Generative models

Outline

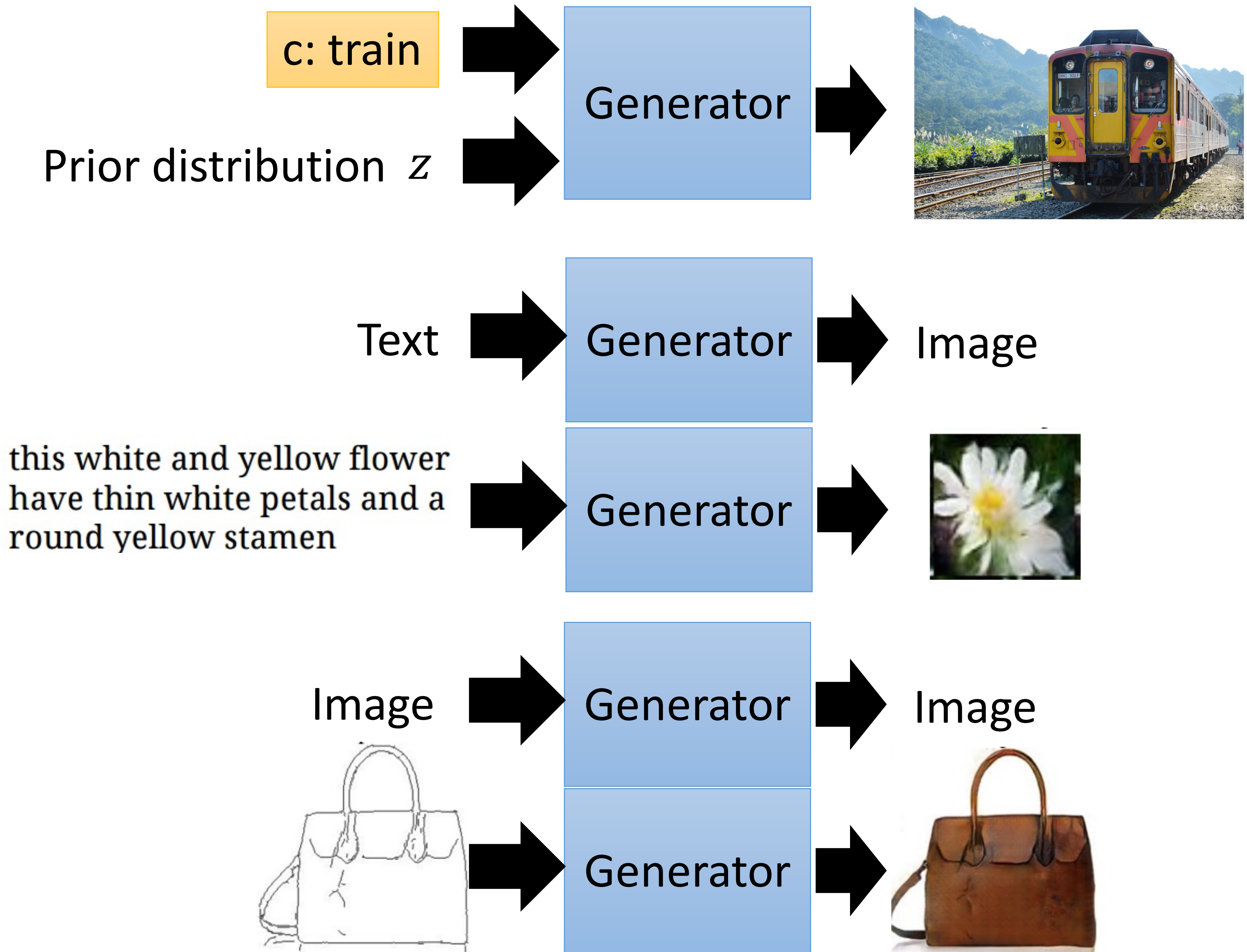
1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. **Conditional GANs**

Generative models

Outline

1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. **Conditional GANs**
    1. **Principle**

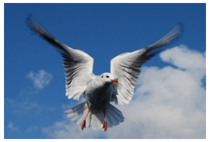
Motivation



c: train
Prior distribution z
Generator
Text
Generator
Image
this white and yellow flower have thin white petals and a round yellow stamen
Generator
Image
Generator
Image
Generator

Conditional GAN

c^1: a dog is running $\hat{x}^1$: 

c^2: a bird is flying $\hat{x}^2$: 

- ***Text to image*** by traditional supervised learning

c^1: a dog is running → NN → Image ↔ $\hat{x}^1$: 

as close as possible

Text: "train"

Target of NN output

A blurry image!

Conditional GAN

c: train

Prior distribution z

G

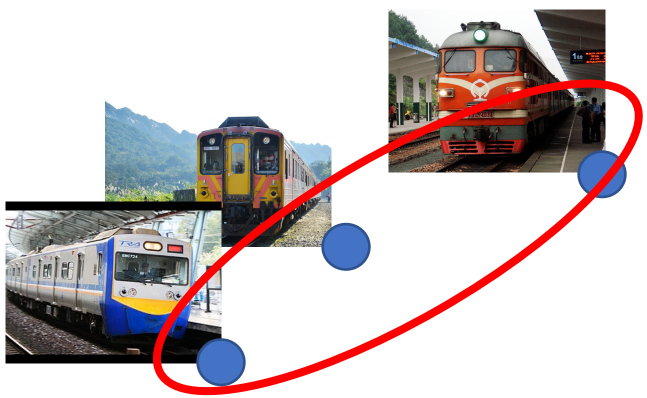
Image

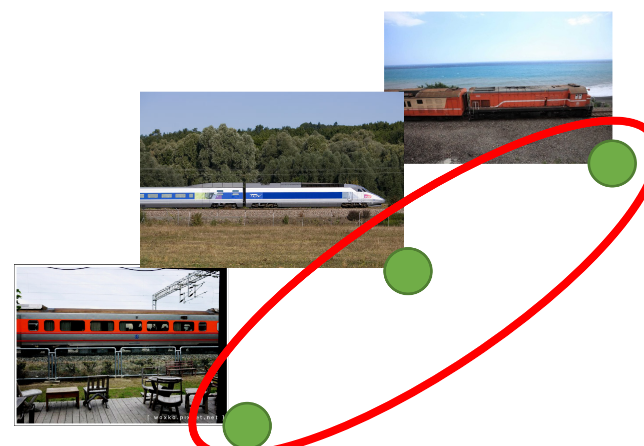
x = G(c,z)

It is a distribution

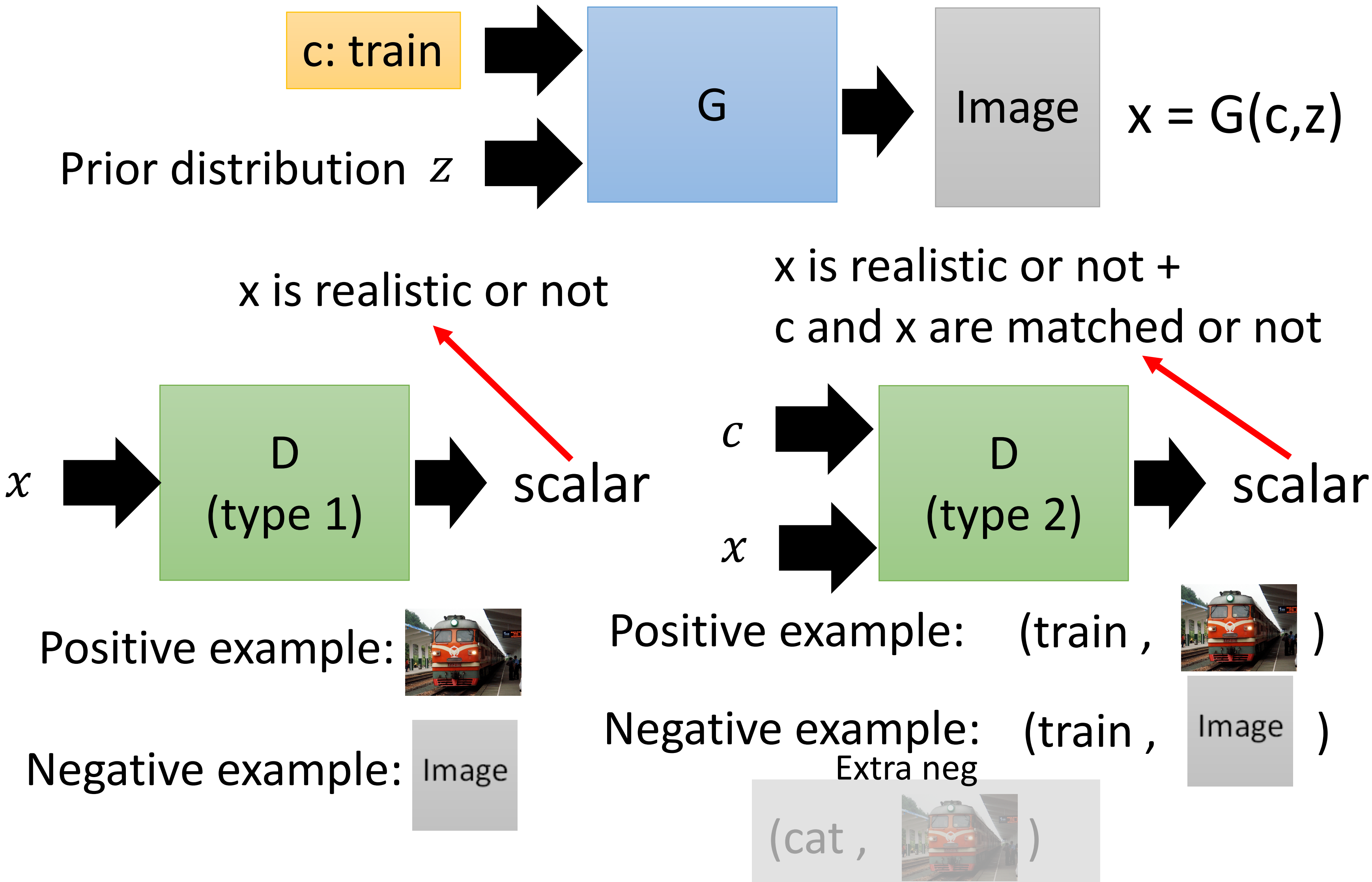
Approximate the distribution of real data

Text: “train”





Conditional GAN



Conditional GAN (cGAN model)

GAN

$$V(G, D) = \mathbb{E}_{x \sim P_{data}}[log D(x)] + \mathbb{E}_{x \sim P_G}[log(1 - D(x))]$$

$$G^* = arg \min_G \max_D V(G, D)$$

cGAN

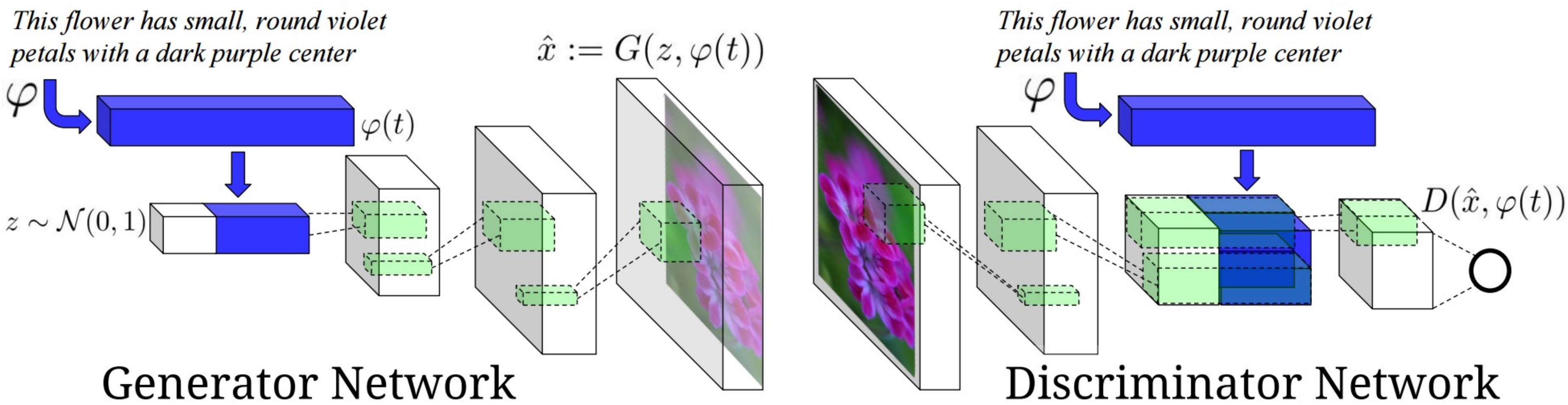
$$\min_G \max_D \left(\mathbb{E}_{\mathbf{x},\mathbf{y} \sim p_{\text{data}}(\mathbf{x},\mathbf{y})} [\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{y} \sim p_{\mathbf{y}},\ \mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z}, \mathbf{y}), \mathbf{y}))] \right)$$

Generative models

Outline

1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. Conditional GANs
    1. Principle
    2. **Text2Image**

Text2Image: architecture example



- Positive samples:
  - real image + right texts
- Negative samples:
  - fake image + right texts
  - Real image + wrong texts

Text2Image results

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma

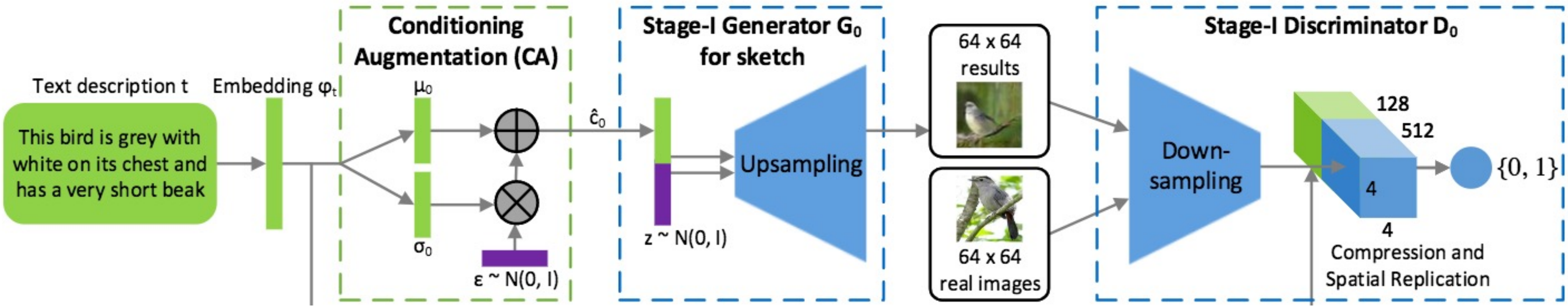
this white and yellow flower have thin white petals and a round yellow stamen

Text2Image results

Caption	Image
this flower has white petals and a yellow stamen	
the center is yellow surrounded by wavy dark purple petals	
this flower has lots of small round pink petals	

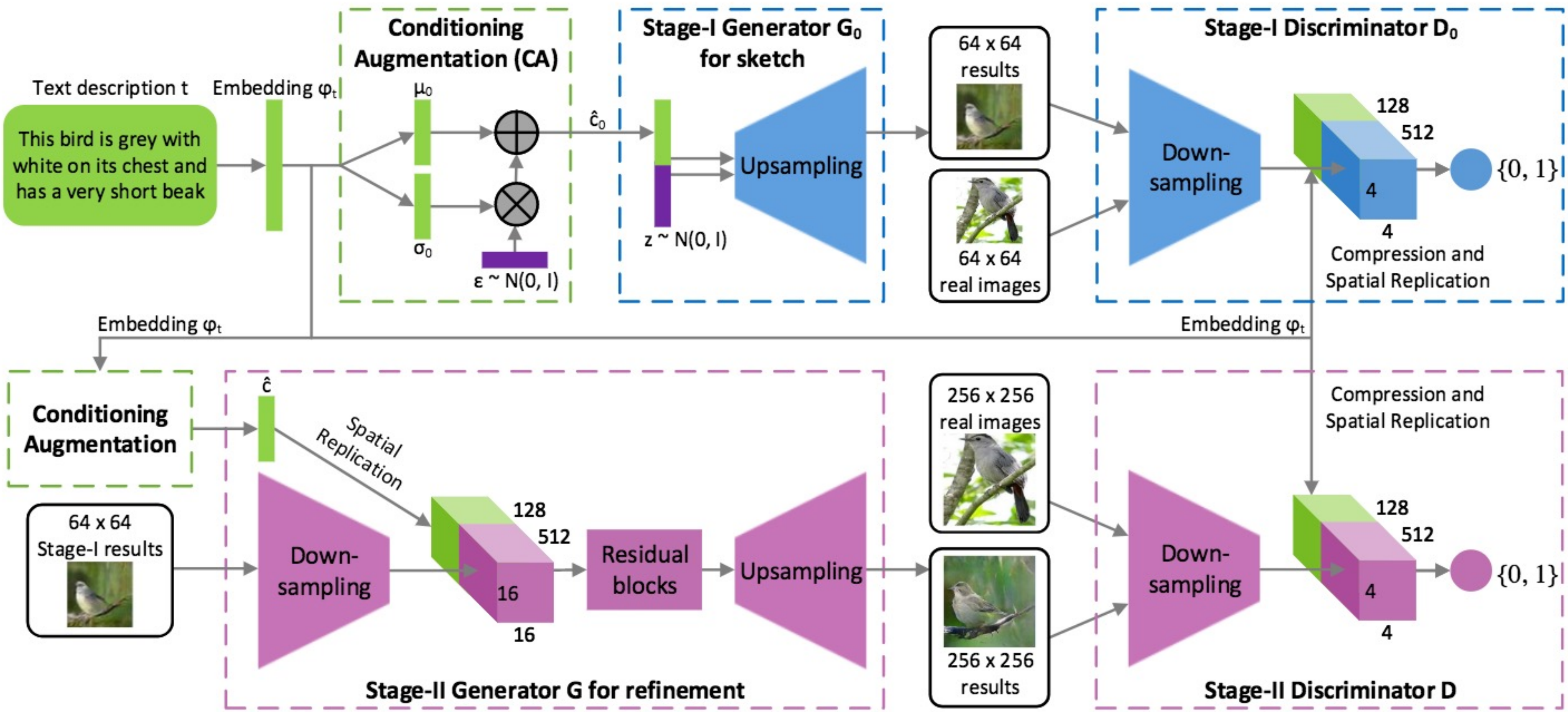
Text2Image: architecture example (2)

Generating higher resolution images (from 64 to 256)

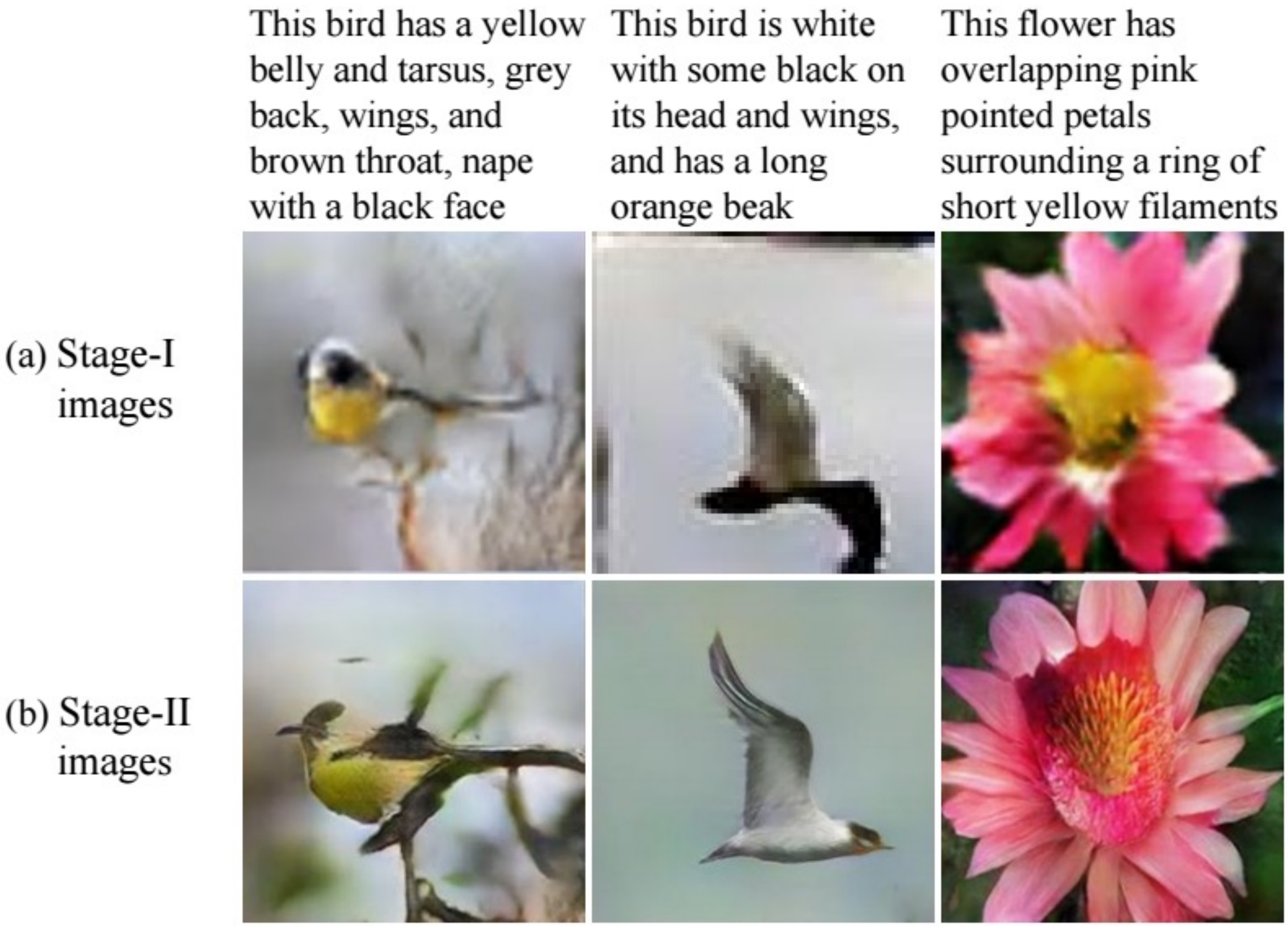


Text2Image: architecture example (2)

Generating higher resolution images (from 64 to 256)



StackGAN results



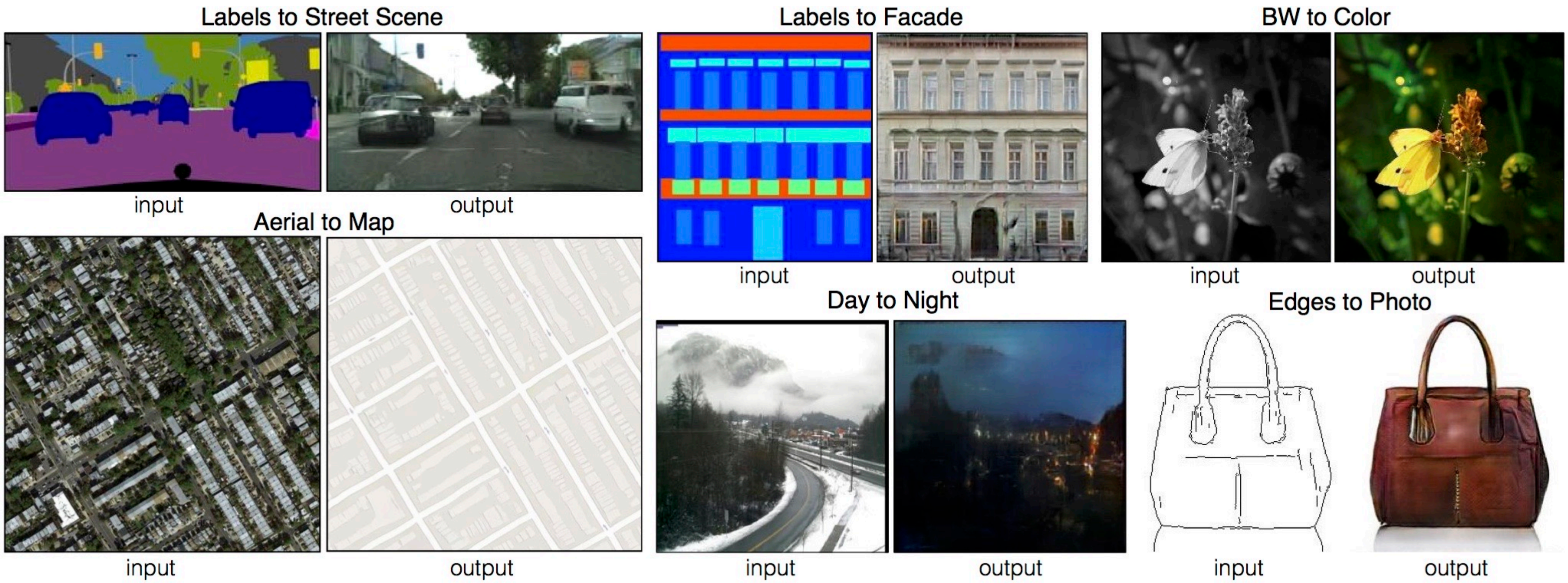
[Zhang et al. 2016]

Generative models

Outline

1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. Conditional GANs
    1. Principle
    2. Text2Image
    3. **Image2Image**

Image-based Conditional GAN



- Conditioned on an image of different modality
- Image-to-Image Translation => pix2pix

Image-to-image pix2pix

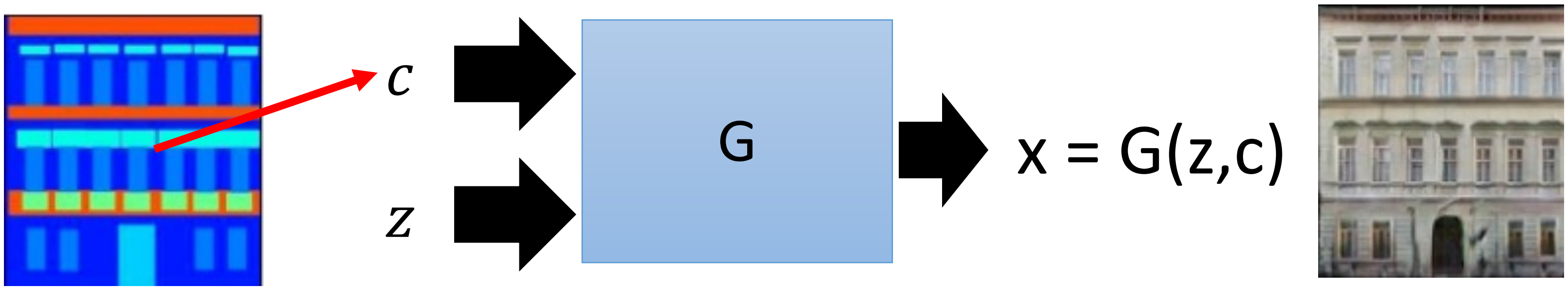
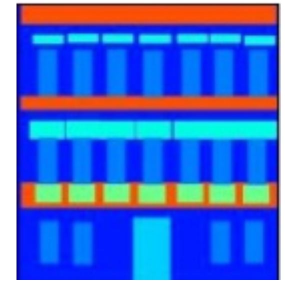
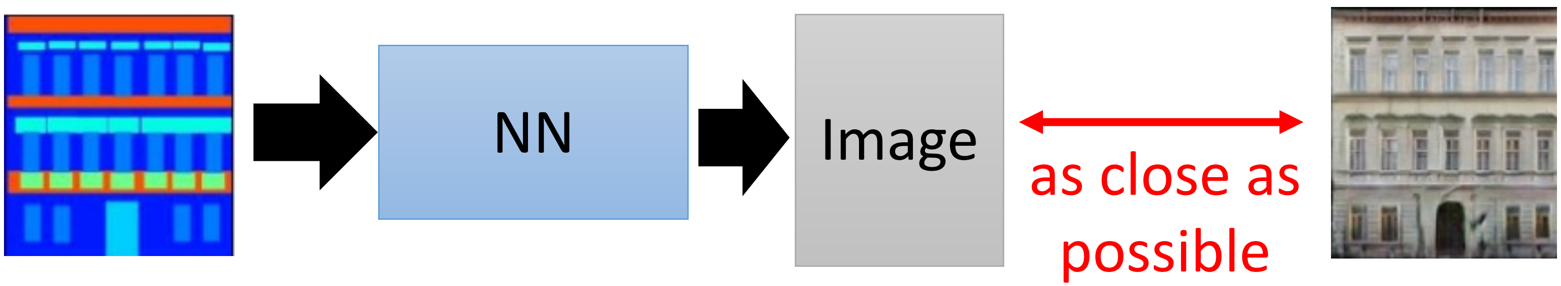


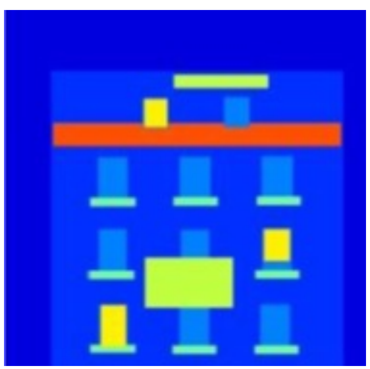
Image-to-image pix2pix



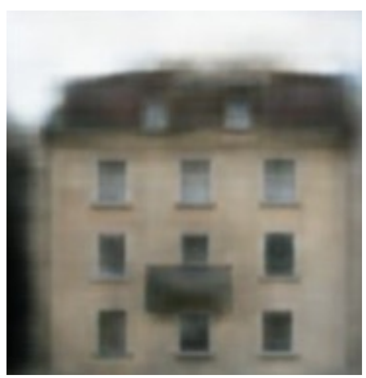
- Traditional supervised approach



Testing:



input



close

It is blurry because it is the average of several images.

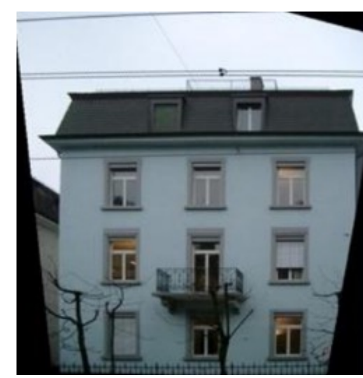
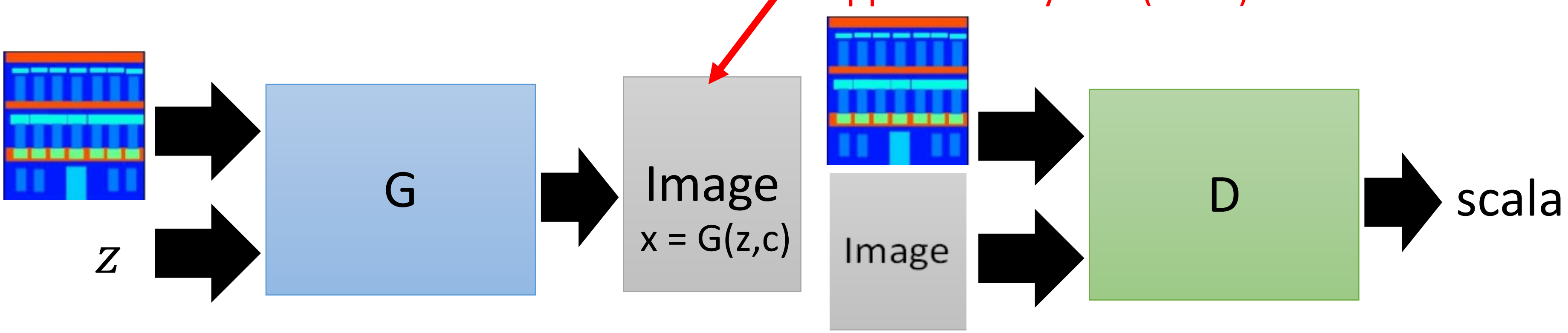


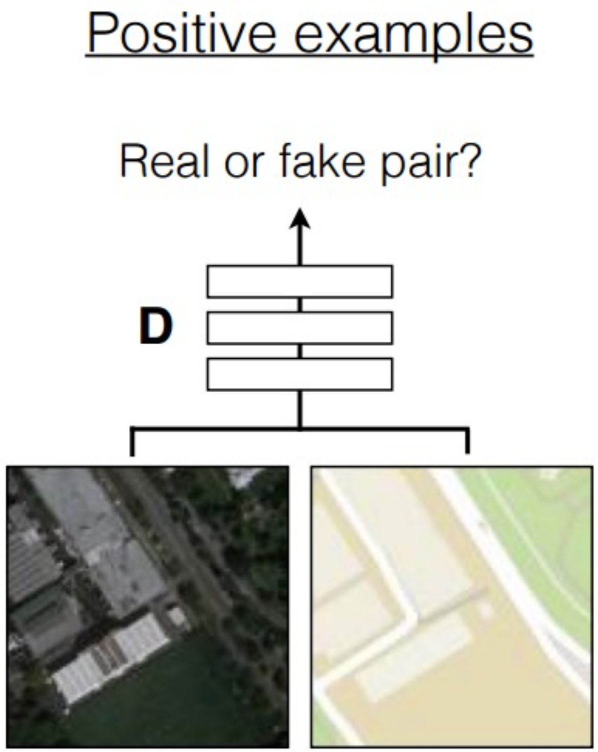
Image-to-image

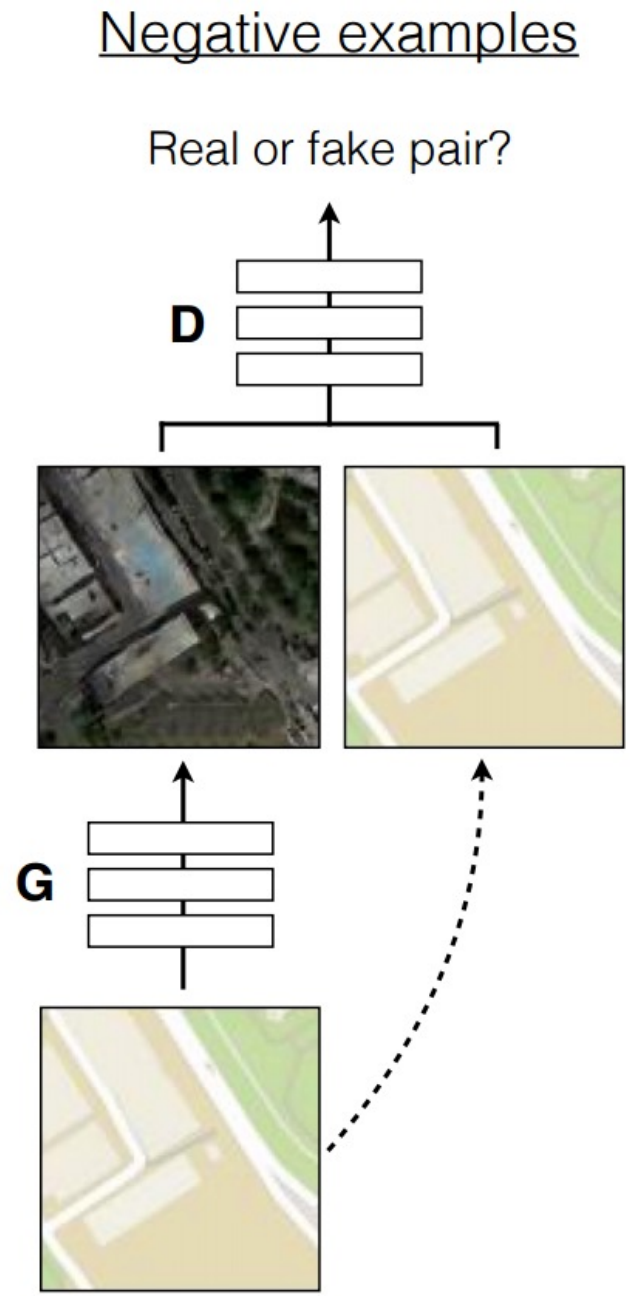
- Conditional GAN



Testing:

input | close | GAN | GAN + close | GT

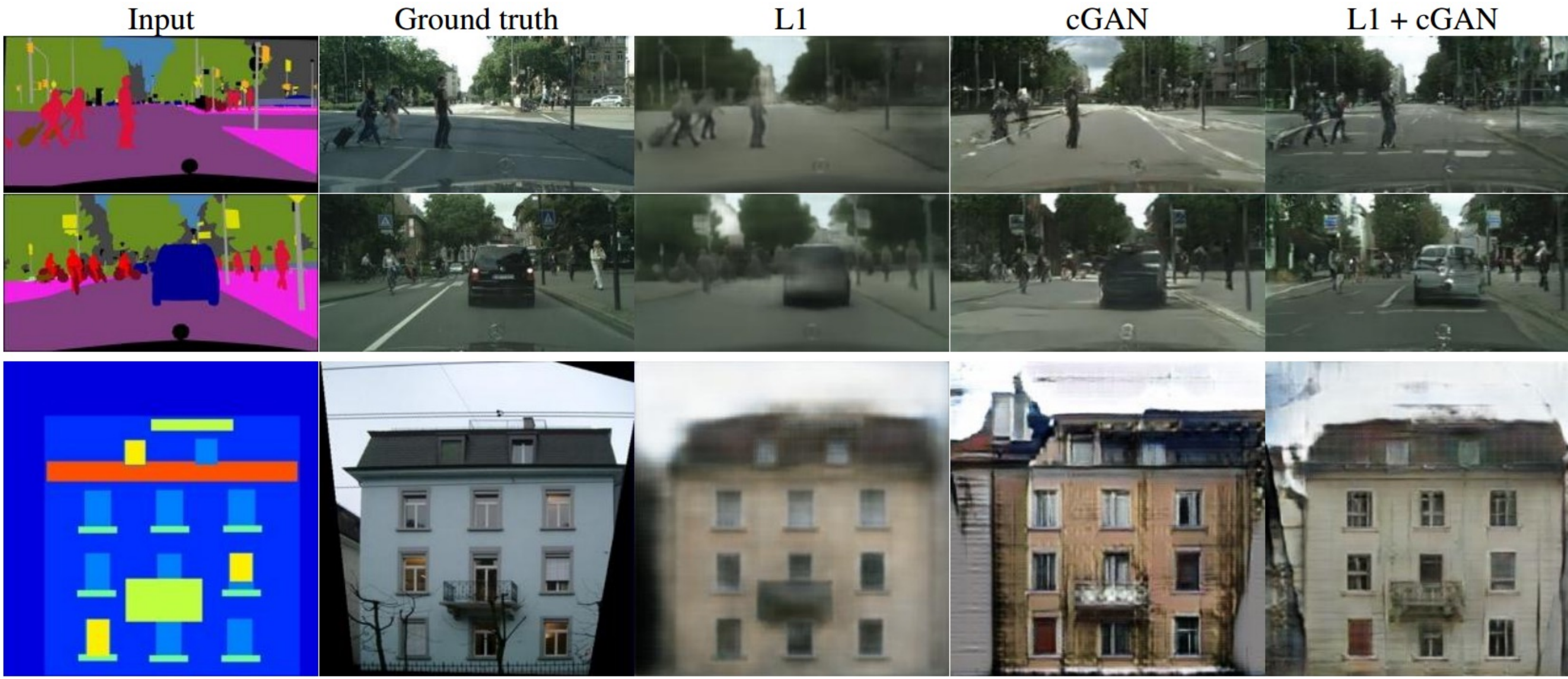




G tries to synthesize fake images that fool **D**

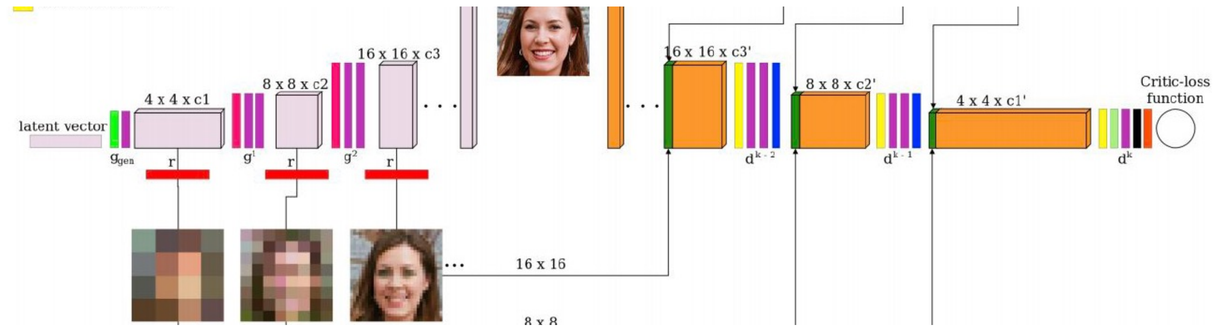
D tries to identify the fakes

Label2Image

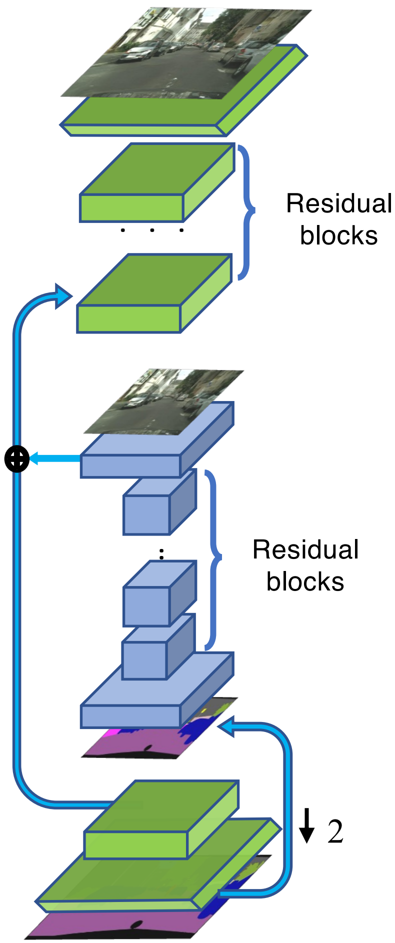


Edges2Image

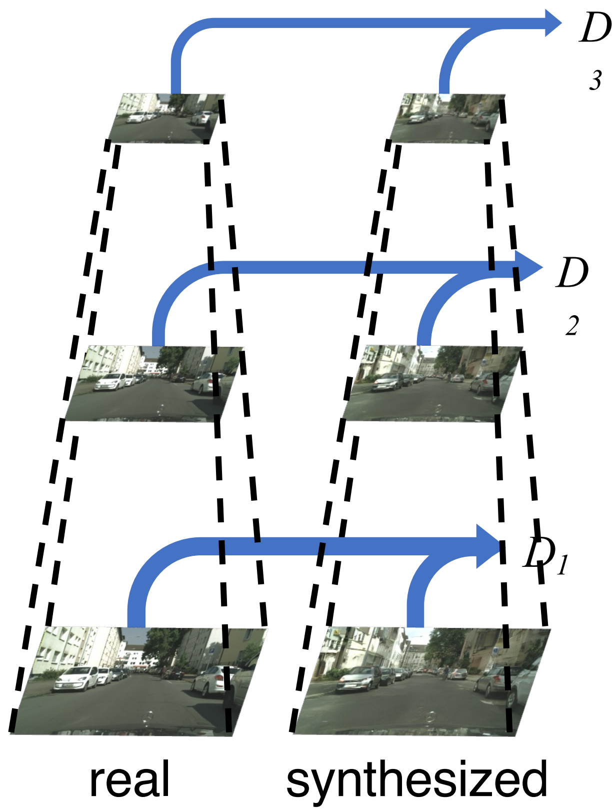
Pix2pixHD



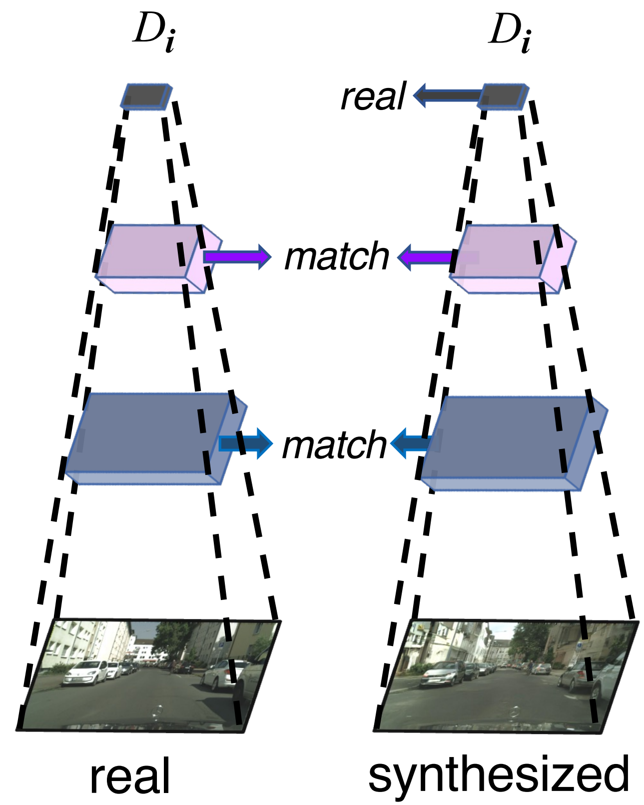
Coarse-to-fine Generator

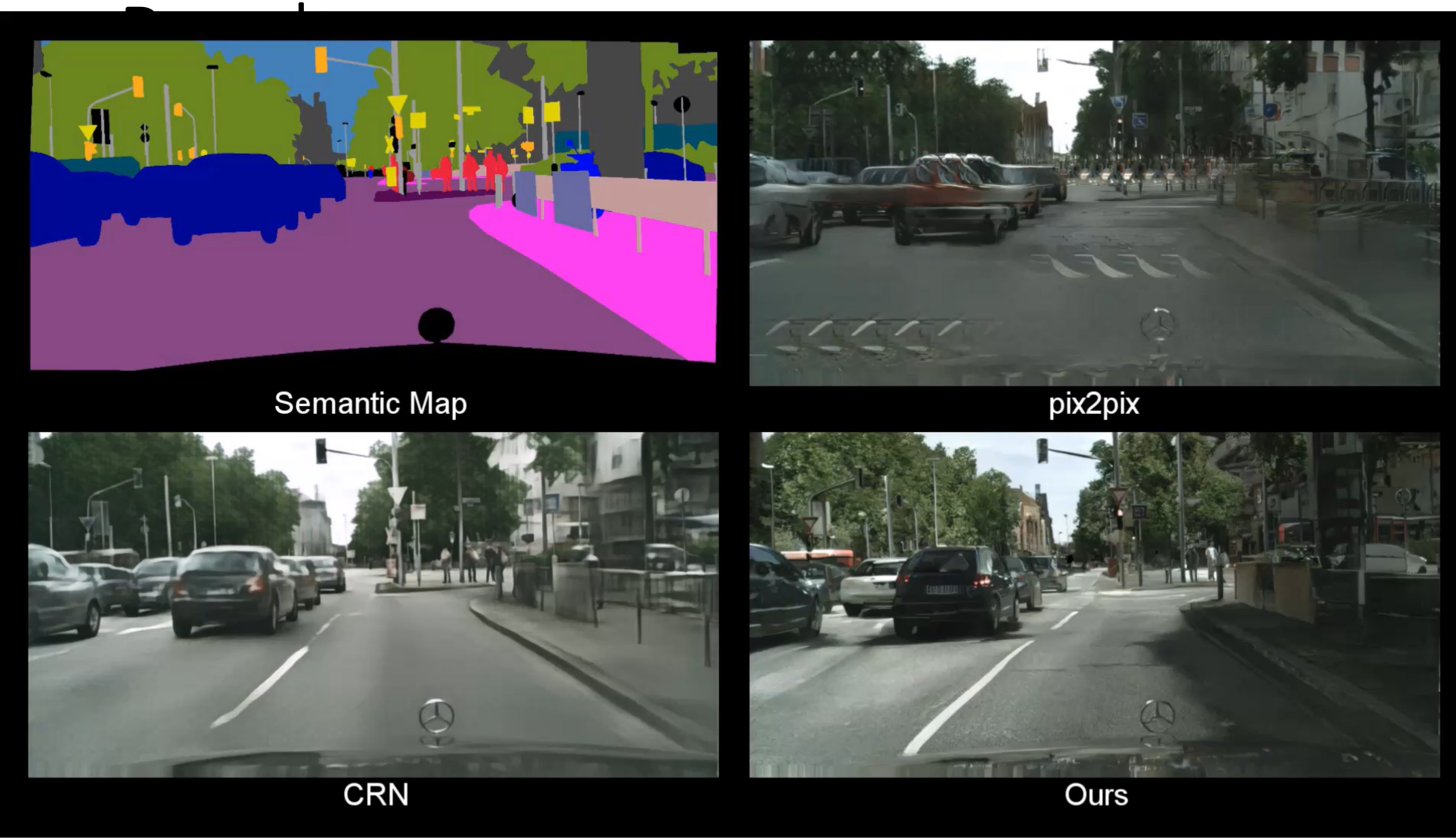


Multi-scale Discriminators



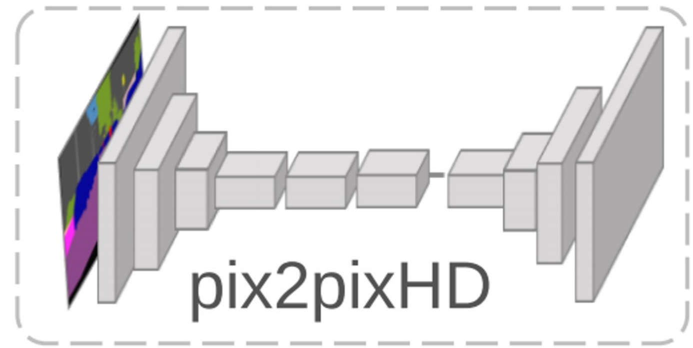
Robust Objective




Semantic Map
pix2pix
CRN
Ours

Improving Segmentation2Image strategy?

Limitation of the approach:



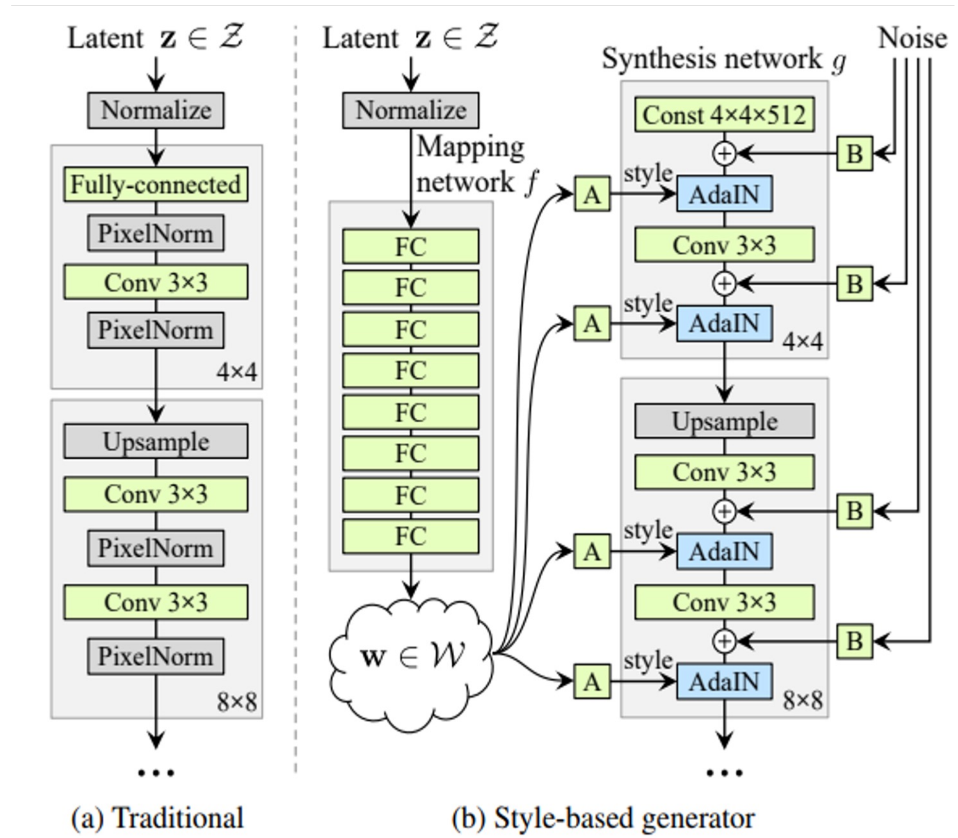
Directly feed the semantic layout as input to the deep network, which is processed through stacks of convolution, normalization, and nonlinearity layers.

However, this is suboptimal as the normalization layers tend to “wash away” semantic information in input semantic segmentation masks.

Improving Segmentation2Image strategy

Proven effective for recent generative adversarial networks such as StyleGAN

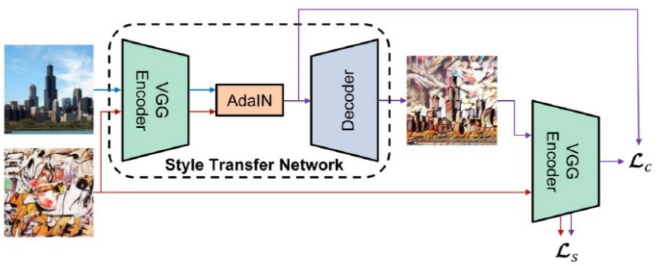
Can we do the same for conditional GAN?
Conditional Normalization Layers?



Improving Segmentation2Image strategy

Recall: Adaptive instance normalization

$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$



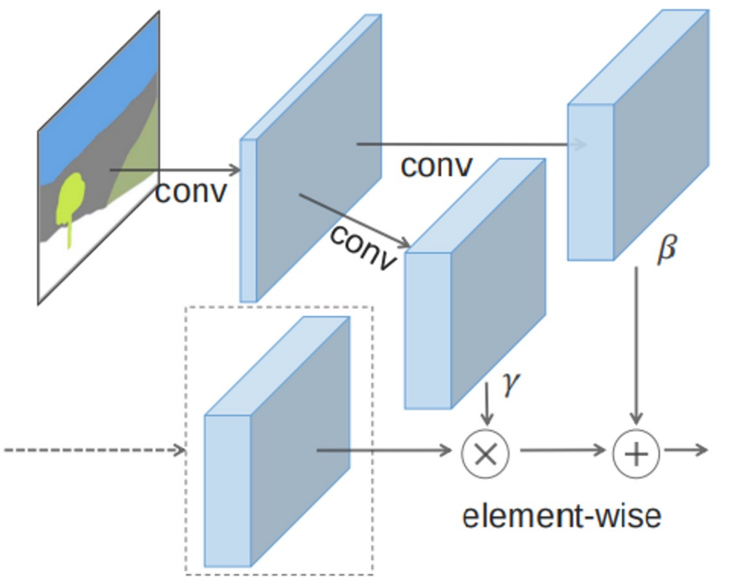
SPADE block= spatially-adaptive denormalization:

Same idea but per class c over each channel i (N=batch size)

$$\gamma^i_{c,y,x}(\mathbf{m})\frac{h^i_{n,c,y,x} - \mu^i_c}{\sigma^i_c} + \beta^i_{c,y,x}(\mathbf{m})$$

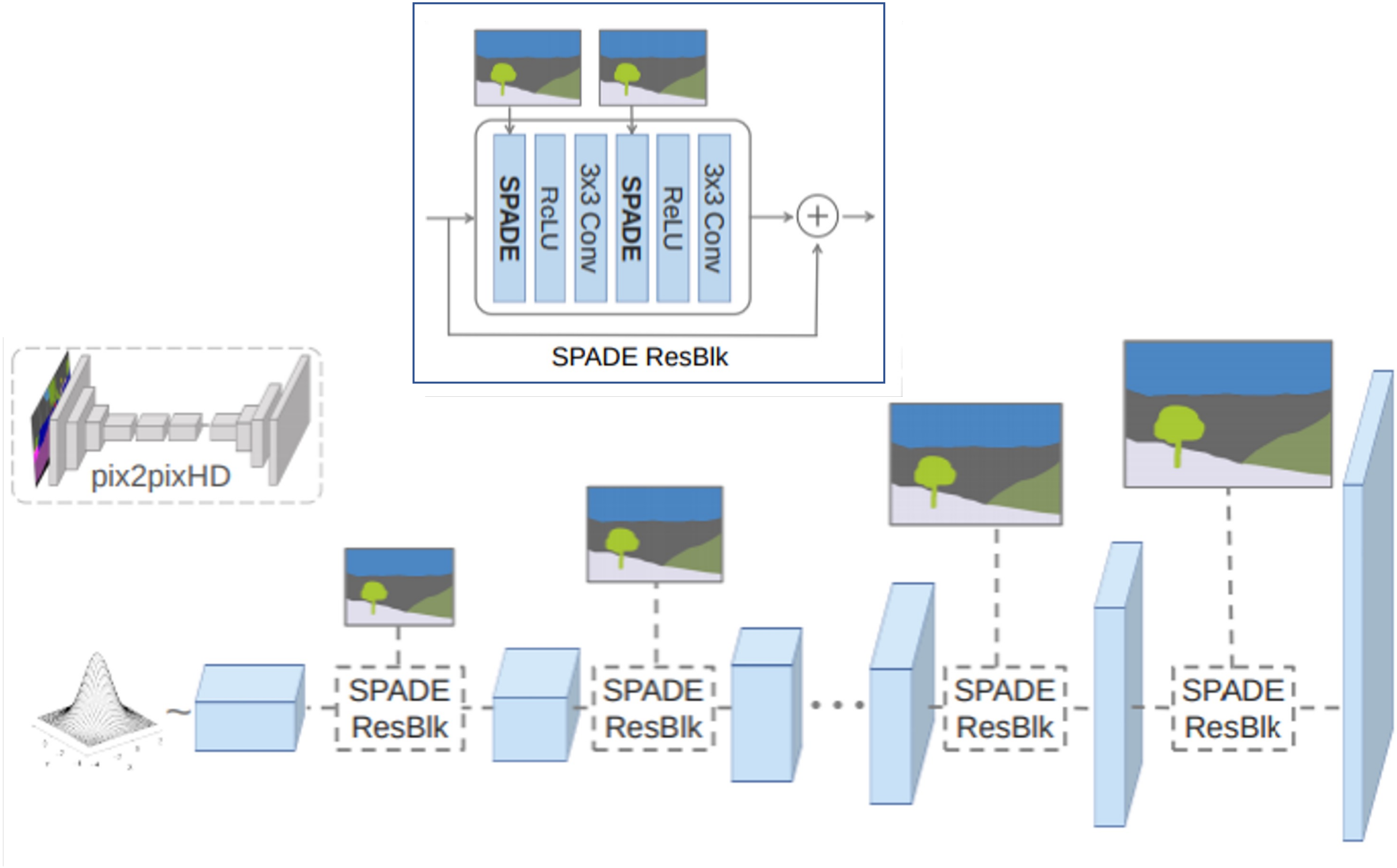
$$\mu^i_c = \frac{1}{NH^iW^i}\sum_{n,y,x} h^i_{n,c,y,x}$$

$$\sigma^i_c = \sqrt{\frac{1}{NH^iW^i}\sum_{n,y,x}(h^i_{n,c,y,x})^2 - (\mu^i_c)^2}$$

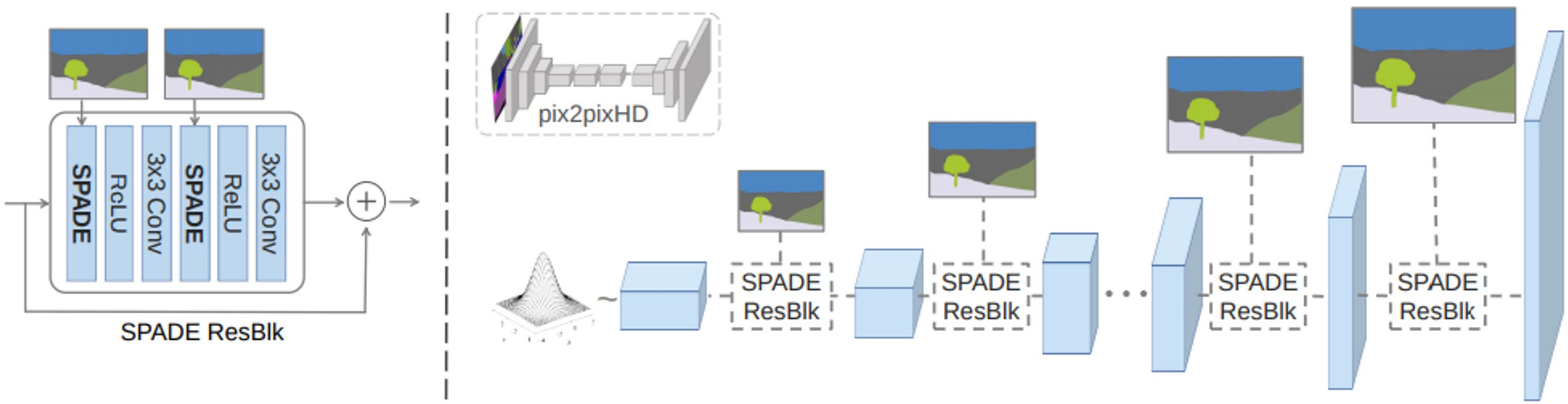


SPADE paper = [Semantic Image Synthesis with Spatially-Adaptive Normalization CVPR 2019]

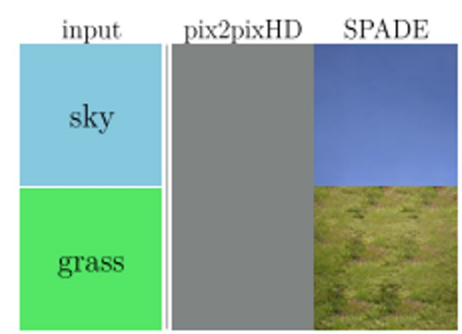
SPADE Generator



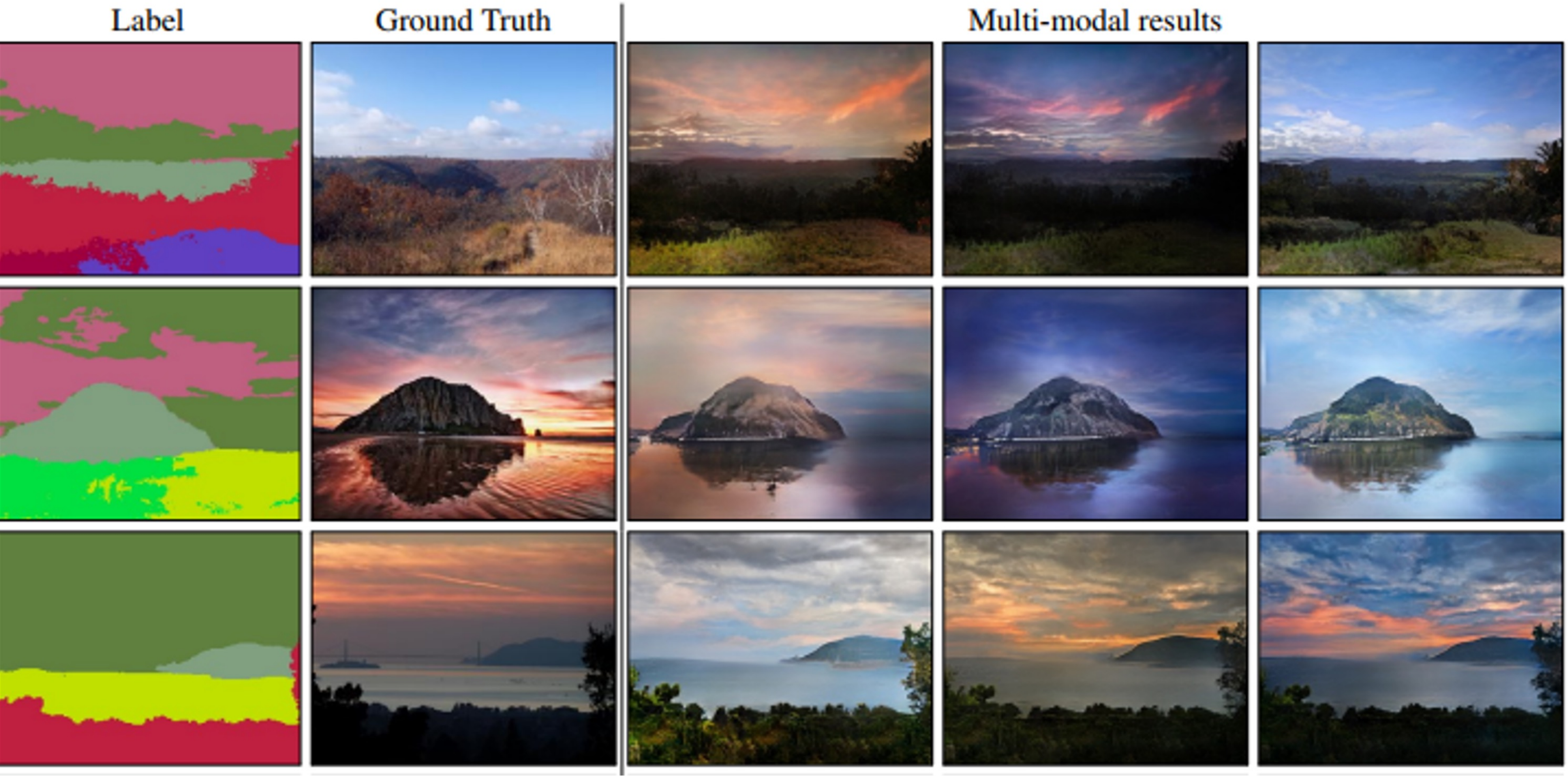
SPADE Generator



Better preserve semantic information
against common normalization layers



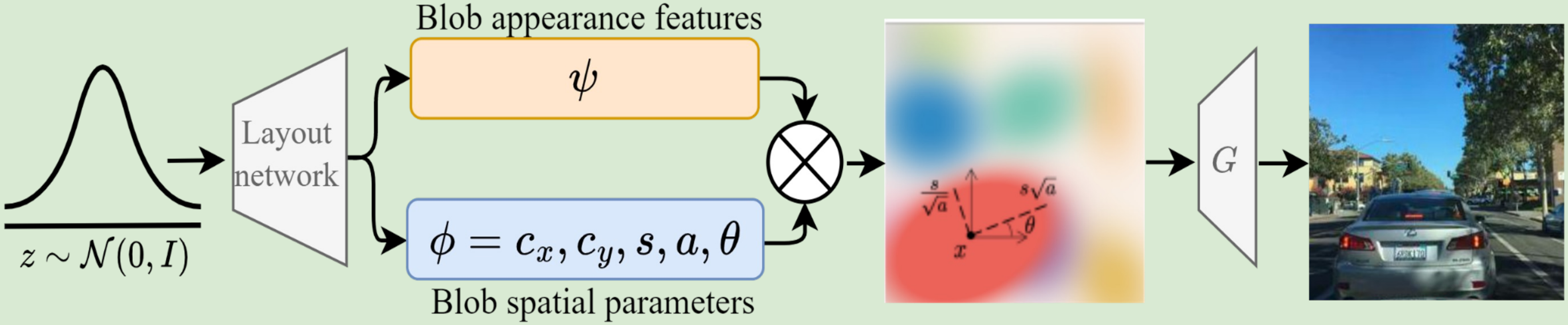
SPADE results



Spade and follow-up approaches

Editing with *conditional-or-structured-latent* GANs

BlobGan [Epstein *et al.* ECCV'22]



Editing with *conditional-or-structured-latent* GANs

Editing with *conditional-or-structured-latent* GANs

Example of Counterfactual optimization for editing

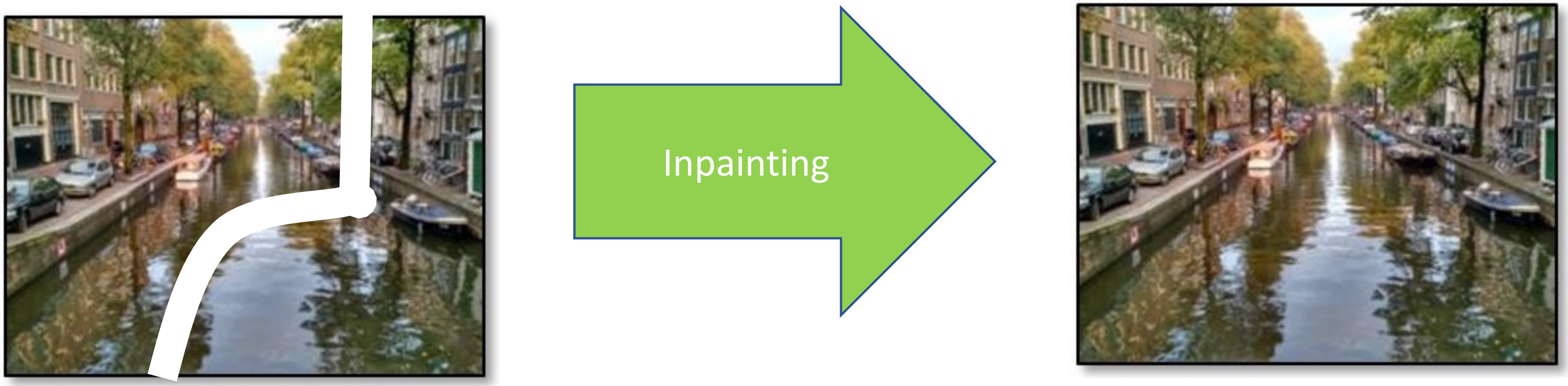
Generative models

Outline

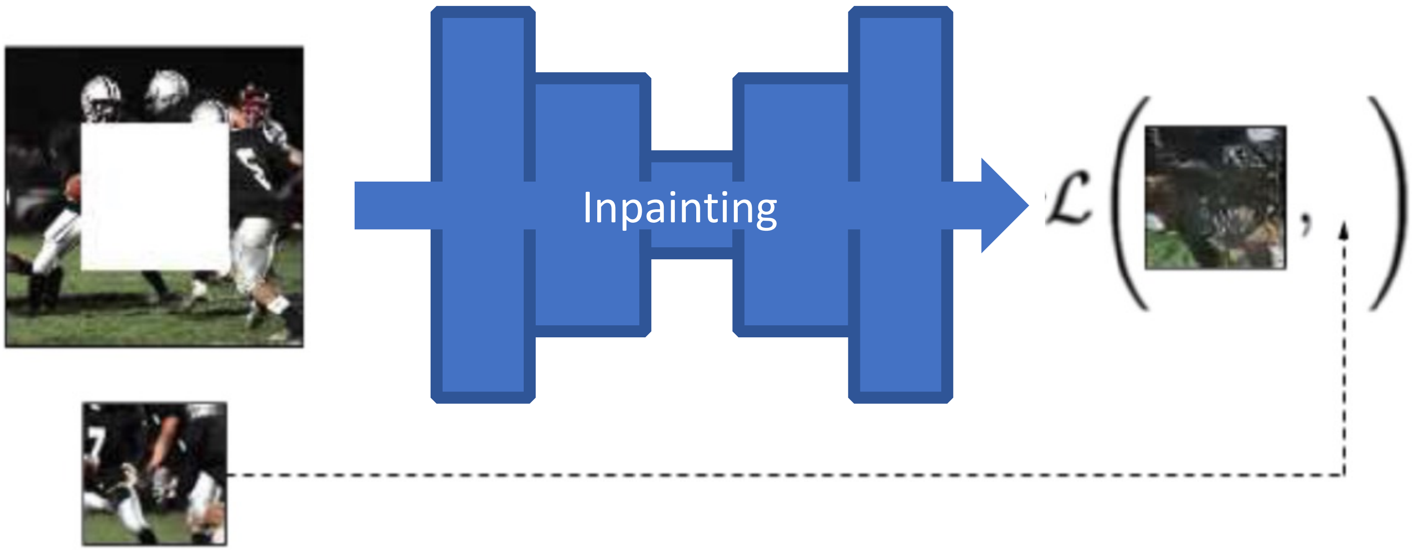
1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. Conditional GANs
   1. Principle
   2. Text2Image
   3. Image2Image
   4. **Inpainting and general missing data encoder**

Inpainting task

- Complete the missing part



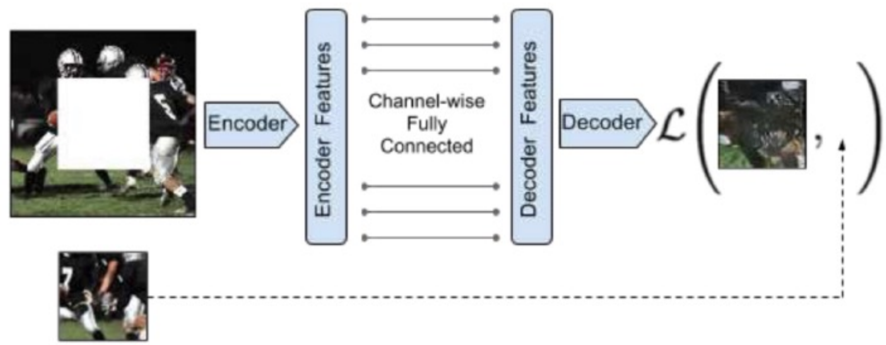
Inpainting as unsupervised learning with GAN loss



Reconstruct missing pixels by decoding using context
Loss defined on the predicted patch and the real one (known at training time)

First proposition -- Architecture

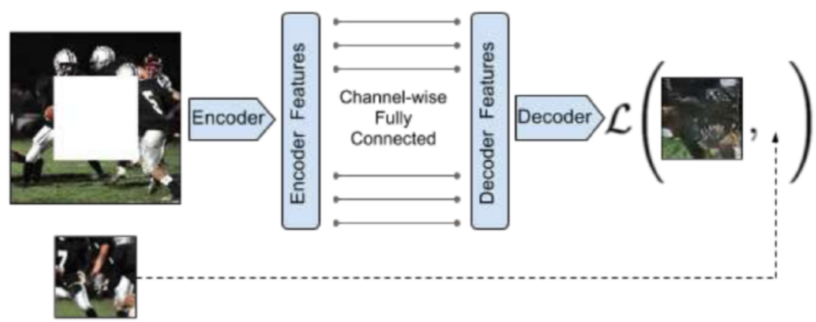
- Architecture: Encoder/Fully connected/Decoder



- DC-GAN for inpainting task
- **Input:** $227 \times 227 \times 3$ image
- **Output:** encoder context features $(6 \times 6 \times 256)$

Channel-wise fully-connected layer

- **Input / output:** $6 \times 6 \times 256$ channels
- **First layer:** Channel-wise fully-connected (each 6×6 input connected to the corresponding 6×6 output)
- **Second layer:** Stride 1 convolution to mix channels

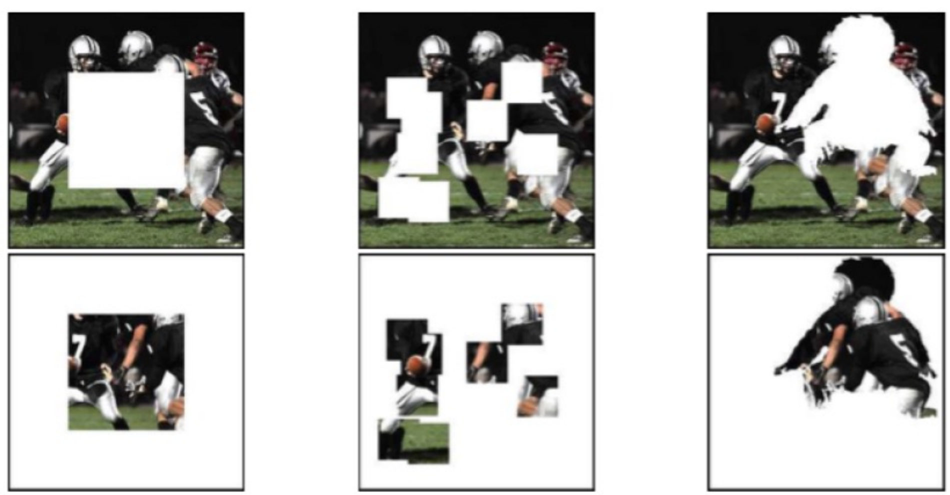


Decoder

- **Architecture:** Same as DC-GAN: 5 up-convolutional layers ("deconv" + ReLU)
- **Input:** decoder context features $6 \times 6 \times 256$
- **Output:** $227 \times 227 \times 3$ image

Training: Masking the images

- **How to define the mask ?**
  - Center region of the image
  - Random regions (chosen solution)
  - Random segmentation mask from VOC (said to be equivalent to random regions)
- **Formal definition:** Defined by a mask $\hat{M} \in \{0, 1\}^{227 \times 227}$ with 1 if the pixel should be masked



Training: Loss - Overview

- Trained completely from scratch to fill-up the masked areas
- **Problem:** multiple plausible solutions
- **Solution:** combining 2 losses:
  - $\mathcal{L}_{rec}$ **L2 reconstruction loss:** learn the structure of the missing region (average multiple modes in prediction)
  - $\mathcal{L}_{adv}$ **Adversarial loss:** make it look real (pick a mode from the distribution)

$$\min_F \mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}$$

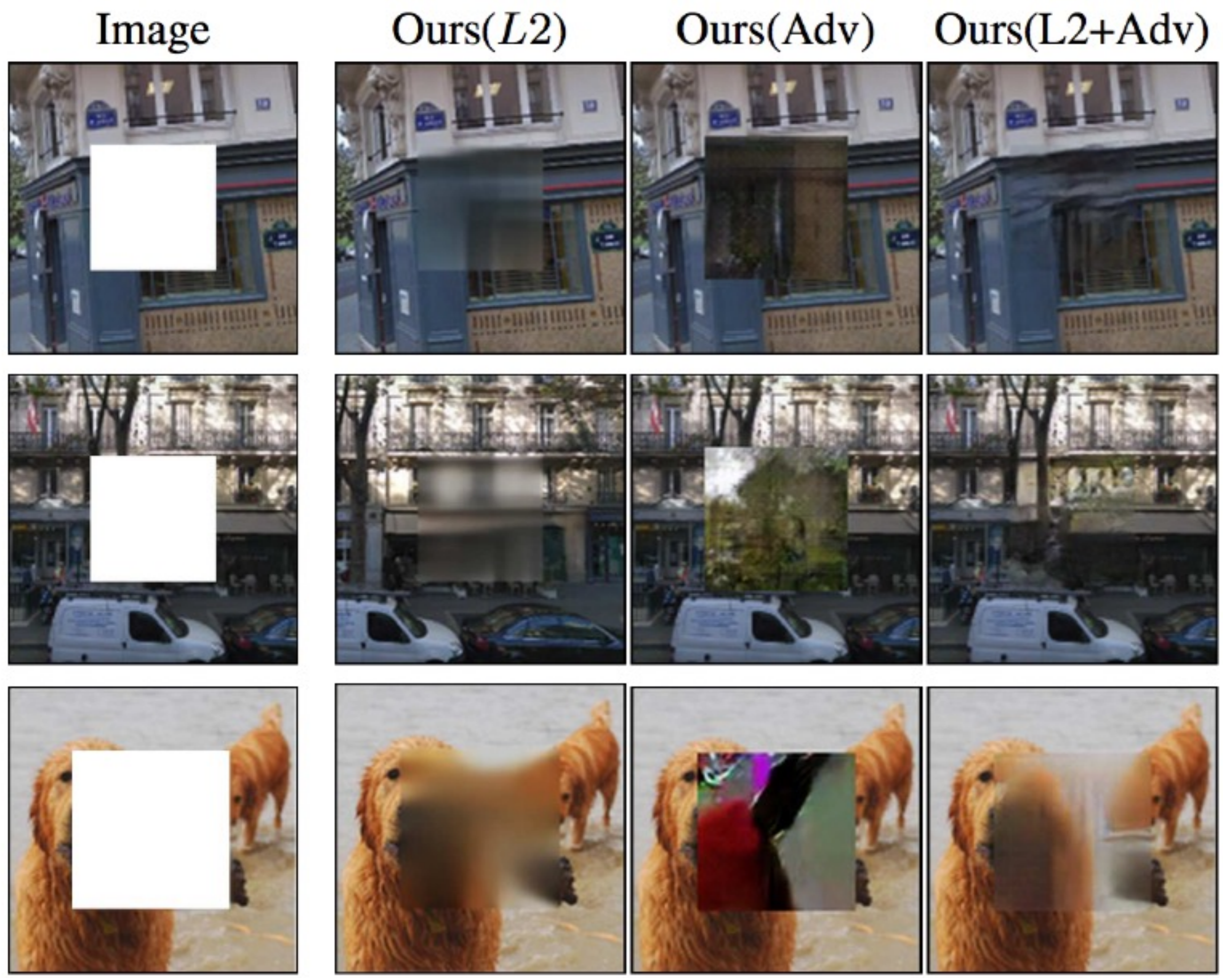
$$\mathcal{L}_{rec}(x) = \left\| \hat{M} \odot \left(x - F((1 - \hat{M}) \odot x) \right) \right\|_2$$

$$\mathcal{L}_{adv} = \max_D \mathbb{E}_{x \in \mathcal{X}} \left[\log(D(x)) + \log\left(1 - D(F((1 - \hat{M}) \odot x))\right) \right]$$

- Rq: The encoder-decoder is the generator, D is a CNN

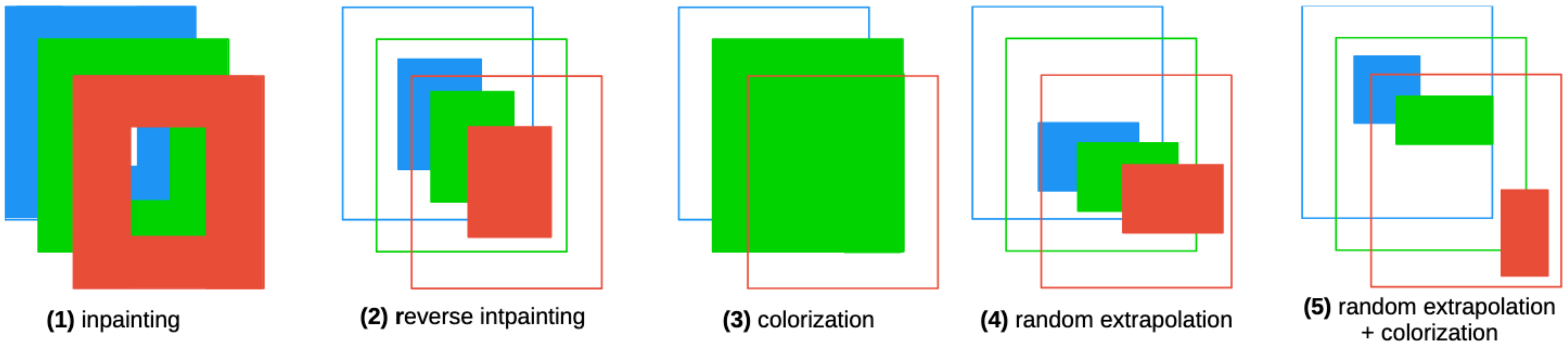
Results

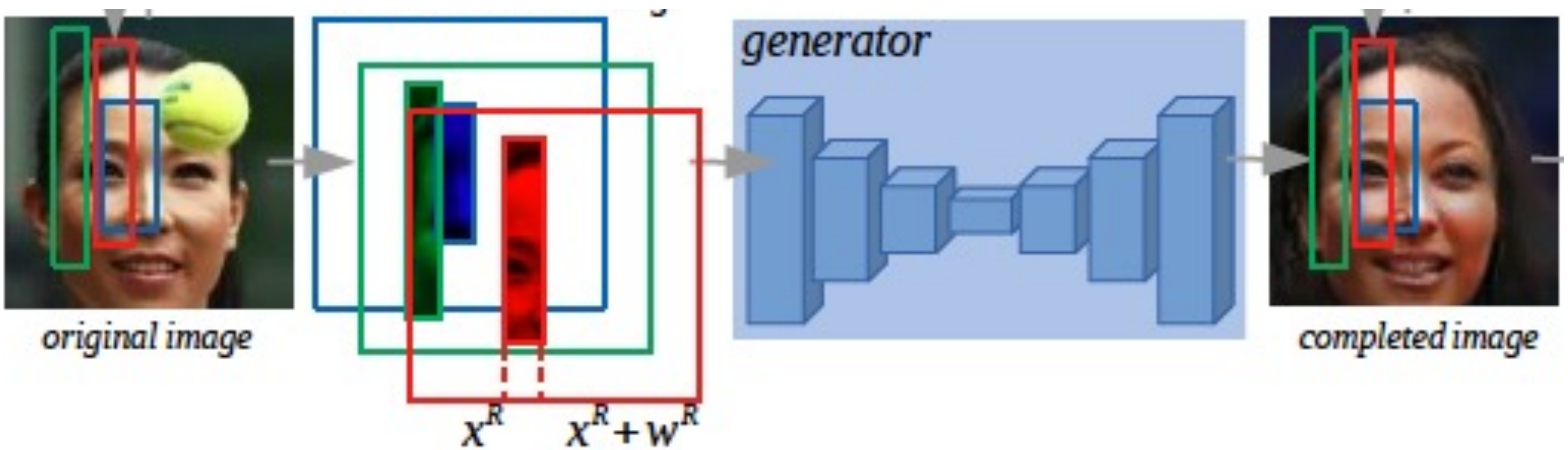
Dataset: StreetView Paris and ImageNet



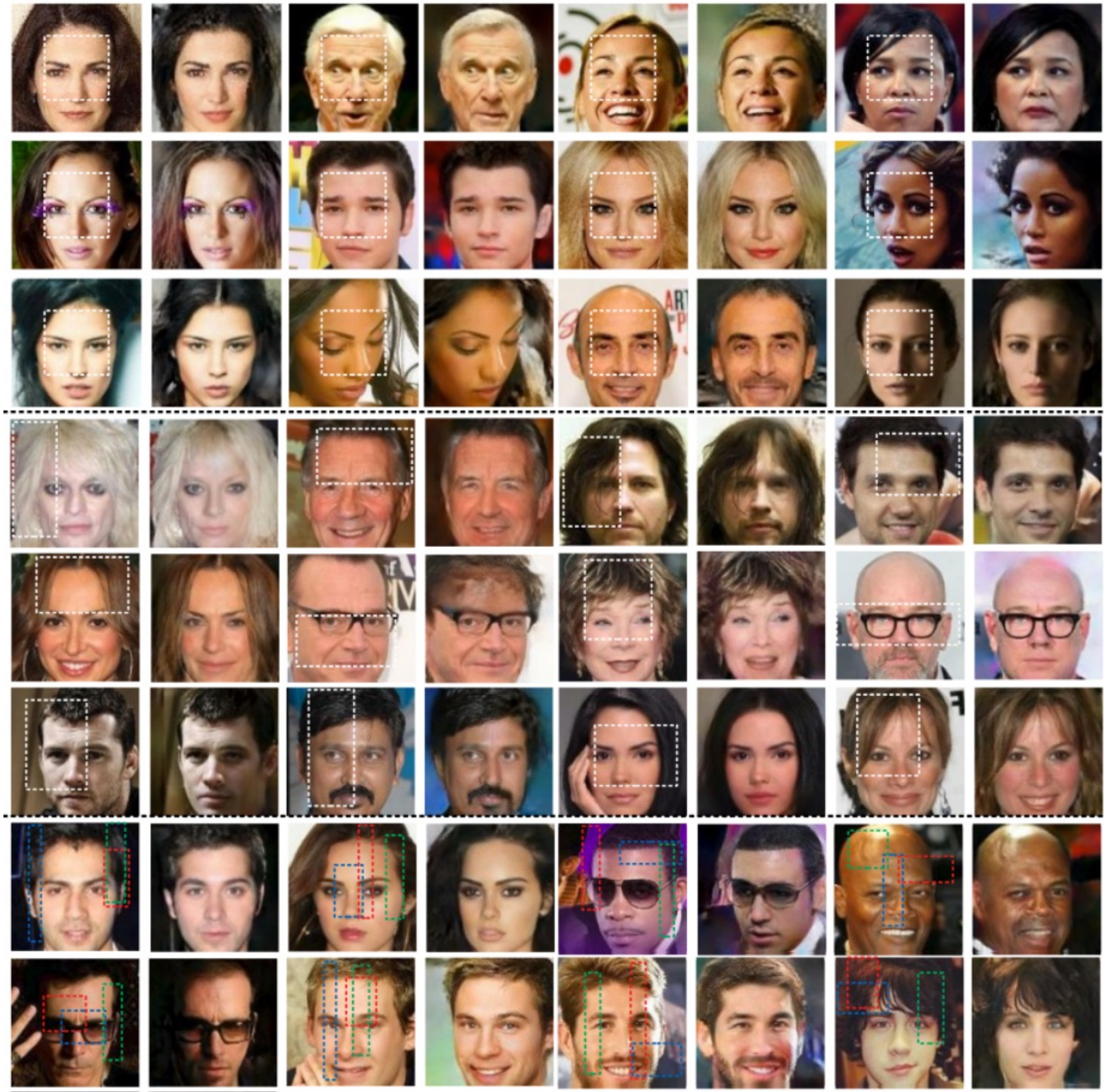
Semantic inpainting - Qualitative results

Generalizing inpainting: missing data encoder





Results



Generative models

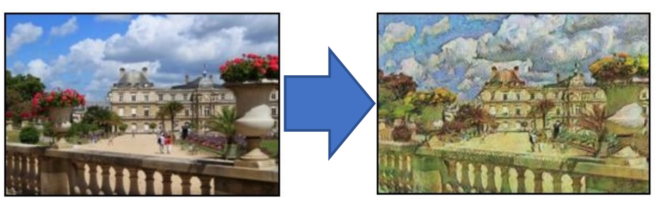
Outline

1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. Conditional GANs
    1. Principle
    2. Text2Image
    3. Image2Image
    4. Inpainting and general missing data encoder
    5. **Learning unpaired Transformation**

Unpaired Transformation

paired data

Transform an object from one domain to another ***without paired data***



photo

van Gogh

Domain X

Domain Y

Cycle GAN

https://arxiv.org/abs/1703.10593
https://junyanz.github.io/CycleGAN/

Domain X

Domain Y

Domain X

ignore input

$G_{X \to Y}$

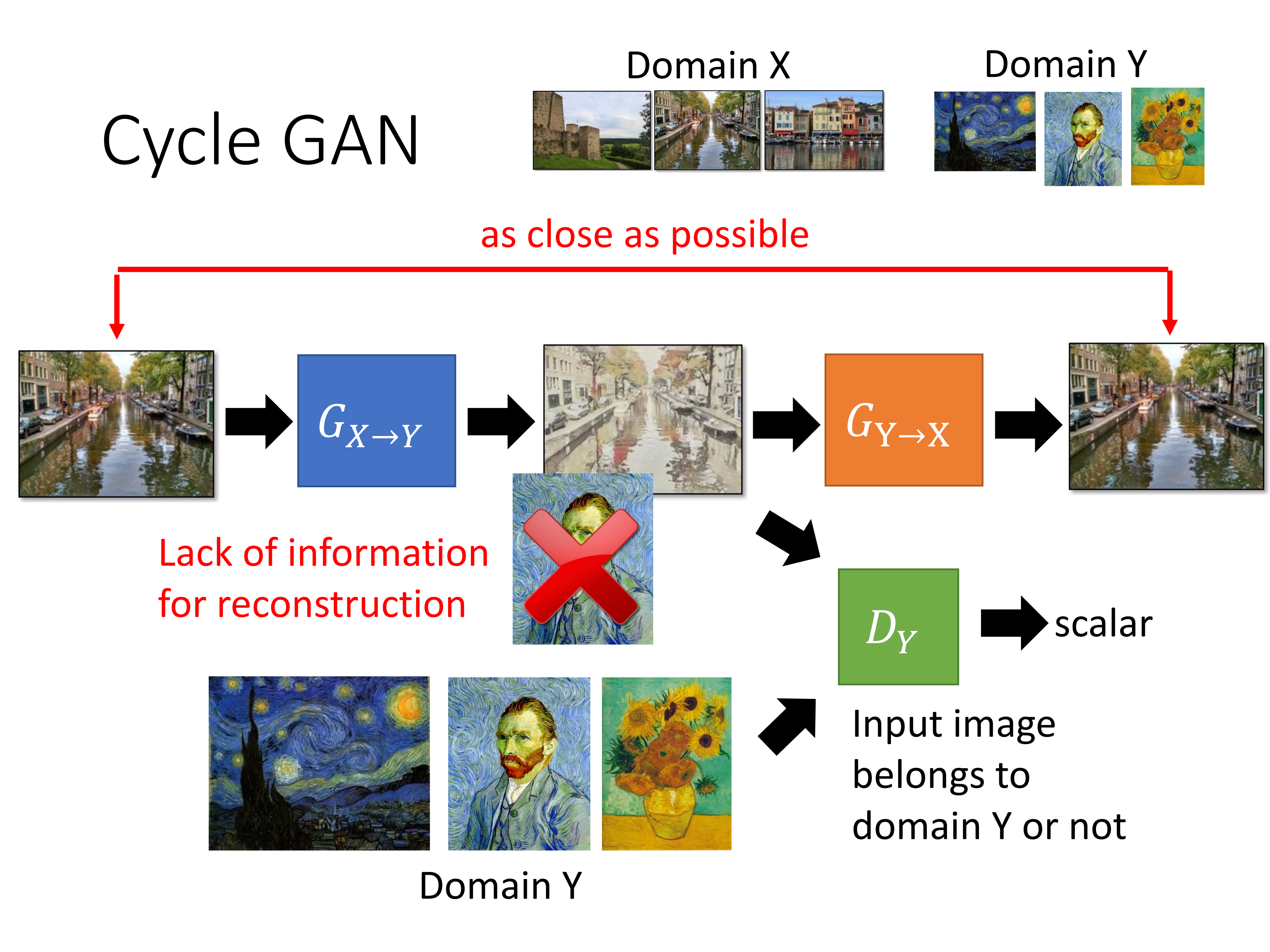
Become similar
to domain Y

Not what we want

D_Y scalar

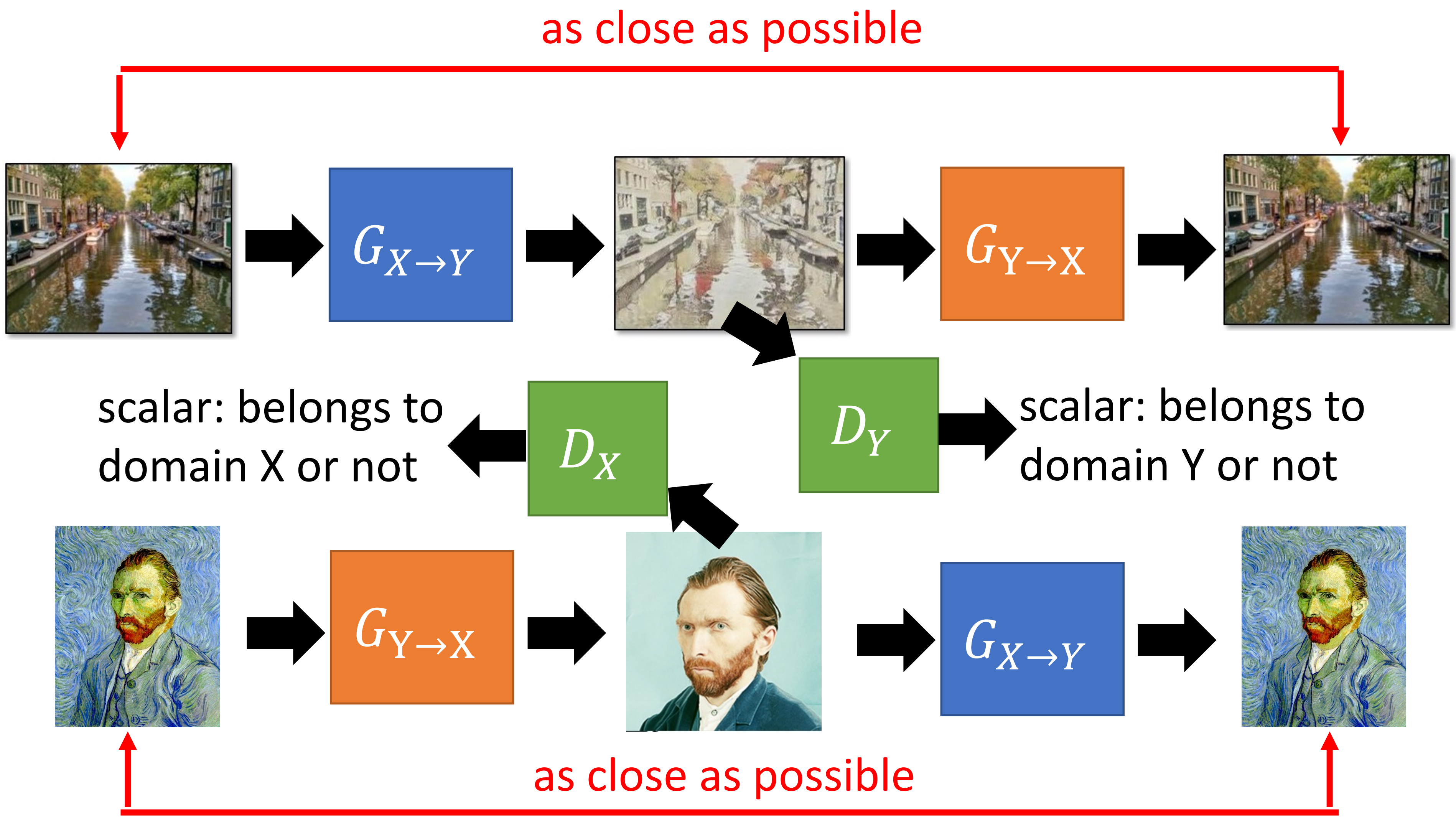
Input image
belongs to
domain Y or not

Domain Y


Cycle GAN
Domain X
Domain Y
as close as possible
$G_{X \to Y}$
$G_{Y \to X}$
Lack of information
for reconstruction
D_Y
scalar
Input image
belongs to
domain Y or not
Domain Y

Cycle GAN


Domain X
Domain Y


as close as possible
$G_{X \to Y}$
$G_{Y \to X}$
D_Y
scalar: belongs to domain Y or not
D_X
scalar: belongs to domain X or not
$G_{Y \to X}$
$G_{X \to Y}$
as close as possible

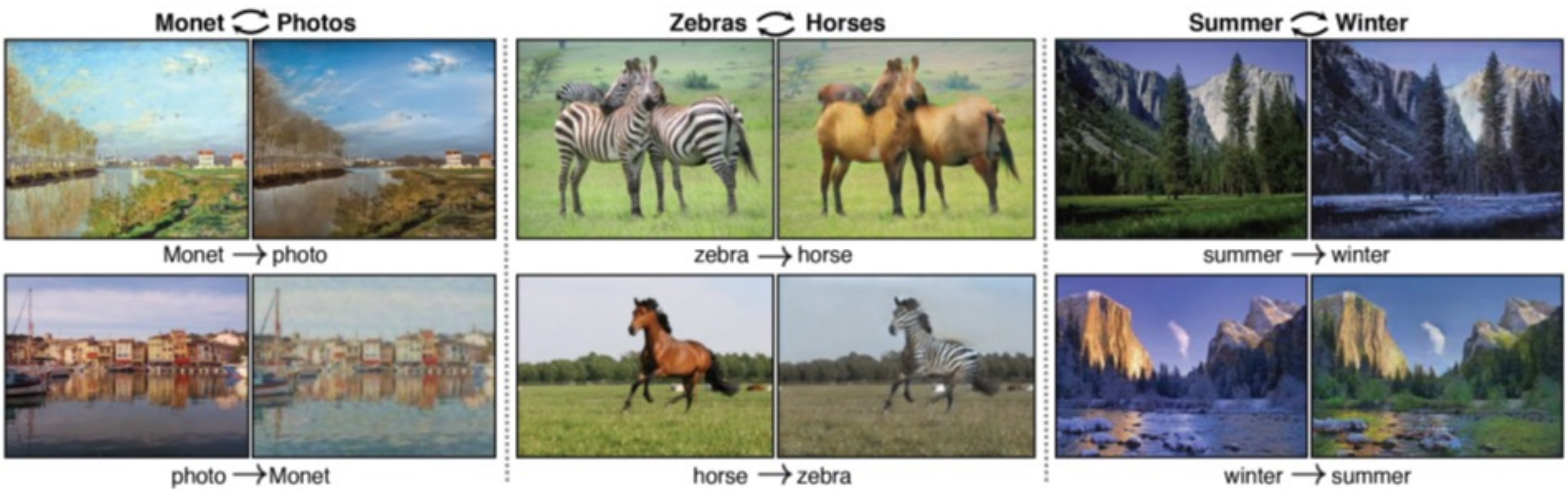
Results -- Cycle GAN

photo

van Gogh

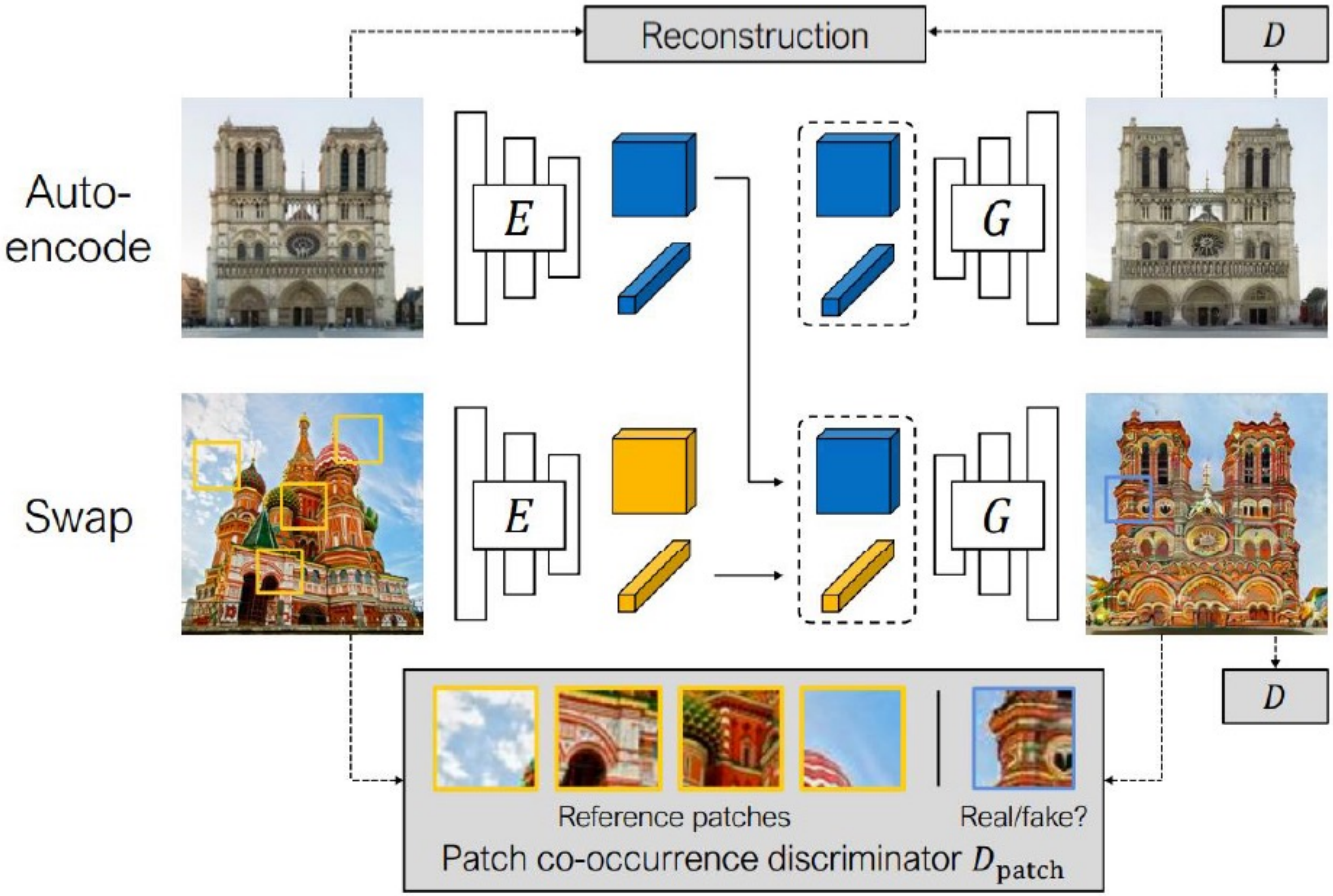
Domain X

Domain Y



GANs: works in progress

A lot of things to better understand, to use, adapt, …

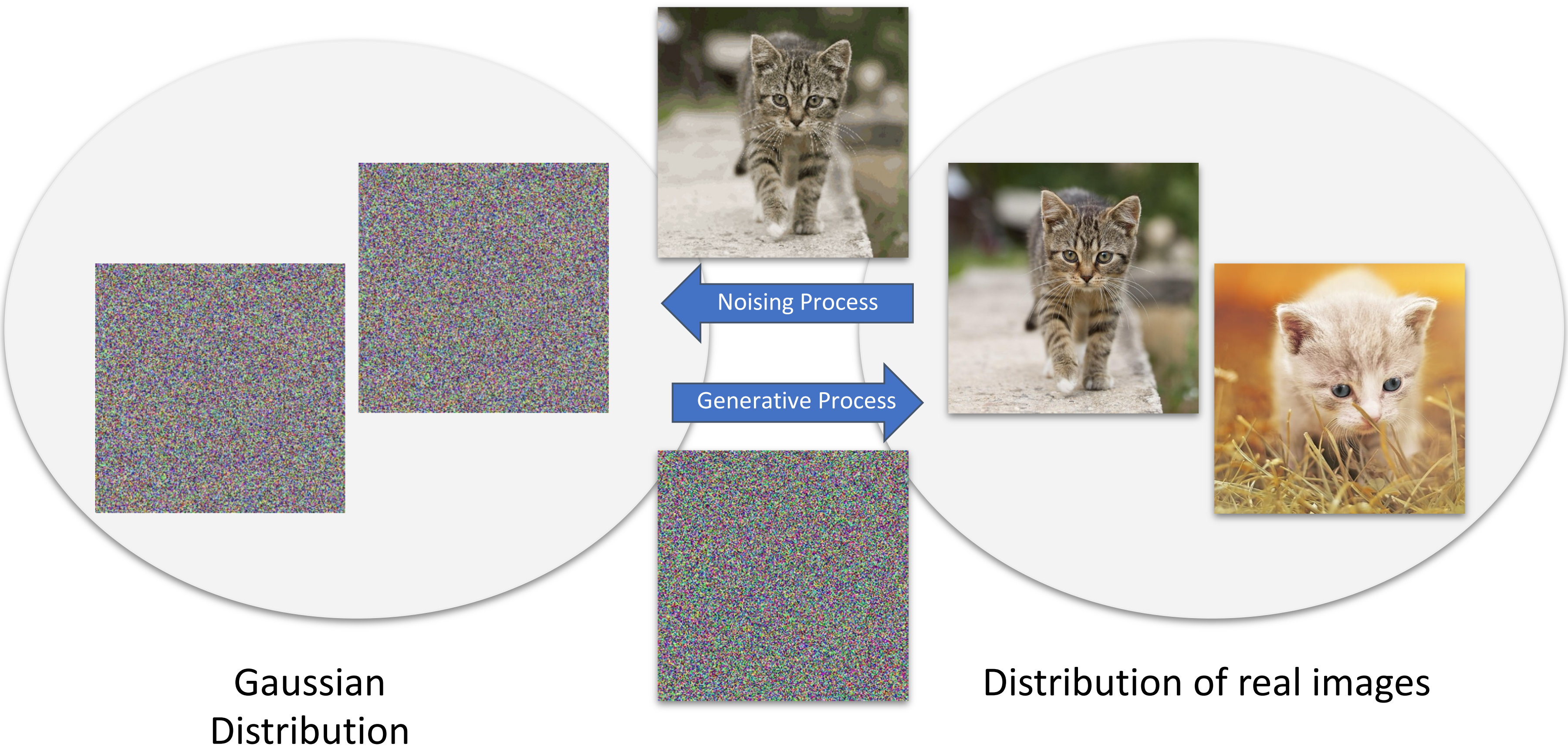


Generative models

Outline

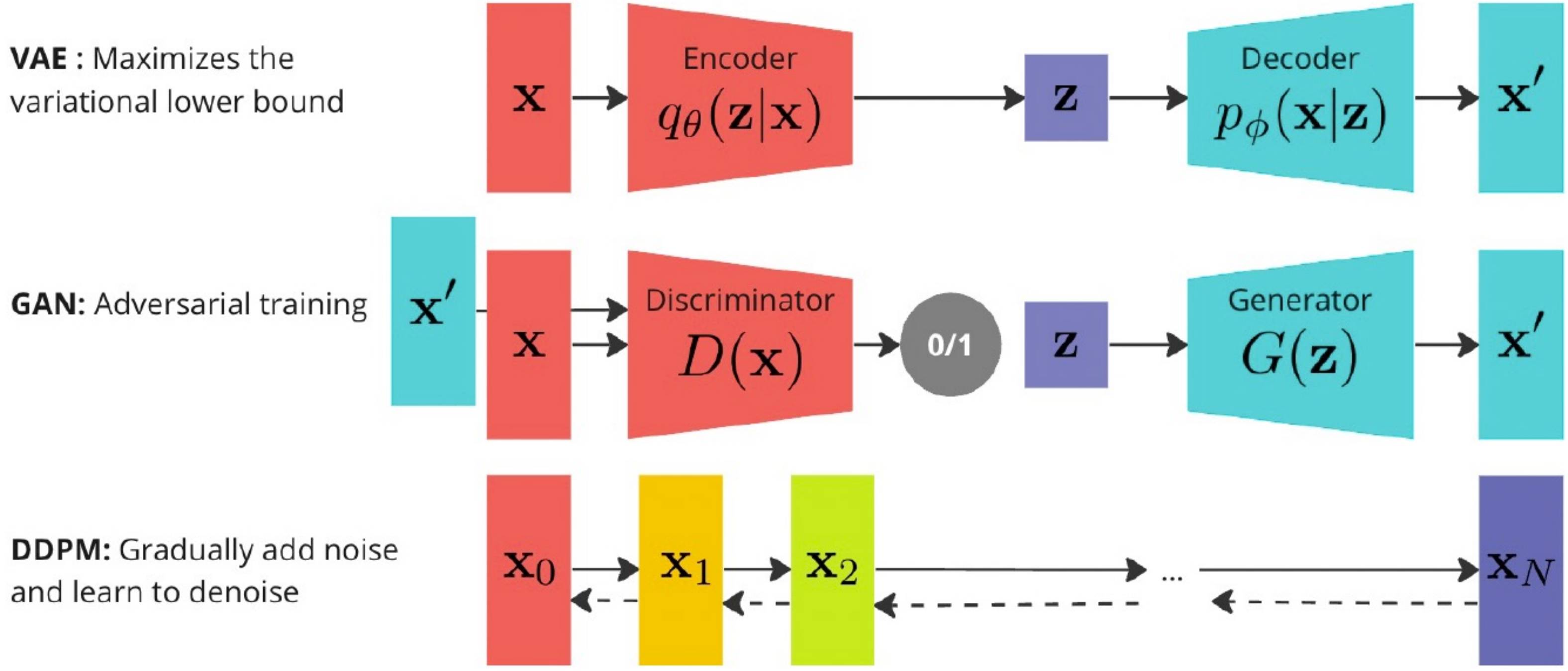
1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. Conditional GANs
6. **Diffusion models**

Generative Modelling with Diffusion models



Generative Modelling with Diffusion models

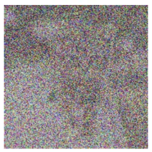
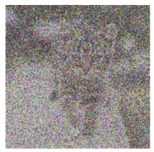
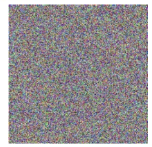
DDPM: Denoising Diffusion Probabilistic Models
In context with other genenrative Models:



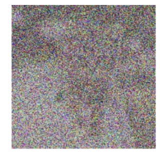
Generative Modelling with Diffusion models

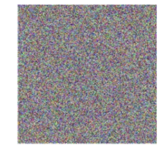
DDPM: Denoising Diffusion Probabilistic Models

$$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\epsilon$$

$$\mathbf{x}_t = \sqrt{\alpha_t}\,\mathbf{x}_0 + \sqrt{1-\alpha_t}\,\epsilon$$





Training:

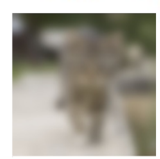
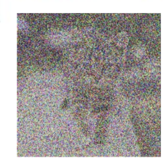
The reverse process refers to learning a model ϵ_θ *which approximates the noise added at a given timestep t:*

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, t, \epsilon} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2$$

Sampling:

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \sqrt{1-\alpha_t}\cdot\epsilon_\theta(\mathbf{x}_t, t)),$$

$$\mathbf{x}_{t-1} = \frac{(\alpha_{t-1}-\alpha_t)\sqrt{\alpha_{t-1}}}{\alpha_{t-1}(1-\alpha_t)}\hat{\mathbf{x}}_0 + \frac{(1-\alpha_{t-1})\sqrt{\alpha_t}}{(1-\alpha_t)\sqrt{\alpha_{t-1}}}\mathbf{x}_t + \sigma_t\mathbf{z}$$

Generative Modelling with Diffusion models

DDPM: Denoising Diffusion Probabilistic Models

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on

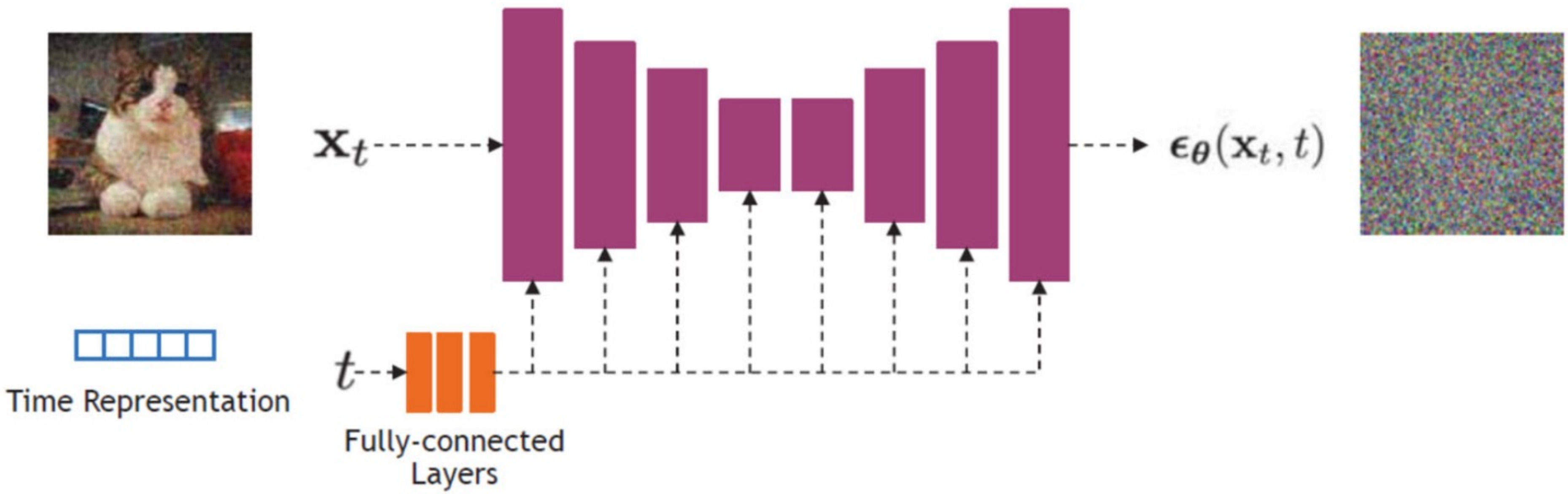
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\boxed{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}, t\right) \right\|^2$$

until converged

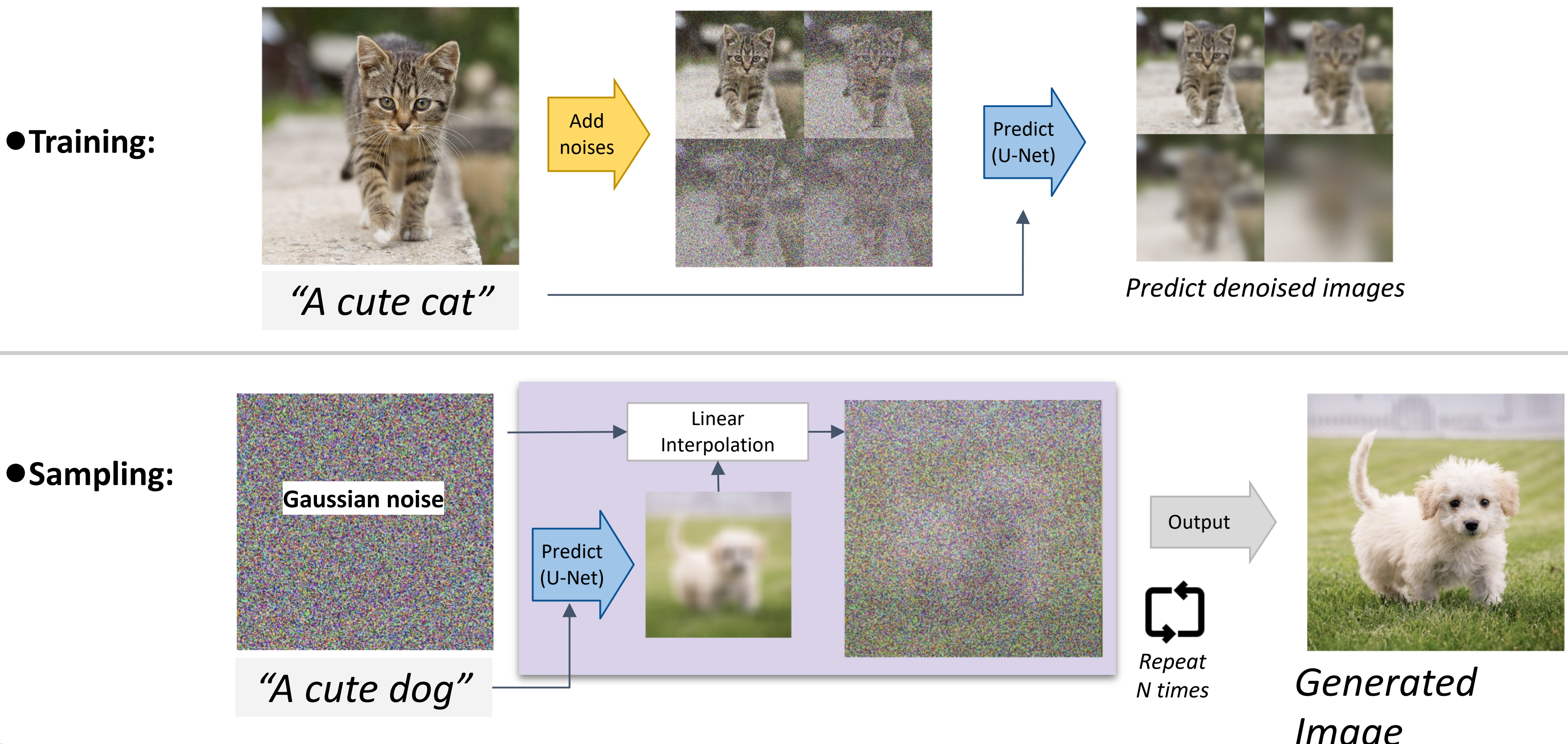
Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)} + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

U-Net for Diffusion models

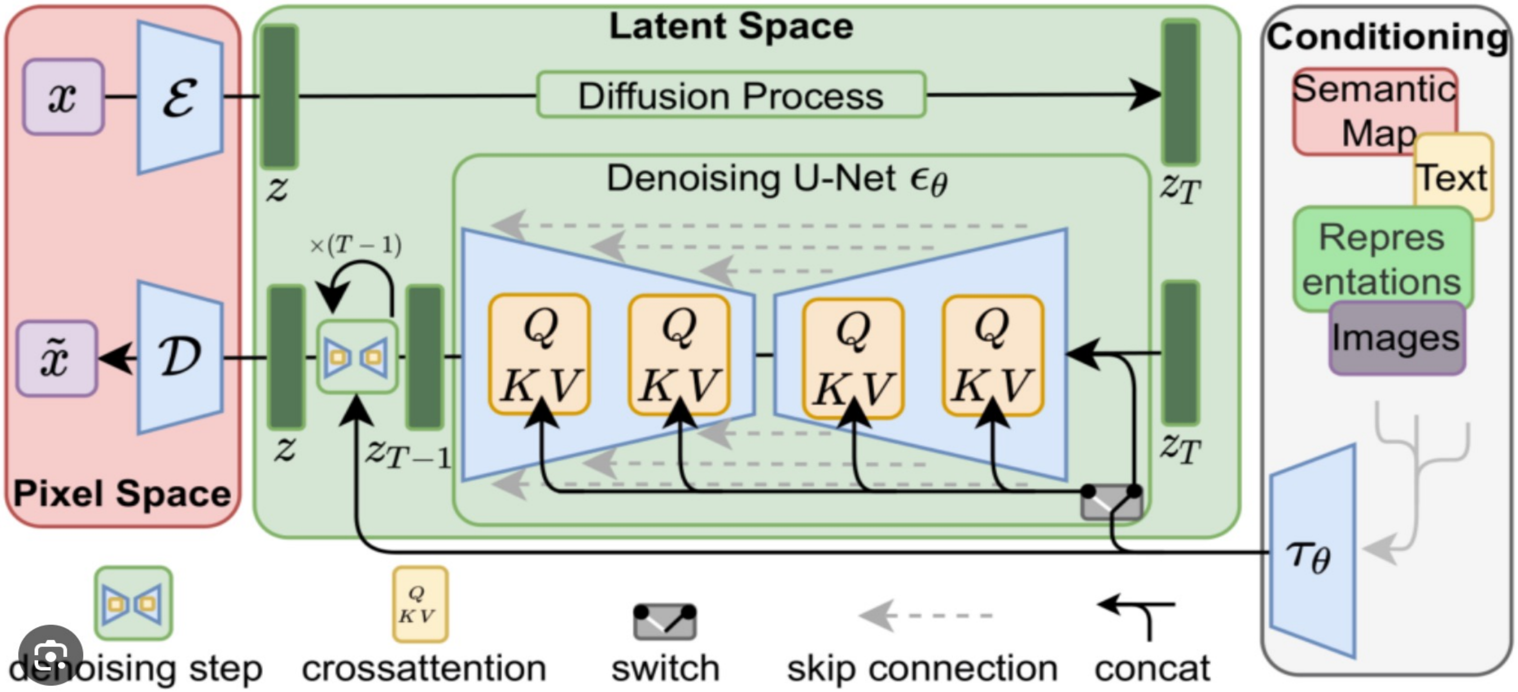


Text-conditional Diffusion models



Denoising Diffusion Probabilistic Models, *Jonathan Ho, Ajay Jain, Pieter Abbeel*, NeurIPS 2020

Text-conditional Diffusion models



Text-conditional Diffusion models

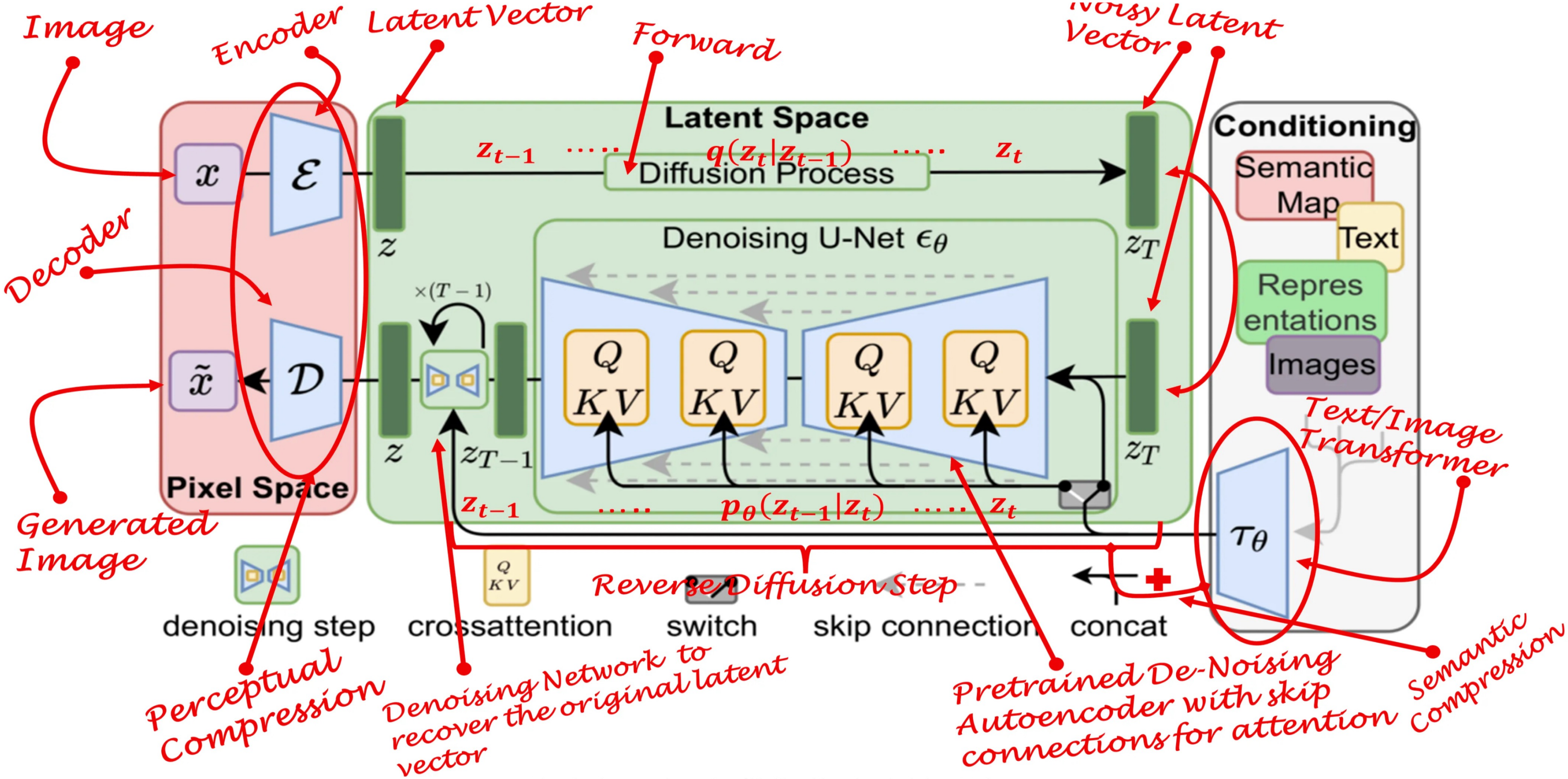


Image Generation using Diffusion Models

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.

A cute corgi lives in a house made out of sushi.

📖Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, *Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, Mohammad Norouzi*, NeurIPS 2022

Generation using Diffusion Models

And the (next) big thing is …

Text to Video