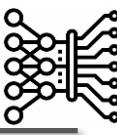


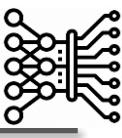
Generative Machine Learning

Generative Model

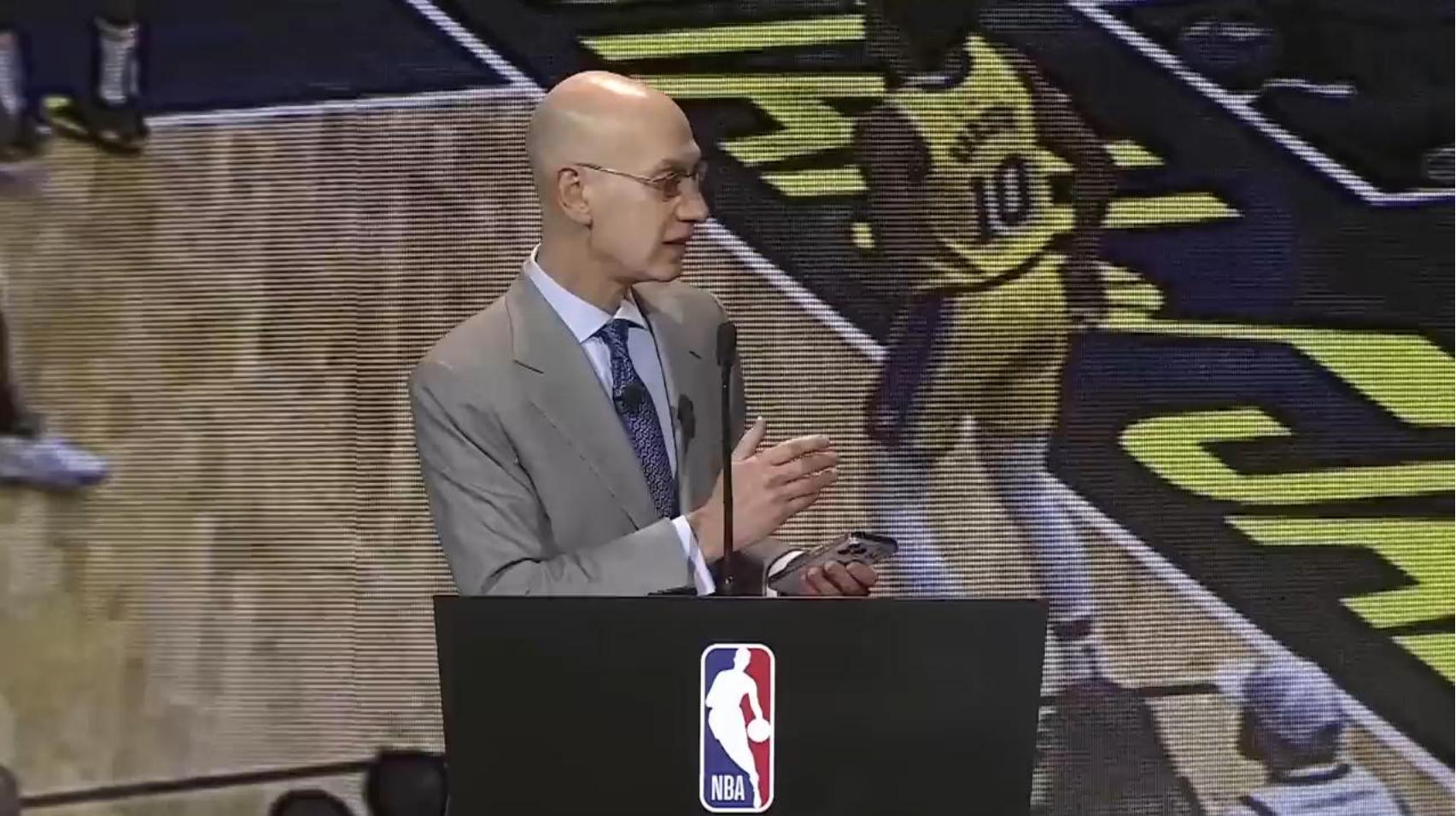
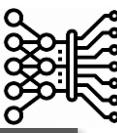


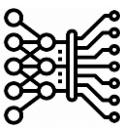
<https://www.youtube.com/watch?v=gLoI9hAX9dw>

Generative Model



Generative Model





What are some recent and potentially upcoming breakthroughs in deep learning?

[Answer](#)[Request](#)

Follow 131

Comment

Share 11

Downvote

...

2 Answers

**Yann LeCun**, Director of AI Research at Facebook and Professor at NYU

Written Jul 29 · Upvoted by Joaquin Quiñonero Candela, studied at Machine Learning and Nikhil Garg, I lead a team of Quora engineers working on ML/NLP problems

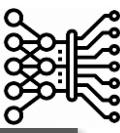


There are many interesting recent development in deep learning, probably too many for me to describe them all here. But there are a few ideas that caught my attention enough for me to get personally involved in research projects.

The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This is an idea that was originally proposed by Ian Goodfellow when he was a student with Yoshua Bengio at the University of Montreal (he since moved to Google Brain and recently to OpenAI).

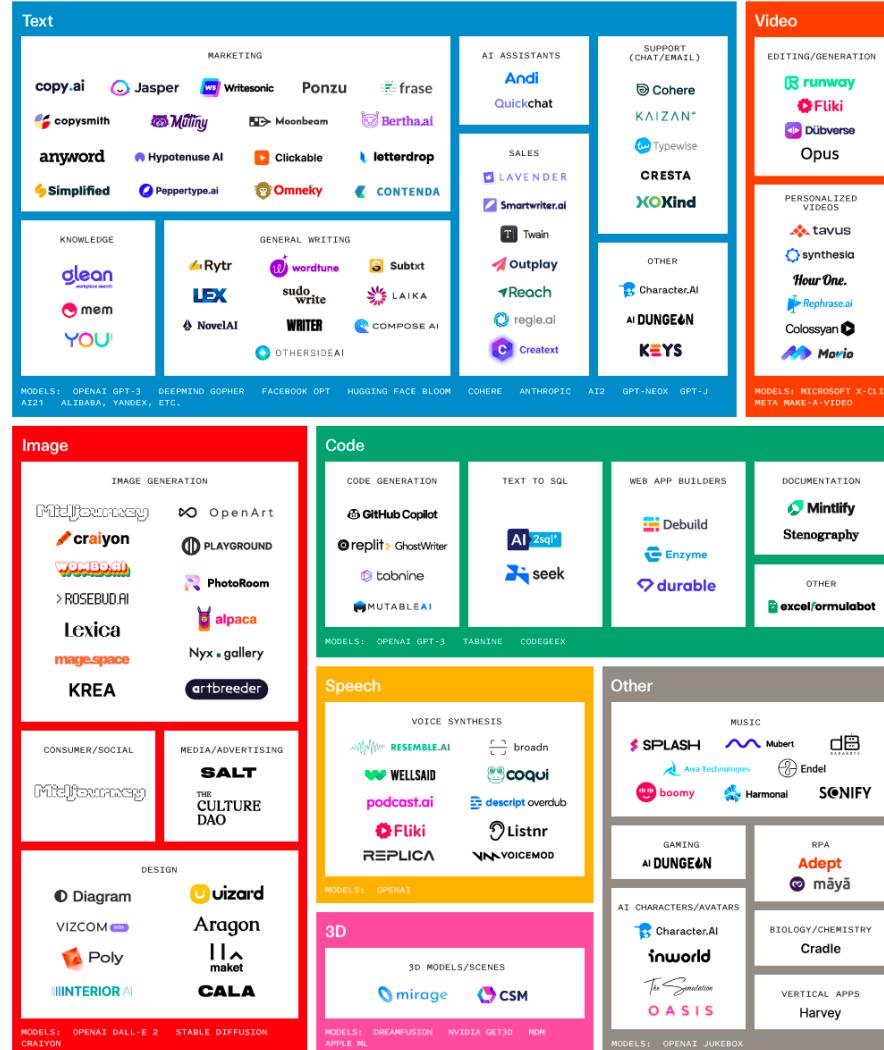
This, and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion.

Generative Model

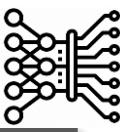


The Generative AI Application Landscape v2

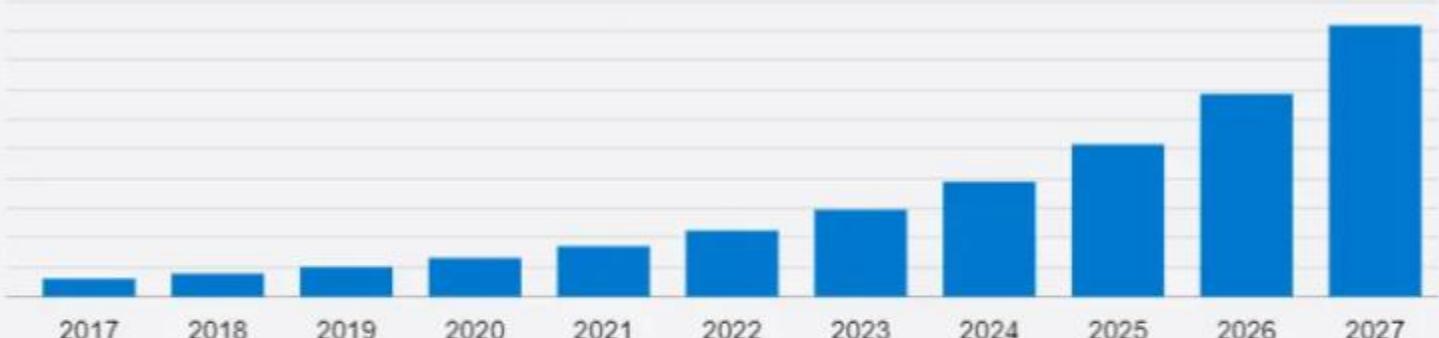
A work in progress



Generative Model



Market Size Outlook (usd Million)



2017 : USD 3,019.66



Year-over-Year
growth rate of 2023



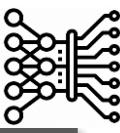
CAGR 2022-2027



Growth Momentum



Market size
growth

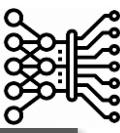


❖ Concepts

눈을 감고 황금 들녘을 상상해보라. 근사한 영상이 떠오른다면 머리 속에 있는 ‘생성 모델 generative model’이 작용한 탓이다. 사람의 생성 모델은 세상에 나타나는 현상을 오랫동안 지켜보면서 학습한 결과이다. 만일 기계 학습이 훈련집합을 사용하여 비슷한 생성 모델을 구축할 수 있다면 강한 인공지능^{strong AI}에 한발 다가설 수 있다. 왜냐하면 생성 모델은 분별 모델에 비해 데이터 생성 과정에 대한 보다 깊은 이해를 필요로 하기 때문이다[Karpathy2015].

❖ GAN

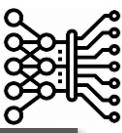
- Data를 만들어내는 **Generator**와 만들어진 data를 평가하는 **Discriminator**가 서로 대립 (**Adversarial**)적으로 학습해가며 성능을 점차 개선해 나가자는 개념



❖ Comparisons

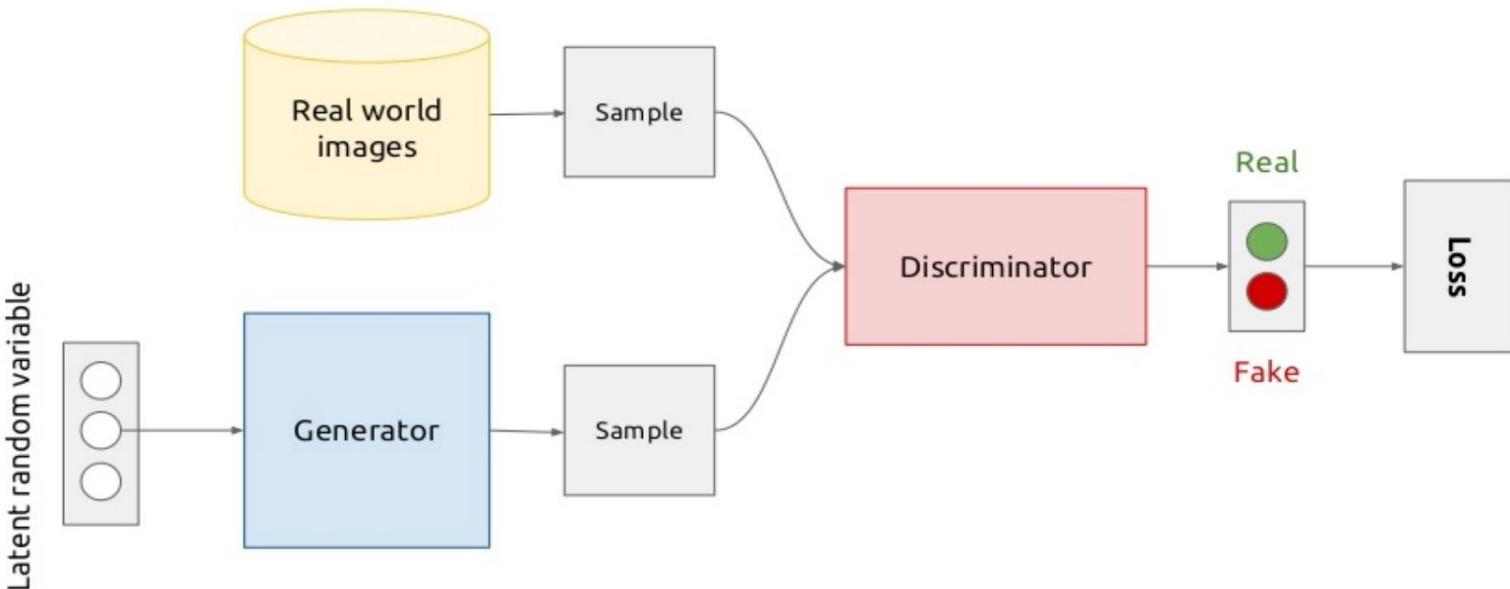
표 4-1 분별 모델과 생성 모델

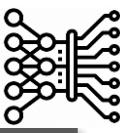
모델	학습 단계가 할 일	예측 단계가 할 일	지도 여부
분별 모델	$P(y x)$ 추정	$f: x \mapsto y$	지도 학습
생성 모델	$P(x)$ 또는 $P(x y)$, $P(x, y)$ 추정	$f: \text{씨앗} \mapsto x$ 또는 $f: \text{씨앗 } y \mapsto x,$ $f: \text{씨앗} \mapsto x, y$	비지도 학습



❖ Basic concepts

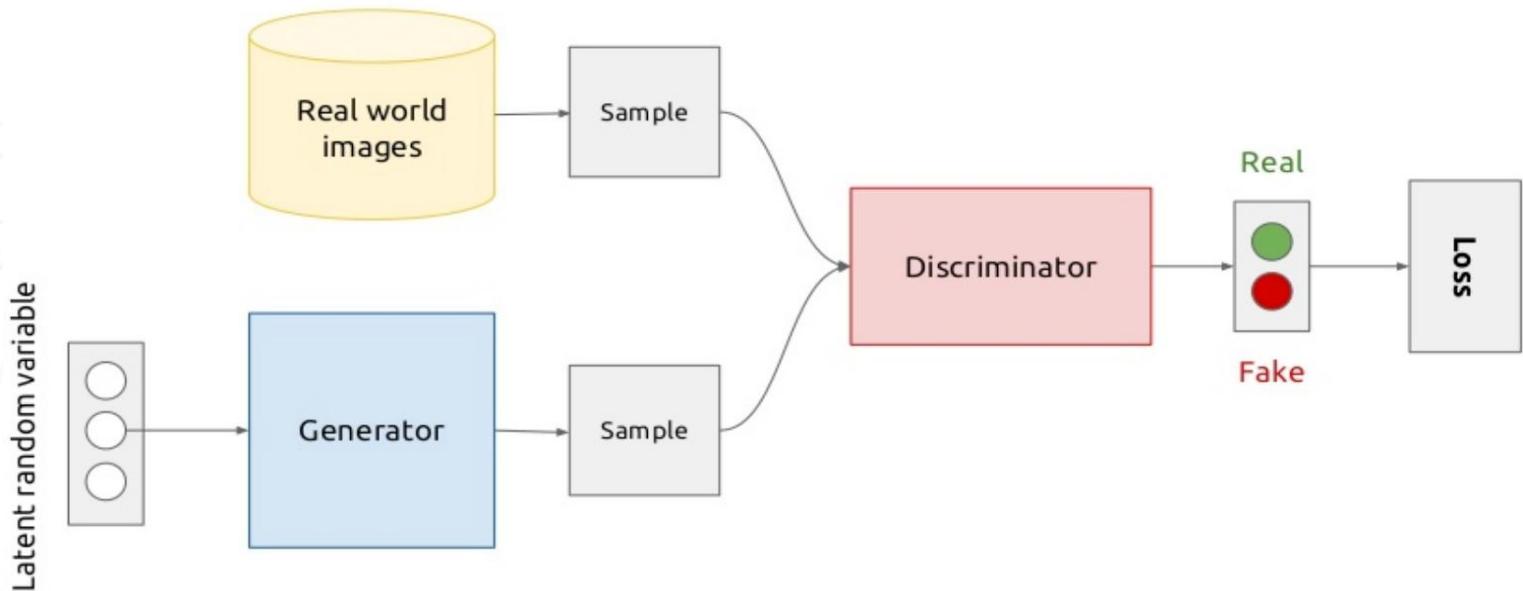
- Discriminator를 학습시킬 때에는 $D(x)$ 가 1이 되고 $D(G(z))$ 가 0이 되도록 학습시킴
 - 진짜 데이터를 진짜로 판별하고, 가짜데이터를 가짜로 판별할 수 있도록
- Generator를 학습시킬 때에는 $D(G(z))$ 가 1이 되도록 학습시킴
 - 가짜 데이터를 discriminator가 구분 못하도록 학습, discriminator를 헷갈리게 하도록

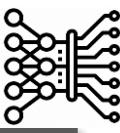




❖ Generator

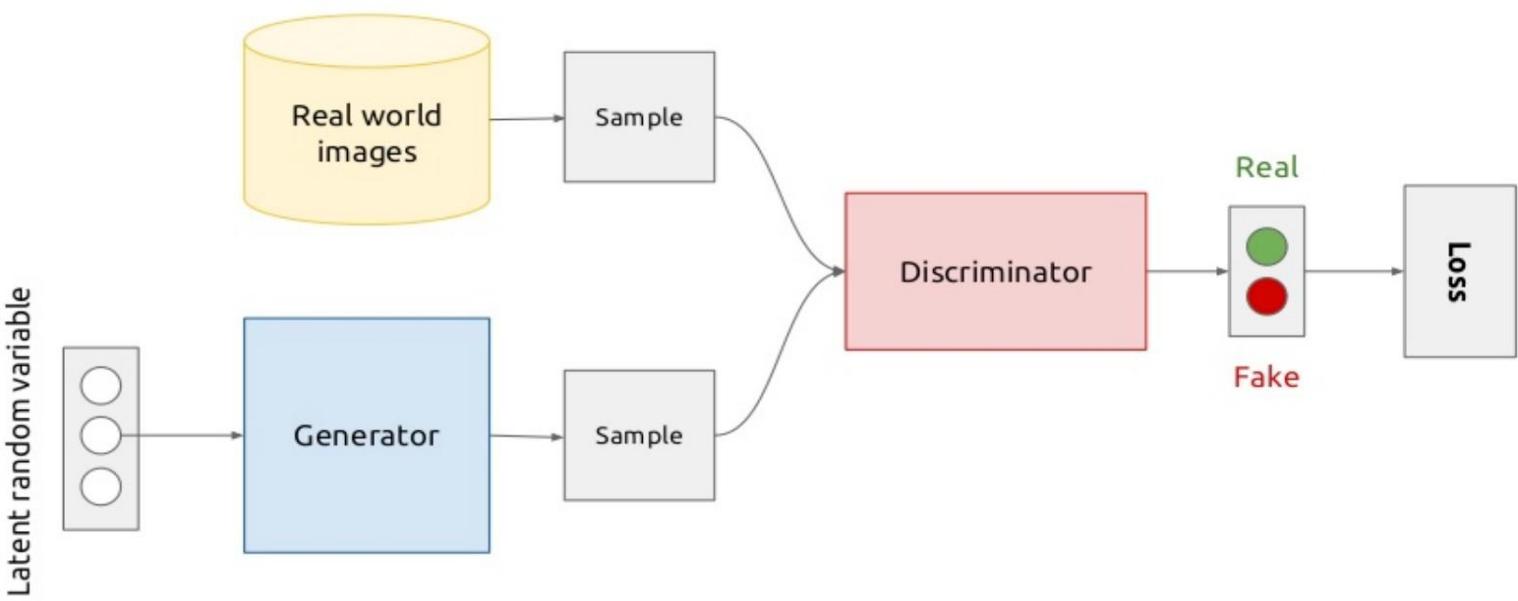
- 주어진 데이터와 유사한 허구데이터(생성 데이터)를 생성
- 저차원 무작위 잡음을 입력 받아 고차원 허구 이미지를 생성
- 실제 이미지를 학습하여 확률분포가 최대한 비슷하도록 허구 이미지를 생성

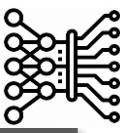




❖ Discriminator

- 허구 데이터인지 실제 데이터인지 구분
 - 입력된 이미지가 실제인지 허구인지 판별
- 허구이미지는 생성망의 학습이 진행됨에 따라 점점 실제 이미지와 유사해짐
 - 판별망은 점진적으로 학습
- 무작위 잡음 벡터 Z를 입력 받아 생성하는 생성망의 결과를 판별

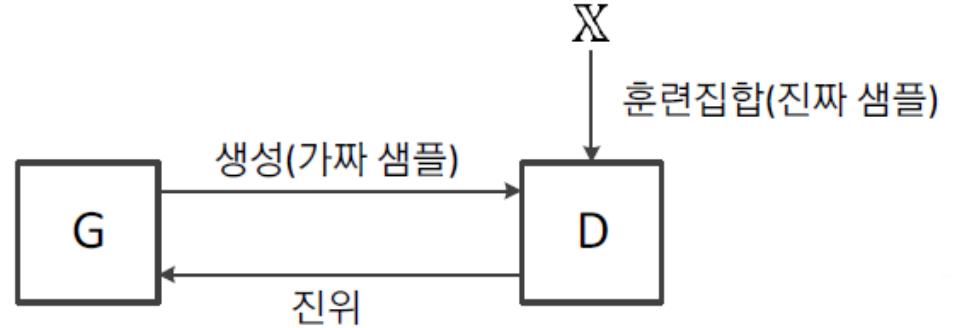




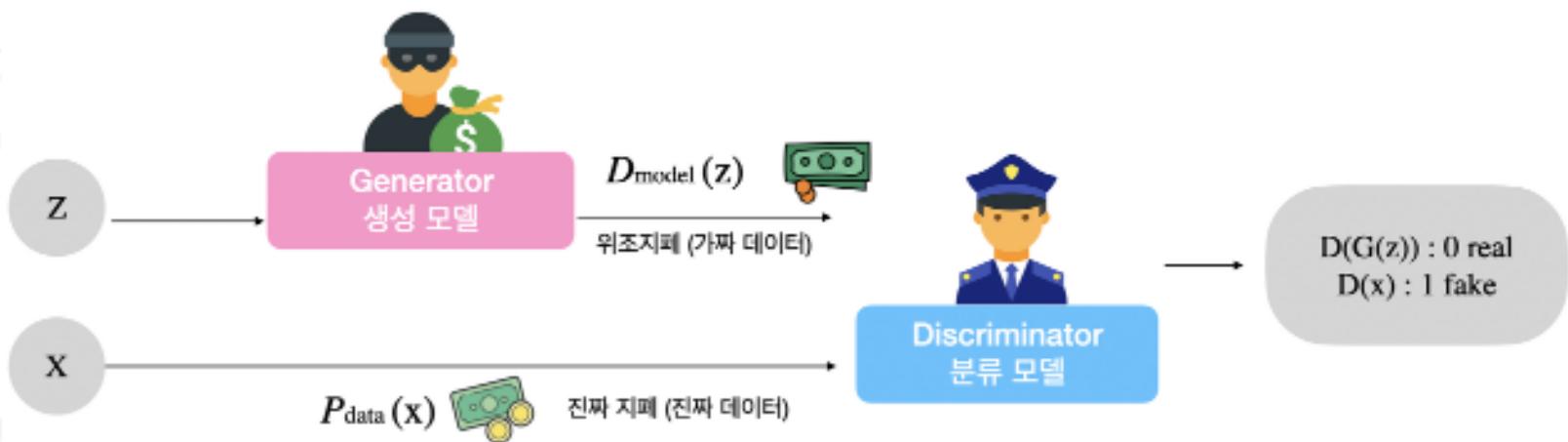
❖ Idea of GAN

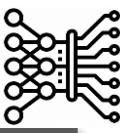
- 생성기 G와 분별기 D의 대립 구도
 - G는 가짜 샘플 생성(위조지폐)
 - D는 가짜와 진짜를 구별(경찰)
- GAN의 목표는 위조지폐의 승리

(G가 만들어내는 샘플을 D가 구별하지 못하는 수준까지 학습)



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

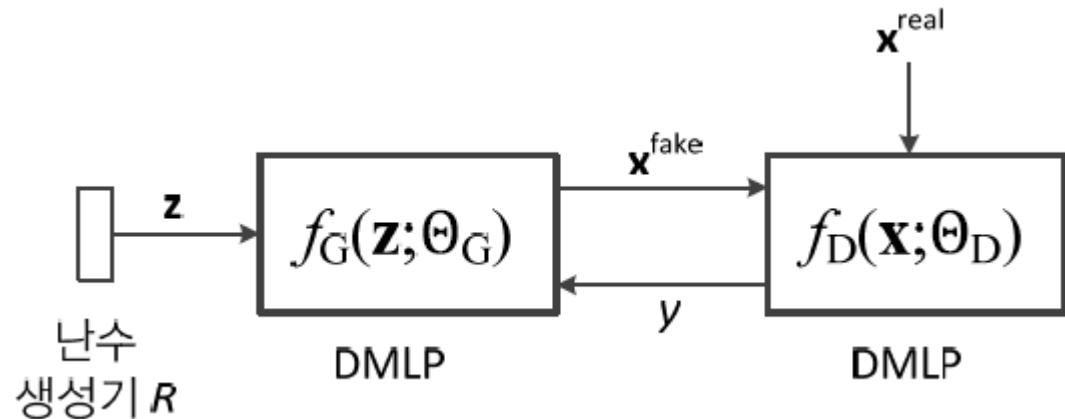




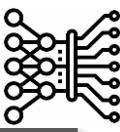
❖ GAN [Goodfellow, 2014]

▪ G와 D를 DMLP로 구현

- G는 $f_G(z; \Theta_G)$, D는 $f_D(x; \Theta_D)$ 로 표기 (Θ_G 와 Θ_D 는 매개변수)
- f_G 는 난수 발생기로 만든 벡터 z 를 입력으로 받아 가짜 영상을 출력
- f_D 는 영상을 입력으로 받아 진짜(1) 또는 가짜(0)를 출력



Structure of GAN



❖ GAN [Goodfellow, 2014]

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k , is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Sample minibatch of m examples $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\mathbf{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D(\mathbf{x}^{(i)}) + \log (1 - D(G(\mathbf{z}^{(i)}))) \right].$$

end for

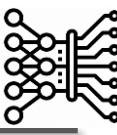
- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(\mathbf{z}^{(i)}))).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Training Procedure for GAN



❖ Notation

- $p_{data}(x)$: 실제 데이터 x 에서 뽑아낸 확률 분포
- $p_z(z)$: 랜덤 노이즈로부터 뽑아낸 확률 분포
- $G(z; \theta_g)$: Generator 함수 (θ_g 를 파라미터로 갖는 뉴럴 네트워크)
- $D(x; \theta_d)$: Discriminator 함수, 출력값은 스칼라 값
 - 입력이 실제 데이터에 왔다면 1을 출력하고 아니면 0을 출력

❖ Loss function

Discriminator는 식을 최대화
Generator는 식을 최소화

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

x 는 실제 데이터로 부터 생성

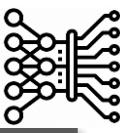
D(real)의 확률

z 는 노이즈로부터 생성

D(fake)의 확률

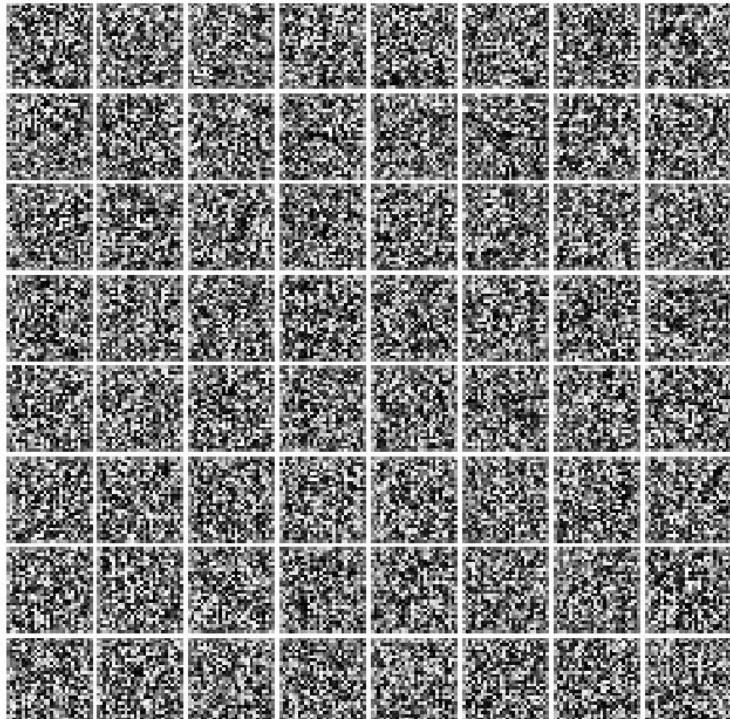
Generator가
생성한 fake

Performance Evaluation

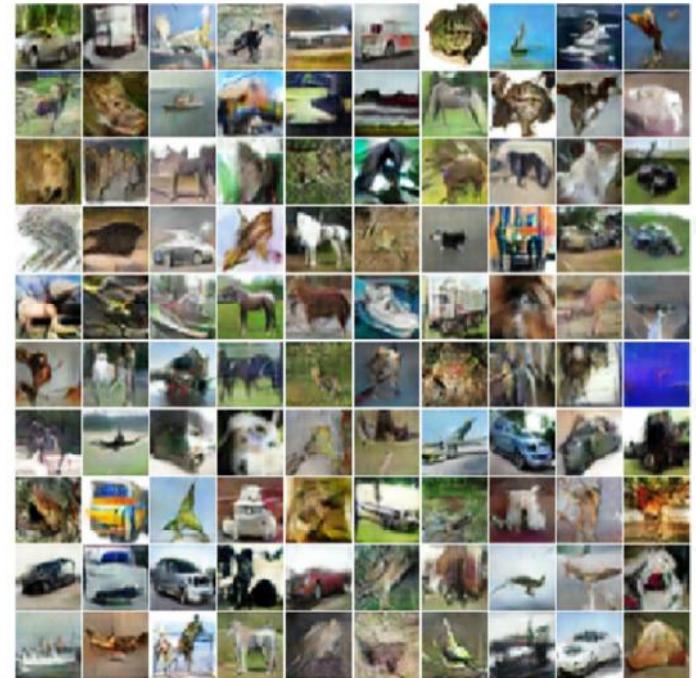


❖ Superior performance

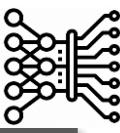
- 사람을 상대로 진짜와 가짜 구별하는 실험에서, MNIST 52.4%, CIFAR-10 78.7% (50%이면 완벽히 속임)



5 8 9 8 9 5 0 2 8 1
4 6 6 6 0 7 1 2 7 8
2 0 1 0 7 4 4 9 5 9
1 7 3 8 3 2 1 6 3 8
0 2 6 0 6 9 3 6 1 6
5 2 9 9 5 8 6 4 4 5
1 5 4 6 7 8 7 9 7 3
7 4 4 7 9 5 6 8 1 1
1 3 9 0 9 1 1 1 8 7
2 6 8 3 0 9 7 1 4 1



Example samples



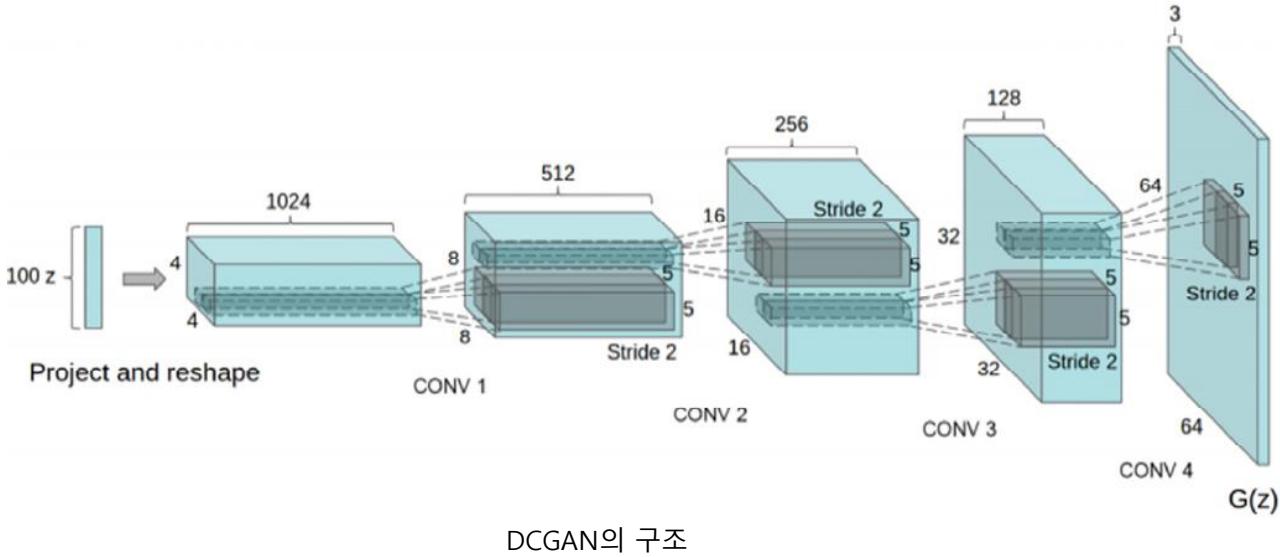
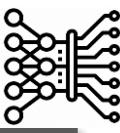
❖ GAN의 단점

- 학습이 매우 불안정하고, learning rate의 영향을 많이 받음
- Learning rate를 높게 주면 거의 모든 경우 학습 실패

❖ 개선된 GAN

- [Salimans2016]
 - 특징 매칭, 가상 배치 정규화, 미니배치 분별 등 기법 적용
 - $P(\mathbf{x})$ 대신 $P(\mathbf{x}, \mathbf{y})$ 를 추정하여 레이블이 있는 샘플을 생성 → 준지도 학습에 활용
- DCGAN
 - DMLP 대신 CNN을 사용
 - CNN과 데이터 흐름이 반대. 즉 벡터를 입력 받아 $3*64*64$ 영상을 출력
 - GAN 모델을 만들 때 고려할 수 있는 가이드라인 제시
 - 거의 대부분의 상황에서 일반 GAN보다 학습이 잘 되는 장점

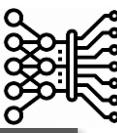
Structure of DCGAN



DCGAN의 구조

- 풀링 레이어를 삭제하고 컨볼루션 레이어 혹은 transpose 컨볼루션 레이어를 통해 특징 맵 크기 감소 / 증가
- Generator와 discriminator에 배치 정규화 사용
(Generator의 출력 레이어와 discriminator의 입력 레이어는 사용 x)
- Fully-connected 레이어 삭제, 대신 Global average pooling 사용
- Generator에는 활성 함수로 ReLU를 사용하고 마지막 레이어는 tanh
- Discriminator에는 leaky ReLU 사용

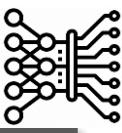
Training Procedure for DCGAN



- ❖ Model is not producing high quality samples via simply overfitting/memorizing training examples
 - 한마디로 Generator가 학습 이미지를 단순히 외우거나 베낀게 아니어야 함
- ❖ "Walking in the latent Space"
 - G의 input z의 공간인 latent Space에서 살짝 이동한다 하더라도 급작스러운 변화가 일어나지 않고 물흐르듯 부드러운 변화를 보여줘야 한다.
- ❖ LSUN Dataset



Training Procedure for DCGAN



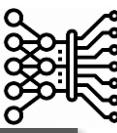
❖ LSUN Dataset



Random filters

Trained filters

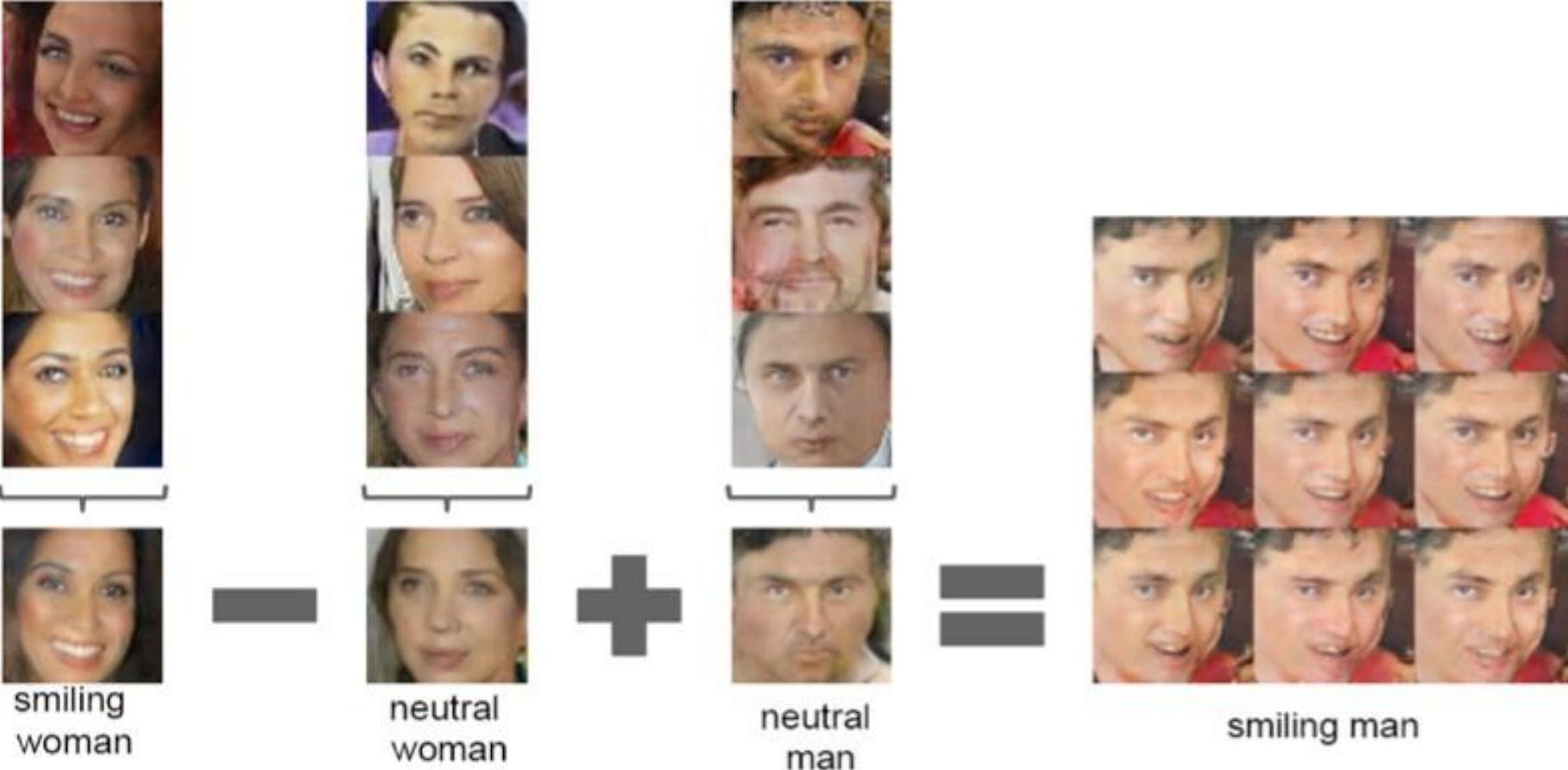
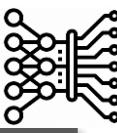
Training Procedure for DCGAN



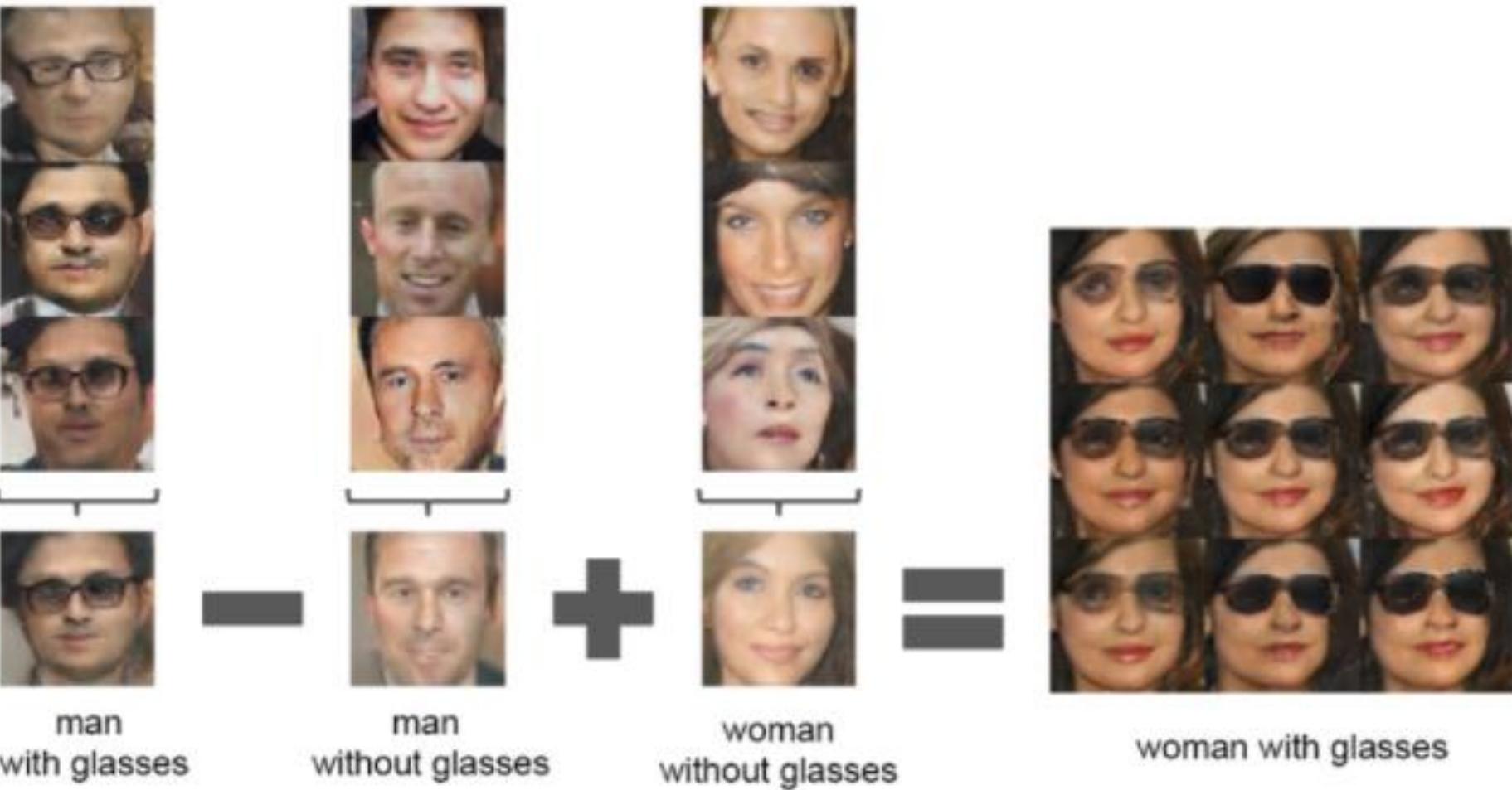
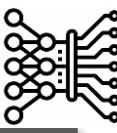
❖ GAN을 Semi-supervised 학습에 응용 가능

- Unsupervised하게 GAN을 학습 시킨 뒤, discriminator의 중간 레이어 출력값을 통해 특징을 추출
- 이미지도 벡터 연산 가능
- 생성된 이미지를 학습 데이터로 사용 가능

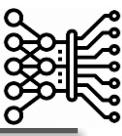
Training Procedure for DCGAN



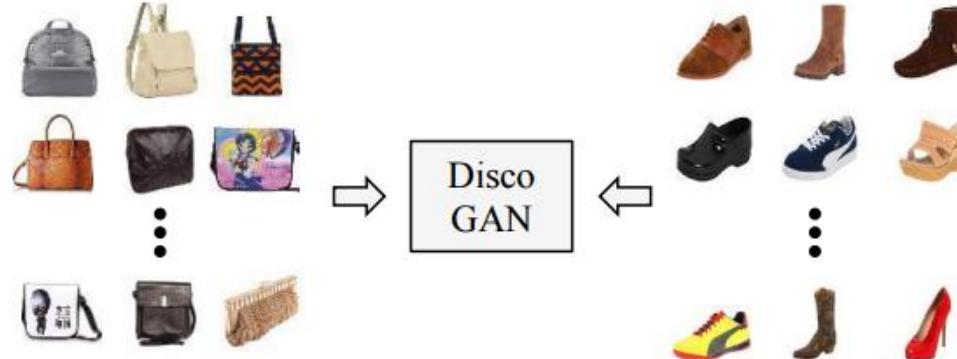
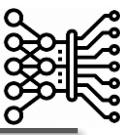
Training Procedure for DCGAN



Training Procedure for DCGAN



왼쪽을 보고 있는 얼굴을 만들어내는 z 들의 평균 vector $z(\text{left})$ 과 오른쪽을 보고 있는 얼굴에 대응하는 z 들의 평균 vector $z(\text{right})$ 를 계산하고 이 두 벡터의 사이를 interpolating하여 Generator에 입력



(a) Learning cross-domain relations **without any extra label**



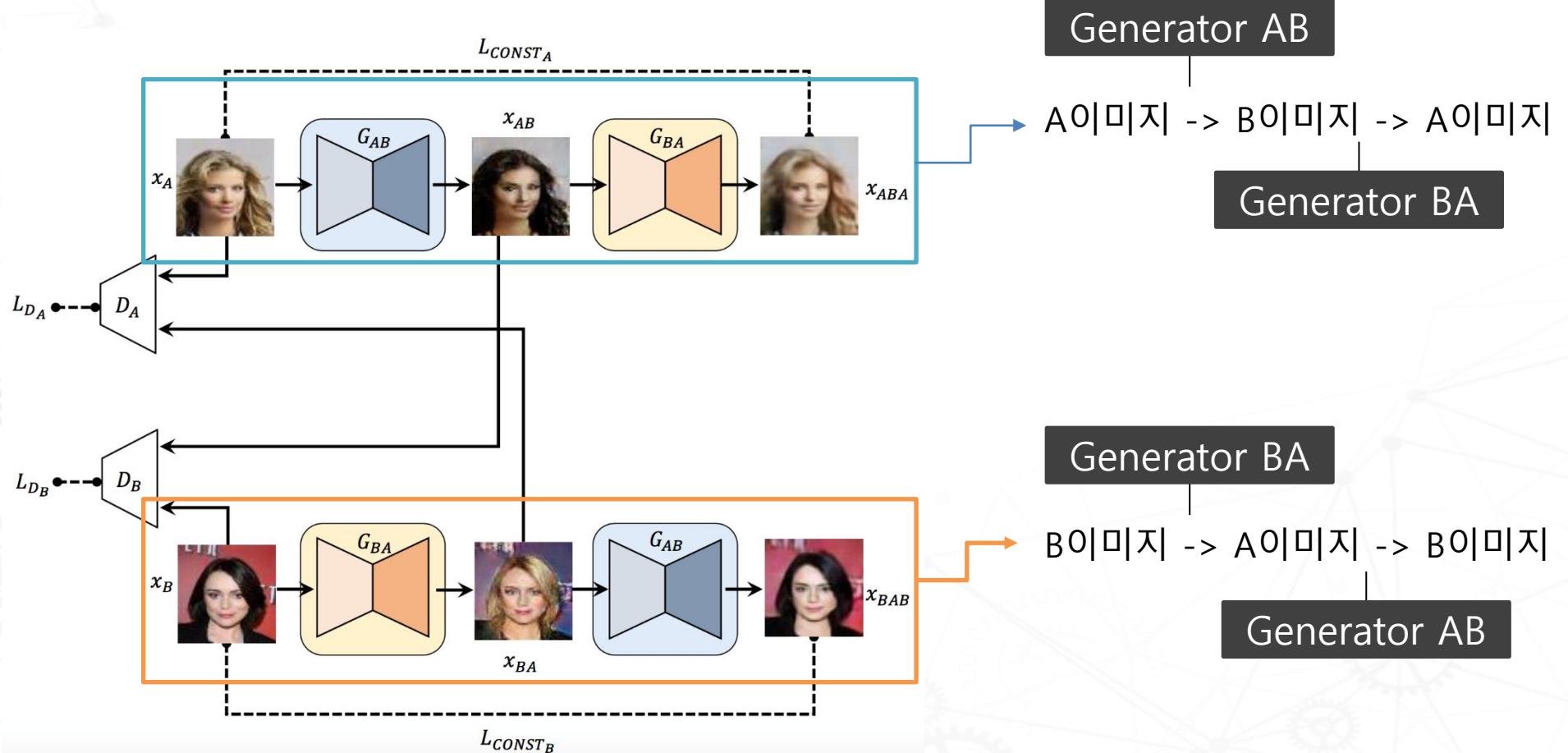
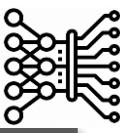
(b) Handbag images (input) & **Generated** shoe images (output)

구두스타일과 비슷한 스타일의 가방을 내보냄

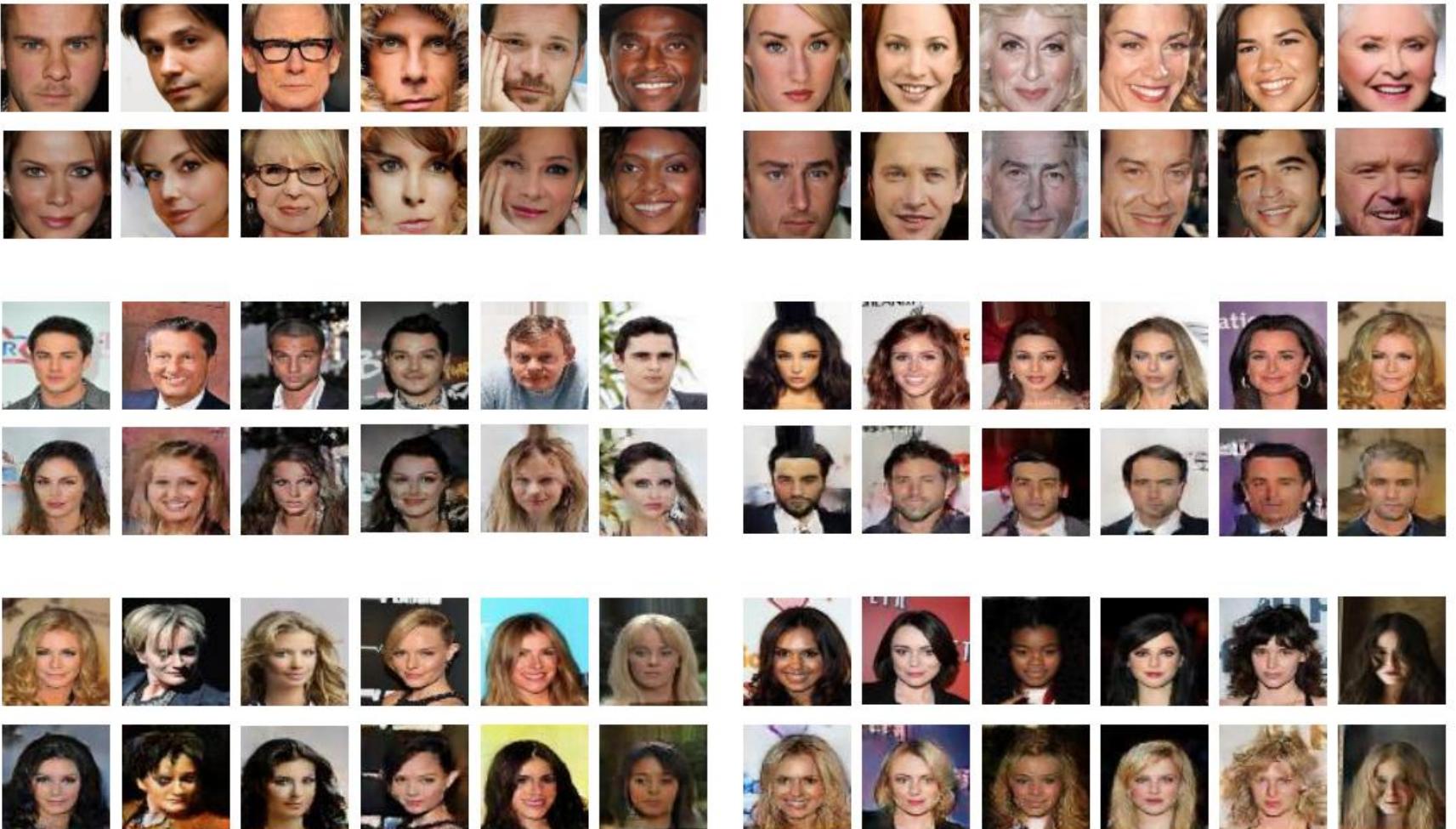
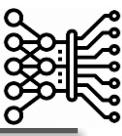


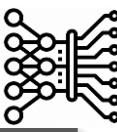
(c) Shoe images (input) & **Generated** handbag images (output)

Structure of DiscoGAN



Training Procedure of DiscoGAN





Cycle GAN



한 도메인의 이미지를 다른 도메인으로 해석하는
IMAGE TO IMAGE TRANSLATION 모델



Photograph



Monet



Van Gogh

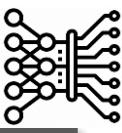


Cezanne

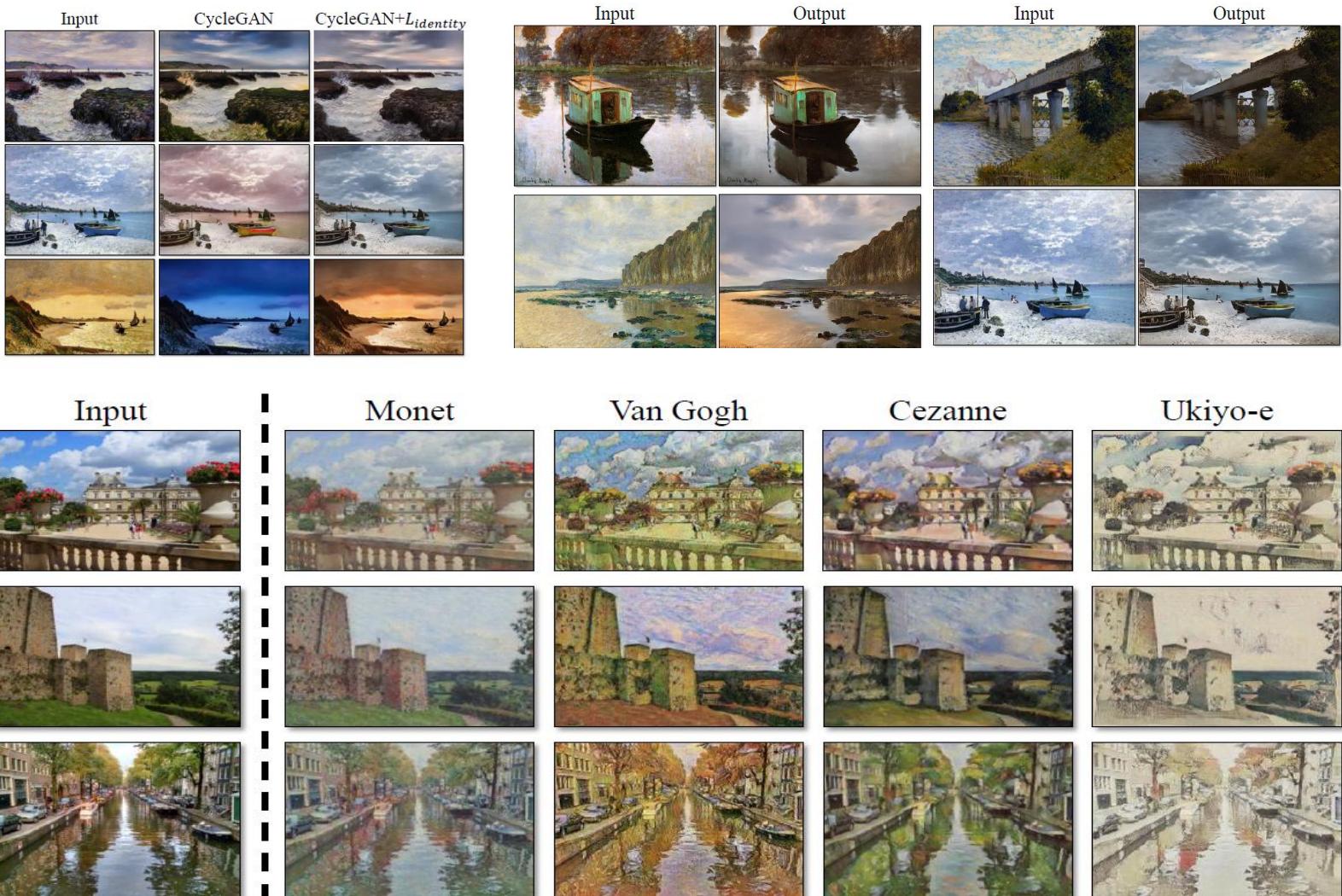
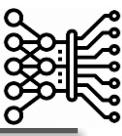


Ukiyo-e

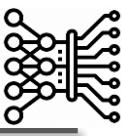
CycleGAN



Training Procedure of CycleGAN



Training Procedure of CycleGAN



Input



Output



Input



Output



Input



Output



horse → zebra



zebra → horse

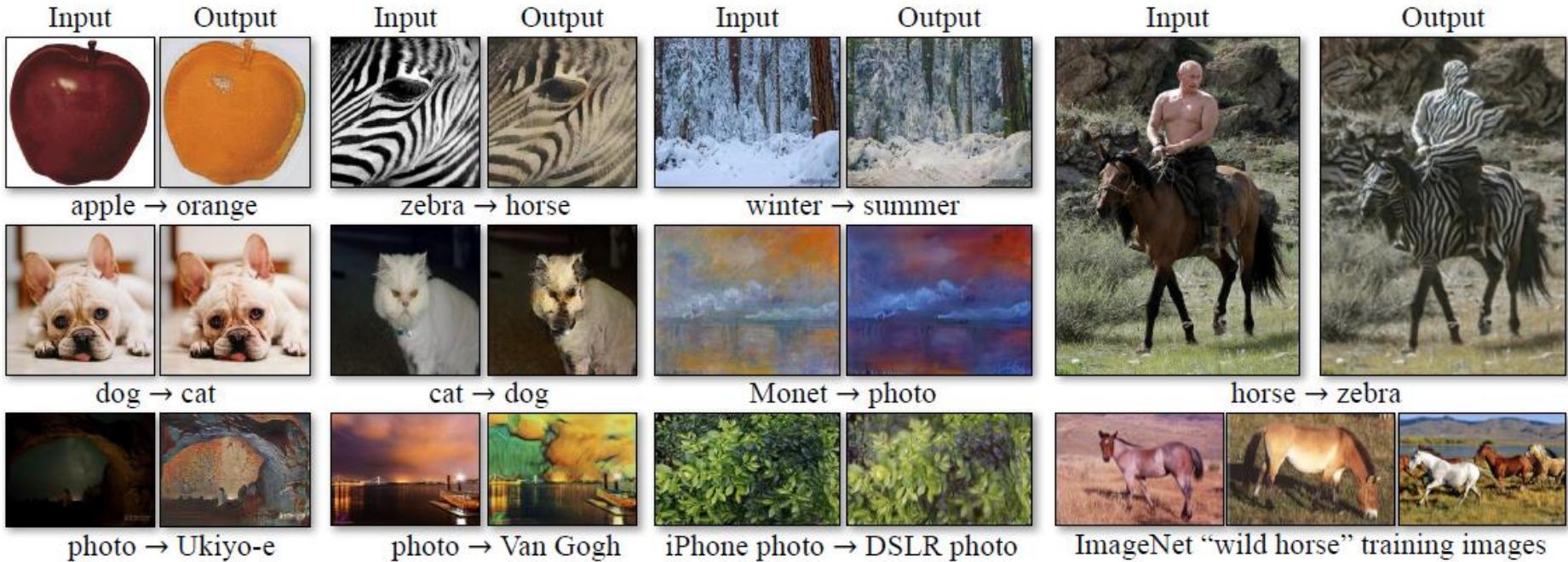
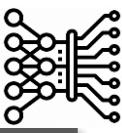


winter Yosemite → summer Yosemite

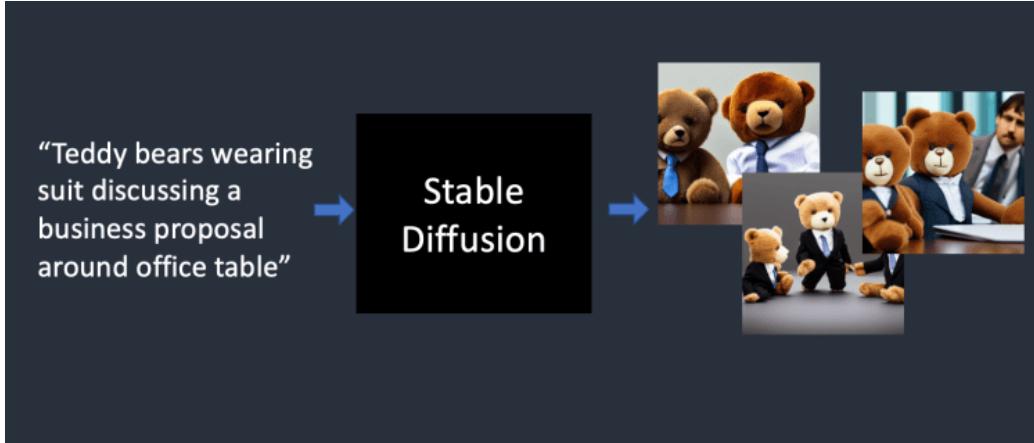
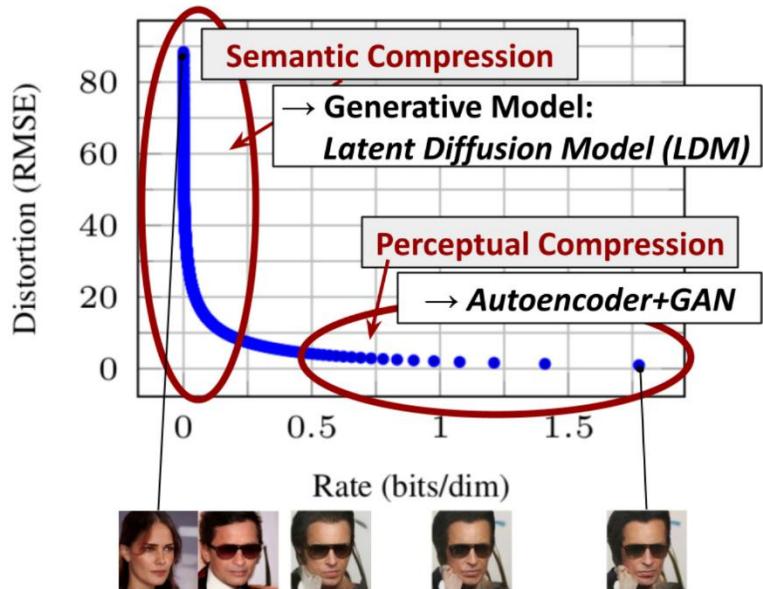
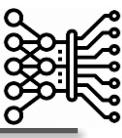


summer Yosemite → winter Yosemite

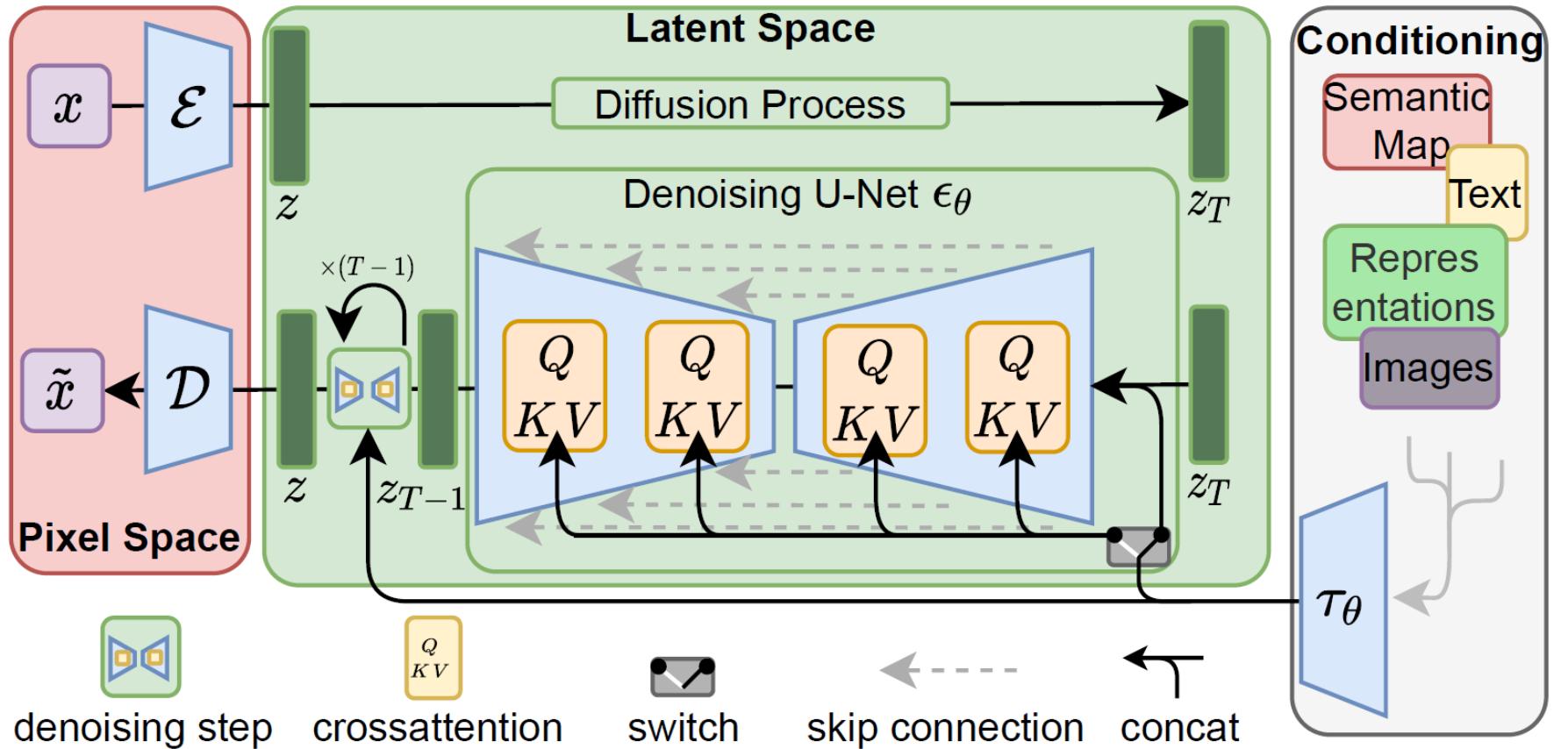
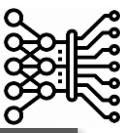
Training Procedure of CycleGAN

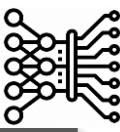


Stable Diffusion



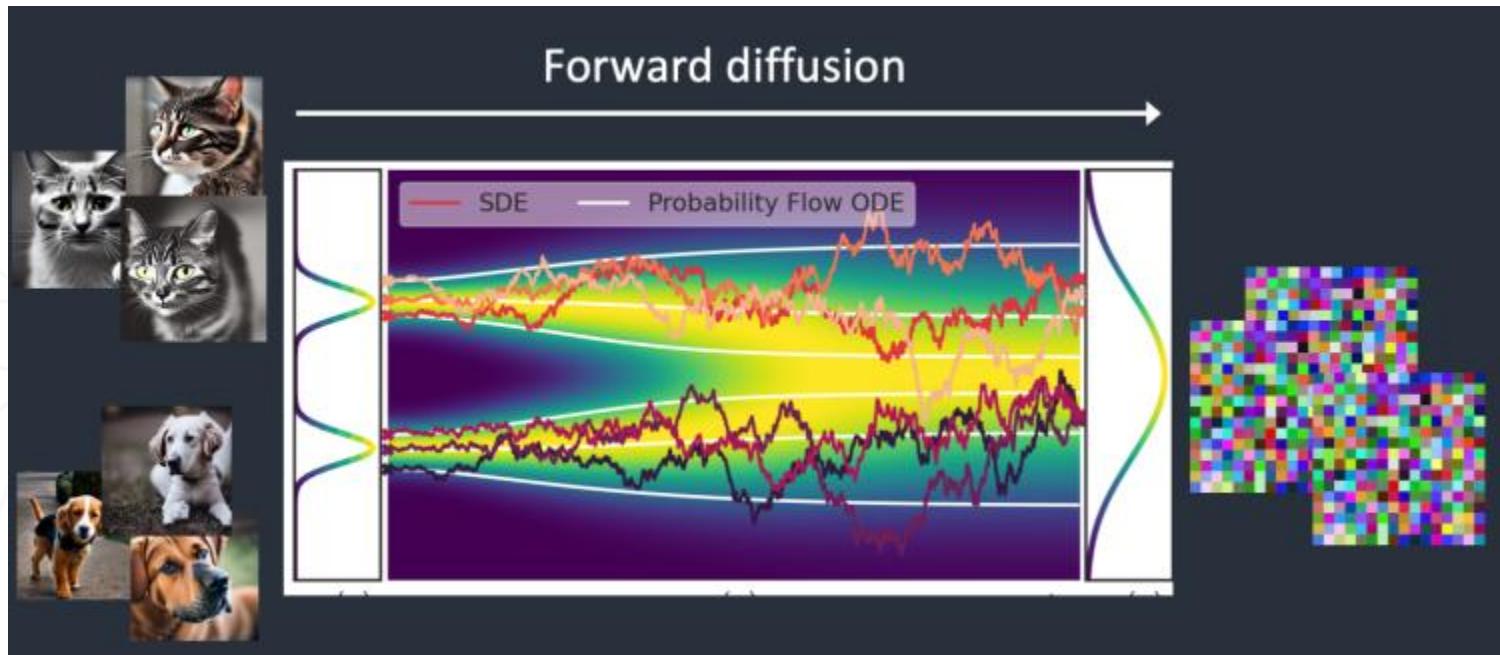
Stable Diffusion



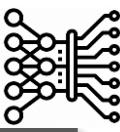


❖ 순방향 확산 (forward diffusion)은 이미지를 잡음으로 바꾸는 과정

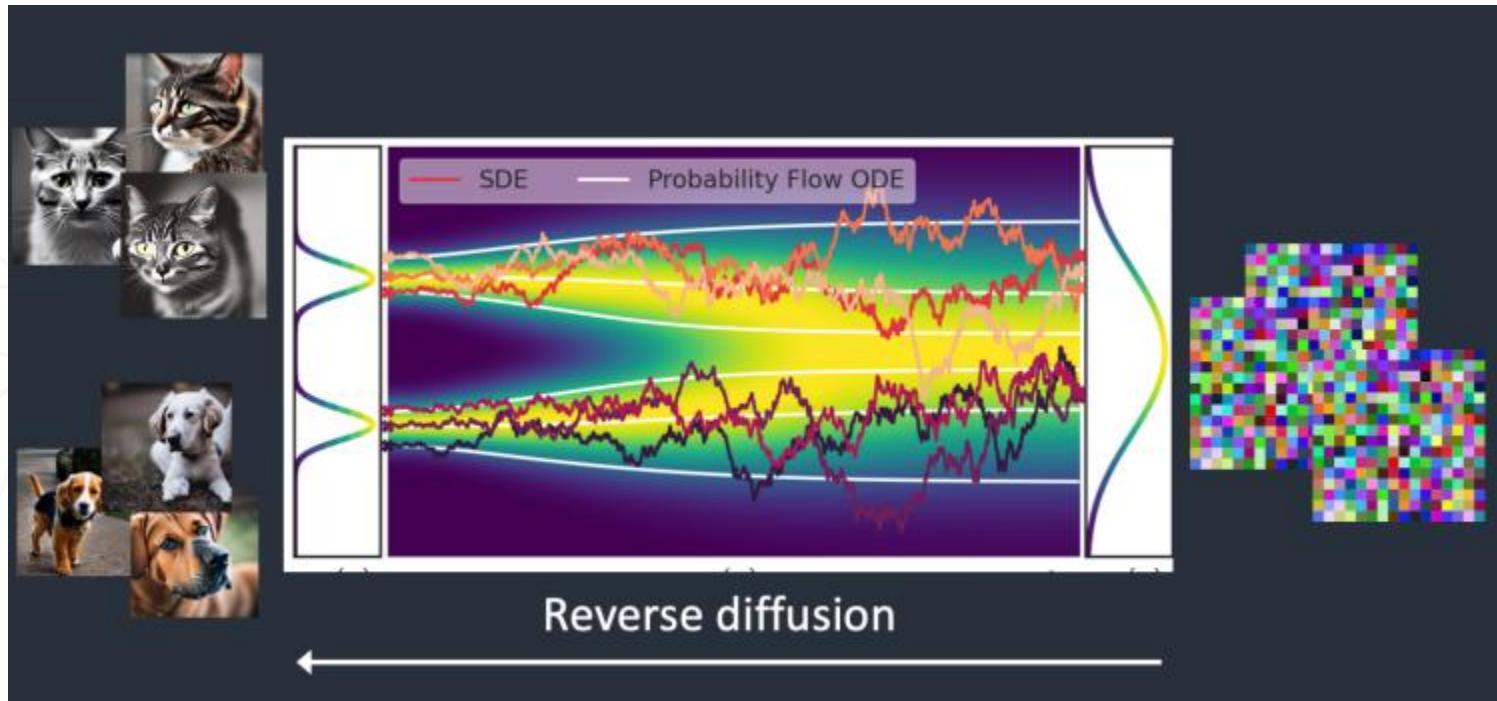
- 학습용 이미지에 점차적으로 잡음을 추가하여, 점점 아무런 특징이 없는 잡음 이미지로 바꿔감
- 아무리 예쁜 이미지도 순방향 프로세스를 거치면 오른쪽에 있는 잡음 이미지가 되어, 결국 강아지 그림인지 고양이 그림인지도 알 수 없게 됨



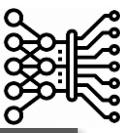
Stable Diffusion



- ❖ 역방향 확산 (reverse diffusion)은 잡음을 이미지로 바꾸는 과정
 - 확산(diffusion)을 뒤집기 위해서는 이미지에 얼마나 많은 잡음이 들어있는지 알아야 함
 - 잡음 예측기(noise predictor), U-Net을 이용하여 추가된 잡음을 예측하도록 학습

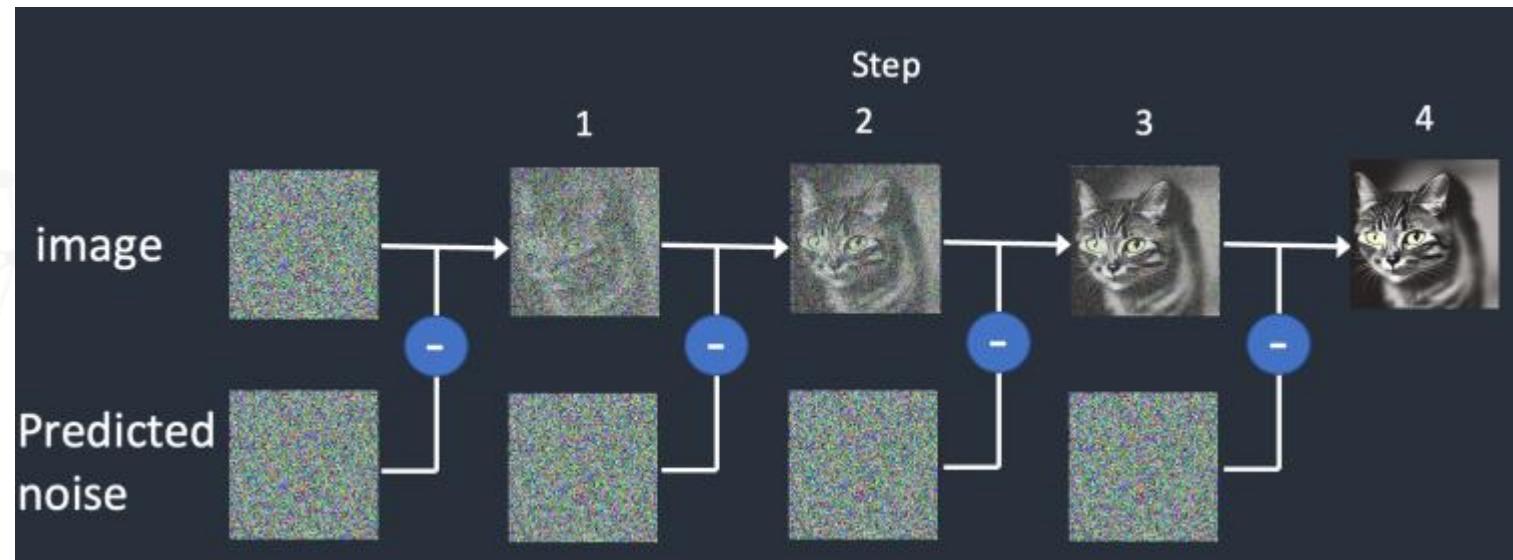


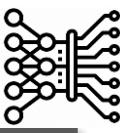
Stable Diffusion



❖ 역방향 확산 (reverse diffusion)은 잡음을 이미지로 바꾸는 과정

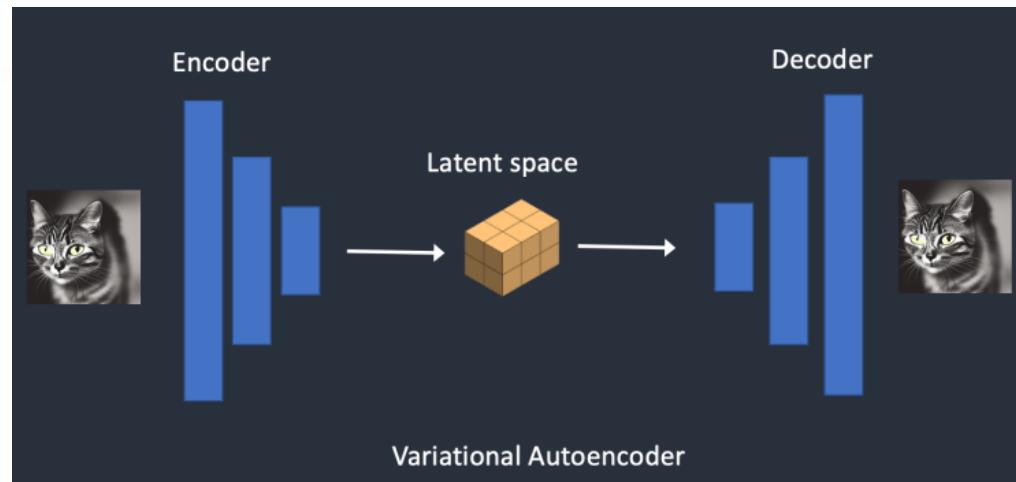
- 학습용 이미지(예: 고양이 이미지)를 선택
- 무작위 잡음 이미지를 생성
- 학습용 이미지에 정해진 단계만큼 무작위 잡음 이미지를 추가하여 손상
- 잡음 예측기에게 잡음 추가량을 학습, 가중치를 조정하고 정답을 보여주는 방식으로 작업이 수행



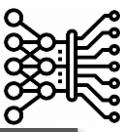


❖ Technical Problem & Solution

- 이미지 공간에서 이렇게 되면 계산이 매우 매우 느림
 - 현존하는 최고의 GPU를 사용한다고 해도, 하나의 GPU로는 계산이 불가능
- 이러한 문제를 잠재 확산 모델(latent diffusion model)을 사용하여 해결
 - 고차원의 이미지 공간에서 작동시키는 것이 아니라, 먼저 잠재 공간(latent space)으로 이미지를 압축한 뒤 연산을 시행
 - 이미지를 잠재 공간으로 압축하는 데는 가변 자동 인코더(variational autoencoder)라는 기법을 사용
- 잡음 이미지를 생성한다고 했지만, 실제로는 잠재 공간상의 무작위 텐서(random tensor, latent noise)를 생성
 - 또한 잡음을 사용해 이미지를 손상시키는 게 아니라, 잠재 공간상의 이미지 표현을 잠재 잡음(latent noise)을 사용해 손상시키는 원리

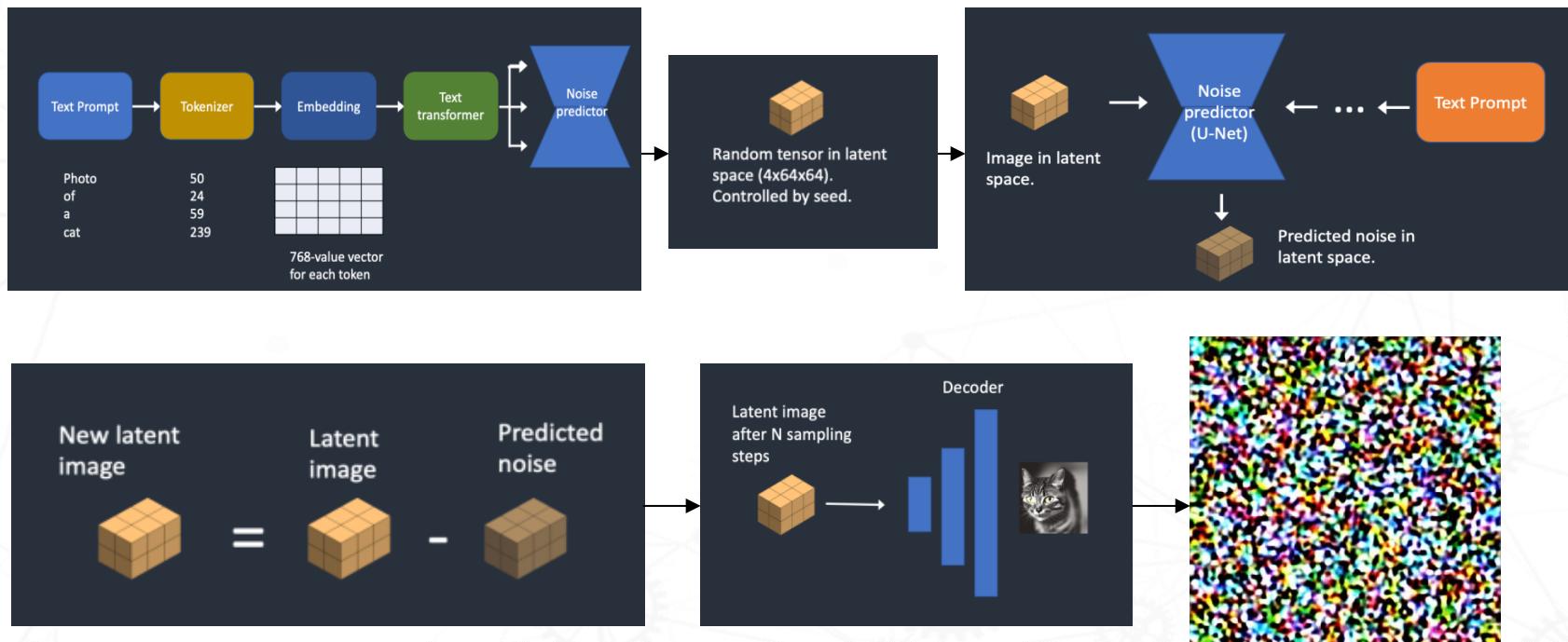


Stable Diffusion

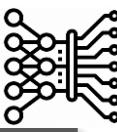


❖ Reverse diffusion for latent space

- 무작위 잠재 공간 행렬을 생성
- 잡음 예측기(noise predictor)가 잠재 행렬의 잡음을 예측
- 예측된 잡음을 잠재 행렬에서 제거
- 지정한 샘플링 단계(sampling step)에 이를 때까지 2,3 단계를 반복
- VAE 디코더(decoder)가 잠재 행렬로부터 최종 이미지로 변환

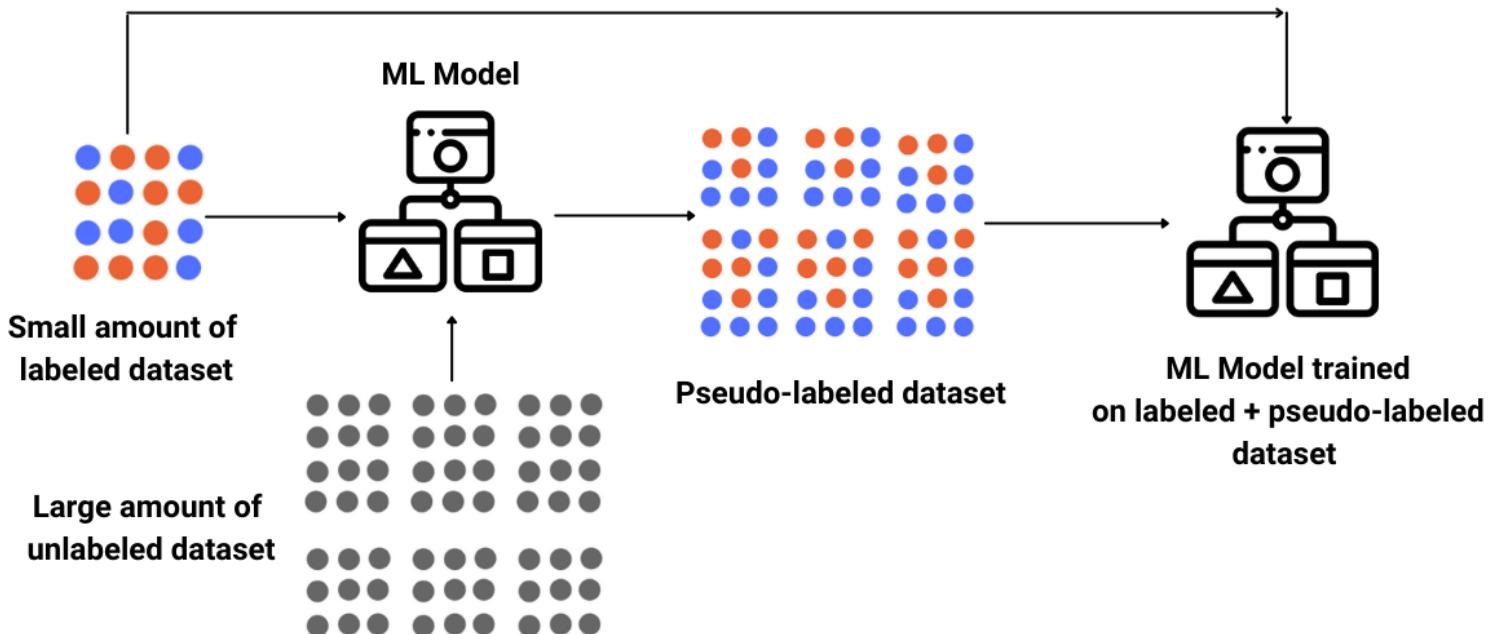


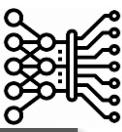
Semi-Supervised Learning



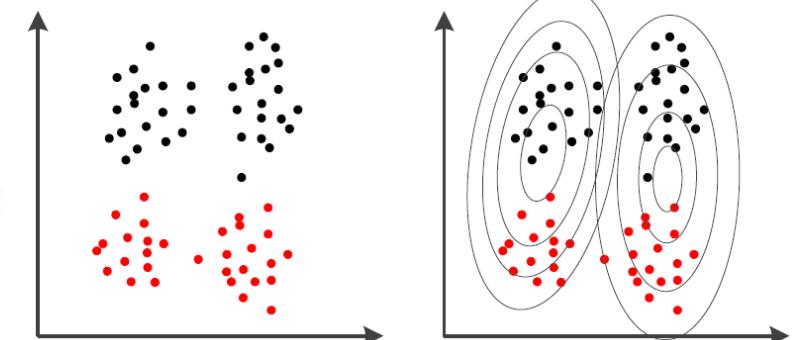
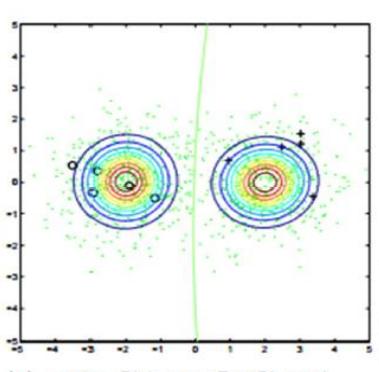
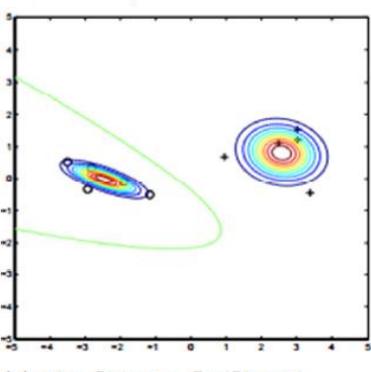
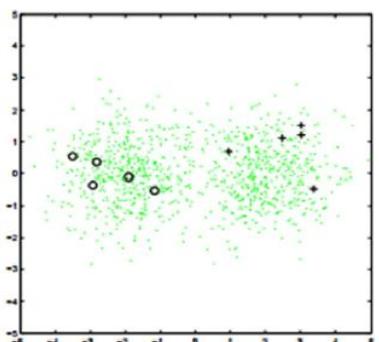
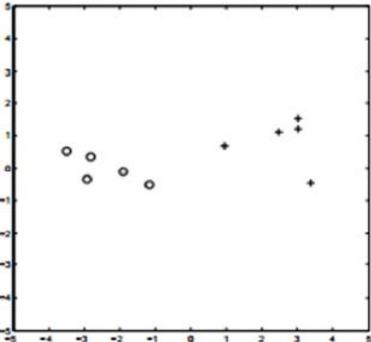
❖ Training set

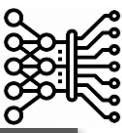
- With label, $\mathbb{X}_l = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}, \mathbb{Y}_l = \{y_1, y_2, \dots, y_n\}$
- Without label, $\mathbb{X}_u = \{\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \dots, \mathbf{x}_{n+m}\}$
- $m \gg n$, there are far more unlabeled samples





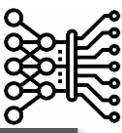
- ❖ Does unlabeled data really help for training
 - Improve performance if you ‘use the right model for the given structure of data’





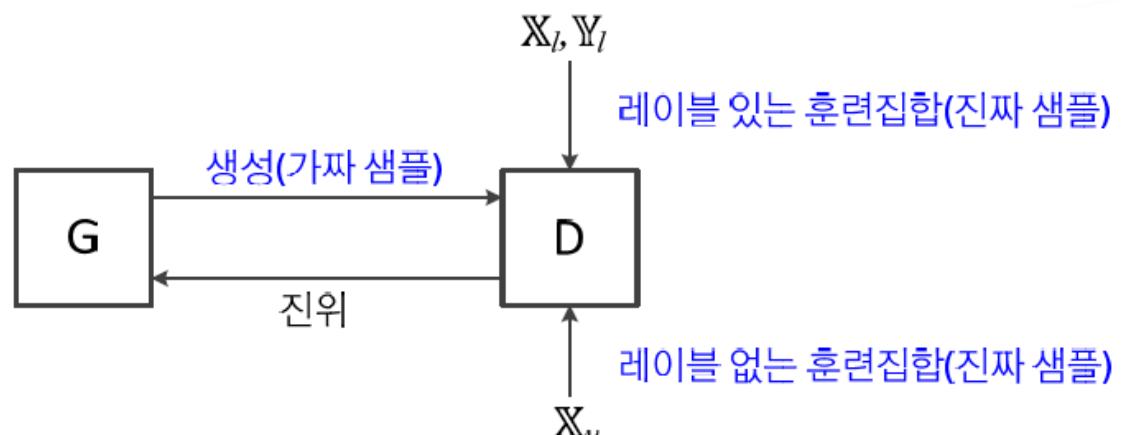
❖ Previous generative model

1. \mathbb{X}_l 과 \mathbb{X}_u 를 가지고 $P(\mathbf{x})$ 를 추정 (\mathbb{Y}_l 을 무시한 비지도 학습)
2. 가우시안 혼합을 사용
3. 각각의 가우시안은 자신에 속한 샘플을 보고 부류를 선택
4. 각 부류는 자신에 속한 가우시안으로 $P(\mathbf{x}|y)$ 를 추정
 - 적용에 한계
 - 가우시안 분포에 맞는 데이터로 국한해야 함
 - 가우시안 혼합을 적용할 때 가우시안 개수를 정확하게 알기 어려움

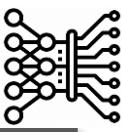


❖ Recent generative model

- 생성 모델 GAN을 사용
- 가짜 샘플에 해당하는 $c+1$ 이라는 레이블을 추가로 사용
- 분별기 D의 목적함수는 세 가지 항을 가짐
 - 가짜 샘플을 $c+1$ 에 배정하는 항
 - \mathbb{X}_u 샘플이 $c+1$ 에 배정되는 것을 막는 항
 - \mathbb{X}_l 샘플을 해당 부류로 배정하는 항



Semi-supervised learning using GAN



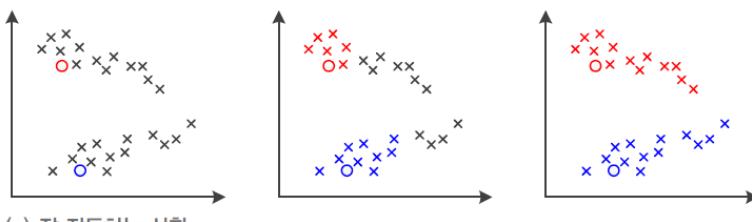
❖ Self learning (자가 학습)

알고리즘 7-1 자가 학습

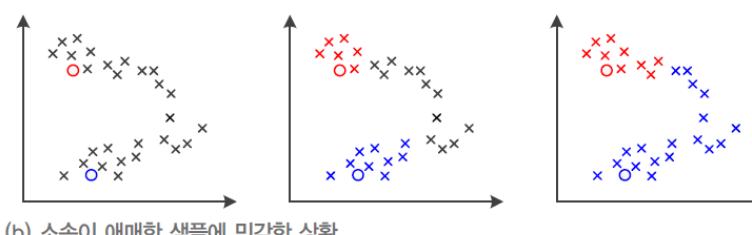
입력: $\mathbb{X}_l = \{x_1, x_2, \dots, x_n\}$, $\mathbb{Y}_l = \{y_1, y_2, \dots, y_n\}$, $\mathbb{X}_u = \{x_{n+1}, x_{n+2}, \dots, x_{n+m}\}$, 학습기 f , 선택 비율 r
출력: $\mathbb{Y}_u = \{y_{n+1}, y_{n+2}, \dots, y_{n+m}\}$, 학습된 f

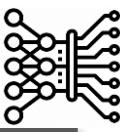
- 1 while (not 멈출 조건)
- 2 \mathbb{X}_l 과 \mathbb{Y}_l 을 훈련집합으로 사용하여 학습기 f 를 학습한다.
- 3 f 로 \mathbb{X}_u 를 분류하여 \mathbb{Y}_u 를 알아낸다.
- 4 분류된 \mathbb{X}_u 중 신뢰도가 높은 순으로 $r\%$ 의 부분집합 \mathbb{X}_{u_subset} 을 선별한다.
- 5 $\mathbb{X}_u = \mathbb{X}_u - \mathbb{X}_{u_subset}$, $\mathbb{X}_l = \mathbb{X}_l \cup \mathbb{X}_{u_subset}$, $\mathbb{Y}_l = \mathbb{Y}_l \cup \mathbb{Y}_{u_subset}$

- Sensitive situations occur for samples of ambiguous affiliation



O: labeled samples
X: Not labeled samples





❖ Co-training (협동 학습)

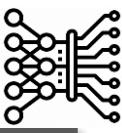
- Two learning machines developed by expanding \mathbb{X}_{expand} in cooperation with each other

알고리즘 7-2 협동 학습(2시점 협동 학습)

입력: $\mathbb{X}_l = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, $\mathbb{Y}_l = \{y_1, y_2, \dots, y_n\}$, $\mathbb{X}_u = \{\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \dots, \mathbf{x}_{n+m}\}$, 학습기 f , 선발 개수 q

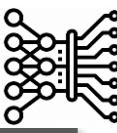
출력: $\mathbb{Y}_u = \{y_{n+1}, y_{n+2}, \dots, y_{n+m}\}$, 학습된 f

- 1 특징 공간 \mathbf{x} 를 2개의 부분공간 $\mathbf{x}_{subset1}$ 과 $\mathbf{x}_{subset2}$ 로 구분한다.
- 2 $\mathbb{X}_{expand} = \mathbb{X}_l$
- 3 while ($\mathbb{X}_u \neq \emptyset$)
 - 4 \mathbb{X}_{expand} 의 $\mathbf{x}_{subset1}$ 시점으로 학습기 f_1 을 학습한다.
 - 5 \mathbb{X}_{expand} 의 $\mathbf{x}_{subset2}$ 시점으로 학습기 f_2 를 학습한다.
 - 6 f_1 로 \mathbb{X}_u 를 분류한다.
 - 7 f_2 로 \mathbb{X}_u 를 분류한다.
 - 8 f_1 의 분류 결과에서 신뢰도가 가장 높은 q 개 샘플을 \mathbb{X}_u 에서 \mathbb{X}_{expand} 로 옮긴다.
 - 9 f_2 의 분류 결과에서 신뢰도가 가장 높은 q 개 샘플을 \mathbb{X}_u 에서 \mathbb{X}_{expand} 로 옮긴다.
 - 10 8~9에서 옮긴 샘플의 레이블을 \mathbb{Y}_u 에 추가한다.



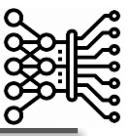
- If two learners teach each other

2. $\mathbb{X}_{expand1} = \mathbb{X}_l$, $\mathbb{X}_{expand2} = \mathbb{X}_l$
3. while ($\mathbb{X}_u \neq \emptyset$)
 4. $\mathbb{X}_{expand1}$ 의 $x_{subset1}$ 시점으로 학습기 f_1 을 학습한다.
 5. $\mathbb{X}_{expand2}$ 의 $x_{subset2}$ 시점으로 학습기 f_2 를 학습한다.
 6. f_1 로 \mathbb{X}_u 를 분류한다.
 7. f_2 로 \mathbb{X}_u 를 분류한다.
 8. f_1 의 분류 결과에서 신뢰도가 가장 높은 q 개 샘플을 \mathbb{X}_u 에서 $\mathbb{X}_{expand2}$ 로 옮긴다.
 9. f_2 의 분류 결과에서 신뢰도가 가장 높은 q 개 샘플을 \mathbb{X}_u 에서 $\mathbb{X}_{expand1}$ 로 옮긴다.

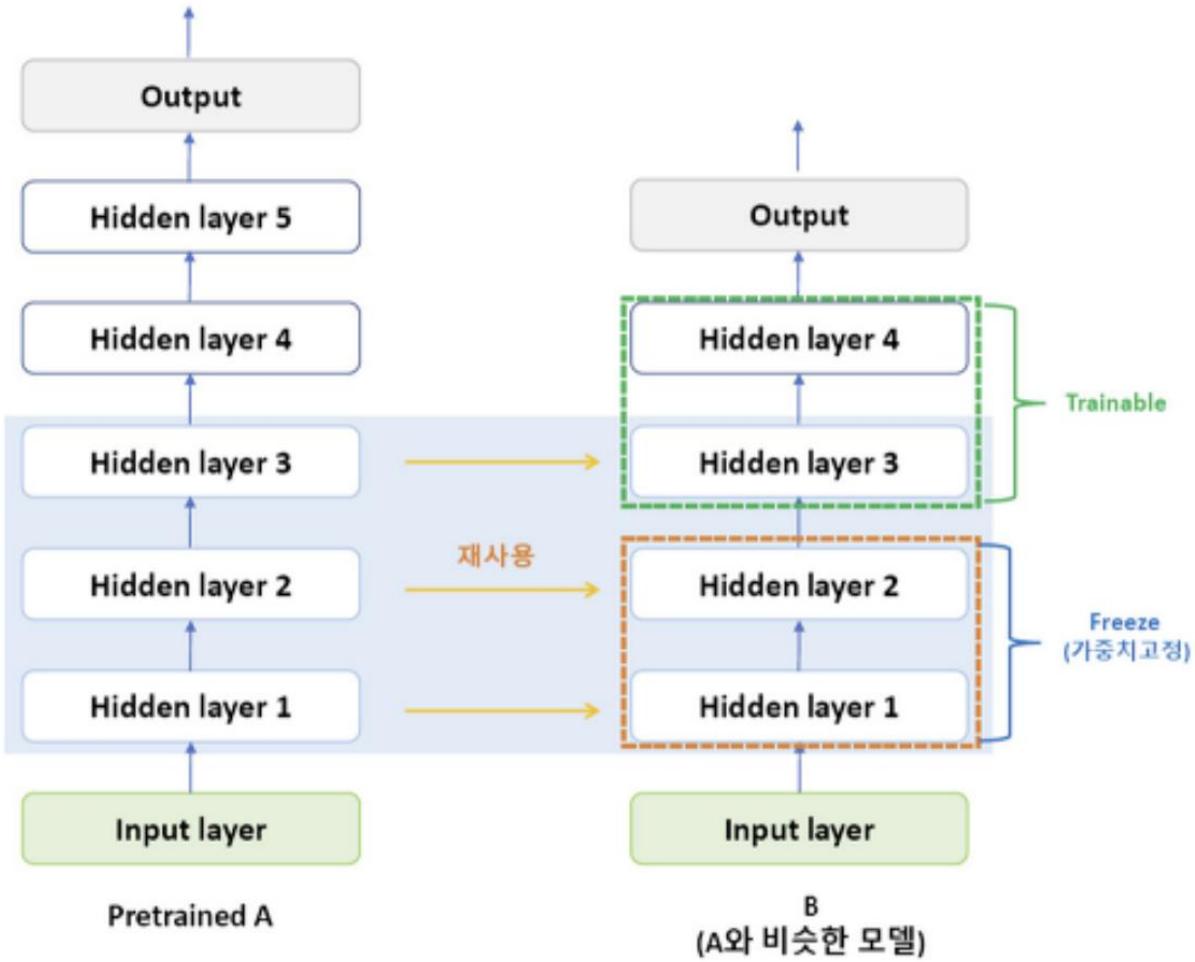


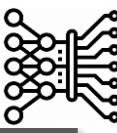
- ❖ Task Transfer Learning
- ❖ Domain Transfer Learning
- ❖ 일상 생활에서 전이 학습
 - 피아노를 칠 줄 아는 사람은 못 치는 사람보다 바이올린을 빨리 배움
 - C언어에 익숙한 학생은 파이썬을 금방 배움
 - 두 영역의 공통 지식을 공유하기 때문
- ❖ 기계 학습에서 전이 학습
 - 어떤 도메인에서 제작한 프로그램을 데이터가 적어 애를 먹는 새로운 도메인에 적용하여 높은 성능을 얻는 기법
 - 현대 기계 학습에서 널리 활용되고 있음

Transfer Learning



❖ Concepts

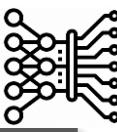




❖ Differentiate between different tasks and different domains

- 과업^{task}이 달라지는 경우: 영상 인식에서 음성 인식으로 전이하는 것처럼 응용분야가 달라질 수 있다. 이는 두 과업 사이의 거리가 아주 먼 상황이며, 아직 이러한 상황에 적용할 수 있는 실용적인 연구 결과는 없다. 반면, 자연영상을 1,000부류로 인식하는 과업을 200종의 나뭇잎을 인식하는 과업으로 전이하는 경우를 생각할 수 있다. 응용분야가 다른 경우보다는 과업 간 거리가 훨씬 짧다. 현재 이런 종류의 전이 학습은 보편적으로 적용할 수 있는 기술이 되었다. 예를 들어, 공개된 VGGNet을 소량의 나뭇잎 영상만 있는 상황에 전이하여 실용적인 나뭇잎 인식 프로그램을 만들어 앱 스토어에 공개한 사례가 여럿 있다.⁵
- 도메인이^{domain} 달라지는 경우: 특징 공간이 다른 경우와 특징 공간은 같은데 데이터의 확률분포가 다른 경우로 구분한다. 전자 사례로는 영불 번역기를 영한 번역기로 전이하거나 한국어 정보 검색기를 베트남어 정보 검색기로 전이하는 상황을 들 수 있다. 두 도메인은 단어 집합이 달라 특징 공간이 다를 수 밖에 없다. 후자 사례는 한국인이 쓴 필기 숫자 데이터베이스로 만든 인식기를 인도에 수출하는 상황에서 발생할 수 있다. 두 도메인은 같은 크기의 숫자 영상을 사용하면 되는데, 필기 습관이 달라 데이터의 확률분포가 다를 것이다. 이때 전이 학습을 사용한다면, 인도인을 대상으로 소량의 데이터만 수집해도 높은 성능을 얻을 수 있다.

Transfer Learning

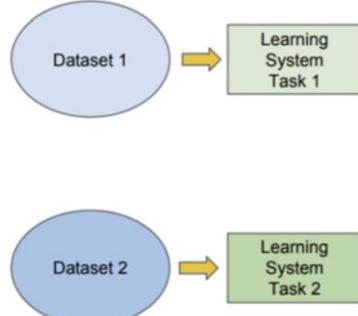


❖ Examples of successful transfer learning

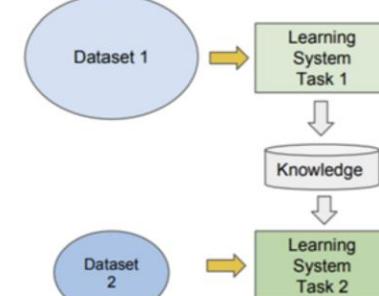
- 2012년 ILSVRC 대회에서 AlexNet은 오류율 16.3%의 경이로운 성능 달성
 - 딥러닝의 가능성을 입증
 - 프로그램, 가중치, 하이퍼 매개변수 등 모든 것을 공개하여 전이 학습을 개척함
- 영상 인식에서 과업 전이가 성공함으로써 전이 학습이 부상

Traditional ML vs Transfer Learning

- Isolated, single task learning:
 - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



- Learning of a new tasks relies on the previous learned tasks:
 - Learning process can be faster, more accurate and/or need less training data



Task Transfer Learning



❖ Traditional CNN characteristics

- Leverages feature extraction from successfully learned neural networks for other tasks

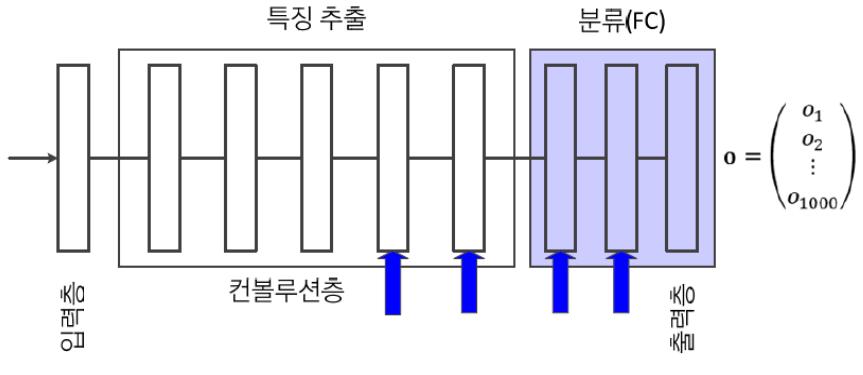


그림 AlexNet에서의 전이 학습(파란색으로 표시된 분류층 3개를 새로운 과업에 맞게 대체함)

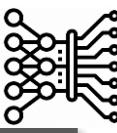
❖ 동결 방식

- 파란색 실선 화살표로 표시된 층 중 하나를 골라 특징을 취함
- 이 특징은 컨볼루션 층을 여럿 통과하면서 정제되었으므로 얇은 신경망(예, MLP)을 사용해도 높은 성능으로 분류할 수 있음

❖ Fine tuning

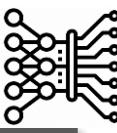
- FC 부분을 떼어낸 후, 새로운 구조를 덧붙여 다시 학습
- 이때 학습률을 낮게 설정해야 함 (높으면 원래 가중치가 훼손)

Task Transfer Learning



❖ CNN advances

기성 CNN	문헌	특성	개발 그룹
AlexNet	[Krizhevsky2012], [Jia2014]	2012년 ILSVRC 우승, 8층 구조	토론토대학
VGGNet	[Simonyan2015]	2014년 ILSVRC 준우승, 19층 구조	옥스퍼드대학
GoogLeNet	[Szegedy2015, Szegedy2016]	2014년 ILSVRC 우승, 27층 구조	구글
ResNet	[He2016a]	2015년 ILSVRC 우승, 152층 구조	마이크로소프트
R-CNN	[Girshick2015]	물체 검출용 CNN	마이크로소프트



❖ Domain Transfer

- 과업은 같은데(즉 레이블 공간이 같음), 도메인이 다른 상황
- 특징 공간이 다른 경우와 특징 공간은 같은데 확률분포가 다른 경우로 나뉨
 - 후자를 도메인 적응이라 부름

❖ Domain Adaptation

- 예, 나뭇잎 인식
 - 원천 도메인은 따낸 나뭇잎 → 목표 도메인은 나무에 붙어있는 나뭇잎 영상

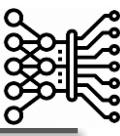


(a) 도메인 1



(b) 도메인 2

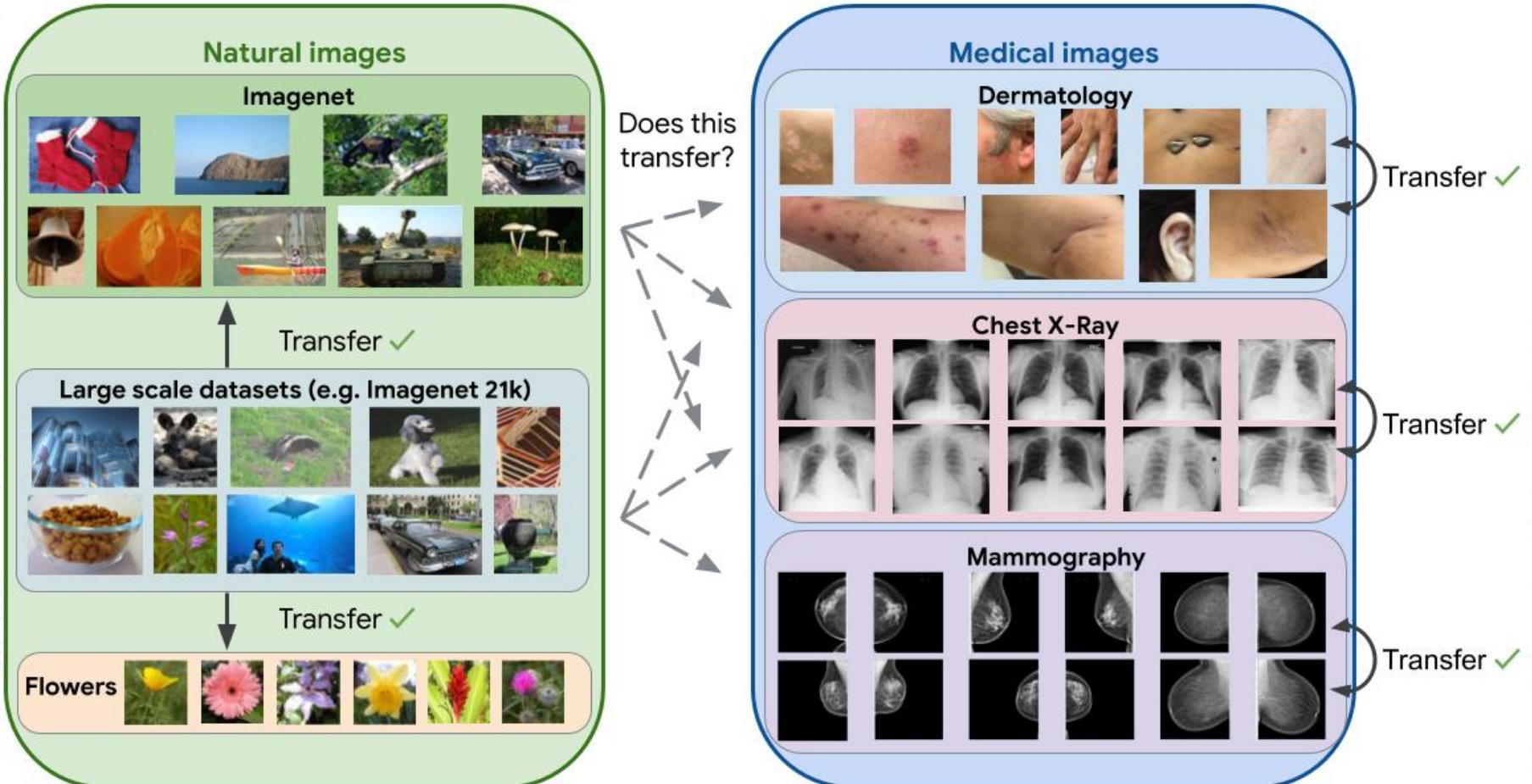
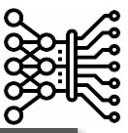




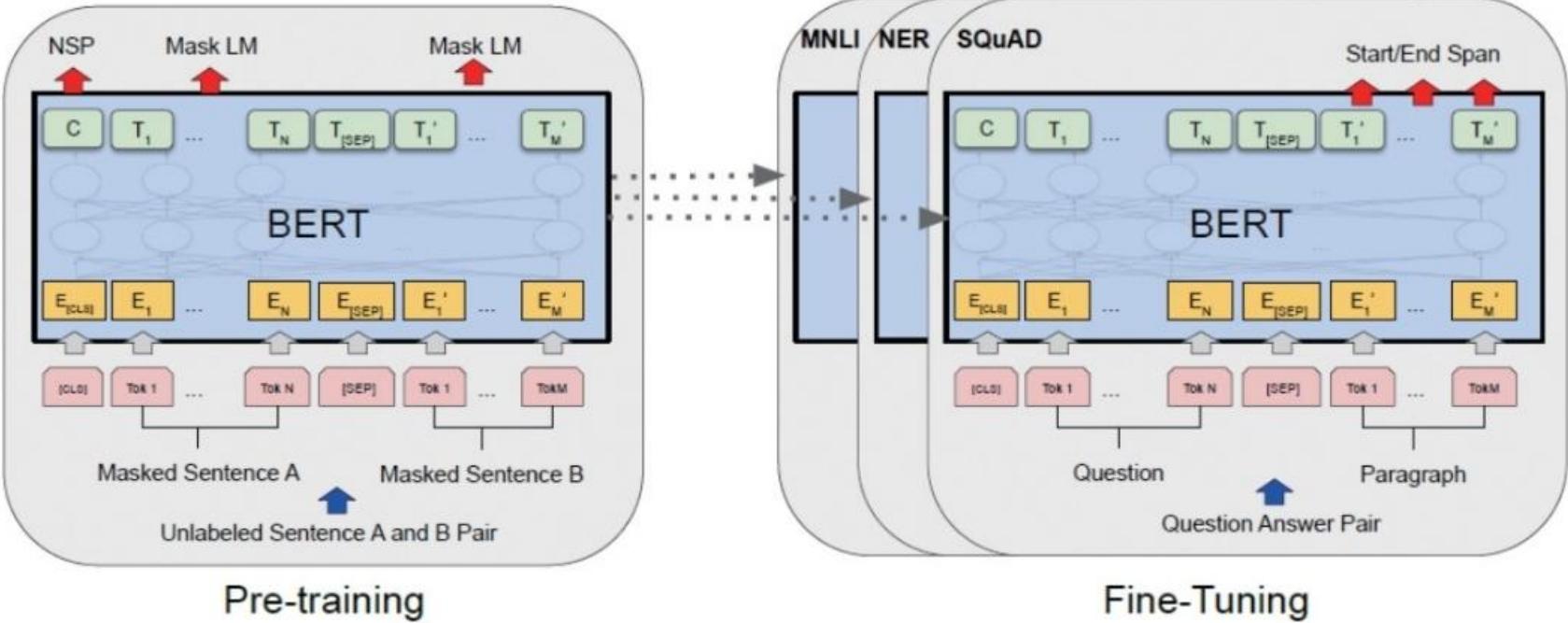
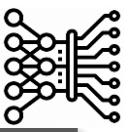
❖ How to

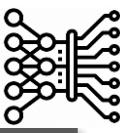
- 원천 도메인의 훈련집합 $\mathbb{X}_s = \{\mathbf{x}_{s1}, \mathbf{x}_{s2}, \dots, \mathbf{x}_{sn}\}$, \mathbb{Y}_s
- 목표 도메인의 훈련집합 $\mathbb{X}_t = \{\mathbf{x}_{t1}, \mathbf{x}_{t2}, \dots, \mathbf{x}_{tn}\}$
- 목표 도메인을 위한 레이블 정보 \mathbb{Y}_t 는 있을 경우(지도 도메인 적응), 없는 경우(비지도 도메인 적응), 일부만 있는 경우(준지도 도메인 적응)로 나뉨

Applications



Applications





“자기 지도 학습”이 가능한 AI 기술

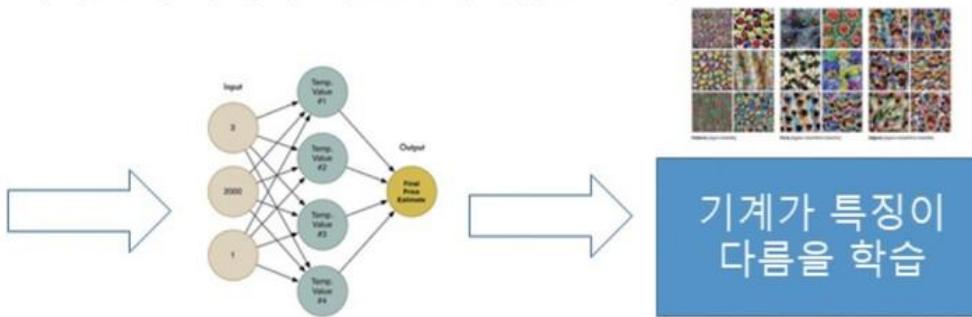
>> 사람이 정의한 데이터가 아닌 기계가 스스로 학습 가능하도록 분류를 찾고 스스로 의미를 부여하여 학습할 수 있는 AI 기술



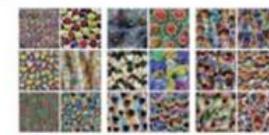
분류 학습



현실 데이터



데이터의 특징을 학습



기계가 특징이
다름을 학습



상식적 추론
시간적 추론
공간적 추론
상황적 추론

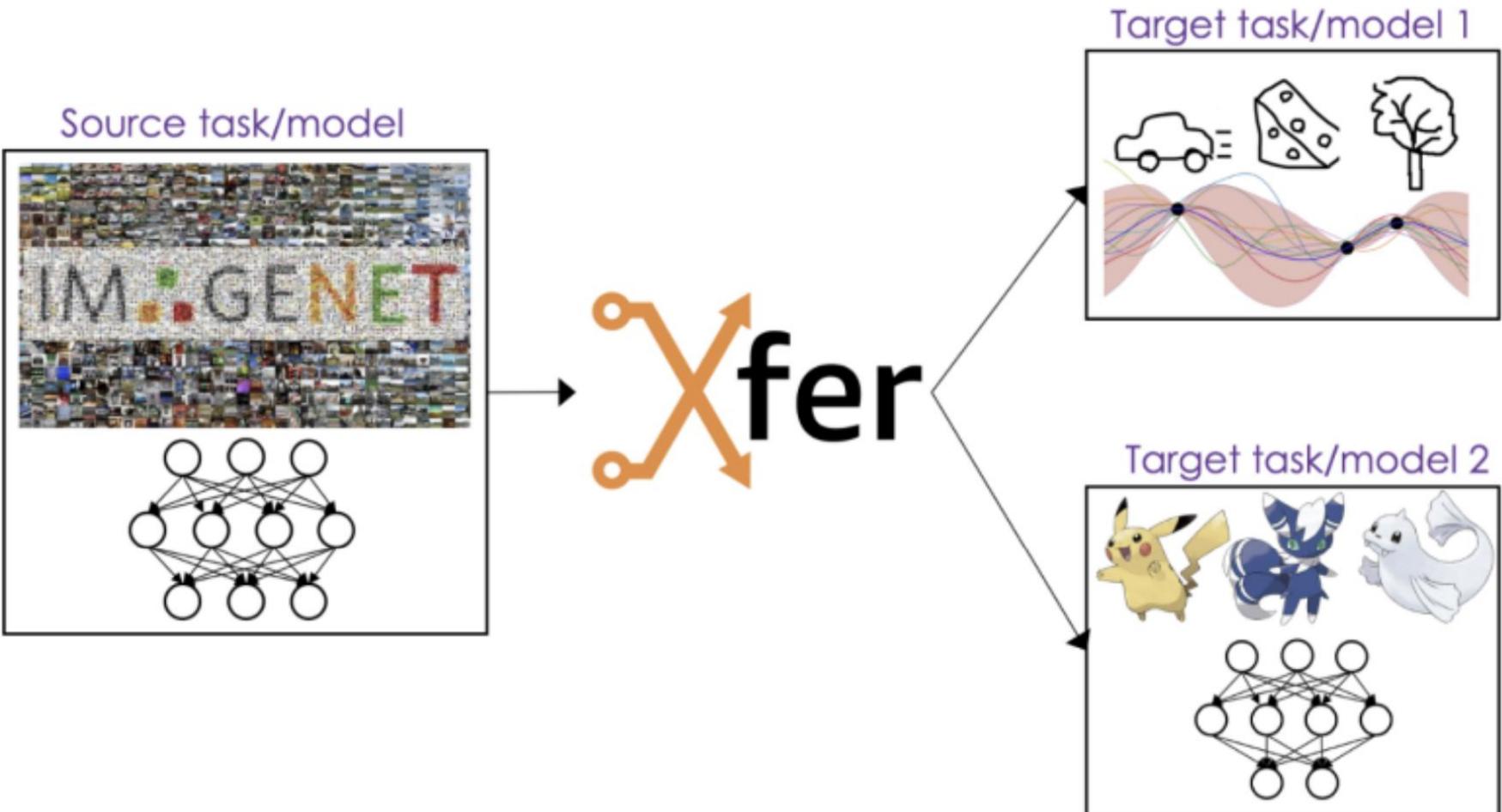
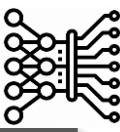


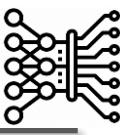
새로운 도메인
데이터의 추론
- 자기 지도 -



전이학습

Applications





- ❖ Generative ML 정의, 설명, 원리
 - 본인이 이해하여 작성 (인용이 필요한 경우 참고문헌 반드시 표시)
- ❖ 다양한 Generative ML 종류에 대하여 설명 (3가지 이상)
 - 다양한 네트워크들에 대한 조사
- ❖ Generative ML을 이용한 응용 분야 정리 (3가지 이상)
 - 구체적인 예시를 들어 설명
- ❖ 연구기관, 산업체에서 Generative ML을 활용하는 사례 조사 (3가지 이상)
 - 국내, 해외 기업 또는 연구기관에서 GAN을 활용하여 서비스를 제공하거나 독창적인 기술 발표들을 정리
- ❖ A4 8장内外, PDF로 제출, 영어로 작성