Generative Model Ver 2

2022. 09. 29/ 안민용

0. 목차

1. Intro

- Why Generative Model?
- Why GAN?

2. GAN

- Intro
- Two-player Game

3. Futher

- GAN의 활용
- Summary

2. Generative Model

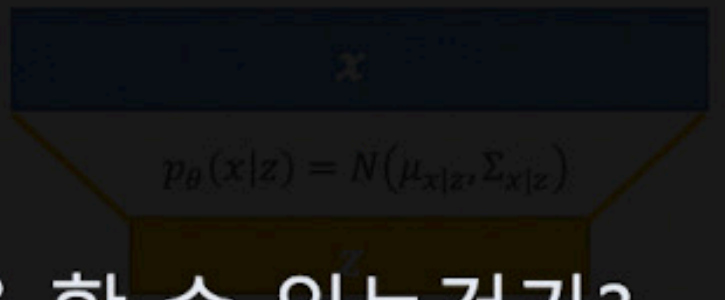
Models-VAE

Sampling(생성) 법

1. Prior of z(Gaussian)으로 표본 생성 : $p_\theta(z)$
2. Decoder로 posterior의 Parameter 계산: $\mu_{x|z}, \Sigma_{x|z}$
3. Posterior 분포로 표본 생성 : $p_\theta(x|z)$

그런데...

꼭 확률을 계산해야 sampling을 할 수 있는건가?



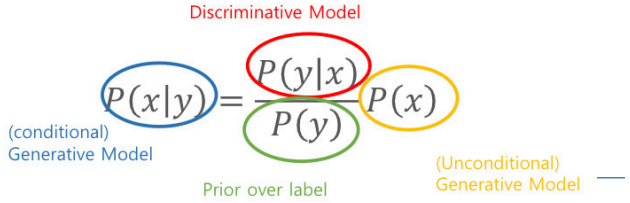
Why Generative Model?

Discriminative Model과 Generative Model

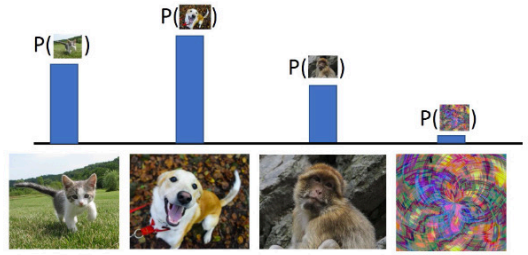
지금까지 살펴본 Generative model의 가장 큰 특징은 다음과 같습니다.

Strong Assumption 즉 feature의 분포에 대한 가정으로부터 시작한다는 점입니다.

1. Intro



Why Generative Model?



*prediction에 있어서

장점 :

입력되는 x 값이 Gaussian 분포(타 모델의 경우 x에 대한 assumption)를 따른다면, 적은 데이터가 들어오더라도 해당 모델의 성능은 좋을 것입니다.

또한 Assumption을 따른다는 가정하에, Discriminative model에 비해 computing 리소스가 적게 소모될 것입니다.

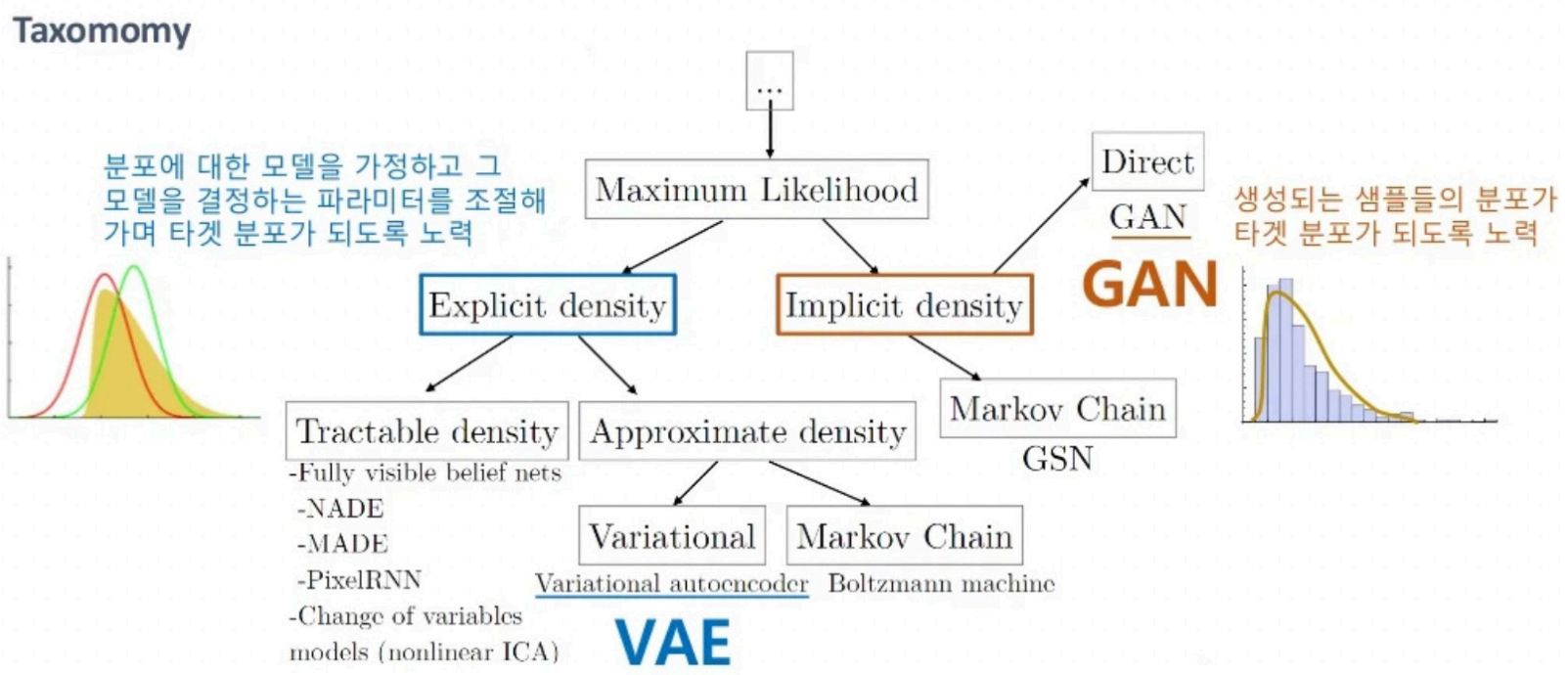
단점 :

x에 대한 assumption이 틀리면 (만약에 x가 gaussian을 따르지 않는다면) 나쁜 성능을 내보낼 것입니다.

그러나 상기 단점은 x분포에 대한 가정이 없는 Discriminative model의 경우 해당되지 않는다는 특징이 있습니다.

1. Intro

Why GAN?



출처 : https://www.slideshare.net/NaverEngineering/ai-83896428

2. GAN

Intro

확률 계산이 아니라, Sampling을 할 수 있는 능력만을 가지고 싶다!!

복잡하고, 고차원 training distribution에서 직접 Sampling을 하는 것은 매우 어려웠다.

2. GAN

Intro

간단한 distribution (예를 들어 random noise)로 부터 Sample하고 싶다.

Training distribution으로의 transformation을 학습!!

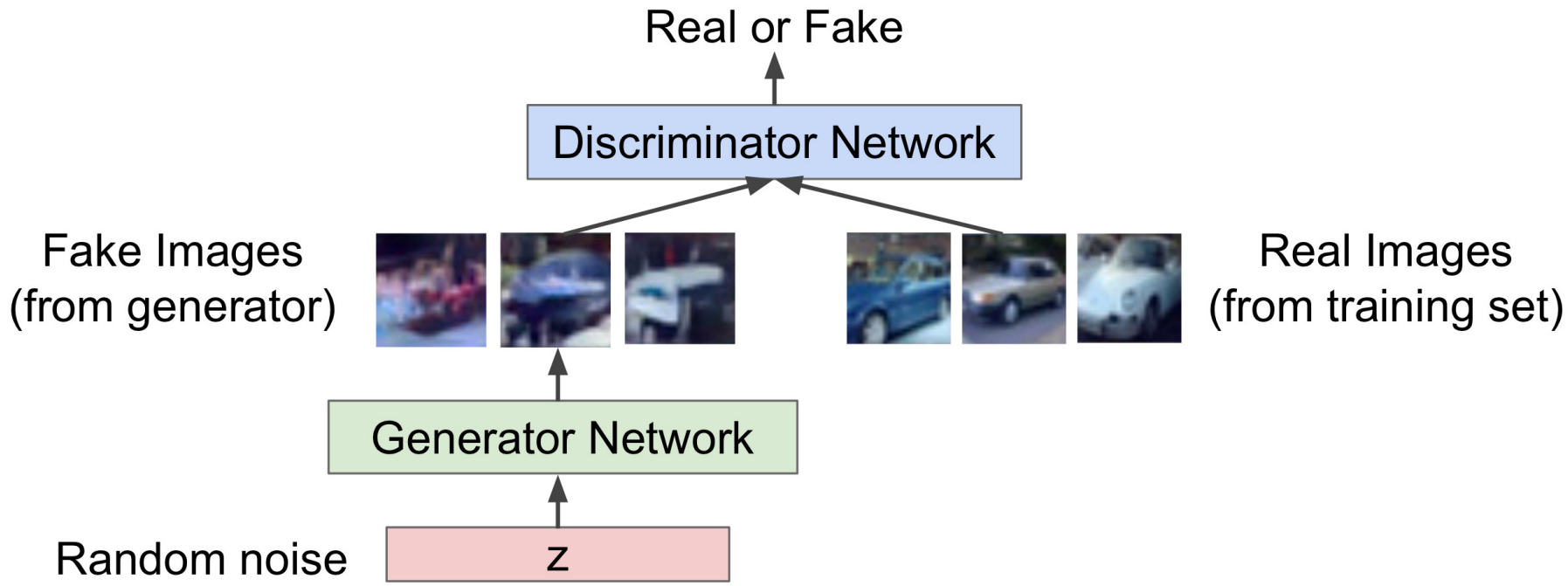
질문 타임

Q) 어떤 방법으로 이런 복잡한 transformation을 나타낼 수 있을까요?!

A) ???

2. GAN

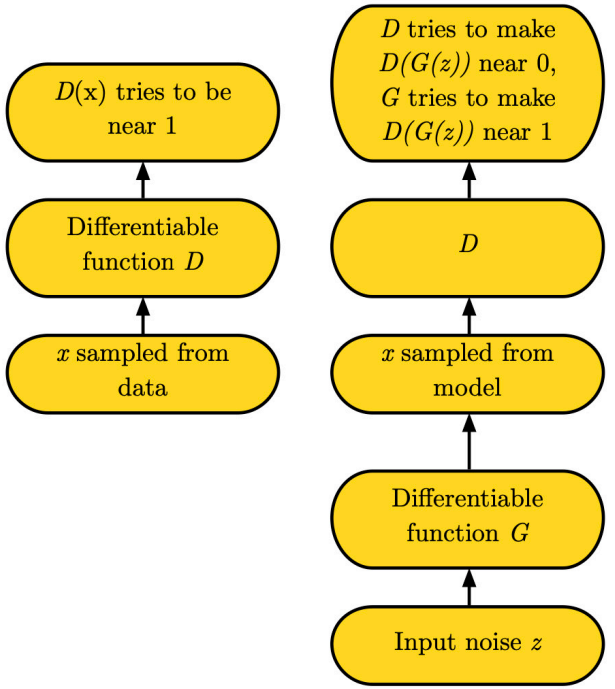
Two-player game



출처 : cs231n_2017_lecture13

Two-player game

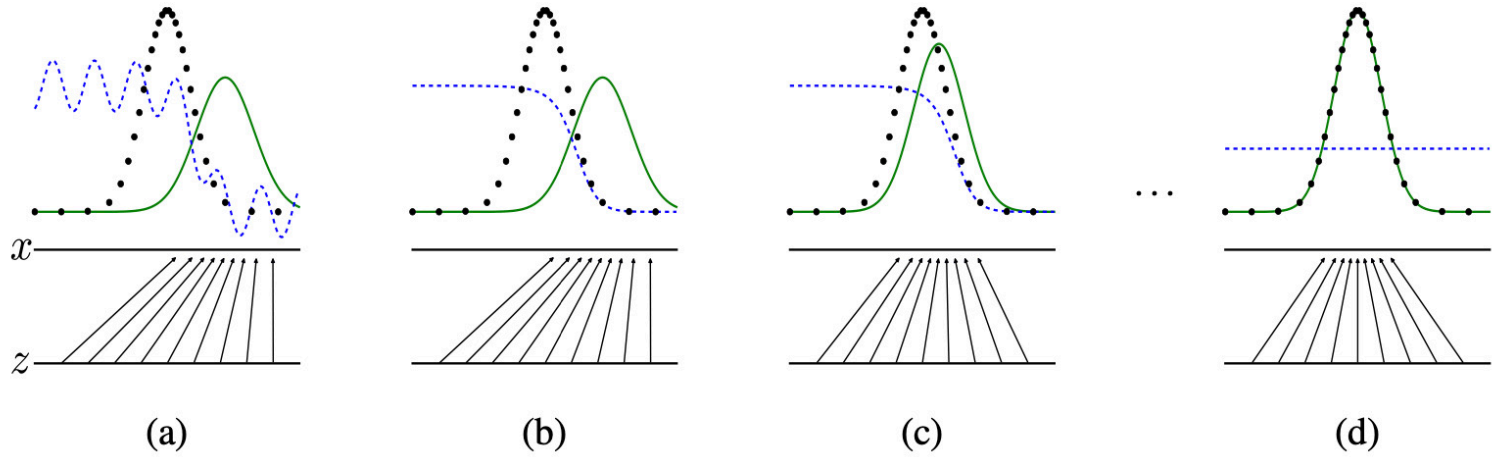
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$



출처 : Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Two-player game

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



출처 : Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 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 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

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

end for

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

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

end for

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

출처 : Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

Two-player game

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

그런데 해당 식은, G로 하여금 초기 학습에 있어서 충분한 Gradient를 제공하지 않을 수 있다.

What?!

2. GAN

for number of training iterations **do**

  for k steps **do**

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

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

출처 : cs231n_2017_lecture13

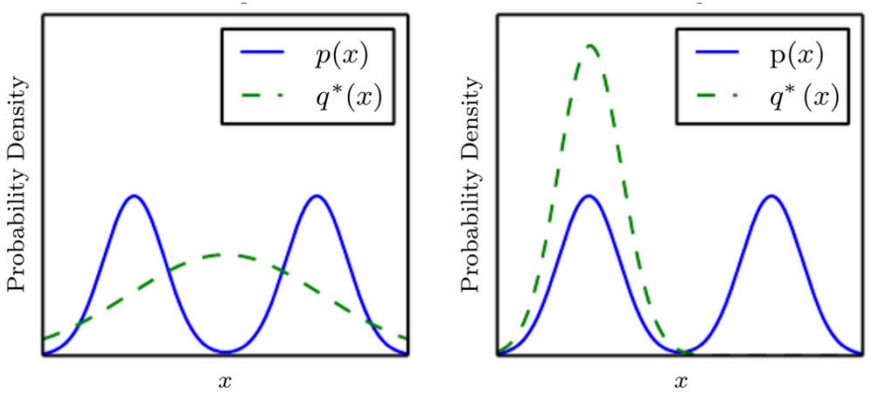
주의할 점

1. Mode Collapse (Lack of Diversity)
2. Unstable Training.
3. Intuition is NOT Enough.
4. Lots of Training Time

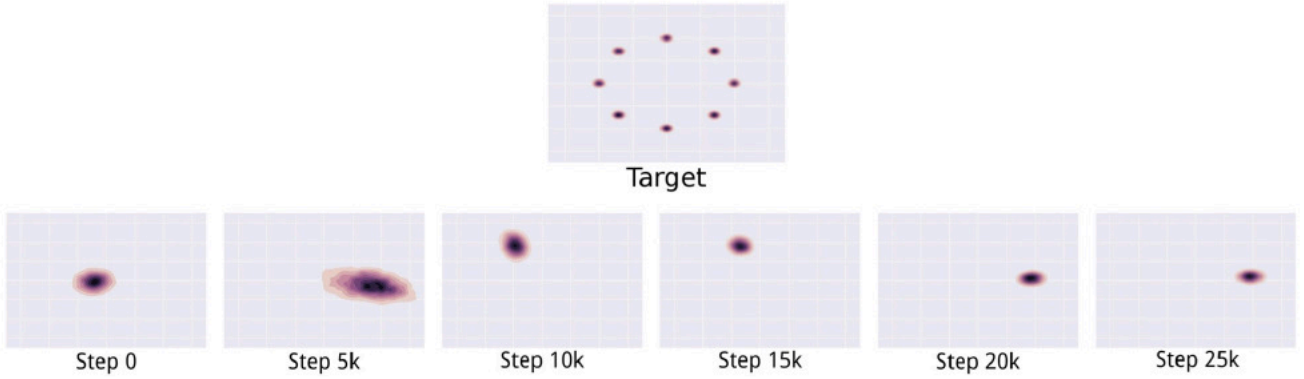
출처 : https://towardsdatascience.com/instability-in-training-text-gan-20273d6a859a

Mode Collapse (Lack of Diversity)

Generator가 다양한 z 값으로부터 동일한 Output을 mapping 하도록 학습하는 경우



* mixture of Gaussians in a two-dimensional space.



Mode Collapse (Lack of Diversity)

사람의 주관적인 판단을 사용하여 측정하는 방법을 사용.

BUT
사람이 볼 때는 하나의 mode로부터 뽑은 sample이 그럴듯하면 이미지의 질이 좋다고 생각하기 쉽다.
그러나 실제로는?!

모든 mode에서 골고로 나오는 건지, 일부 한정된 mode에서 적당한 이미지가 나오면 속아넘어가기 마련..

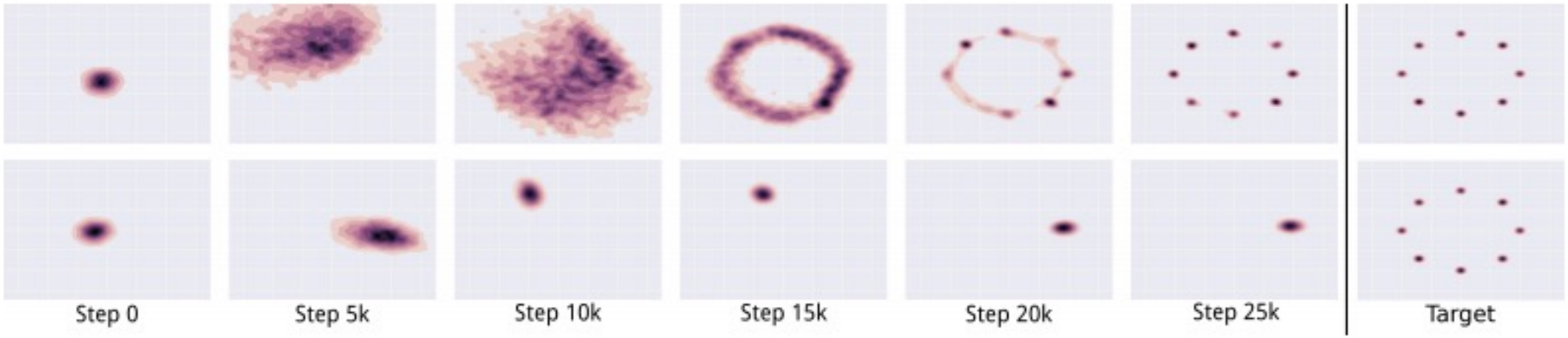
Unrolled GAN

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

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

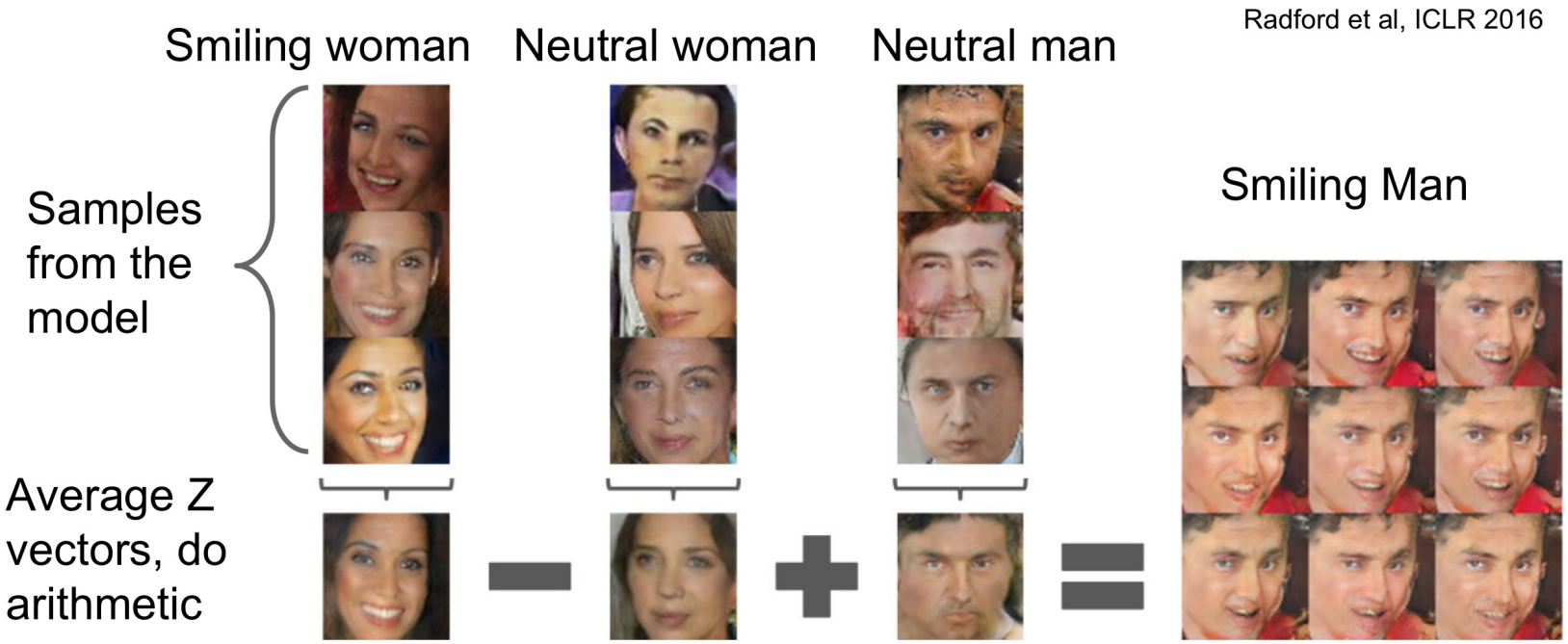
G입장에서는 D를 속일 수 있는 sample 하나만 내보내면 된다.

Unrolled GAN



3. Futher

GAN의 활용



3. Futher

GAN의 활용 - StyleGAN

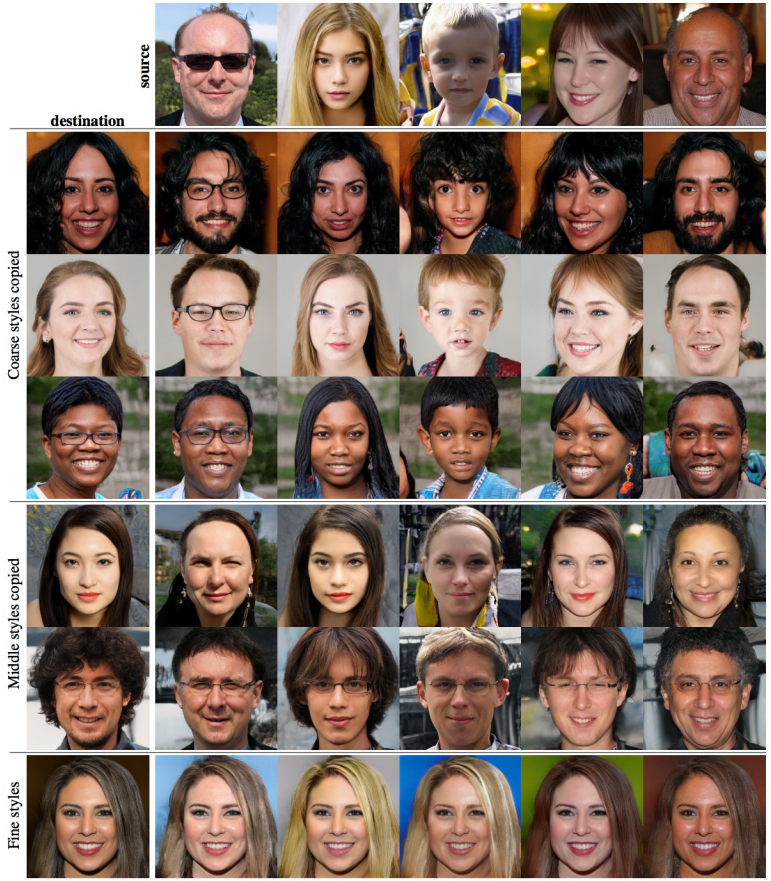
Generator를 통한 이미지 합성 과정은 Black Box로서, 이미지의 attribute를 조절하기 어렵다.

⇒ StyleGAN

Style Transfer에 기반한 새로운 Generator 구조

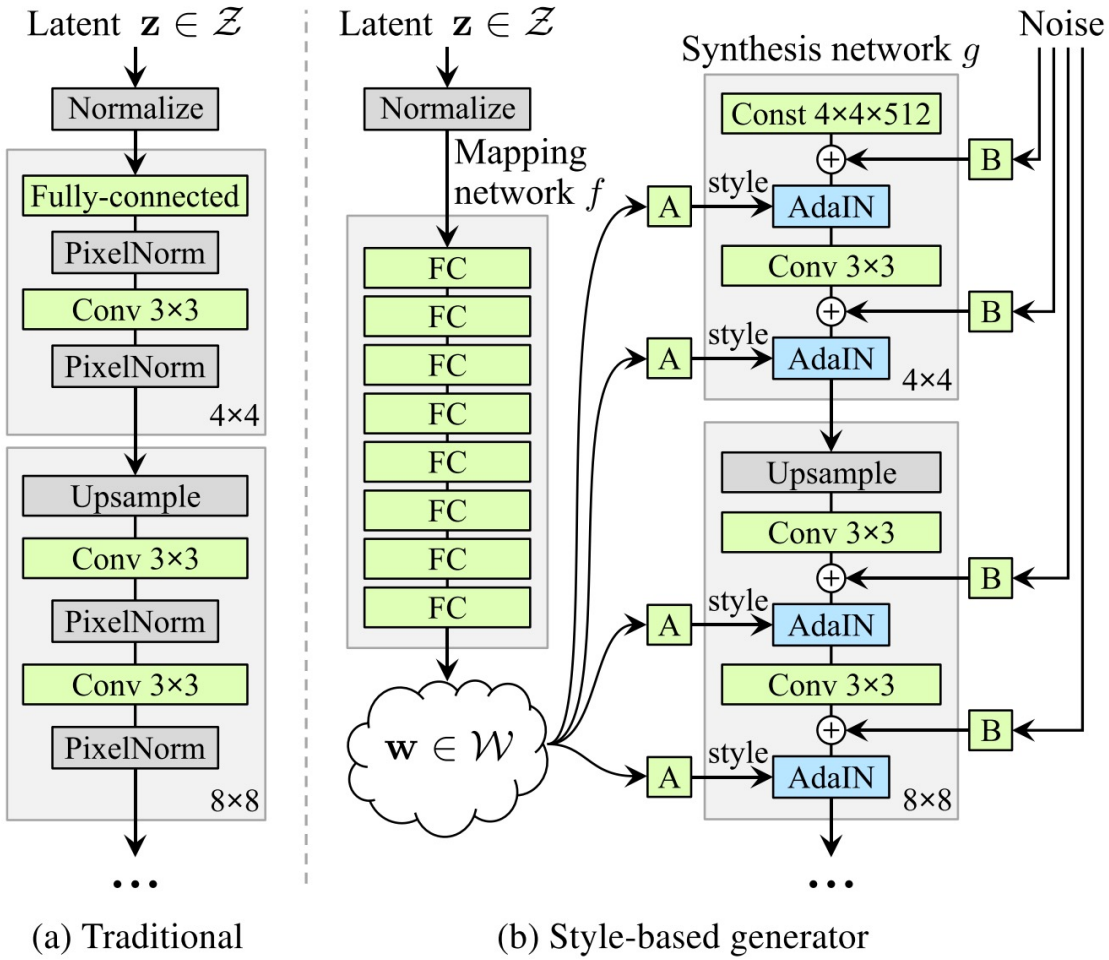
이미지를 style의 조합으로 보고, Generator의 각 layer마다 style 정보를 입히는 방식으로 이미지 합성.

e.g.) 성별, 포즈, 머리색, 피부톤 등등



3. Futher

GAN의 활용 - StyleGAN



(a) Traditional (b) Style-based generator

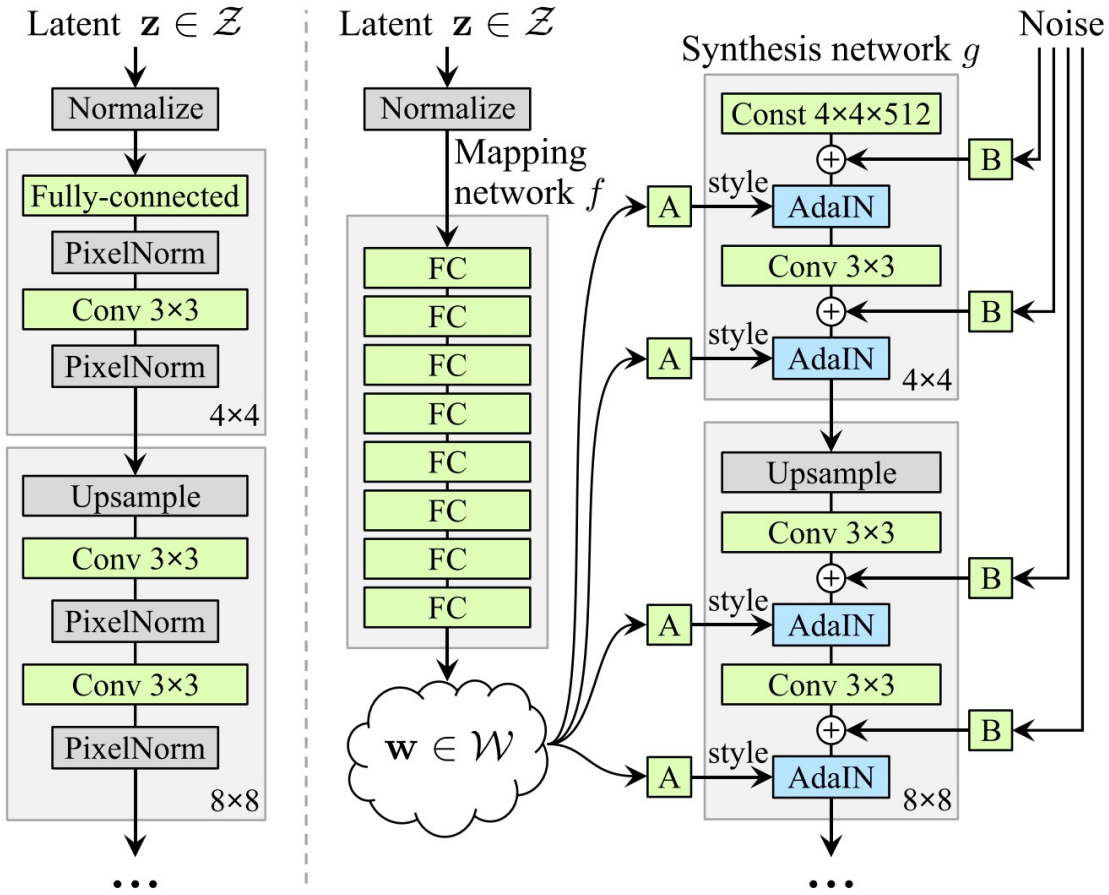
learnt intermediate latent space, W를 사용.

Standard gaussian latent space에 비해, training data의 분포를 더 잘 나타내는 space

이에 더해 disentangled property또한 가지고 있다.

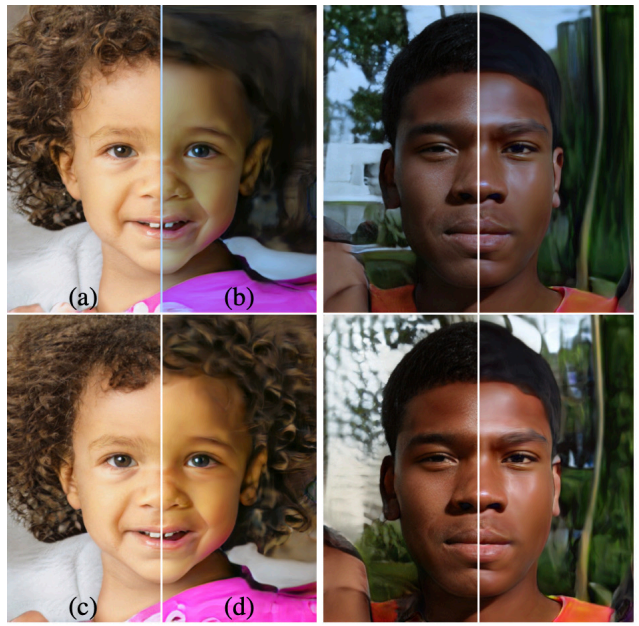
3. Futher

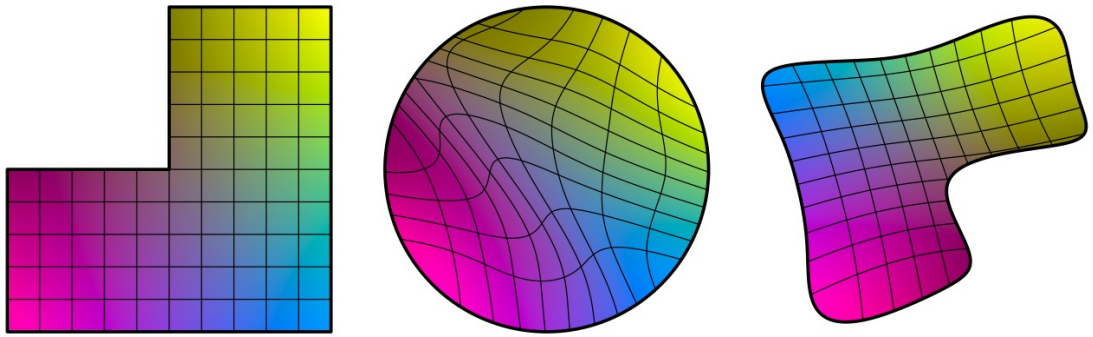
GAN의 활용 - StyleGAN



(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

3. Futher

GAN Inversion - e4e

실제 이미지에, StyleGAN에서의 Style transfer와 같은 manipulation을 하려면?!

위에서 살펴본 GAN 친구들은, "walking" direction을 찾는 방향으로 진행.
e.g.) fully-supervised 환경에서는, 라벨을 통해 young 〈-〉 old 이런식으로 방향을 찾음.

실제 주어진 이미지를 latent space W로 invert 해야 한다.
이런 w를 구할 수 있고 + editing capabilites까지 가지고 싶다!

3. Futher

GAN Inversion - e4e

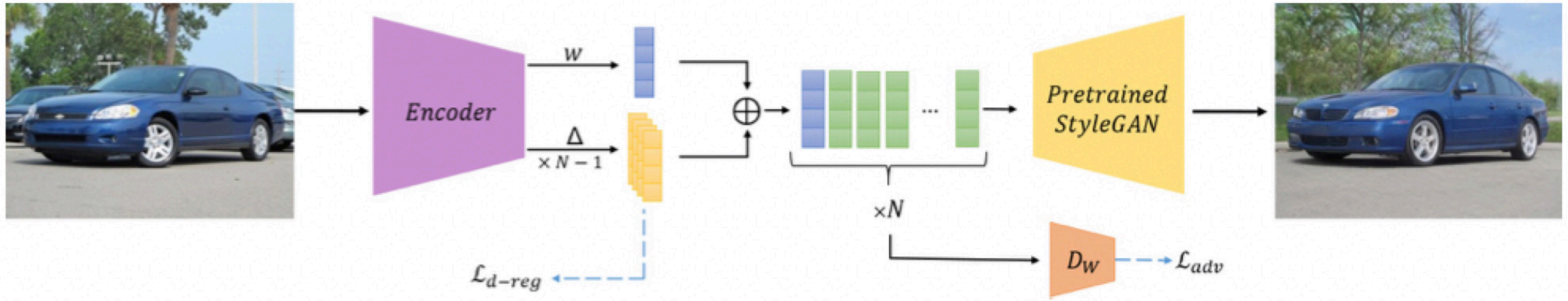
핵심은 이미지를 W에 가장 인접하게 inverting하는 것에 있다.

1. 다른 style vector들 사이에서 low variance
2. Distribution W 안에 위치해야 한다.

이 논문은 이것을 해낼 수 있는 e4e라는 encoder를 제시

3. Futher

GAN Inversion - e4e



$$E(x) = (w_0, w_1, ..., w_{N-1})$$

$$(w, w + \Delta_1, ..., w + \Delta_{N-1}).$$

$$\mathcal{L}_{\text{adv}}^{D} = - \mathop{\mathbb{E}}_{w \sim \mathcal{W}} [\log D_{\mathcal{W}}(w)] - \mathop{\mathbb{E}}_{x \sim p_X} [\log(1 - D_{\mathcal{W}}(E(x)_i)] + \frac{\gamma}{2} \mathop{\mathbb{E}}_{w \sim \mathcal{W}} \left[\|\nabla_w D_{\mathcal{W}}(w)\|_2^2 \right], \quad (2)$$

$$\mathcal{L}_{\text{adv}}^{E} = - \mathop{\mathbb{E}}_{x \sim p_X} [\log D_{\mathcal{W}}(E(x)_i)]. \quad (3)$$

3.Futher

E4E 뿐만이 아니라, PTI, StyleClip 등등…

예시: https://huggingface.co/spaces/rinong/StyleGAN-NADA

3. Futher

GAN의 활용 - StackGAN

GAN을 통한 Text to Image의 대표적인 논문

단순히 Vanila GAN에 upsampling layers를 더한 GAN_INT-CLS 보다는,

GAN을 두 층을 쌓아서(Stack) 더 좋은 성능을 낼 수 있음을 보여준다.

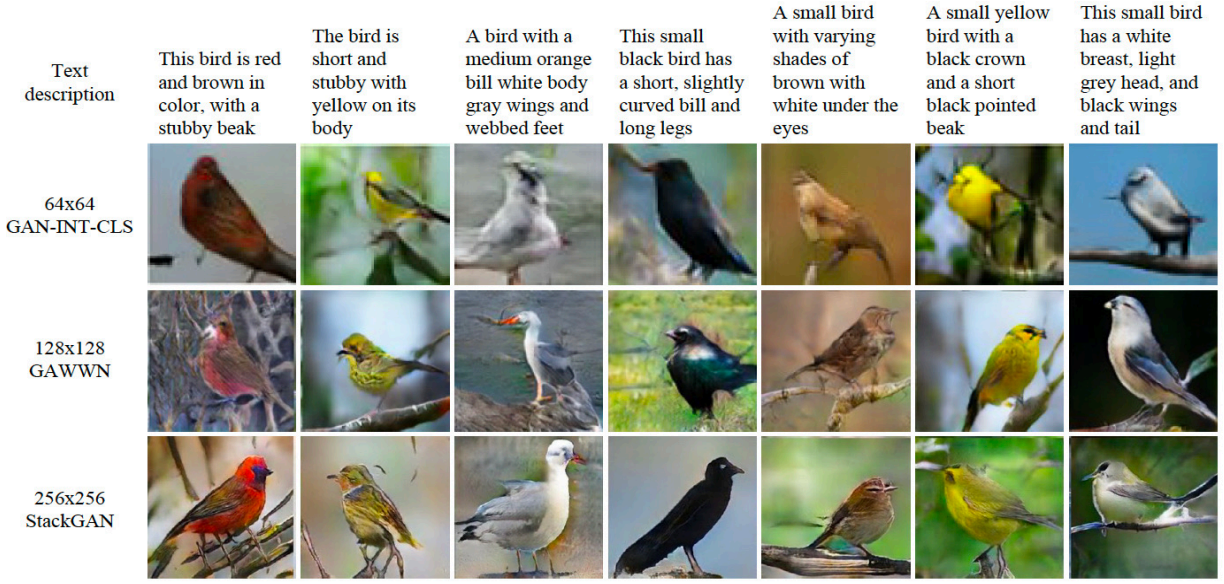


Figure 3. Example results by our StackGAN, GAWWN [24], and GAN-INT-CLS [26] conditioned on text descriptions from CUB test set.

GAN의 활용 - StackGAN

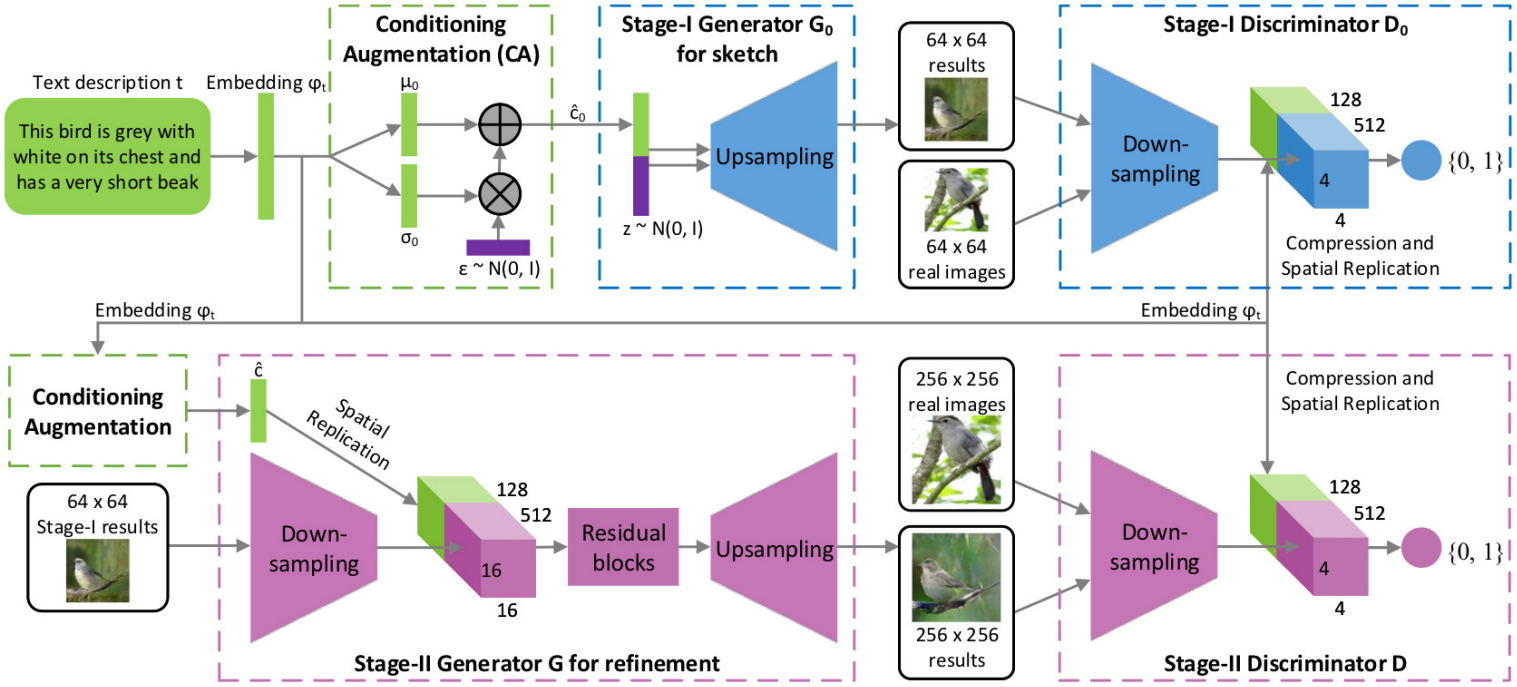


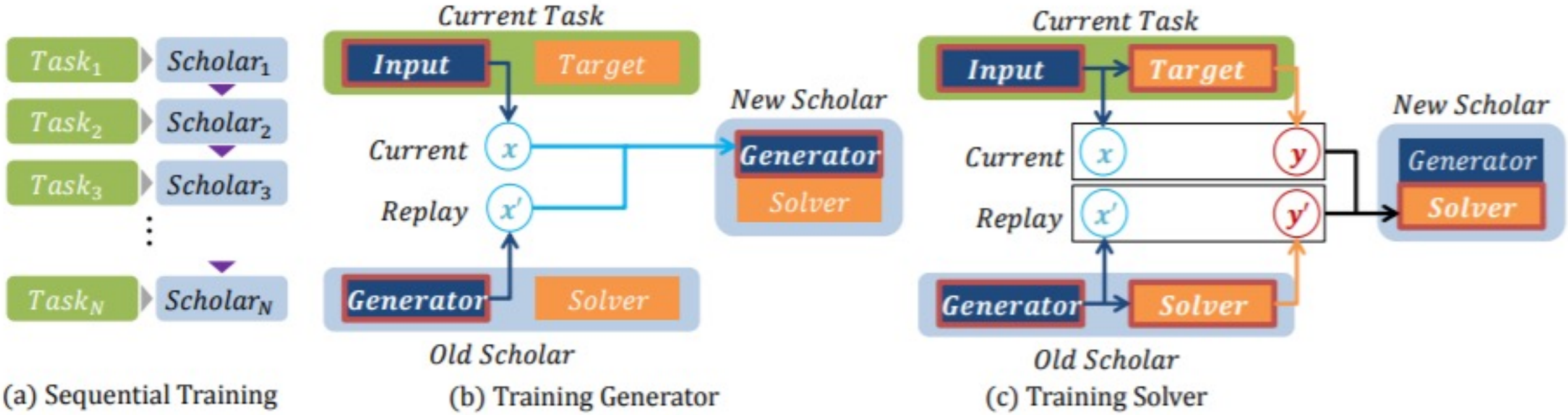
Figure 2. The architecture of the proposed StackGAN. The Stage-I generator draws a low-resolution image by sketching rough shape and basic colors of the object from the given text and painting the background from a random noise vector. Conditioned on Stage-I results, the Stage-II generator corrects defects and adds compelling details into Stage-I results, yielding a more realistic high-resolution image.

3. Futher

Continual Learning with Deep Generative Replay, nips2017, Shin et al

이미 학습된 모델에 새로운 도메인, 클래스를 학습시키는 경우

지금까지 catastrophic forgetting을 완화시키기 위한 연구는
이전데이터를 저장하는 episodic memory system에 기반,
상기 논문을 통해 GANs를 사용한 생성 모델을 학습시켜서 과거 데이터를 모방

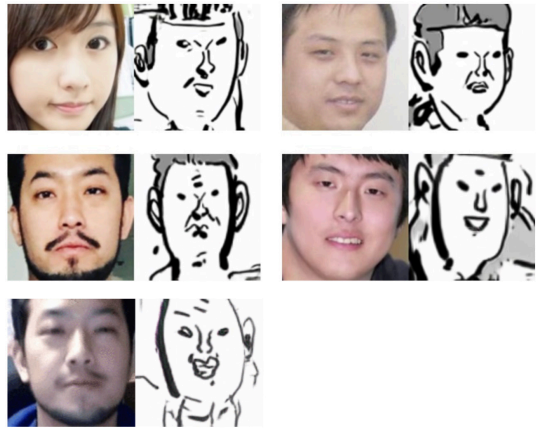


(a) Sequential Training (b) Training Generator (c) Training Solver

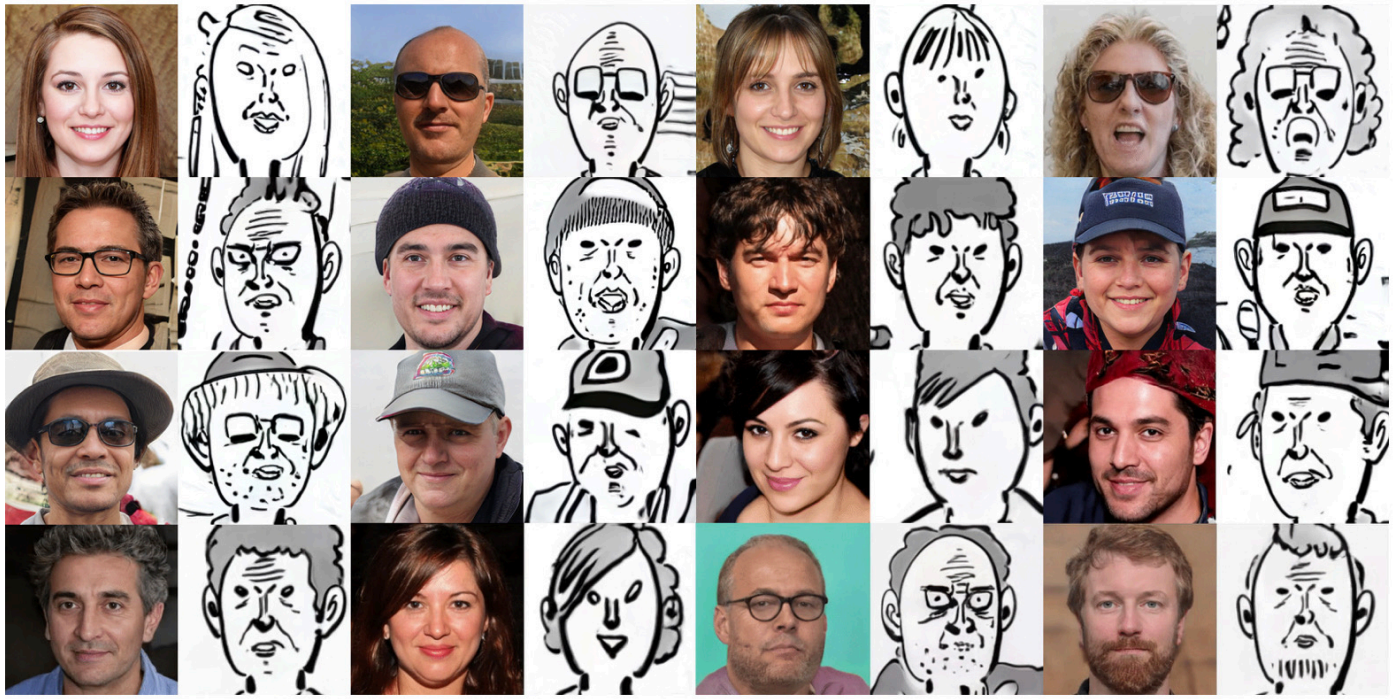
3. Futher

GAN의 활용 - StyleGAN

이말년 변환기



https://github.com/bryandlee/malnyun_faces



END

참고

- 1시간만에 GAN(Generative Adversarial Network) 완전 정복하기
- CS231N_2017_Lecture 13: Generative Model
- Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014
- NIPS 2016 Tutorial: Generative Adversarial Networks
- https://alsdyd98.github.io/Stack-GAN.html