

인공지능 기반 설계 이론 및 사례 연구

11차) Generative Adversarial Network (GAN)

2020년 11월

강남우

기계시스템학부
숙명여자대학교



Generative Adversarial Network (GAN) – How to work

VAEs define intractable density function with latent \mathbf{z} :

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

Cannot optimize directly, derive and optimize lower bound on likelihood instead

What if we give up on explicitly modeling density, and just want ability to sample?

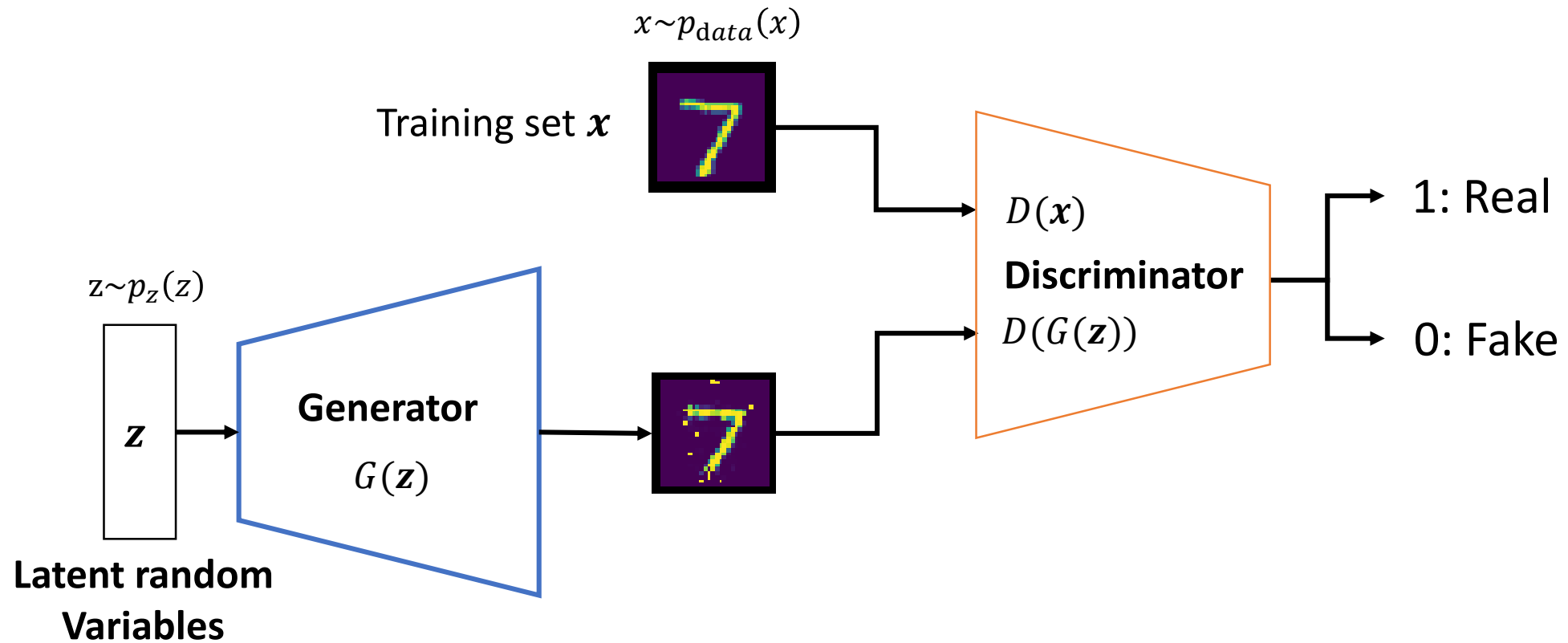
GANs: don't work with any explicit density function!

Instead, take game-theoretic approach: learn to generate from training distribution through 2-player game

GAN – How to work

Training GANs: Two-player game

- **Generator network:** try to fool the discriminator by generating real-looking images
- **Discriminator network:** try to distinguish between real and fake images

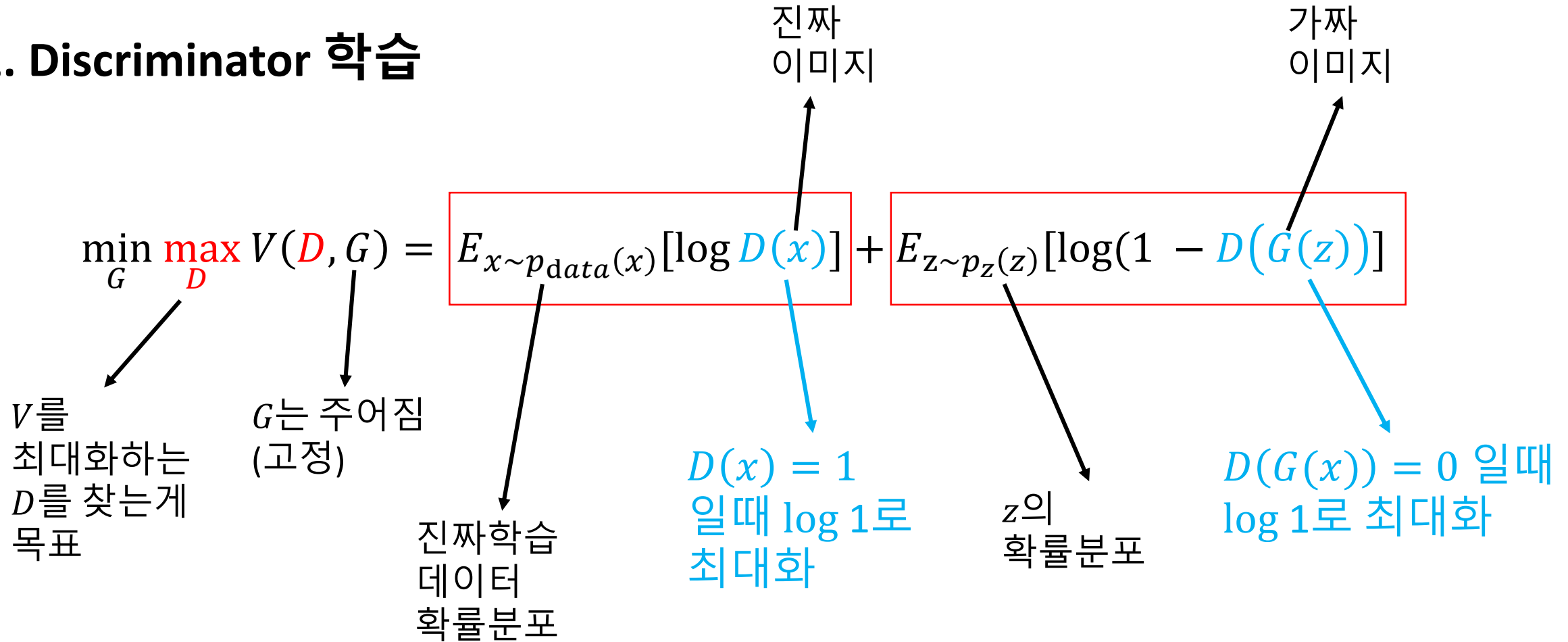


Objective function (minimax game)

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

GAN – How to work

1. Discriminator 학습



GAN – How to work

2. Generator 학습

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

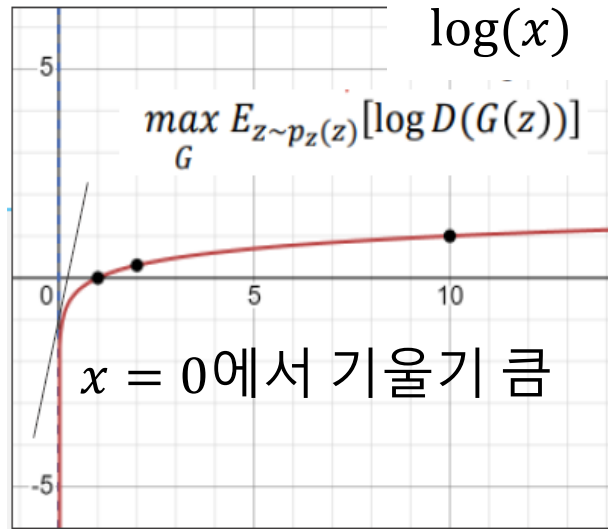
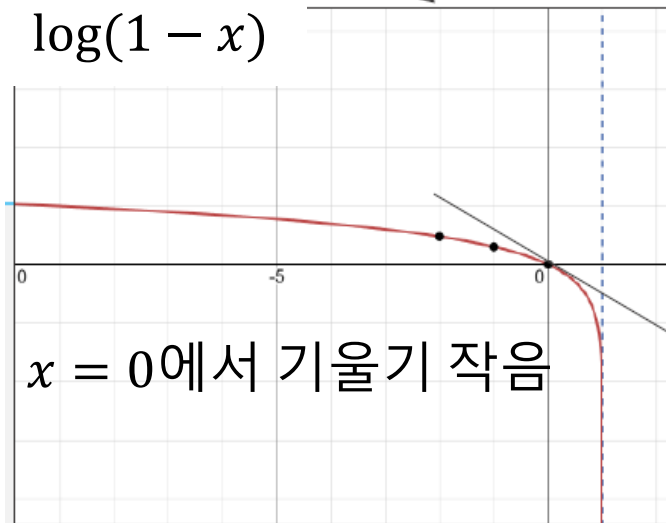
V 를
최대화하는
 G 를 찾는게
목표

D 는 주어짐
(고정)

학습하지 않음

가짜 이미지

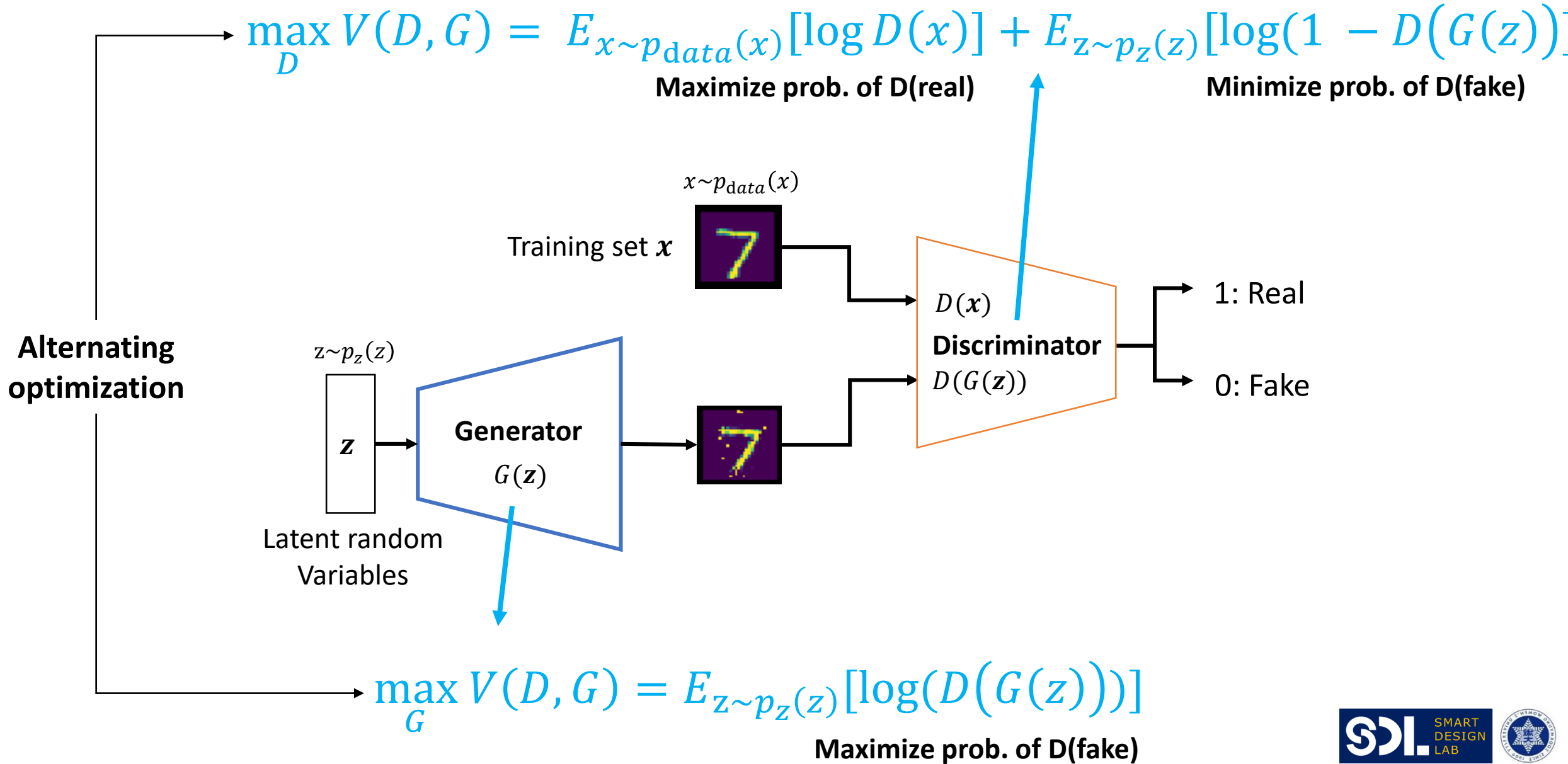
$D(G(x)) = 1$ 일때
 $\log 0$ 으로 최소화



처음에는 $D(G(z))$ 가 0에 가까움

$$\max_G E_{z \sim p_z(z)} [\log(D(G(z)))]$$

GAN – How to work



GAN - Proof

GAN eventually minimizes the distance between the **real data distribution** and the **model distribution**.

$$\min_G \max_D V(D, G) = \min_{G, D} JSD(p_{data} || p_g)$$

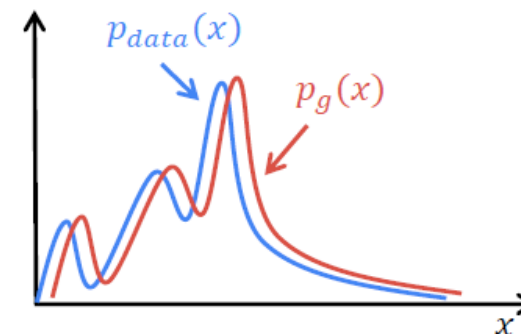
Objective function of GANs

$$E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Jenson-Shannon divergence

$$JSD(P || Q) = \frac{1}{2} KL(P || M) + \frac{1}{2} KL(Q || M)$$

$$\text{Where } M = \frac{1}{2} (P + Q)$$



GAN - Proof

1. Discriminator 최적해 증명 (1/2)

High dimensional vector (e.g. 128x128) Low dimensional vector (e.g. 128)

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

Objective function of GAN Real data distribution Gaussian distribution

Fix G to make it a function of D

$$D^*(x) = \arg \max_D V(D) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Optimal D Get D when V(D) is maximum

Sampling x from p_g instead of sampling z from p_z

$z \rightarrow G(z) \rightarrow x \rightarrow D(x)$

$$= E_{x \sim p_{data}(x)} [\log D(x)] + E_{x \sim p_g(x)} [\log(1 - D(x))]$$

$$= \int_x p_{data}(x) \log D(x) dx + \int_x p_g(x) \log(1 - D(x)) dx$$

$E_{x \sim p(x)} [f(x)] = \int_x p(x) f(x) dx$

Basic property of integral

$$= \int_x p_{data}(x) \log D(x) + p_g(x) \log(1 - D(x)) dx$$

Now we need to find D(x) which makes the function inside integral maximum.

GAN - Proof

1. Discriminator 최적해 증명 (2/2)

$$D^*(x) = \arg \max_D V(D) \quad \text{The function inside integral}$$

$$= \arg \max_D p_{data}(x) \log D(x) + p_g(x) \log(1 - D(x))$$

Substitute $a = p_{data}(x), y = D(x), b = p_g(x)$

$$a \log y + b \log(1 - y)$$

Differentiate with respect to $D(x)$ using $\frac{d}{dx} \log f(x) = \frac{f'(x)}{f(x)}$
Note that $D(x)$ cannot affect to $p_{data}(x)$ and $p_g(x)$

$$\frac{a}{y} + \frac{-b}{1-y} = \frac{a - (a+b)y}{y(1-y)}$$

Find the point where the derivative value is 0

$$\frac{a - (a+b)y}{y(1-y)} = 0$$

It has a maximum value when $y = \frac{a}{a+b}$

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

Substitute $a = p_{data}(x), y = D(x), b = p_g(x)$

GAN - Proof

2. Generator 최적해 증명

$$\min_G \max_D V(D, G) = \min_G V(D^*, G) \quad \text{Optimal D}$$

$$V(D^*, G) = E_{x \sim p_{data}(x)} [\log D^*(x)] + E_{x \sim p_g} [\log(1 - D^*(x))] \quad E_{x \sim p(x)} [f(x)] = \int_x p(x) f(x) dx$$

G should minimize

$$= \int_x p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} dx + \int_x p_g(x) \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} dx$$

$$= -\log 4 + \log 4 + \int_x p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} dx + \int_x p_g(x) \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} dx$$

$$= -\log 4 + \int_x p_{data}(x) \log \frac{2 \cdot p_{data}(x)}{p_{data}(x) + p_g(x)} dx + \int_x p_g(x) \log \frac{2 \cdot p_g(x)}{p_{data}(x) + p_g(x)} dx$$

$$KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

$$= -\log 4 + KL(p_{data} || \frac{p_{data} + p_g}{2}) + KL(p_g || \frac{p_{data} + p_g}{2})$$

$$JSD(P||Q) = \frac{1}{2} KL(P||M) + \frac{1}{2} KL(Q||M)$$

$$= -\log 4 + 2 \cdot JSD(p_{data} || p_g) \quad \text{Optimizing } V(D, G) \text{ is same as minimizing } JSD(p_{data} || p_g)$$

$$\Rightarrow p_{data}(x) = p_g(x) \Rightarrow D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} = \frac{1}{2}$$

VAE vs. GAN

VAE



Blurry

Tend to remember input images

Smooth

GAN



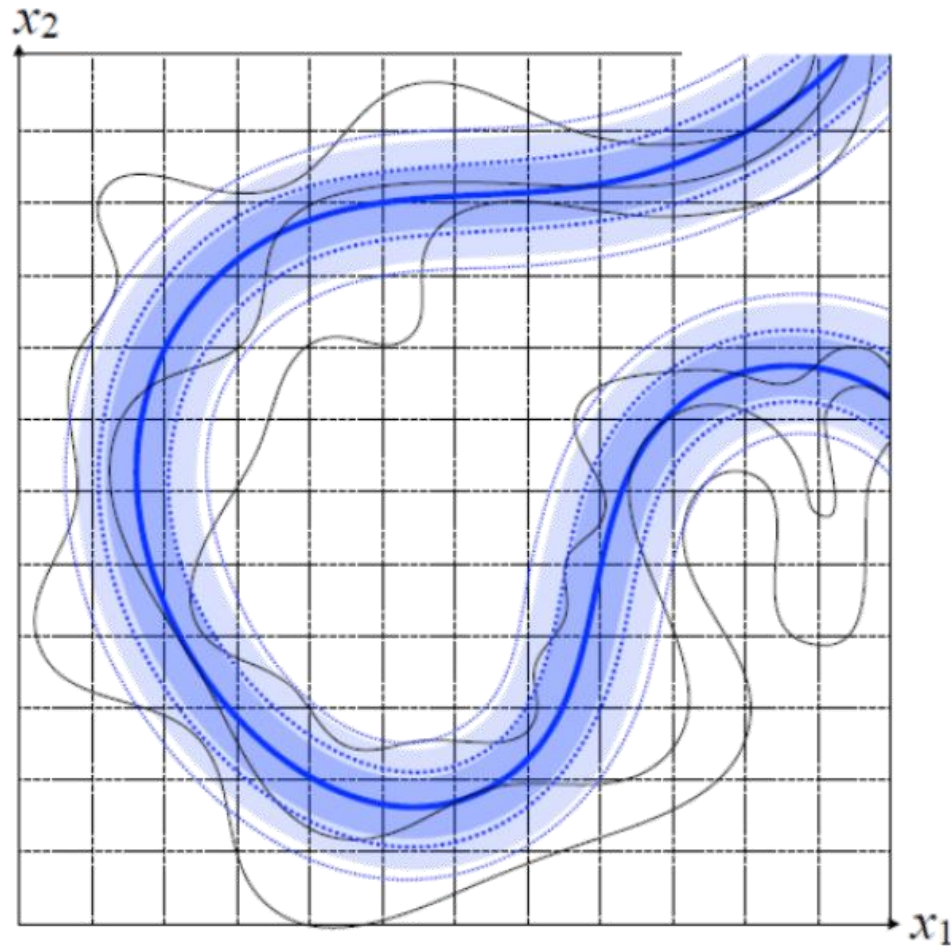
Sharp

Generate new unseen images

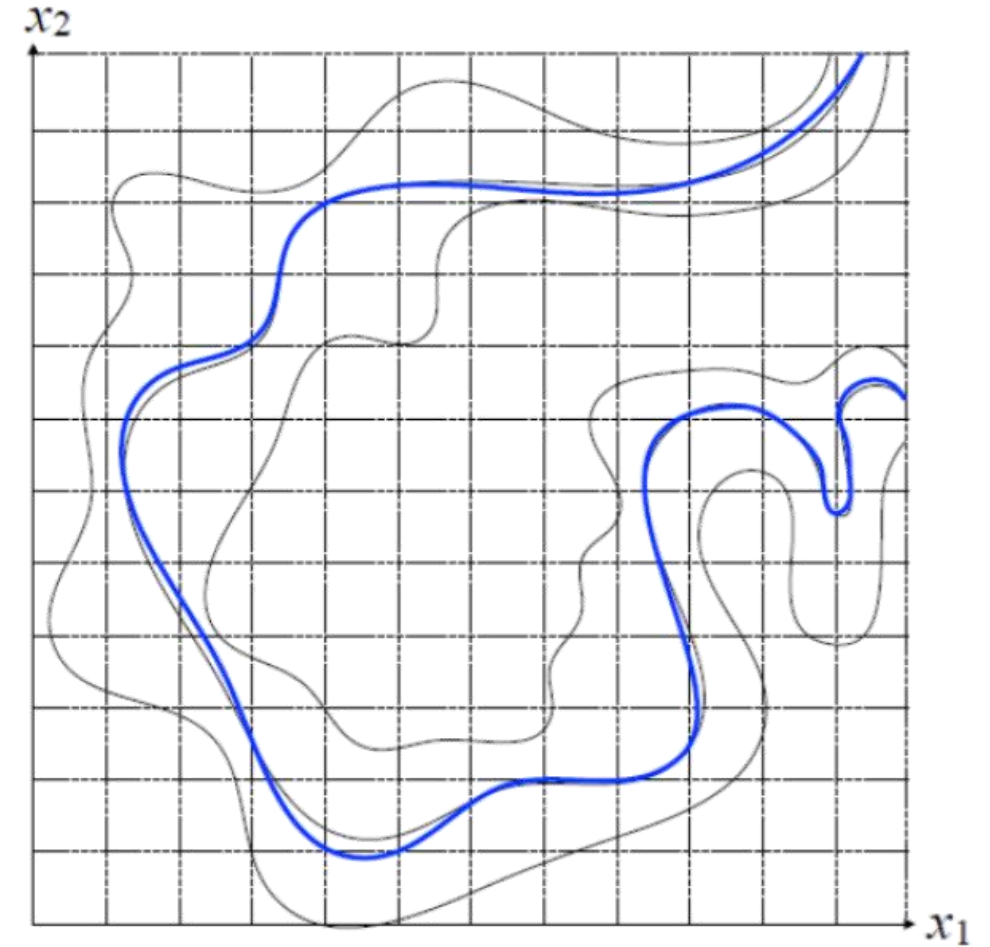
Mode collapse

Unstable convergence

VAE vs. GAN



**VAE : maximum likelihood
approach**



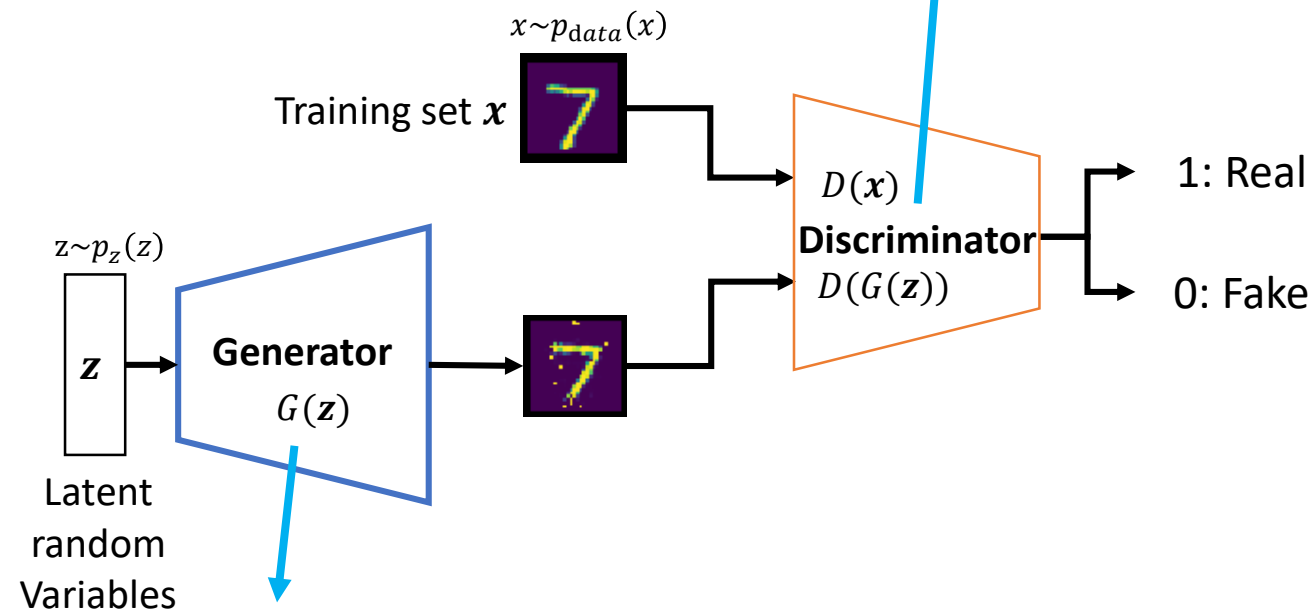
GAN

GAN Coding

```
loss_D = tf.reduce_mean(tf.log(D_real) + tf.log(1 - D_gene))
```

$$\max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

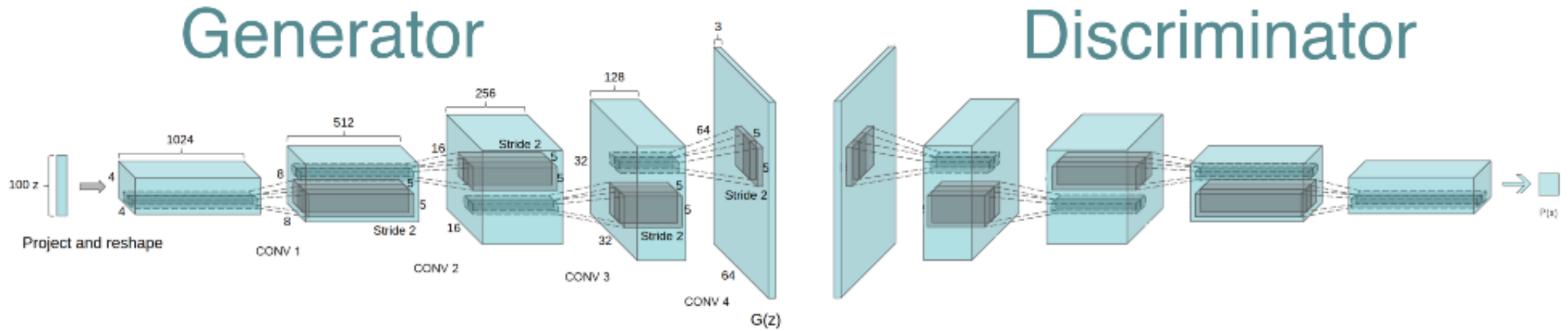
Alternating
optimization



$$\max_G V(D, G) = E_{z \sim p_z(z)} [\log(D(G(z)))]$$

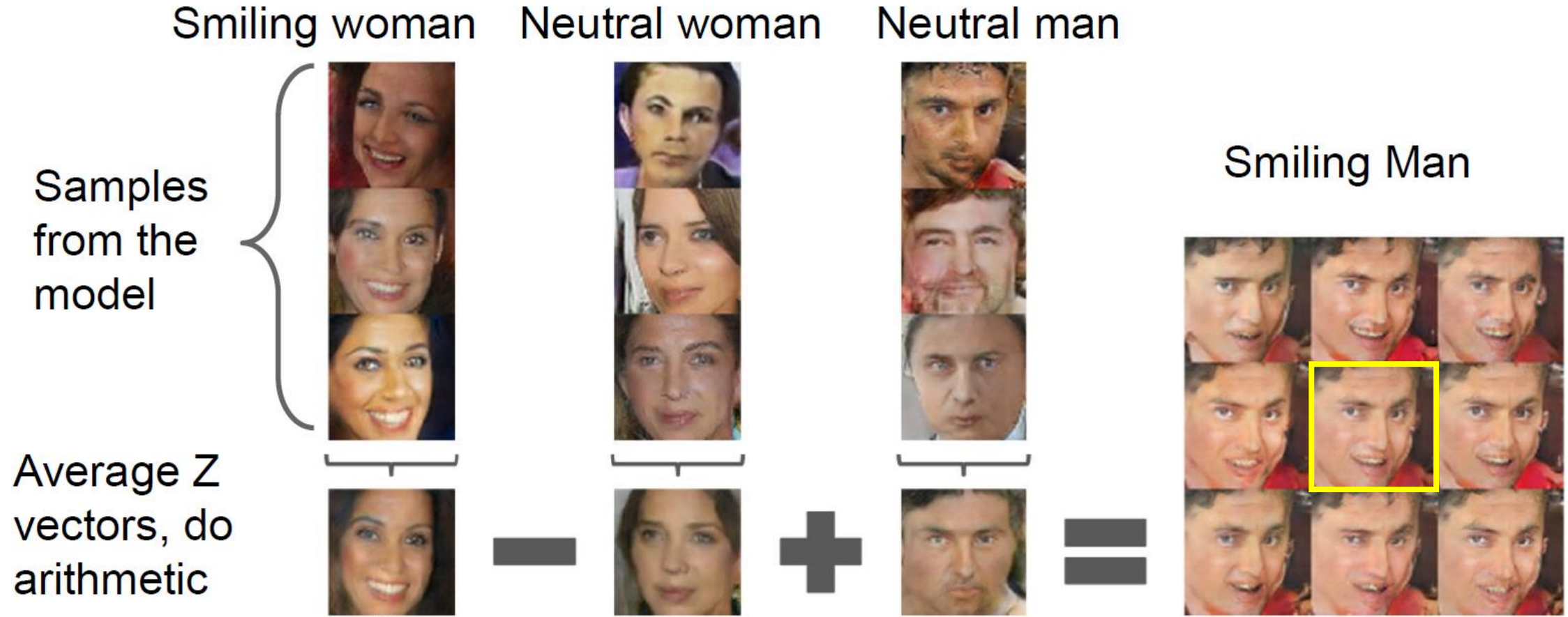
```
loss_G = tf.reduce_mean(tf.log(D_gene))
```

Deep Convolutional GAN (DCGAN)



	Generator	Discriminator
Pooling Layers	Not used. But use stride convolutions instead	
Batch normalization	Use except output layer	Use except input layer
Fully connected hidden layers	Not used	
Activation function	ReLU for all layers except for the output, which uses Tanh	LeakyReLU for all layers

DCGAN: Interpretable Vector Math



Results of doing the same arithmetic in pixel space

DCGAN: Interpretable Vector Math

Glasses man



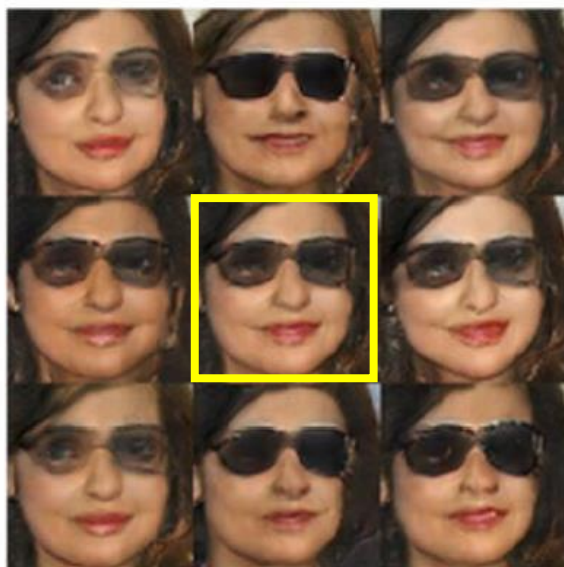
No glasses man



No glasses woman



Woman with glasses















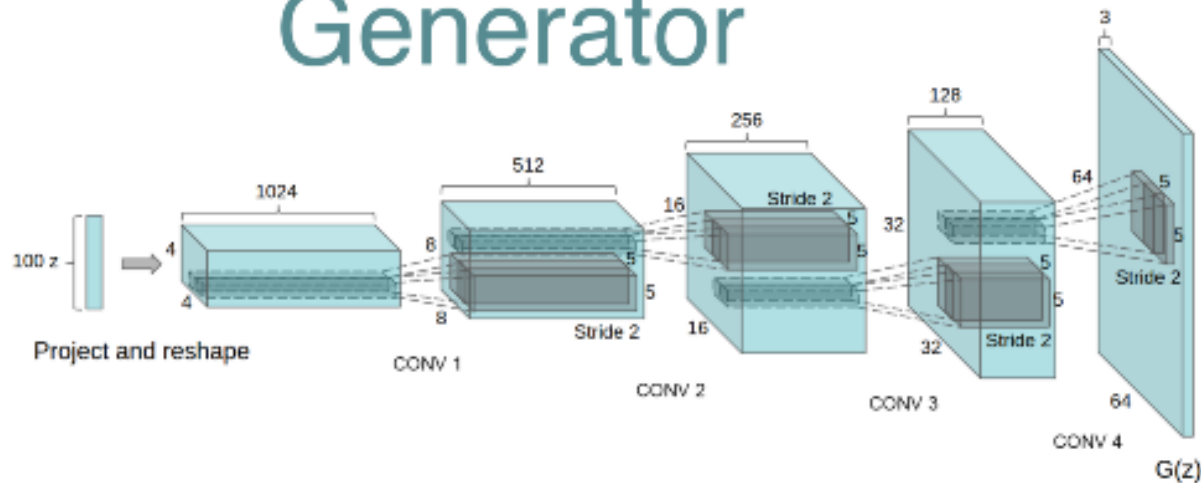


Results of doing the same arithmetic in pixel space

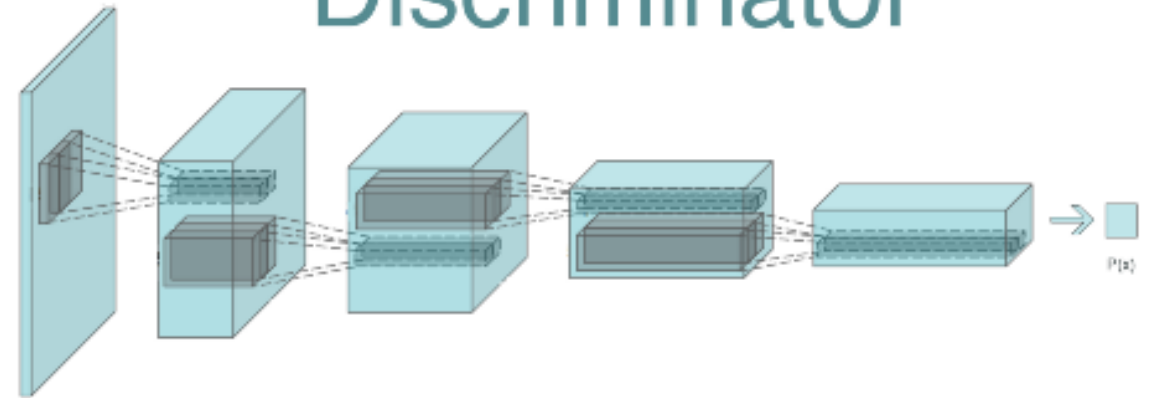
DCGAN Coding

Go to <http://www.smartdesignlab.org/dl.html>

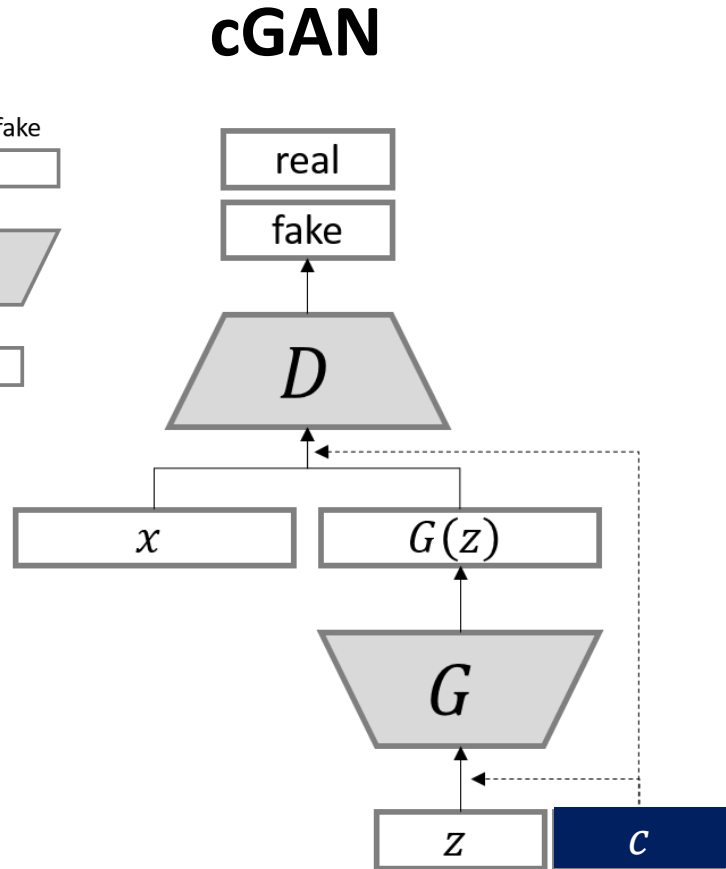
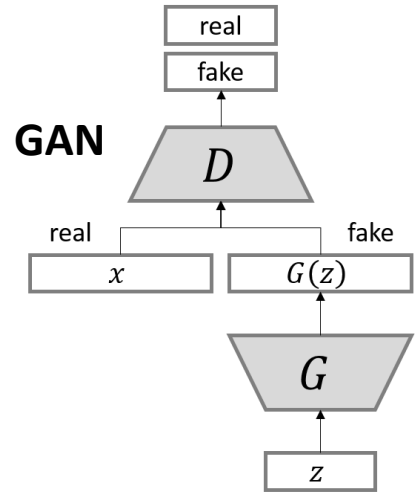
Generator



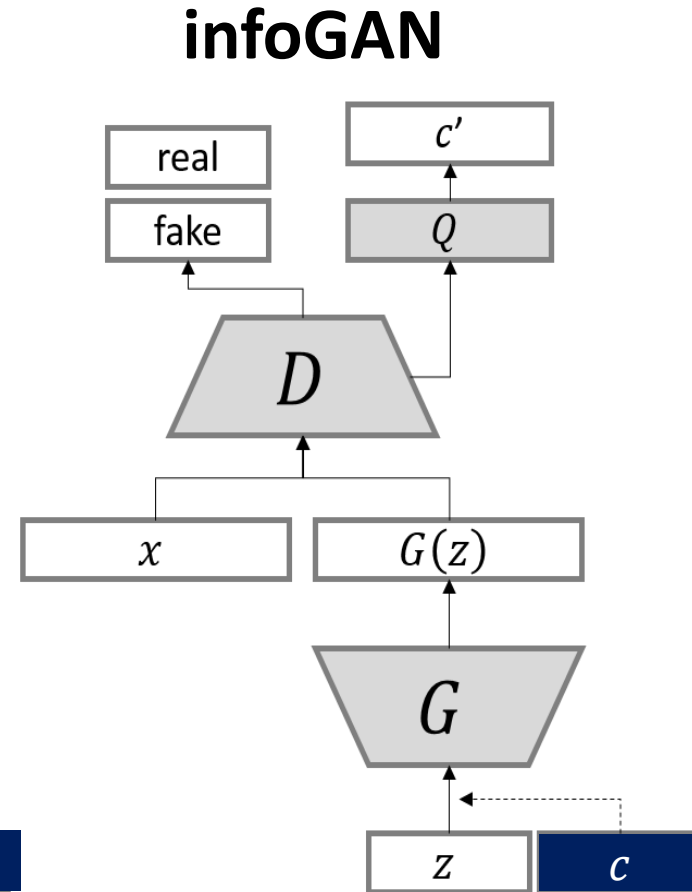
Discriminator



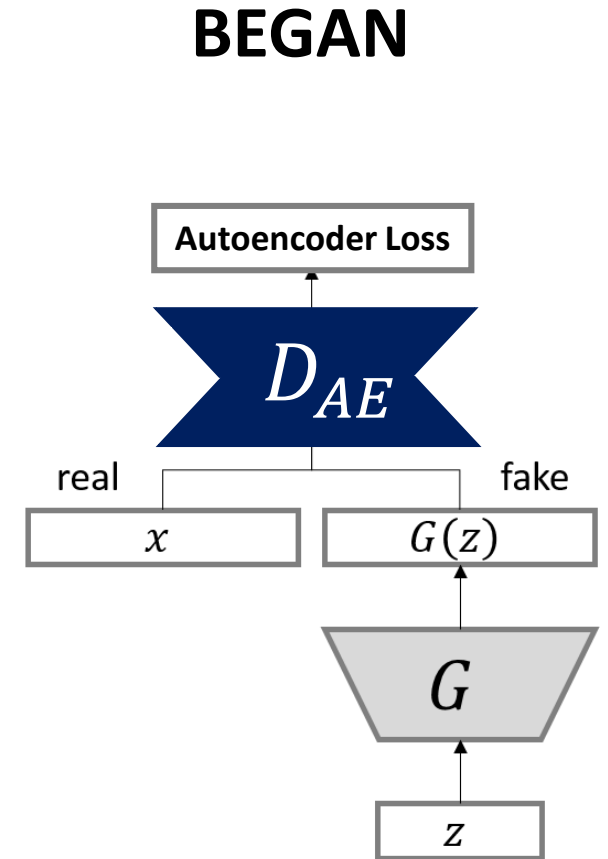
Various GAN Structures



(Mirza & Osindero, 2014)

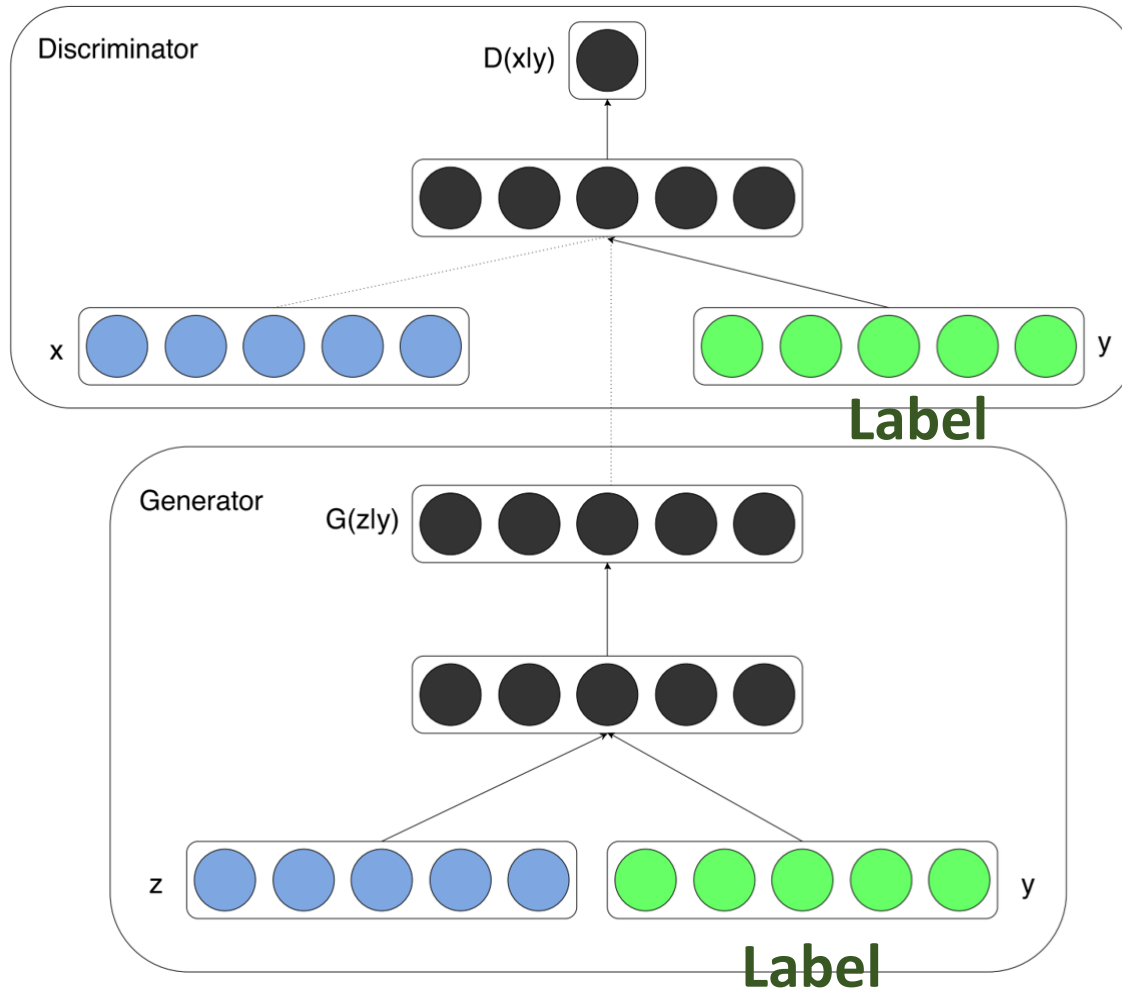


(Chen, et al., 2016)

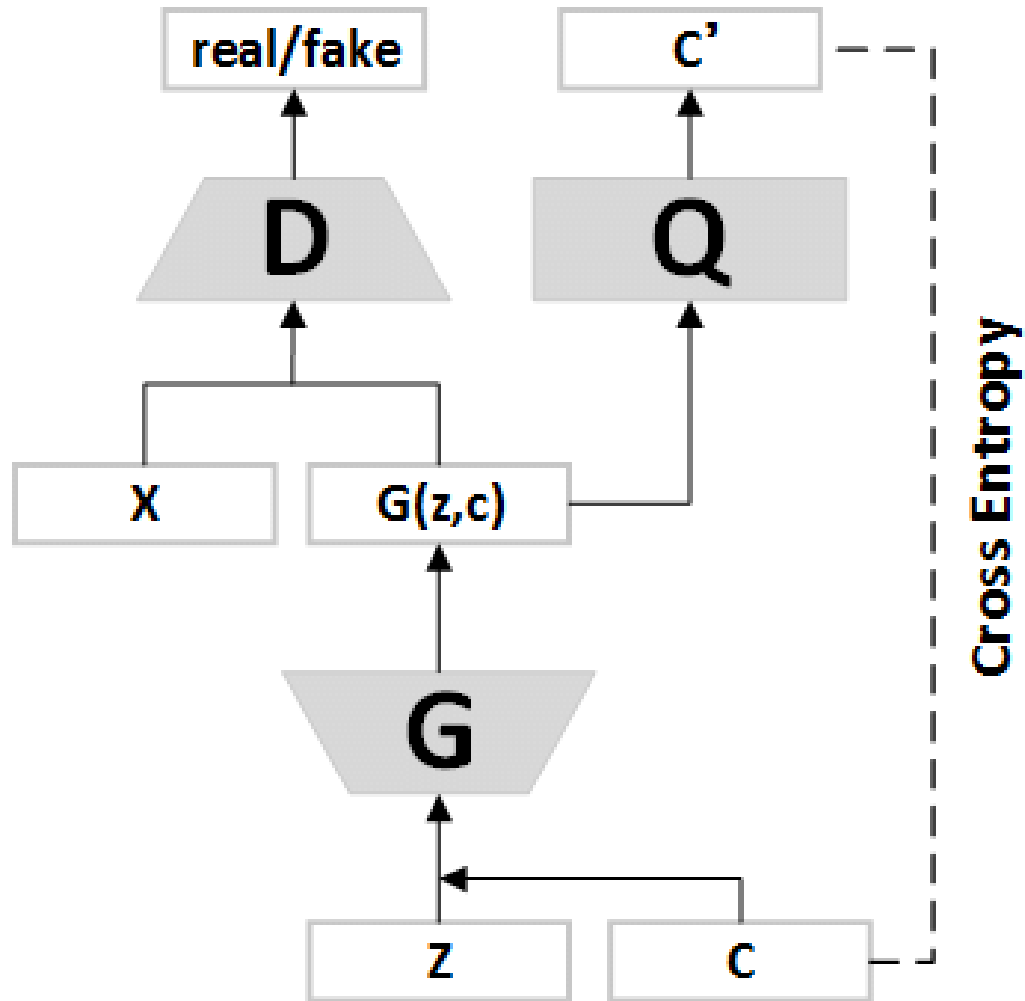


(Berthelot et al., 2017)

Conditional GAN (cGAN)

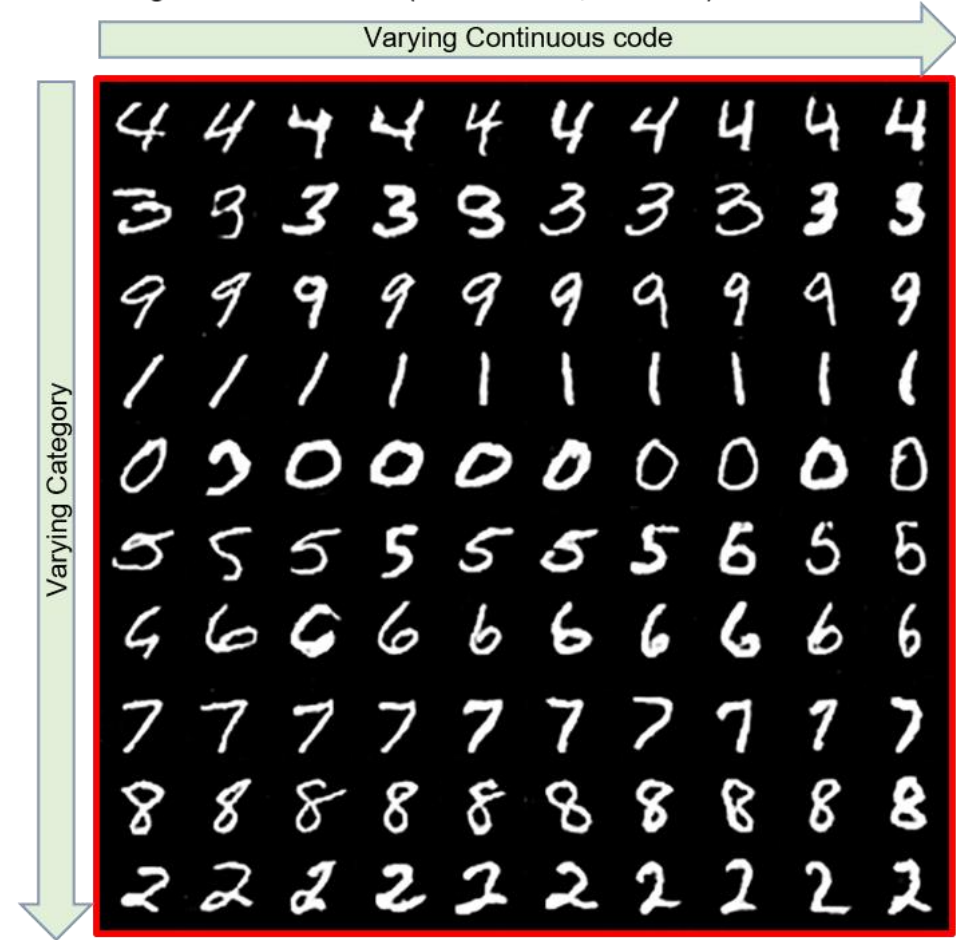


InfoGAN



MNIST digits generated using InfoGAN

Infogan는 의미있는 개념(ex. 숫자 유형, 기울기 등)을 포착한다.



Row는 Latent categorical value에 따라 대응하며, Column은 Latent continuous variable에 따라 대응한다.

Boundary Equilibrium GAN (BEGAN)

Autoencoder loss: $\mathcal{L}(v) = |v - A(v)|^\eta$

where

$$\begin{cases} \mathbb{R}^{N_x} \rightarrow \mathbb{R}^{N_x} & \text{is an autoencoder function} \\ \eta \in \{1, 2\} & \text{is a target norm} \\ v \in \mathbb{R}^{N_z} & \text{is a sample of dimension } N_z \end{cases}$$

Discriminator: $\mathcal{L}_D = \mathcal{L}(x) - k_t \mathcal{L}(G(z_D))$ for θ_D

진짜가
복원이 잘되도록 가짜가
복원이 안되도록

Generator:

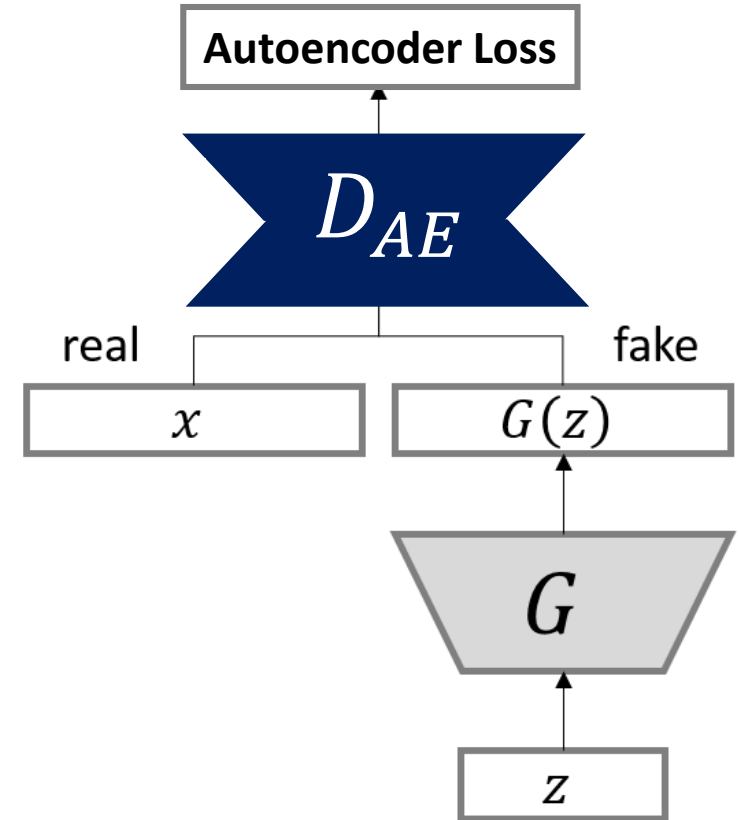
$\mathcal{L}_G = \mathcal{L}(G(z_G))$ for θ_G

가짜가
복원이 잘되도록

Discriminator와 Generator의 균형 맞추기

0부터 점점
커지기 $k_{t+1} = k_t + \lambda_k (\gamma \mathcal{L}(x) - \mathcal{L}(G(z_G)))$

Learning
rate



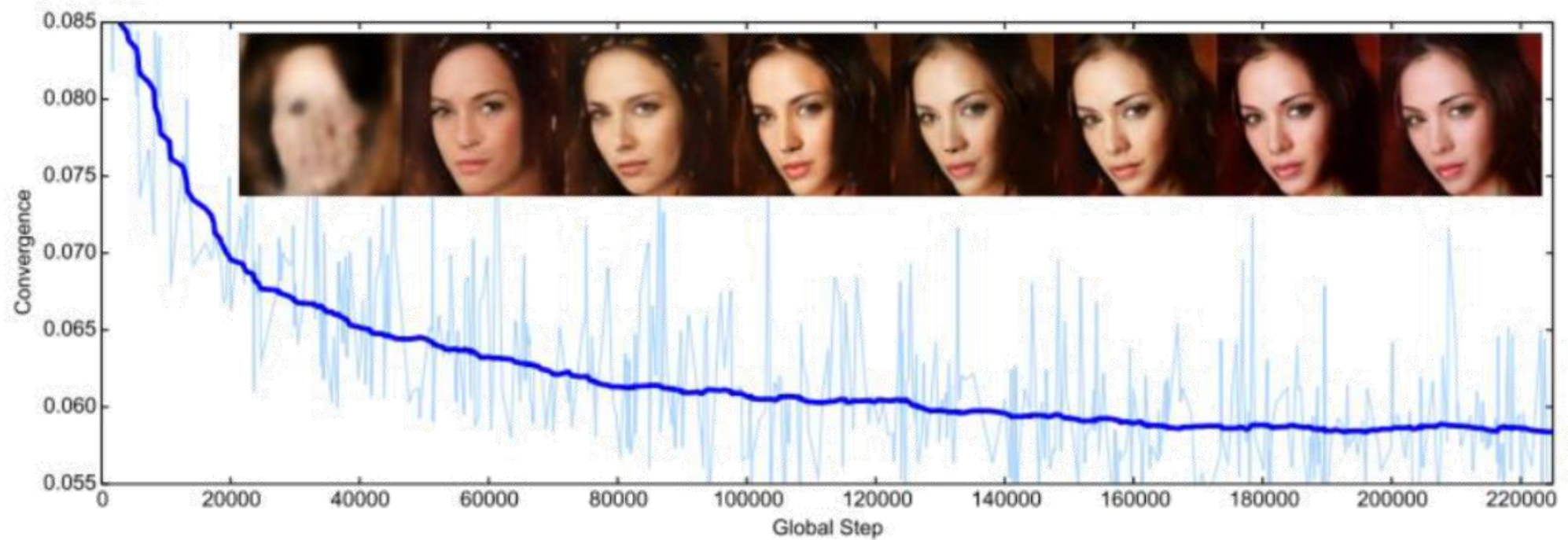
(Berthelot et al., 2017)

$\gamma \downarrow \rightarrow \mathcal{L}(x)$ 에 집중 $\rightarrow G(z)$ 다양성 떨어짐, 퀄리티 증가
 $\gamma \uparrow \rightarrow \mathcal{L}(G(z))$ 에 집중 $\rightarrow G(z)$ 다양성 증가, 퀄리티 감소

Boundary Equilibrium GAN (BEGAN)

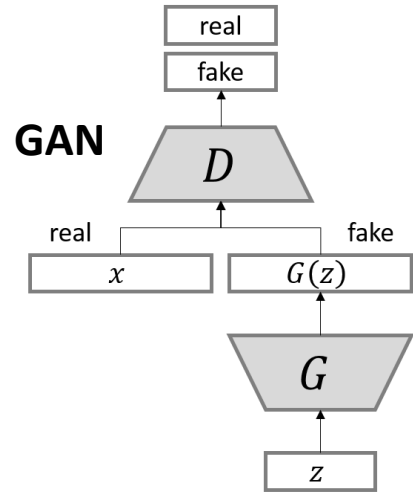
Convergence Measure

$$M_{global} = \mathcal{L}(x) + |\gamma(\mathcal{L}(x)) - \mathcal{L}(G(z_G))|$$

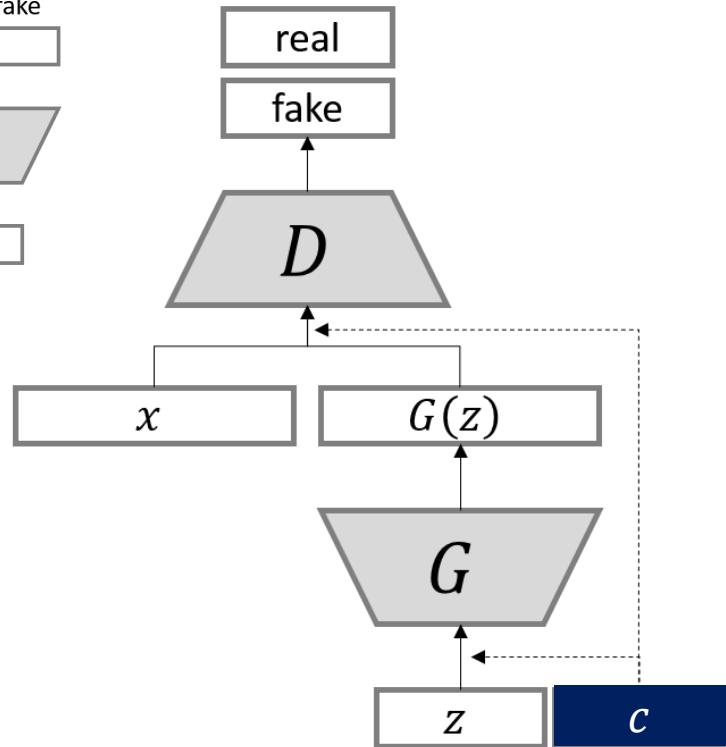


Various GANs Coding

Go to <http://www.smartdesignlab.org/dl.html>

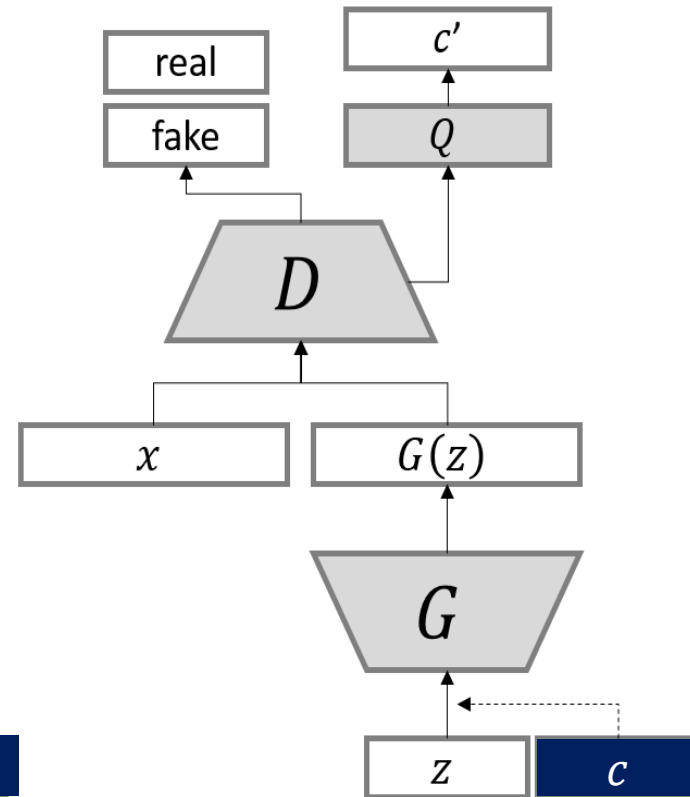


cGAN



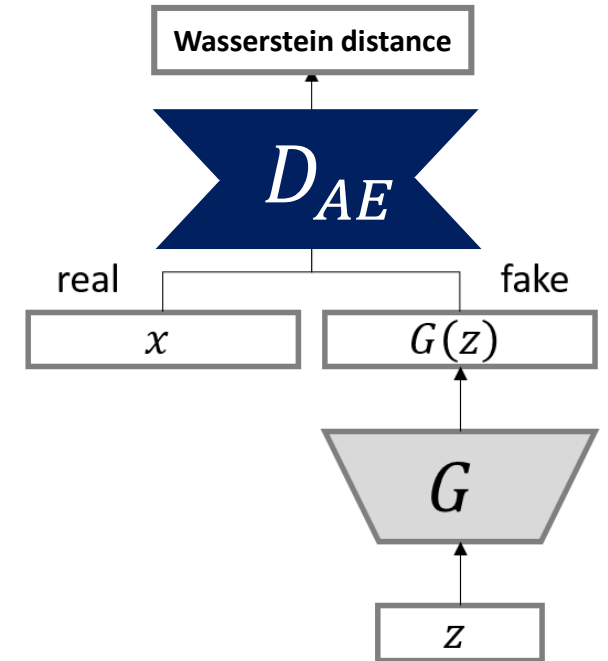
(Mirza & Osindero, 2014)

infoGAN



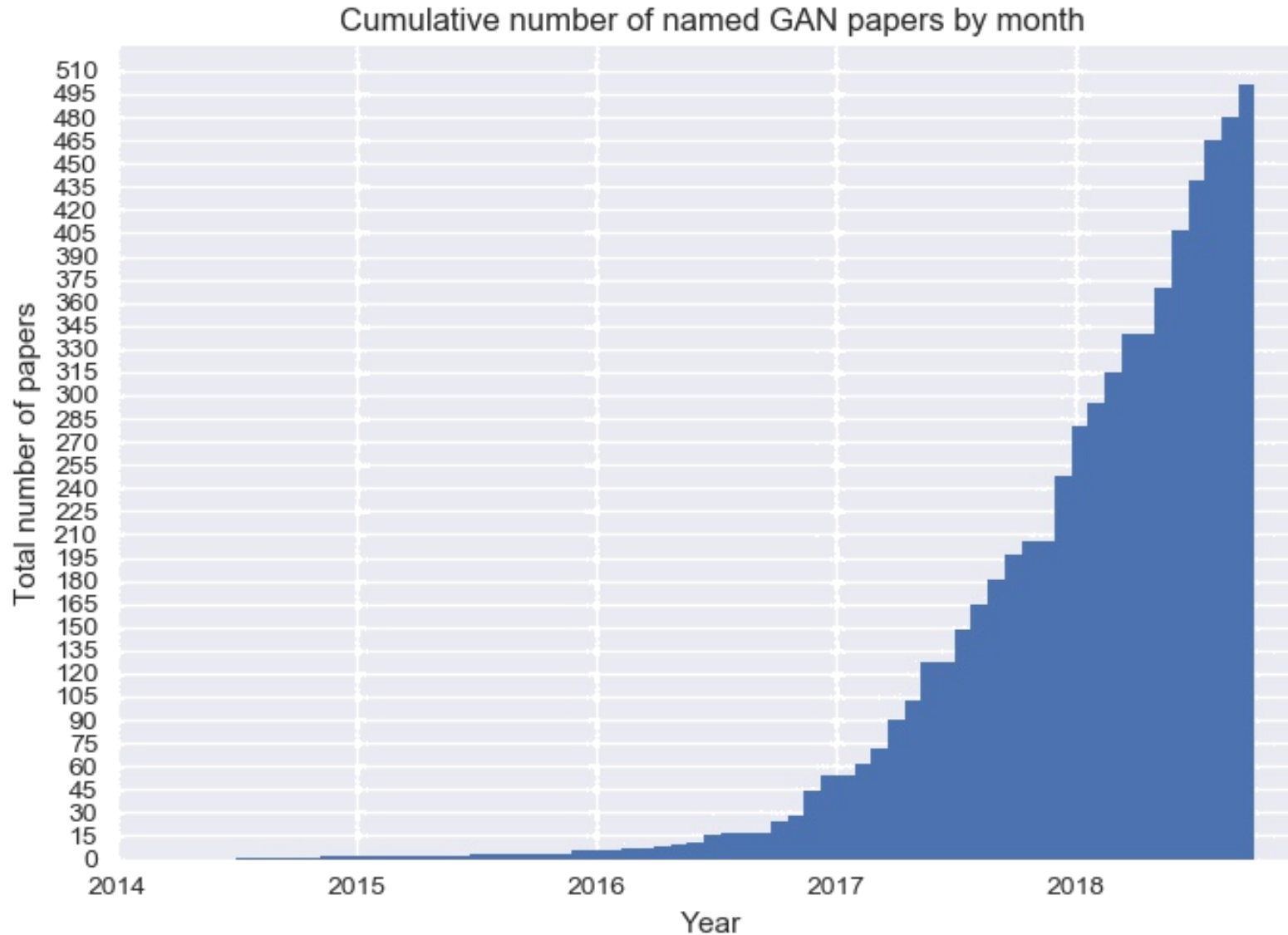
(Chen, et al., 2016)

BEGAN



(Berthelot et al., 2017)

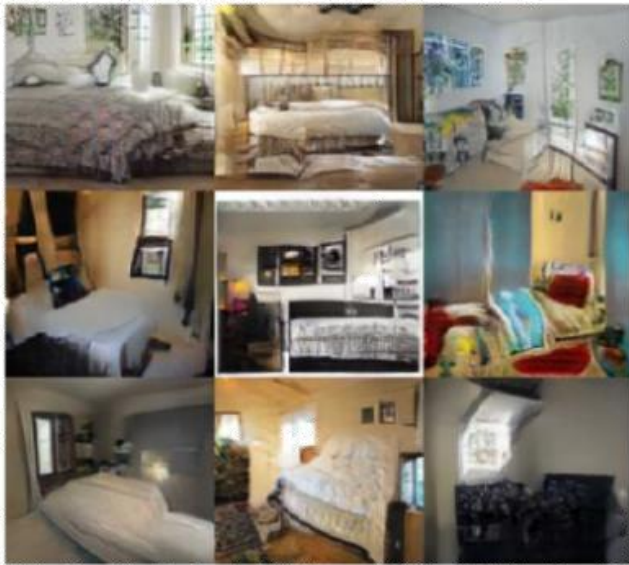
GAN ZOO



- GAN
- DCGAN
- WGAN
- CGAN
- LSGAN
- SGAN
- ACGAN
- InfoGAN
- CycleGAN
- ⋮

2017: Explosion of GANs

Better training and generation



LSGAN, Zhu 2017.



Wasserstein GAN,
Arjovsky 2017.
Improved Wasserstein
GAN, Gulrajani 2017.



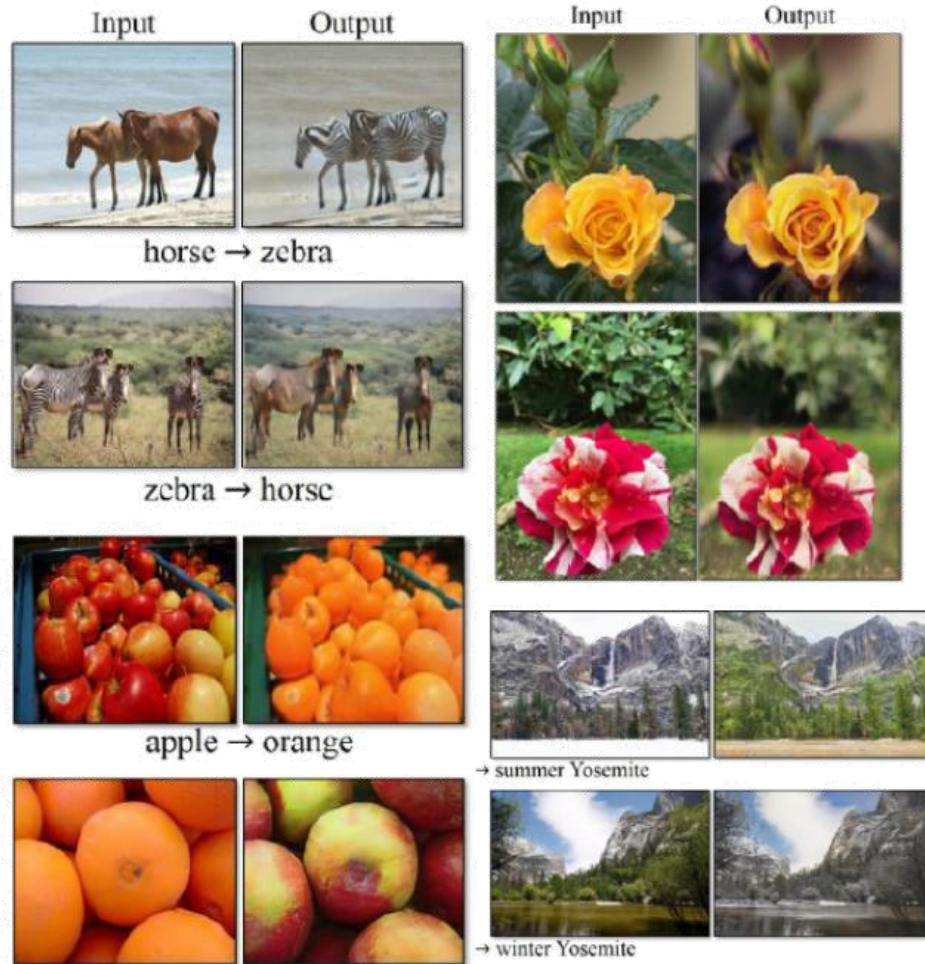
Progressive GAN, Karras 2018.

CelebA-HQ
1024 × 1024

Progressive growing

2017: Explosion of GANs

Source->Target domain transfer



CycleGAN. Zhu et al. 2017.

Text -> Image Synthesis

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.



Reed et al. 2017.

Many GAN applications



Pix2pix. Isola 2017. Many examples at <https://phillipi.github.io/pix2pix/>

2019: BigGAN



Figure 6: Additional samples generated by our model at 512×512 resolution.

2020: StyleGAN2



What Questions Do You Have?

nwkang@sm.ac.kr

www.smartdesignlab.org

