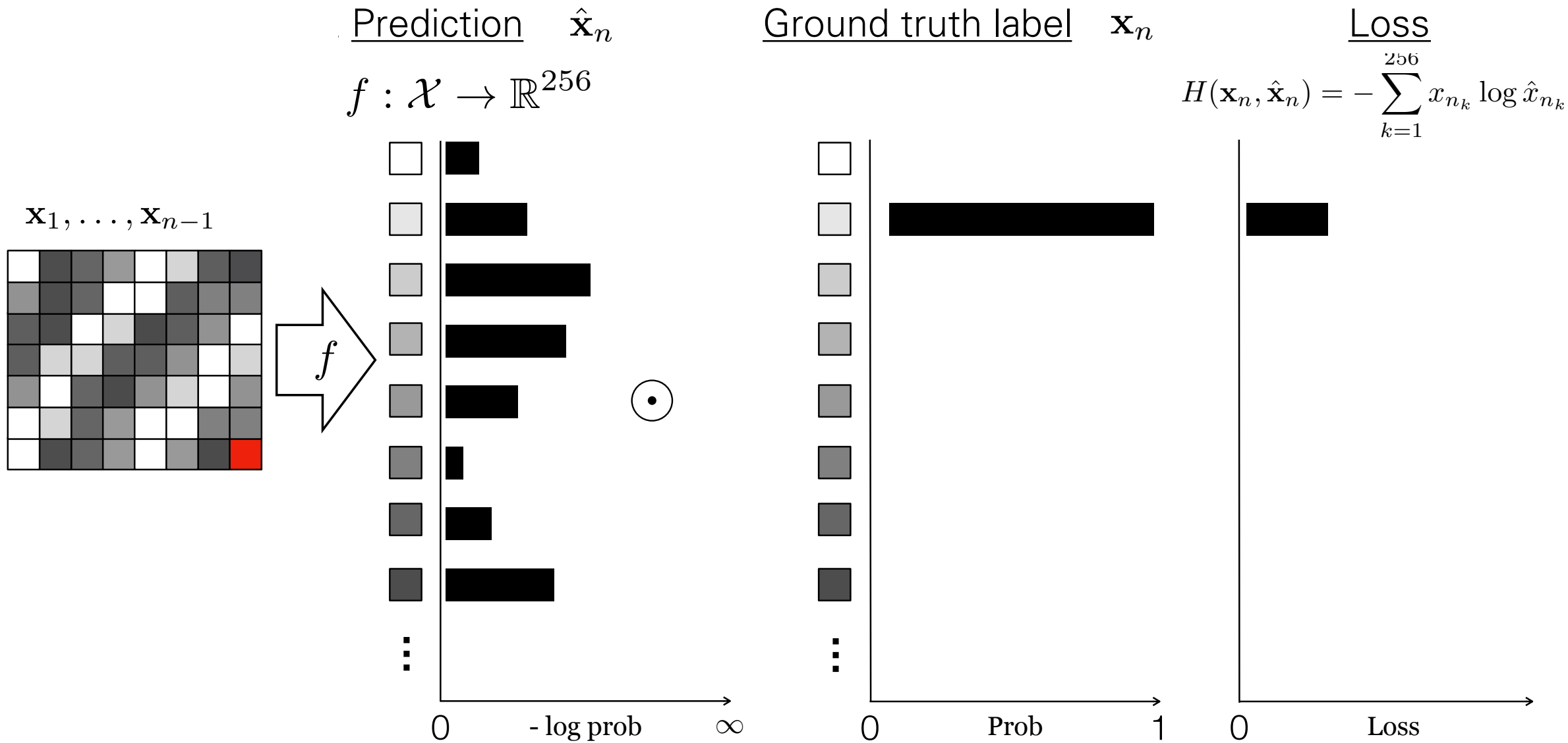
How to model $p(\text{time}|\text{Once, upon, a})$?

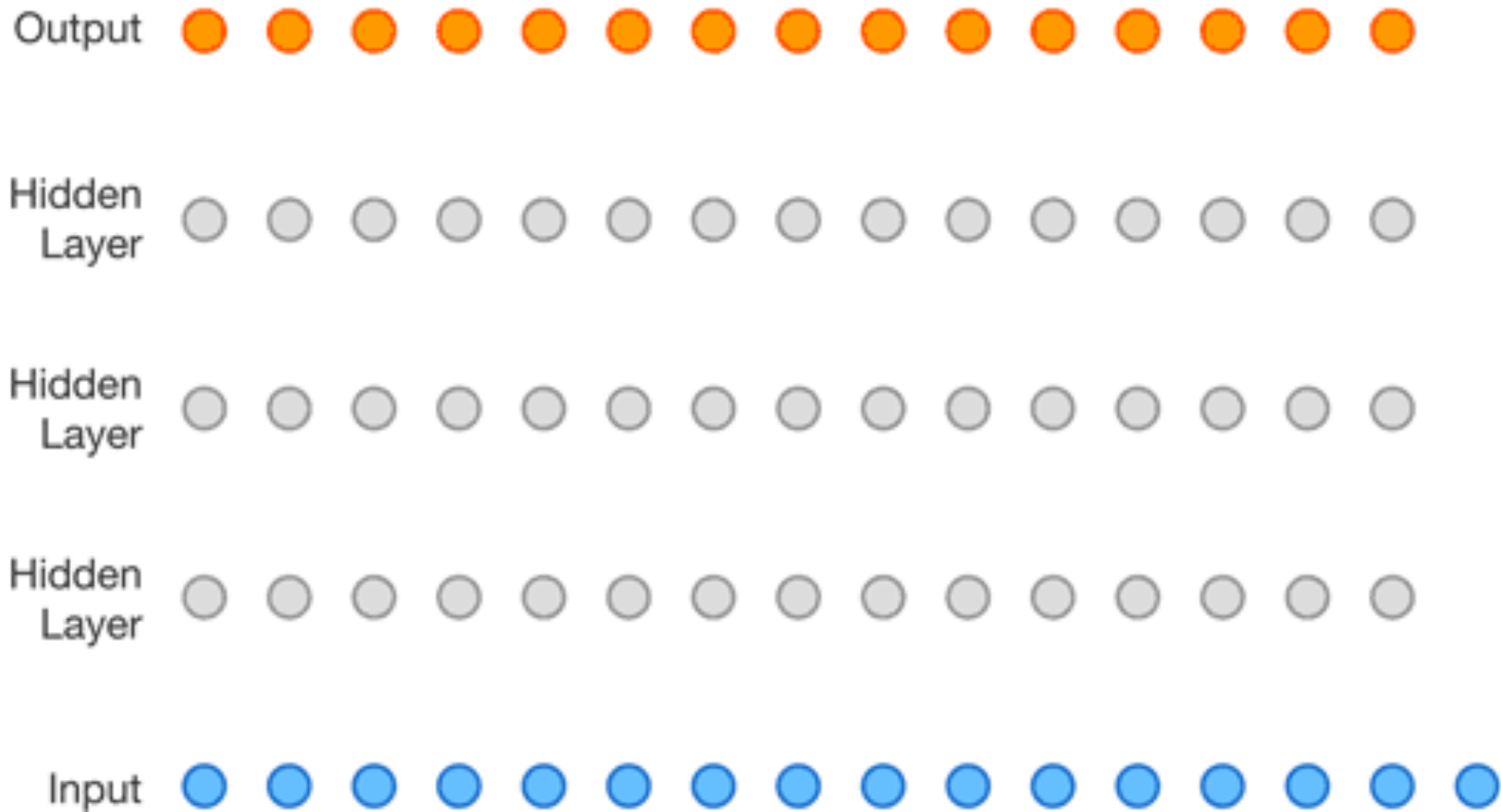
Just treat it as a next word classifier!



Autoregressive model of pixels

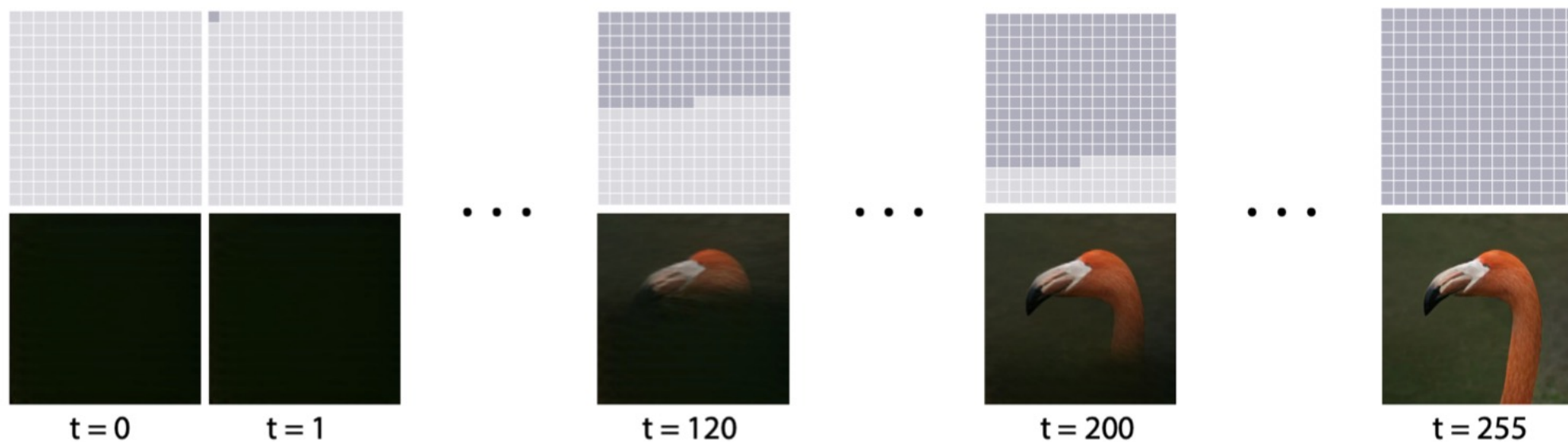




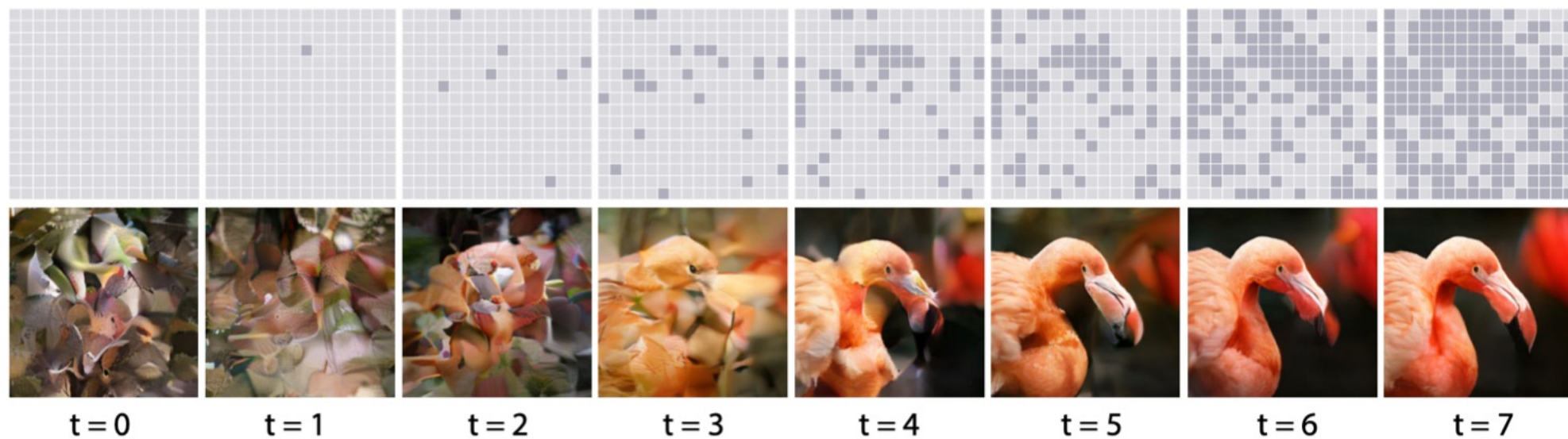


[**Wavenet**, <https://www.deepmind.com/blog/wavenet-a-generative-model-for-raw-audio/>]

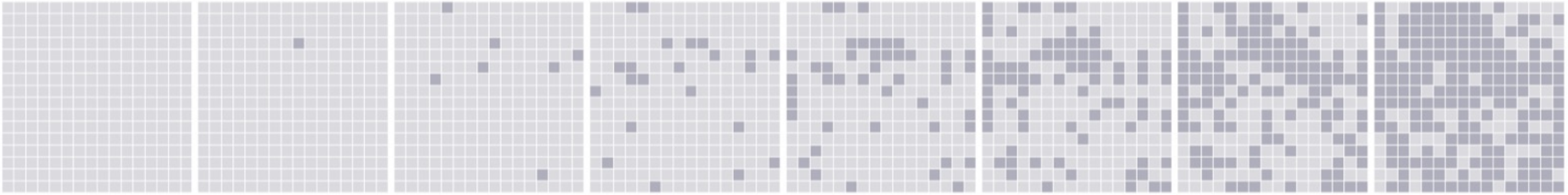
Sequential
Decoding
with Autoregressive
Transformers



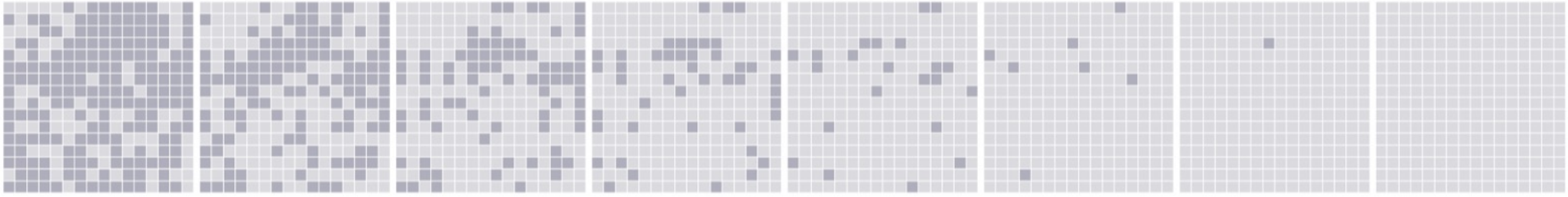
Scheduled
Parallel
Decoding
with MaskGIT



Diffusion models



Diffusion models



Corrupting the input

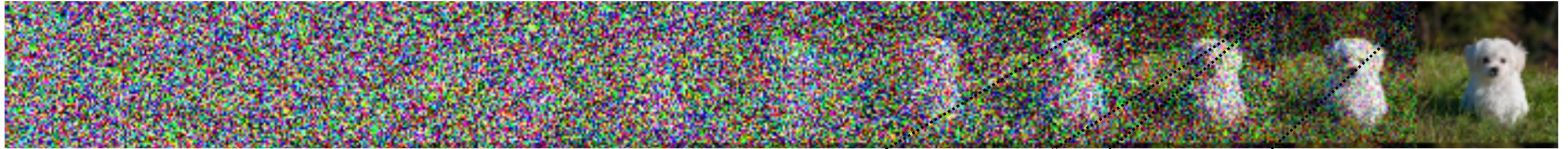


Corrupting the input

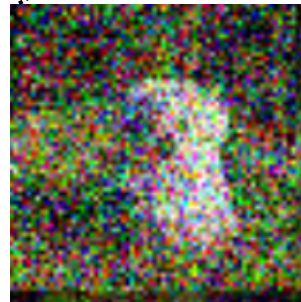
Diffusion models

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

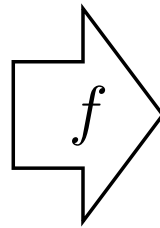
\mathbf{x}



Denoising



\mathbf{x}_t

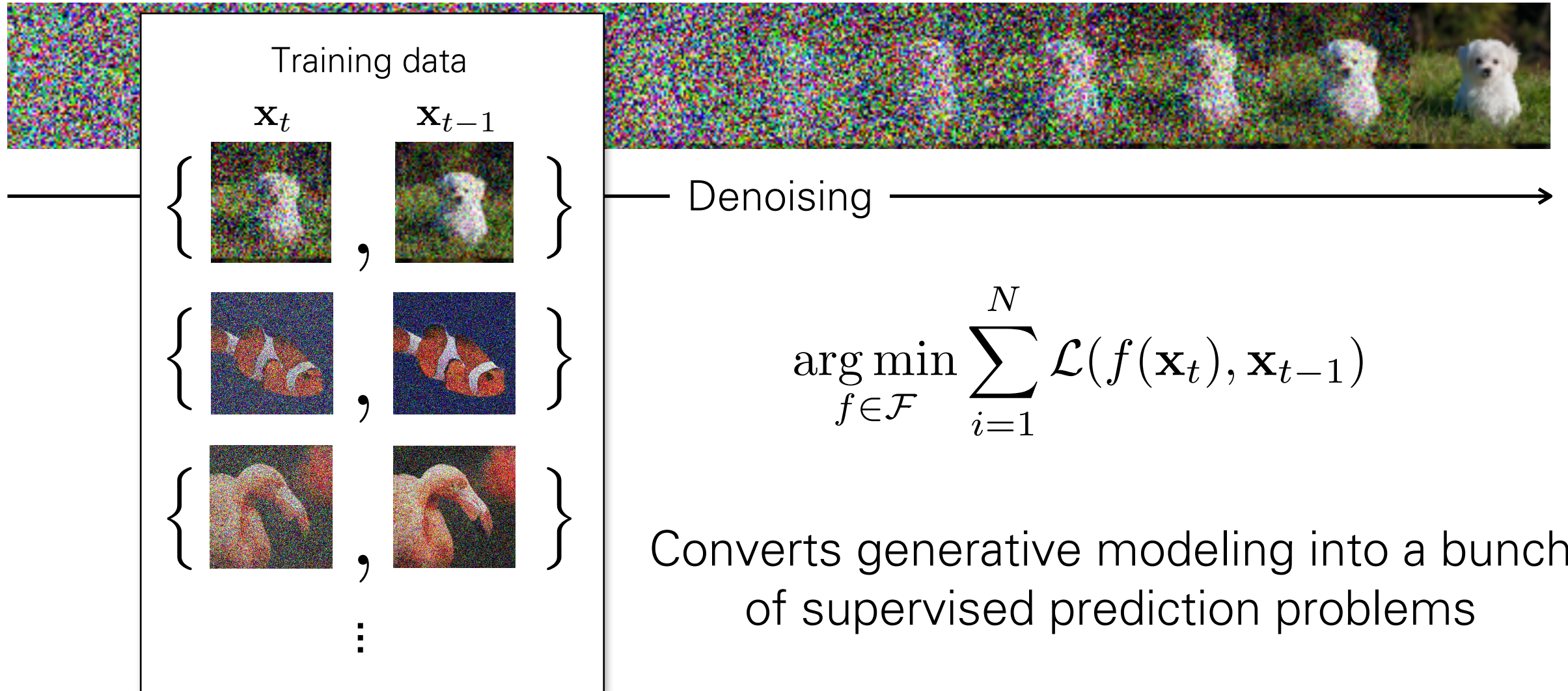


\mathbf{x}_{t-1}

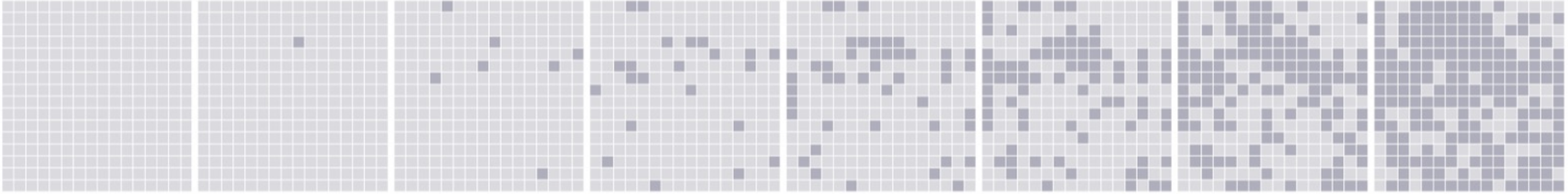
Diffusion models

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

\mathbf{x}



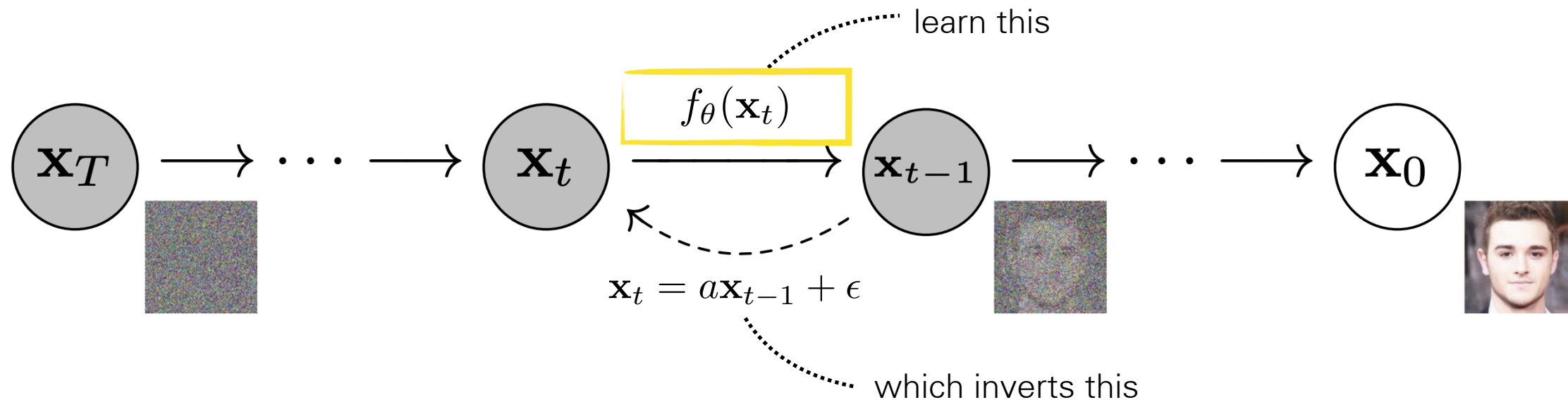
Diffusion models



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



Gaussian diffusion models



Forward process:

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

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

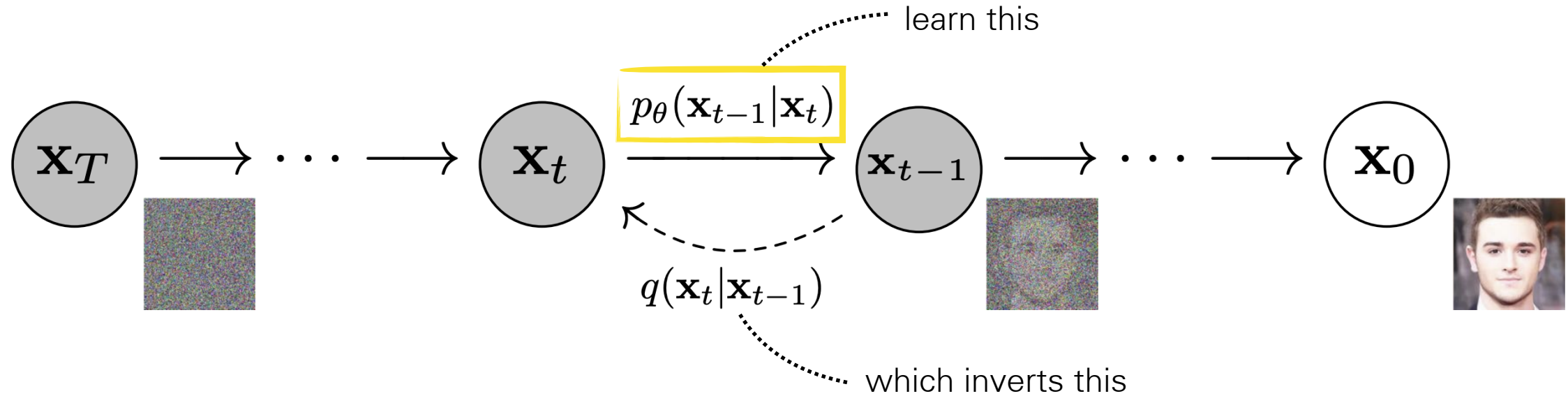
Reverse process:

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

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

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

Gaussian diffusion models



Forward process:

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

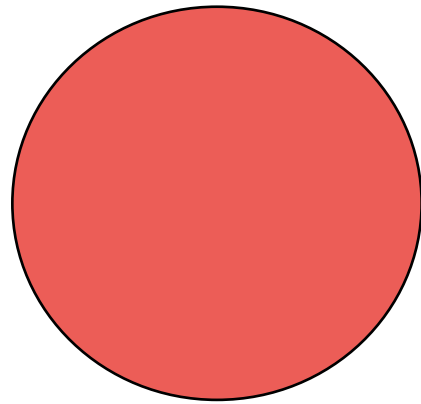
Reverse process:

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

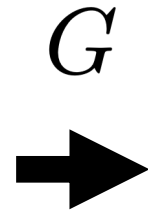
a , b , and c are hyperparameters
(see Ho, Jain, Abbeel for one
way to set them)

Deep generative models are distribution transformers

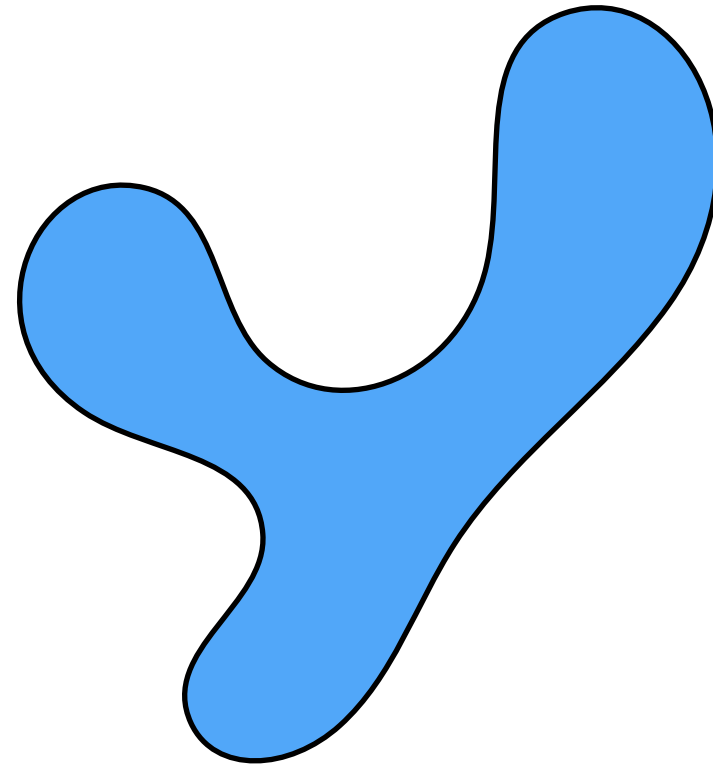
Prior distribution



$p(z)$

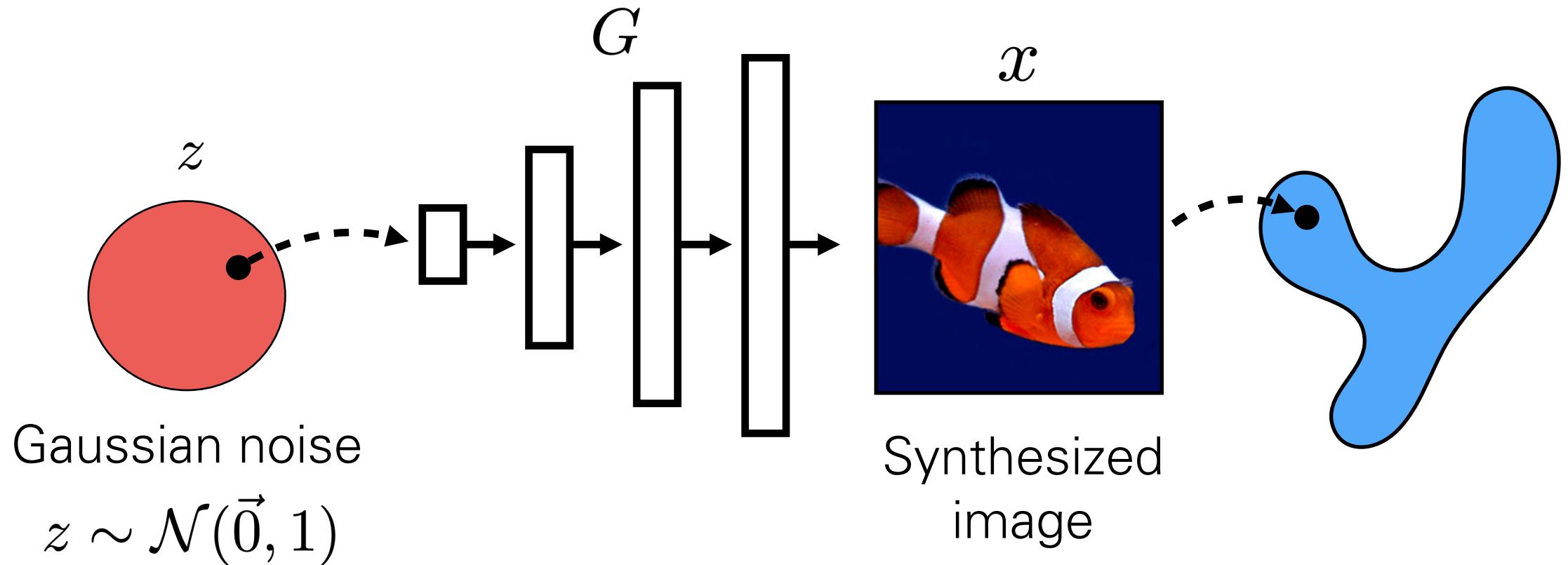


Target distribution

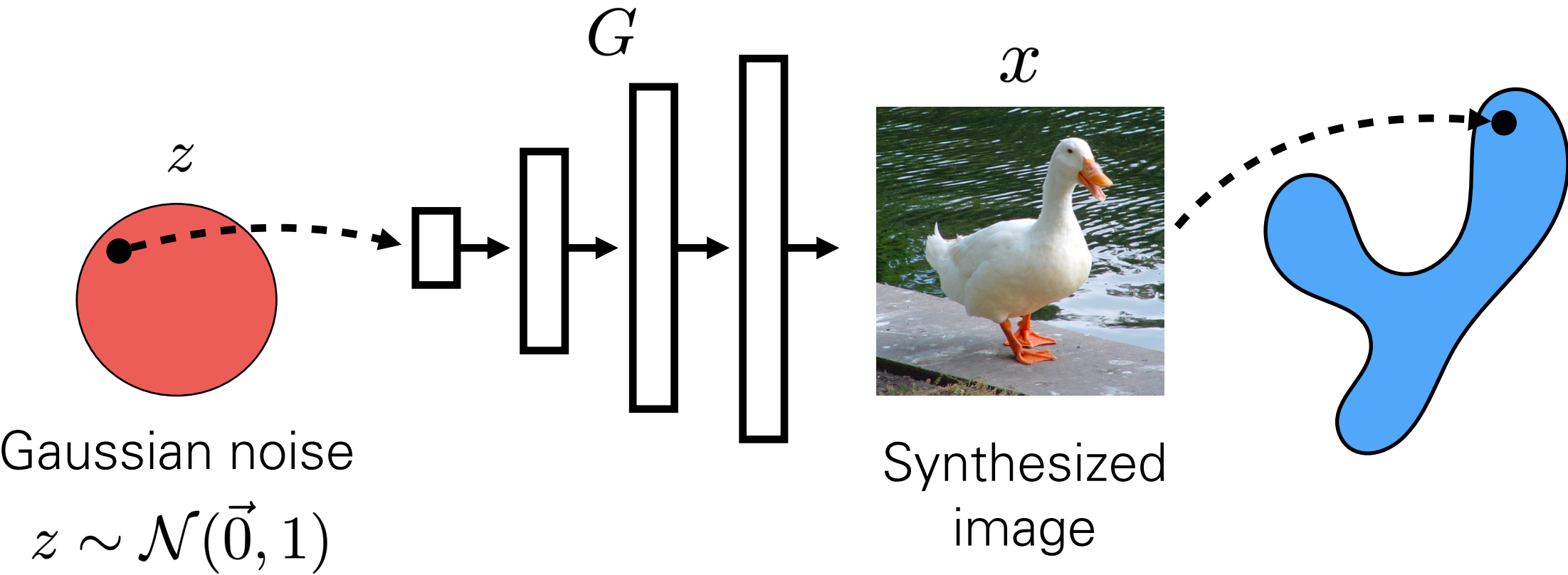


$p(x)$

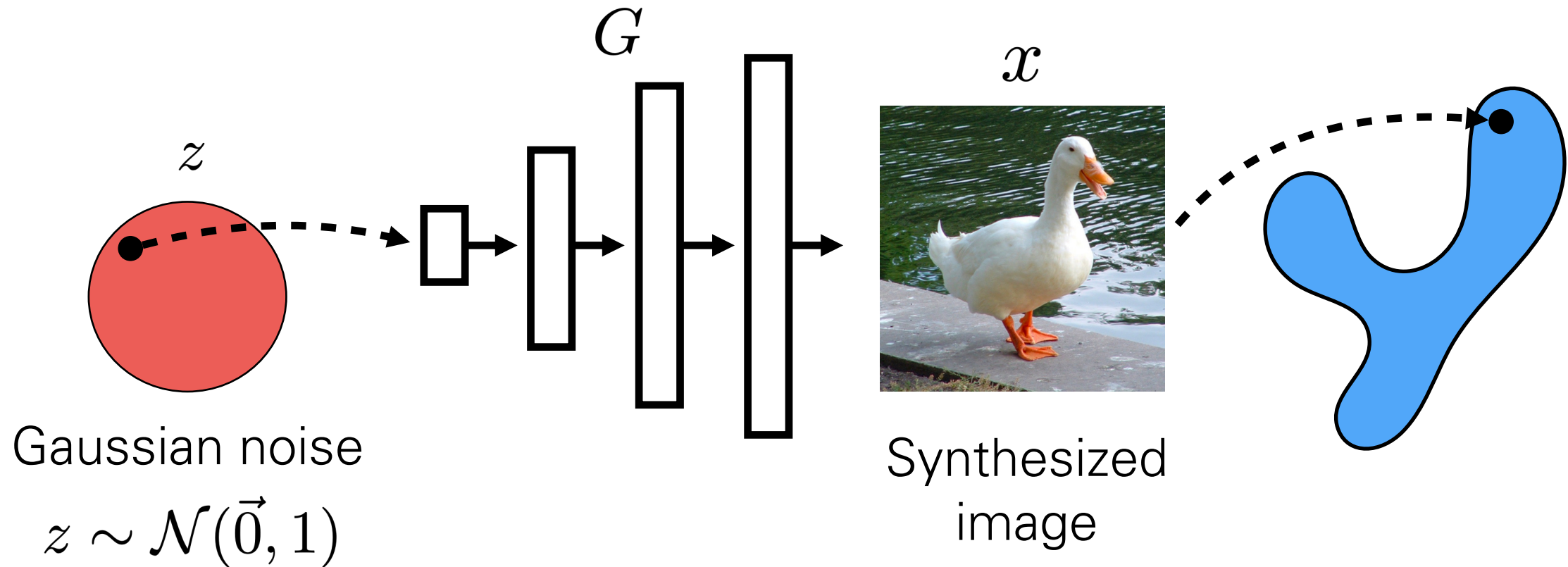
Deep generative models are distribution transformers

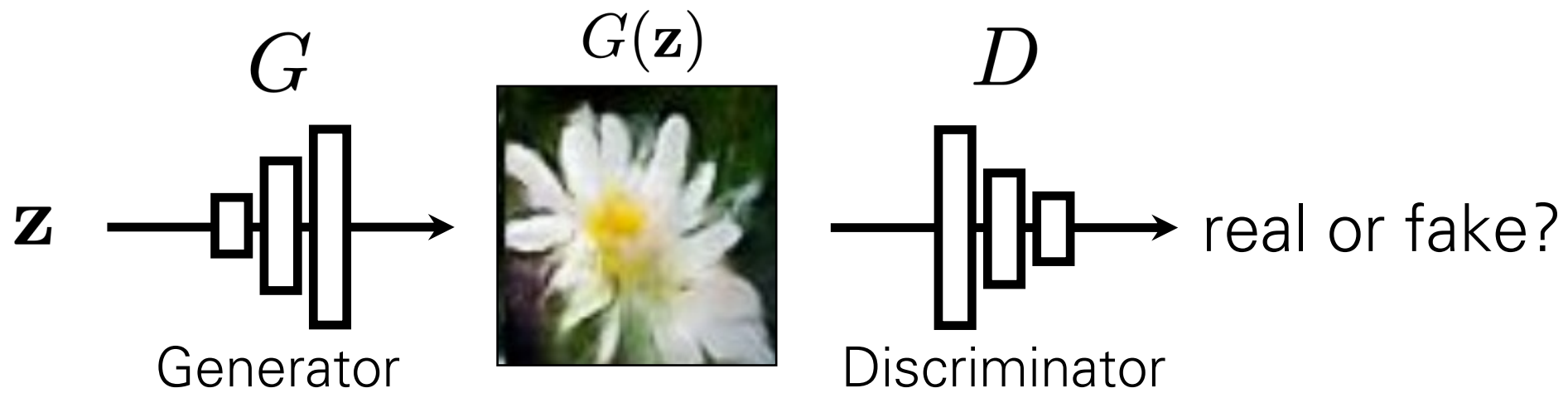


Deep generative models are distribution transformers



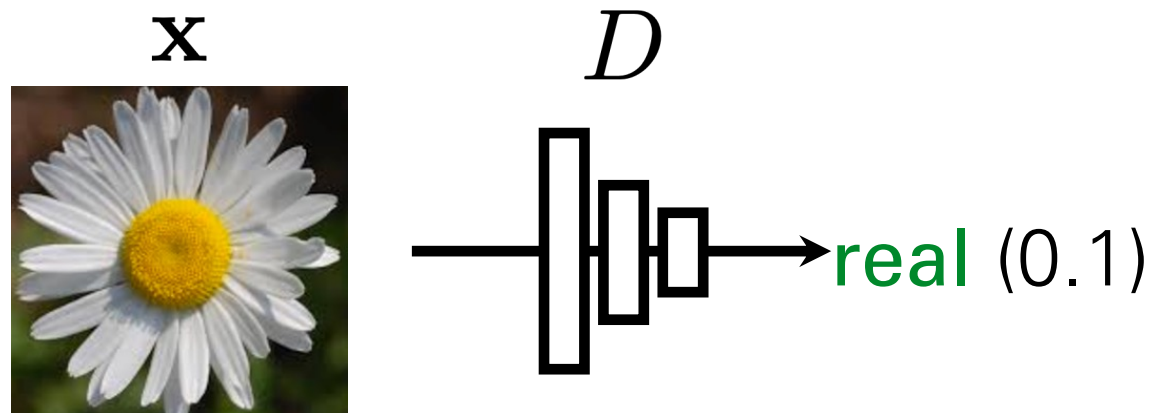
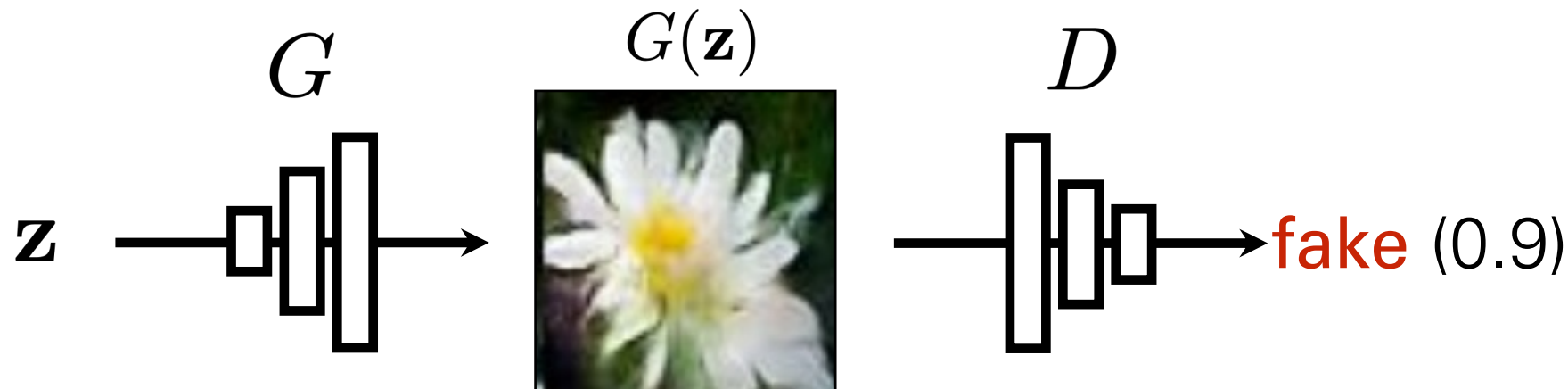
Generative Adversarial Networks (GANs)



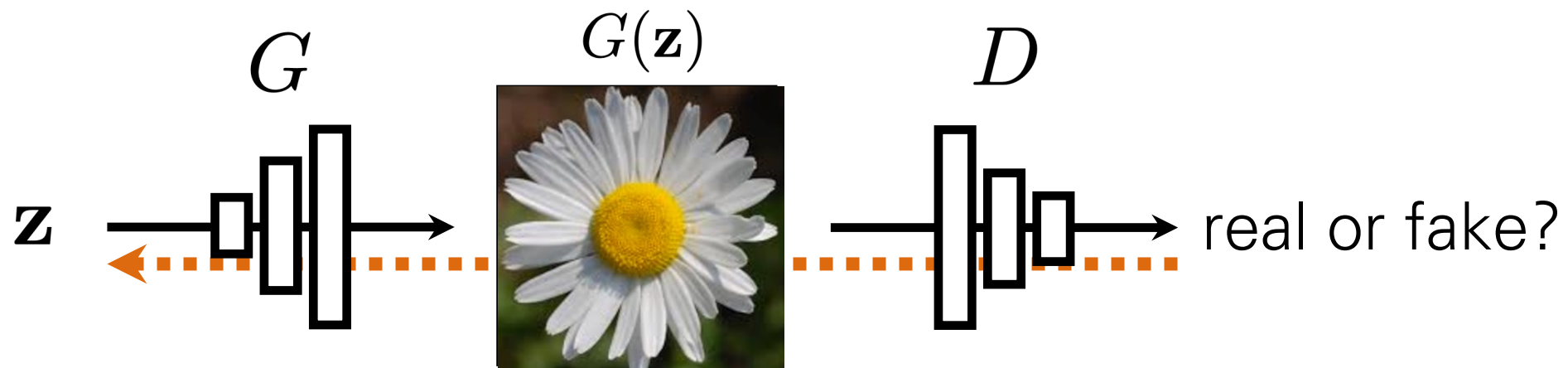


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

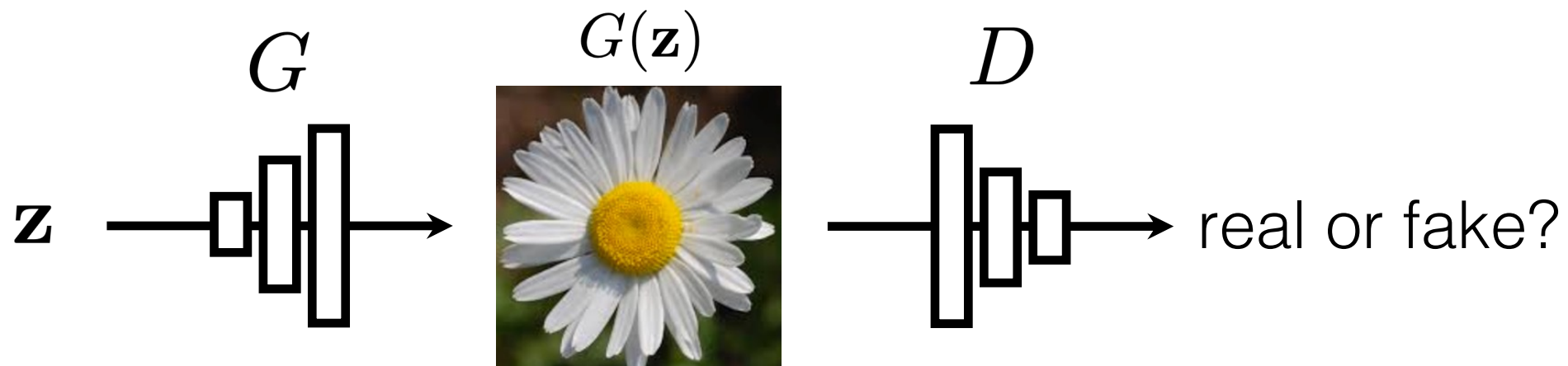


$$\arg \max_D \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\log D(G(\mathbf{z})) + \log (1 - D(\mathbf{x}))]$$



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

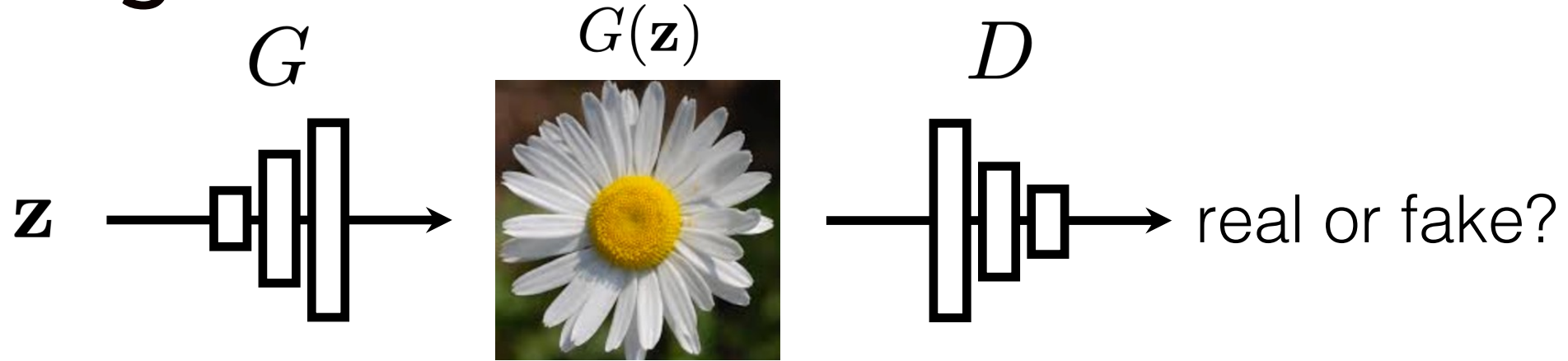
$$\arg \min_G \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\log D(G(\mathbf{z})) + \log (1 - D(\mathbf{x}))]$$



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

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

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

Cross-entropy
loss for binary
classification

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)

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

Real data

Noise vector used to

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

$$J^{(G)} = -\frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} [\log (1 - D_{\omega}(G_{\theta}(\mathbf{z})))]$$

**Important question is
"Does this converge??"**

cross-entropy loss for binary classification

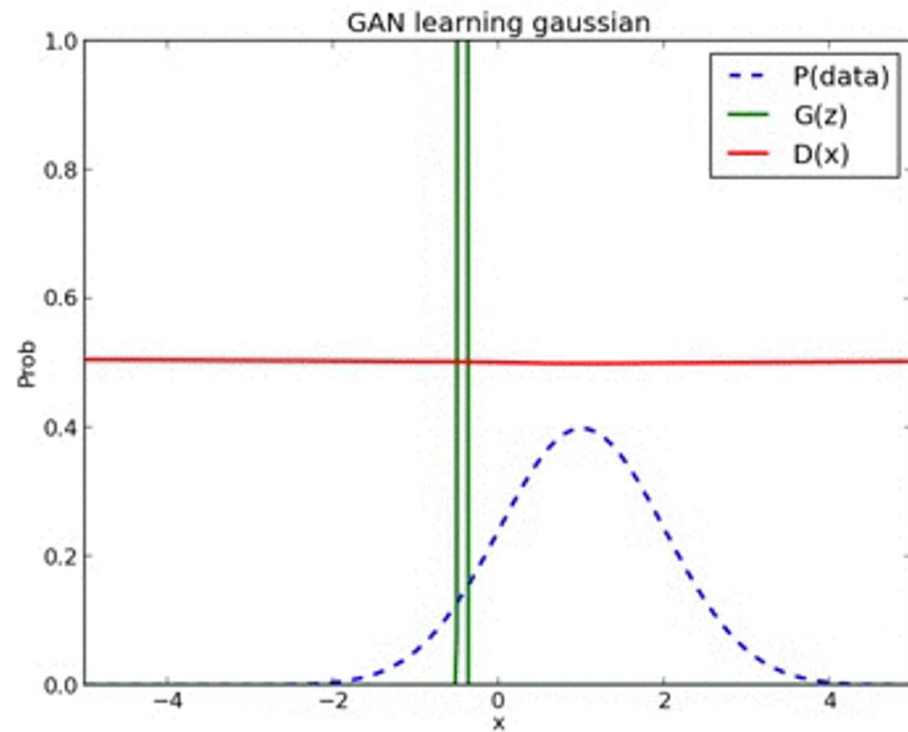
probability

of the discriminator being mistaken

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

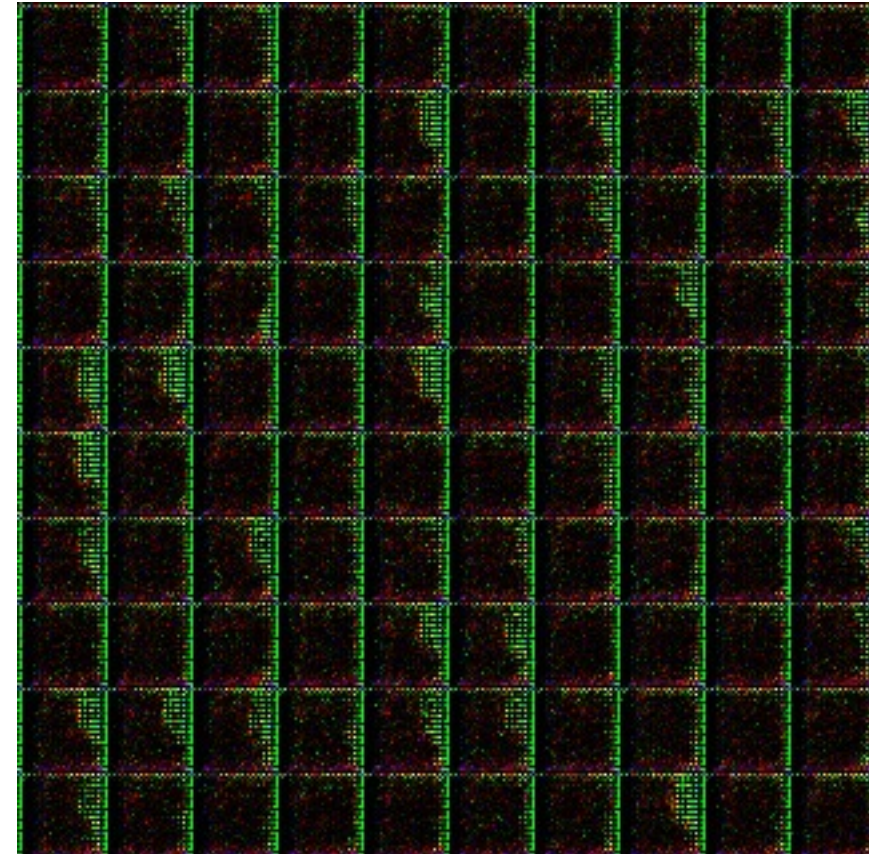
Training Procedure

(Goodfellow et al., 2014)



Source: Alec Radford

Generating 1D points



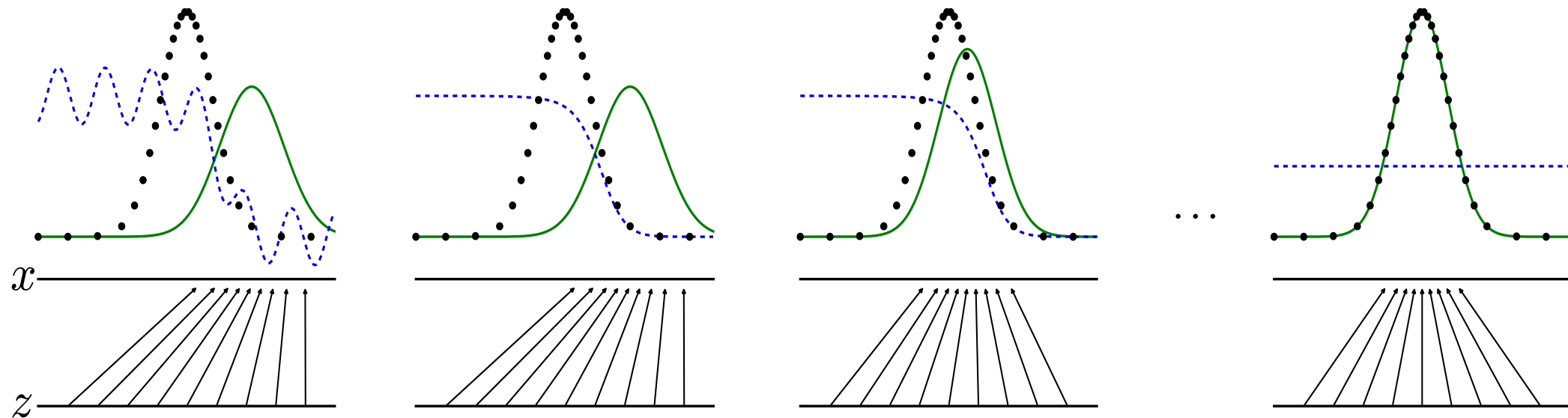
Source: OpenAI blog

Generating images

Training Procedure

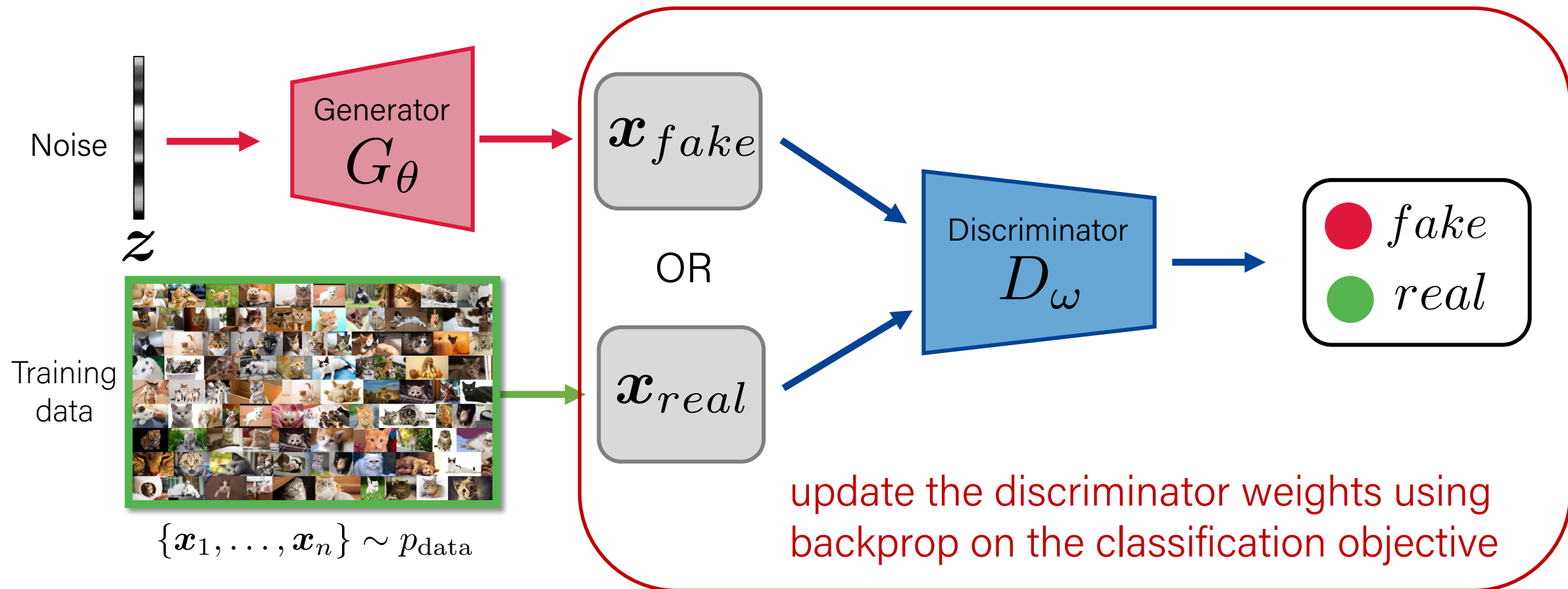
(Goodfellow et al., 2014)

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



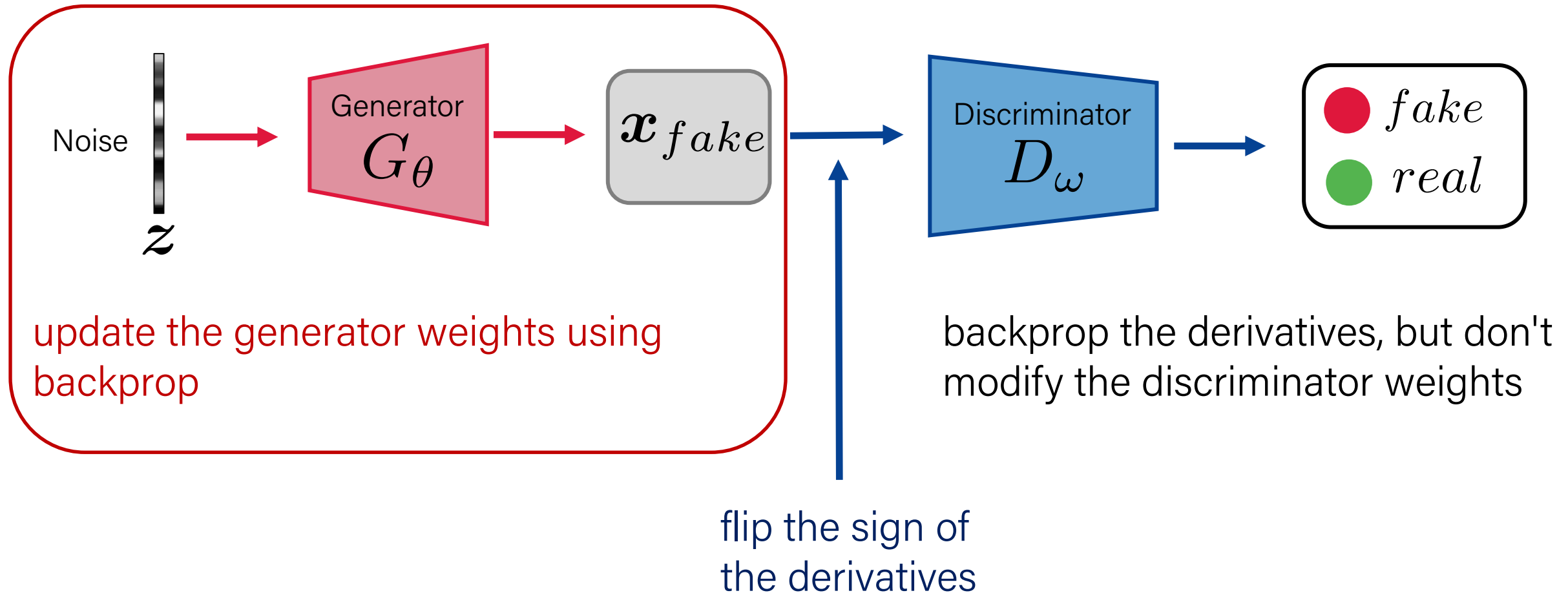
Training Procedure

- Updating the discriminator:



Training Procedure

- Updating the generator:



Early Results

(Goodfellow et al., 2014)

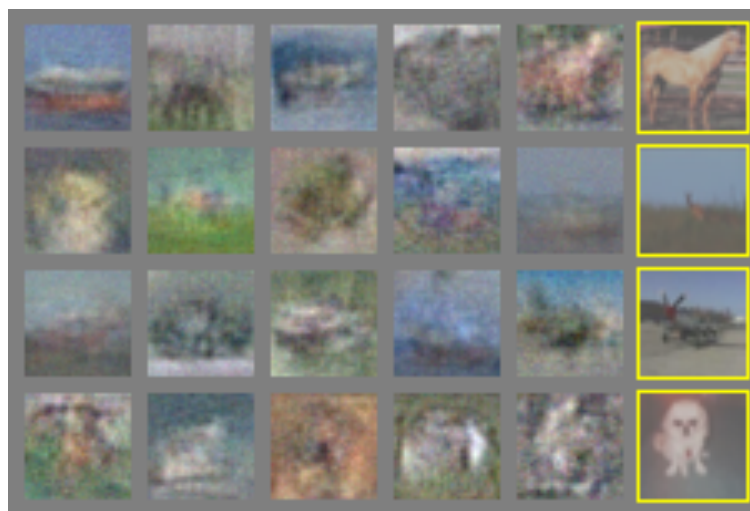
- The generator uses a mixture of rectifier linear activations and/or sigmoid activations
- The discriminator net used maxout activations.



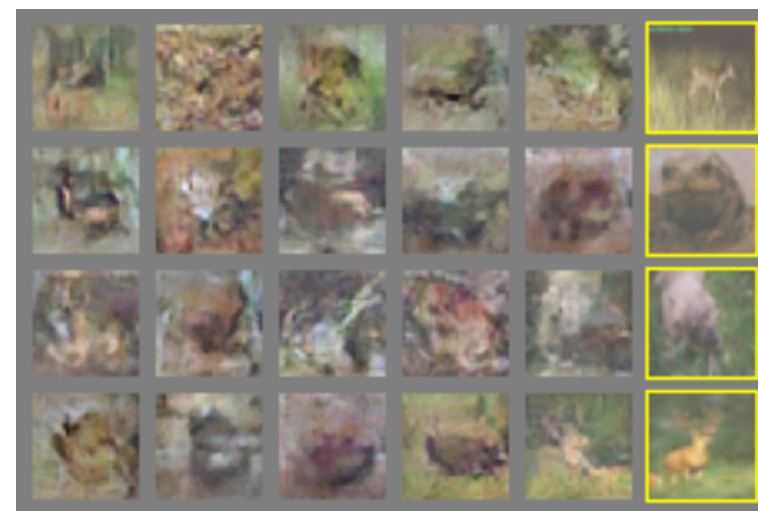
MNIST samples



TFD samples



CIFAR10 samples
(fully-connected model)



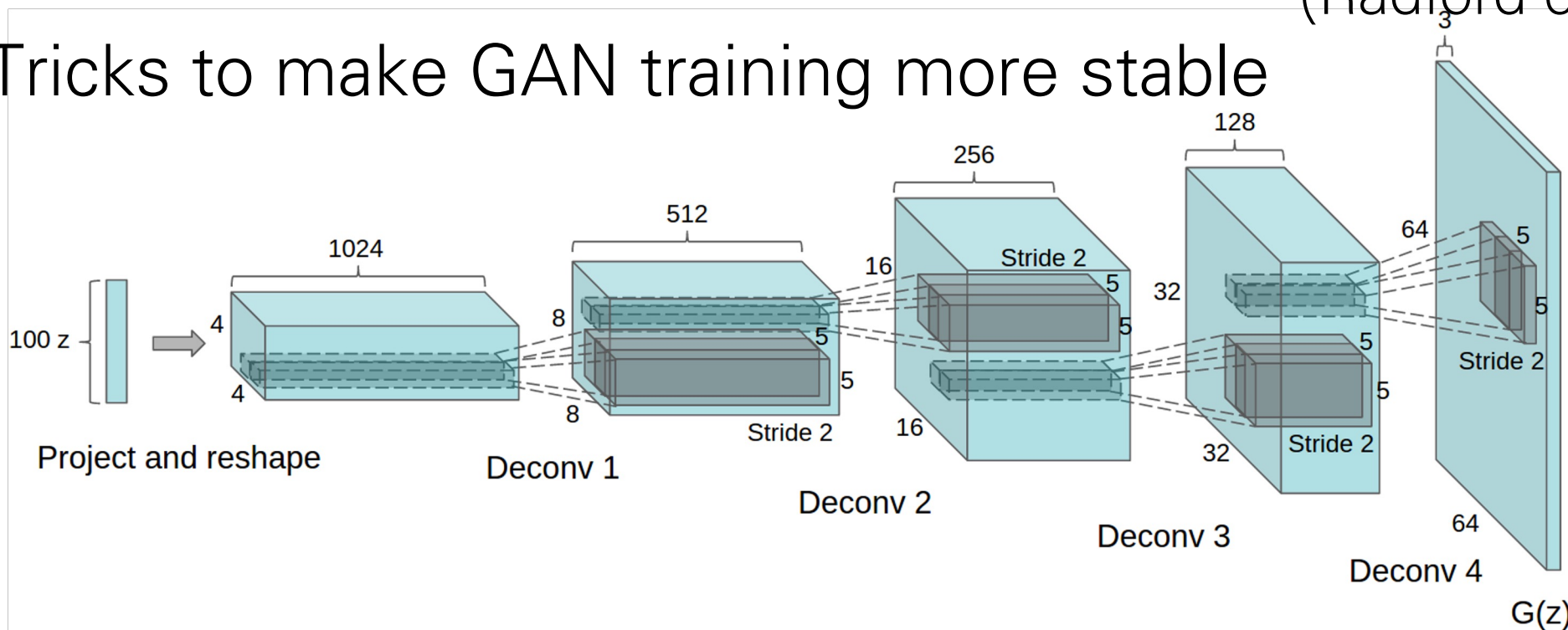
CIFAR10 samples
*(convolutional discriminator,
deconvolutional generator)*

Deep Convolutional GANs (DCGAN)



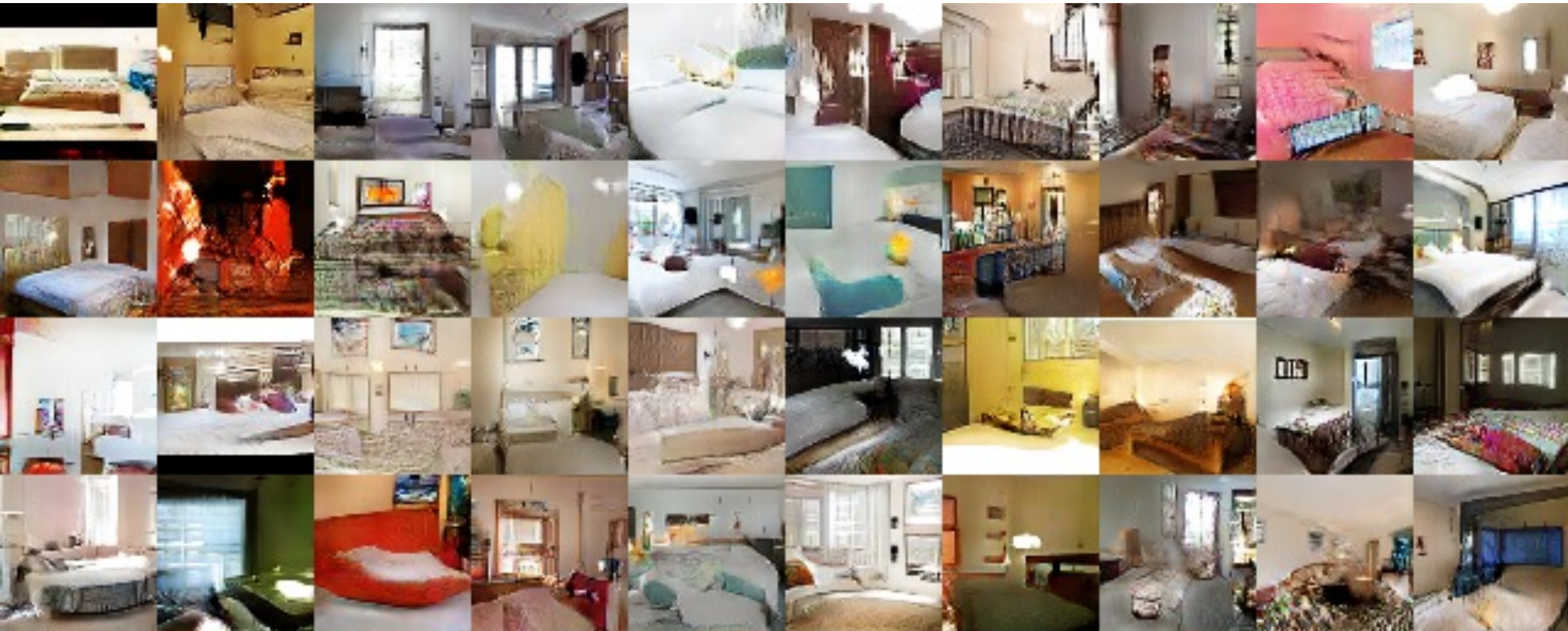
(Radford et al., 2015)

- 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 ($lr=0.0002$, $b1=0.5$)

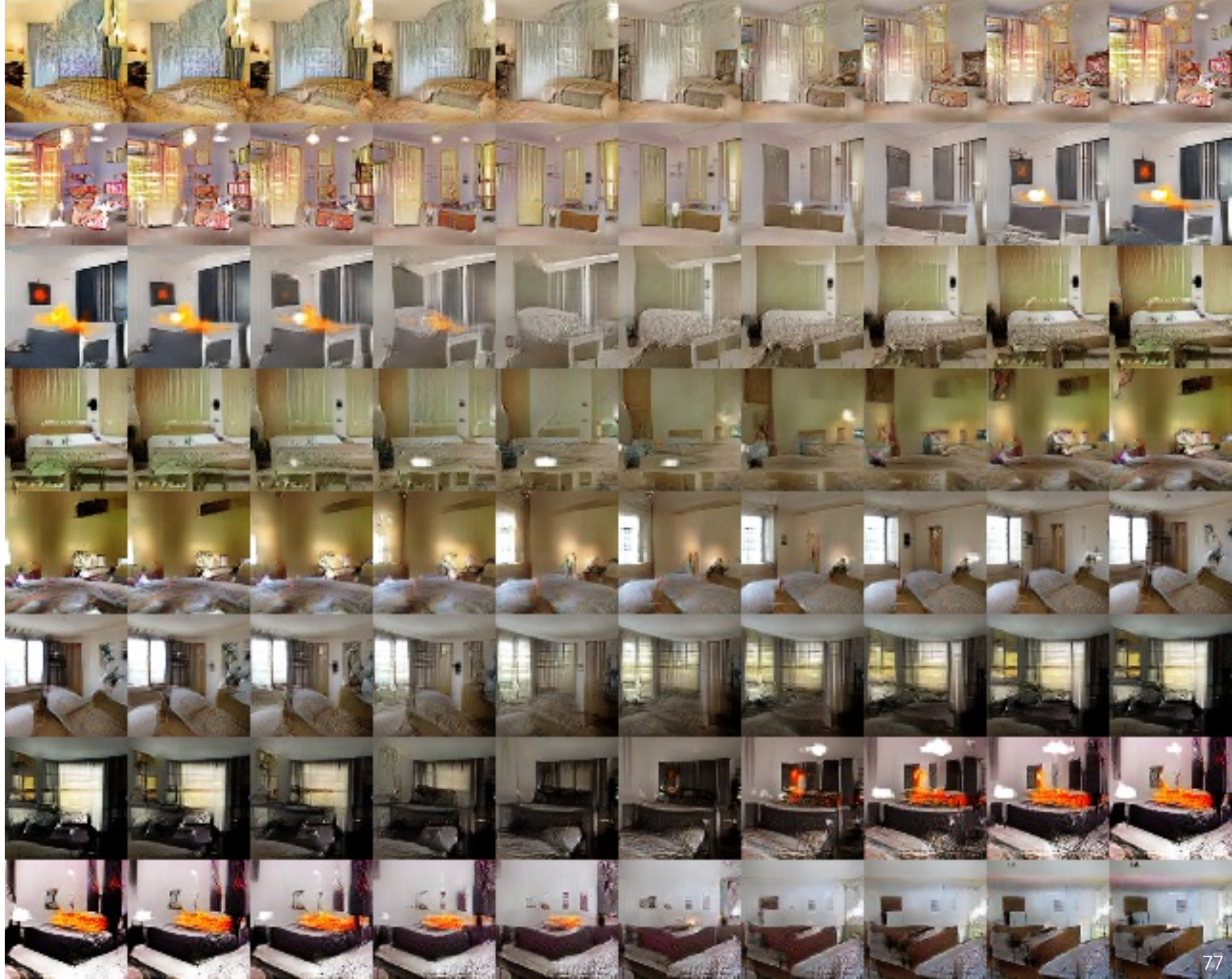
DCGAN for LSUN Bedrooms 64×64 pixels ~3M images (Radford et al., 2015)



Walking over the latent space

(Radford et al., 2015)

- Interpolation suggests non-overfitting behavior



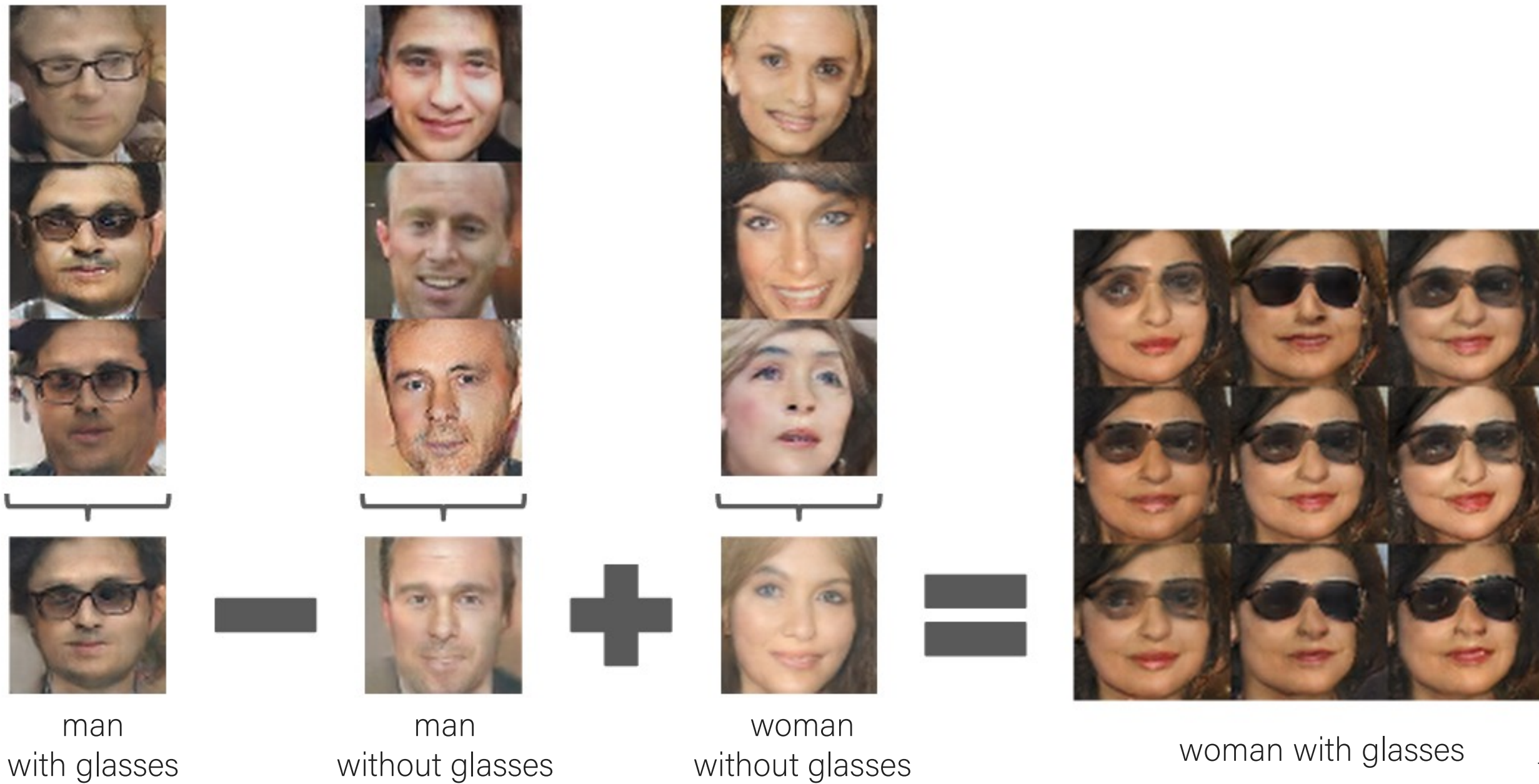
Walking over the latent space

(Radford et al, 2015)



Vector Space Arithmetic

(Radford et al., 2015)

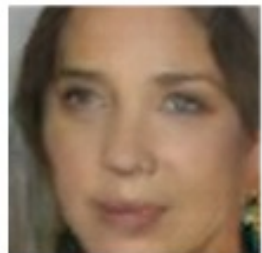


Vector Space Arithmetic

(Radford et al., 2015)



smiling woman



neutral woman

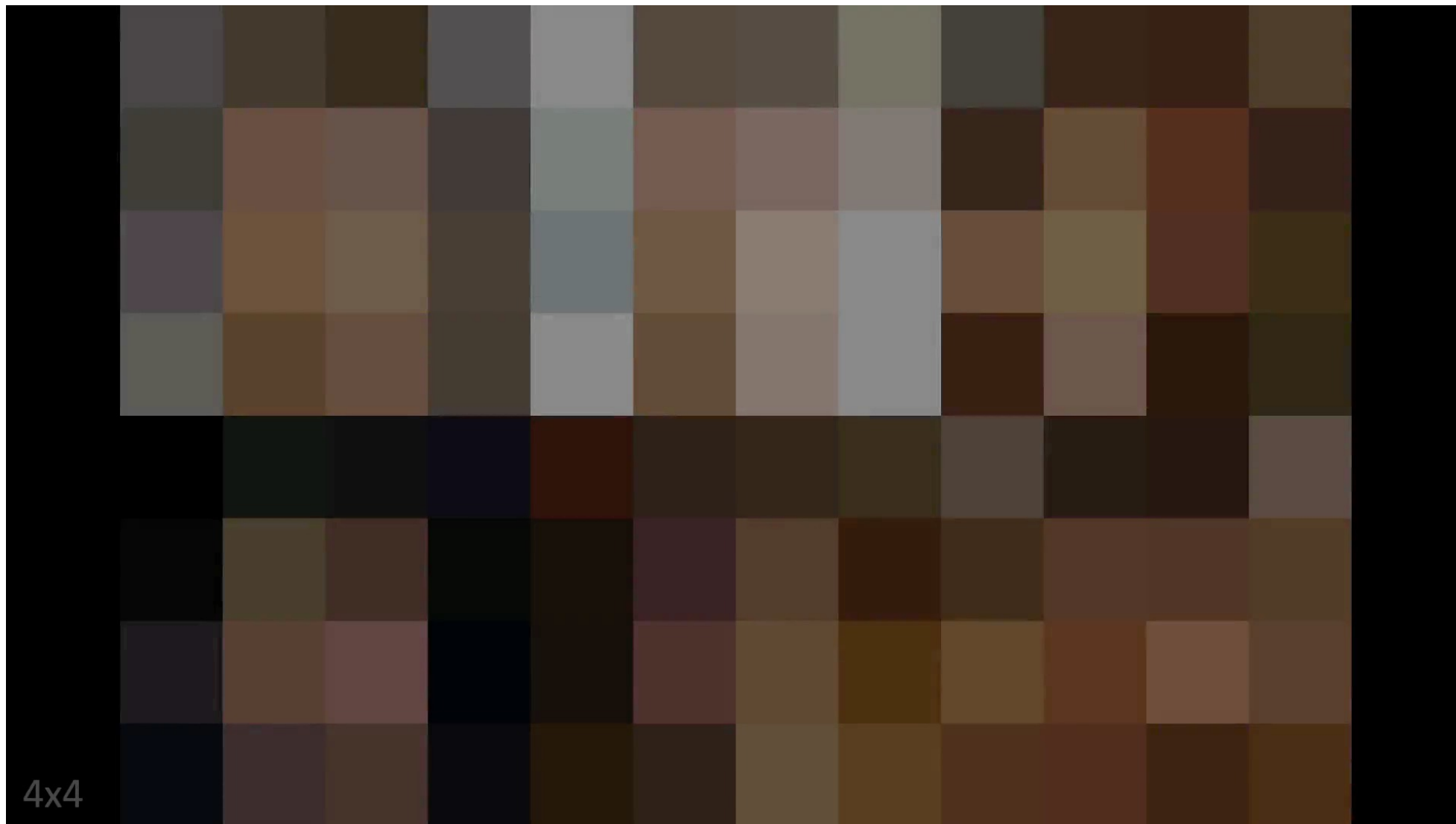


neutral man



smiling man

Progressive GANs (Karras et al., 2018)



Progressive GANs (Karras et al., 2018)



CelebA-HQ
random interpolations

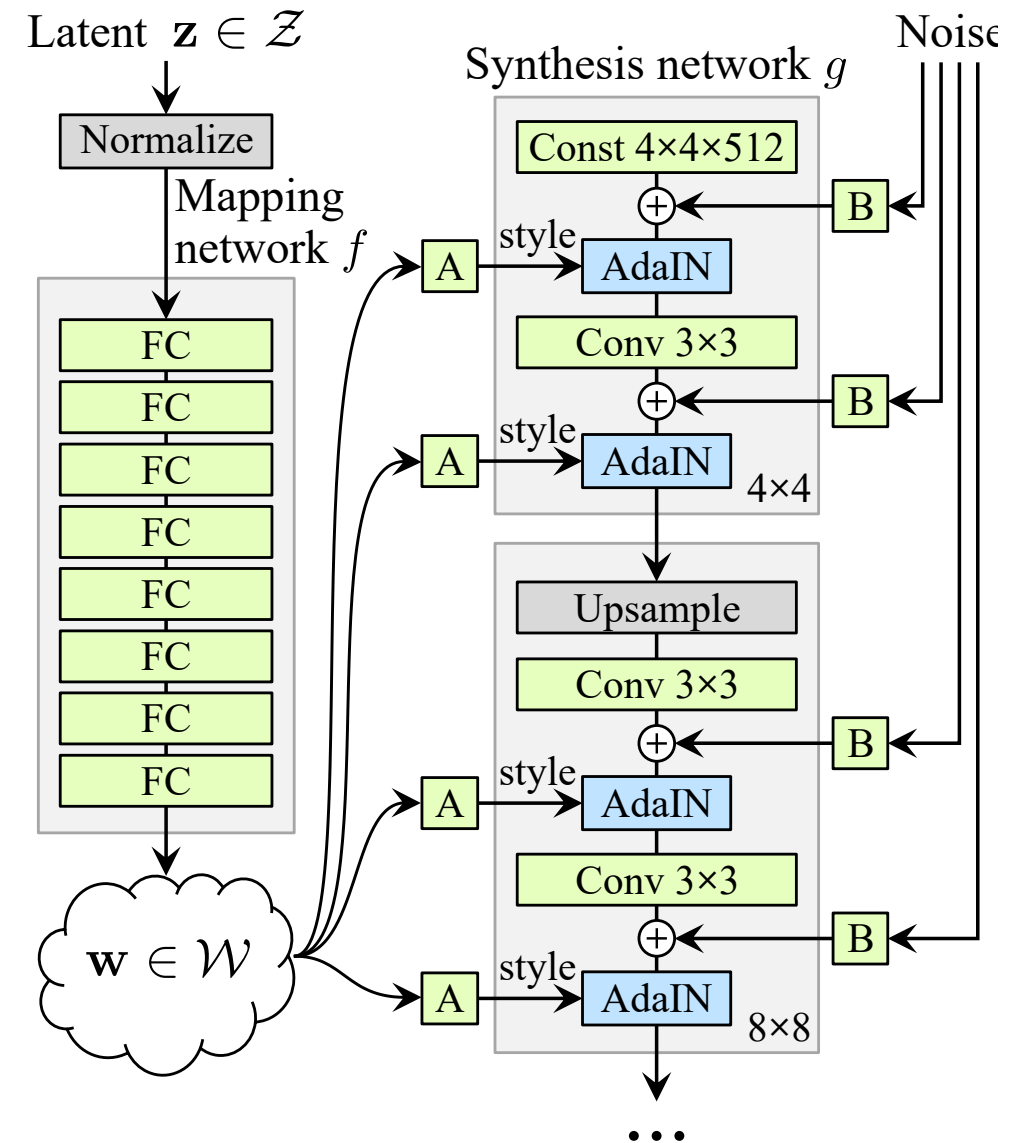
Samples from BigGAN

[Brock et al. 2018]



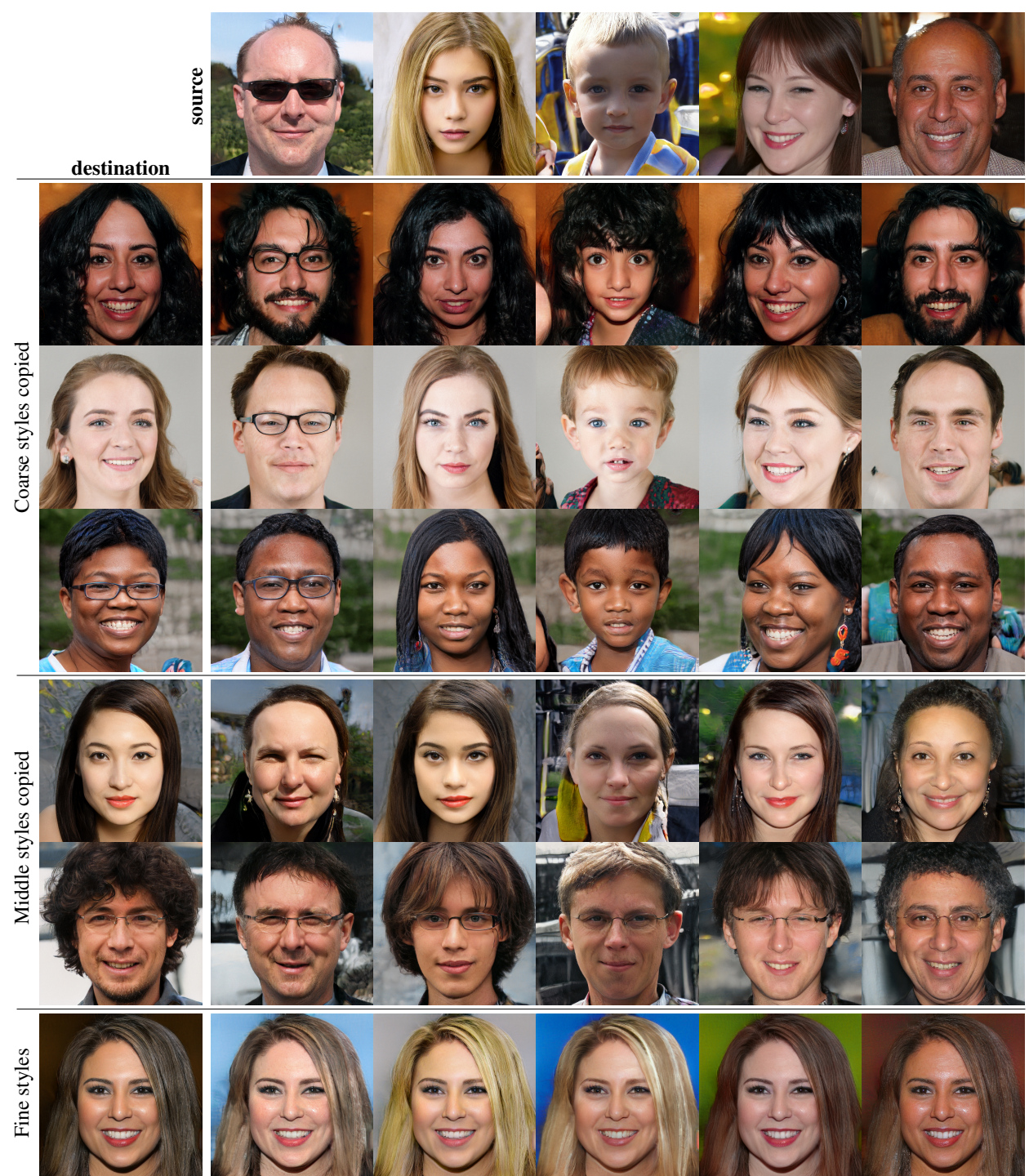
StyleGANs (Karras et al., 2019)

- 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)

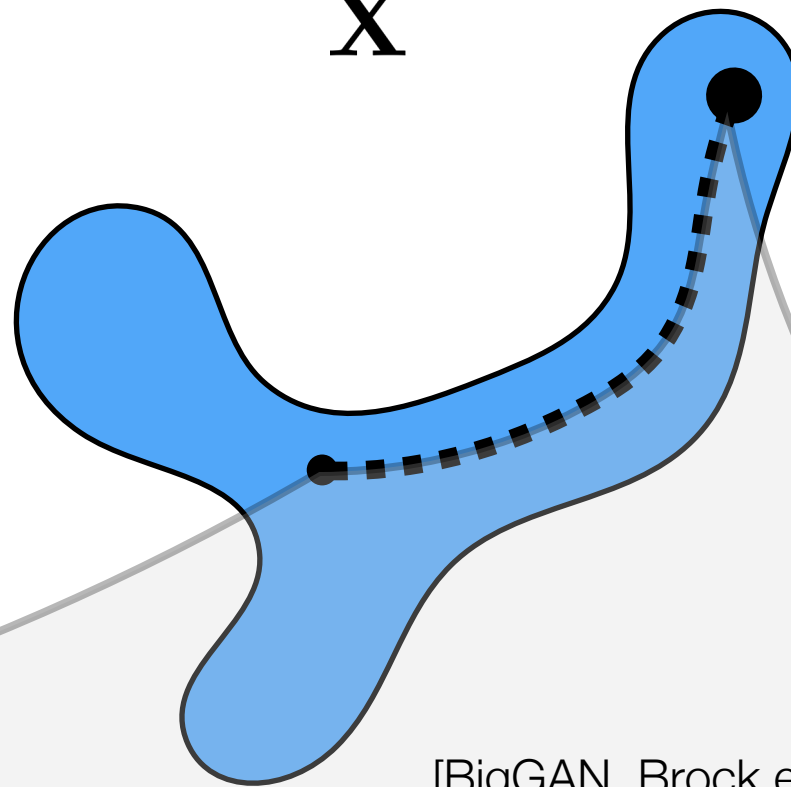
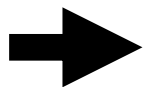
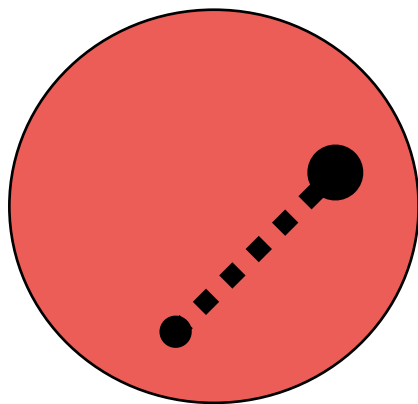


Latent space
(Gaussian)

Data space
(Natural image manifold)

\mathbf{z}

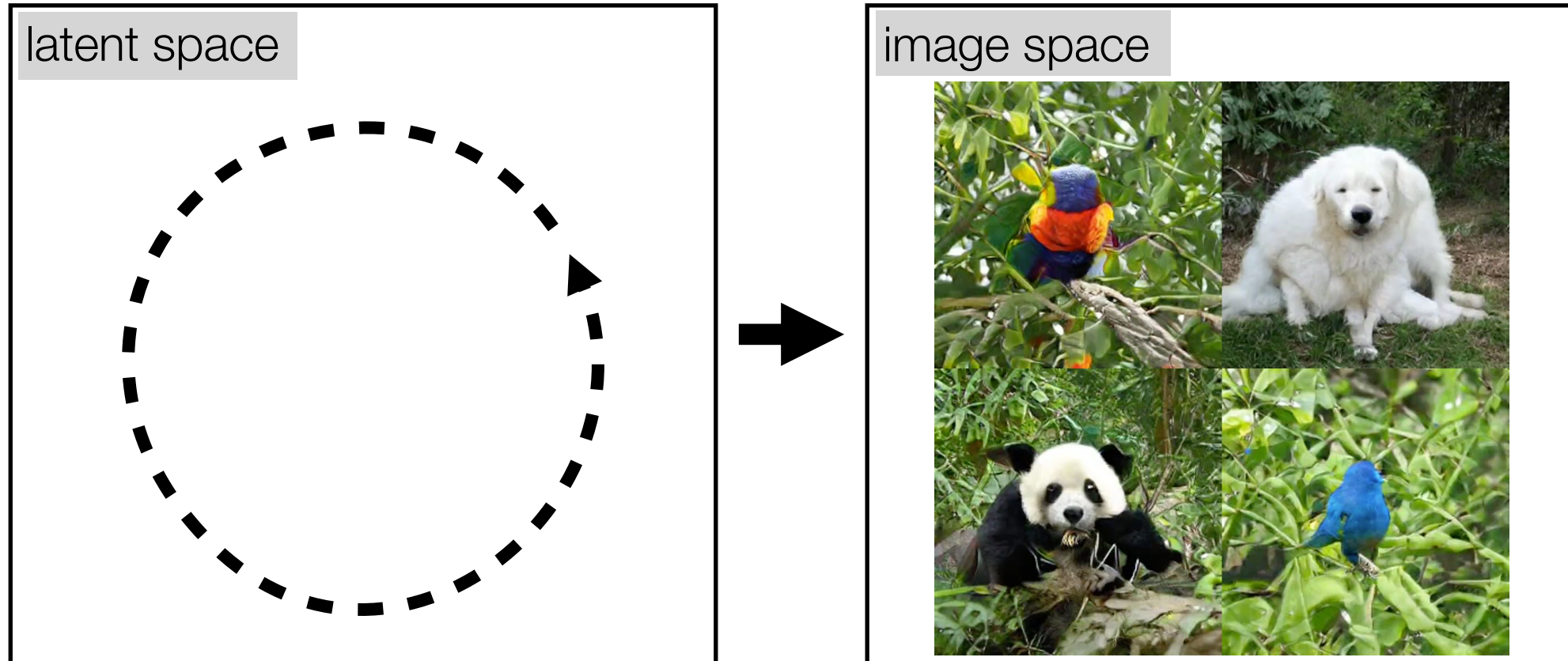
\mathbf{X}



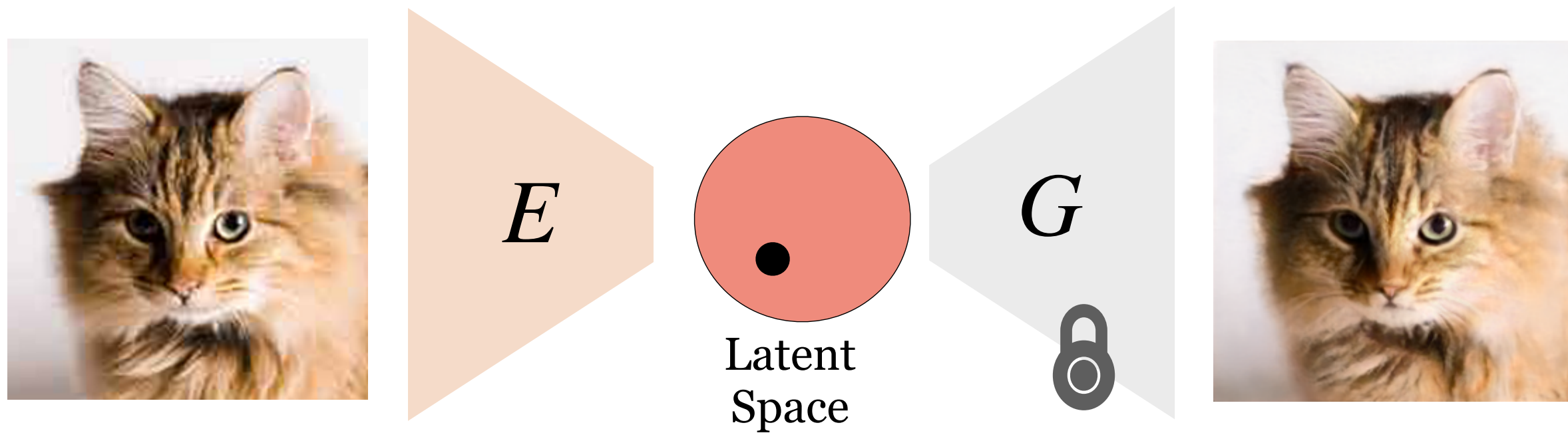
[BigGAN, Brock et al. 2018]



Generative models organize the manifold of natural images

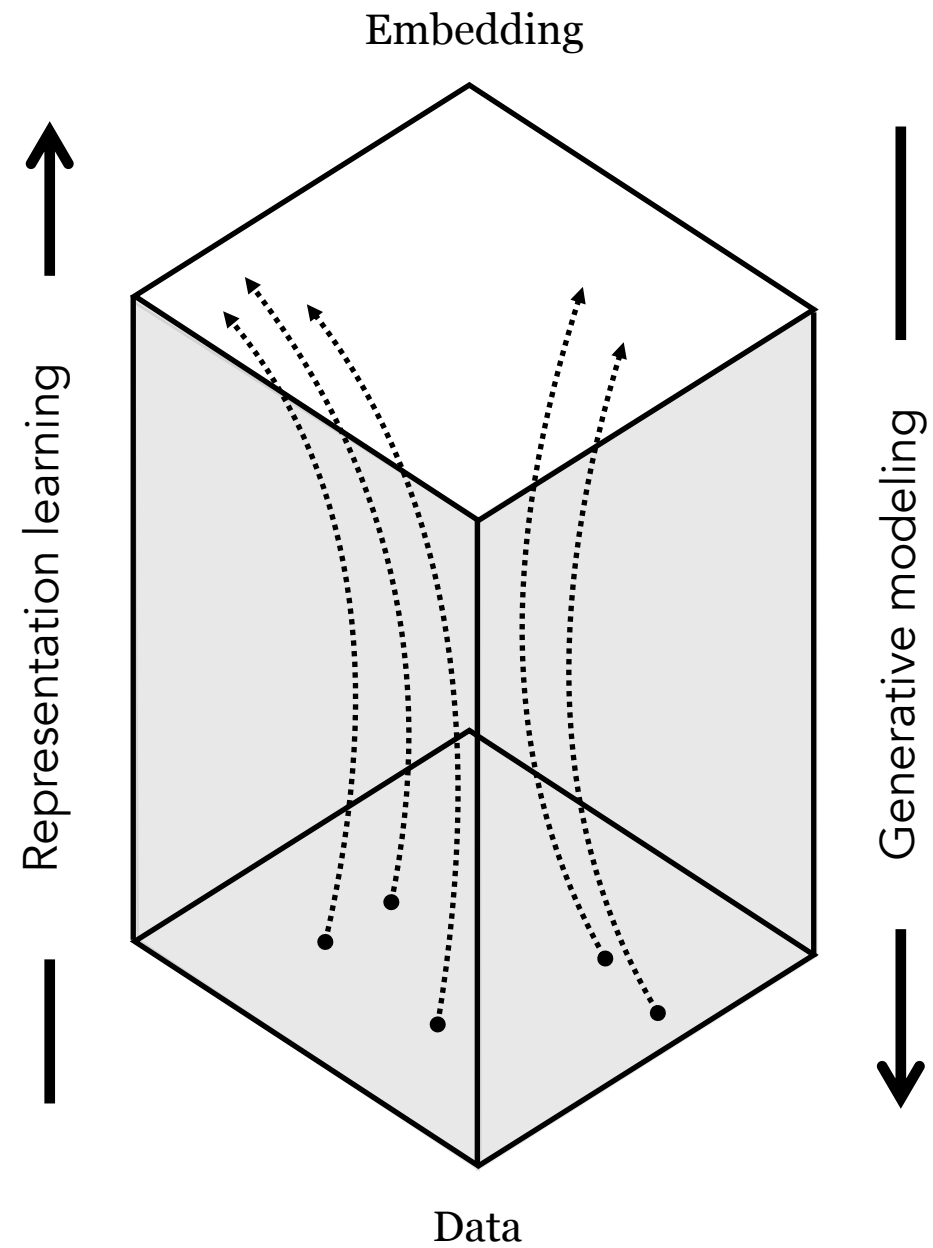


Projecting images into GAN latent space



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

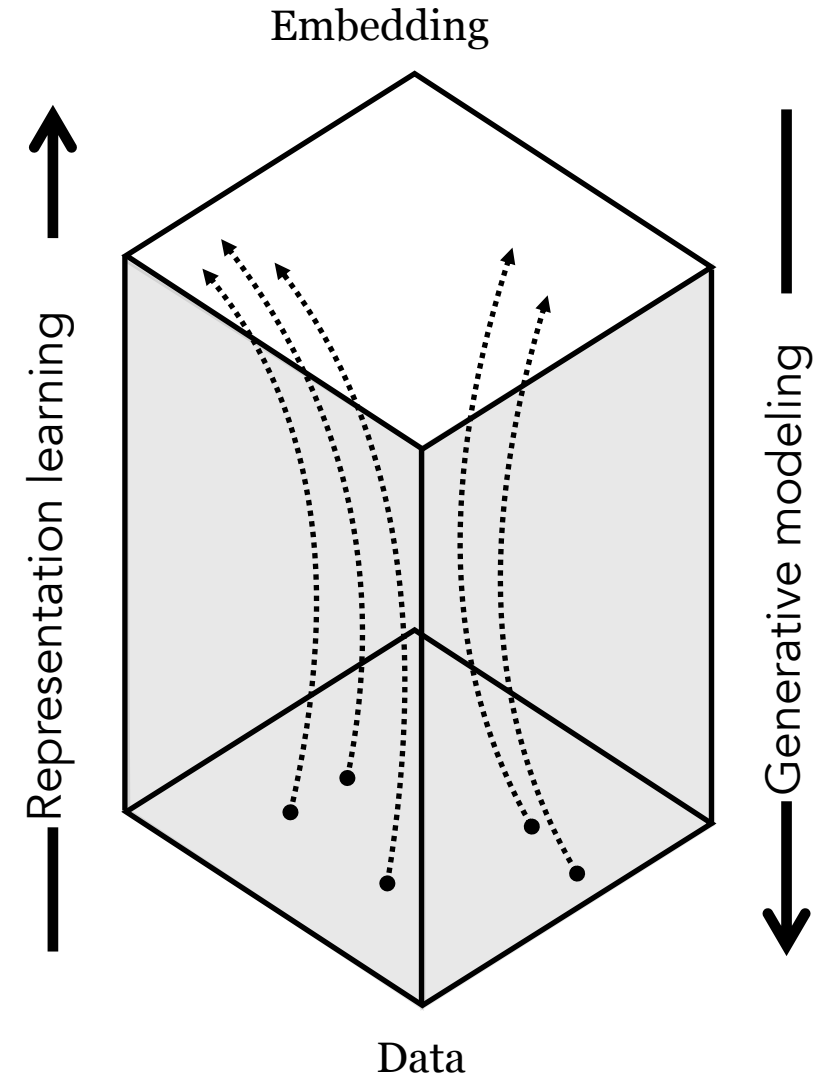
Representation Learning



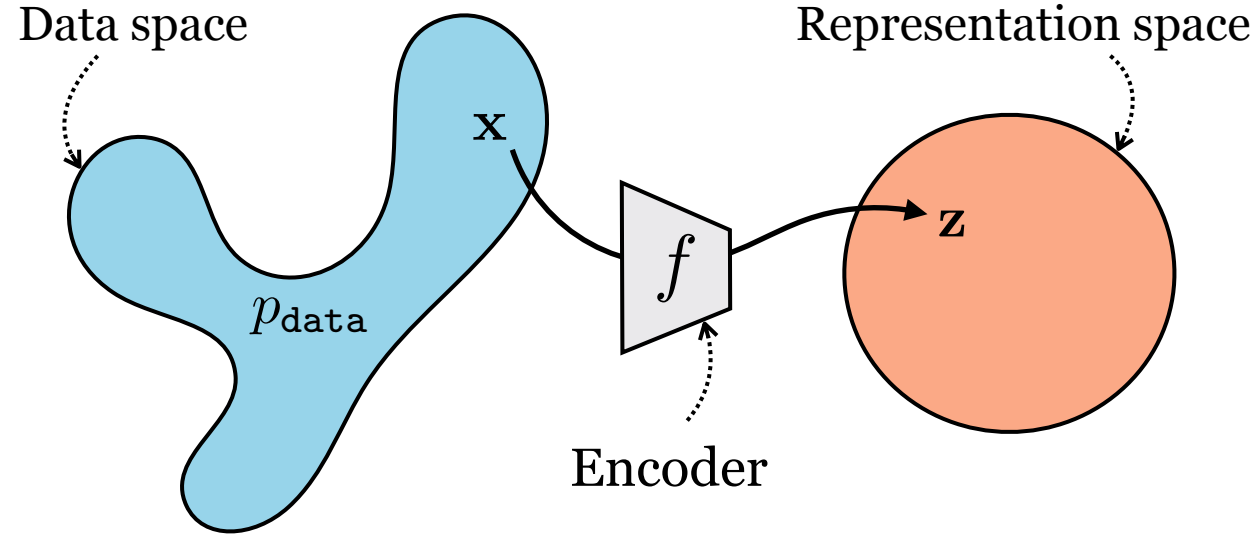
Generative modeling vs Representation learning

Representation learning:
mapping data to abstract representations
(analysis)

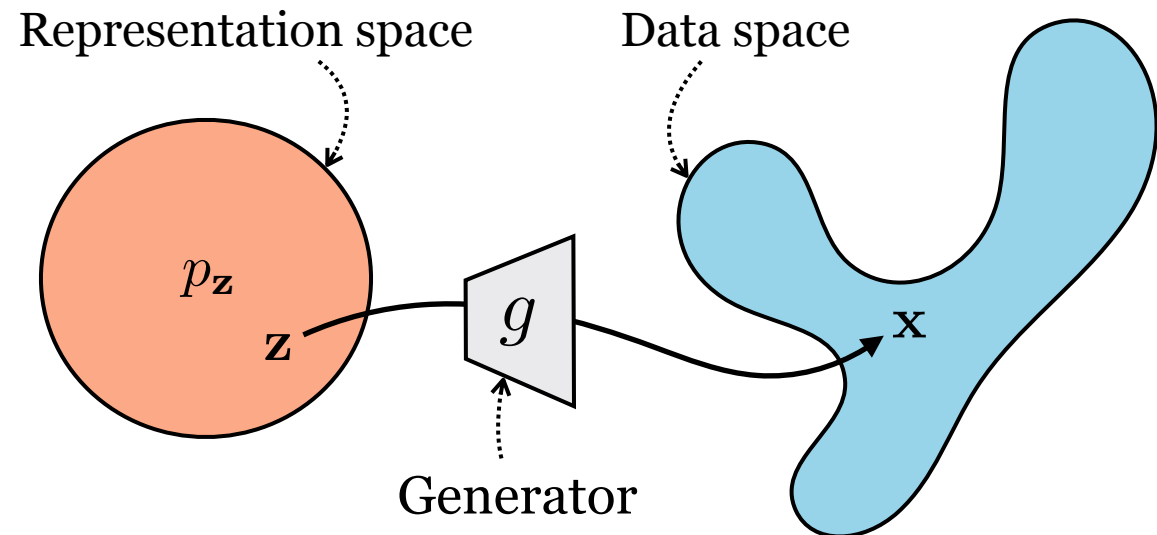
Generative modeling:
mapping abstract representations to data
(synthesis)



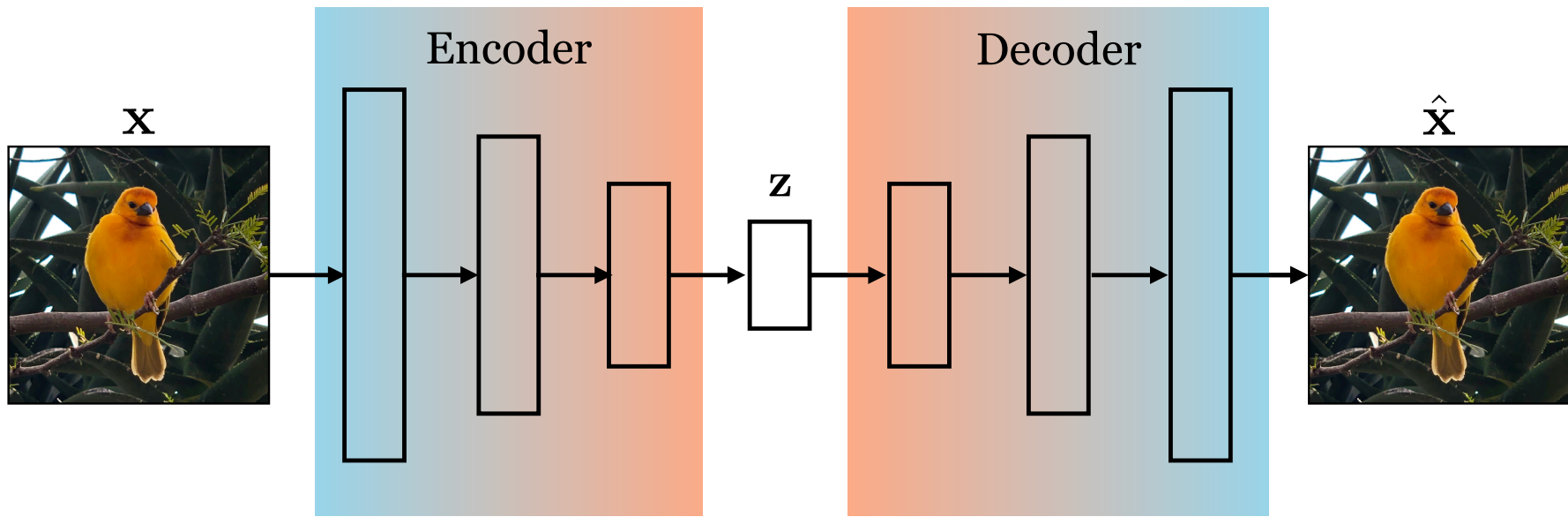
Representation learning



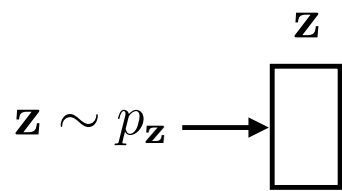
Generative modeling



Autoencoder

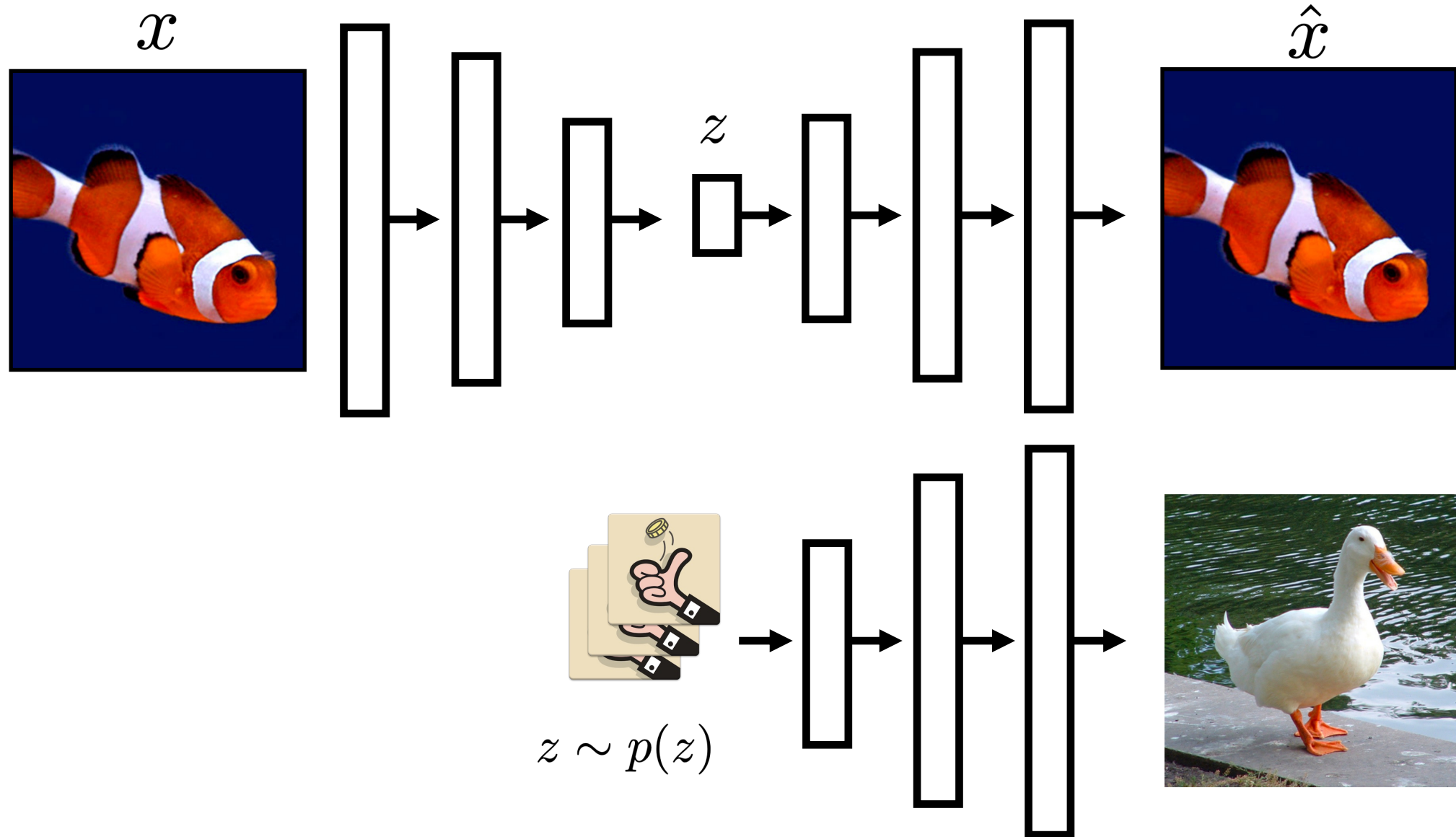


Generative model



⋮

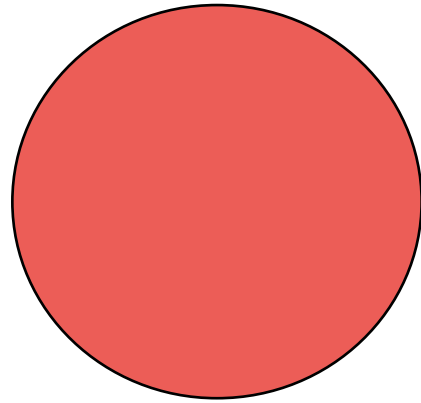
Autoencoder → Generative model



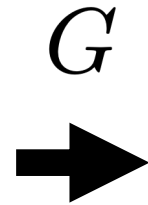
Variational Autoencoder (VAE)

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

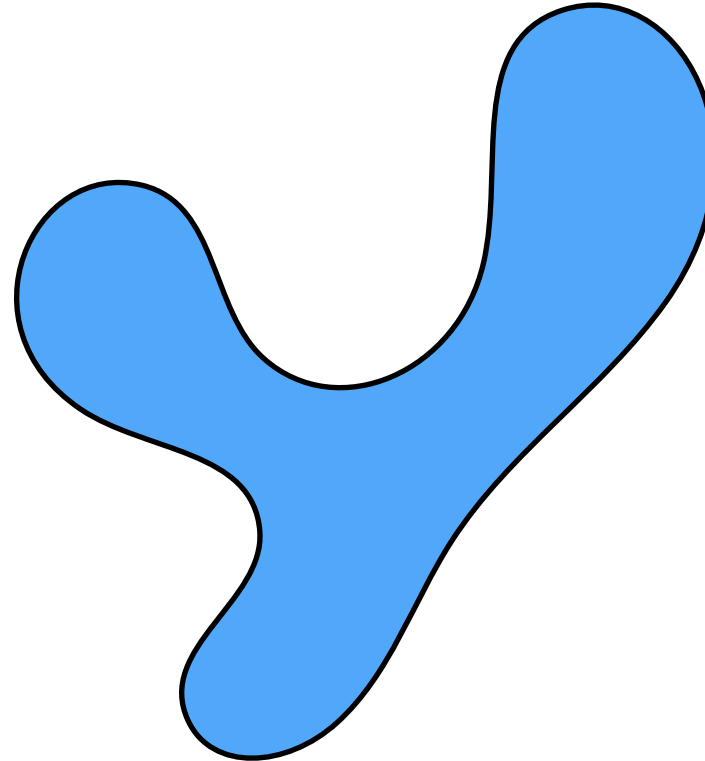
Prior distribution



$p(z)$

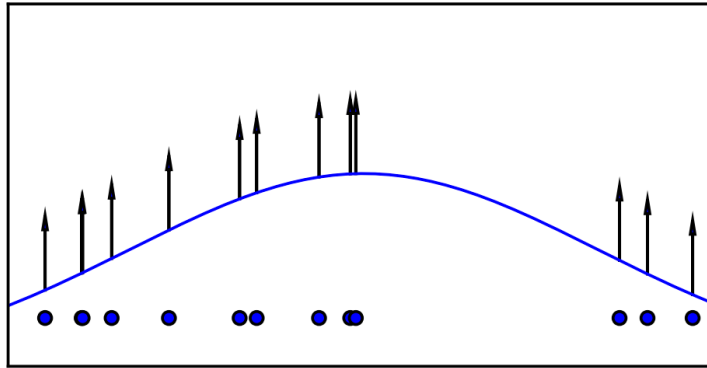


Target distribution



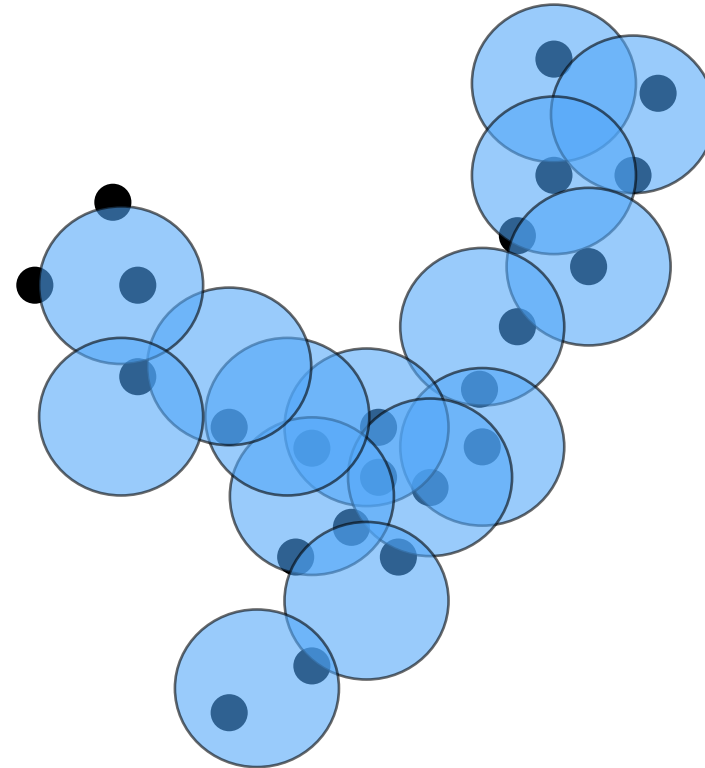
$p(x)$

Mixture of Gaussians



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

Target distribution



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

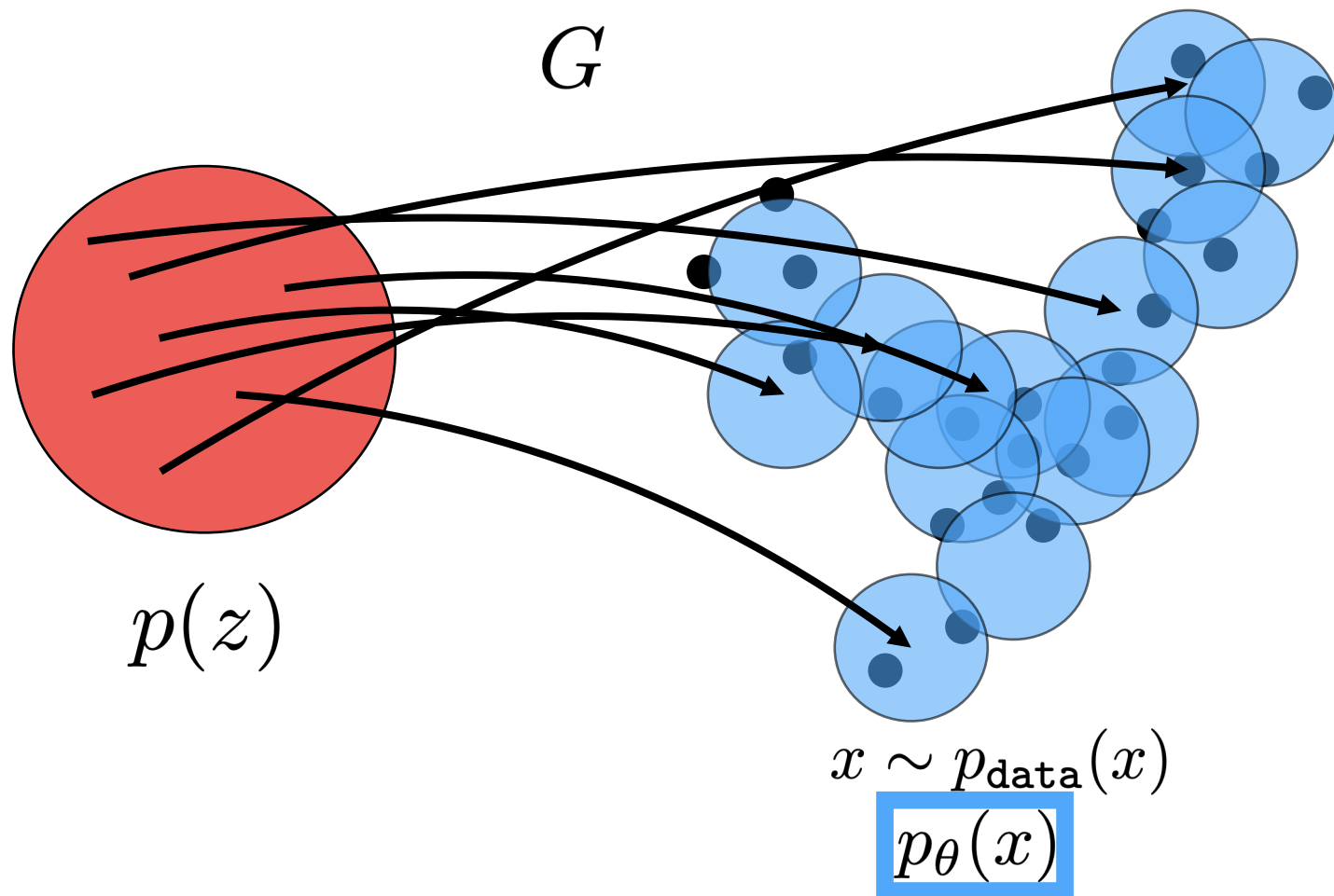
$$p_{\theta}(x)$$

Variational Autoencoder (VAE)

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

Prior distribution

Target distribution



Density model:

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

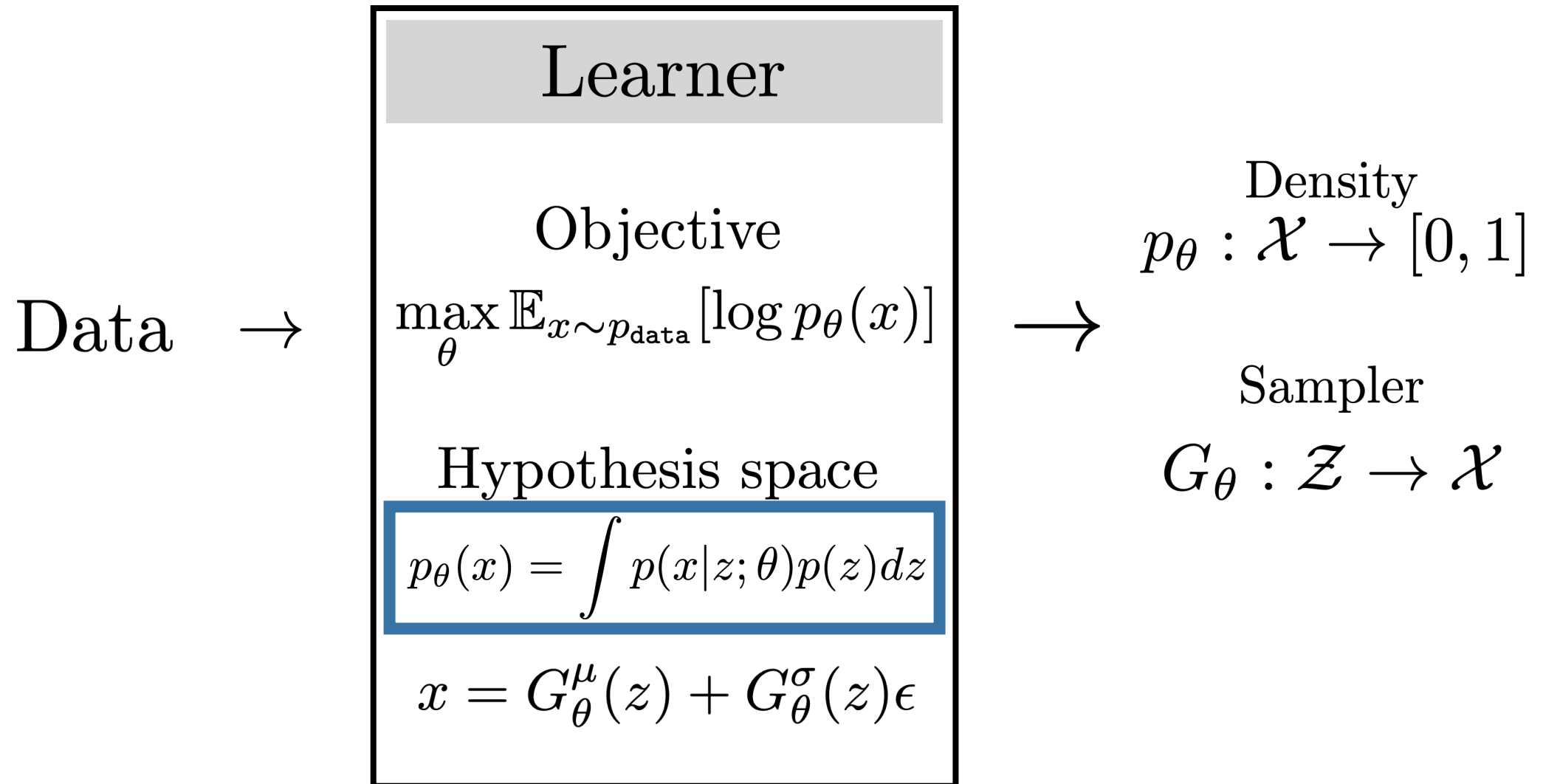
$$p(x|z; \theta) \sim \mathcal{N}(x; G_\theta^\mu(x), G_\theta^\sigma(x))$$

Sampling:

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

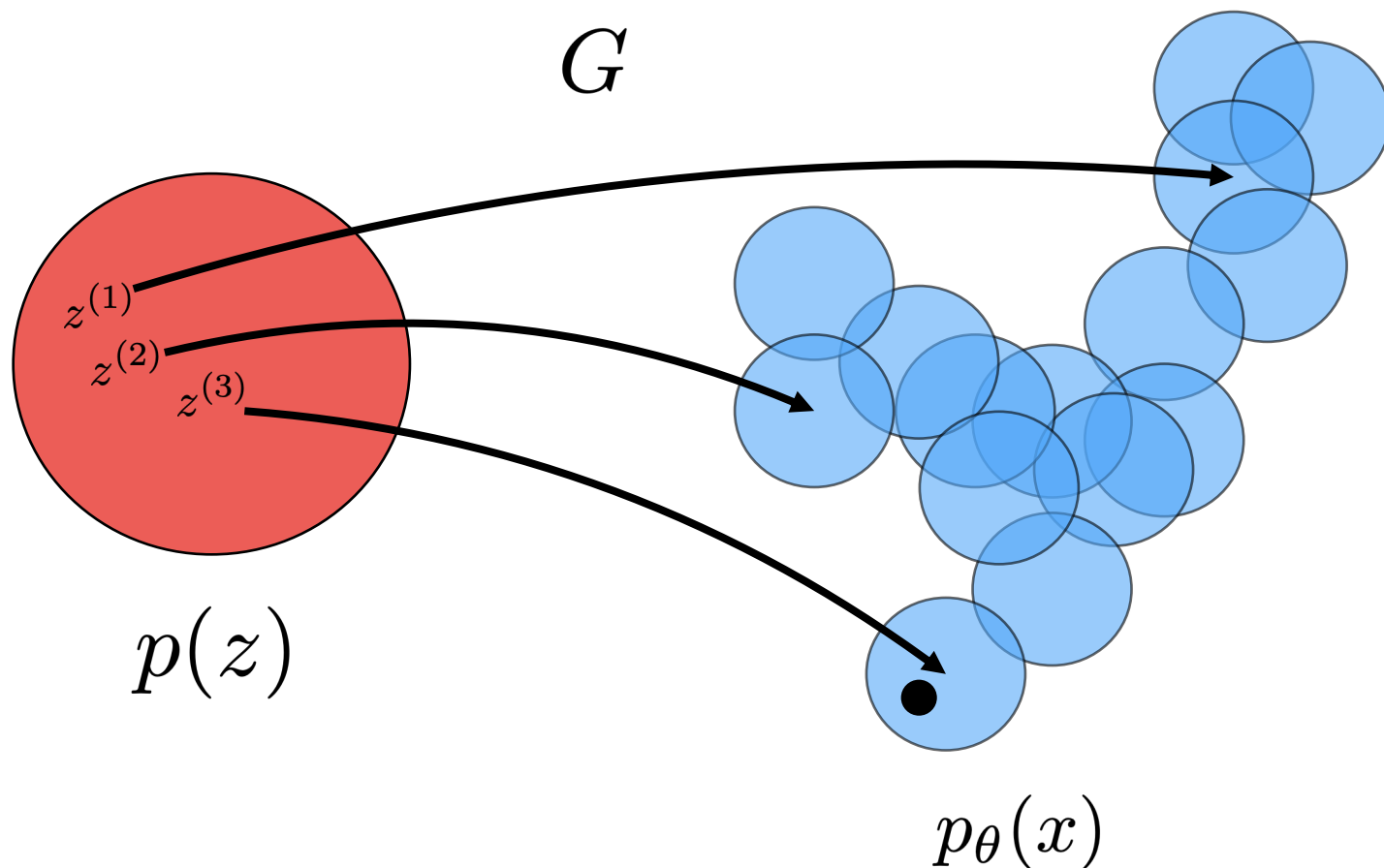
$$x = G_\theta^\mu(z) + G_\theta^\sigma(z)\epsilon$$

Variational Autoencoder (VAE)



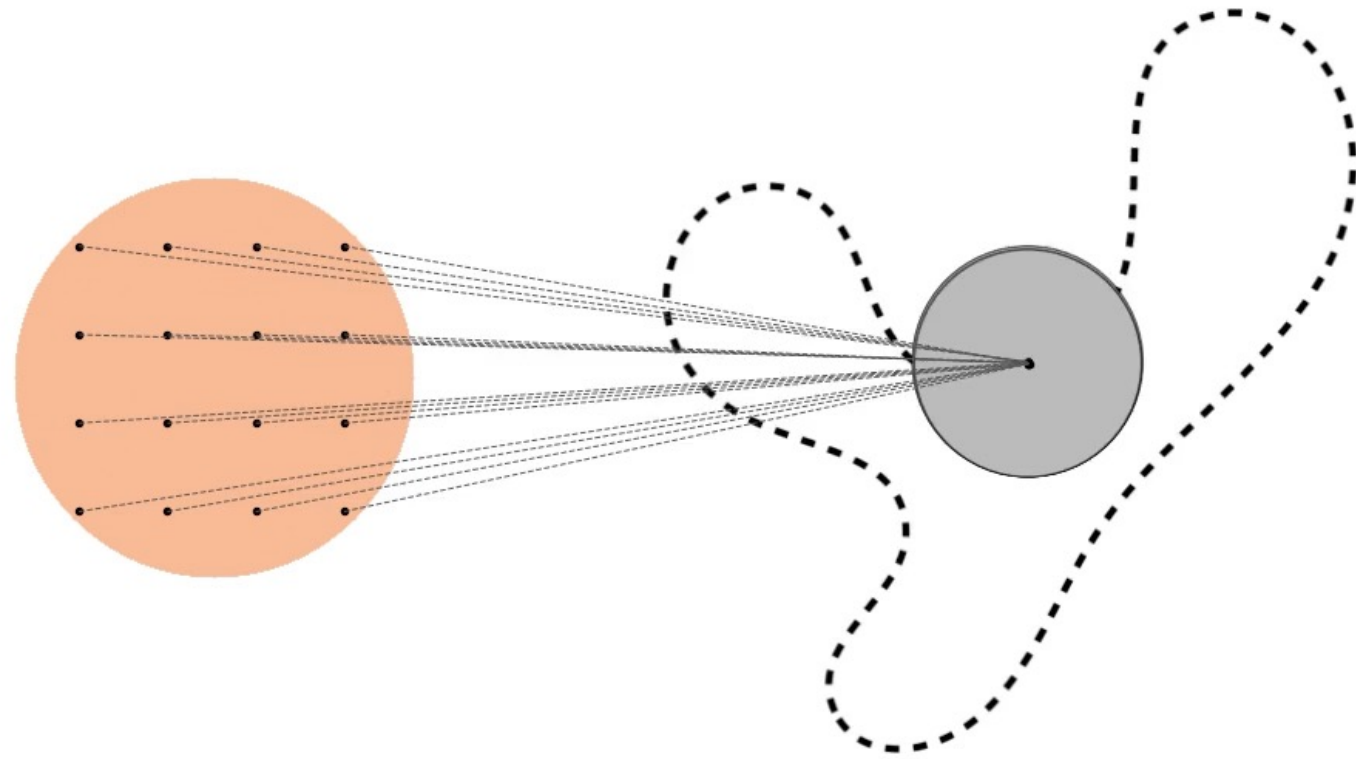
Prior distribution

Current model of target distribution



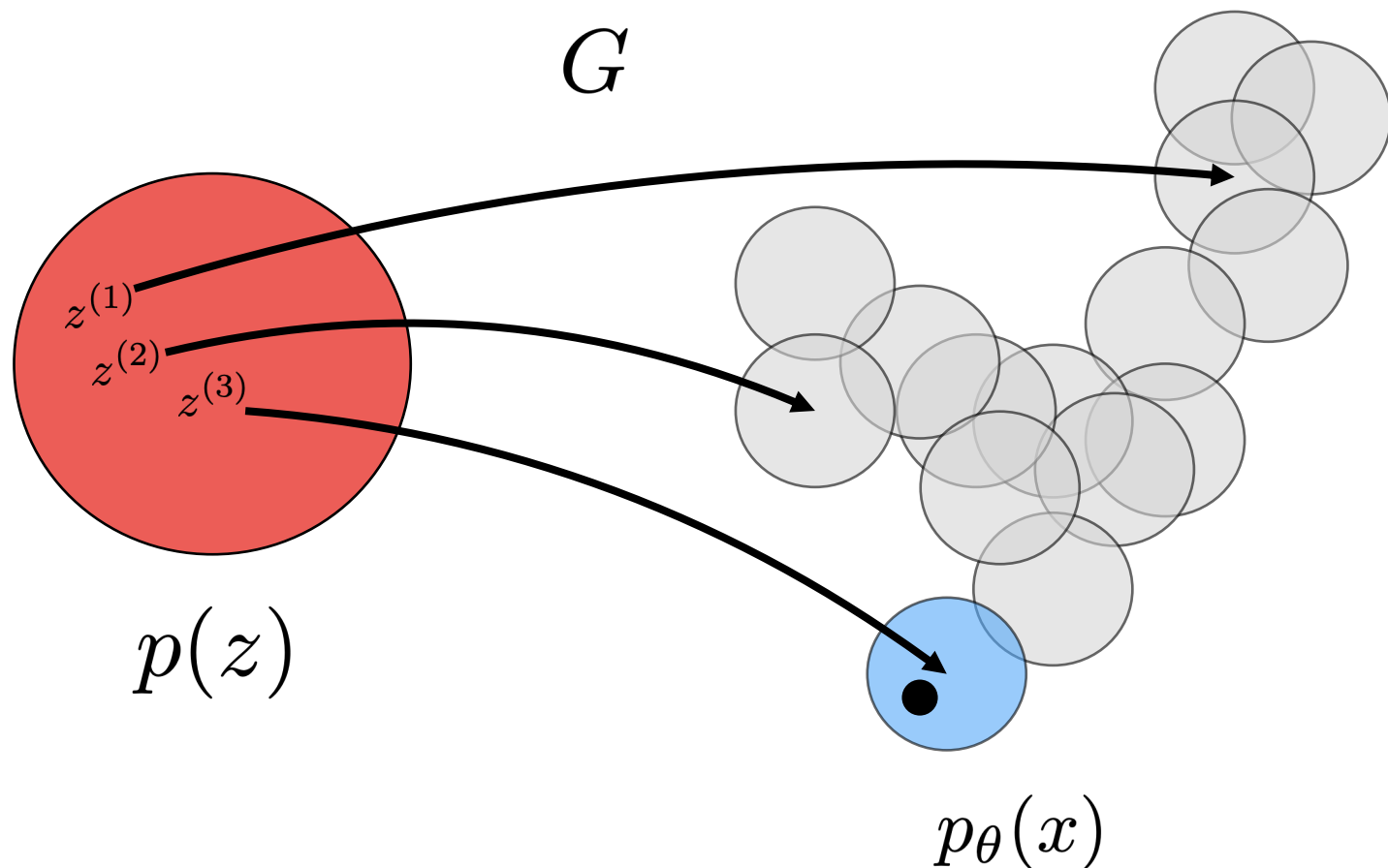
In order to optimize our model, we need to measure the likelihood it assigns to each datapoint x

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



Prior distribution

Current model of target distribution

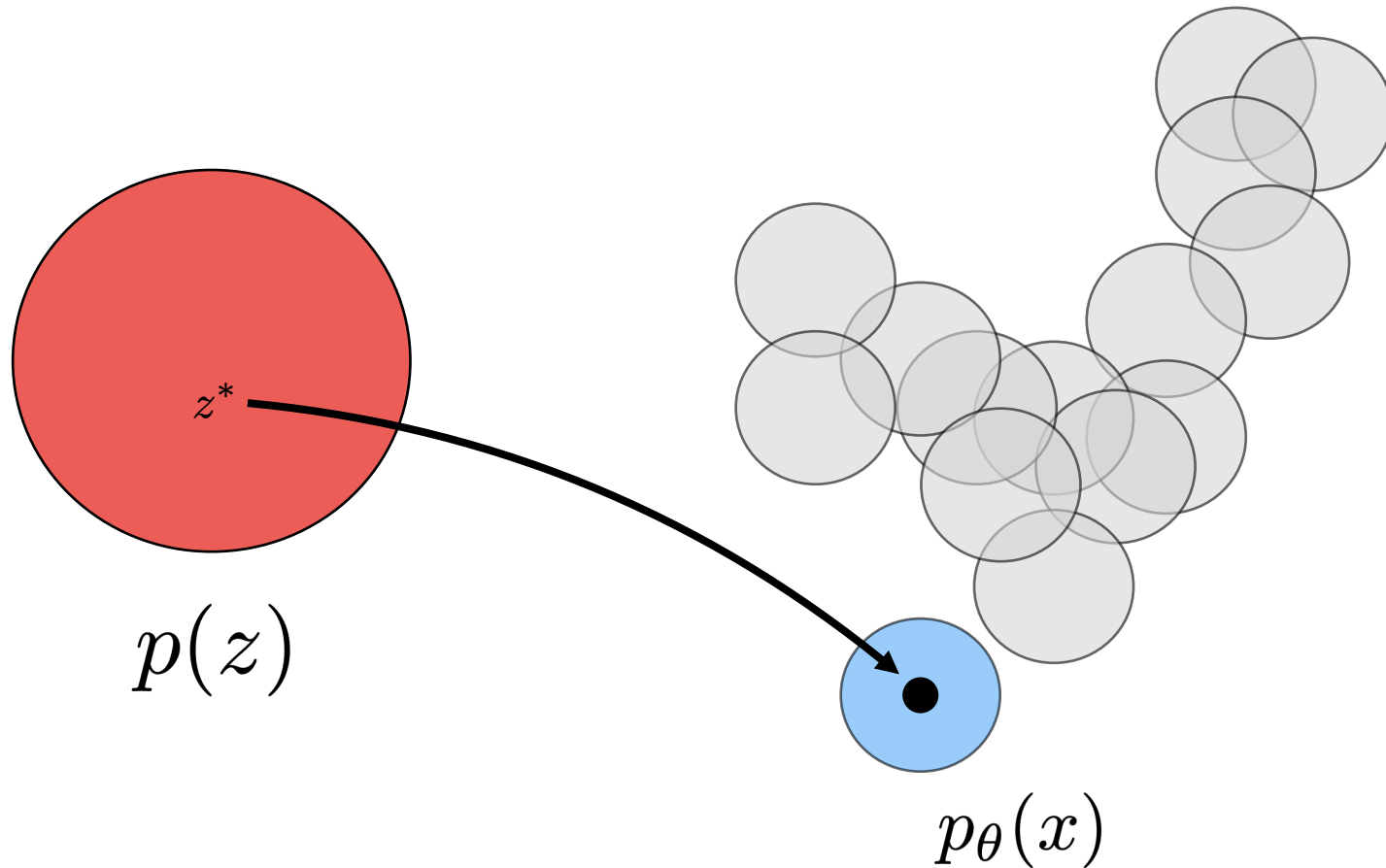


In order to optimize our model, we need to measure the likelihood it assigns to each datapoint x

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

Prior distribution

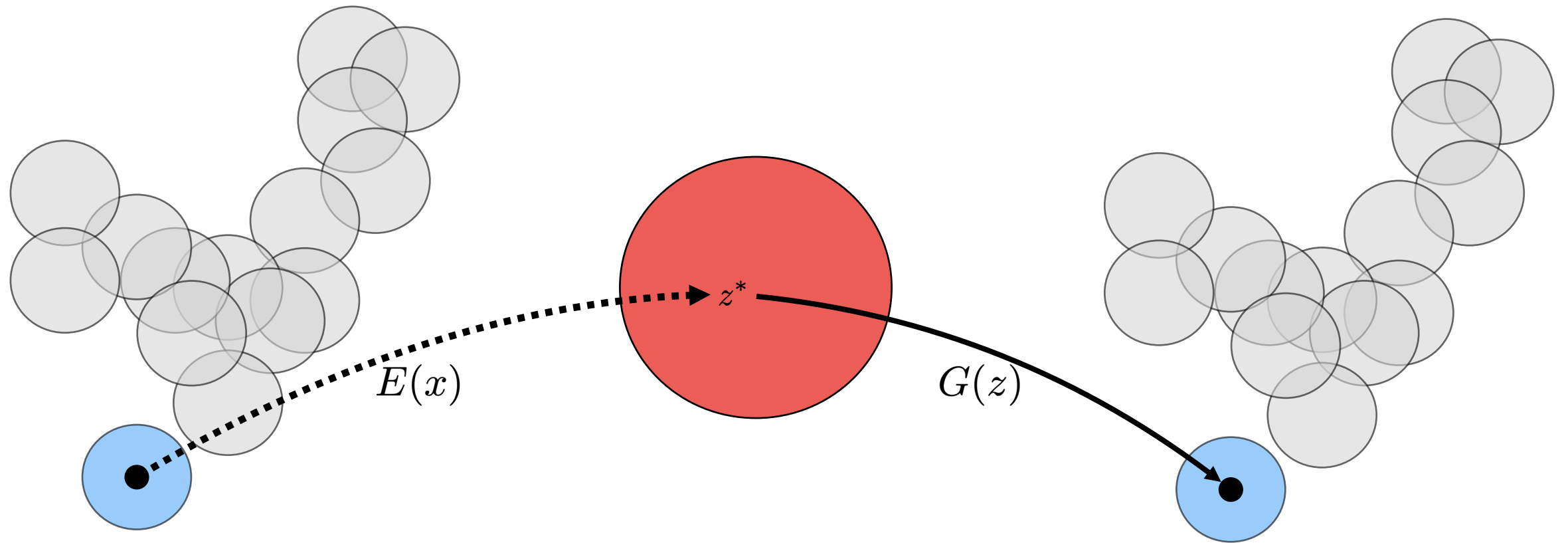
Current model of
target distribution

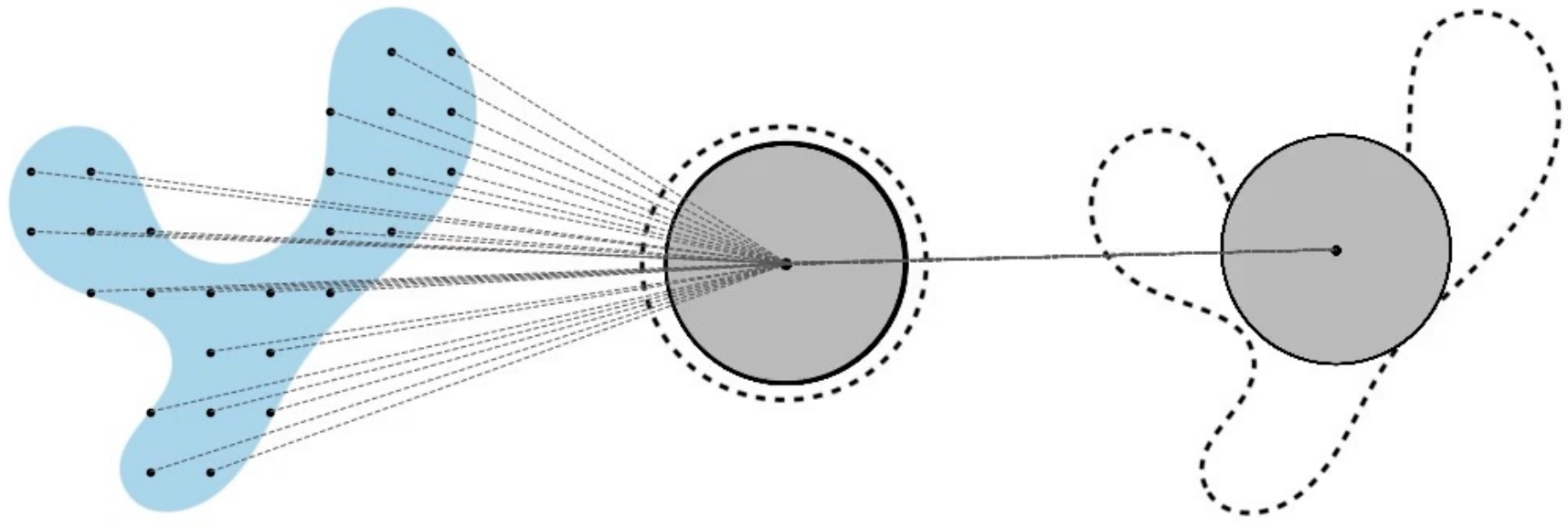


If only we knew z^* , we
wouldn't need the integral...

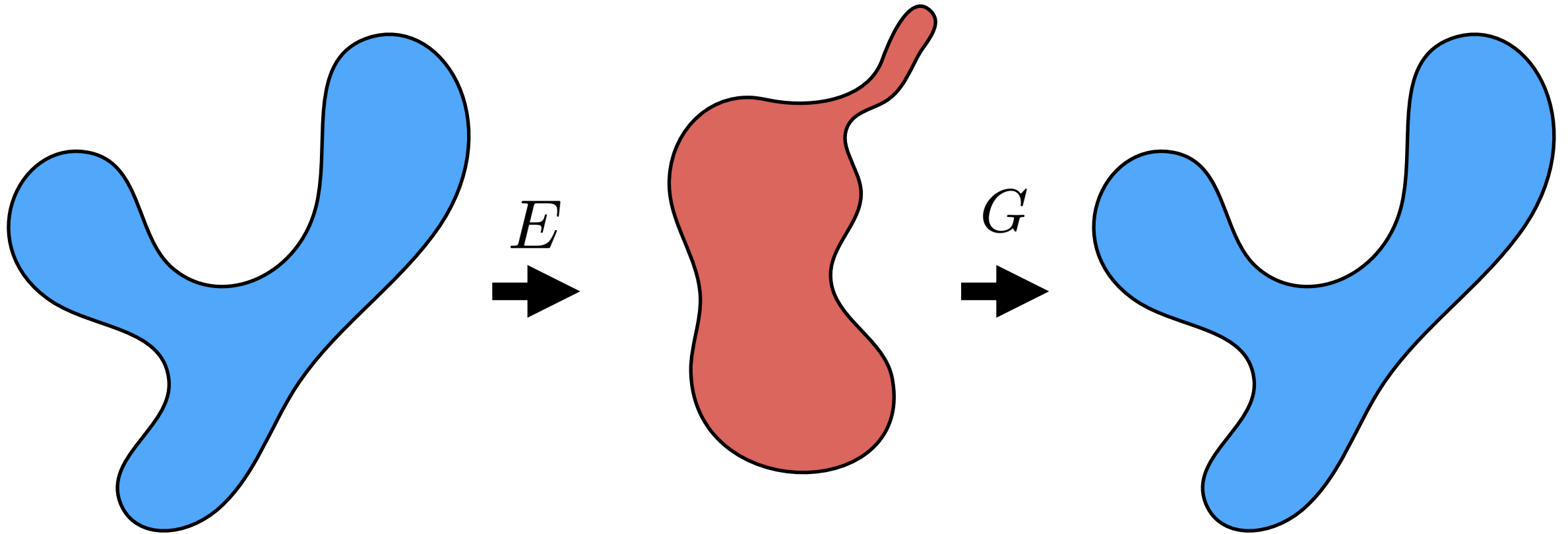
$$p_\theta(x) = \int p(x|z; \theta)p(z)dz$$
$$\approx p(x|z^*; \theta)p(z^*)$$

Autoencoder!



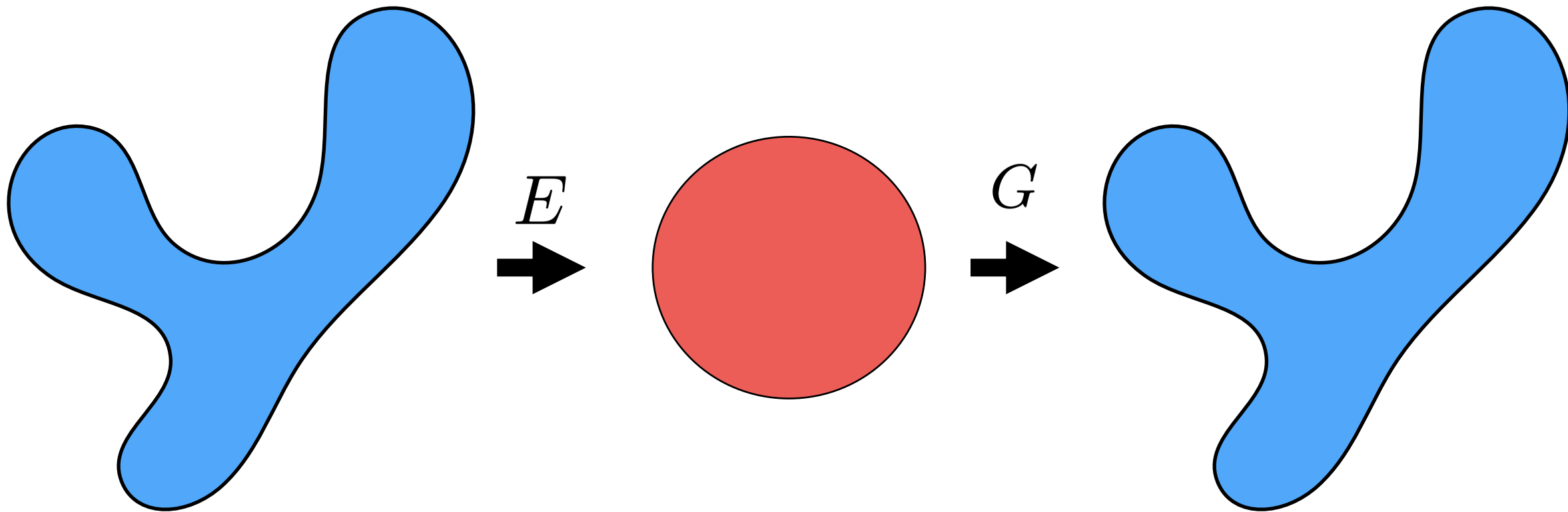


Classical Autoencoder



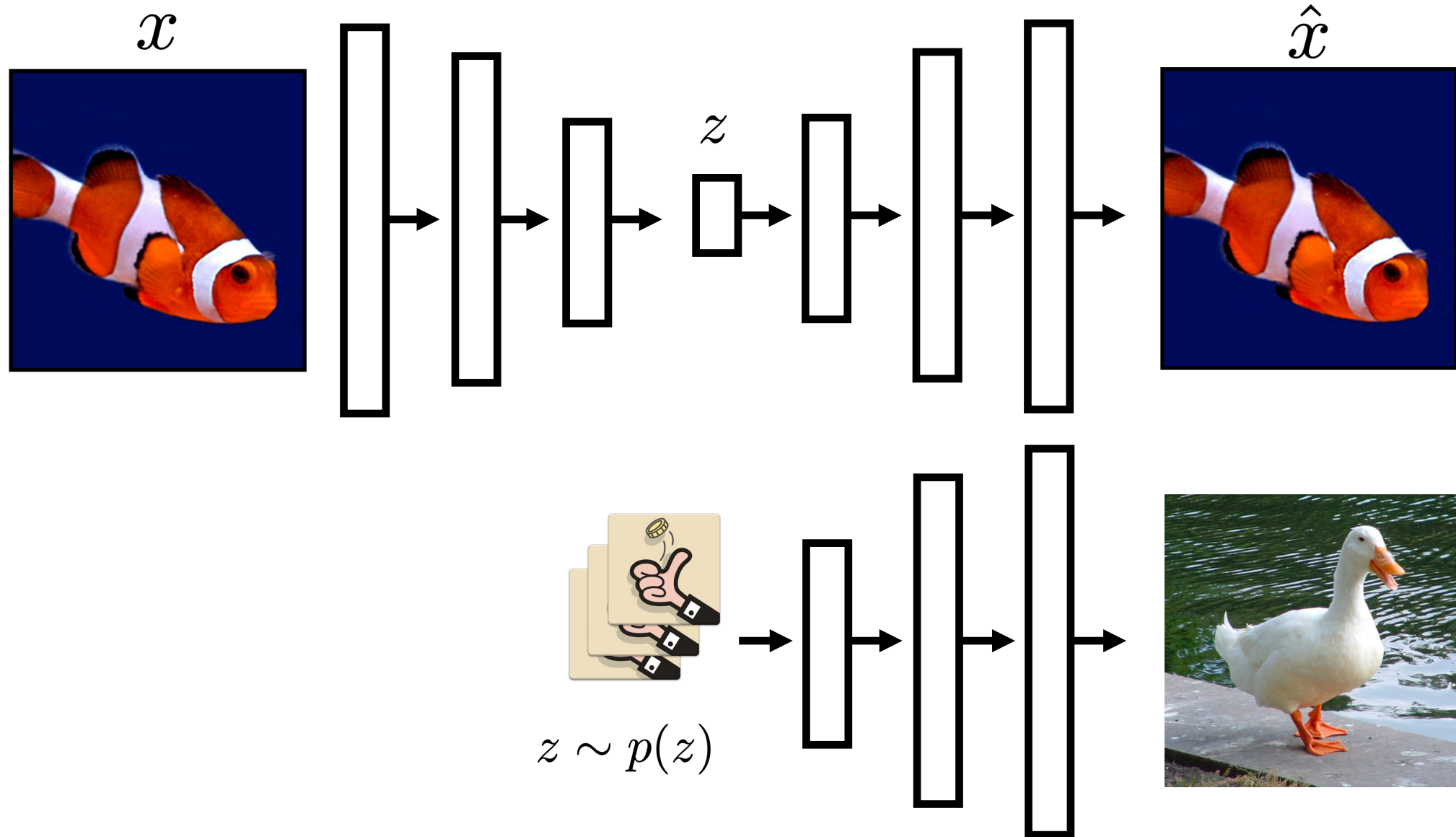
$$\arg \min_{G, E} \mathbb{E}_x [\|G(E(x)) - x\|_2^2]$$

Variational Autoencoder



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

Variational Autoencoder



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

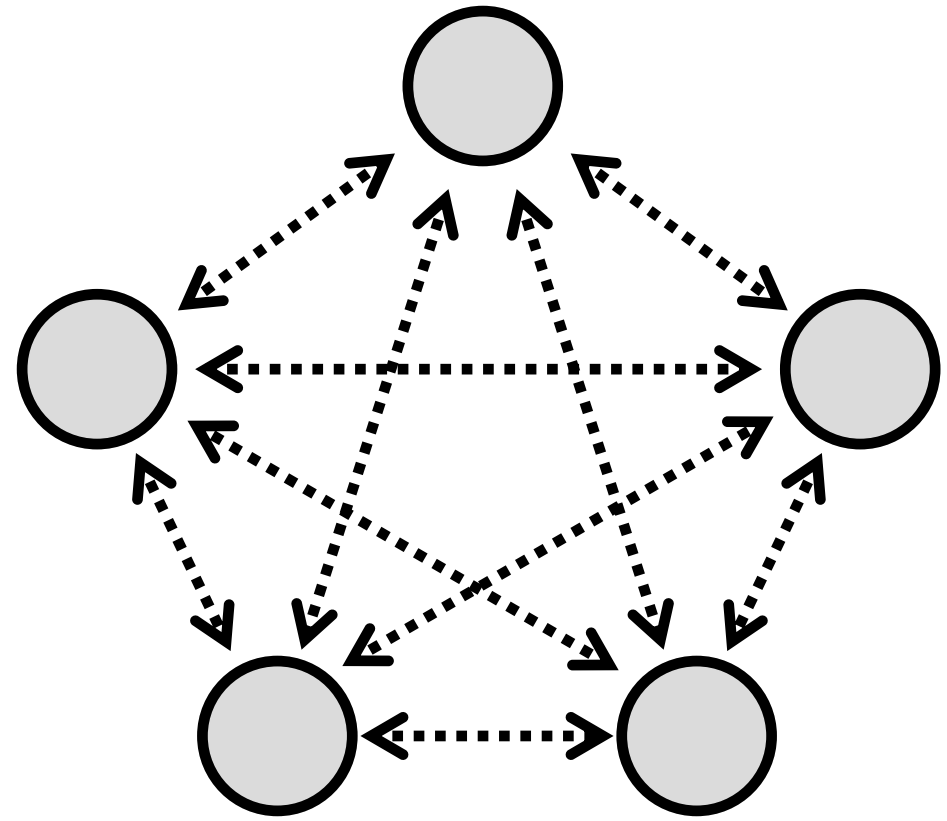
Vector Quantized VAEs (VQ-VAE) 2019



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

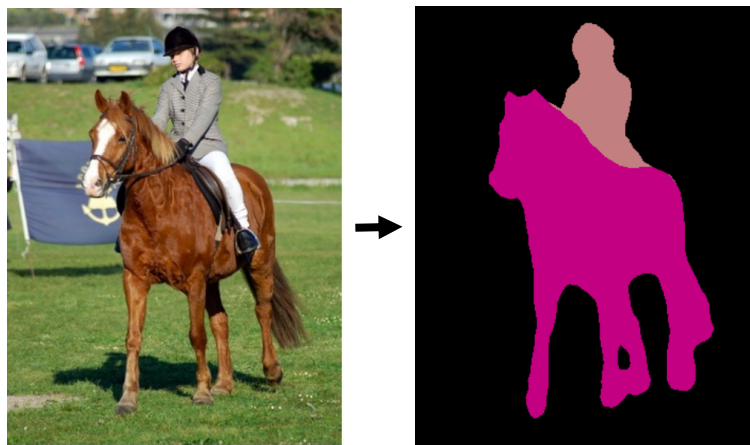
VQ-VAE-2, Razavi et al., NeurIPS 2019

Data Translation



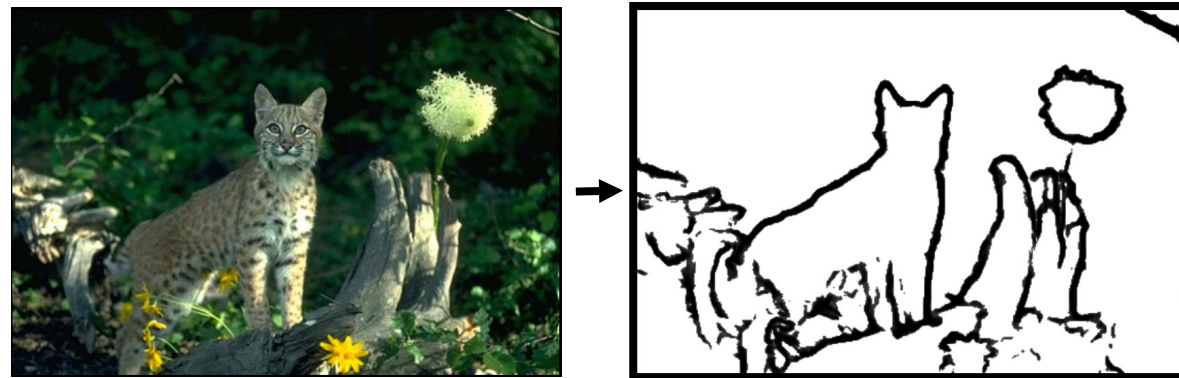
Data translation problems (“structured prediction”)

Semantic segmentation



[Long et al. 2015, ...]

Edge detection



[Xie et al. 2015, ...]

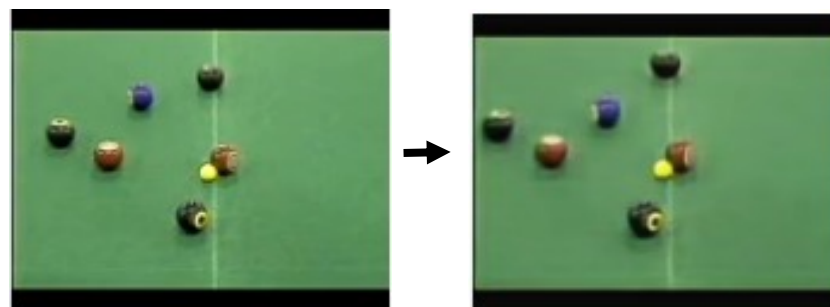
Text-to-image

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



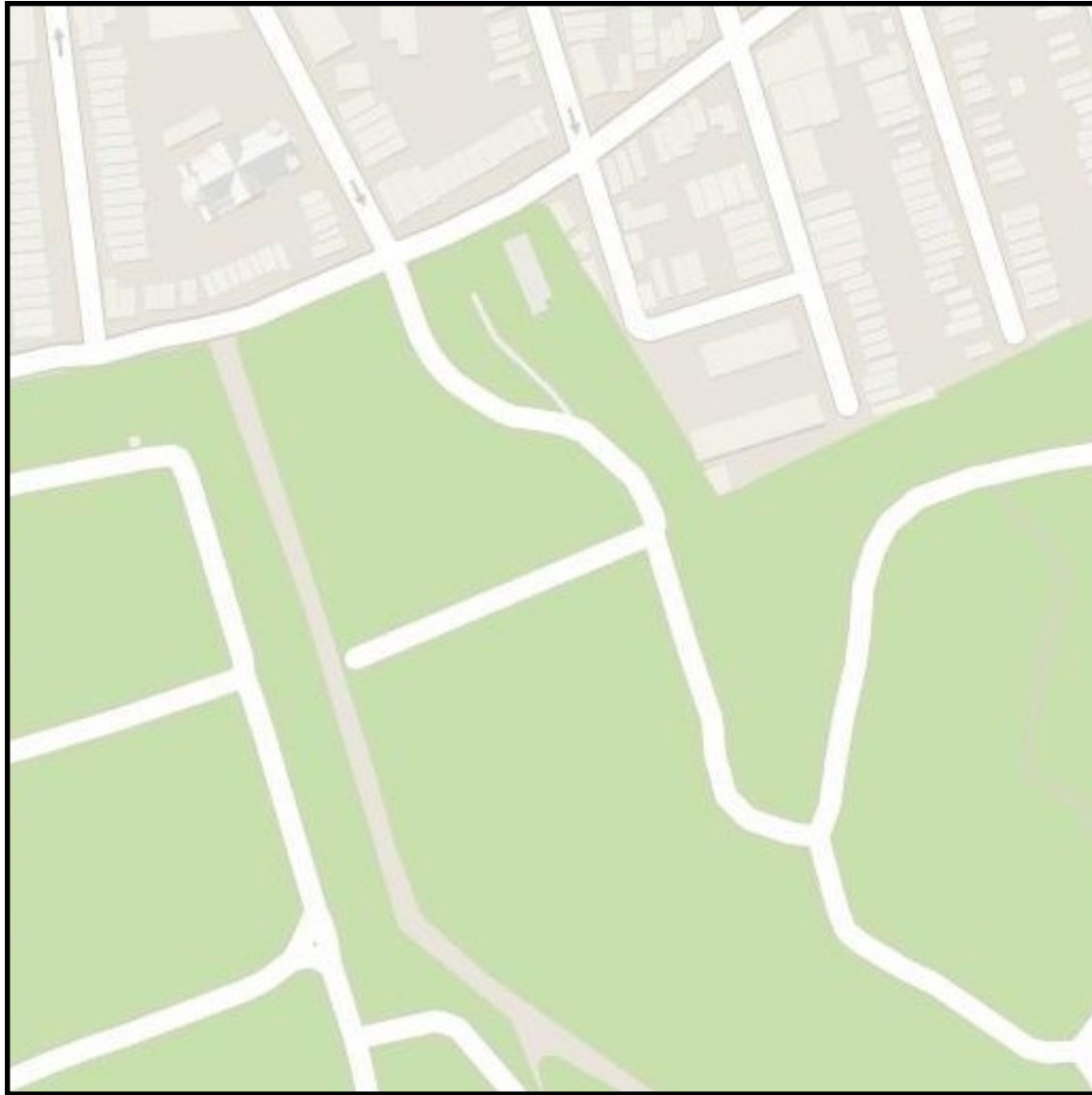
[Reed et al. 2014, ...]

Future frame prediction



[Mathieu et al. 2016, ...]

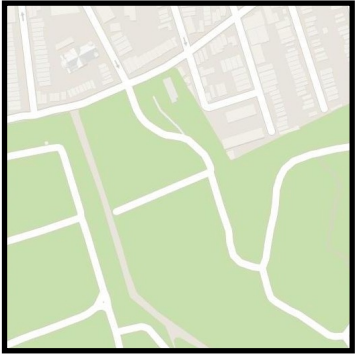
Input



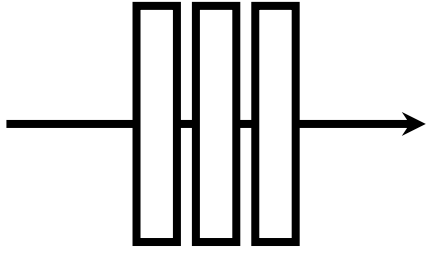
Deep net output



\mathbf{x}



G



Generator

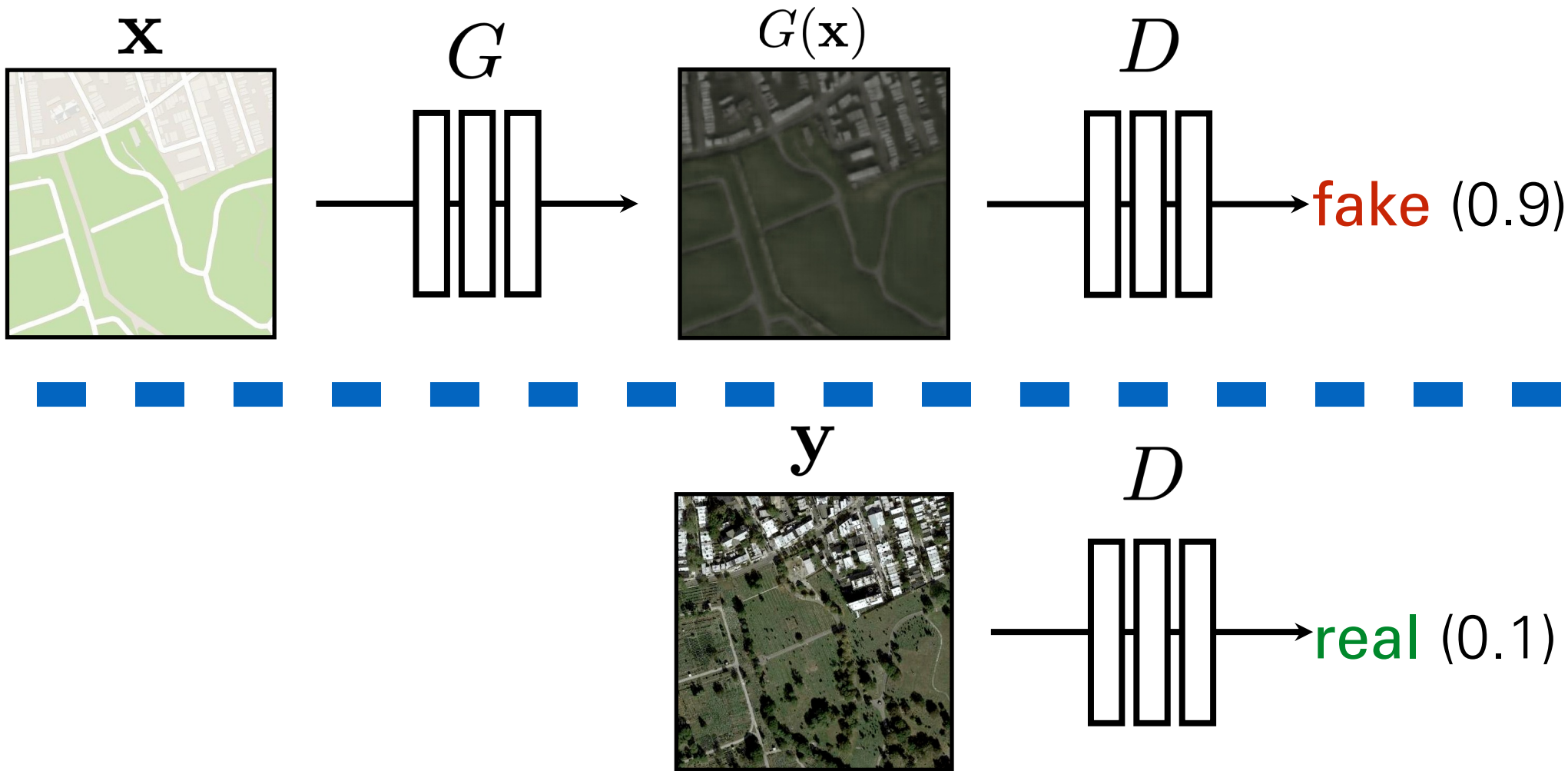
$G(\mathbf{x})$



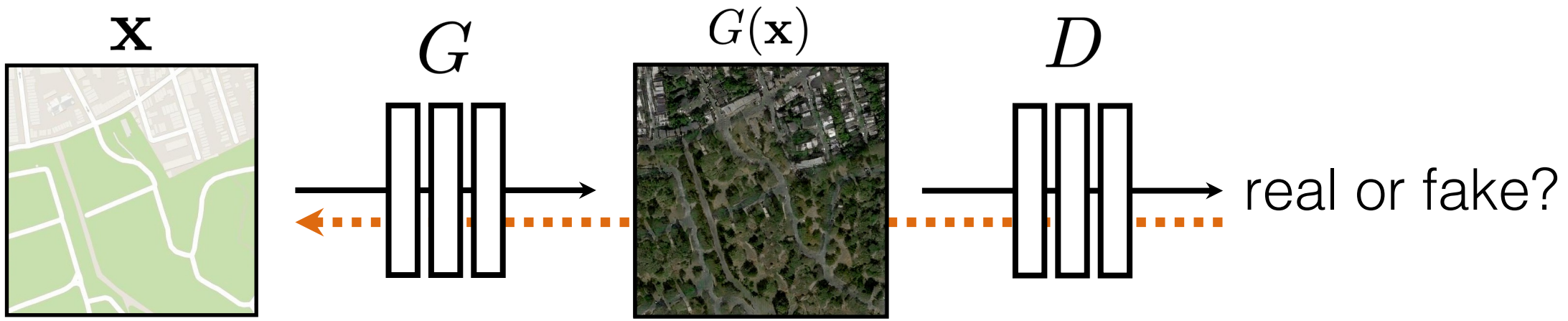


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

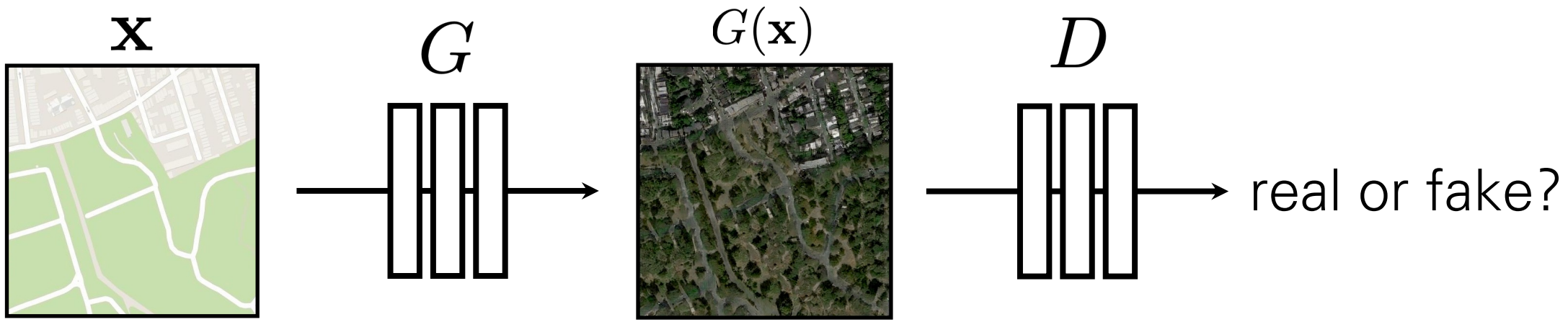


$$\arg \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$



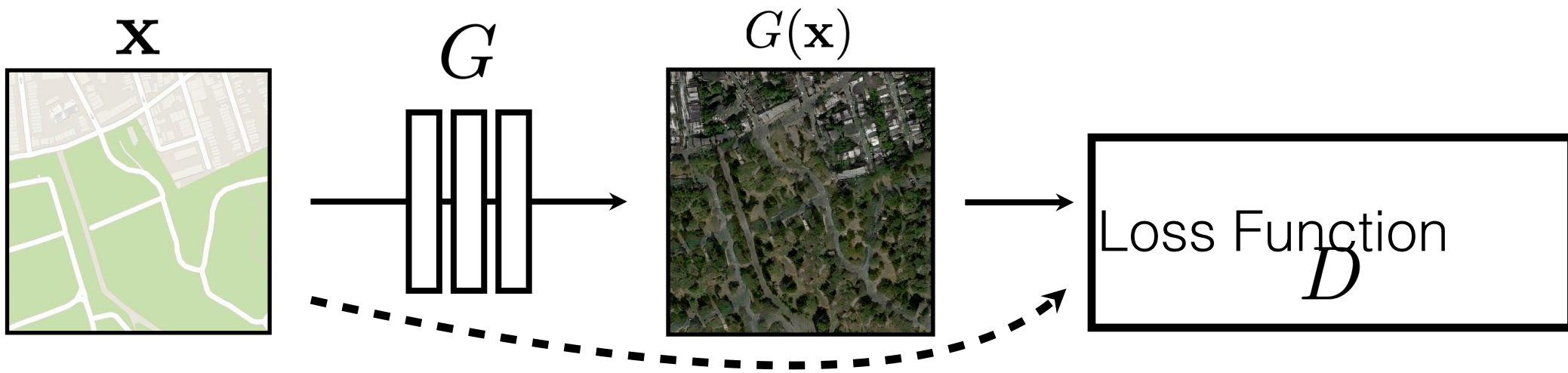
G tries to synthesize fake images that *fool* **D**:

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



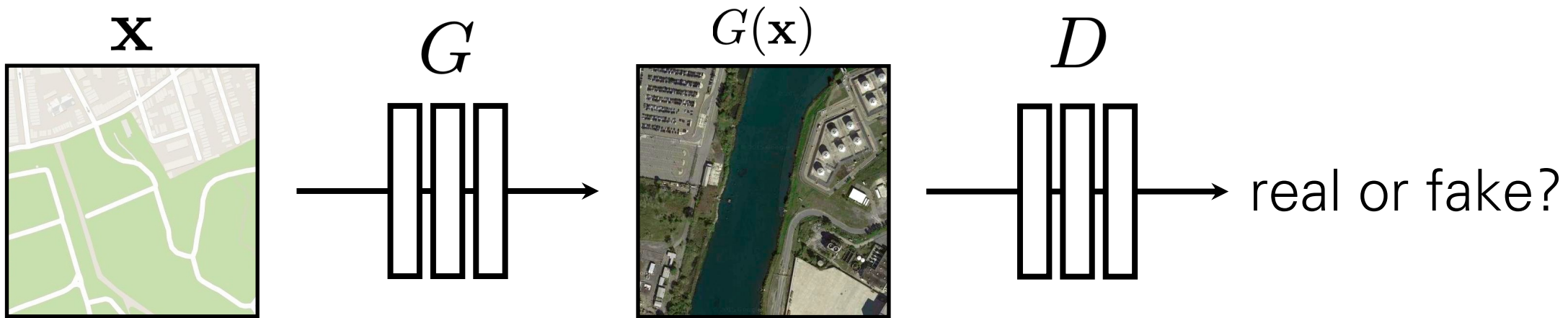
G tries to synthesize fake images that *fool* the *best* D :

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

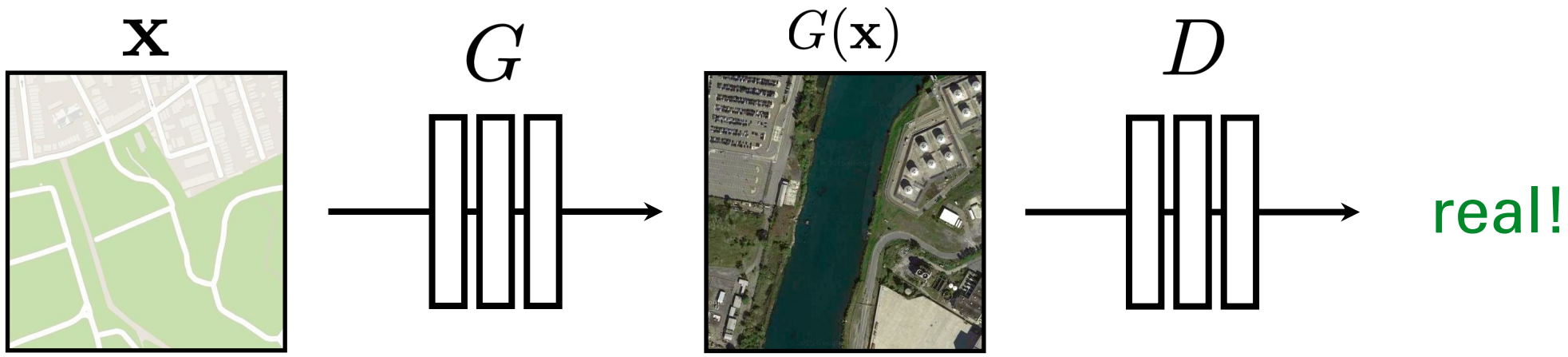


G's perspective: **D** is a loss function.

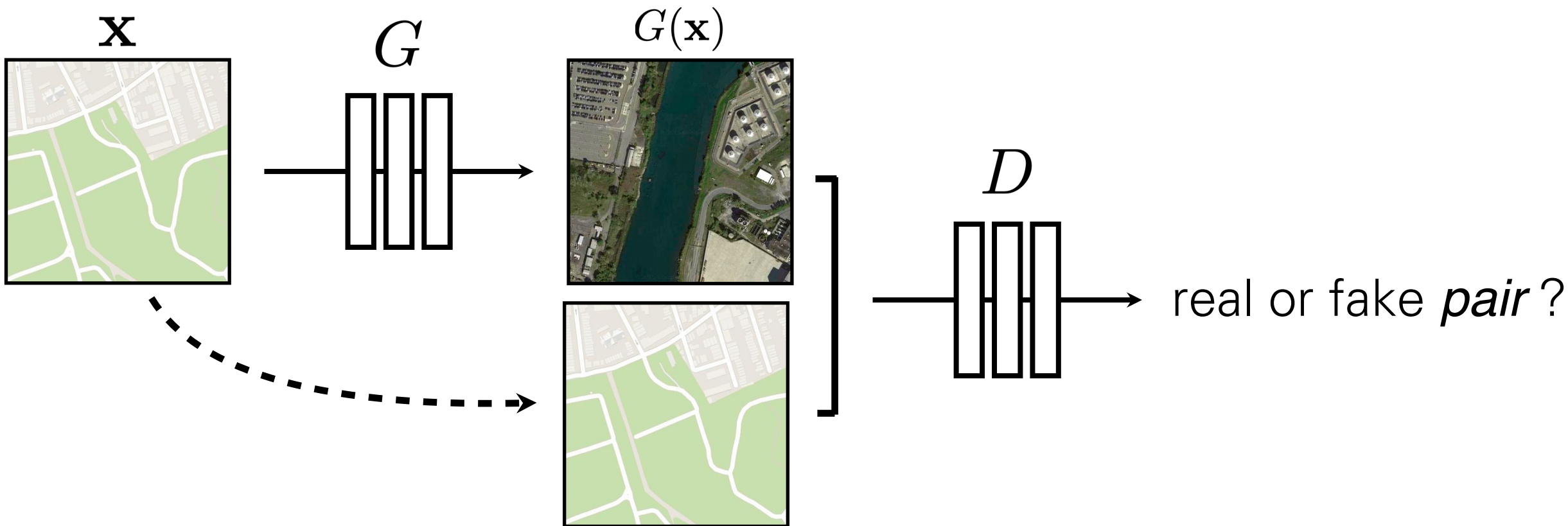
Rather than being hand-designed, it is *learned* and *highly structured*.



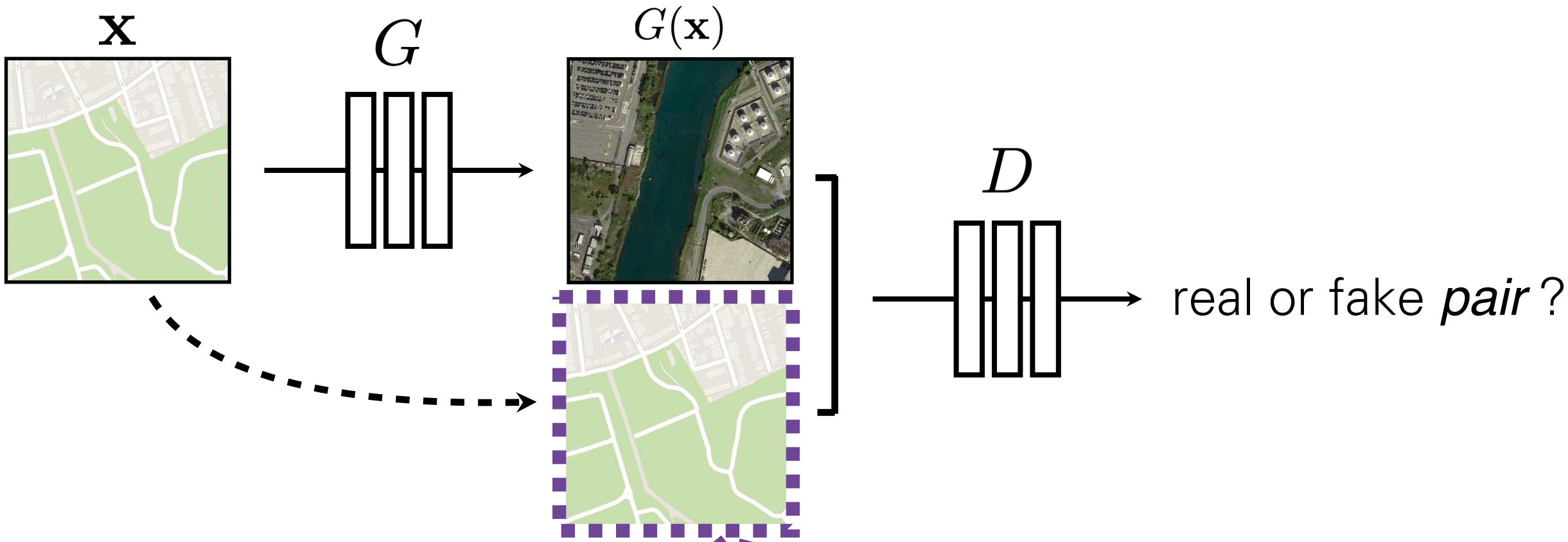
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$



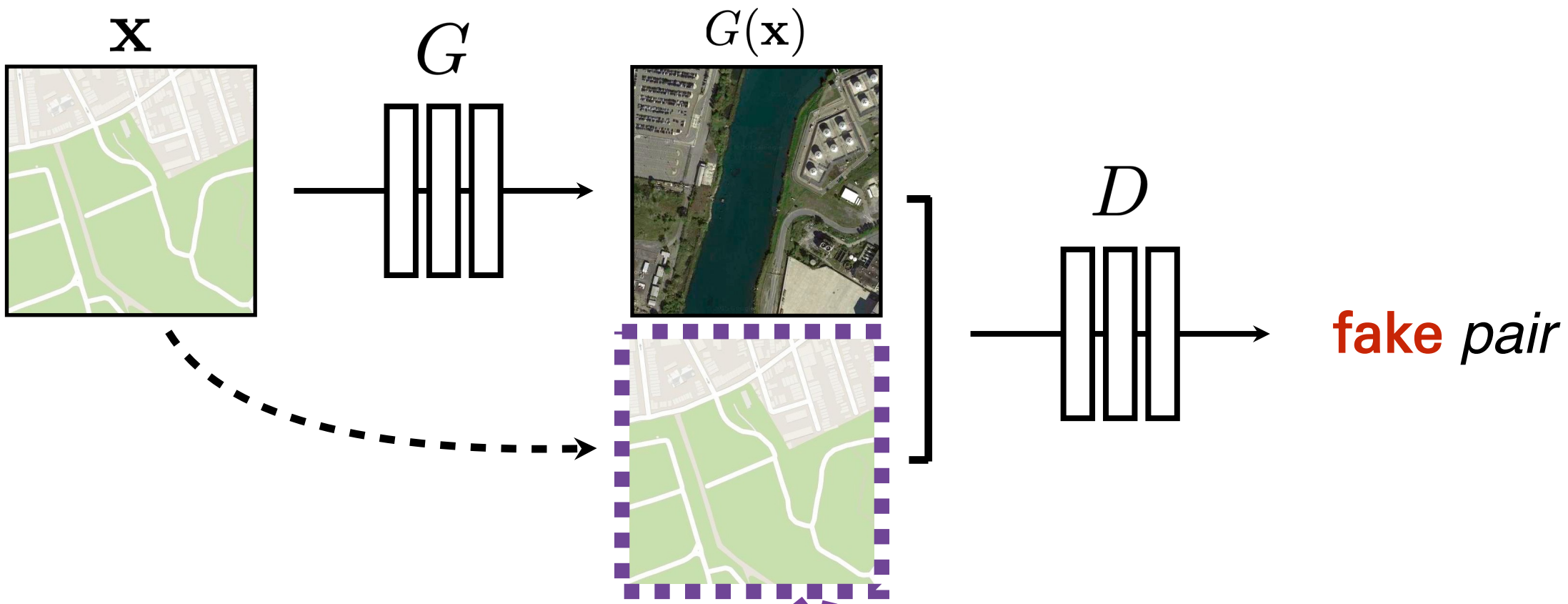
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$



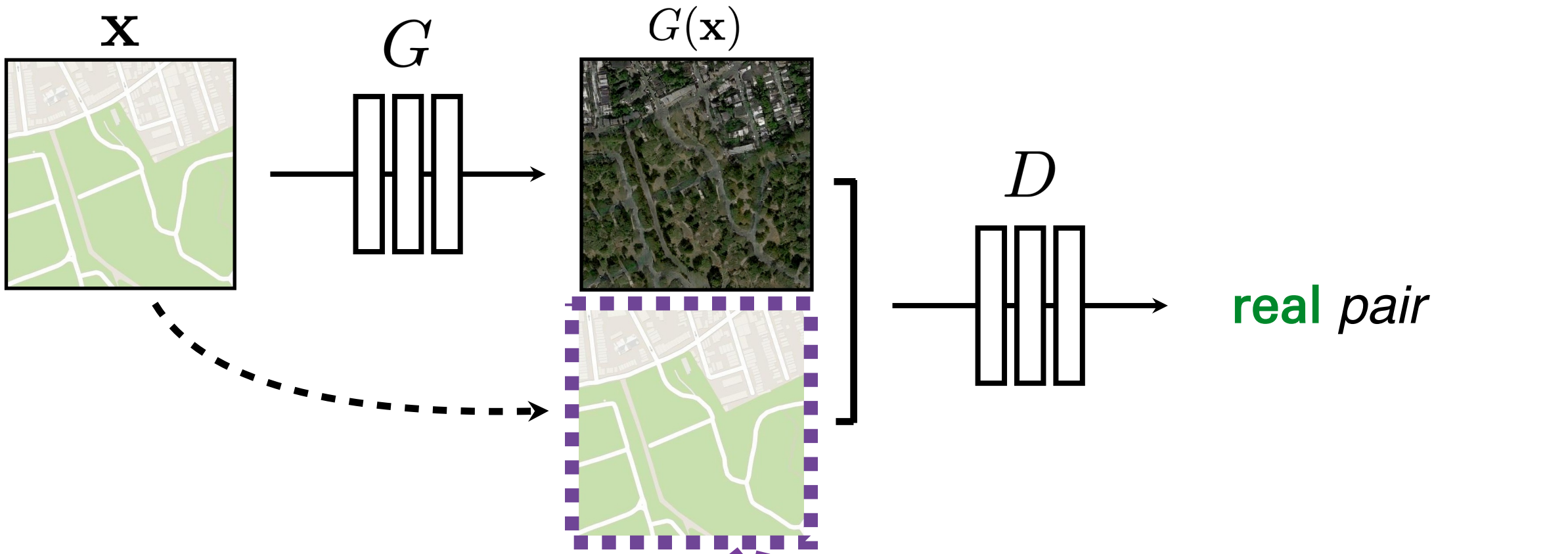
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$



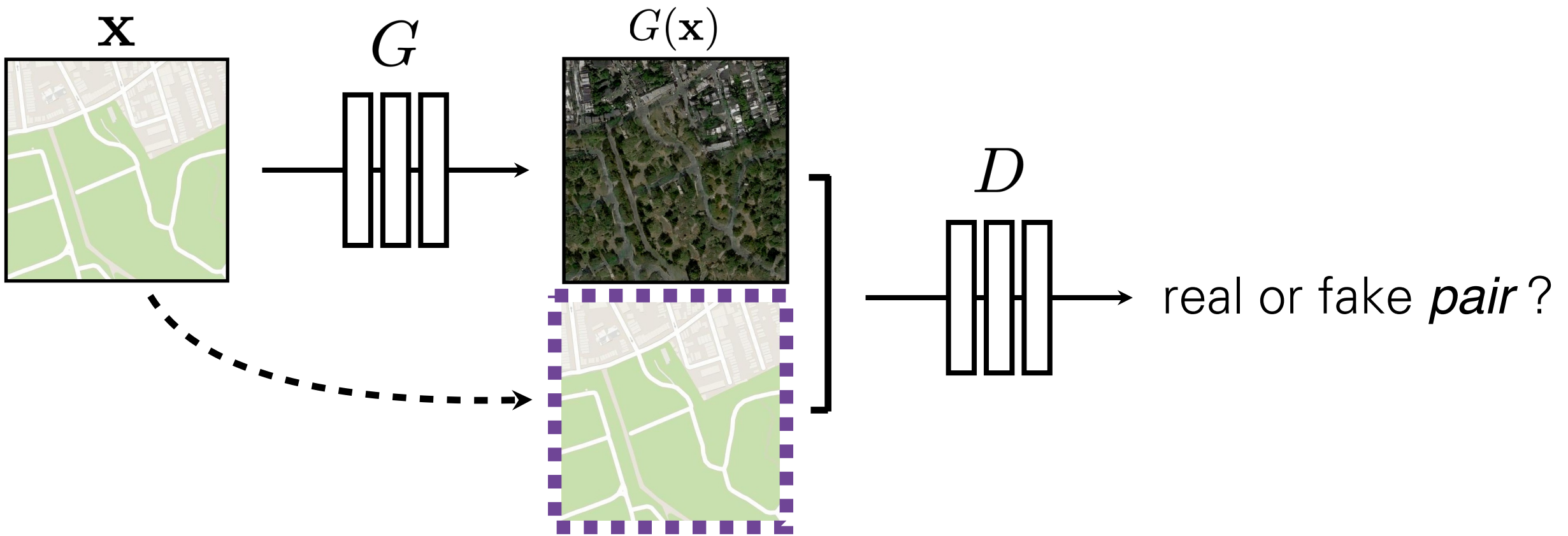
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))]$$



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



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



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

Training Details: Loss function

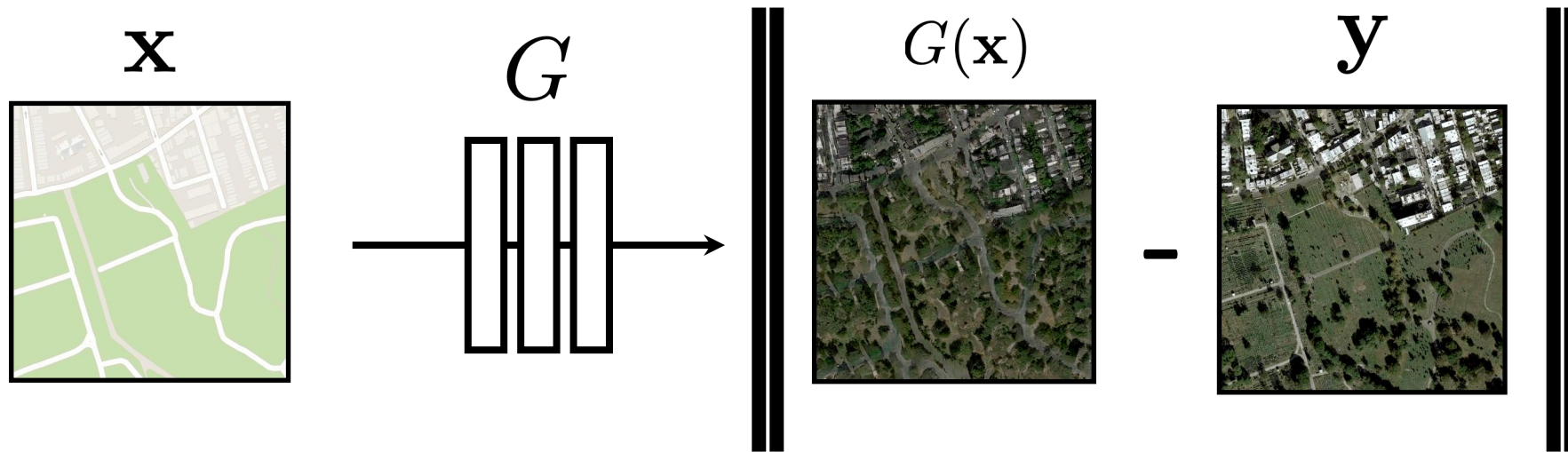
Conditional GAN

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

Training Details: Loss function

Conditional GAN

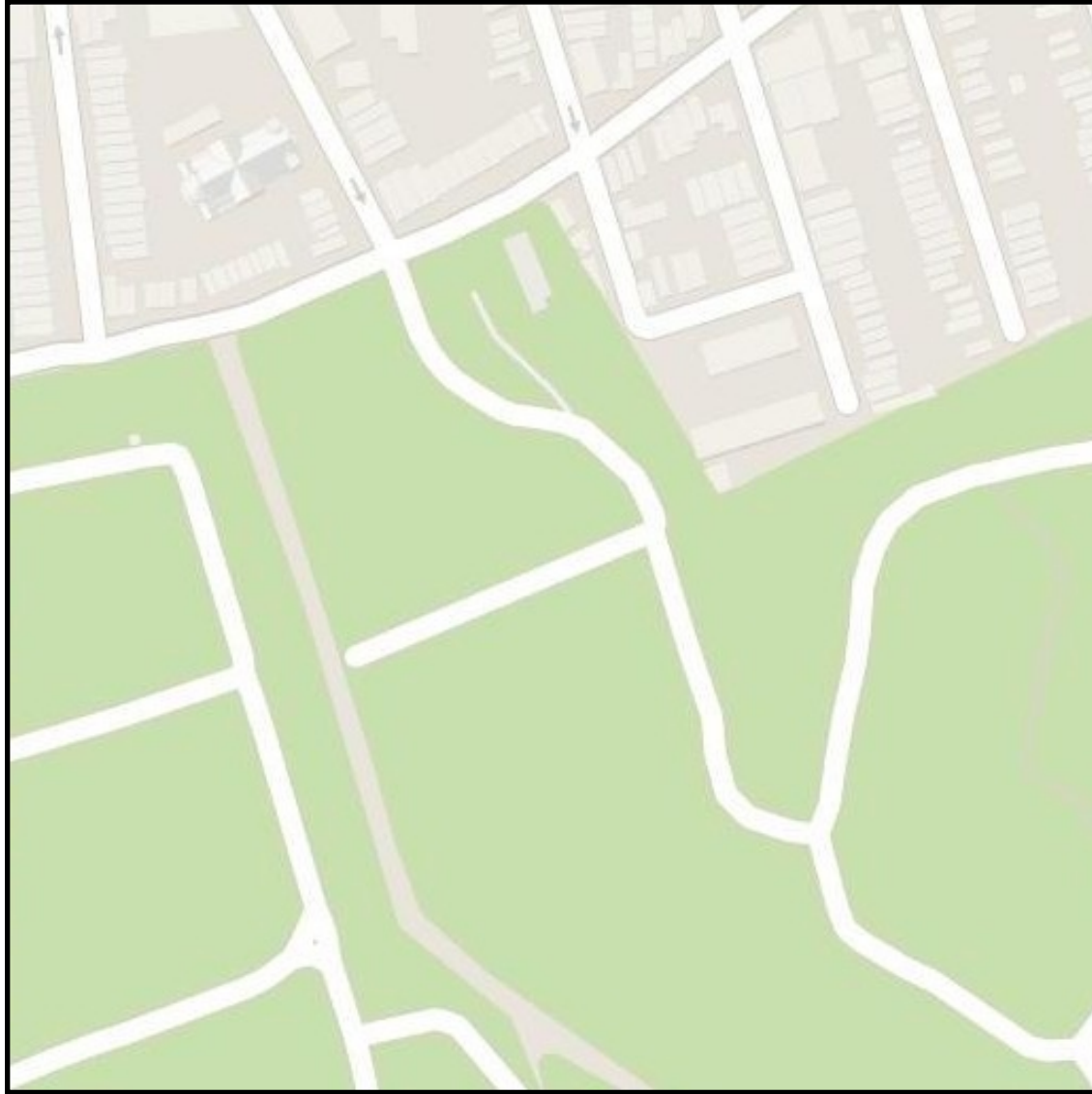
$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$



Stable training + fast convergence

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

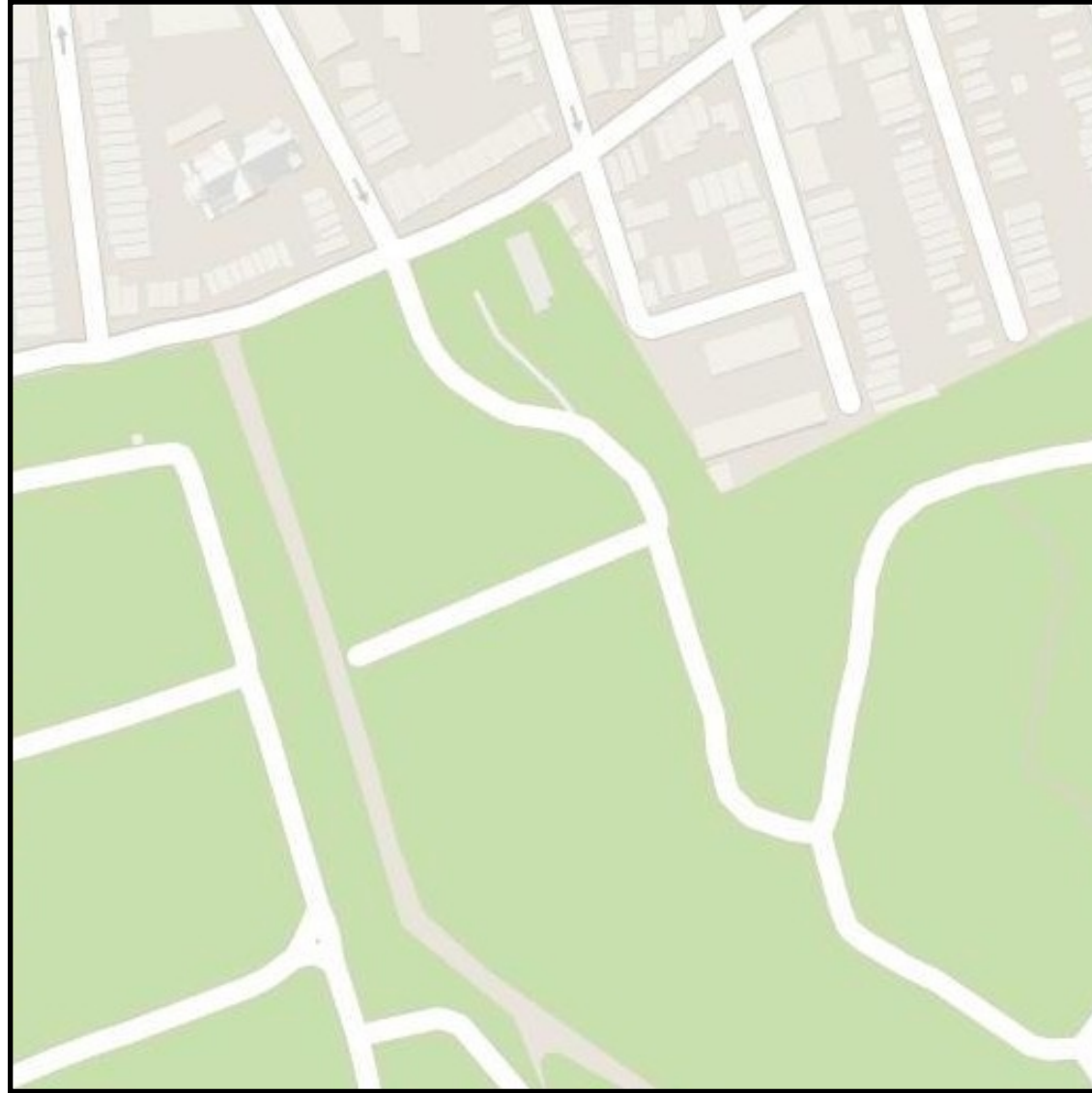
Input



Unstructured prediction (L1)



Input



Structured Prediction (cGAN)



Input



Output



Groundtruth



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



Input

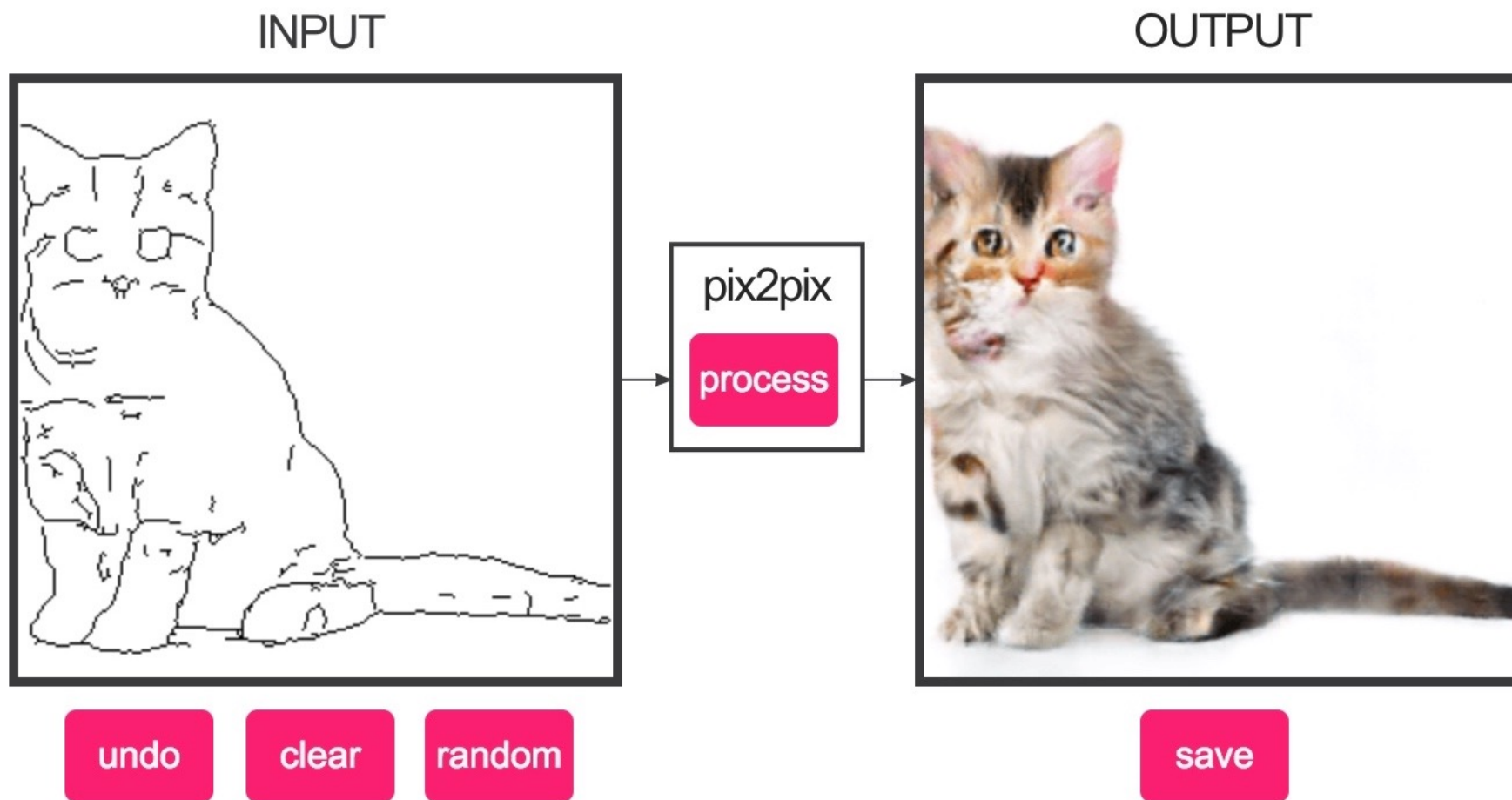
Output

Groundtruth



Data from [maps.google.com]

#edges2cats [Chris Hesse]

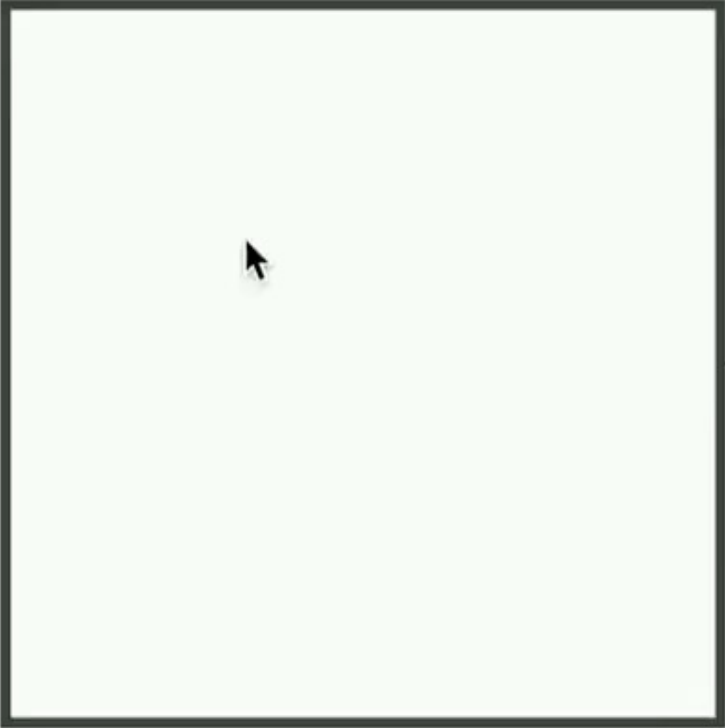


edges2cats

TOOL

line
 eraser

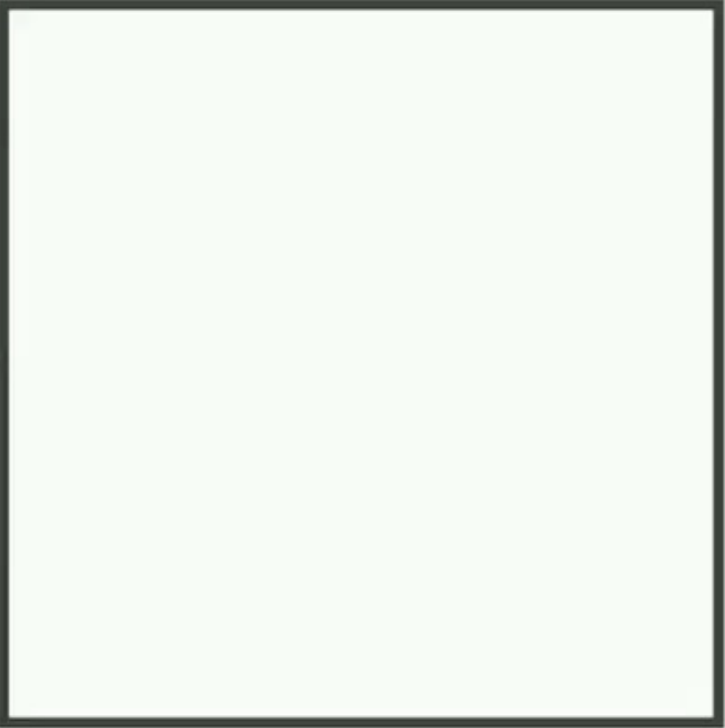
INPUT



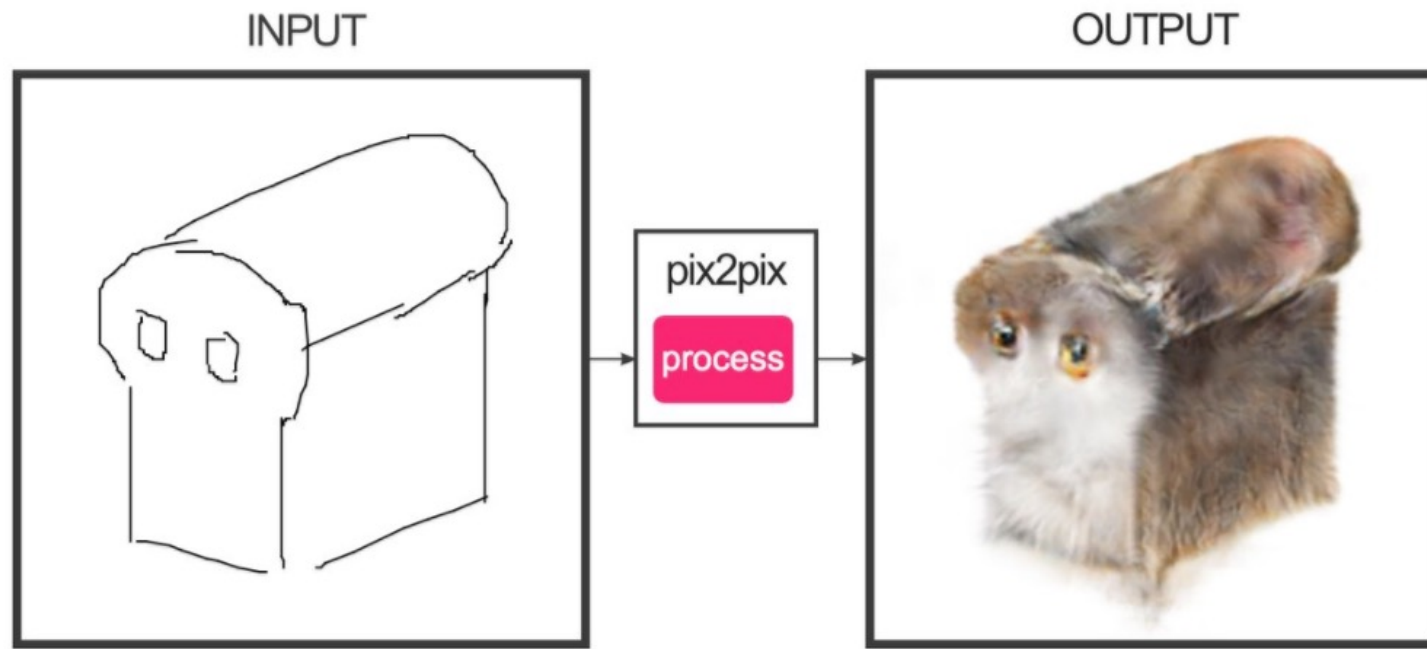
undo clear random

pix2pix
process

OUTPUT



save

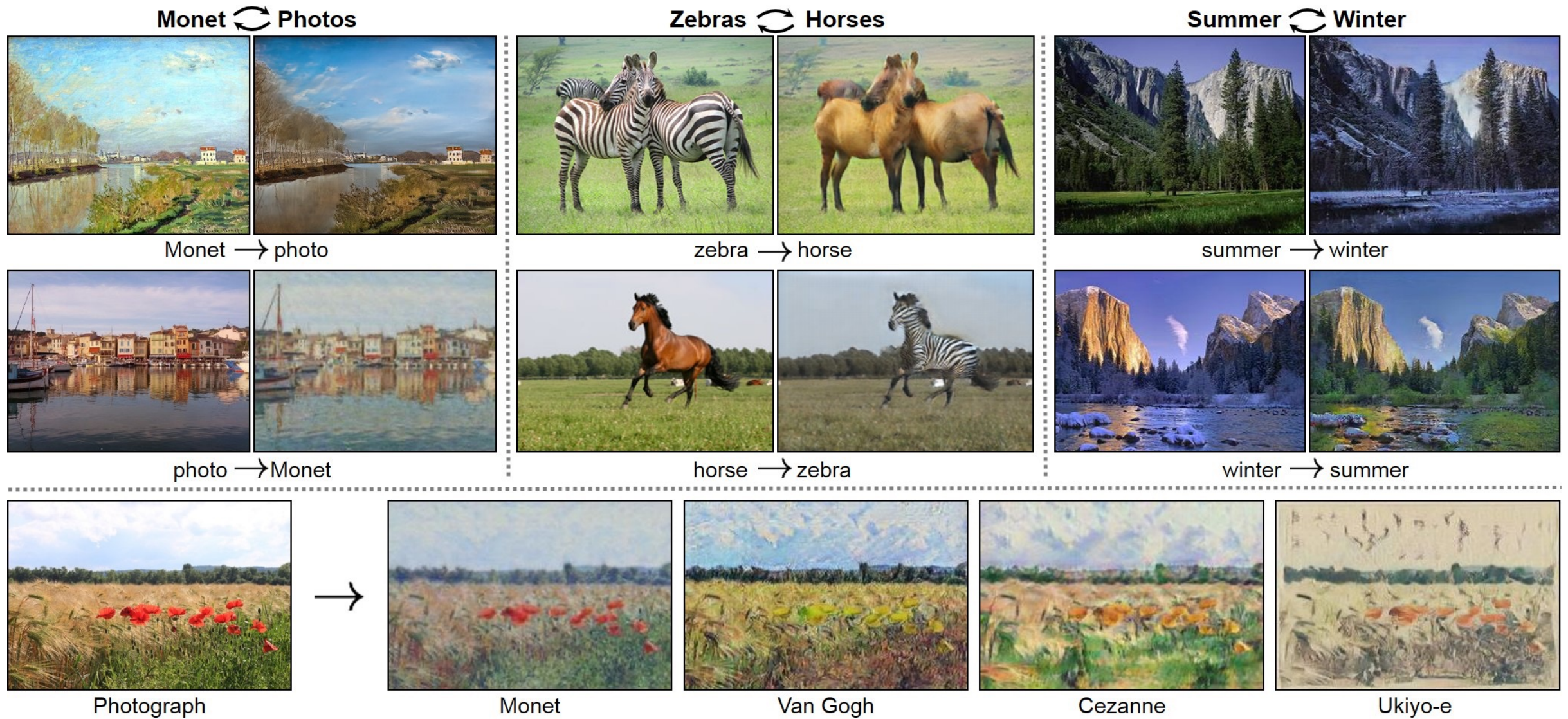


Ivy Tasi @ivymyt



Vitaly Vidmirov @vvid

CycleGAN: Pix2Pix w/o input-output pairs

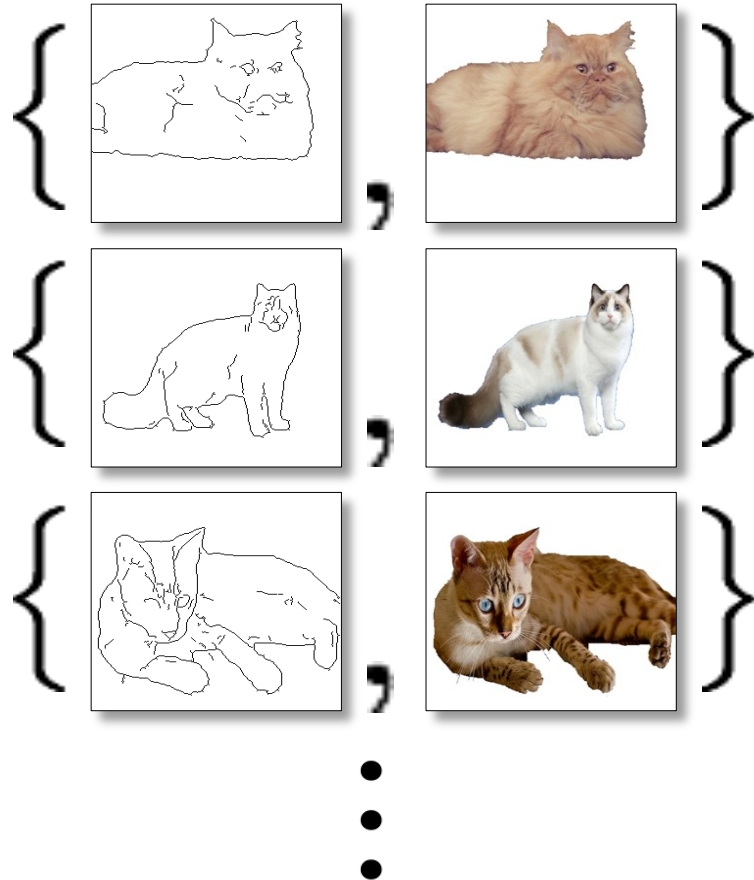


(Zhu et al. 2017)

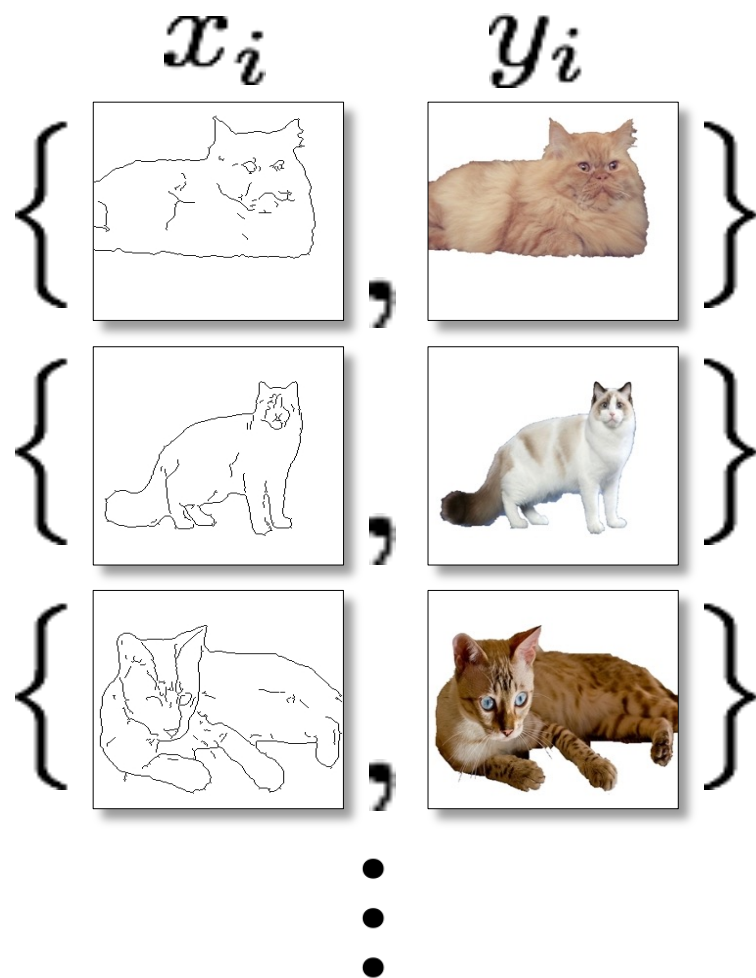
Paired data

x_i

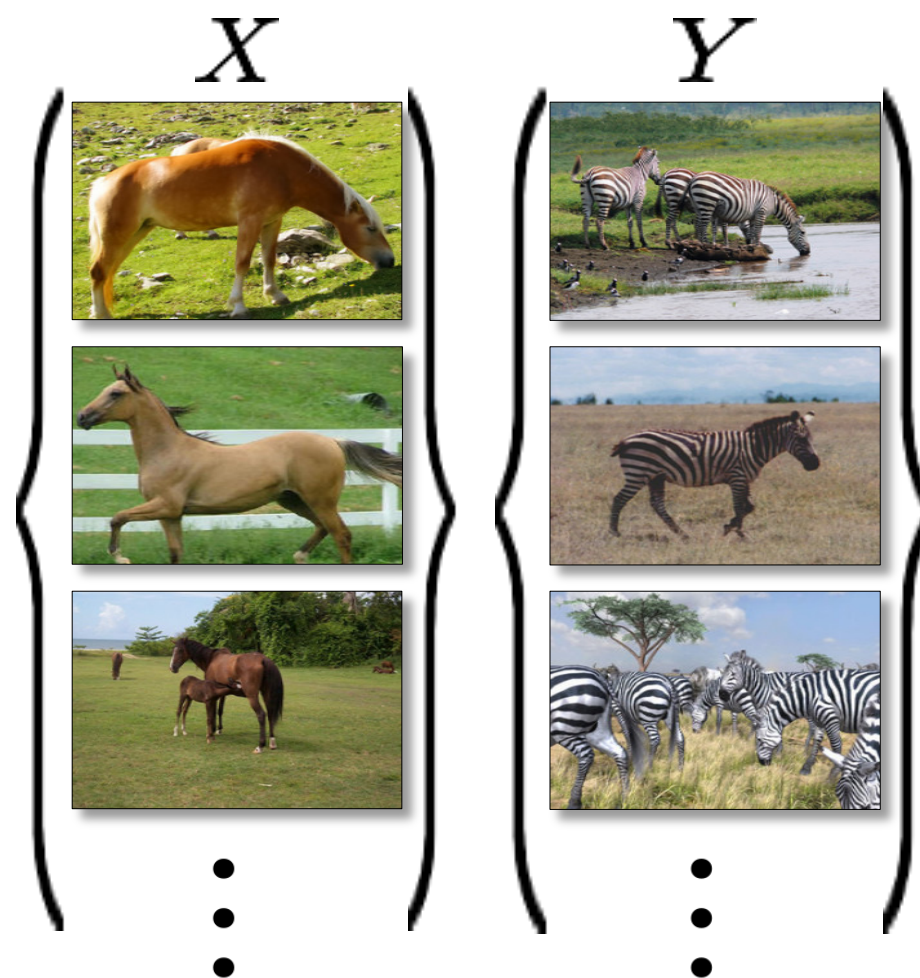
y_i

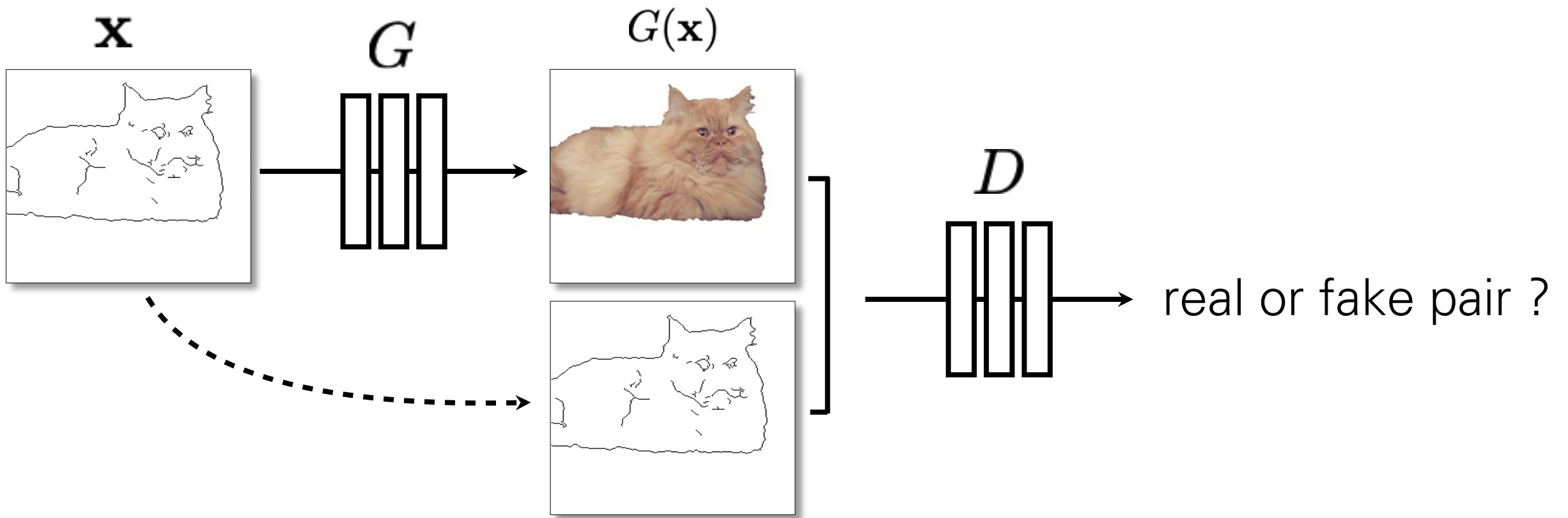


Paired data

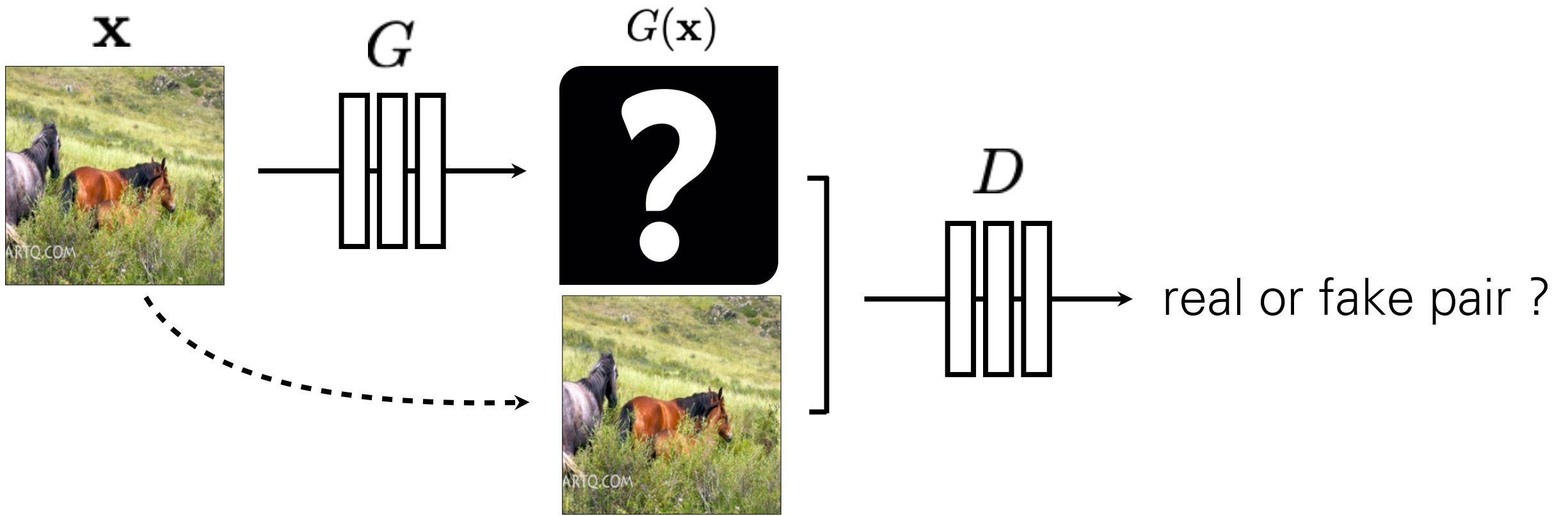


Unpaired data



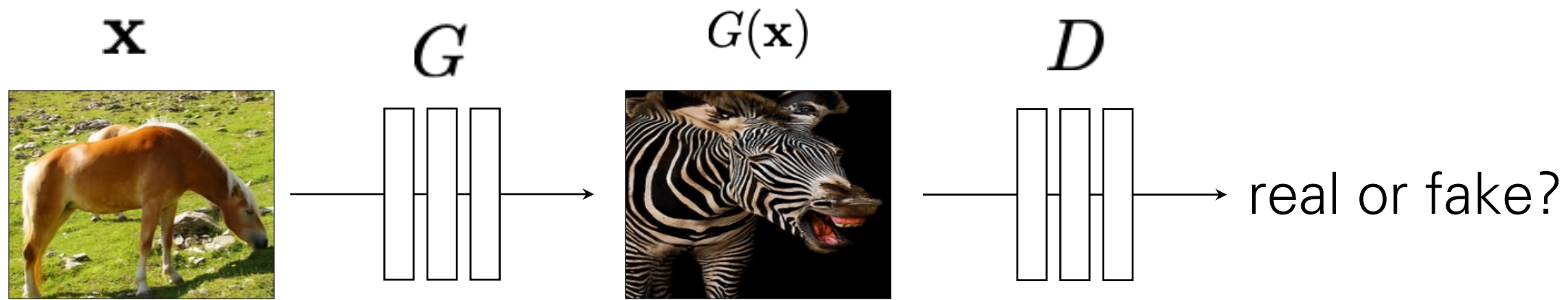


$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))]$$



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

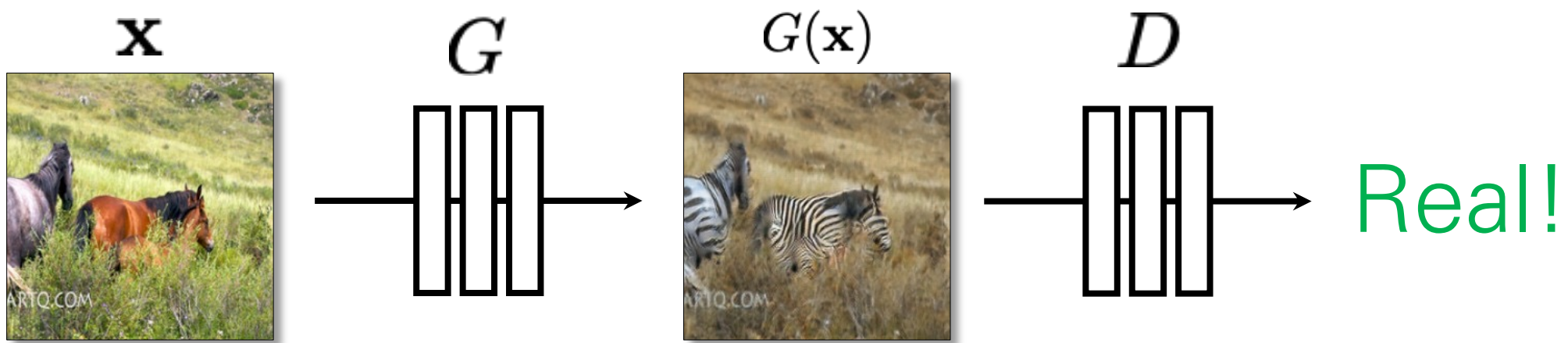
No input-output pairs!

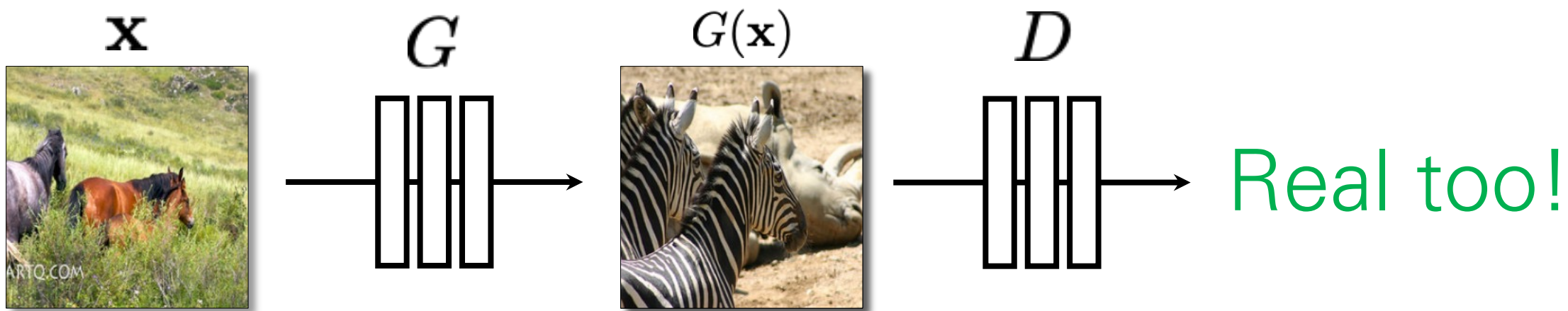


$$\arg \min_G \max_D \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\log D(G(\mathbf{x})) + \log(1 - D(\mathbf{y}))]$$

Usually loss functions check if output matches a target instance

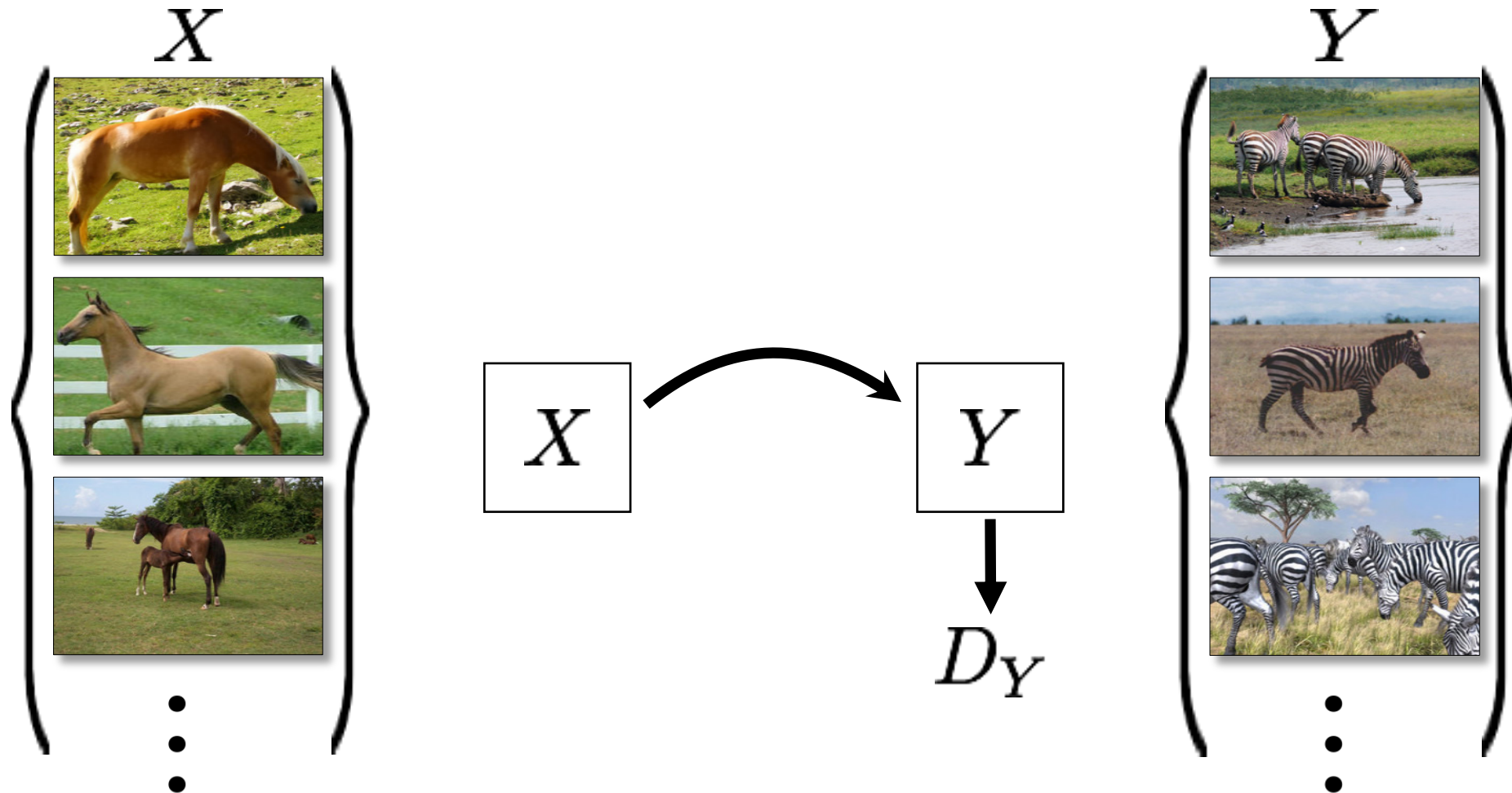
GAN loss checks if output is part of an admissible set





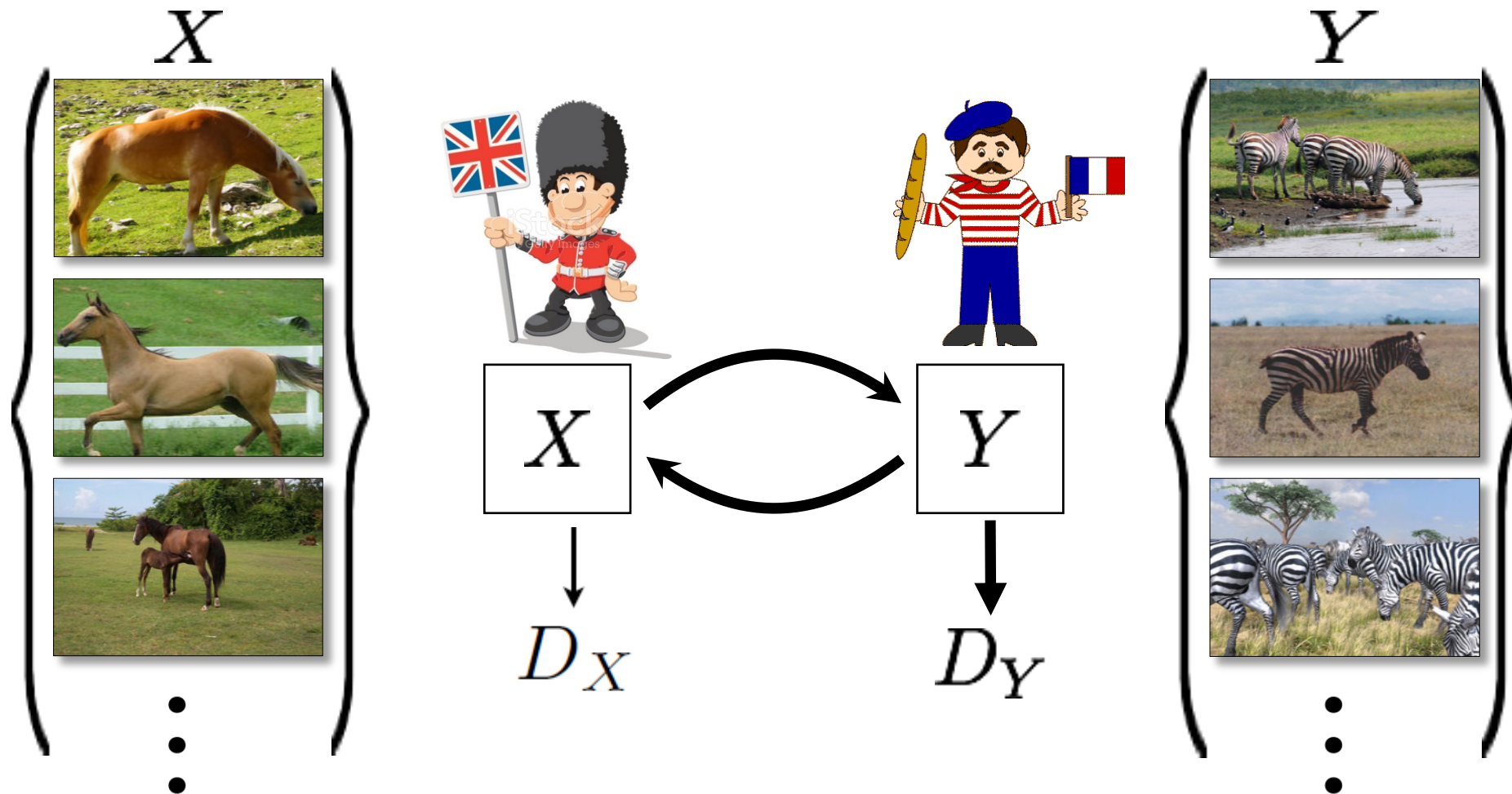
Nothing to force output to correspond to input

Cycle-Consistent Adversarial Networks

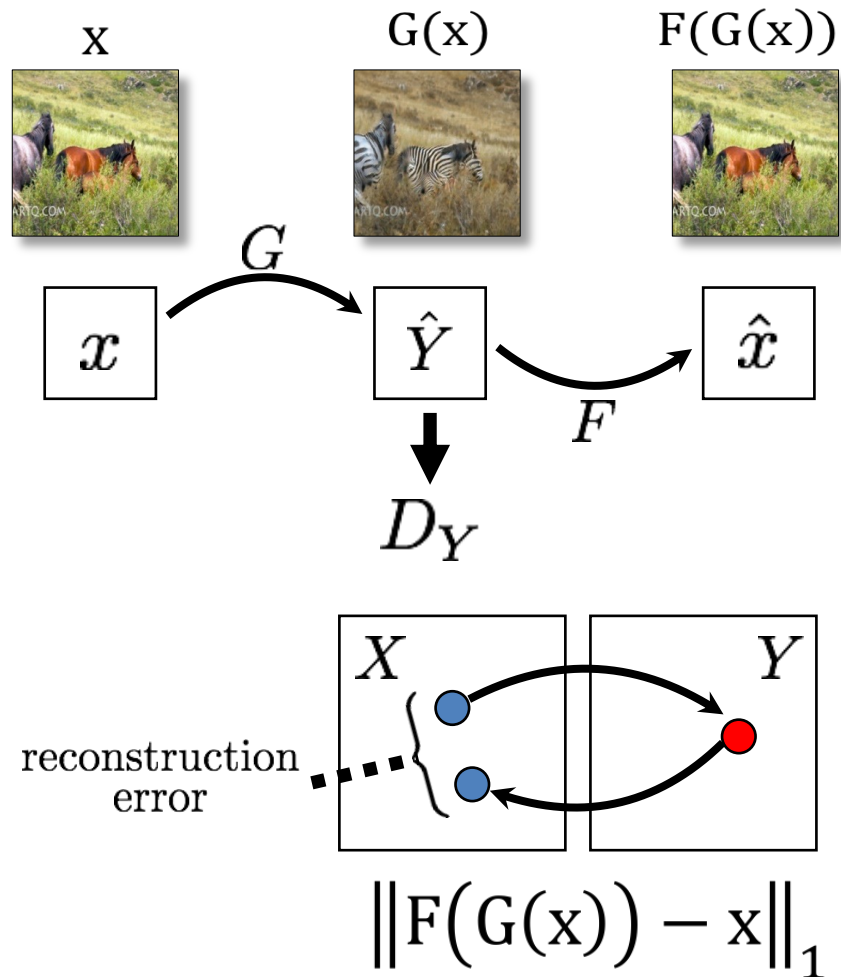


[Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

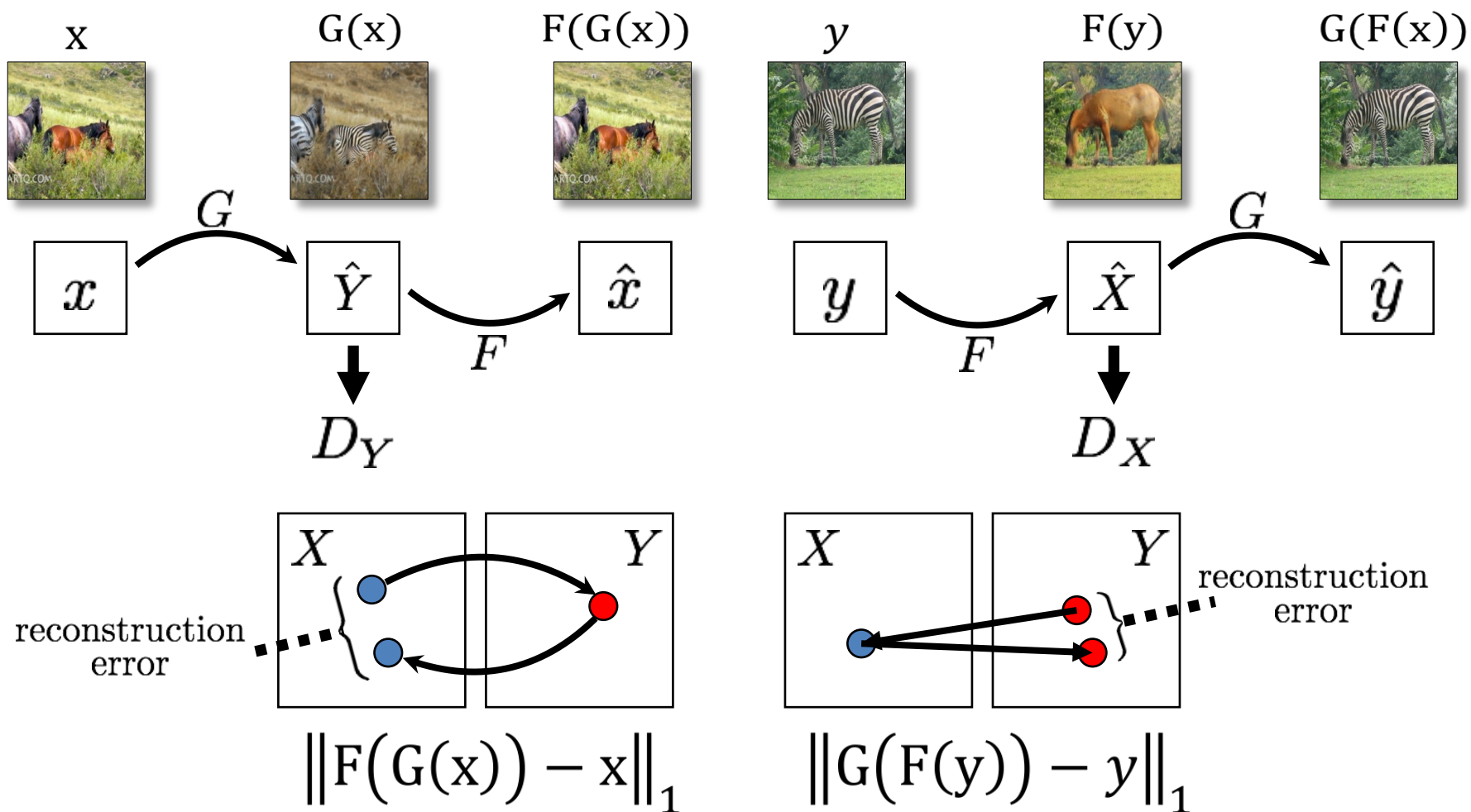
Cycle-Consistent Adversarial Networks



Cycle Consistency Loss



Cycle Consistency Loss





Input



Monet



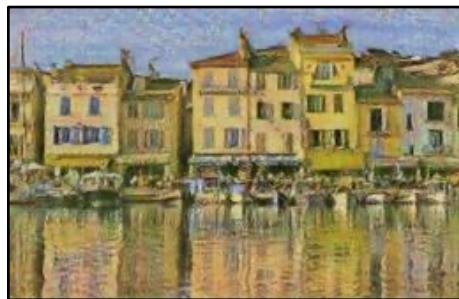
Van Gogh



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



Deep learning



Representations
(encoders)

Models
(decoders)

Foundation models

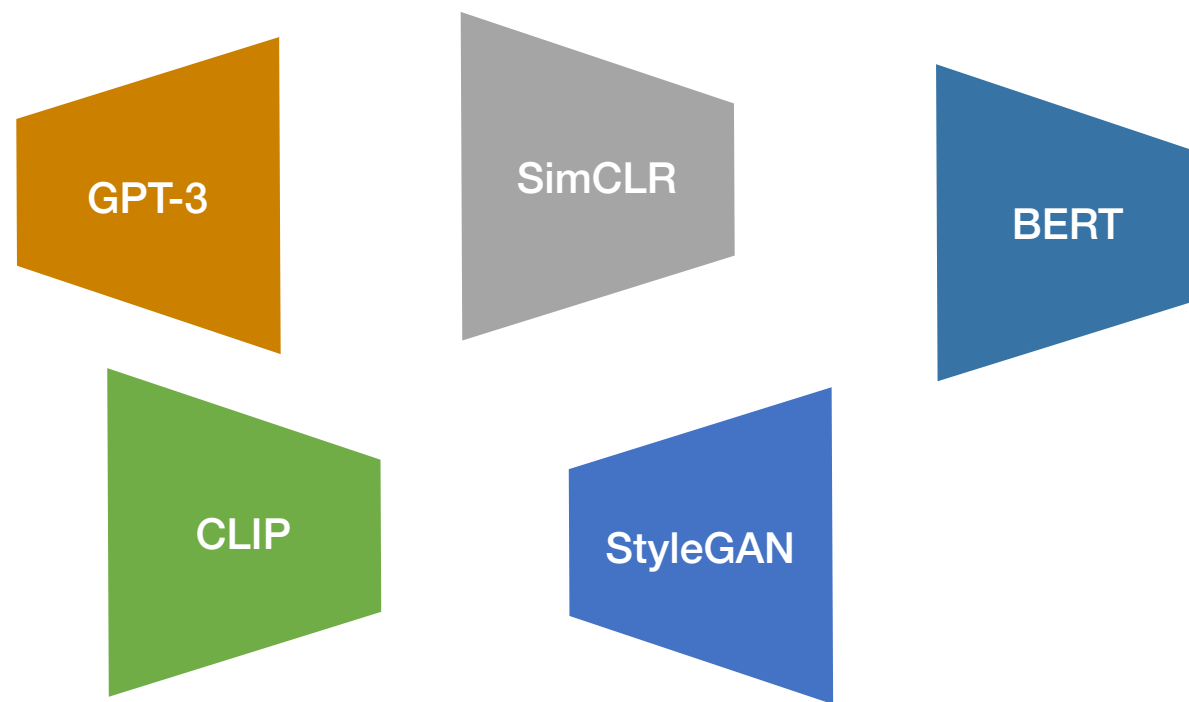
[Bommasani et al. 2021] <https://arxiv.org/pdf/2108.07258.pdf>

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

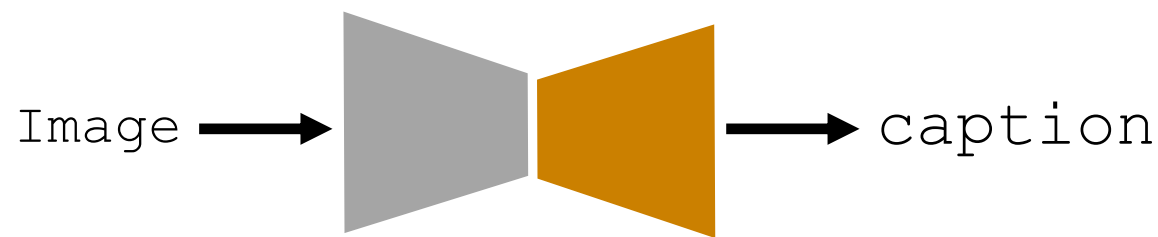


[Blind Orion Searching for the Rising Sun by Nicolas Poussin, 1658]

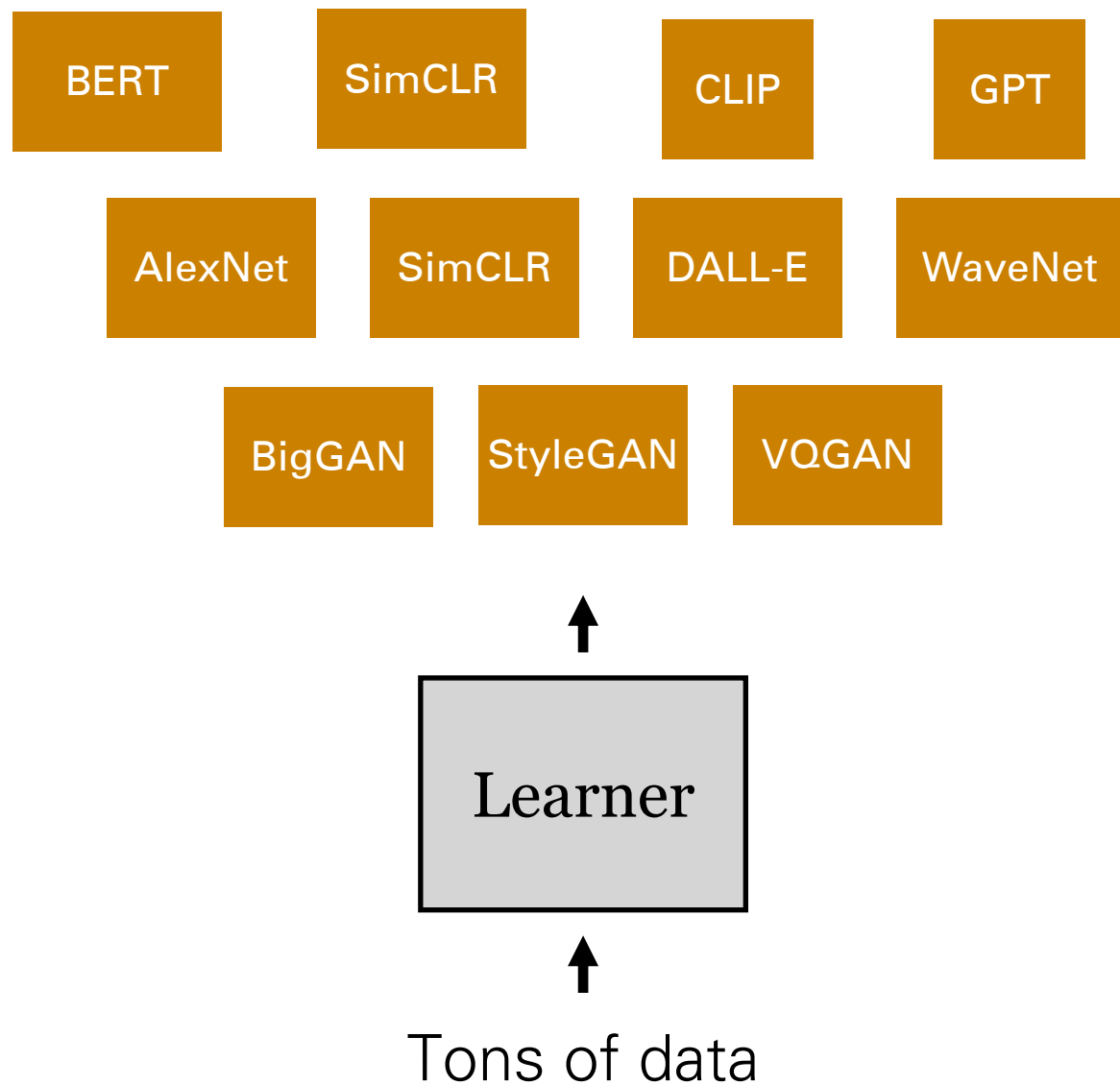
1. Learn foundation model **encoders** and **decoders** for each domain



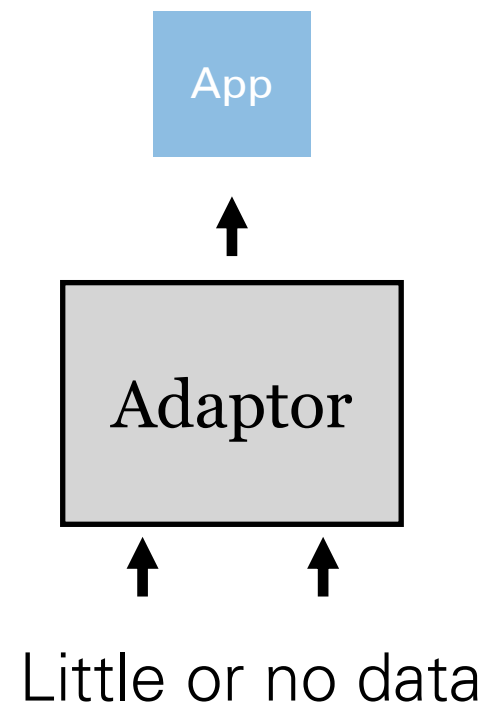
2. Plug them together to translate between modalities (may require finetuning)



Learn foundation models

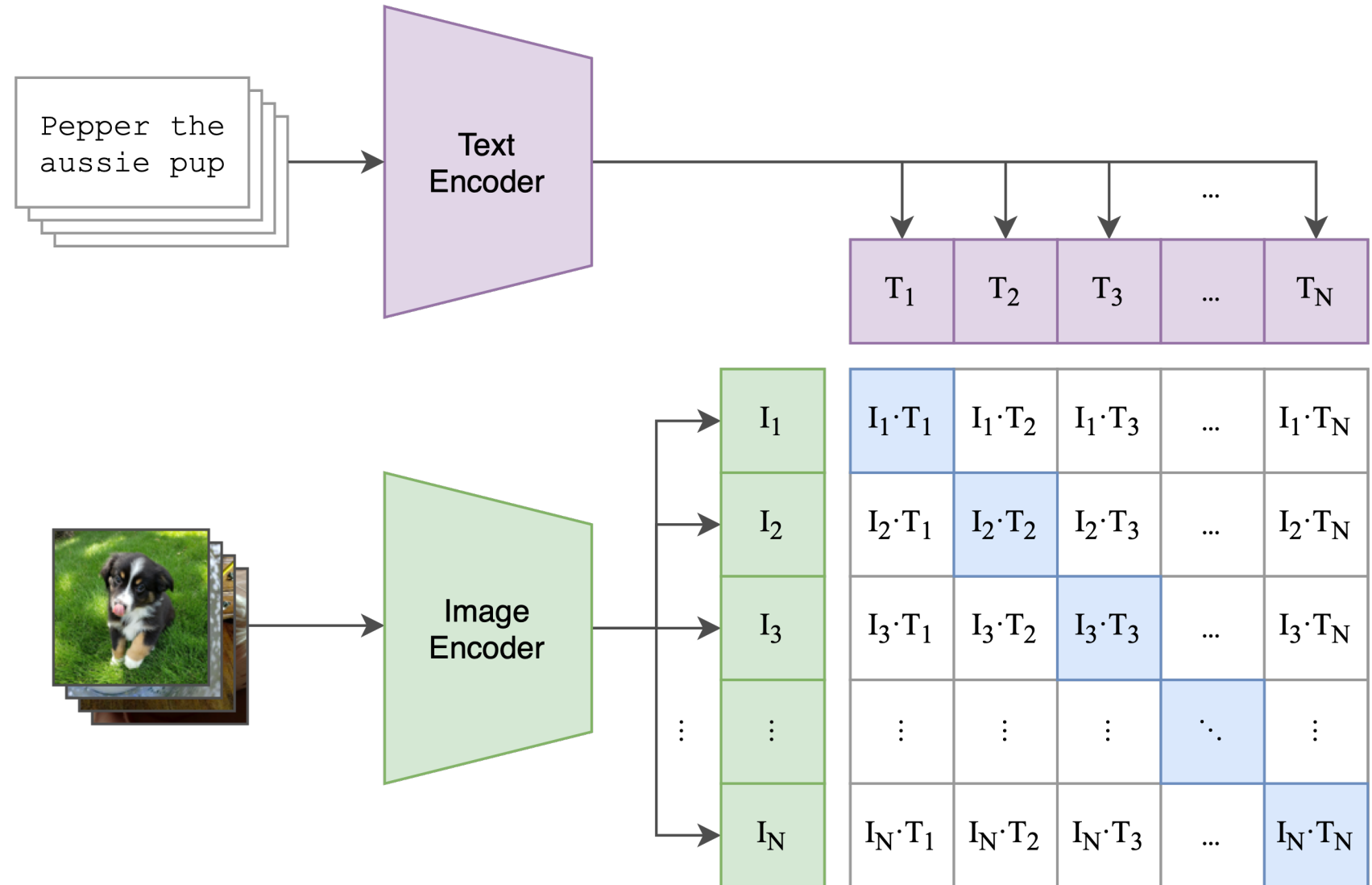


Use/adapt foundations to solve new problems



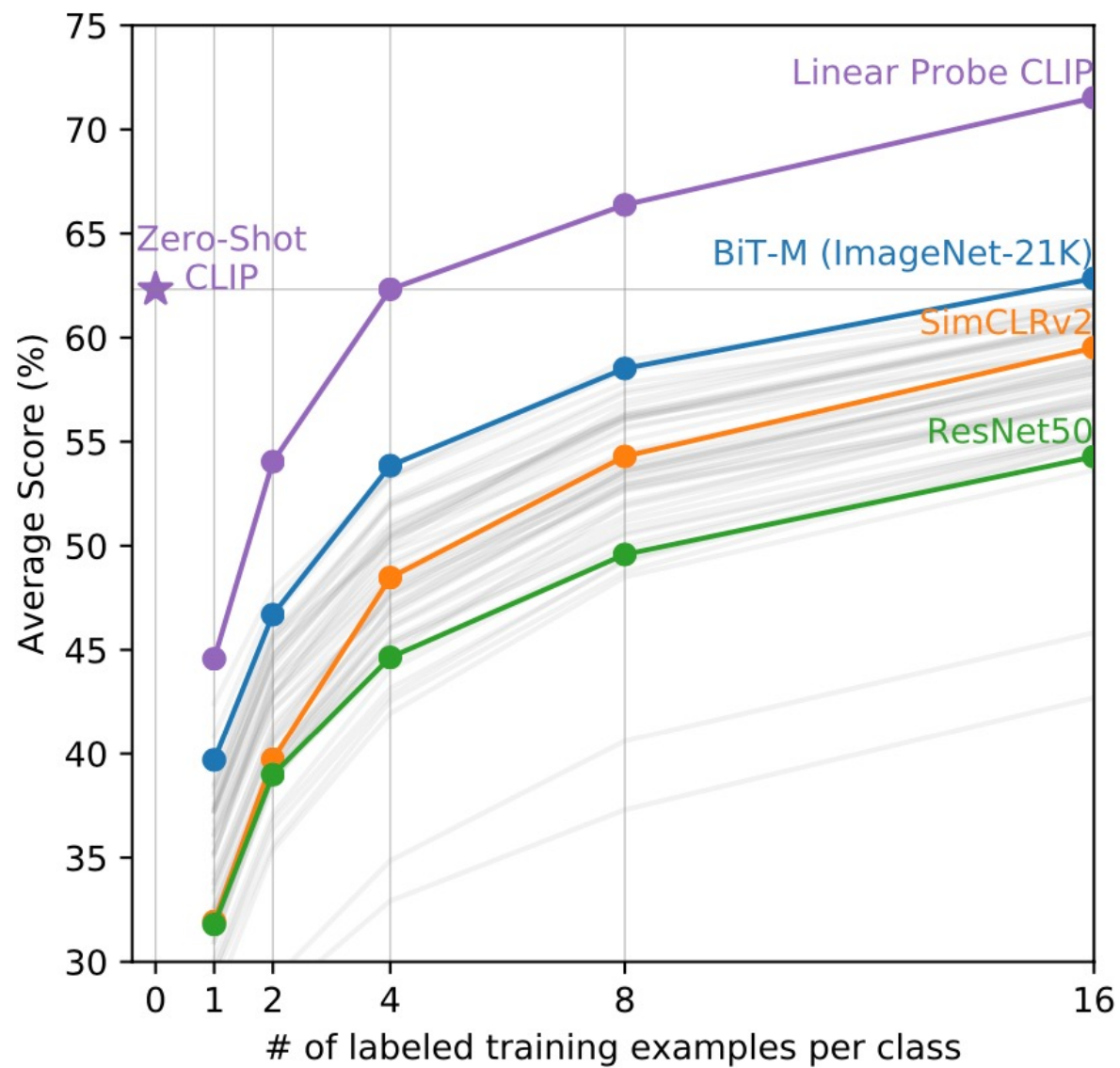
CLIP

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



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

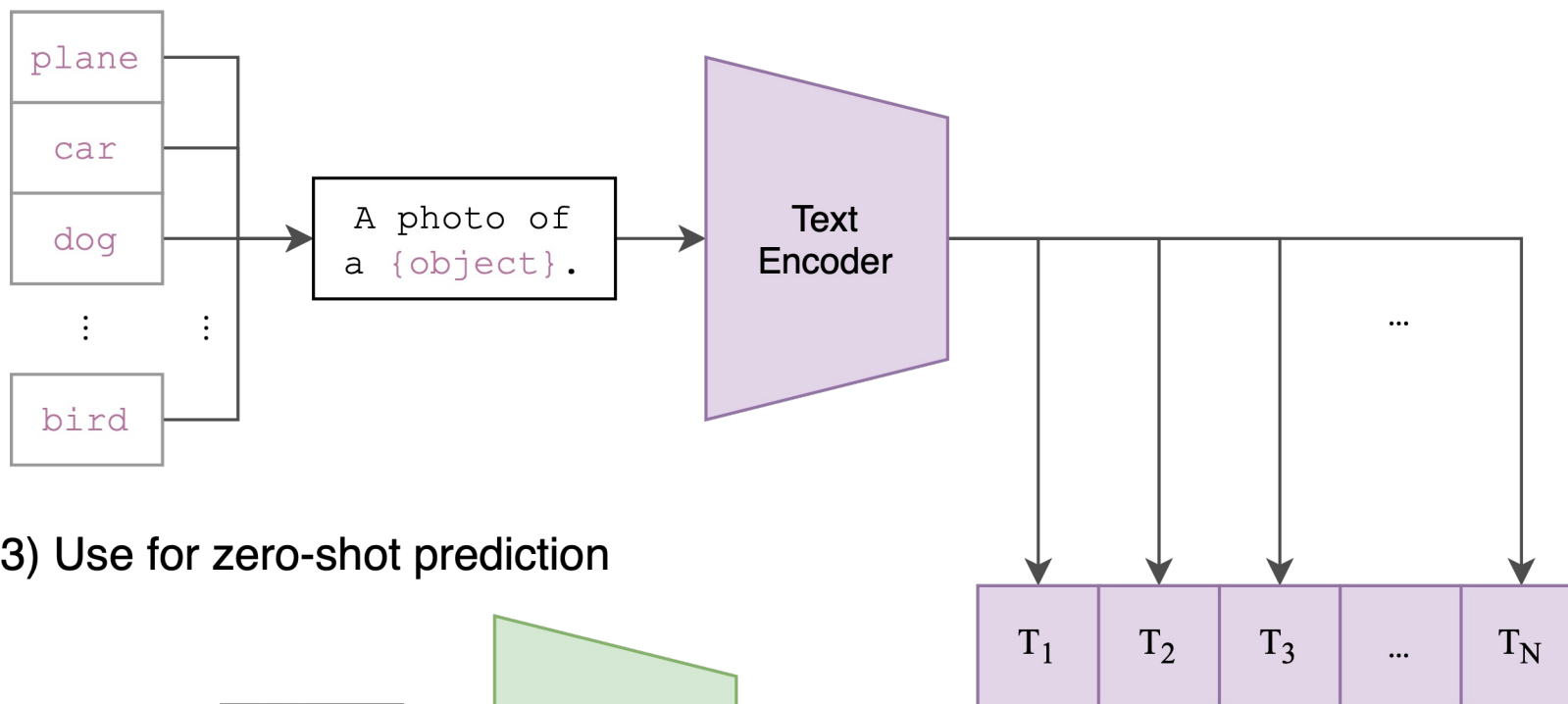
2. Adaptor: Linear classifier on top of image encodings



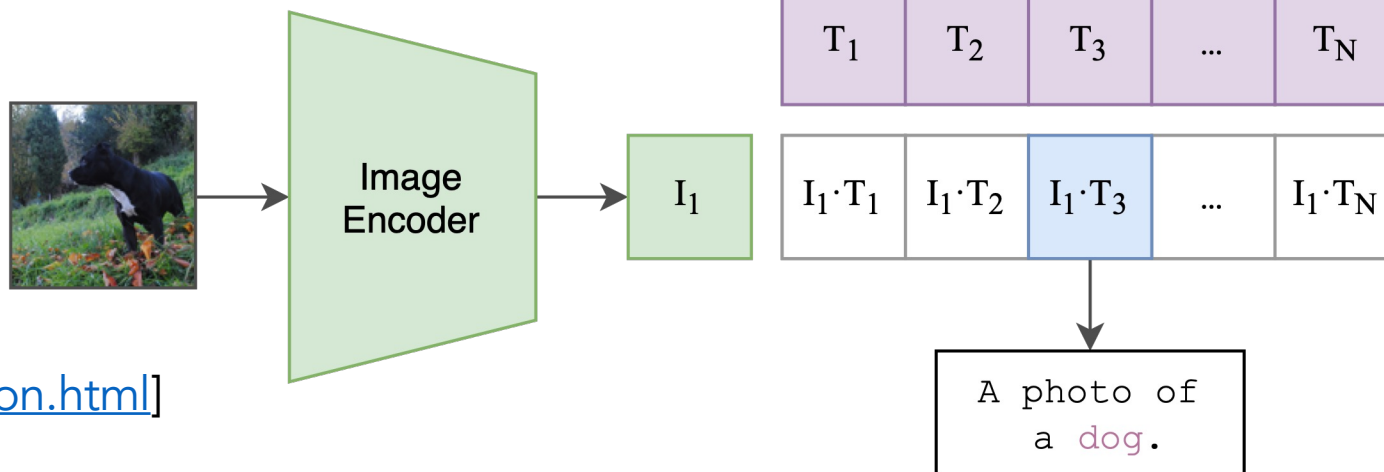
CLIP

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

(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>]

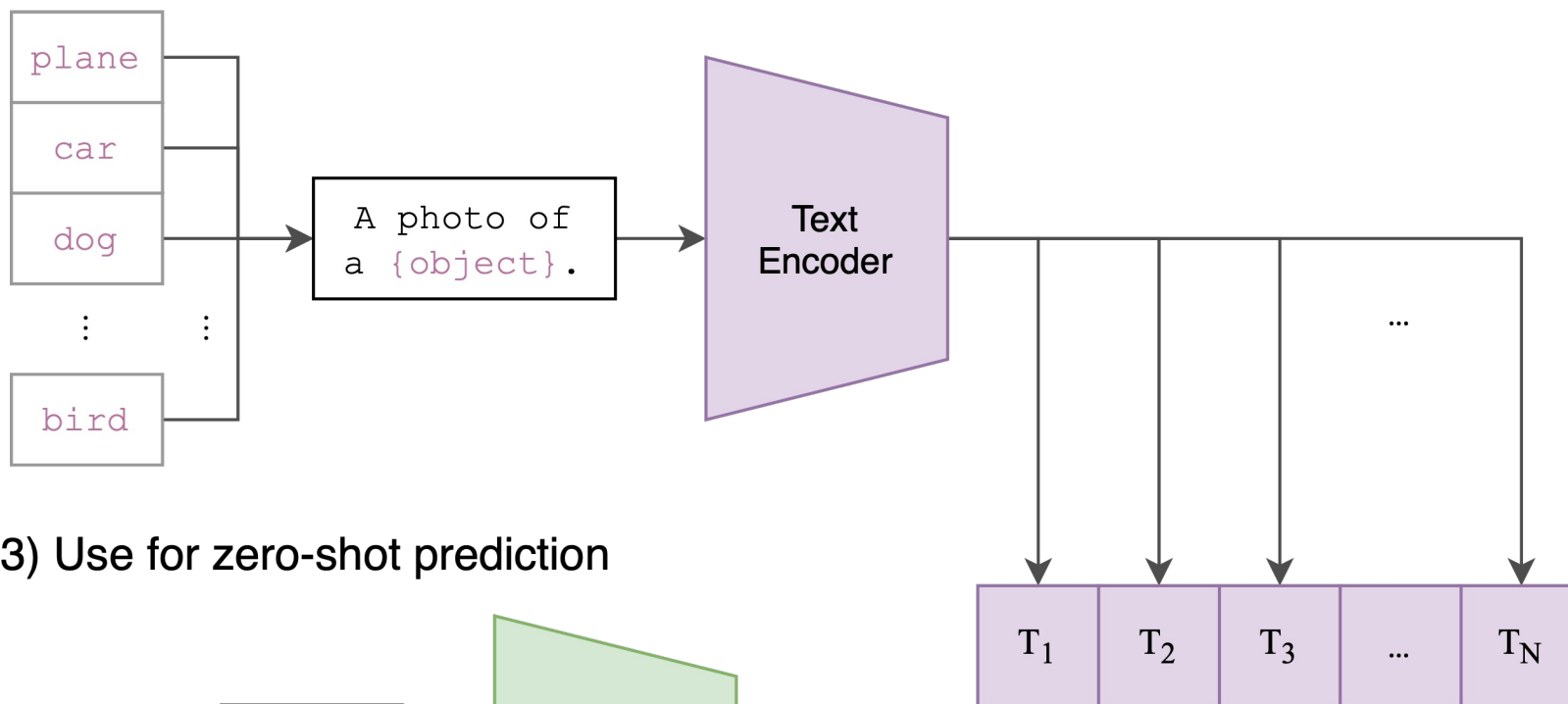
[<https://openai.com/blog/clip/>]

CLIP

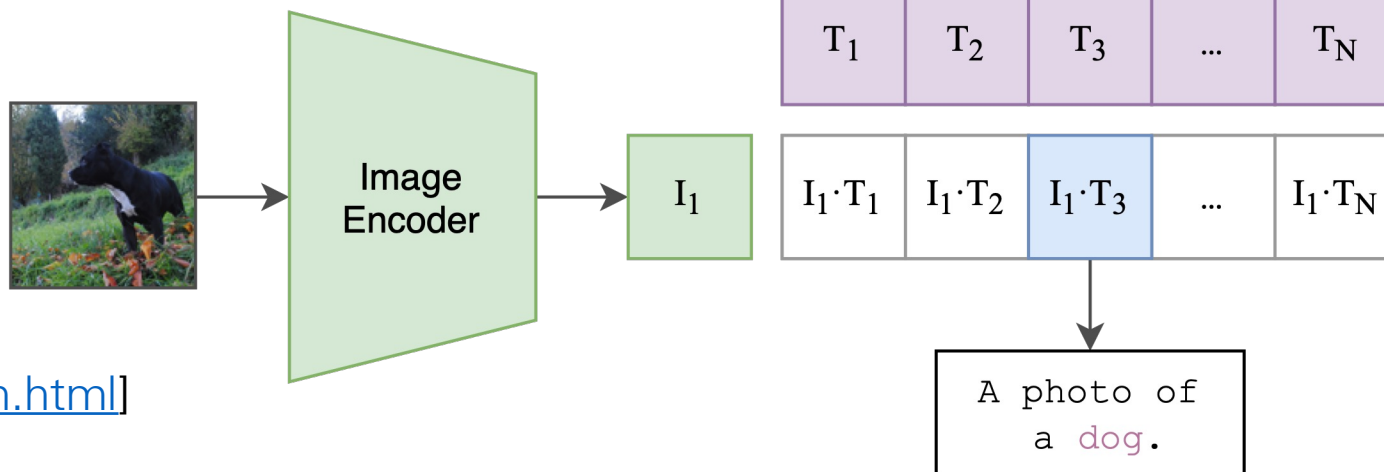
[Radford et al., 2021] <https://arxiv.org/pdf/2103.00020.pdf>

2. Adaptor: Just ask

(2) Create dataset classifier from label text



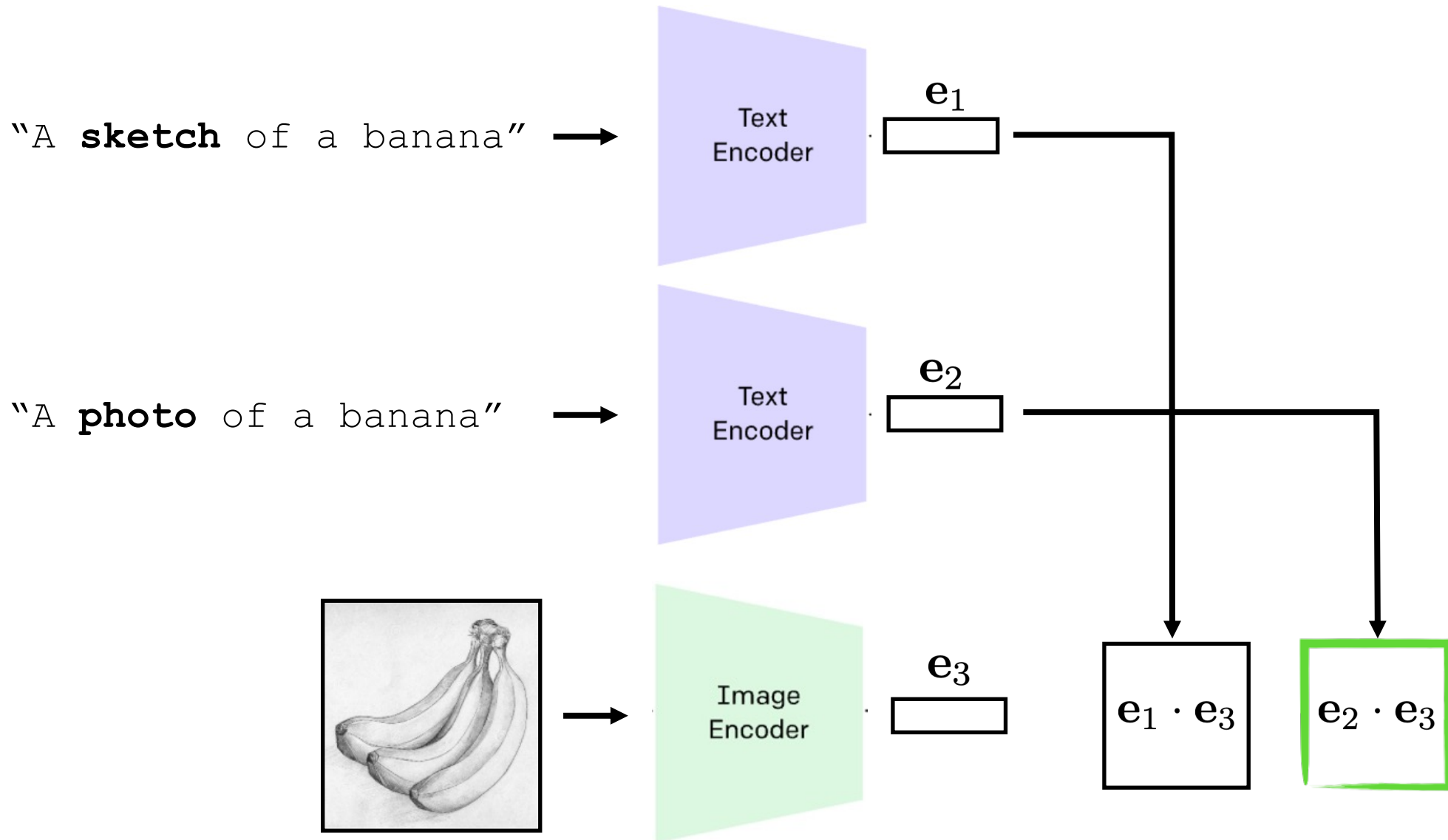
(3) Use for zero-shot prediction



[\[https://evjang.com/2021/10/23/generalization.html\]](https://evjang.com/2021/10/23/generalization.html)

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

Z

Image Generator

OUTPUT:



Text Encoder

e_1

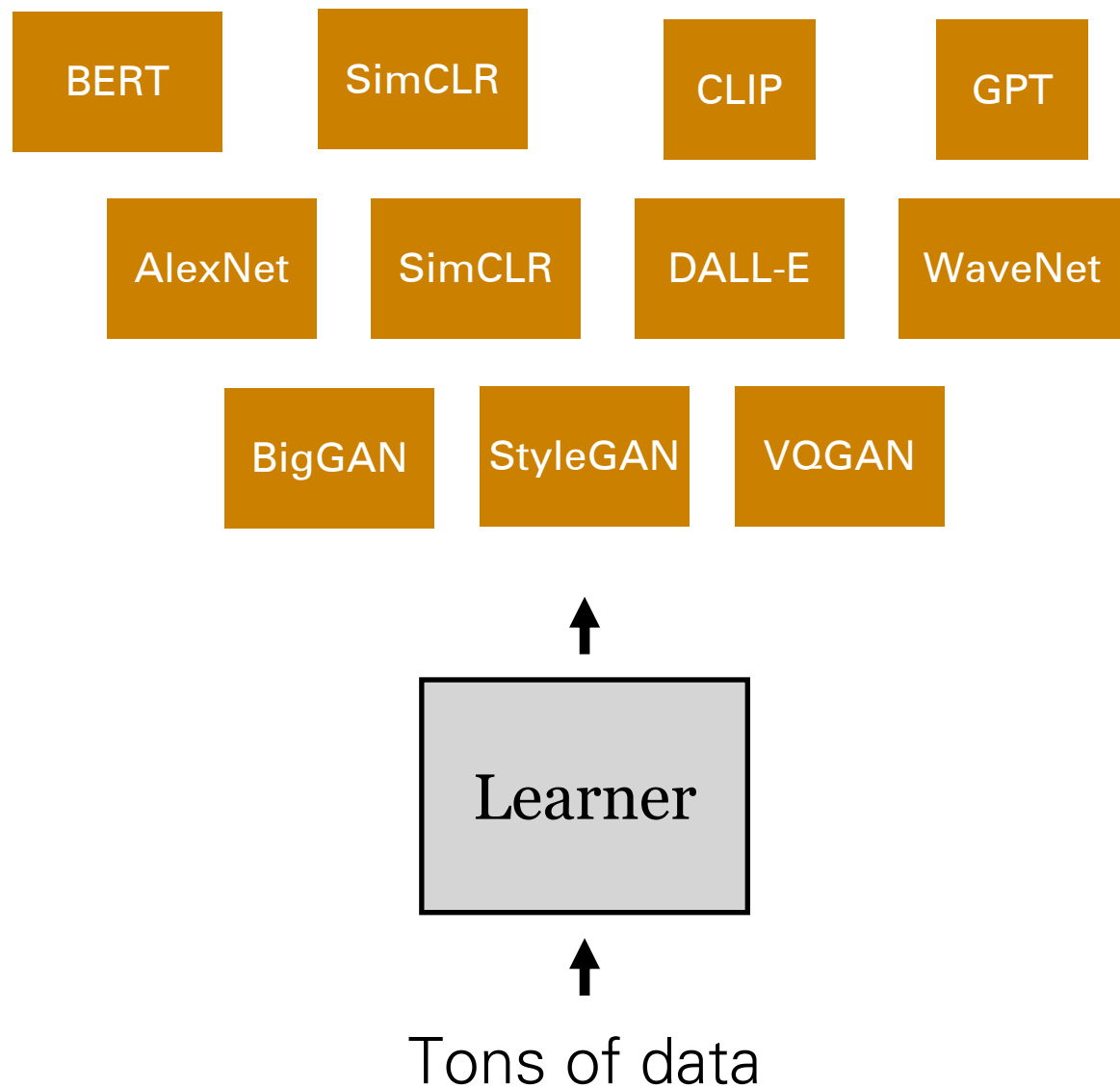
Image Encoder

e_2

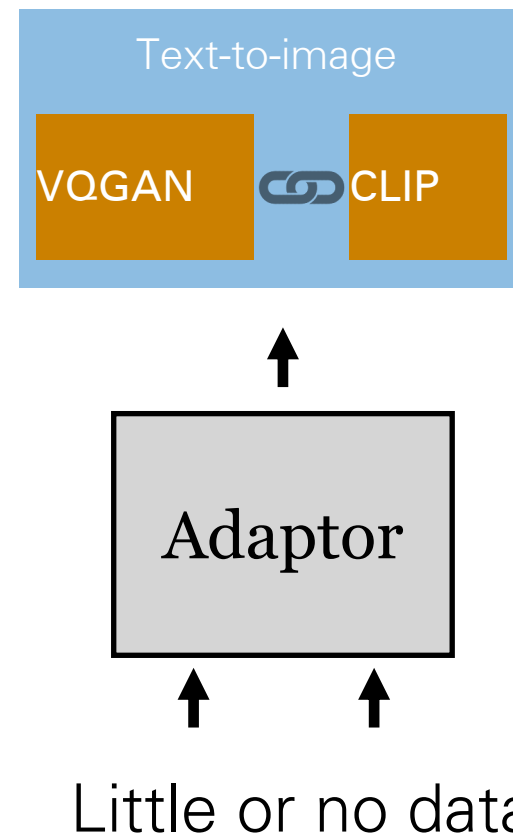
e_1 e_2

To maximize this

Learn foundation models



Use/adapt foundations to solve new problems



DALL-E

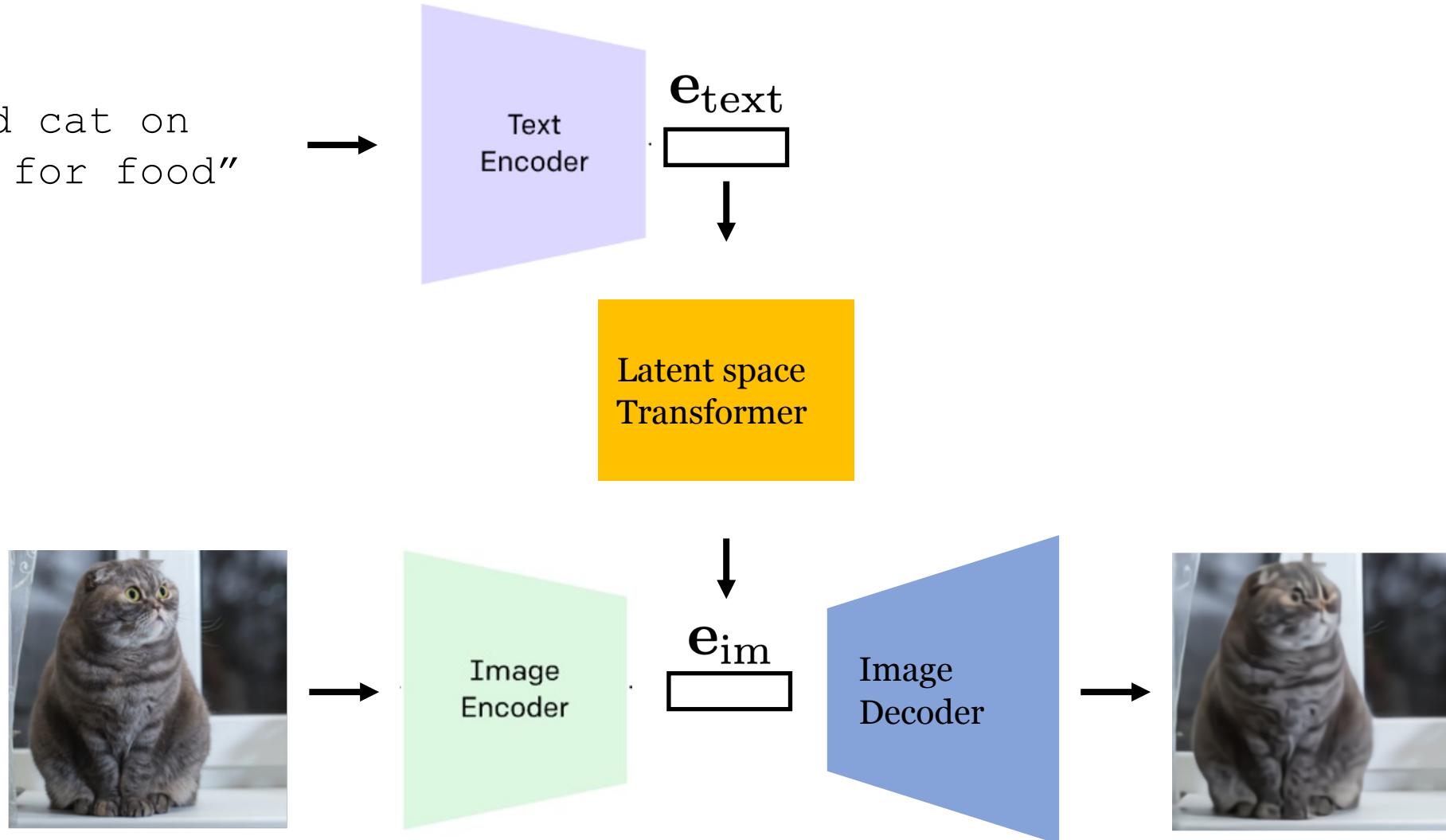
[Ramesh et al. 2021]

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

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

INPUT:

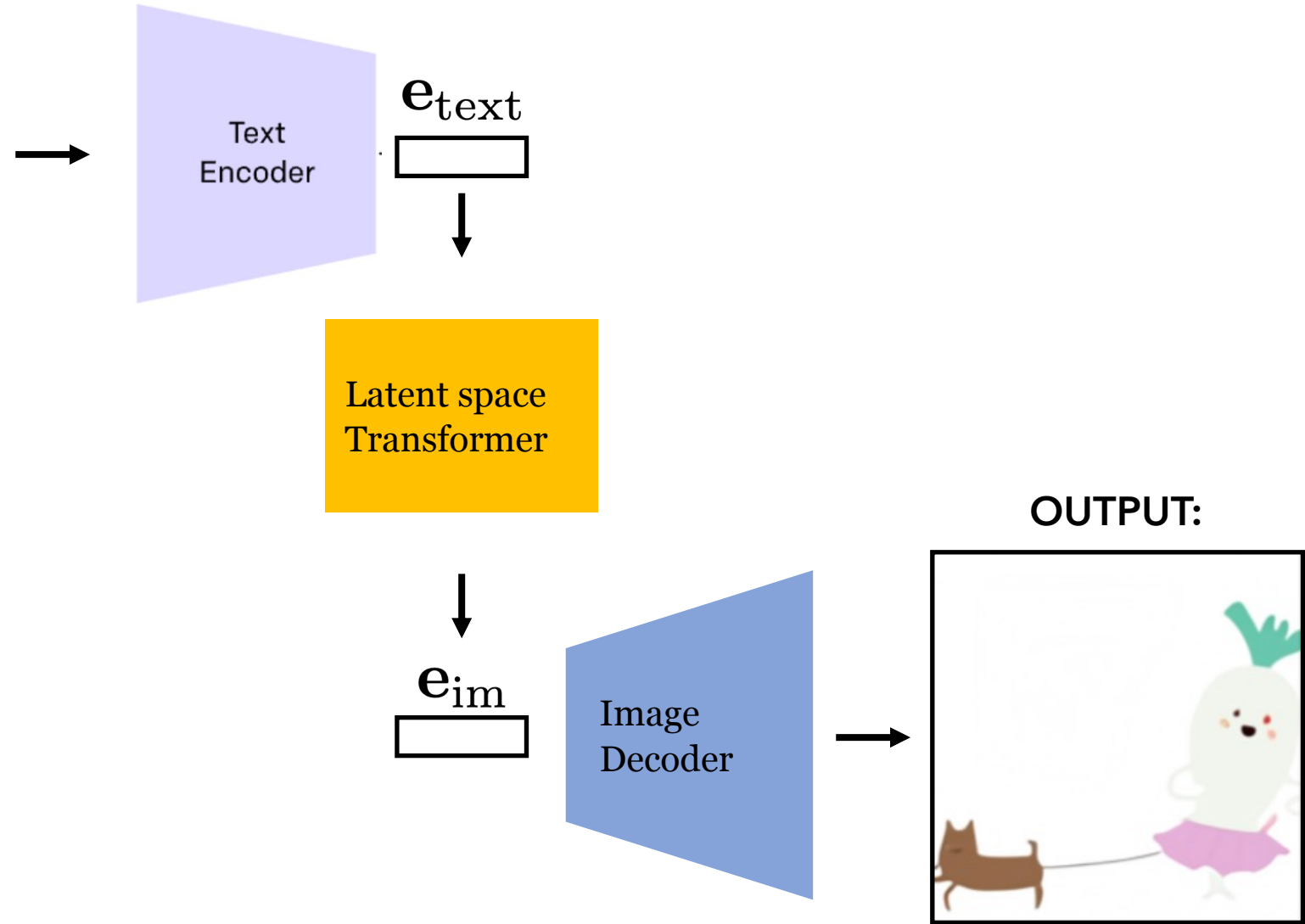
"A wide-eyed cat on
the lookout for food"



Text-to-image translation

INPUT:

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



New capabilities by just asking: product design

TEXT PROMPT

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

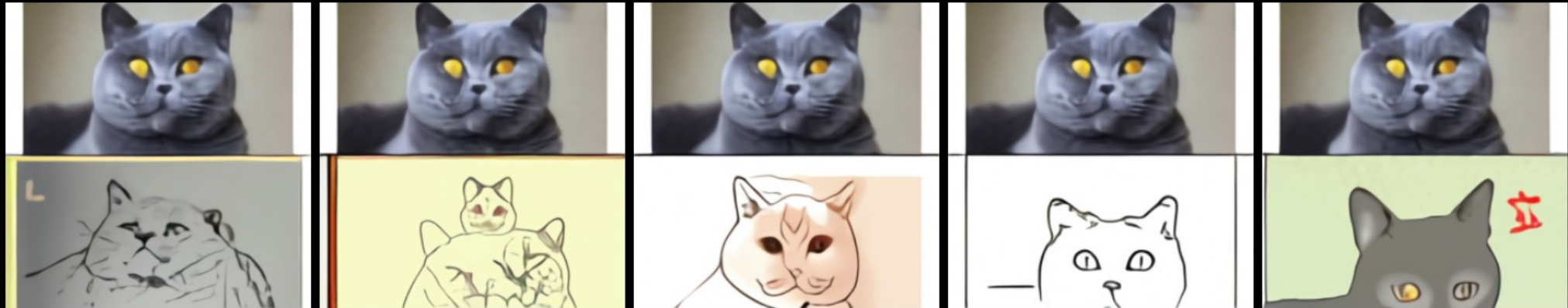
AI-GENERATED
IMAGES



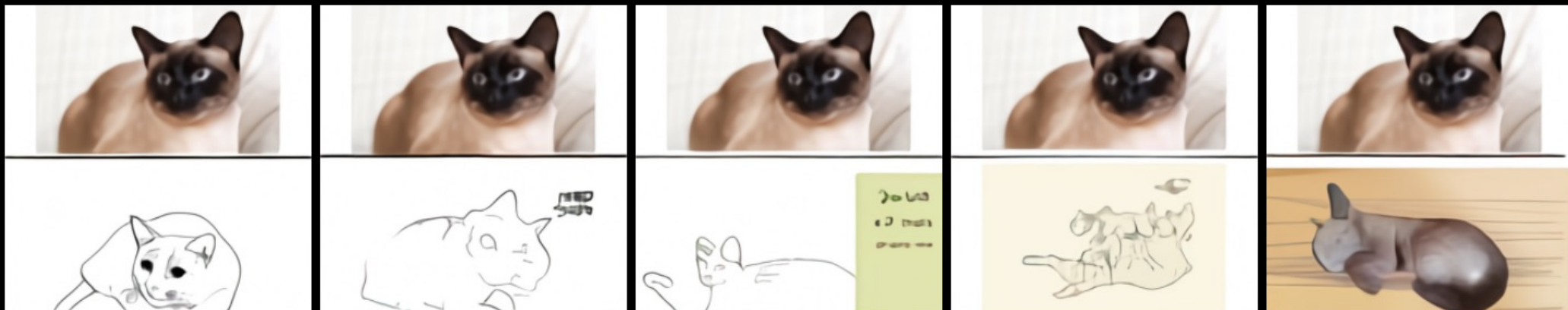
New capabilities by just asking: image translation

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

AI-GENERATED
IMAGES



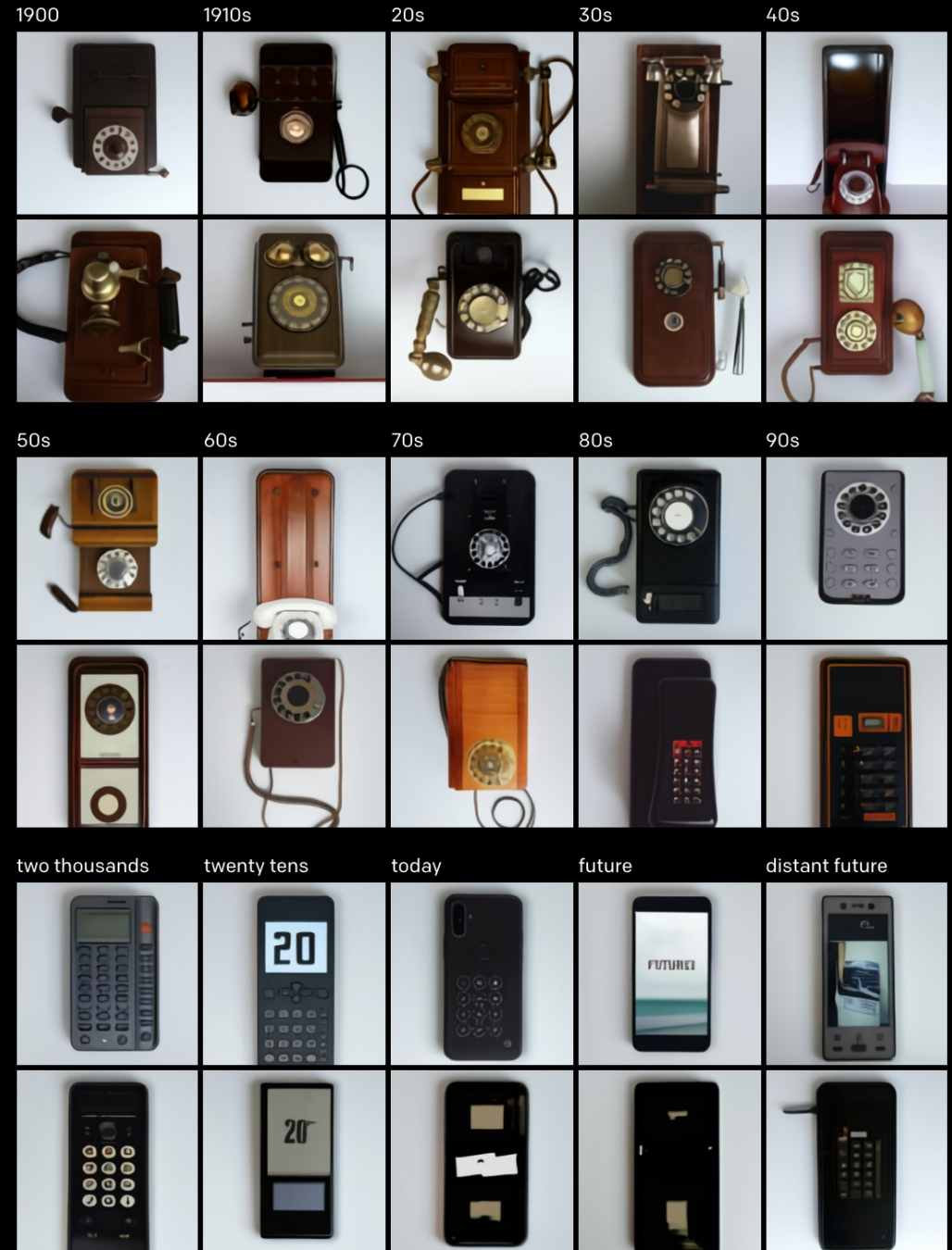
AI-GENERATED
IMAGES



TEXT PROMPT

a photo of a phone from the ...

AI-GENERATED
IMAGES



TEXT PROMPT

a photo of a computer from the ...

AI-GENERATED
IMAGES

1900

1910s

20s

30s

40s



50s

60s

70s

80s

90s



two thousands

twenty tens

today

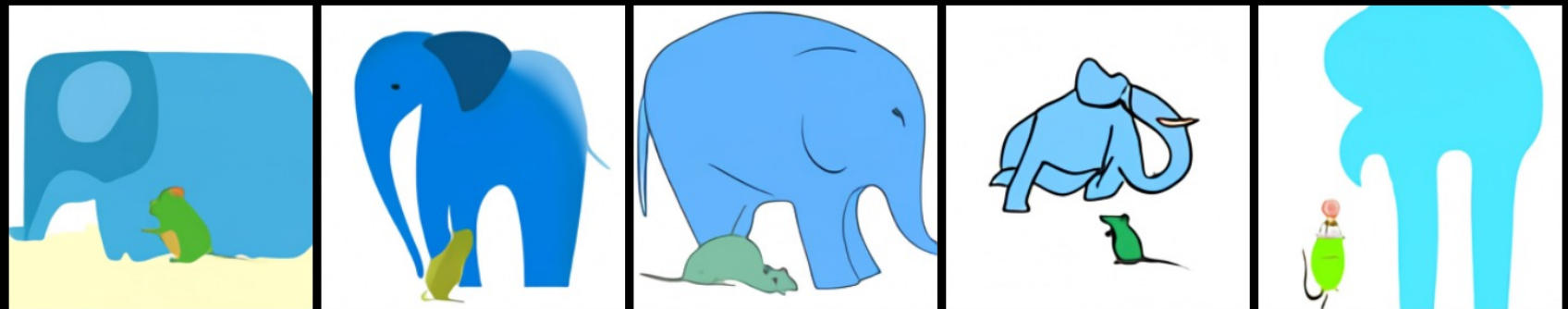
future

distant future



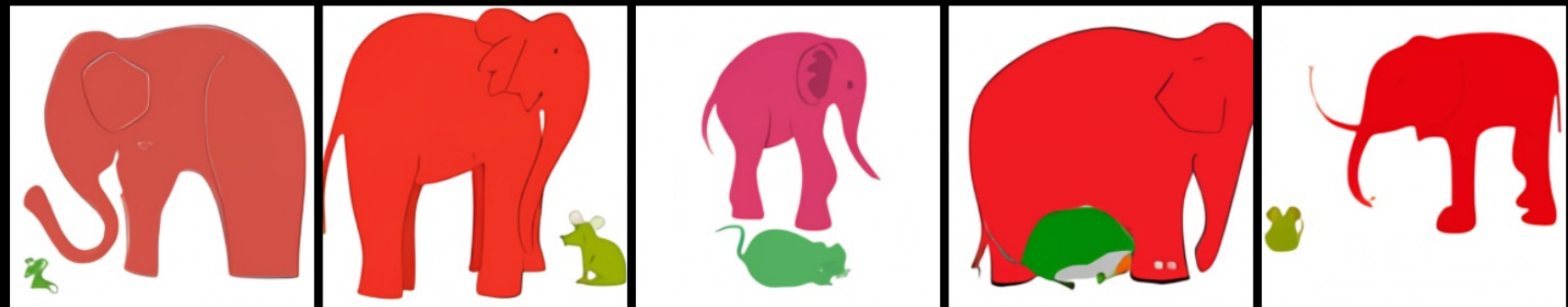
TEXT PROMPT an illustration of a small green mouse sitting below a large blue elephant

AI-GENERATED IMAGES



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


AI-GENERATED IMAGES

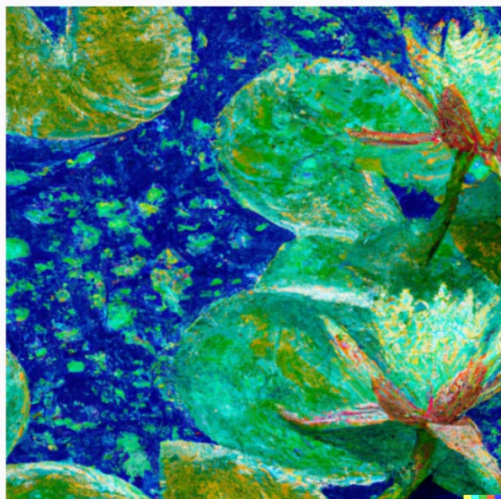
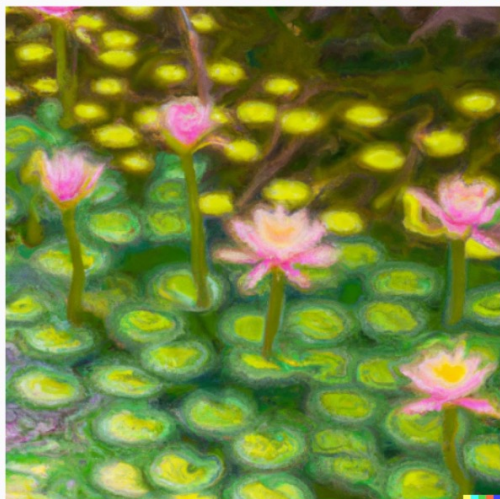
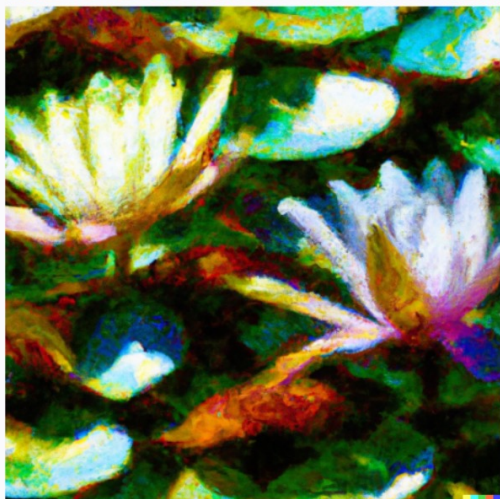
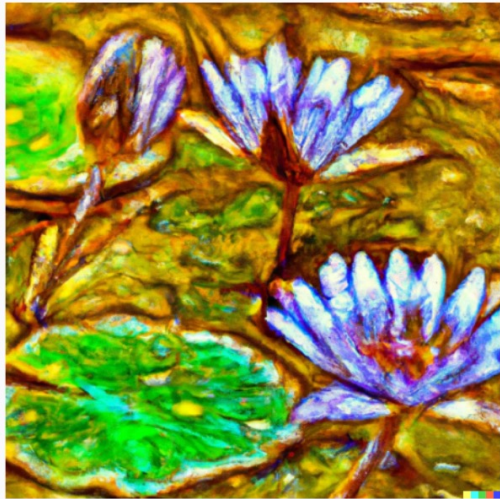
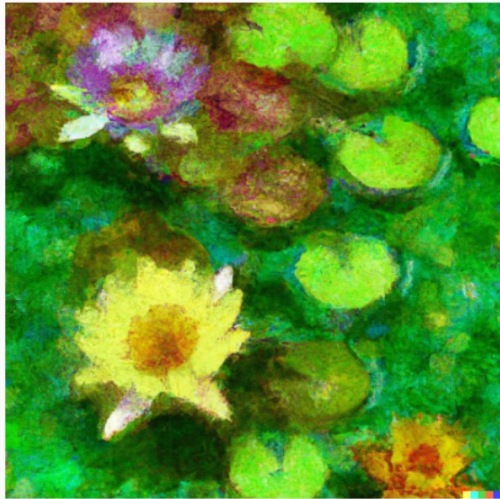


DALL-E 2

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



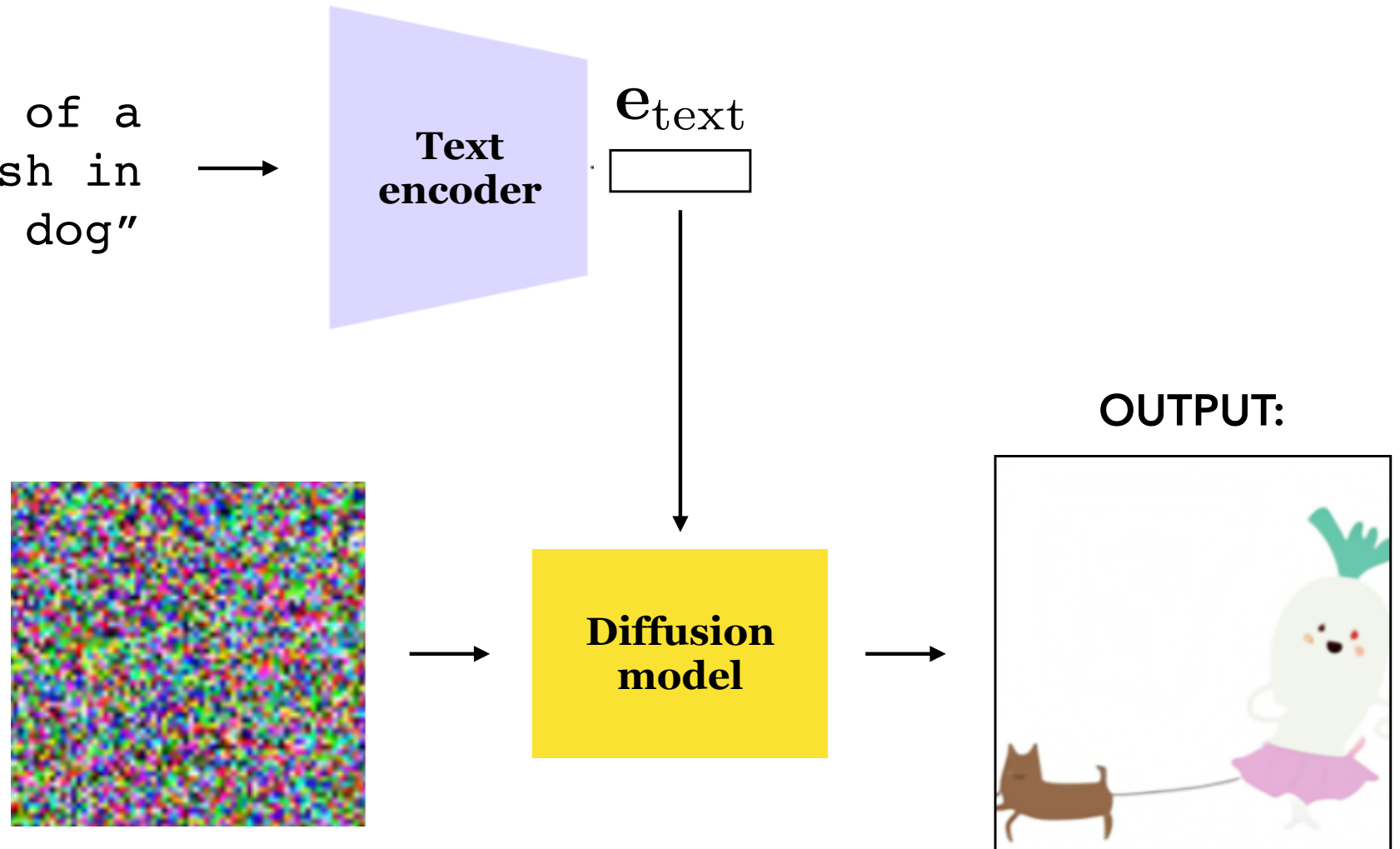
Report issue 



DALL-E 2 [Ramesh et al. 2022] <https://cdn.openai.com/papers/dall-e-2.pdf>

INPUT:

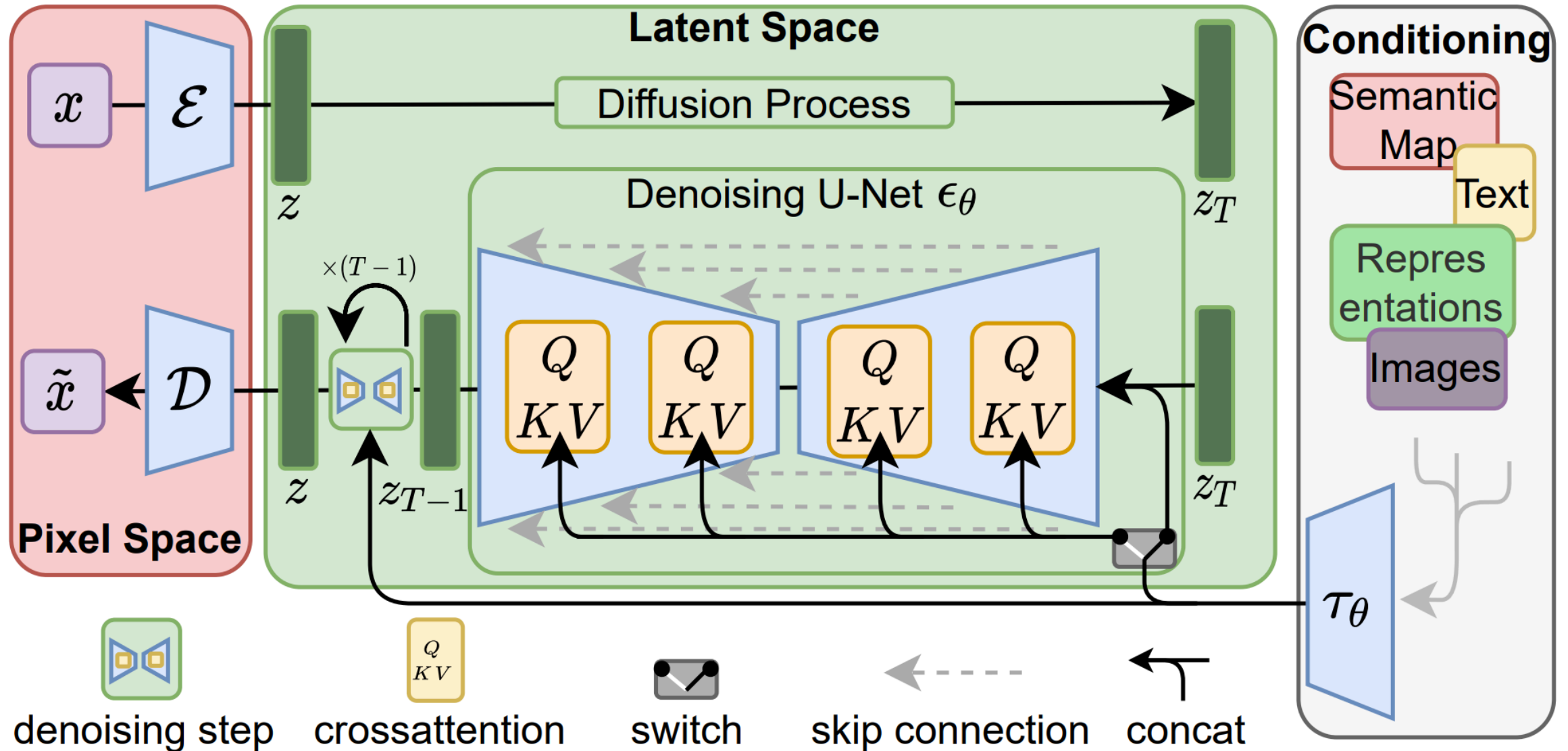
"An illustration of a baby daikon radish in a tutu walking a dog"



Latent diffusion

[Rombach*, Blattman* et al. 2022]

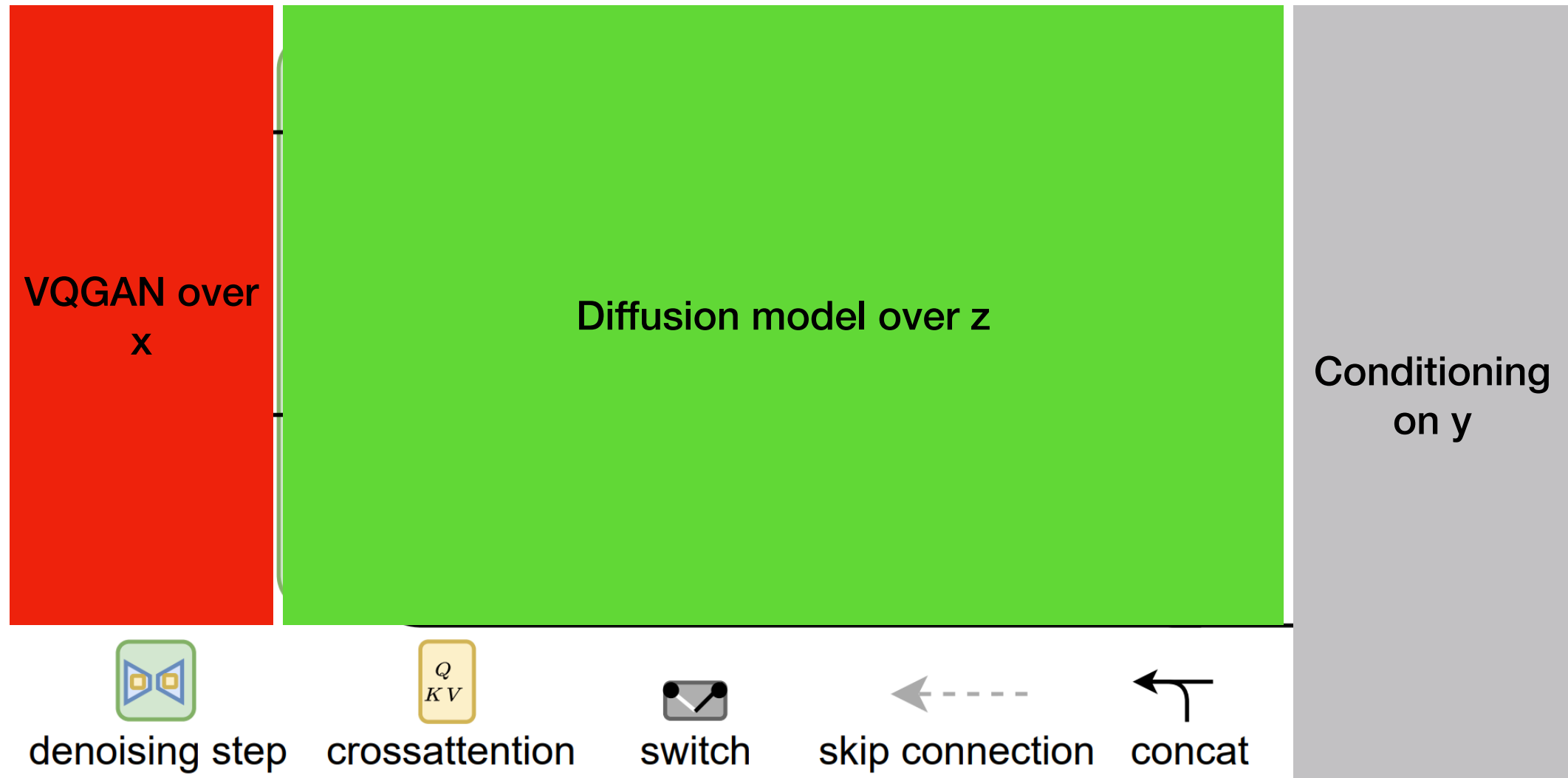
<https://arxiv.org/abs/2112.10752>



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

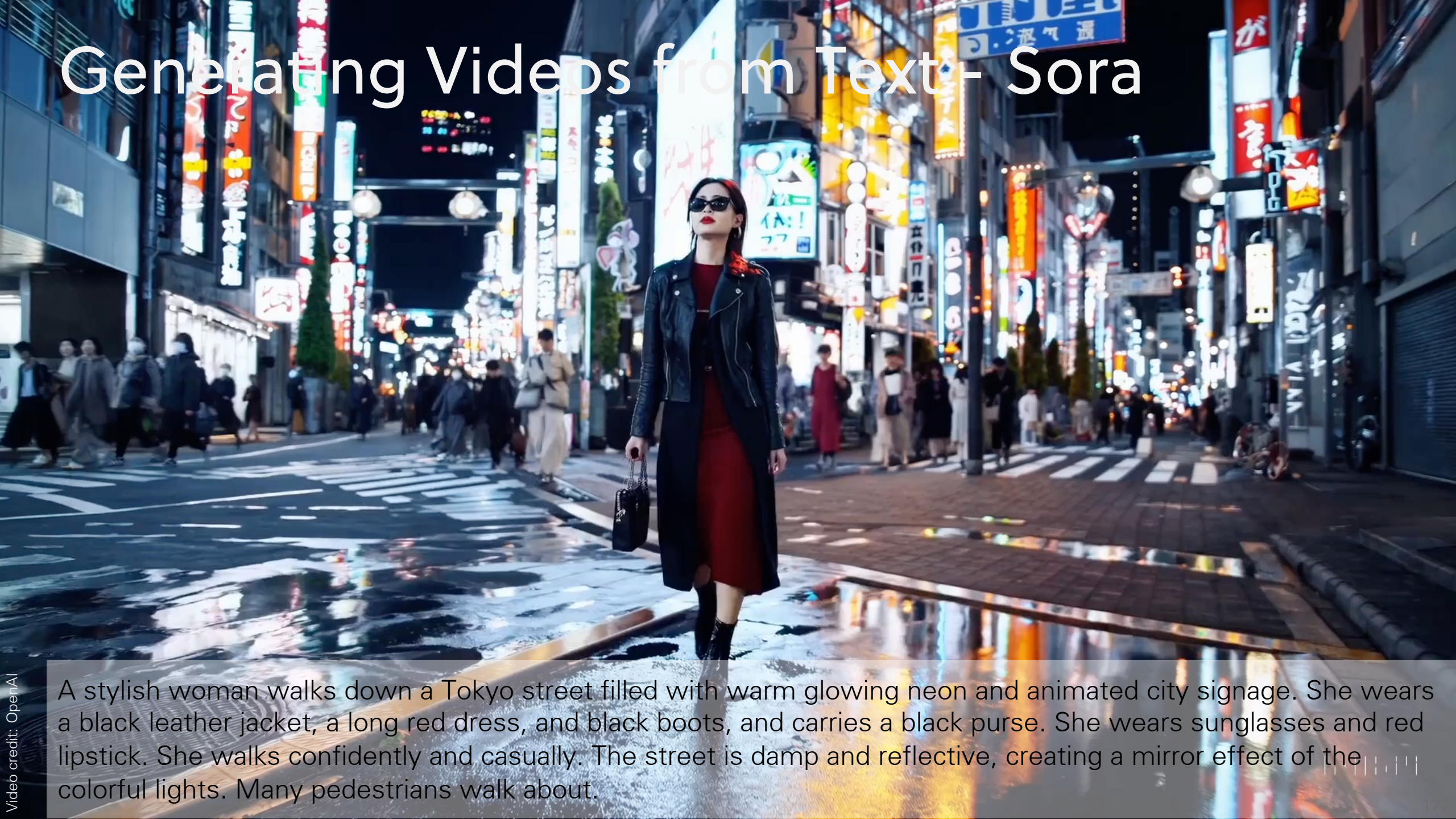


A British shorthair jumping over a couch



A swarm of bees flying around their hive

Generating Videos from Text - Sora



A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.

Generating Videos from Text - Sora



A young man at his 20s is sitting on a piece of cloud in the sky, reading a book.

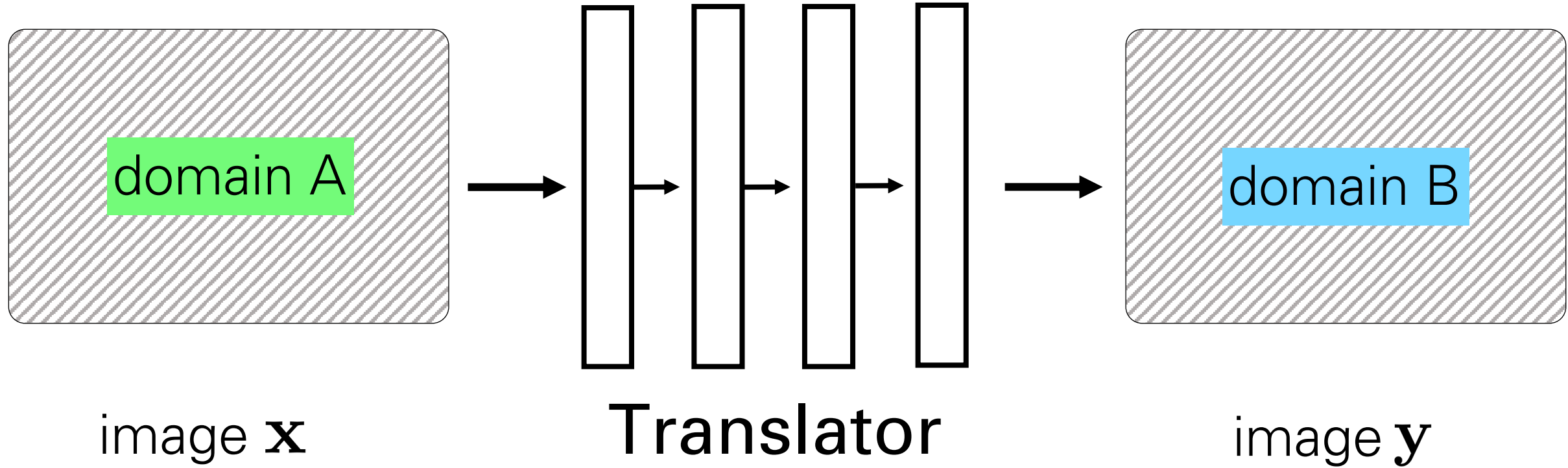
Generating Videos from Text - Sora



The camera directly faces colorful buildings in burano italy. An adorable dalmation looks through a window on a building on the ground floor. Many people are walking and cycling along the canal streets in front of the buildings.

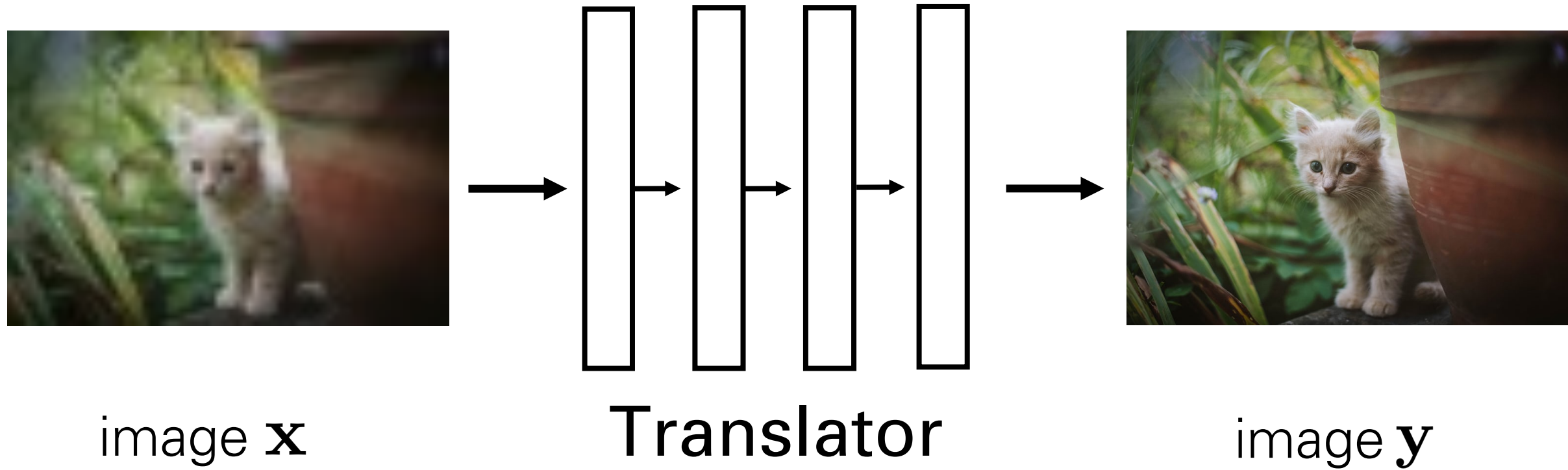
Applications in Computational Photography

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

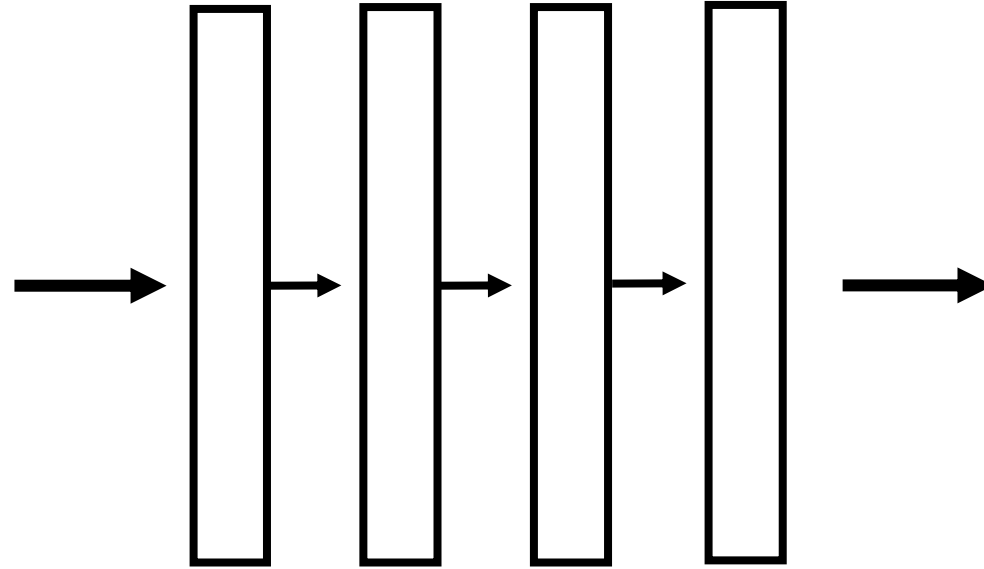


- **Input:** low-res image
- **Output:** high-res image

Colorization



image x



Translator



image y

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

BW → Color

Input

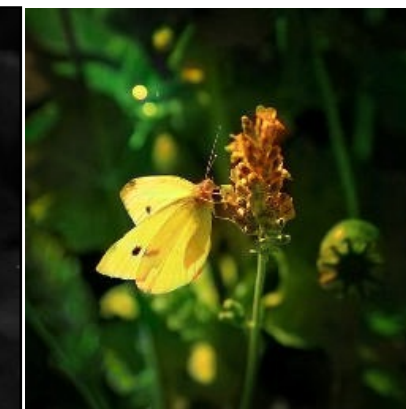
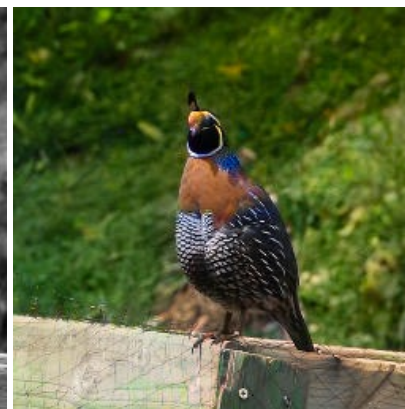
Output

Input

Output

Input

Output

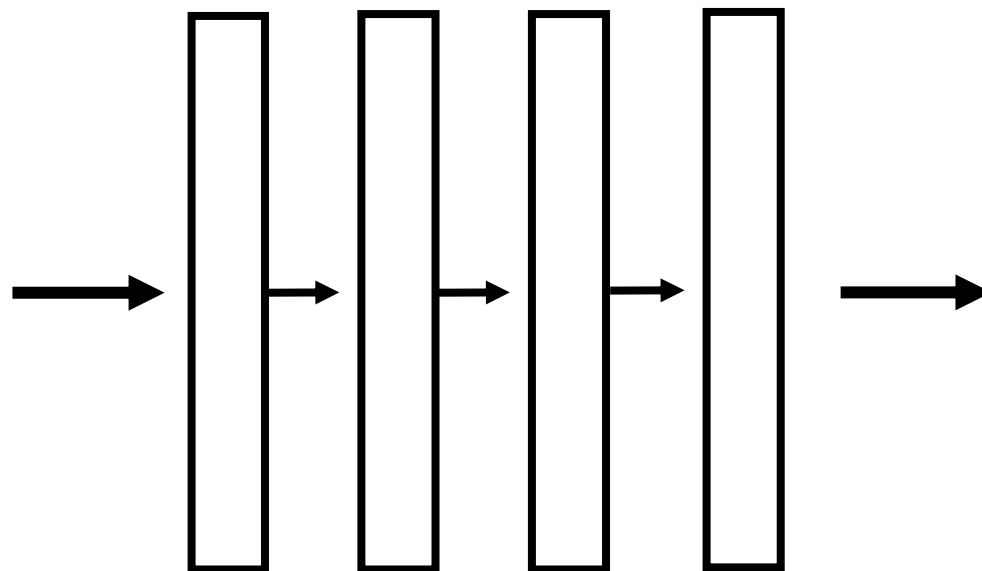


Data from [Russakovsky et al. 2015]

Day to Night



image x



Translator



image y

- Input: day image
- Output: night image

Photo Style Transfer

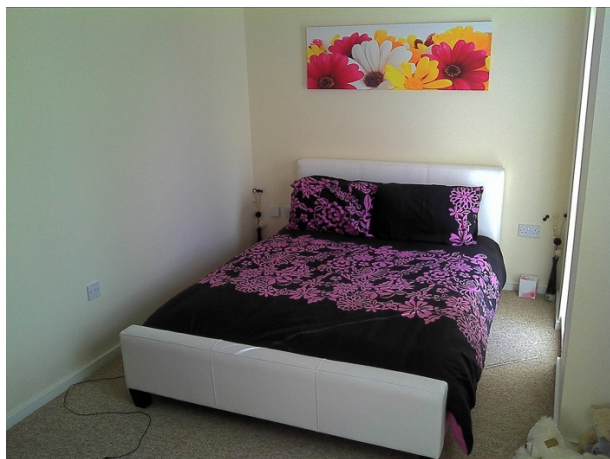


image x

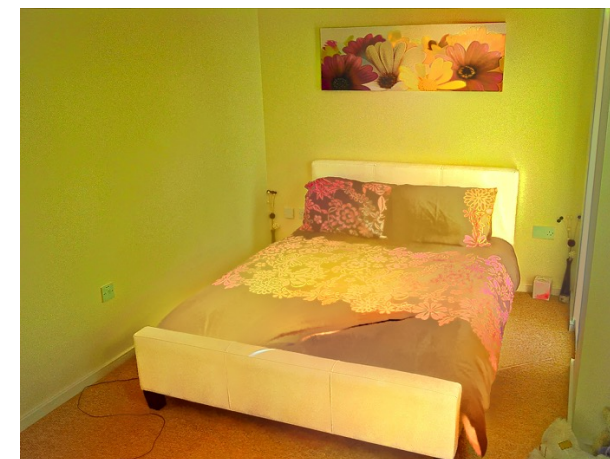
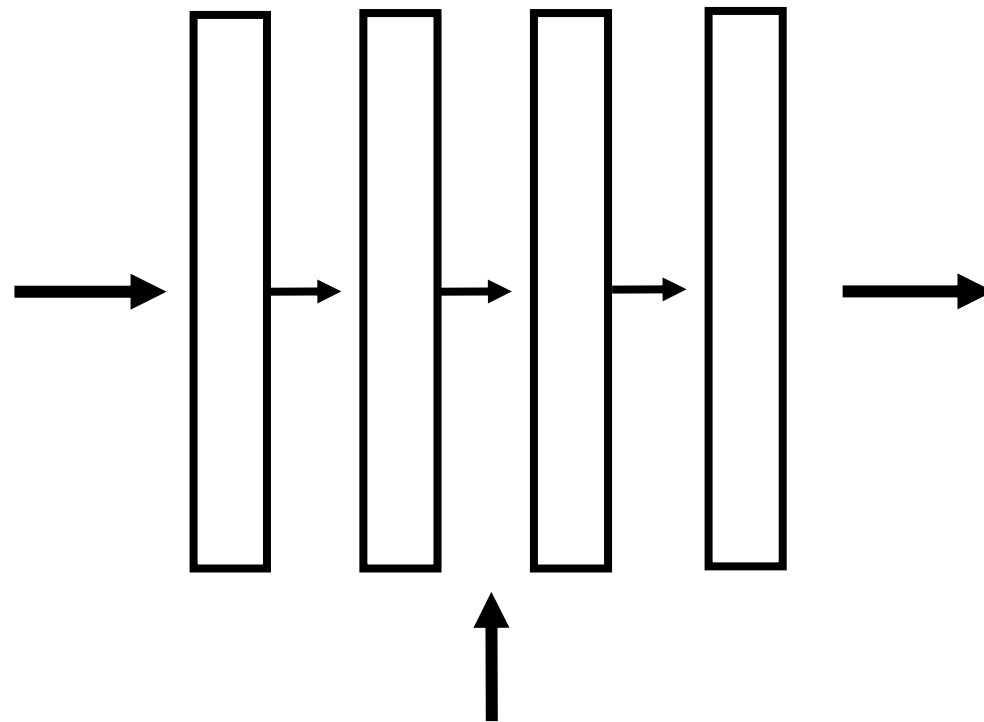
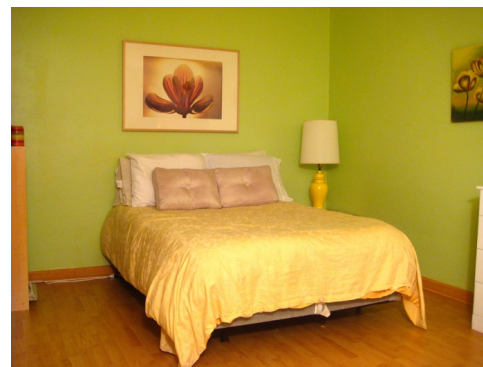


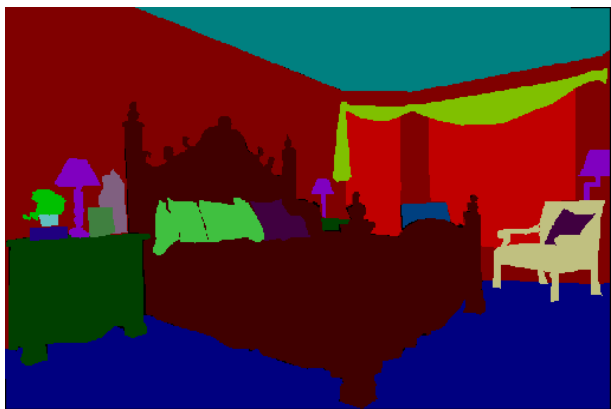
image y

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

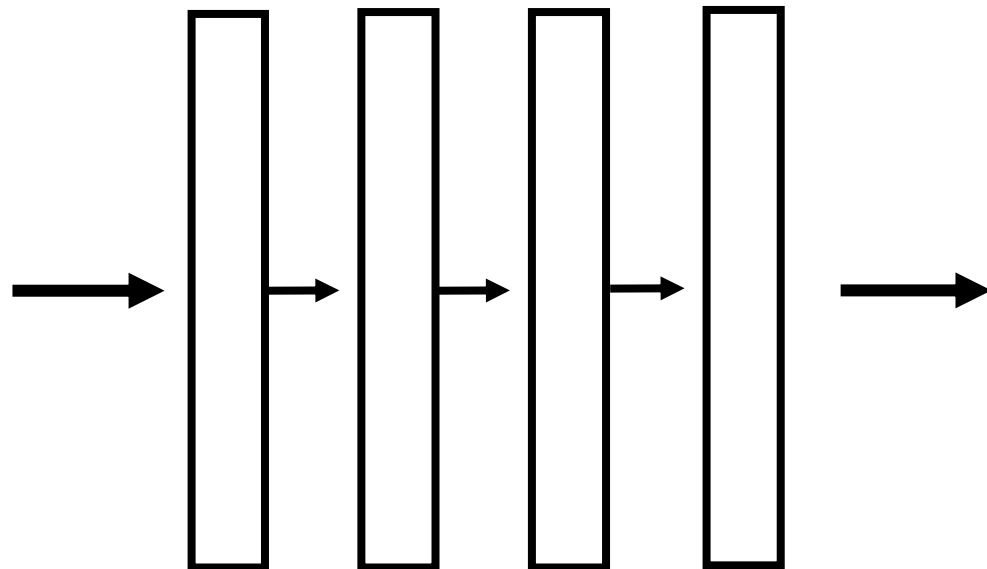


Target style
image t

Semantic Image Synthesis



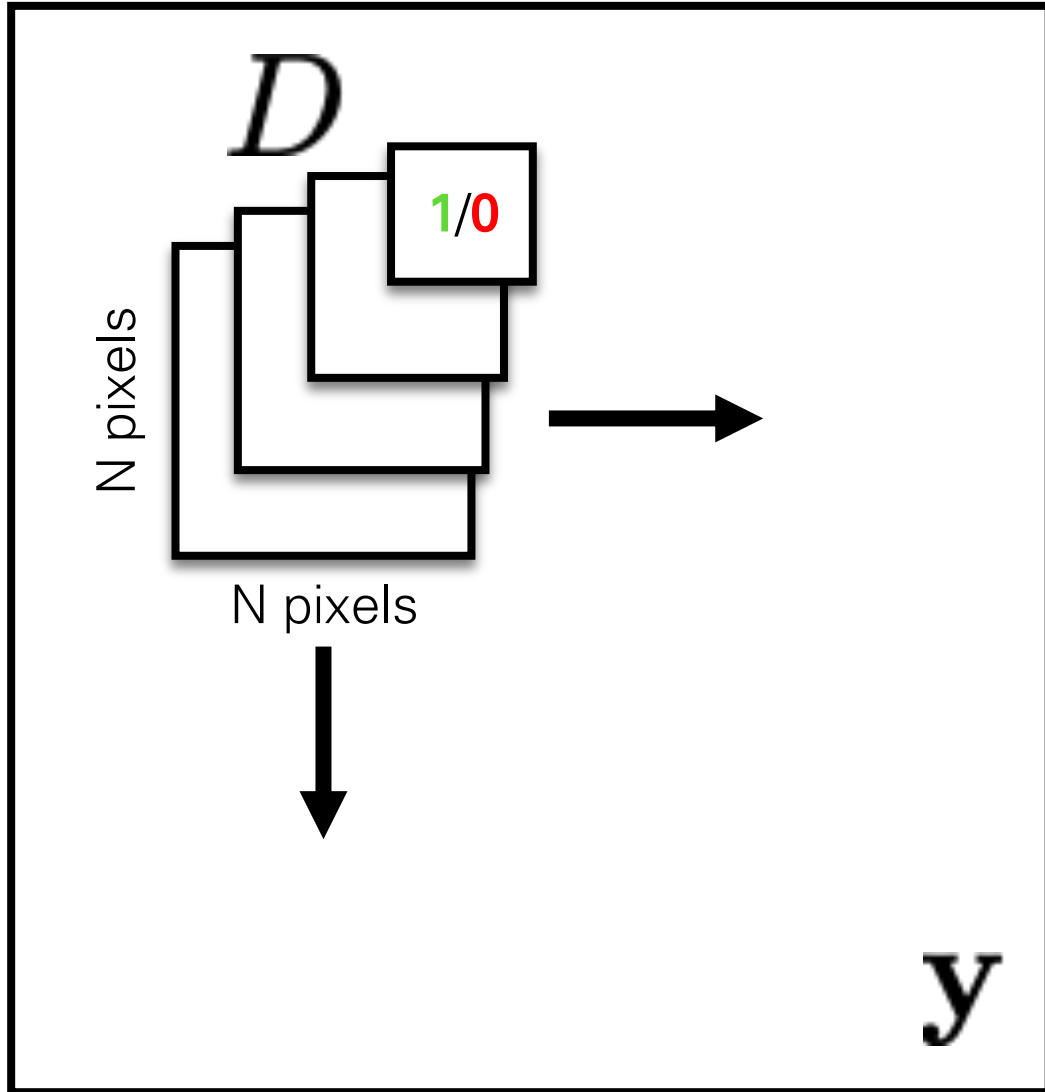
semantic
layout \mathbf{x}



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

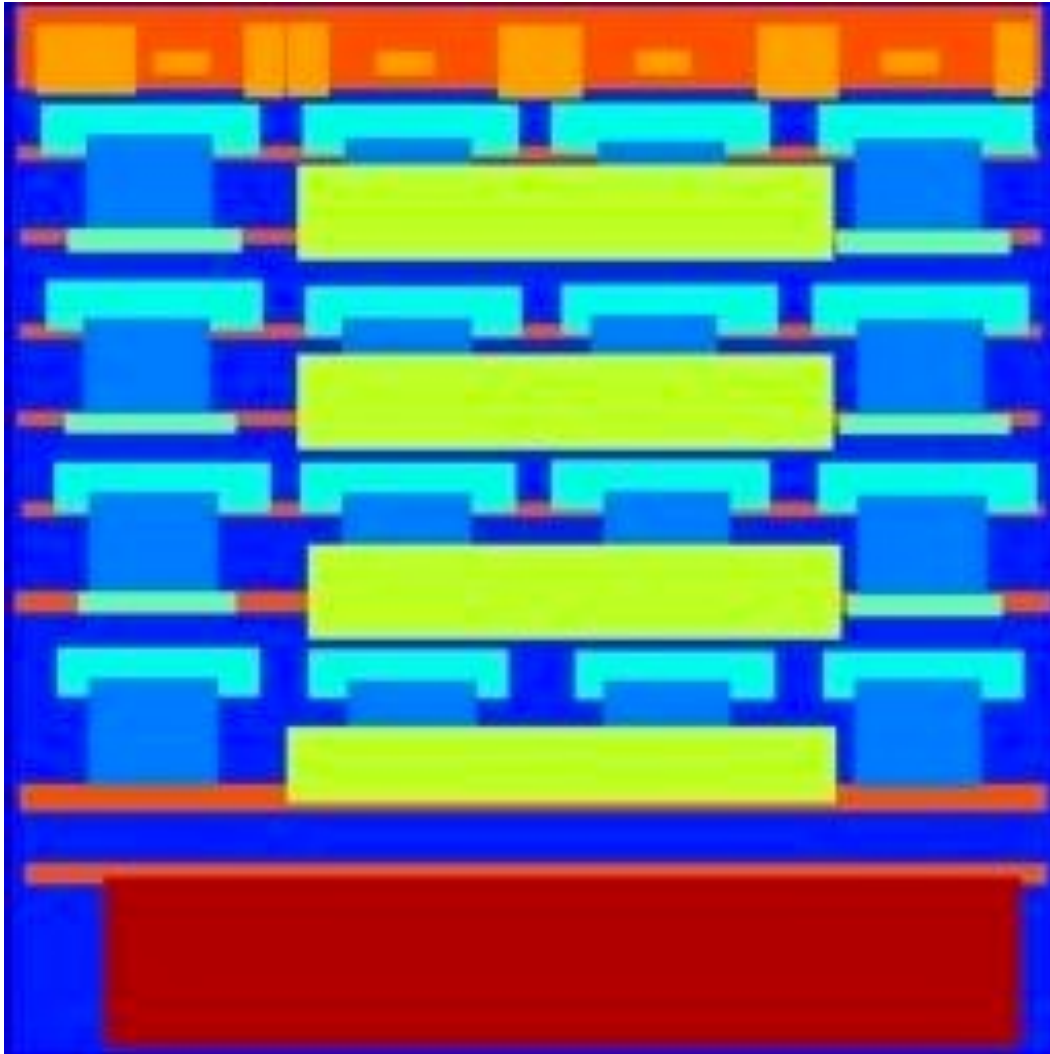
[Li & Wand 2016]

[Shrivastava et al. 2017]

[Isola et al. 2017]

Labels \rightarrow Facades

Input

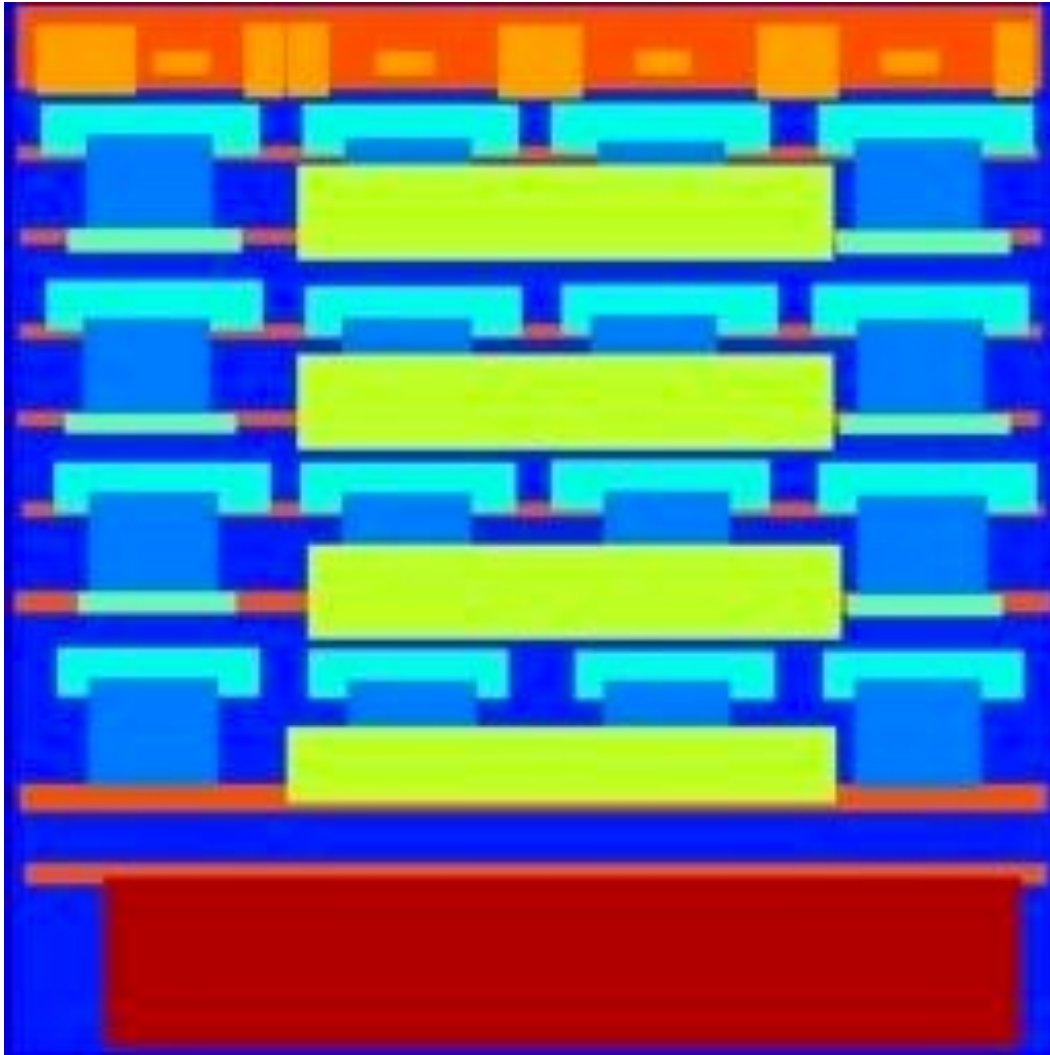


1x1 Discriminator



Labels \rightarrow Facades

Input

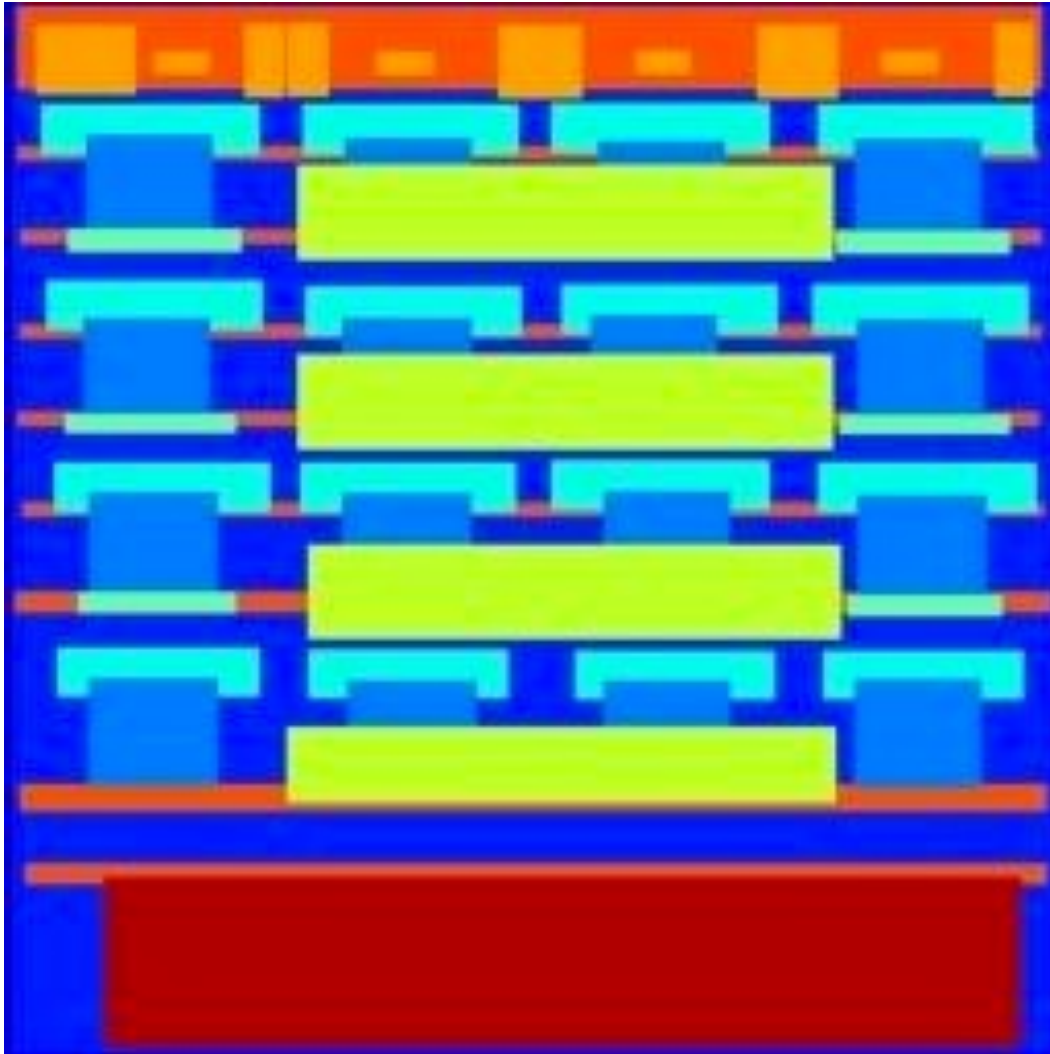


16x16 Discriminator



Labels → Facades

Input

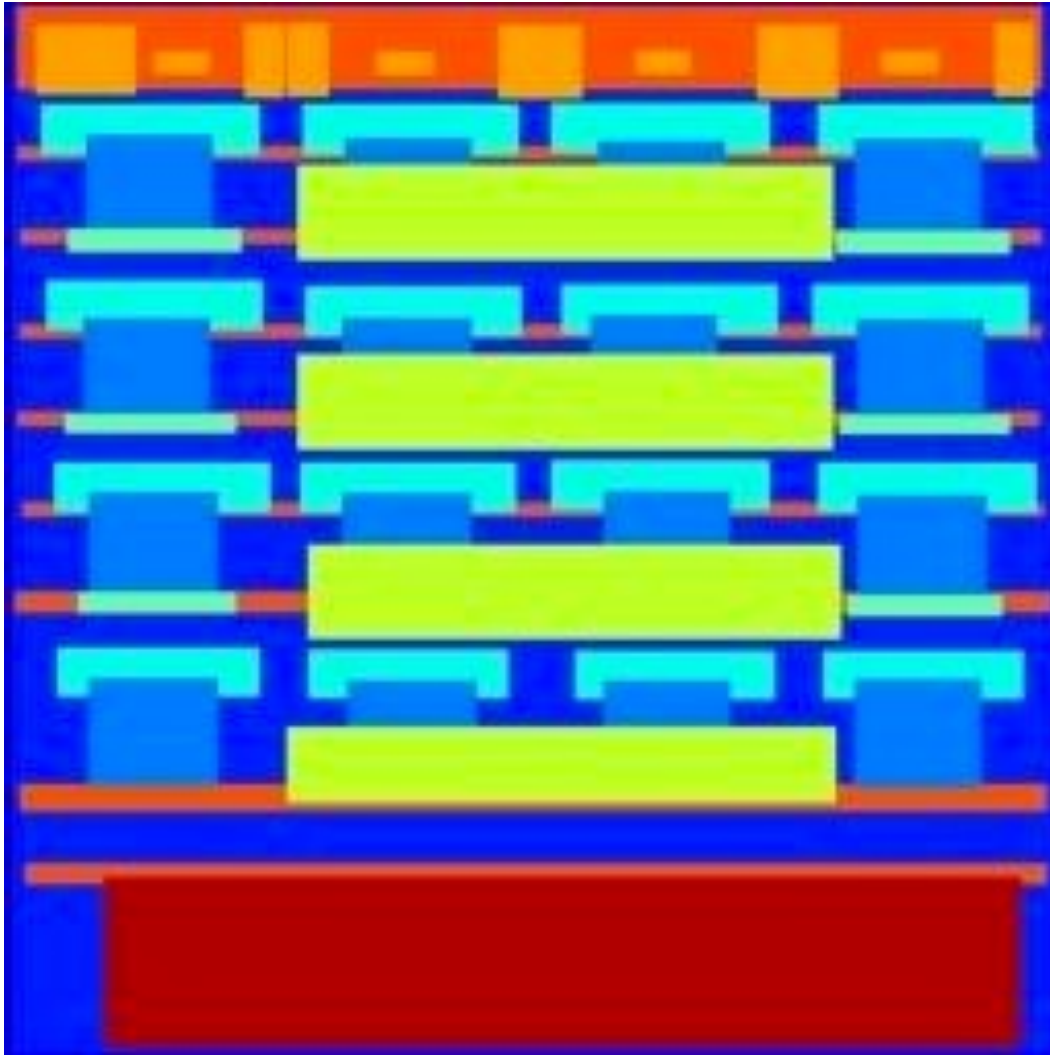


70x70 Discriminator



Labels \rightarrow Facades

Input



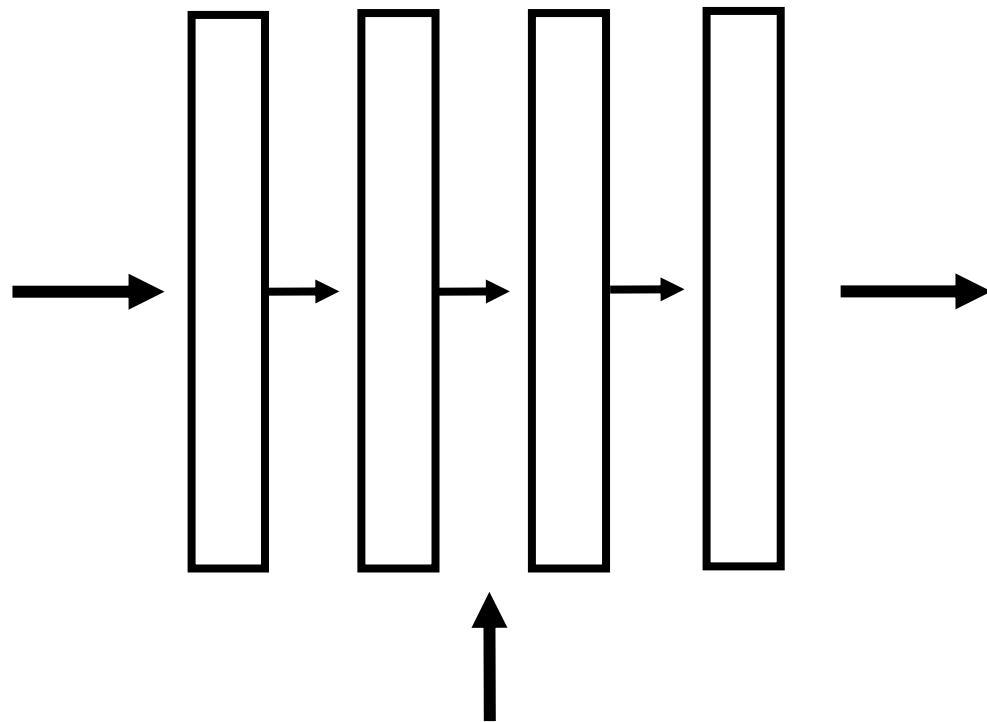
Full image Discriminator



Semantic Image Synthesis



semantic layout \mathbf{x}



synthesized image \mathbf{y}

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

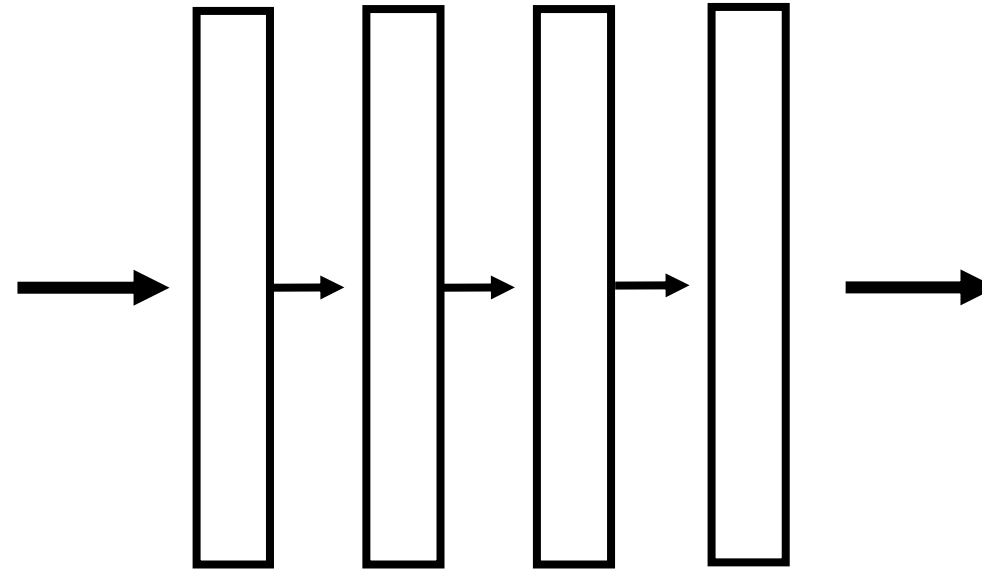


Target style image \mathbf{t}

Semantic Image Editing



image \mathbf{x}



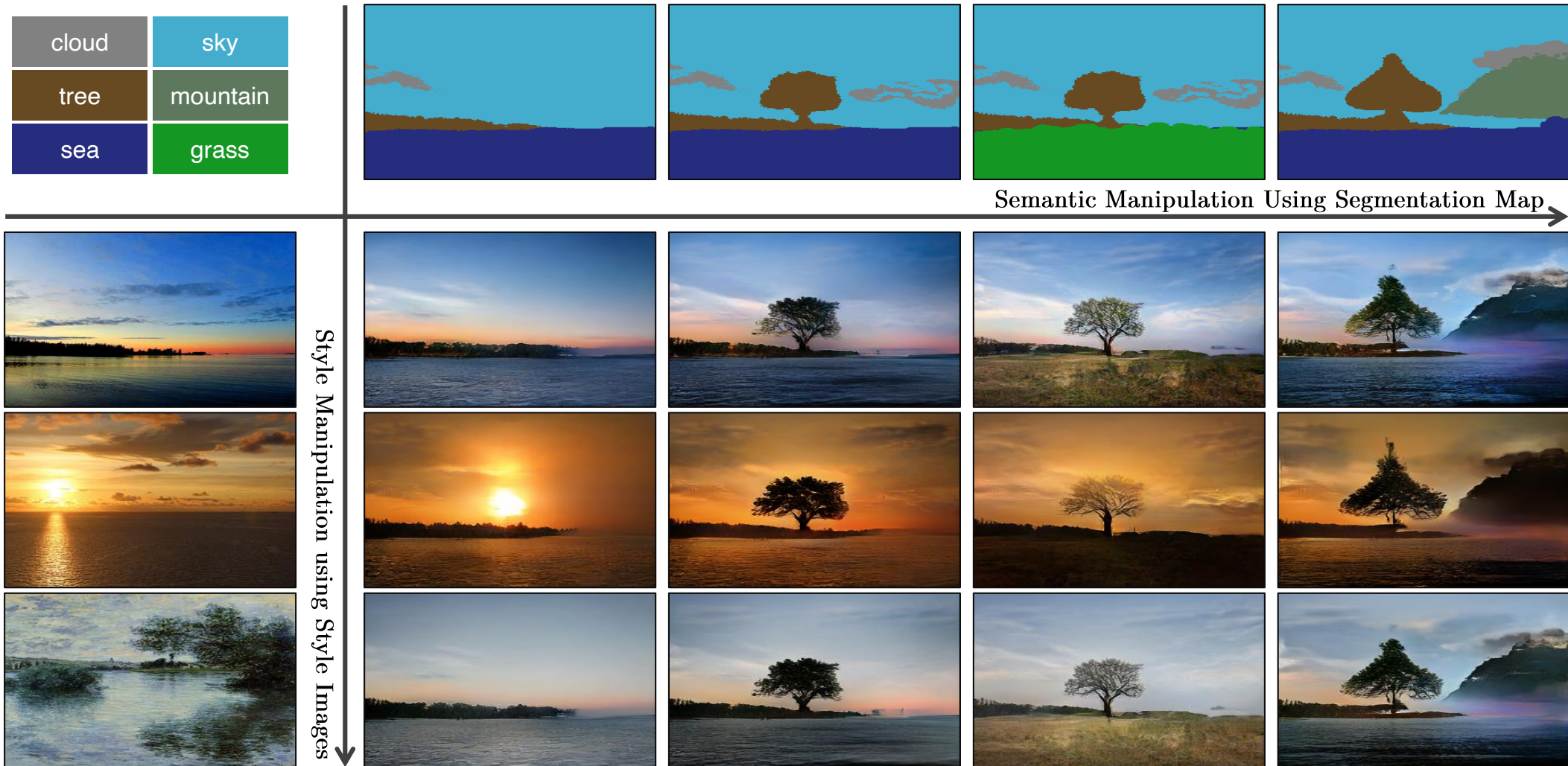
manipulated
image \mathbf{y}

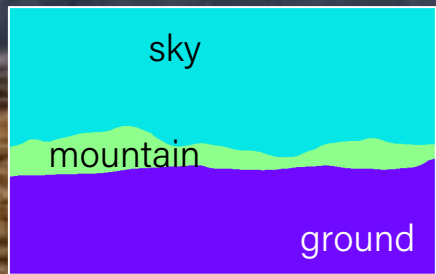
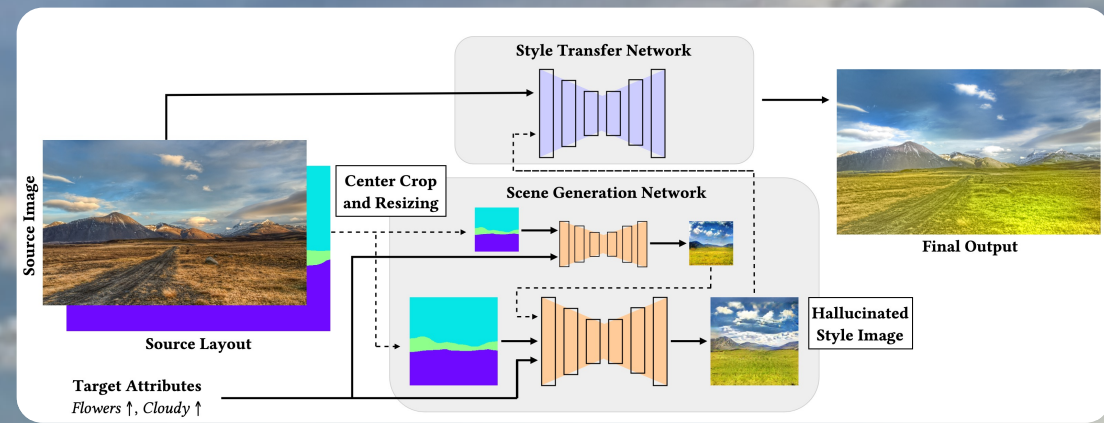
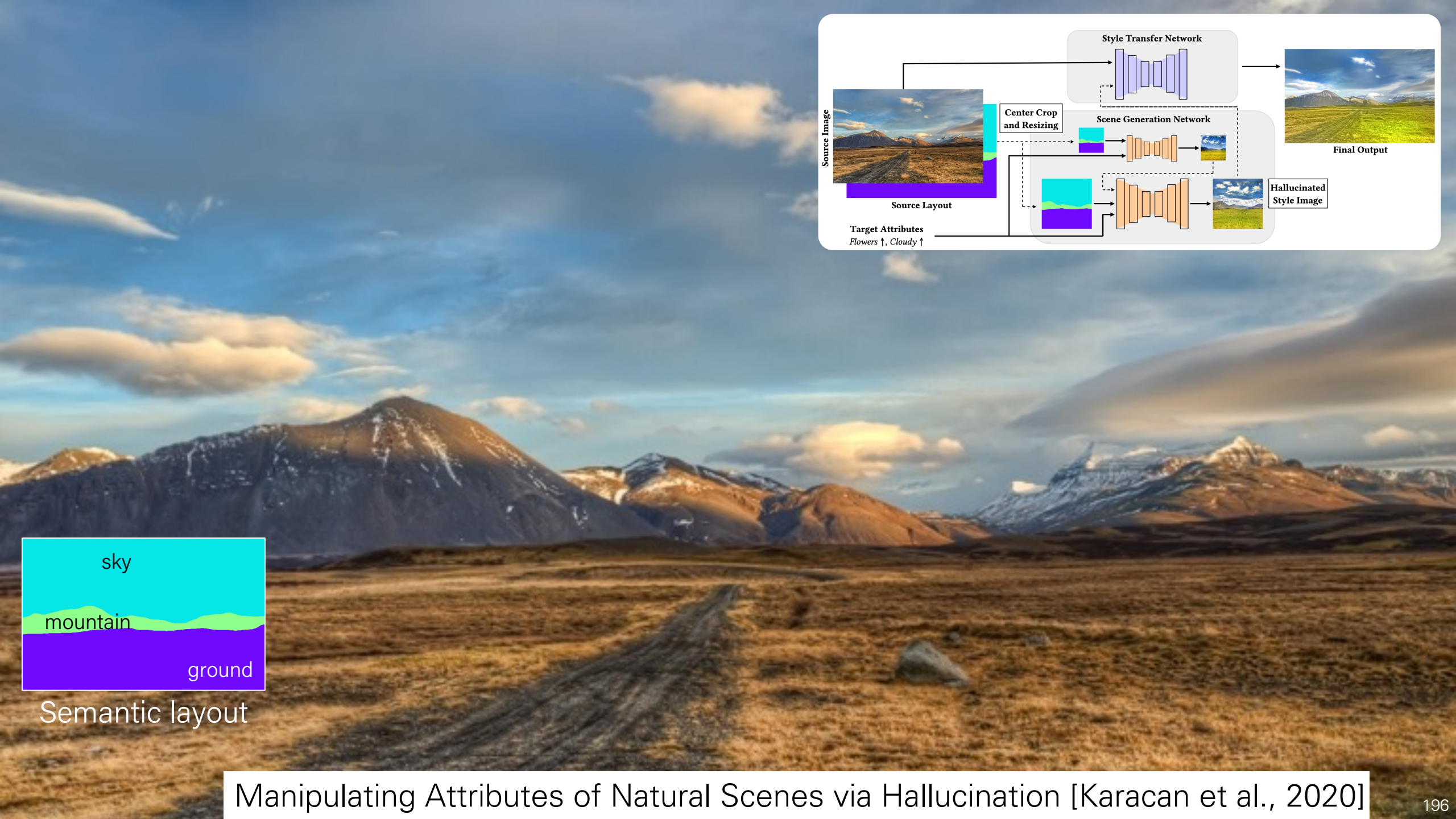
- **Input:** input image
(+ semantic layout)
+ target attribute
- Output:** manipulated image

“more flowers”
“more cloudy”
Target scene
attribute(s) \mathbf{t}

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

- Image generation conditioned on semantic layouts



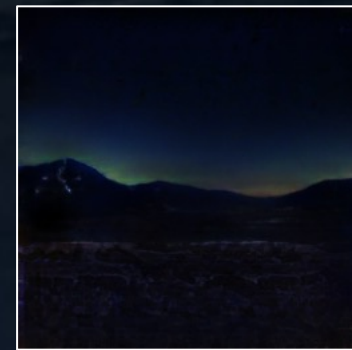


Semantic layout



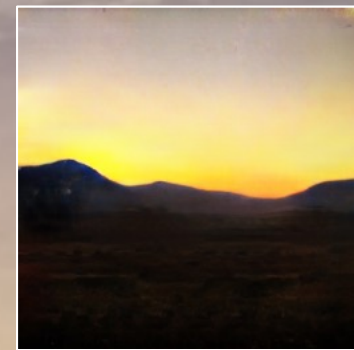
Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

night

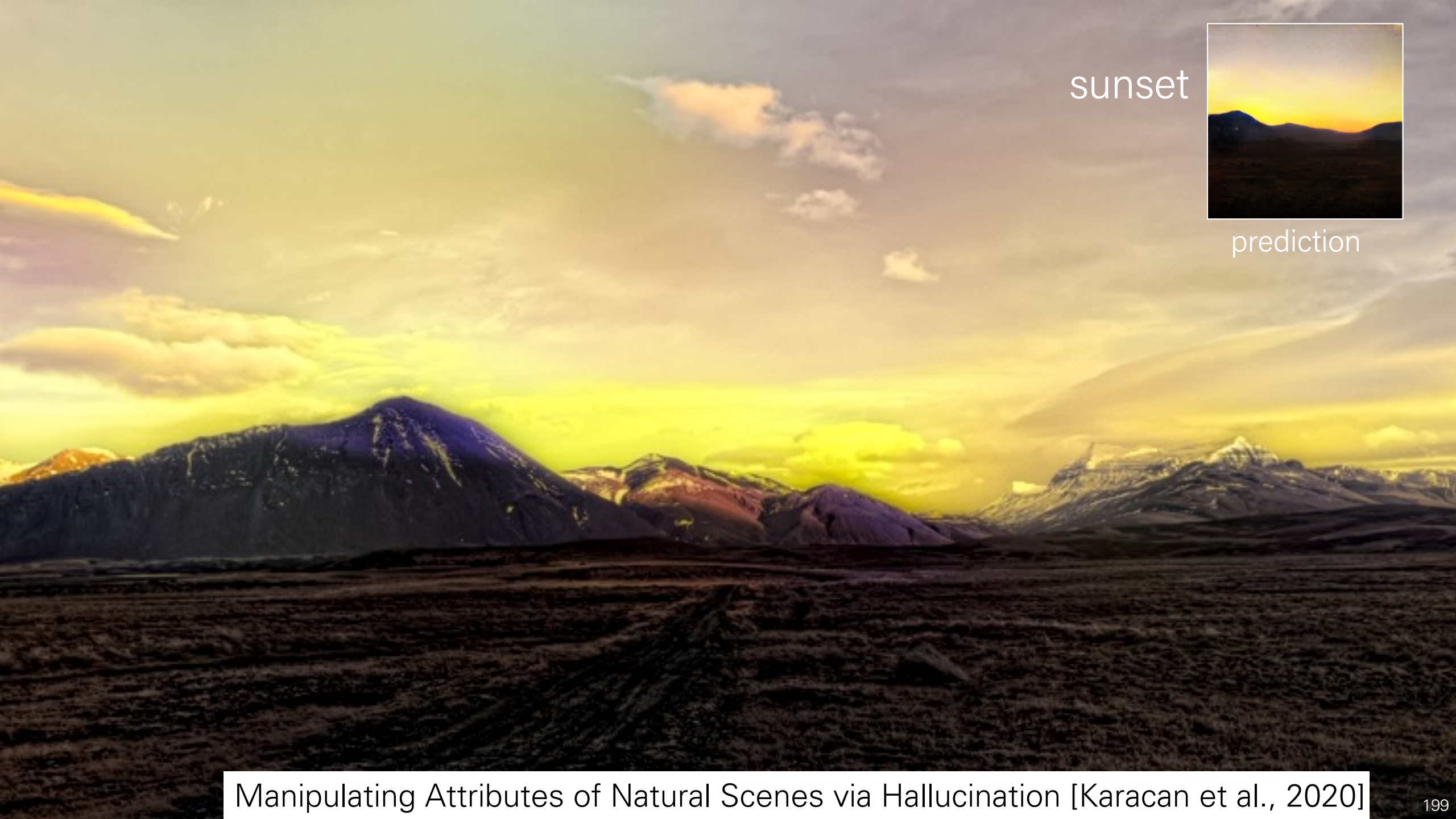


prediction

sunset



prediction



snow



prediction



winter



prediction



Spring and clouds



prediction



Moist, rain and fog

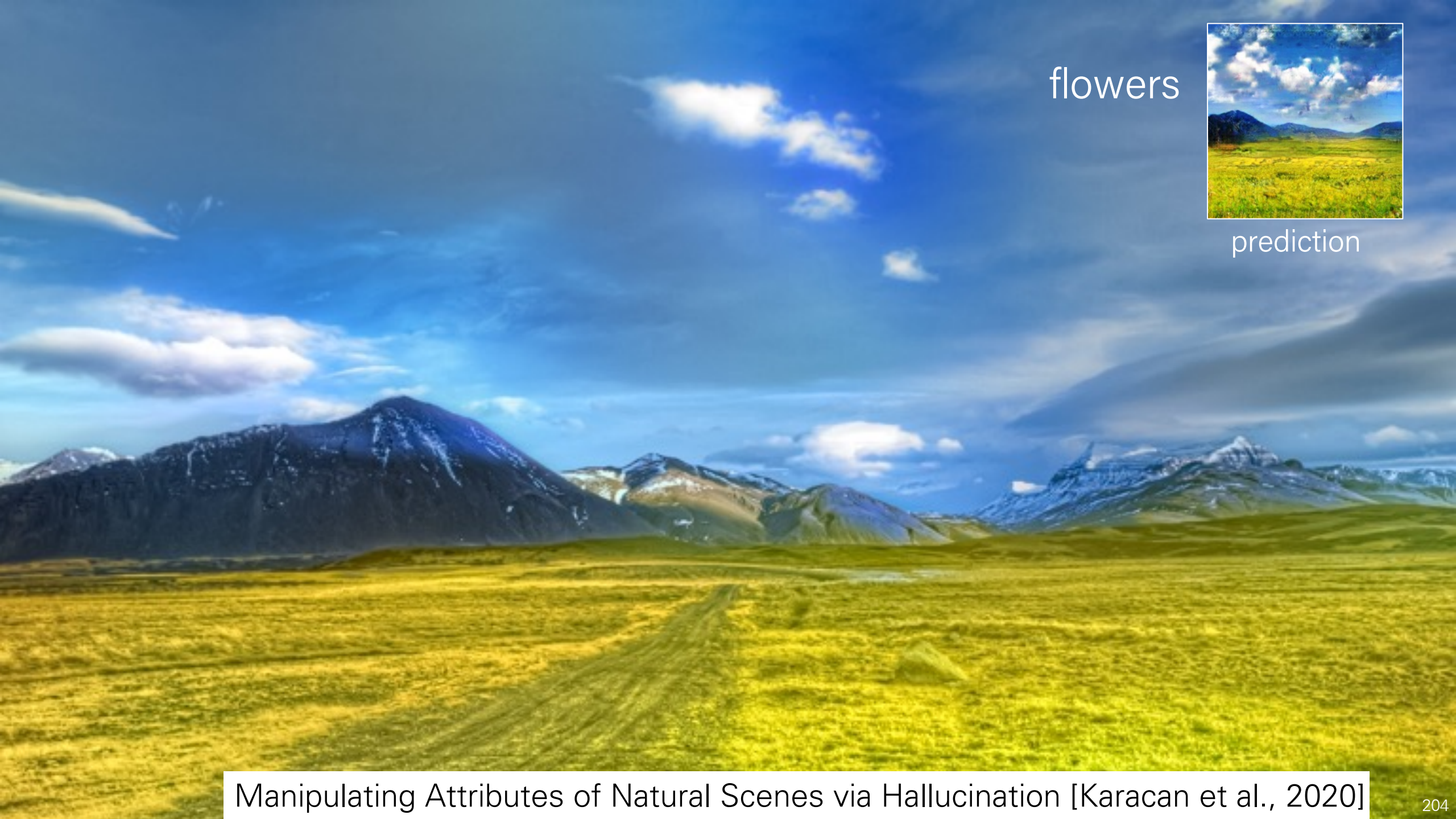


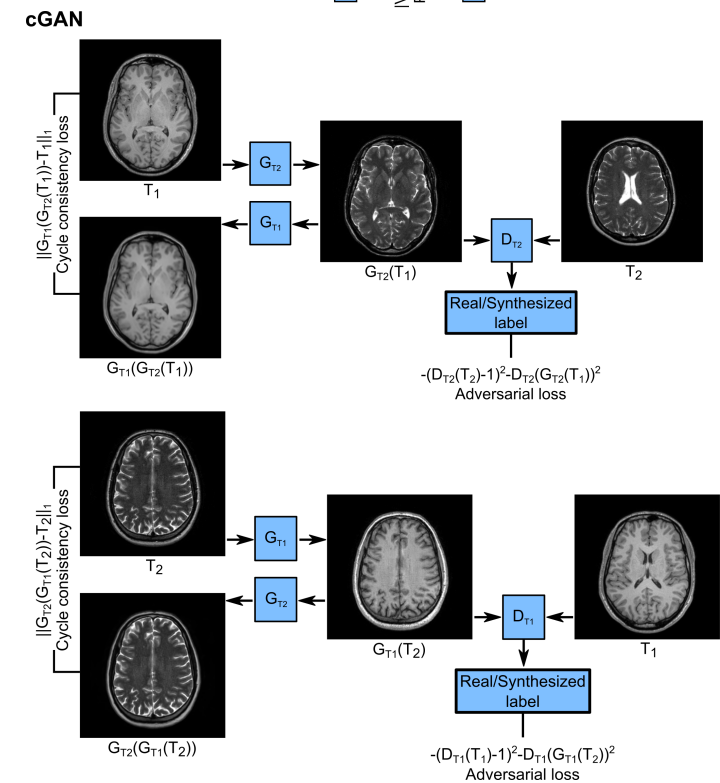
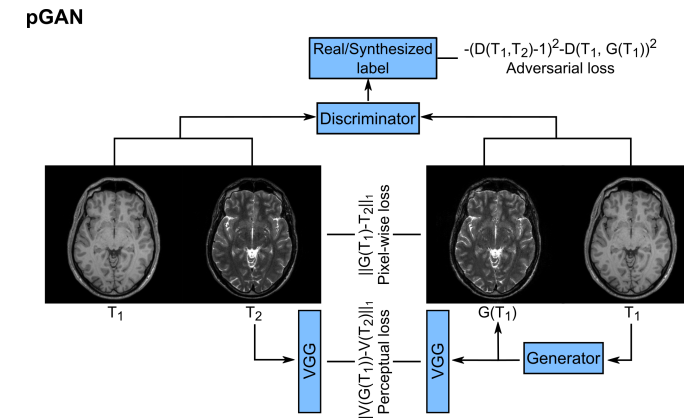
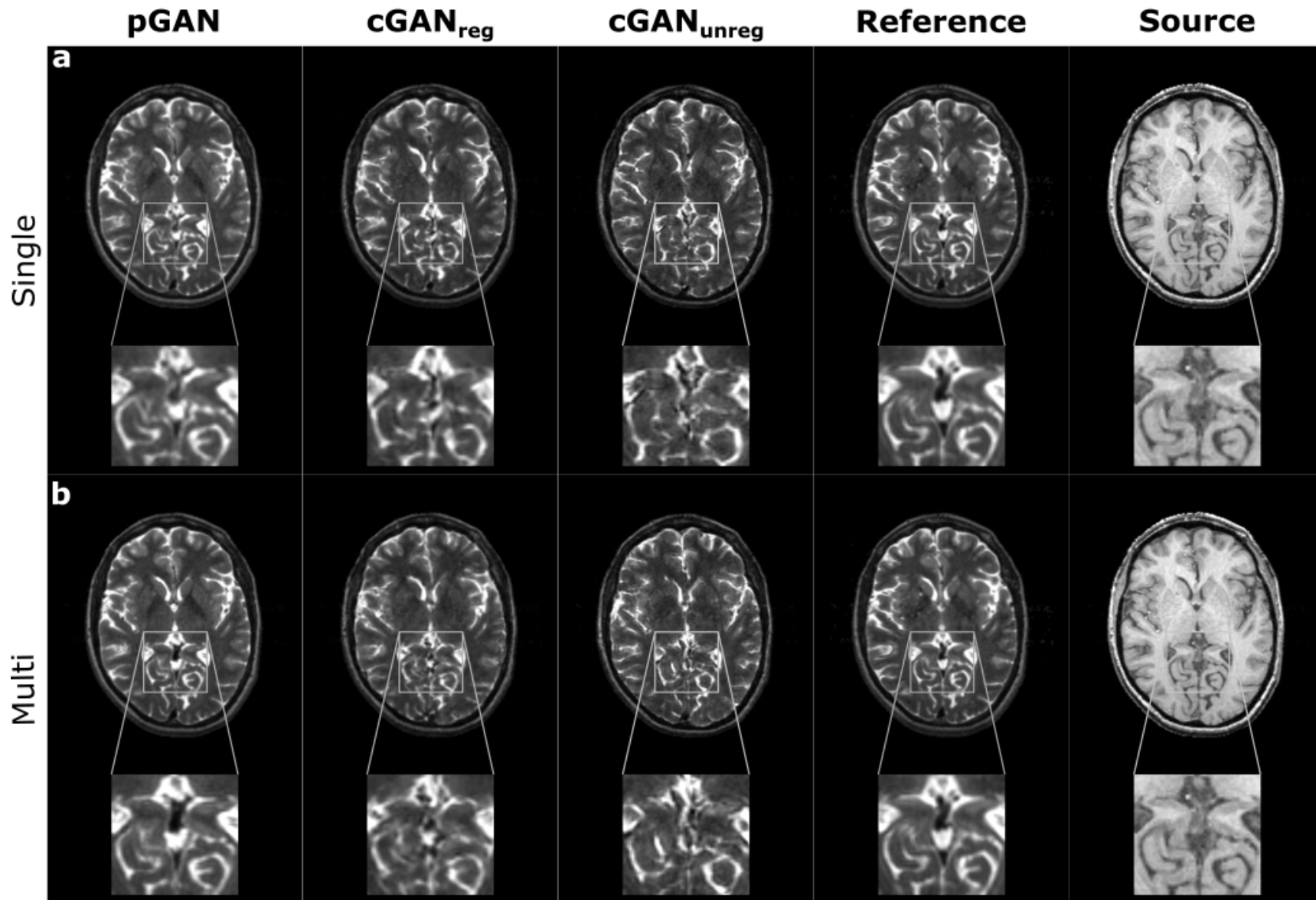
prediction

flowers

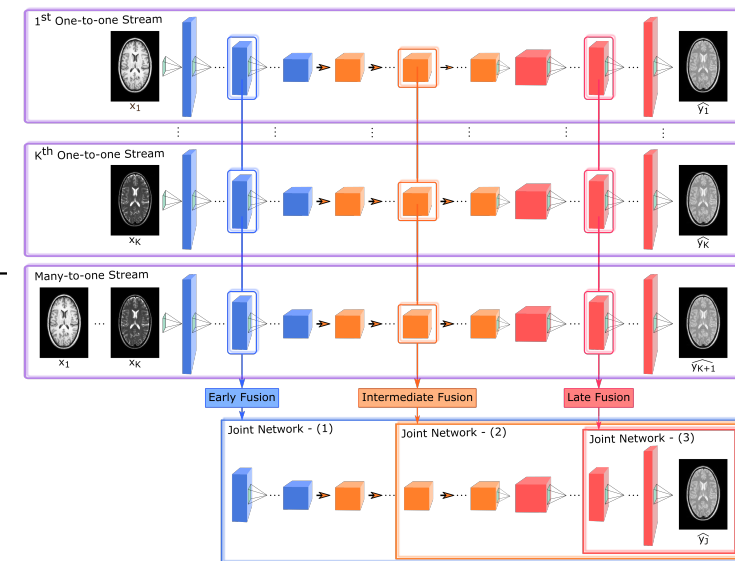
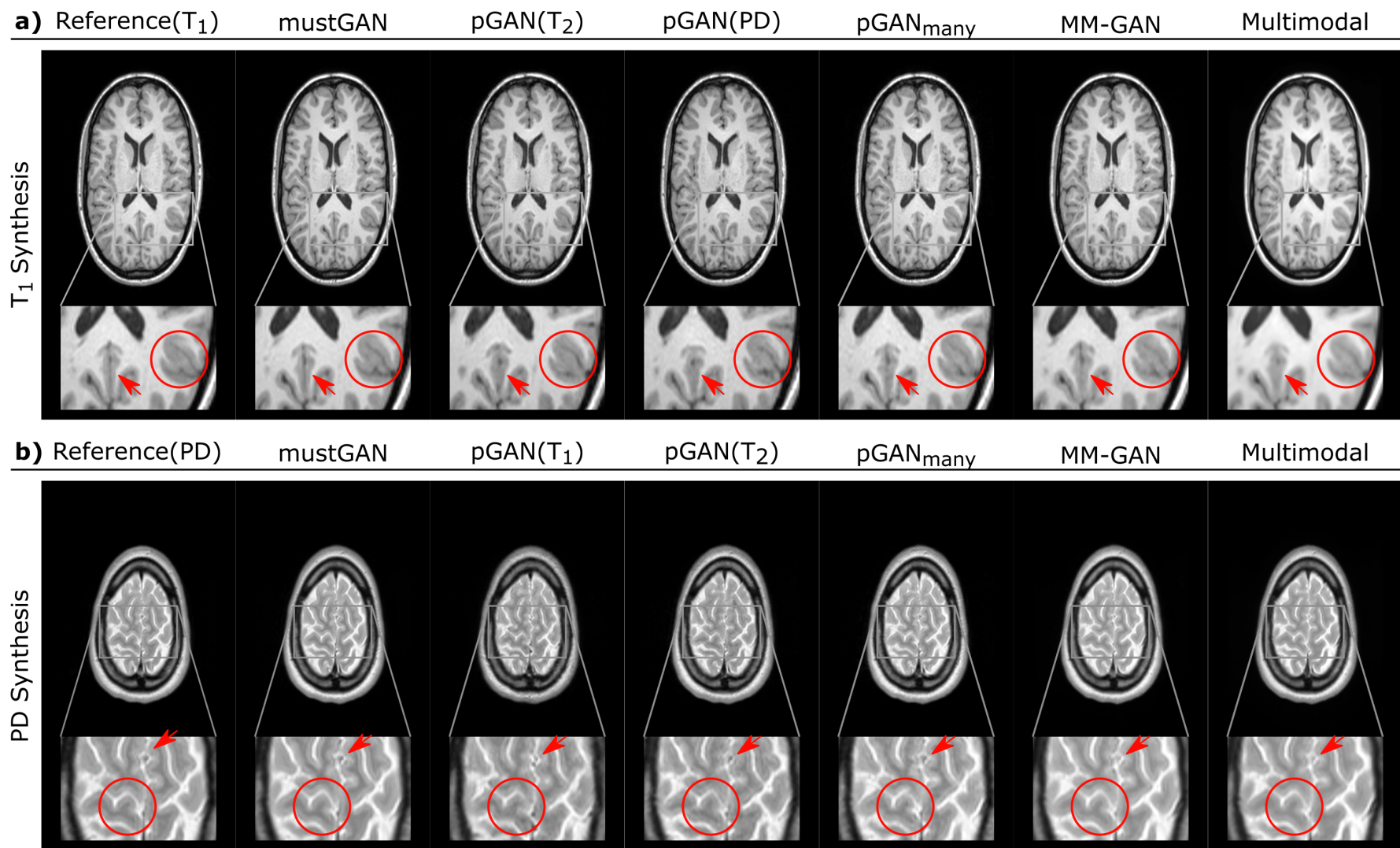


prediction





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



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

Single Image Super-Resolution

bicubic



SRResNet



SRGAN

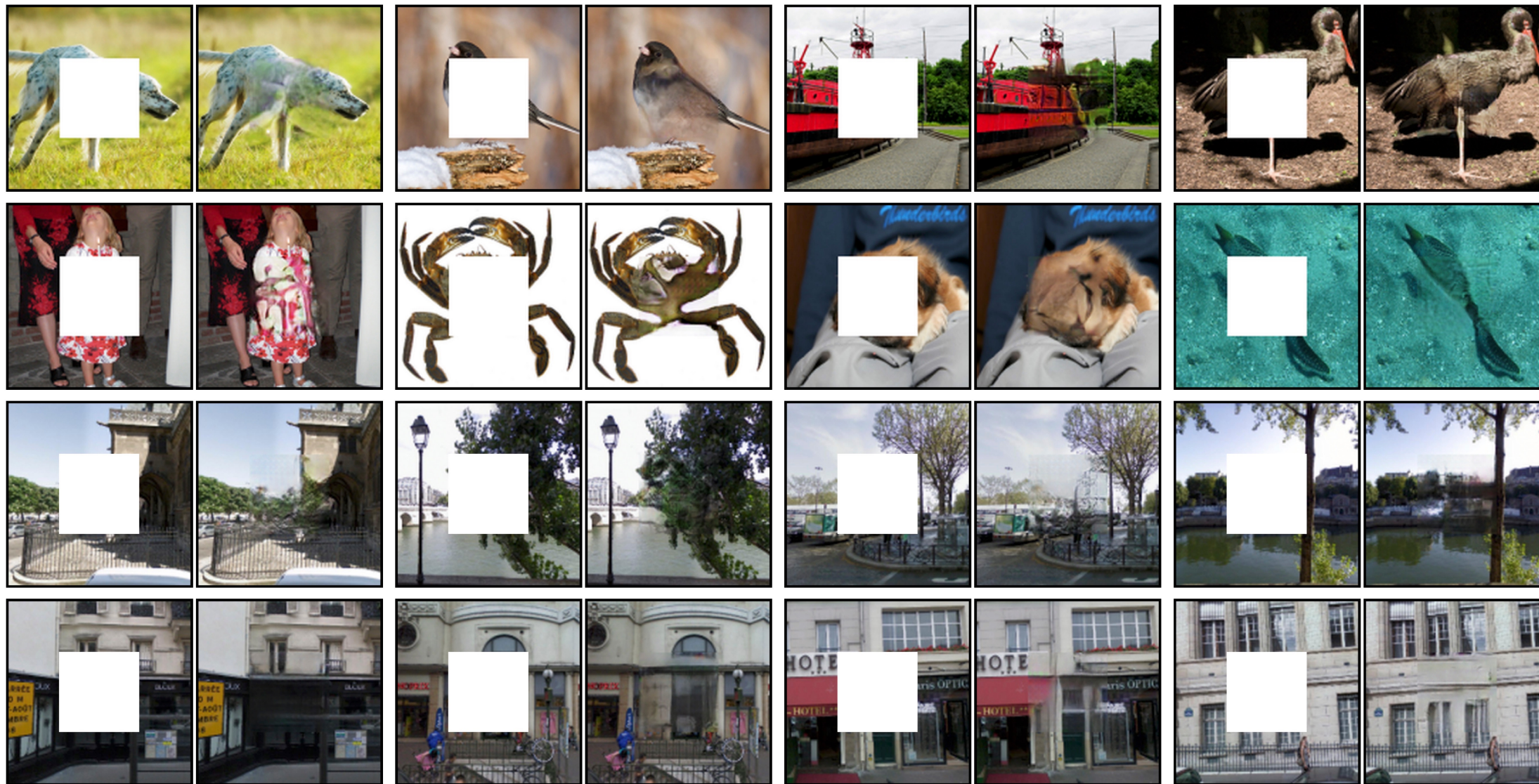


original



- Key idea: Combine content loss with adversarial loss

Image Inpainting (Pathak et al., 2016)



- Key idea: Combine content loss with adversarial loss

Image Deblurring

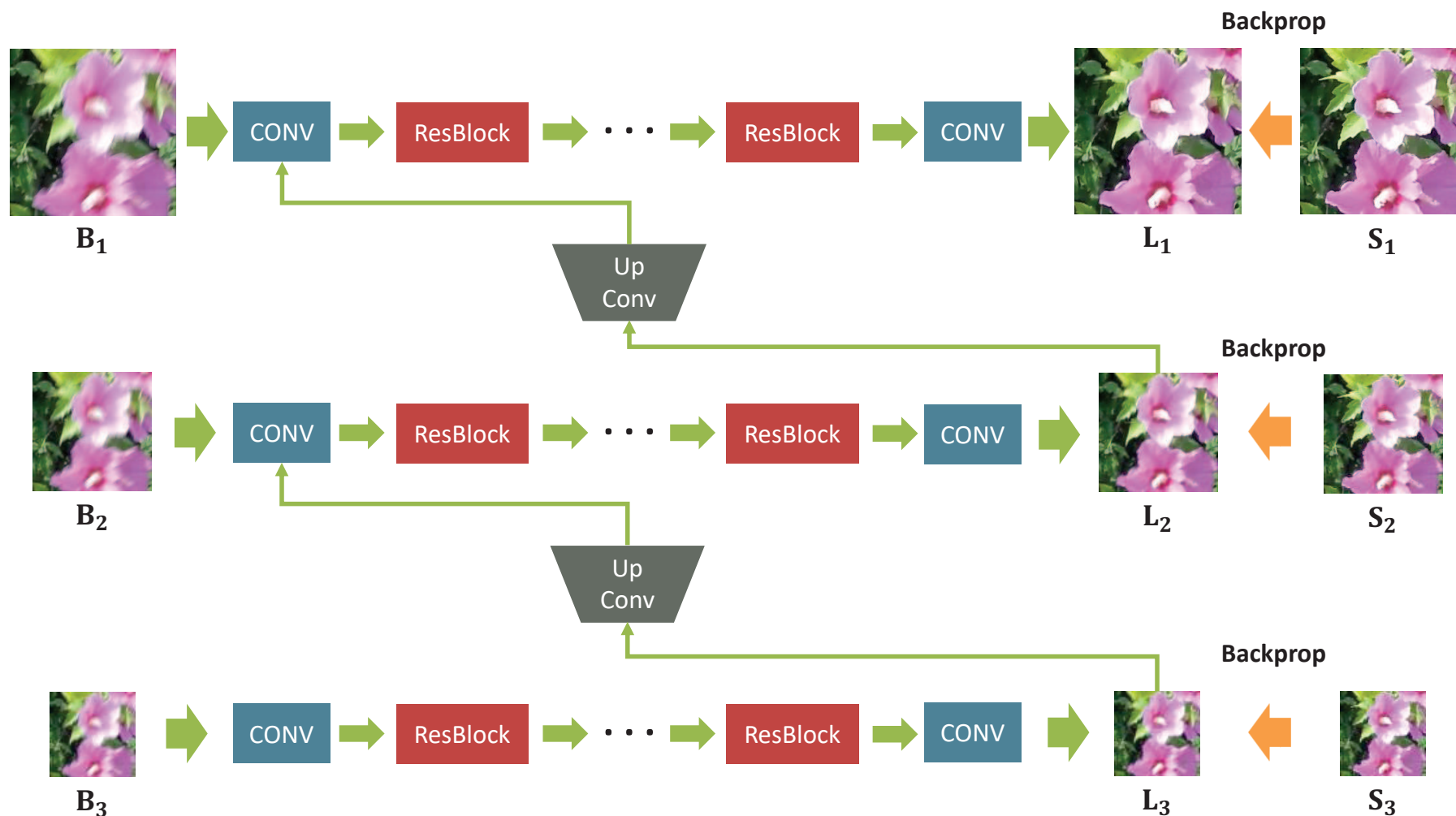
- non-uniform motion blur from a single blurry image.
- **Key idea:** Use multiscale CNNs to restore sharp images in an end-to-end manner

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

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

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

Image Deblurring



- Coarser scale features aid finer scale image deblurring

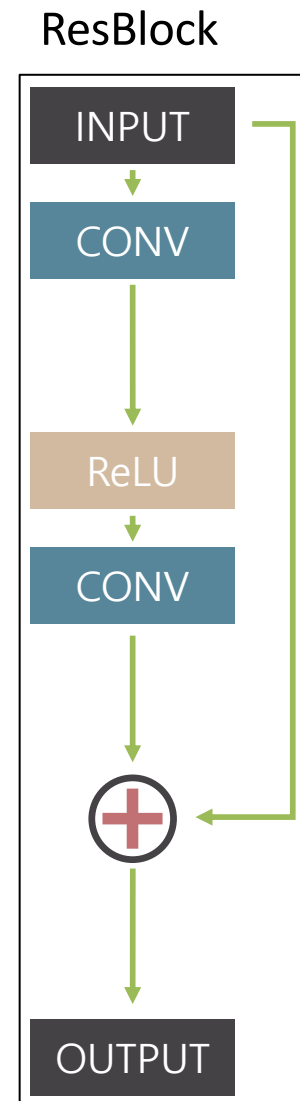
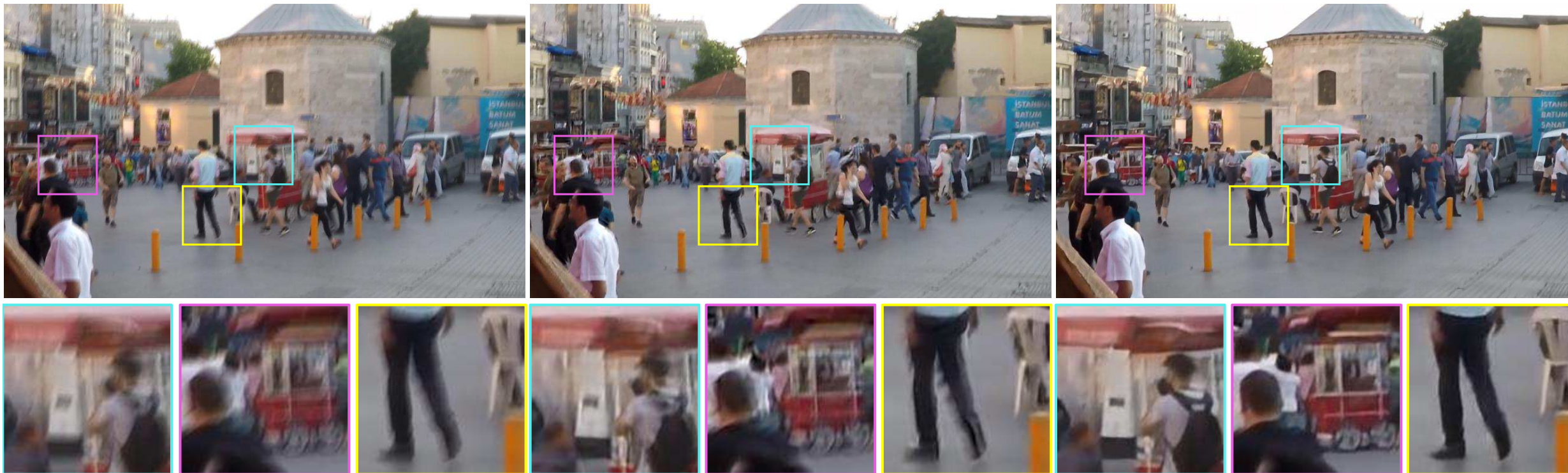


Image Deblurring



Blurred images

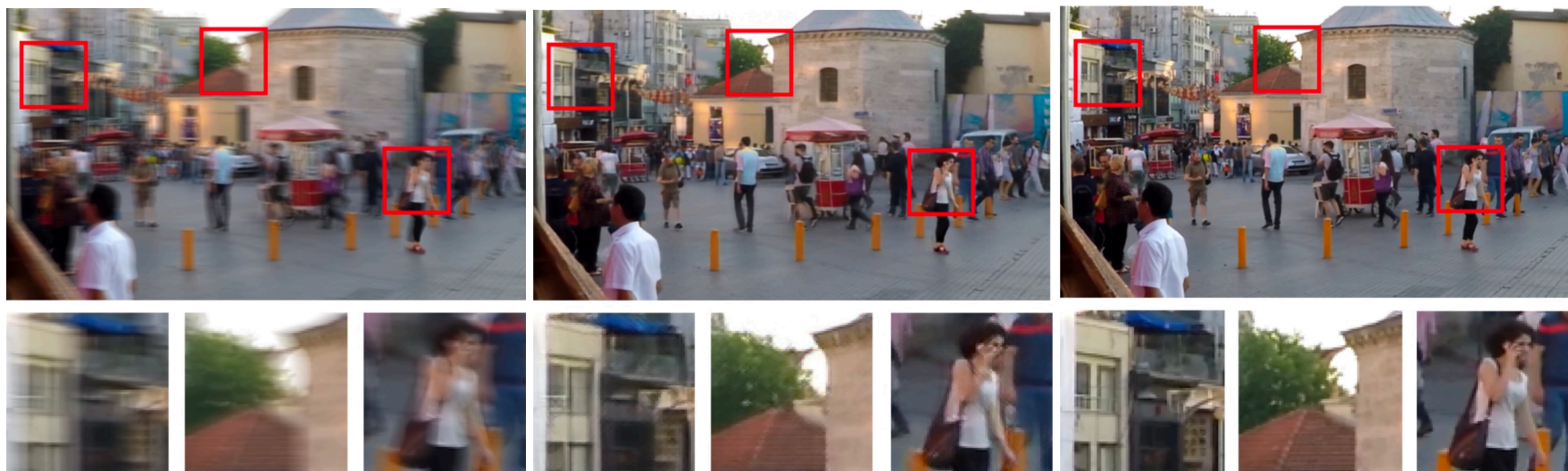
Sun et al., CVPR 2015

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

$$\mathcal{L} = \underbrace{\mathcal{L}_{GAN}}_{adv\ loss} + \underbrace{\lambda \cdot \mathcal{L}_X}_{content\ loss} \quad \underbrace{\hspace{10em}}_{total\ loss}$$
$$\mathcal{L}_{GAN} = \sum_{n=1}^N -D_{\theta_D}(G_{\theta_G}(I^B)) \quad \mathcal{L}_X = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^S)_{x,y} - \phi_{i,j}(G_{\theta_G}(I^B))_{x,y})^2$$



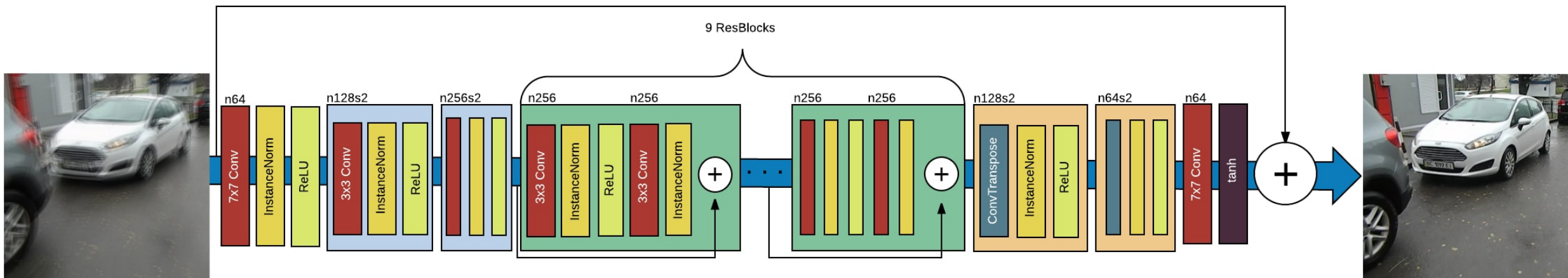
Blurred images

Groundtruth

Predicted

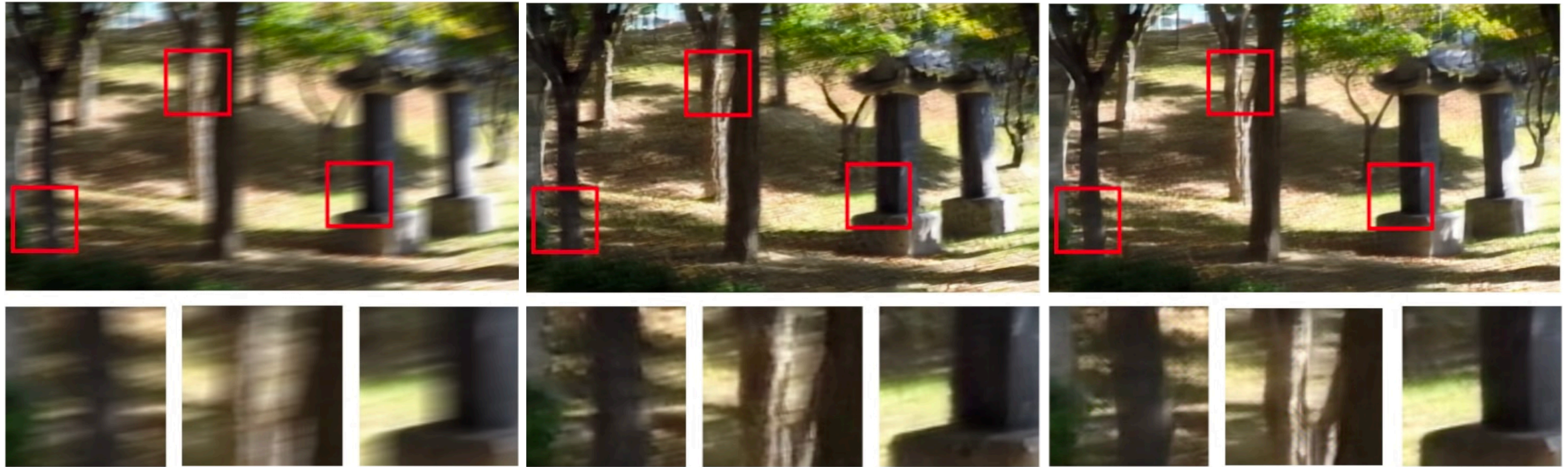
Image Deblurring

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



213

Image Deblurring



Blurred images

Nah et al., CVPR 2017

Kupyn et al., CVPR 2018

Image Deblurring



Image Deblurring

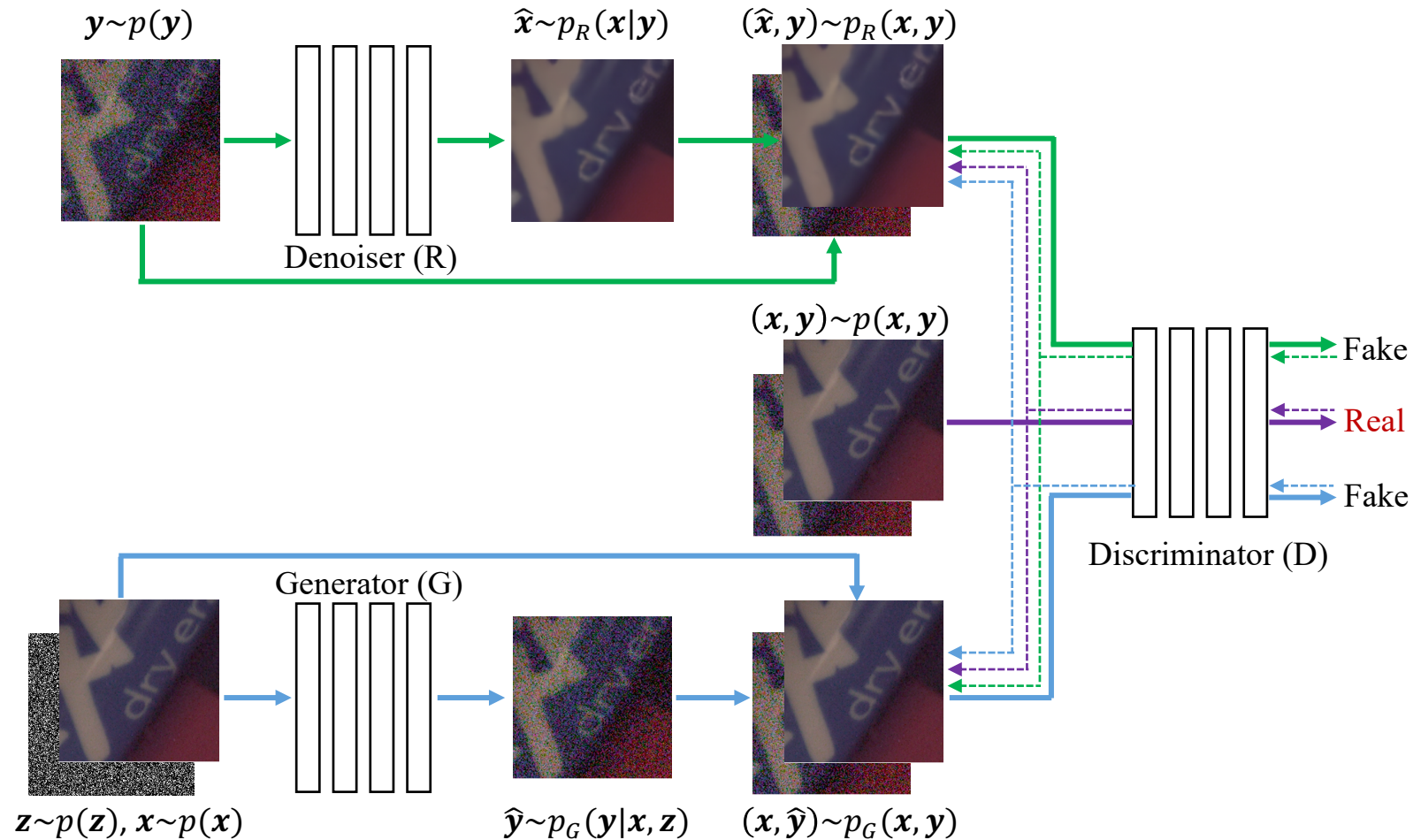


Blurred images

Nah et al., CVPR 2017

Kupyn et al., CVPR 2018

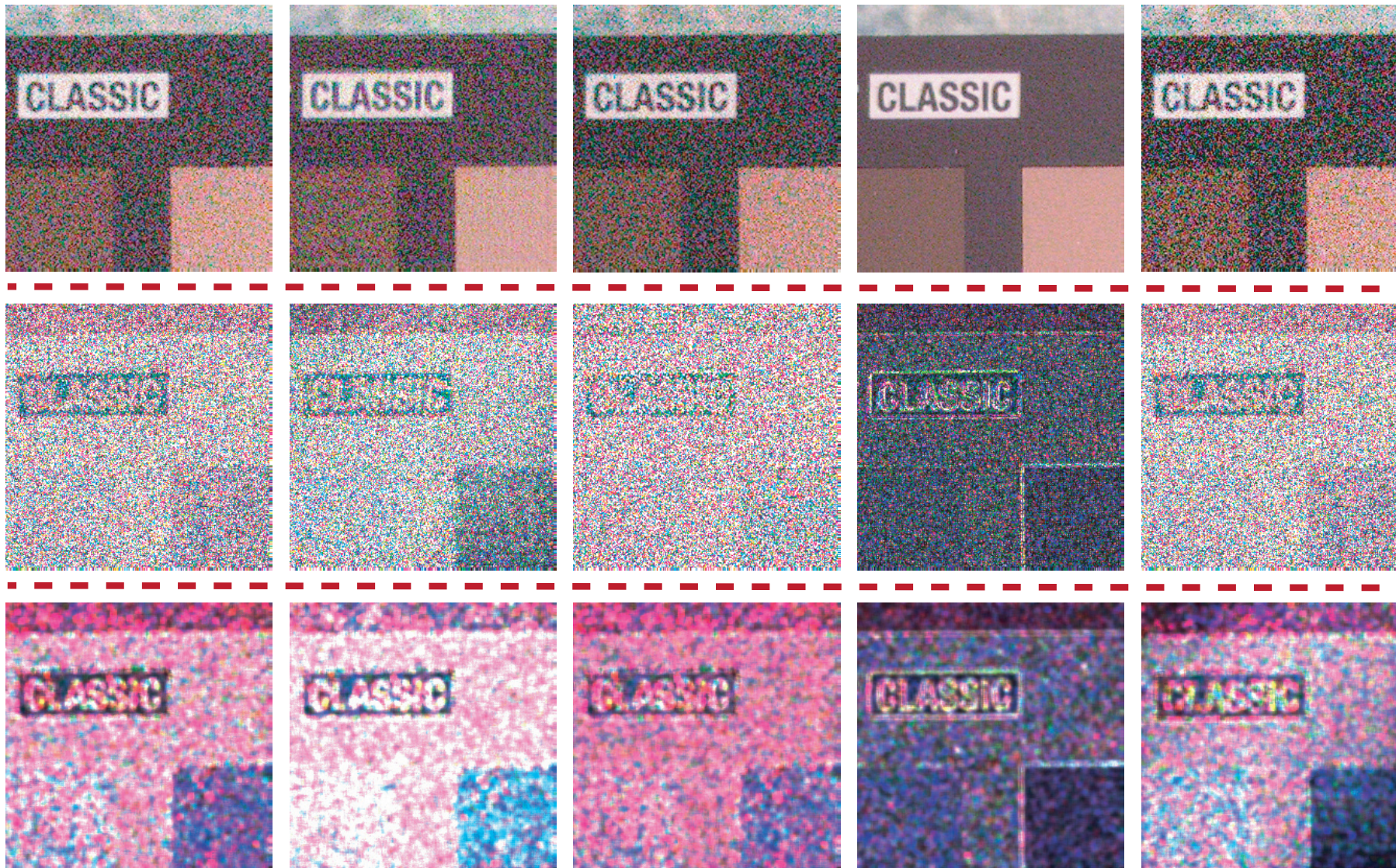
Image Denoising



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

Image Denoising

Generated Noisy Images



(a) Real

(b) CBDNet

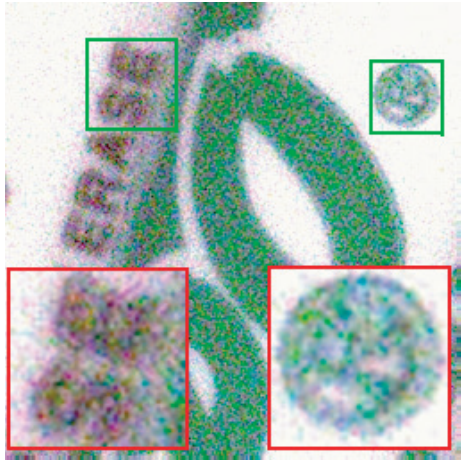
(c) ULRD

(d) GRDN

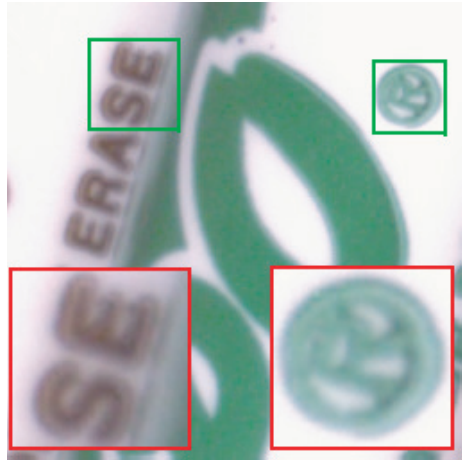
(e) DANet

Image Denoising

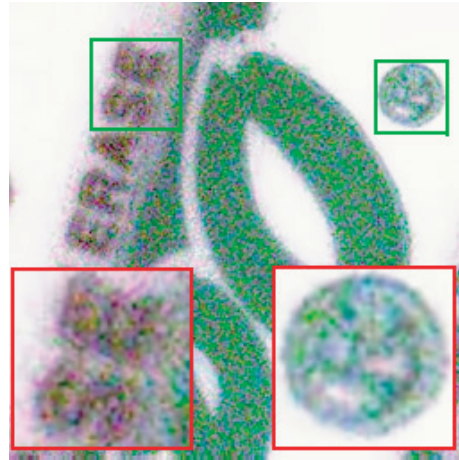
Denoising results



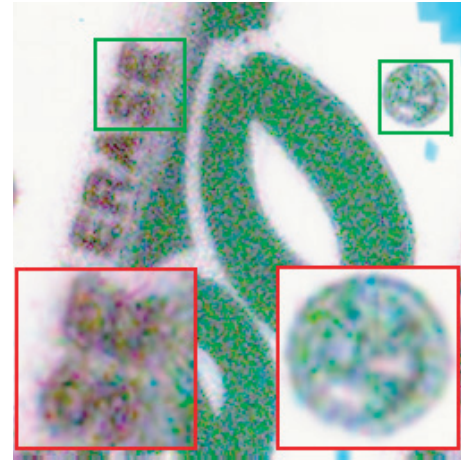
(a): Noisy



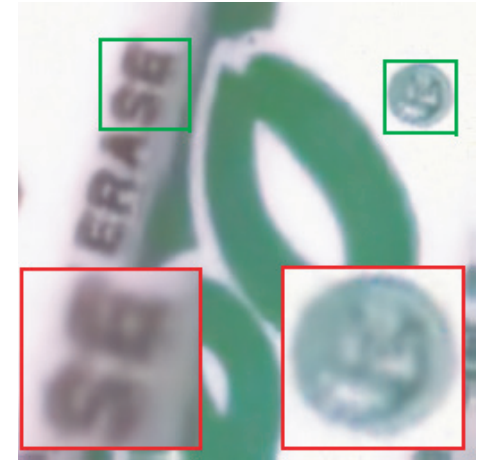
(b): GroundTruth



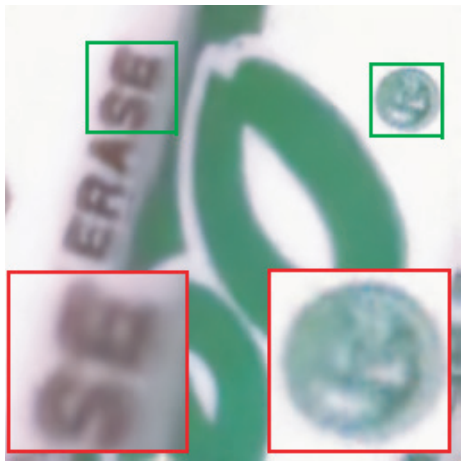
(c): BM3D



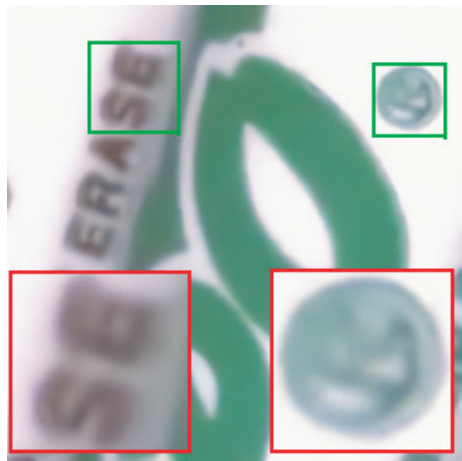
(d): WNNM



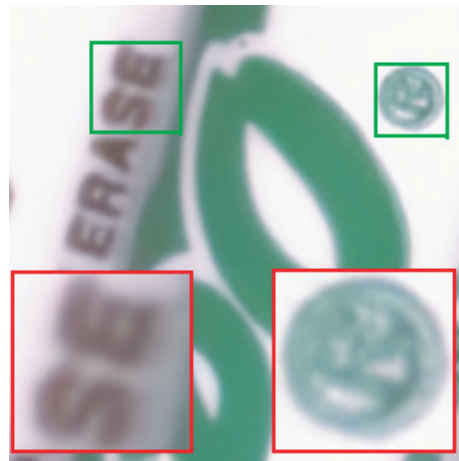
(e): DnCNN



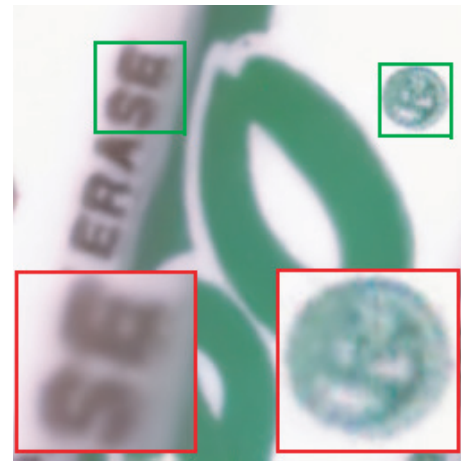
(f): CBDNet



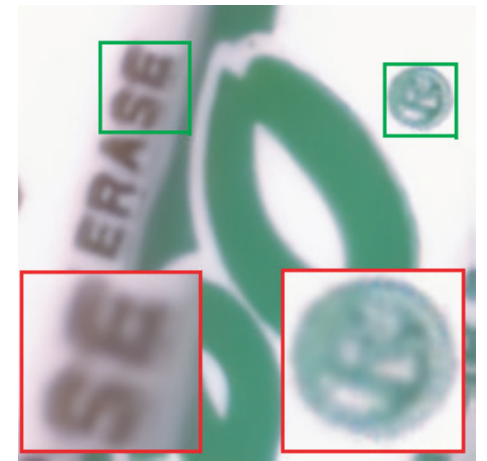
(g): RIDNet



(h): VDN

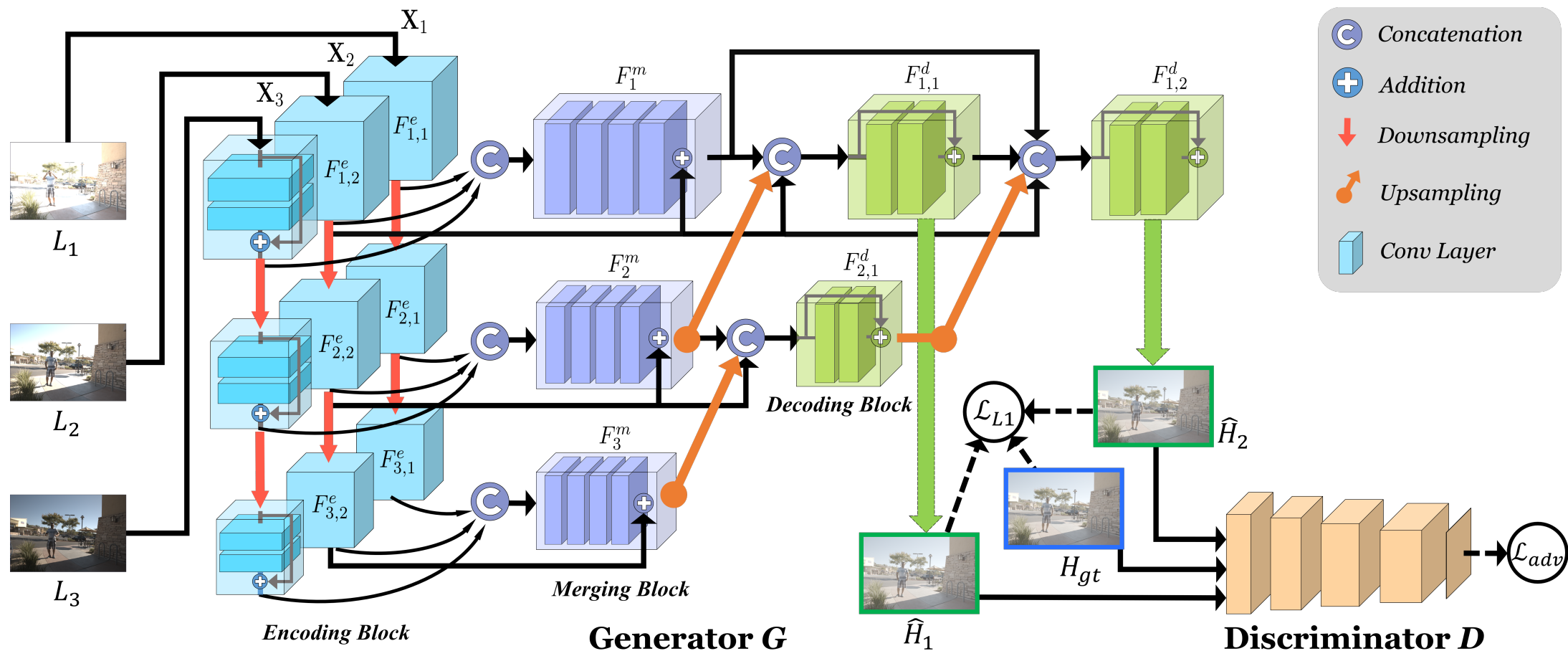


(i): DANet



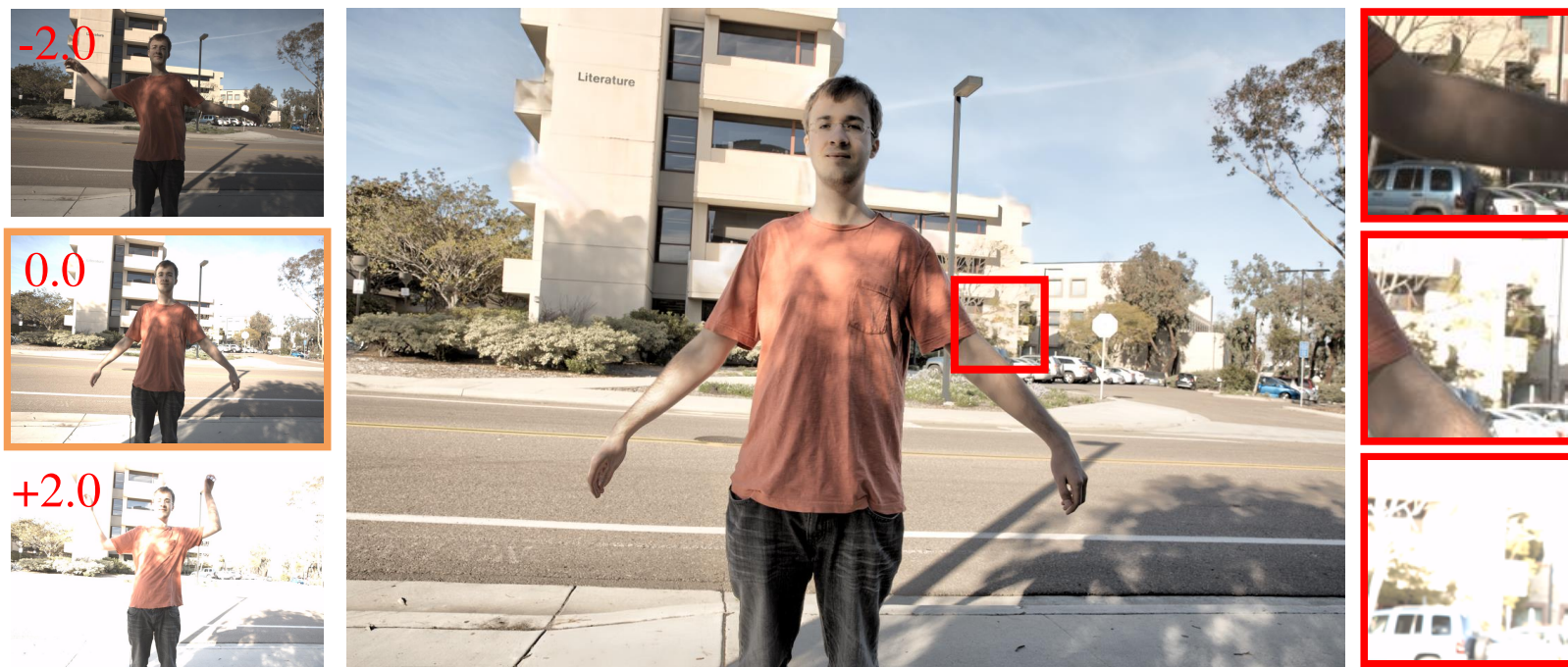
(j): DANet+

High Dynamic Range Imaging



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

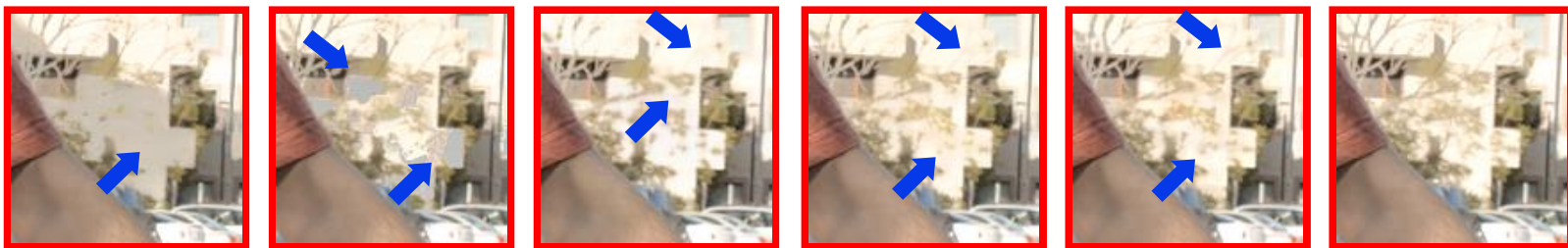
High Dynamic Range Imaging



LDR

Our generated tonemapped HDR image

LDR patches



Sen *et al.*

Kalantari *et al.*

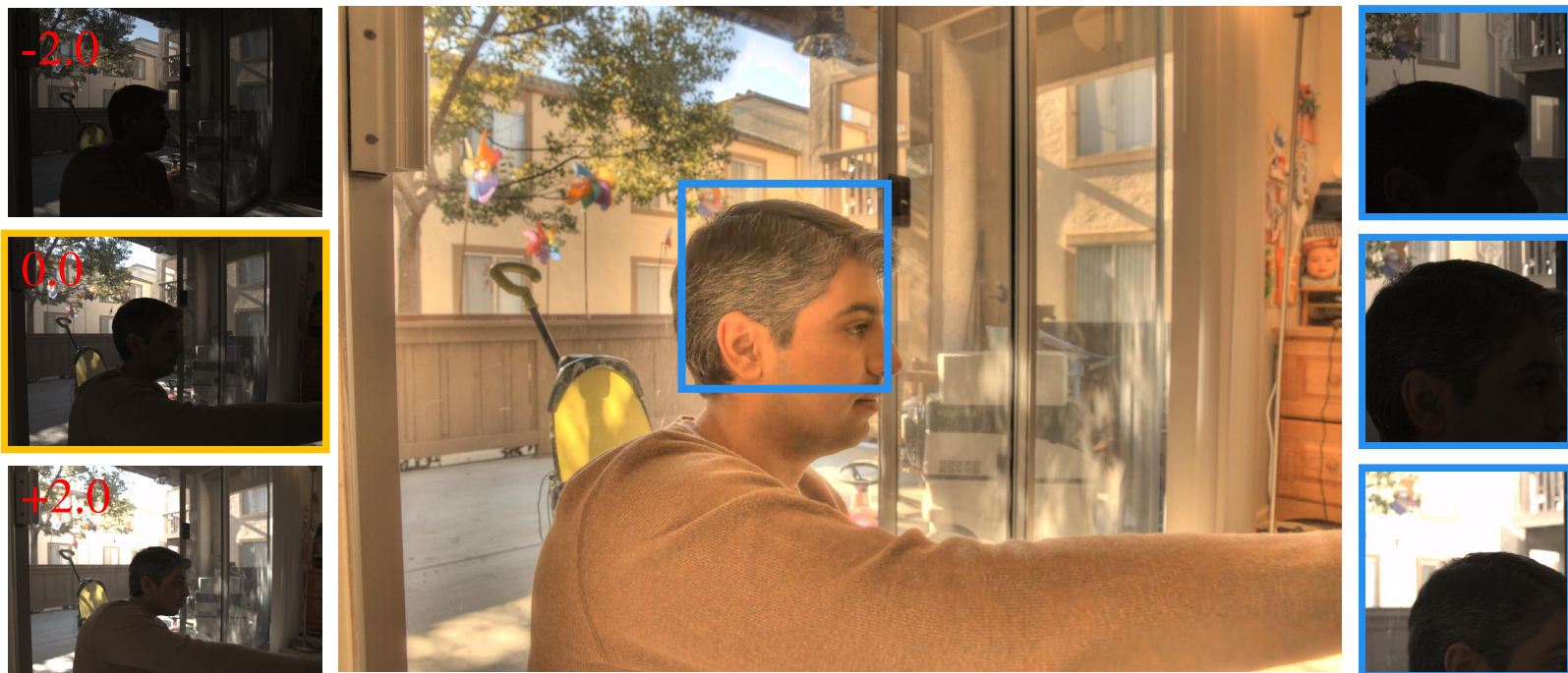
DeepHDR

AHDRNet

Ours

GT

High Dynamic Range Imaging



LDR

Our generated tonemapped HDR image

LDR Patches



Sen *et al.*

Kalantari *et al.*

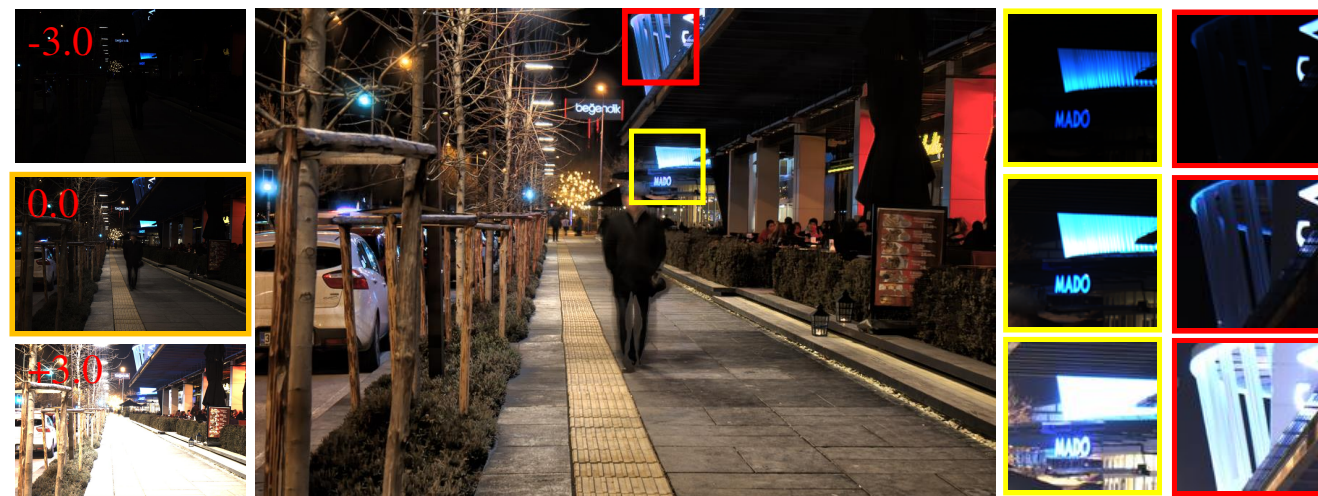
DeepHDR

AHDRNet

Ours

GT

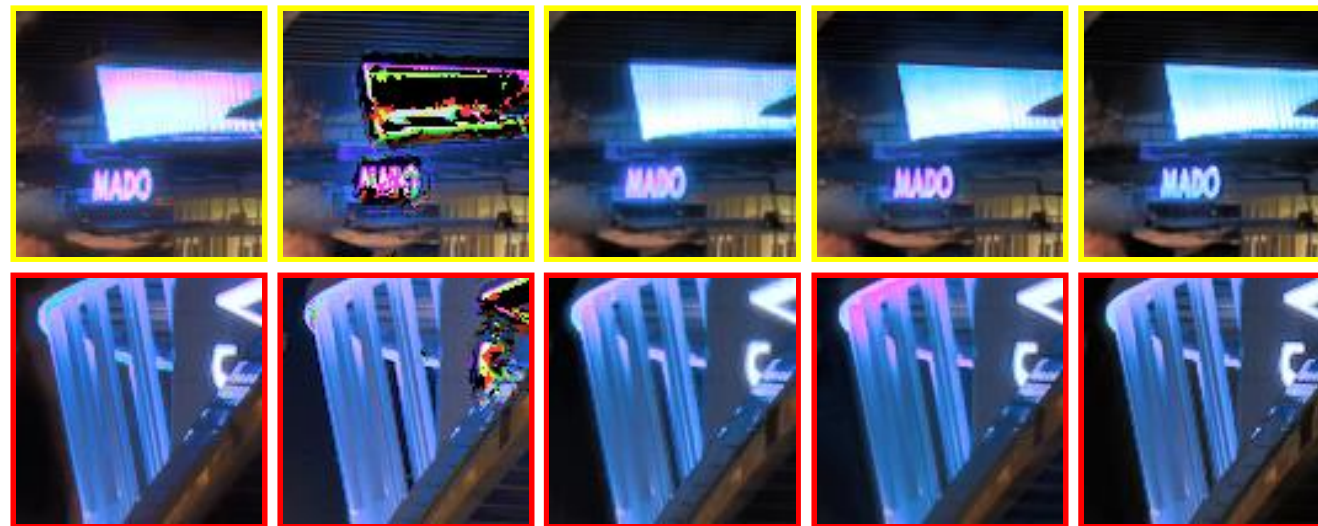
High Dynamic Range Imaging



LDRs

Our generated tonemapped HDR image

LDR Patches



Sen *et al.*

Kalantari *et al.*

DeepHDR

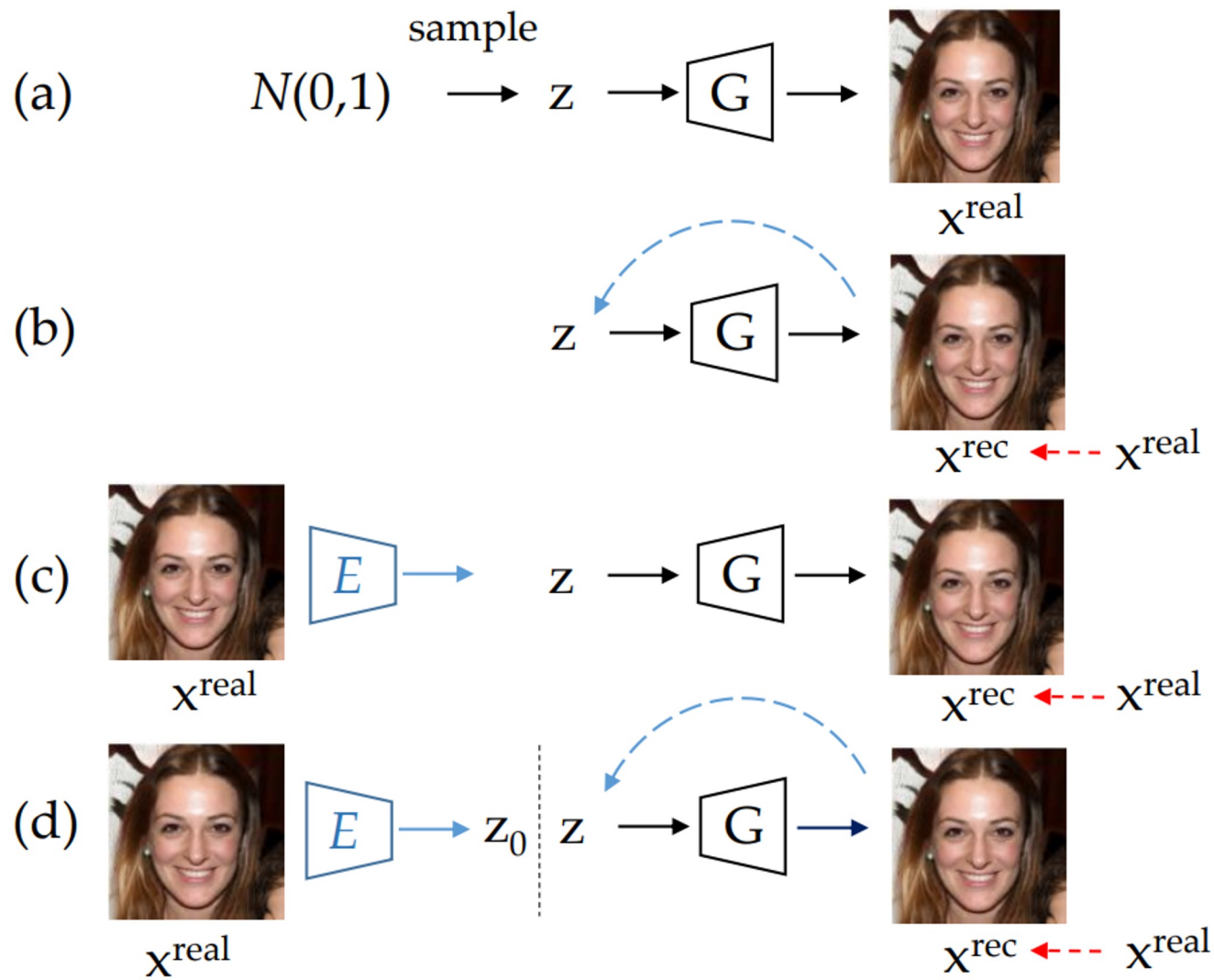
AHDRNet

Ours

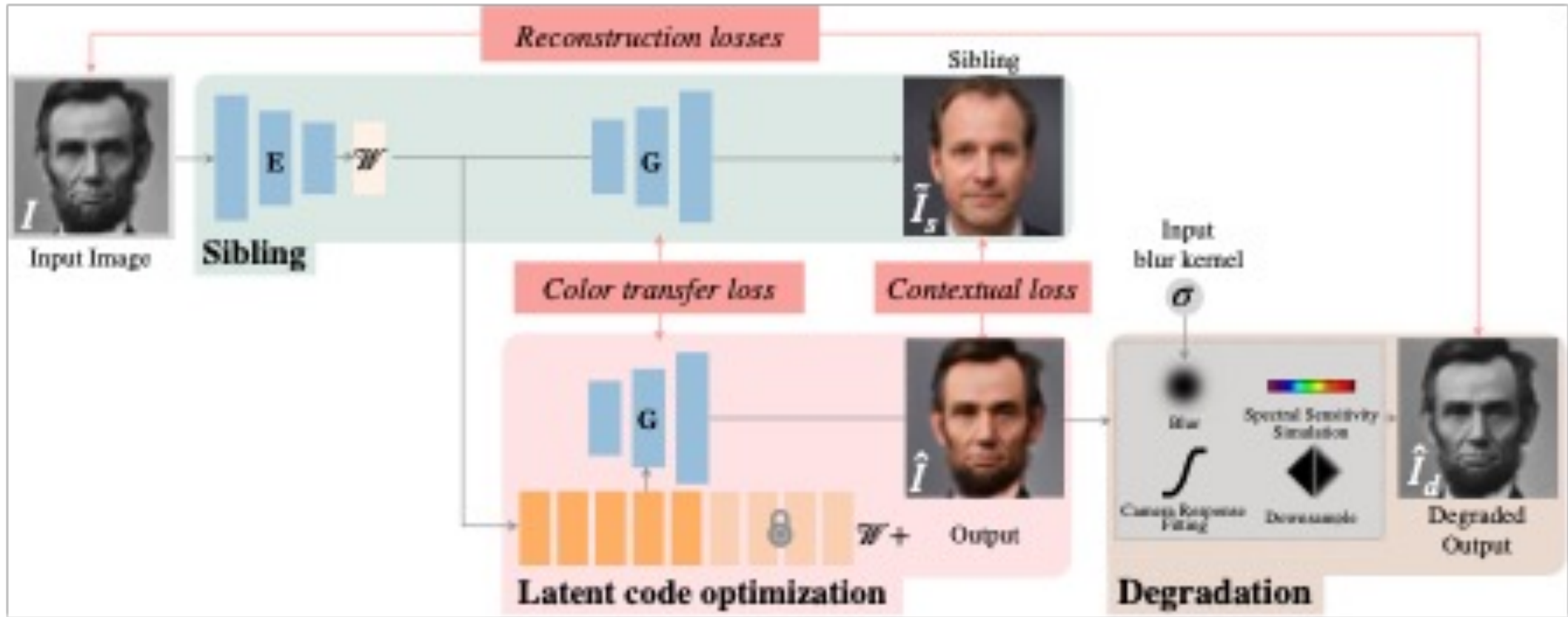
GAN Inversion

3 methods of inversion:

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

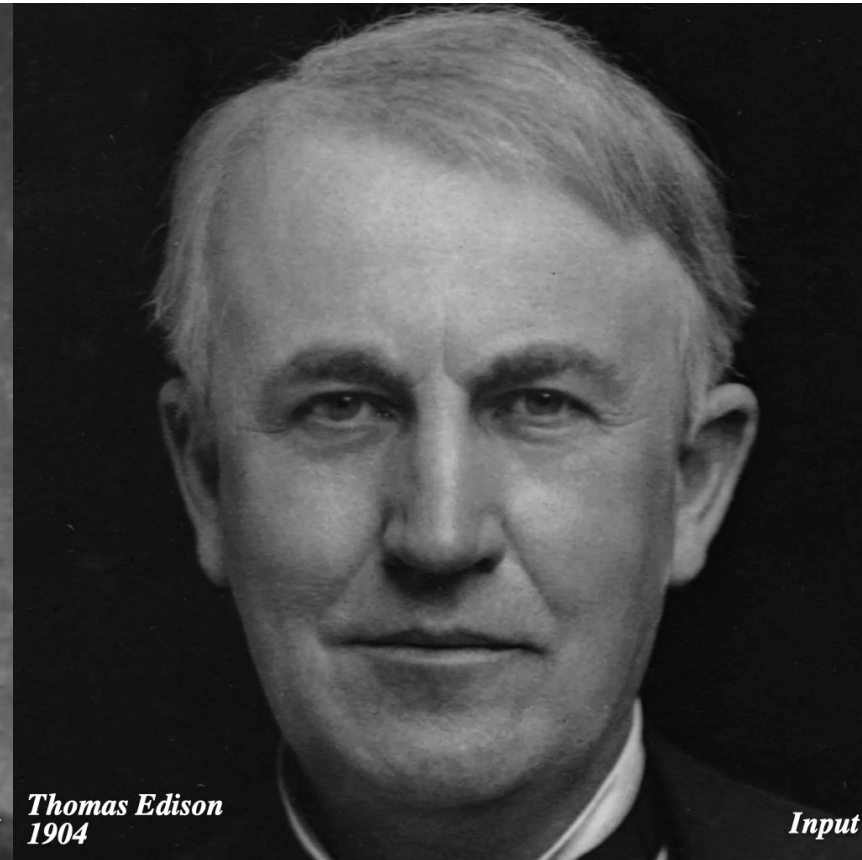


Time-travel Rephotography



- 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



Input

Ours

DeOldify

InstColorization

Zhang

Zhang (FFHQ)

Palette: Image-to-Image Diffusion Models

Input

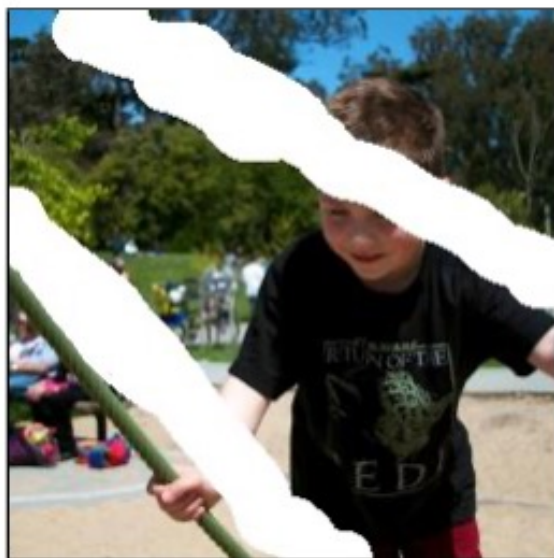
Output

Original

Colorization



Inpainting



[Saharia et al. 2022]

Palette: Image-to-Image Diffusion Models

Input

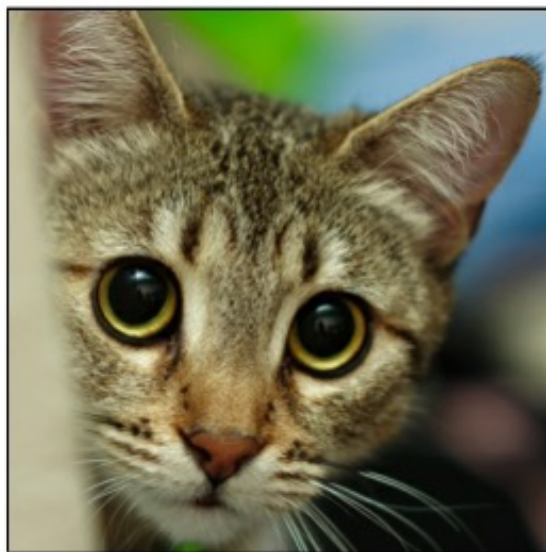
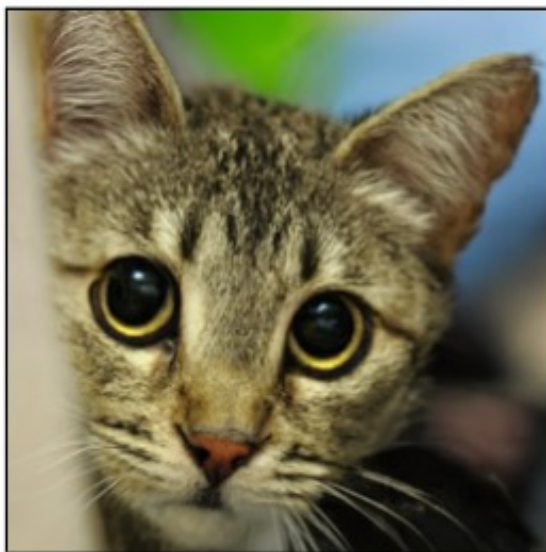
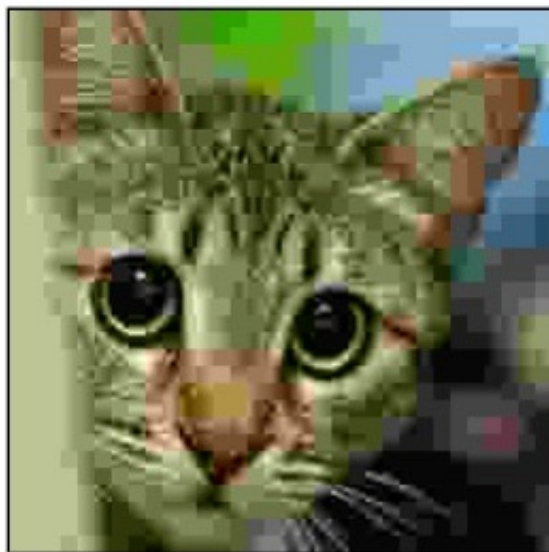
Output

Original

Uncropping



JPEG restoration



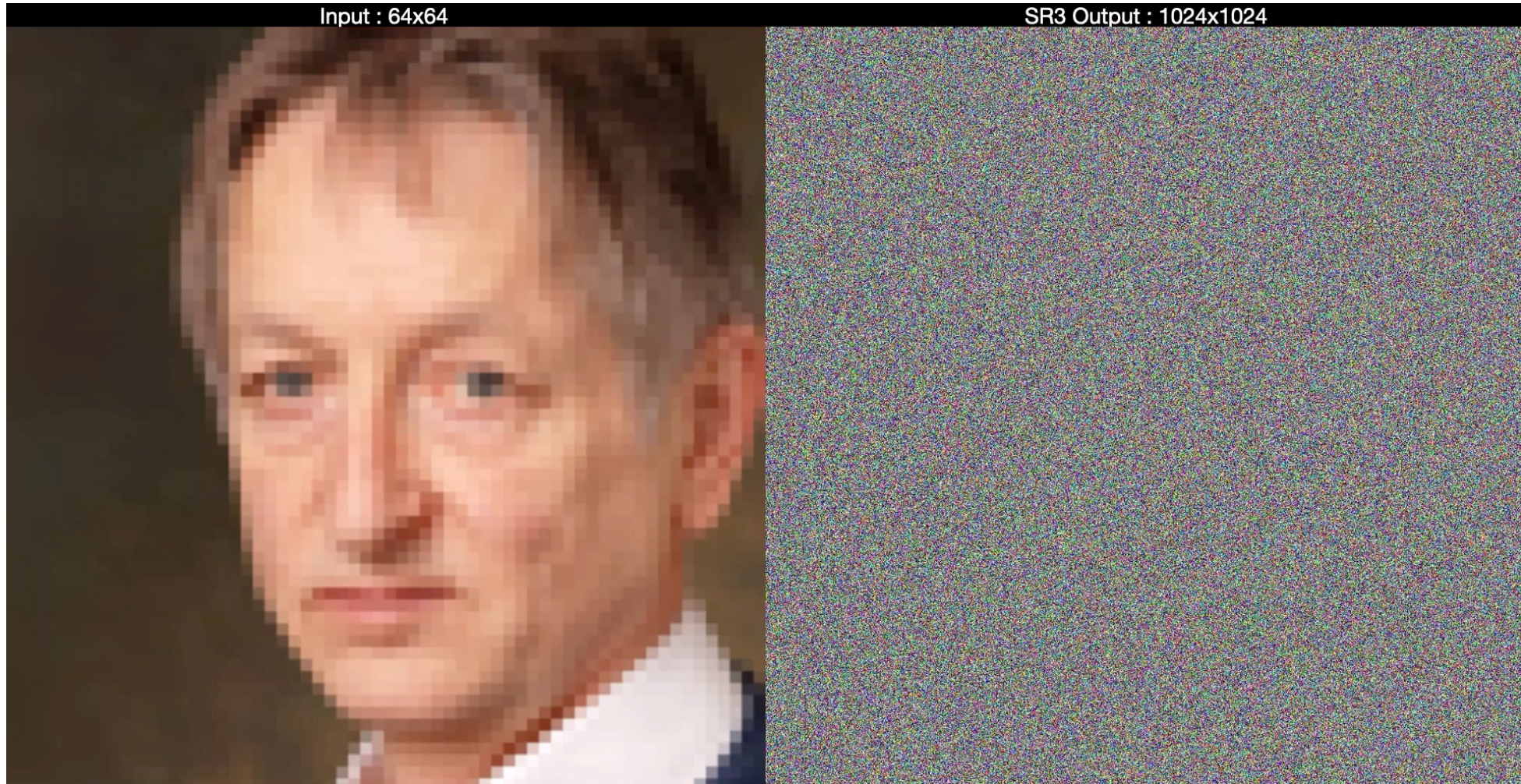
[Saharia et al. 2022]

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

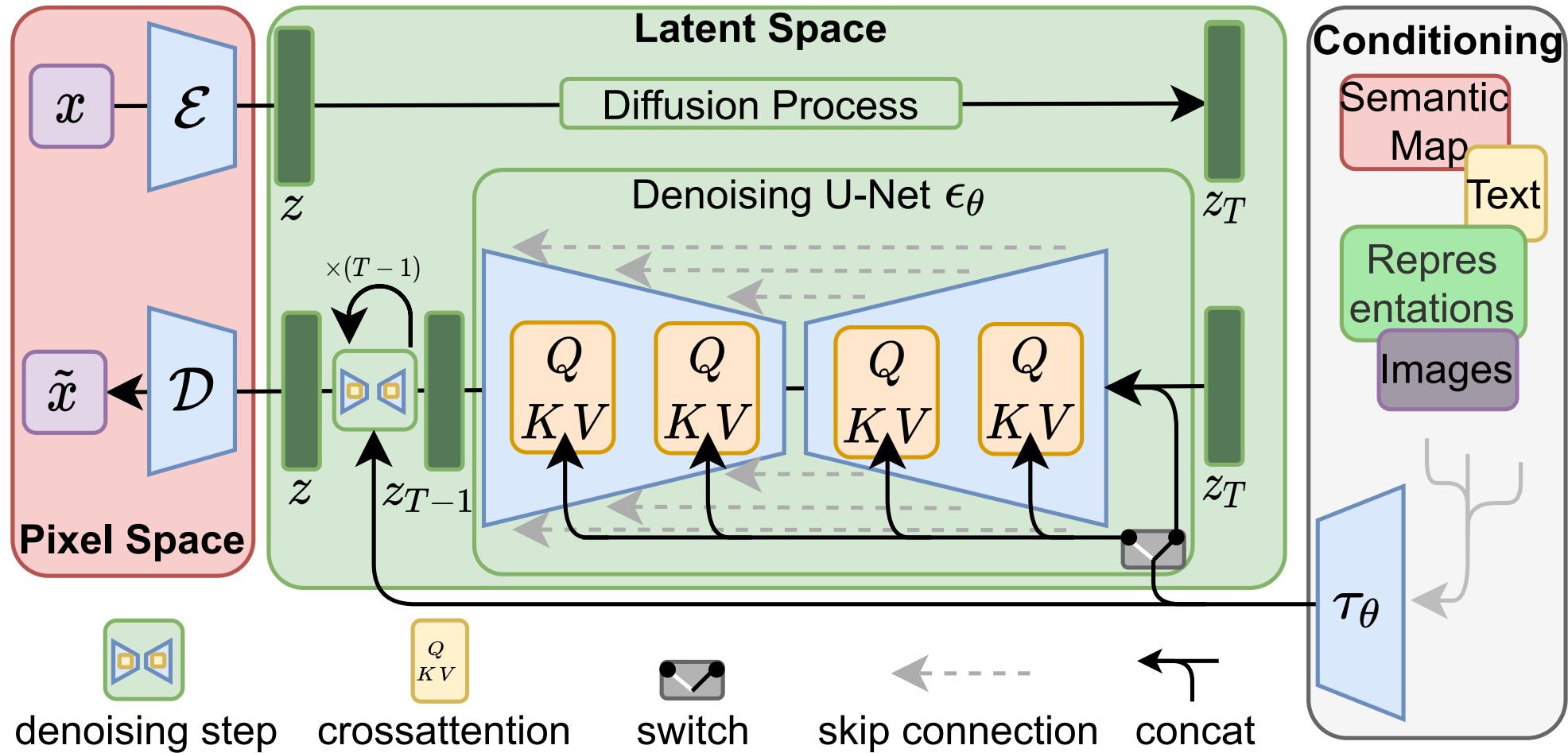
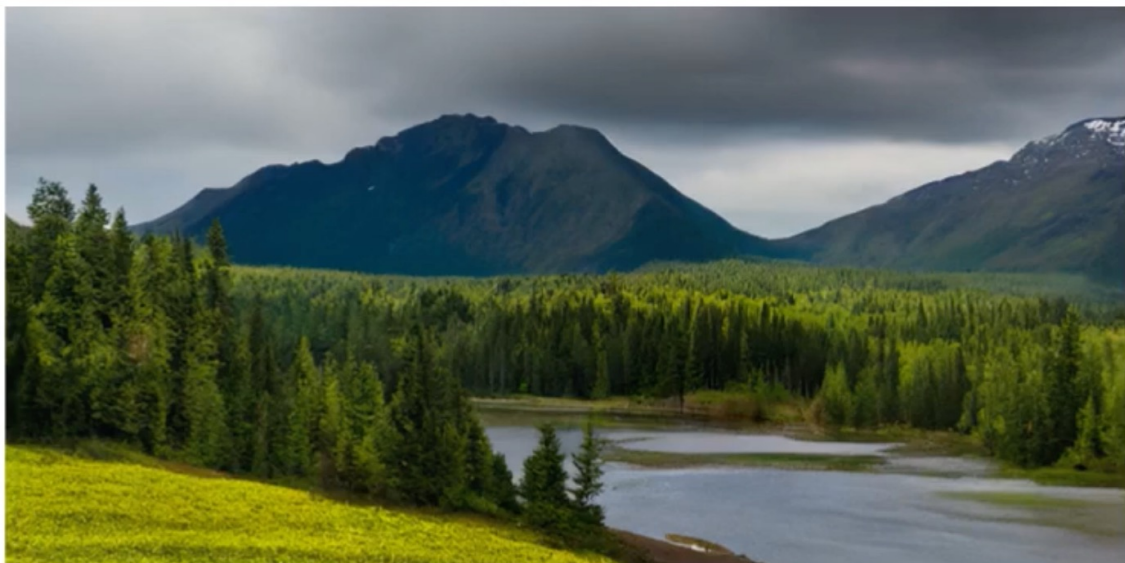


Image Inpainting



Semantic Image Synthesis

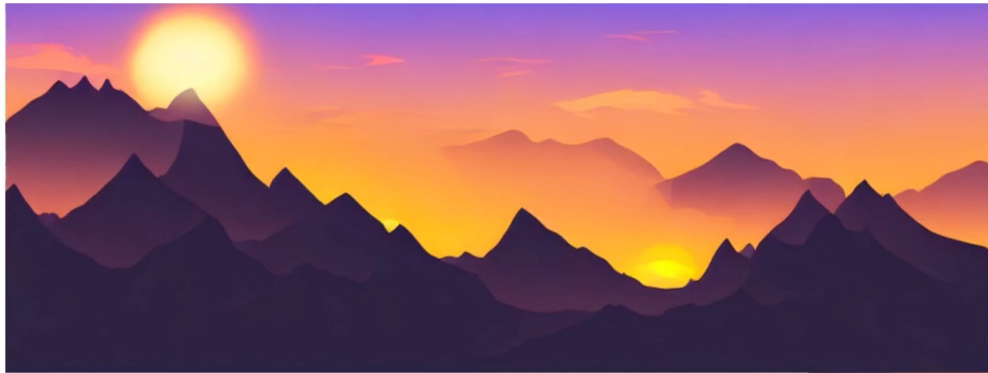


Generating Images from Text



"A sunset over a mountain, vector image"

"a portrait of a cyberpunk rabbit, trending on artstation"



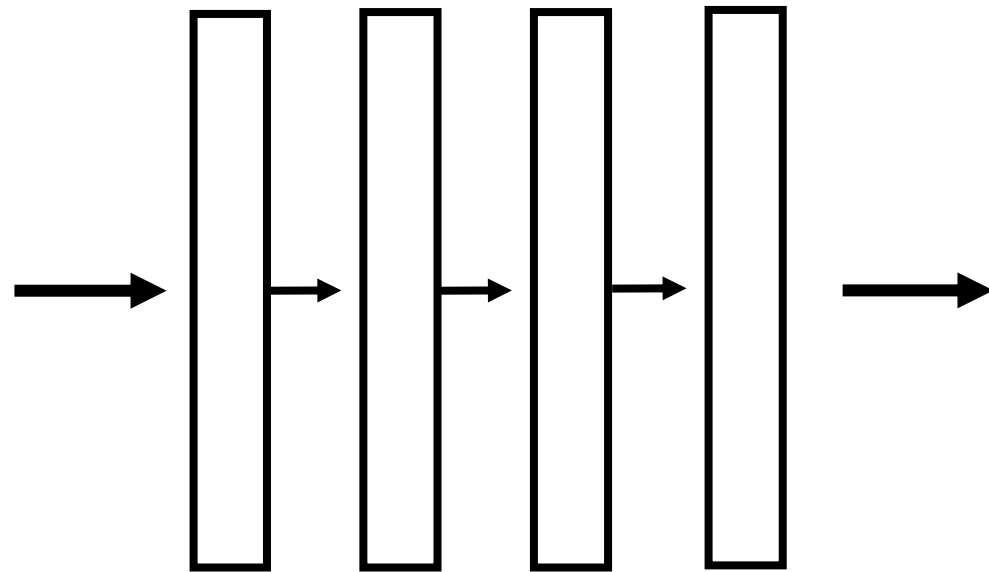
"A sunset over a mountain, oil on canvas"



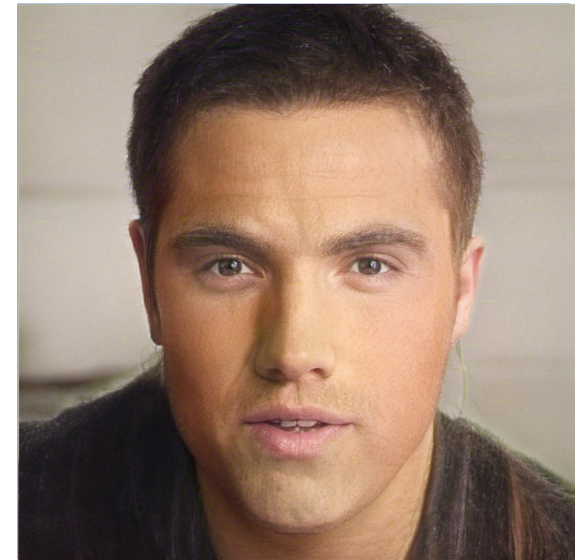
Semantic Image Editing



image \mathbf{x}



"he is tanned"



manipulated
image \mathbf{y}

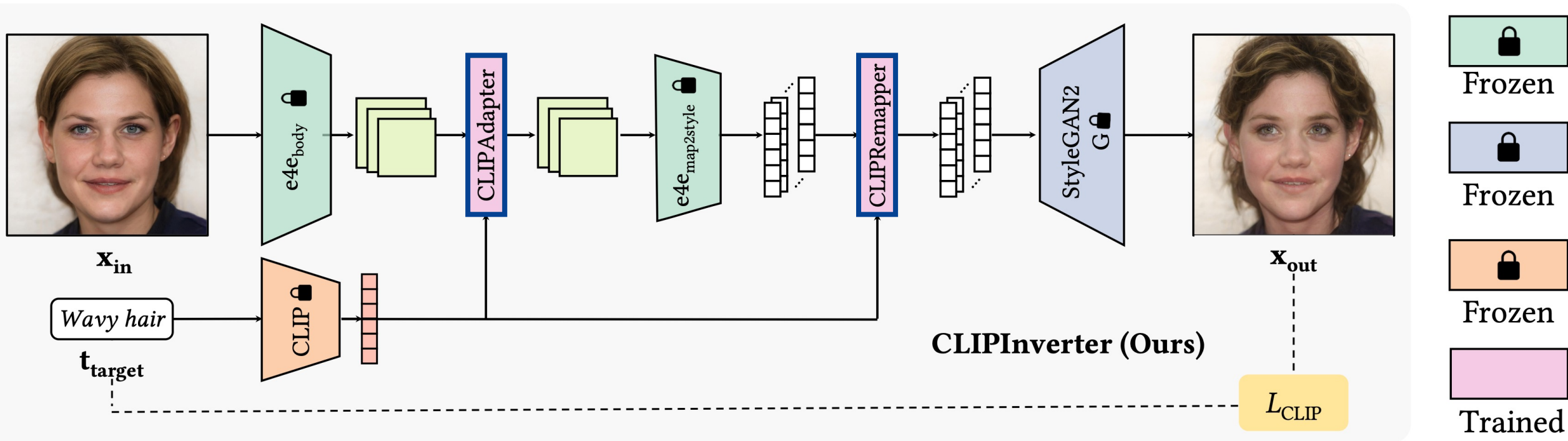
- **Input:** input image

+ target description

Output: manipulated image

Target description \mathbf{t}

CLIPInverter

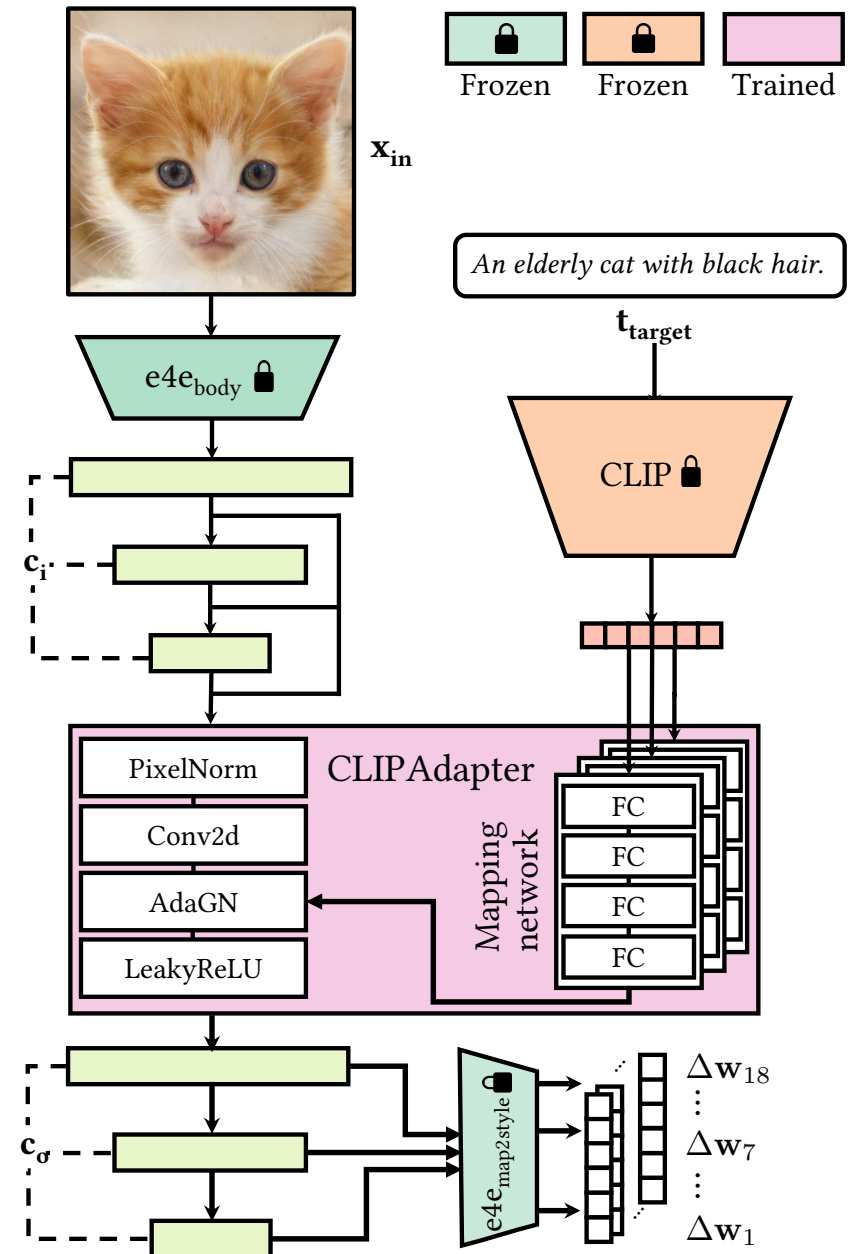


- **CLIPAdapter:** Finds and follows semantic paths in latent space for initial image edits.
- **CLIPRemapper:** Refines edits by aligning the latent code with text prompt embeddings.

CLIPAdapter

- Extract feature maps from the image.
- Modulate these features with CLIP text embeddings.
- Input modulated features into the encoder to produce residual latents.

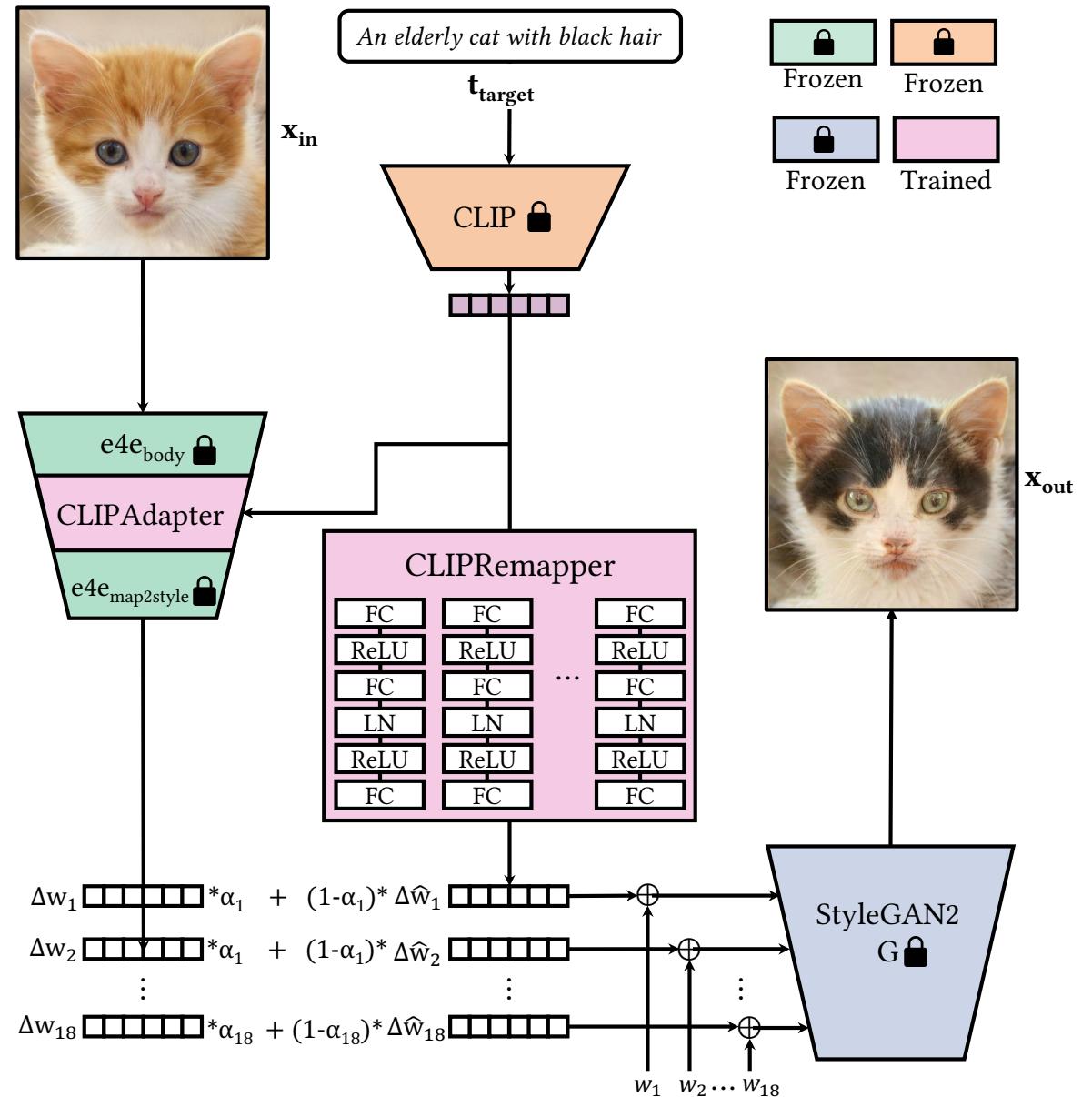
We will show that this improves editability!



CLIPRemapper

- CLIPAdapter provides residual latents.
- Remapper calculates corrections using text prompts.
- Combines residuals with corrections to input into the generator for final image output.

This further improves editing accuracy!





Target: This person has mustache



Target: A cat with ginger hair



Target: This bird has wings that are blue and has a white belly 242

Original



Smiling



Smiling+Chubby



Smiling+Chubby+Beard



Original



TediGAN-B



CLIPInverter (Ours)



Target: He wears necktie. He has bushy eyebrows, mouth slightly open, bags under eyes, big nose, and high cheekbones. He is smiling.

Original



StyleCLIP-LO



CLIPInverter (Ours)



Target: He wears necktie. He has bushy eyebrows, mouth slightly open, bags under eyes, big nose, and high cheekbones. He is smiling.

Original



StyleCLIP-LO



CLIPInverter (Ours)



Target: He wears necktie. He has bushy eyebrows, mouth slightly open, bags under eyes, big nose, and high cheekbones. He is smiling.

Original



StyleMC



CLIPInverter (Ours)



Target: He wears necktie. He has bushy eyebrows, mouth slightly open, bags under eyes, big nose, and high cheekbones. He is smiling.

Original



HairCLIP



CLIPInverter (Ours)



Target: He wears necktie. He has bushy eyebrows, mouth slightly open, bags under eyes, big nose, and high cheekbones. He is smiling.

Limitations

- Directional CLIP loss may unintentionally alter gender representation.

Original



Wavy Hair



Limitations

- Directional CLIP loss may unintentionally alter gender representation.
- Adding a specific pronoun to the target text description helps prevent this.

Original

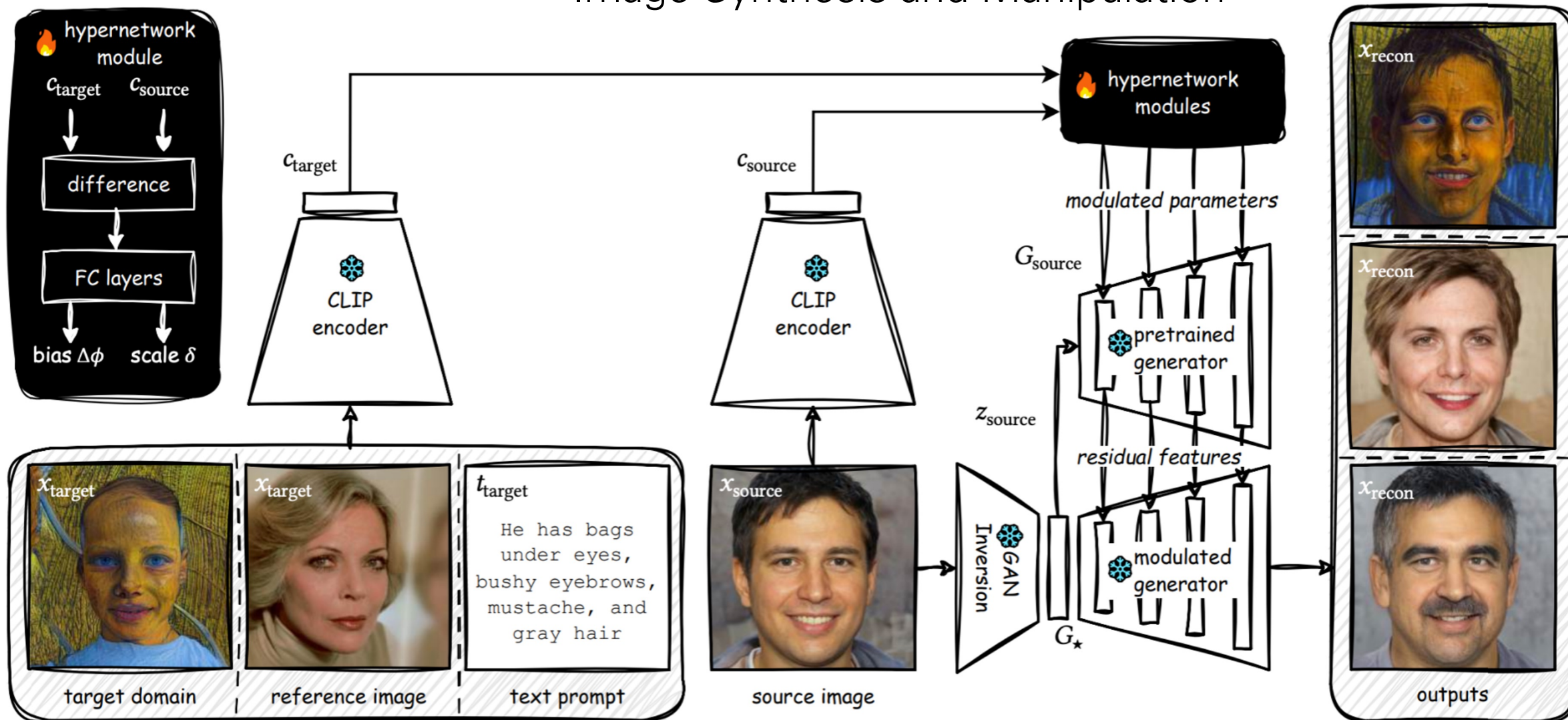
Wavy Hair

She has wavy hair

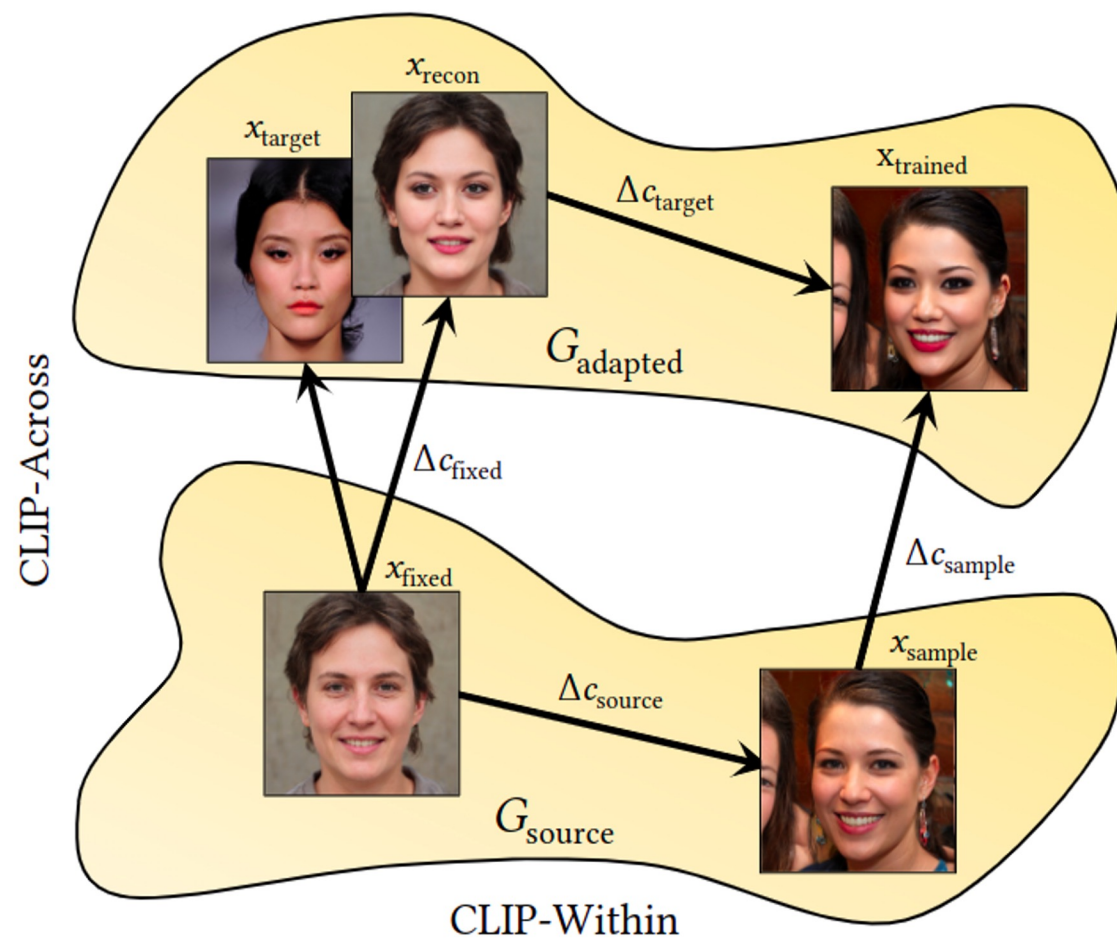
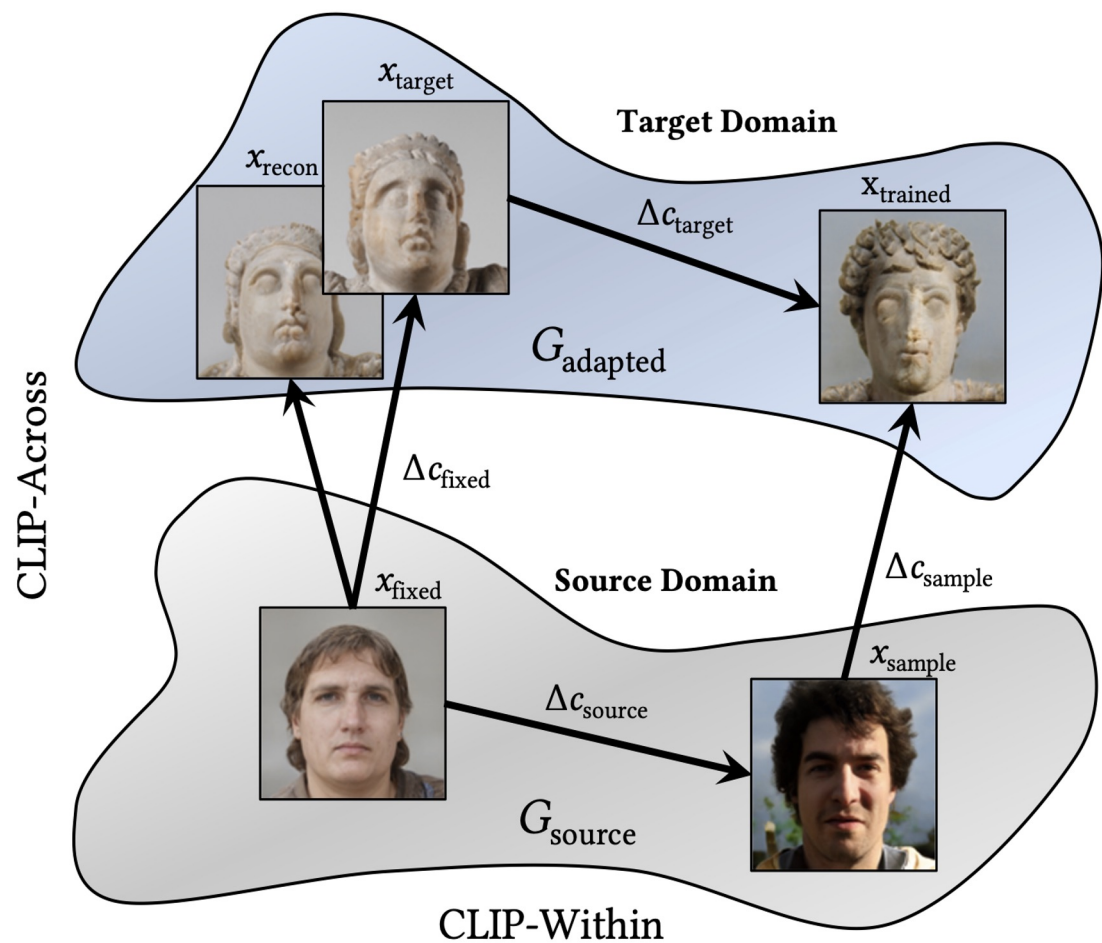


HyperGAN-CLIP

A Unified Framework for Domain Adaptation, Image Synthesis and Manipulation



HyperGAN-CLIP



Domain Adaptation Results



source



target



result

Domain Adaptation Results

Source

Target

StyleGAN-NADA

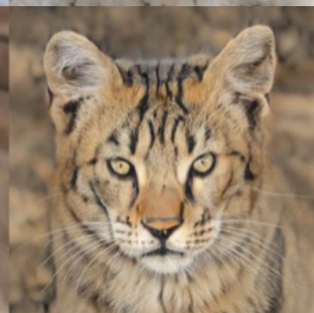
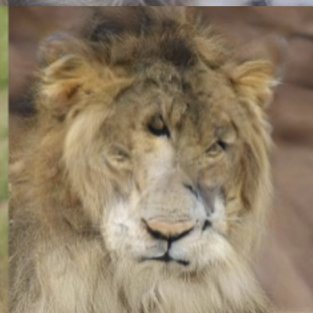
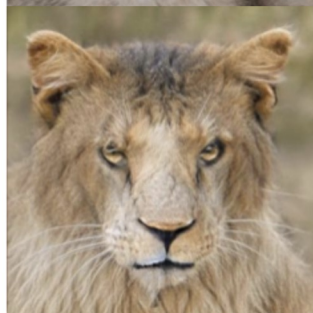
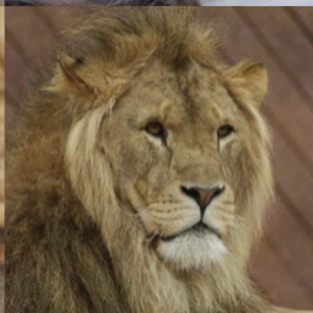
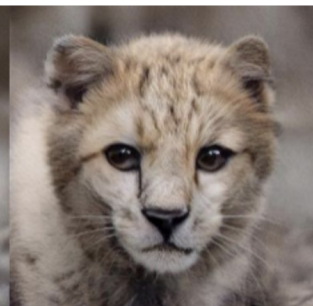
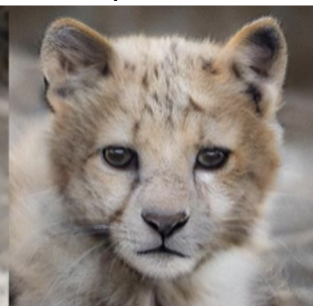
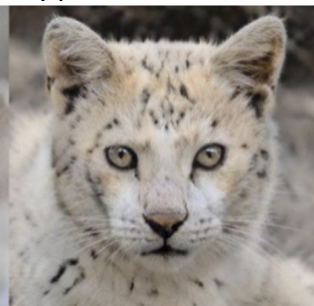
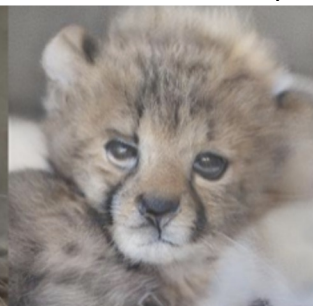
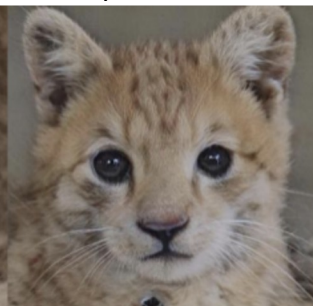
Adaptation-SCR

Mind-the-Gap

HyperDomainNet

DynaGAN

Ours



Reference-Guided Synthesis Results



source

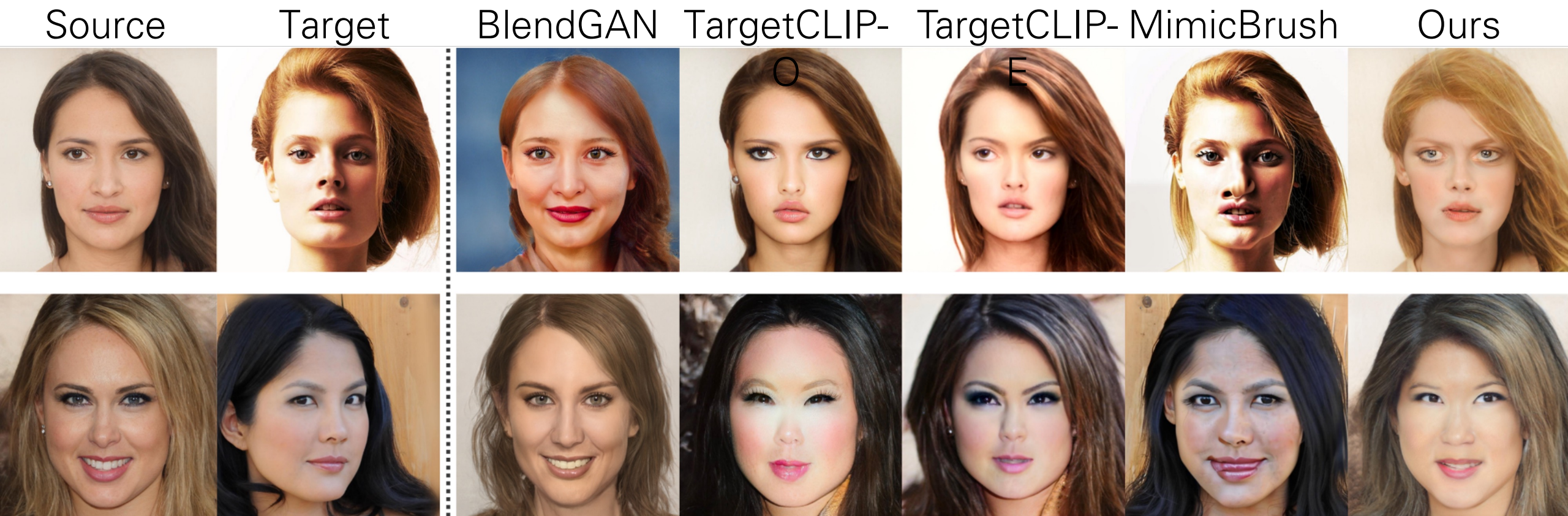


target



result

Reference-Guided Synthesis Results



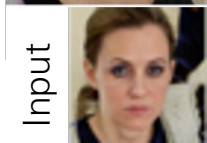
Text-Guided Image Manipulation Results



"Elsa from Frozen"



Text-Guided Image Manipulation Results

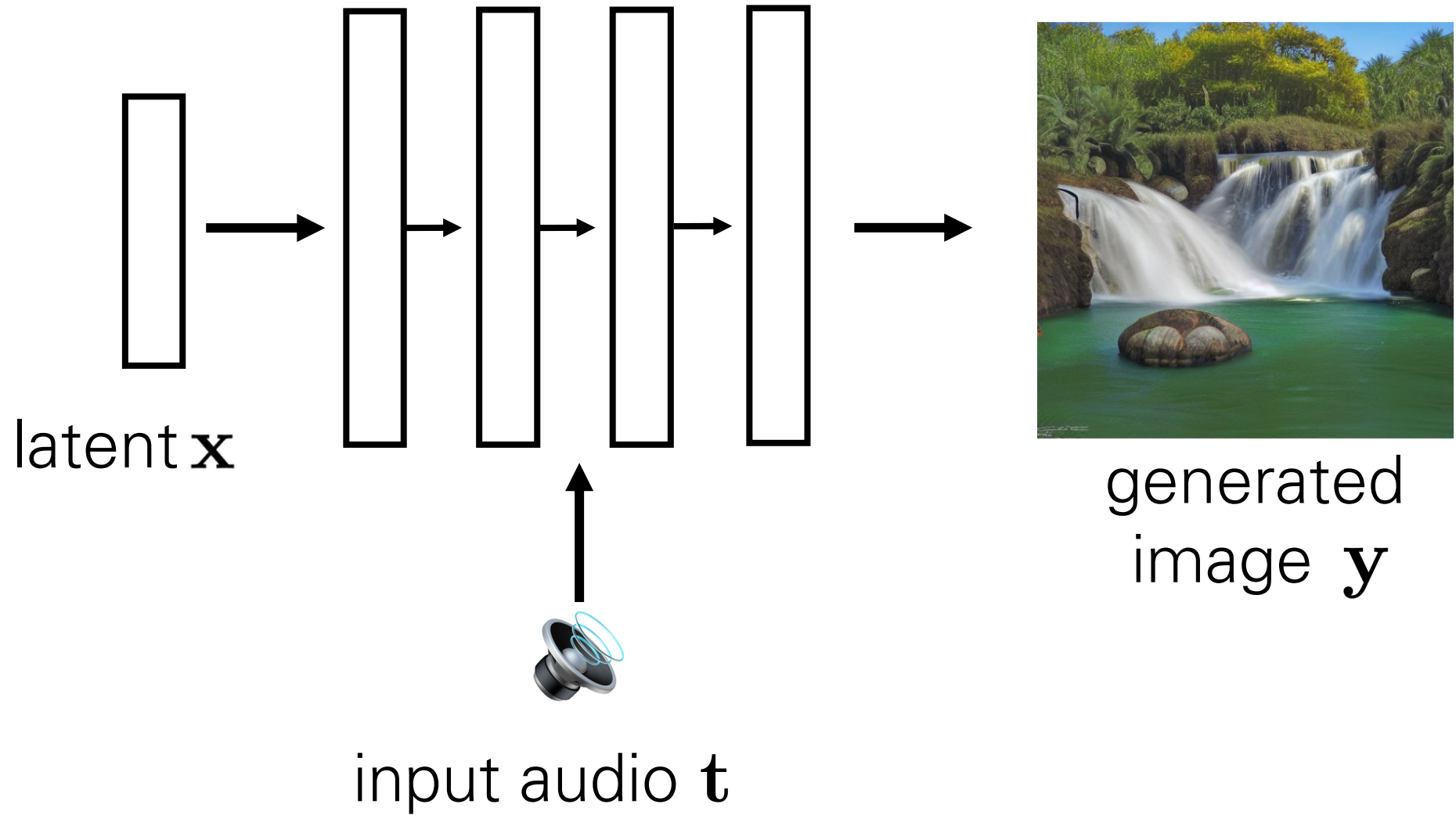


"The person is wearing heavy makeup, lipstick. She has wavy hair, rosy cheeks, and pointy nose."



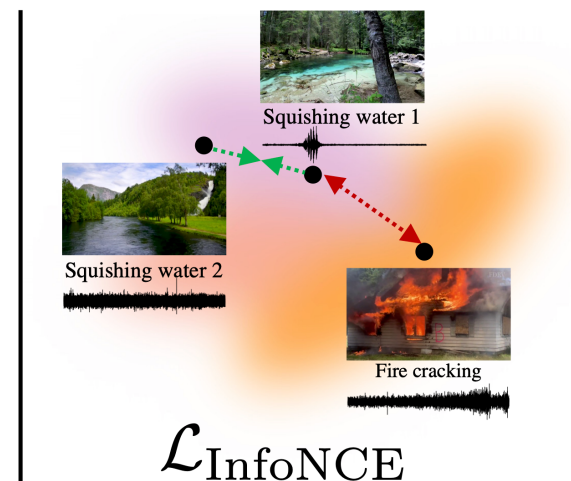
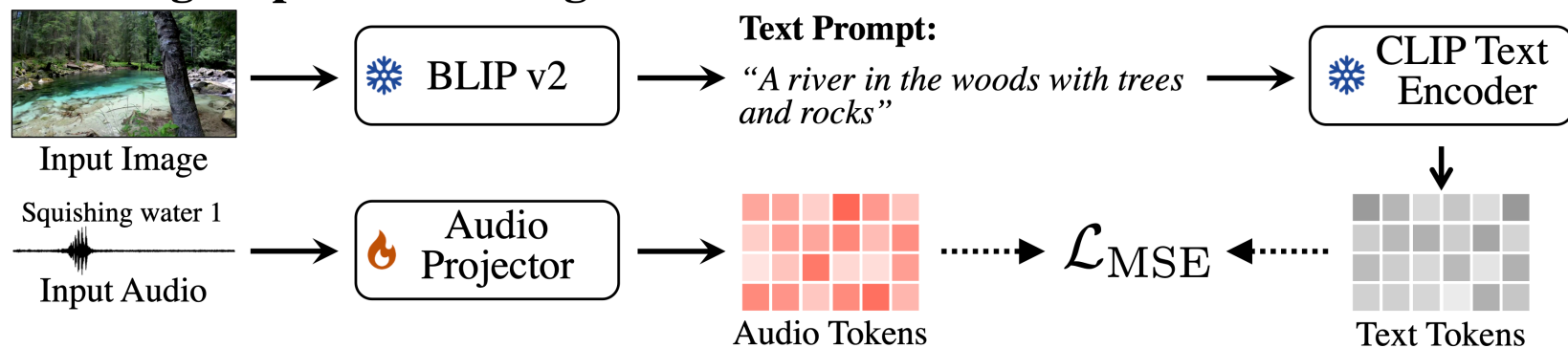
"Surprised"

Audio-Guided Image Synthesis

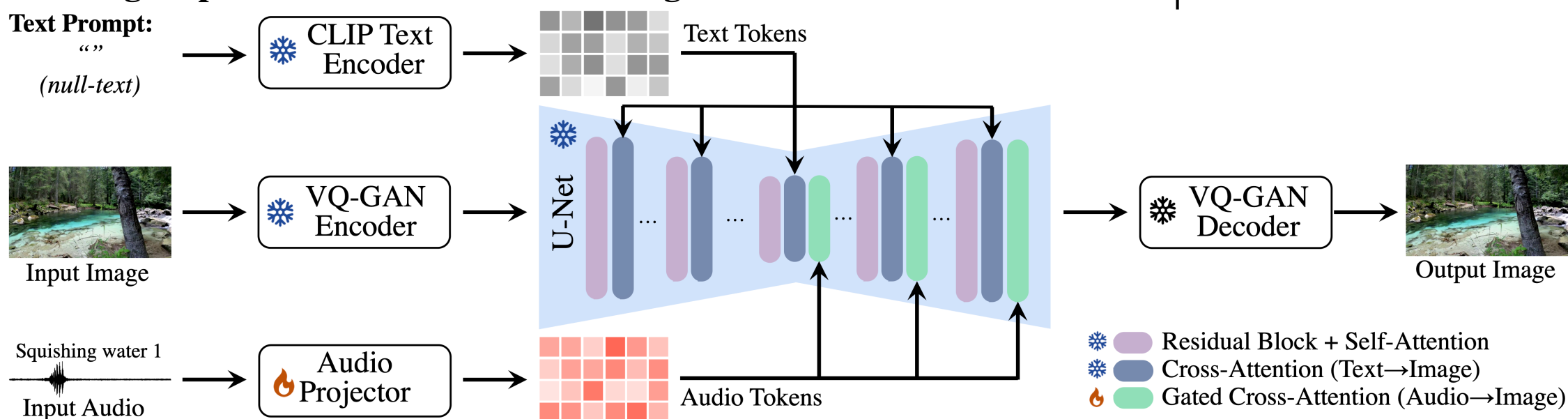


SonicDiffusion – Training

Training Step 1: Initial Alignment

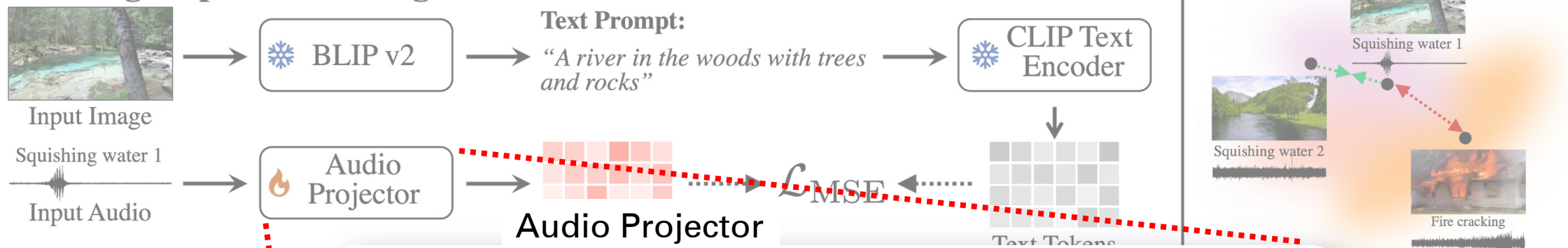


Training Step 2: Parameter Efficient Tuning

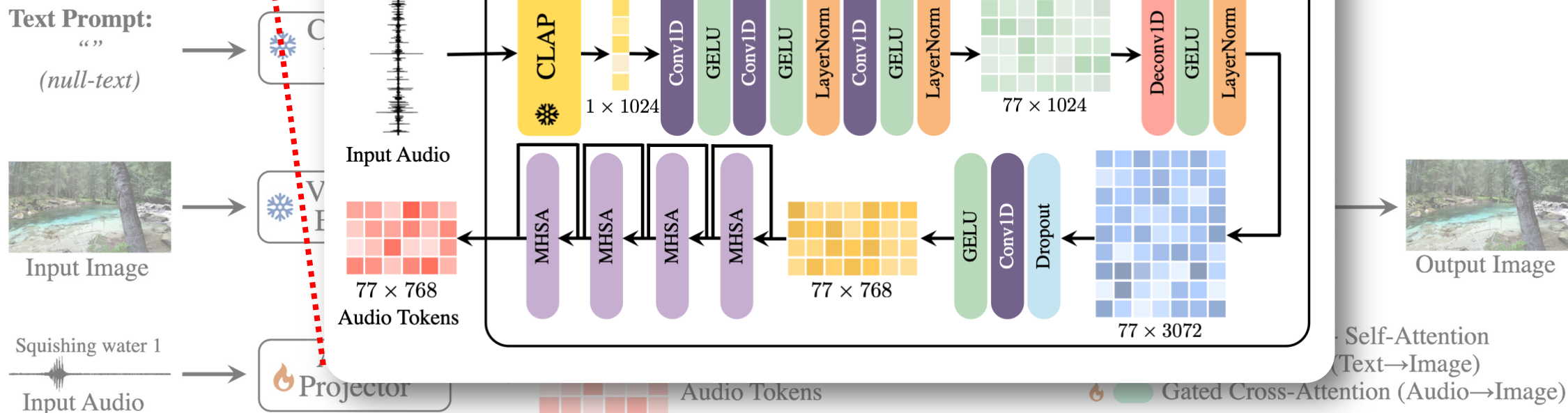


SonicDiffusion – Training

Training Step 1: Initial Alignment

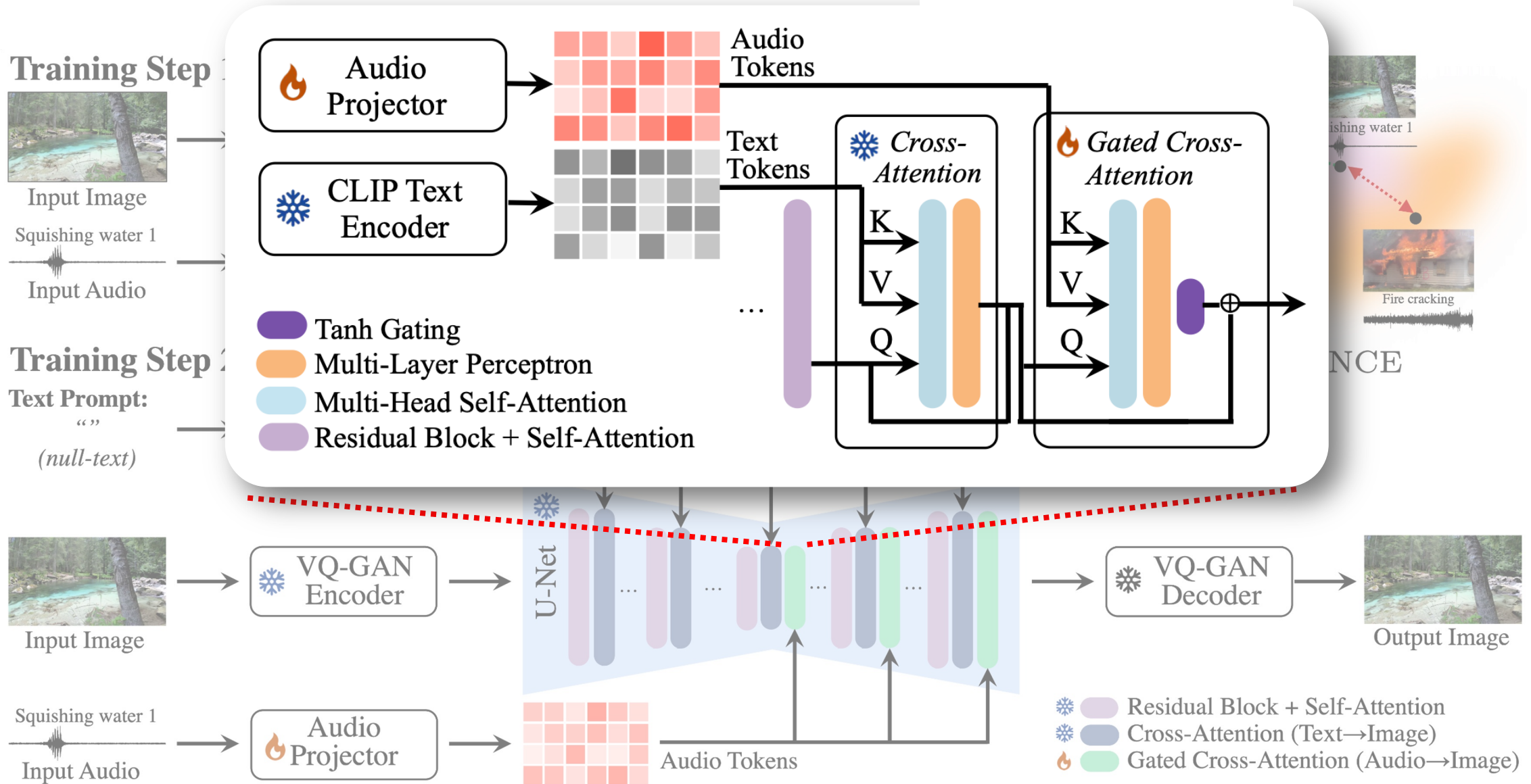


Training Step 2: Parallel



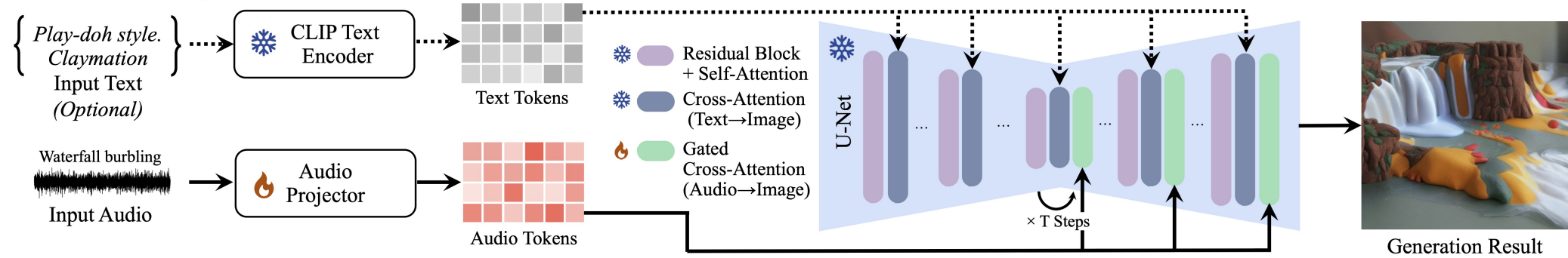
SonicDiffusion – Training

Gated Cross-Attention



SonicDiffusion – Inference

Audio-Driven Image Generation



Audio-Driven Image Editing

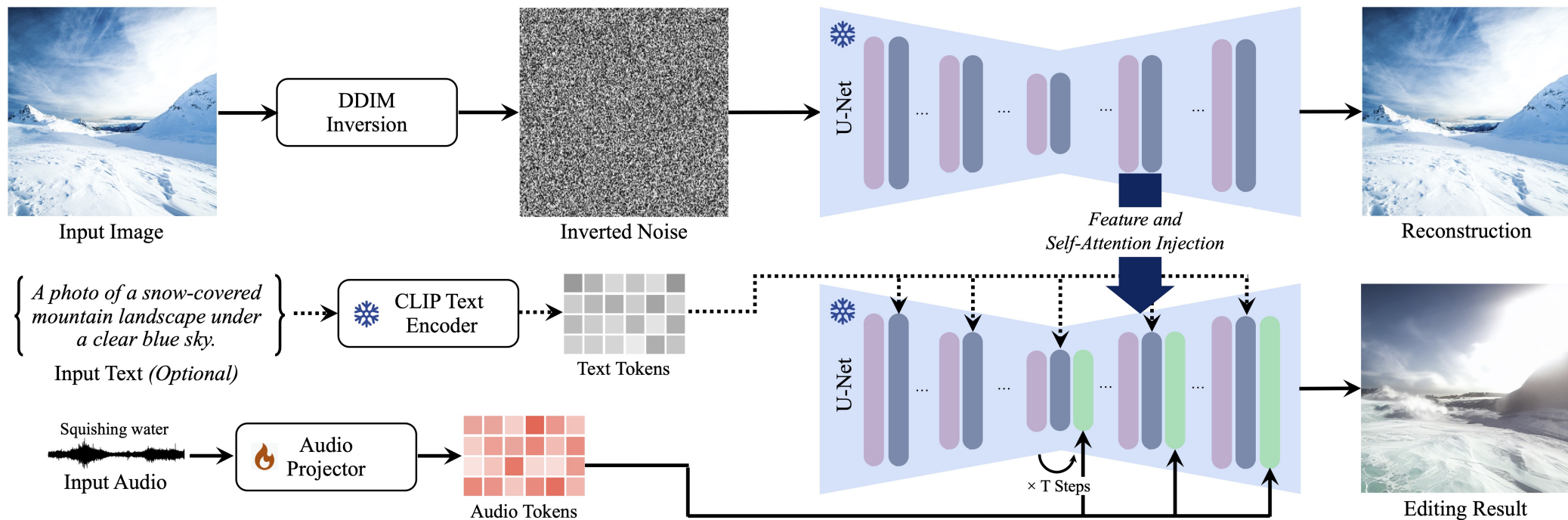


Image Generation



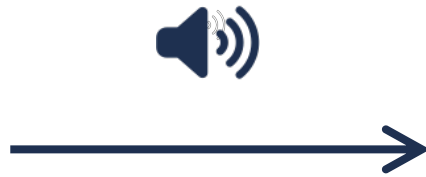
Image Editing



Image Editing



Image Editing



Instruction-Based Object Removal

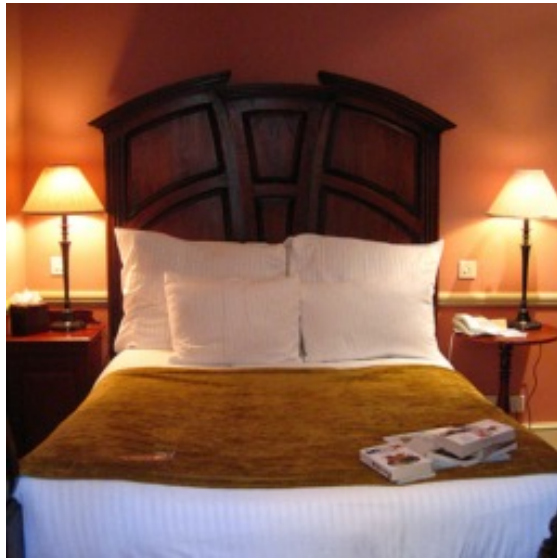
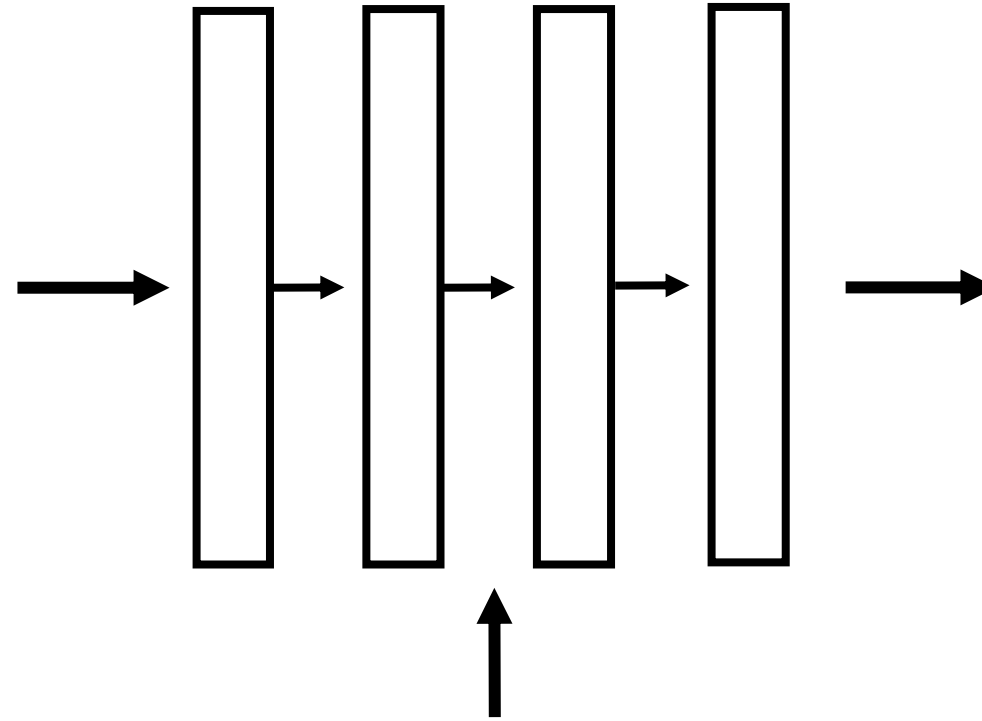
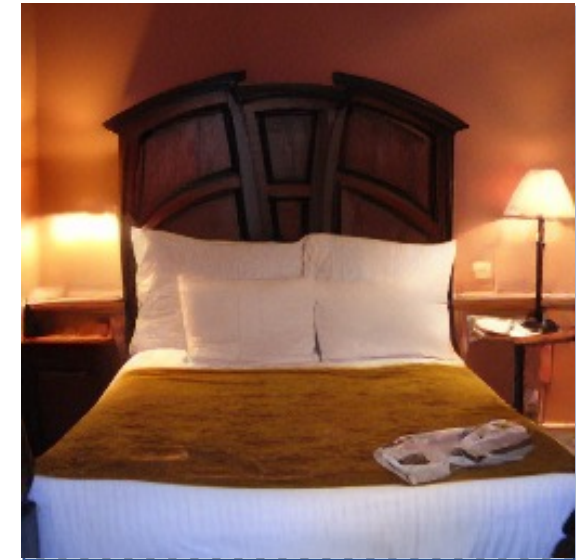


image \mathbf{x}



“remove the lamp at the left of the large headboard”

Instruction \mathbf{t}



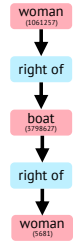
inpainted image \mathbf{y}

- **Input:** input image + instruction

Output: inpainted image

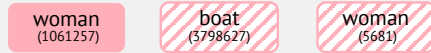
Inst-Inpaint – Dataset Generation

Training Data Generation for GQA-Inpaint

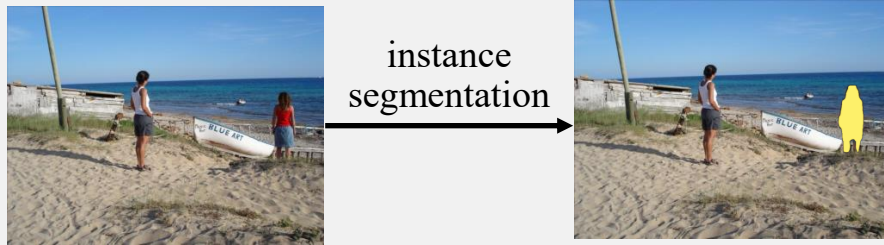


An image from GQA dataset and its scene graph

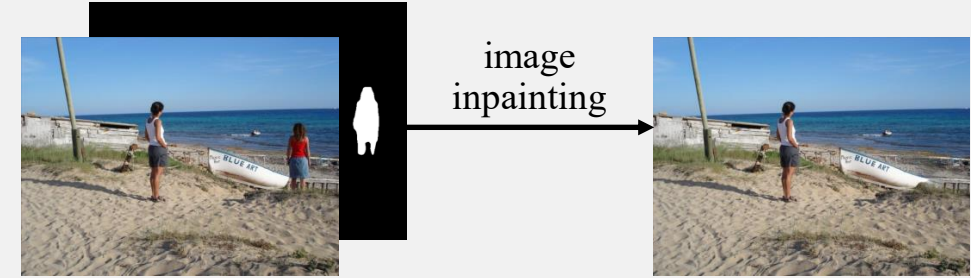
(a) Selecting an object from the scene graph:



(b) Extracting segmentation mask:



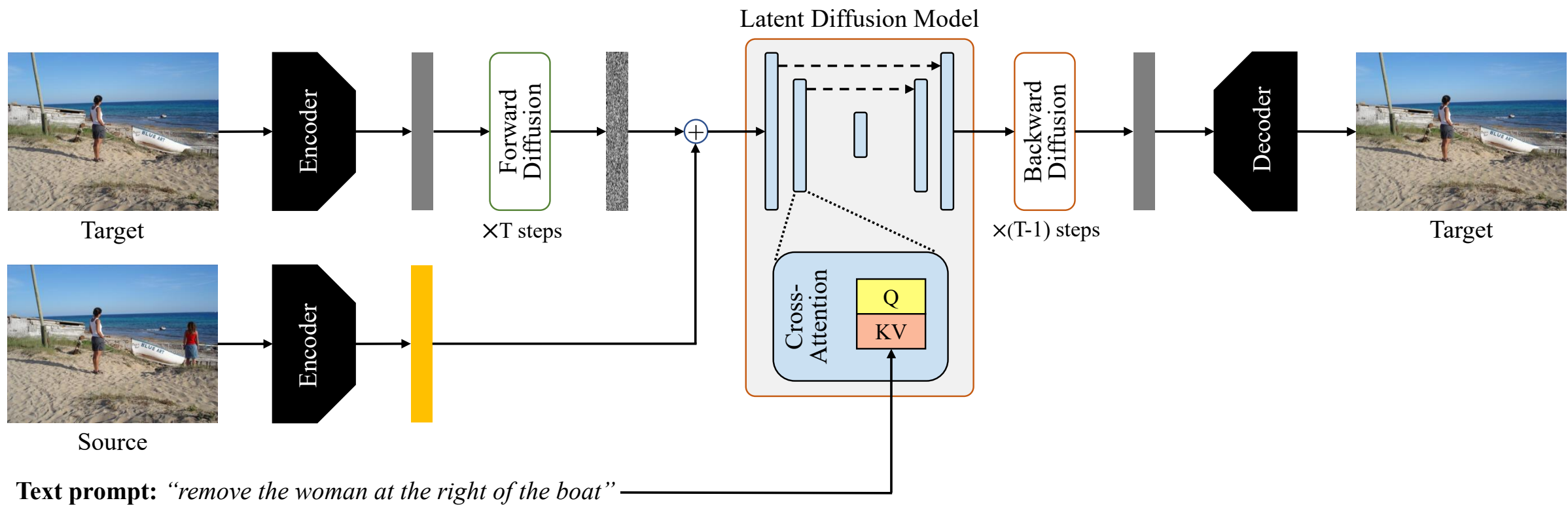
(c) Removing object from the image:



(d) Generating textual prompt:
“remove the woman at the right of the boat”

Inst-Inpaint – Model Training

Training Inst-Inpaint for Instructional Image Inpainting



Instruction-Based Object Removal



Instruction-Based Object Removal



remove the gray kite at the left



remove the street light at the left



remove the man at the right of the man



remove the red car at the left of the tall ladder



remove the colorful train at the right



remove the boat at the right of the small boat

Instruction-Based Object Removal

Source

InstPix2Pix

X-Decoder

CLIPSeg

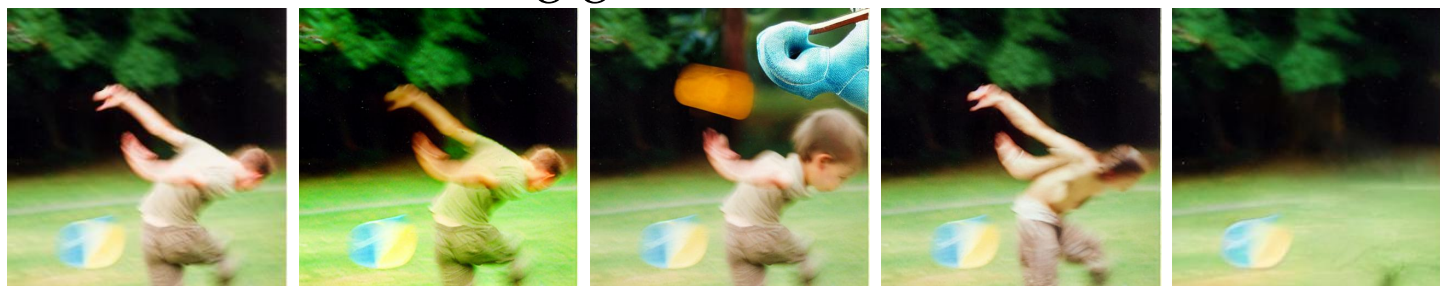
Ours



➔ Remove the gray kite



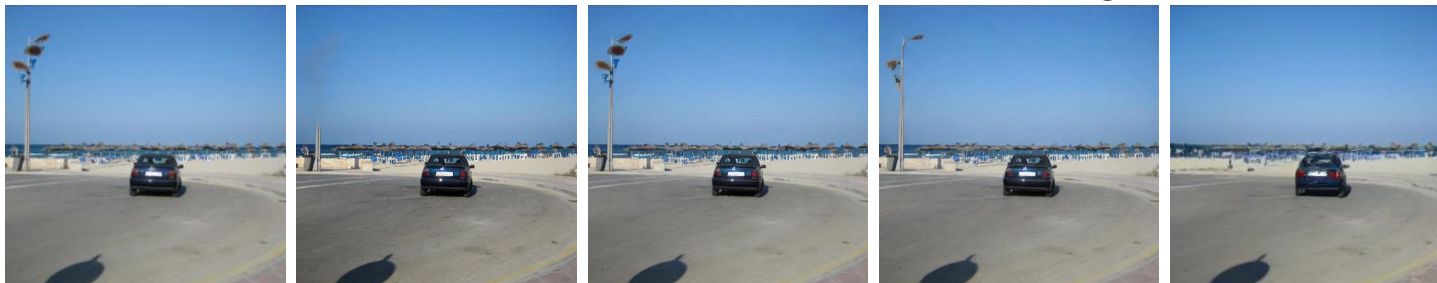
➔ Remove the surfing girl on the blue surfboard



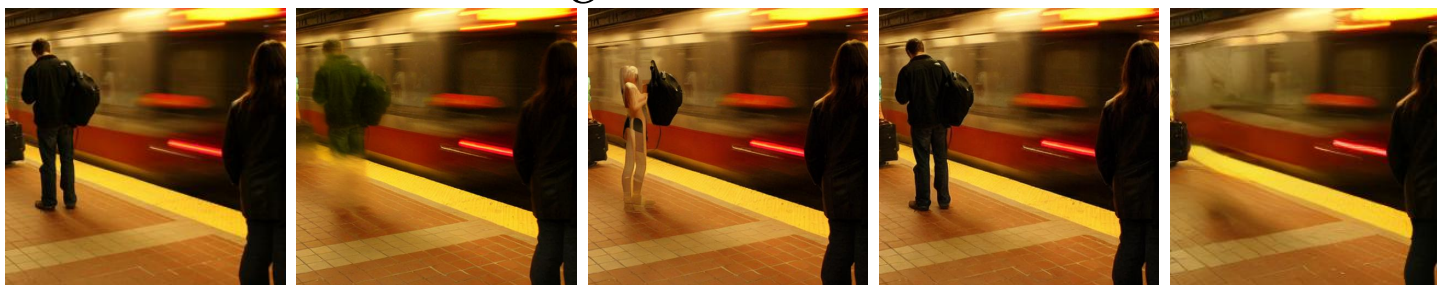
➔ Remove the boy

Instruction-Based Object Removal

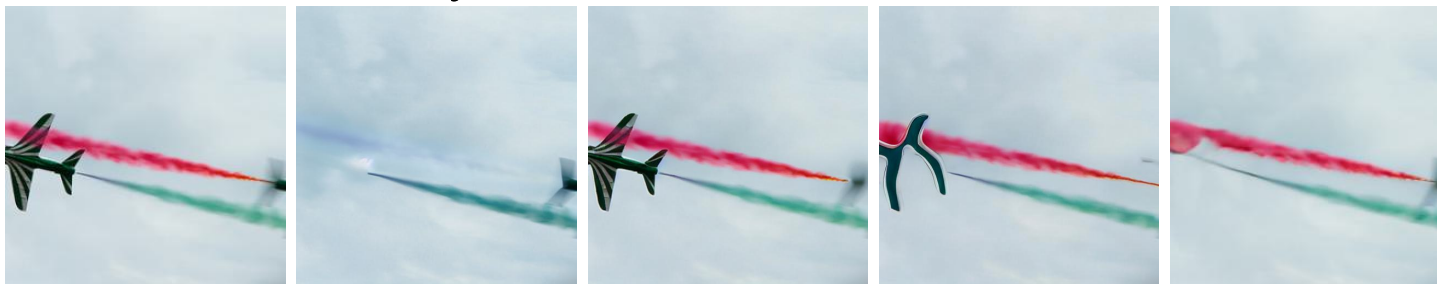
Source InstPix2Pix X-Decoder CLIPSeg Ours



➔ Remove the street light at the left



➔ Remove the boy at the left



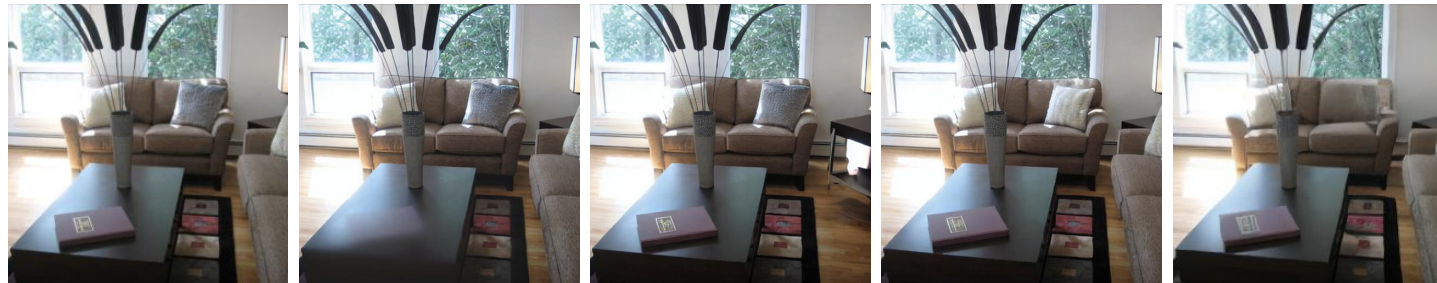
➔ Remove the airplane at the left

Instruction-Based Object Removal

Source InstPix2Pix X-Decoder CLIPSeg Ours



➔ Remove the television at the center

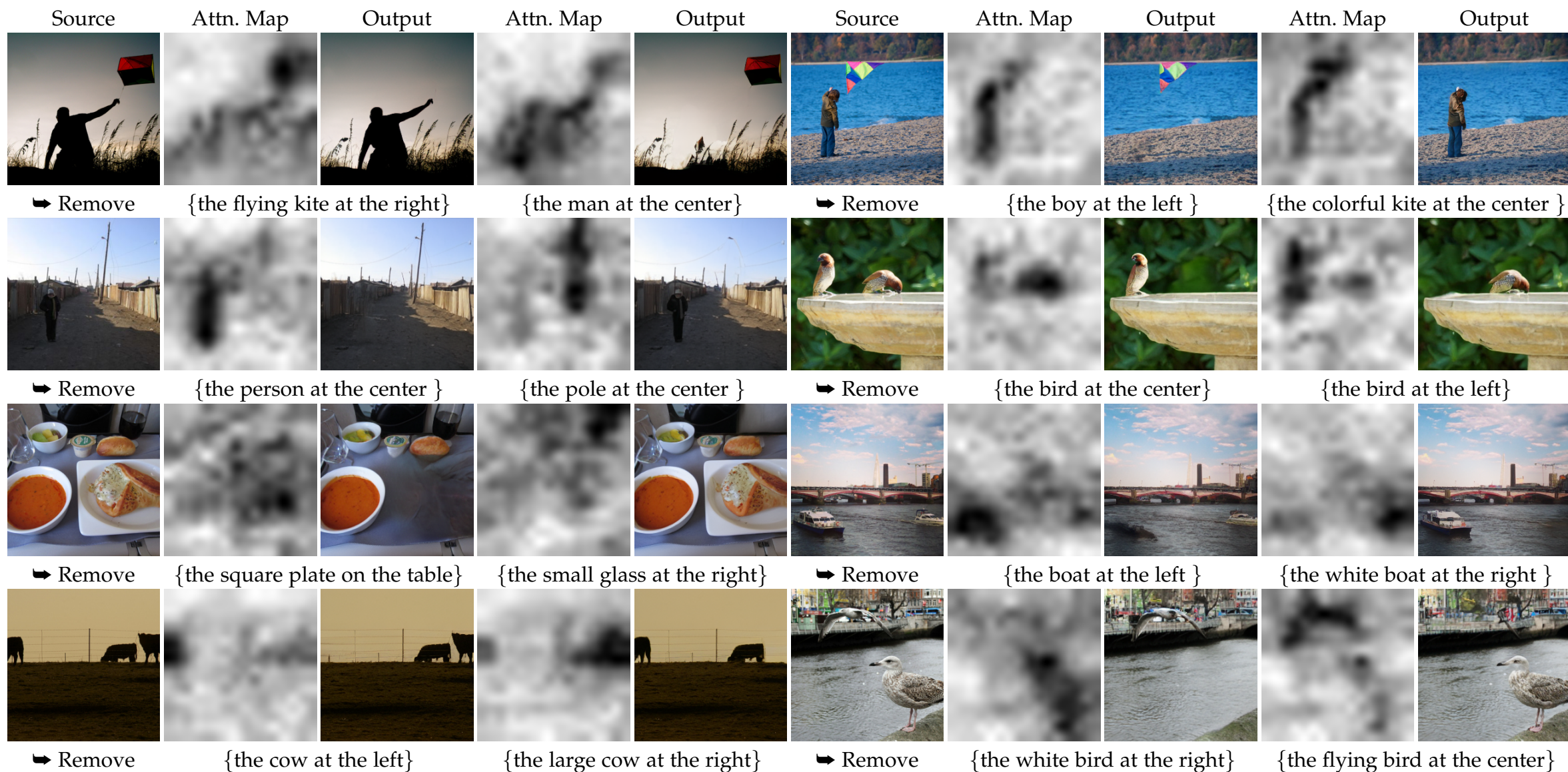


➔ Remove the pillow at the right

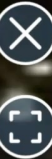


➔ Remove the man

Instruction-Based Object Removal



360° Panoramic Image Generation



GENERATION PROMPT:

The image depicts a nighttime scene of a European town square. The square is paved with dark, glossy tiles that reflect the lights from the surrounding buildings and street lamps. The buildings are multi-storied and painted in various pastel colors, including shades of pink, yellow, and green. The architecture is traditional, with shuttered windows and balconies adorned with plants and flowers. In the center of the square, there is a fountain with water spouting from its top, surrounded by benches for people to sit and enjoy the view. The square is illuminated by several street lamps that cast a warm glow on the surroundings. There are also some bicycles parked along the edges of the square. The sky above is dark, indicating that it is nighttime. The overall atmosphere of the scene is peaceful and serene, with no people visible in the image.



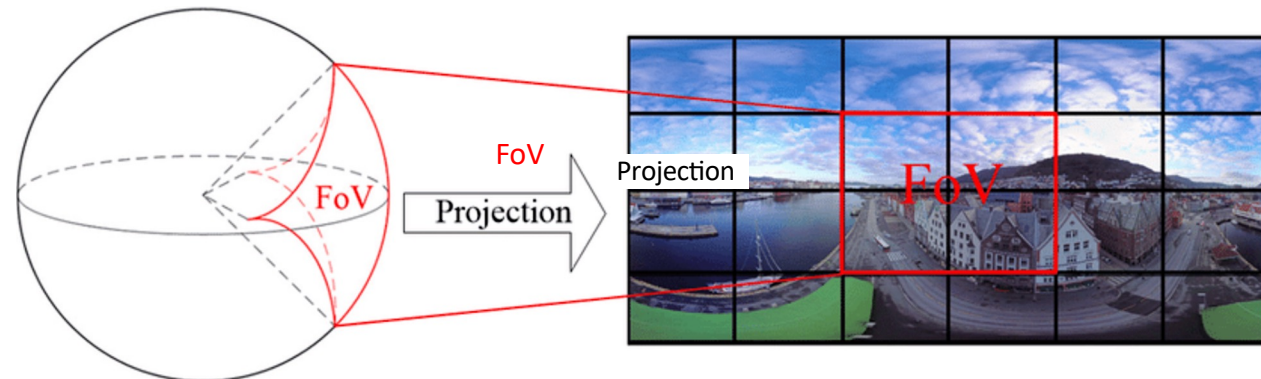
Omnidirectional Images



- Omnidirectional images (360° images) are becoming more and more popular
- Applications in virtual reality, robotics, and surveillance
- A new form of visual data

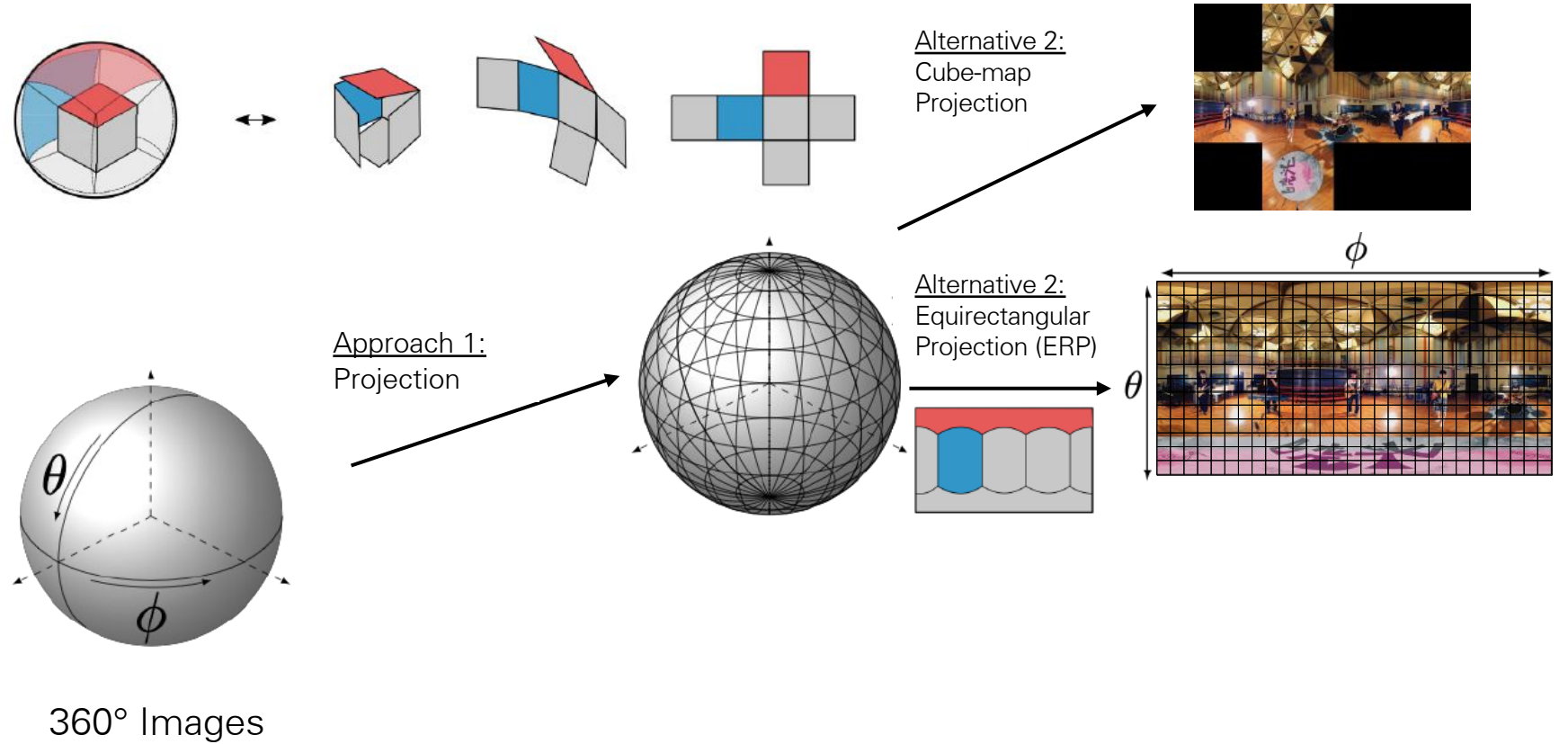
Challenges in 360° Vision

- Spherical distortion
- Limited training data
- Input size (+4K images/videos)



Representing Omnidirectional Images

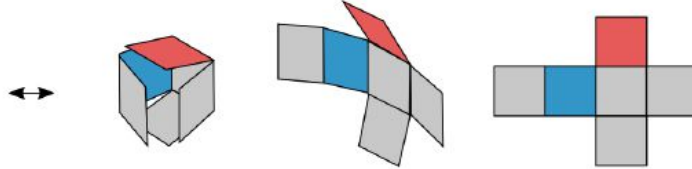
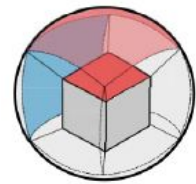
- Approach 1:
Mapping onto 2D
 - Cube-map Projection
 - Equirectangular Projection



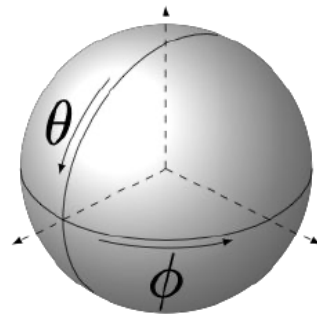
Representing Omnidirectional Images

- Approach 1:
Mapping onto 2D

- Cube-map Projection
- Equirectangular Projection

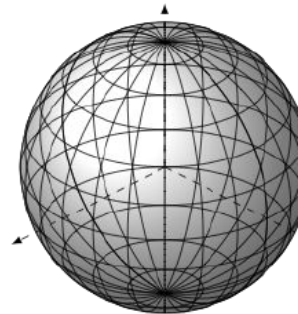


Alternative 2:
Cube-map
Projection

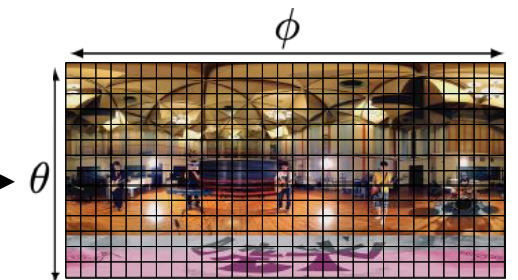
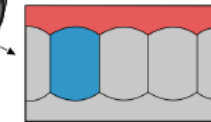


360° Images

Approach 1:
Projection

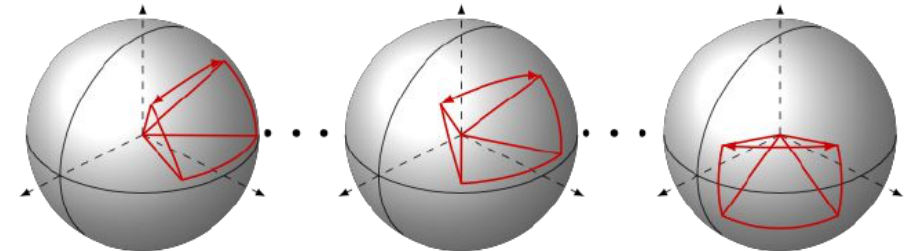


Alternative 2:
Equirectangular
Projection (ERP)



- Approach 2:
Extracting
multiple
viewports

Approach 2:
Extracting
viewports



Viewport 1



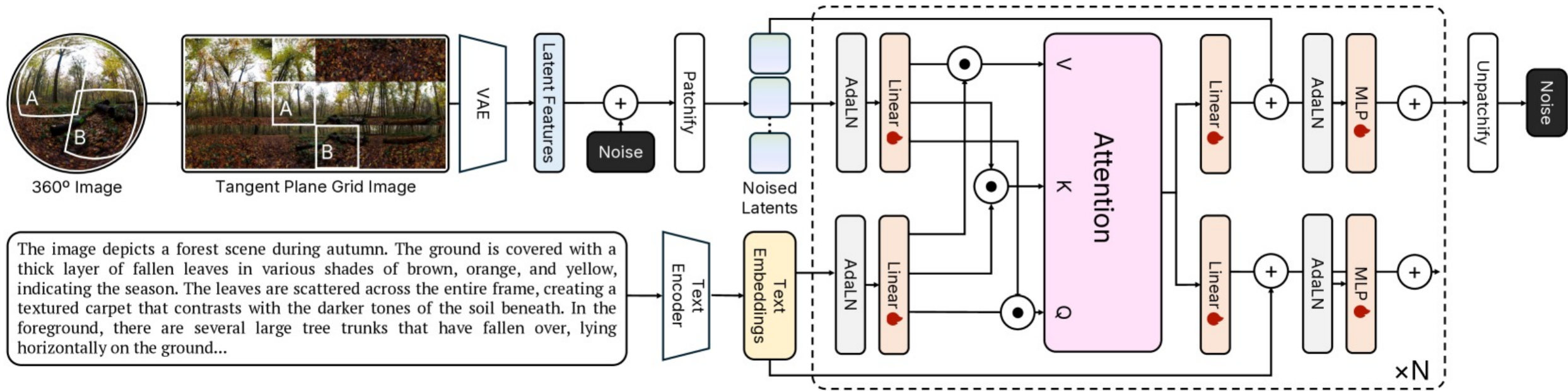
Viewport 2

...



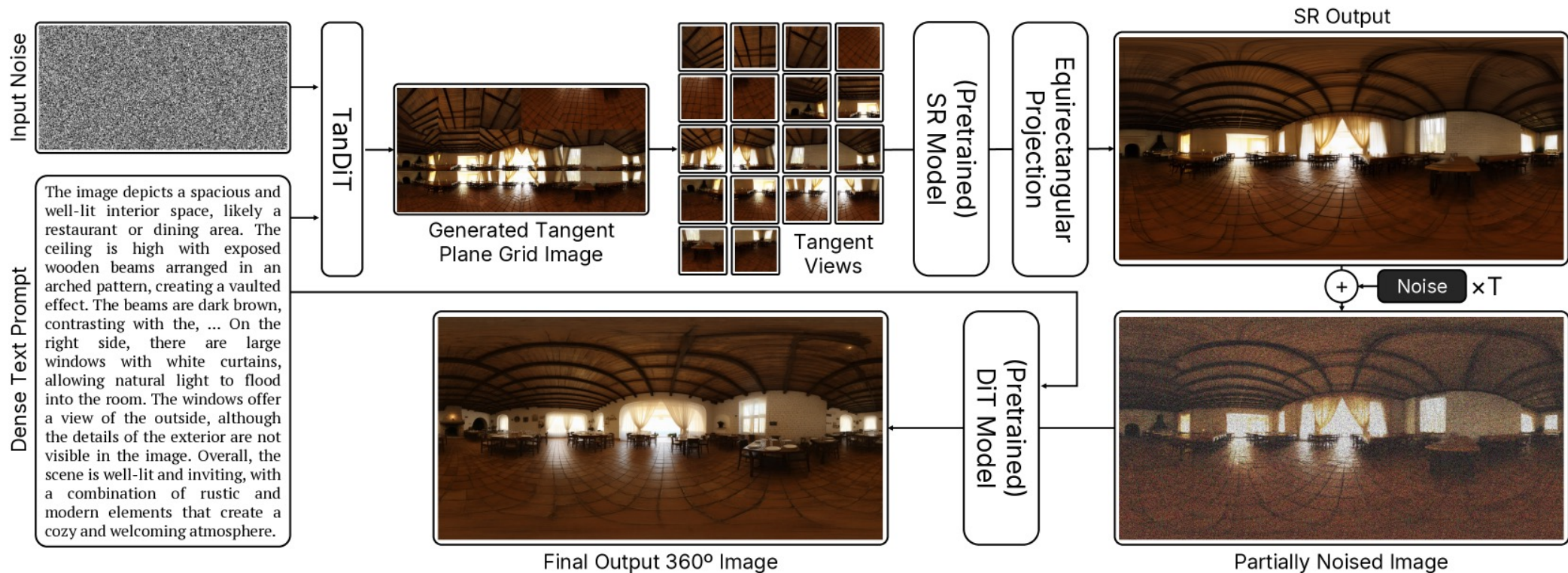
Viewport n

TanDiT Training Pipeline



- Convert 360° panoramas into a grid of tangent-plane crops (gnomonic projection).
- Arrange overlapping crops adjacently for spatial consistency.
- Use dense captions to enrich scene context.
- Train with a latent-space denoising diffusion objective.

TanDiT Inference Pipeline



- Generate a grid of tangent-plane views conditioned on the text prompt.
- Upscale the views and reproject them into an intermediate equirectangular panorama.
- Add mild latent noise and refine the panorama using a pre-trained DiT model conditioned on the same prompt.
- Output a high-quality, high-resolution 360° image.

360° Panorama Generation



GENERATION PROMPT:

The panoramic image depicts a modern, open-plan living space that seamlessly integrates the kitchen, dining area, and living room. The kitchen features sleek white cabinetry with black countertops, complemented by a large island in the center of the room. Above the island, there are three pendant lights with a metallic finish, casting a warm glow over the space. The backsplash is made of dark tiles, providing a striking contrast to the white cabinets. Adjacent to the kitchen is the dining area, which includes a rectangular table surrounded by four chairs. The table is set with a few plates and cutlery, suggesting a meal might be about to take place or has just concluded. A large window with a wooden frame allows natural light to flood into the room, enhancing the overall brightness of the space. The living room area is furnished with a large, L-shaped sectional sofa in a dark gray color. The sofa is adorned with several throw pillows in shades of yellow and white, adding a pop of color to the otherwise neutral palette. In front of the sofa, there is a low coffee table with a glass top, upon which rests a few decorative items. The walls of the living room are painted in a light beige color, creating a warm and inviting atmosphere. The flooring throughout the space is composed of large, light-colored tiles that reflect the ample natural light, contributing to the room's airy feel. The ceiling is high, with recessed lighting installed to provide additional illumination when needed. In the background, there is a hallway leading to other parts of the house, with a door on the right side. The hallway is lined with a few framed pictures and a small console table against the wall. The overall design of the space is contemporary, with clean lines and a minimalist aesthetic.

360° Panoramic Image Generation



GENERATION PROMPT:

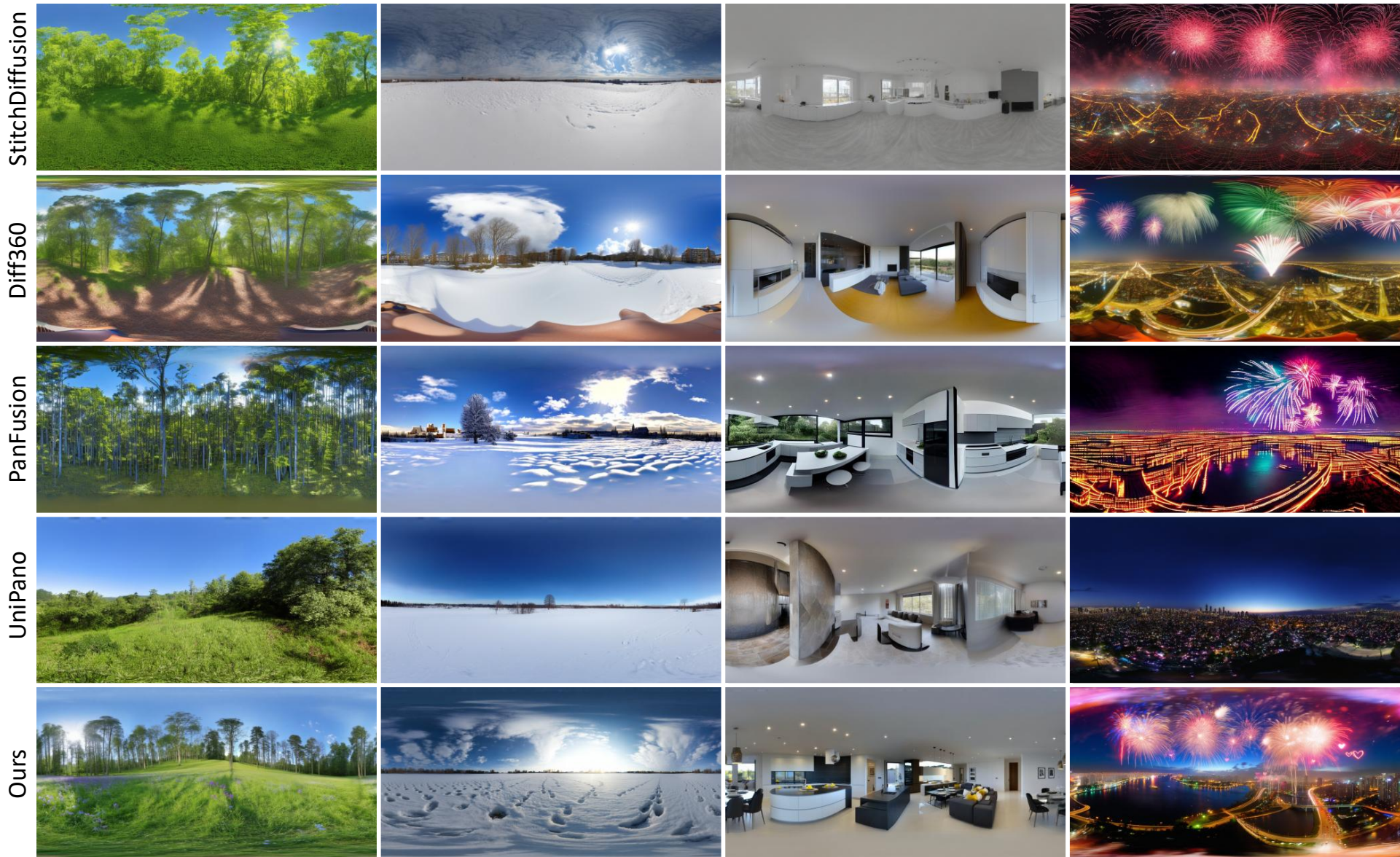
Rolling hills, a red barn, grazing cows, and sunflowers swaying in the breeze. The sun sets low behind the trees, casting golden light over everything.

STYLE:

Papercraft Diorama



Qualitative Comparisons



- ERP-based methods, StitchDiffusion, Diffusion360, show polar distortions and seam artifacts.
- PanFusion and UniPano lack fine detail.
- TanDiT generates globally coherent spherical images with sharper details and pole/seam consistency.

Qualitative Comparisons

StitchDiffusion



Diff360



We ran a user study with 30 participants viewing 33 rotating 360° image pairs, and TanDiT was chosen as the **more realistic** option in most comparisons.

UniPano



Ours



- ERP-based methods, StitchDiffusion, Diffusion360, show polar distortions and seam artifacts.
- PanFusion and UniPano lack fine detail.
- TanDiT generates globally coherent spherical images with sharper details and pole consistency.

Next Lecture: Visual Quality Assessment