

꼼꼼한 딥러닝 논문 리뷰와 코드 실습

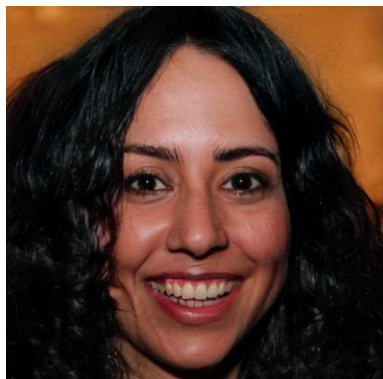
Deep Learning Paper Review and Code Practice

나동빈(dongbinna@postech.ac.kr)

Pohang University of Science and Technology

논문 소개: StyleGAN (CVPR 2019)

- 고화질 이미지 생성에 적합한 아키텍처를 제안합니다.
 - ① PGGAN 베이스라인 아키텍처의 성능을 향상시켰습니다.
 - ② Disentanglement 특성을 향상시켰습니다.
 - ③ 고해상도 얼굴 데이터셋(FFHQ)을 발표했습니다.



Source A



Source B



Coarse Styles
from Source B



관련 연구: Generative Adversarial Networks (GAN)

- 생성자(generator)와 판별자(discriminator) 두 개의 네트워크를 활용한 생성 모델입니다.
- 다음의 목적 함수(objective function)를 통해 생성자는 이미지 분포를 학습할 수 있습니다.

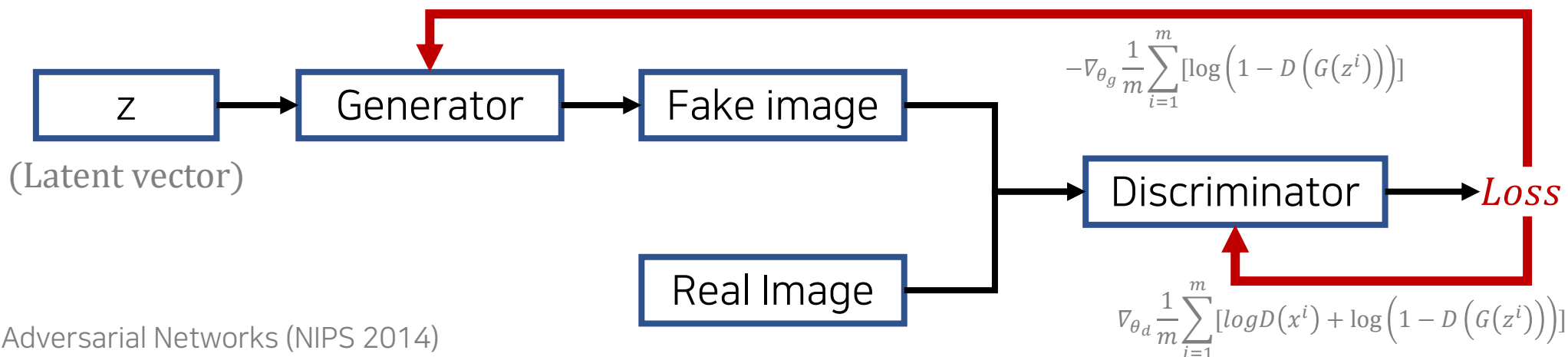
$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Generator

$G(z)$: new data instance

Discriminator

$D(x)$ = Probability: a sample came from the real distribution (Real: 1 ~ Fake: 0)



Generative Adversarial Networks (NIPS 2014)

관련 연구: DCGAN (ICLR 2016)

- Deep Convolutional Layers를 이용하여 이미지 도메인에서의 높은 성능을 보입니다.

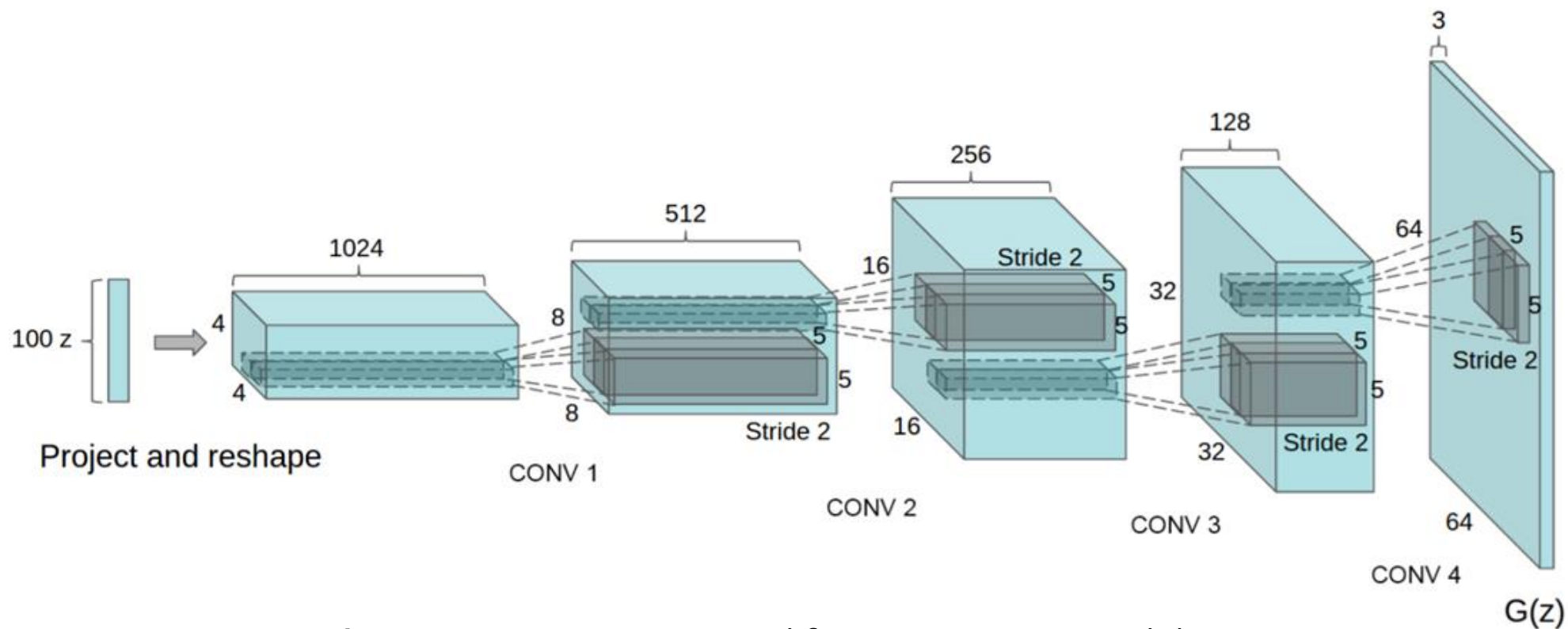
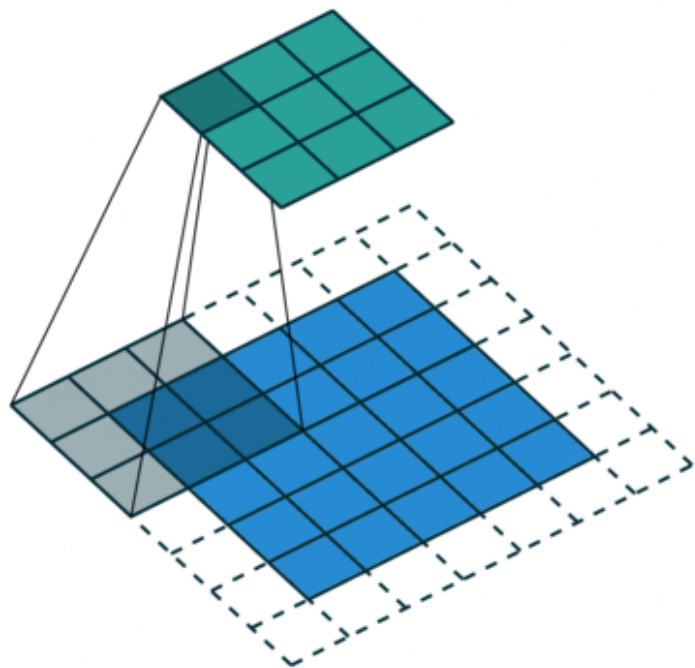


Fig. DCGAN generator used for LSUN scene modeling.

Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (ICLR 2016)

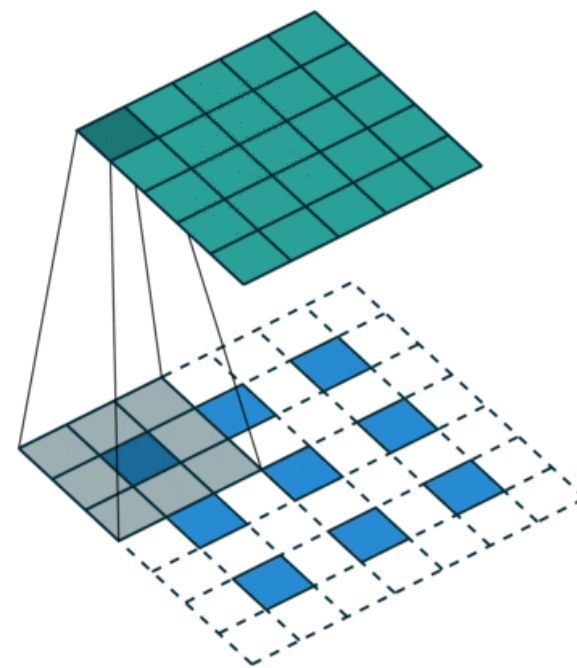
관련 연구: DCGAN에서의 Convolutional 필터(Filter) 소개

판별자(Discriminator)



Strided Convolution: 너비와 높이가 감소

생성자(Generator)



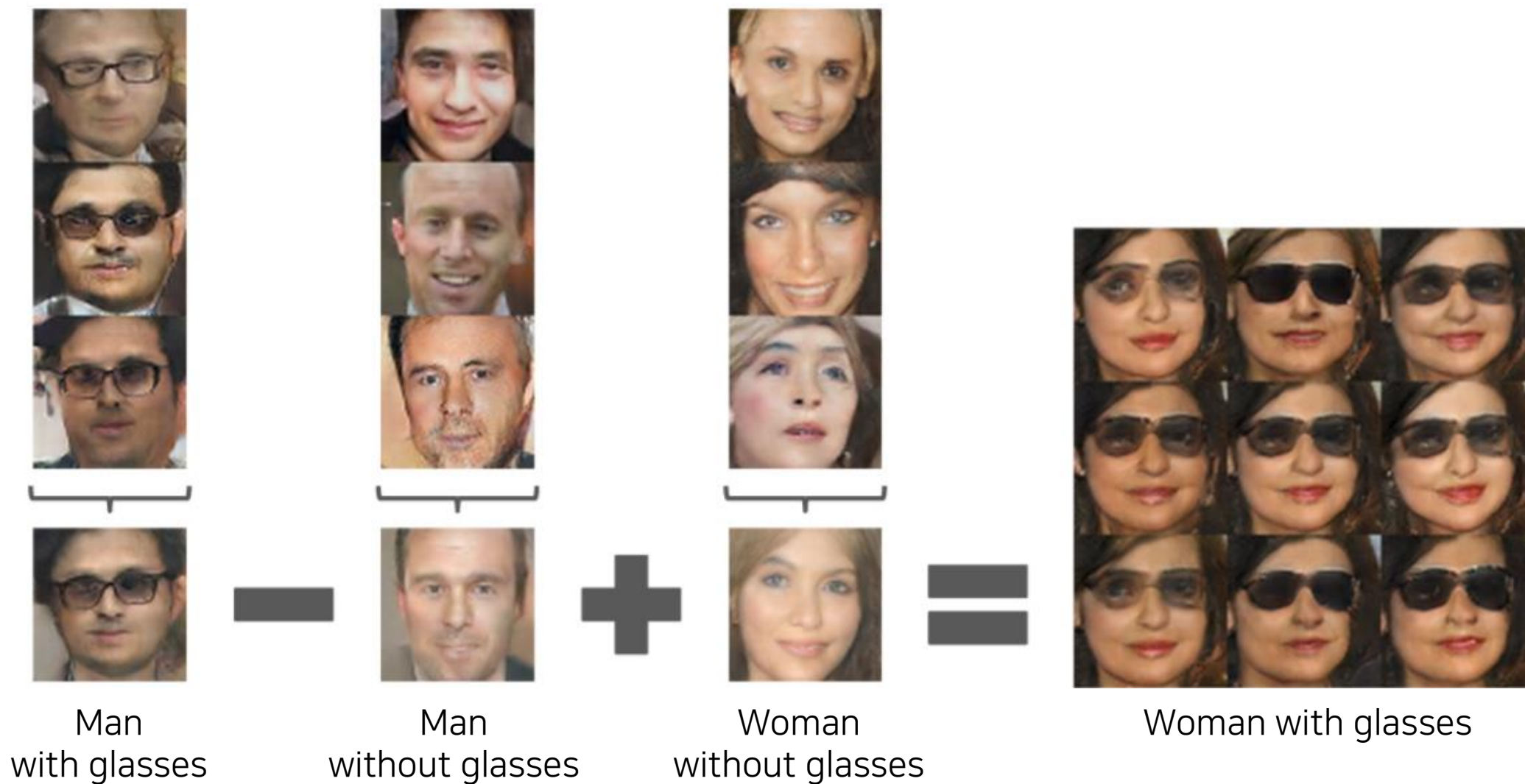
Transposed Convolution: 너비와 높이가 증가

Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (ICLR 2016)

A Style-Based Generator Architecture for Generative Adversarial Networks (CVPR 2019)

나동빈

관련 연구: DCGAN에서의 벡터 연산(Vector Arithmetic)



관련 연구: WGAN-GP

- WGAN은 함수가 1-Lipshichtz 조건을 만족하도록 하여 안정적인 학습을 유도합니다.
 - 본래 WGAN 논문은 weight clipping을 이용하여 제약 조건을 만족하도록 합니다.
- **WGAN-GP**에서는 gradient penalty를 이용하여 WGAN의 성능을 개선합니다.

$$L = \underbrace{\mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g} [D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} [D(\mathbf{x})]}_{\text{Original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]}_{\text{Gradient penalty}}$$

(Unsupervised case)
Inception Score 측정 결과



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

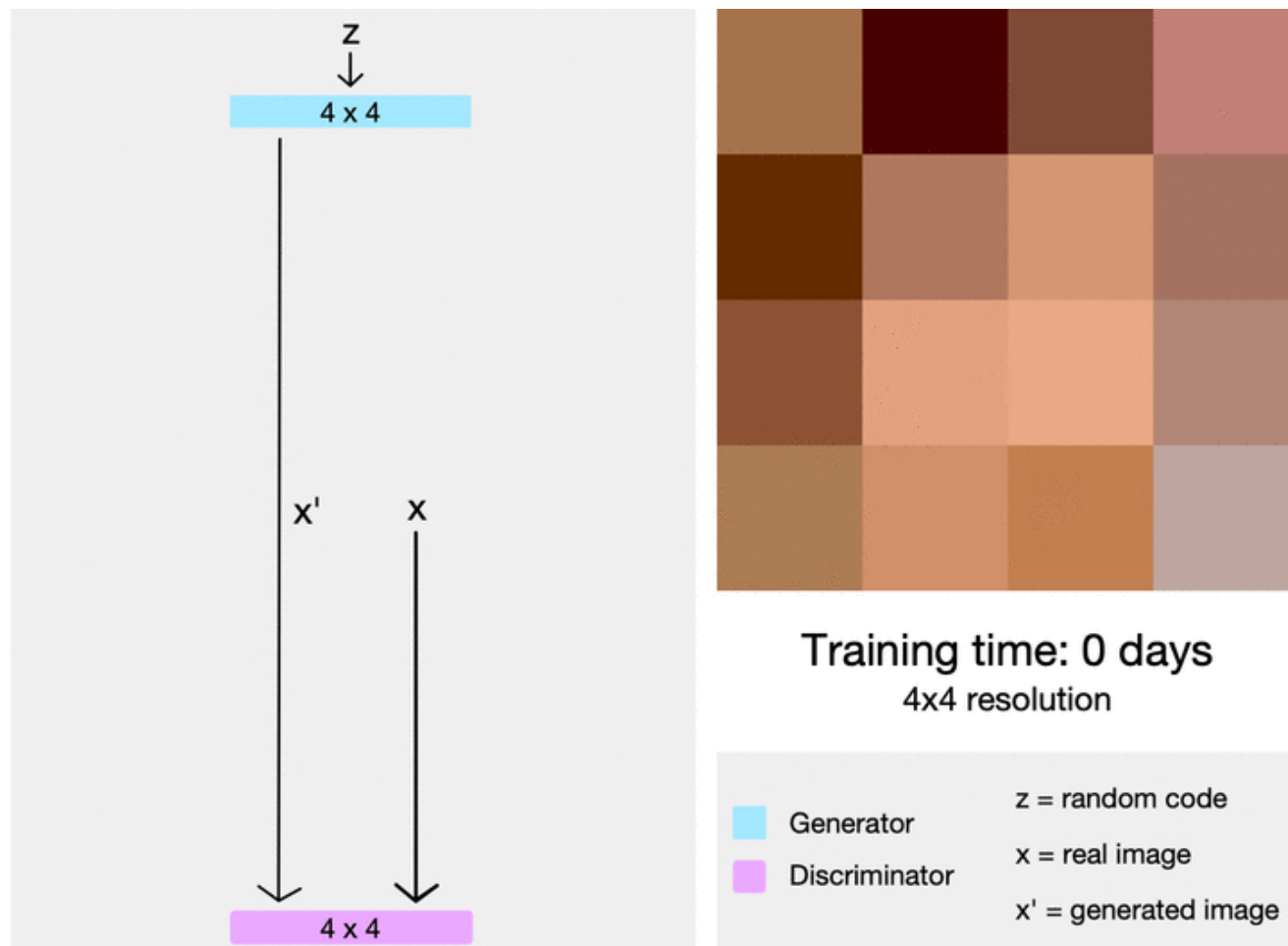
Improved Training of Wasserstein GANs (NIPS 2017)

연구 배경: Progressive Growing of GANs (PGGAN = ProGAN)

- 메인 아이디어
 - 학습 과정에서 레이어를 추가
 - 고해상도 이미지 학습 성공
- 한계점
 - 이미지의 특징 제어가 어려움



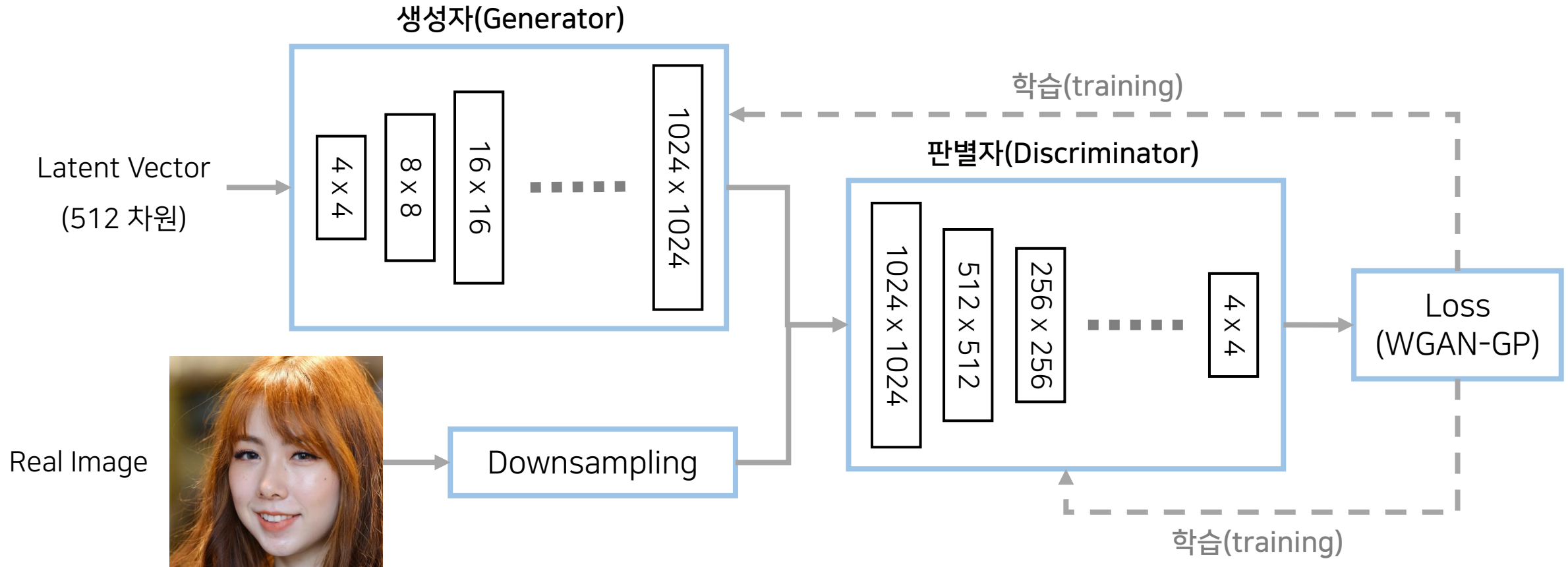
StyleGAN에서 개선됩니다.



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

연구 배경: Progressive Growing of GANs (PGGAN = ProGAN)

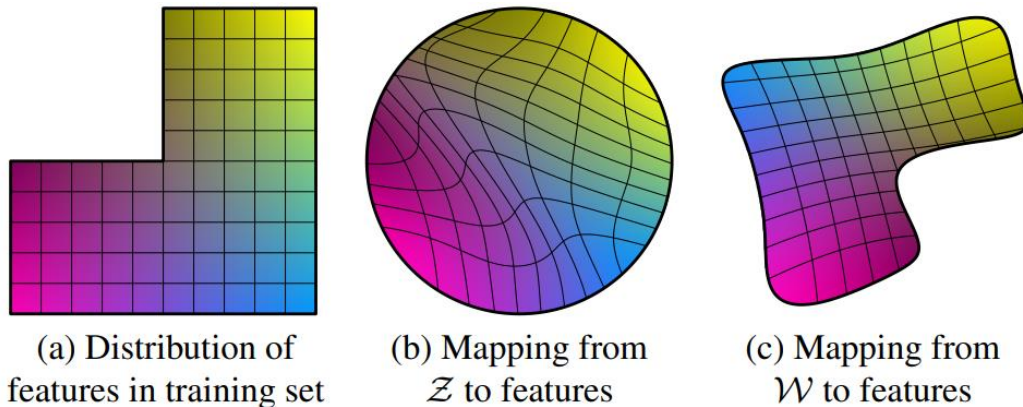
- 학습을 진행하는 과정에서 점진적으로(progressively) 네트워크의 레이어를 붙여 나갑니다.



<https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431>

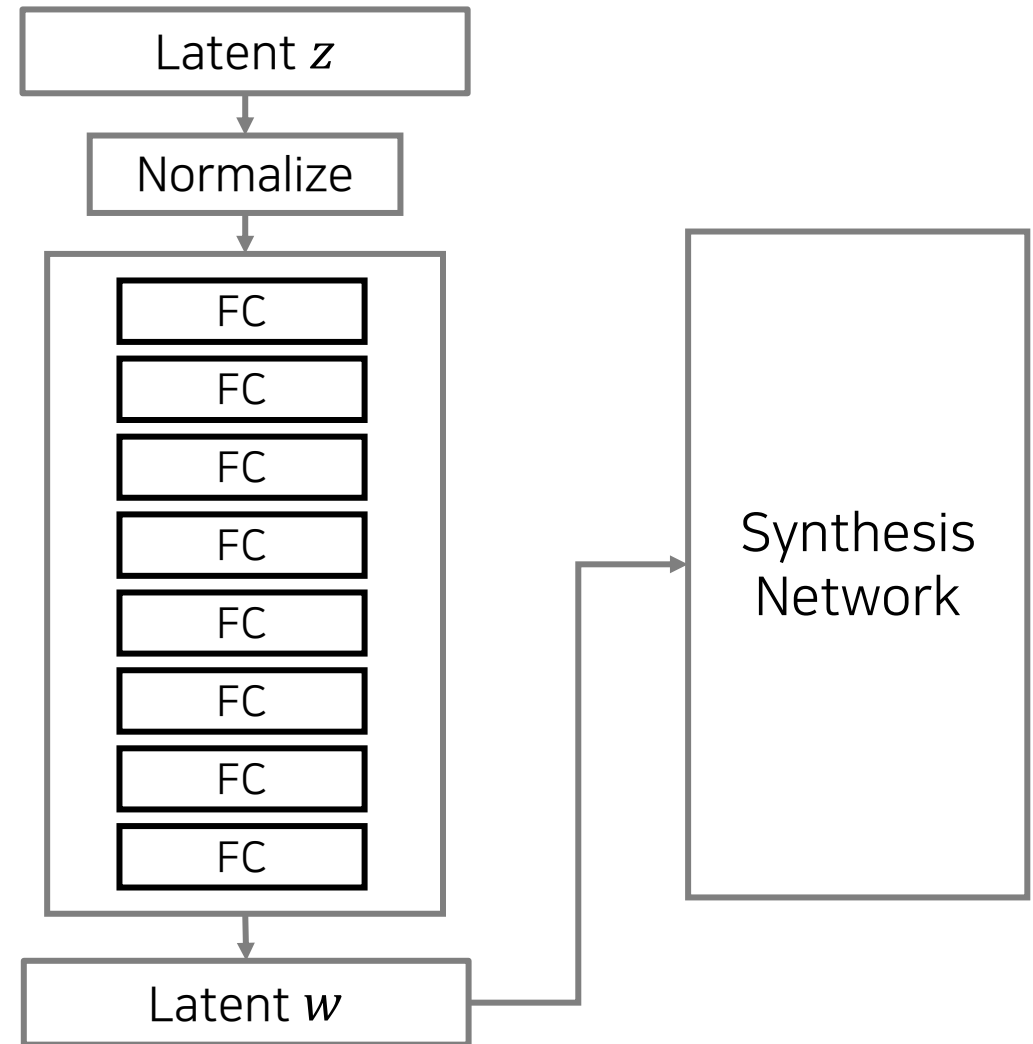
StyleGAN의 핵심 아이디어: 매핑 네트워크 (Mapping Network)

- 512차원의 z 도메인에서 w 도메인으로 매핑을 수행합니다.
- 가우시안 분포에서 샘플링한 z 벡터를 직접 사용하지 않습니다.
 - 계산된 w 벡터를 사용할 때가 효과가 좋습니다.



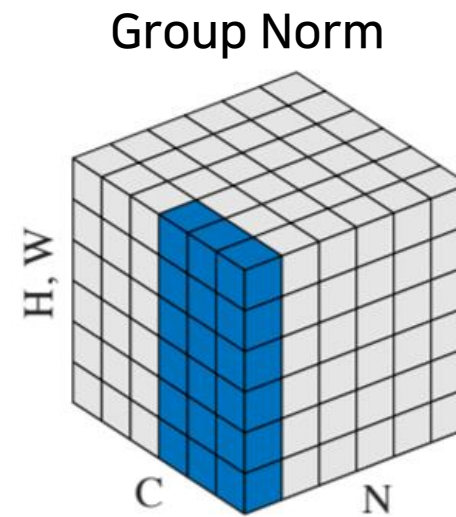
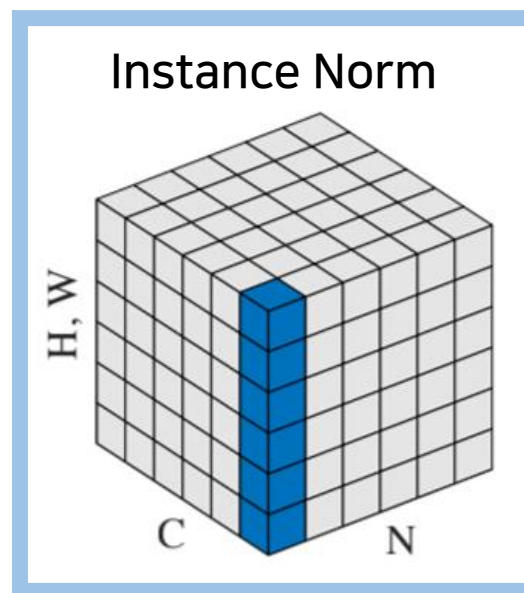
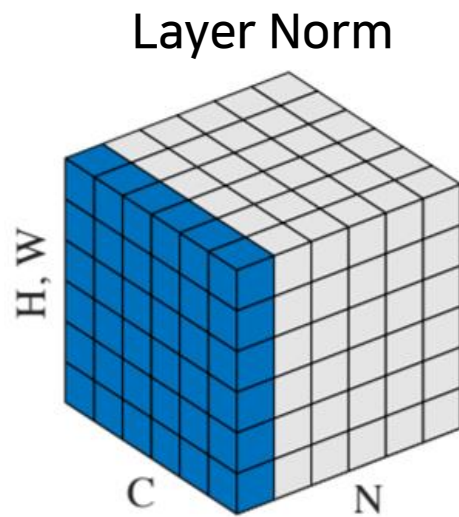
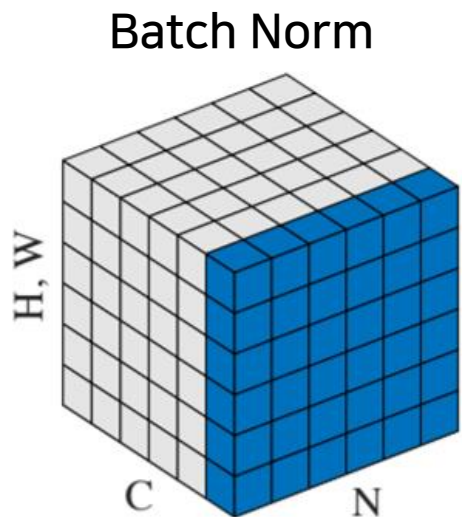
In W space, the factors of variation become more linear.

Z : Fixed distribution
Learned mapping $f: z \rightarrow w$



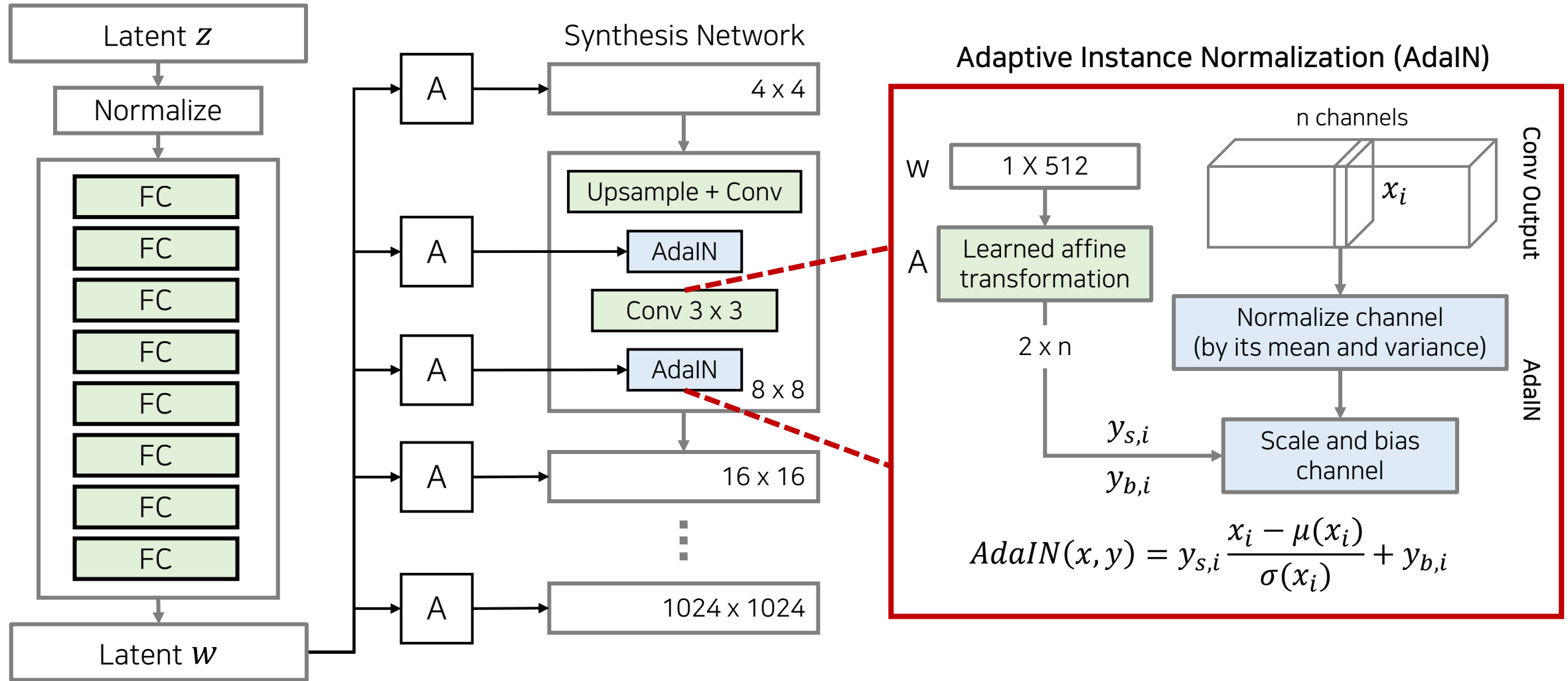
관련 연구: Adaptive Instance Normalization (AdaIN)

- AdaIN을 이용하면 다른 원하는 데이터로부터 스타일(style) 정보를 가져와 적용할 수 있습니다.
 - 학습시킬 파라미터가 필요하지 않습니다. (γ 와 β 사용하지 않음)
 - feed-forward 방식의 style transfer 네트워크에서 사용되어 좋은 성능을 보입니다.



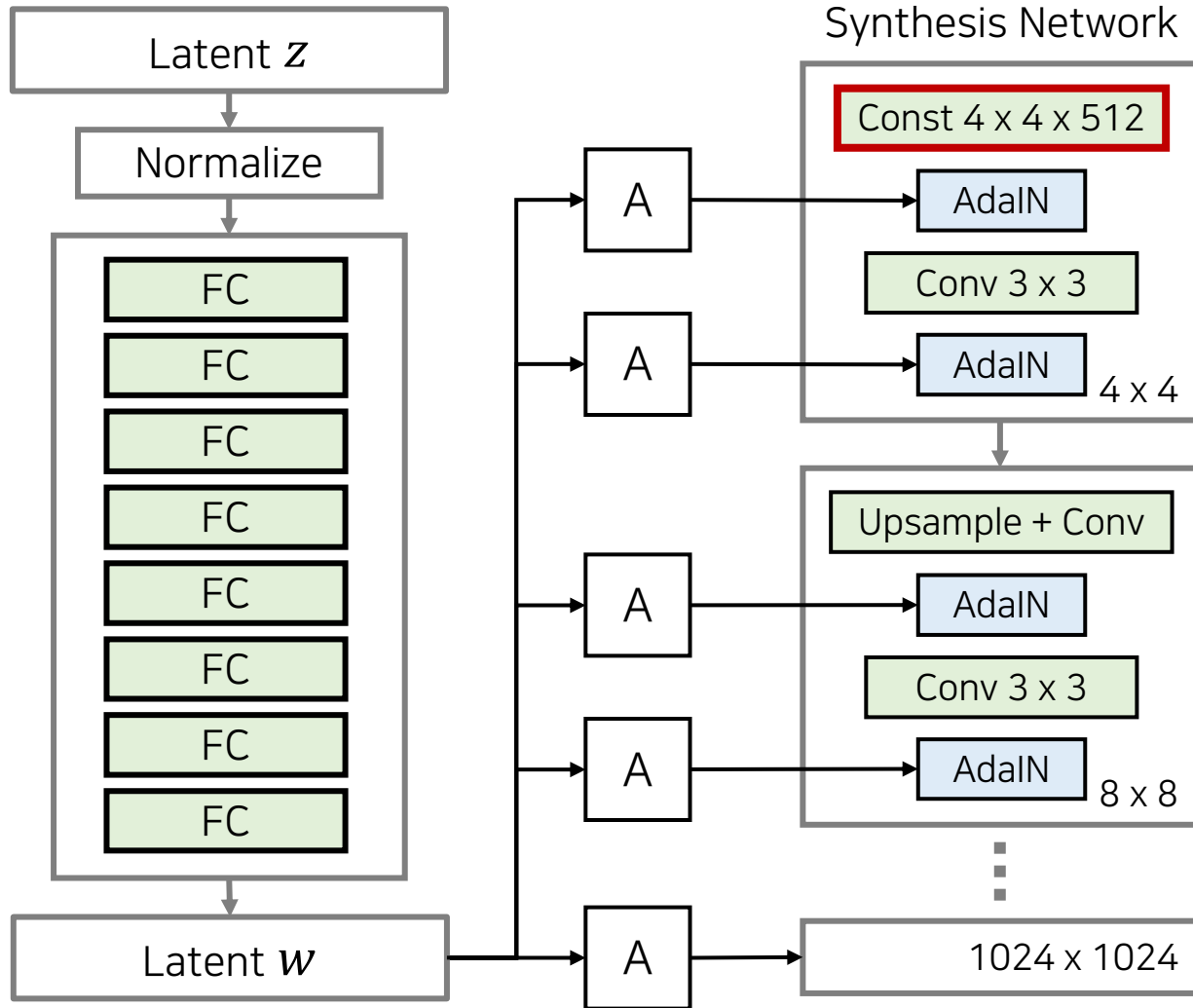
Group Normalization (ECCV 2018)

Style Modules (AdaIN)



<https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431>

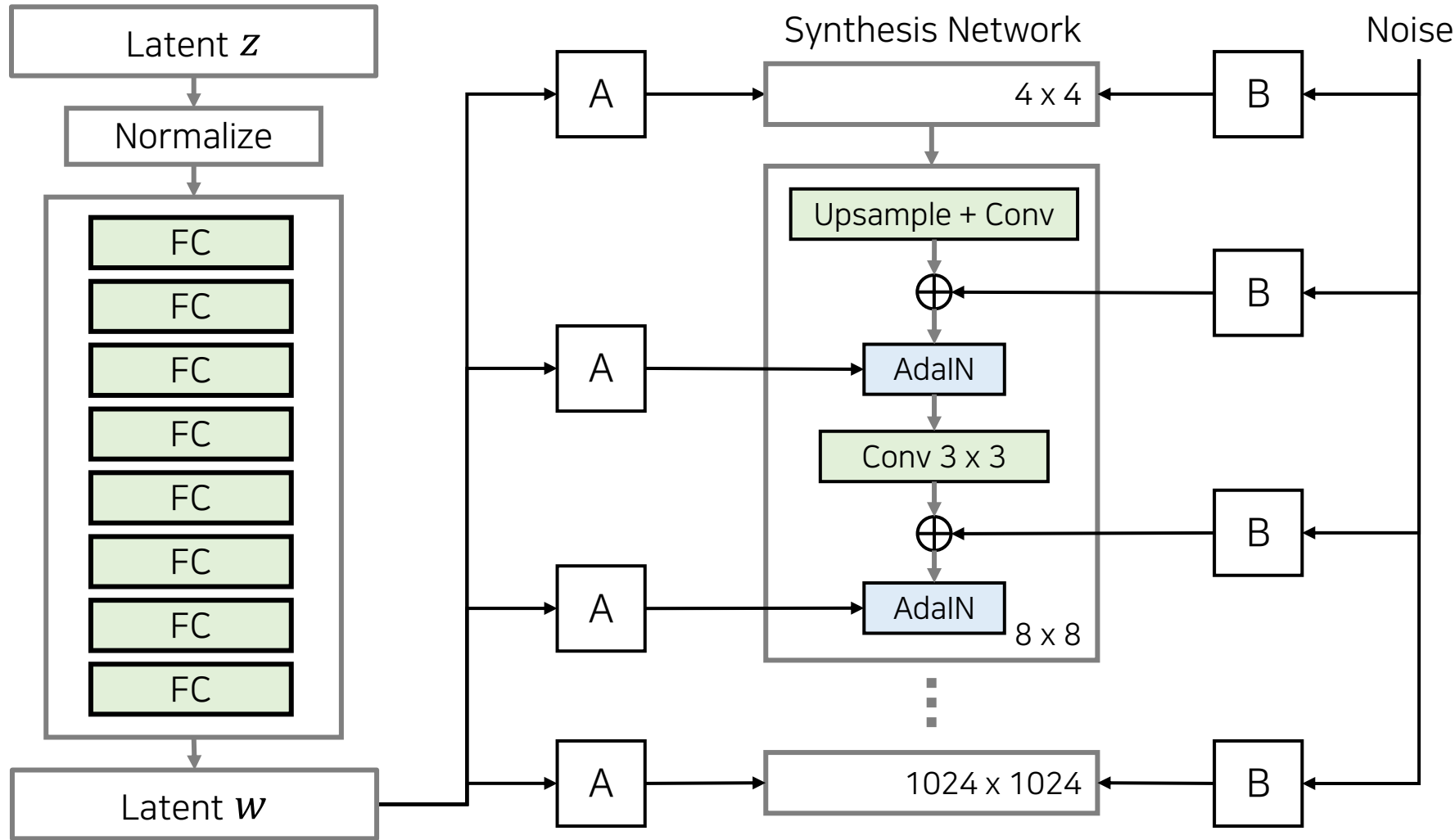
Removing Traditional Input



- 초기 입력을 상수(constant)로 대체합니다.
- 경험적으로 성능이 향상됩니다.

<https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431>

Stochastic Variation



다양한 확률적인 측면 컨트롤

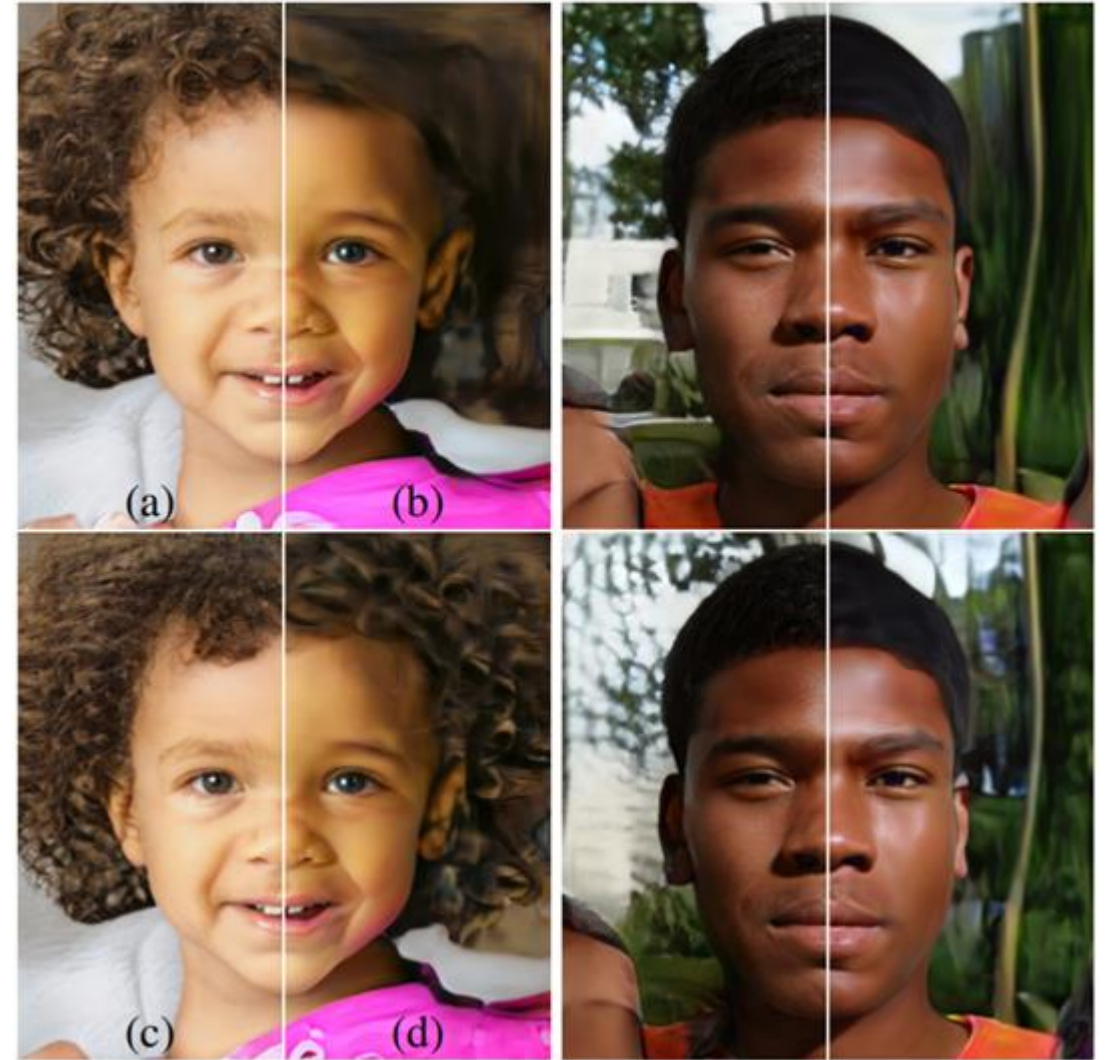
- 주근깨
- 머리카락의 배치

<https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431>

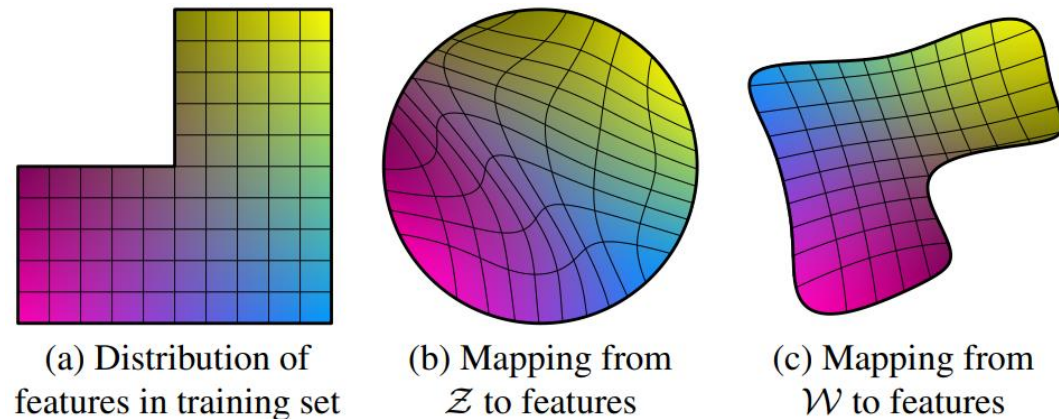
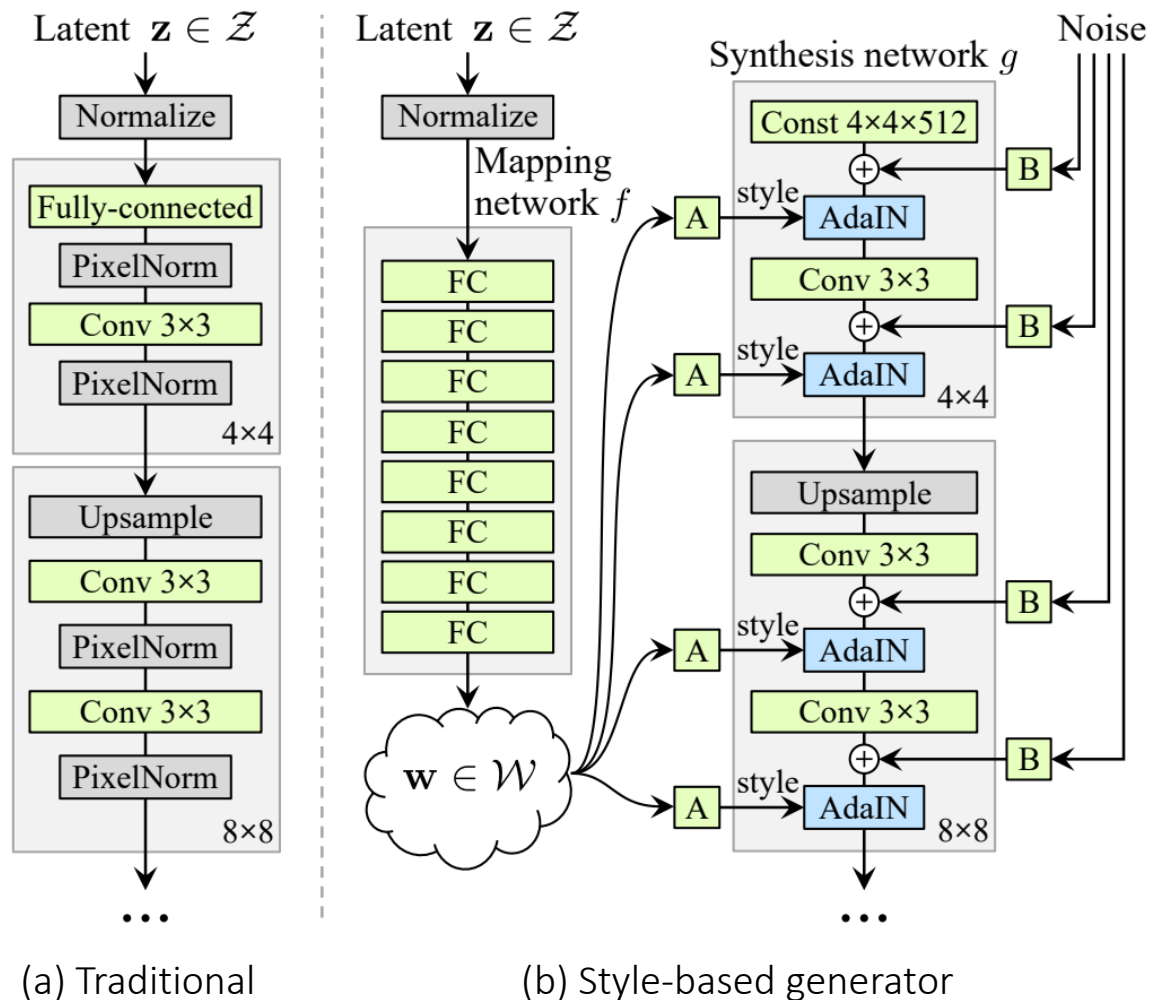
Stochastic Variation

- 스타일: high-level global attributes
 - 얼굴형, 포즈, 안경의 유무 등
- 노이즈: stochastic variation
 - 주근깨(freckle), 피부 모공(skin pore)
 - Coarse noise: 큰 크기의 머리 곱슬거림, 배경 등
 - Fine noise: 세밀한 머리 곱슬거림, 배경 등

- (a) 모든 레이어에 노이즈 적용
- (b) 노이즈 적용하지 않음
- (c) Fine layer에 적용
- (d) Coarse layer에 적용



StyleGAN 아키텍처: Disentanglement Properties of StyleGAN



In \mathcal{W} space, the factors of variation become more linear.

\mathcal{Z} : Fixed distribution
Learned mapping $f: z \rightarrow w$

StyleGAN의 생성자는 더욱 linear하며 덜 entangled되어 있습니다.

Latent Vector Meanings of StyleGAN



Evaluation: FID 값 비교/분석

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

[Table] Frechet Inception Distance (FID) for various generator designs.

(A) PGGAN 베이스라인

(B) Bilinear up/downsampling operations

(C) Mapping Network + Adaln

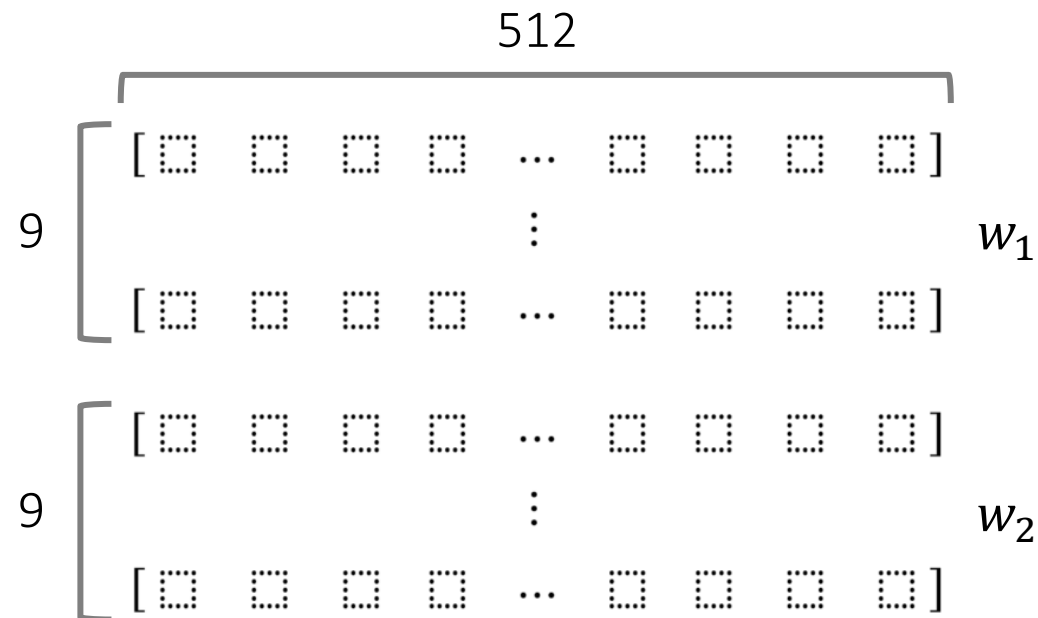
(D) Input 레이어로 학습된 $4 \times 4 \times 512$ 상수 텐서 사용

(E) 노이즈 입력

(F) Mixing Regularization

Style Mixing (Mixing Regularization)

- 인접한 레이어 간의 스타일(style) 상관관계를 줄입니다.
- 구체적인 Mixing Regularization 방법 설명
 - 1) 두 개의 입력 벡터를 준비합니다.
 - 2) 크로스오버(crossover) 포인트를 설정합니다.
 - 3) 크로스오버 이전은 w_1 , 이후는 w_2 를 사용합니다.
- 스타일은 각 레이어에 대하여 지역화(localized)됩니다.



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

Disentanglement 관련 두 가지 성능 측정 지표 제안

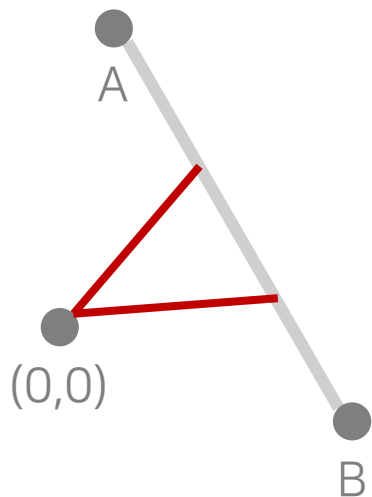
1. Path Length: 두 벡터를 보간(interpolation)할 때 얼마나 급격하게 이미지 특징이 바뀌는지
2. Separability: latent space에서 attributes가 얼마나 선형적으로 분류될 수 있는지 평가

Method	FID	Path length		Separa- bility
		full	end	
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

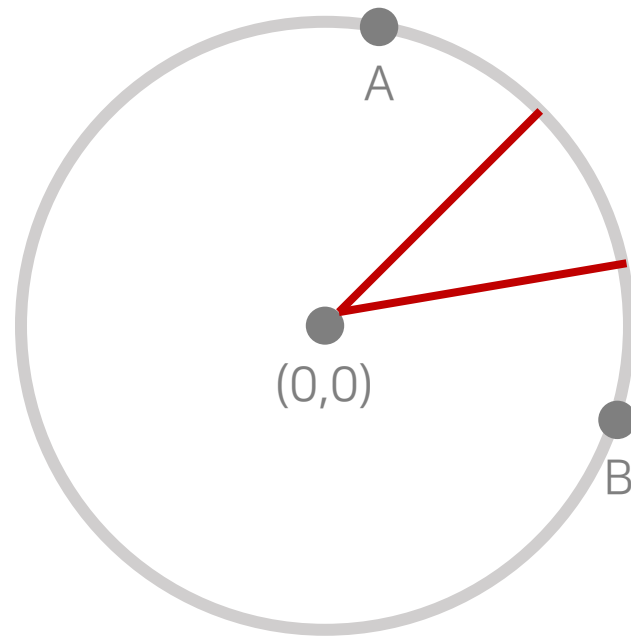
- \mathcal{W} 공간(space)이 \mathcal{Z} 공간보다 이상적인 성질을 가지고 있습니다.

Latent Vector Interpolation

- 두 개의 latent codes를 보간(interpolation)하는 방법으로는 다음과 같은 것들이 있습니다.



Linear Interpolation (LERP)



Spherical Linear Interpolation (SLERP)

Perceptual Path Length

- 두 개의 latent codes를 보간(interpolation)할 때 얼마나 급격하게 (부드럽지 않게) 바뀌는지 체크합니다.
 - 지점 t 와 $t + \epsilon$ 사이에서의 VGG 특징(features)의 거리가 얼마나 먼지 계산할 수 있습니다.

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

$$Loss_W = E \left[\frac{1}{\epsilon^2} d \left(G(\underbrace{\text{lerp}(f(z_1), f(z_2); t))}_{w_1}, \underbrace{G(\text{lerp}(f(z_1), f(z_2); t + \epsilon))}_{w_2}) \right) \right]$$

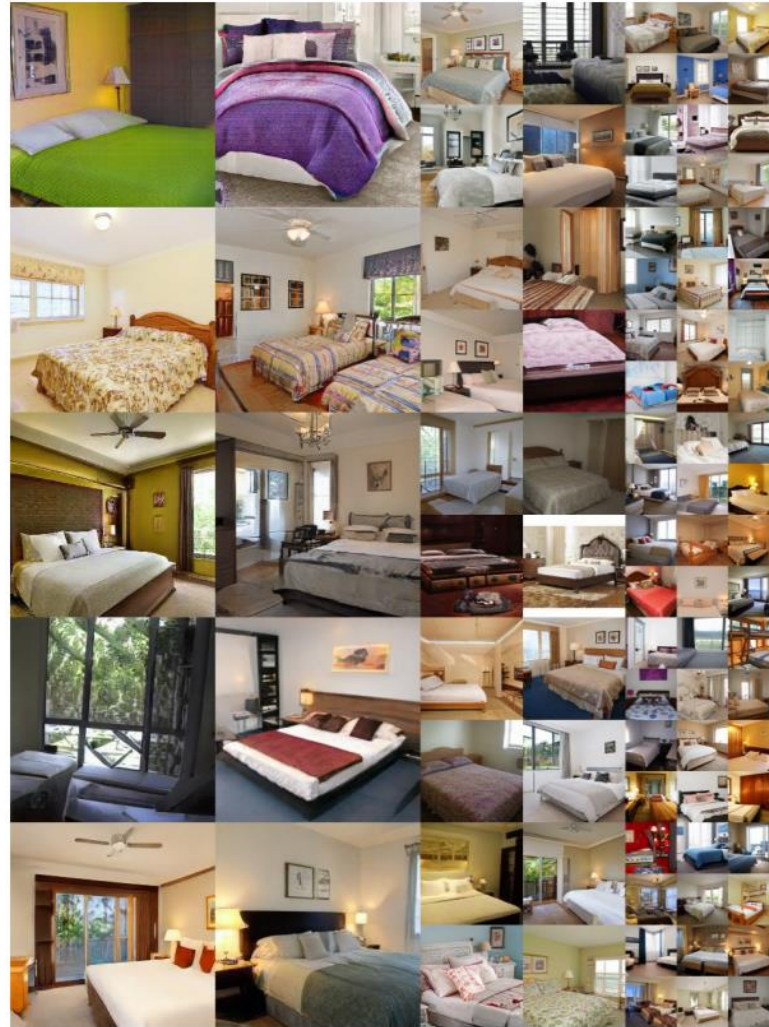
Linear Separability

- CelebA-HQ: 얼굴마다 성별(gender) 등의 40개의 **binary attributes**가 명시되어 있는 데이터셋
 - 이를 이용해 40개의 분류(classification) 모델을 학습합니다.
- 하나의 속성(attribute)마다 200,000개의 이미지를 생성하여 분류 모델에 넣습니다.
 - 이후에 confidence가 낮은 절반은 제거하여 100,000개의 레이블이 명시된 latent vector를 준비합니다.
 - 이렇게 준비된 100,000개의 데이터를 학습 데이터로 사용합니다.
- 매 attribute마다 **linear SVM 모델**을 학습합니다.
 - 이때 전통적인(traditional) GAN에서는 z , Style GAN에서는 w 를 이용합니다.
- 각 linear SVM 모델을 활용하여 다음의 값을 계산합니다. (i = 각 attribute의 인덱스)

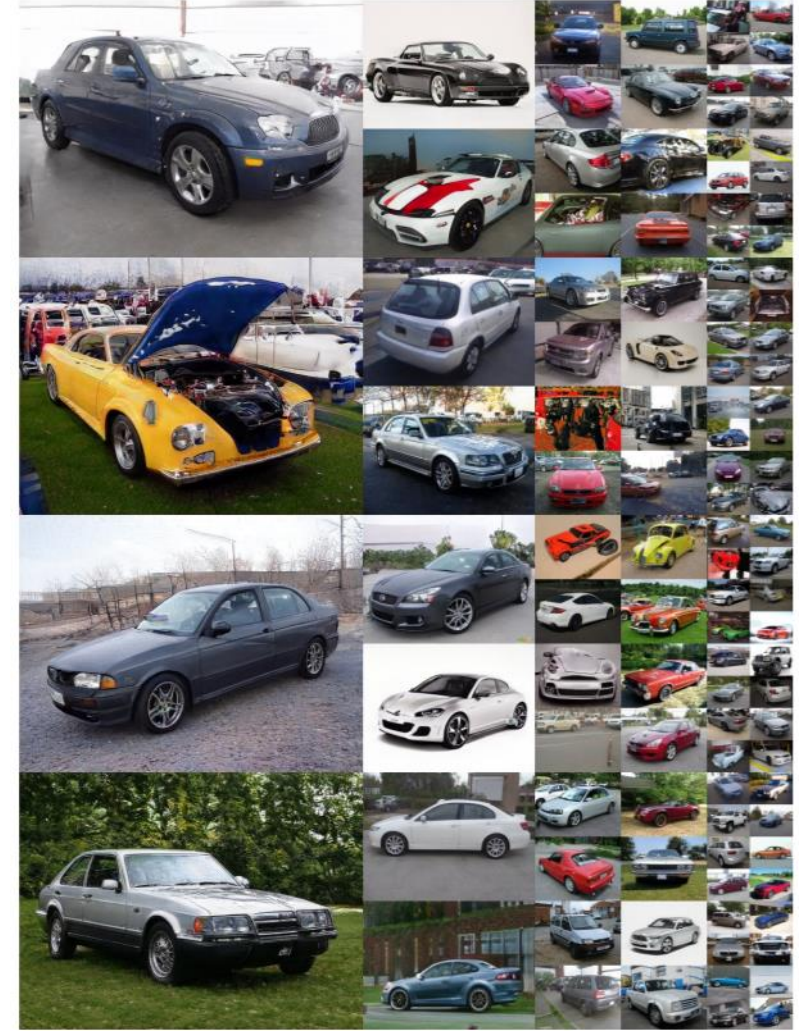
$$\exp \left(\sum_i H(Y_i | X_i) \right)$$

추가적인 실험 결과

- 동일한 세팅으로 추가 실험
 - LSUN Bedroom 데이터셋
 - LSUN Car 데이터셋
- Coarse styles 변화
 - 카메라 구도
- Middle styles 변화
 - 특정 가구
- Fine styles 변화
 - 세밀한 색상, 재질 등



LSUN Bedroom



LSUN Car