# Generative Adversarial Network (GAN)

**Namwoo Kang**

Smart Design Lab
CCS Graduate School of Green Transportation
KAIST

# Reference

## ❑ 강의 슬라이드 및 실습코드는 아래의 링크에서 받으실 수 있습니다

- http://www.smartdesignlab.org/dl_aischool_2021.html
- Contributors: 김성신, 유소영, 이성희, 김은지

## ❑ 강의 소스

- Andrew Ng의 ML Class (www.holehouse.org/mlclass/)
- Fei-Fei Li & Justin Johnson & Serena Yeung, CS231n: Convolutional Neural Networks for Visual Recognition, Stanford (http://cs231n.stanford.edu/)
- Stefano Ermon & Aditya Grover, CS 236: Deep Generative Models , Stanford (https://deepgenerativemodels.github.io/)
- 모두를 위한 딥러닝 (https://hunkim.github.io/ml/)
- 모두를 위한 딥러닝 시즌 2 (https://deeplearningzerotoall.github.io/season2/lec_tensorflow.html)
- 이활석, Autoencoders (https://www.slideshare.net/NaverEngineering/ss-96581209)
- 최윤제, 1시간만에 GAN(Generative Adversarial Network) 완전 정복하기 (https://www.slideshare.net/NaverEngineering/1-gangenerative-adversarial-network?qid=c53ce33f-6643-4437-8e93-88776c9cebb1&v=&b=&from_search=5)
- 김성범, [핵심 머신러닝] Principal Component Analysis (PCA, 주성분 분석) (https://youtu.be/FhQm2Tc8Kic)

# Contents

- **Ch1: Introduction to Unsupervised Learning Part I** → Probability & Maximum Likelihood
- **Ch2: Introduction to Unsupervised Learning Part II** → Generative Model & Dimensionality Reduction
- **Ch3: Principal Component Analysis (PCA)** → Machine Learning Model
- **Ch4: Autoencoder & Anomaly Detection** + 실습
- **Ch5: Variational AutoEncoder (VAE)** + 실습
- **Ch6: Generative Adversarial Network (GAN)** + 실습

→ Deep Learning Models

- **Ch7: Application: Mechanical Design + AI** → CAD/CAM/CAE/Design Optimization + AI

# Generative Adversarial Network (GAN) – How to work

VAEs define intractable density function with latent **z**:

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

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

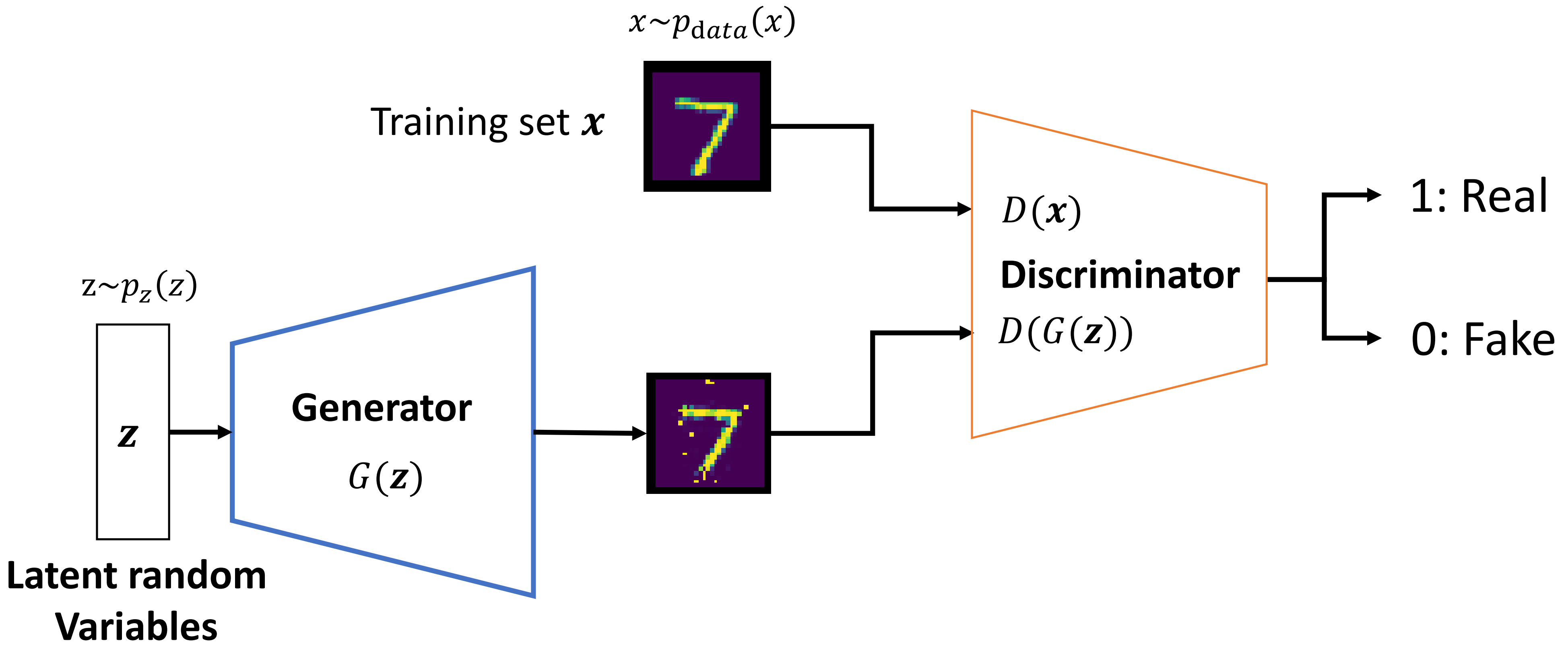
What if we give up on explicitly modeling density, and just want ability to sample?

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)**

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

## 1. Discriminator 학습

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

진짜
이미지

가짜
이미지

$V$를
최대화하는
$D$를 찾는게
목표

$G$는 주어짐
(고정)

진짜학습
데이터
확률분포

$D(x) = 1$
일때 $\log 1$로
최대화

$z$의
확률분포

$D(G(x)) = 0$ 일때
$\log 1$로 최대화

# GAN – How to work

## 2. Generator 학습

가짜 이미지

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

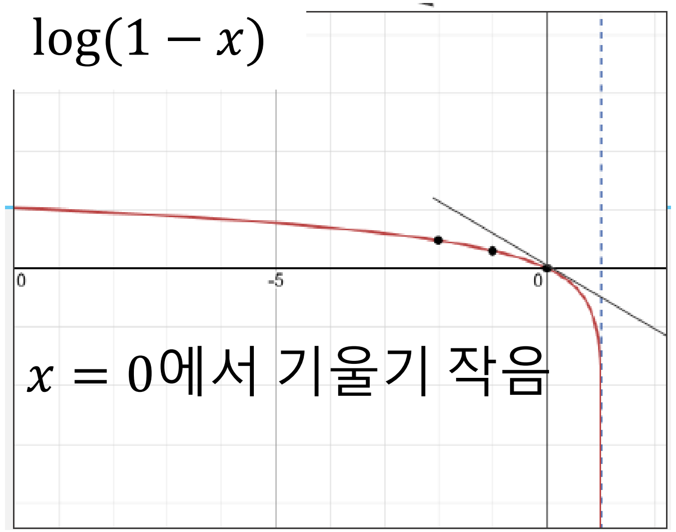
학습하지 않음

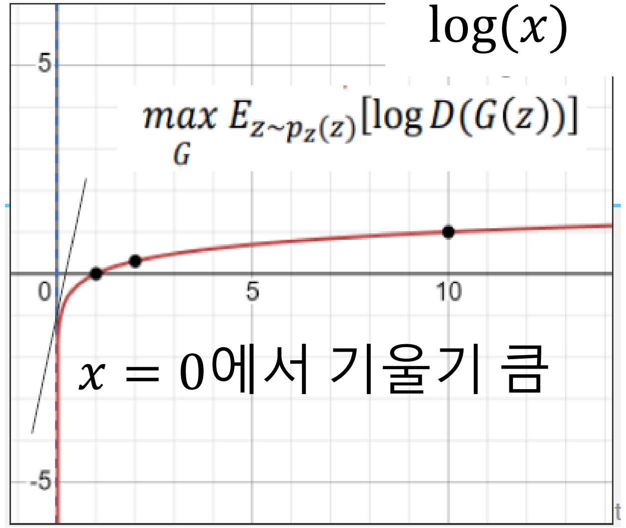
$V$를 최대화하는 $G$를 찾는게 목표

$D$는 주어짐 (고정)

$D(G(x)) = 1$ 일때 $\log 0$ 으로 최소화

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




처음에는 $D(G(z))$가 0에 가까움



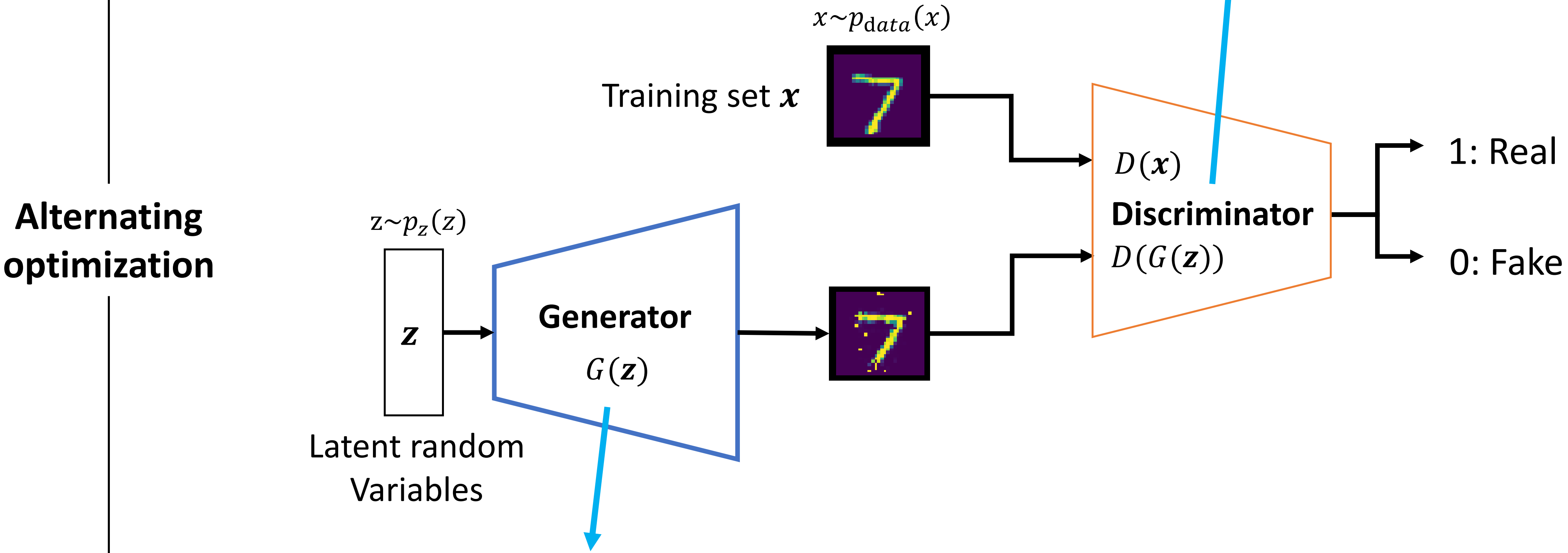
SDL SMART DESIGN LAB KAIST

# GAN – How to work

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

**Maximize prob. of D(real)**

**Minimize prob. of D(fake)**




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

**Maximize prob. of D(fake)**

# GAN - Proof

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

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

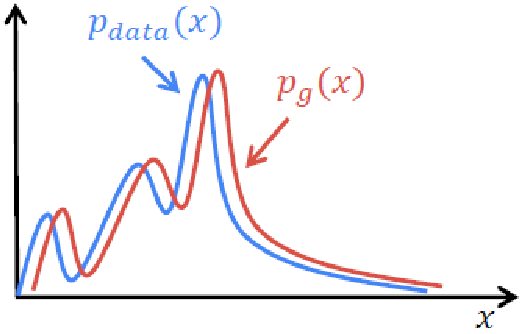
Objective function of GANs

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

Jenson-Shannon divergence

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

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

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

Sampling x from $p_g$ instead of sampling z from $p_z$

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

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

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

Basic property of integral

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

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



SMART DESIGN LAB KAIST

# GAN - Proof

## 1. Discriminator 최적해 증명 (2/2)

$$D^*(x) = arg \max_D V(D)$$

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

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

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

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

G should minimize

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

$$= -log4 + log4 + \int_x p_{data}(x)\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}dx + \int_x p_g(x)\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}dx$$

$$= -log4 + \int_x p_{data}(x)\log\frac{2\cdot p_{data}(x)}{p_{data}(x)+p_g(x)}dx + \int_x p_g(x)\log\frac{2\cdot p_g(x)}{p_{data}(x)+p_g(x)}dx$$

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

$$= -log4 + KL(p_{data}||\frac{p_{data}+p_g}{2}) + KL(p_g||\frac{p_{data}+p_g}{2})$$

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

$$= -log4 + 2\cdot JSD(p_{data}||p_g)$$

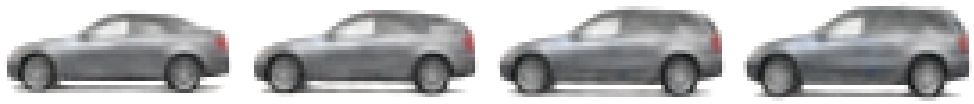
Optimizing V(D,G) is same as minimizing $JSD(p_{data}||p_g)$

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

SDL SMART DESIGN LAB KAIST

# VAE vs. GAN

## VAE



Blurry

Tend to remember input images

Smooth

## GAN

Sharp

Generate new unseen images

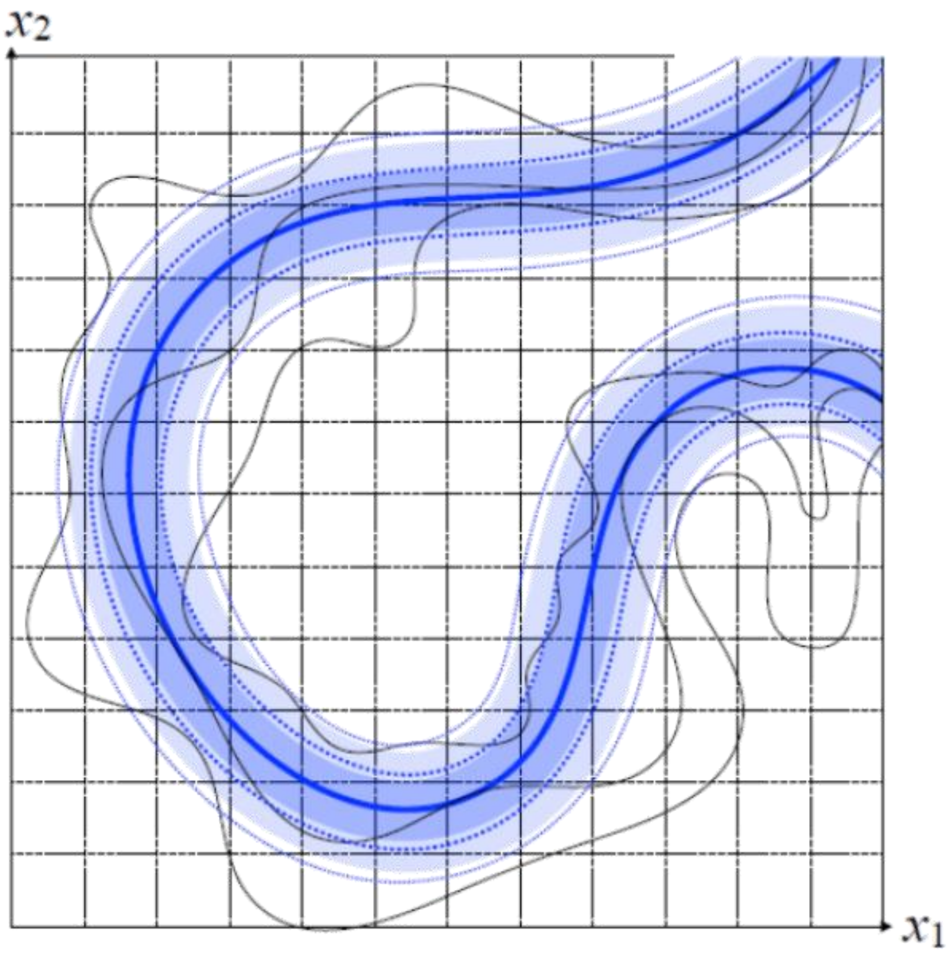
Mode collapse

Unstable convergence

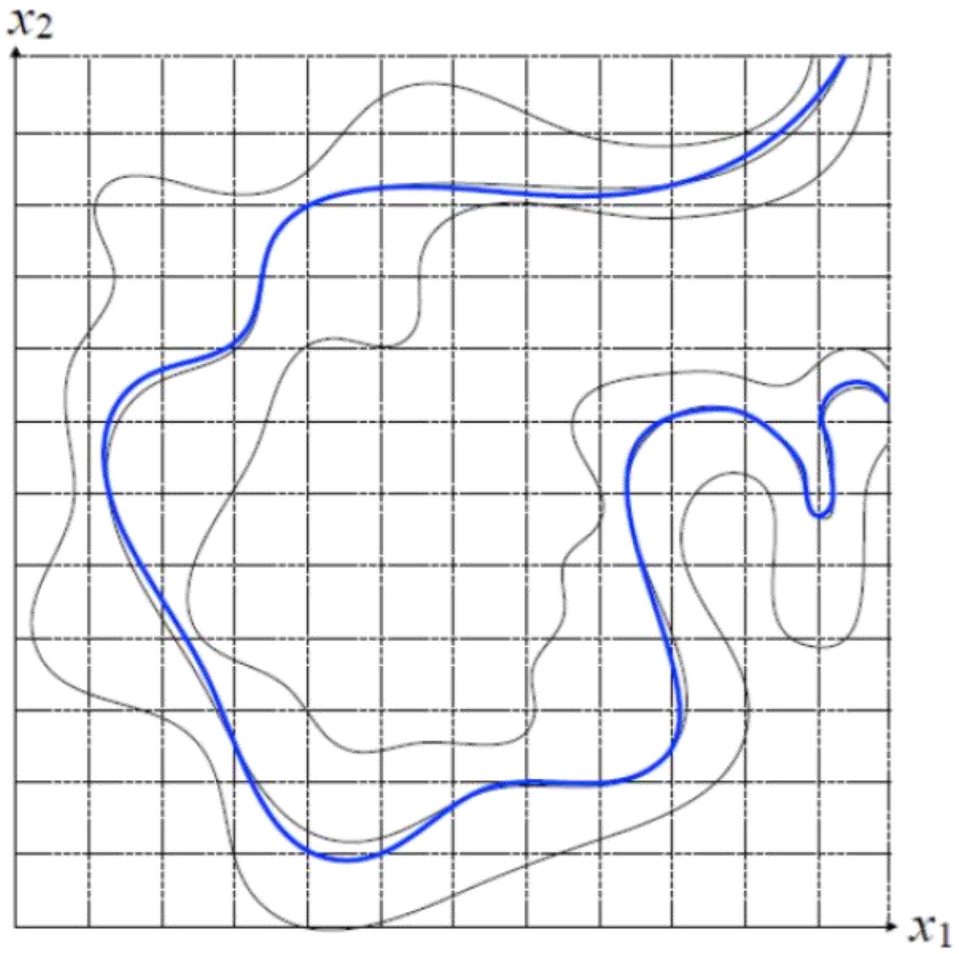


<Burnap, A., Timoshenko, A., and Hauser, J., ``Design and Evaluation of Product Aesthetics: A Human-Machine Hybrid Approach>

# VAE vs. GAN




VAE : maximum likelihood approach

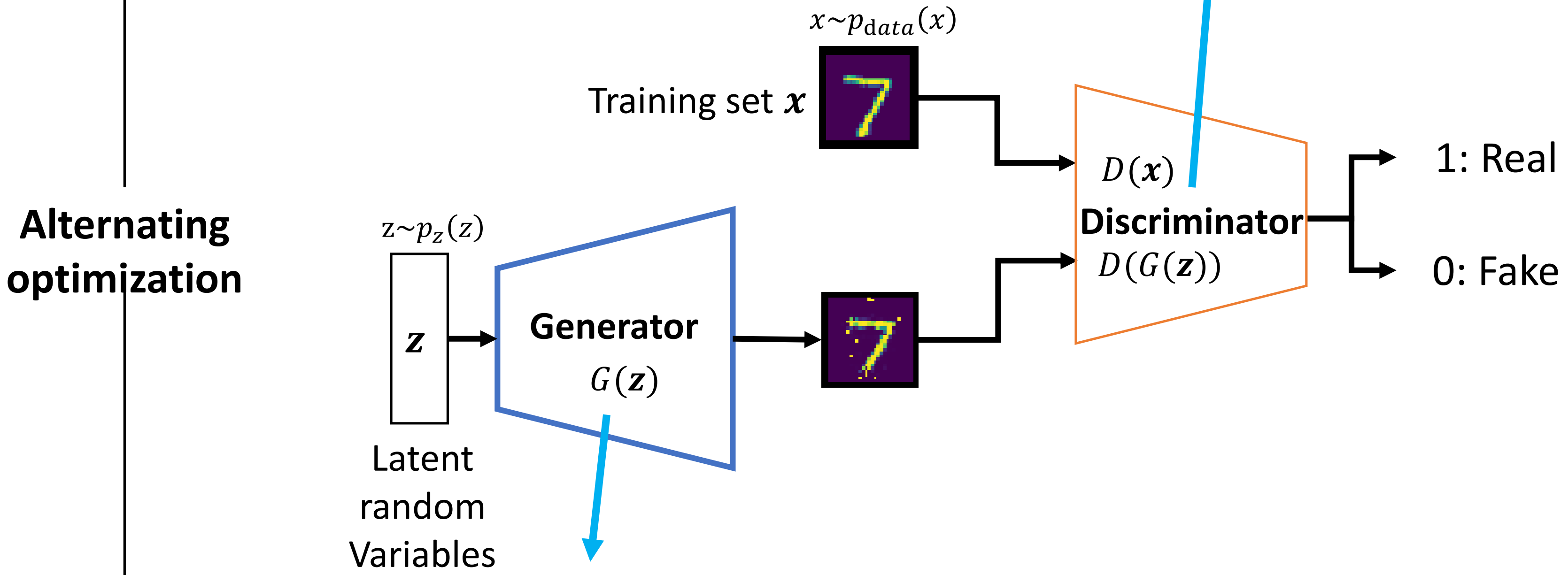



GAN

# GAN Coding

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

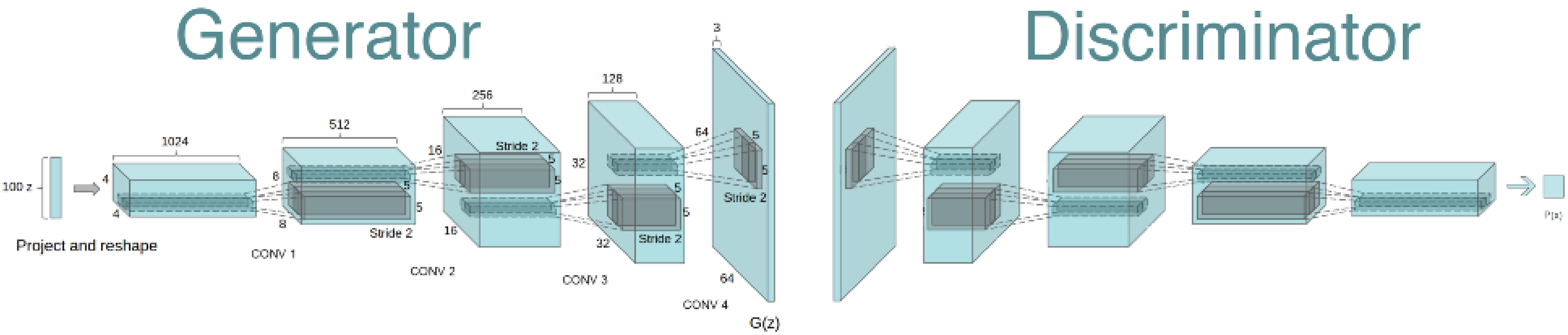
$$\max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z))]$$




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

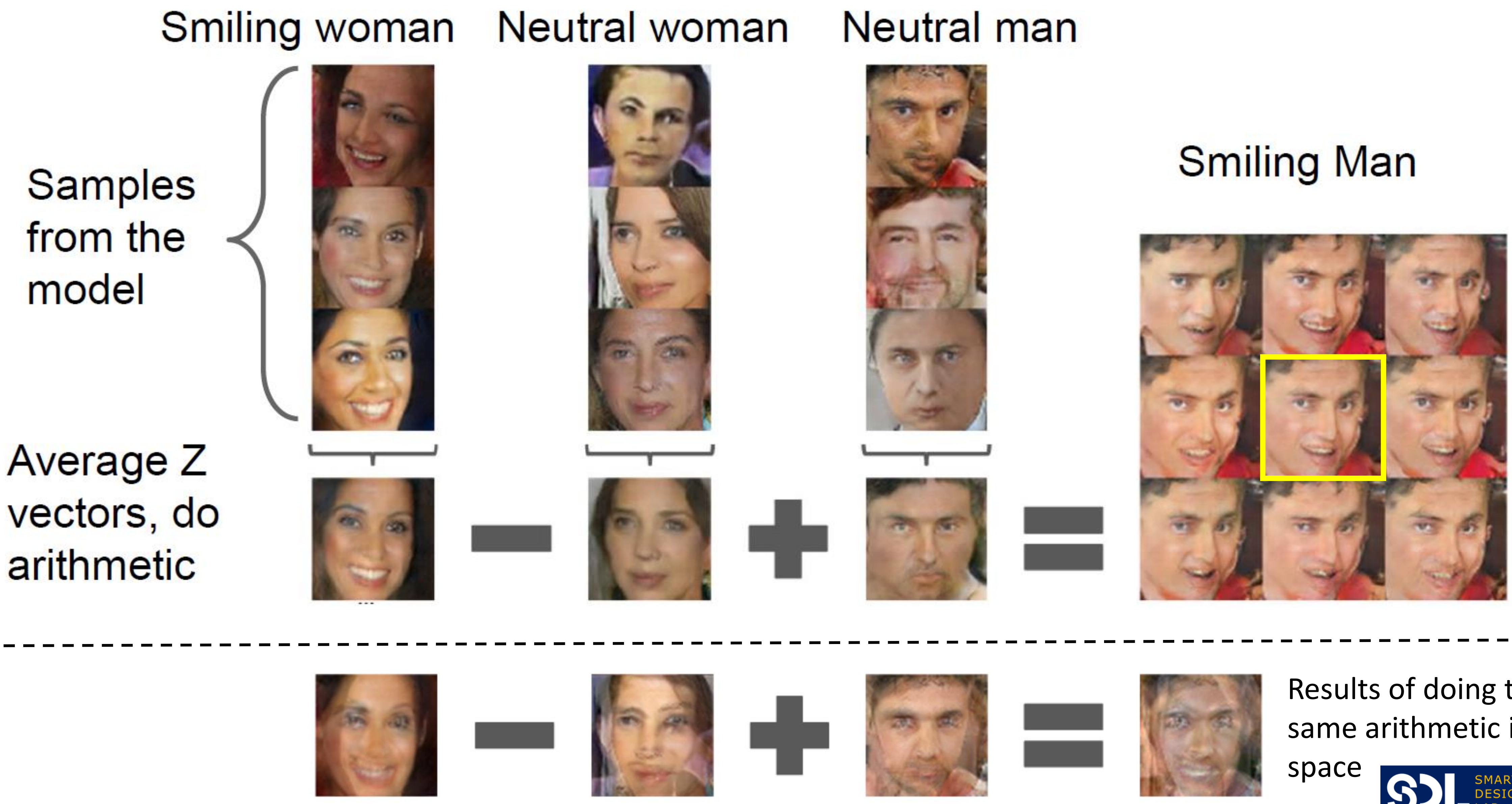
```
loss_G = tf.reduce_mean(tf.log(D_gene))
```

# Deep Convolutional GAN (DCGAN)



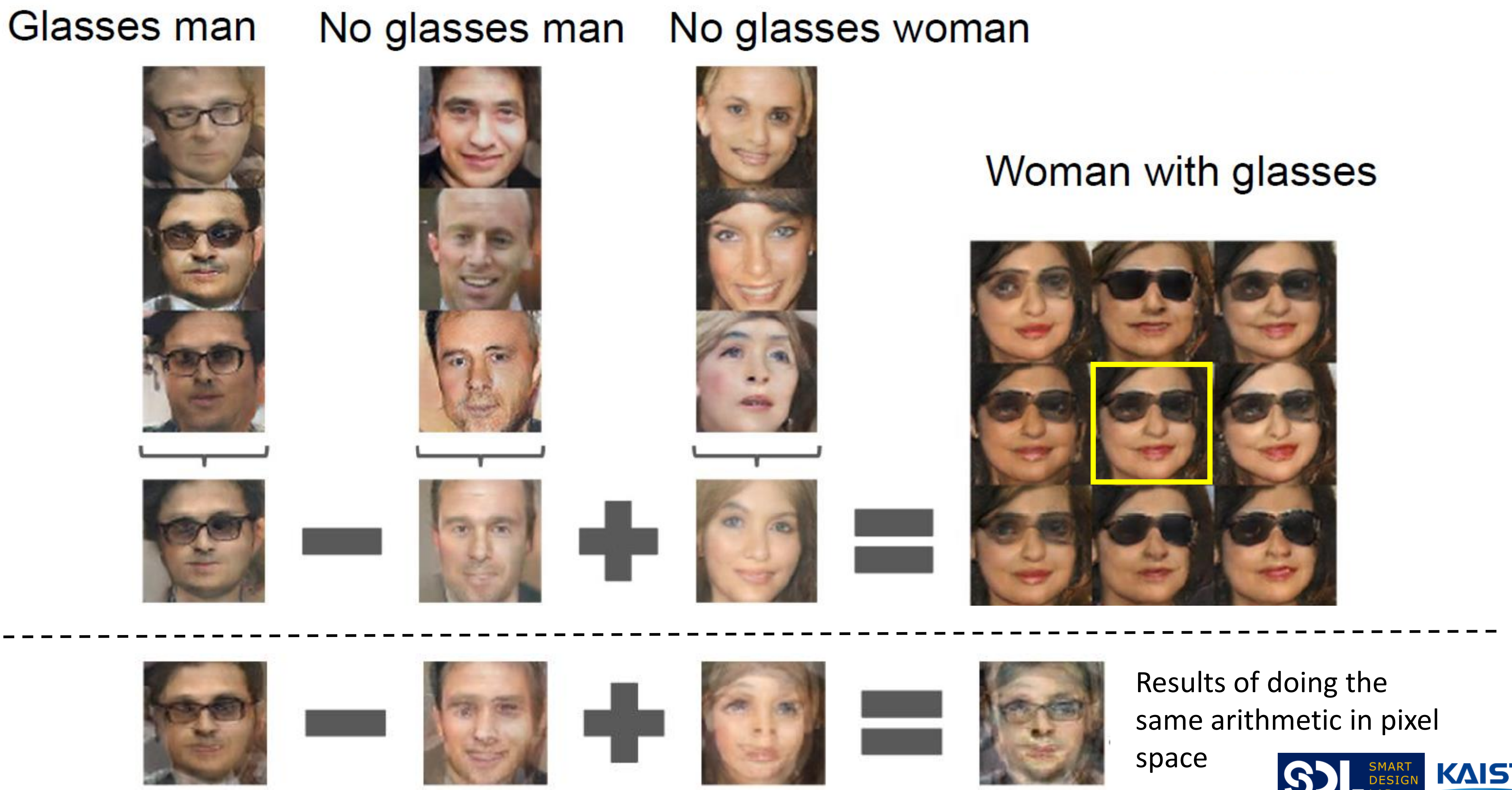

| | Generator | Discriminator |
|---|---|---|
| Pooling Layers | Not used. But use stride convolutions instead | |
| Batch normalization | Use except output layer | Use except input layer |
| Fully connected hidden layers | Not used | |
| Activation function | ReLU for all layers except for the output,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

# DCGAN: Interpretable Vector Math



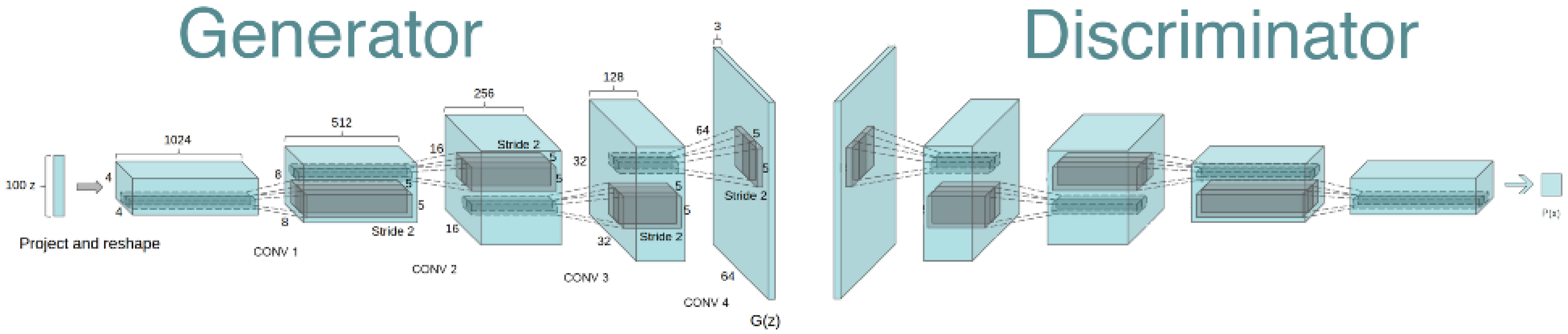

Results of doing the same arithmetic in pixel space

# DCGAN: Interpretable Vector Math



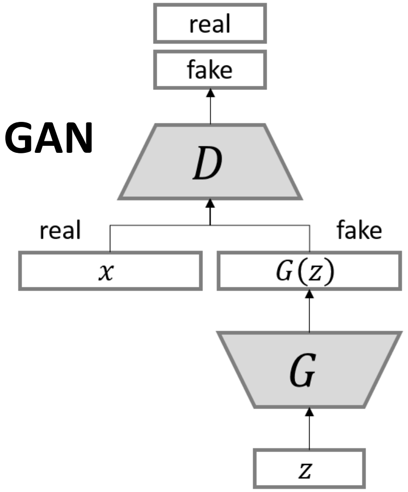

Results of doing the same arithmetic in pixel space

# DCGAN Coding

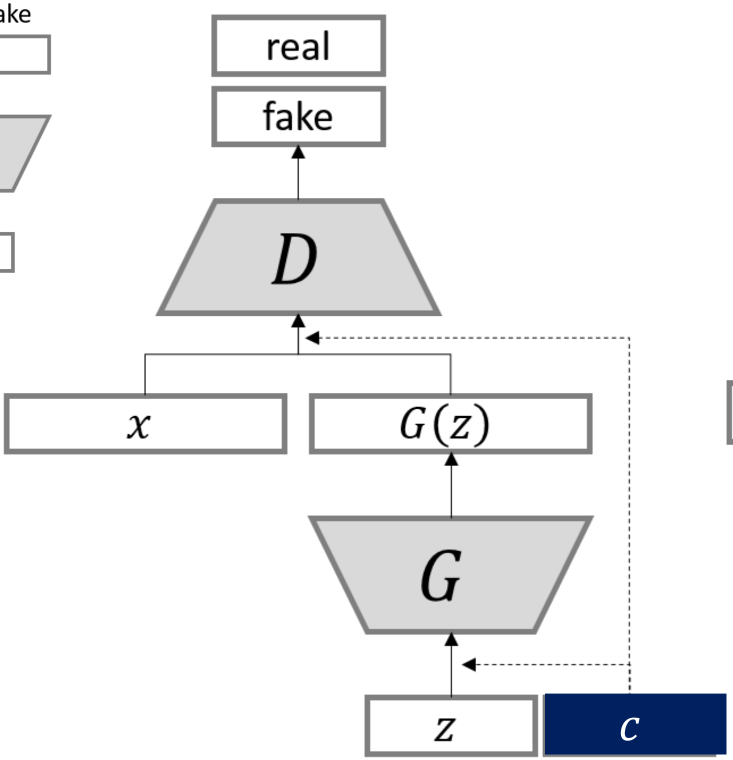


# Various GAN Structures

## GAN

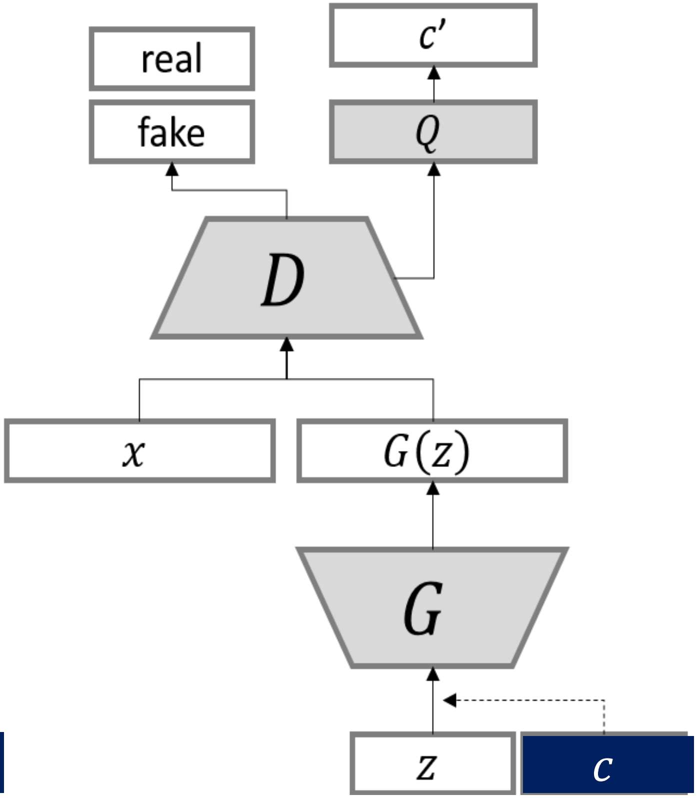



## cGAN



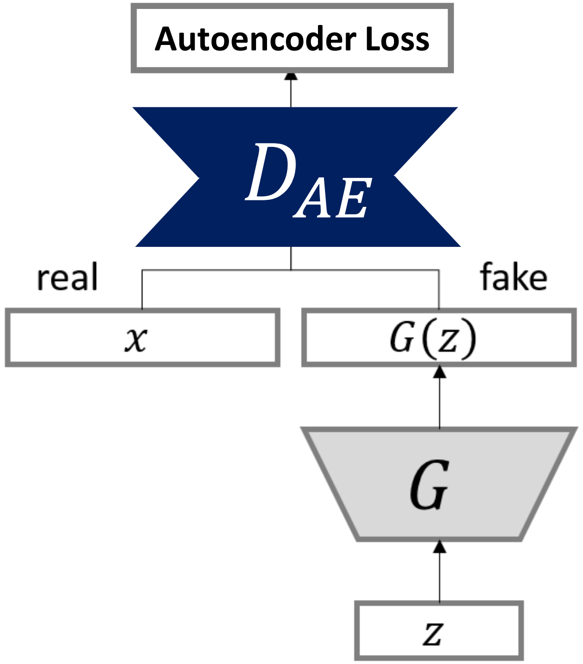

(Mirza & Osindero, 2014)

## infoGAN




(Chen, et al., 2016)

## BEGAN

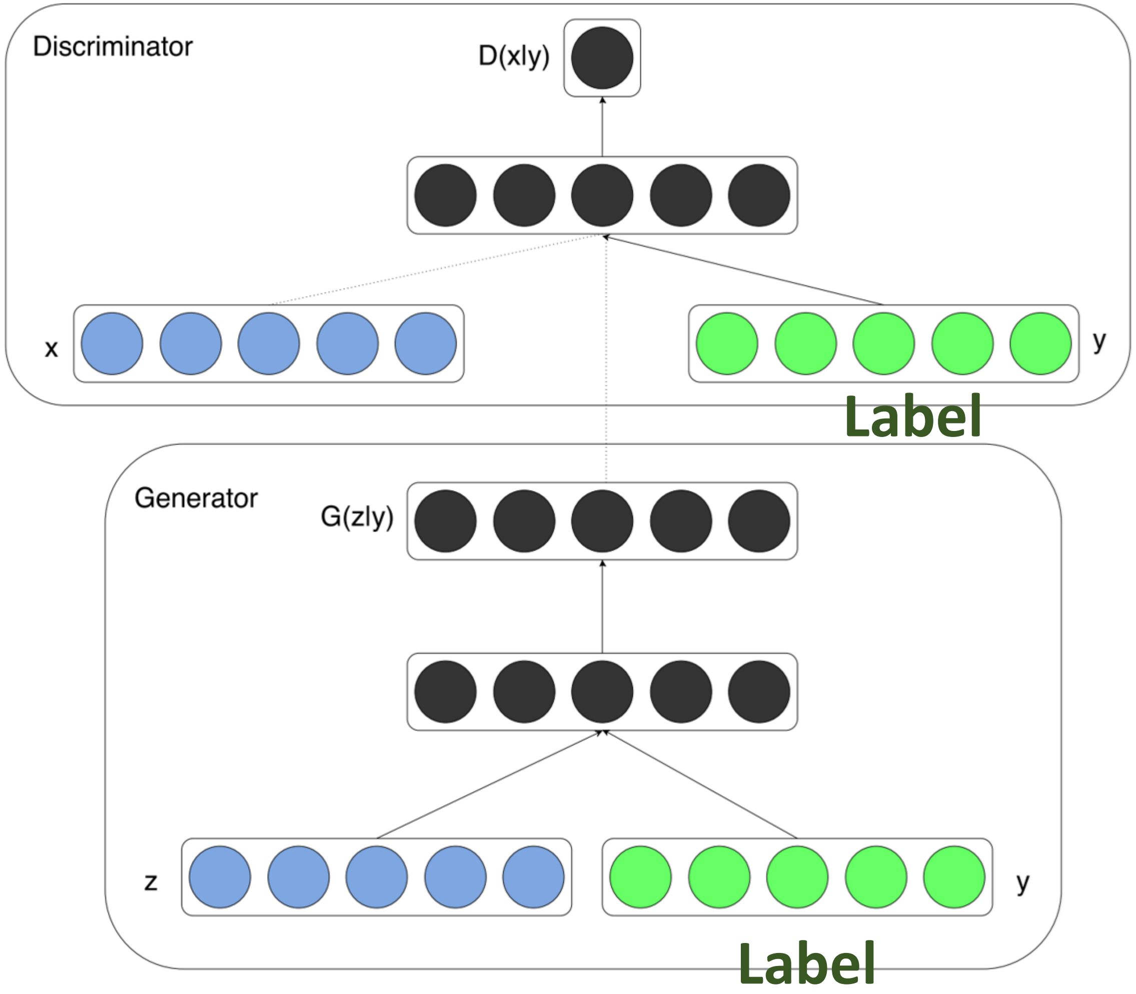



(Berthelot et al., 2017)

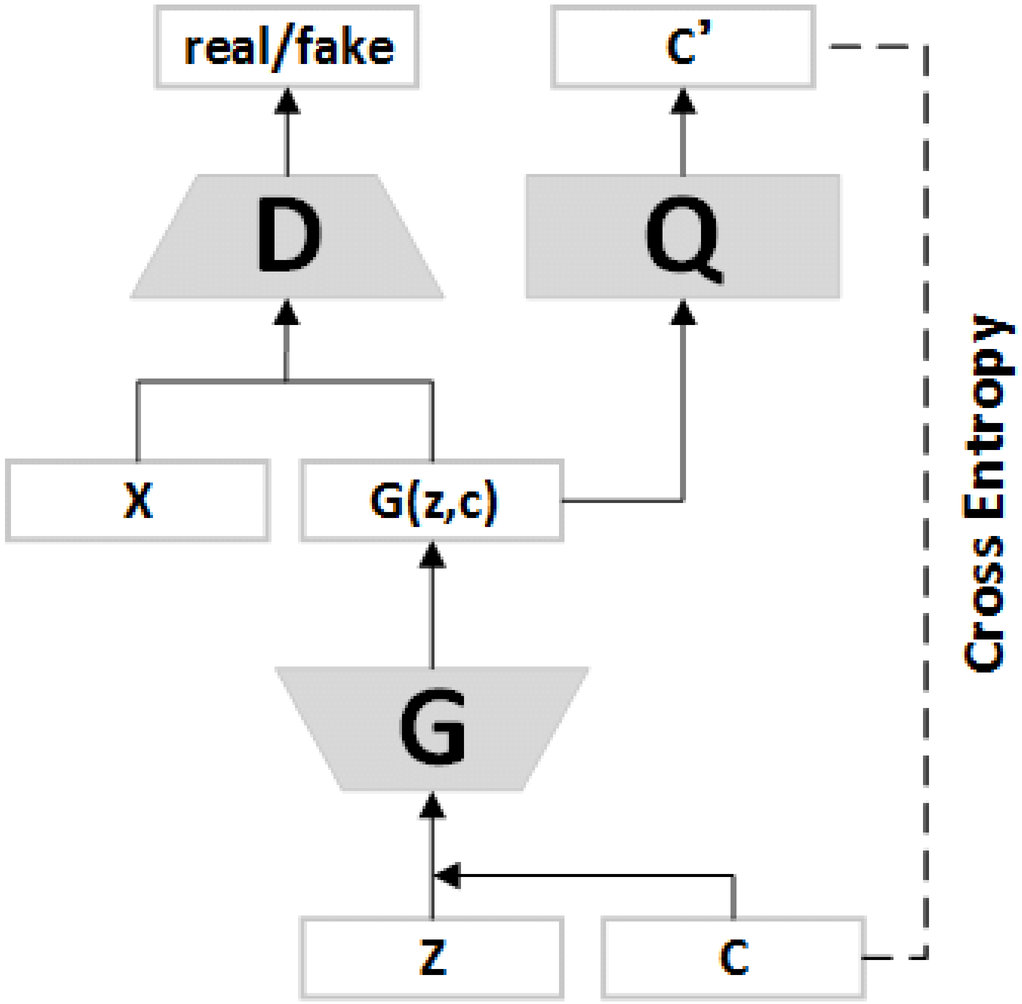


<https://github.com/hwalsuklee/tensorflow-generative-model-collections>

# Conditional GAN (cGAN)

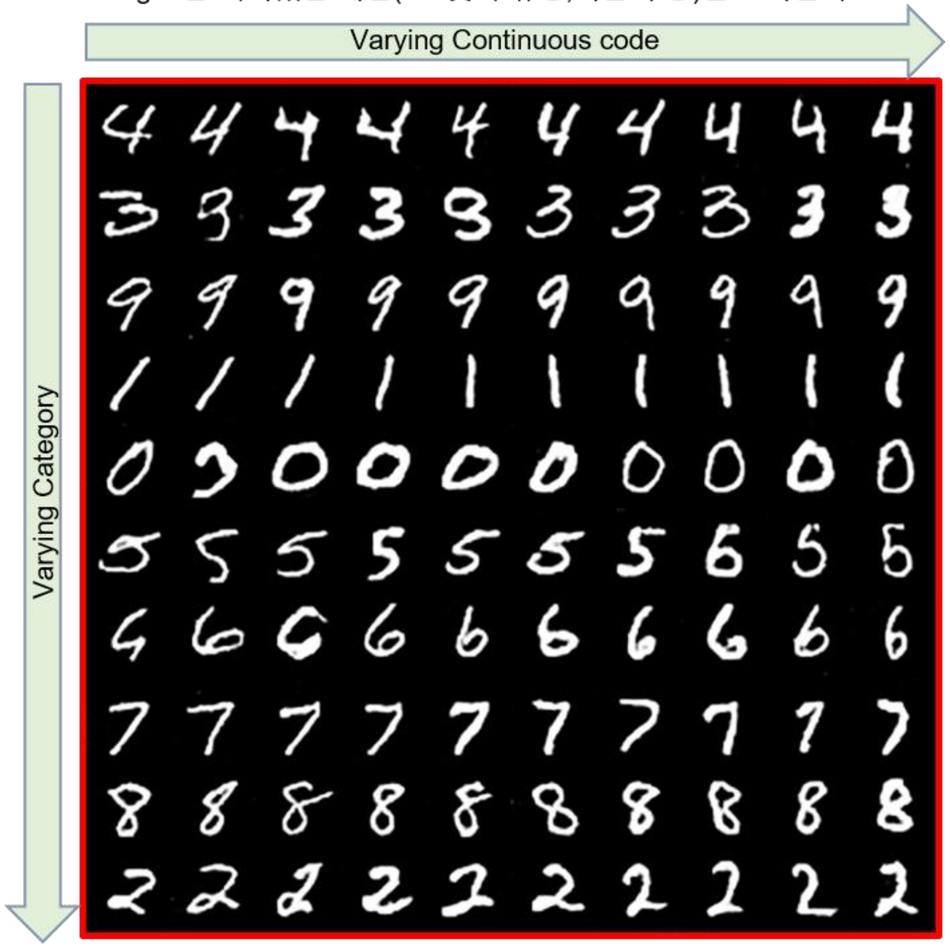



# InfoGAN




**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

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

**Discriminator:** $\mathcal{L}_D = \boxed{\mathcal{L}(x)} - k_t \boxed{\mathcal{L}(G(z_D))} \quad \text{for} \quad \theta_D$

진짜가
복원이 잘되도록

가짜가
복원이 안되도록

**Generator:** $\mathcal{L}_G = \boxed{\mathcal{L}(G(z_G))} \quad \text{for} \quad \theta_G$

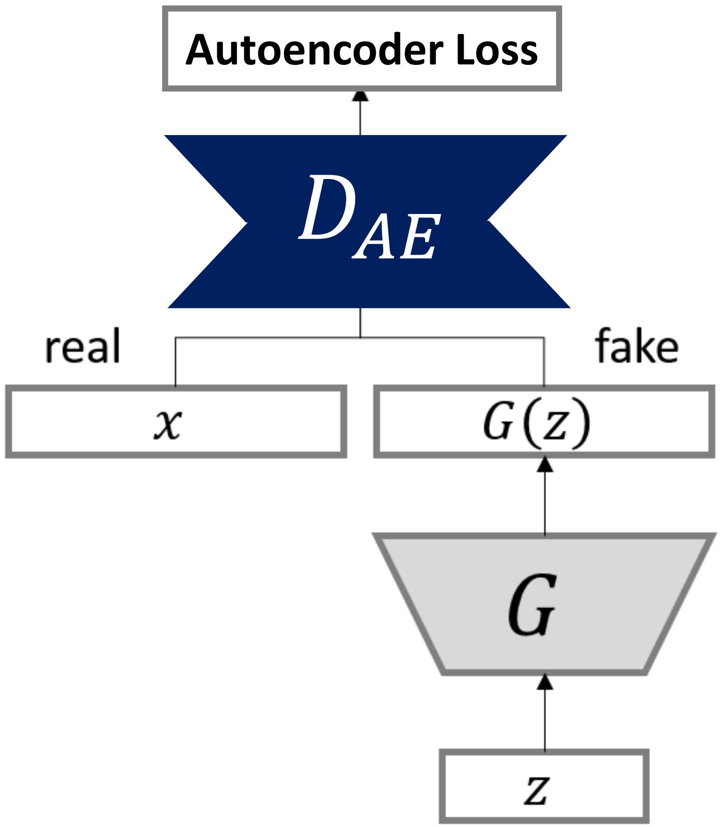
가짜가
복원이 잘되도록

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)

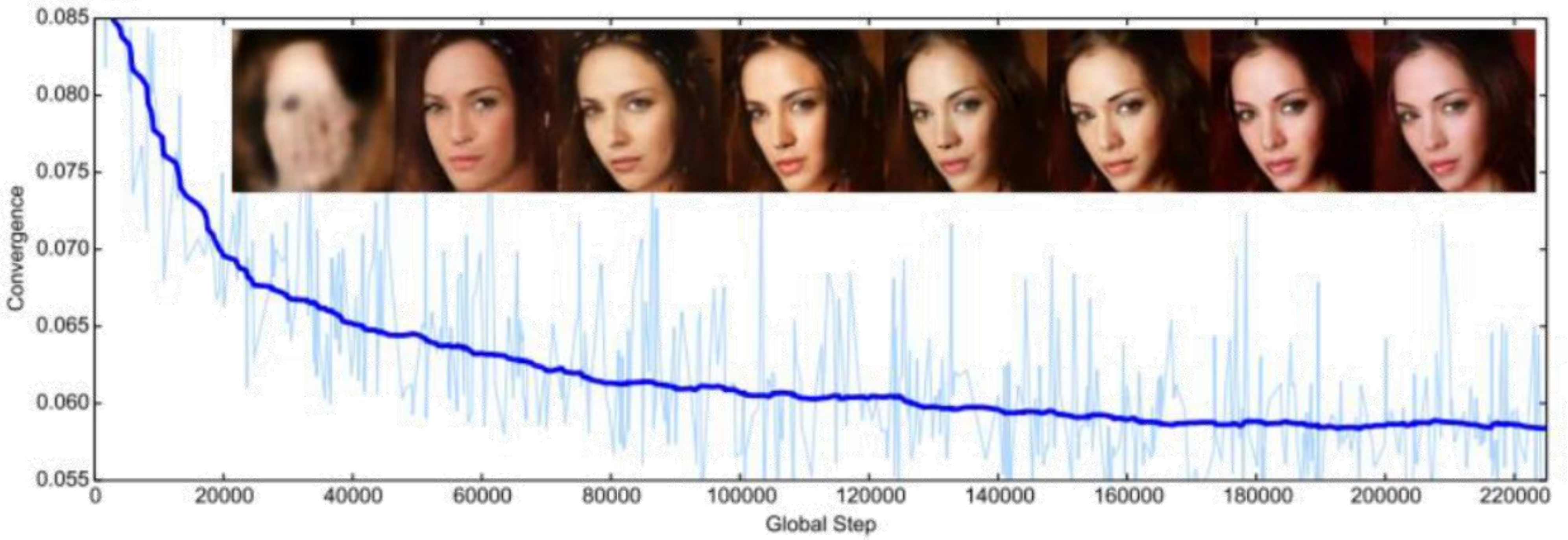
$\gamma$ ⬇ ➔ $L(x)$ 에 집중 ➔ $G(z)$ 다양성 떨어짐, 퀄리티 증가

$\gamma$ ⬆ ➔ $L(G(z))$ 에 집중 ➔ $G(z)$ 다양성 증가, 퀄리티 감소

# Boundary Equilibrium GAN (BEGAN)

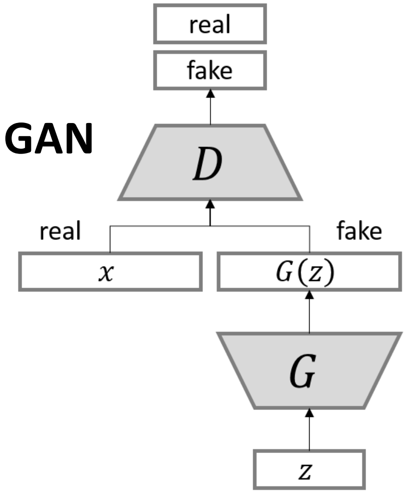
***Convergence Measure***

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

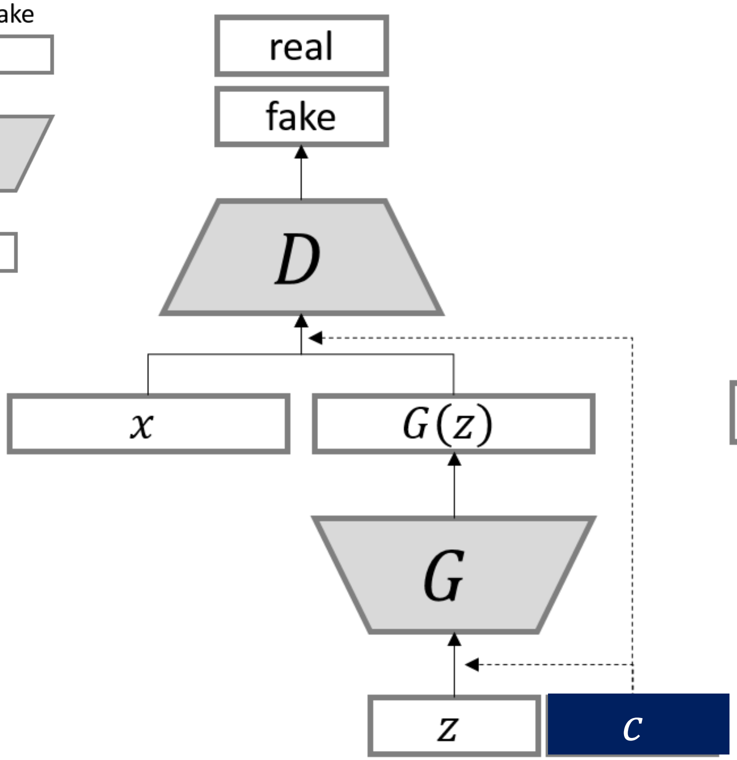


# Various GANs Coding

GAN

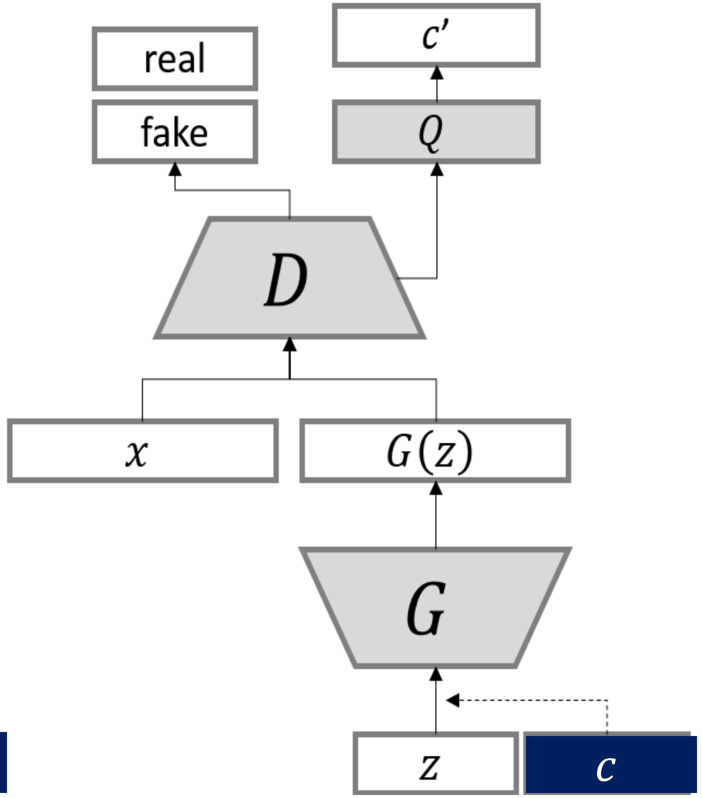



cGAN



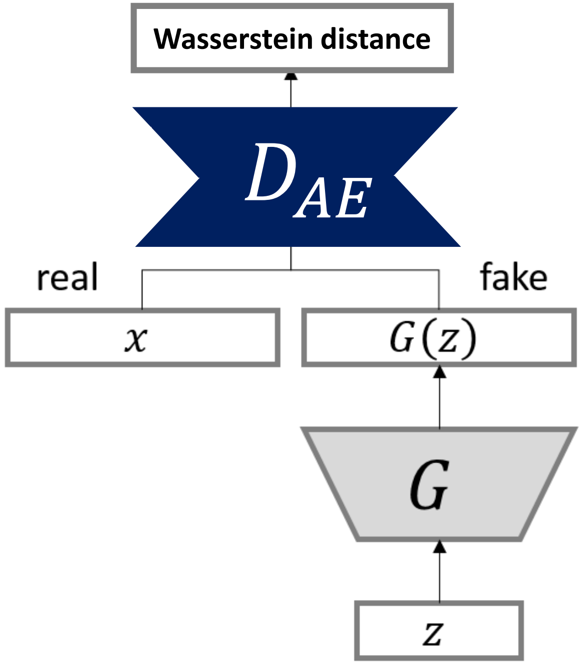

(Mirza & Osindero, 2014)

infoGAN




(Chen, et al., 2016)

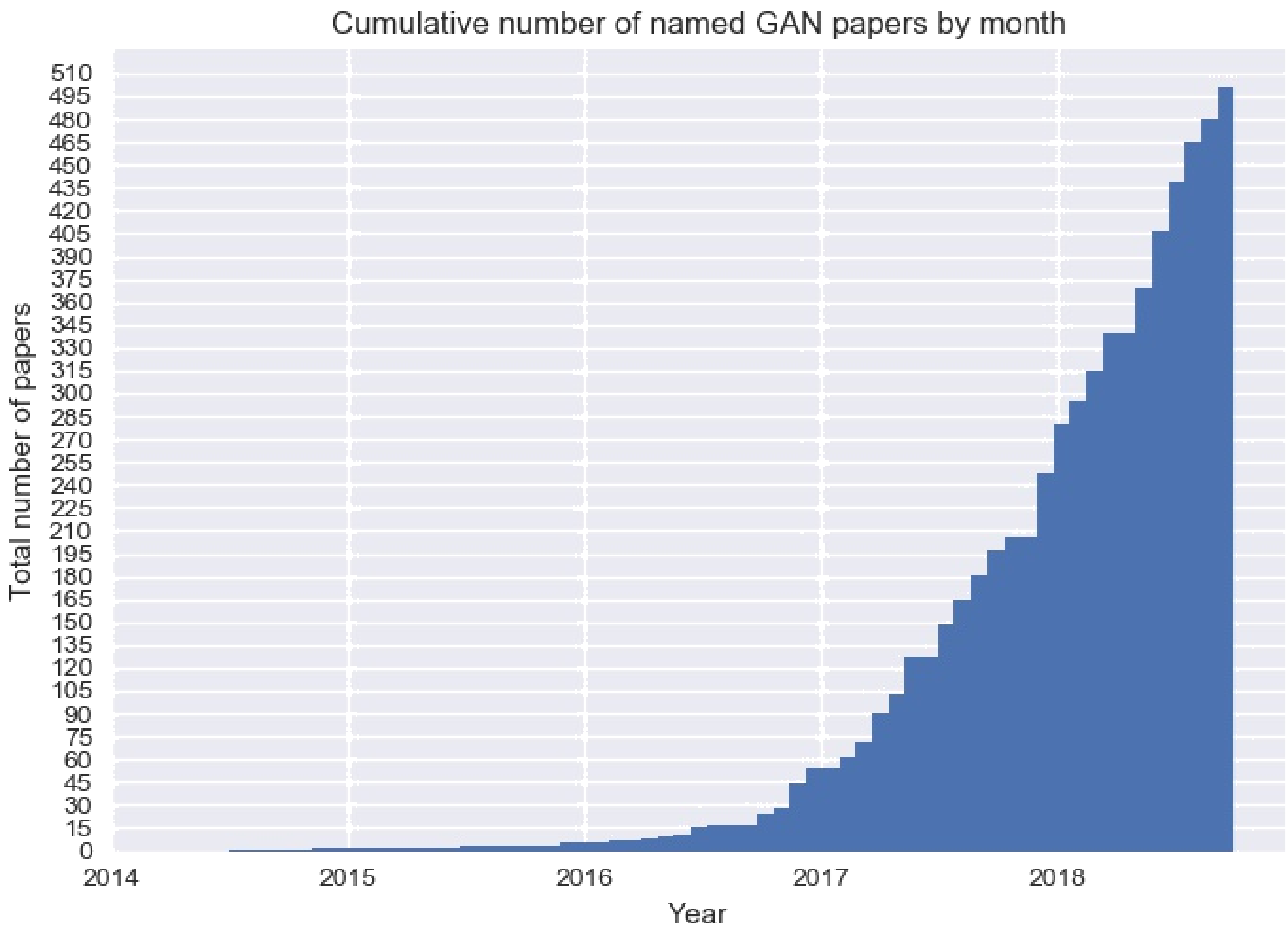
BEGAN




(Berthelot et al., 2017)


<https://github.com/hwalsuklee/tensorflow-generative-model-collections>

# GAN ZOO




GAN
DCGAN
WGAN
CGAN
LSGAN
SGAN
ACGAN
InfoGAN
CycleGAN

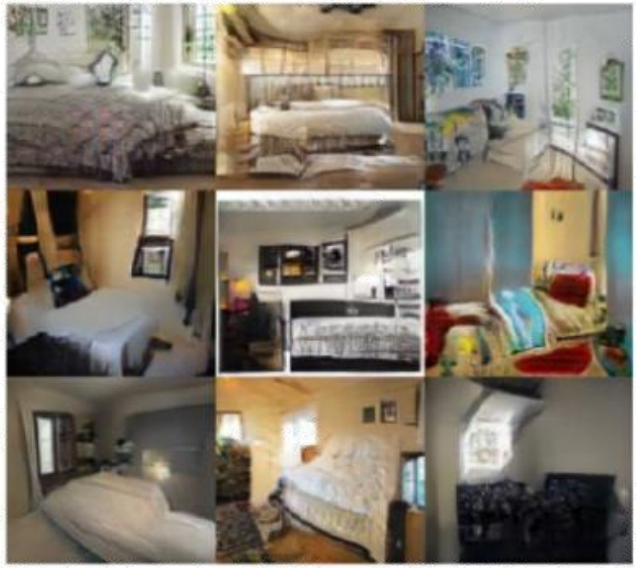
⋮



<https://github.com/hindupuravinash/the-gan-zoo>

# 2017: Explosion of GANs

## Better training and generation



LSGAN, Zhu 2017.

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

Progressive GAN, Karras 2018.

SMART DESIGN LAB KAIST

# CelebA-HQ

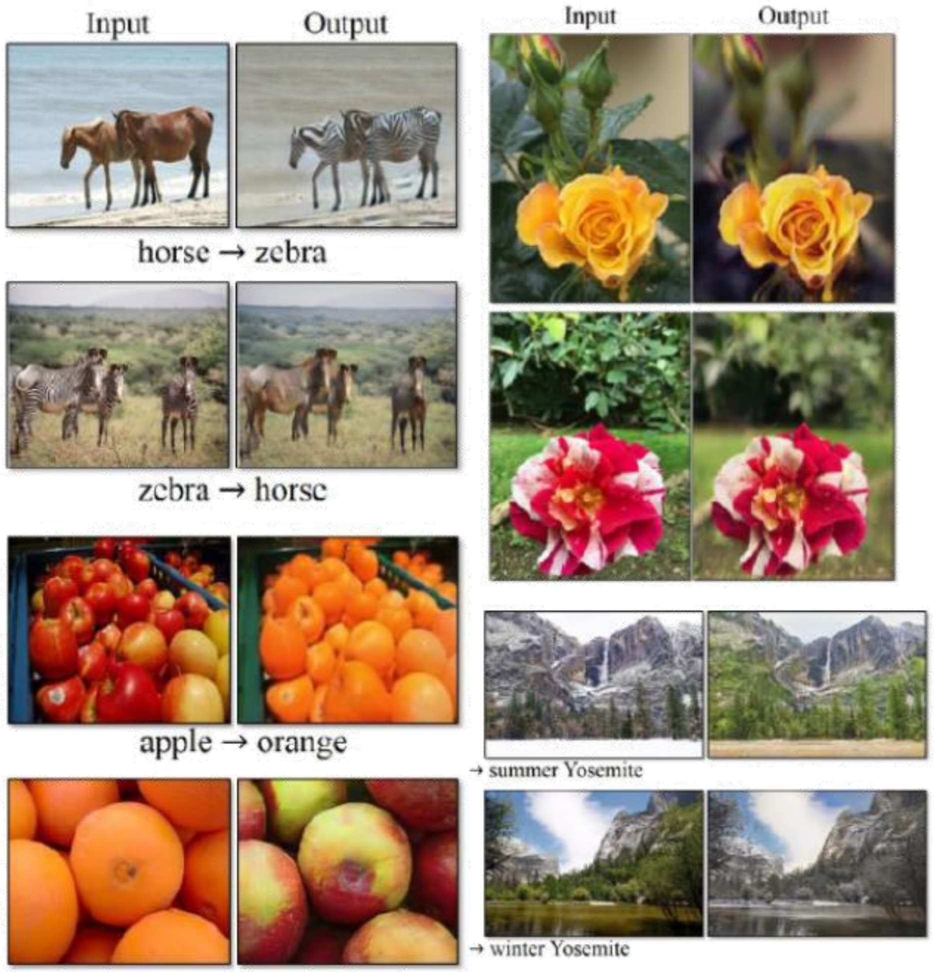
1024 × 1024

# Progressive growing

<Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.>

# 2017: Explosion of GANs

## Source->Target domain transfer




CycleGAN. Zhu et al. 2017.

## Text -> Image Synthesis


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.

Brock et al., 2019

# 2020: StyleGAN2

Karras et al., 2020



https://thispersondoesnotexist.com/

# What Questions Do You Have?

nwkang@kaist.ac.kr

www.smartdesignlab.org

SMART DESIGN LAB KAIST