DATA
SCIENCE LAB

Generative model 2

23.03.23 / 8기 황진우

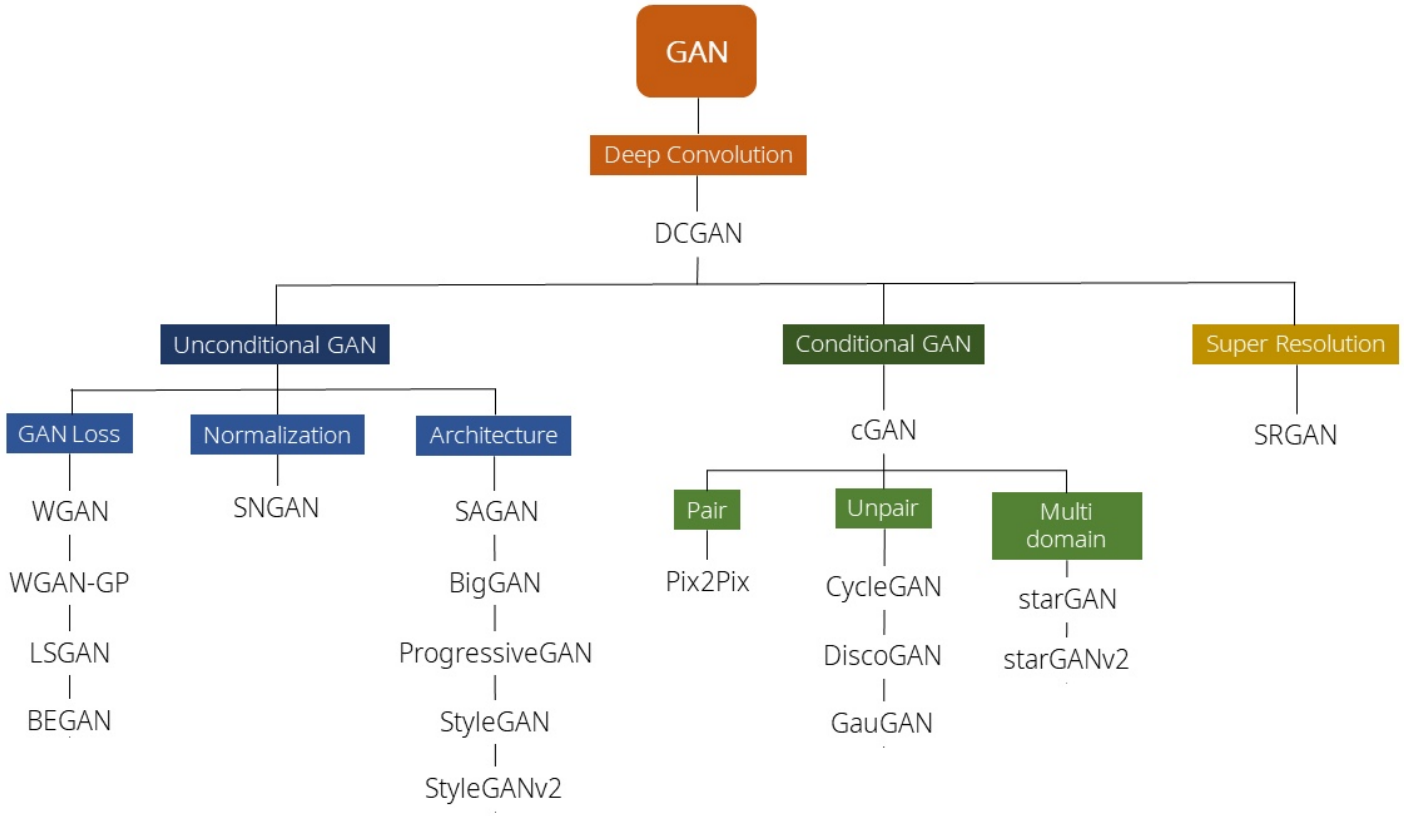
CONTENTS

1. GAN

목적

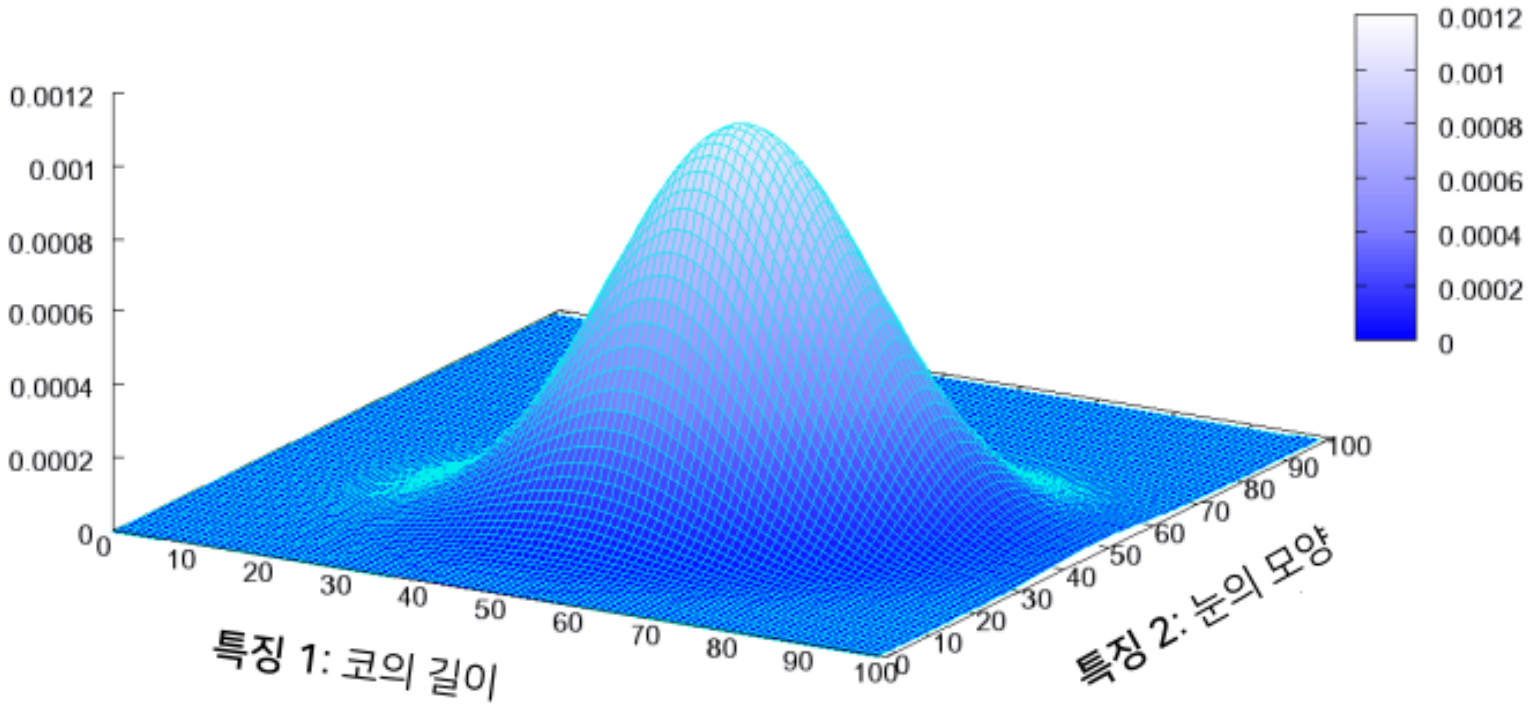
- Gan은 실존하지 않지만 있을 법한 이미지(그 외의 다른 것 포함)를 생성할 수 있는 모델을 의미한다
- 데이터의 분포를 근사하는 모델 G를 만드는 것이 생성 모델의 목표!

1. GAN


GAN
Deep Convolution
DCGAN
Unconditional GAN
Conditional GAN
Super Resolution
GAN Loss
Normalization
Architecture
cGAN
SRGAN
WGAN
WGAN-GP
LSGAN
BEGAN
SNGAN
SAGAN
BigGAN
ProgressiveGAN
StyleGAN
StyleGANv2
Pair
Unpair
Multi domain
Pix2Pix
CycleGAN
DiscoGAN
GauGAN
starGAN
starGANv2

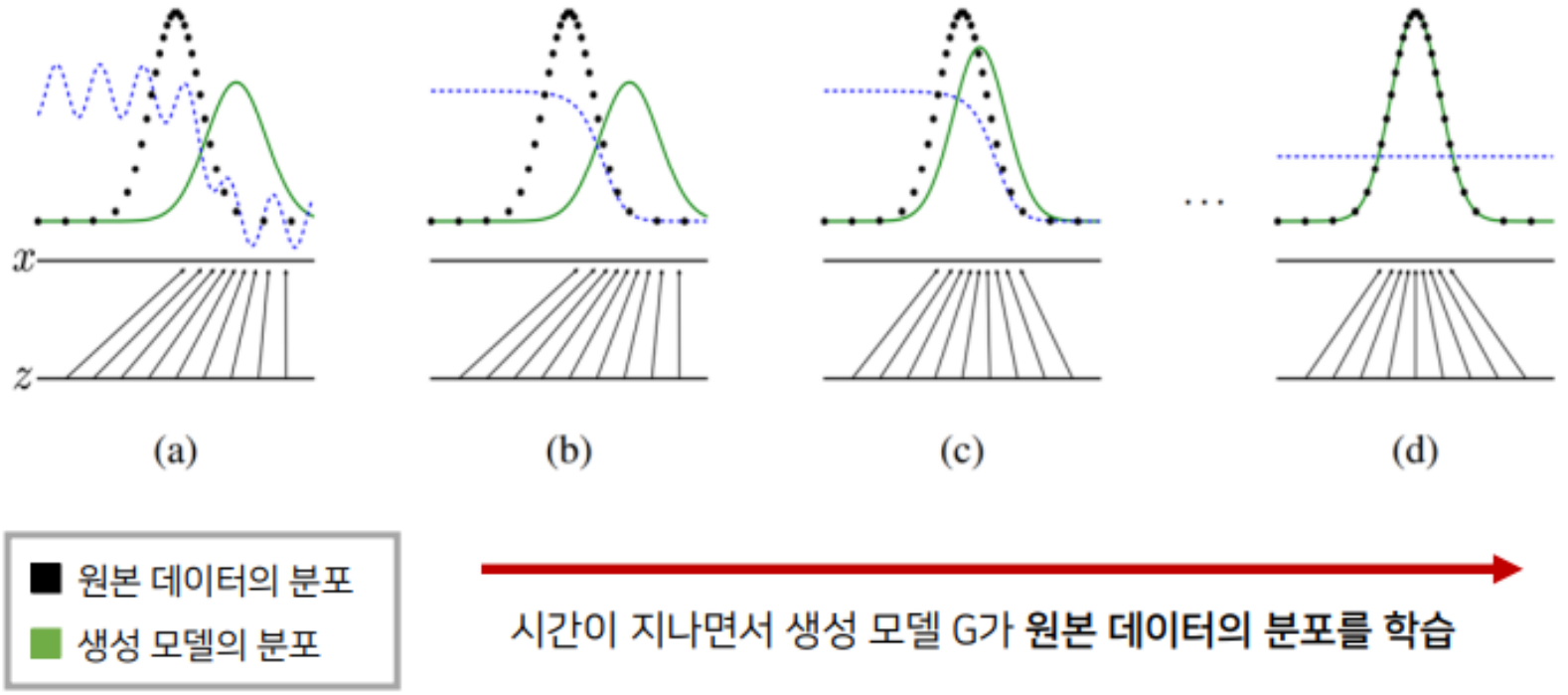
1. GAN

확률 분포


0.0012
0.001
0.0008
0.0006
0.0004
0.0002
0
0
10
20
30
40
50
60
70
80
90
100
특징 1: 코의 길이
0
10
20
30
40
50
60
70
80
90
100
특징 2: 눈의 모양

1. GAN

분포



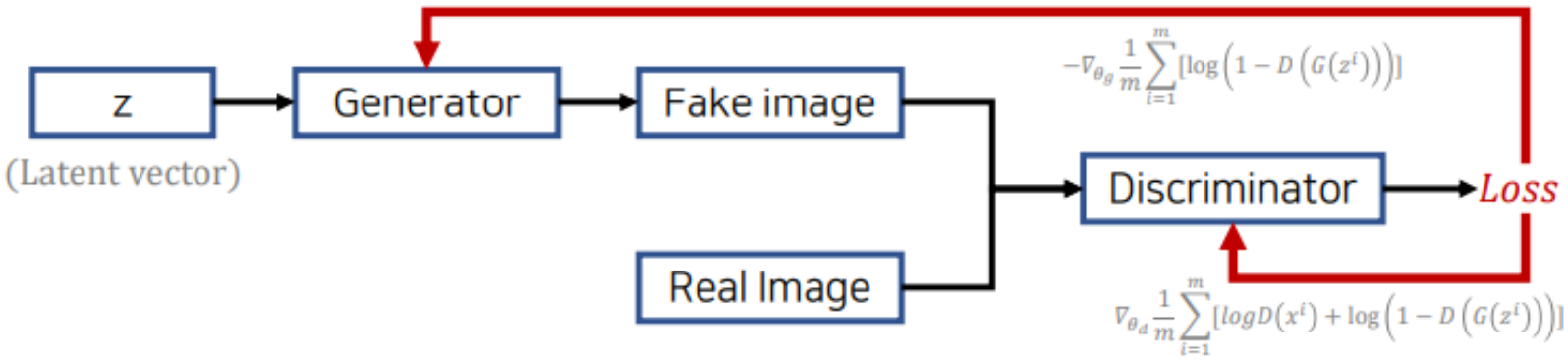
1. GAN

Objective Func

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

G(x): new data instance

D(x): probability (sample from the real dist {Real:1, Fake:0})



1. GAN

Global Optimality

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

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

$$= \int_x p_{data}(x)[logD(x)]dx + \int_z P_z(z)\left[\log\left(1 - D\left(G(z)\right)\right)\right]dz$$

$$= \int_x p_{data}(x)[logD(x)] + P_x(x)\left[\log\left(1 - D\left(G(x)\right)\right)\right]dx$$

1. GAN

Global Optimality

$$p_g = p_{data}$$

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

$$= E_{x\sim p_{data}(x)}\left[log\frac{p_{data}(x)}{P_{data}(x)+p_g(x)}\right] + E_{z\sim p_z(z)}\left[log\frac{p_{data}(x)}{P_{data}(x)+p_g(x)}\right]$$

$$= E_{x\sim p_{data}(x)}\left[log\frac{2*p_{data}(x)}{P_{data}(x)+p_g(x)}\right] + E_{z\sim p_z(z)}\left[log\frac{2*p_{data}(x)}{P_{data}(x)+p_g(x)}\right] - \log 4$$

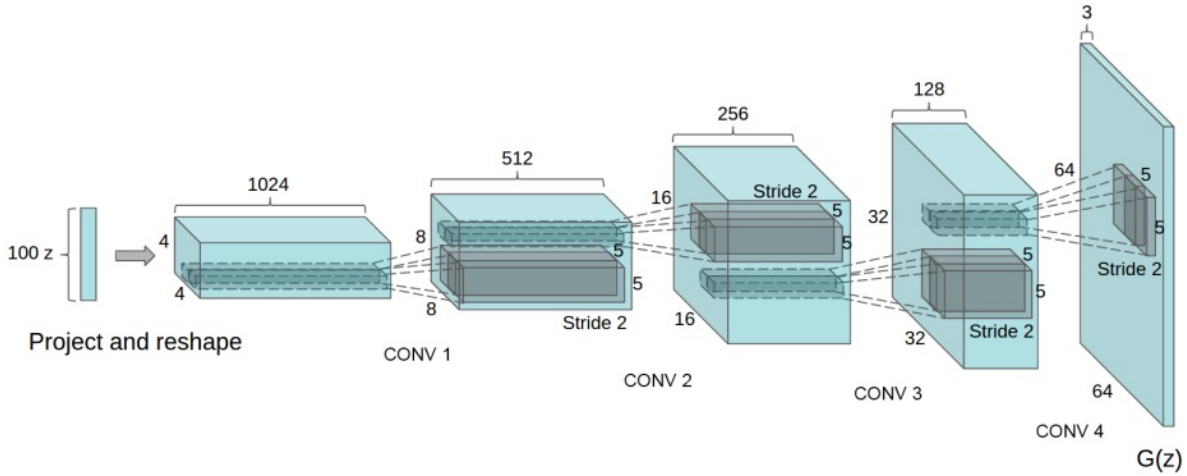
$$= KL(p_{data}||\frac{p_{data}(x)+p_g(x)}{2}) + KL(p_{data}||\frac{p_{data}(x)+p_g(x)}{2}) - \log 4$$

$$= 2*JSD(p_{data}||p_g) - log4$$

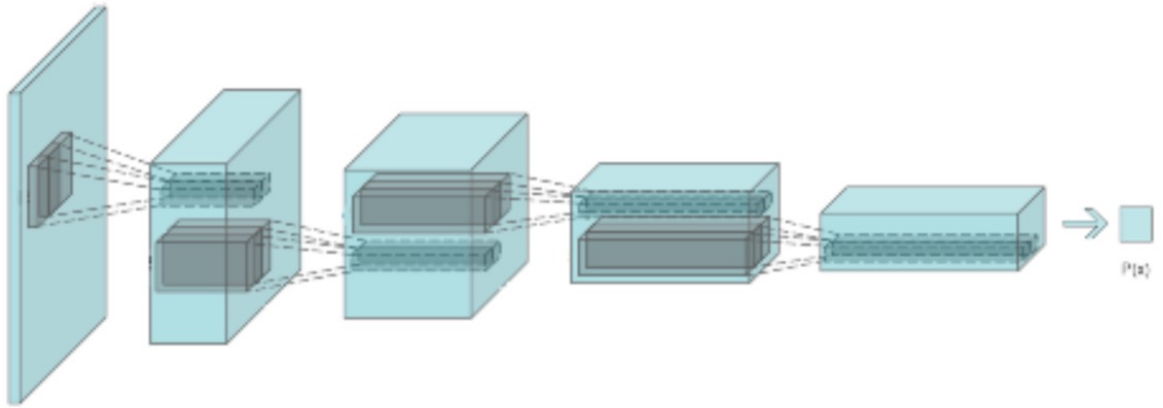
1. GAN

DCGAN

Generator



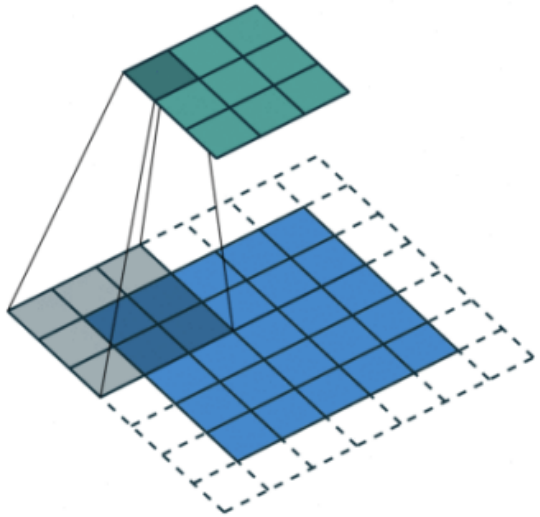
Discriminator



1. GAN

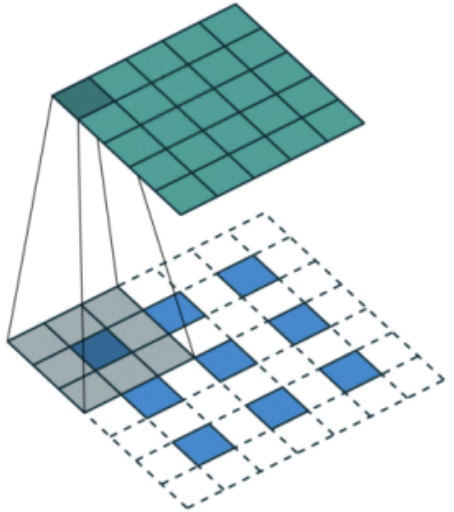
DCGAN

판별자(Discriminator)



Strided Convolution: 너비와 높이가 감소

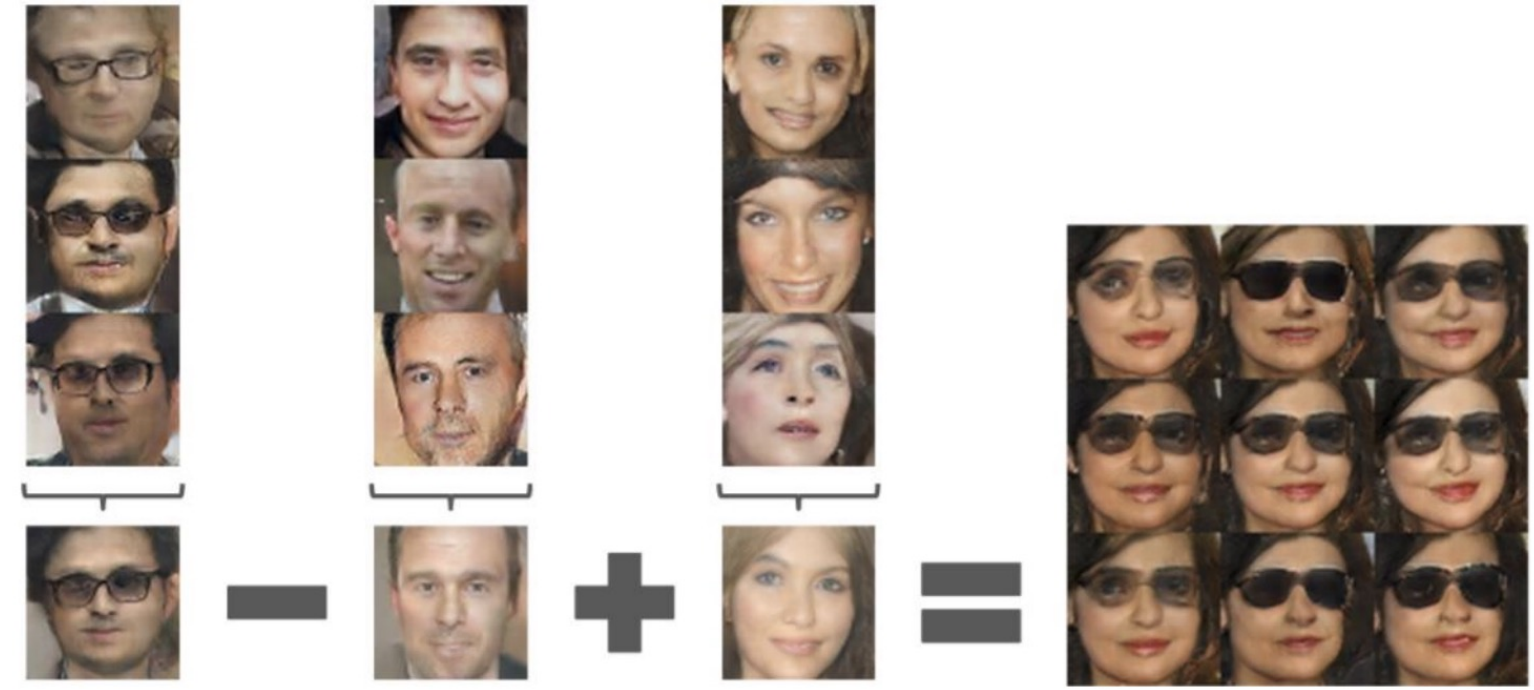
생성자(Generator)



Transposed Convolution: 너비와 높이가 증가

1. GAN

DCGAN


Man
with glasses
Man
without glasses
Woman
without glasses
Woman with glasses

1. GAN

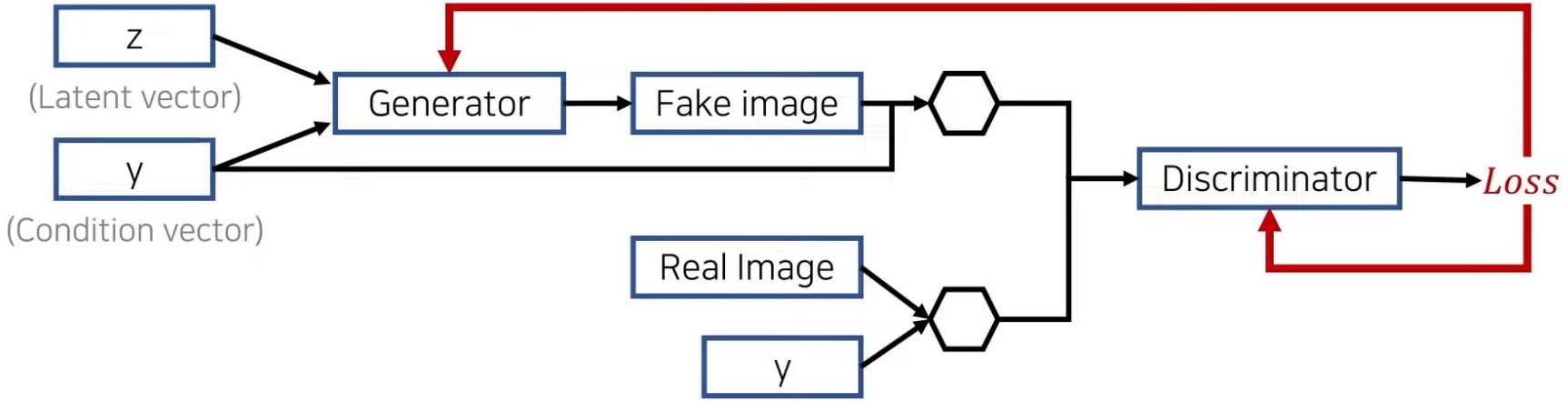
단점

- mode_collapse가 일어난다
- 생성하고자 하는 이미지의 조건을 정할 수 없다

2. Conditional GAN

개요

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



2. Conditional GAN

개요

2. Conditional GAN

Pix2Pix

- L1 loss를 이용하여 GAN 성능 향상 (ground-truth 와 유사)
- L2 loss에 비해 blurring 현상이 덜 발생

$$\mathrm{G}^{*} = \arg\min_{\mathrm{G}} \max_{\mathrm{D}} \mathcal{L}_{cGAN}(\mathrm{G},\mathrm{D}) + \lambda\, \mathcal{L}_{\mathrm{L1}}(\mathrm{G},\mathrm{D})$$

$$\mathcal{L}_{cGAN}(\mathrm{G},\mathrm{D}) = E_{x,\mathrm{y}}\left[logD(x,\mathrm{y})\right] + E_{\mathrm{x,z}}\left[\log\left(1 - D\left(G(\mathrm{x},\mathrm{z})\right)\right)\right]$$

$$\mathcal{L}_{\mathrm{L1}}(\mathrm{G}) = \mathrm{E}_{\mathrm{x,y,z}}[\left|\left|\mathrm{y} - \mathrm{G}(\mathrm{x},\mathrm{z})\right|\right|_1]$$

2. Conditional GAN

Pix2Pix loss

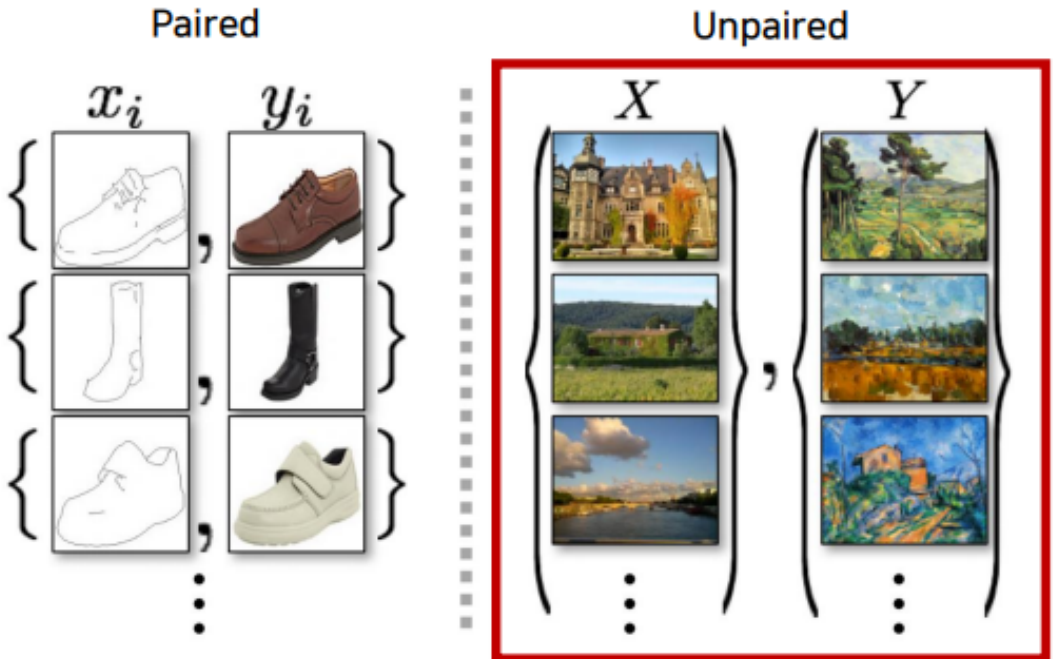
- 여전히 blurry한 결과를 보임
- cGan loss만 사용하는 경우 visual artifact가 존재함
- 두 loss를 섞어 사용함으로써 우수한 결과가 나옴

2. Conditional GAN

Pix2Pix 단점

- 서로 다른 두 도메인의 데이터를 한쌍으로 묶어서 학습을 진행

➔ 데이터를 모으기 힘듦

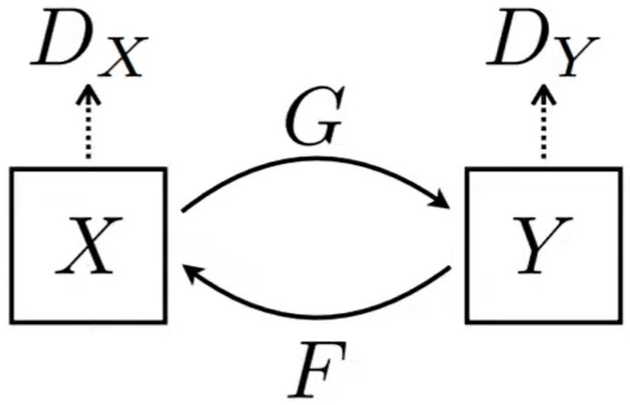


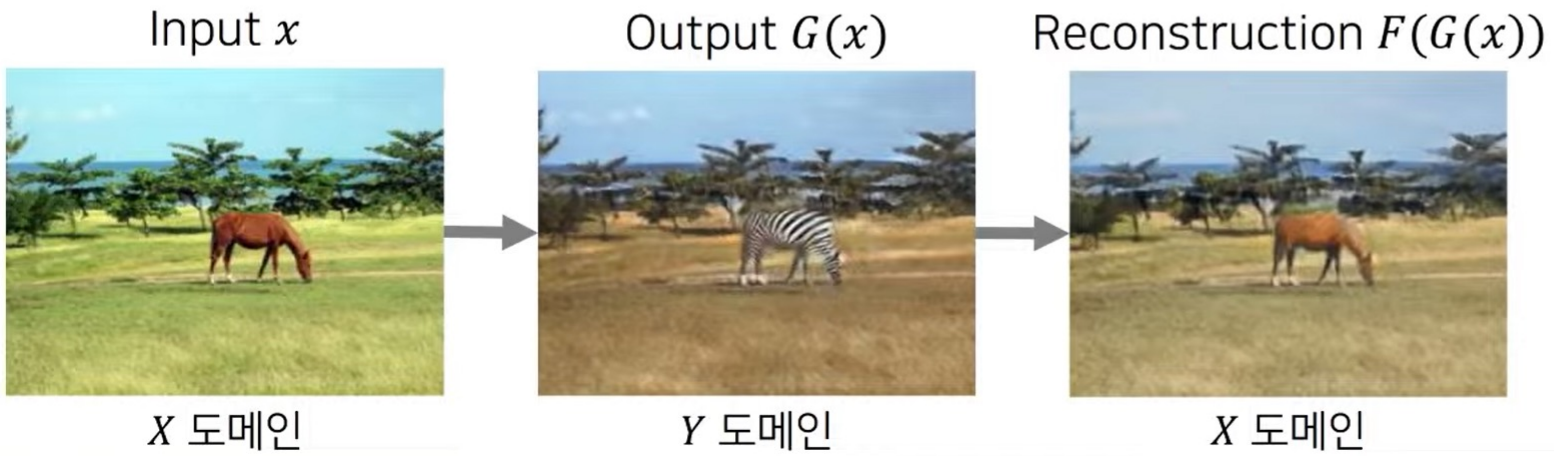
3. CYCLE GAN

개요

목적: paired 되지 않은 데이터셋으로 image to image translation

- 해당 목적을 위해 두 개의 변환기를 사용
  - G: X -> Y
  - F: Y -> X





3. CYCLE GAN

Objective function

$$\mathcal{L}(\mathrm{G}, \mathrm{F}, \mathrm{D_X}, \mathrm{D_y}) = \mathcal{L}_{\mathrm{GAN}}(\mathrm{G}, \mathrm{D_Y}, \mathrm{X}, \mathrm{Y}) + \mathcal{L}_{\mathrm{GAN}}(\mathrm{G}, \mathrm{D_X}, \mathrm{Y}, \mathrm{X}) + \lambda \mathcal{L}_{\mathrm{cyc}}(\mathrm{G}, \mathrm{F})$$

- $\mathcal{L}_{\mathrm{GAN}}(\mathrm{G}, \mathrm{D_Y}, \mathrm{X}, \mathrm{Y}) = E_{\mathrm{y}\sim p_{data}(\mathrm{y})}[logD_{\mathrm{Y}}(\mathrm{y})] + E_{\mathrm{x}\sim p_{\mathrm{data}}(\mathrm{x})}[\log\left(1 - D_{\mathrm{Y}}\left(G(\mathrm{x})\right)\right)$

- $\mathcal{L}_{\mathrm{cyc}}(\mathrm{G}, \mathrm{F}) = E_{\mathrm{x}\sim p_{\mathrm{data}}(\mathrm{x})}[\left|\left|F\left(G(x)\right) - x\right|\right|_1] + E_{\mathrm{y}\sim p_{data}(\mathrm{y})}[\left|\left|F\left(G(\mathrm{y})\right) - \mathrm{y}\right|\right|_1]$

3. CYCLE GAN

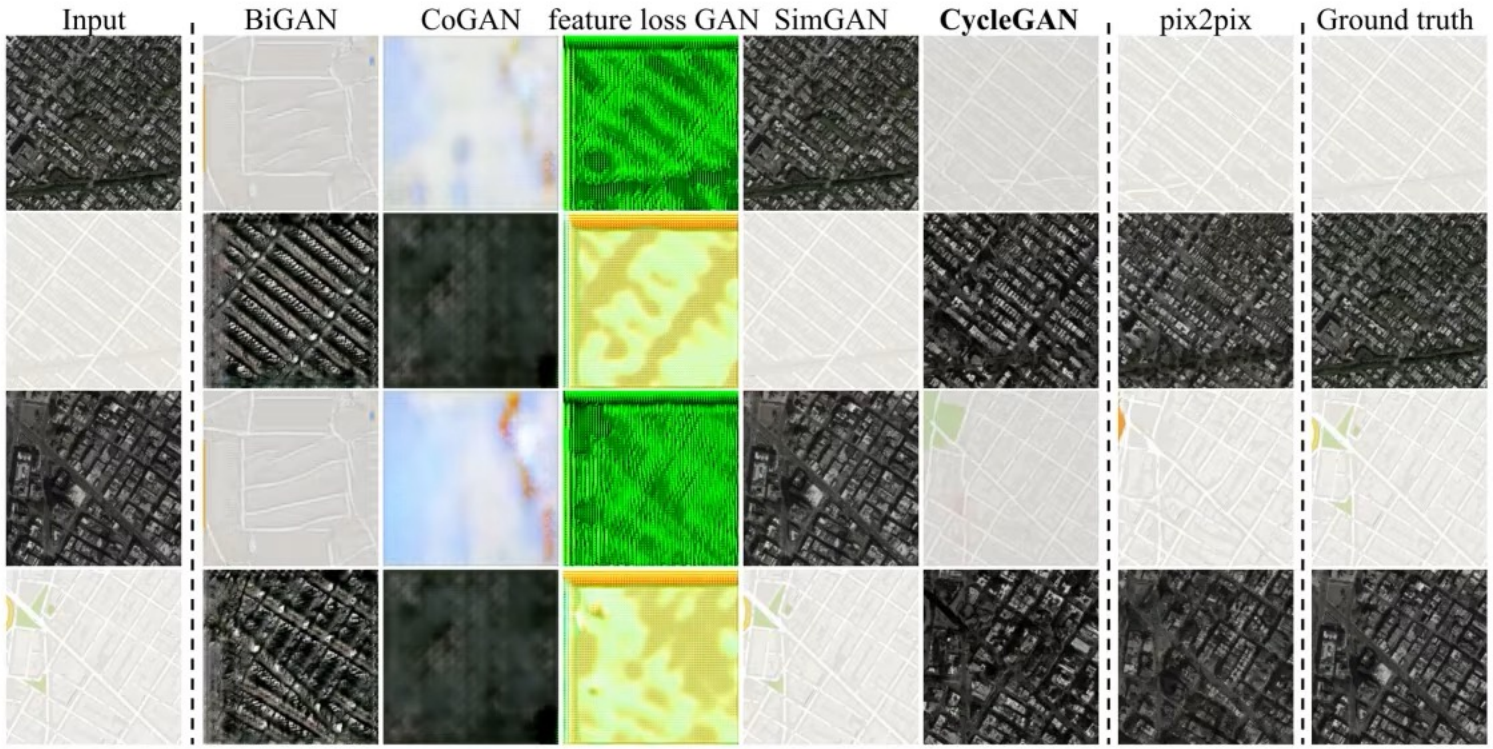
구현

목적: paired 되지 않은 데이터셋으로 image to image translation

- 아키텍처
  - Instance nomarlization
- 학습 방법
  - Least-Square loss
  - Replay beffer

3. CYCLE GAN

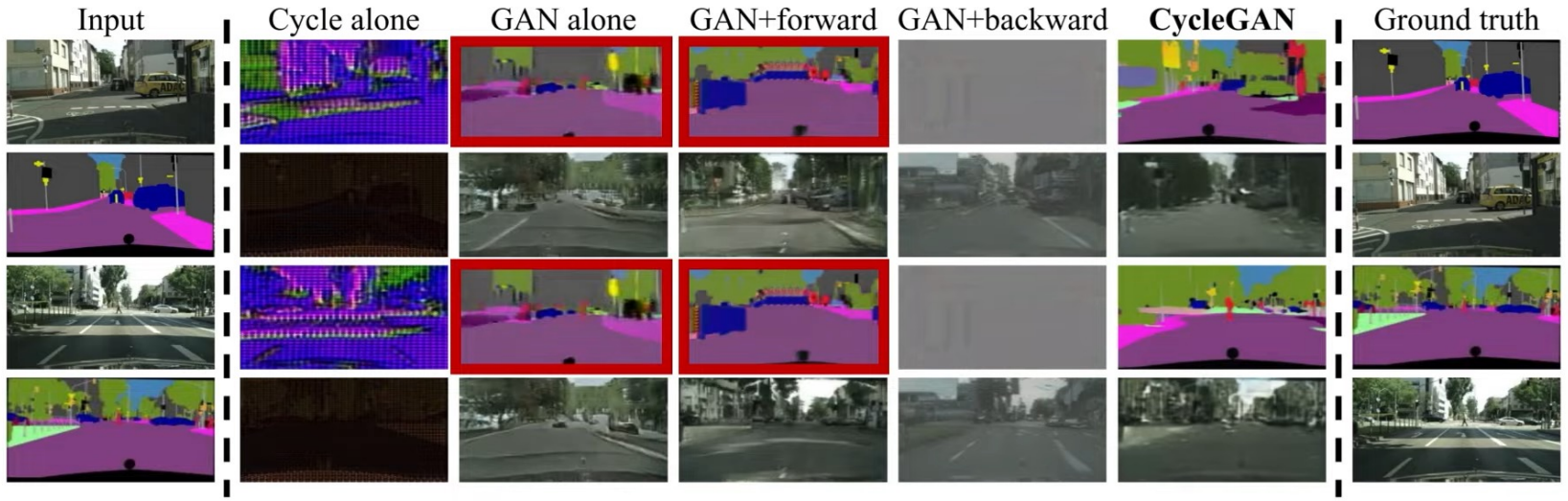
성능



3. CYCLE GAN

Loss에 따른 성능

모든 loss function을 사용했을 때, 우수한 결과가 나온다



3. CYCLE GAN

Identity loss

목적: 색상 구성을 보존해야 할 때, 사용한다

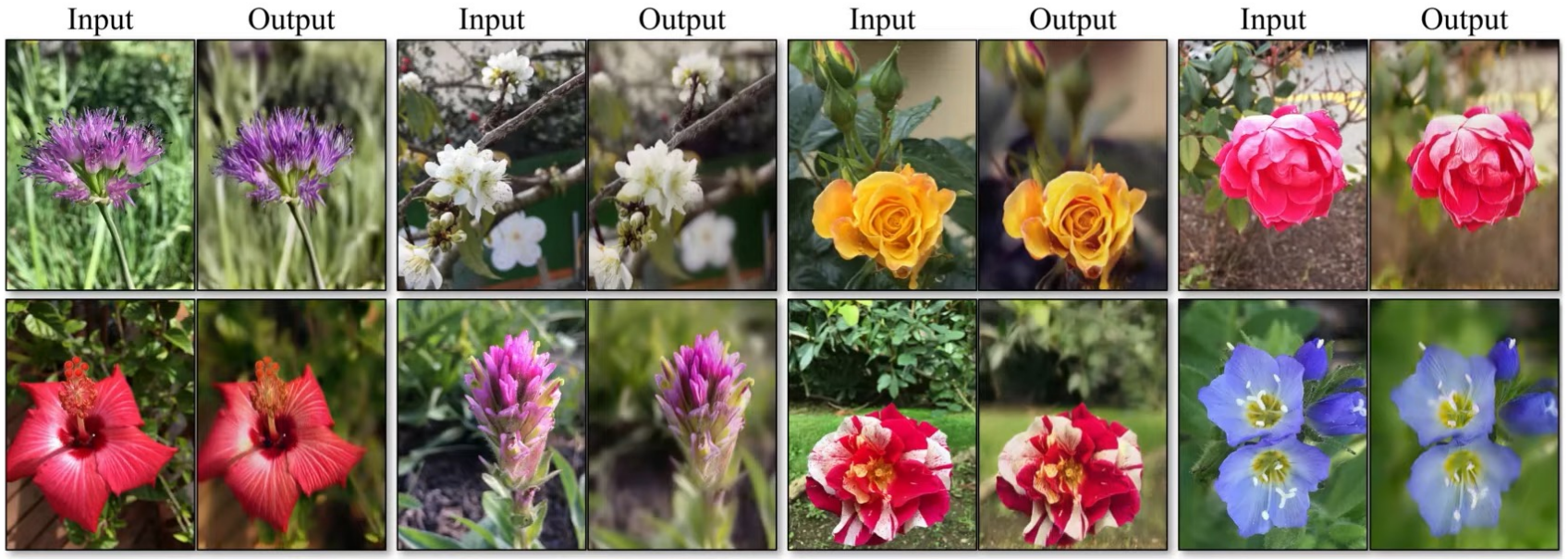
$$\mathcal{L}_{\text{identity}}(\mathrm{G},\mathrm{F}) = \mathrm{E}_{\mathrm{y}\sim \mathrm{p}_{\mathrm{data}}(\mathrm{y})}[\left|\left|G(y)-y\right|\right|_1] + \mathrm{E}_{\mathrm{x}\sim \mathrm{p}_{\mathrm{data}}(\mathrm{x})}[\left|\left|G(\mathrm{x})-\mathrm{x}\right|\right|_1]$$

3. CYCLE GAN

Style Transfer 성능

3. CYCLE GAN

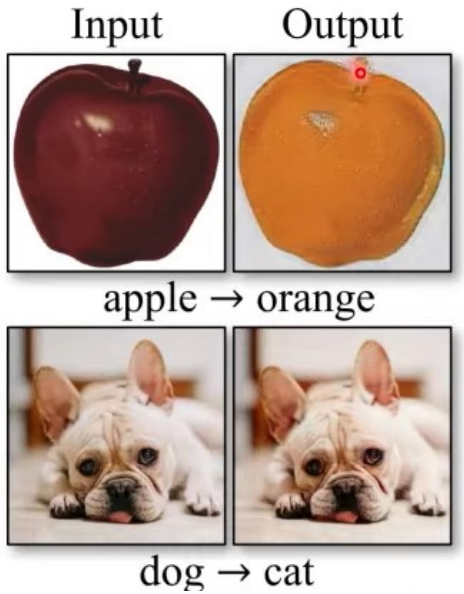
성능

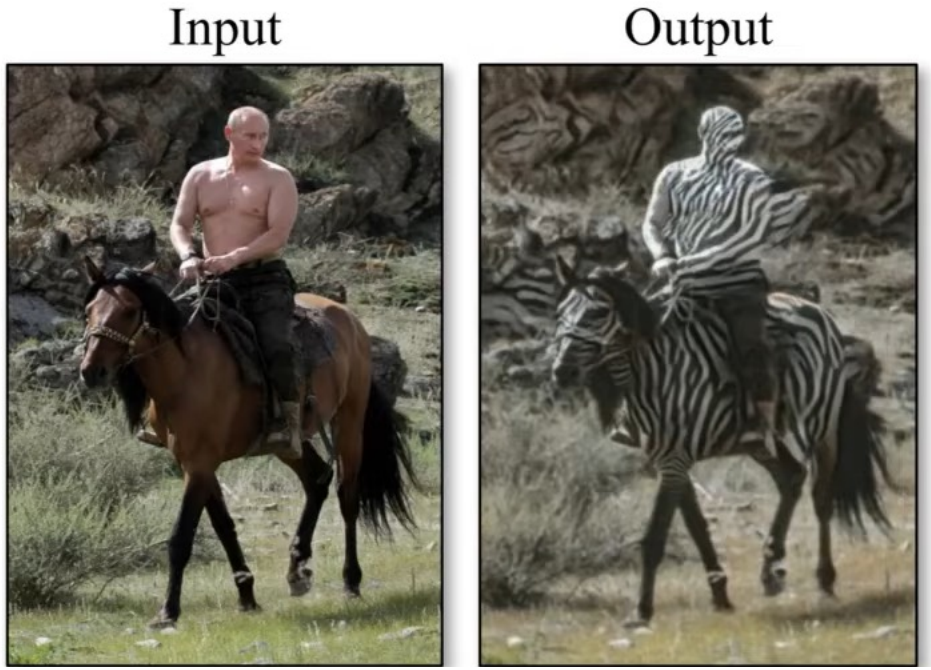


3. CYCLE GAN

단점

- 모양 정보를 포함한 transfer가 필요한 경우
- 학습 데이터에 포함되지 않은 사물을 처리하는 경우

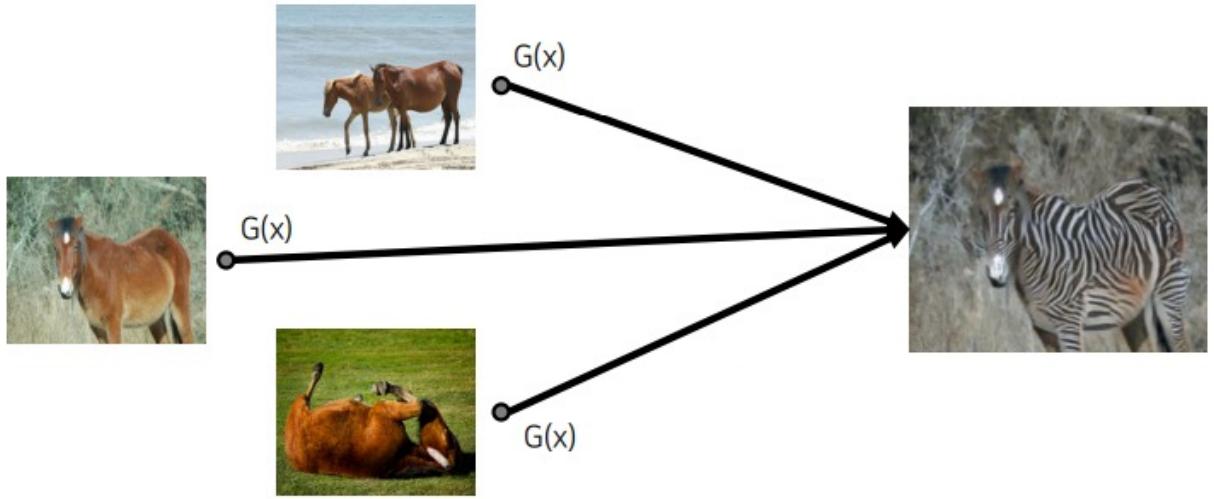




3. CYCLE GAN

단점2

- 별도의 제약 조건 없이 이미지를 바꾸고자 한다면 입력과 관계없이 어떤 도메인에 대해서는 하나의 이미지만 제시할 수도 있다



4. WGAN-GP

원리

- 함수가 1-Lipshichtz 조건을 만족하도록 하여 안정적인 학습 유도
  - Weight clipping을 이용하여 제약 조건 만족
- Gradient penalty로 WGAN 성능을 개선

4. WGAN-GP

LIpshitz

- Gradient의 변화에 제약을 건다

$$\forall x, \forall y, |f(x) - f(y)| \leq \mathcal{L}\left||x - y|\right|_2$$

4. WGAN-GP

Wasserstein loss

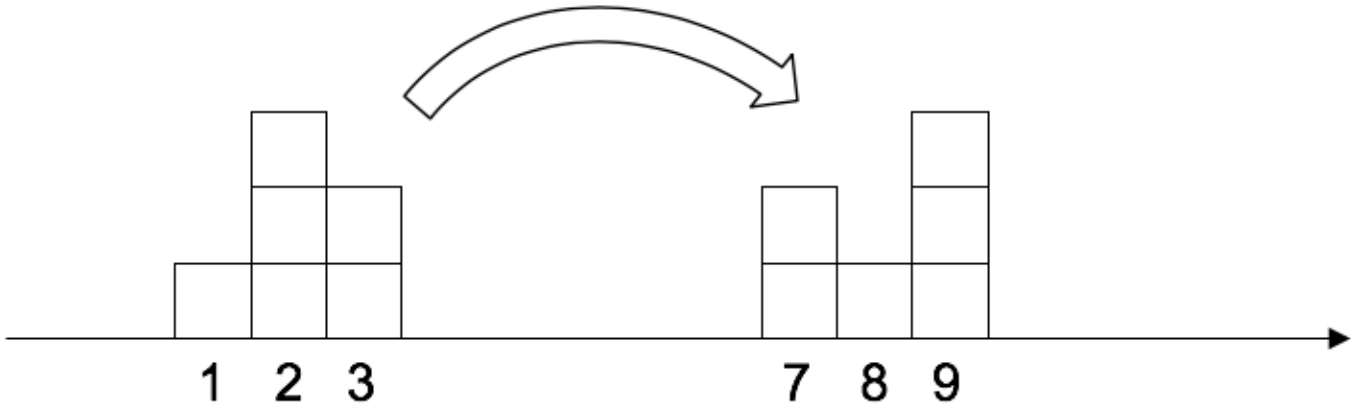
- Distance의 종류이다

$$W(P_r, P_g) = \inf_{\gamma \in \prod(P_r, P_g)} E_{(x,y)\sim\gamma}[||x - y||]$$

4. WGAN-GP

Wasserstein loss 수식 이해

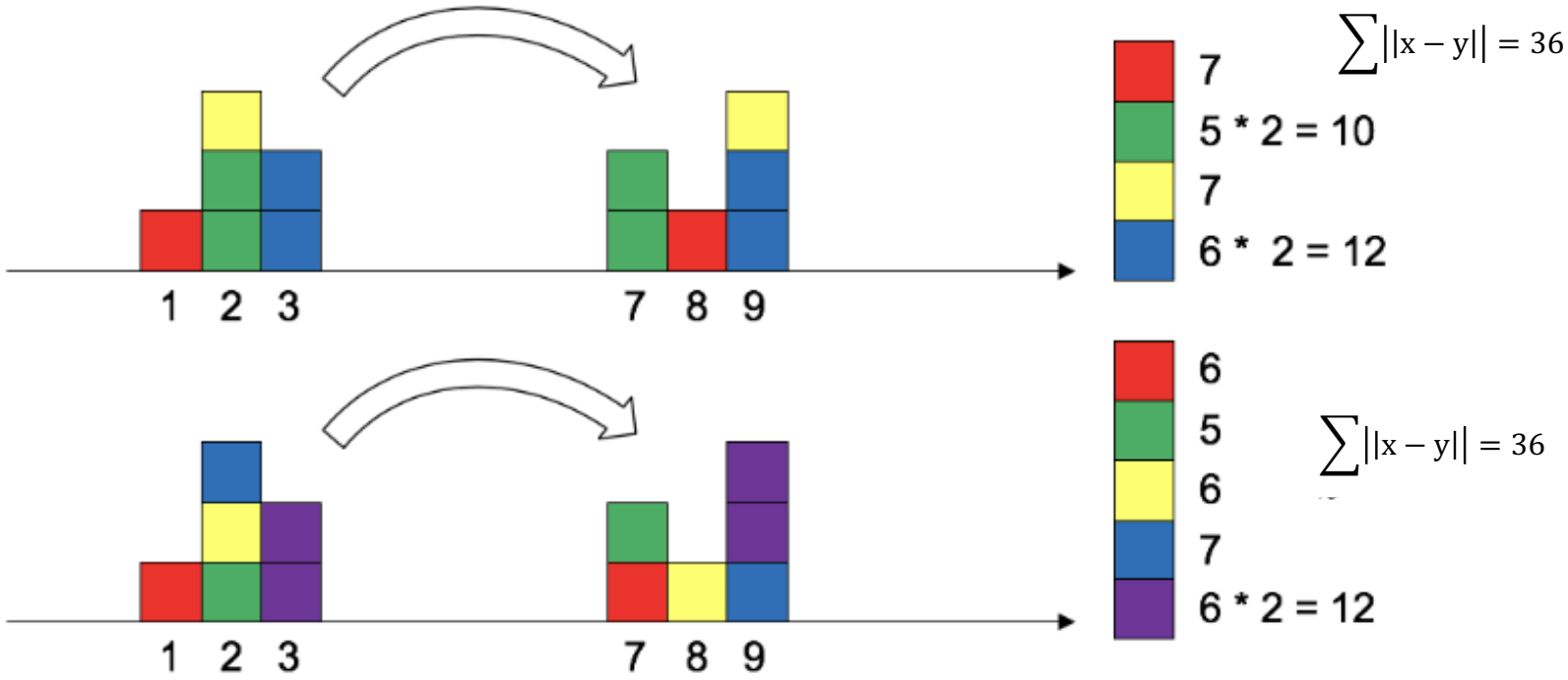
다음 블록을 옮기는데 걸리는 비용을 계산 해보자



4. WGAN-GP

Wasserstein loss

블록을 옮기는 예시 2개



4. WGAN-GP

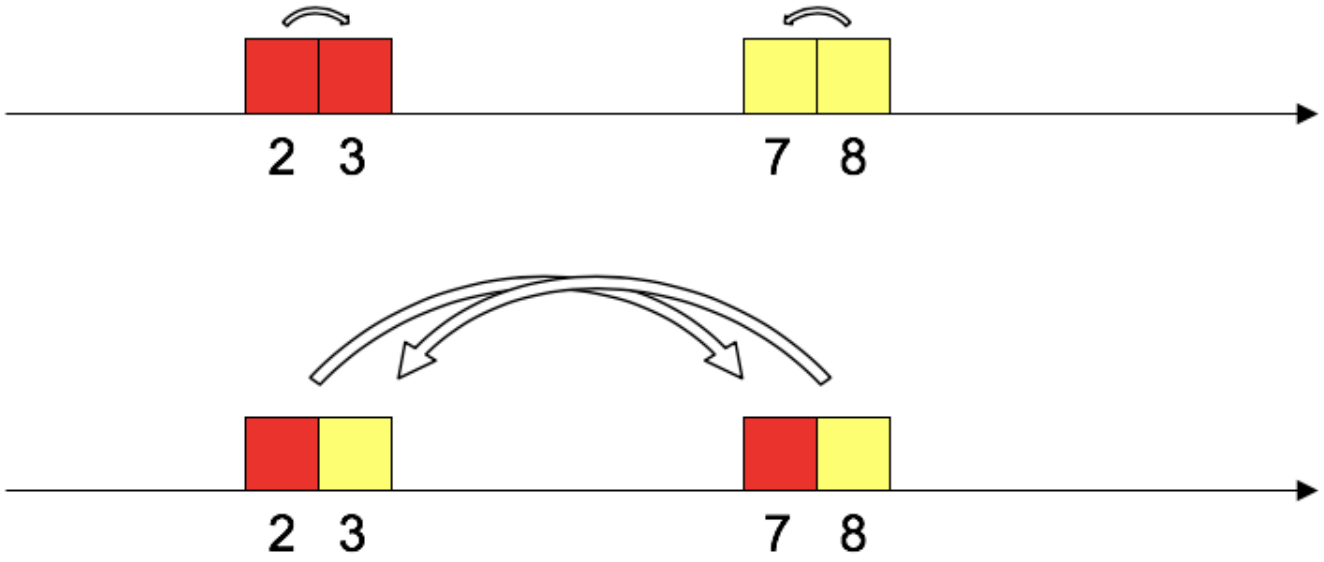
Wasserstein loss

비용이 달라지는 예시(검은색과 회색을 바꾸어보자)

4. WGAN-GP

Wasserstein loss

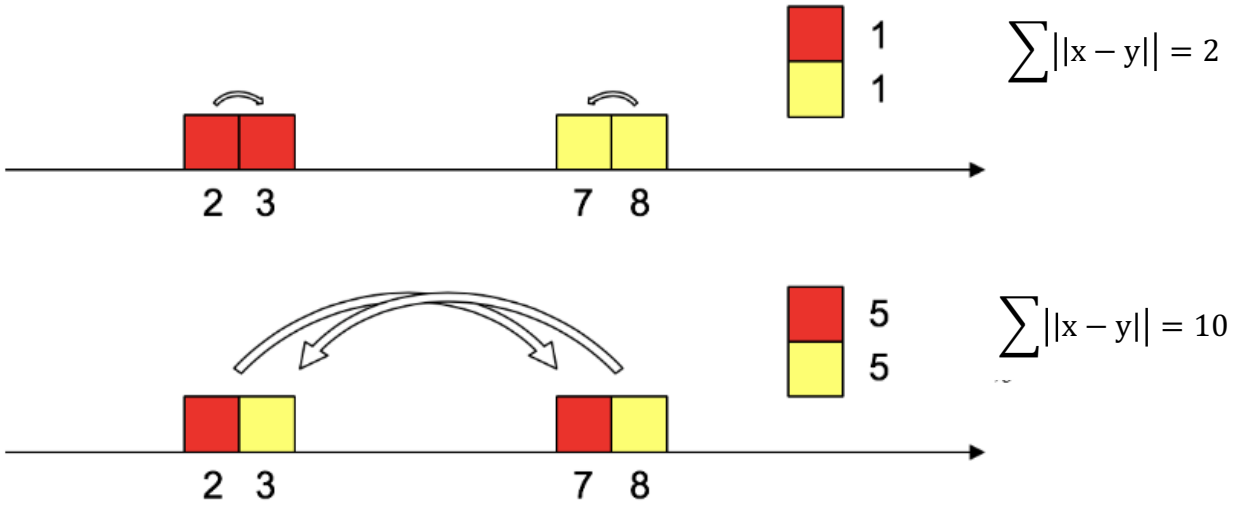
비용이 달라지는 예시(검은색과 회색을 바꾸어보자)



4. WGAN-GP

Wasserstein loss

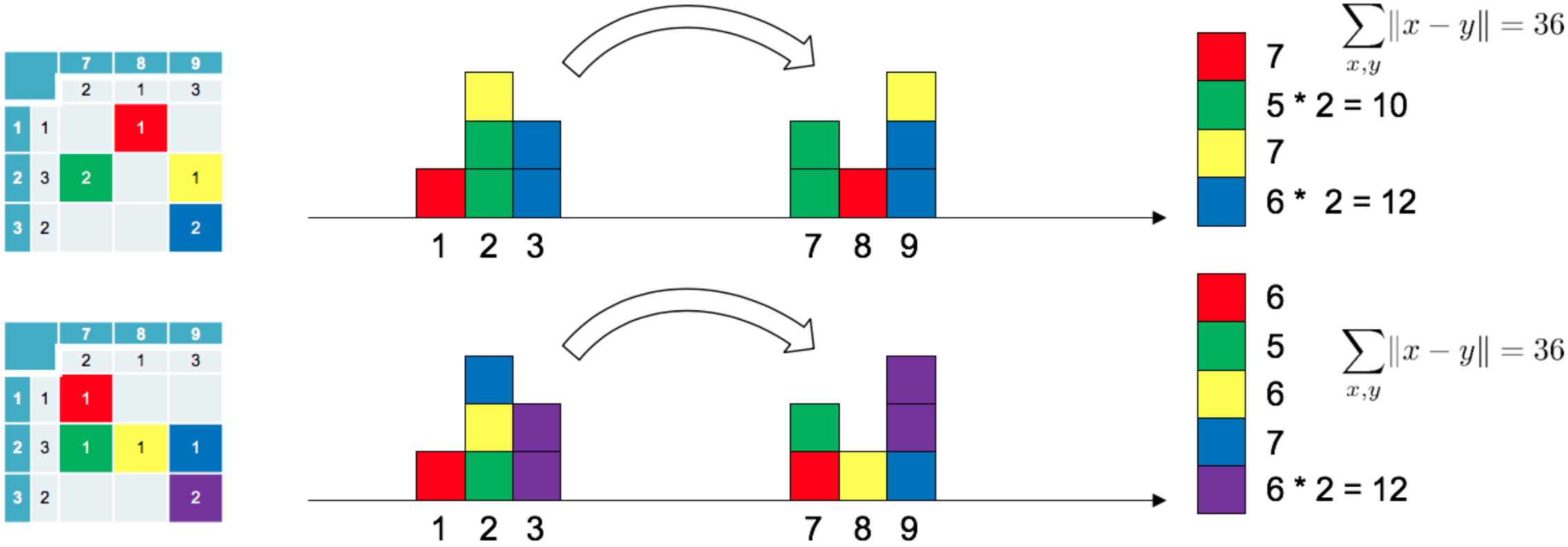
- 비용이 달라졌다!



4. WGAN-GP

Wasserstein loss

- Joint probability distribution으로 볼 수 있다



4. WGAN-GP

Wasserstein loss

$$W(P_r, P_g) = \gamma_{X,Y}(1,8)\times|1-8| + \gamma_{X,Y}(2,7)\times|2-7| + \gamma_{X,Y}(2,9)\times|2-9| + \gamma_{X,Y}(3,9)\times|3-9|$$

$$= \frac{1}{6}\times 7 + \frac{1}{3}\times 5 + \frac{1}{6}\times 7 + \frac{1}{3}\times 6 = 6$$

		7	8	9
		2	1	3
1	1		1	
2	3	2		1
3	2			2

		7	8	9
		1/3	1/6	1/2
1	1/6		1/6	
2	1/2	1/3		1/6
3	1/3			1/3

4. WGAN-GP

Wasserstein loss

- Kantorovich_Rubinsein Duality Theorem

$$W(P_r, P_g) = \inf_{\gamma \in \prod(P_r, P_g)} E_{(x,y)\sim\gamma}[||x - y||]$$

$$W(P_r, P_g) = \sup_{||f||_L \leq 1} E_{x\sim P_r,}[f(x)] - E_{x\sim P_g}[f(x)]$$

$$W(P_r, P_g) = \max_{w \leq W} E_{x\sim P_r,}[f(x)] - E_{z\sim P(Z)}[f(G(z))]$$

4. WGAN-GP

Wasserstein loss

- Weight clipping

$$W = [-0.01, 0.01]^d$$

4. WGAN-GP

loss

$$L = E_{x` \sim P_{g,}}[D(x`)] - E_{x \sim P_r}[D(x)] + \lambda_{x` \sim P_g} E[\left( ||\nabla D(x`)||_2 - 1 \right)^2]$$

(Unsupervised case)

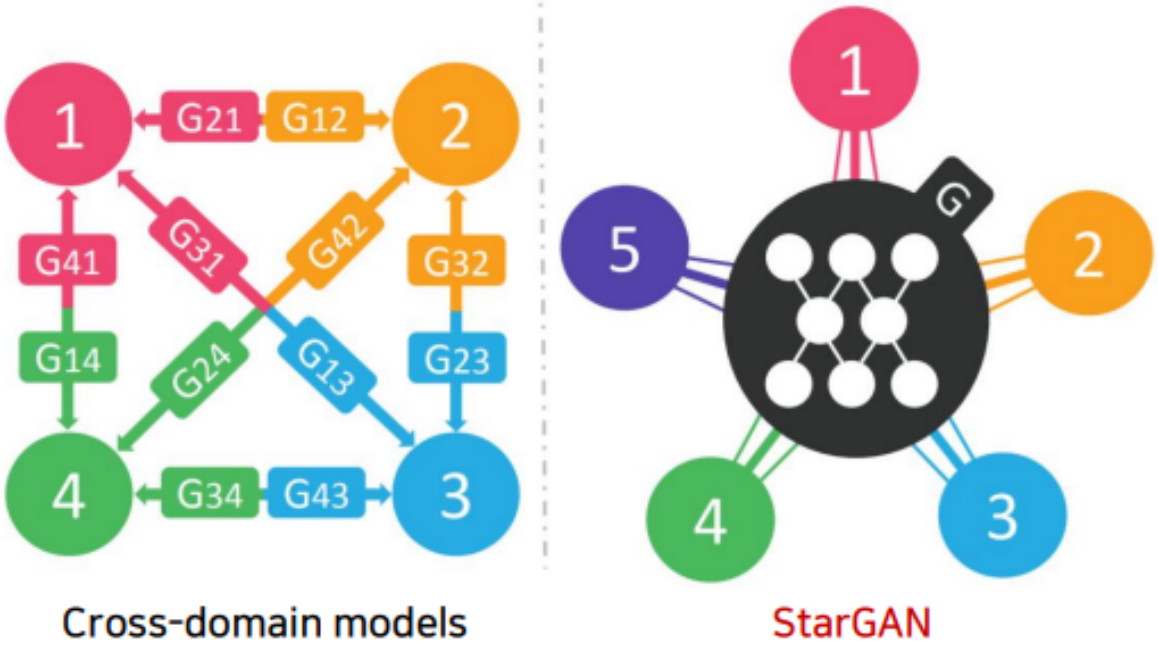
Inception Score 측정 결과

Method	Score
ALI [8] (in [27])	$5.34 \pm .05$
BEGAN [4]	5.62
DCGAN [22] (in [11])	$6.16 \pm .07$
Improved GAN (-L+HA) [23]	$6.86 \pm .06$
EGAN-Ent-VI [7]	$7.07 \pm .10$
DFM [27]	$7.72 \pm .13$
WGAN-GP ResNet (ours)	$7.86 \pm .07$

5. STAR GAN

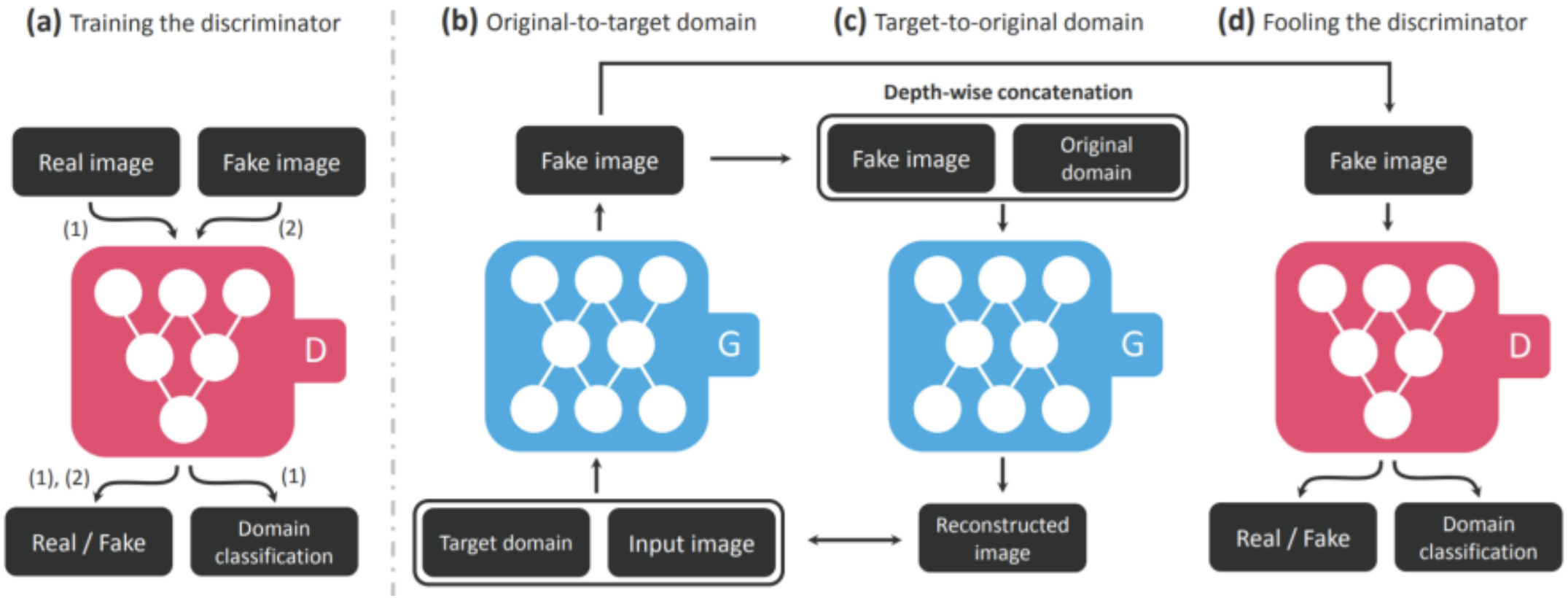
개요

목적: 하나의 뉴럴 네트워크를 통해 다중 도메인 사이에서의 이미지 변환 가능



5. STAR GAN

개요


(a) Training the discriminator
(b) Original-to-target domain
(c) Target-to-original domain
(d) Fooling the discriminator
Depth-wise concatenation
Real image
Fake image
(1)
(2)
D
(1), (2)
(1)
Real / Fake
Domain classification
Fake image
G
Target domain
Input image
Fake image
Original domain
G
Reconstructed image
Fake image
D
Real / Fake
Domain classification

5. STAR GAN

Loss

- StarGAN: ① Adversarial loss + ② Domain classification loss + ③ Reconstruction loss

Adversarial $$\mathcal{L}_{adv} = \mathbb{E}_x\left[\log D_{src}(x)\right] + \mathbb{E}_{x,c}[\log\left(1 - D_{src}(G(x,c))\right)]$$

Domain classification

$$\begin{cases} \mathcal{L}_{cls}^{f} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))] \\ \mathcal{L}_{cls}^{r} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)] \end{cases}$$

Reconstruction $$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1]$$

최종 목적 함수

$$\begin{cases} \mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}_{cls}^{r} \\ \mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}_{cls}^{f} + \lambda_{rec}\,\mathcal{L}_{rec} \end{cases}$$

5. STAR GAN

Mask Vector

- Multiple dataset에서의 학습을 위하여 mask vector m을 이용할 수 있음

$$\tilde{c} = [c_1, \ldots, c_n, m]$$ ← [·]: Concatenation

CelebA label: Black / Blond / Brown / Male / Young

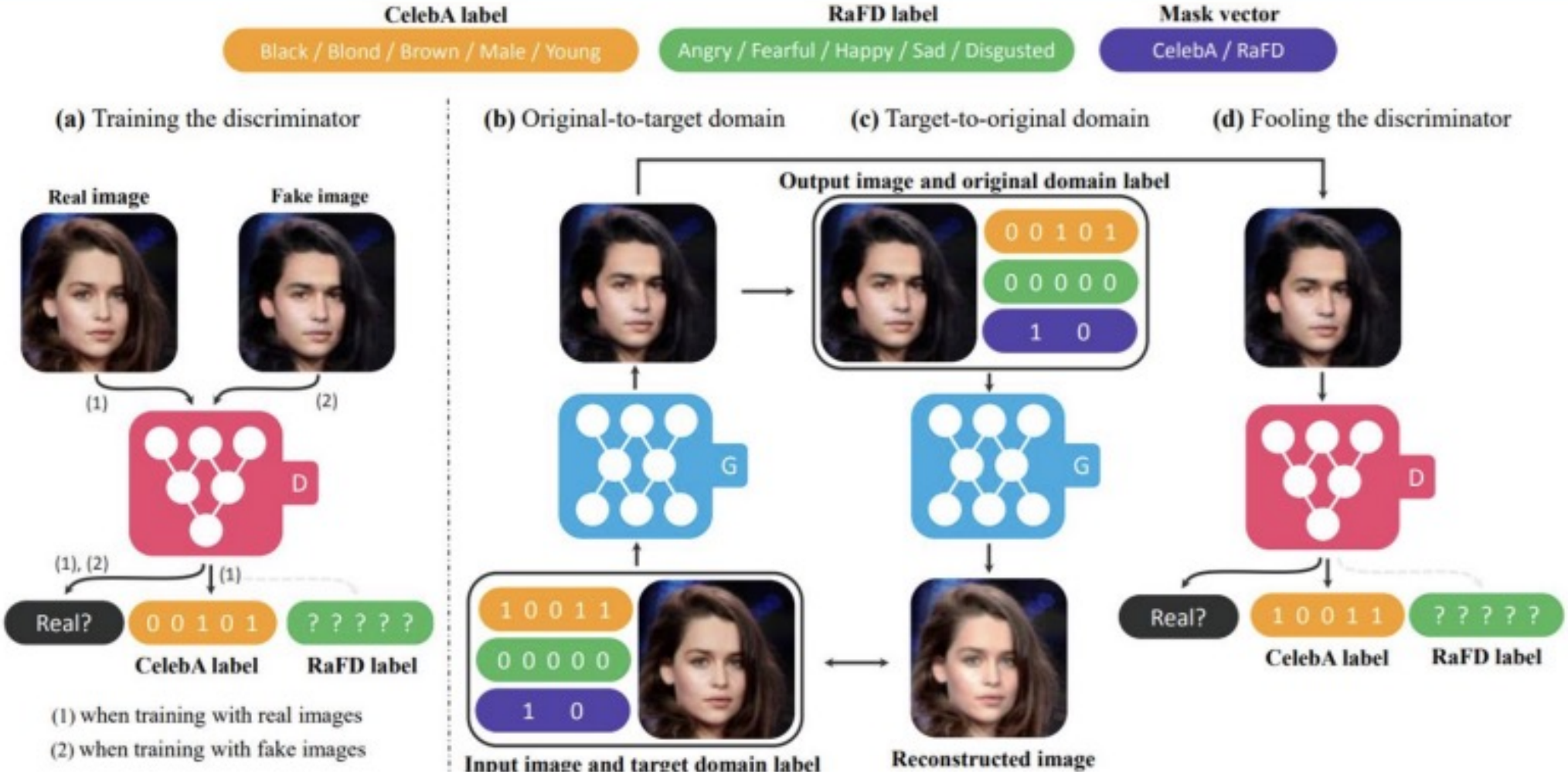
RaFD label: Angry / Fearful / Happy / Sad / Disgusted

Mask vector: CelebA / RaFD

더 많은 데이터를 활용한 쪽이 **높은 성능**

5. STAR GAN

학습


CelebA label
Black / Blond / Brown / Male / Young
RaFD label
Angry / Fearful / Happy / Sad / Disgusted
Mask vector
CelebA / RaFD
(a) Training the discriminator
(b) Original-to-target domain
(c) Target-to-original domain
(d) Fooling the discriminator
Real image
Fake image
(1)
(2)
D
G
Output image and original domain label
0 0 1 0 1
0 0 0 0 0
1 0
(1), (2)
(1)
Real?
0 0 1 0 1
? ? ? ? ?
CelebA label
RaFD label
1 0 0 1 1
0 0 0 0 0
1 0
Input image and target domain label
Reconstructed image
(1) when training with real images
(2) when training with fake images

6. STYLE GAN

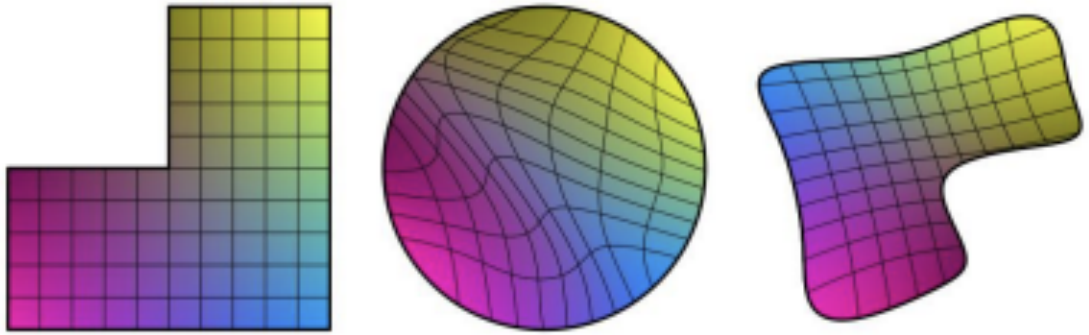
ProGAN

- 메인 아이디어
  - 학습 과정에서 레이어 추가
- 한계점
  - 이미지 특징 제어가 어려움

6. STYLE GAN

Mapping Network

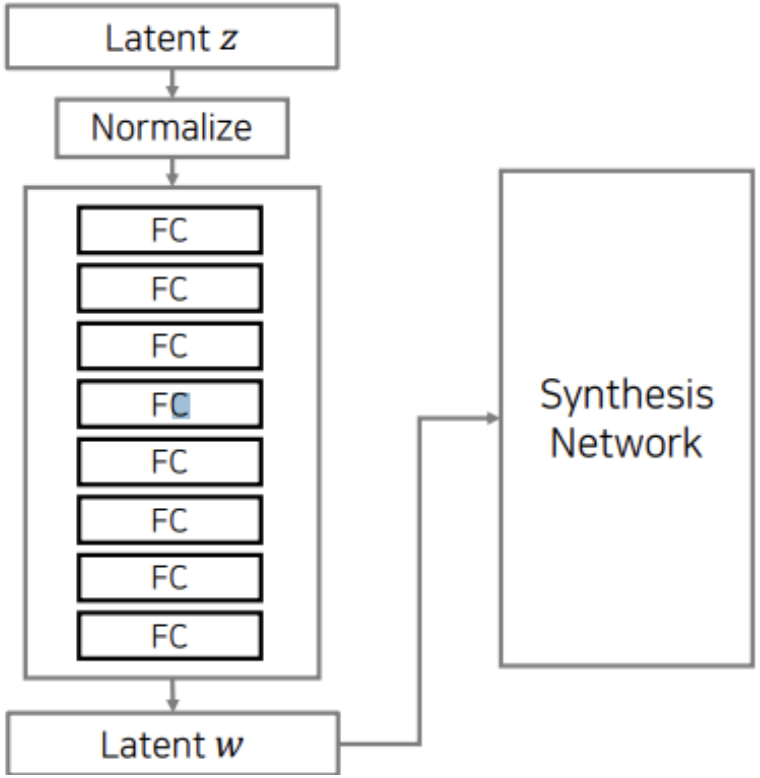
- 512차원의 z도메인에서 w도메인으로 매핑
- W벡터의 성능이 훨씬 좋다
- W벡터와 Z벡터의 크기는 동일하다



(a) Distribution of features in training set

(b) Mapping from $\mathcal{Z}$ to features

(c) Mapping from $\mathcal{W}$ to features

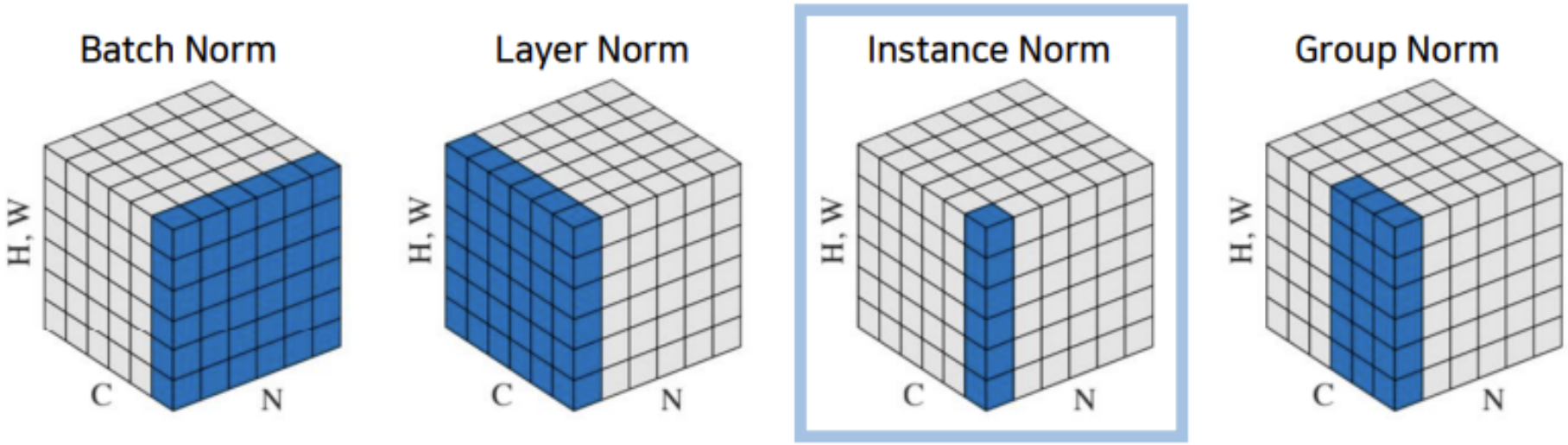


6. STYLE GAN

ADaIN

Adaptive Instance Normalization

- 학습시킬 파라미터가 필요하지 않다
- Style transfer에 좋다



6. STYLE GAN

ADaIN

Batch Normalization

$$BN(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta$$

Instance Normalization

$$IN(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta$$

Adain

$$AdaIN(x,y) = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y)$$

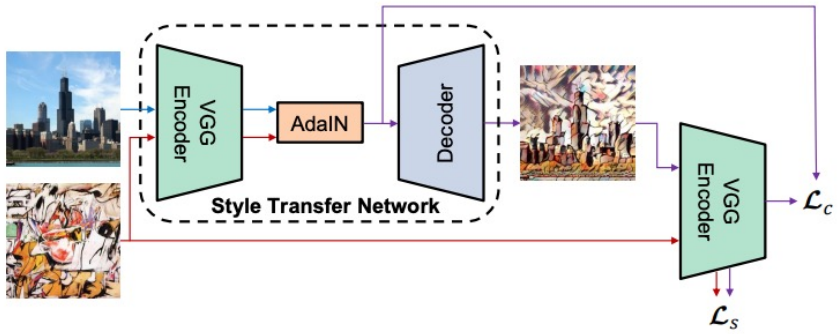
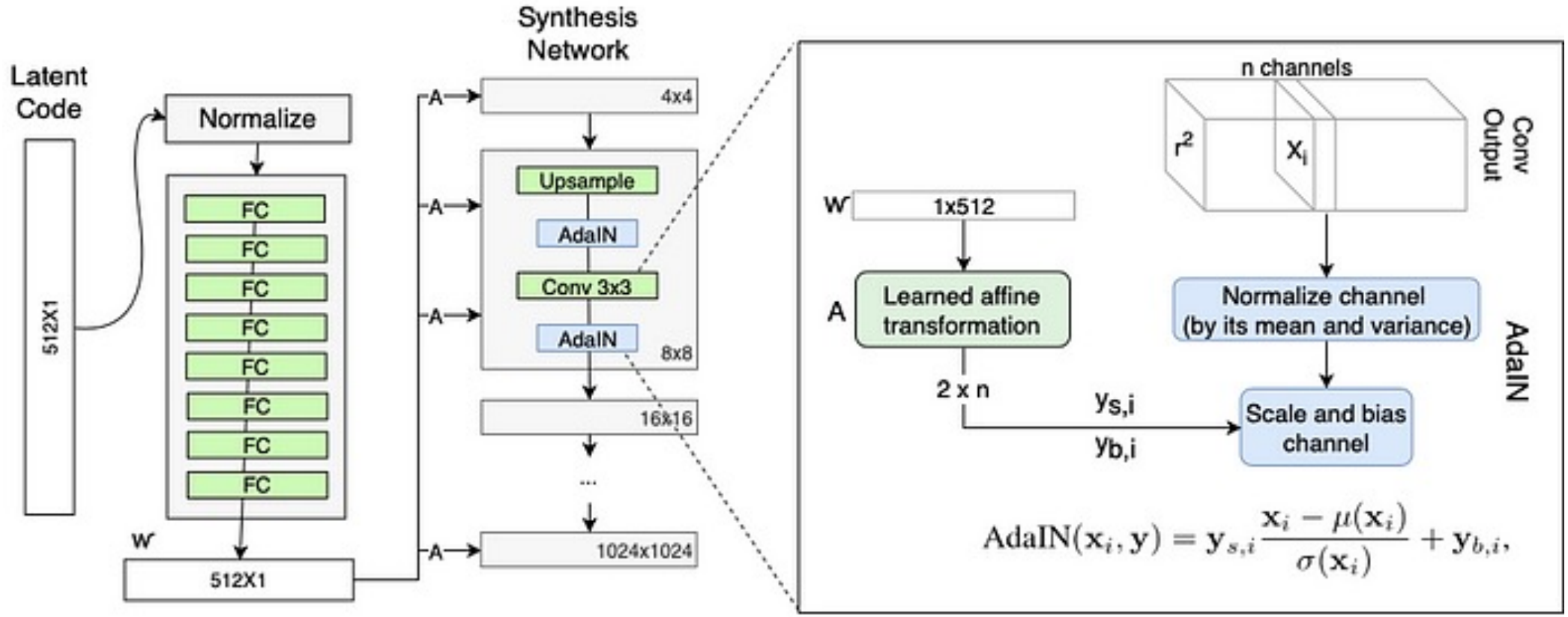


Figure 2. An overview of our style transfer algorithm. We use the first few layers of a fixed VGG-19 network to encode the content and style images. An AdaIN layer is used to perform style transfer in the feature space. A decoder is learned to invert the AdaIN output to the image spaces. We use the same VGG encoder to compute a content loss $\mathcal{L}_c$ (Equ. 12) and a style loss $\mathcal{L}_s$ (Equ. 13).

6. STYLE GAN

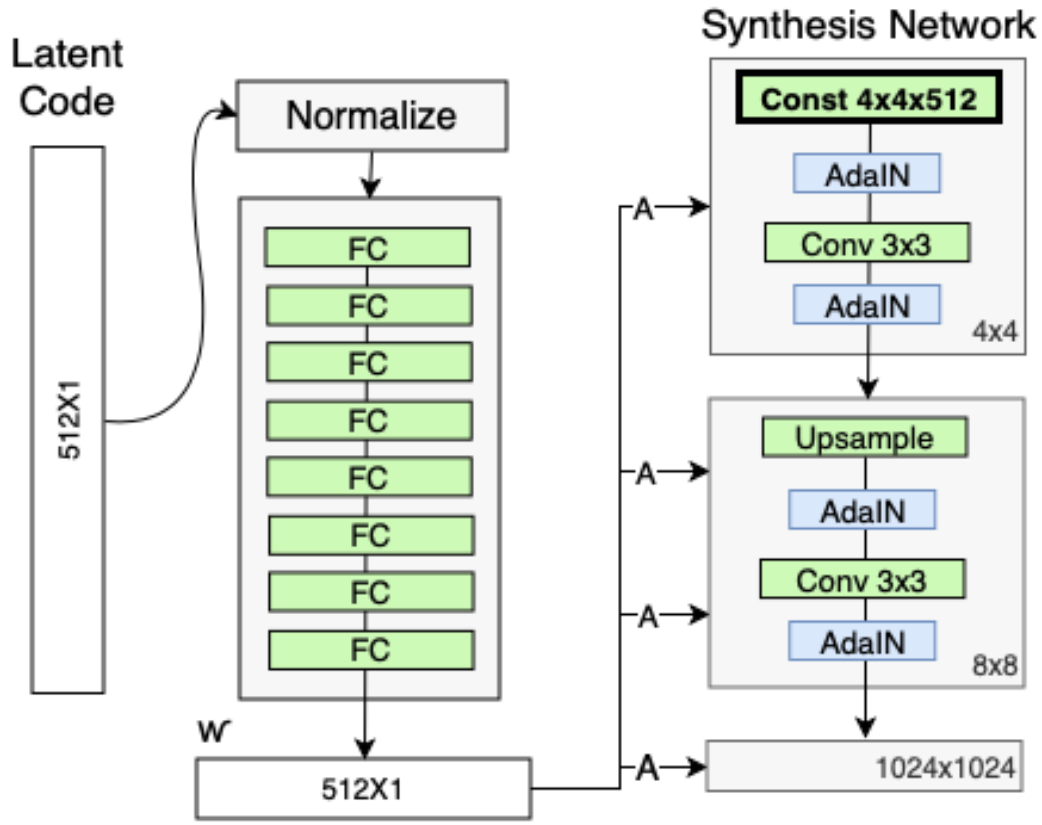
AdaIN



6. STYLE GAN

Removing Traditional Input

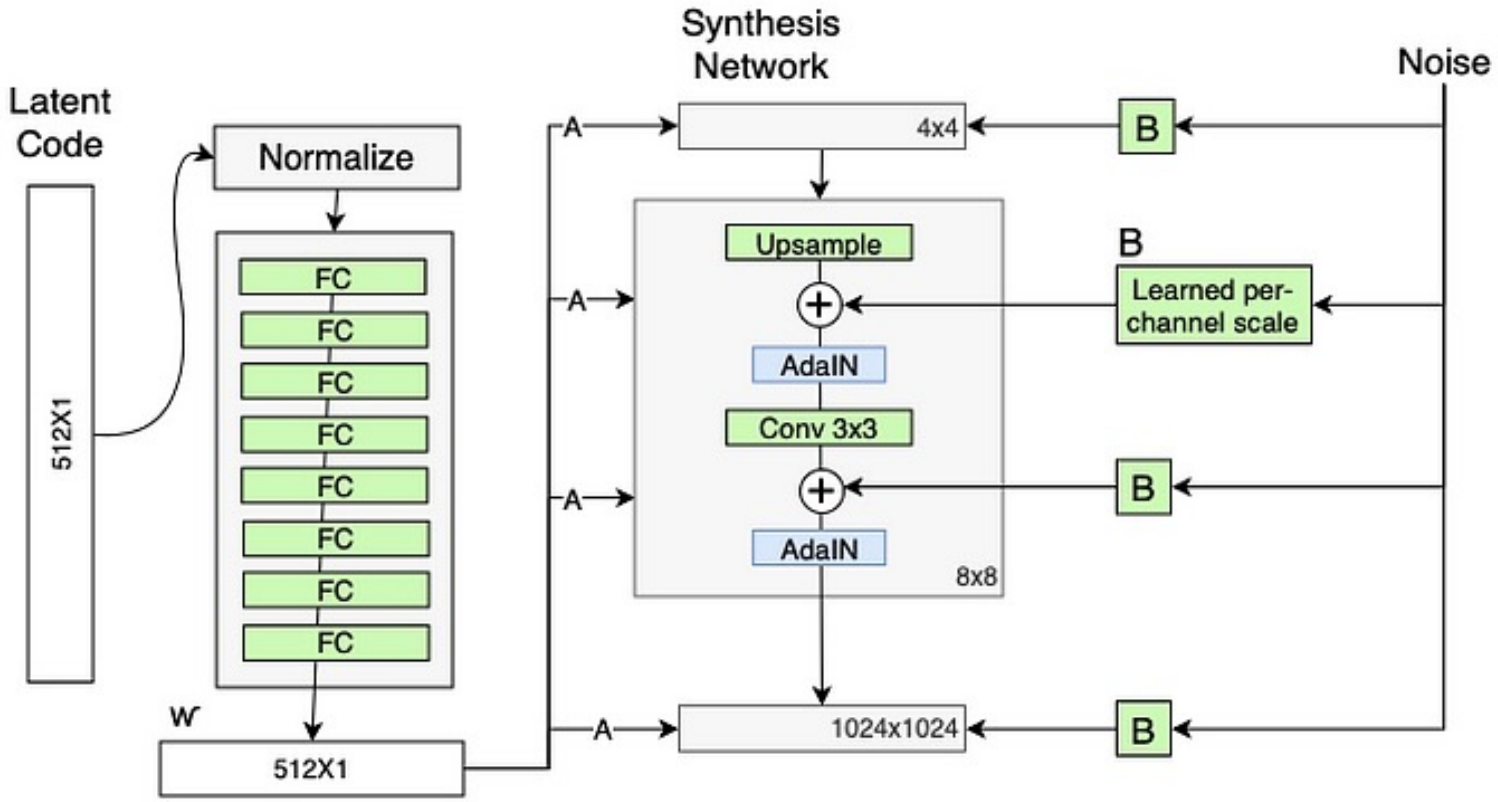
초기 입력값은 확률 분포가 아닌 상수!



6. STYLE GAN

Stochastic Variation

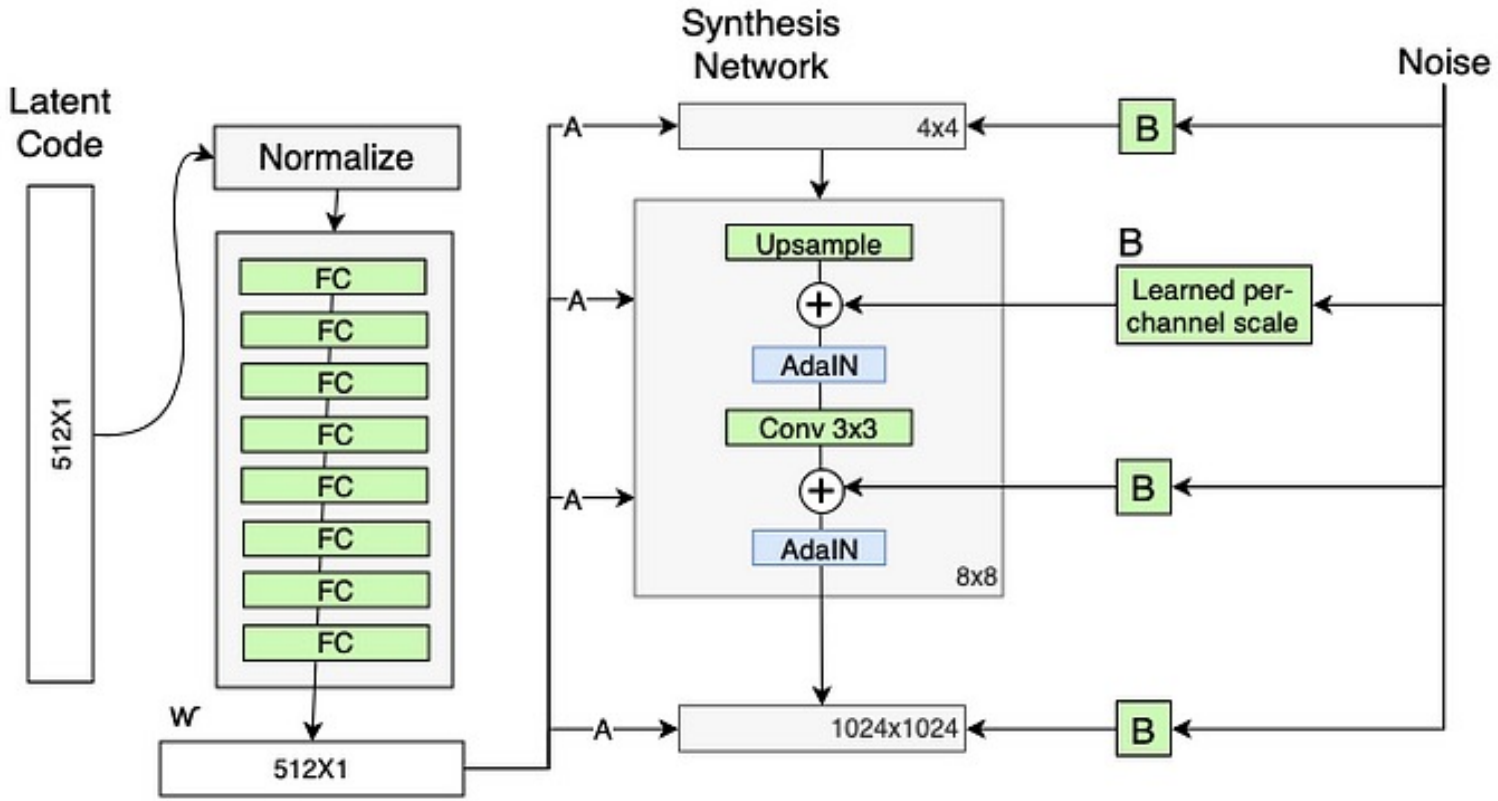
다양한 확률적인 측면 제어



6. STYLE GAN

Stochastic Variation

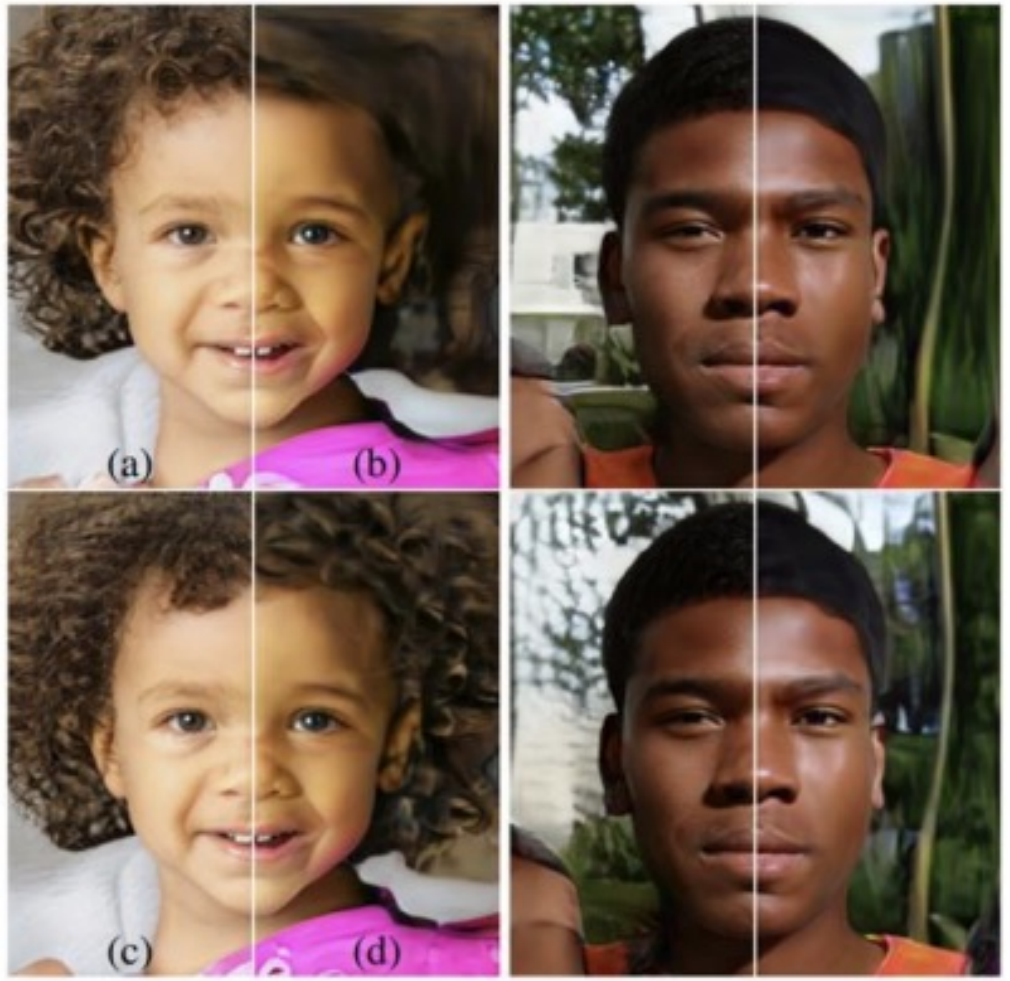
다양한 확률적인 측면 제어



6. STYLE GAN

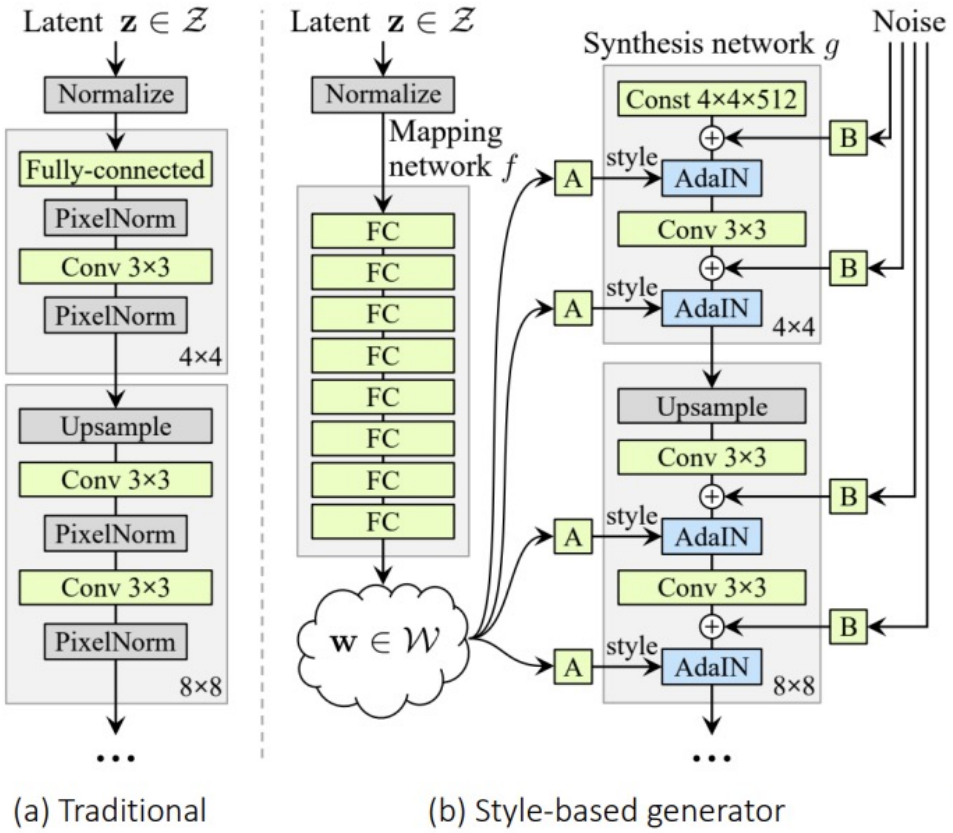
Stochastic Variation

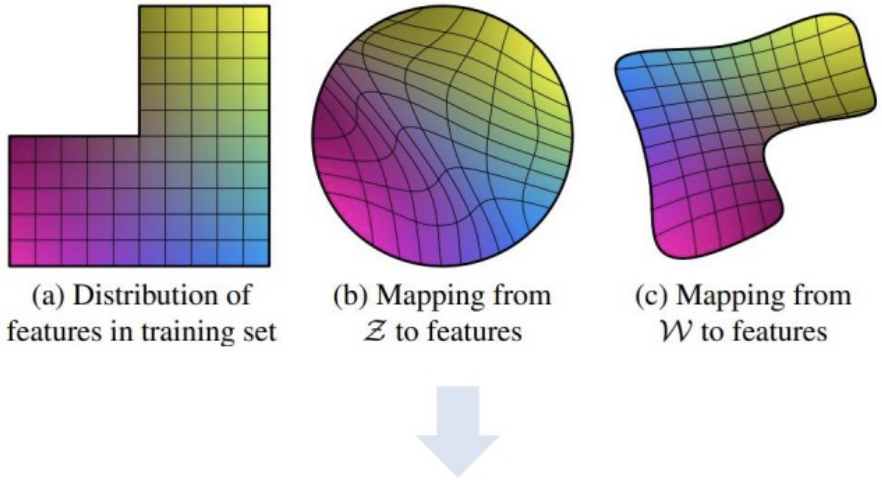
- 스타일: high level global attributes
  - 얼굴형, 안경 유무
- 노이즈: stochastic variation
  - 주근깨, 피부모공, 곱슬거림



6. STYLE GAN

Architecture





(a) Distribution of features in training set

(b) Mapping from $\mathcal{Z}$ to features

(c) Mapping from $\mathcal{W}$ to features

StyleGAN의 생성자는 더욱 linear하며
덜 entangled하다

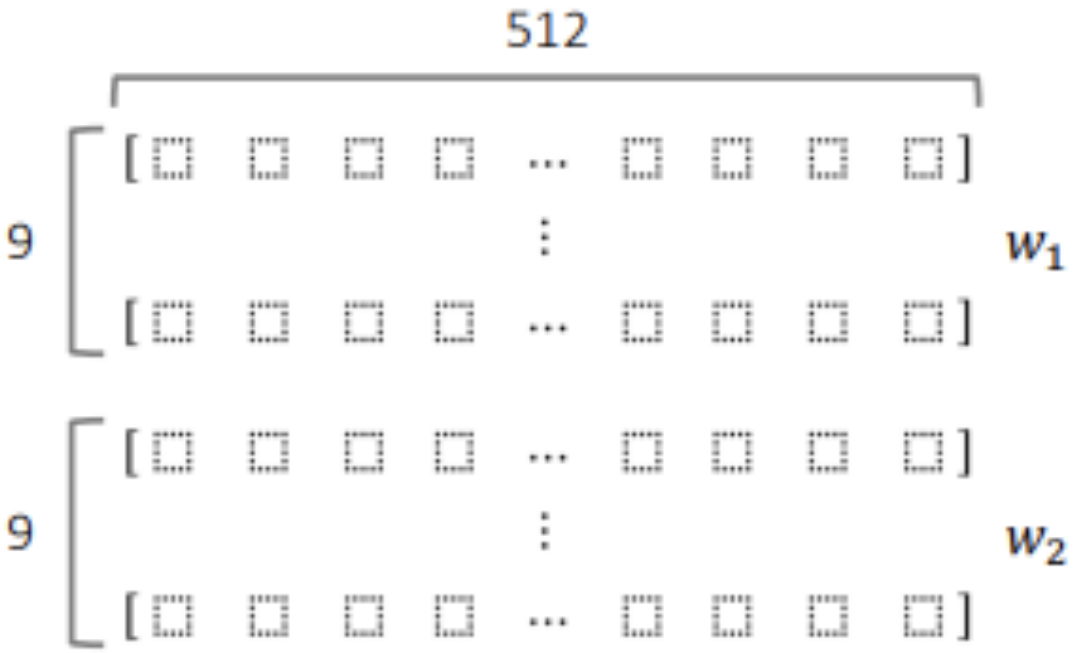
6. STYLE GAN

Latent Vector

6. STYLE GAN

Style Mixing

- 인접한 레이어 간의 스타일 상관관계를 줄입니다
- 방법
  - 두 개의 입력 벡터를 준비
  - 크로스오버 포인트를 이용
  - 크로스오버 이전에는 w_1, 이후에는 w_2를 이용
- 스타일은 각 레이어에 대해 localized된다



7. Summary

Gan

- 실제 데이터에 근사한 분포를 통해 이미지 생성
- Discriminator(판별자)와 Generator(생성자)로 학습
- 생성 이미지에 조건을 걸 수 있음
- 지금도 계속 새로운 Gan이 나오고 있다!

Reference

Gan

- https://www.youtube.com/@dongbinna
- https://haawron.tistory.com/21

DATA SCIENCE LAB

발표자 황진우 010-7938-5173
E-mail: @@@@gmail.com