[Karras19-CVPR] A Style-Based Generator Architecture for Generative Adversarial Networks

Jiin Kim

Abstract

1. 본 논문은 GAN 을 위한 새로운 **generator** 를 제안한다.
2. **Interpolation quality 와 disentanglement 를 정량화** 하기 위해 두 가지 새로운 기법을 제안한다.
3. 새로운 **데이터셋**을 만들었다.

1. Introduction

- GAN 의 한계점
  - Generator 가 **동작하는 원리**를 구체적으로 알기 어렵다.
  - **Latent space** 와 관련된 연구가 부족하다.
  - 일반적으로 입증된 latent space interpolation 방법들은 서로 다른 generator 들을 비교할 수 있는 **정량적인 방법**이 없다.

- StyleGAN 의 특징1
  - (1) 이미지 합성을 control (2) Generator 구조만 설계
    - 본 논문은 **style transfer** 논문[27]에서 영감을 받아 generator 가 이미지 합성 과정을 control 할 수 있도록 만들었다.
    - 본 논문은 **generator** 의 구조에만 신경 썼고, discriminator 와 loss function 은 건드리지 않았다.

[27] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. CoRR, abs/1703.06868, 2017

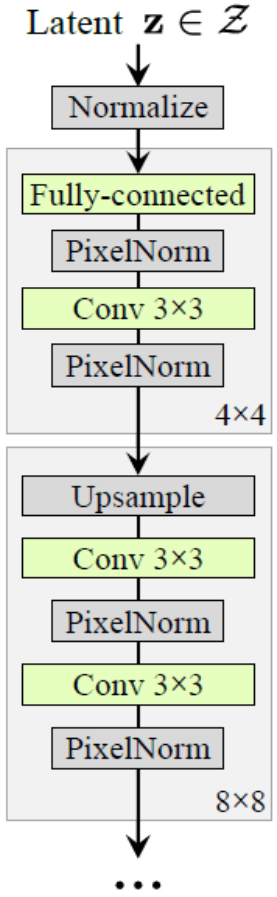
- StyleGAN 의 특징2
  - Disentanglement
    - 본 논문의 generator 는 중간 latent space 에 input latent code 를 삽입하는데, 이것은 variation 을 control 하는 것과 밀접한 관련이 있다.
    - Input latent space 는 training data 의 확률 밀도를 따라야 하고, 이것은 곧 entanglement 를 이끈다.
    - 그러나 본 논문은 input latent space 가 training data 의 확률 밀도를 따르도록 설계하지 않았다.
    - Disentanglement 에 대한 자세한 설명은 나중에 그림과 함께 하겠다.

- StyleGAN 의 특징3
  - (1) 새로운 Disentanglement 측정방법 제시 (2) Dataset 제작
    - latent space disentanglement 를 측정하는 새로운 두 가지 방법을 제시한다.
    - FFHQ dataset 을 제안한다.

2. Style-based generator

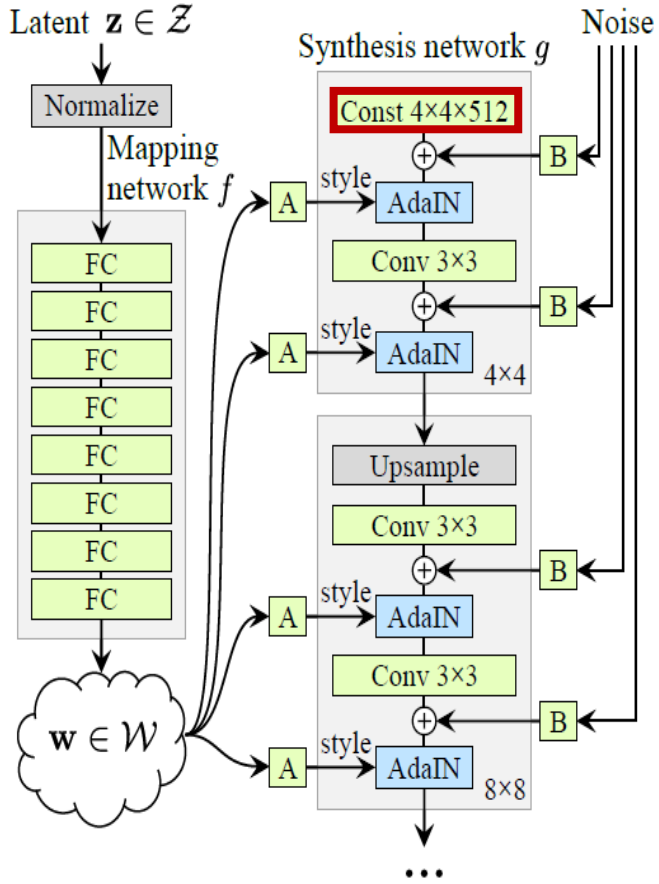
2. Style-based generator

Traditional Generator



Latent code z 는 오직 feed-forward network 의 첫 번째 레이어 만을 통과한다.

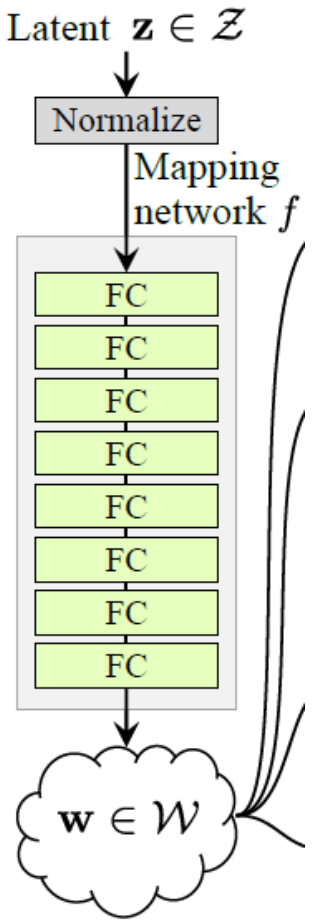
Style-based generator



학습된 constant 가 입력으로 들어간다.

Style-based generator

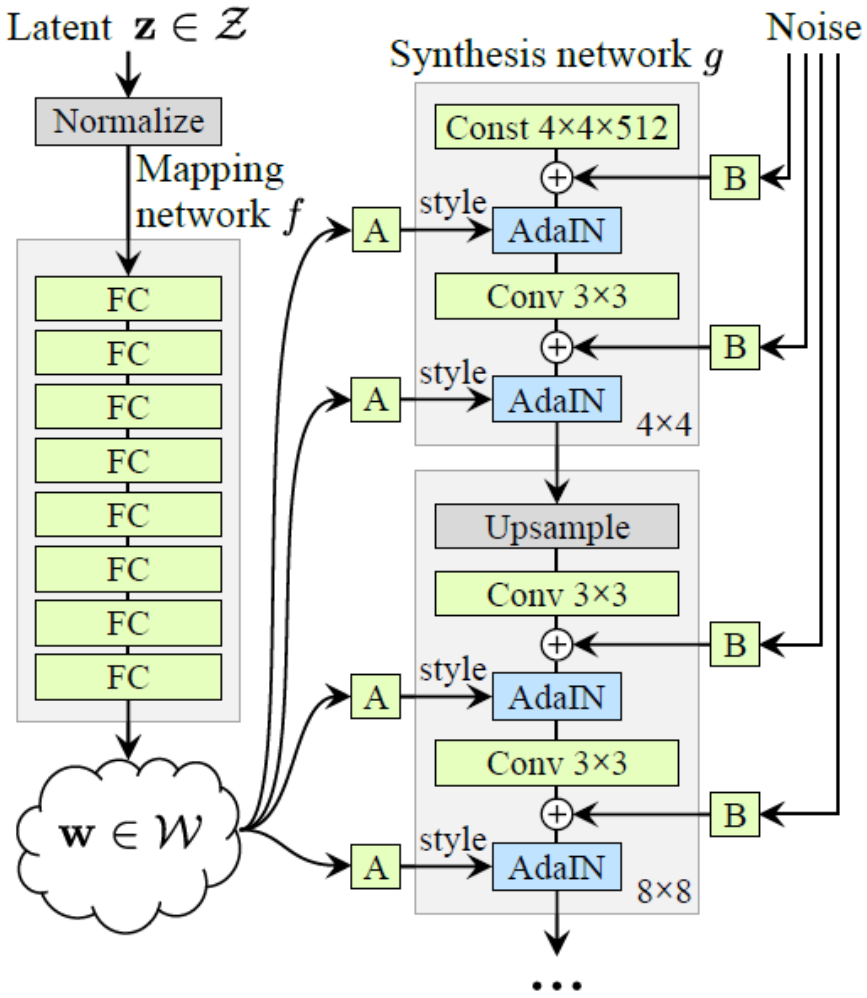
- Mapping network 인 $f : \mathcal{Z} \rightarrow \mathcal{W}$ 는 $z \in \mathcal{Z}$ 로 $w \in \mathcal{W}$ 를 만든다.
- $\mathcal{Z}$ 와 $\mathcal{W}$ 는 모두 512차원이고, f 는 8-layer MLP 로 이루어진다.



Style-based generator

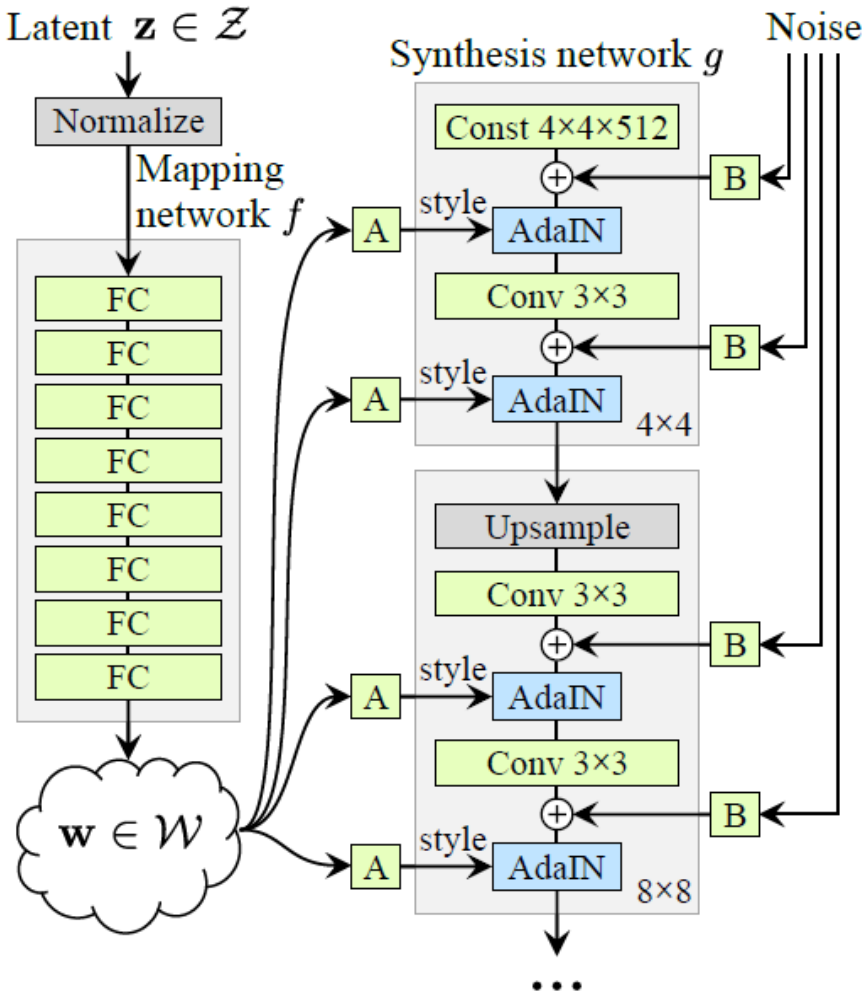
- w 는 학습된 affine transformation(A) 을 통해서 '*style*' 이라고 불리는 $y = (y_s, y_b)$ 를 생성하고, 생성된 y 는 AdaIN 을 control 한다. (i 는 channel)

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$



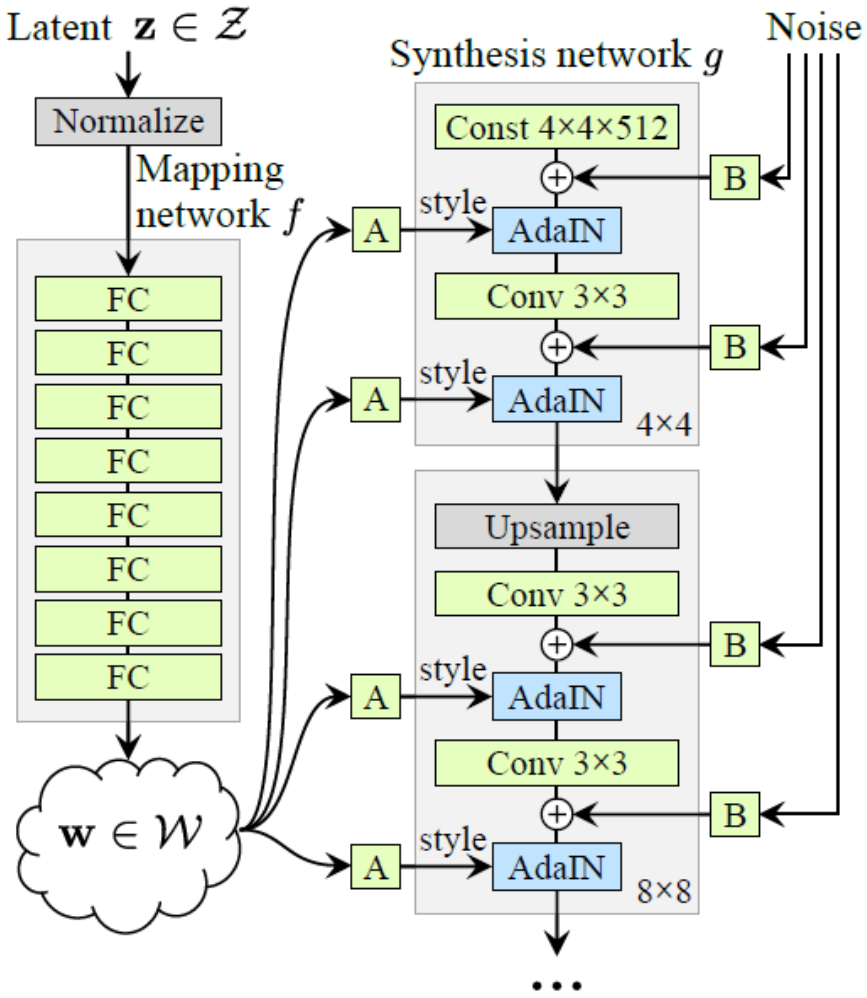
Style-based generator

- 확률적인 변이(주근깨, 주름, 머리카락 등)를 생성하기 위해 noise 를 도입했다.
- 노이즈는 상관관계가 없는 가우시안 노이즈로 구성된 1개의 channel 을 가진 이미지이다.
- 학습된 per-channel scaling factors 를 노이즈 이미지에 적용(B)한 결과는 합성곱의 출력과 더해진다.



Style-based generator

- g 는 18개의 레이어로 이루어진다. ($4^2 - 1024^2$ → 각 resolution 당 2개의 convolution = 9 x 2 = 18)
- 마지막 레이어의 출력은 분리된 1×1 합성곱을 사용하여 RGB 로 변환된다.



2.1. Quality of generated images

2.1. Quality of generated images

- 다양한 generator 들을 사용해서 FID 를 계산하였다.
- Truncation trick 을 사용하지 않았다.

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [30]	7.79	8.04
B	+ Tuning (incl. bilinear up/down)	6.11	5.25
C	+ Add mapping and styles	5.34	4.85
D	+ Remove traditional input	5.07	4.88
E	+ Add noise inputs	**5.06**	4.42
F	+ Mixing regularization	5.17	**4.40**

(A) PGGAN

(B) Bilinear up/downsampling operations

(C) Mapping Network + AdaIN

(D) 4 x 4 x 512 상수 텐서를 입력으로 사용

(E) 노이즈 입력 사용

(F) Mixing Regularization

2.1. Quality of generated images

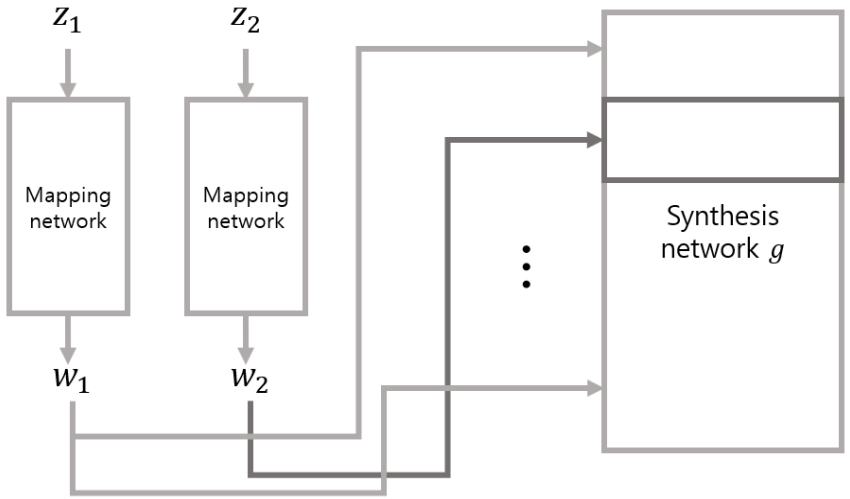
- 본 논문에서 제안한 style-based generator 를 사용해서 생성한 이미지들이다.
- Truncation trick 를 사용했다.

3. Properties of the style-based generator

3.1. Style mixing

3.1. Style mixing

- 인접한 style 사이의 연관성을 없애기 위해 Mixing regularization 을 사용했다.
- Training 동안에 “머리가 검은색인 사람은 항상 안경을 꼈다“와 같은 작은 artifact 가 만들어졌다고 하자.
  - 하나의 z 에서 나온 하나의 w 로만 계속 학습을 하면 안경을 낀 사람은 머리색을 전부 검은색으로 만들어버린다. 검은색 머리와 안경 사이의 correlation 이 생긴 것이다.
  - 작은 스케일의 style 과 큰 스케일의 style 이 w 단에서 correlation 이 생기는 것을 방지하기 위해 style mixing 연산을 사용해 머리가 노란색인 사람도 안경을 낄 수 있게 만들었다.



Style mixing

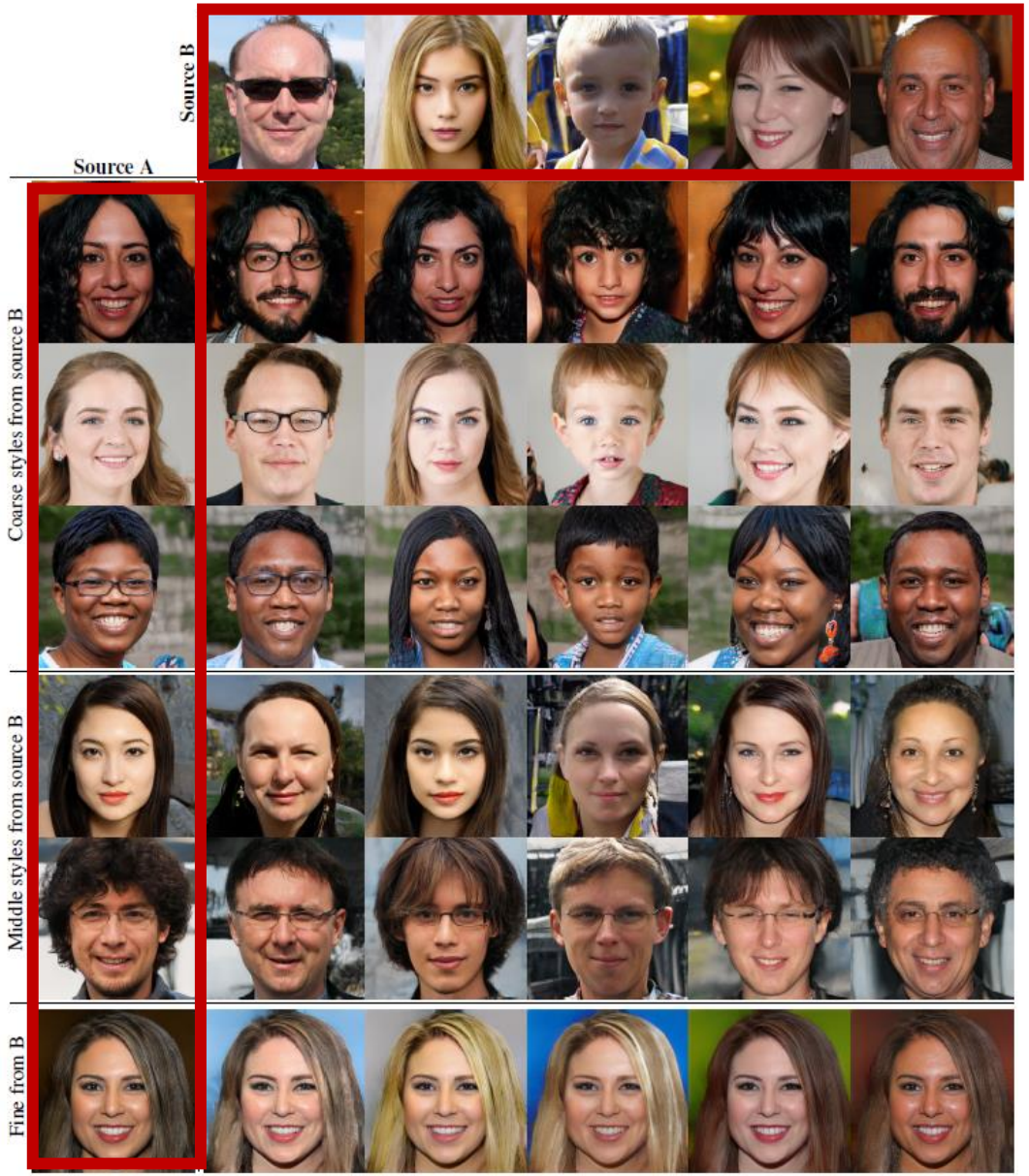
3.1. Style mixing

	Mixing regularization	Number of latents during testing 1	2	3	4
E	0%	4.42	8.22	12.88	17.41
	50%	4.41	6.10	8.71	11.61
F	90%	**4.40**	**5.11**	6.88	9.03
	100%	4.83	5.17	**6.63**	**8.40**

- Mixing regularization 을 많이 사용할 수록 FID 값이 작아진다.

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [30]	7.79	8.04
B	+ Tuning (incl. bilinear up/down)	6.11	5.25
C	+ Add mapping and styles	5.34	4.85
D	+ Remove traditional input	5.07	4.88
E	+ Add noise inputs	**5.06**	4.42
F	+ Mixing regularization	5.17	**4.40**

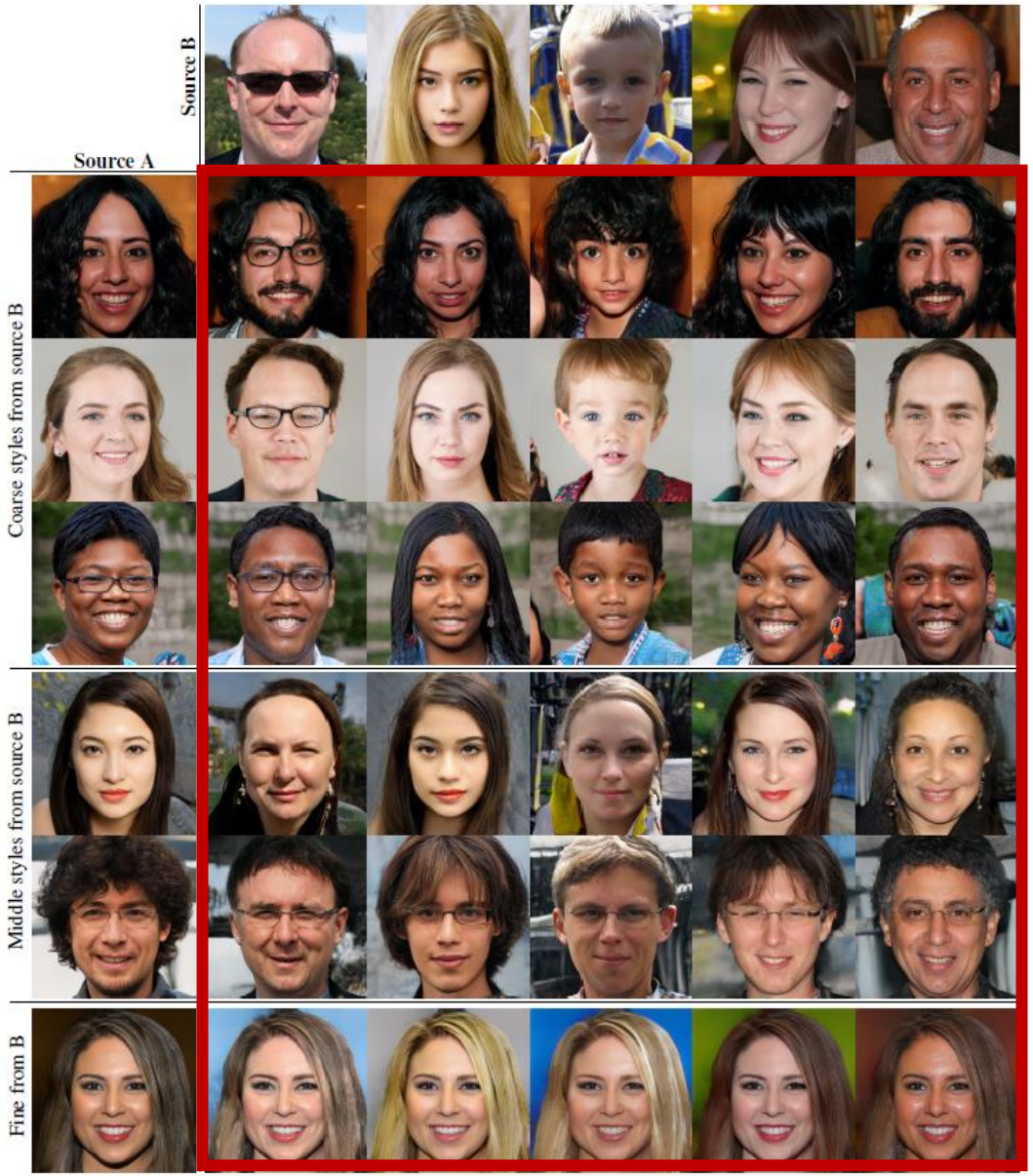
3.1. Style mixing



각 latent code 를 사용해서 두 개의 이미지 집합을 만들었다.

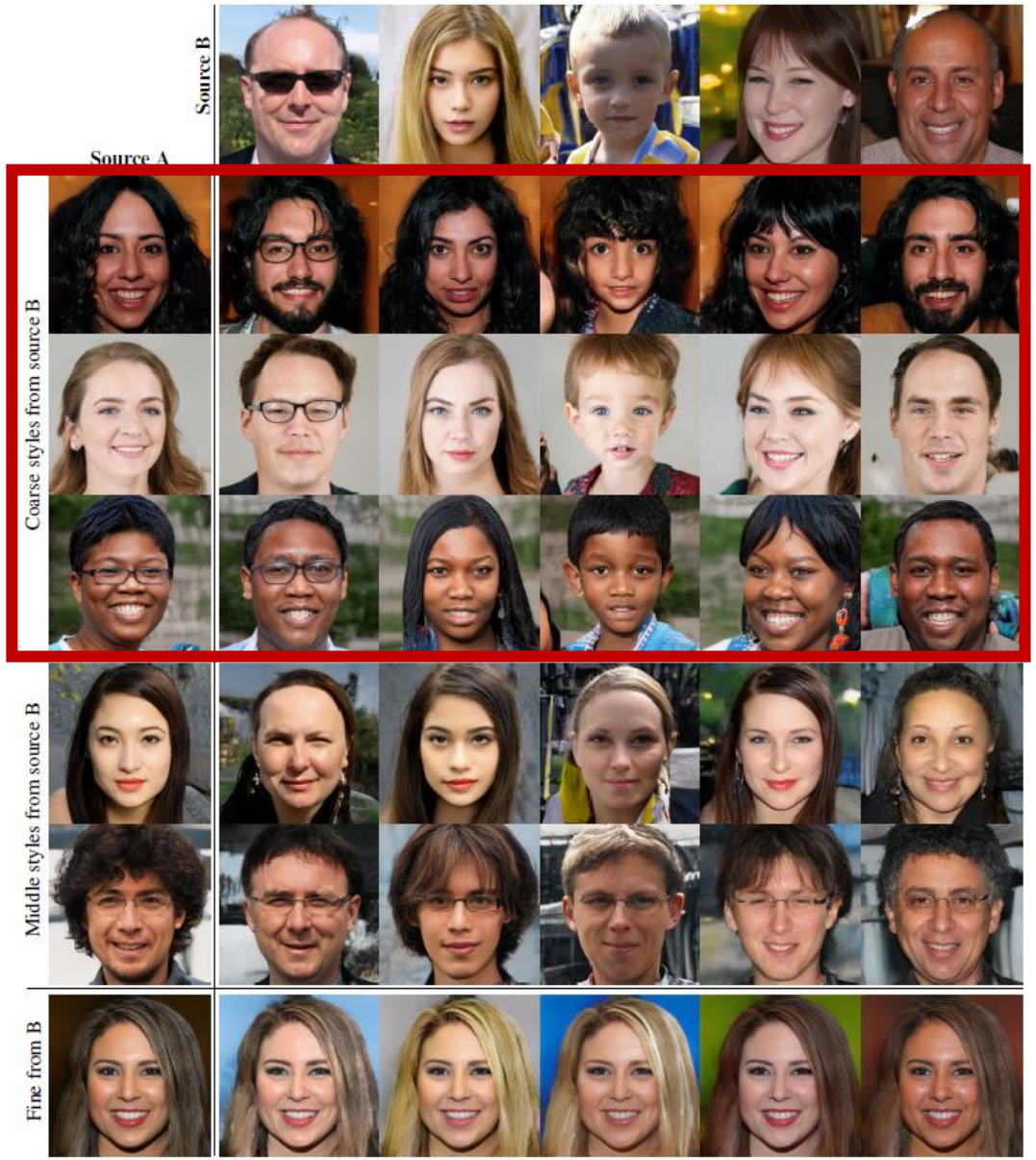
Source A & Source B

3.1. Style mixing



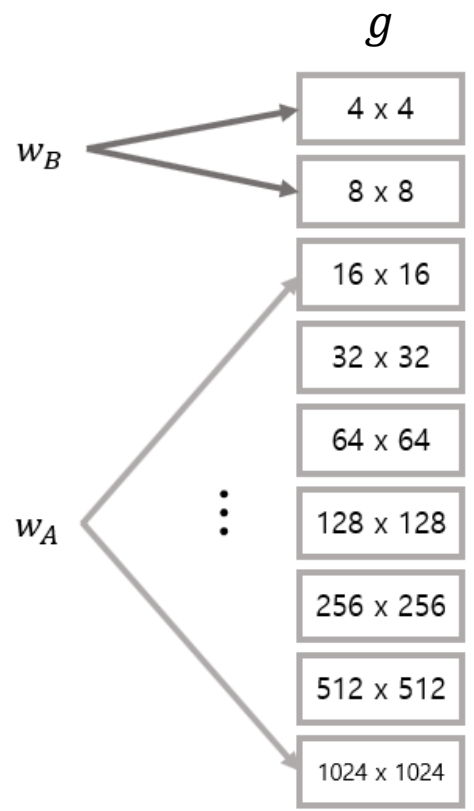
Source B 의 특정 style 을 복사하고 나머지 부분에는 source A 의 style 을 복사해서 이미지들을 생성했다.

3.1. Style mixing

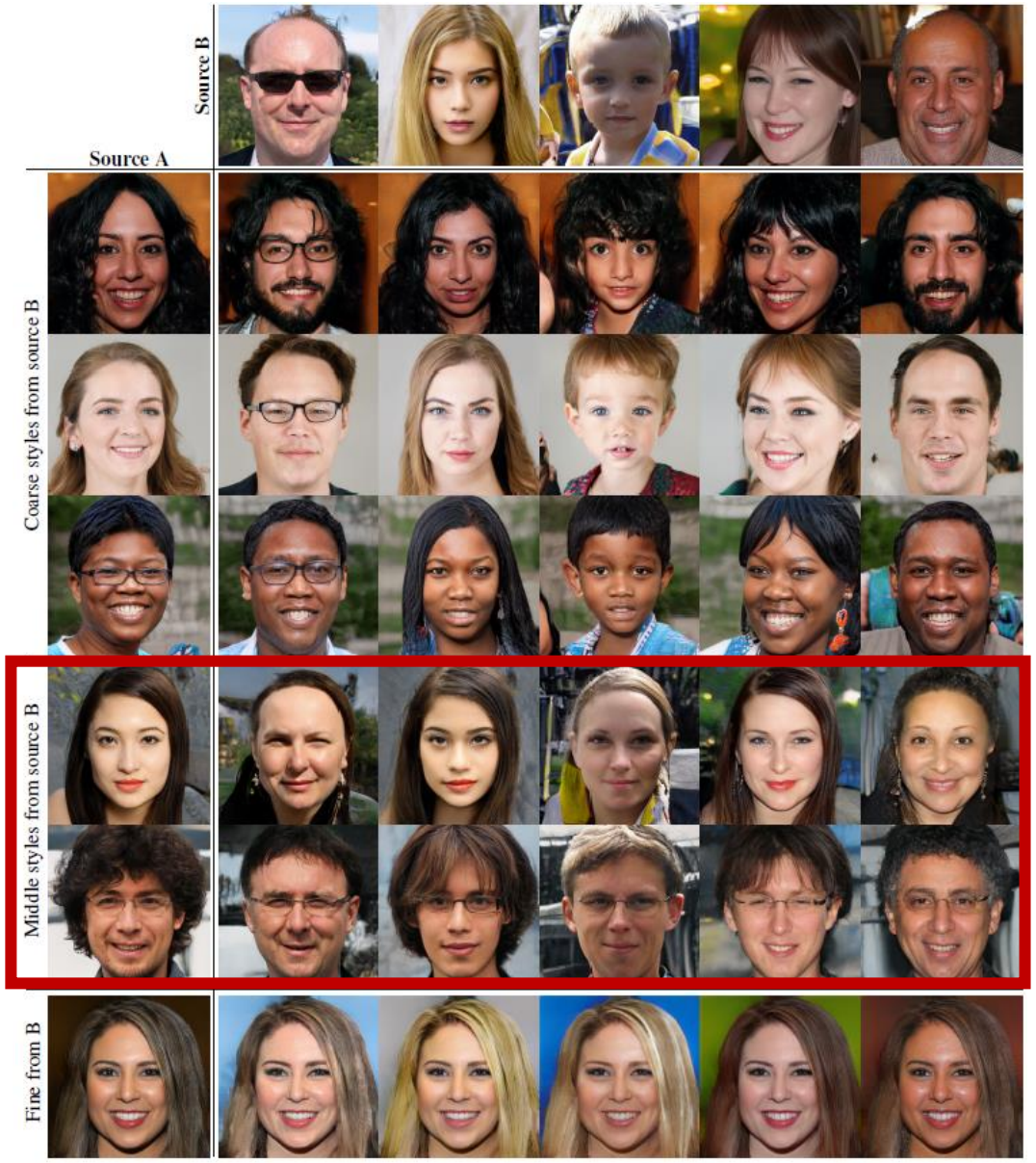


Coarse spatial resolutions $(4^2 - 8^2)$ 의 styles 을 B 에서 복사

- High-level aspects (pose, general hair style, face shape, and eyeglasses) 을 B 에서 복사했다.
- Finer facial features (eyes, hair, lighting 의 color) 는 A 에서 복사했다.

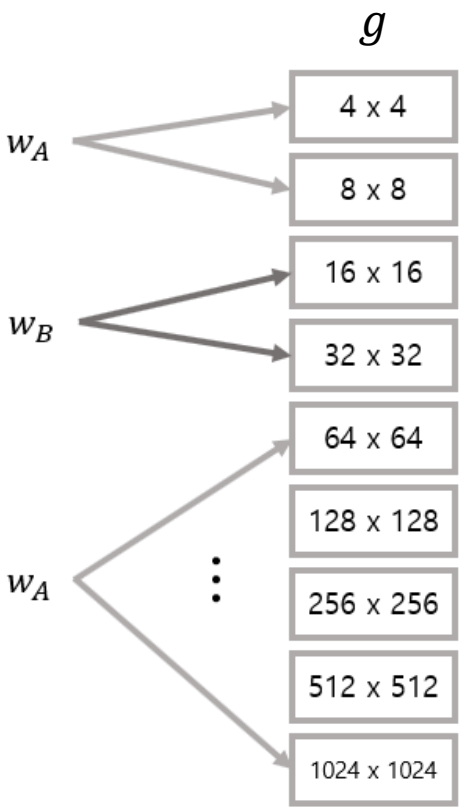


3.1. Style mixing

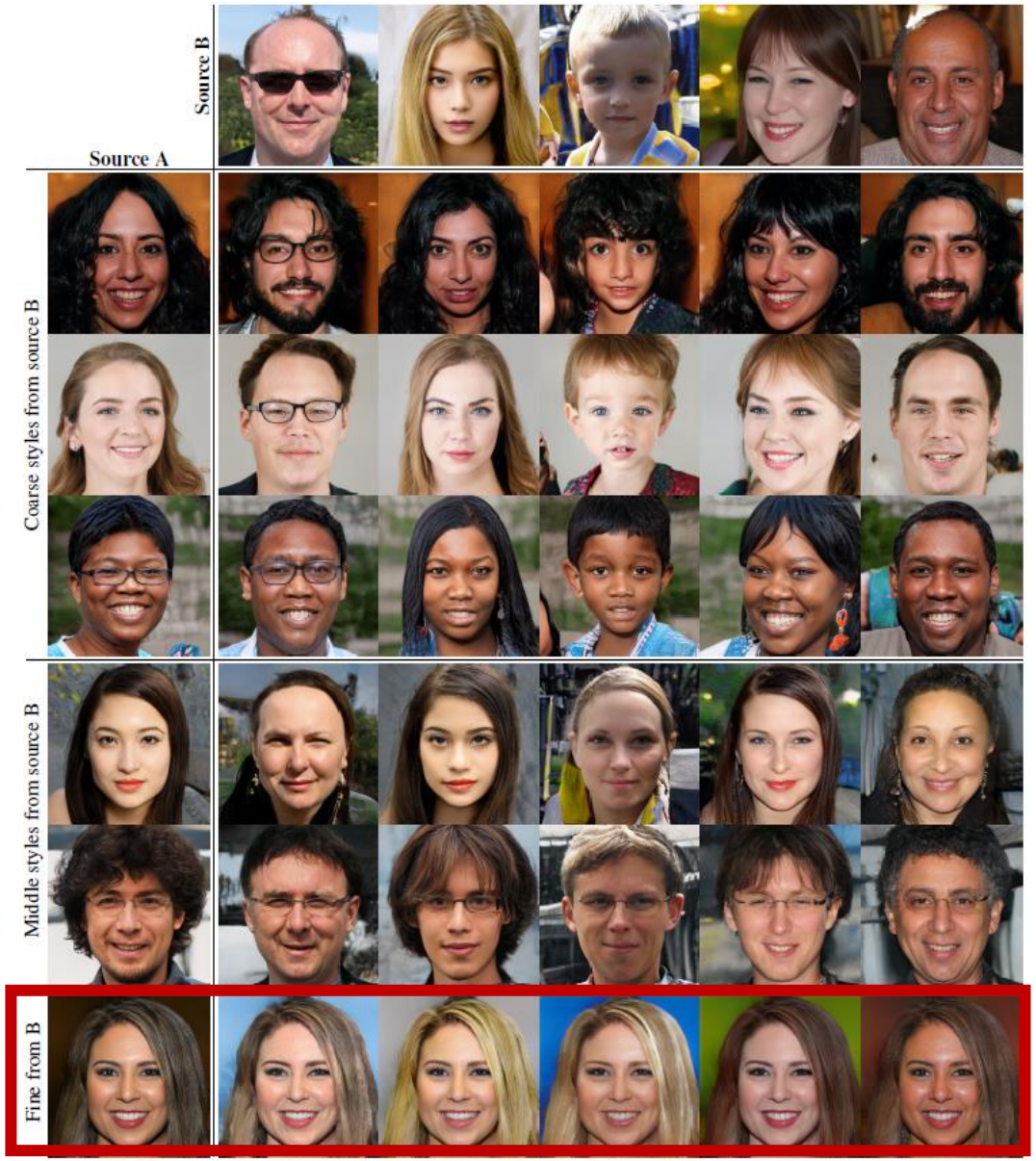


Middle resolutions $(16^2 - 32^2)$ 의 styles 를 B 에서 복사

- Smaller scale facial features, hair style, eyes open/closed 를 B 에서 복사했다.
- Pose, general face shape, and eyeglasses 를 A 에서 복사했다.

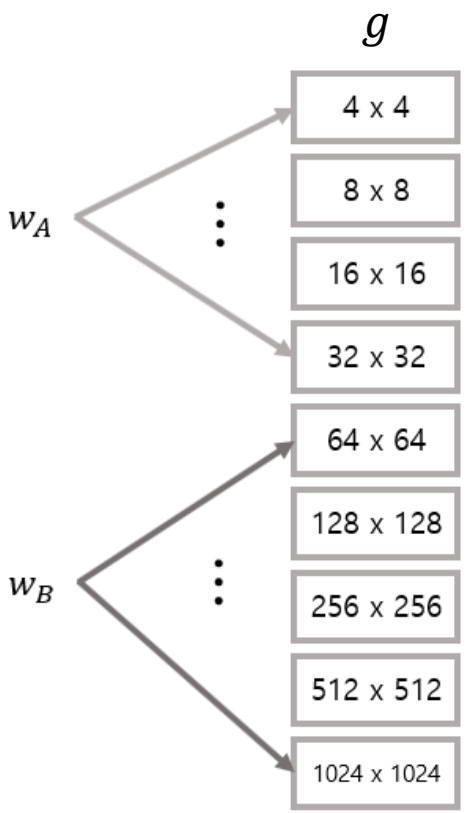


3.1. Style mixing



Fine styles ($64^2 - 1024^2$) 를 B 에서 복사

- 주로 color scheme 과 microstructure 를 B 에서 복사했다.



3.2. Stochastic variation

3.2. Stochastic variation

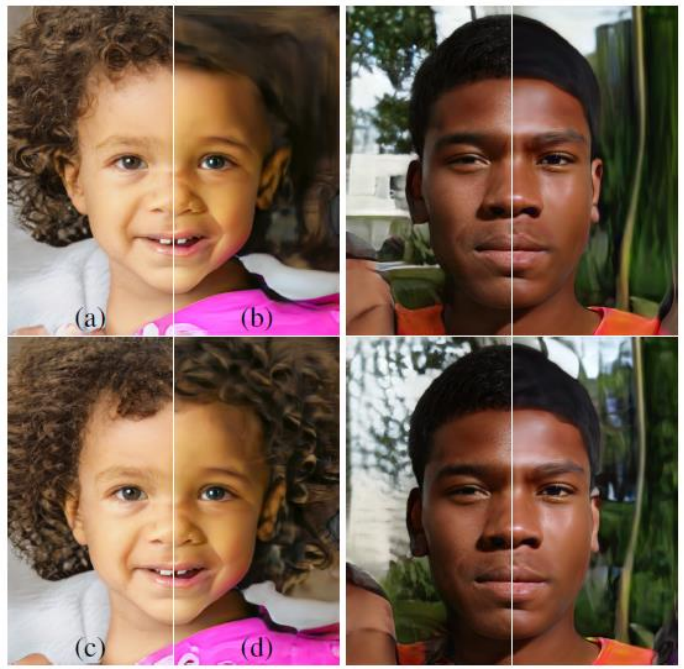
- 이미지를 건드리지 않고 stochastic variation 을 랜덤으로 이미지에 추가할 수 있다.

3.2. Stochastic variation

(a) Generated image (b) Stochastic variation (c) Standard deviation

- Stochastic variation 의 예시를 나타낸다.
  - (b) 다양한 입력 노이즈를 적용한 결과이다. 전체적인 외모는 거의 동일하지만, 머리카락이 다르다.
  - (c) 100 개의 다른 realizations 을 통해 각 픽셀의 표준편차를 구하고, 노이즈에 영향을 받는 부분을 강조했다. 머리카락, 실루엣, 배경, 눈 반사 등이 영향을 받는다. Identity 나 pose 같은 전체적인 측면은 stochastic variation 에 영향을 받지 않는다.

3.2. Stochastic variation



- 본 논문의 generator 의 여러 레이어에 noise 를 넣었을 때의 효과를 나타낸다.
  - (a) Noise 가 모든 레이어에 적용되었다.
  - (b) Noise 가 사용되지 않았다.
  - (c) Noise 가 fine layer 인 $64^2 - 1024^2$ 레이어에서 사용되었다.
  - (d) Noise 가 course layer 인 $4^2 - 32^2$ 레이어에서 사용되었다.
  - Coarse noise 는 머리카락에서 굵은 곱슬을 만들고 배경이 더 크게 나타난다.
  - 반면 fine noise 는 머리카락에 얇은 곱슬을 만들고, 더 자세한 배경을 만들고, 모공도 만든다.

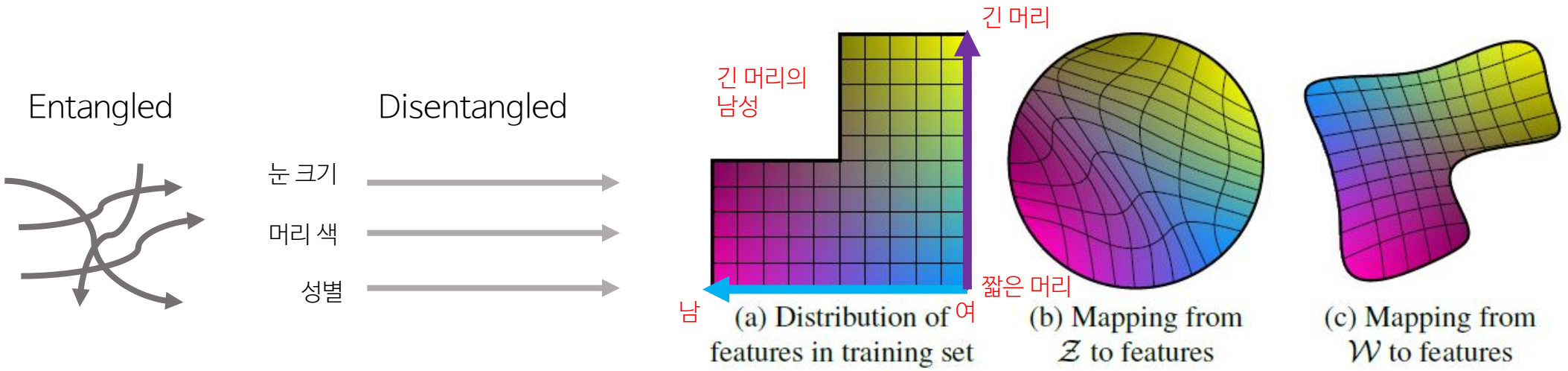
3.3. Separation of global effects from stochasticity

3.3. Separation of global effects from stochasticity

- Style 을 바꾸는 것
  - 이미지의 전체적인 부분에 영향을 미친다.
- Noise 를 추가하는 것
  - 중요하지 않은 stochastic variation에 영향을 미친다.

4. Disentanglement studies

4. Disentanglement studies



- Disentanglement 는 선형적으로 잘 분리되어 있는 것을 말한다.
- 예 : 성별 & 머리카락 길이
  - (a) “긴 머리의 남성”이 training set 에 없다.
  - (b) Training set 의 분포를 그대로 고정된 분포인 $\mathcal{Z}$ 에 매핑했다. 억지로 training set 의 분포를 $\mathcal{Z}$ 에 매핑하다 보니 $\mathcal{Z}$ 와 image feature 의 매핑이 곡선이 되었다. Latent space 의 차원을 조금씩 움직였을 때, 짧은 머리 다음에 바로 긴 머리가 나오는 등 매우 예측하기 힘든 일이 일어난다.
  - (c) 중간 latent space $\mathcal{W}$ 는 고정된 분포에 따라 샘플링 할 필요가 없다. $\mathcal{W}$ 는 학습된 매핑 $f(z)$ 에 의해 유도되고, 이 맵핑은 latent space 의 subspace 를 선형적을 만든다.

4. Disentanglement studies

- Disentanglement 를 정량화 하기 위해 최근에 제안된 방법들은 input images 를 latent codes 로 매핑하는 encoder network 를 요구한다. 이 방법들은 본 논문의 baseline GAN 이 그러한 encoder 가 부족하기 때문에 본 논문의 목적에는 부적합하다.
- 본 논문은 **두 개의 새로운 disentanglement 를 정량화 하는 방법**을 제안한다.

4.1. Perceptual path length

4.1. Perceptual path length

- Latent-space vectors 를 interpolation 을 하면, 이미지에서 non-linear 한 변화를 만든다.

남

아기

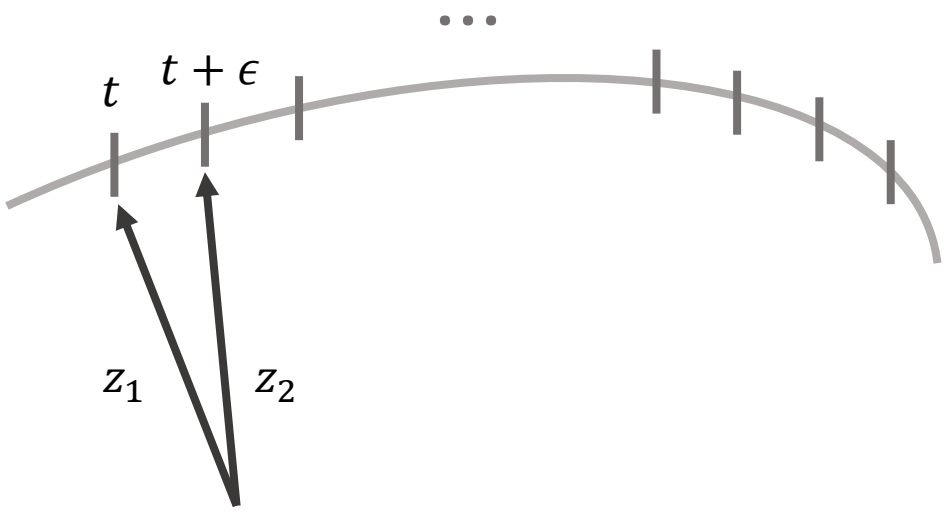
여

Interpolation

- 이것은 latent space 가 entangled 되어 있고 variation 의 요인이 적절히 분리되어 있지 않다는 것을 의미한다.
- Disentanglement 를 정량화 하기 위해, latent space 에서 **interpolation** 을 수행할 때 이미지가 얼마나 변하는지 측정할 것이다.

4.1. Perceptual path length

- Perceptually-based pairwise image distance
  - 두 개의 VGG16 embeddings 사이에서 weighted difference 를 계산한다.
  - 먼저, Latent space interpolation path 를 linear segments 로 나눈다.
  - Total perceptual length 는 각 segment 에 대한 perceptual differences 의 합으로 정의한다.

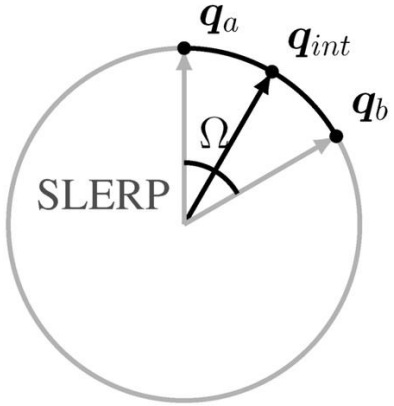


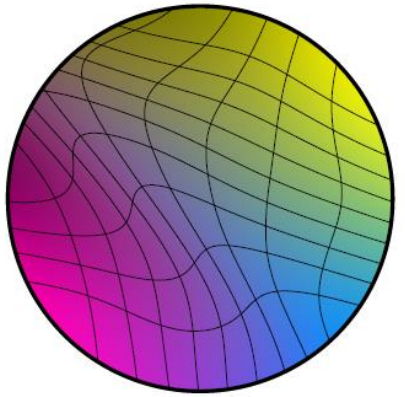
4.1. Perceptual path length

- 모든 가능한 endpoints 에 대해 latent space Z 에서 perceptual path length 의 평균은 다음과 같다.

$$l_Z = \mathbb{E}\left[\frac{1}{\epsilon^2} d\left(G\left(slerp(z_1, z_2; t)\right), G\left(slerp(z_1, z_2; t+\epsilon)\right)\right)\right]$$

- $\epsilon = 10^{-4}$
- $z_1, z_2 \sim P(z)$
- $t \sim U(0,1)$
- G 는 generator
- $d(\cdot,\cdot)$ 는 perceptual distance 계산



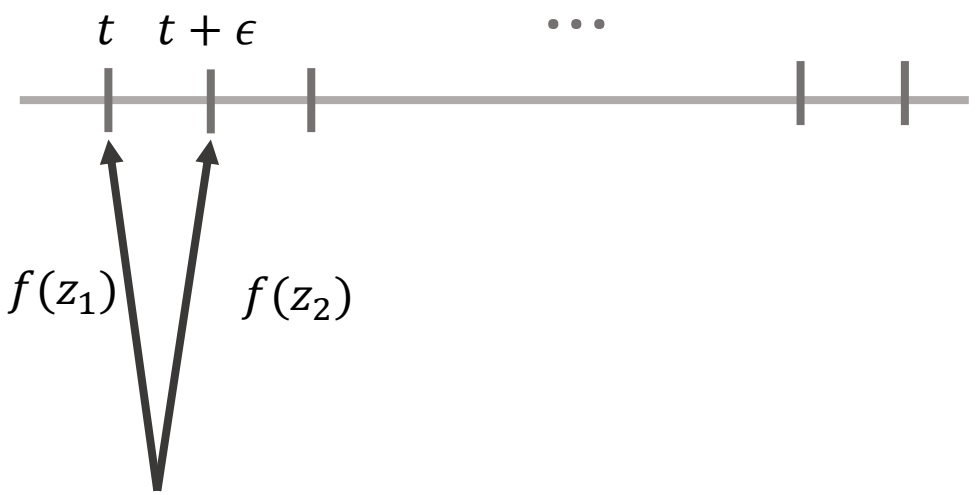


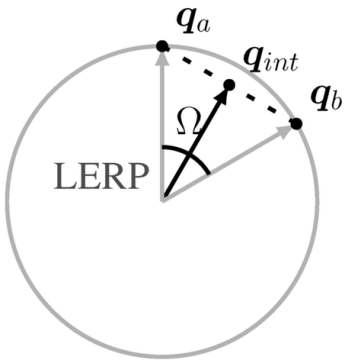
- $slerp$ 는 spherical interpolation 이다. 이것은 normalized input latent space 에서 가장 적절한 interpolation 방법이다.
- Pairwise image metric 을 계산하기 전에 생성된 이미지가 오직 face 만을 가지도록 crop 했다.
- d 가 quadratic [65] (모든 식이 2차 항)하기 때문에, ϵ^2 로 나누었다.
- 100,000 개의 샘플들을 가지고 기댓값을 계산했다.

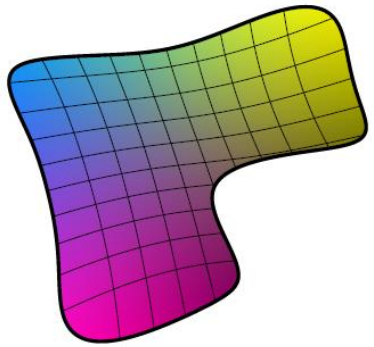
4.1. Perceptual path length

- W 에서의 average perceptual path length 도 역시 비슷한 방식으로 계산한다.

$$l_W = \mathbb{E}\left[\frac{1}{\epsilon^2} d(g(lerp(f(z_1), f(z_2); t)), g(lerp(f(z_1), f(z_2); t + \epsilon)))\right]$$







4.1. Perceptual path length

	Method	Path length full	Path length end	Separability
B	Traditional generator $\mathcal{Z}$	412.0	415.3	10.78
D	Style-based generator $\mathcal{W}$	446.2	376.6	3.61
E	+ Add noise inputs $\mathcal{W}$	**200.5**	**160.6**	3.54
	+ Mixing 50% $\mathcal{W}$	231.5	182.1	**3.51**
F	+ Mixing 90% $\mathcal{W}$	234.0	195.9	3.79

	Method	FID	Path length full	Path length end	Separability
B	Traditional 0 $\mathcal{Z}$	5.25	412.0	415.3	10.78
	Traditional 8 $\mathcal{Z}$	4.87	896.2	902.0	170.29
	Traditional 8 $\mathcal{W}$	4.87	324.5	212.2	6.52
	Style-based 0 $\mathcal{Z}$	5.06	283.5	285.5	9.88
	Style-based 1 $\mathcal{W}$	4.60	219.9	209.4	6.81
	Style-based 2 $\mathcal{W}$	4.43	**217.8**	199.9	6.25
F	Style-based 8 $\mathcal{W}$	**4.40**	234.0	**195.9**	**3.79**

- Method name 의 숫자는 mapping network 의 **depth** 를 나타낸다.

4.2. Linear separability

4.2. Linear separability

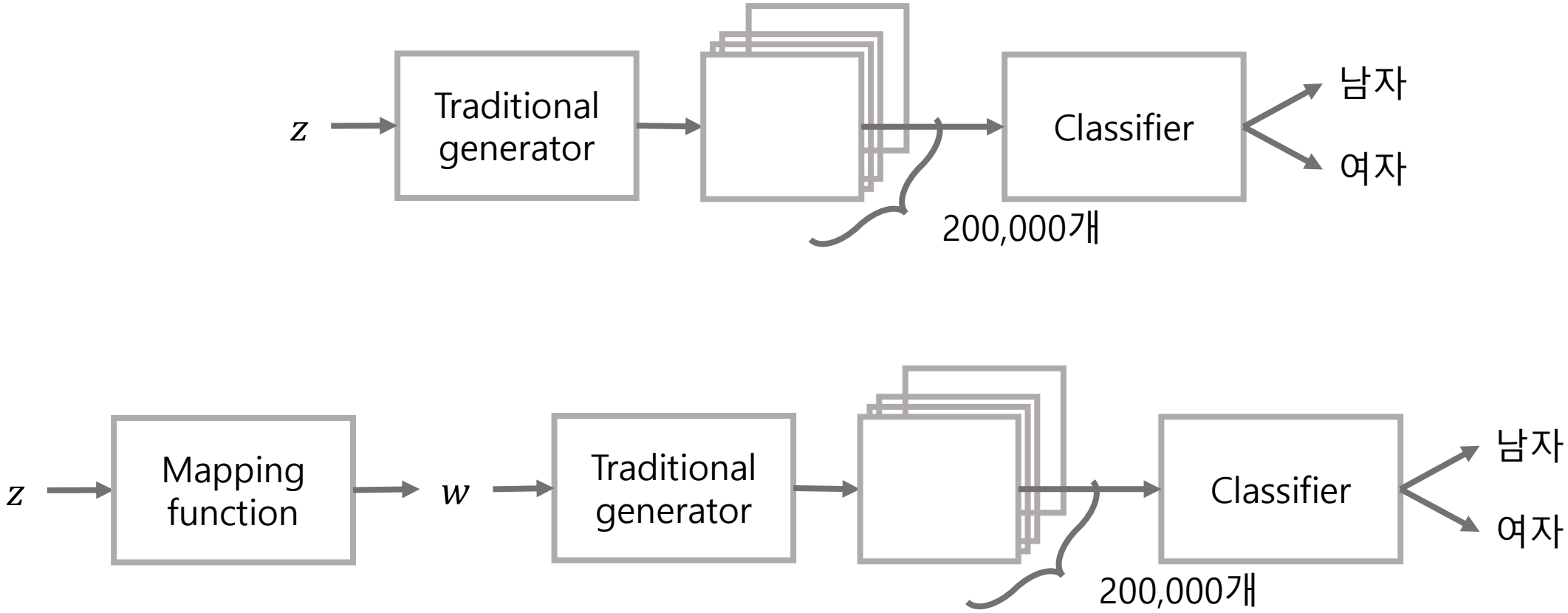
- Latent space 가 충분히 disentangled 되었다면, 각 factors of variation 당 상응하는 방향 벡터가 있어야 한다.
- 본 논문은 latent-space points 가 linear hyperplane 을 통해서 얼마나 잘 두 개의 구별된 sets 로 분리될 수 있는지 측정함으로써 이 효과를 정량화 하는 또다른 기법을 제안한다.

4.2. Linear separability

- 이미지를 생성하고, 생성된 이미지들에 label 을 붙이기 위해 이진 분류를 수행하는 하나의 **classification network** 를 train 했다.
- 이 classifiers 는 우리의 discriminator 와 같은 구조를 가졌고, **40 개의 attributes 를 가진 CELEBA-HQ 데이터셋**을 사용해서 train 했다.

4.2. Linear separability

- 하나의 attribute 에 대해 separability 를 측정하기 위해, $z \sim P(z)$ 로 200,000 개의 이미지를 생성하고 classification network 를 사용해서 분류했다.



4.2. Linear separability

- Classifier confidence 에 따라 샘플들을 정렬하고 신뢰도가 가장 낮은 샘플 수의 ½을 제거했다.
- 그래서 100,000개의 labeled latent space vectors 를 만들었다.

Traditional generator

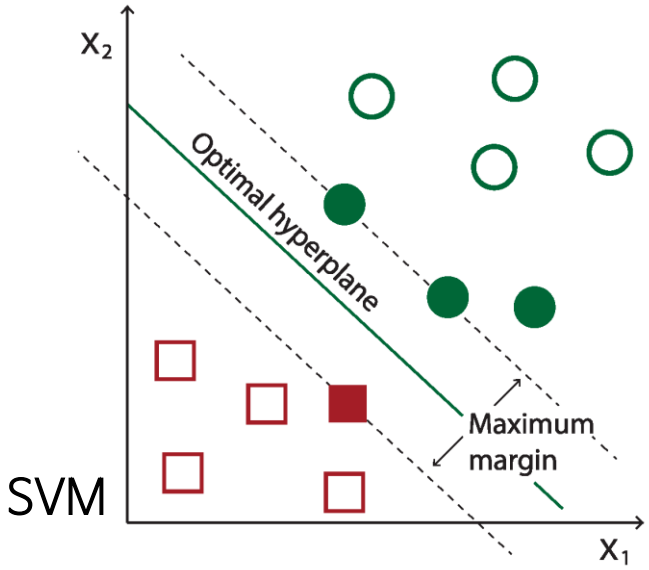
z_1	z_2	z_3	…	$z_{100,000}$
남	여	여	…	남

Style-based generator

w_1	w_2	w_3	…	$w_{100,000}$
남	여	여	…	남

4.2. Linear separability

- 각 attribute 에 대해서, latent-space point 에 기반한 label 을 예측하기 위해 linear SVM 을 적용했다.
- 그리고 조건부 엔트로피 $H(Y|X)$ 를 계산한다. (Z와 W에 대해 계산했다.)
  - X 는 SVM 에 의해 예측된 classes 이다.
  - Y 는 pre-trained classifier 에 의해 결정된 classes 이다.
  - 조건부 엔트로피는 어떤 확률 변수 X가 다른 확률변수 Y의 값을 예측하는데 도움이 되는지를 측정하는 방법 중의 하나이다.
- $\exp(\sum_i H(Y_i|X_i))$ 로 final separability score 를 계산하고, i 는 40 attributes 를 순회한다.



4.2. Linear separability

	Method	Path length full	Path length end	Separability
B	Traditional generator $\mathcal{Z}$	412.0	415.3	10.78
D	Style-based generator $\mathcal{W}$	446.2	376.6	3.61
E	+ Add noise inputs $\mathcal{W}$	**200.5**	**160.6**	3.54
	+ Mixing 50% $\mathcal{W}$	231.5	182.1	**3.51**
F	+ Mixing 90% $\mathcal{W}$	234.0	195.9	3.79

	Method	FID	Path length full	Path length end	Separability
B	Traditional 0 $\mathcal{Z}$	5.25	412.0	415.3	10.78
	Traditional 8 $\mathcal{Z}$	4.87	896.2	902.0	170.29
	Traditional 8 $\mathcal{W}$	4.87	324.5	212.2	6.52
	Style-based 0 $\mathcal{Z}$	5.06	283.5	285.5	9.88
	Style-based 1 $\mathcal{W}$	4.60	219.9	209.4	6.81
	Style-based 2 $\mathcal{W}$	4.43	**217.8**	199.9	6.25
F	Style-based 8 $\mathcal{W}$	**4.40**	234.0	**195.9**	**3.79**

- W 가 Z 보다 훨씬 특징 분류를 잘 하는 것을 보여준다. (Separability가 낮다.)
- 더욱이, Mapping network 의 depth 를 증가시키는 것은 image quality 와 W 에서의 separability 를 향상시킨다.

Thank you for listening