이미지 생성 AI

구름

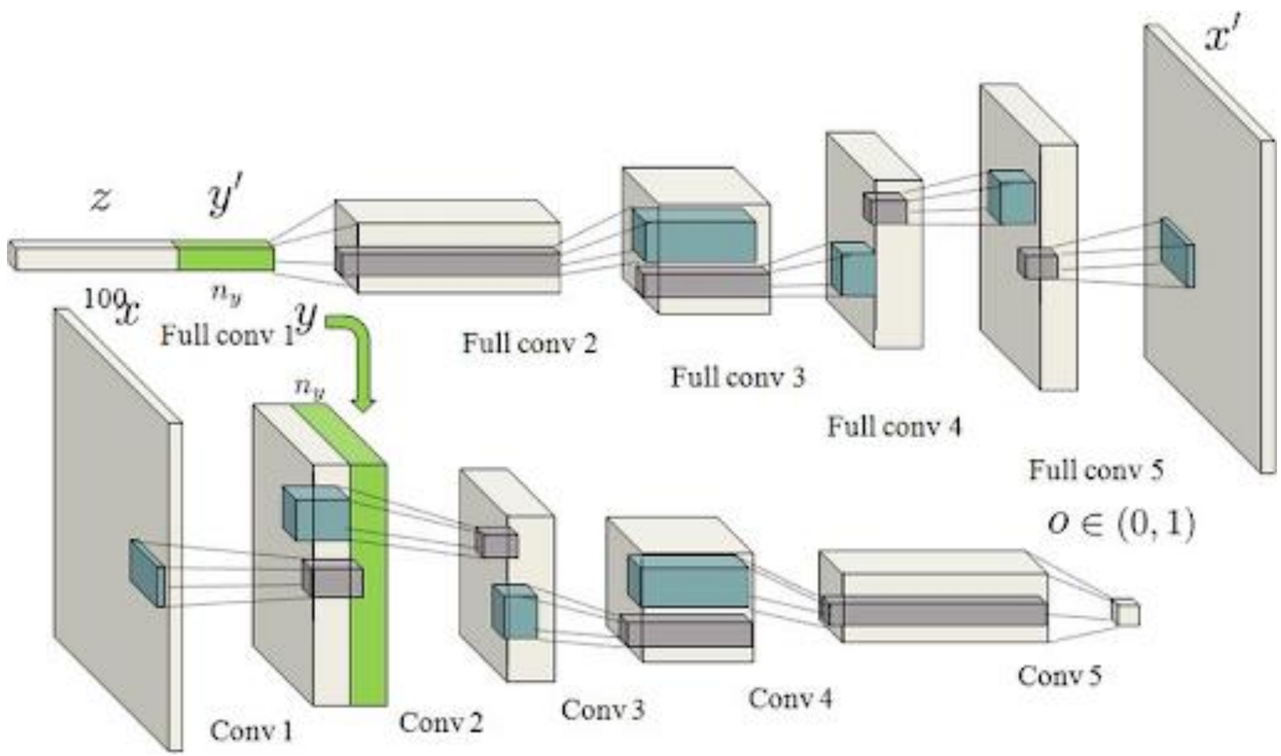
빅밸류

1. **GAN**
2. Autoencoder
3. Diffusion

Generative Adversarial Nets (GAN)

https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

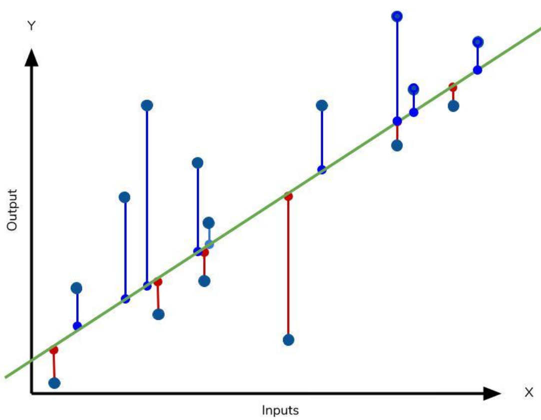
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$



Loss Function

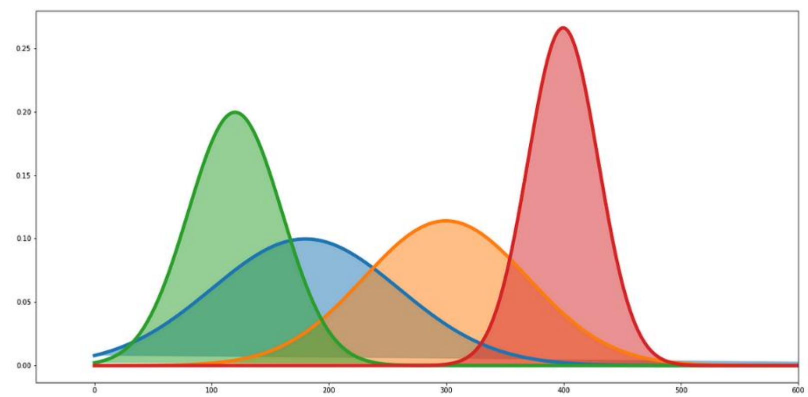
MSE

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$



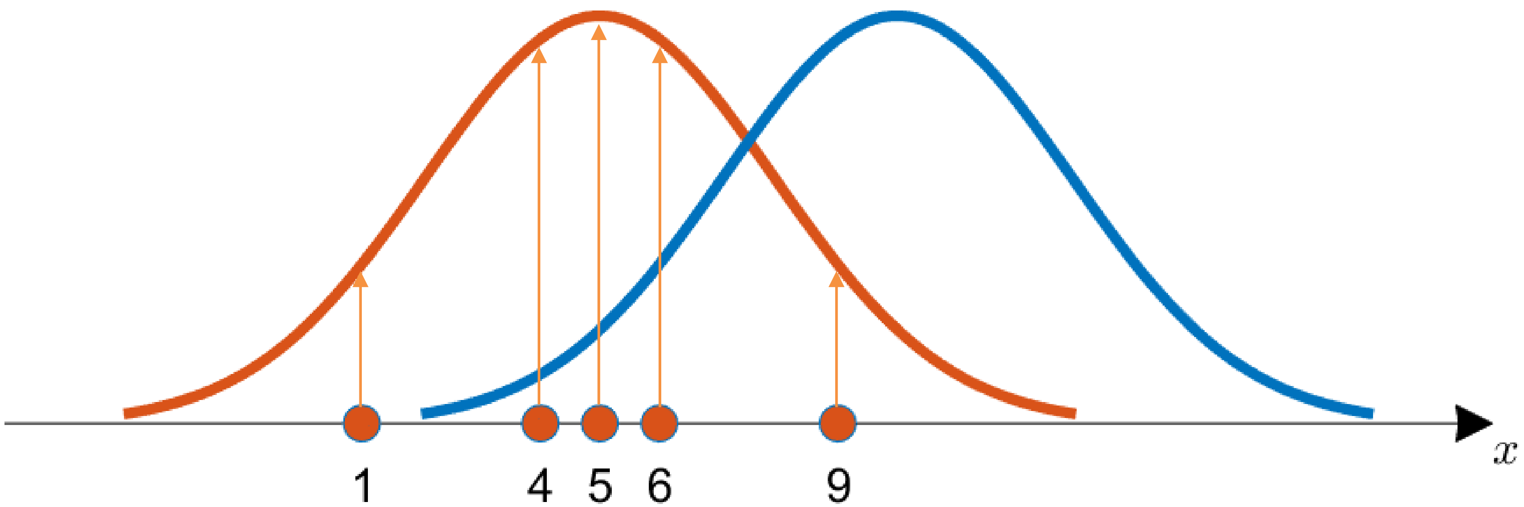
MLE

$$L(\theta) = p(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta)$$



Maximum Likelihood

https://angeloyeo.github.io/2020/07/17/MLE.html



Likelihood

정보이론 (information theory)

저장공간 :

0	1	1	0	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	0

코드 :

0 : A	1 : B	10 : C	11 : D	100 : E	101 : F	110 : G	111 : H	1000 : I

데이터 :

EGG	100110110
BED	110011
HI	1111000
HEAD	111100011

발생확률

코드	A	B	C	D	E	F	G	H	I	합계
빈도	1	1	0	2	3	0	2	2	1	12
확률	8%	8%	0%	17%	25%	0%	17%	17%	8%	
길이	1	1	2	2	3	3	3	3	4	
총길이	0.08	0.08	0	0.33	0.75	0	0.5	0.5	0.33	2.58

코드 :

0 : A	1 : B	10 : C	11 : D	100 : E	101 : F	110 : G	111 : H	1000 : I

확률 :

8%	8%	0%	17%	25%	0%	17%	17%	8%

변환 :

100 : A	101 : B	111 : C	1 : D	0 : E	1000 : F	10 : G	11 : H	110 : I

데이터 :

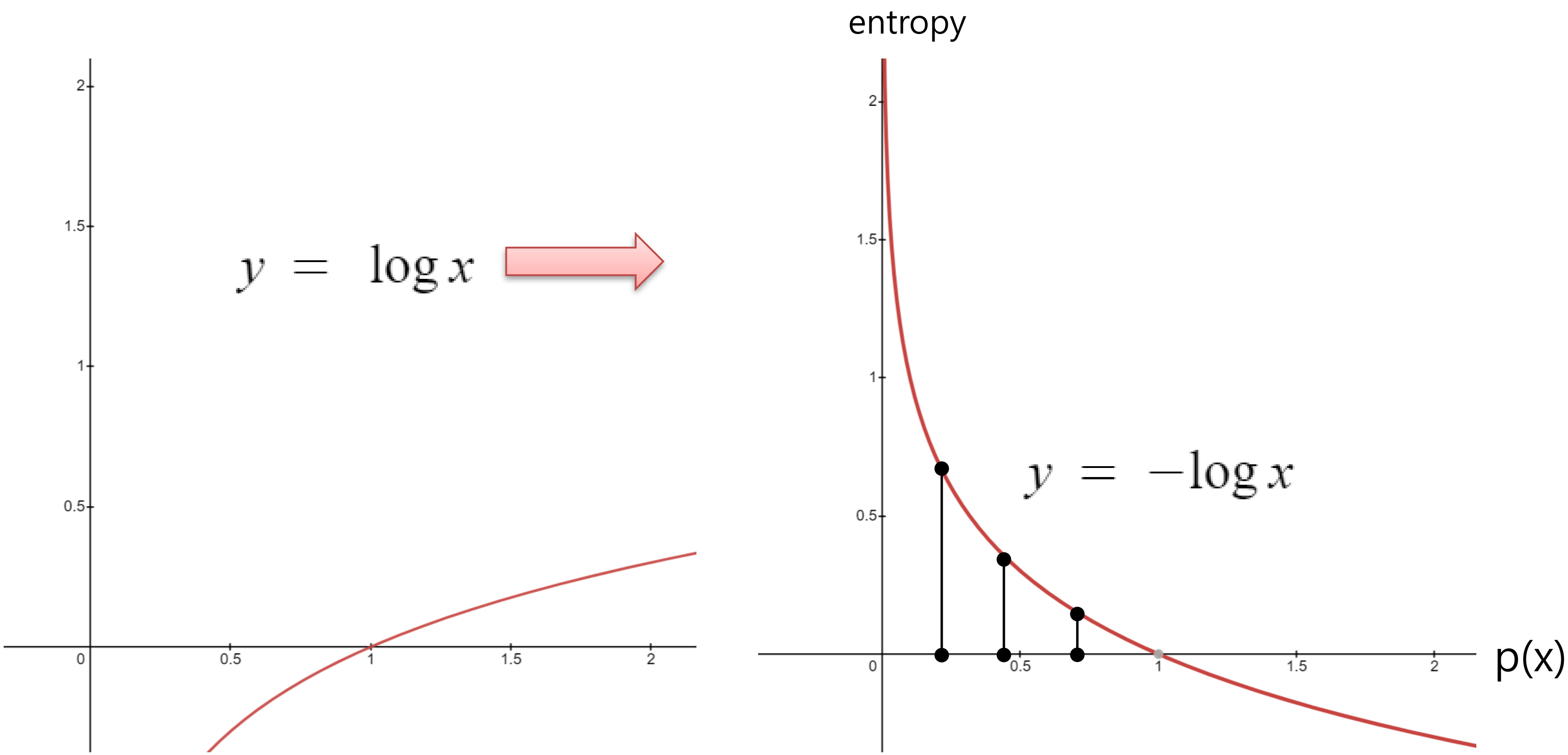
EGG	100110110	01010
BED	110011	10101
HI	1111000	11110
HEAD	111100011	1101001

발생확률

코드	A	B	C	D	E	F	G	H	I	합계
빈도	1	1	0	2	3	0	2	2	1	12
확률	8%	8%	0%	17%	25%	0%	17%	17%	8%	
길이	3	3	3	1	1	4	2	2	3	
총길이	0.24	0.24	0	0.17	0.25	0	0.34	0.34	0.24	1.82
이전	0.08	0.08	0	0.33	0.75	0	0.5	0.5	0.33	2.58

엔트로피 (Entropy)

$$H(x) = -\sum_{i=1}^{n} p(x_i) log p(x_i)$$



Cross Entropy

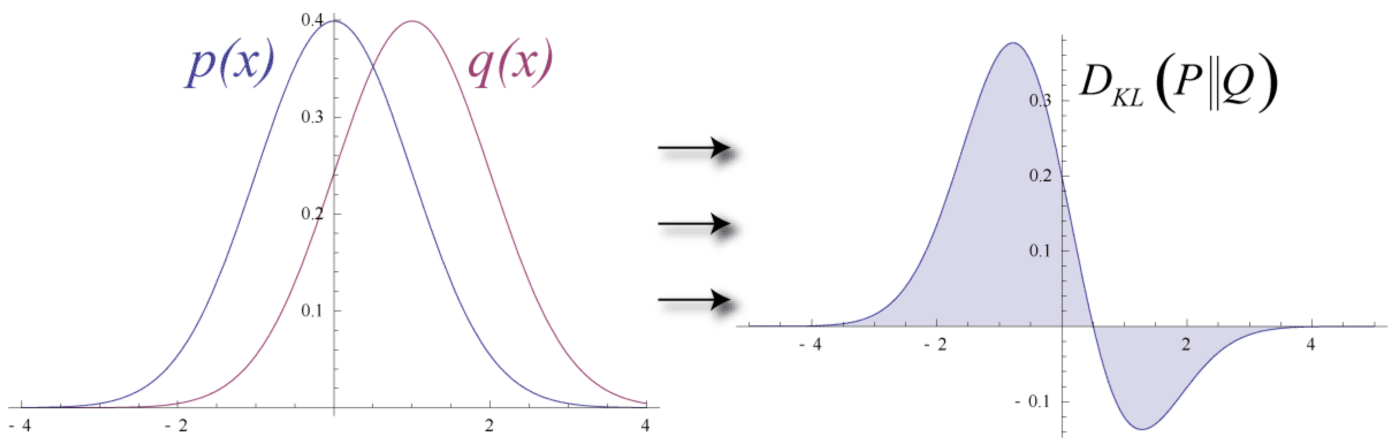
$$H(P^*|P) = -\sum_i \underbrace{P^*(i)}_{\text{TRUE CLASS DISTIRBUTION}} \log \underbrace{P(i)}_{\text{PREDICTED CLASS DISTIRBUTION}}$$

	P*(x)	-LOG(p*(x))	entropy	P(X)	-LOG(p(x))	cross-entropy
A	10%	1.00	0.10	80%	0.10	0.01
B	50%	0.30	0.15	45%	0.35	0.17
C	12%	0.92	0.11	70%	0.15	0.02
D	5%	1.30	0.07	90%	0.05	0.00
E	1%	2.00	0.02	99%	0.00	0.00
F	90%	0.05	0.04	10%	1.00	0.90
G	70%	0.15	0.11	30%	0.52	0.37
			0.60			1.47

KL Divergence (Kullback-Leibler divergence)

$$D_{KL}(P \parallel Q) = \sum_{i=0}^{n} p(x_i)log(p(x_i)) - \sum_{i=0}^{n} p(x_i)log(q(x_i))$$

entropy Cross Entropy



Original Gaussian PDF's KL Area to be Integrated

JS Divergence (Jenson-Shannon divergence)

$$JSD(p\|q) = \frac{1}{2}KL(p\|M) + \frac{1}{2}KL(q\|M)$$
$$where, M = \frac{1}{2}(p+q)$$

Generative Adversarial Nets (GAN)

https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

판별 함수에 원본 이미지가 들어가면 1이 나오면 커짐

임의의 z 를 생성자에 넣어서 나온 생성 이미지가 판별함수에서 0이 나오면 커짐

D 최대

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

G 최소

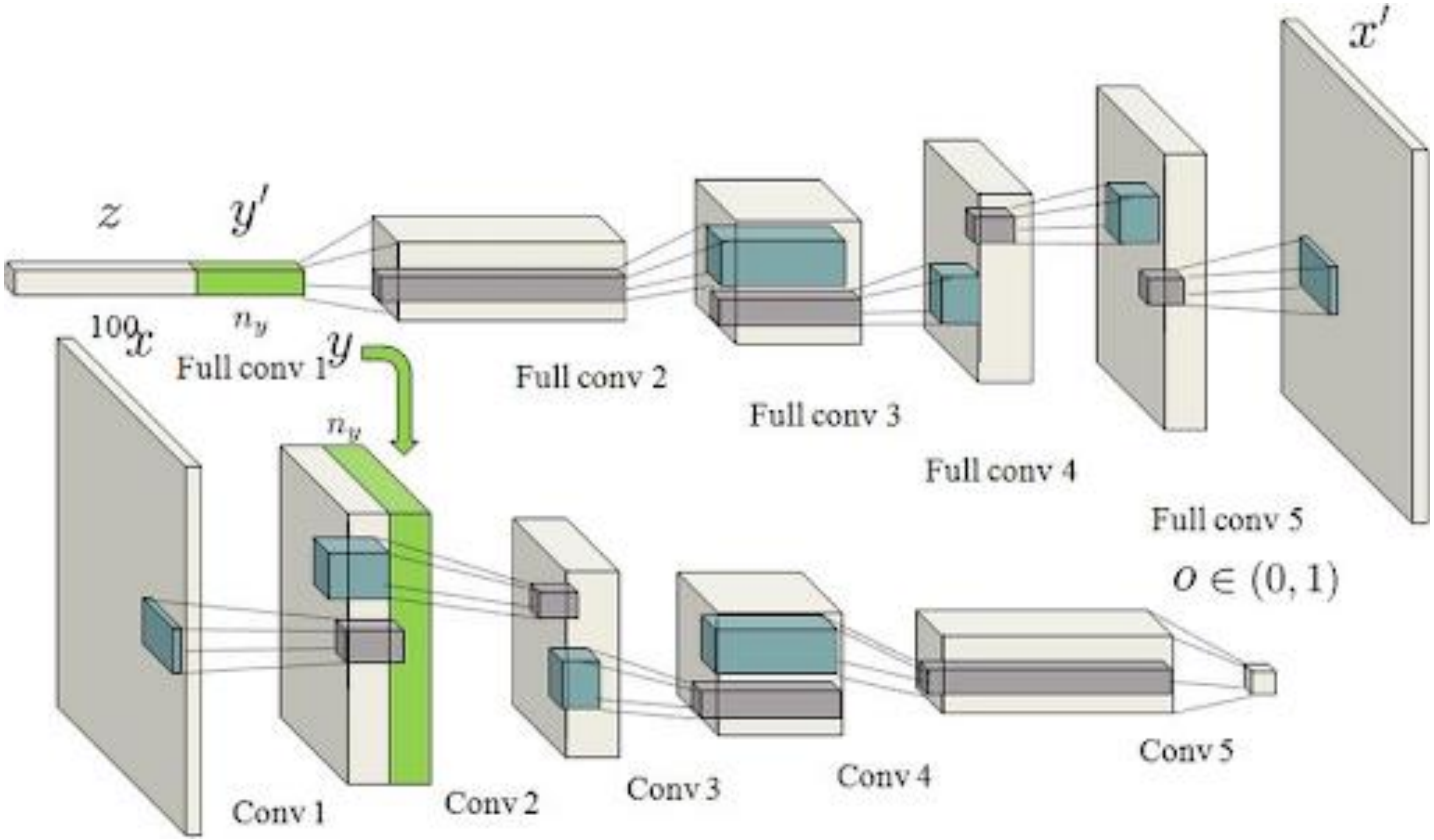
생성함수가 생성한 이미지가 판별함수에서 1에 가까운 숫자가 나와야 작아짐

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)$$

실제데이터와 생성된 데이터의 분포가 같으면 Global Optimum에 도달

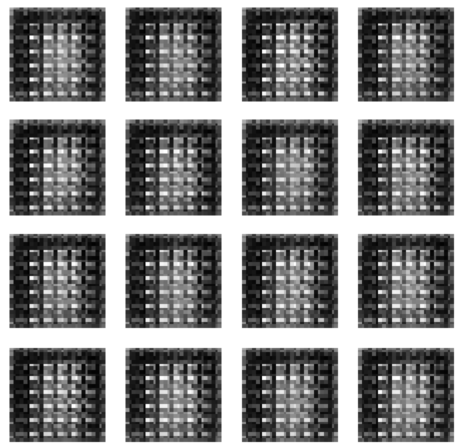
$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \,\|\, p_g\right)$$

Generative Adversarial Nets (GAN)



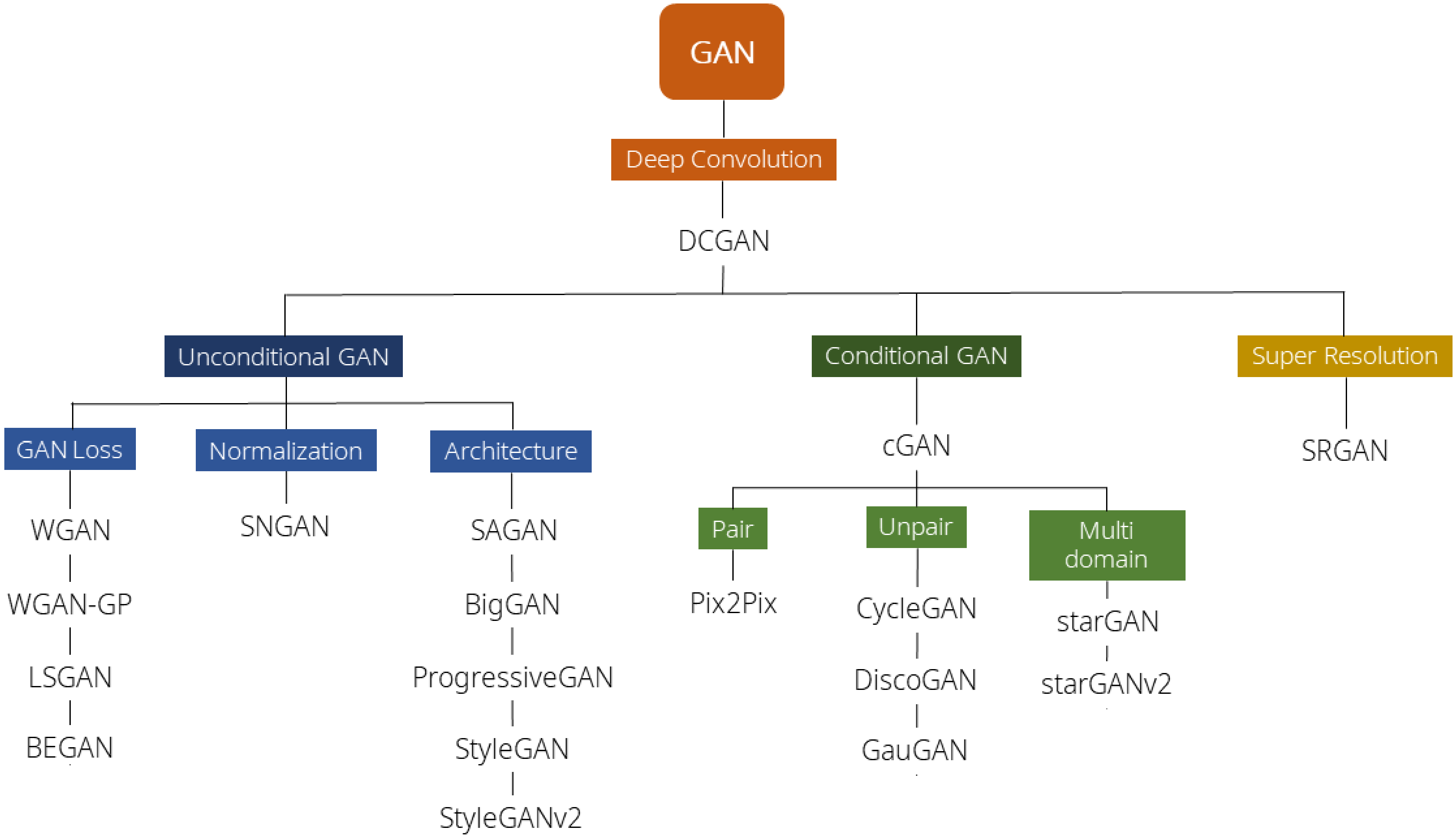
심층 합성곱 생성적 적대 신경망
MNIST 활용

https://colab.research.google.com/github/tensorflow/docs-l10n/blob/master/site/ko/tutorials/generative/dcgan.ipynb?hl=ko



GAN 의 종류

https://baobao.tistory.com/66



UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS (DCGAN)

https://arxiv.org/pdf/1511.06434.pdf

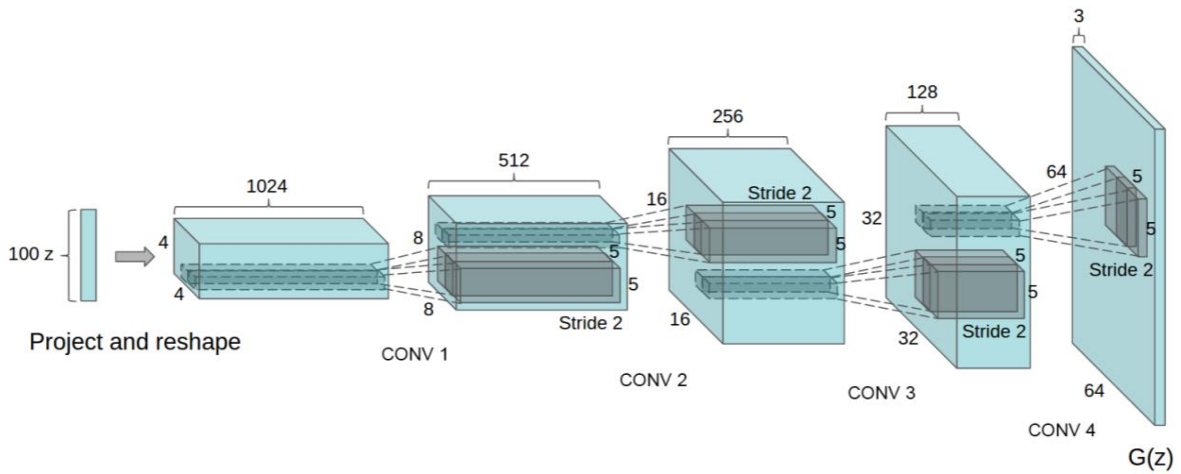


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

Wasserstein GAN (WGAN)

https://arxiv.org/abs/1701.07875

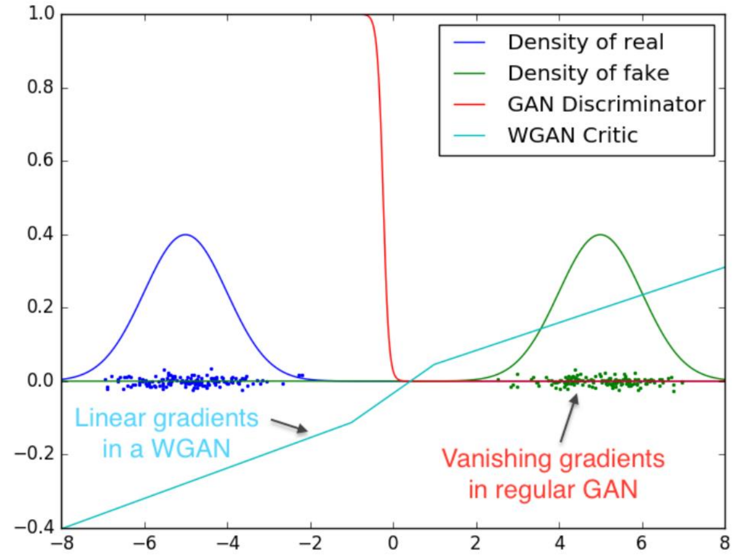
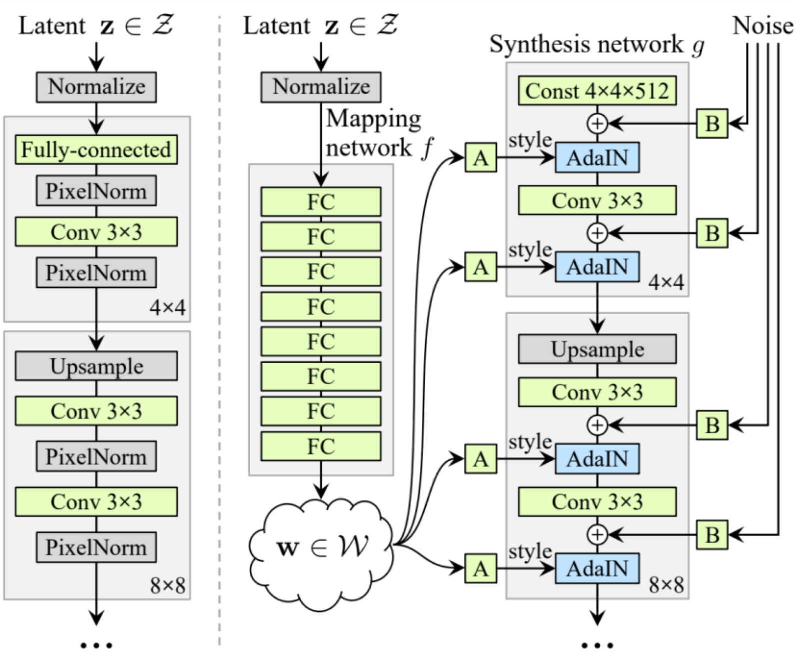


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the discriminator of a minimax GAN saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

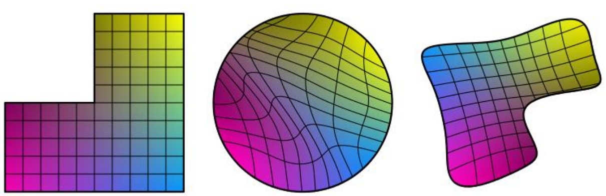
Figure 5: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: standard GAN formulation. Both algorithms produce high quality samples.

A Style-Based Generator Architecture for Generative Adversarial Networks

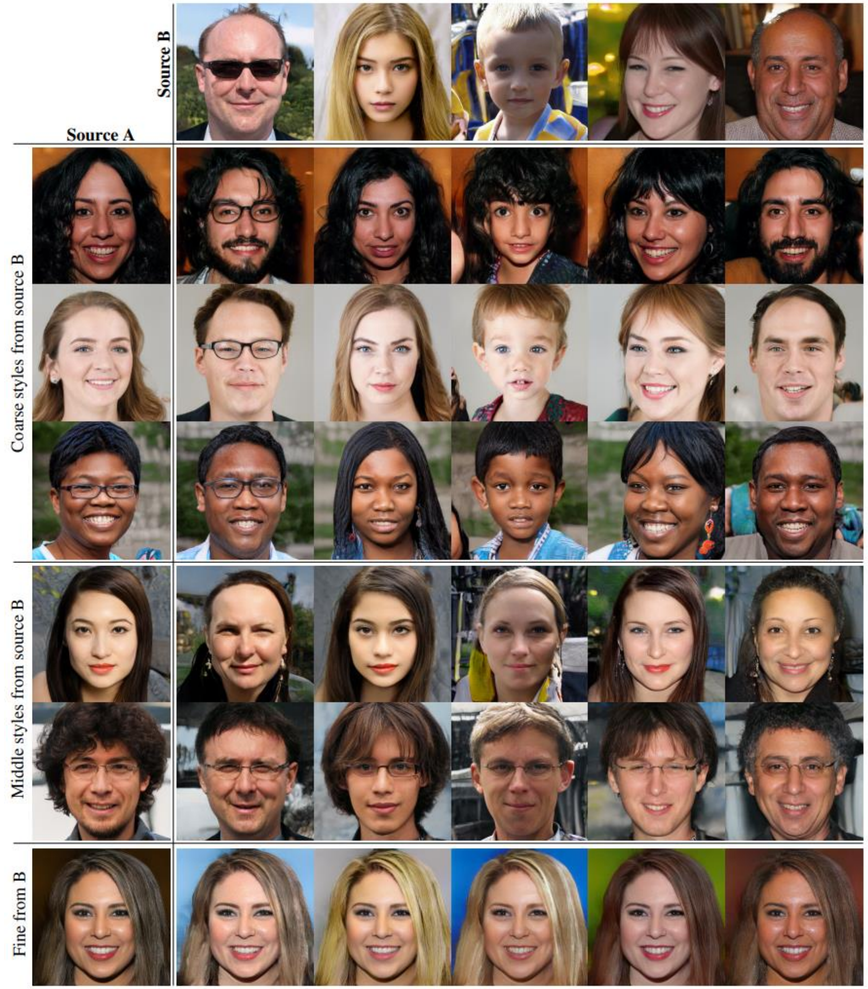
https://arxiv.org/pdf/1812.04948.pdf



(a) Traditional (b) Style-based generator



(a) Distribution of features in training set (b) Mapping from $\mathcal{Z}$ to features (c) Mapping from $\mathcal{W}$ to features



Conditional Generative Adversarial Nets

https://arxiv.org/pdf/1411.1784.pdf

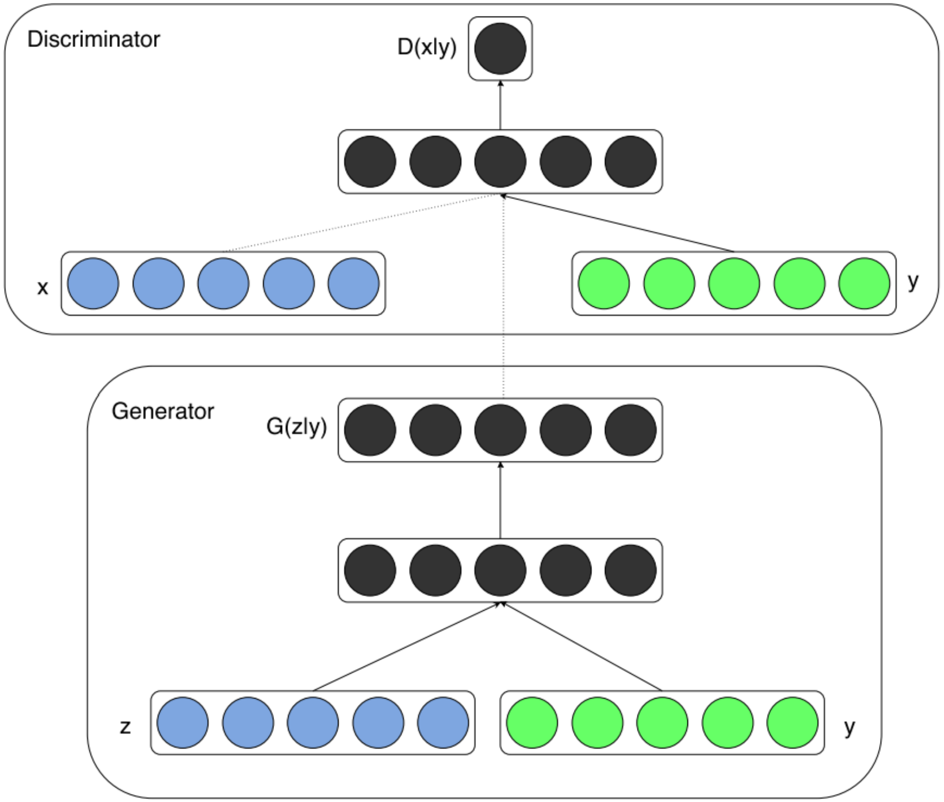


Figure 1: Conditional adversarial net

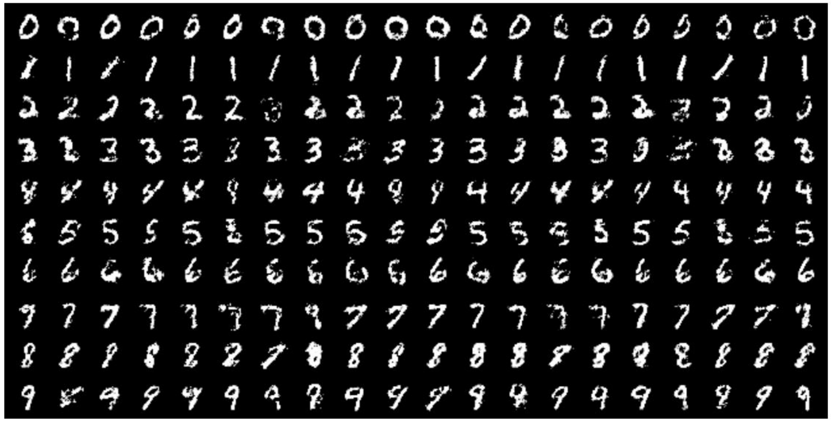


Figure 2: Generated MNIST digits, each row conditioned on one label

Image-to-Image Translation with Conditional Adversarial Networks

https://arxiv.org/pdf/1611.07004.pdf

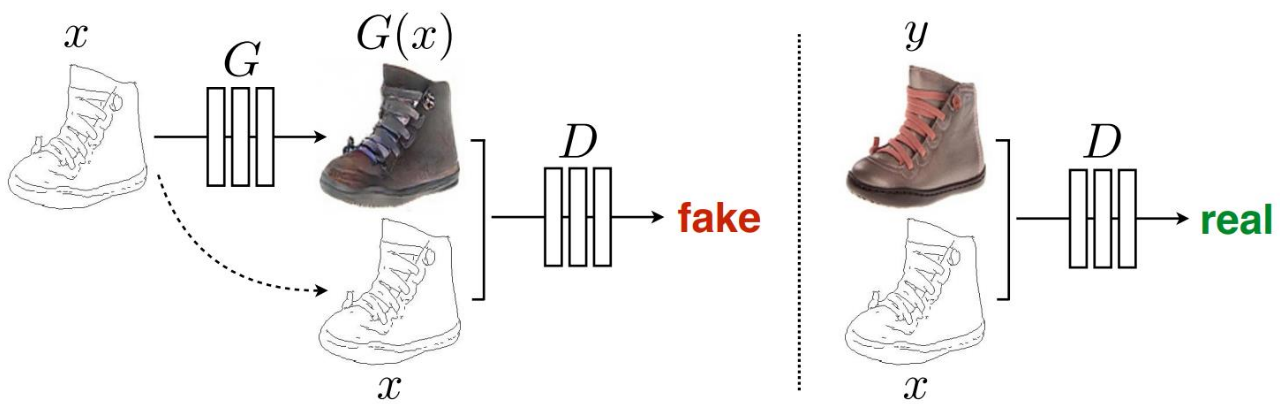
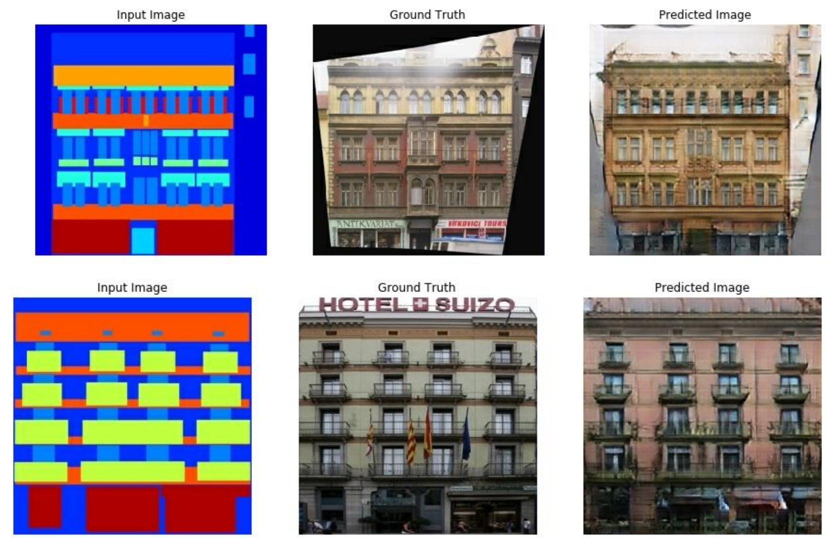


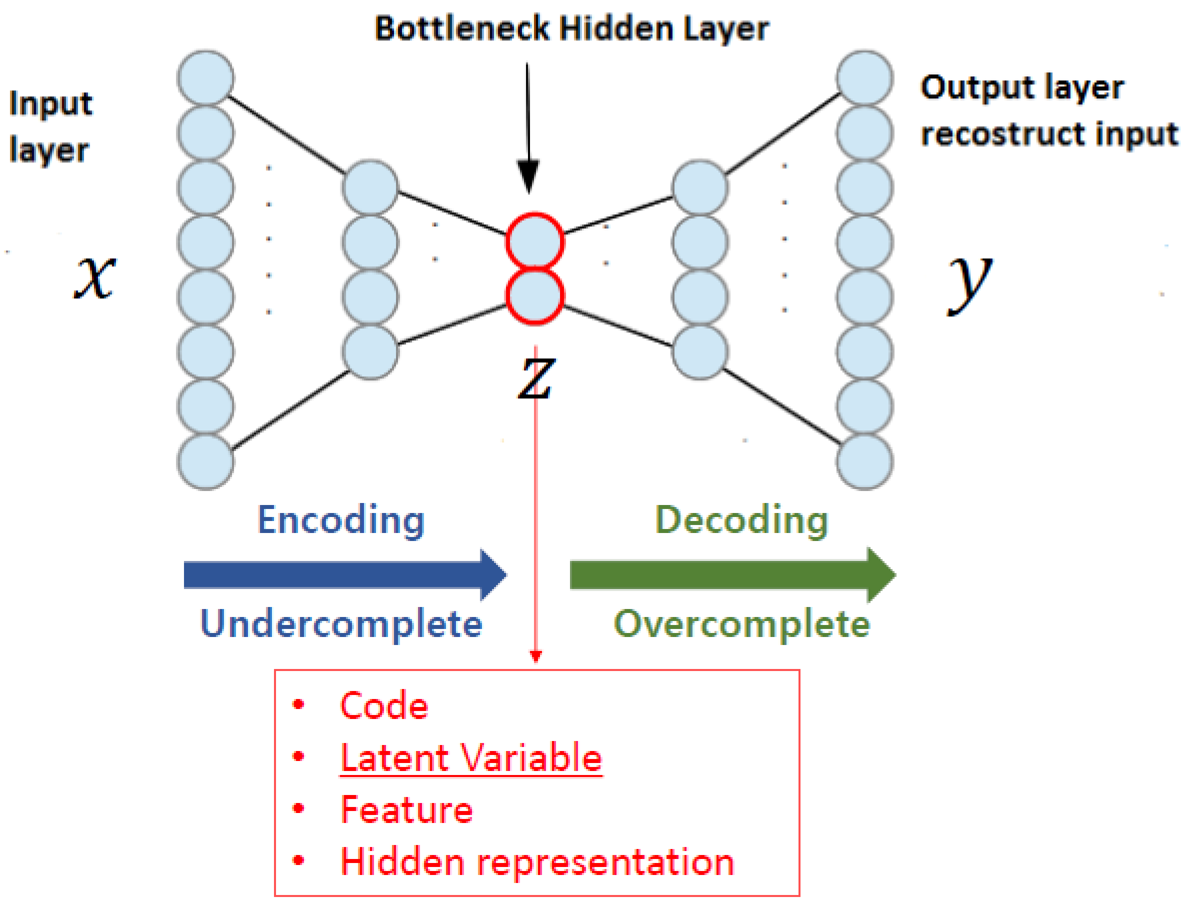
Figure 8: Example results on Google Maps at 512x512 resolution (model was trained on images at 256 × 256 resolution, and run convolutionally on the larger images at test time). Contrast adjusted for clarity.



1. GAN
2. **Autoencoder**
3. Diffusion

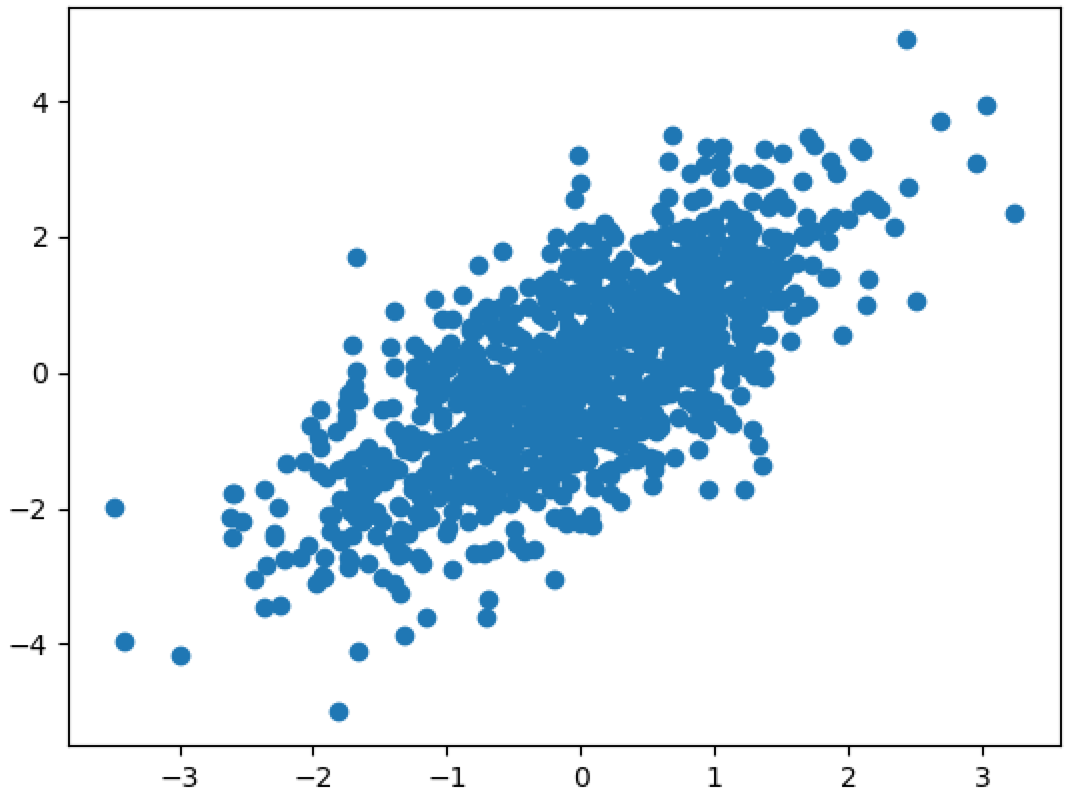
AutoEncoder

https://youtu.be/o_peo6U7IRM

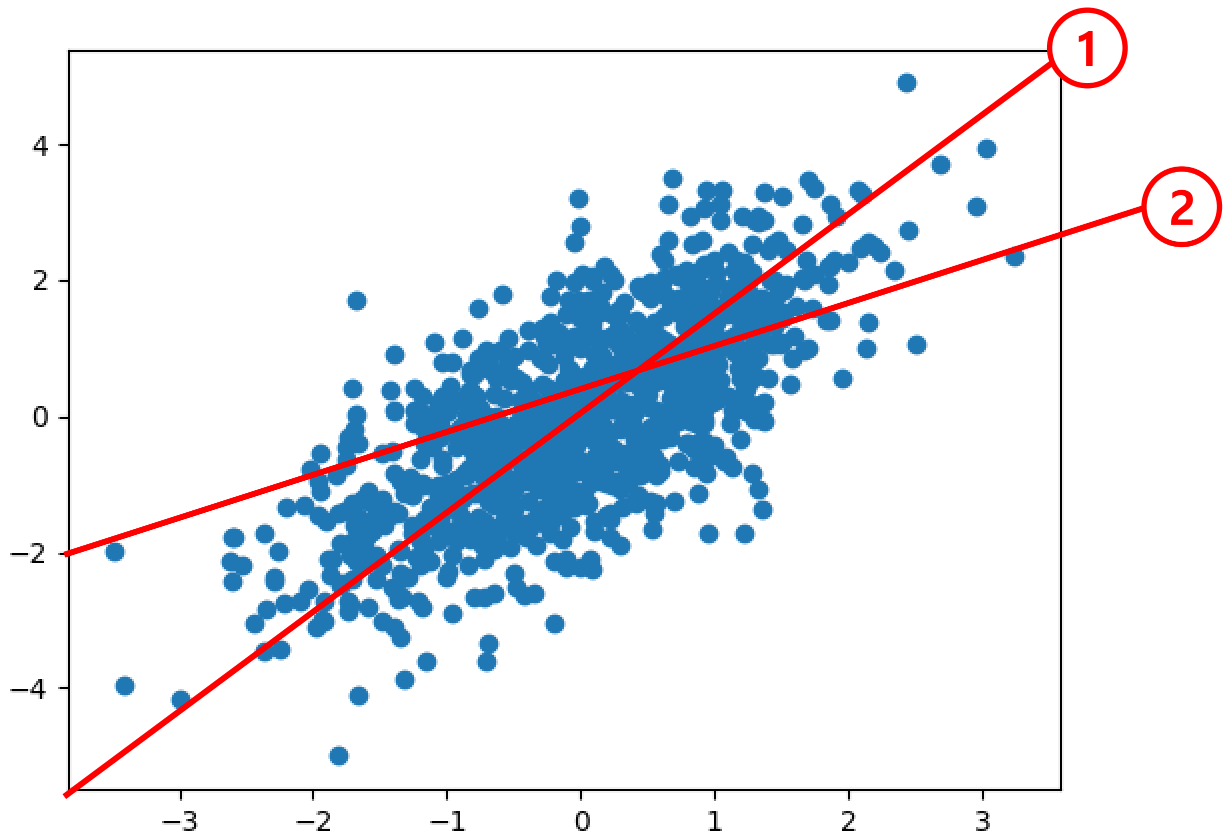


PCA 주성분 분석 (Principle Component Analysis)

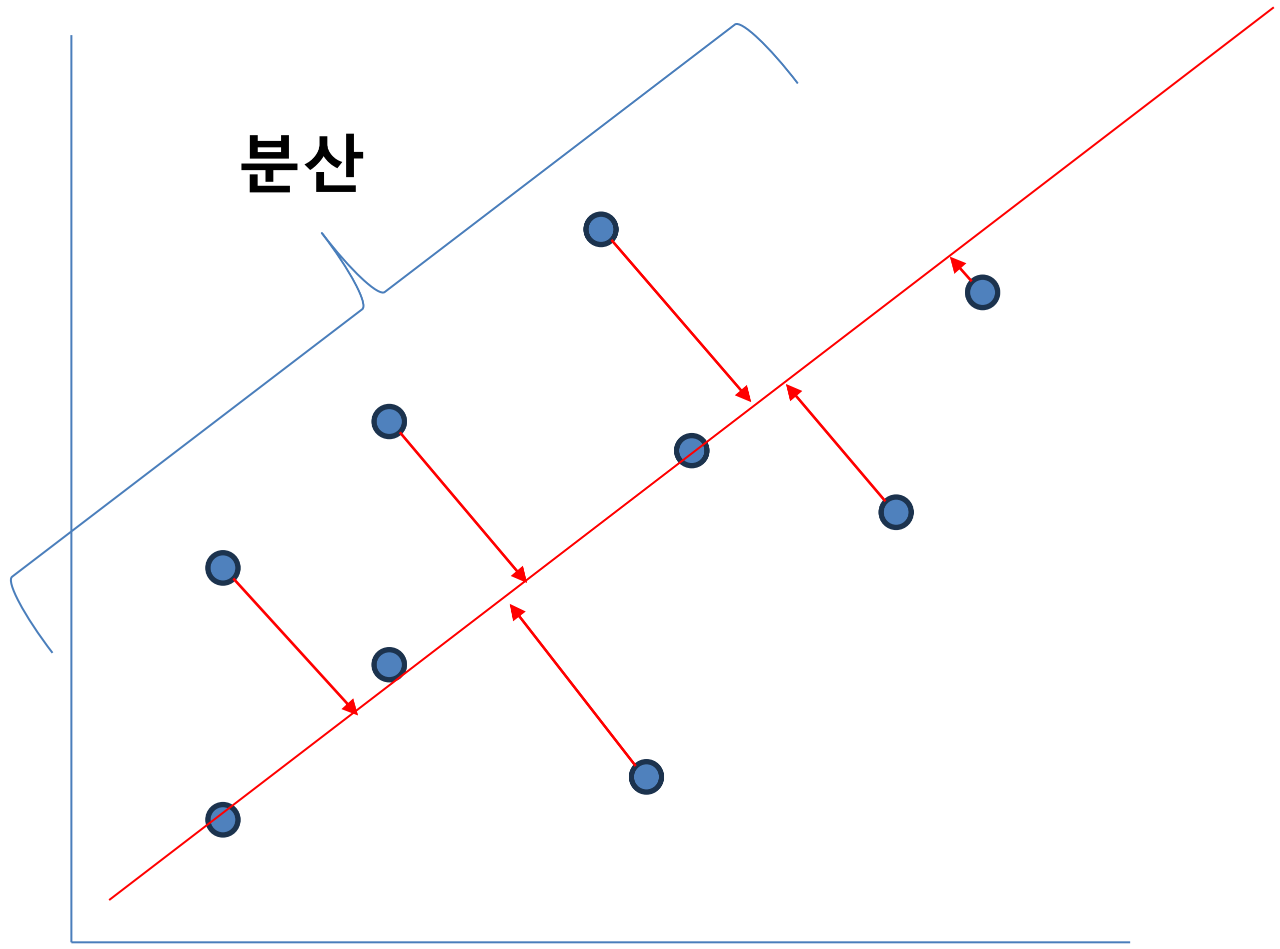
https://youtu.be/VcrEEjFpqCQ?si=mhHyyLgFRrQnD2ef



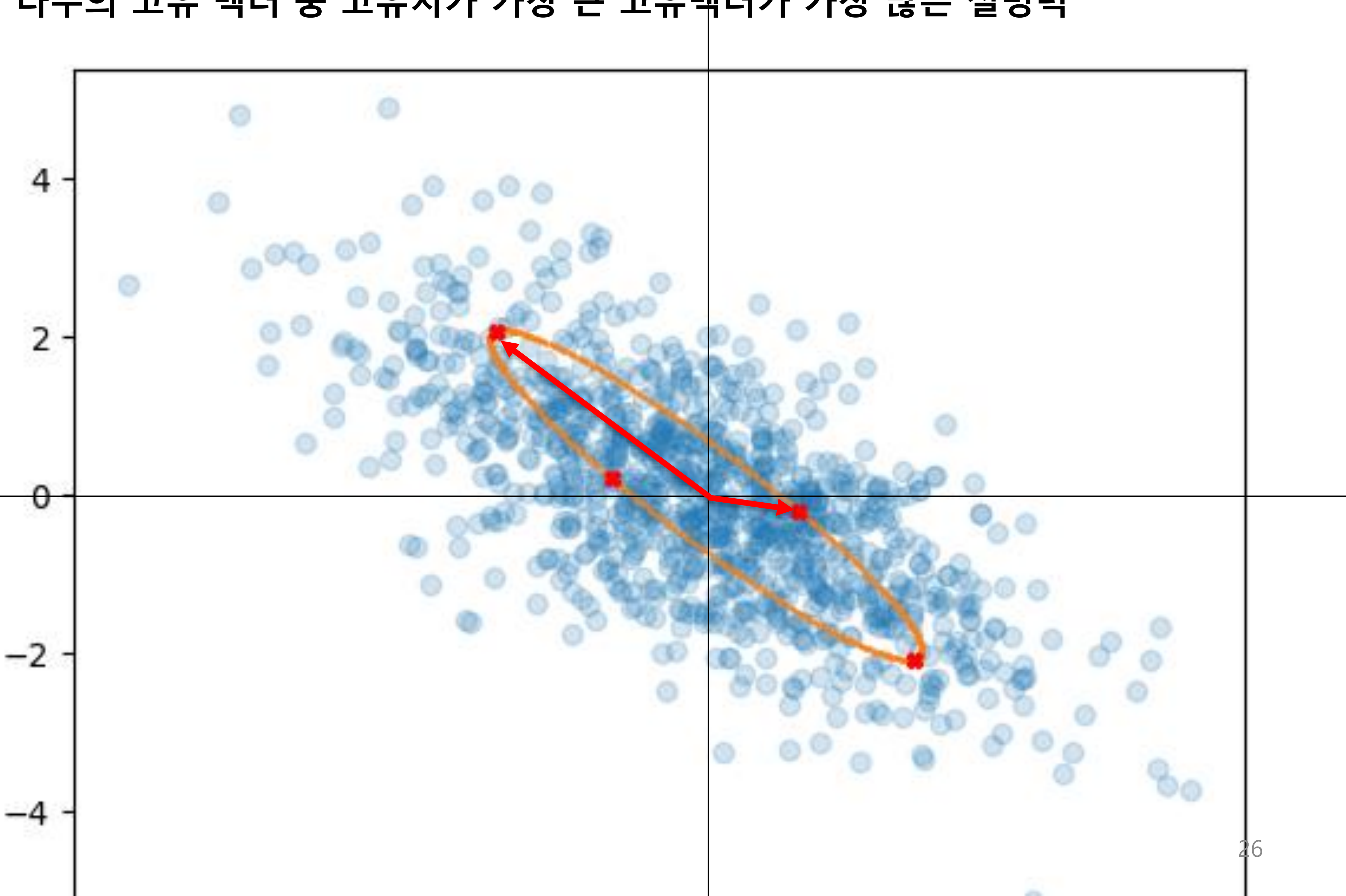
2차원 > 1차원으로 데이터 축소를 하려면 어느 축으로?



분산이 크면 가장 많은 설명력을 담은 저차원 곡선이다.

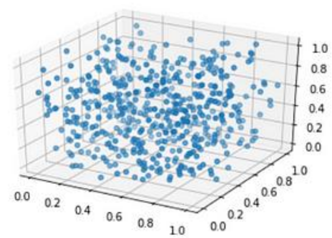
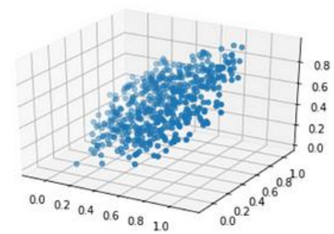
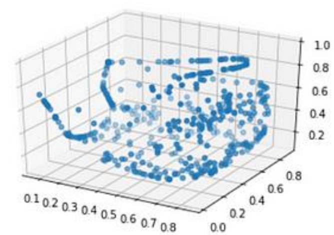


PCA는 공분산의 고유벡터가 가장 데이터를 잘 설명함
다수의 고유 벡터 중 고유치가 가장 큰 고유벡터가 가장 많은 설명력



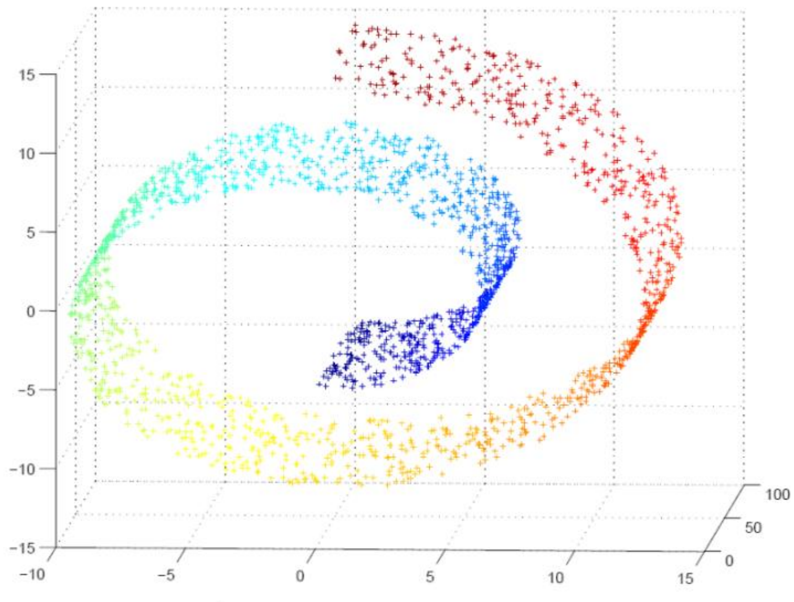
PCA 주성분 분석 vs AutoEncoder

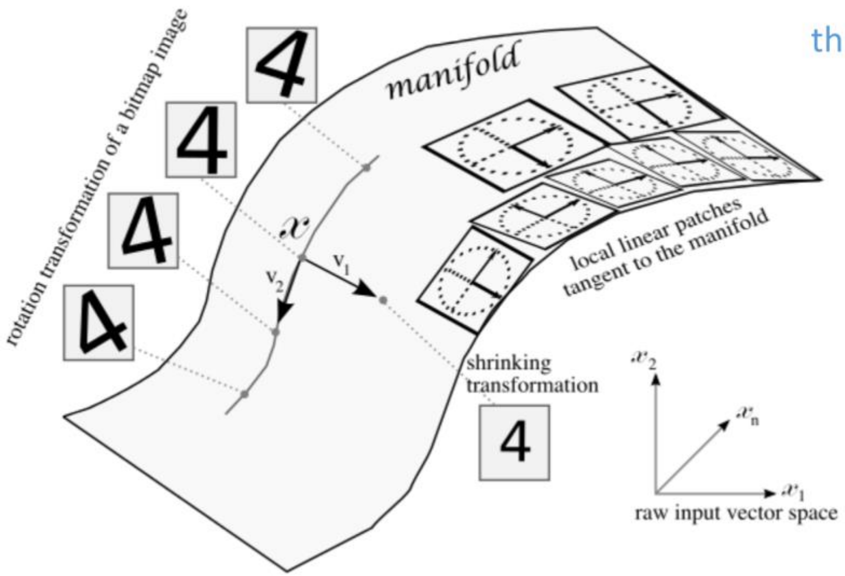
https://towardsdatascience.com/autoencoders-vs-pca-when-to-use-which-73de063f5d7

	Feature Space	PCA Reconstruction	Auto Encoder Reconstruction
Random Data			
Reconstruction Cost (MSE)		0.024	0.010

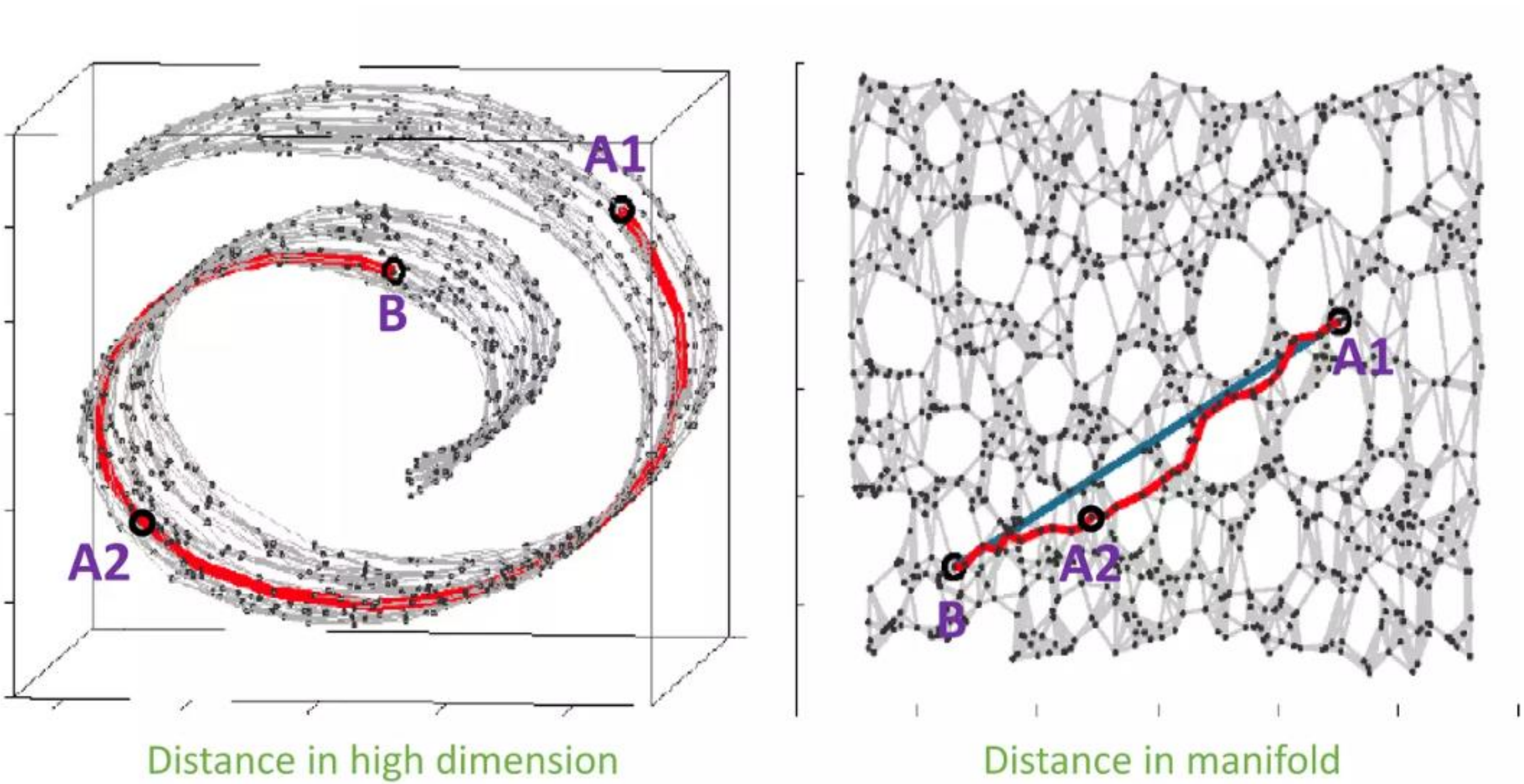
Manifold

http://vision-explorer.reactive.ai/#/galaxy?_k=37rsjx

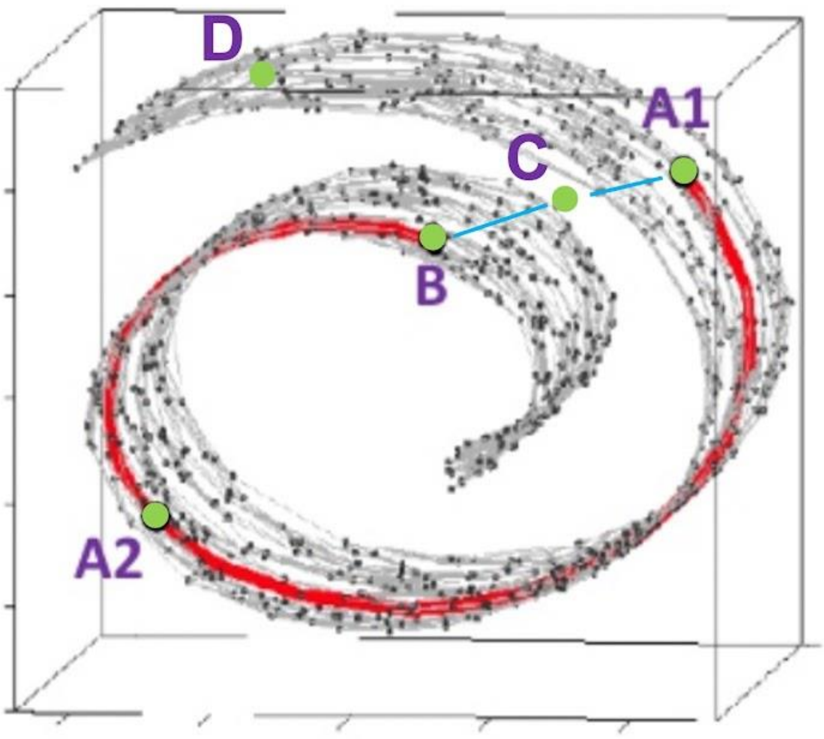


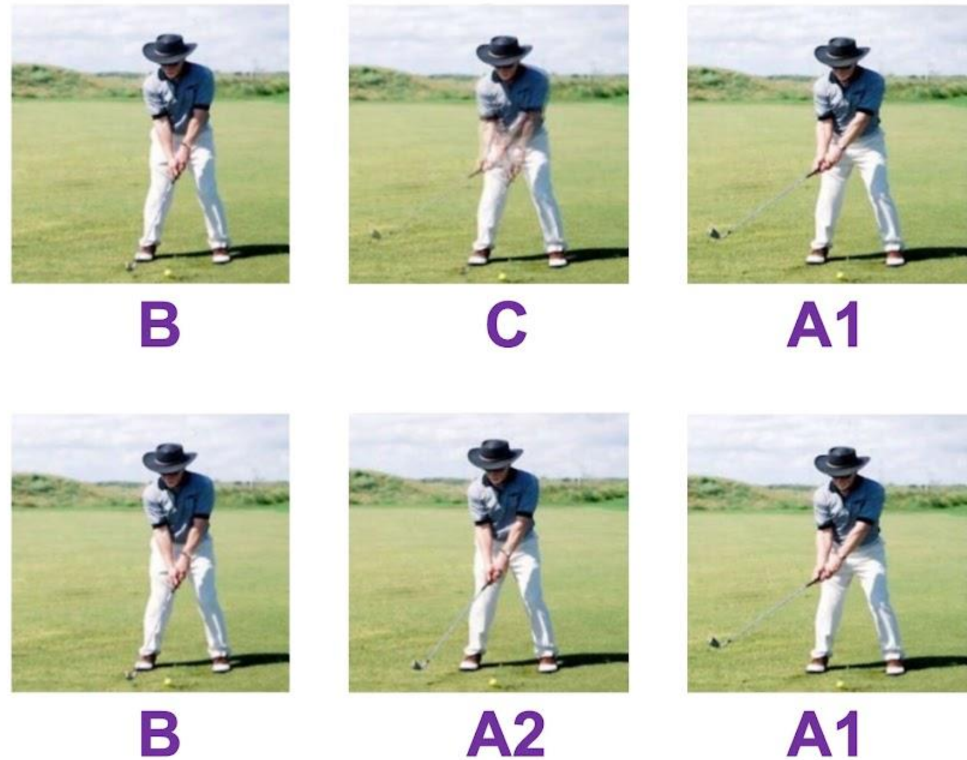


고차원 공간에서의 거리와 Manifold에서의 거리 차이



Manifold를 벗어난 이미지





Manifold Learning Algorithm

Dimensionality Reduction

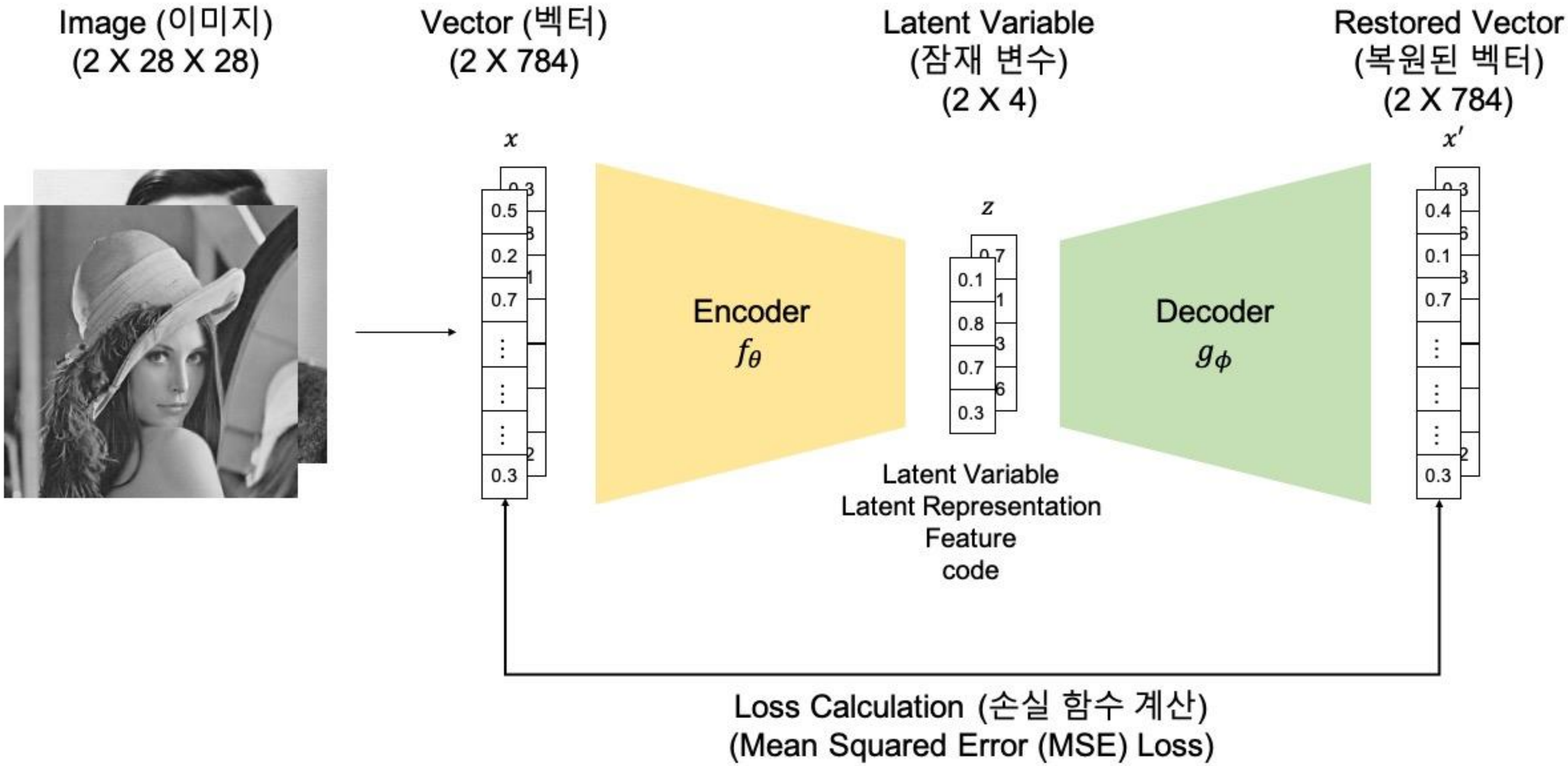
Linear

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- etc..

Non-Linear

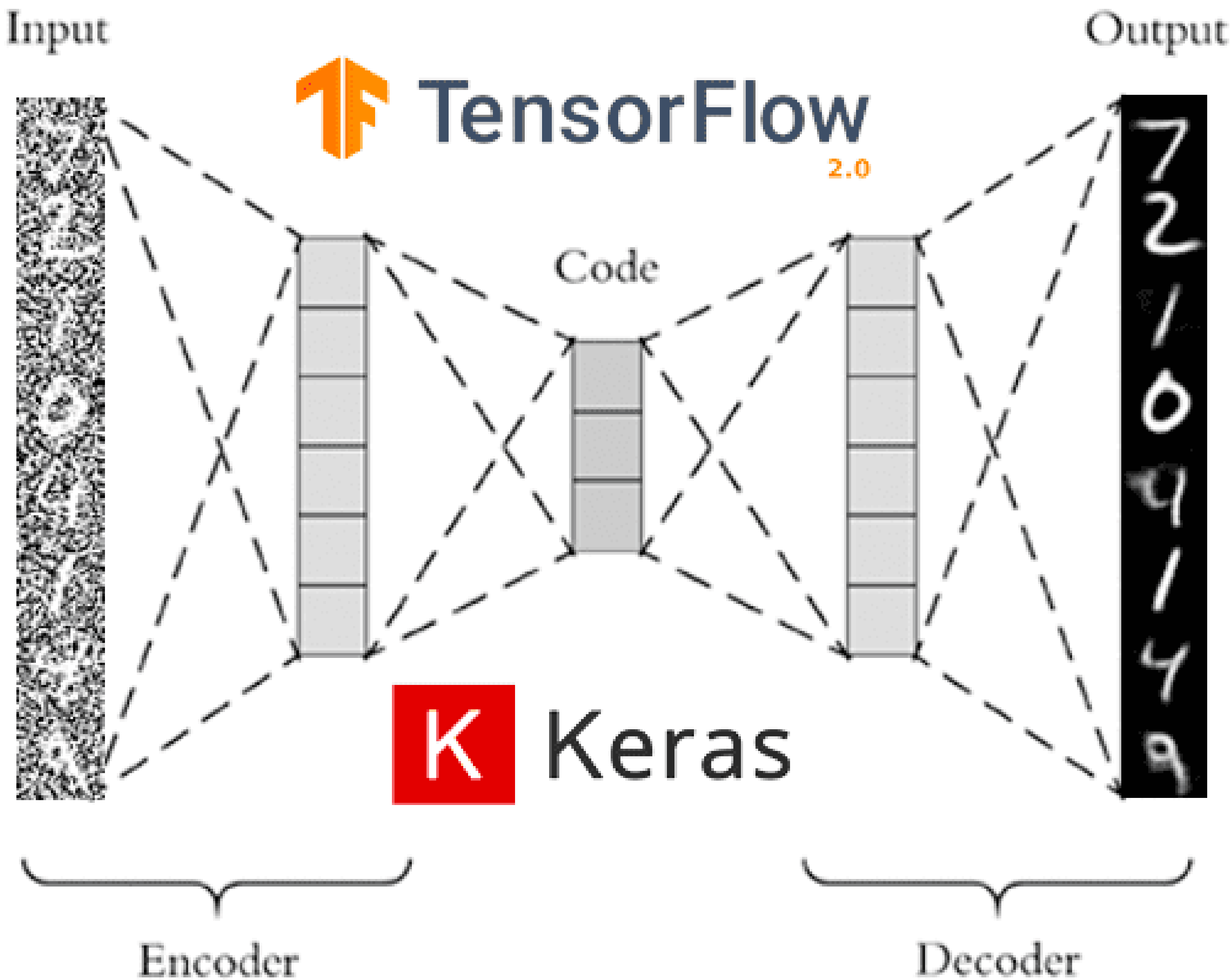
- Autoencoders (AE)
- t-distributed stochastic neighbor embedding (t-SNE)
- Isomap
- Locally-linear embedding (LLE)
- etc..

Autoencoder



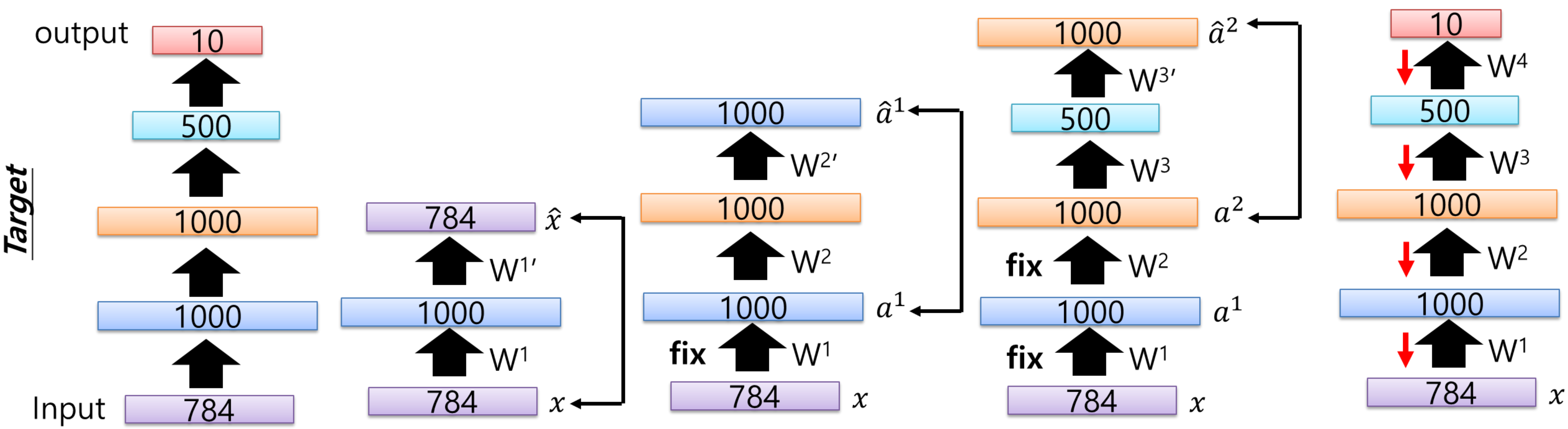
Denoising Autoencoder

https://keras.io/examples/vision/autoencoder/



Denoising Autoencoder

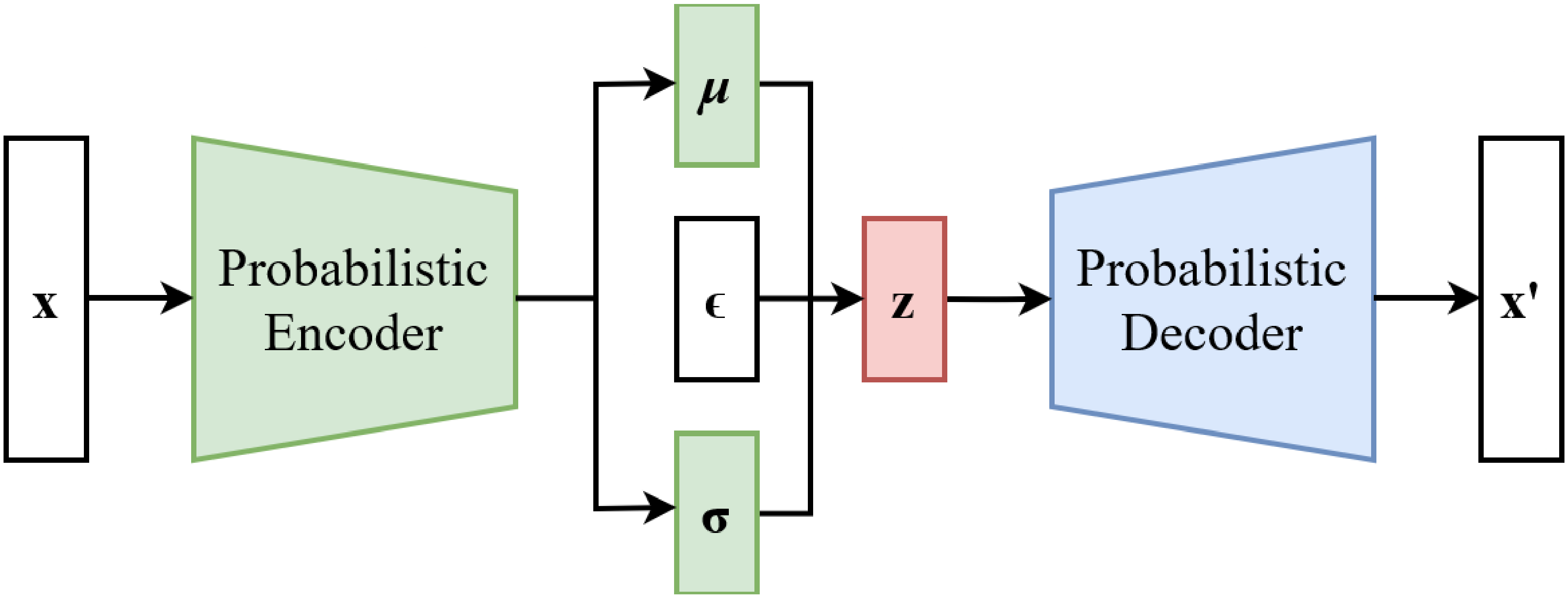
https://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/auto.pptx



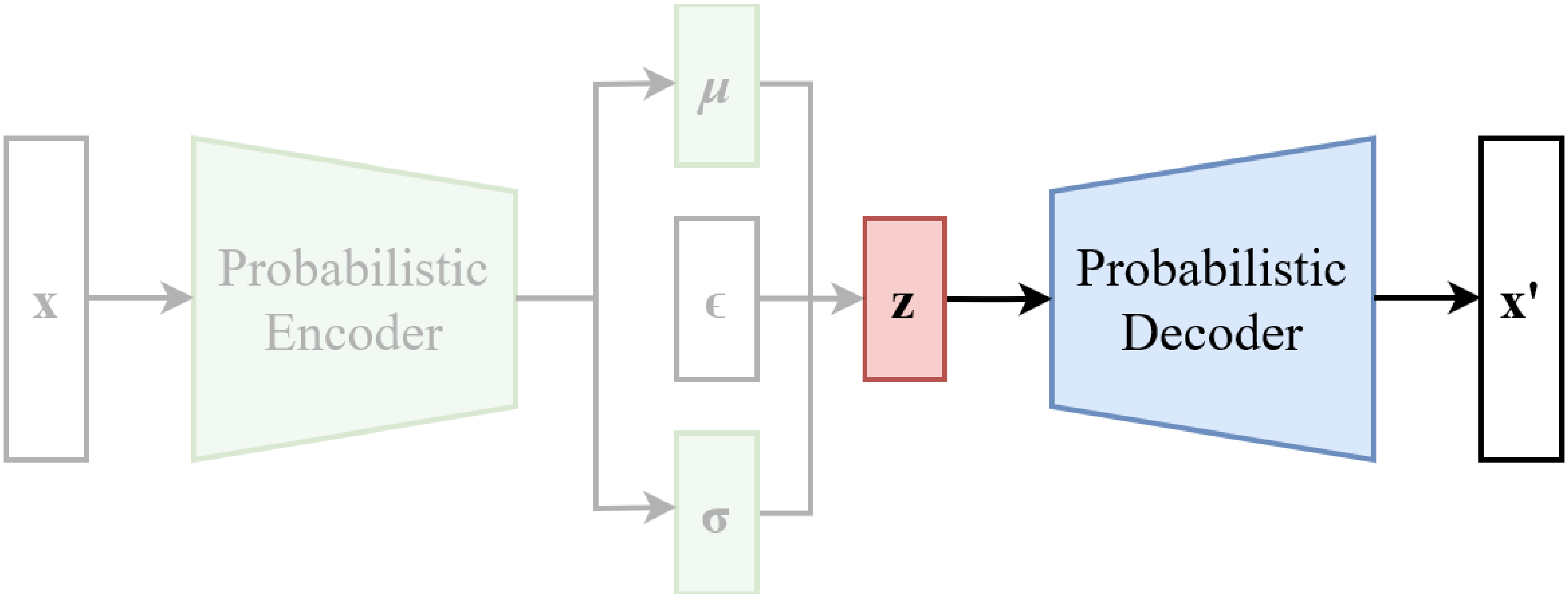
Auto-Encoding Variational Bayes (VAE)

https://arxiv.org/abs/1312.6114

https://arxiv.org/pdf/1606.05908.pdf

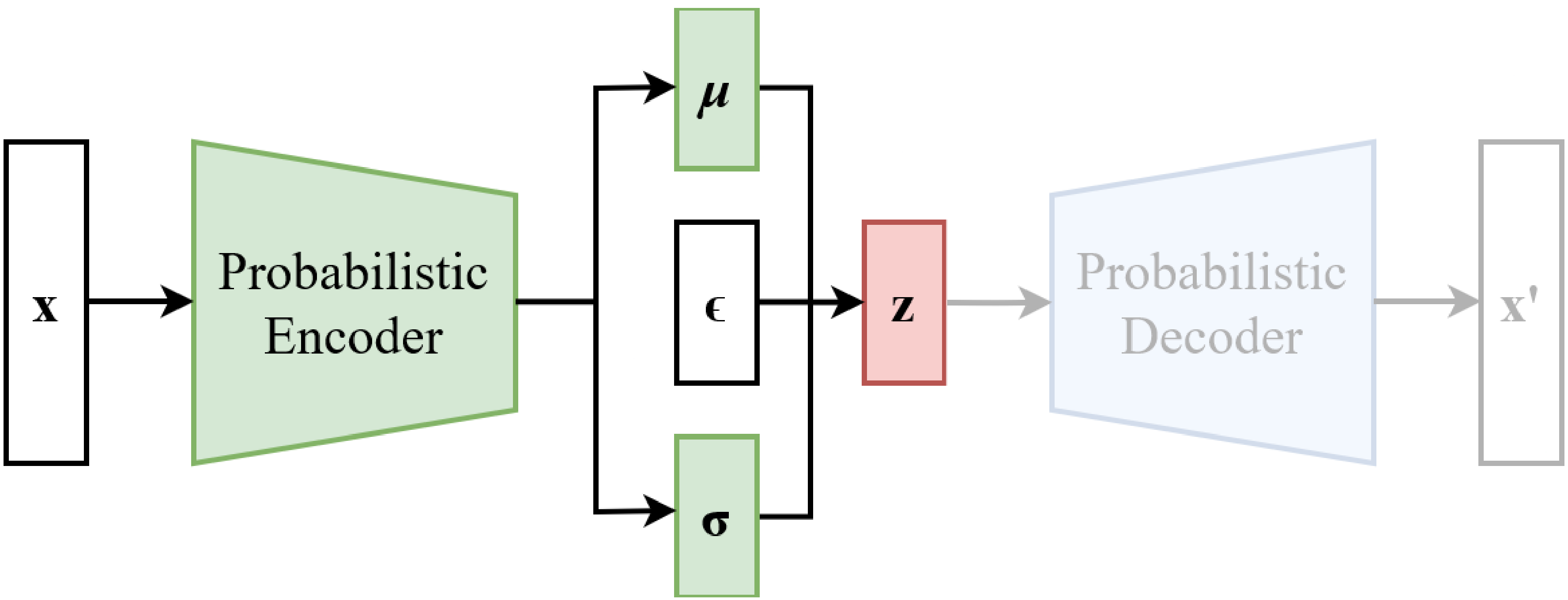


Auto-Encoding Variational Bayes (VAE)



Auto-Encoding Variational Bayes (VAE)

https://keras.io/examples/generative/vae/



Adversarial Autoencoders

https://arxiv.org/abs/1511.05644

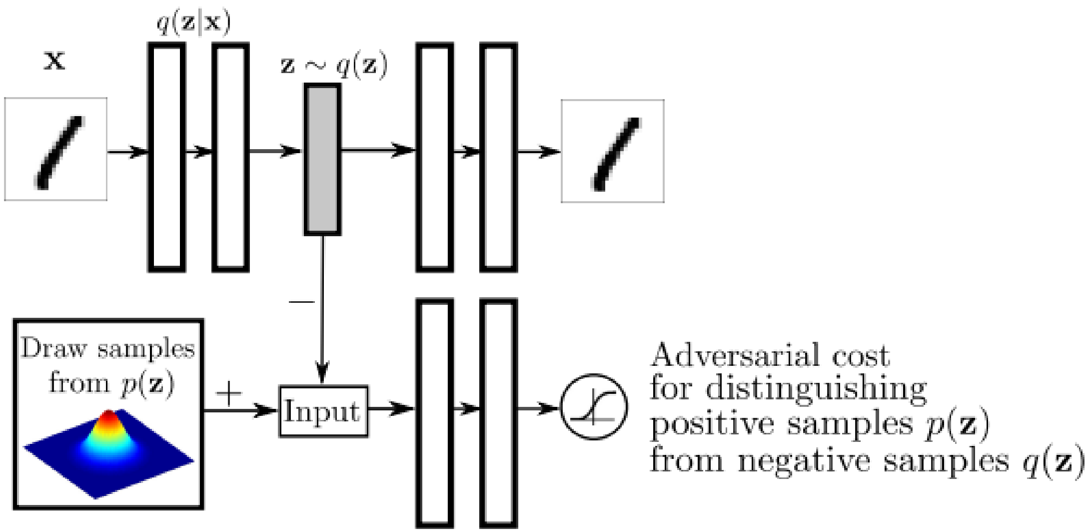
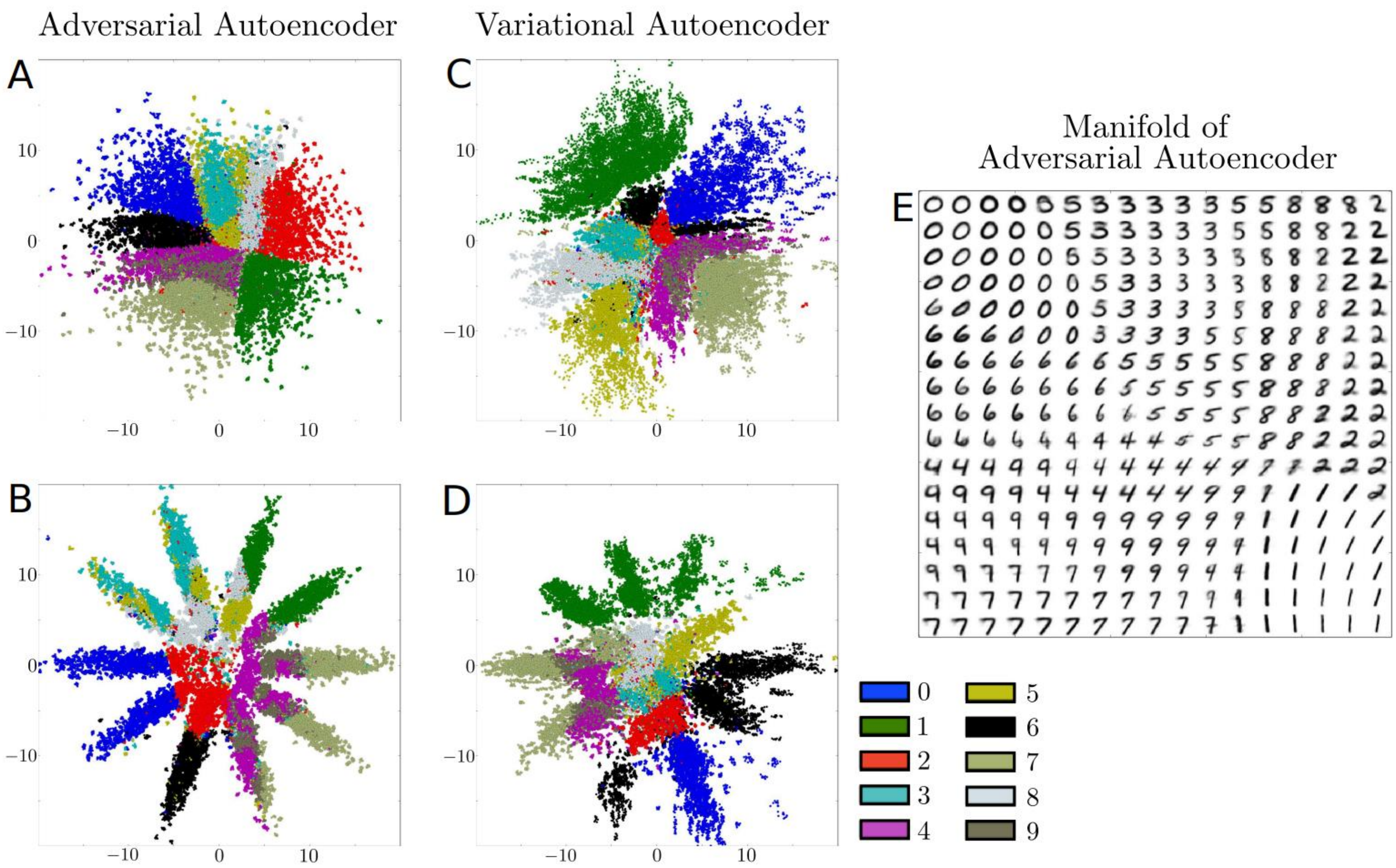


Figure 1: Architecture of an adversarial autoencoder. The top row is a standard autoencoder that reconstructs an image $\mathbf{x}$ from a latent code $\mathbf{z}$. The bottom row diagrams a second network trained to discriminatively predict whether a sample arises from the hidden code of the autoencoder or from a sampled distribution specified by the user.

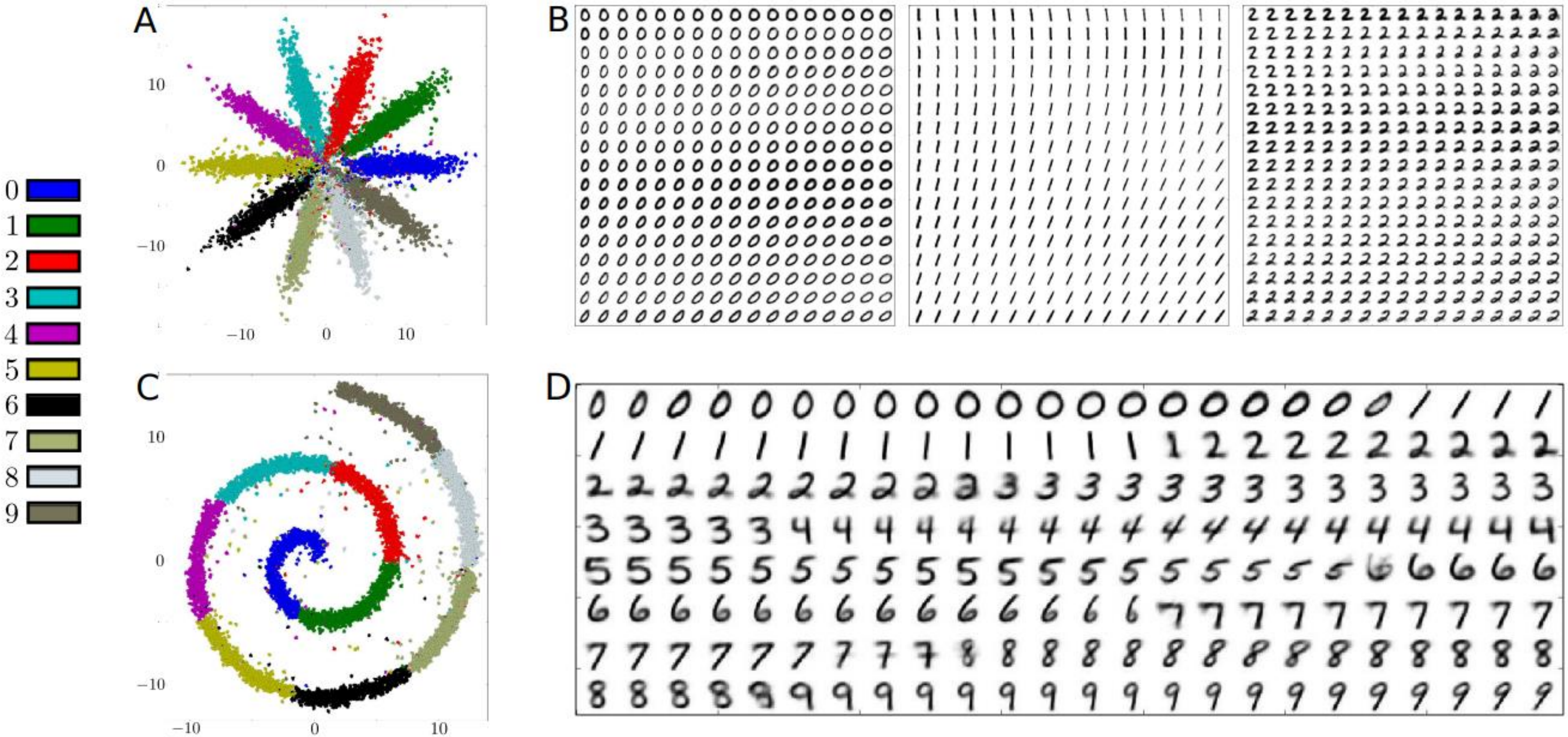
Adversarial Autoencoders

https://arxiv.org/abs/1511.05644



Adversarial Autoencoders

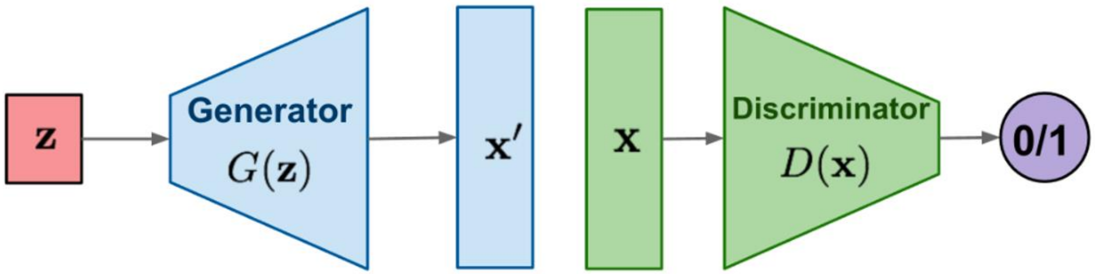
https://arxiv.org/abs/1511.05644



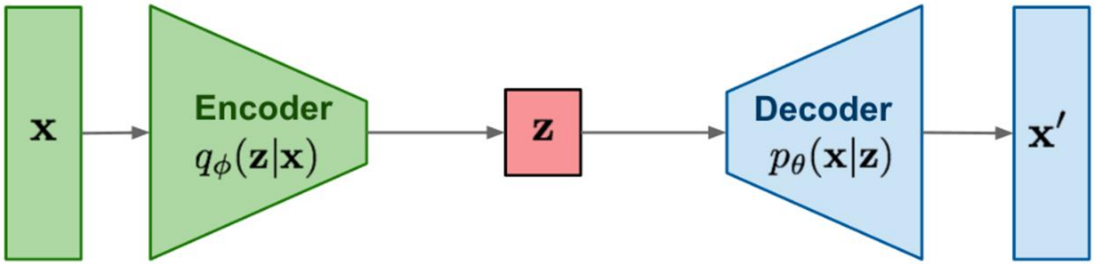
1. GAN
2. Autoencoder
3. **Diffusion**

생성모델 종류

GAN: Adversarial training



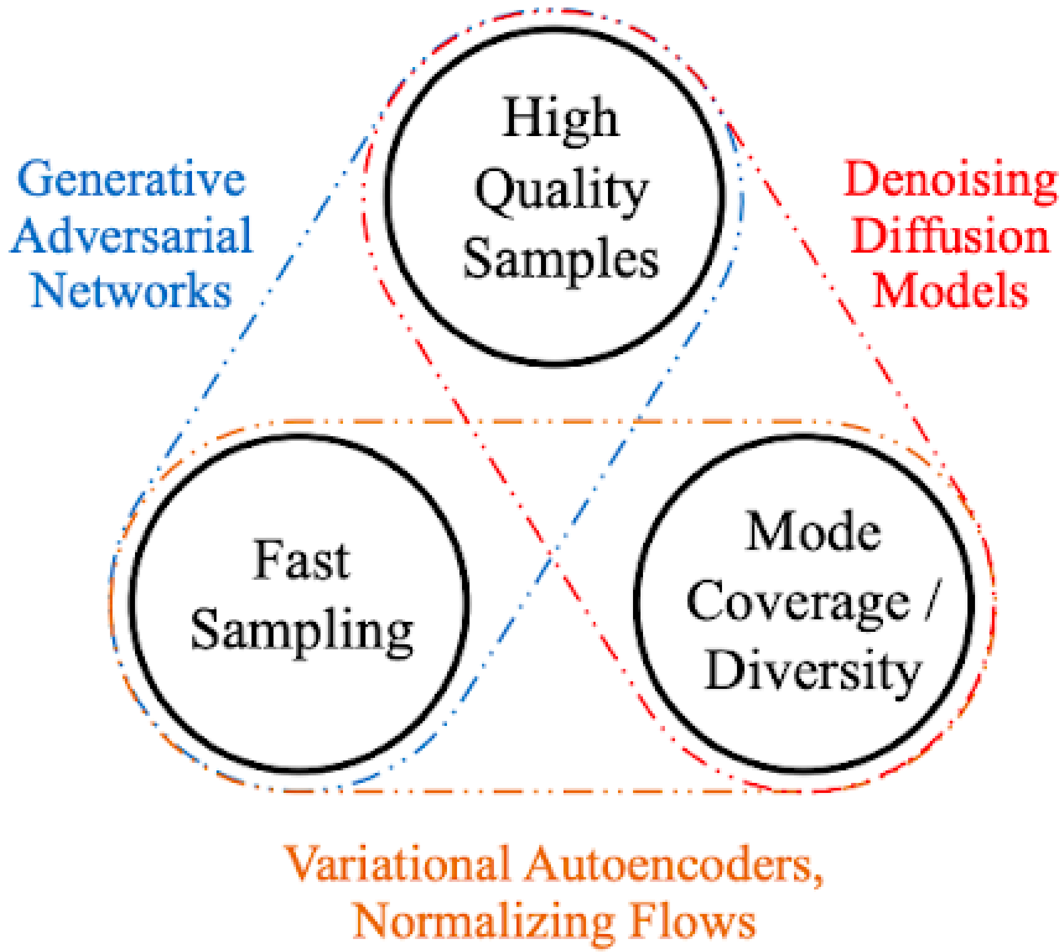
VAE: maximize variational lower bound



Diffusion models: Gradually add Gaussian noise and then reverse

Tackling the Generative Learning Trilemma with Denoising Diffusion GANs

https://arxiv.org/abs/2112.07804

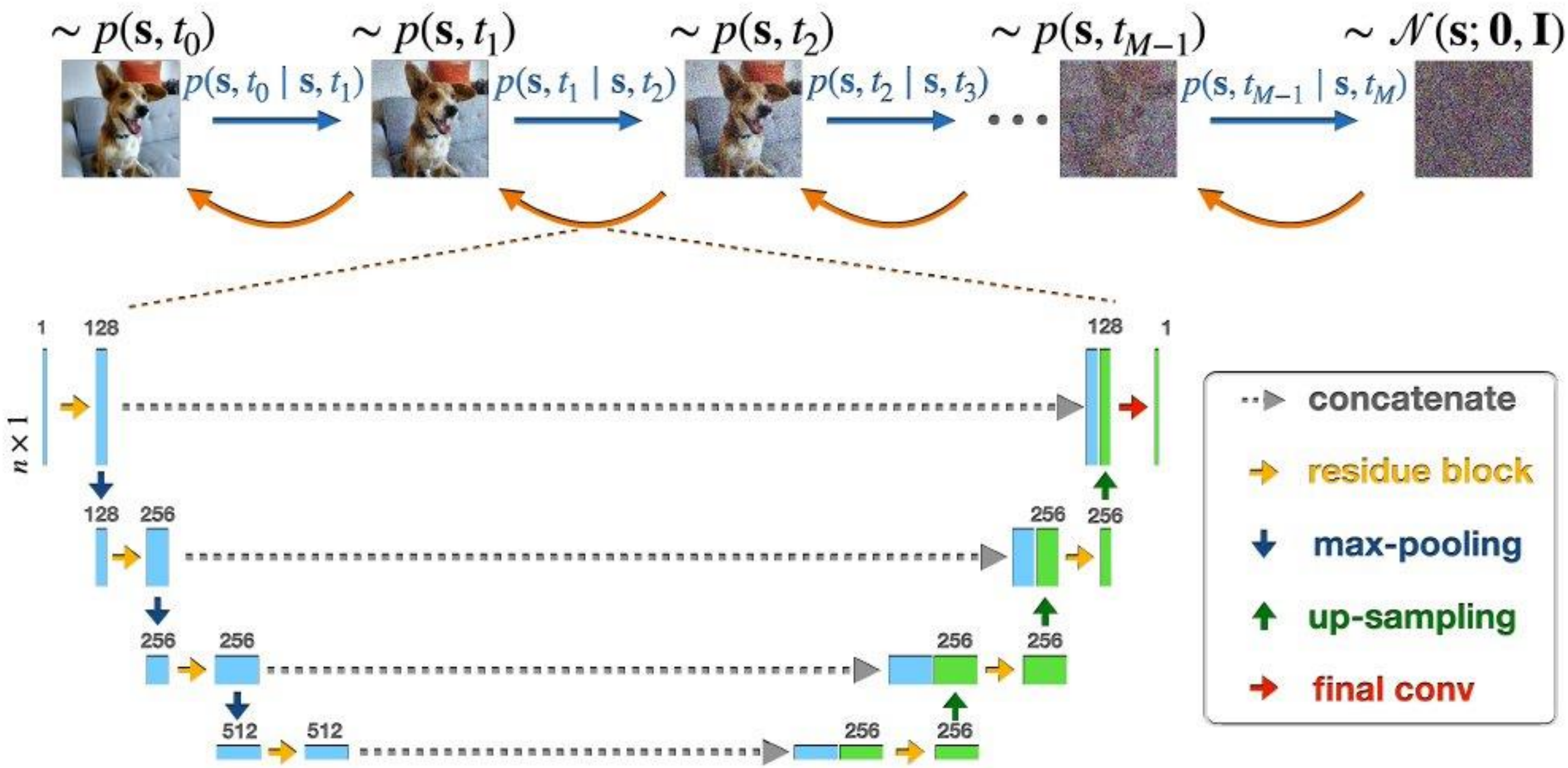


Denoising Diffusion Probabilistic Model

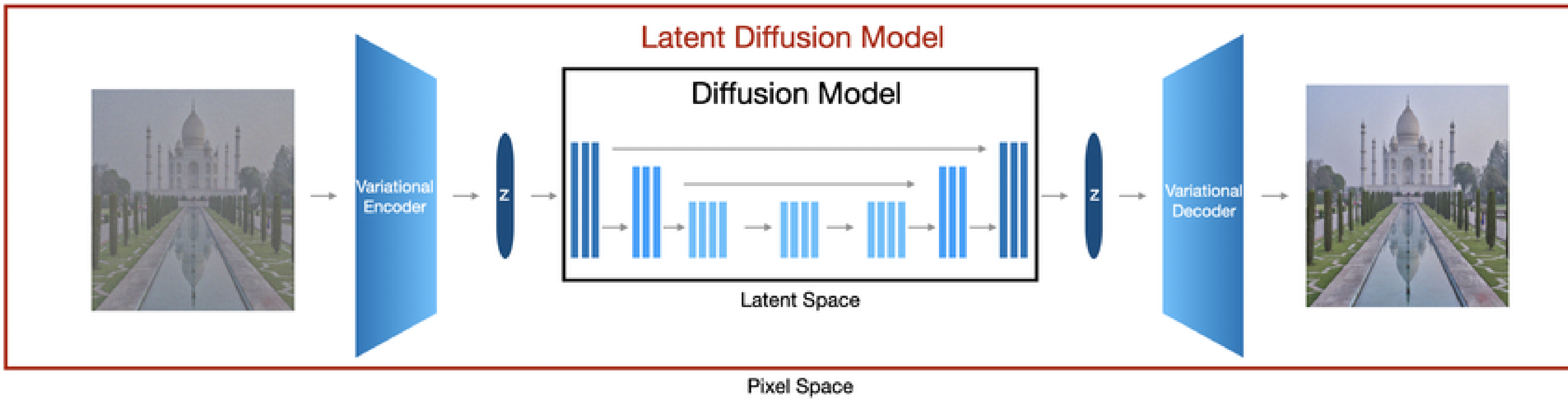
https://arxiv.org/abs/2006.11239

https://keras.io/examples/generative/ddpm/

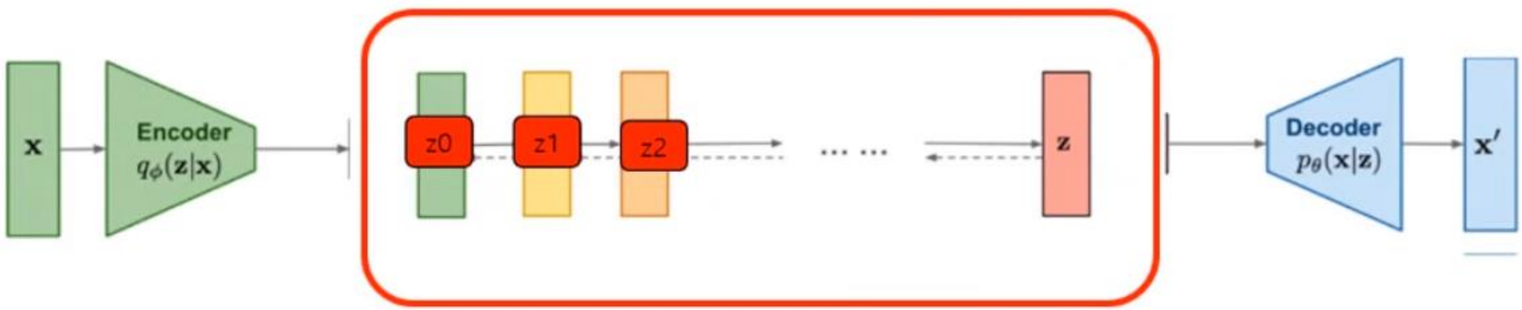
https://colab.research.google.com/?hl=ko



Latent Diffusion Models

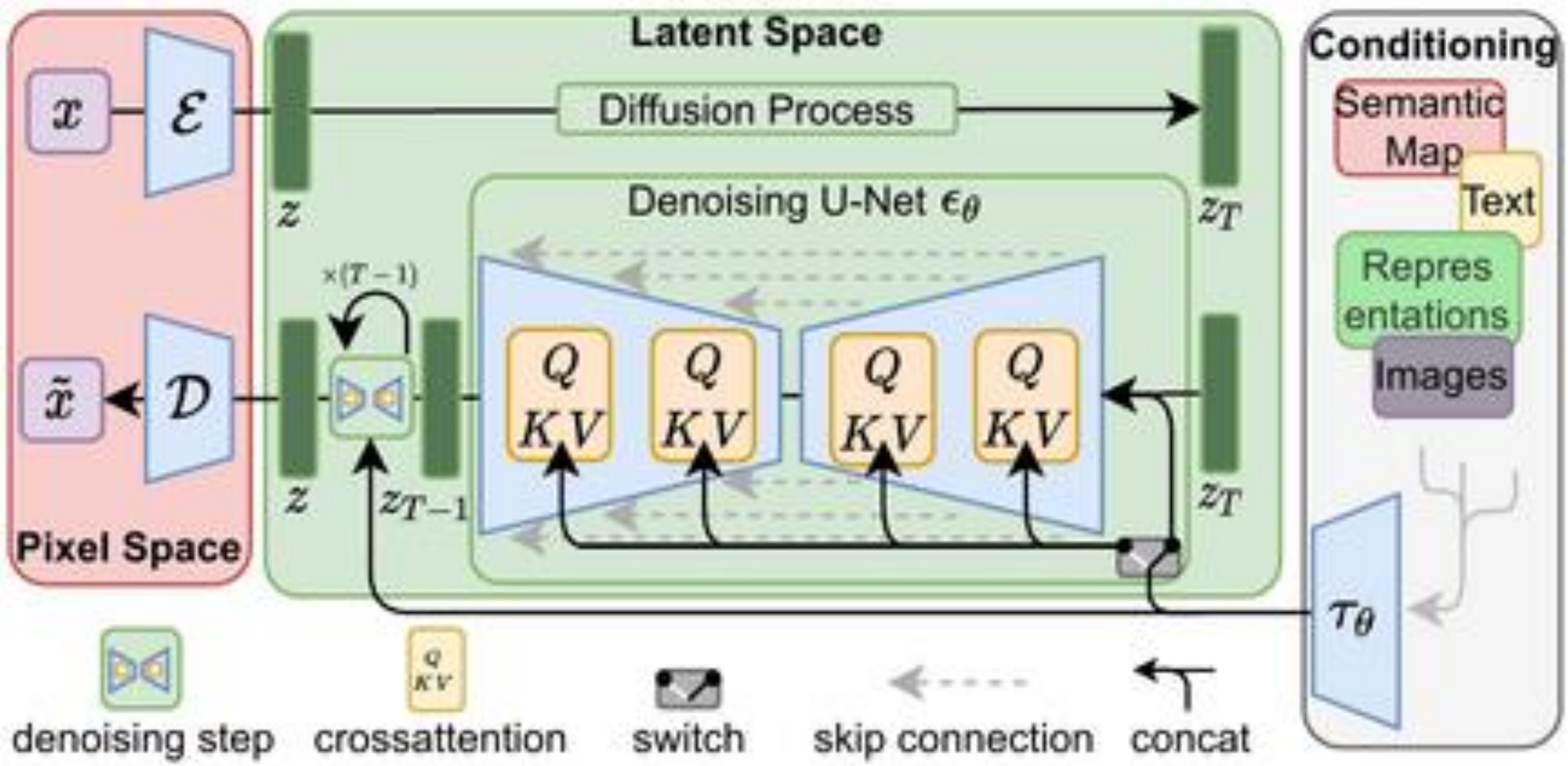


Stable Diffusion



High-Resolution Image Synthesis with Latent Diffusion Models

https://arxiv.org/abs/2112.10752



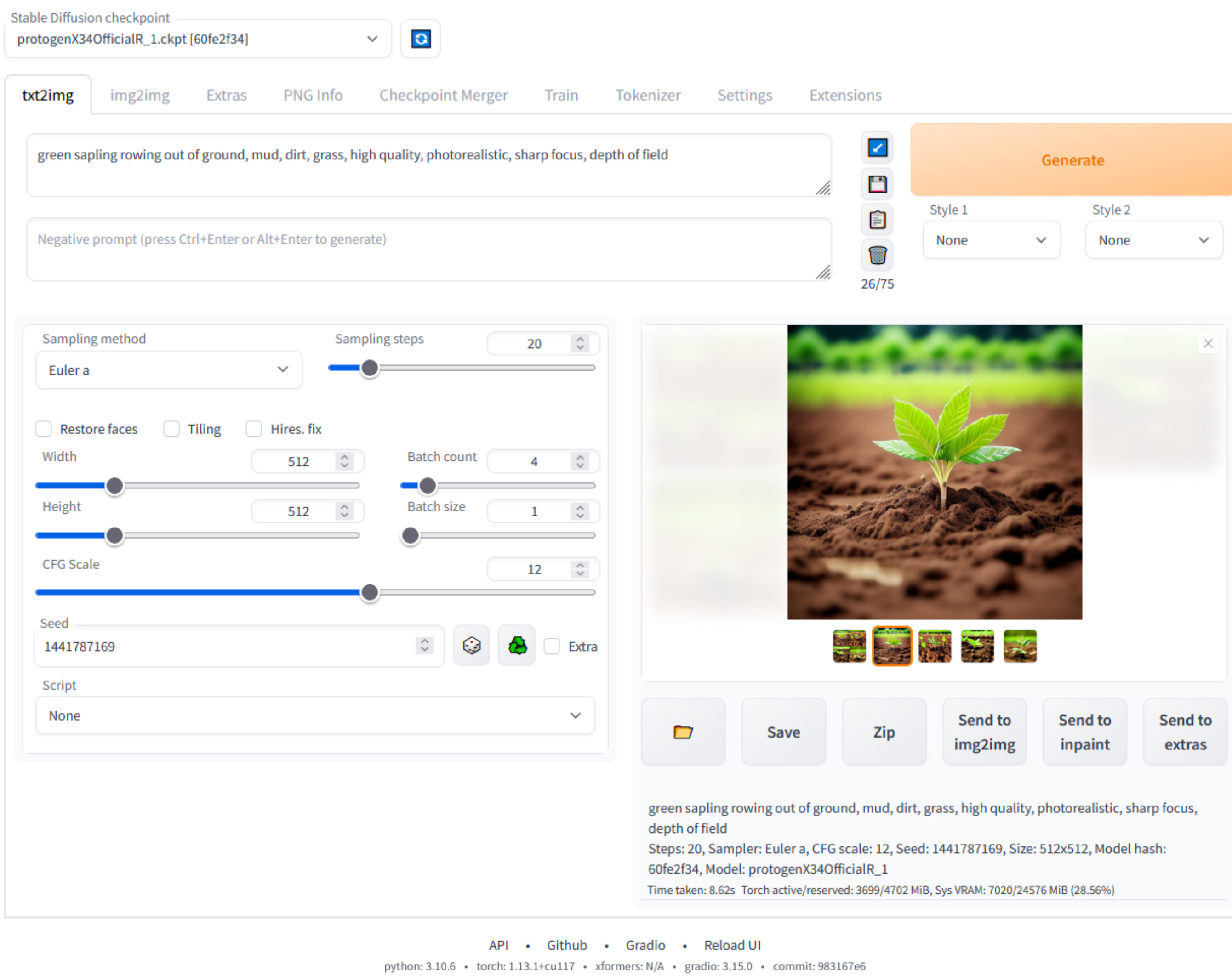
https://huggingface.co/

The AI community building the future.

Build, train and deploy state of the art models powered by the reference open source in machine learning.

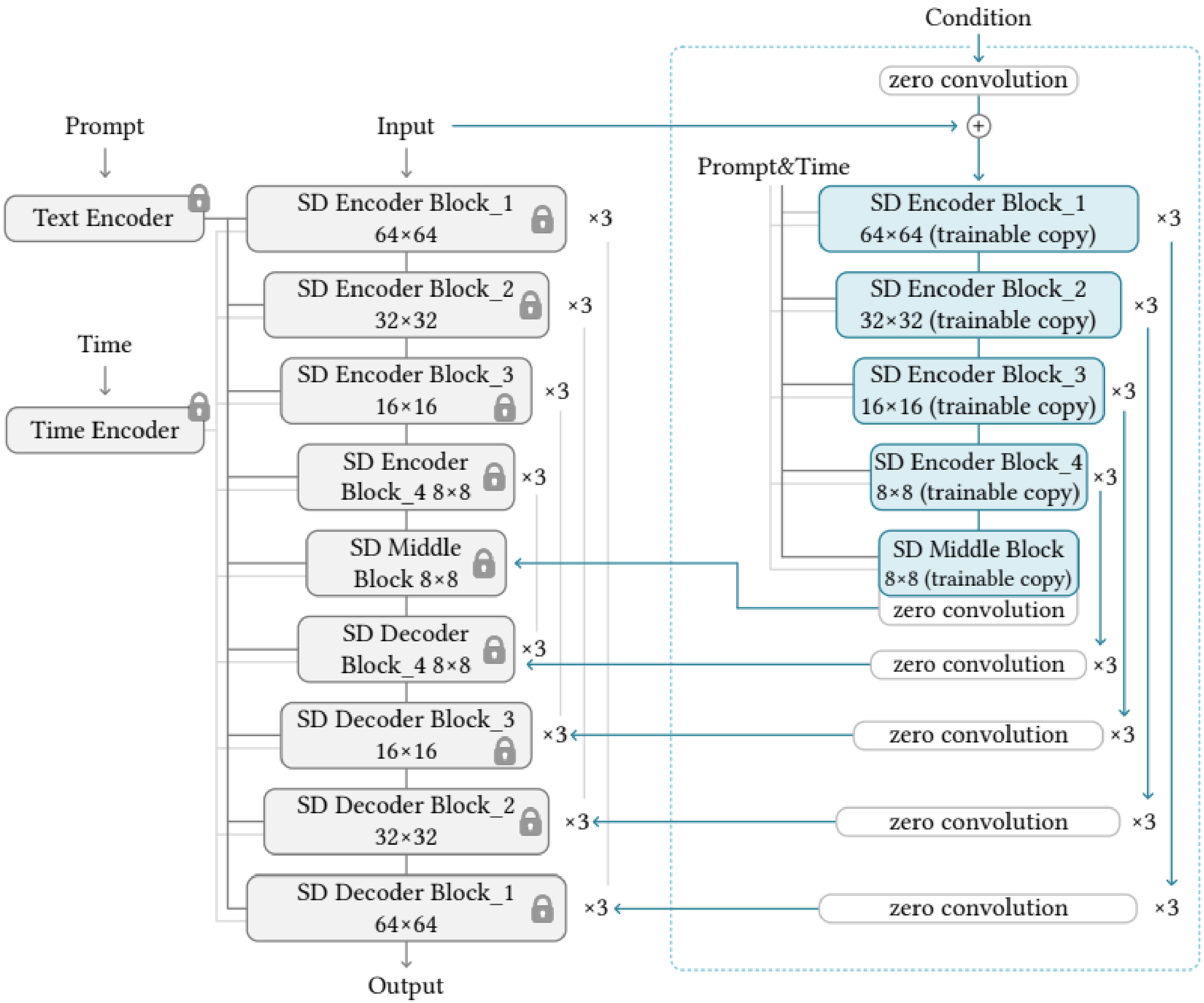
Stable Diffusion

https://github.com/AUTOMATIC1111/stable-diffusion-webui



Adding Conditional Control to Text-to-Image Diffusion Models

https://arxiv.org/abs/2302.05543



(a) Stable Diffusion (b) ControlNet