Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

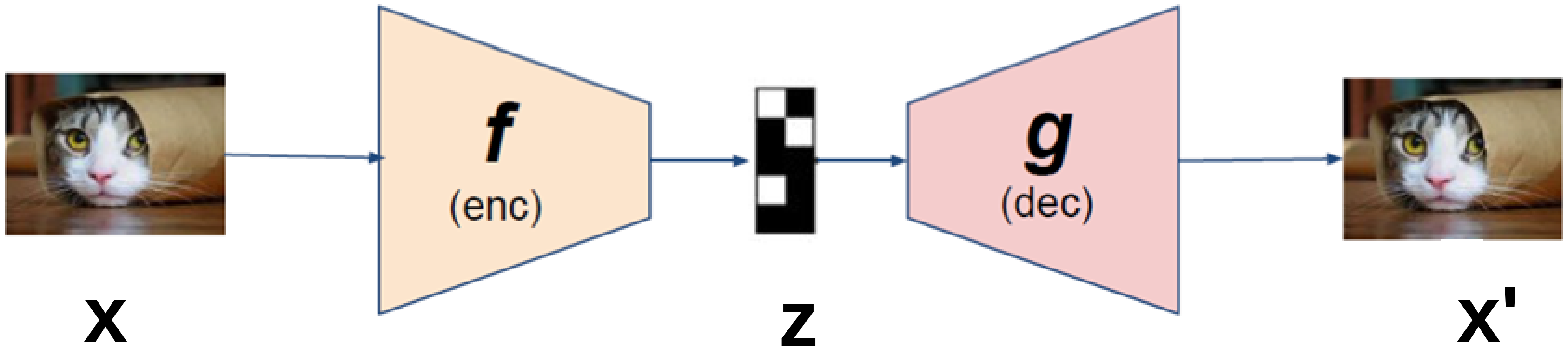
Distribution Learning using GANs

(A crash course)

Danda Pani Paudel
Computer Vision Lab @ ETH Zurich, NAAMII

Auto-Encoders:
(*As Generative Models*)

- Reconstruction with penalization on latent variables
  - $\min_{f}(f,g) = \mathbb{E}_{q(z)}[\log p(x|z)] - KL(q(z)||p(z))$



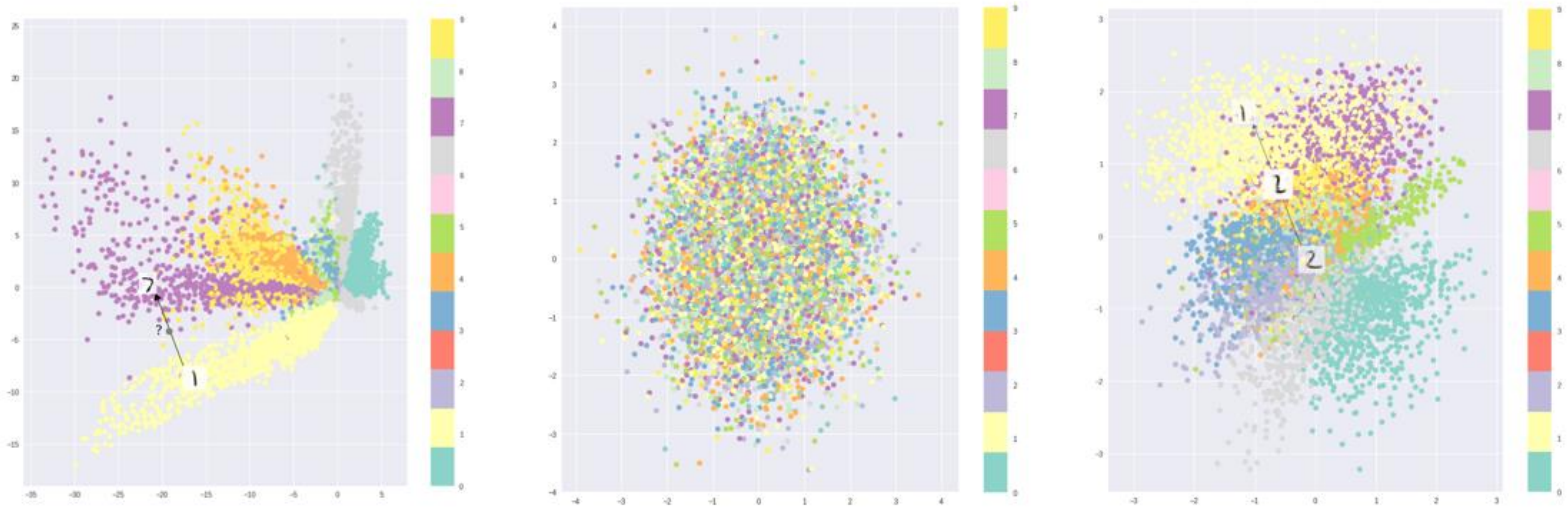
Supervised Learning
with L-2 loss (MSE:=||x-x'||) + KL Regularizer

Divergence and Reconstruction loss

Only reconstruction loss

Only KL divergence

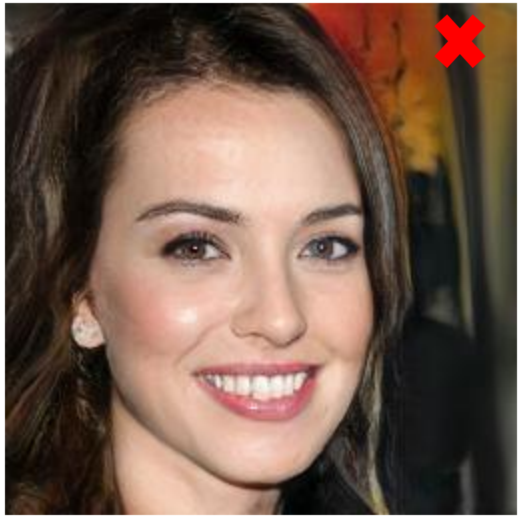
Combination



We all have friends we love dearly that couldn't pass for human in a strict Turing test.

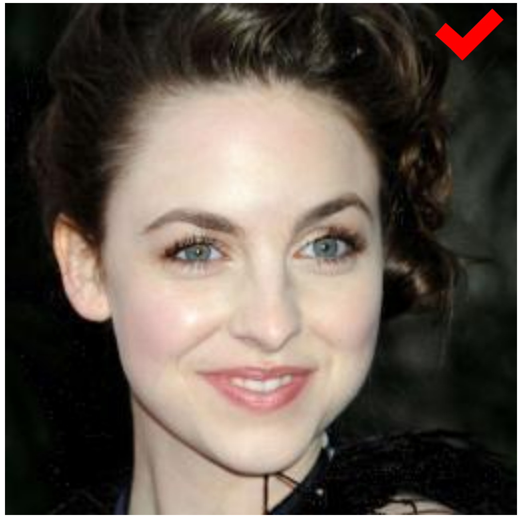
— *Penn Jillette* —

Real or Fake?


A

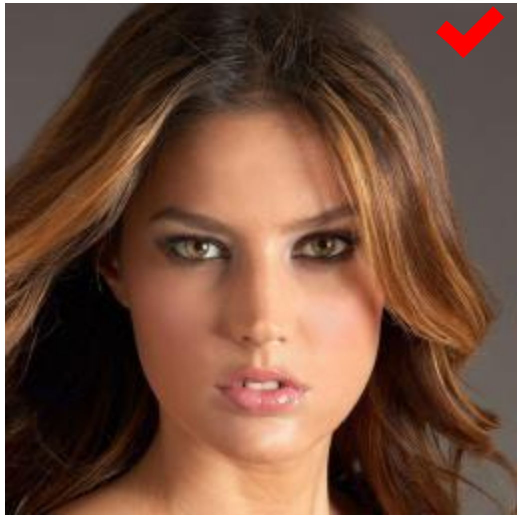

B


C


D

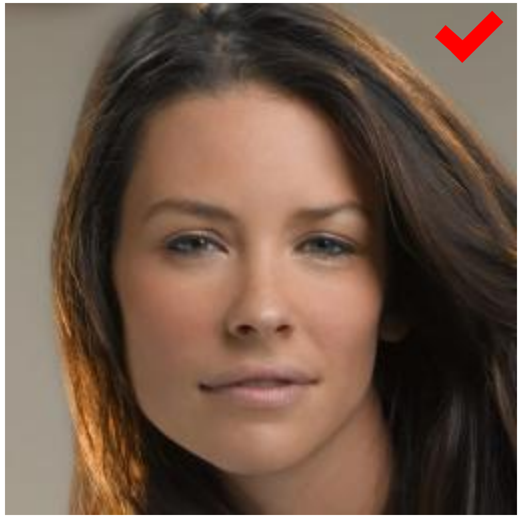

E


F


G


H

Real or Fake?


A


B


C


D


E


F


G

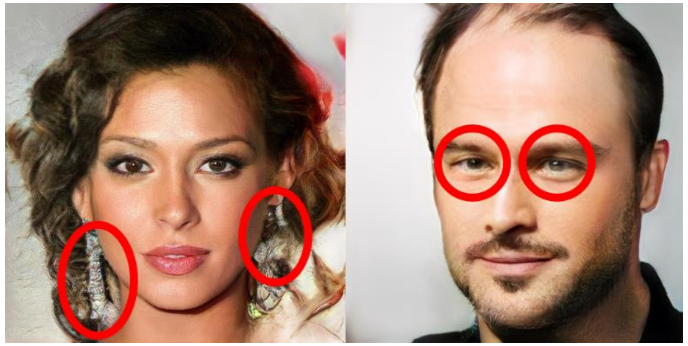

H

Hints for Real or Fake?

Text is uninterpretable

Background is surreal

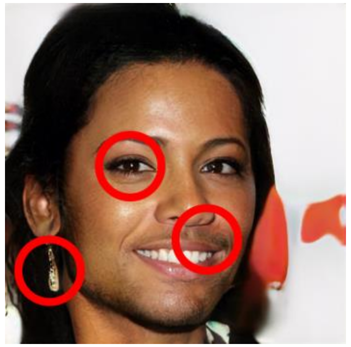
Asymmetry



Weird teeth

Messy hair

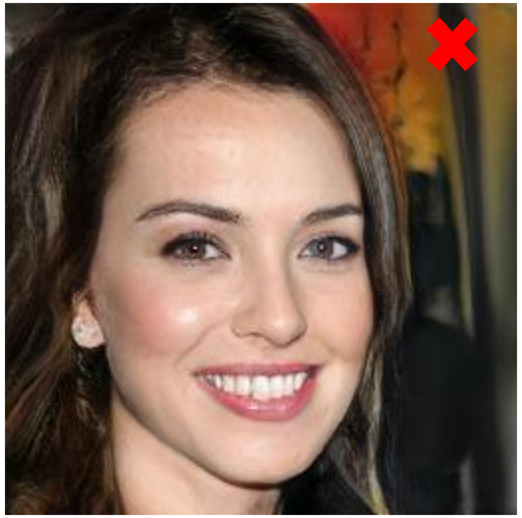
Non-stereotypical gender



Images from https://medium.com/@kcimc/how-to-recognize-fake-ai-generated-images-4d1f6f9a2842

✔ ✖

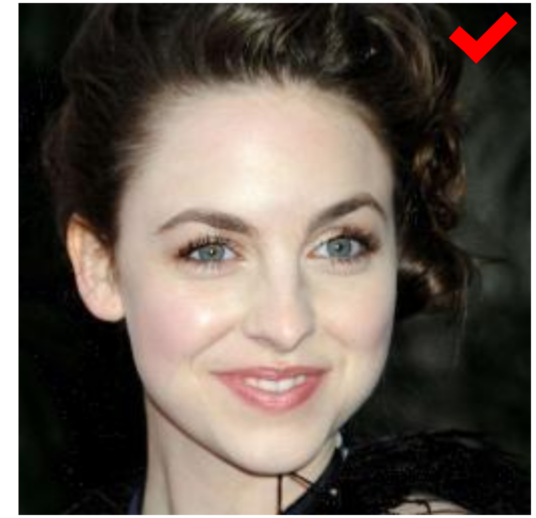
Real or Fake?



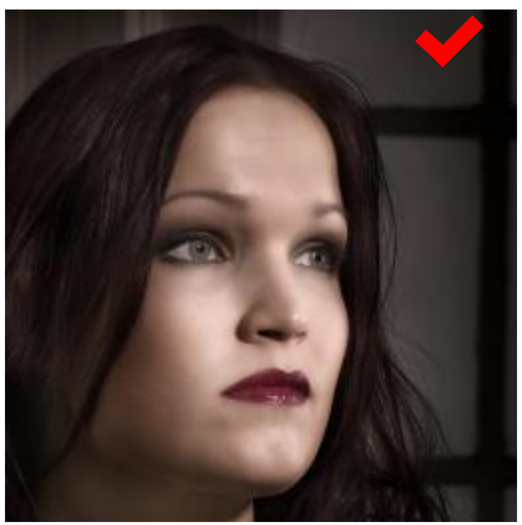
A

B

C

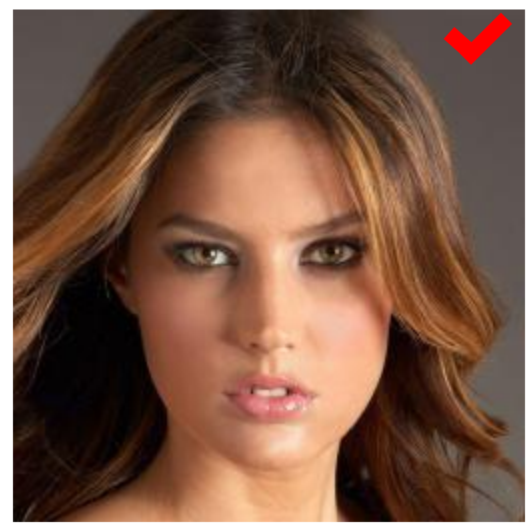


D



E

F



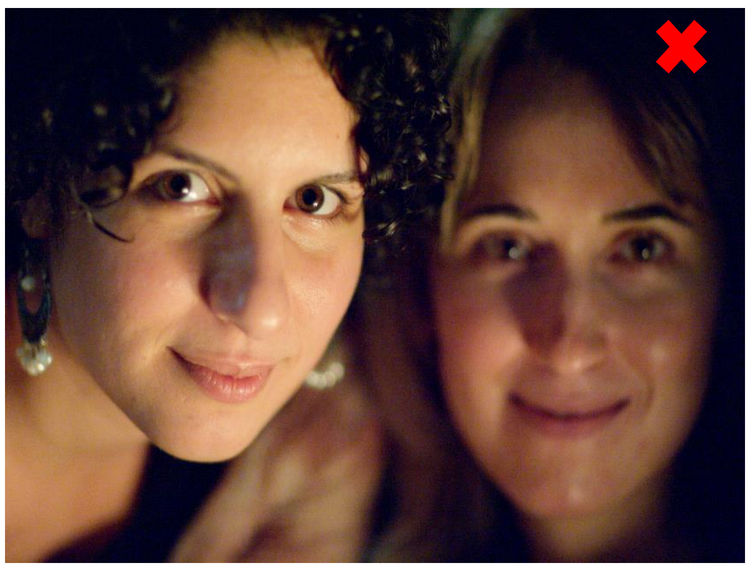
G

H

Real or Fake?

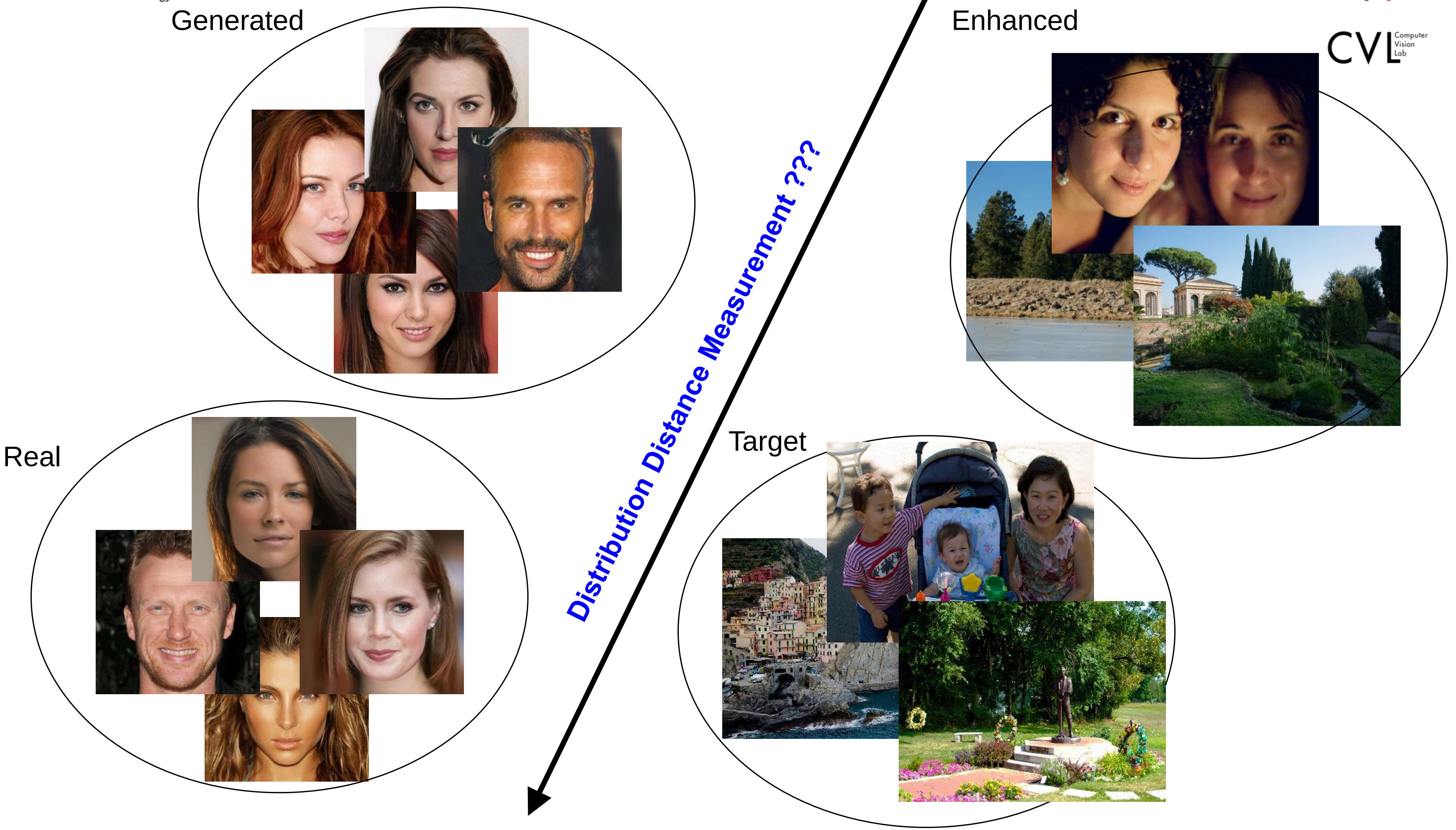
Before or After?



Img Generation

Img Enhancement



Generative Models

(VAE limitations.. Examples)

This bird is white with some black on its head and wings, and has a long orange beak

VAEs are suitable for data compression or generating semantics. GANs are suitable for generating data.

GAN को ज्ञान

GAN "predicts" data given a label, Instead of label given data.

The Minimax Theorem

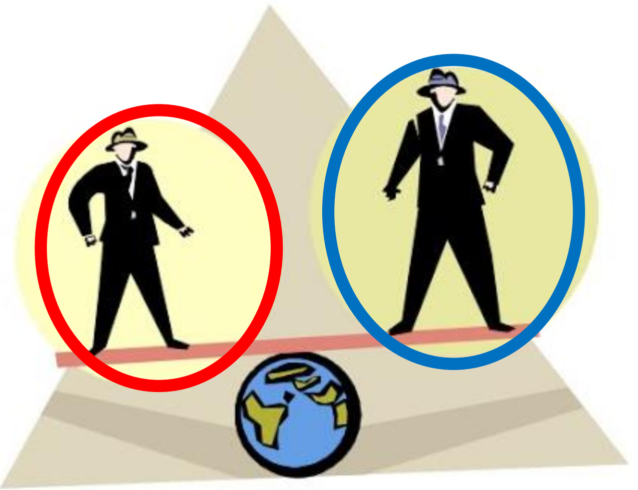
"As far as I can see, there could be no theory of games … without that theorem … I thought there was nothing worth publishing until the Minimax Theorem was proved"

-- John von Neumann, 1928

Zero-sum Game
(An Example)

- A zero-sum game is *strictly a competitive* game.

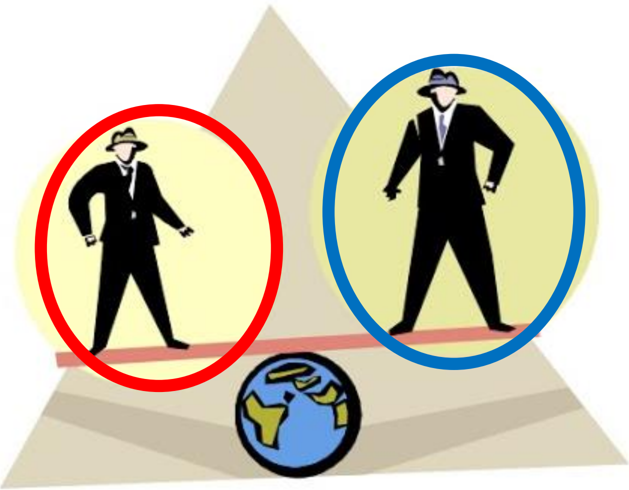
Blue Red	*A*	*B*	*C*
1	−30 30	10 −10	−20 20
2	10 −10	−20 20	20 −20



Zero-sum Game
(An Example)

- A zero-sum game is *strictly a competitive* game.

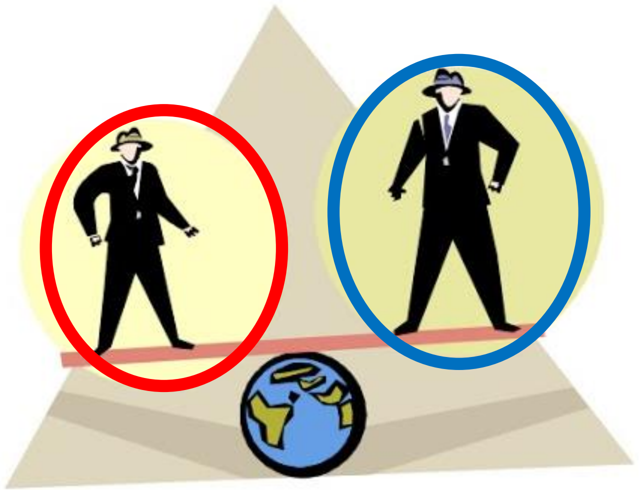
Blue / Red	*A*	*B*	*C*
1	−30 / 30	10 / −10	−20 / 20
2	10 / −10	−20 / 20	20 / −20



Zero-sum Game
(An Example)

- A zero-sum game is *strictly a competitive* game.

Blue Red	*A*	*B*	*C*
1	−30 30	10 −10	−20 20
2	10 −10	−20 20	20 −20

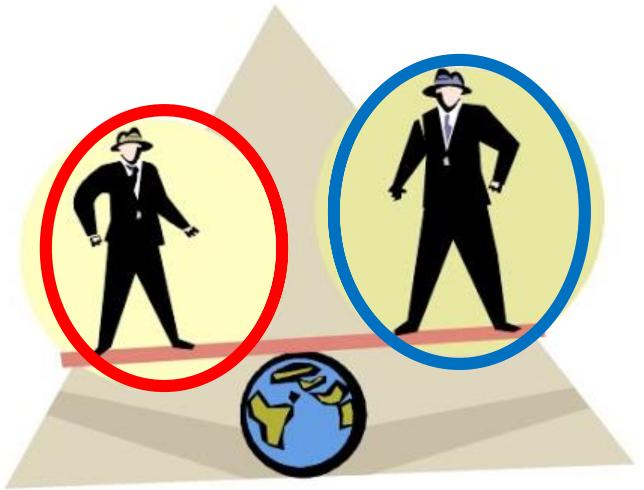


- Two players assign probabilities to their respective actions.
- Each player computes the probabilities to minimize the maximum expected point-loss.
- Once’s strategy is independent that of its opponent.
- This leads to a minmax problem with the optimal strategies for each player.

Zero-sum Game
(An Example)

- A zero-sum game is *strictly a competitive* game.

Blue Red	*A*	*B*	*C*
1	−30 30	10 −10	−20 20
2	10 −10	−20 20	20 −20



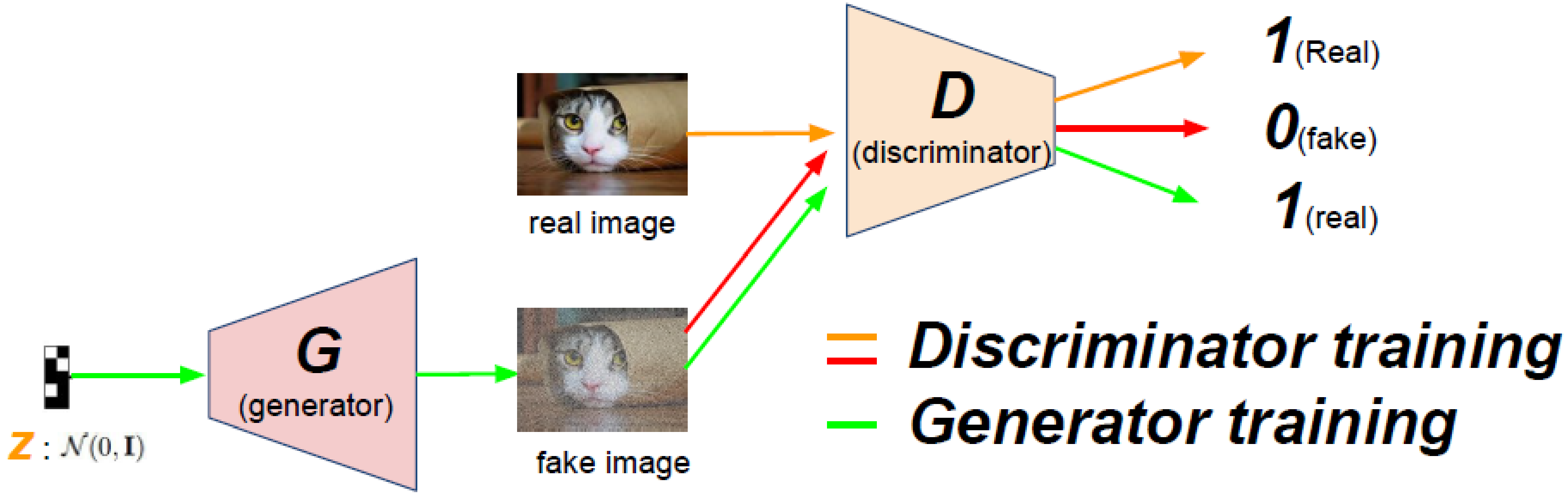
$$\max_{x\in X}\min_{y\in Y} f(x,y)=\min_{y\in Y}\max_{x\in X} f(x,y).$$

$f(\cdot,y):X\to\mathbb{R}$ is concave for fixed y, and
$f(x,\cdot):Y\to\mathbb{R}$ is convex for fixed x.

Can Compute the Nash Equilibrium!!

GAN को ज्ञान

- Two-player game (min-max objective function)
  - $\min_{G} \max_{D}(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_z(z)}[\log(1 - D(G(z)))]$



Images from Namju Kim

Conditional GAN
(As Joint distribution learning)

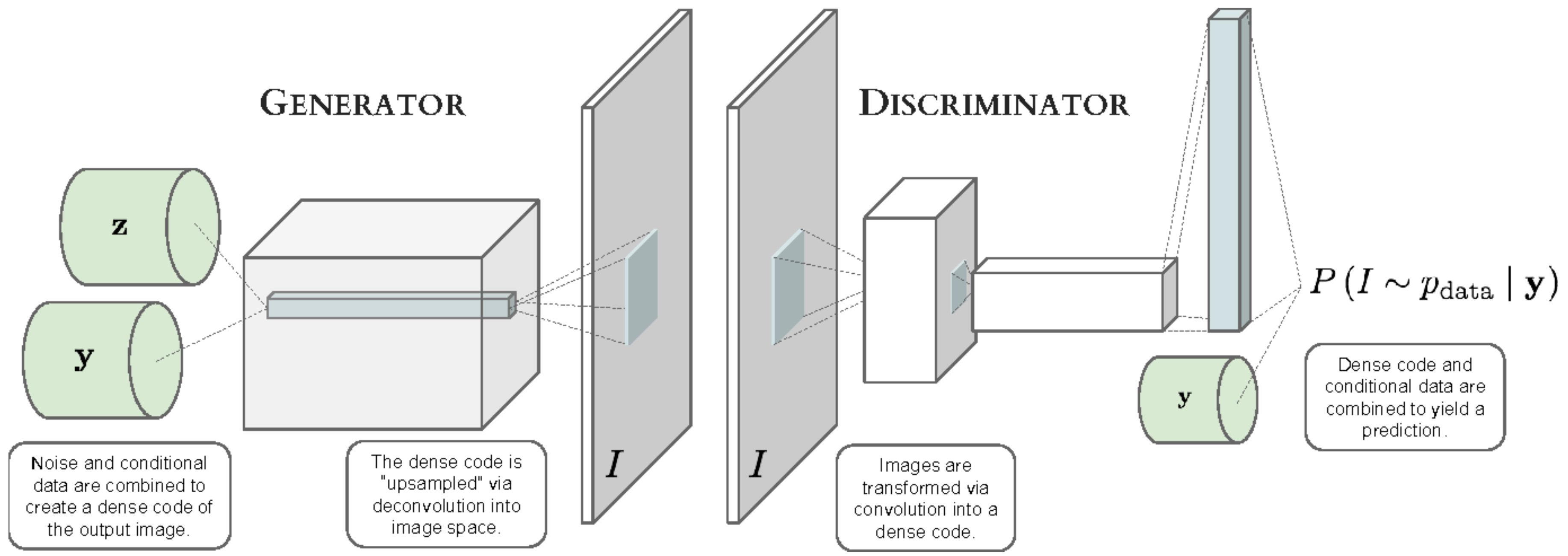
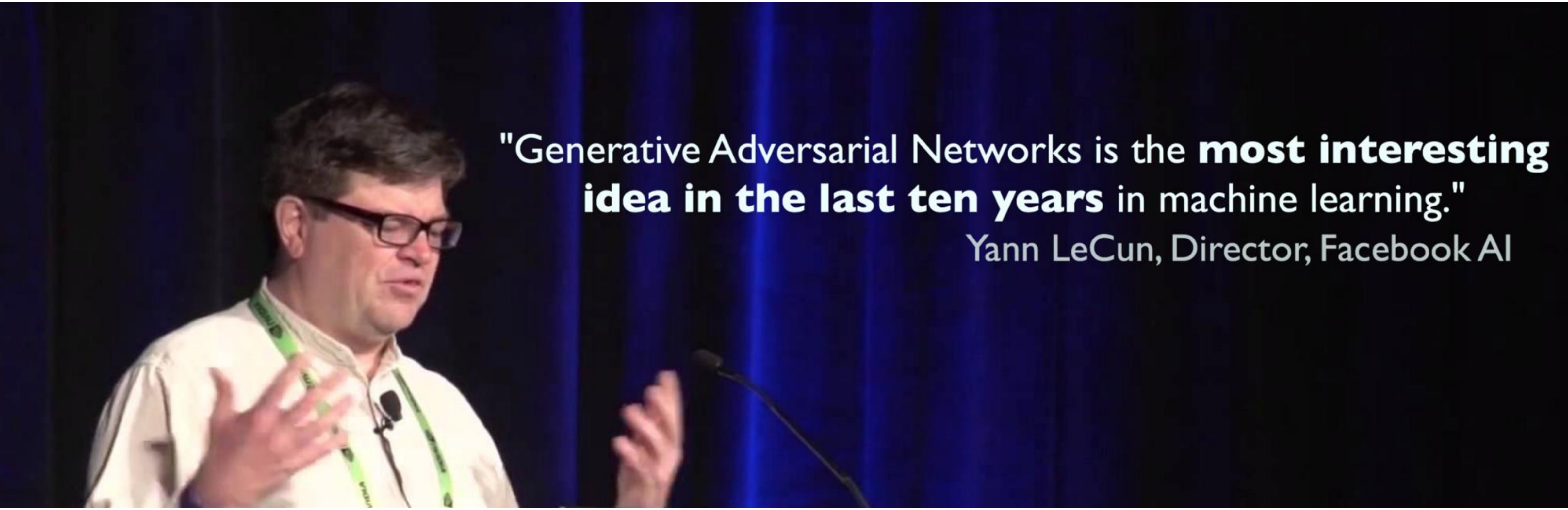


Image from: [Jon Gauthier]



The general idea of learning via competition between players dates back to at least 1959 with Arthur Samuel.

How GANs evolve? Two branches

"Those who do not move, do not notice their chains."
— **Rosa Luxemburg**

Image GAN	**GAN** Goodfellow'14	**DCGAN** Radford'15	**WGAN** Arjovsk'17	**PWGAN** Karras'18	**StyleGAN** Karras'19
Conditional GAN	**Pix2pix** Isola'17	**CycleGAN** Zhu'17	**StarGAN** Choi'18	**SPADE** Park'19	**SinGAN** Shaham'19

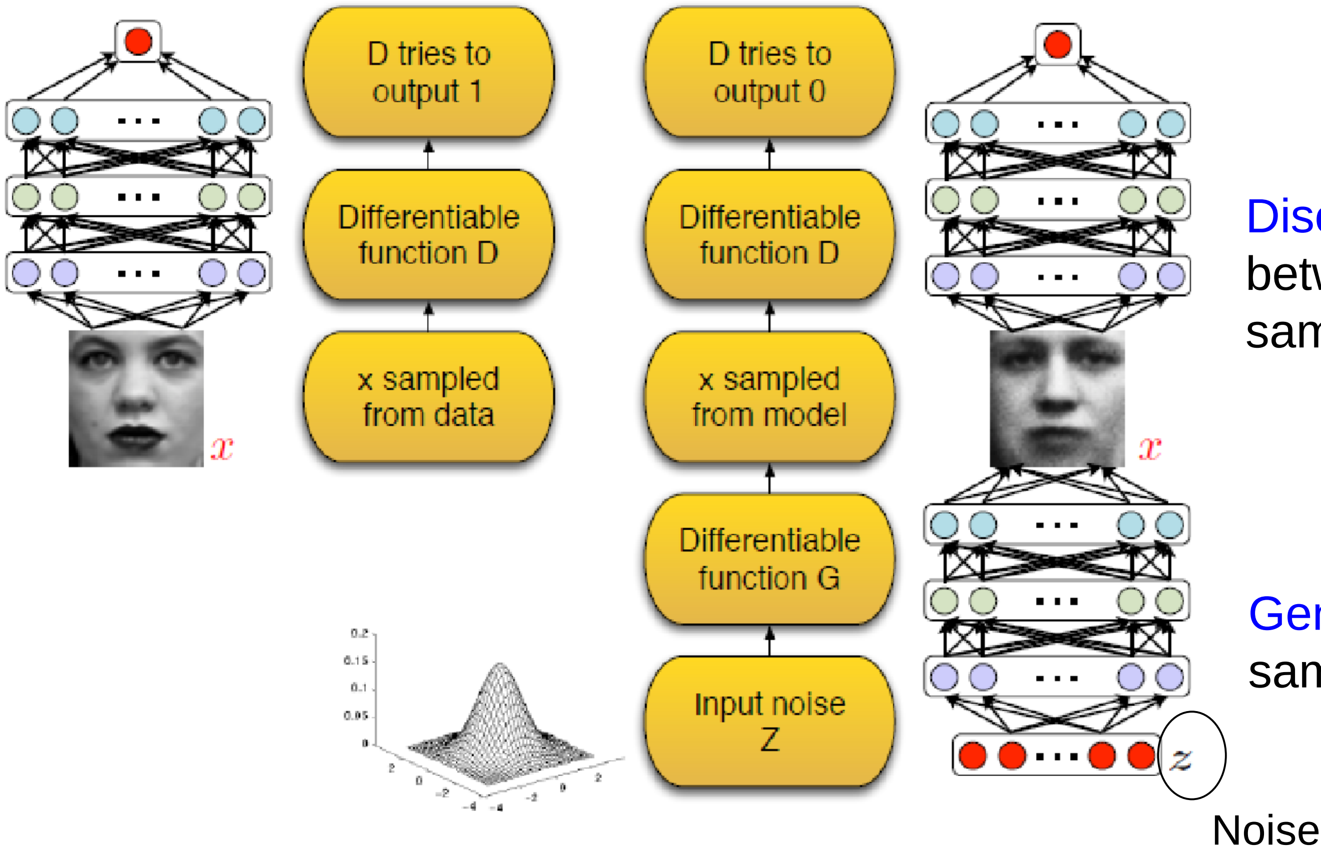
Image GAN

GAN Goodfellow'14 → **DCGAN** Radford'15 → **WGAN** Arjovsk'17 → **PWGAN** Karras'18 → **StyleGAN** Karras'19

From ImageNet

Image GAN

GAN Goodfellow'14 → **DCGAN** Radford'15 → **WGAN** Arjovsk'17 → **PWGAN** Karras'18 → **StyleGAN** Karras'19



Discriminator: distinguish between real and fake samples

Generator: generate samples from noises

Ian Goodfellow, NIPS'2014 workshop

GAN by Example

```
# Generator
```

```
# Discriminator
D = nn.Sequential(nn.Linear(image_size, hidden_size),
```

```
# Binary cross entropy loss and optimizer
criterion = nn.BCELoss()
d_optimizer = torch.optim.Adam(D.parameters(), lr=0.0002)
g_optimizer = torch.optim.Adam(G.parameters(), lr=0.0002)

for epoch in range(num_epochs):
for i, (images, _) in enumerate(data_loader):
images = images.reshape(batch_size, -1).to(device)

# Create the labels which are later used as input for the BCE loss
real_labels = torch.ones(batch_size, 1).to(device)
fake_labels = torch.zeros(batch_size, 1).to(device)
```

```
# Train the discriminator #
# Compute BCE_Loss using real images where
# BCE_Loss(x, y): - y * log(D(x)) - (1-y) * log(1 - D(x))
# Second term of the loss is always zero since real_labels == 1

outputs = D(images)
d_loss_real = criterion(outputs, real_labels)
real_score = outputs

# Compute BCELoss using fake images
```

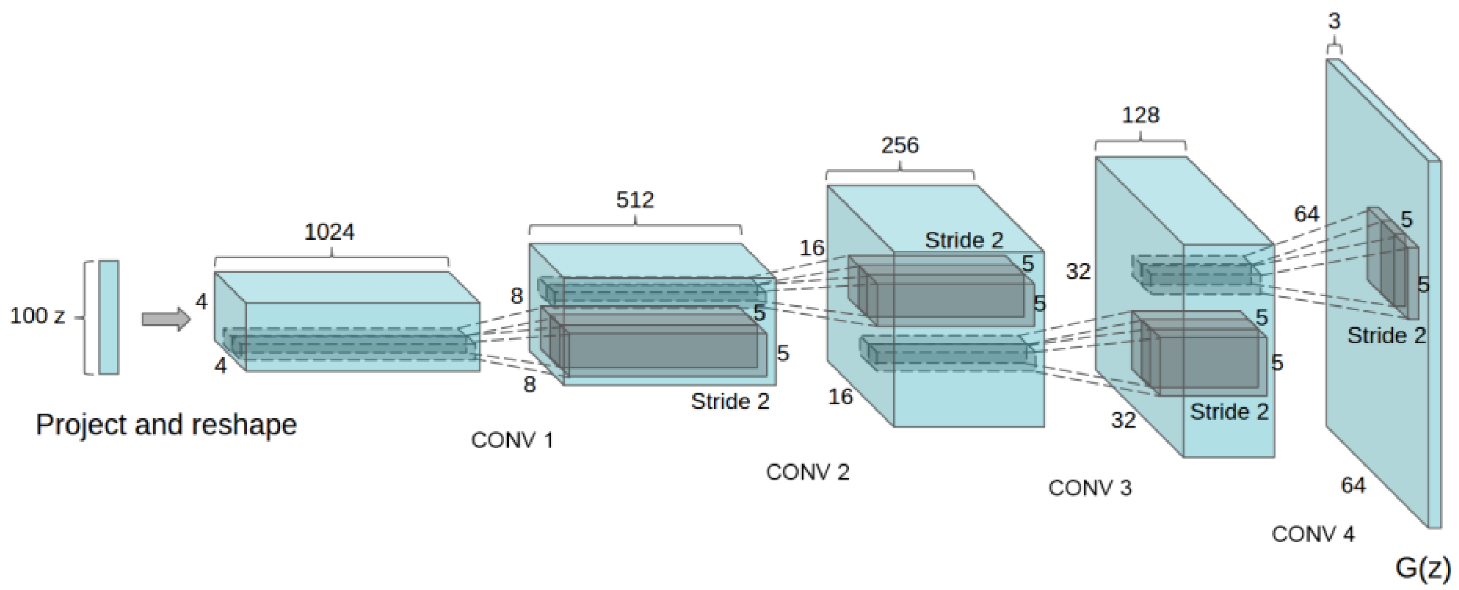
```
# Train the generator #
# Compute loss with fake images
z = torch.randn(batch_size, latent_size).to(device)

fake_images = G(z)
outputs = D(fake_images)

# We train G to maximize log(D(G(z)) instead of minimizing log(1-D(G(z)))
g_loss = criterion(outputs, real_labels)
# Backprop and optimize
reset_grad()
g_loss.backward()
```

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

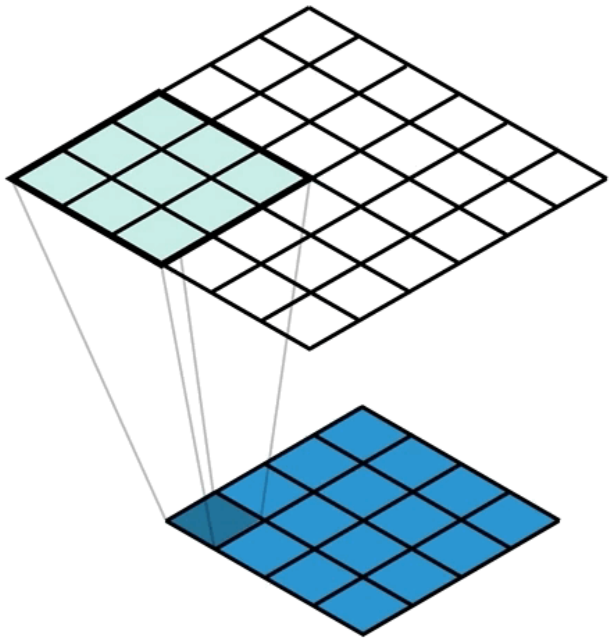
Image GAN | GAN Goodfellow'14 | DCGAN Radford'15 | WGAN Arjovsk'17 | PWGAN Karras'18 | StyleGAN Karras'19

- GAN with explicit usage of convolutional and convolutional-transpose layers!



What Actually is Transpose Convolution?

-- Distributing Values…….



Input

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Kernel

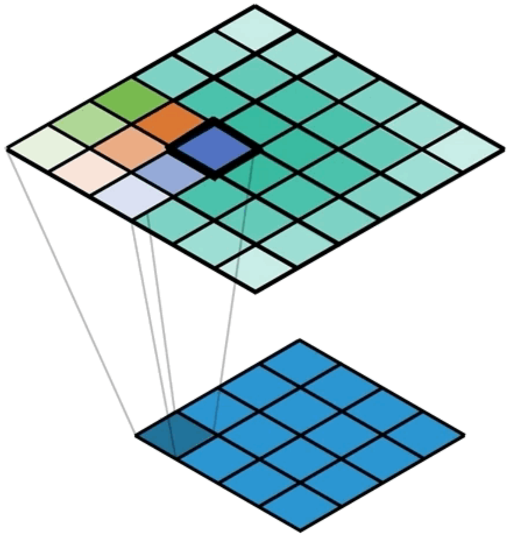
1	1	1
1	1	1
1	1	1

Output

1	2	3	3	2	1
2	4	6	6	4	2
3	6	9	9	6	3
3	6	9	9	6	3
2	4	6	6	4	2
1	2	3	3	2	1

What Actually is Transpose Convolution?

-- Collecting Values.......



Input

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	3	2	1	0	0
0	0	1	3	3	1	0	0
0	0	2	1	1	3	0	0
0	0	3	2	3	3	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Kernel

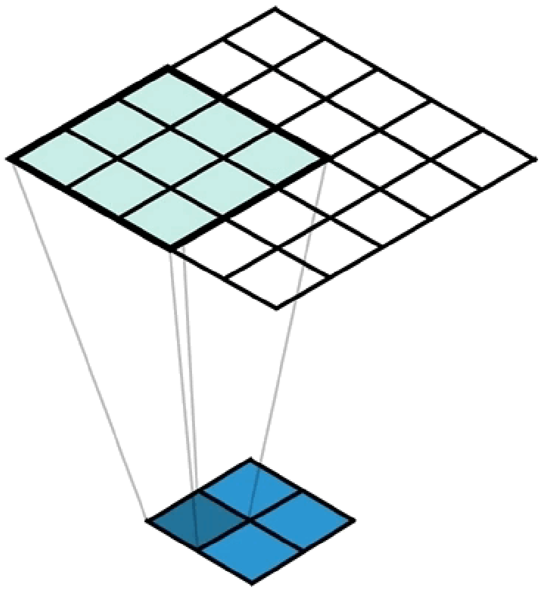
1	2	3
0	1	0
2	1	2

Output

1	5	11	14	8	3
1	6	15	18	12	3
4	13	21	21	15	11
5	17	28	27	25	11
4	7	9	12	8	6
6	7	14	13	9	6

What Actually is Transpose Convolution?

-- Playing with strides…….



Input

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	3	0	3	0	0
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Kernel

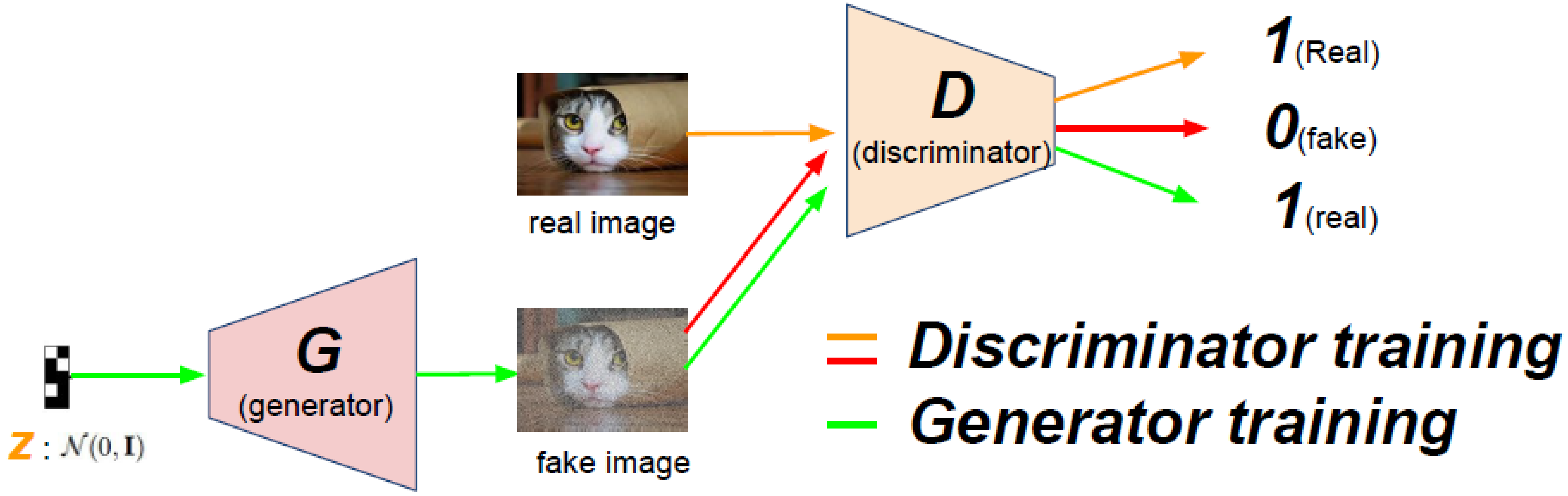
1	2	3
0	1	0
2	1	2

Output

3	6	12	6	9
0	3	0	3	0
7	5	16	5	9
0	1	0	1	0
2	1	4	1	2

GAN को ज्ञान

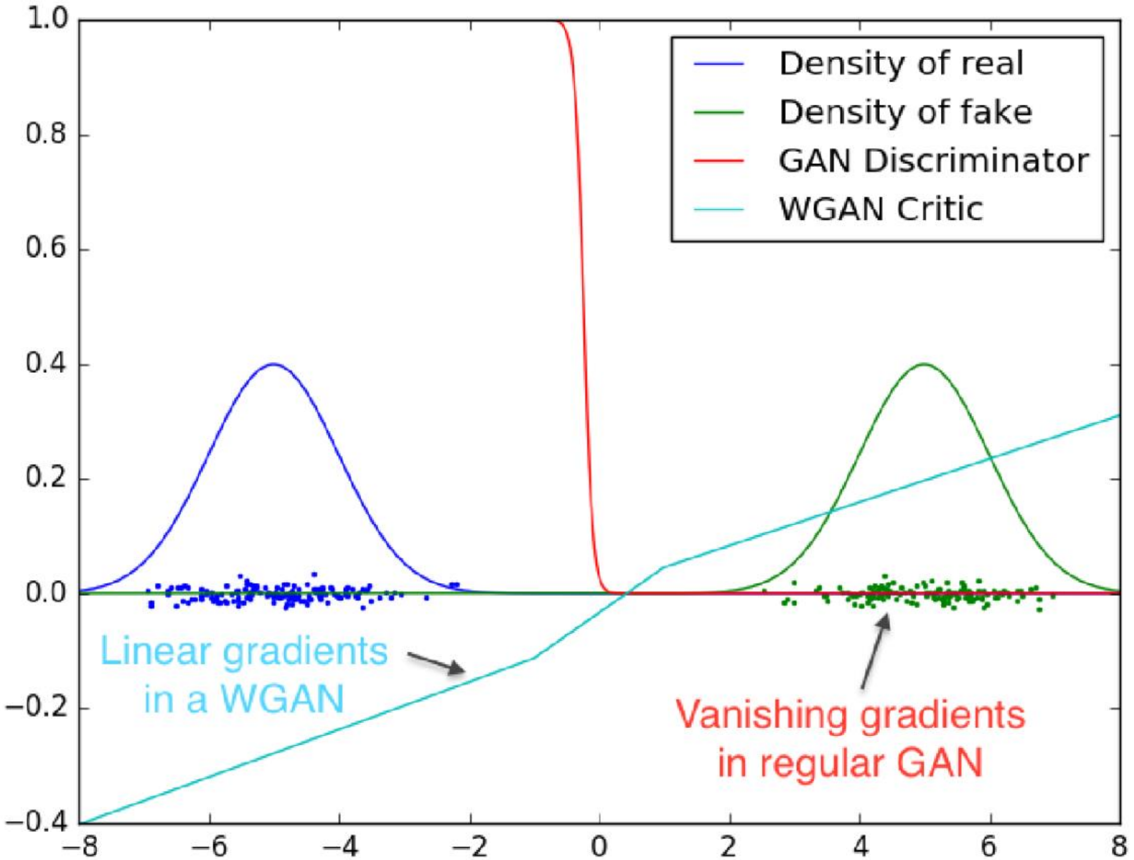
- Two-player game (min-max objective function)
  - $\min_{G} \max_{D}(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x \sim p_{z}(z)}\left[\log(1 - D\left(G(z)\right))\right]$



Images from Namju Kim

The Mode Collapse Problem

A DCGAN model is trained with an MLP network with 4 layers, 512 units and ReLU activation function, configured to lack a strong inductive bias for image generation. The results shows a significant degree of mode collapse. (Image source: [Arjovsky, Chintala, & Bottou, 2017.](#))



Arjovsky et al., ICML'17

Early GANs Goodfellow et al., NIPS'14; Radford et al., arXiv'15

(JS divgergence): locally saturated

$$\min_G \max_D (D, G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{x\sim p_z(z)}[\log(1 - D(G(z)))]$$

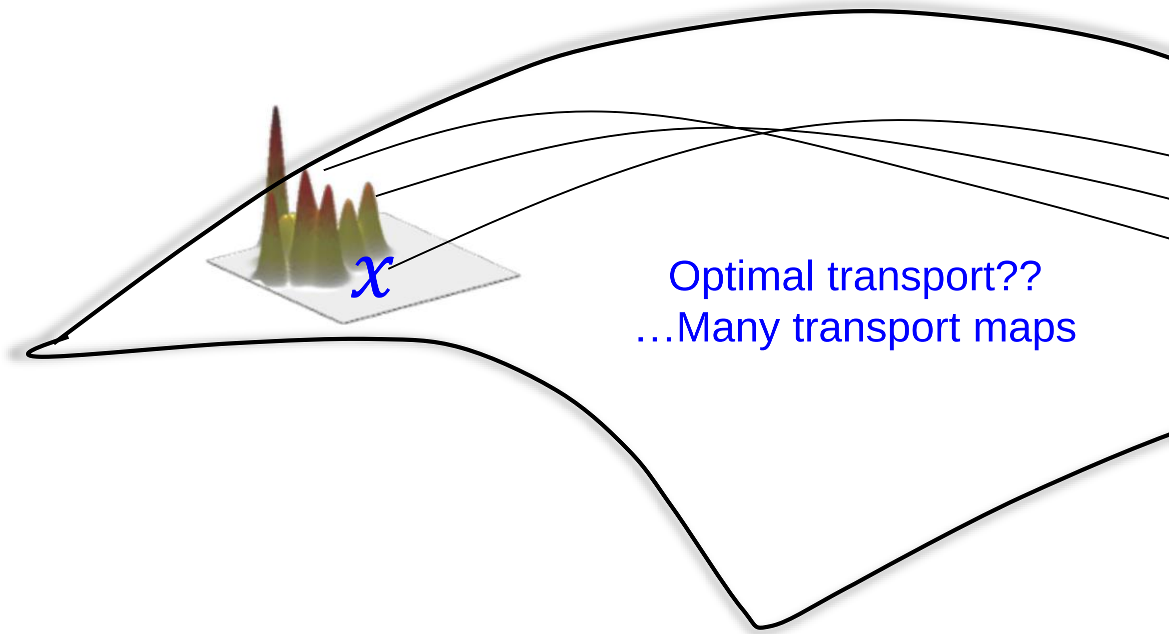
Wasserstein GANs *Arjovsky et al., ICML'17; Gulrajani et al., NIPS'17; Wei et al., ICLR'18*

(Wasserstein distance): continuous and differentiable

$$\min_G \max_{\|C\|_L \le 1} (C, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[C(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_Z(\boldsymbol{z})}[C(G(\boldsymbol{z}))]$$

Wasserstein Distance

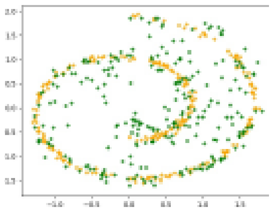
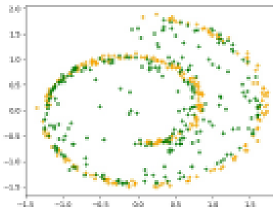
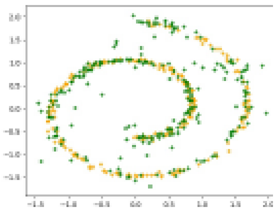
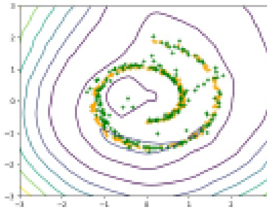
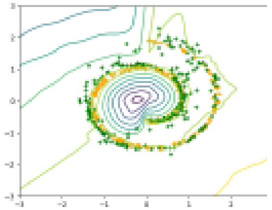
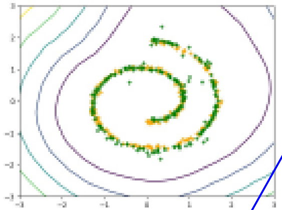
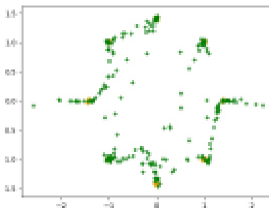
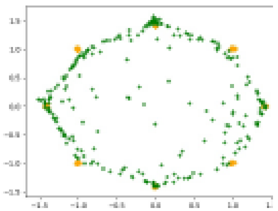
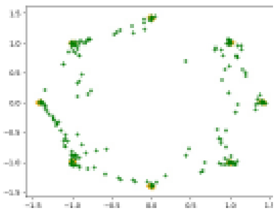
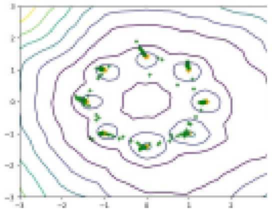
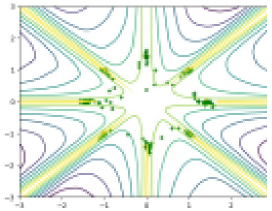
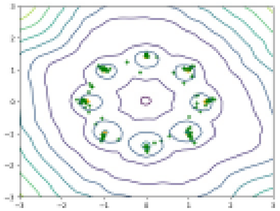
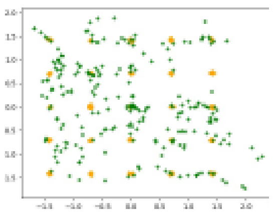
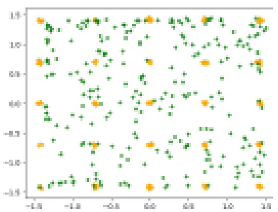
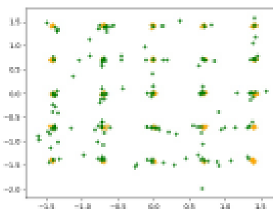
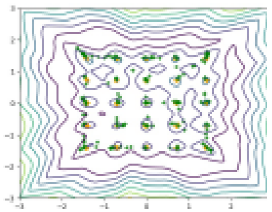
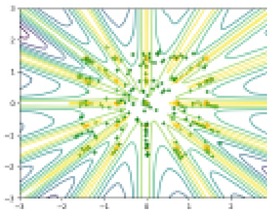
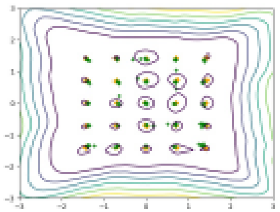
$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$



Optimal transport??
…Many transport maps

```
# train with real
real_cpu, _ = data
netD.zero_grad()
batch_size = real_cpu.size(0)

if opt.cuda:
    real_cpu = real_cpu.cuda()
input.resize_as_(real_cpu).copy_(real_cpu)
inputv = Variable(input)

errD_real = netD(inputv)
errD_real.backward(one)

# train with fake
noise.resize_(opt.batchSize, nz, 1, 1).normal_(0, 1)
noisev = Variable(noise, volatile = True) # totally freeze netG
fake = Variable(netG(noisev).data)
inputv = fake
errD_fake = netD(inputv)
errD_fake.backward(mone)
errD = errD_real - errD_fake
optimizerD.step()
```

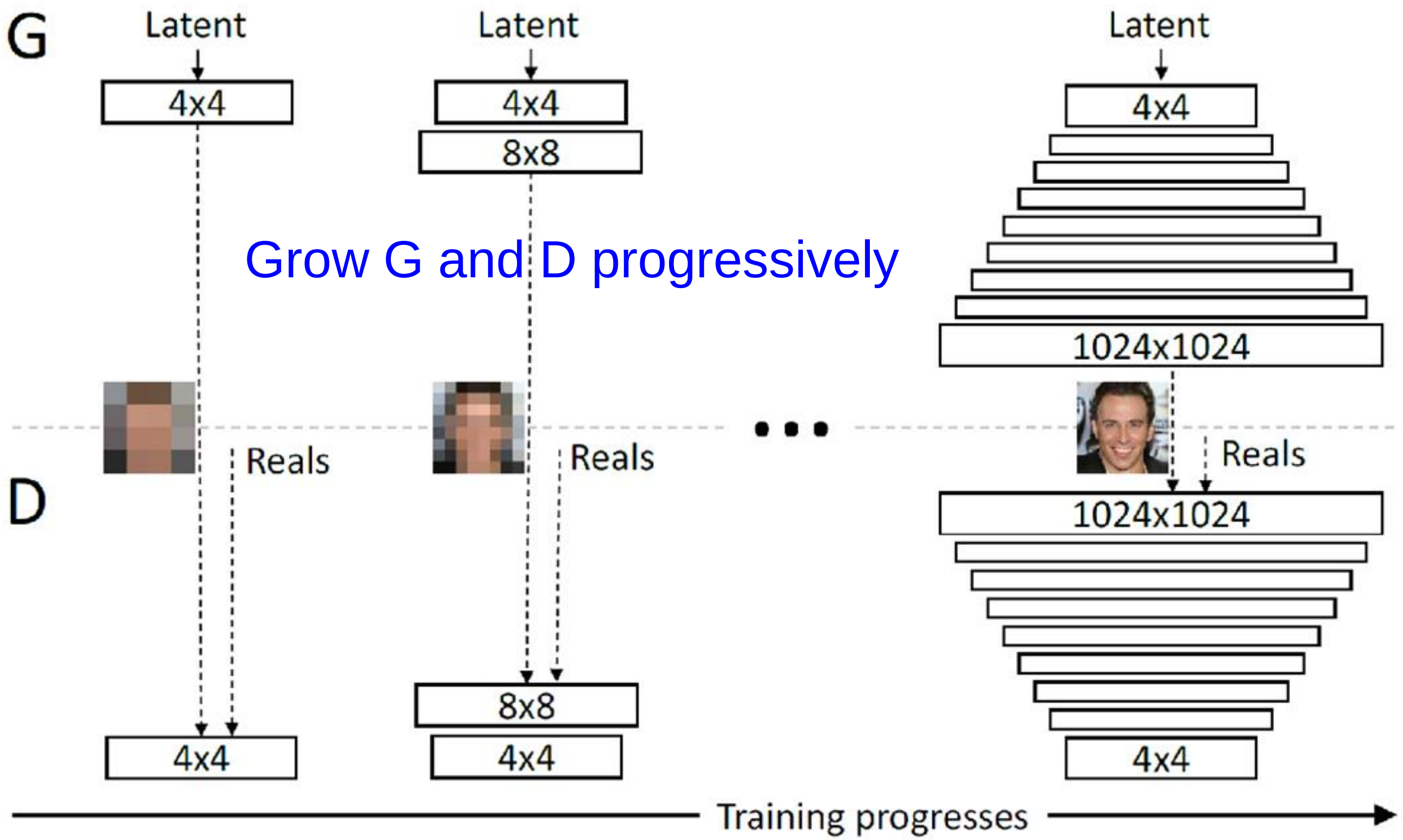
$$W_p(P, Q) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim p_r}[f(x)] - \mathbb{E}_{y\sim p_\theta}[f(y)]$$

WAE	AE +100 IDT	**SWAE**	CT-GAN	SWG	**SWGAN**
					
0.04 (0.06)	0.05 (0.06)	**0.04 (0.04)**	0.01	0.03	**0.01**
					
0.07 (0.04)	0.06 (0.06)	**0.03 (0.02)**	0.03	0.05	**0.02**
					
0.05 (0.03)	0.05 (0.06)	**0.04 (0.02)**	0.02	0.05	**0.01**

Frechet Inception Distance (FID)

$$\mathrm{FID} = ||\mu_r - \mu_g||^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

Progressive growing of GANs:



Grow G and D progressively

Wasserstein GAN loss

Karras et al., ICLR'18

Generated HQ samples

Image GAN

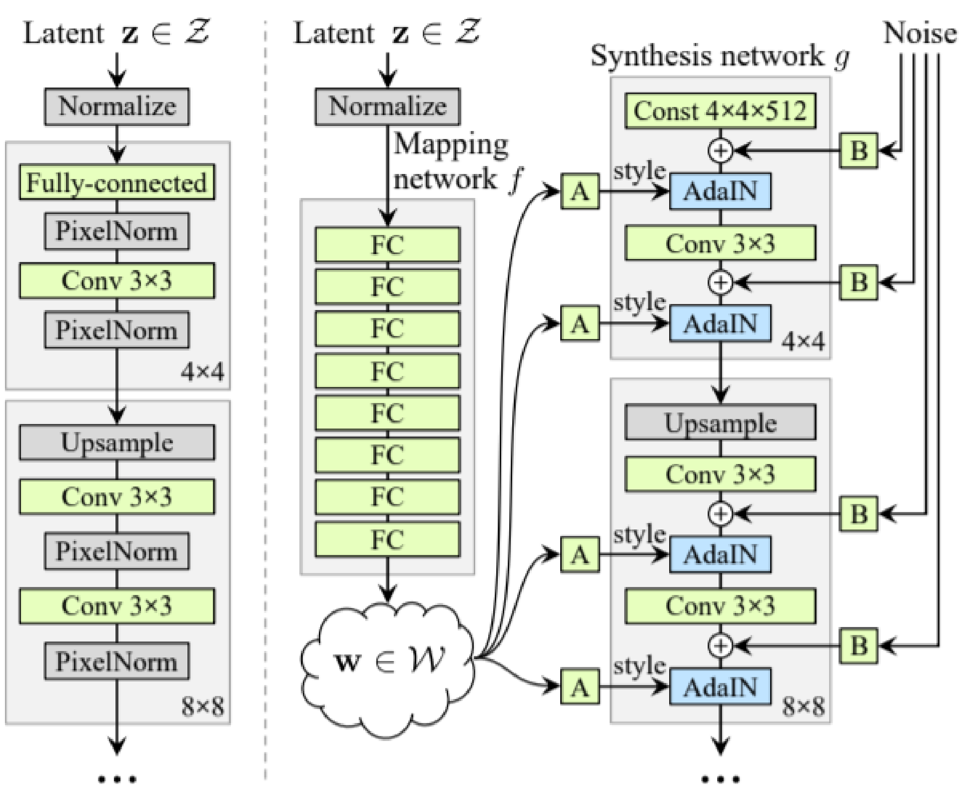
GAN Goodfellow'14 → **DCGAN** Radford'15 → **WGAN** Arjovsk'17 → **PWGAN** Karras'18 → **StyleGAN** Karras'19

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

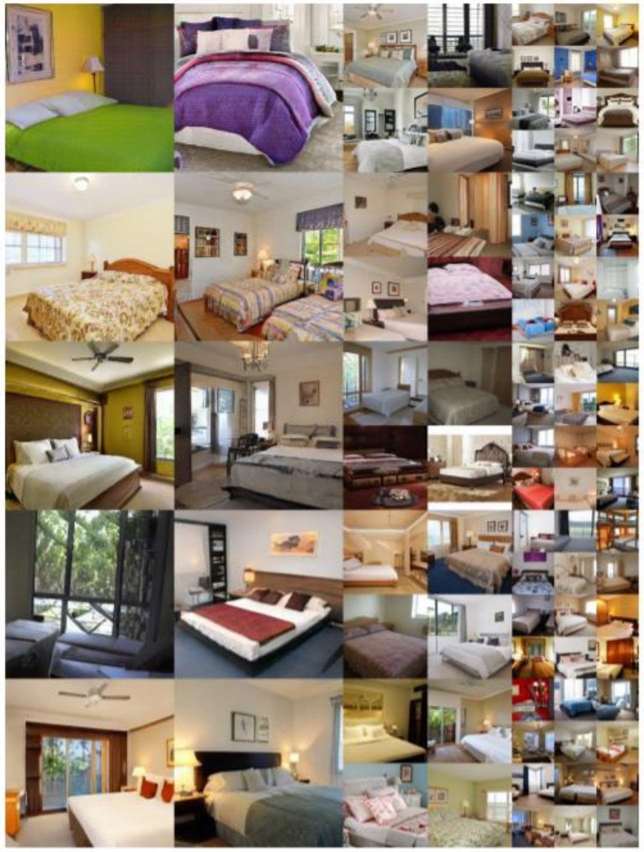
Image GAN

GAN Goodfellow'14 → DCGAN Radford'15 → WGAN Arjovsk'17 → PWGAN Karras'18 → **StyleGAN Karras'19**



(a) Traditional

(b) Style-based generator



This bird is white with some black on its head and wings, and has a long orange beak

This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face

This flower has overlapping pink pointed petals surrounding a ring of short yellow filaments

Zhang et al., ICCV'17

Conditional GAN

Pix2pix Isola'17

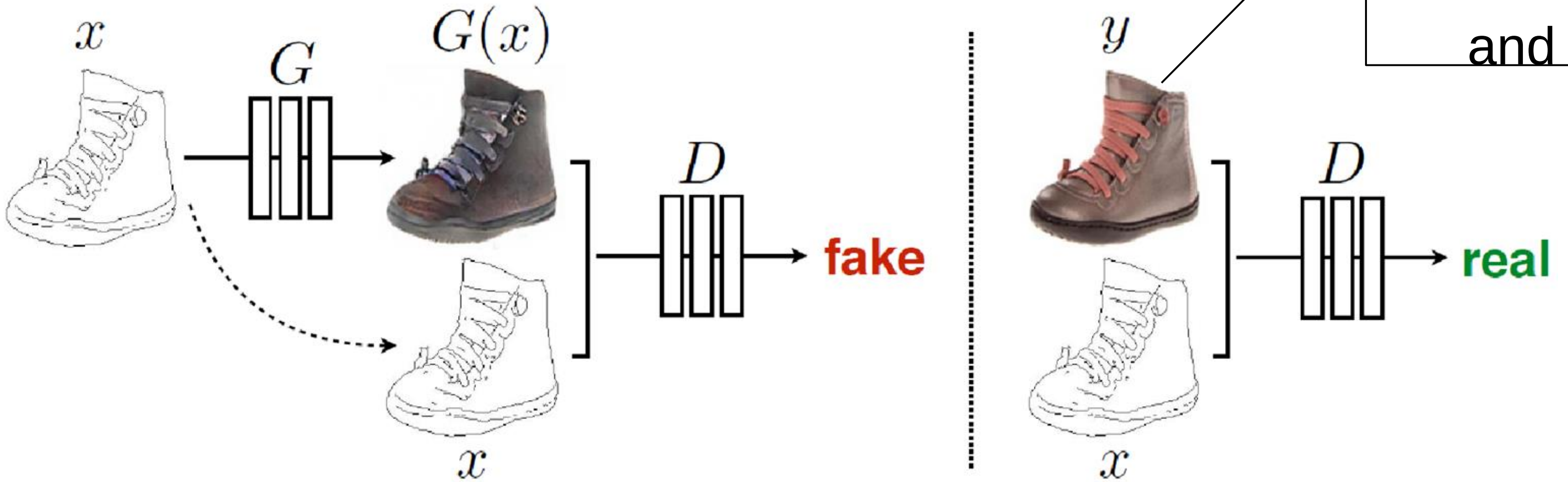
CycleGAN Zhu'17

StarGAN Choi'18

SPADE Park'19

SinGAN Shaham'19

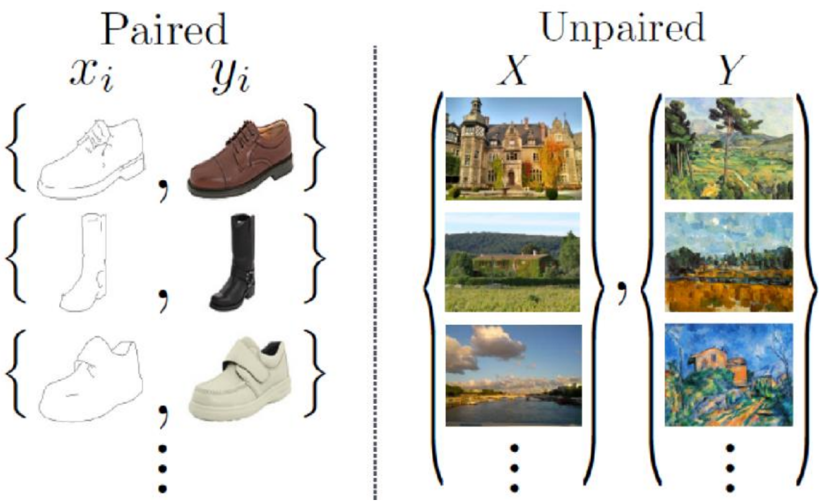
Tuples of edge and photo

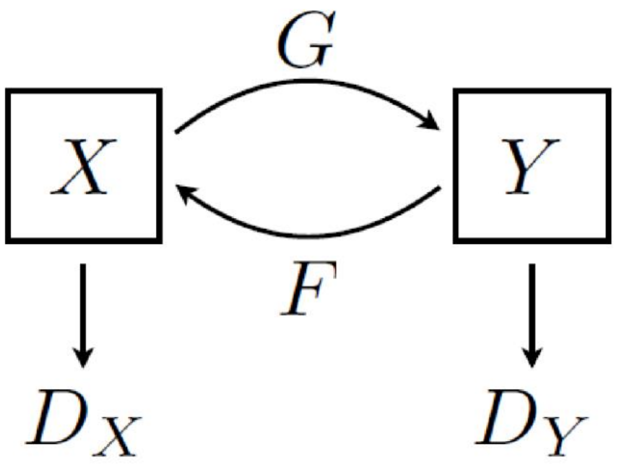


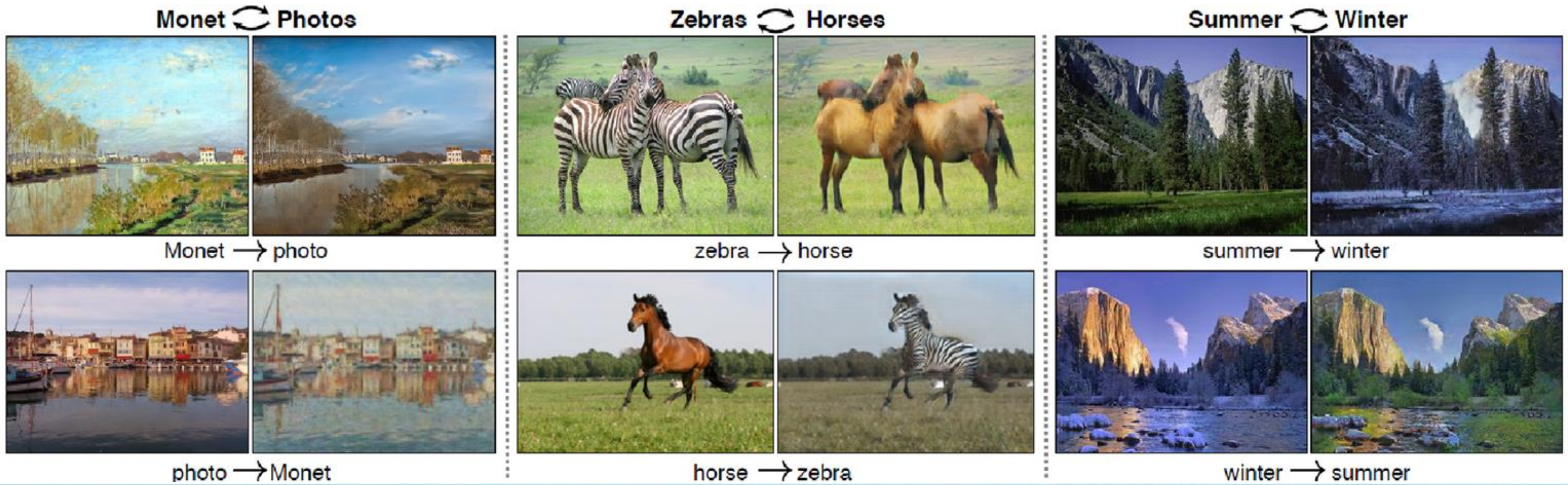
Map to aerial photo

Aerial photo to map

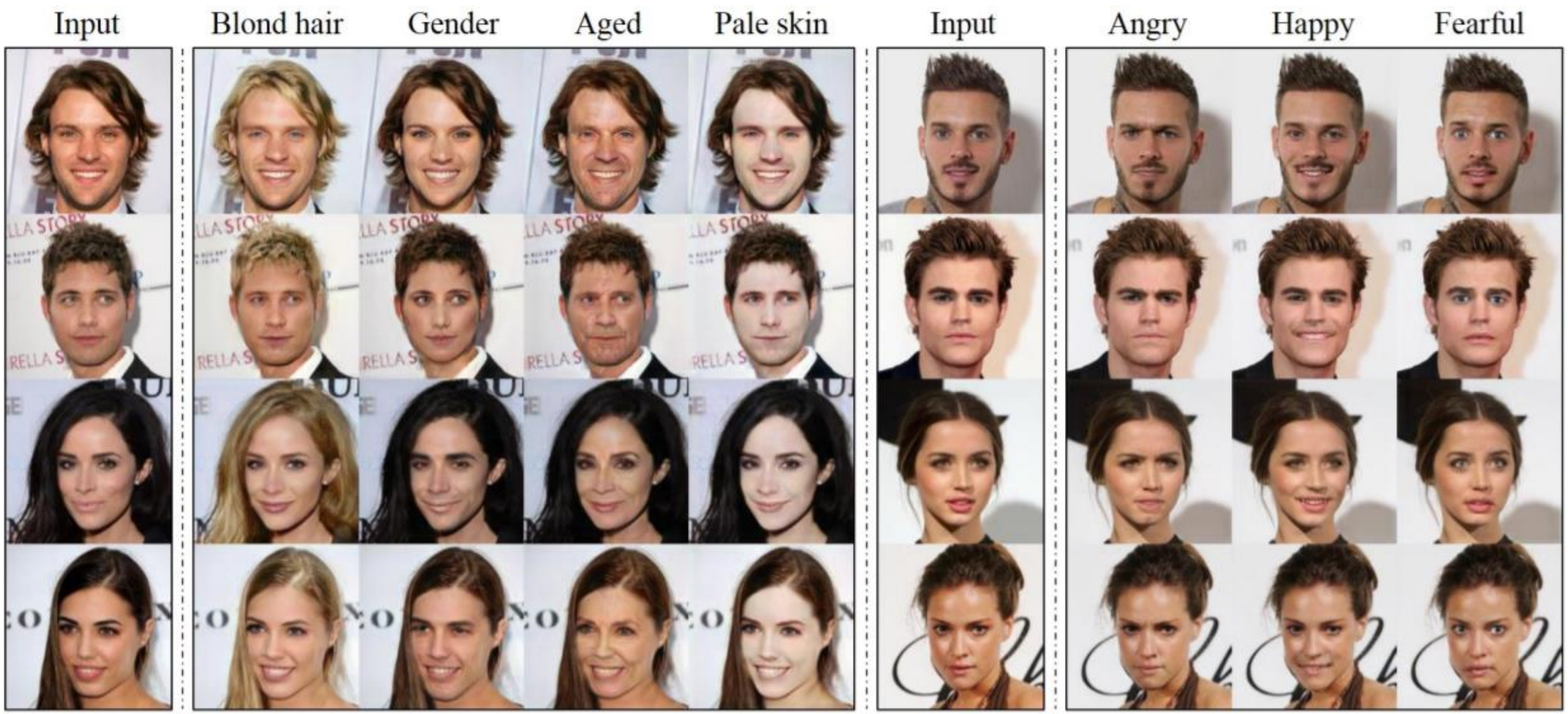
Conditional GAN | Pix2pix Isola'17 | CycleGAN Zhu'17 | StarGAN Choi'18 | SPADE Park'19 | SinGAN Shaham'19

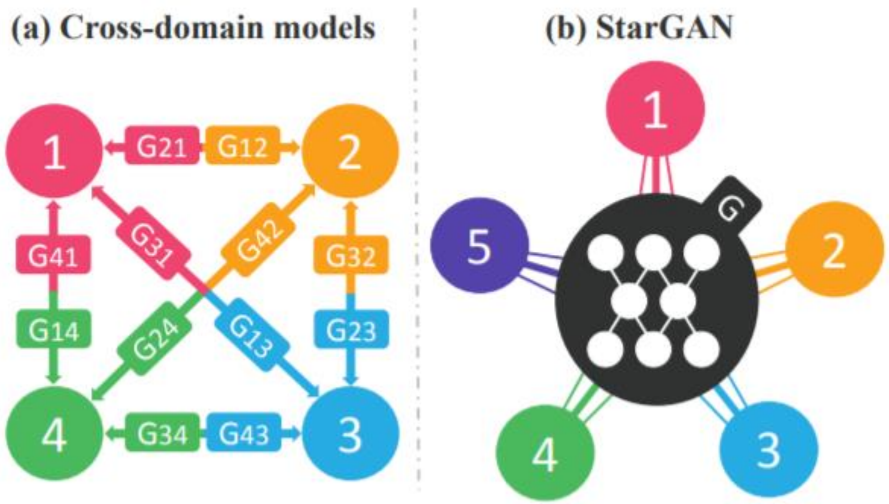


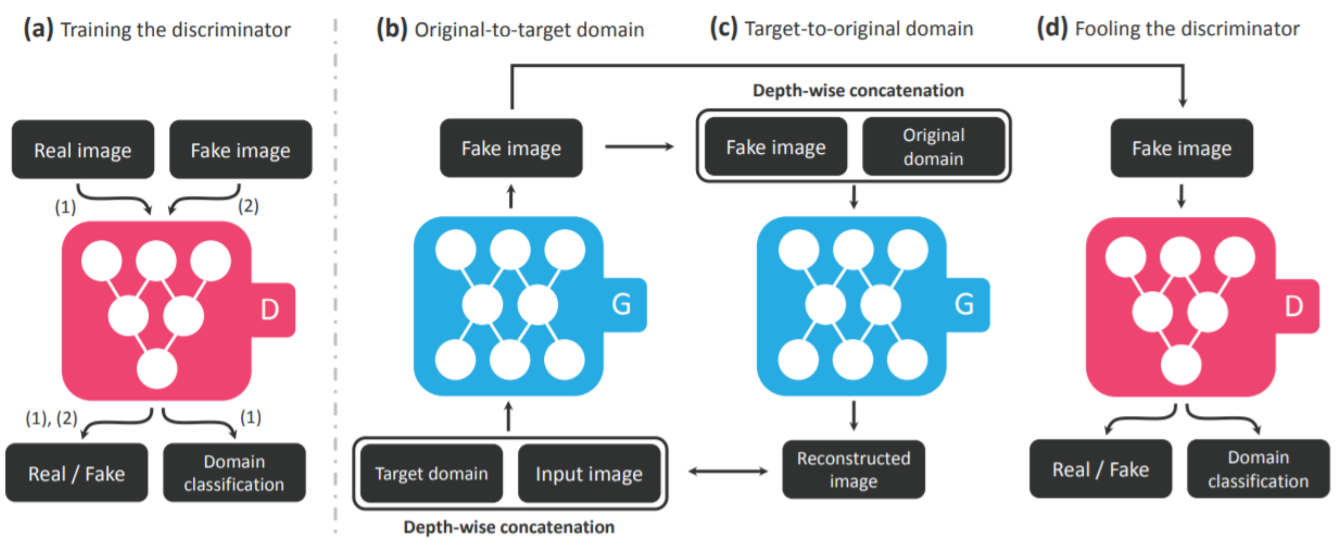


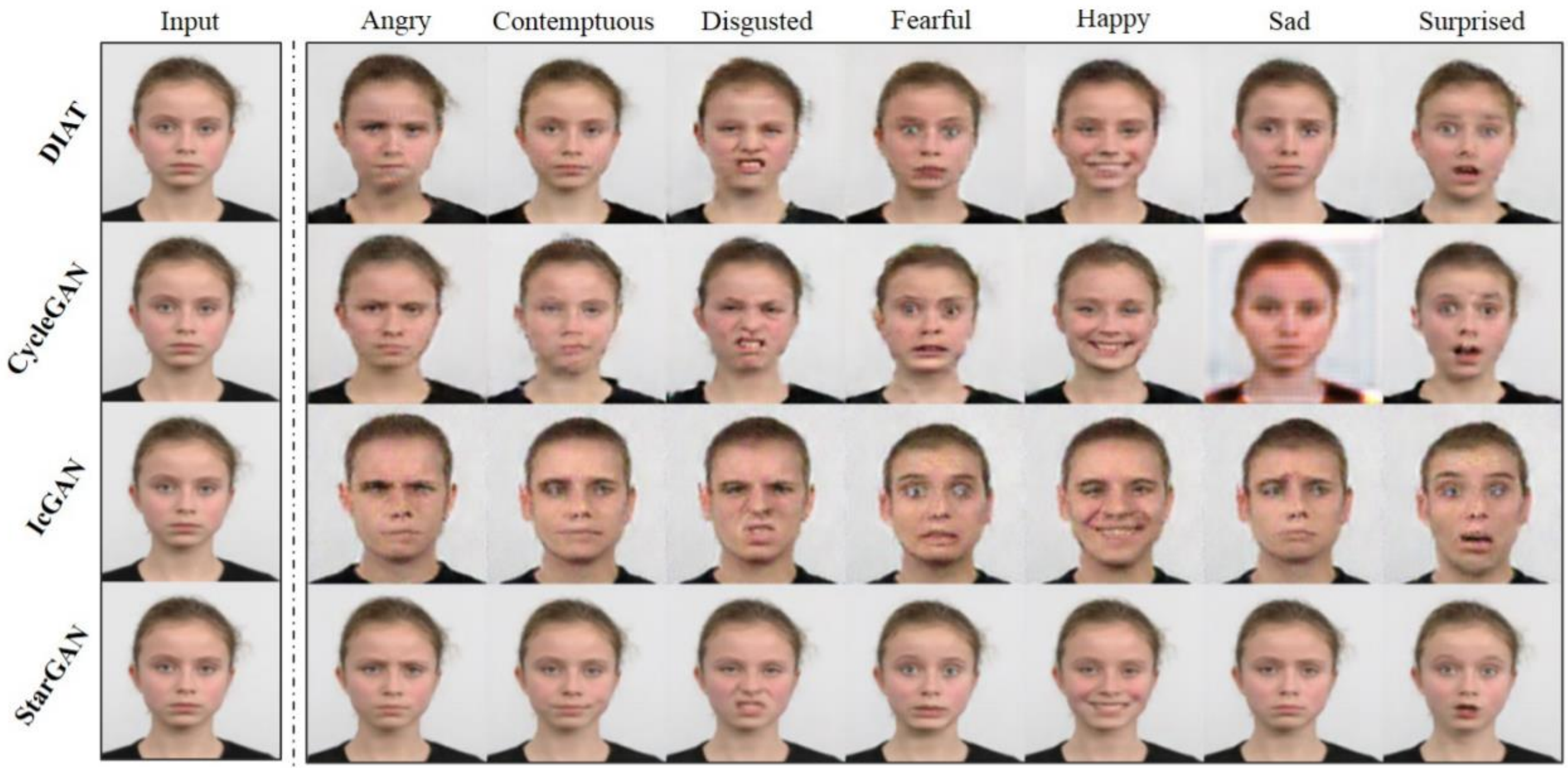


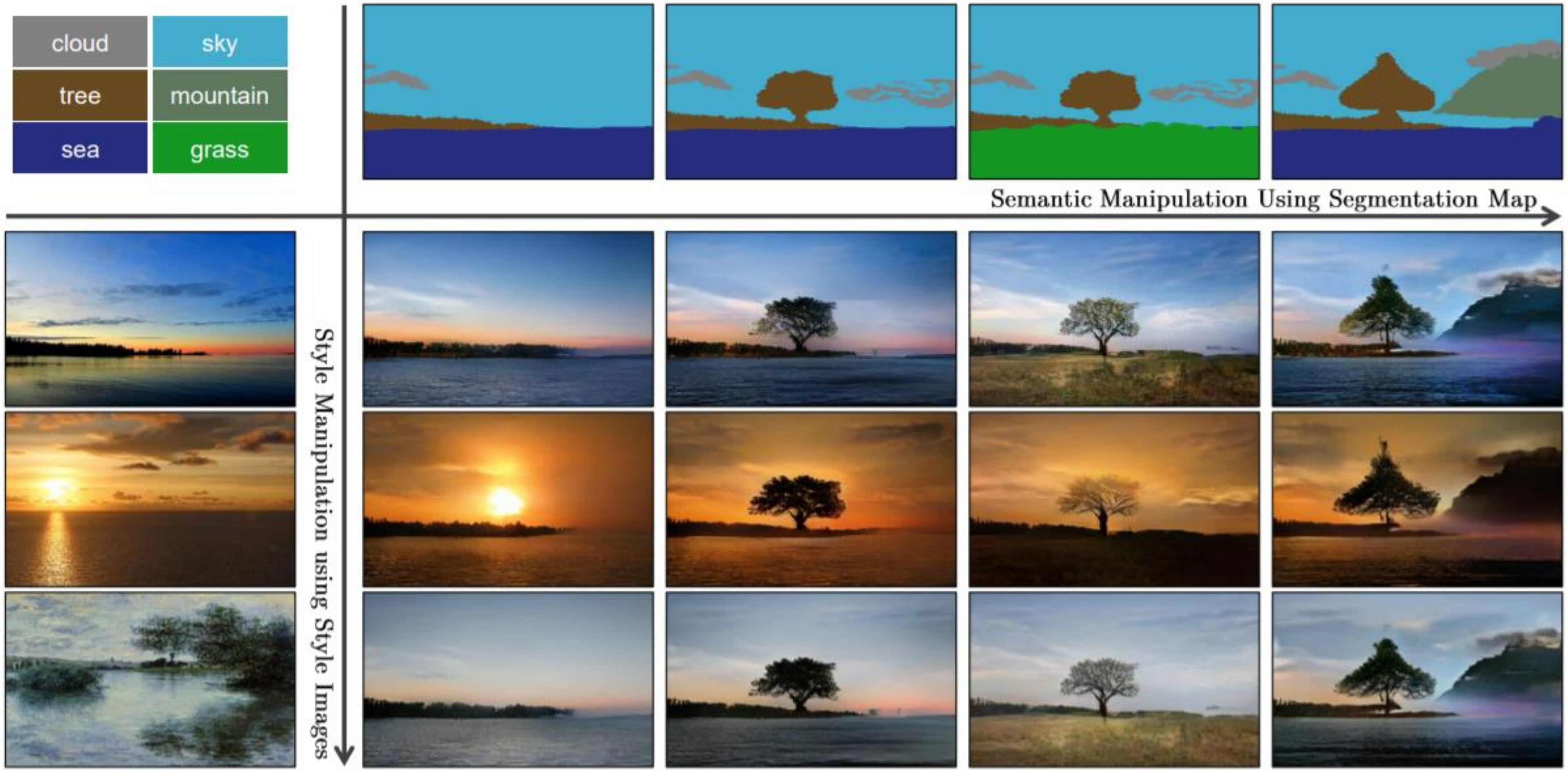
Conditional GAN | **Pix2pix** Isola'17 | **CycleGAN** Zhu'17 | **StarGAN** Choi'18 | **SPADE** Park'19 | **SinGAN** Shaham'19

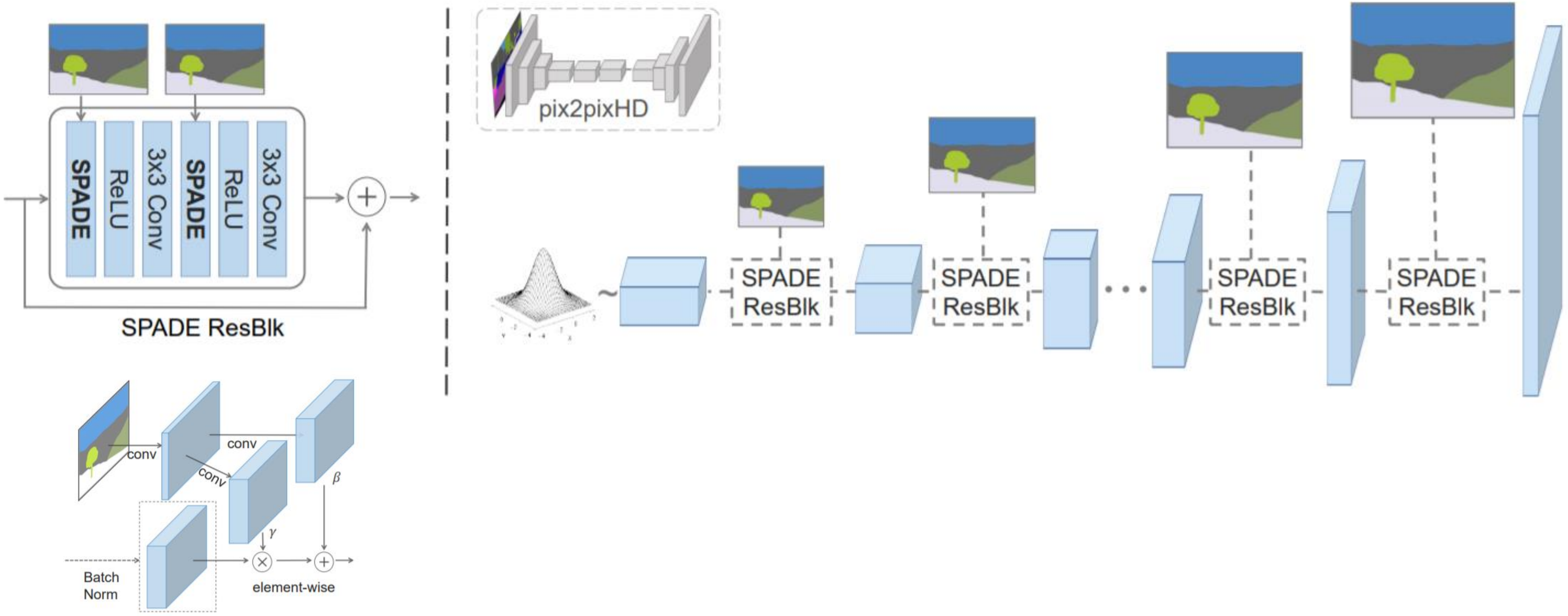



(a) Cross-domain models
(b) StarGAN
G21
G12
G41
G31
G42
G32
G14
G24
G13
G23
G34
G43
G
1
2
3
4
5


(a) Training the discriminator
(b) Original-to-target domain
(c) Target-to-original domain
(d) Fooling the discriminator
Real image
Fake image
(1)
(2)
D
(1), (2)
(1)
Real / Fake
Domain classification
Depth-wise concatenation
Fake image
G
Target domain
Input image
Depth-wise concatenation
Fake image
Original domain
G
Reconstructed image
Fake image
D
Real / Fake
Domain classification

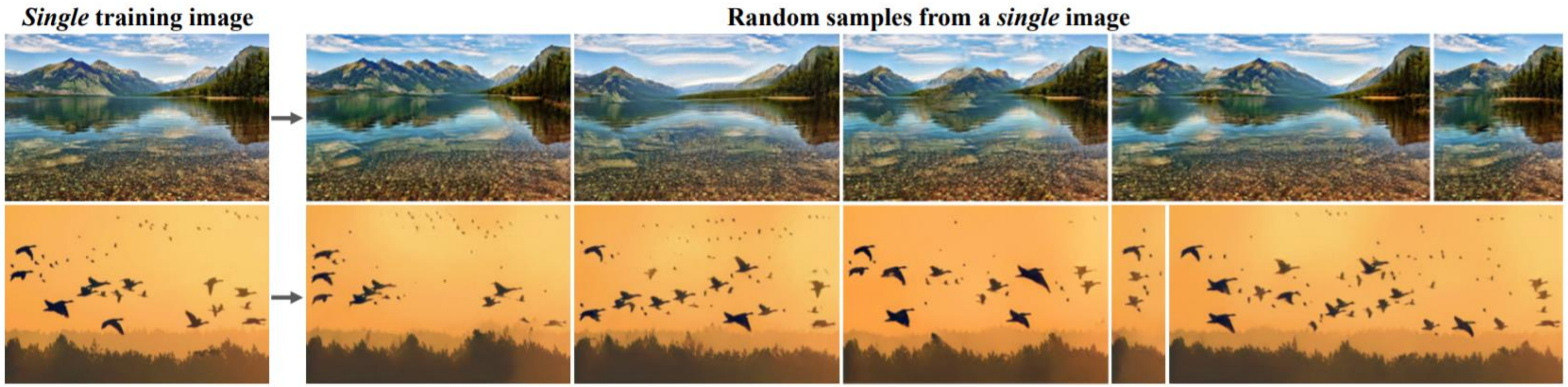

Input
Angry
Contemptuous
Disgusted
Fearful
Happy
Sad
Surprised
DIAT
CycleGAN
IcGAN
StarGAN

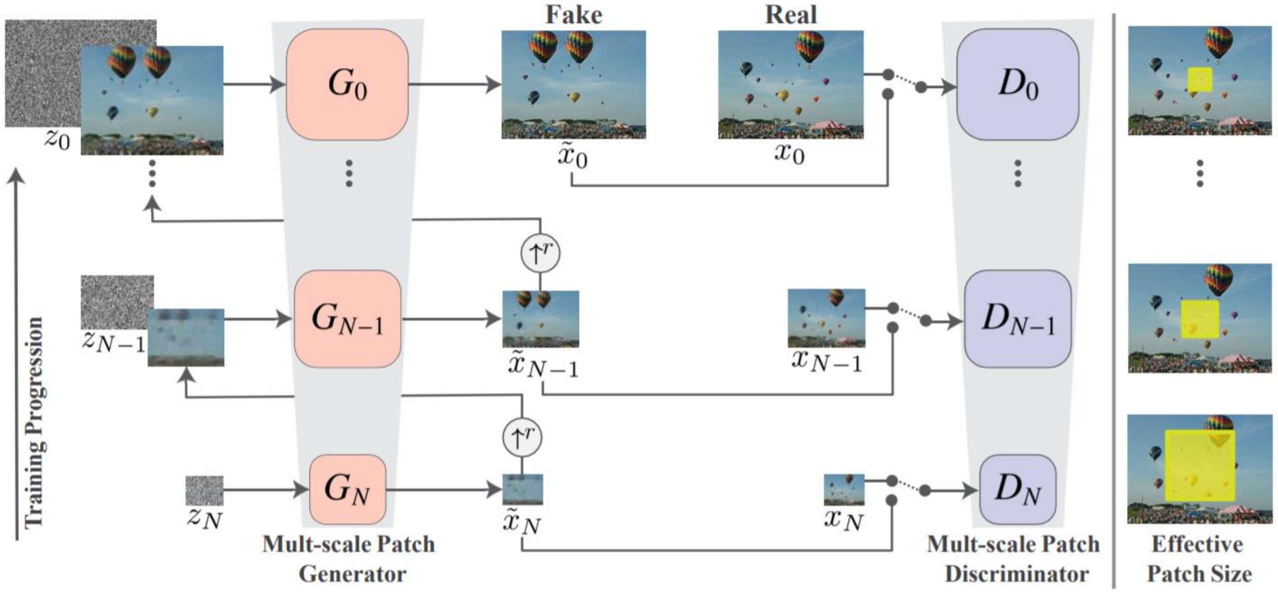

cloud
sky
tree
mountain
sea
grass
Semantic Manipulation Using Segmentation Map
Style Manipulation using Style Images

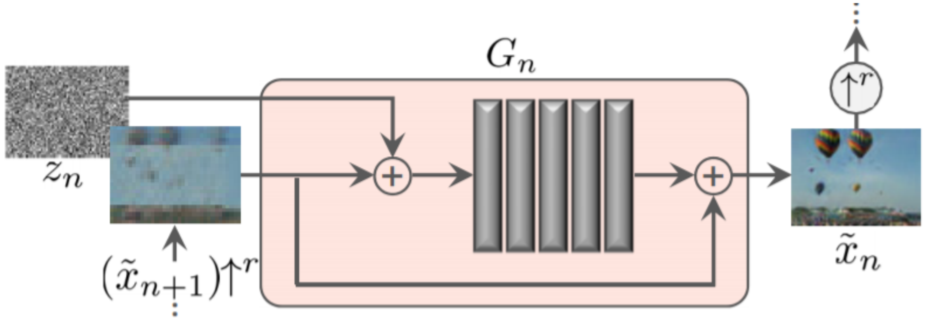


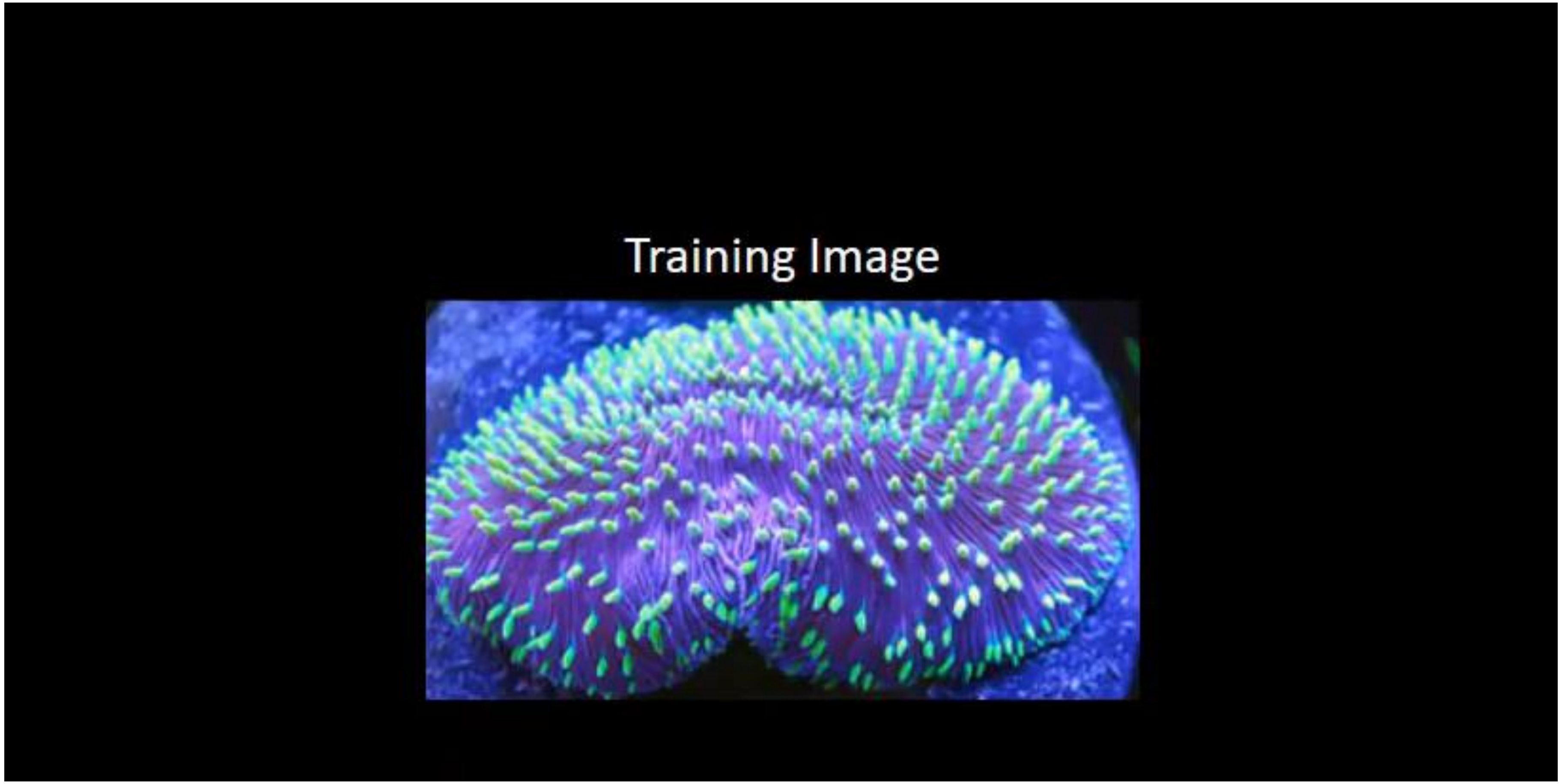
SPADE
ReLU
3x3 Conv
SPADE
ReLU
3x3 Conv
SPADE ResBlk
conv
conv
conv
β
γ
Batch Norm
element-wise
pix2pixHD
SPADE ResBlk
SPADE ResBlk
SPADE ResBlk
SPADE ResBlk


NVIDIA.
Interactive Demo
Semantic Image Synthesis with Spatially Adaptive Normalization
CVPR 2019


Single training image
Random samples from a *single* image


Fake
Real
G_0
G_{N-1}
G_N
D_0
D_{N-1}
D_N
z_0
z_{N-1}
z_N
$\tilde{x}_0$
x_0
$\tilde{x}_{N-1}$
x_{N-1}
$\tilde{x}_N$
x_N
$\uparrow^r$
Training Progression
Mult-scale Patch Generator
Mult-scale Patch Discriminator
Effective Patch Size

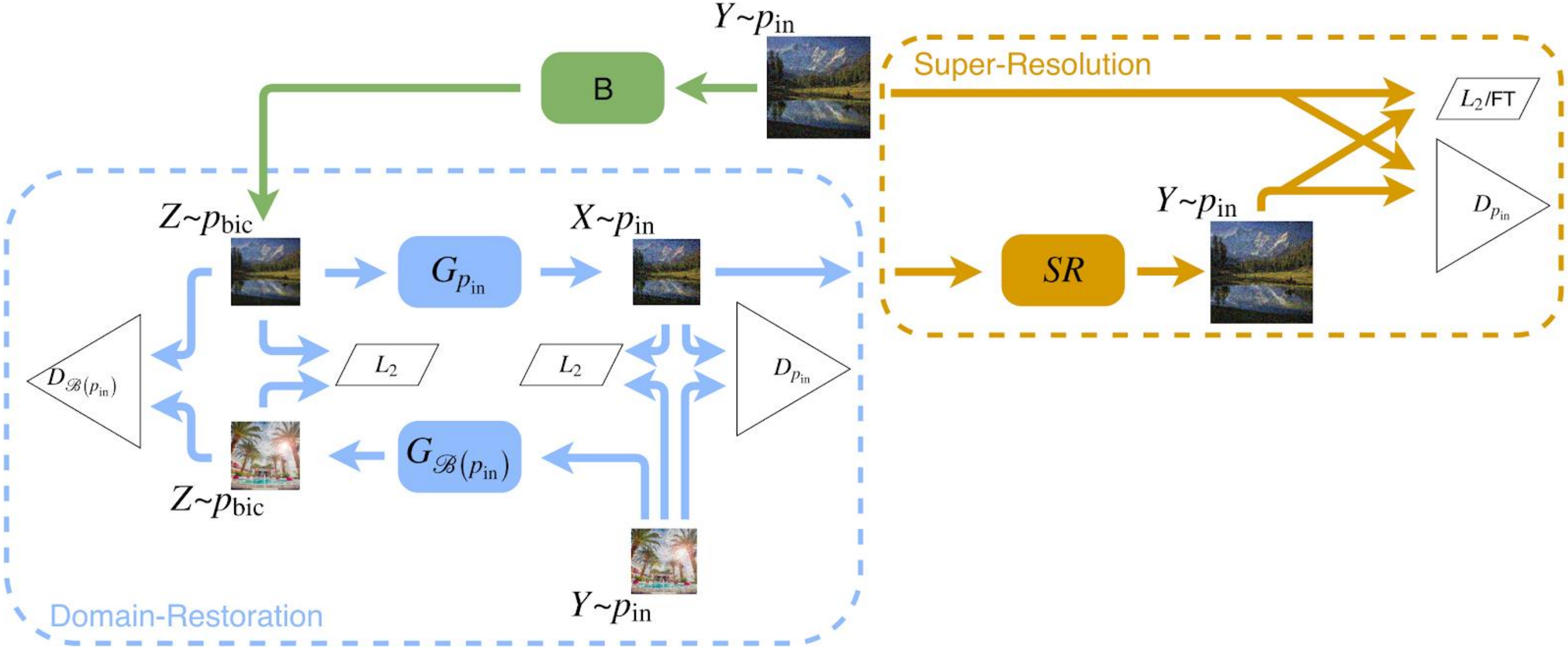

G_n
z_n
$(\tilde{x}_{n+1})\uparrow^r$
$\uparrow^r$
$\tilde{x}_n$


Training Image

And Few More Applications……..

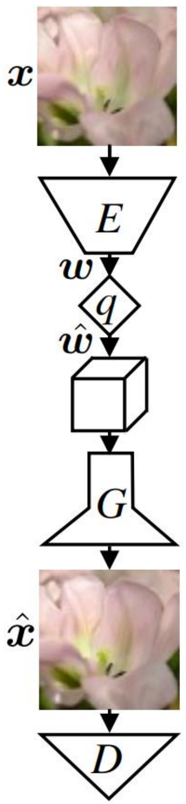
Unsupervised Image Super-Resolution

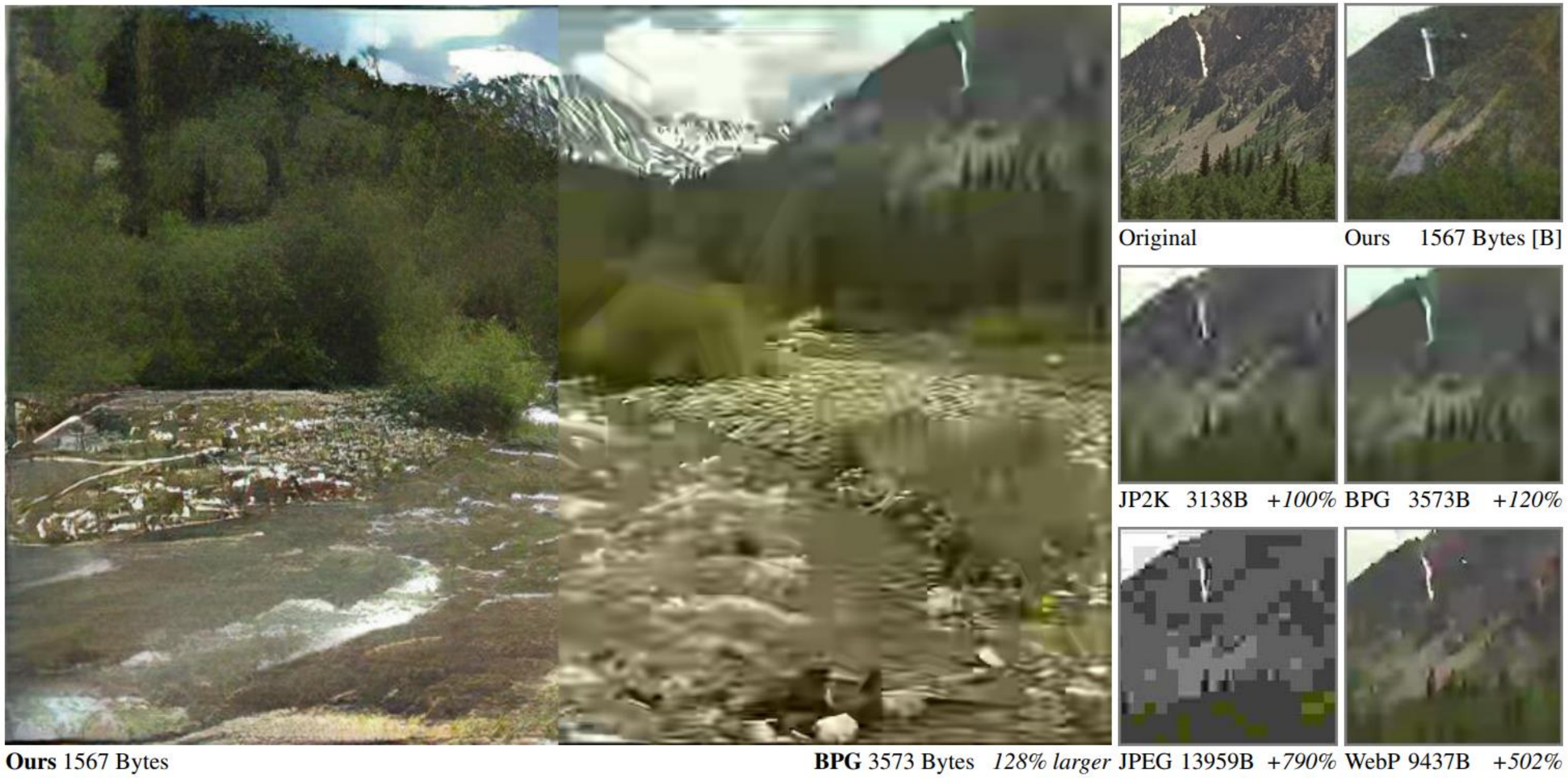
- Learn the downsampling operation with unpaired images
- Train the super-resolution network on the generated paired data



Compression

- Learned image compression for better quality
- Practical lossless compression

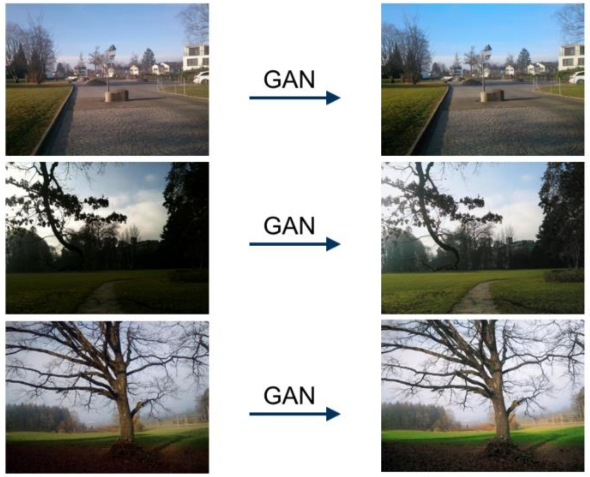




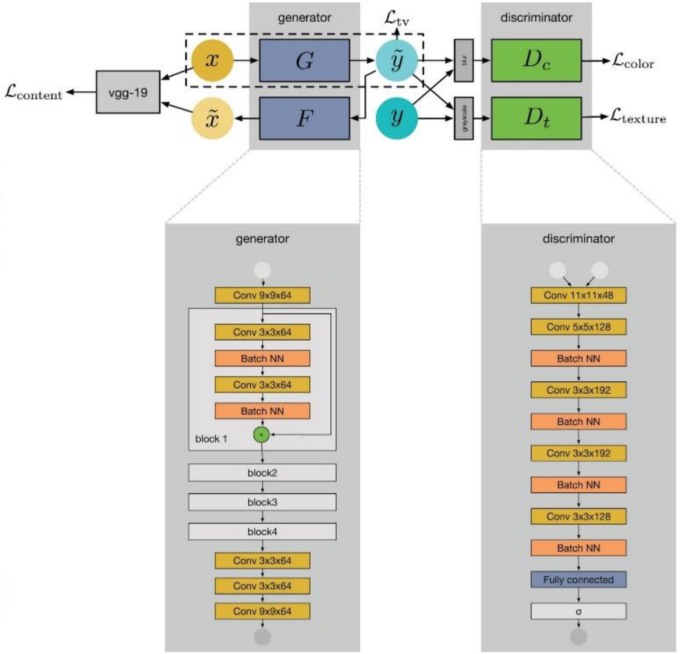
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

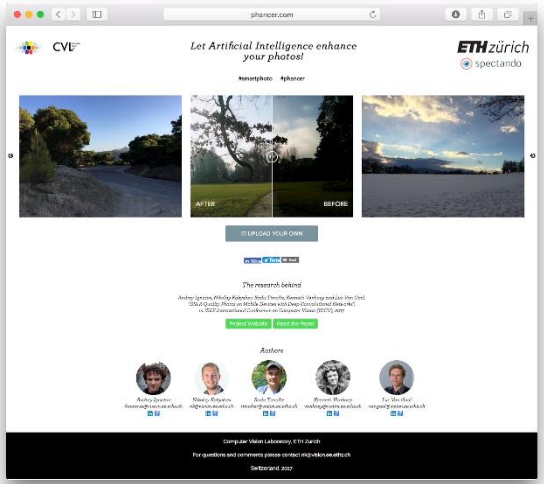
Deep Image Enhancement

- Make pictures taken by a cheap smartphone camera look as if taken by a DSLR!
- Transform 3D sequences into photorealistic movie



Bad quality images Enhanced





Input

WESPE [Ignatov, CVPRW'18]

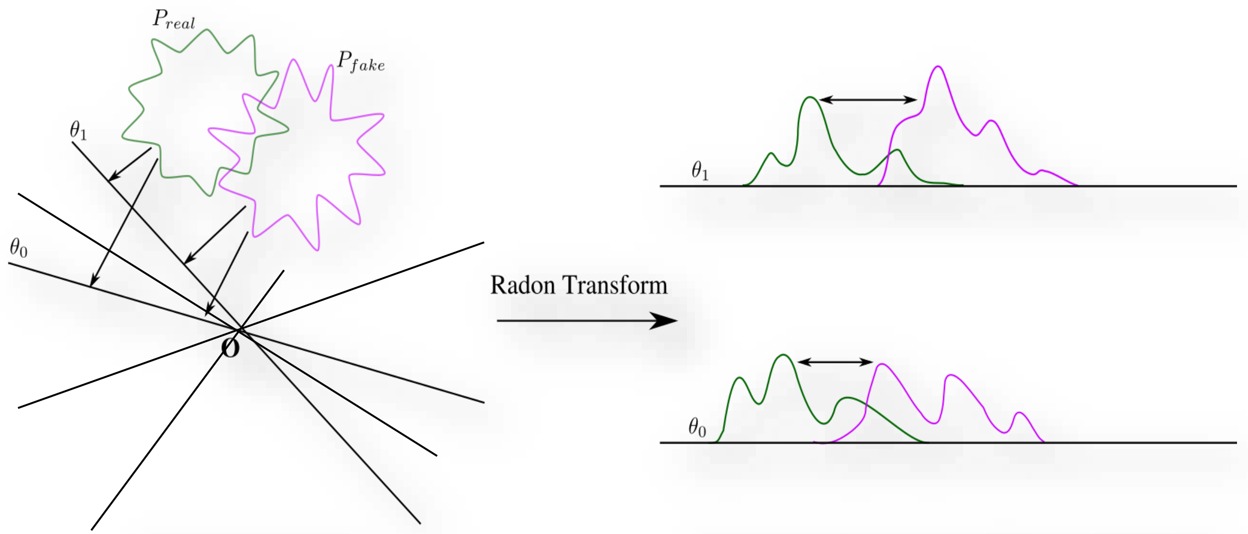
DPE [Chen, CVPR'18]

Proposed MUSPE [Under submission]

A Better Loss……

Sliced Wasserstein Generative Models

Jiqing Wu*[1] Zhiwu Huang*[1] Dinesh Acharya[1] Wen Li[1]
Janine Thoma[1] Danda Pani Paudel[1] Luc Van Gool[1,2]
[1] Computer Vision Lab, ETH Zurich, Switzerland [2] VISICS, KU Leuven, Belgium



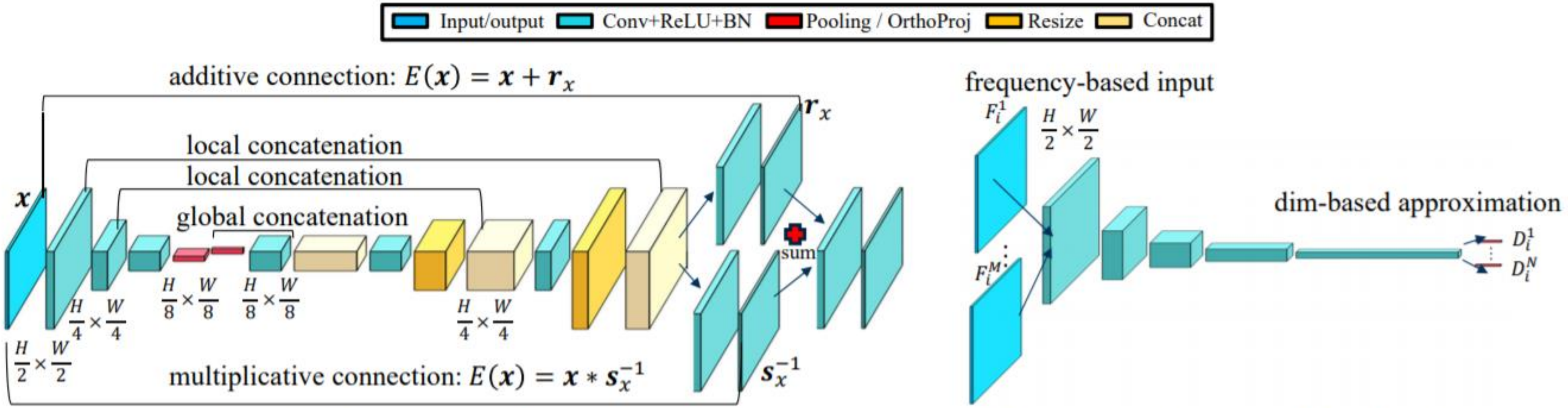
Orthogonal Projection
Using Radon Transform

Wasserstein Distance of 1-D Marginal Distributions
is integrated over all possible theta.

A Better Architecture ….

DIVIDE-AND-CONQUER ADVERSARIAL LEARNING FOR HIGH-RESOLUTION IMAGE AND VIDEO ENHANCEMENT

Zhiwu Huang[†], Danda Pani Paudel[†], Guanju Li[†], Jiqing Wu[†], Radu Timofte[†], Luc Van Gool[†‡]
[†]Computer Vision Lab, ETH Zurich, Switzerland, [‡]VISICS, KU Leuven, Belgium

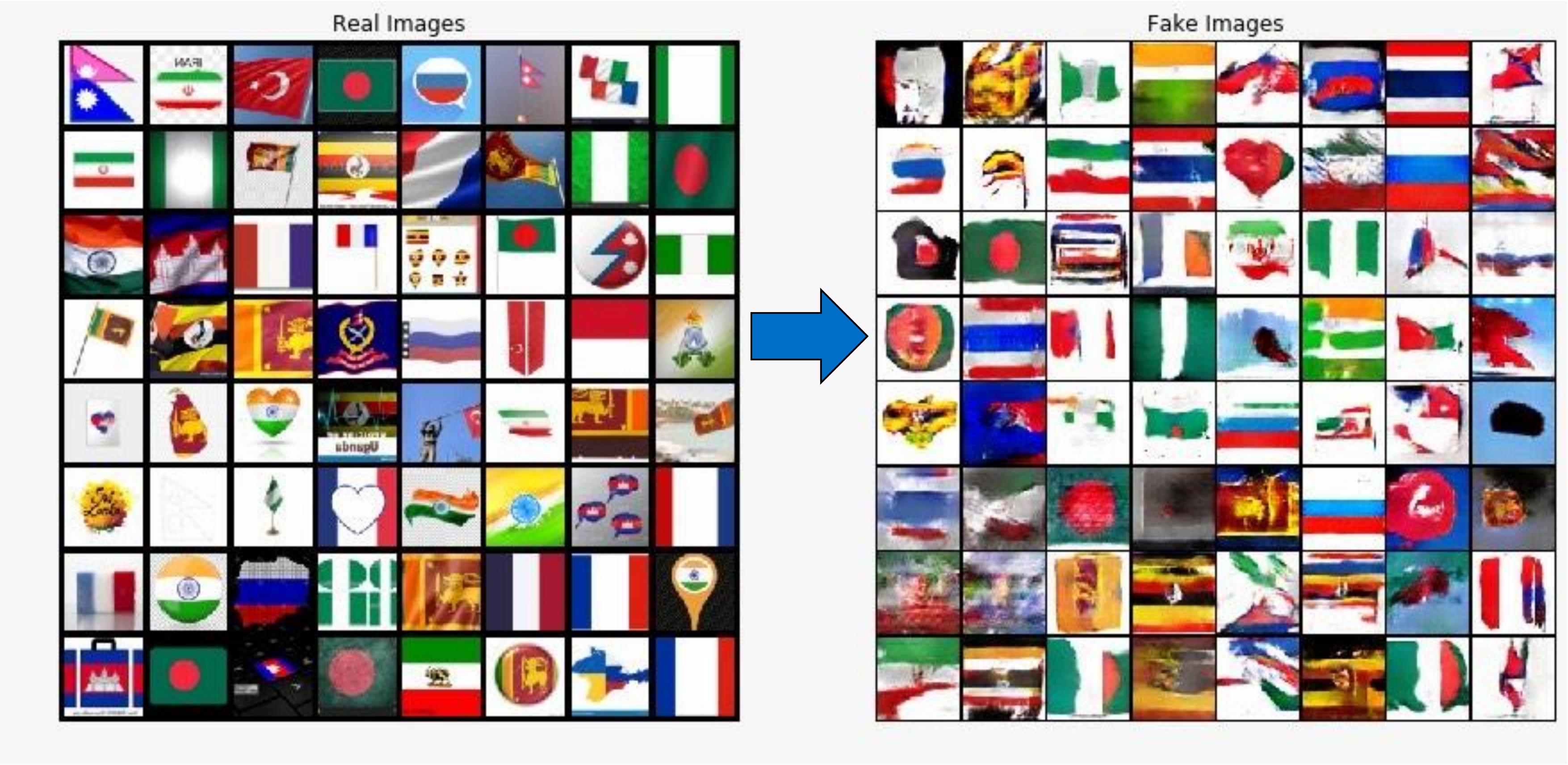


(a) Enhancer for Peception-based Division

(b) Discriminator for Freq- and Dim-based Division


Yes,
we GAN.
BOREALIS AI

Lab Session



Q&A