Generative models

Outline

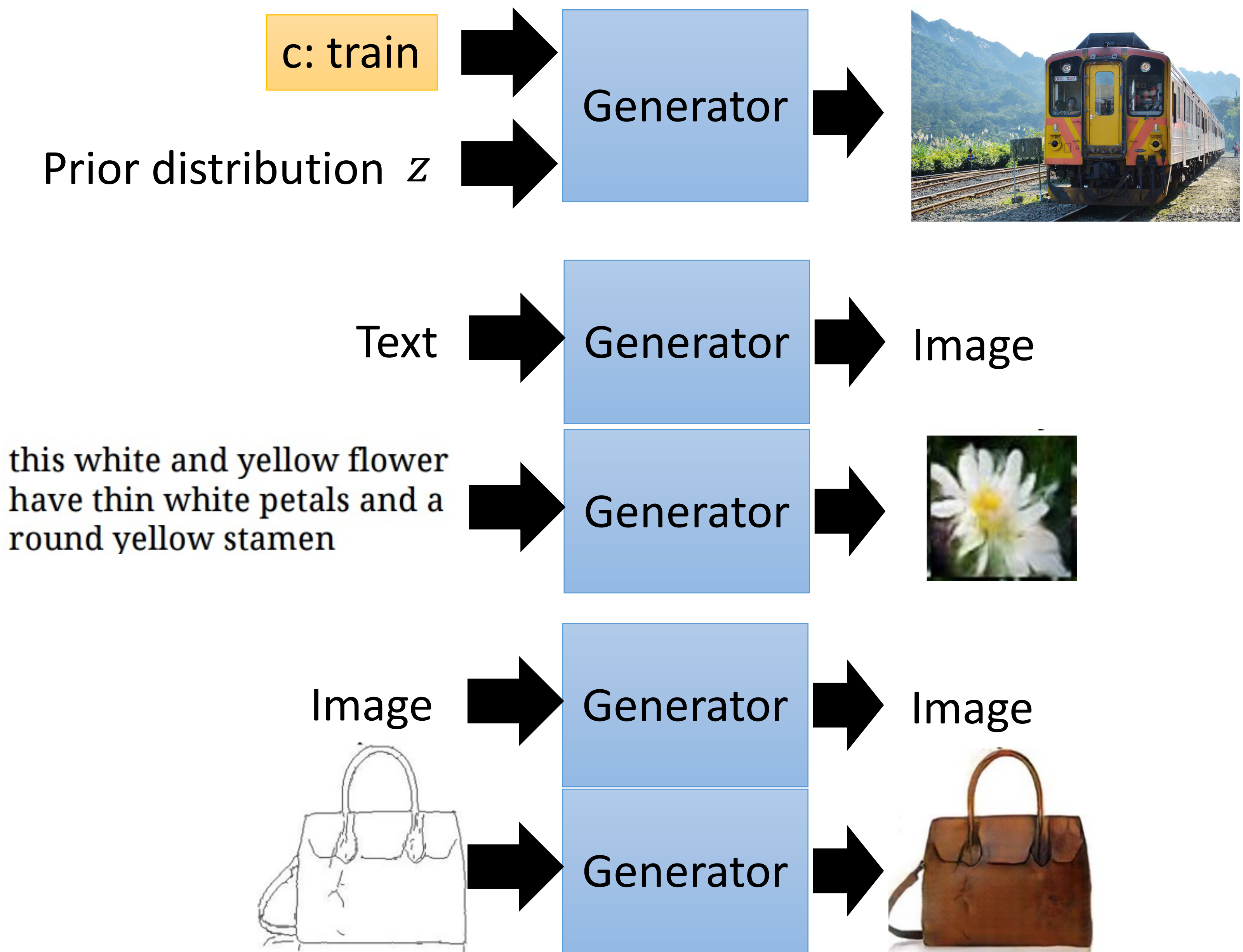
1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. **Conditional GANs**

Generative models

Outline

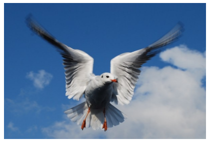
1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. **Conditional GANs**
    1. **Principle**

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

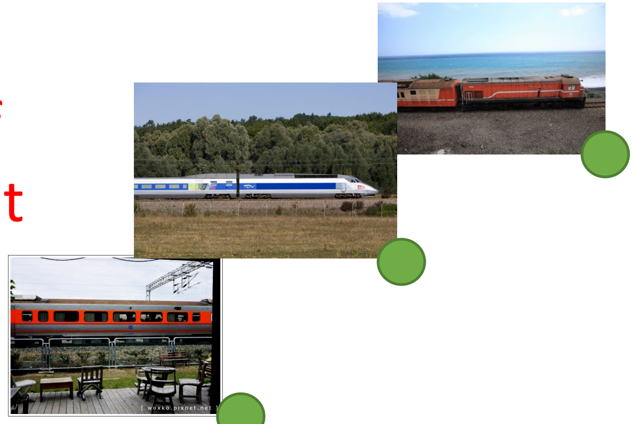
c^1: a dog is running → NN → Image ↔ $\hat{x}^1$: 

as close as possible

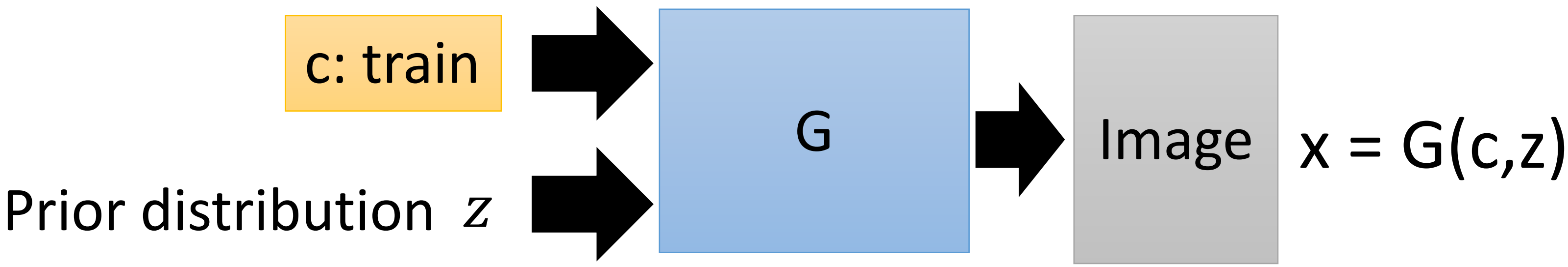
Text: "train"

Target of NN output

A blurry image!



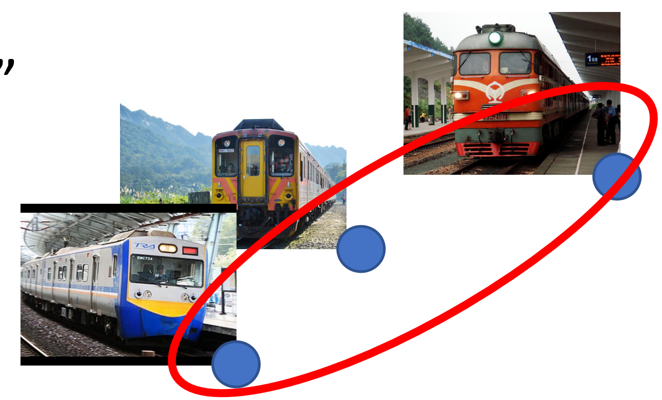
Conditional GAN

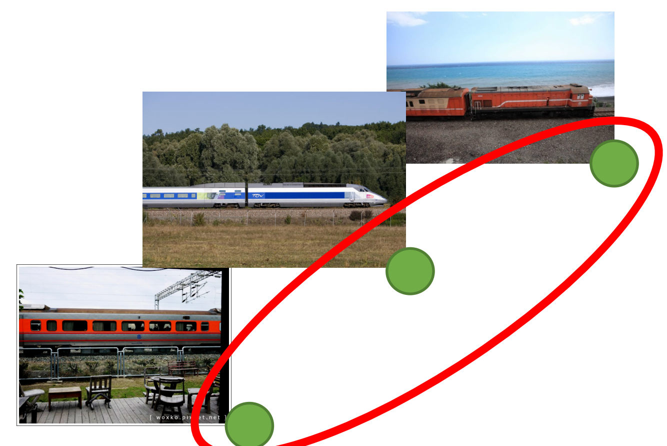


It is a distribution

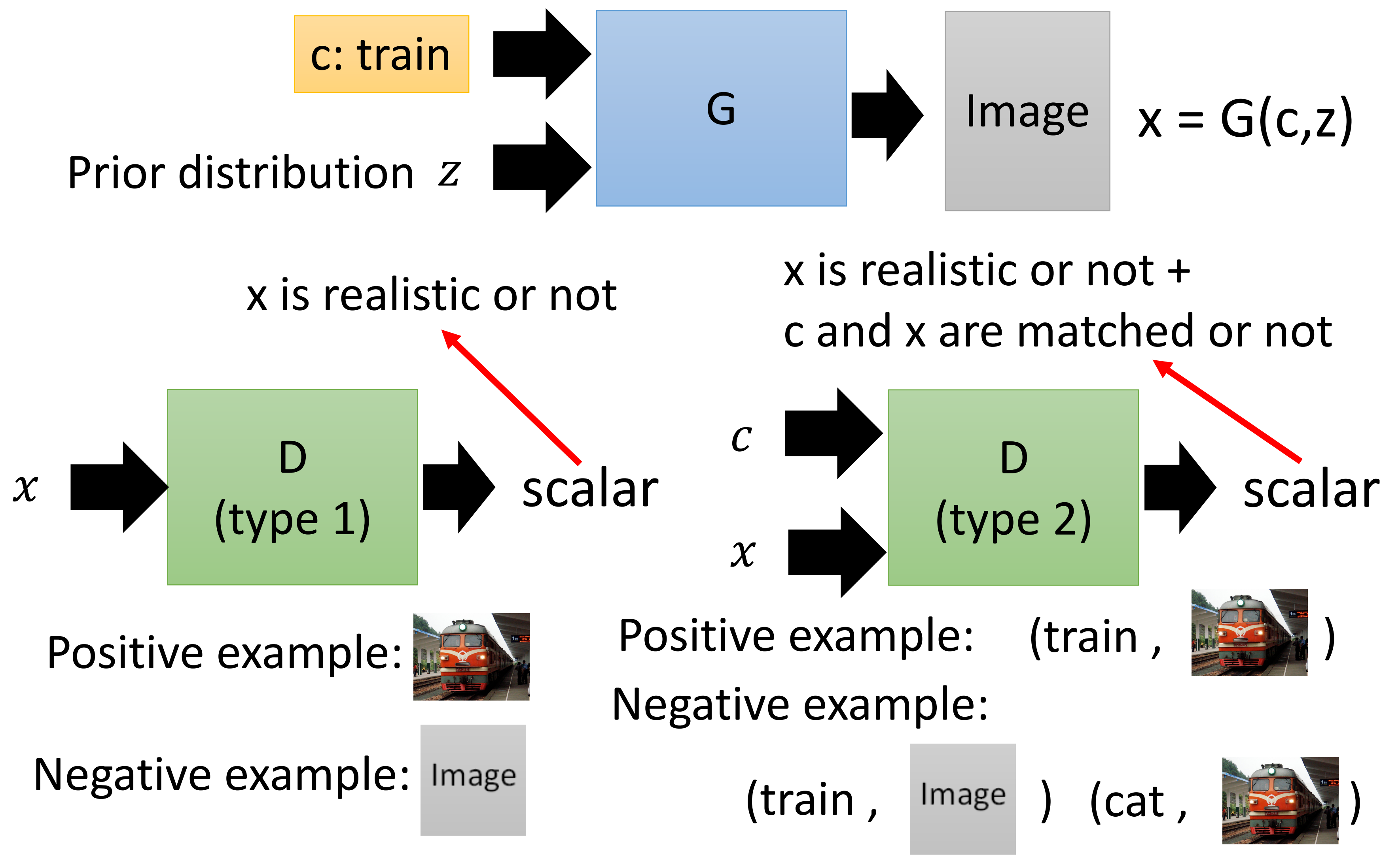
Approximate the distribution of real data

Text: "train"



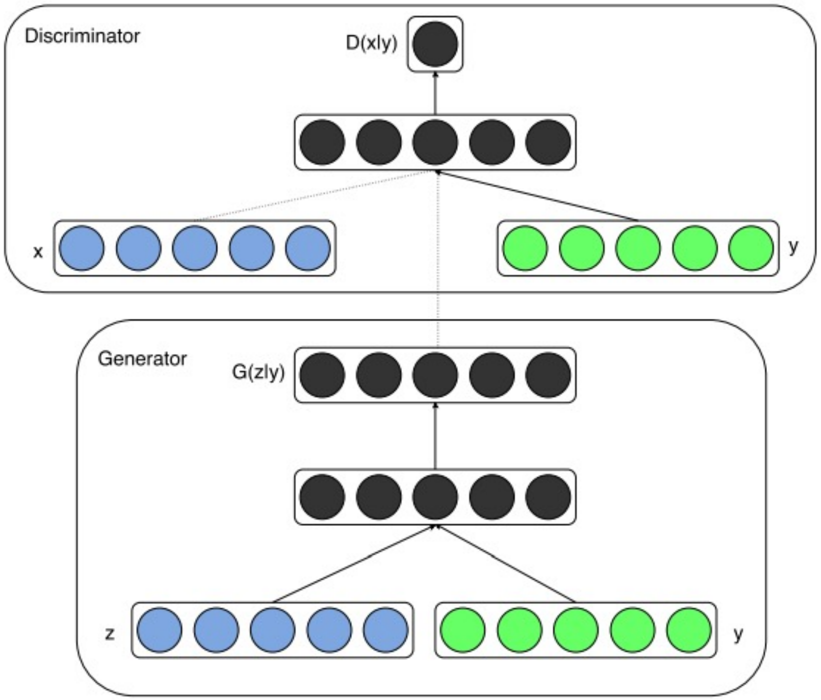


Conditional GAN



Conditional GAN (CGAN model)

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

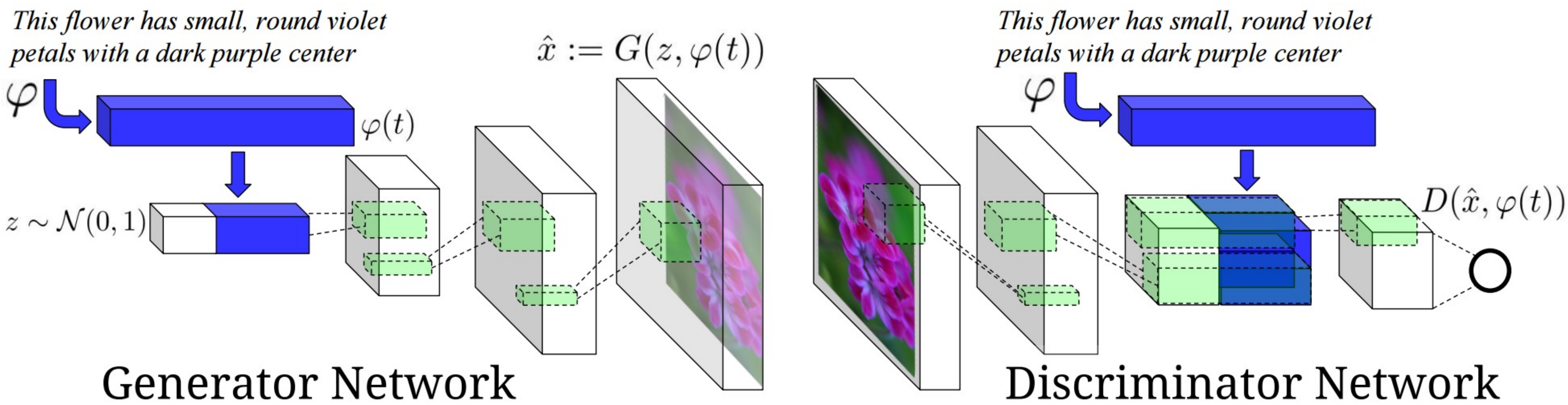


Generative models

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

[Reed et al. ICML 2016]

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

[Reed et al. ICML 2016]

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)

StackGAN: similar idea with LapGan to generate higher resolution images

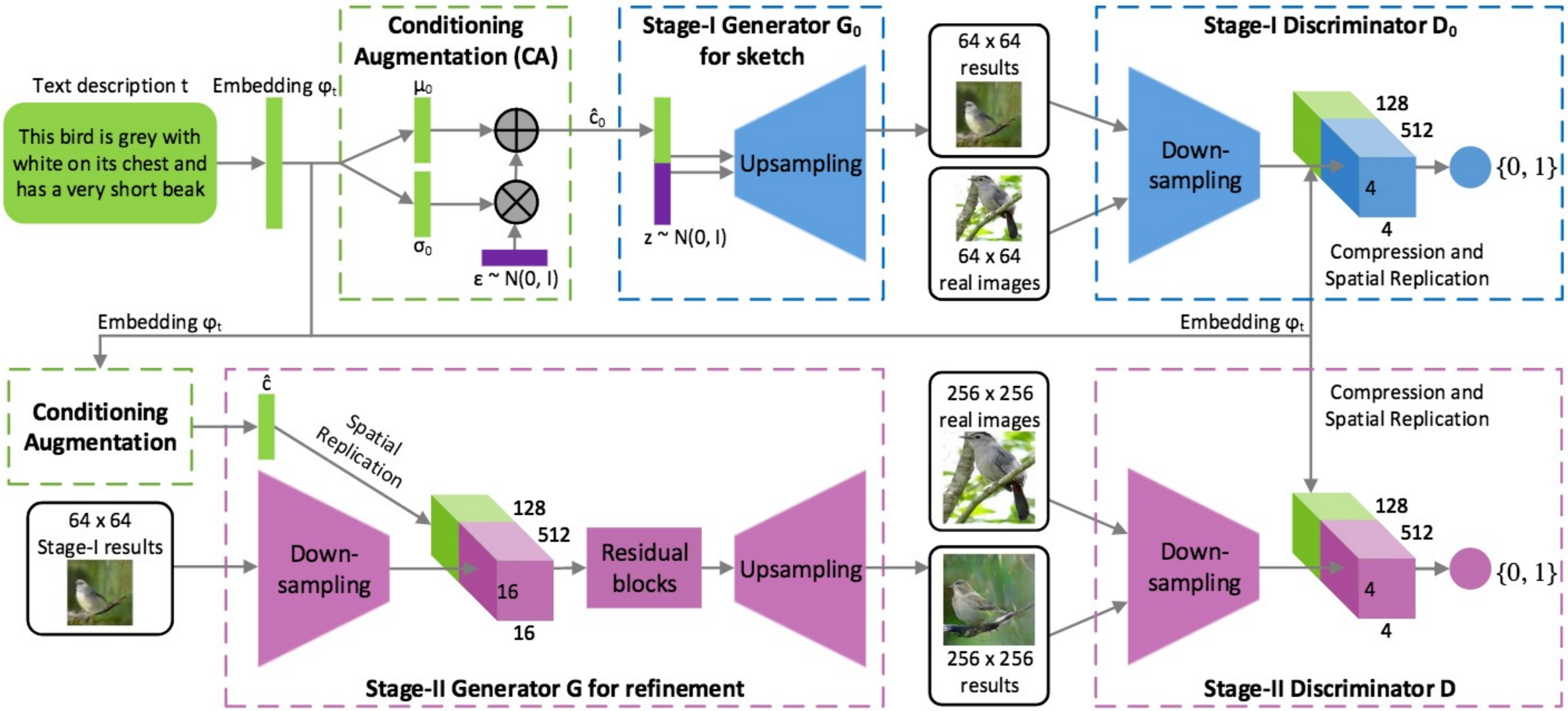
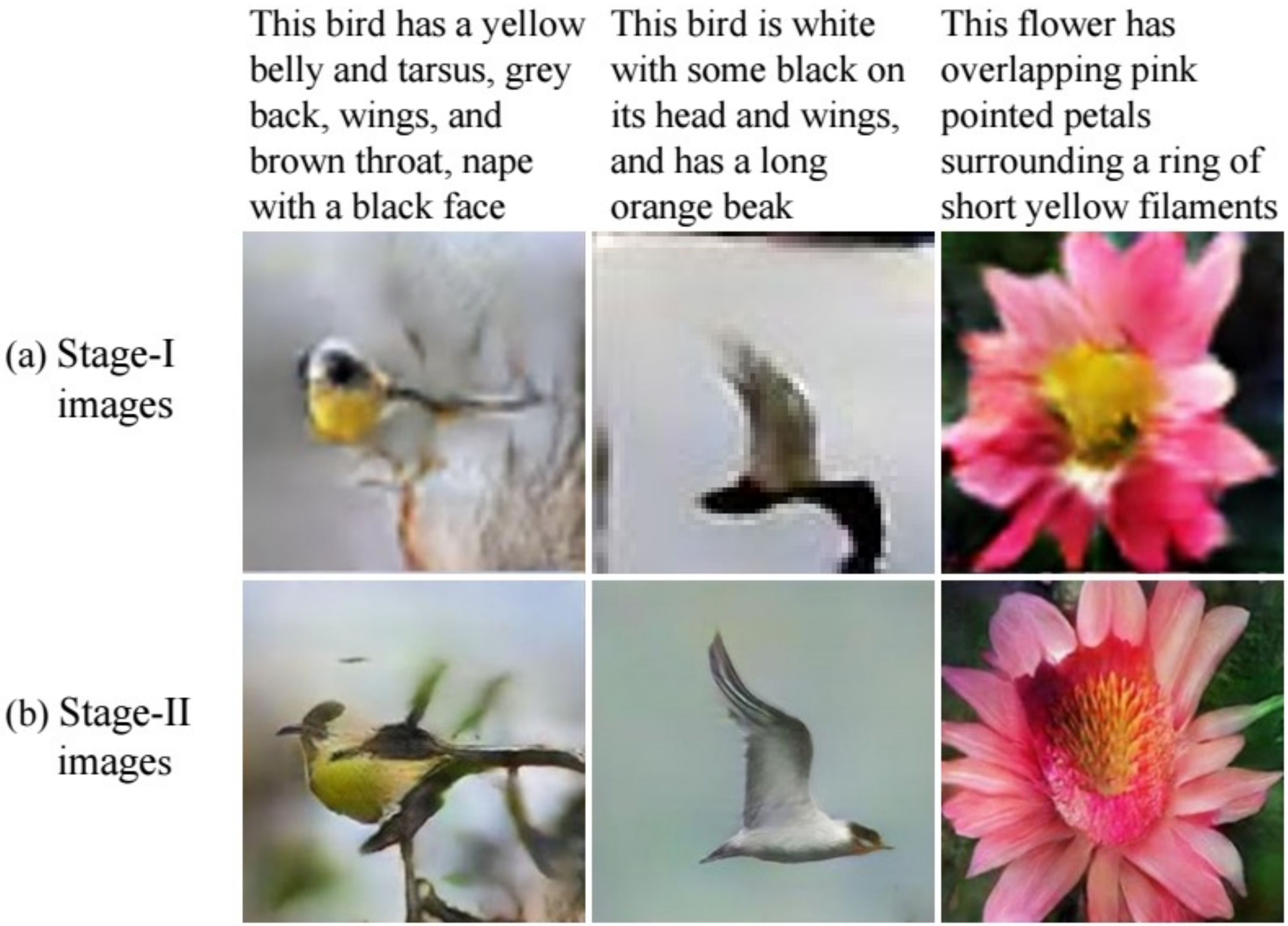


Figure 2. The architecture of the proposed StackGAN. The Stage-I generator draws a low-resolution image by sketching rough shape and basic colors of the object from the given text and painting the background from a random noise vector. Conditioned on Stage-I results, the Stage-II generator corrects defects and adds compelling details into Stage-I results, yielding a more realistic high-resolution image.

[Zhang et al. 2016]

StackGAN results



[Zhang et al. 2016]

Text2Image results

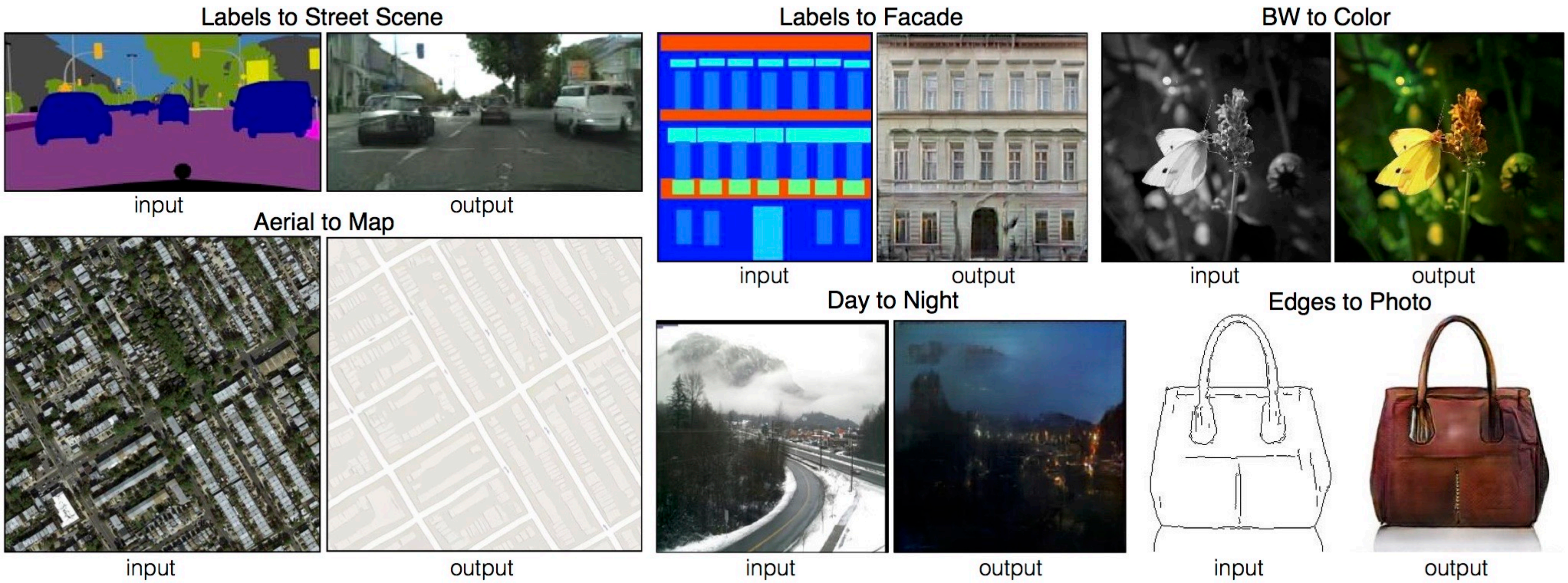
Caption	Image
a pitcher is about to throw the ball to the batter	
a group of people on skis stand in the snow	
a man in a wet suit riding a surfboard on a wave	

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**

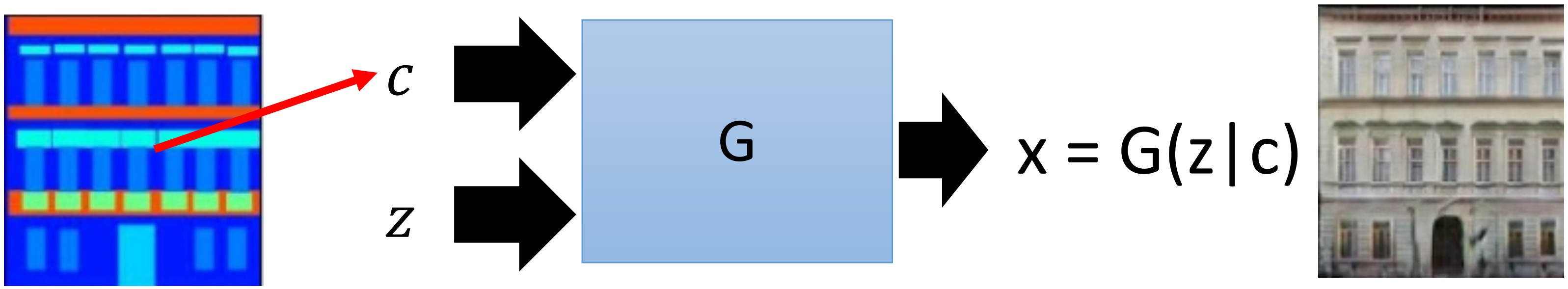
Image-to-Image Translation pix2pix

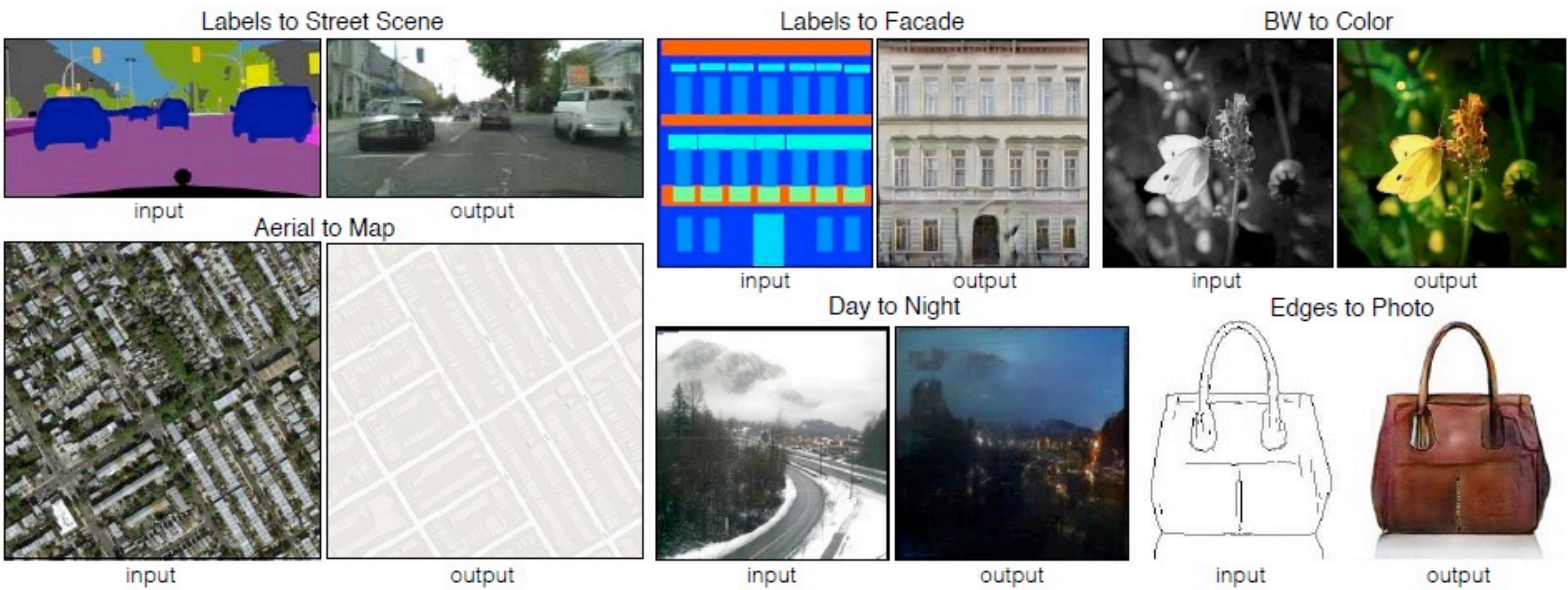


- Conditioned on an image of different modality
- No need to specify the loss function

[Isola et al. CVPR 2017]

Image-to-image pix2pix

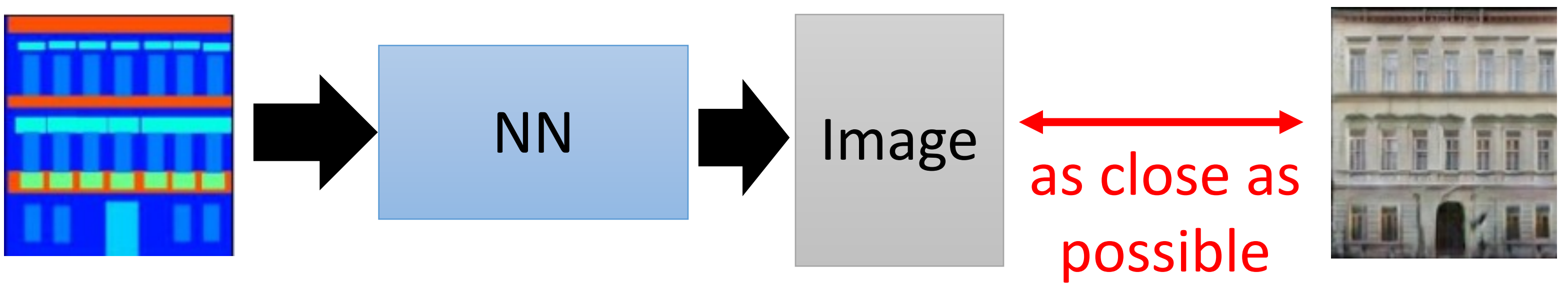




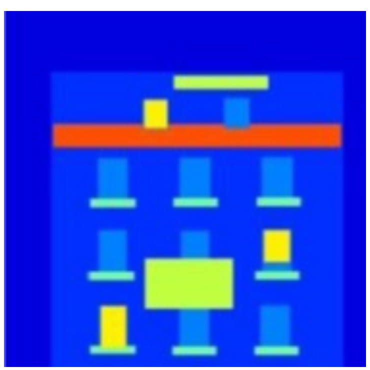
https://arxiv.org/pdf/1611.07004

Image-to-image pix2pix

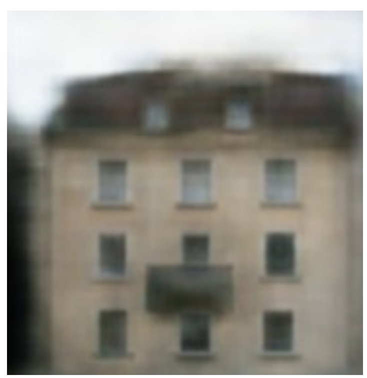
- Traditional supervised approach



Testing:



input

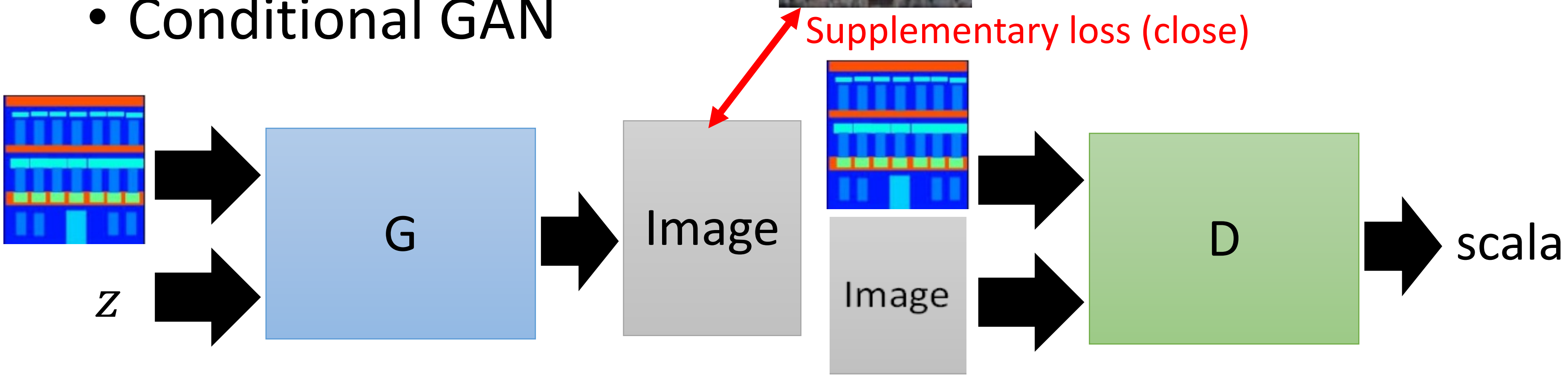


close

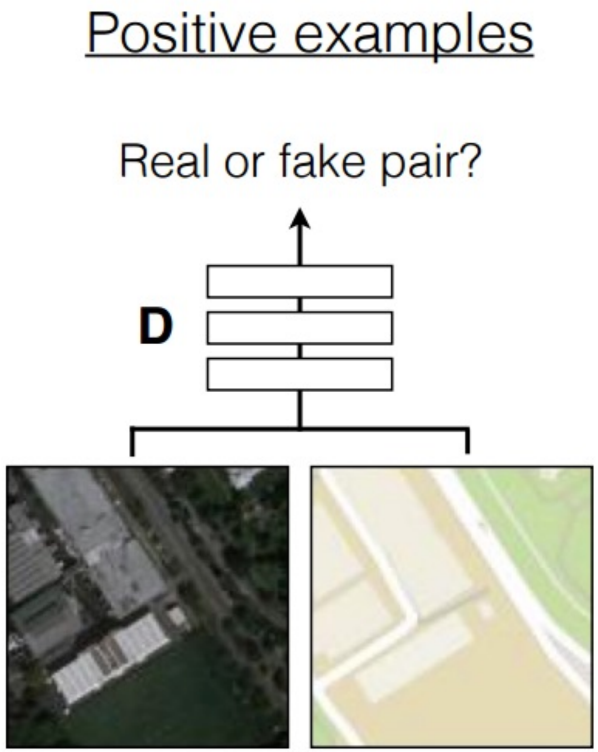
It is blurry because it is the average of several images.

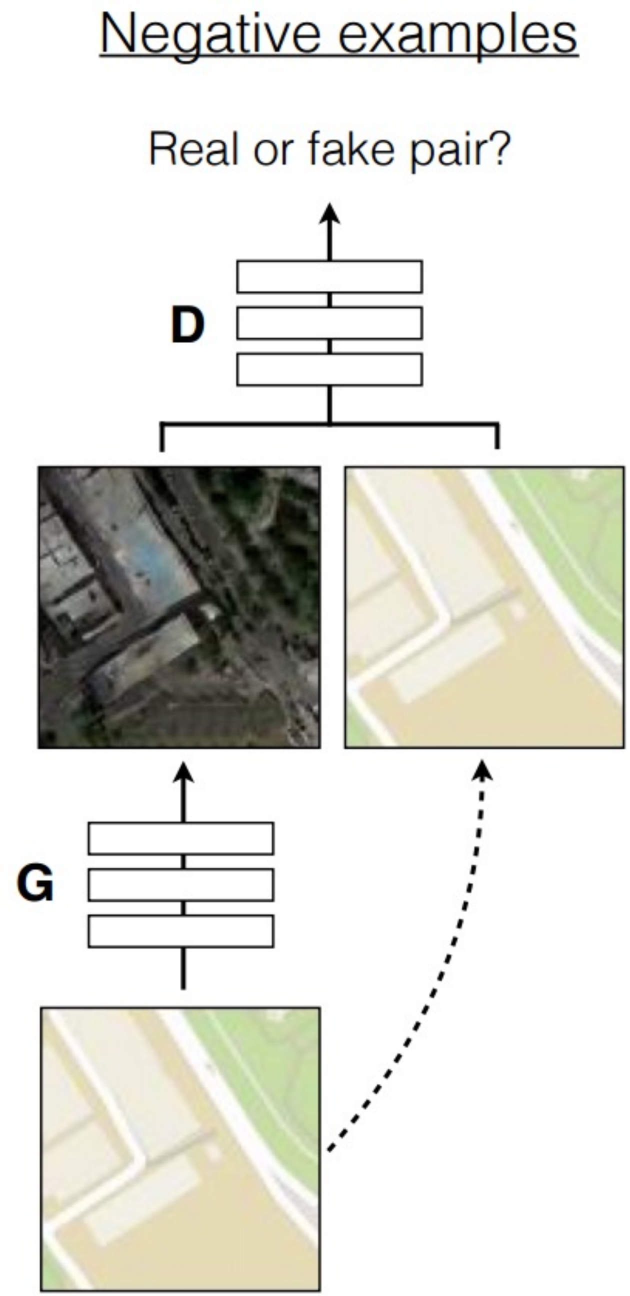
Image-to-image

- Conditional GAN



Testing:



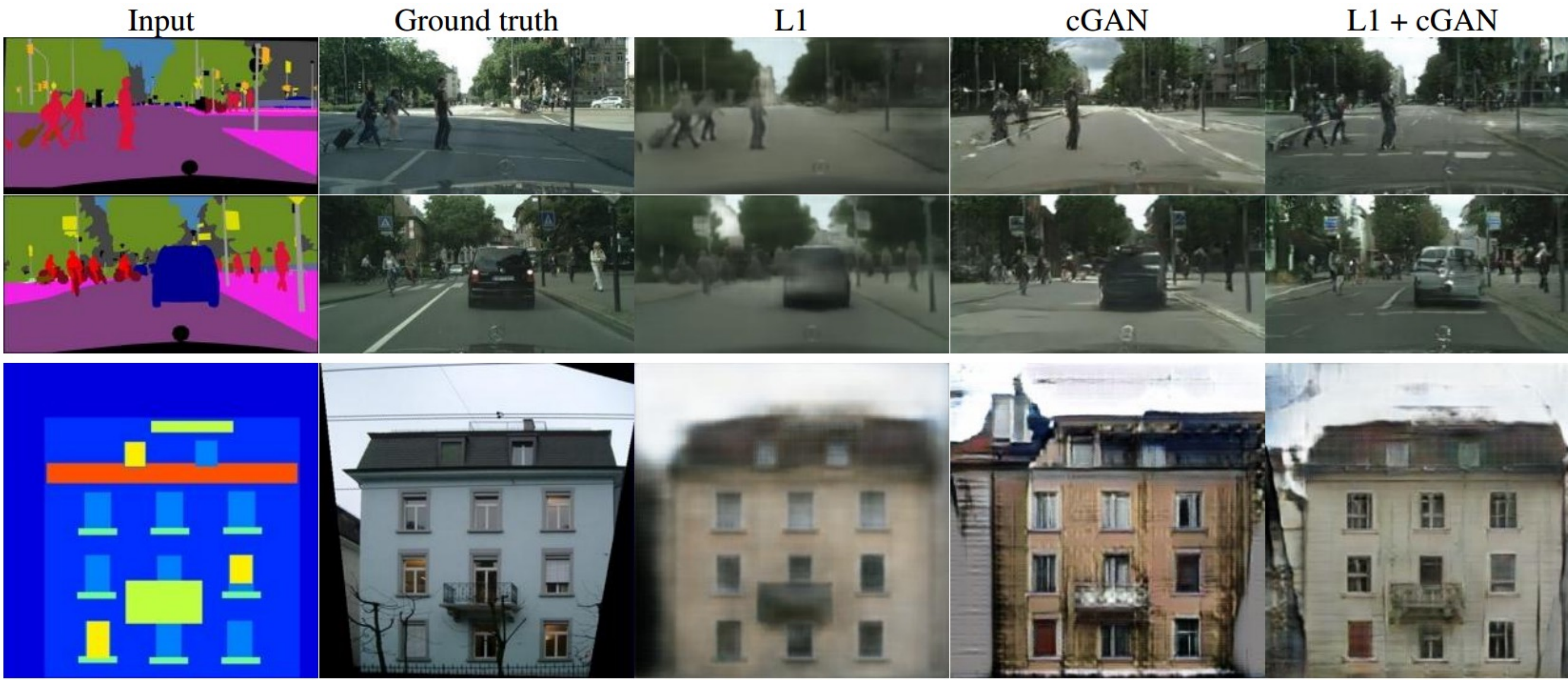


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

[Isola et al. CVPR 2017]

Label2Image



[Isola et al. CVPR 2017]

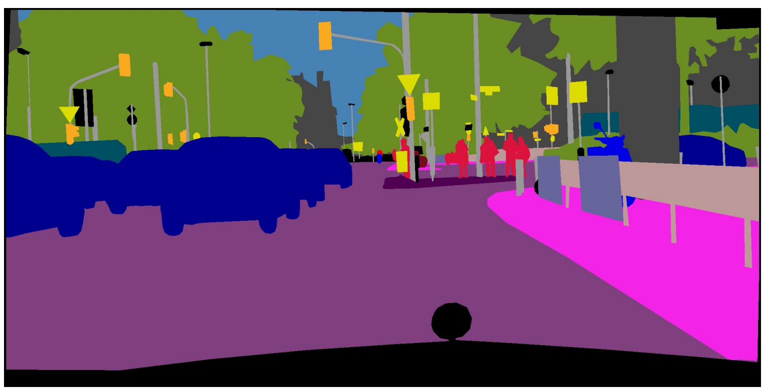
Edges2Image

[Isola et al. CVPR 2017]

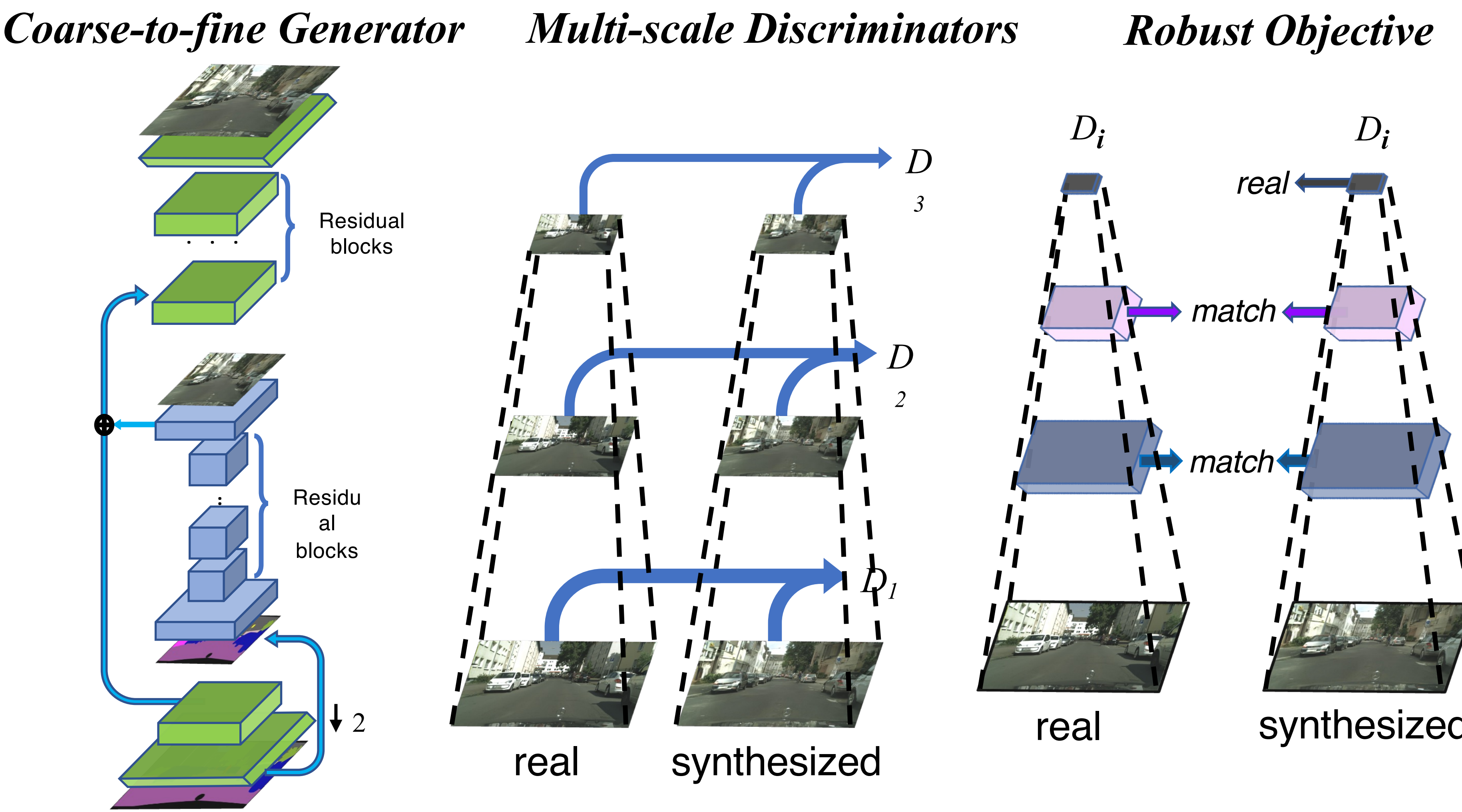
Pix2pixHD [CVPR 2018]

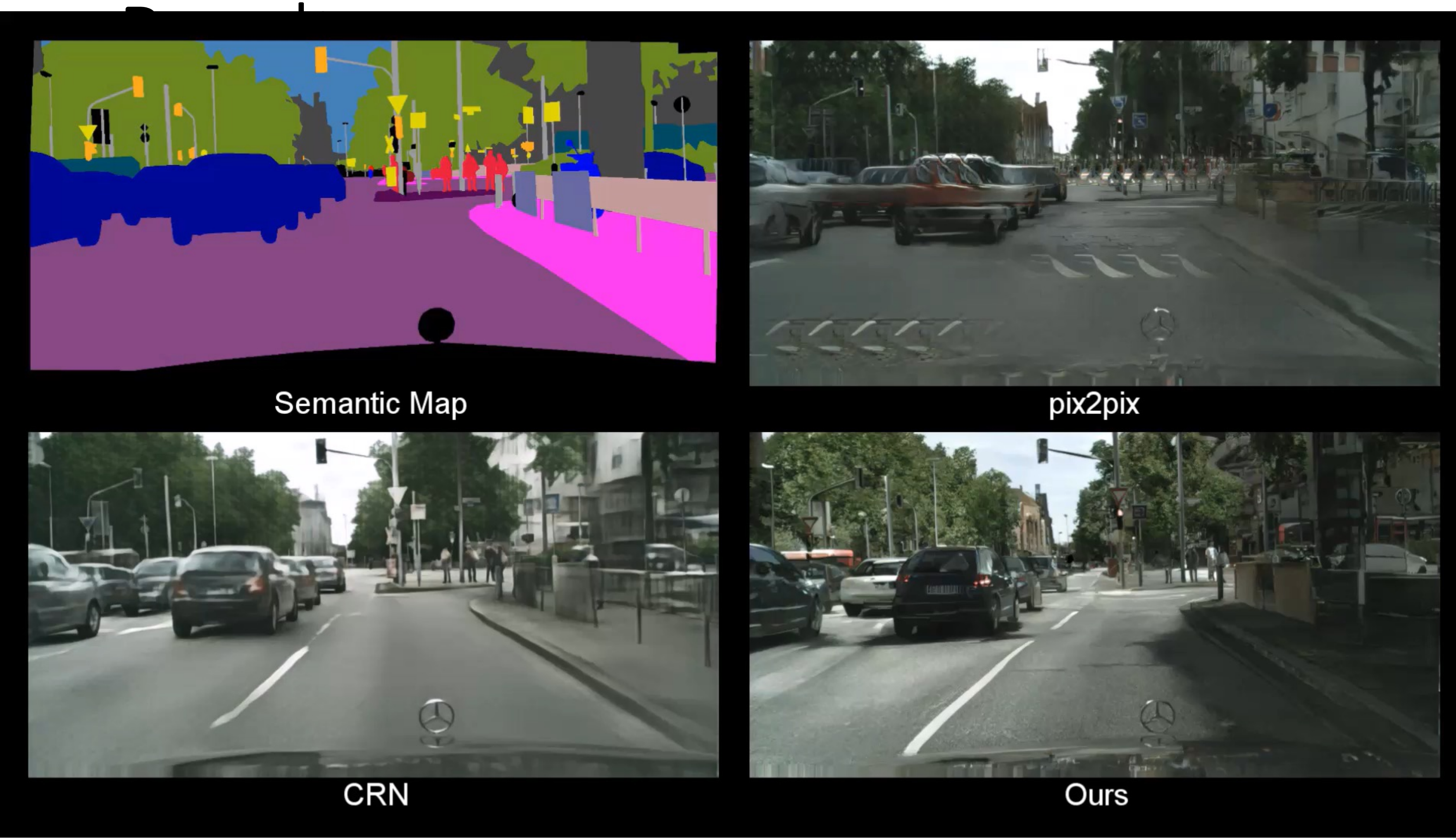
High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, Bryan Catanzaro



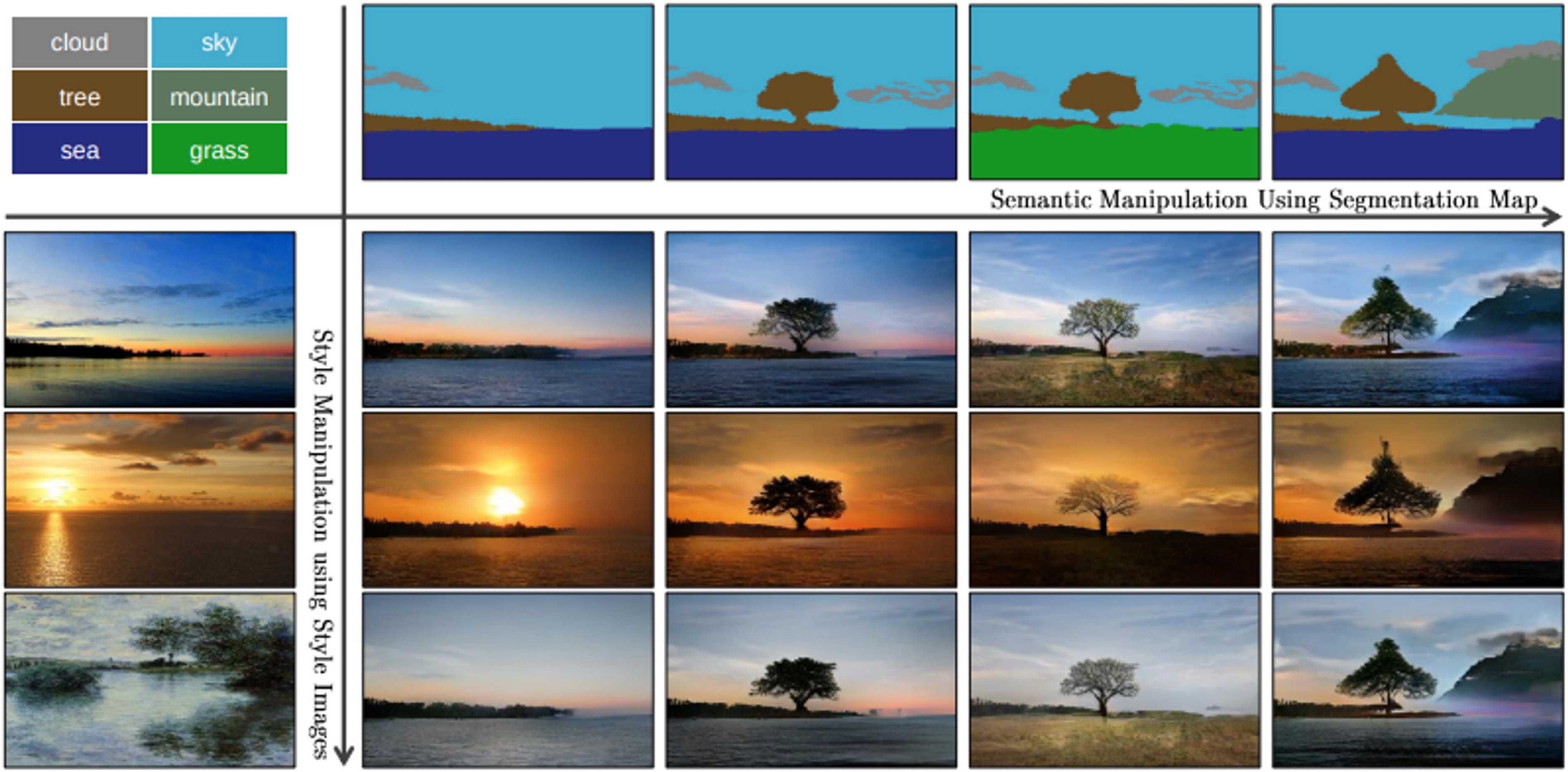
Pix2pixHD [CVPR 2018]




Semantic Map
pix2pix
CRN
Ours

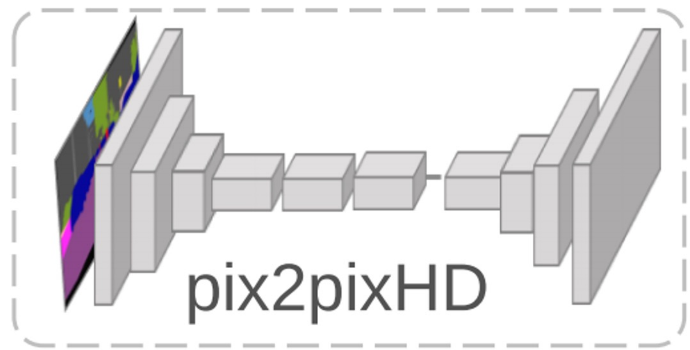
Improving Segmentation2Image strategy

[SPADE: Semantic Image Synthesis with Spatially-Adaptive Normalization, CVPR19]



Improving Segmentation2Image strategy

Previous approach:



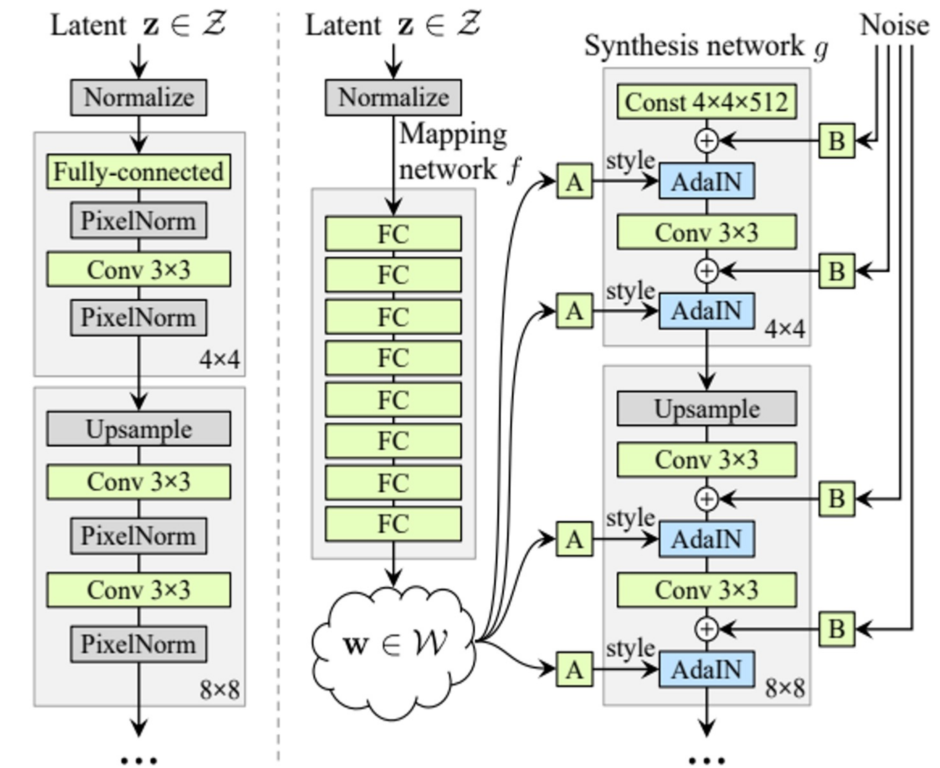
Directly feed the semantic layout as input to the deep network, which is processed through stacks of convolution, normalization, and nonlinearity layers.

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?



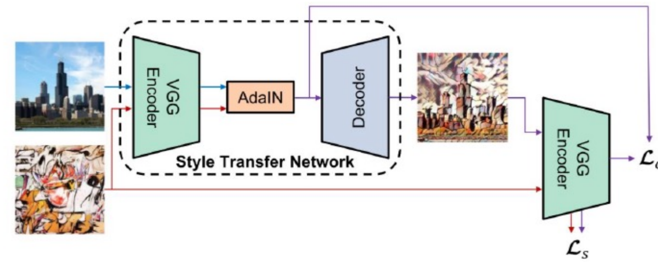
(a) Traditional

(b) Style-based generator

Improving Segmentation2Image strategy

Recall: Adaptive instance normalization

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



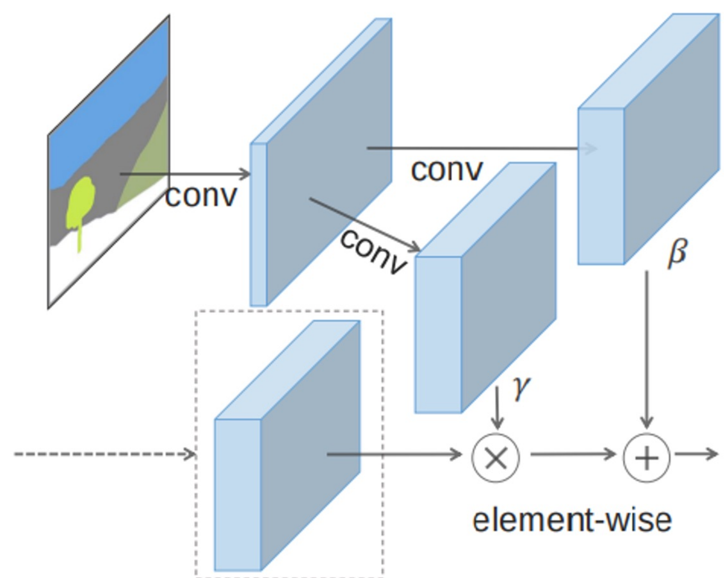
SPADE block= spatially-adaptive denormalization:

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

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

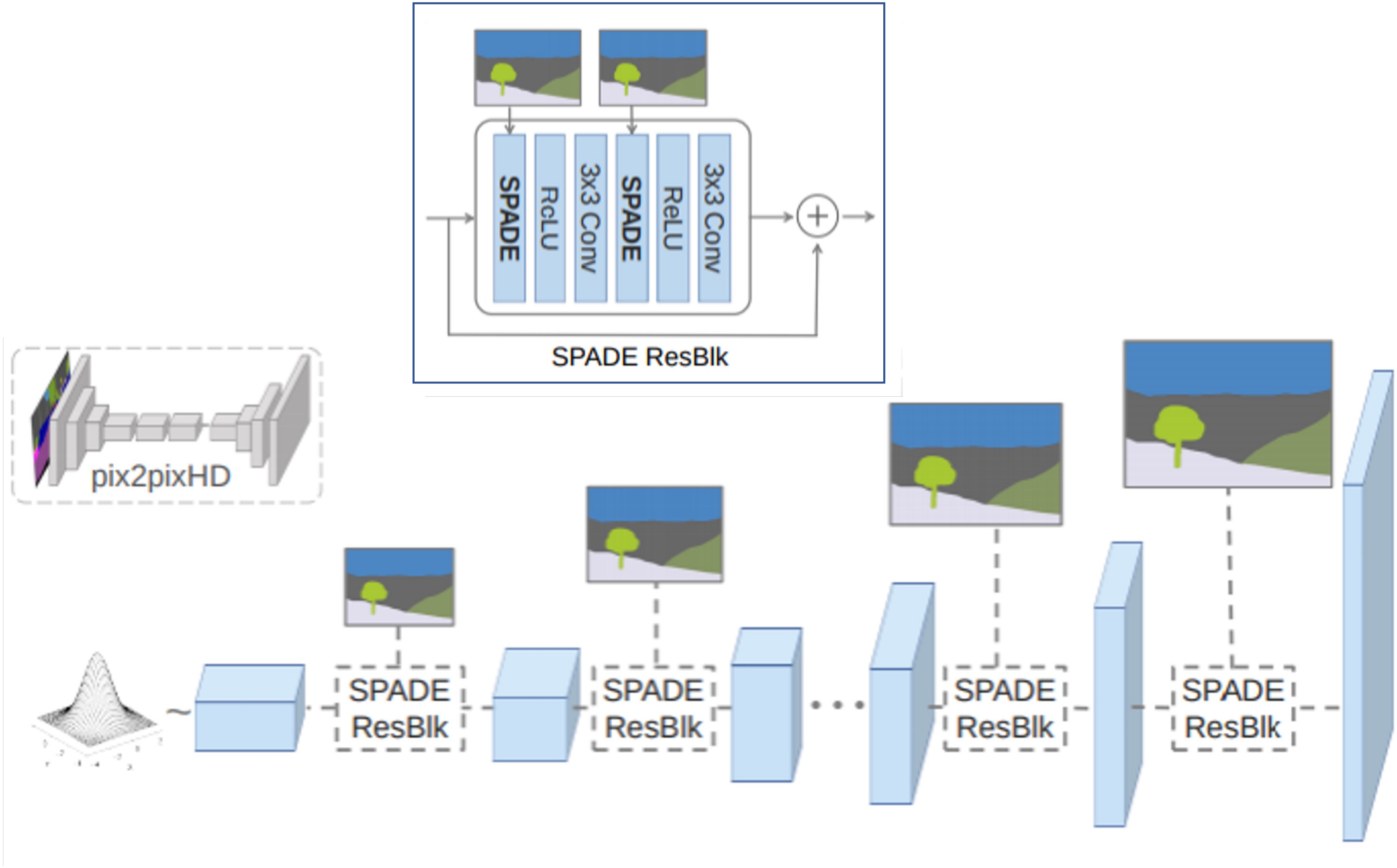
$$\mu^i_c = \frac{1}{NH^iW^i} \sum_{n,y,x} h^i_{n,c,y,x}$$

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

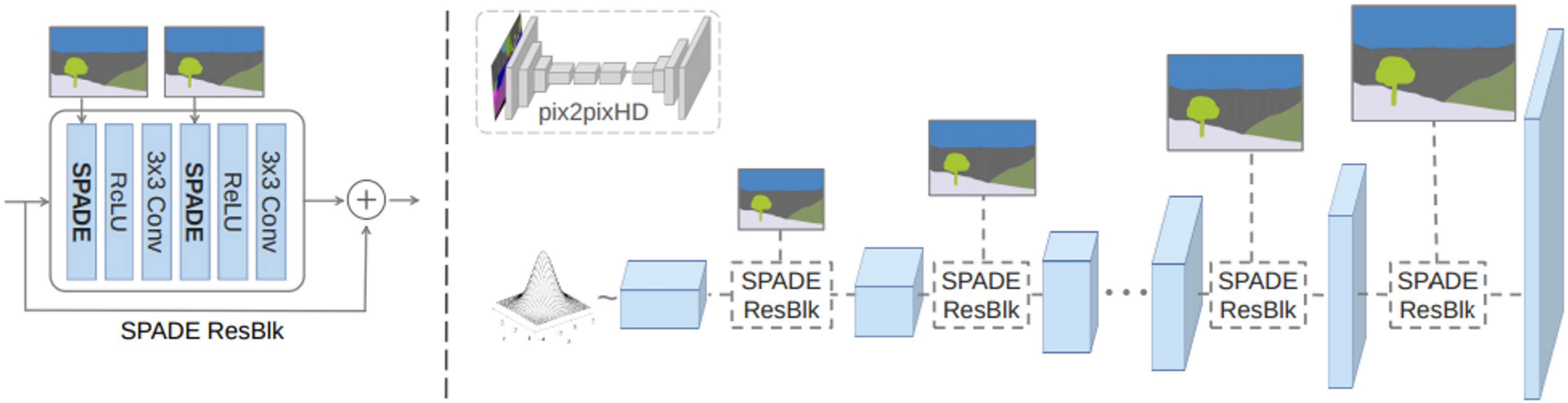


SPADE paper = [Semantic Image Synthesis with Spatially-Adaptive Normalization CVPR 2019]

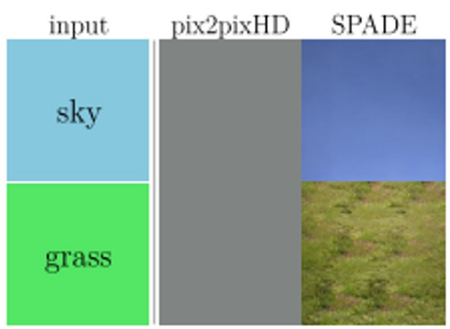
SPADE Generator



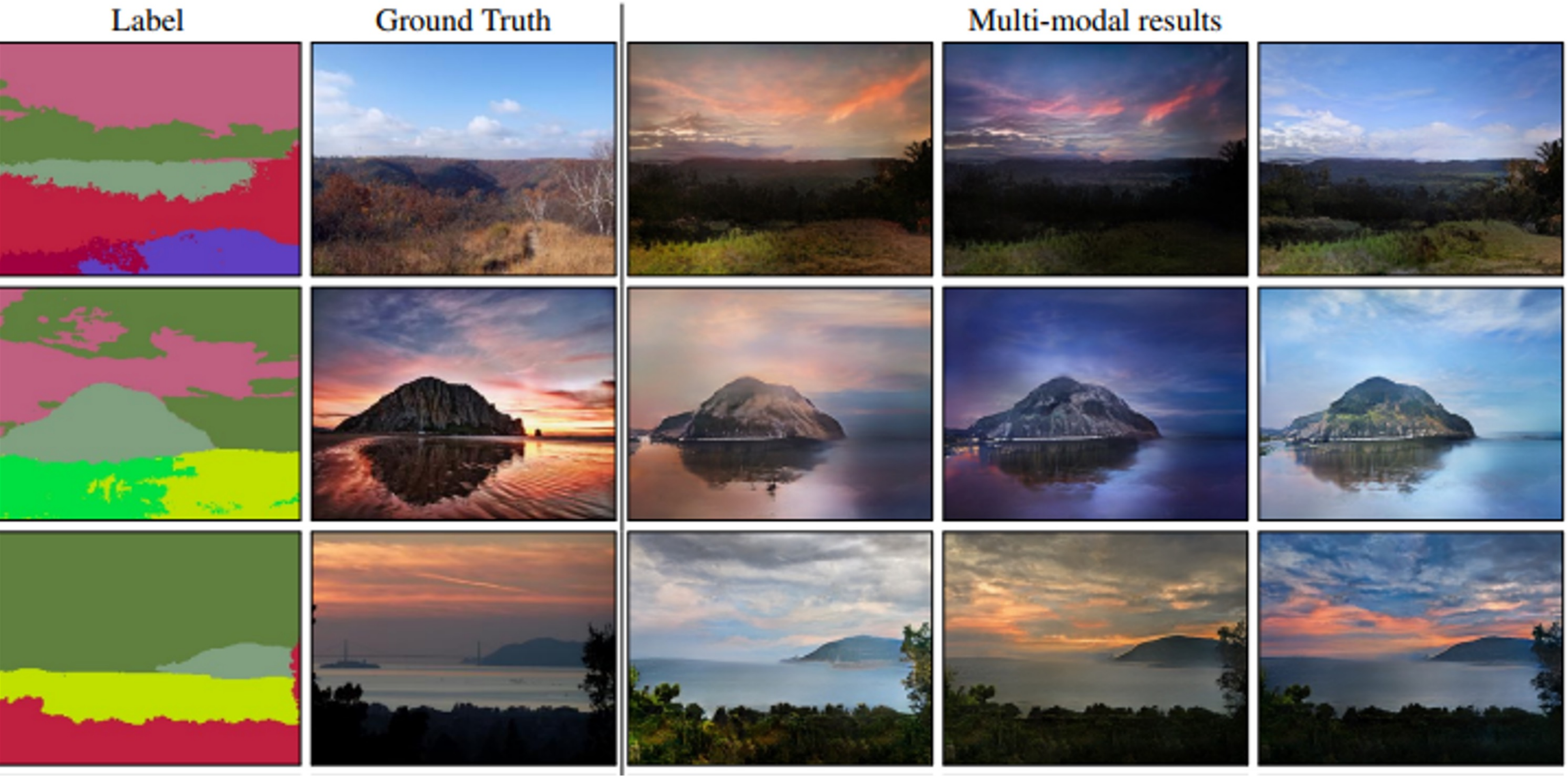
SPADE Generator



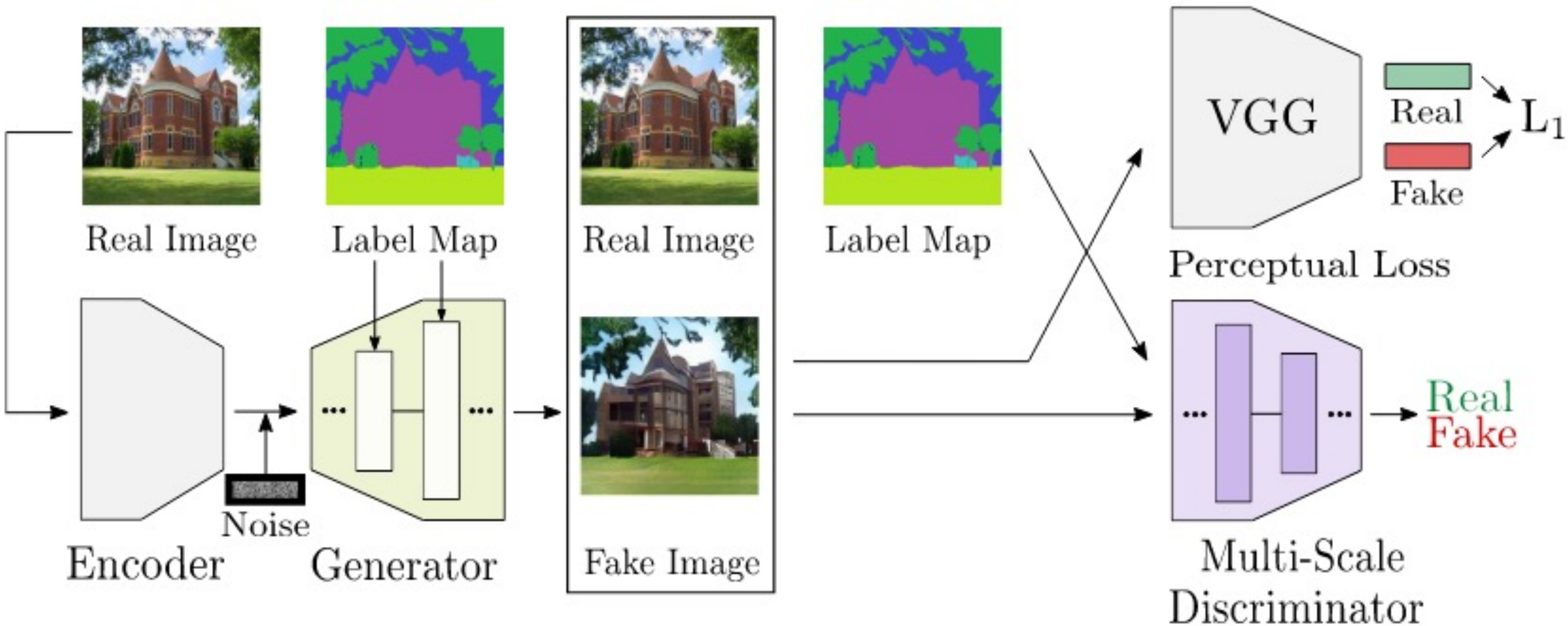
Better preserve semantic information against common normalization layers



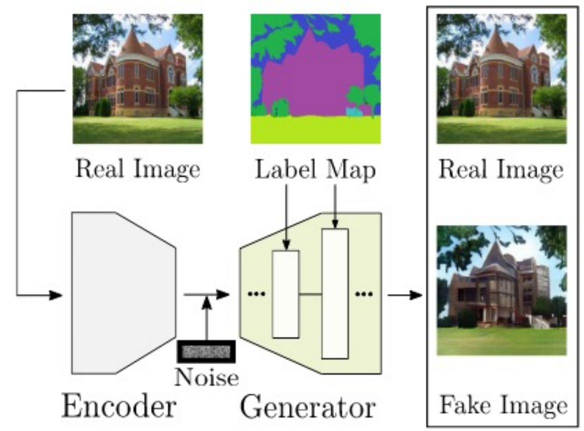
SPADE results

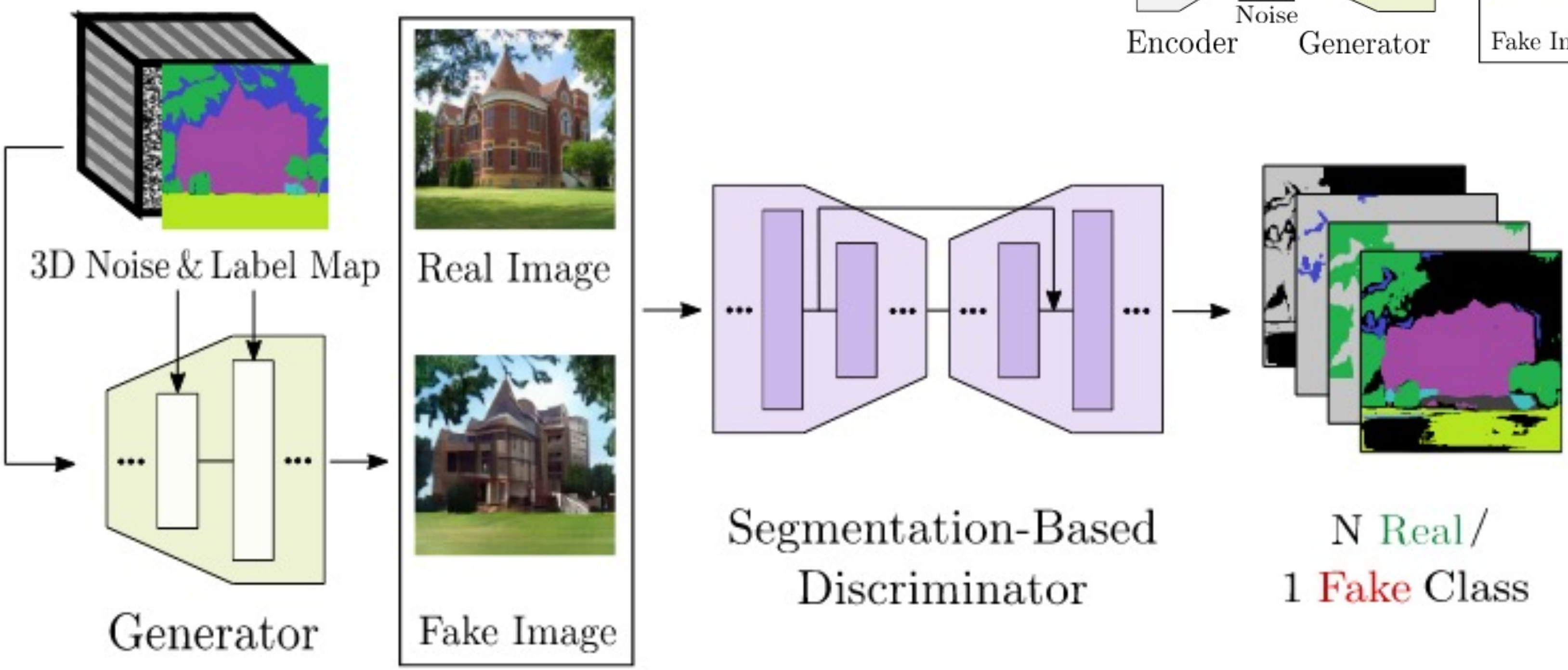


SPADE with real image:



[OASIS iclr 2021] (follow-up paper of SPADE) with real image:





[OASIS iclr 2021] (follow-up paper of SPADE) with real image:

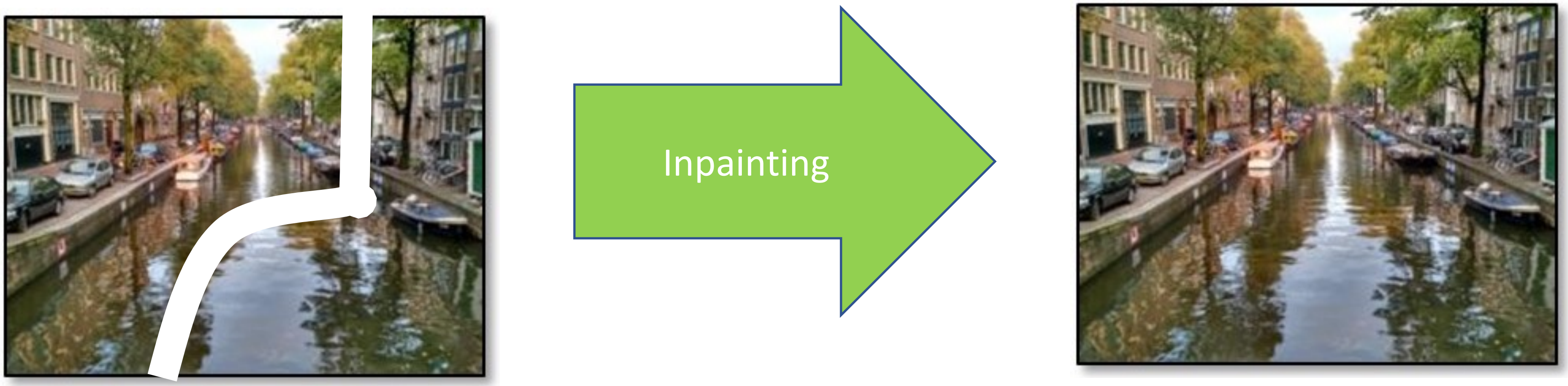
Generative models

Outline

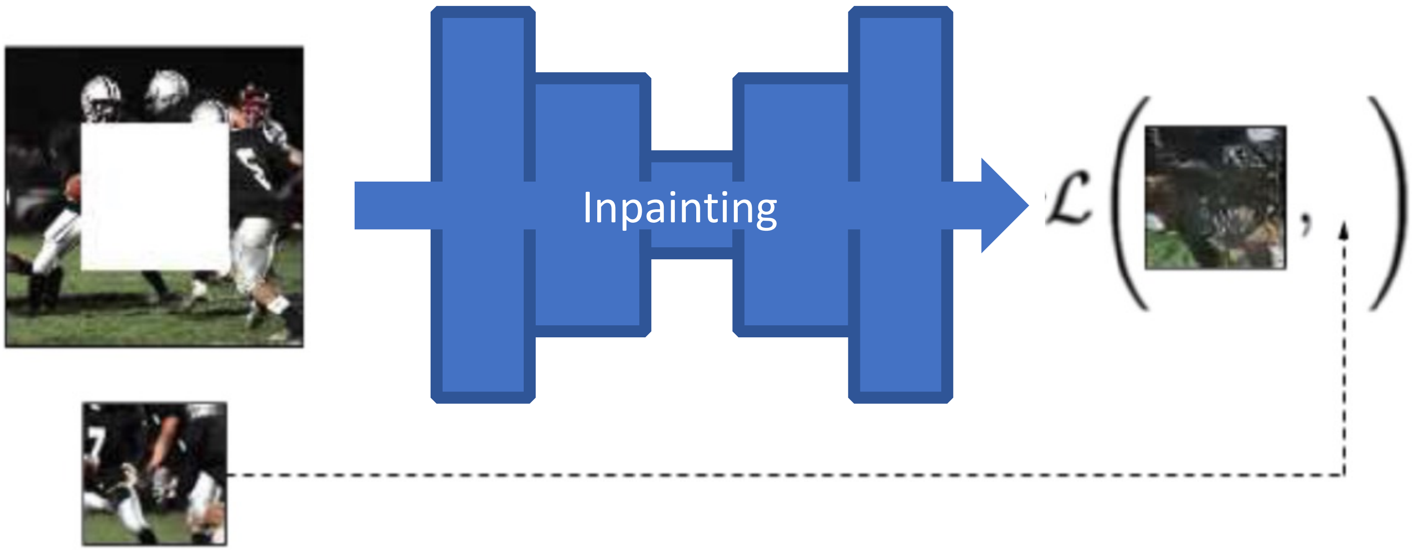
1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
3. GAN architectures
4. Editing
5. Conditional GANs
    1. Principle
    2. Text2Image
    3. Image2Image
    4. **Inpainting and general missing data encoder**

Inpainting task

- Complete the missing part



Inpainting as unsupervised learning with GAN loss

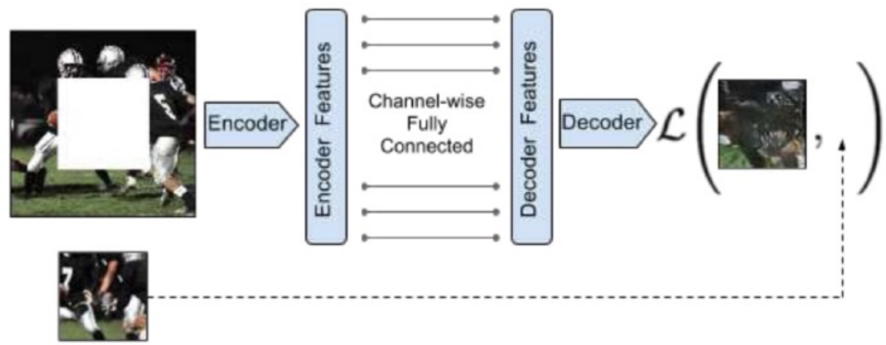


Reconstruct missing pixels by decoding using context

Loss defined on the predicted patch and the real one (known at training time)

First proposition -- Architecture

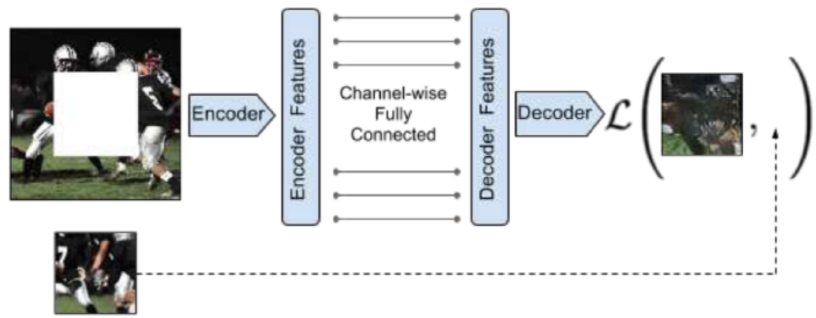
- Architecture: Encoder/Fully connected/Decoder



- 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

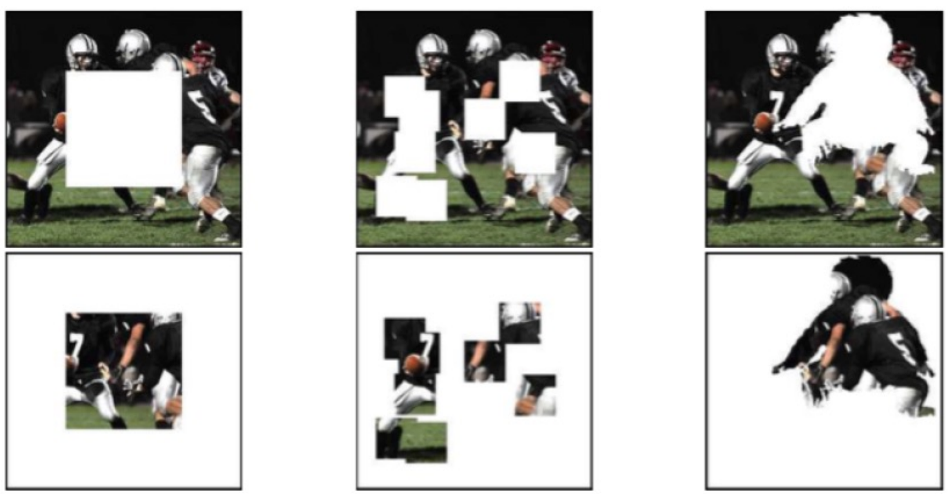


Decoder

- **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)

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

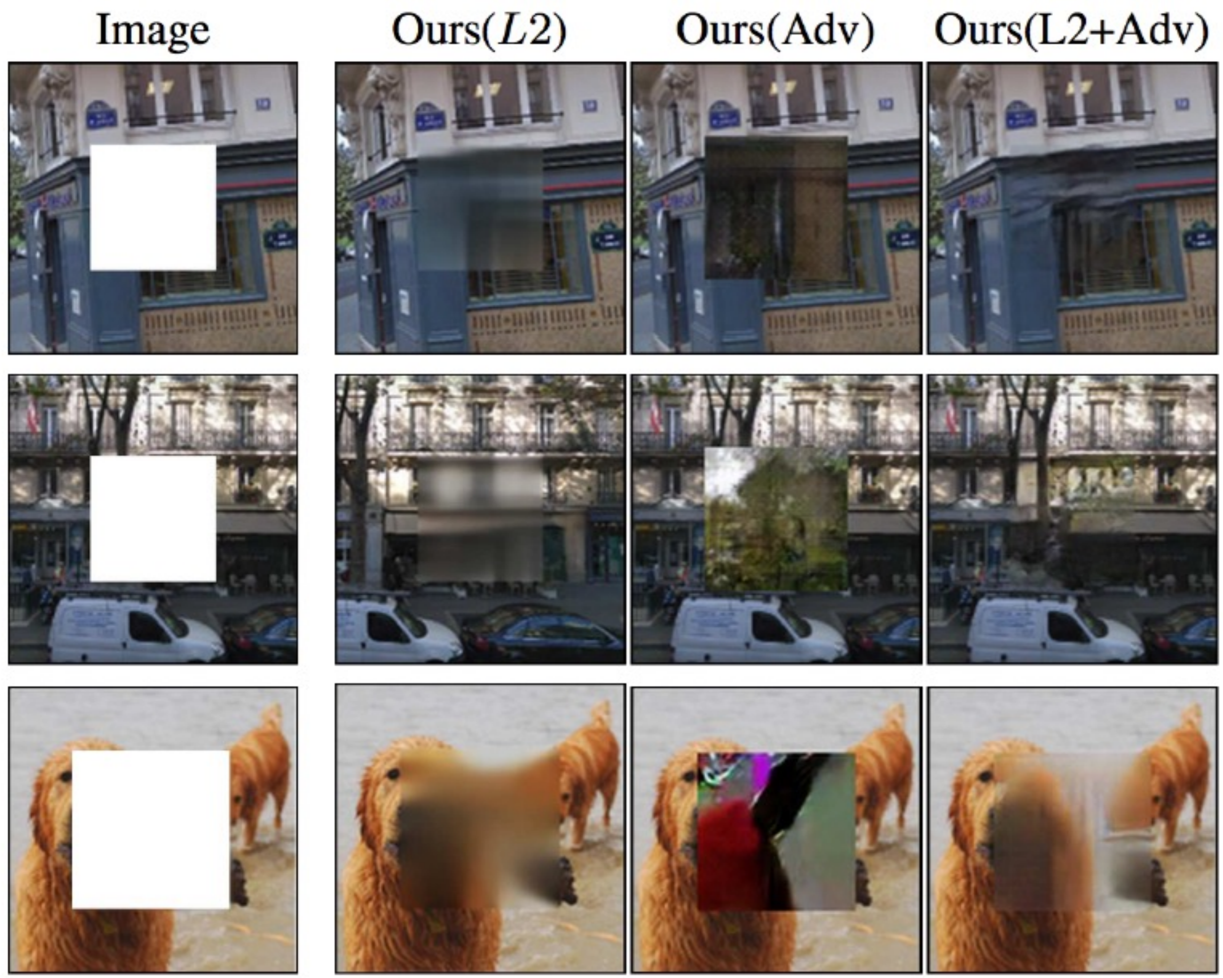
$$\mathcal{L}_{rec}(x) = \left\| \hat{M} \odot \left(x - F((1 - \hat{M}) \odot x) \right) \right\|_2$$

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

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

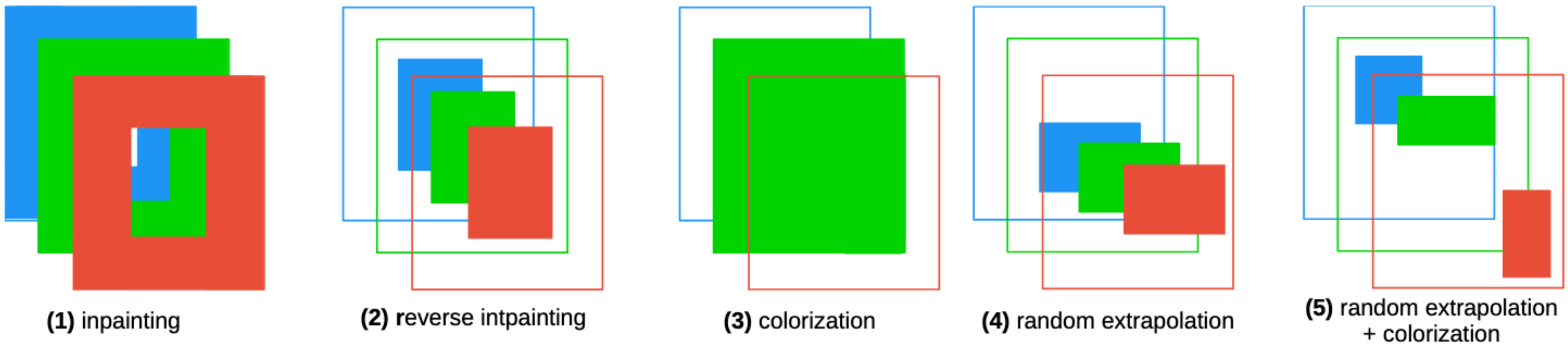
Results

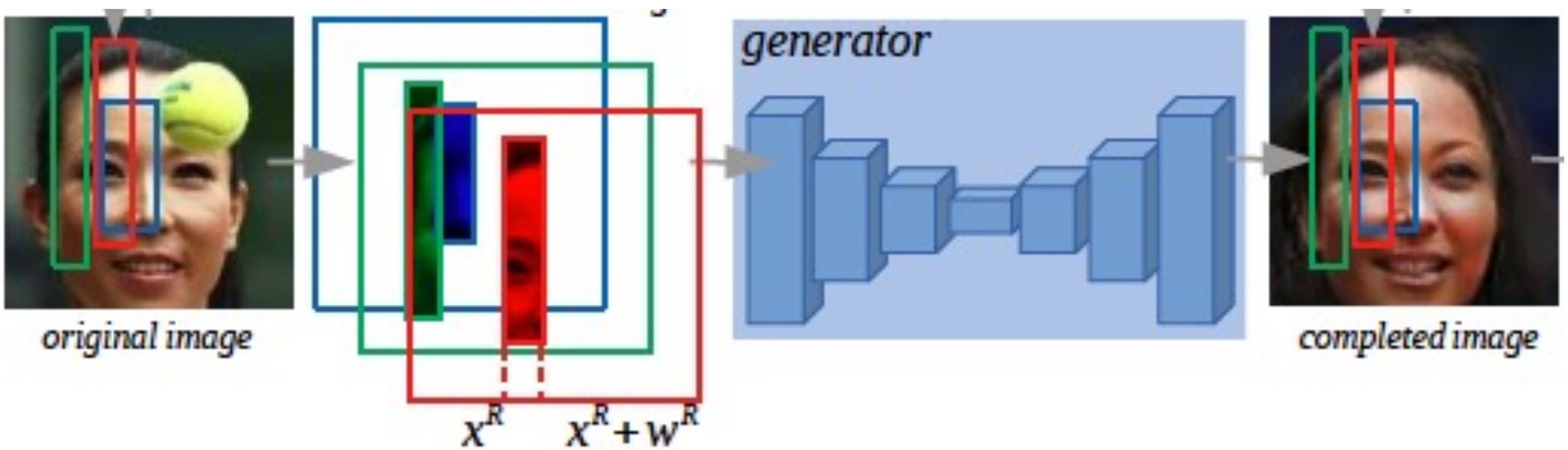
Dataset: StreetView Paris and ImageNet



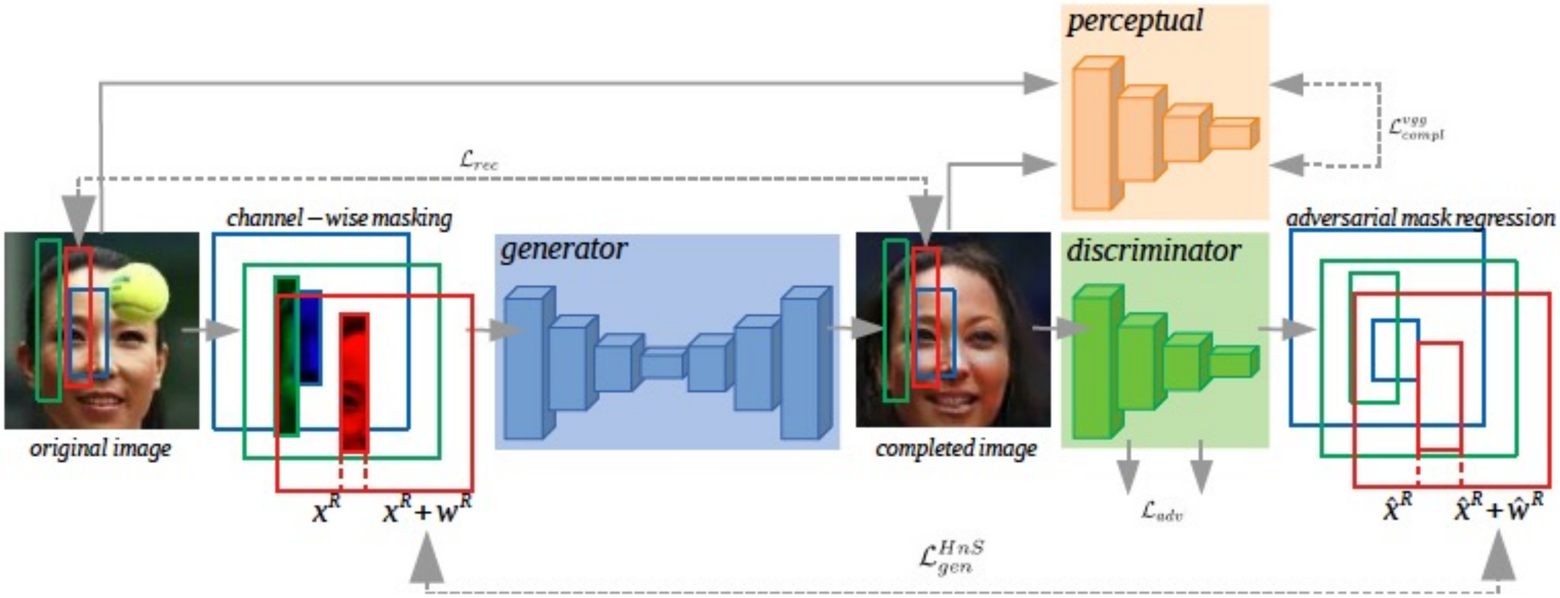
Semantic inpainting - Qualitative results

Generalizing inpainting: missing data encoder





Adding perceptual loss, BB regression loss



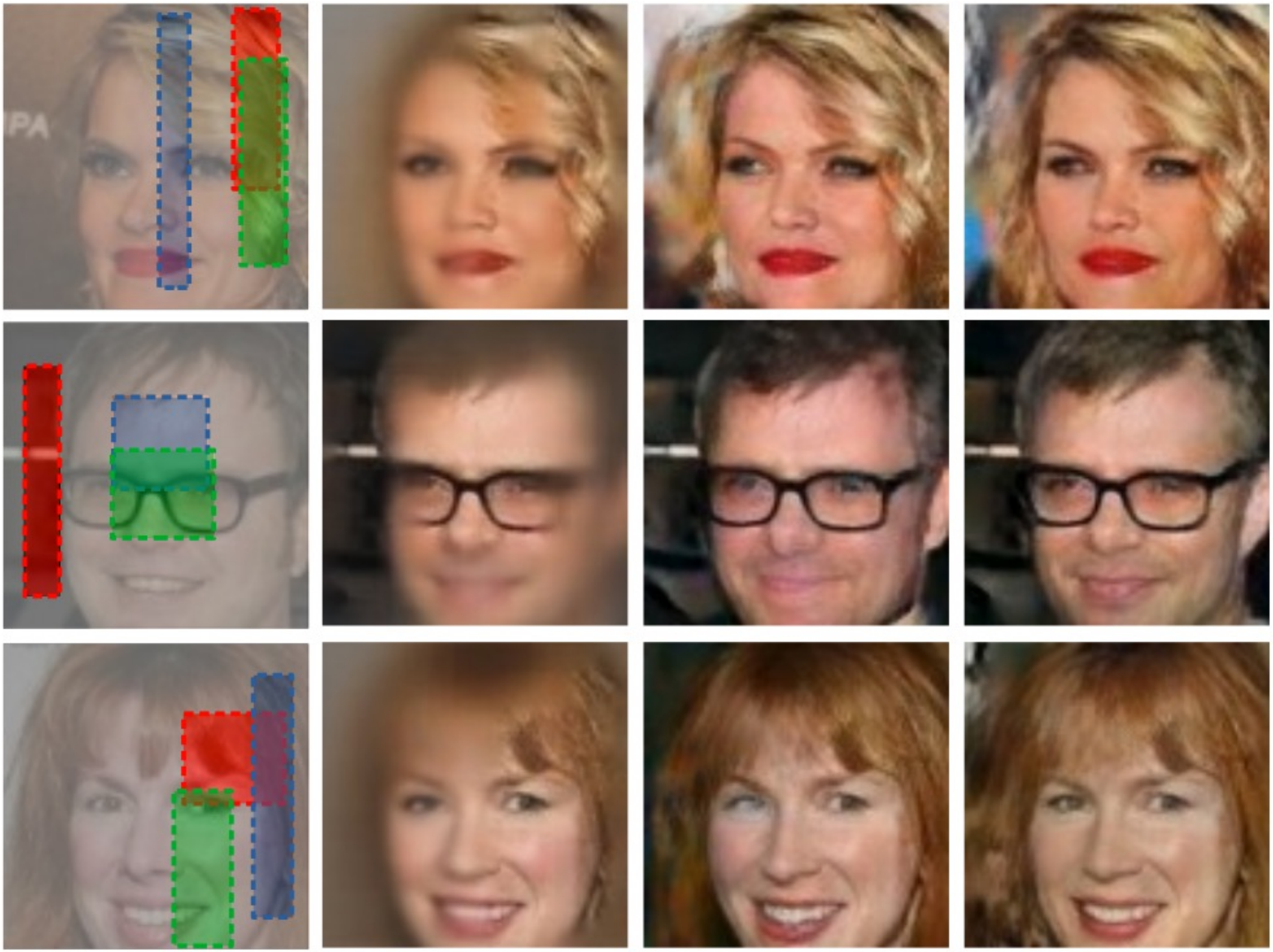
$$r^c = (x^c/W, y^c/H, (x^c + w^c)/W, (y^c + h^c)/H)$$

$$\mathcal{L}_{disc}^{HnS}(\theta_d) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\|r_i^c - \hat{r}_i^c(\theta_g, \theta_d)\|$$

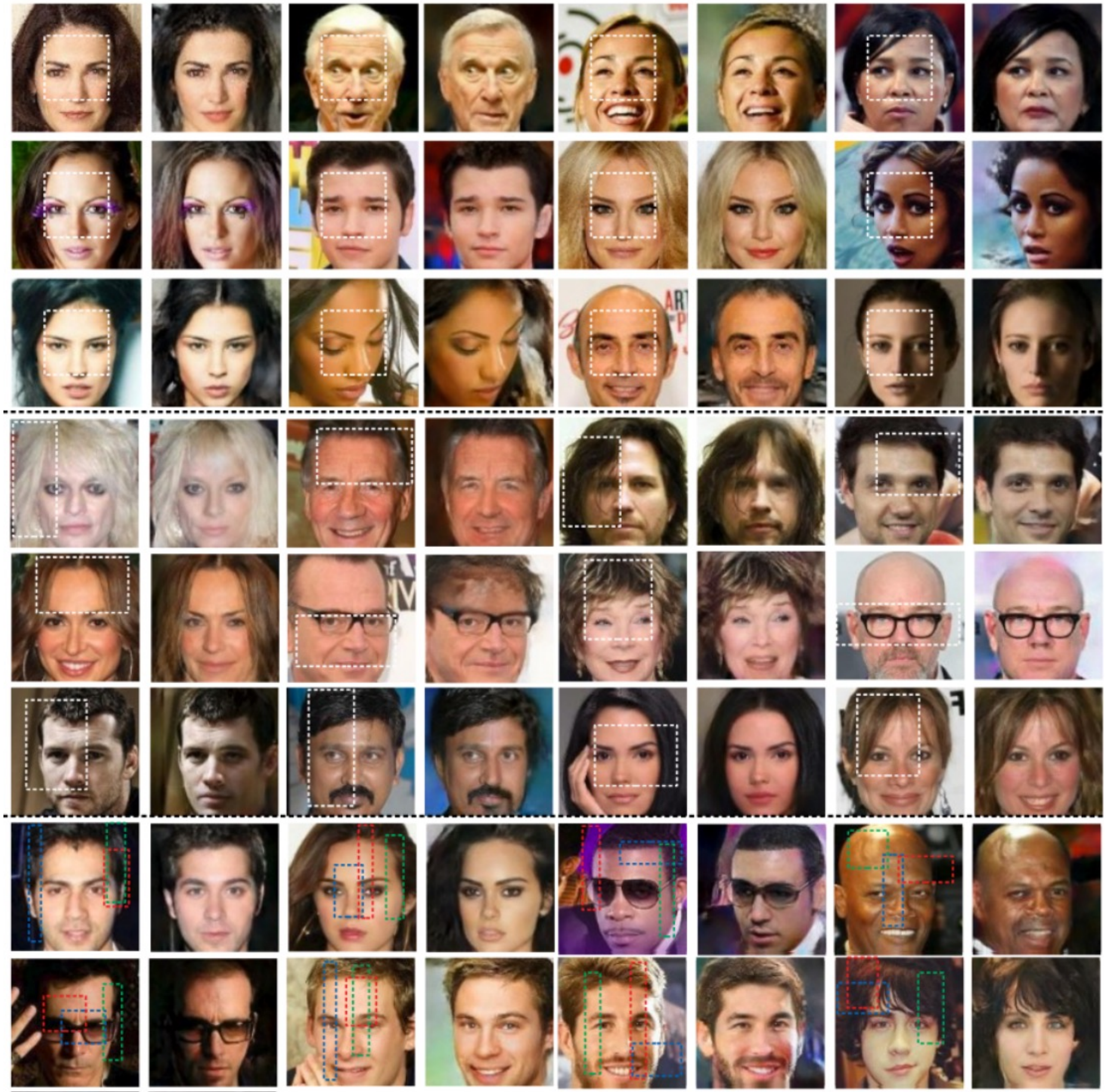
$$\mathcal{L}_{gen}^{HnS}(\theta_g) = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\|q_i^c - \hat{r}_i^c(\theta_g, \theta_d)\|$$

$$\mathcal{L}_{tot}(\theta_g, \theta_d) = \mathcal{L}_{rec}(\theta_g) + \lambda_{compl}\mathcal{L}_{compl}^{vgg}(\theta_g) + \lambda_{adv}\mathcal{L}_{adv}(\theta_g, \theta_d) + \lambda_{HnS}\mathcal{L}_{coord}^{HnS}(\theta_g, \theta_d)$$

Results



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
- Cycle GAN, Disco GAN

paired data

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

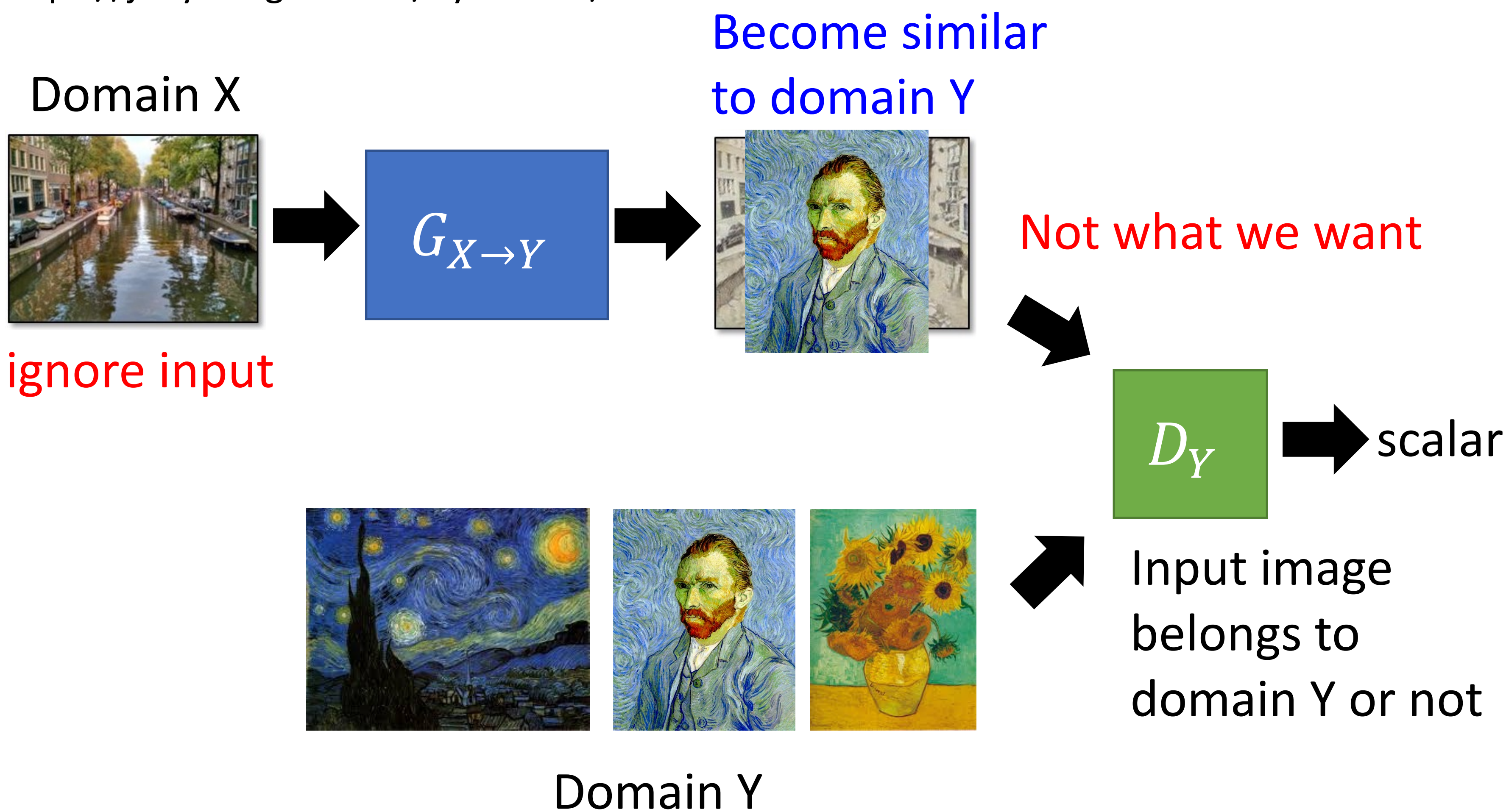
photo van Gogh

Domain X

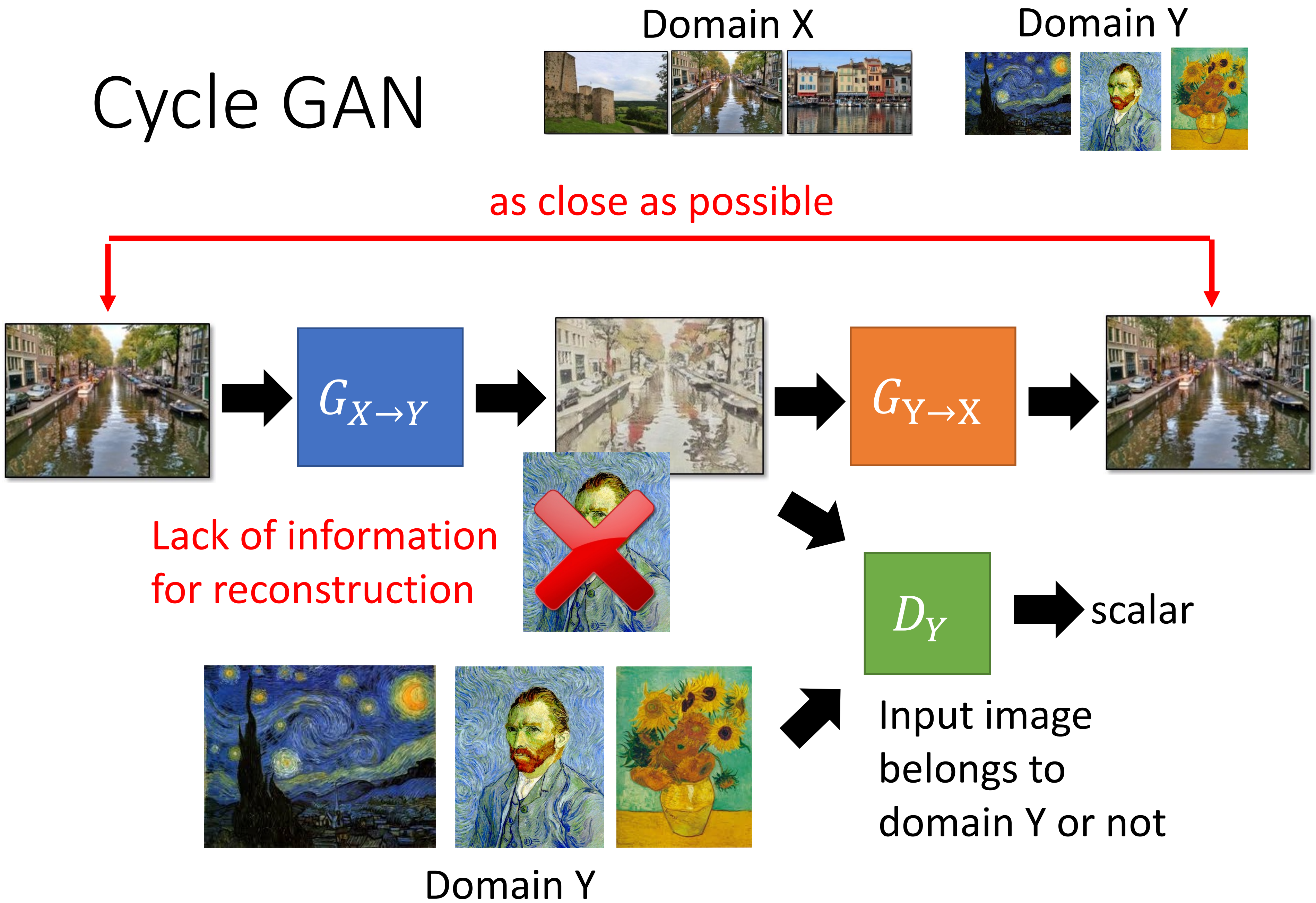
Domain Y

Cycle GAN

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



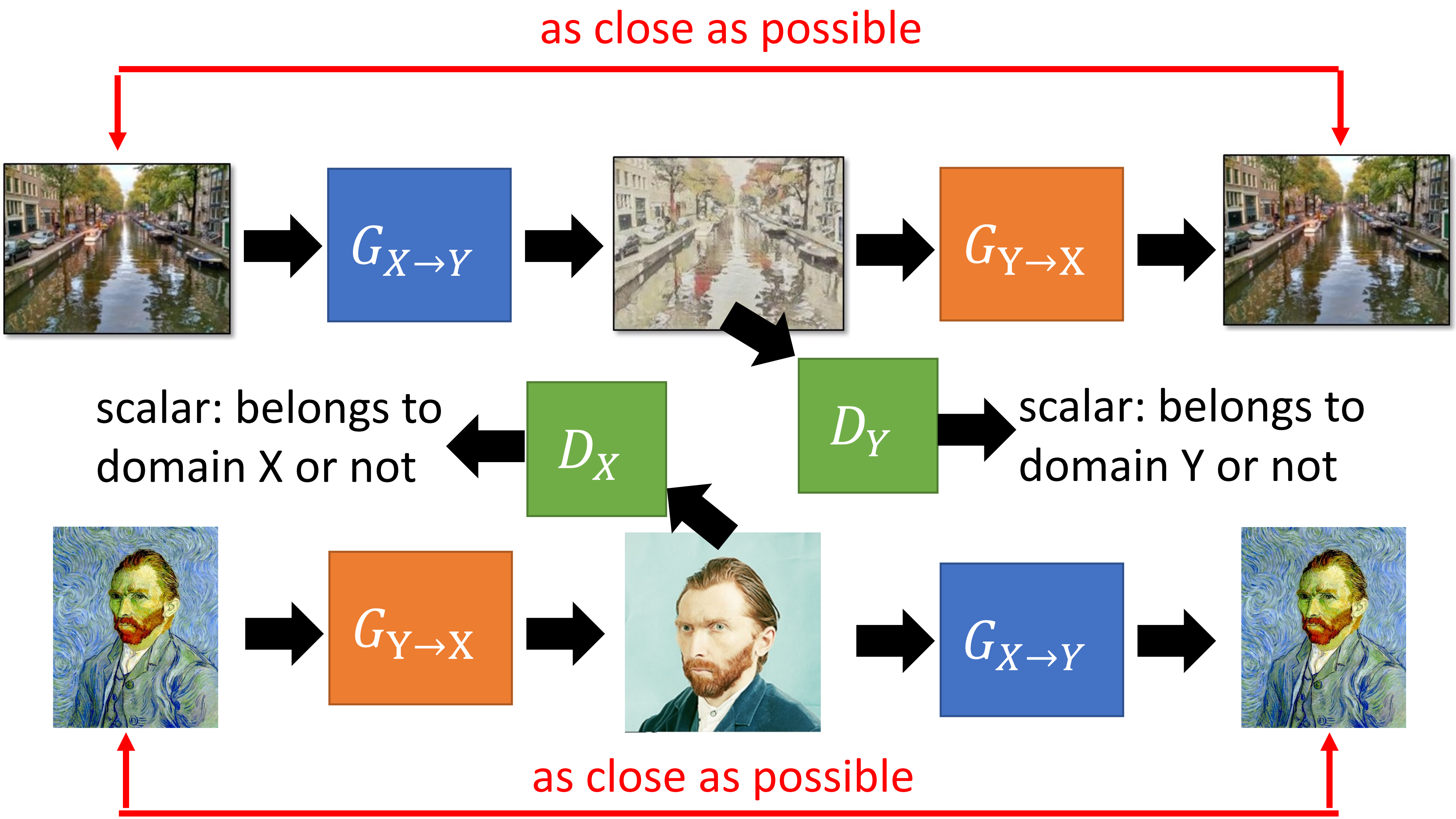
Cycle GAN



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

Results -- Cycle GAN

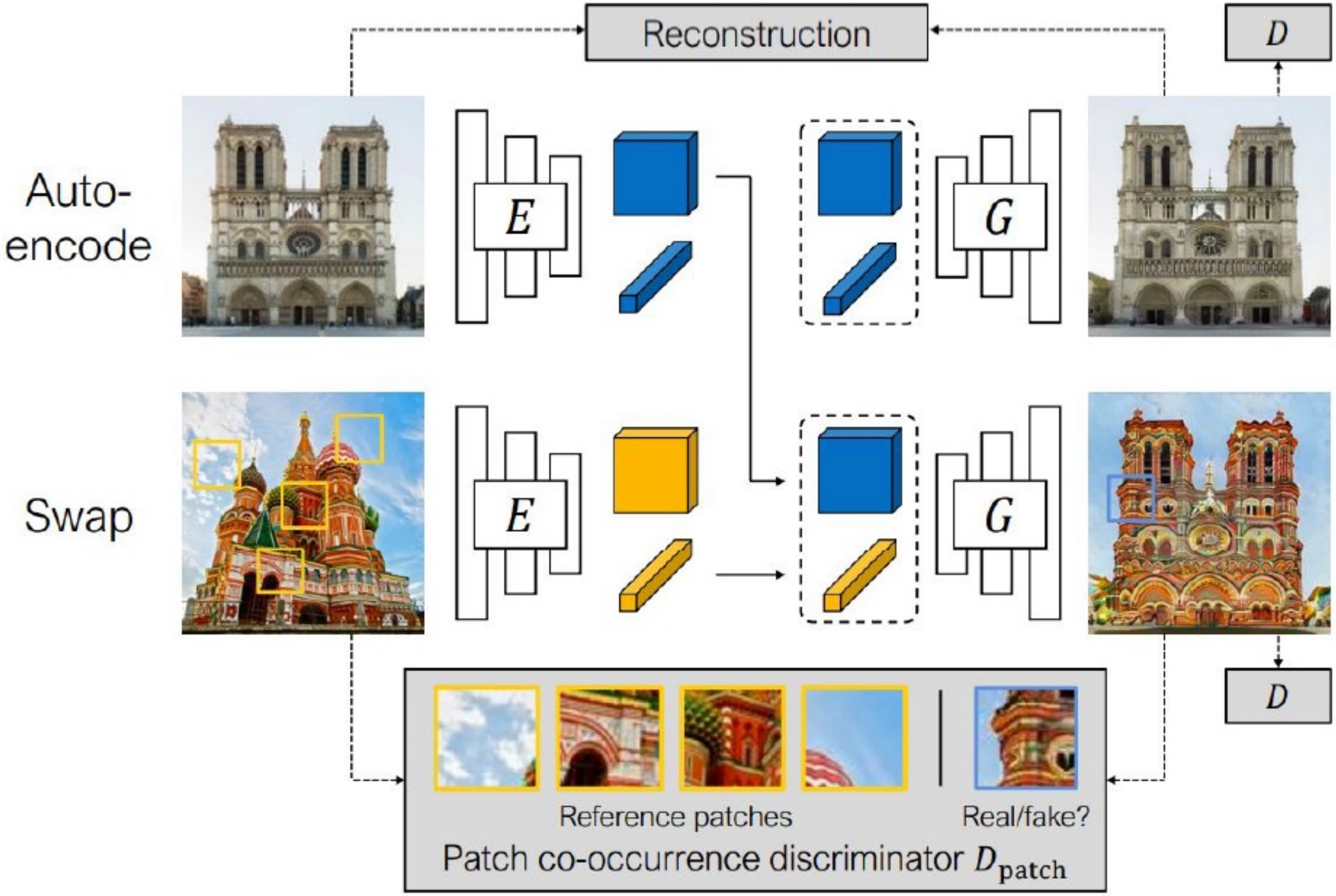
photo van Gogh

Domain X

Domain Y

GANs: works in progress

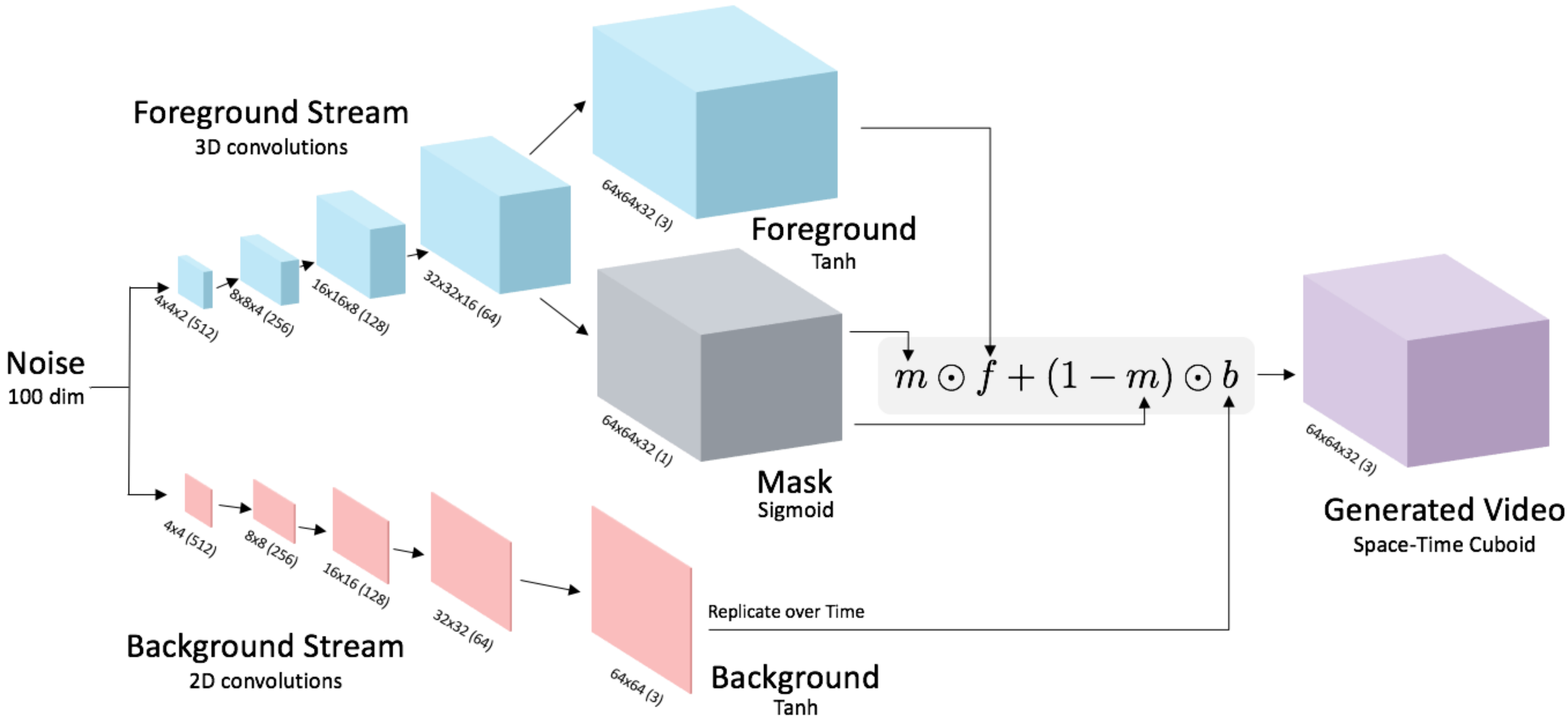
A lot of things to better understand, to use, adapt, …



Appendix

GANs for Video, 3D, etc.

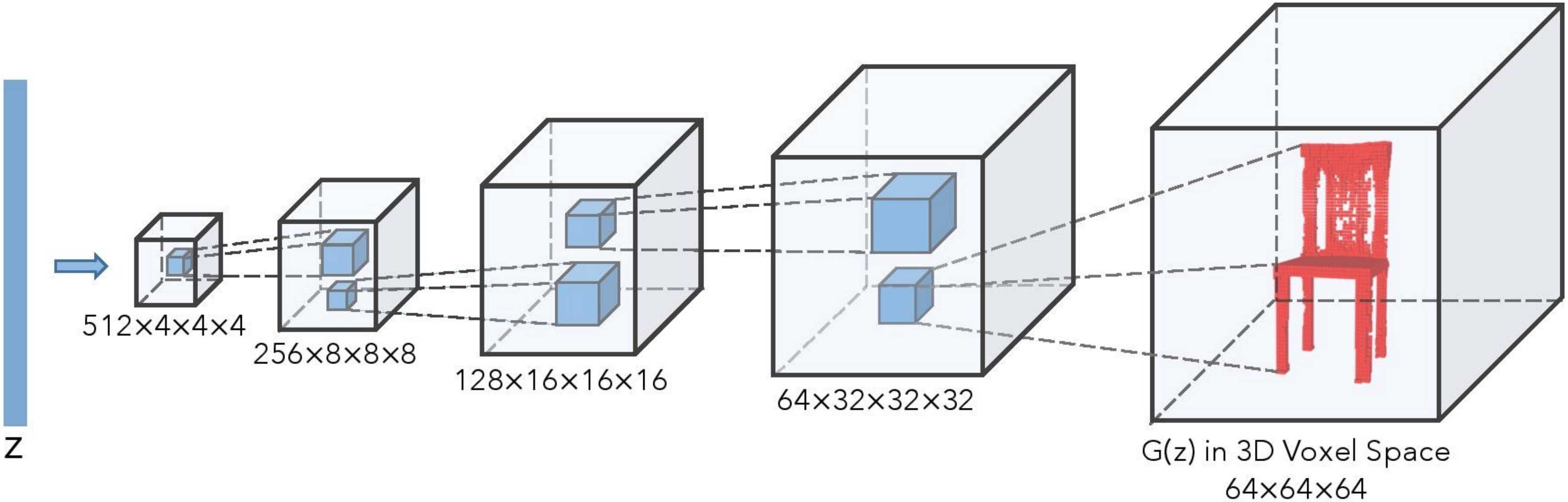
Video GAN



- Videos http://web.mit.edu/vondrick/tinyvideo/

[Vondrick et al. NIPS 2016]

Shape modeling using 3D Generative Adversarial Network



[Wu et al. NIPS 2016]