Style GAN :

A Style-Based Generator Architecture for Generative Adversarial Networks

DA 문구영

Contents

ProGAN (PGGAN)

Training time: 13 days
1024x1024 resolution

Generator
Discriminator
z = random code
x = real image
x' = generated image

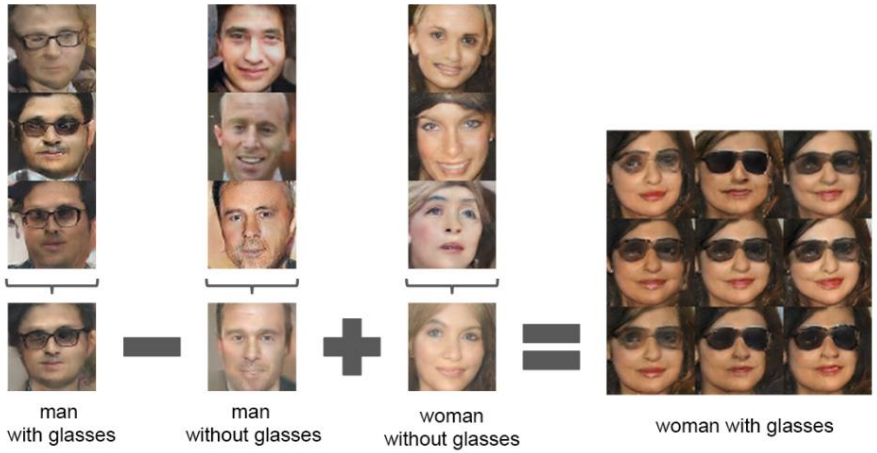
https://towardsdatascience.com/progan-how-nvidia-generated-images-of-unprecedented-quality-51c98ec2cbd2

메인 아이디어

- 학습 초기부터 고해상도 층을 사용하면 손실 함수의 공간을 탐색하기 어려움 (생성자에 유의미한 gradient를 제공하지 못함)
- 학습 과정에서 **점진적으로 네트워크의 layer를 붙여나감 (progressive growing)**
- 생성 이미지의 해상도를 점진적으로 증대 (점진적으로 부드럽게 천천히 복잡한 모습이 확대되어 나타나도록)

한계

- 고해상도 이미지의 학습에 성공했으나 **이미지의 특징을 제어하기 어려움**



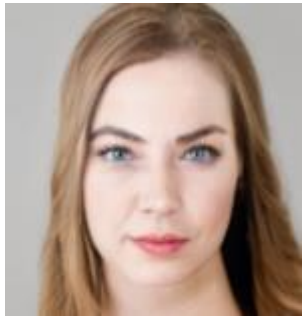
Abstract – Style GAN

Style GAN

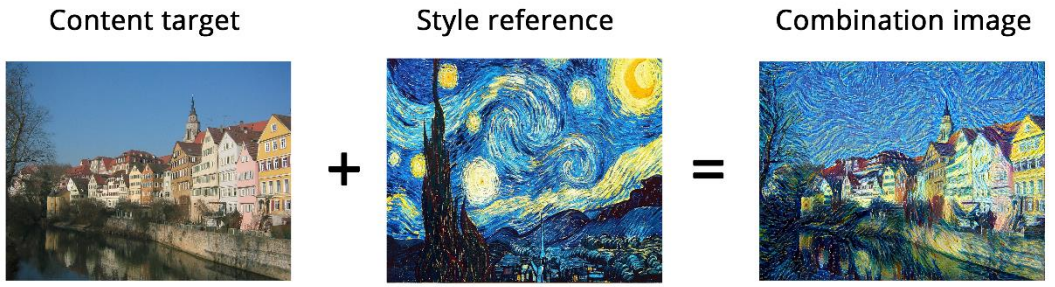
1. Style transfer literature

Original

Source B



Coarse Styles
from Source B



→ 원본 이미지의 high level attributes는 유지한 채 style만 변경

2. 이미지 생성 과정의 제어가 가능하며 고화질 이미지 생성에 적합한 GAN 아키텍처

→ 고해상도의 이미지를 생성하나 생성된 이미지의 구체적인 특징을 컨트롤하기 힘든 Pro GAN 아키텍처를 개선

→ Style의 disentanglement 성능을 향상

→ 다양한 유형을 가진 고해상도 얼굴 데이터셋을 제안

3. Generator 아키텍처를 개선

→ 생성자의 초기 입력을 latent code가 아닌 **learned constant**로 변경

→ 생성자의 매 Convolutional layer에서 이미지의 style을 조정

4. Mapping Network

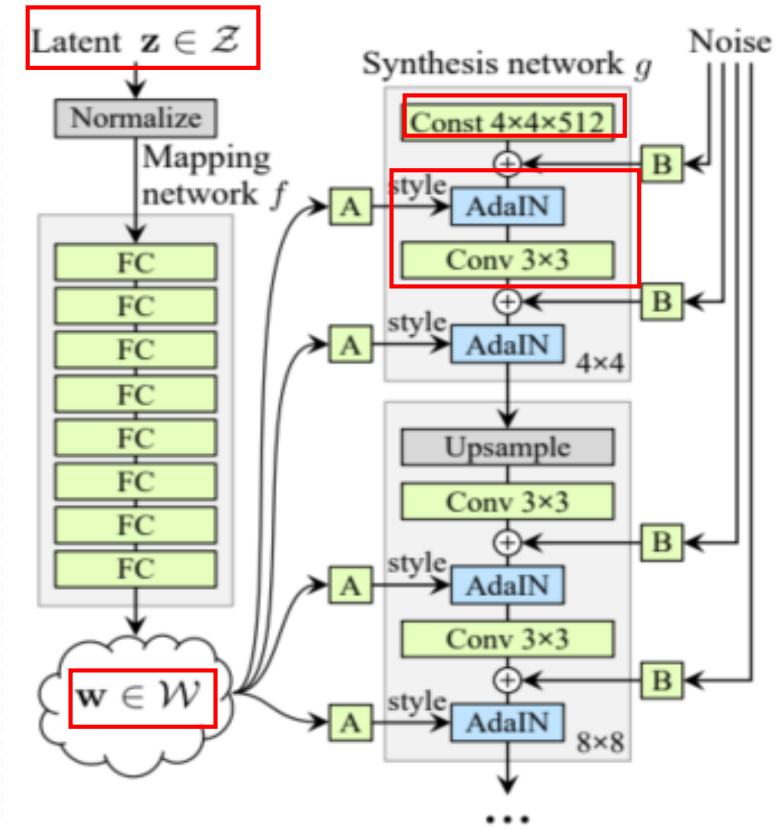
→ Latent code z를 intermediate latent space w로 mapping

→ z는 특정 확률 분포를 따르기 때문에 style의 **entanglement**를 야기

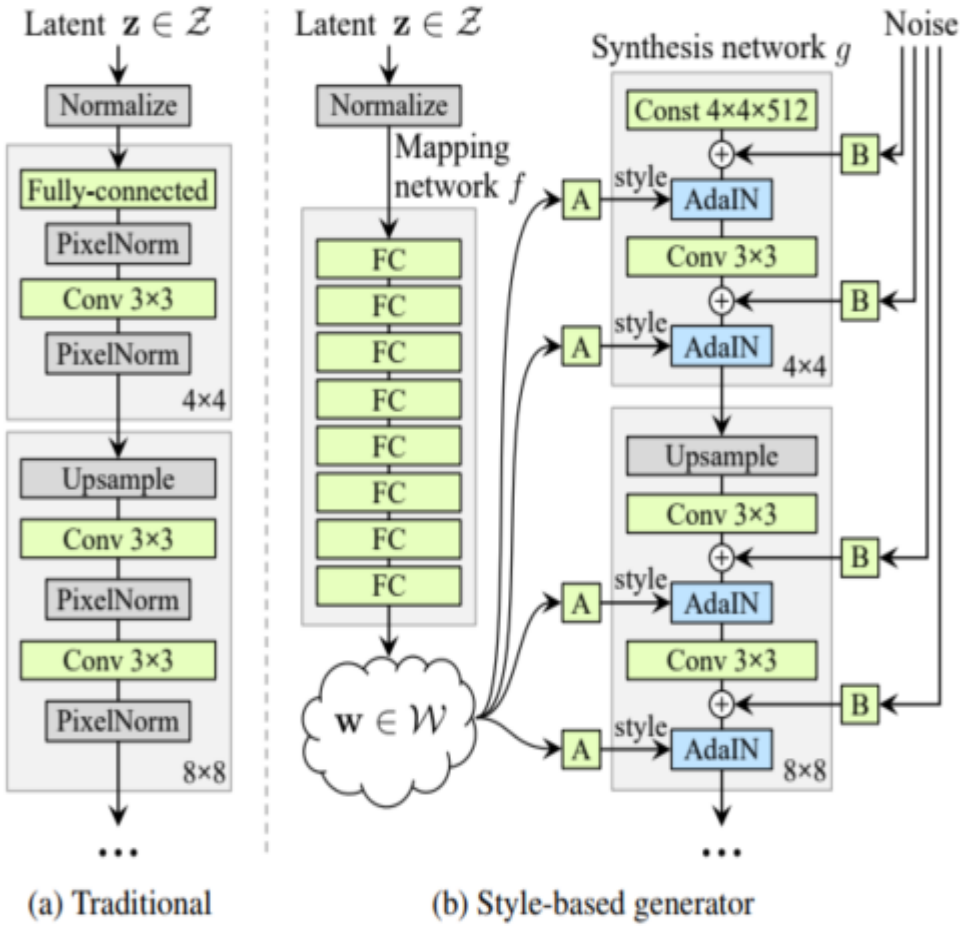
→ w는 확률 분포 제약 조건에서 자유로움 (disentanglement)

5. Latent space disentanglement에 대한 평가 지표 제시

→ Perceptual path length, Linear separability



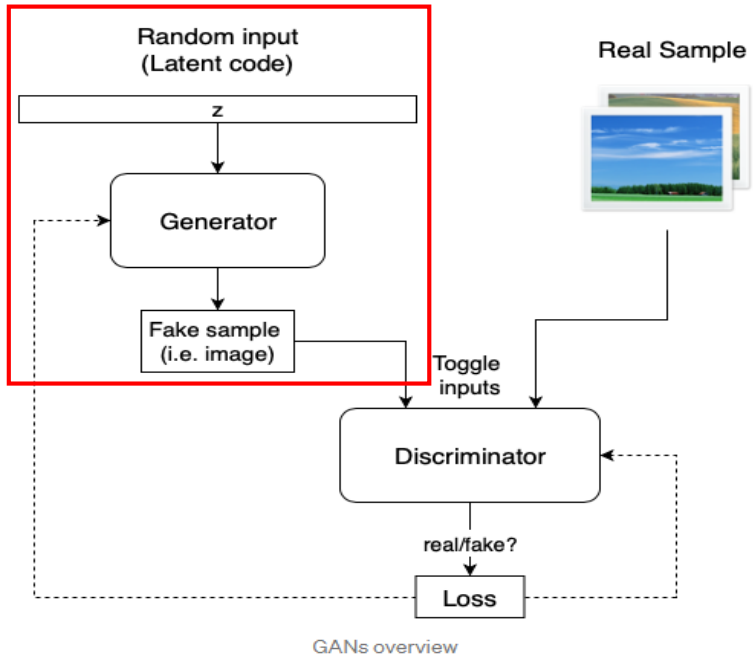
Style Based Generator



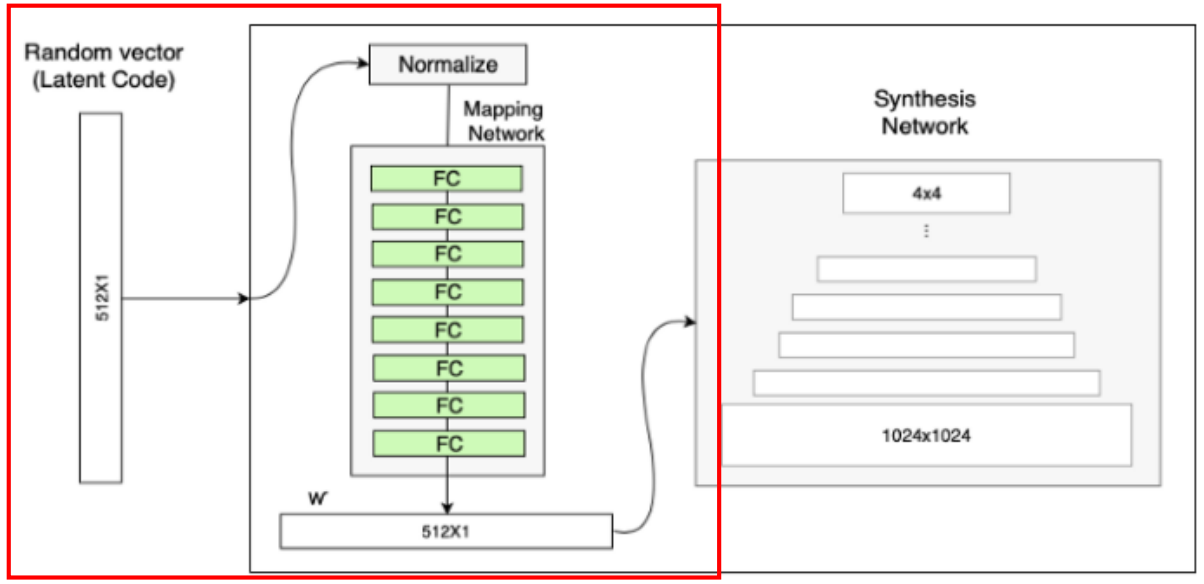
- 생성자는 latent code가 아닌 **learned constant**를 입력으로 받음
- Mapping network f가 입력 latent code z를 **intermediate latent space w**로 변환
- W는 affine 변환, AdaIN을 통해 매 convolutional layer에서 생성자의 **style**을 조정
- Noise는 생성된 이미지의 **stochastic features**를 조정
- f는 8 layers, 생성자는 18 layers로 구성
- 생성자의 최종 layer에서 1x1 convolution을 통해 RGB 이미지가 생성
- 기존 GAN보다 복잡한 모델 (26.2M > 23.1M)

StyleGAN 생성자는 linear하고 disentangled 되어 있음

Style Based Generator – Mapping Network



GANs overview



The generator with the Mapping Network (in addition to the ProGAN synthesis network)

좌 : 기존 GAN

우 : Style GAN

- 512 차원의 z 도메인에서 w 도메인으로의 **mapping** 수행
- 가우시안 분포에서 샘플링한 **z 벡터를 직접 사용하지 않음**

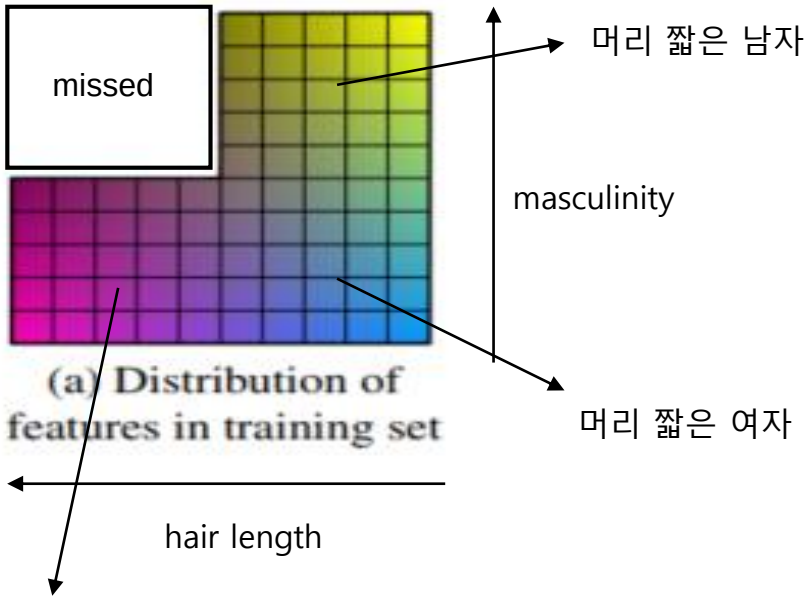
→ In W space, the factors of variation becomes more linear

Z : fixed distribution
Learned non-linear mapping **f : z → w**

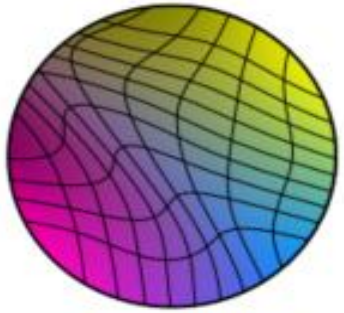
Disentanglement : *"latent space that consists of linear subspaces, each of which controls one factor of variation"*

- Z의 특정한 값을 바꿀 때 이미지의 여러 특성이 한꺼번에 변화하면 entangle

Training Image Set



Interpolation 수행



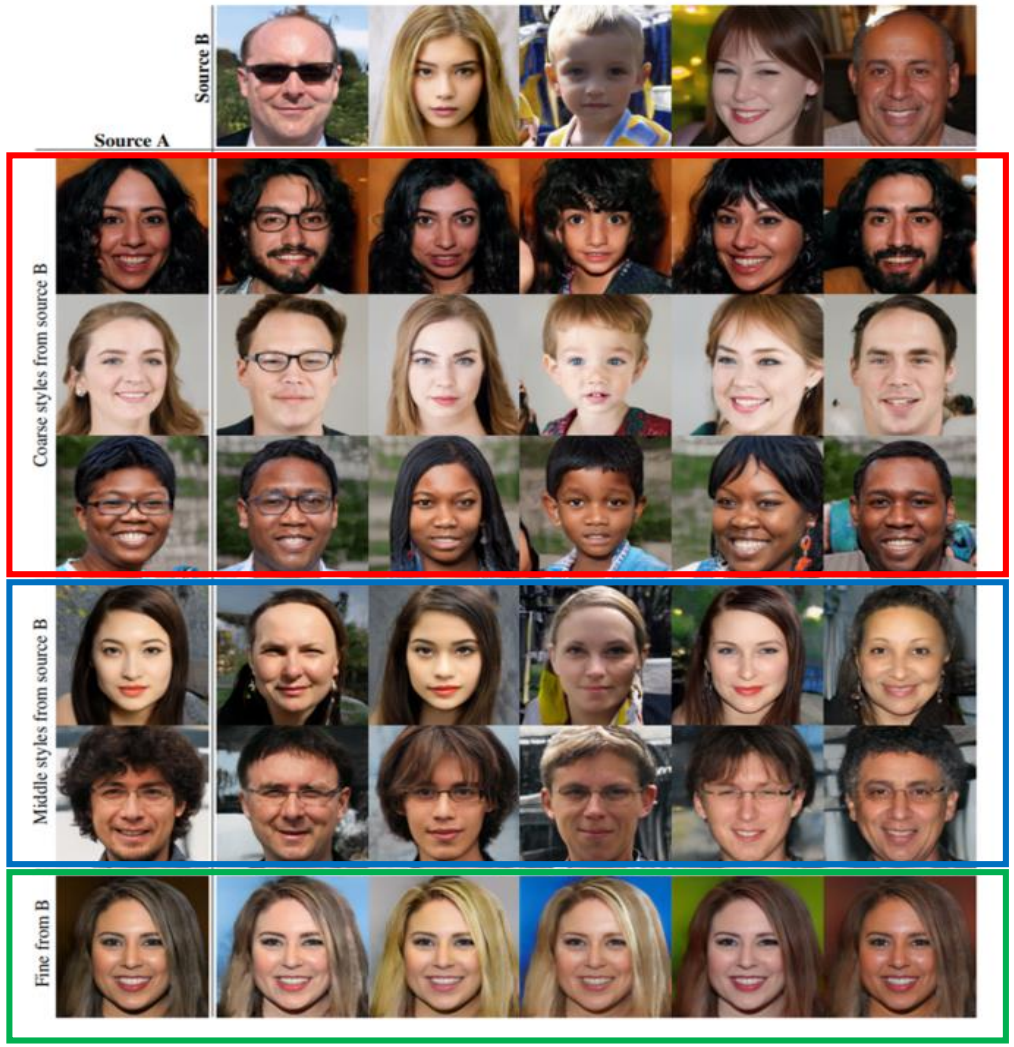
(b) Mapping from $\mathcal{Z}$ to features

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

- 가우시안 분포의 제약
- 이미지의 급격한 변화 : entanglement

- 확률 분포에 대한 제약 x
- 각각의 image feature들이 잘 분리 될 수 있는 형태로 학습 : linearly separable

Latent vector W



- 이미지 A에 이미지 B의 스타일을 적용
- Style GAN의 생성자는 총 18 layers로 구성
- Coarse styles (4 x 512)
- Middle styles (4 x 512)
- Fine styles (10 x 512)

AdaIN

- Style transfer : 특정 이미지에서 style, 다른 이미지에서 contents를 추출하여 합성 ex) 나무 + 의자

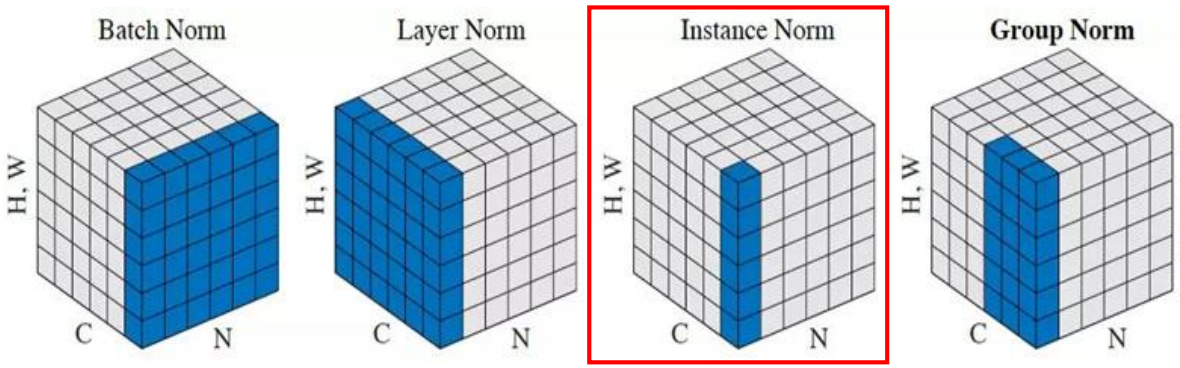
  여러 연구를 통해 이미지의 feature space상의 여러 statistics는 style을 표현하는데 유용한 것으로 밝혀짐

- 다른 원하는 데이터로부터 style 정보를 가져와 현재 이미지의 style 정보를 갱신 (학습 시킬 별도의 파라미터 x)

$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y) \qquad \mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i}\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}, \qquad (1)$$

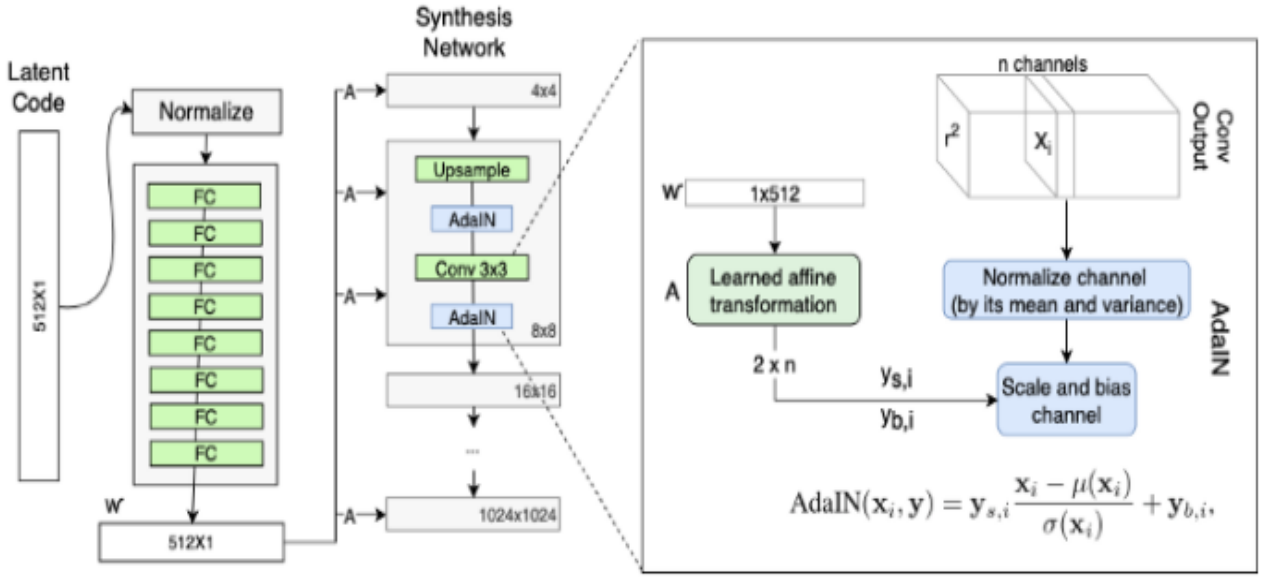
→ 내가 원하는 contents를 담고 있는 이미지의 feature x 에서, 이미지의 style을 빼주고, 입히고 싶은 style을 더해주는 방식

- Feed forward 방식의 style transfer 네트워크에서 자주 사용



→ 배치 사이즈에 관계없이 개별 이미지의 각 채널에 대해 정규화 수행

Style Based Generator – Style Modules (AdaIN)

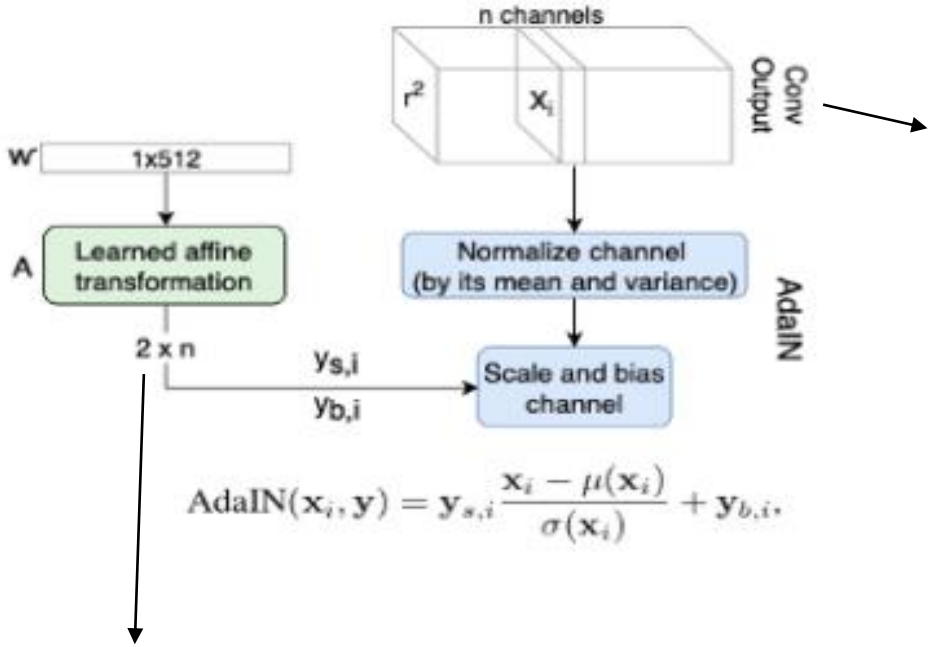


The generator's Adaptive Instance Normalization (AdaIN)

- 생성자의 각 layer는 2 개의 convolutional layer, AdaIN layer로 구성
- AdaIN layer는 convolutional 연산의 결과를 처리
- 18개의 layer를 거치며 고해상도 이미지로 upsampling 됨

- W는 학습된 affine transformation에 의해 style vector $\boldsymbol{y} = (\boldsymbol{y_s}, \boldsymbol{y_b})$로 변환, spatially invariant style y from vector w
- style 정보는 AdaIN layer의 입력으로 제공

Style Based Generator – Style Modules (AdaIN)

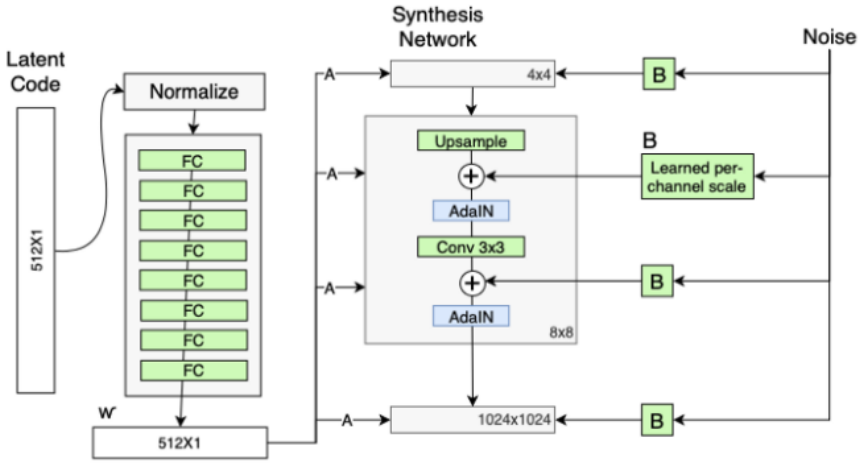


- Conv output은 n개의 채널로 구성된 Tensor
- x_i : 각 채널의 feature map

- 각 n개의 채널 마다 2개의 style 정보를 생성 $y = (y_s, y_b)$
- 각 정규화된 feature map에 대해 어느 정도 scaling하고, bias를 더할 지 결정

→ Style 반영 : y를 이용해 feature의 통계량을 변경

Style Based Generator – Stochastic Variation (Noise)



Adding scaled noise to each resolution level of the synthesis network

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

- Noise Input

→ 다양한 확률적인 특성들을 컨트롤, 미세한 특정 부분만 변경 (리터칭)

ex) 주근깨, 머리카락

→ broadcasted to all feature maps

- Noise는 확률적 측면에만 영향을 끼치고 overall composition, high level aspects는 보존

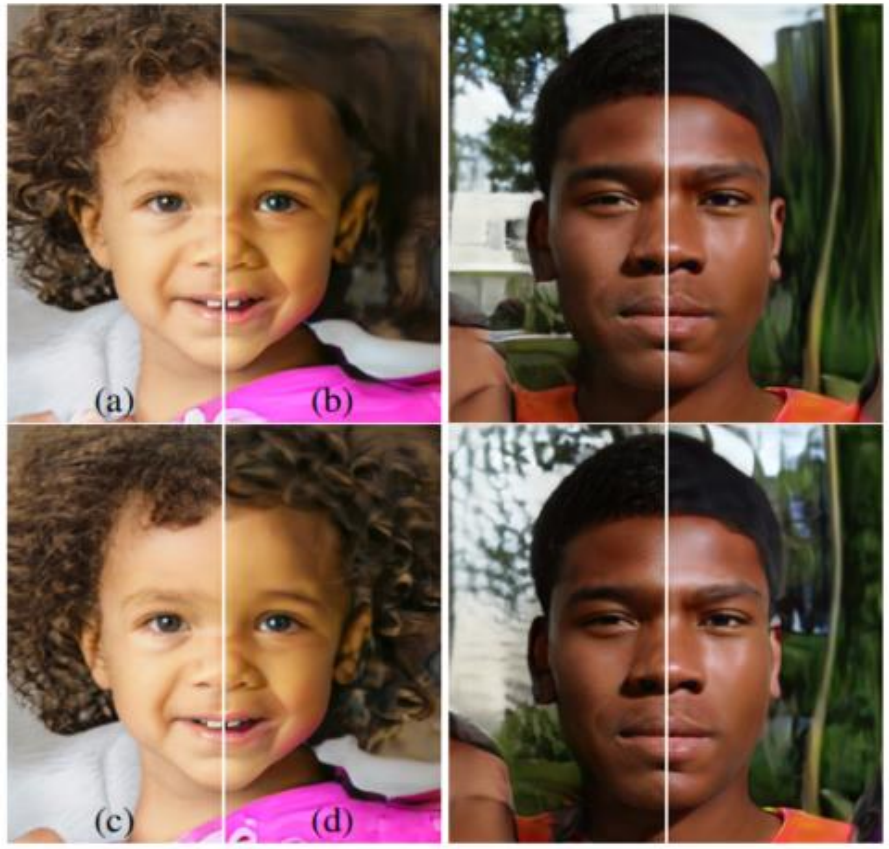
* Style : high level attributes ex) 얼굴형, 안경, 포즈

(c) : Standard deviation of each pixel

→ 이미지의 어느 부분이 noise의 영향을 받는 지 표시 (영향을 크게 받으면 흰색)

→ 인종, 포즈 같은 Global aspects는 변하지 않음

Style Based Generator – Stochastic Variation



- Noise 제거는 생성된 이미지의 품질을 저하
- Coarse noise (d) : 큰 크기의 머리 곱슬, 배경
- Fine noise (c) : 세밀한 머리 곱슬, 배경

(a) : 모든 layer에 noise를 입력으로 제공

(b) : Noise x

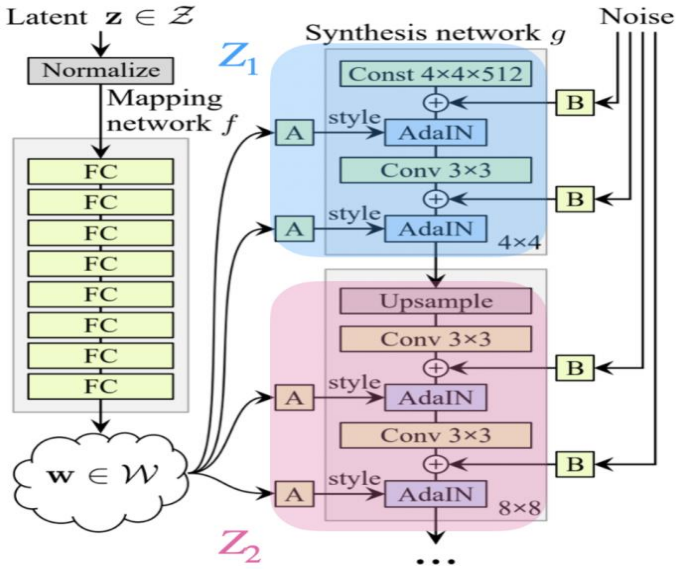
(c) : Fine layers에만 noise 제공

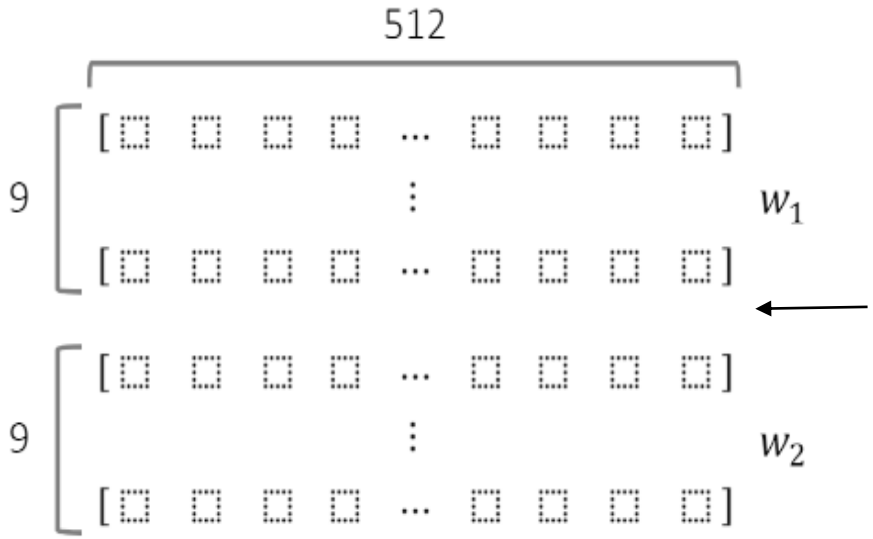
(d) : Coarse layers에만 noise 제공

Properties of Style GAN – Style Mixing (Mixing Regularization)

- 인접한 layer간의 style correlation을 줄임 – style을 각 layer에 대해 localize
- 동일한 latent vector z_1에서 출력된 w_1 하나만 이용해 계속 학습하면 style correlation이 발생
- Multi latent z 이용

→ Latent space에서 추출한 $z_1, z_2, \ldots z_n$을 mapping network f에 통과시켜 $w_1, w_2, \ldots w_n$ 생성





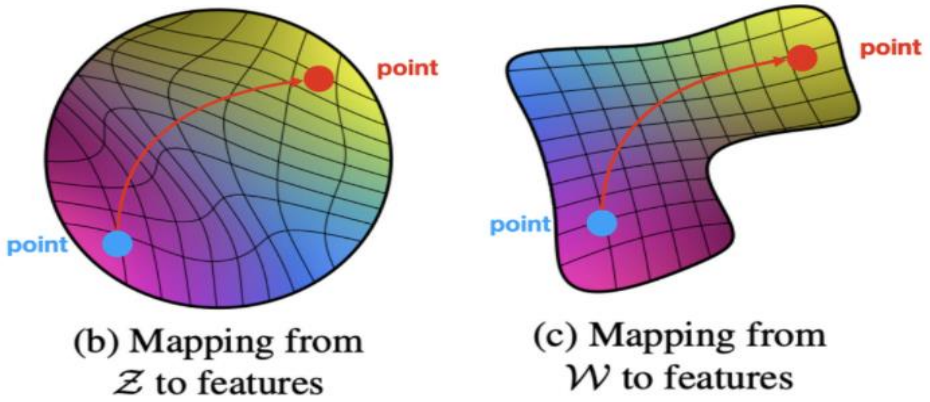
Crossover point

Crossover 이전은 w_1, 이후는 w_2 사용

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**

- Intermediate latent space W의 샘플링은 고정된 확률 분포의 제약을 받지 않음 (학습된 mapping network에 의해 샘플링 분포가 결정됨)

  → The factors of variation becomes more linear



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

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

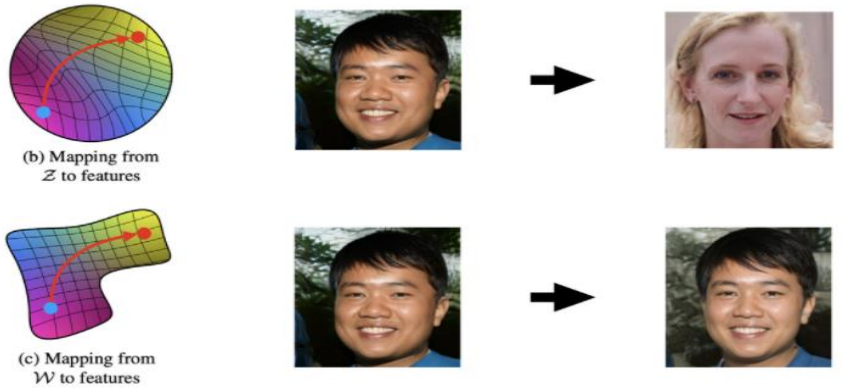
- Disentangled representation을 통해 생성자는 보다 현실적인 이미지를 생성

- 기존에는 입력 이미지를 latent code로 변환하는 encoder 네트워크를 필요로 했음 (Style GAN에 부합 x)

  → Disentanglement에 대한 새로운 평가 지표를 제안

Disentanglement studies – Perceptual Path length

- 2개의 latent vectors를 interpolation할 때 얼마나 급격하게 이미지의 특성이 변화하는 지 (surprisingly non-linear changes)
  ex) 2 벡터의 endpoint에 없는 특징들이 linearly interpolated 이미지에 등장
- 이미지의 급격한 변화는 latent space가 entangled되어 있다는 것을 의미함



계산

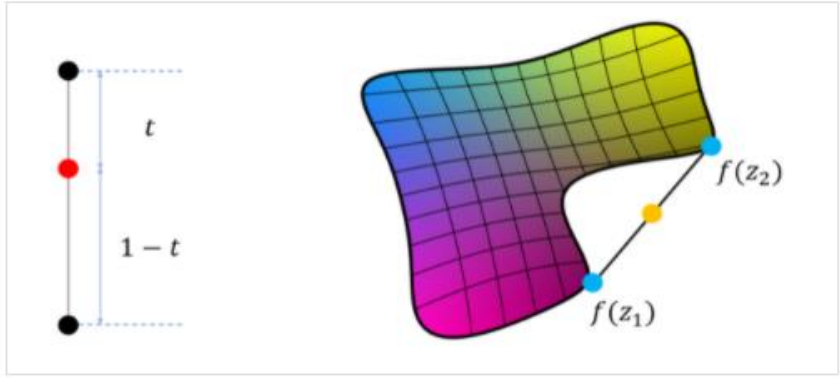
1. 사전 학습된 VGG16에 z_1, z_2로 생성된 이미지를 입력으로 넣어 embedding
2. Embedding된 features를 바탕으로 perceptual difference를 계산
   → slerp, lerp : interpolation operation

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

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

→ 지점 t, $t+\epsilon$ 사이에서의 VGG features의 거리가 얼마나 먼지 계산
(weighted difference btw two VGG16 embeddings)

- Full path : $t \sim U(0,1)$
- End path : $t \in \{0,1\}$



Disentanglement studies – Linear Separability

- Latent space가 잘 disentangled되어 있다면 개별 variation factor를 구별하는 정확한 방향 벡터를 찾을 수 있어야 함
- One style – one style direction vector

- 웃음 벡터의 연속 공간
  (웃음과 관련된 요소들만 변화함)

- Latent space의 점들을 linear hyperplane으로 잘 구별할 수 있다면 latent space는 disentangled

→ 얼굴 마다 성별 등 40개의 binary attributes가 명시되어 있는 CelebA-HQ 데이터를 사용하여 분류 모델 학습

Disentanglement studies – Linear Separability

계산

1. 매 attribute 마다 200000개의 이미지를 생성하고, Auxiliary Classification Network의 입력으로 제공
2. Confidence가 낮은 절반을 제거 (100000개의 label이 명시된 latent space vector 준비)
3. Latent space point를 바탕으로 class label을 예측하는 linear SVM을 학습
4. Conditional entropy $\boldsymbol{H(Y|X)}$계산 (X : SVM 예측 라벨, Y : A.C.N으로 예측된 라벨)

   → true class를 결정하기 위해 additional information이 얼마나 필요한 지 (낮을 수록 예측이 잘 되었다는 것을 의미)
5. Separability score $\mathbf{exp}(\sum_i \boldsymbol{H(Y_i|X_i)})$, $i = \#\ of\ attributes$ 계산

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**

- W가 z보다 separable함 (less entangled representation)

생성 모델의 과제 & 해결 방안

- 훈련 데이터에 잘 표현되지 않는 (low density) 영역의 이미지들을 생성
- 생성자는 훈련 데이터에 적게 나타난 특징들을 학습할 수 없고, 그와 닮은 이미지를 생성할 수 없거나 낮은 품질의 이미지를 생성

→ W의 truncated/shrunk space에서 latent vector를 샘플링

→ 이미지 품질을 높이지만 variation의 손실 발생

BigGAN의 truncated trick

- 샘플링한 입력 노이즈 중 ths를 넘는 것은 다시 샘플링하여 ths안에 포함되도록 함

오른쪽으로 갈 수록 ths 값이 낮아짐

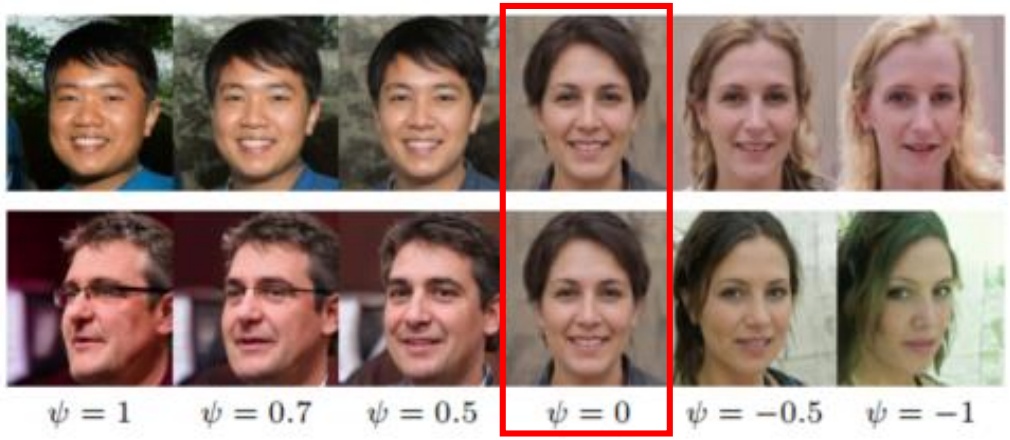
- Ths가 최소일 때는 다양성이 손실 (loss of variation)
- Ths가 최대일 때는 이미지의 품질이 낮아짐

Truncation trick in w

Center of mass of w : $\bar{w} = E_{z\sim P(z)}[f(z)]$ – average face

Scale the deviation of a given w from the center : $W' = \bar{w} + \varphi(w - \bar{w})$, φ : style scale

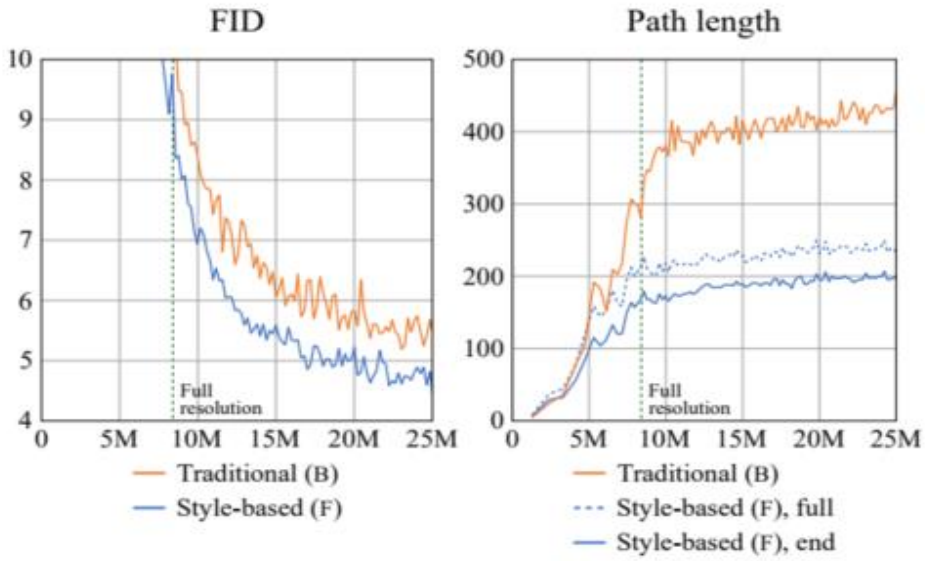
→ 평균적인 이미지로부터 얼마나 거리가 먼 이미지를 생성할 지 결정



- $\varphi = 0$: average face로 수렴
- 양 극단을 비교할 때 안경, 나이, 머리 길이, 성별 등 high level attributes (style)이 서로 반대

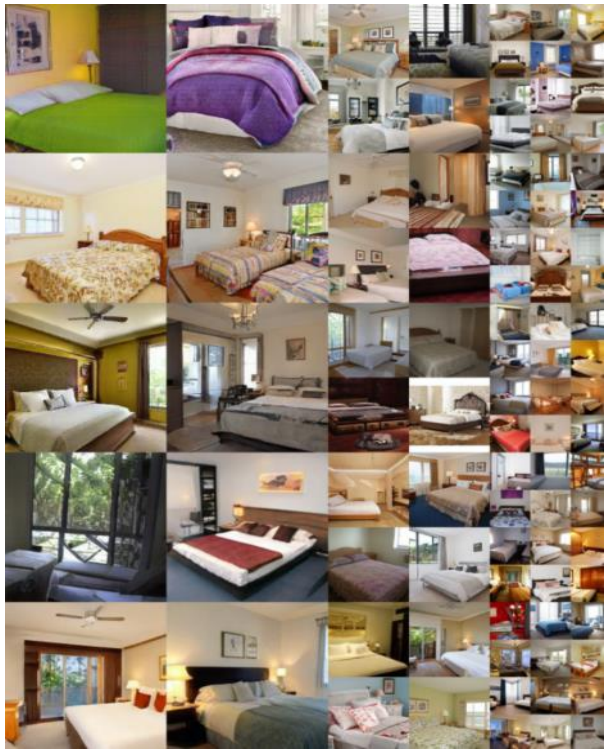
→ Truncation in w space seems to work reliably even without changes to the loss function

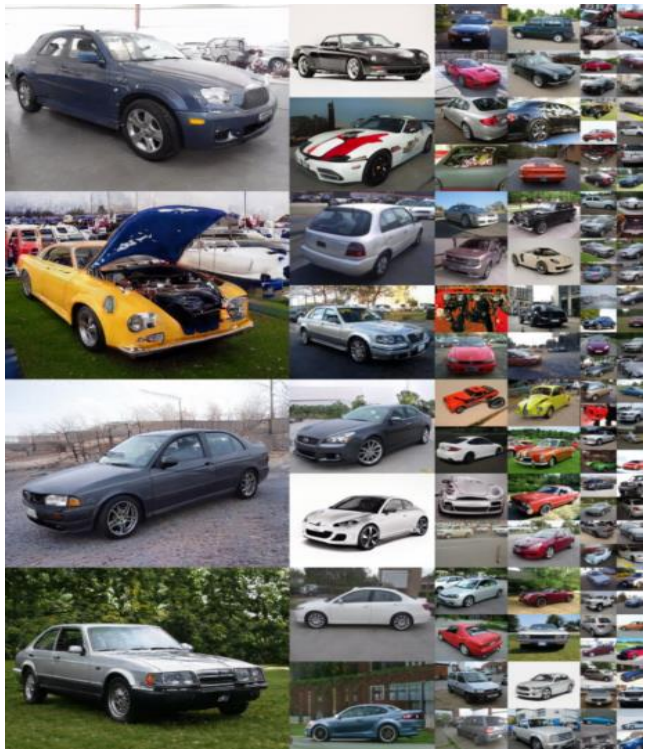
Appendix – Training details & Experiments



FID : Frechet Inception Distance

(거리가 가까울 수록 좋은 영상)





다양한 유형의 고해상도 이미지를 생성 →

Figure 7. The FFHQ dataset offers a lot of variety in terms of age, ethnicity, viewpoint, lighting, and image background.