STYLEGAN

16011105 이재성

목차

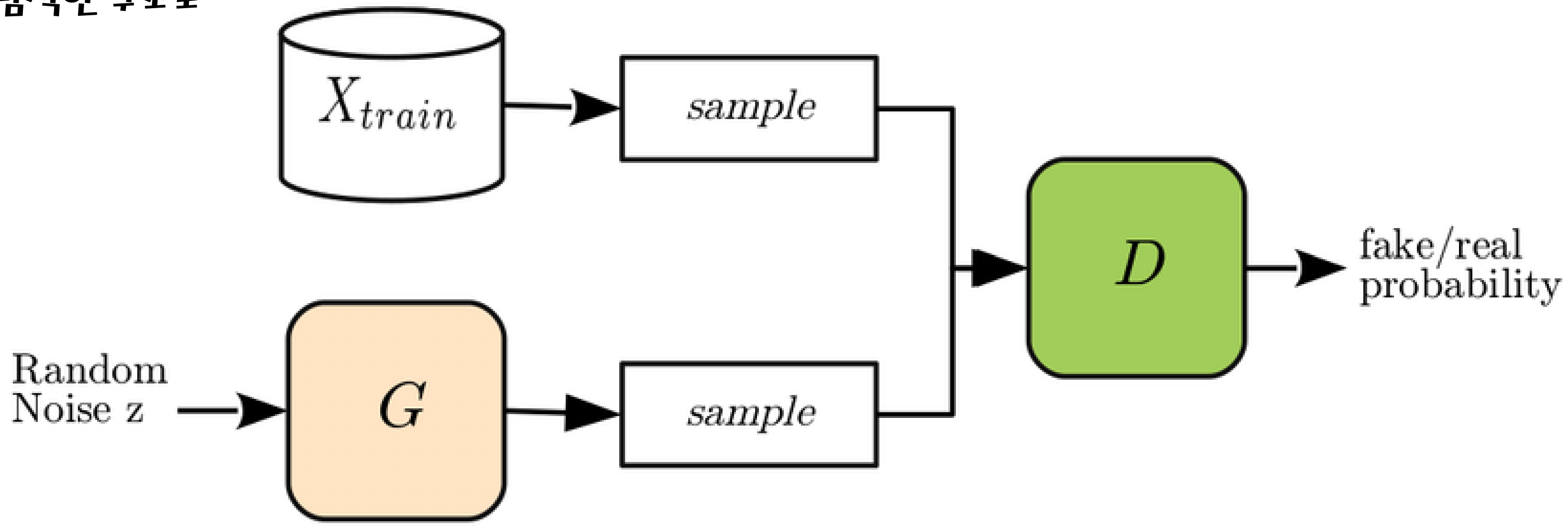
- 1. GAN의 기본 구조 (복습)
- 2. STYLEGAN 의 전신 PROGRESSIVE GAN
- 3. STYLEGAN

GAN 기본구조(복습)

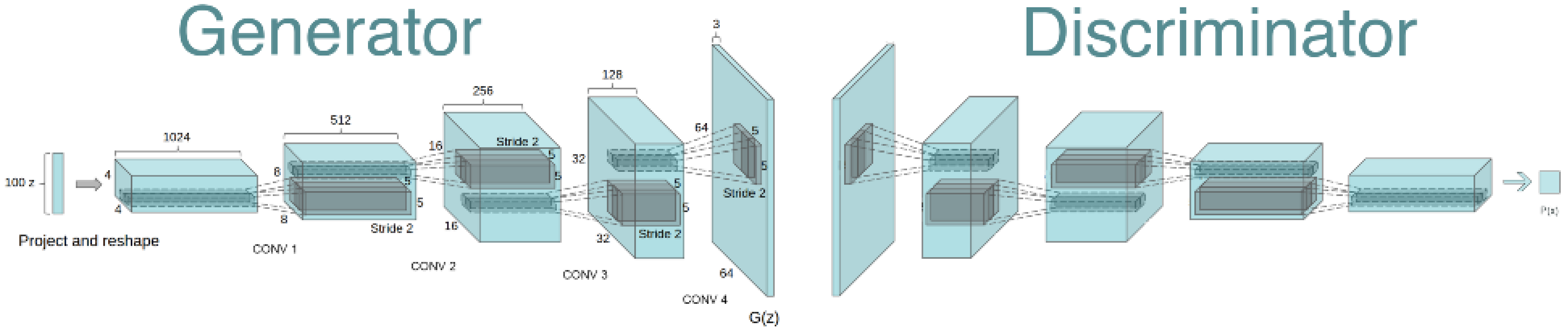
Generative
Adversarial
Network

(A) 생성기
→ Latent Space(잠재공간)를 기반으로 이미지를 생성함
→ 목표는 Discriminator를 속일 수 있는 정교한 이미지를 만들어 내는 것
(B) 분류기
→ 주어진 Real 데이터와 Fake 데이터를 기반으로 훈련
→ 목표는 진짜 이미지와 가짜 이미지를 구별하는 것

1. 개념적인 구조도

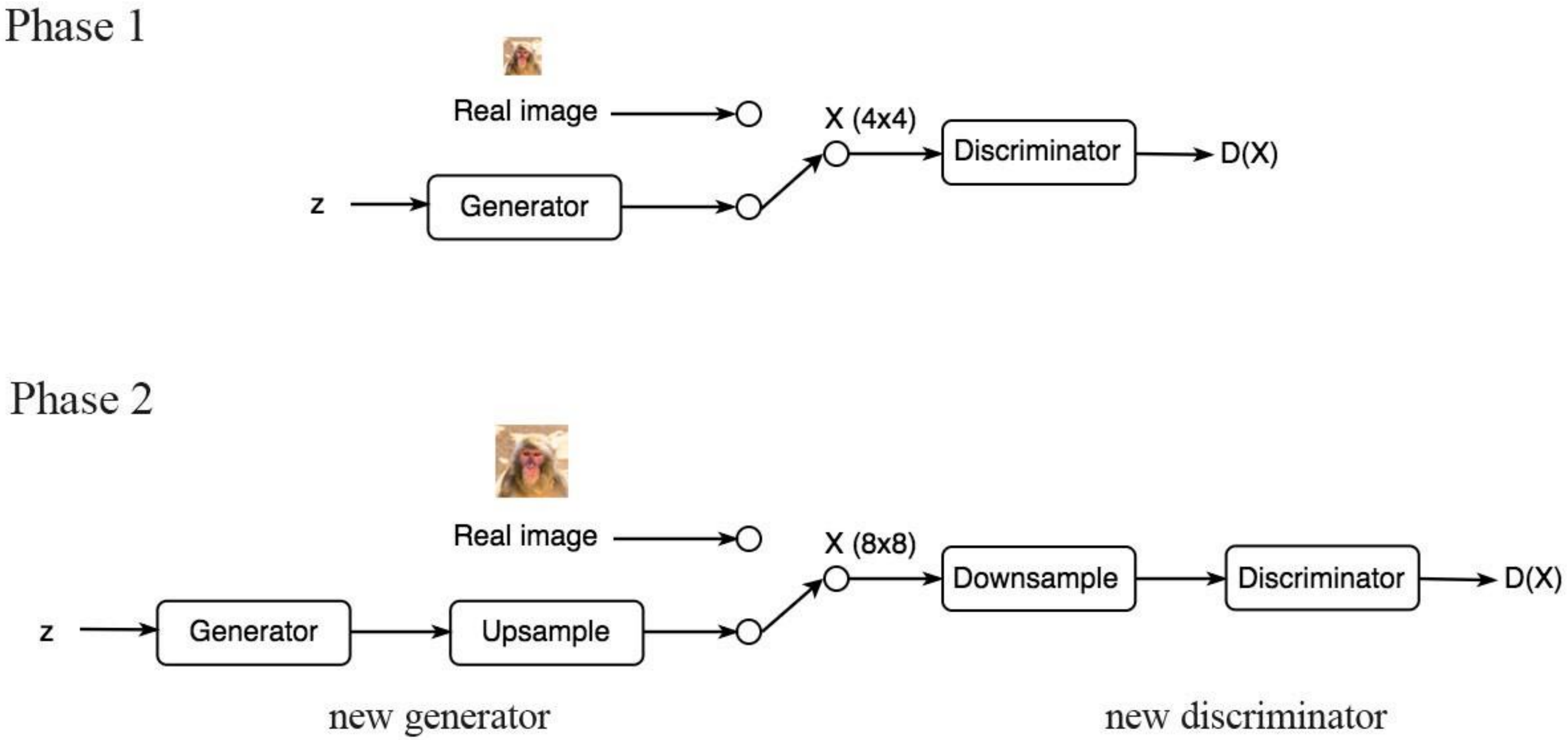


2. DCGAN의 구조도 예시



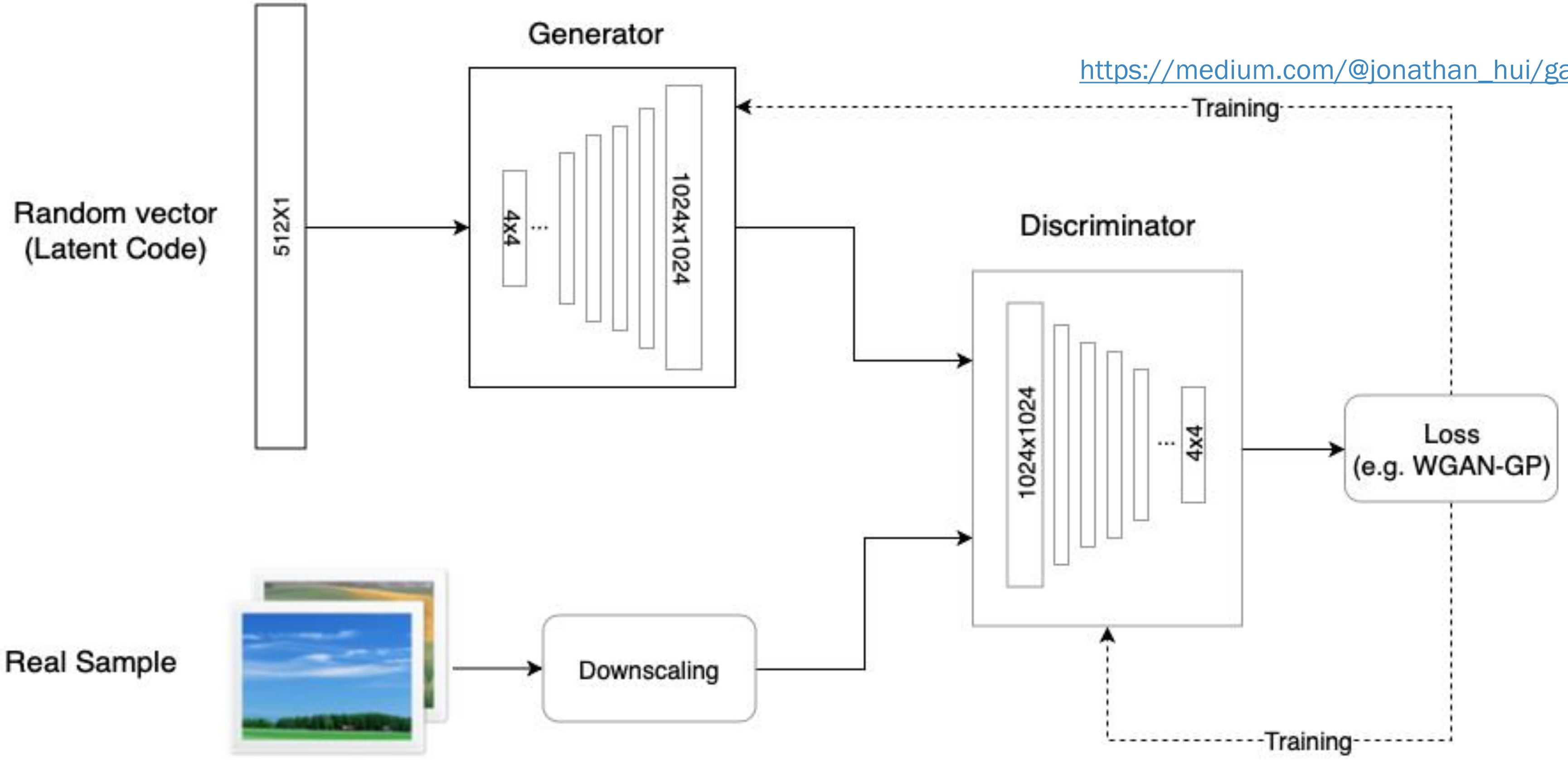
STYLEGAN 의 전신 PROGRESSIVE GAN

[HTTPS://ARXIV.ORG/PDF/1710.10196.PDF](https://arxiv.org/pdf/1710.10196.pdf)



PROGAN ARCHITECTURE

https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490



Wasserstein GAN with gradient penalty (WGAN-GP)

	Discriminator/Critic	**Generator**
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$
WGAN	$\boxed{\nabla_w \frac{1}{m} \sum_{i=1}^{m} \left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right) \right]}$	$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f\left(G\left(z^{(i)}\right)\right)$

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

PROGRESSIVE GROWING GAN

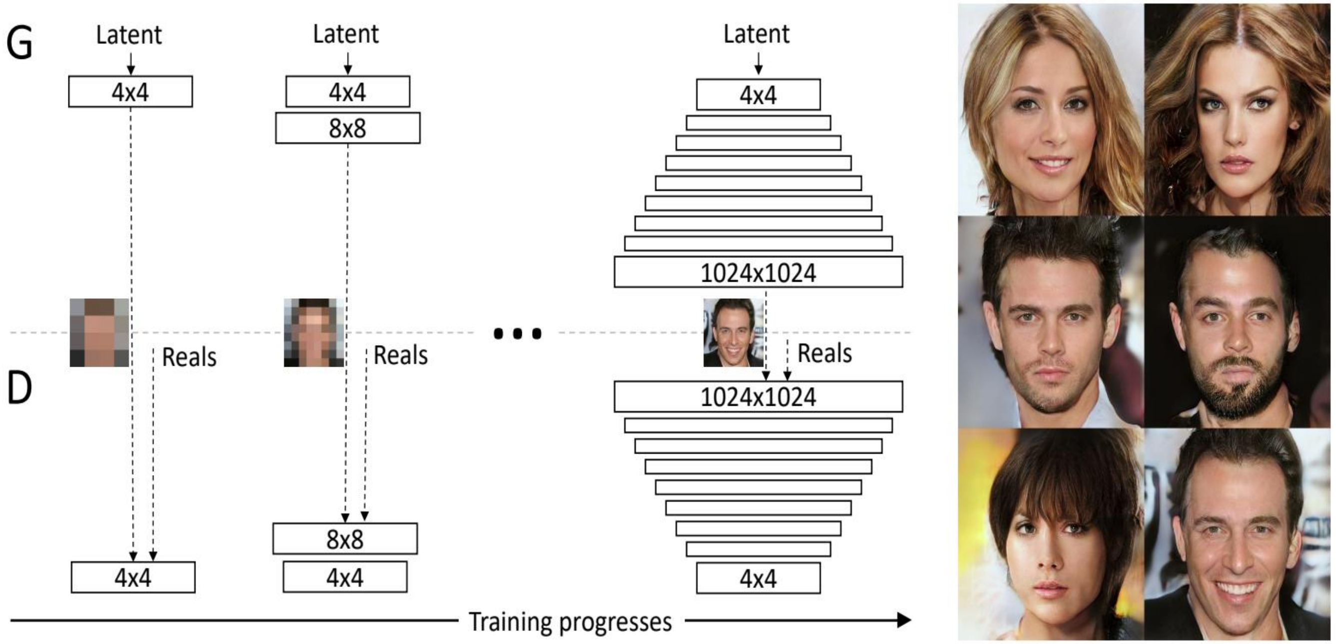
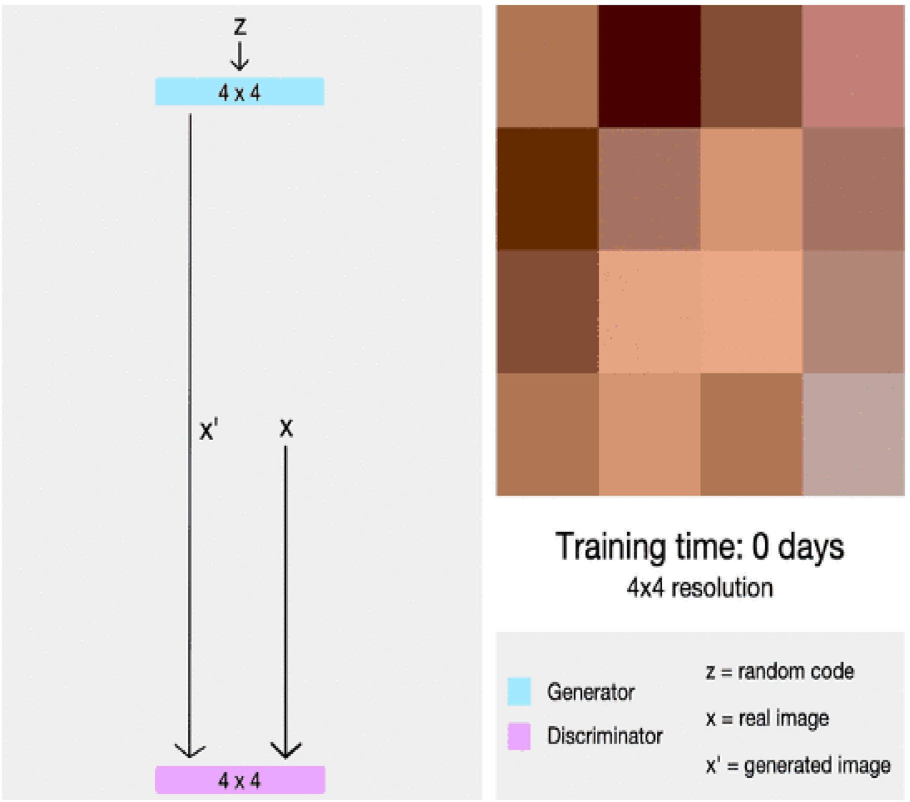


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $\boxed{N \times N}$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

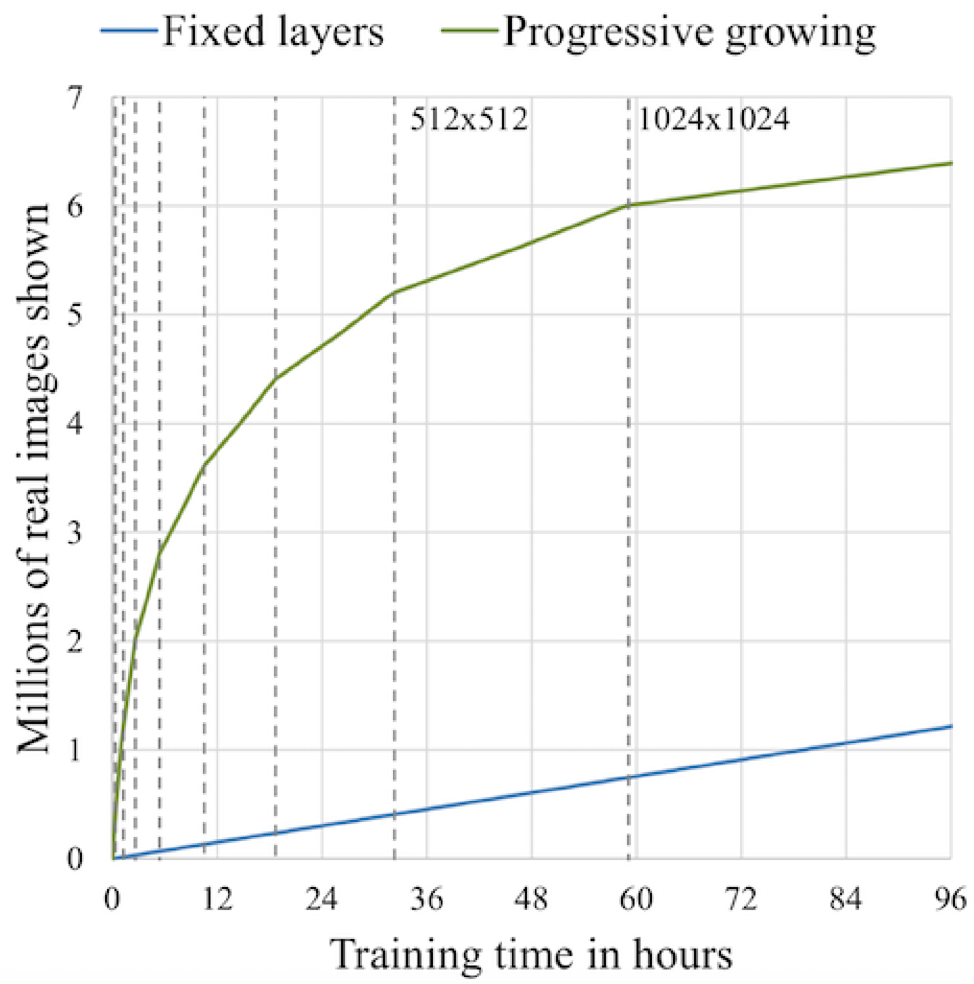


PROGRESSIVE GAN ARCHITECTURE

Generator	Act.	Output shape	Params
Latent vector	–	512 × 1 × 1	–
Conv 4 × 4	LReLU	512 × 4 × 4	4.2M
Conv 3 × 3	LReLU	512 × 4 × 4	2.4M
Upsample	–	512 × 8 × 8	–
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Upsample	–	512 × 16 × 16	–
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Upsample	–	512 × 32 × 32	–
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Upsample	–	512 × 64 × 64	–
Conv 3 × 3	LReLU	256 × 64 × 64	1.2M
Conv 3 × 3	LReLU	256 × 64 × 64	590k
Upsample	–	256 × 128 × 128	–
Conv 3 × 3	LReLU	128 × 128 × 128	295k
Conv 3 × 3	LReLU	128 × 128 × 128	148k
Upsample	–	128 × 256 × 256	–
Conv 3 × 3	LReLU	64 × 256 × 256	74k
Conv 3 × 3	LReLU	64 × 256 × 256	37k
Upsample	–	64 × 512 × 512	–
Conv 3 × 3	LReLU	32 × 512 × 512	18k
Conv 3 × 3	LReLU	32 × 512 × 512	9.2k
Upsample	–	32 × 1024 × 1024	–
Conv 3 × 3	LReLU	16 × 1024 × 1024	4.6k
Conv 3 × 3	LReLU	16 × 1024 × 1024	2.3k
Conv 1 × 1	linear	3 × 1024 × 1024	51
Total trainable parameters			**23.1M**

Discriminator	Act.	Output shape	Params
Input image	–	3 × 1024 × 1024	–
Conv 1 × 1	LReLU	16 × 1024 × 1024	64
Conv 3 × 3	LReLU	16 × 1024 × 1024	2.3k
Conv 3 × 3	LReLU	32 × 1024 × 1024	4.6k
Downsample	–	32 × 512 × 512	–
Conv 3 × 3	LReLU	32 × 512 × 512	9.2k
Conv 3 × 3	LReLU	64 × 512 × 512	18k
Downsample	–	64 × 256 × 256	–
Conv 3 × 3	LReLU	64 × 256 × 256	37k
Conv 3 × 3	LReLU	128 × 256 × 256	74k
Downsample	–	128 × 128 × 128	–
Conv 3 × 3	LReLU	128 × 128 × 128	148k
Conv 3 × 3	LReLU	256 × 128 × 128	295k
Downsample	–	256 × 64 × 64	–
Conv 3 × 3	LReLU	256 × 64 × 64	590k
Conv 3 × 3	LReLU	512 × 64 × 64	1.2M
Downsample	–	512 × 32 × 32	–
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Downsample	–	512 × 16 × 16	–
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Downsample	–	512 × 8 × 8	–
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Downsample	–	512 × 4 × 4	–
Minibatch stddev	–	513 × 4 × 4	–
Conv 3 × 3	LReLU	512 × 4 × 4	2.4M
Conv 4 × 4	LReLU	512 × 1 × 1	4.2M
Fully-connected	linear	1 × 1 × 1	513
Total trainable parameters			**23.1M**

Table 2: Generator and discriminator that we use with CELEBA-HQ to generate 1024×1024 images.

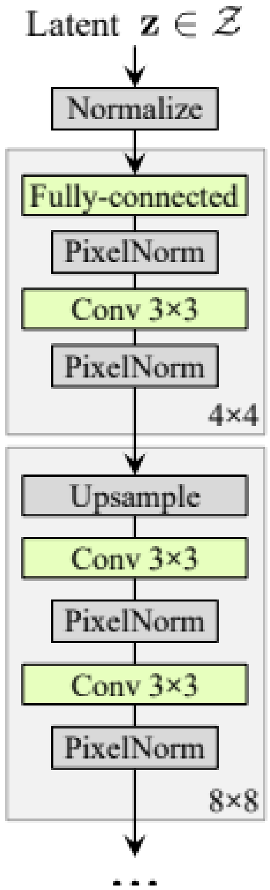


STYLEGAN (GENERATOR 부분만 바꿈)
[나머지 부분(LOSS, DISCRIMINATOR)등은 PROGAN과 동일]

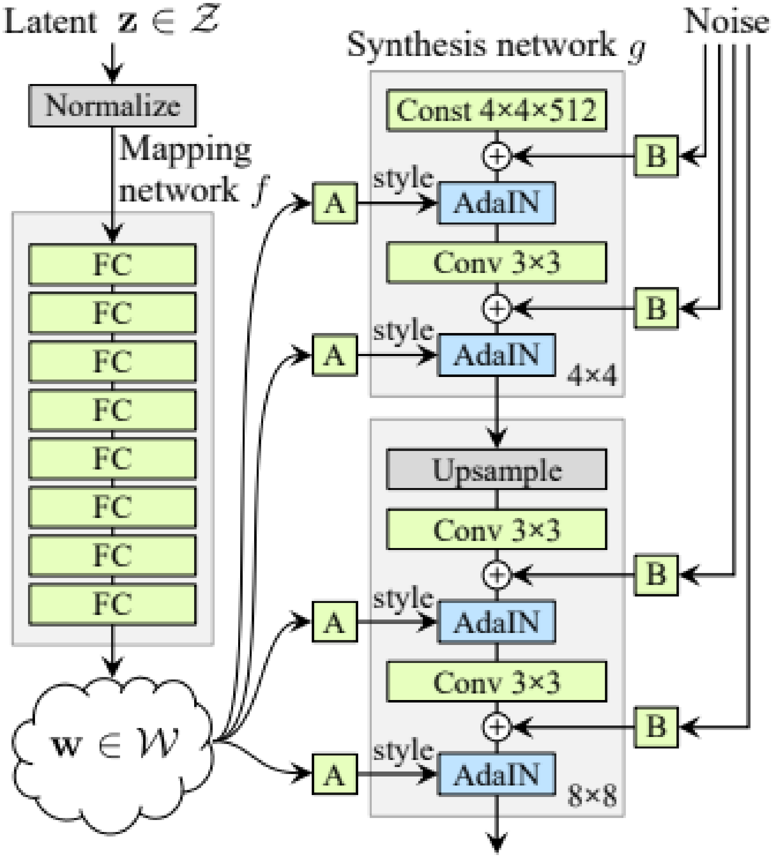
Traditional
→ PROGRESSIVE GAN(PROGAN)
장점→ scale specific한 style조절 가능
차이점:

(1) Latent Space에서 Mapping 함수f의 유무성
(2) 이미지 생성시 Style을 입히기 위한 AdaIn
(3) Stochastic Variation을 위한 Noise Injection

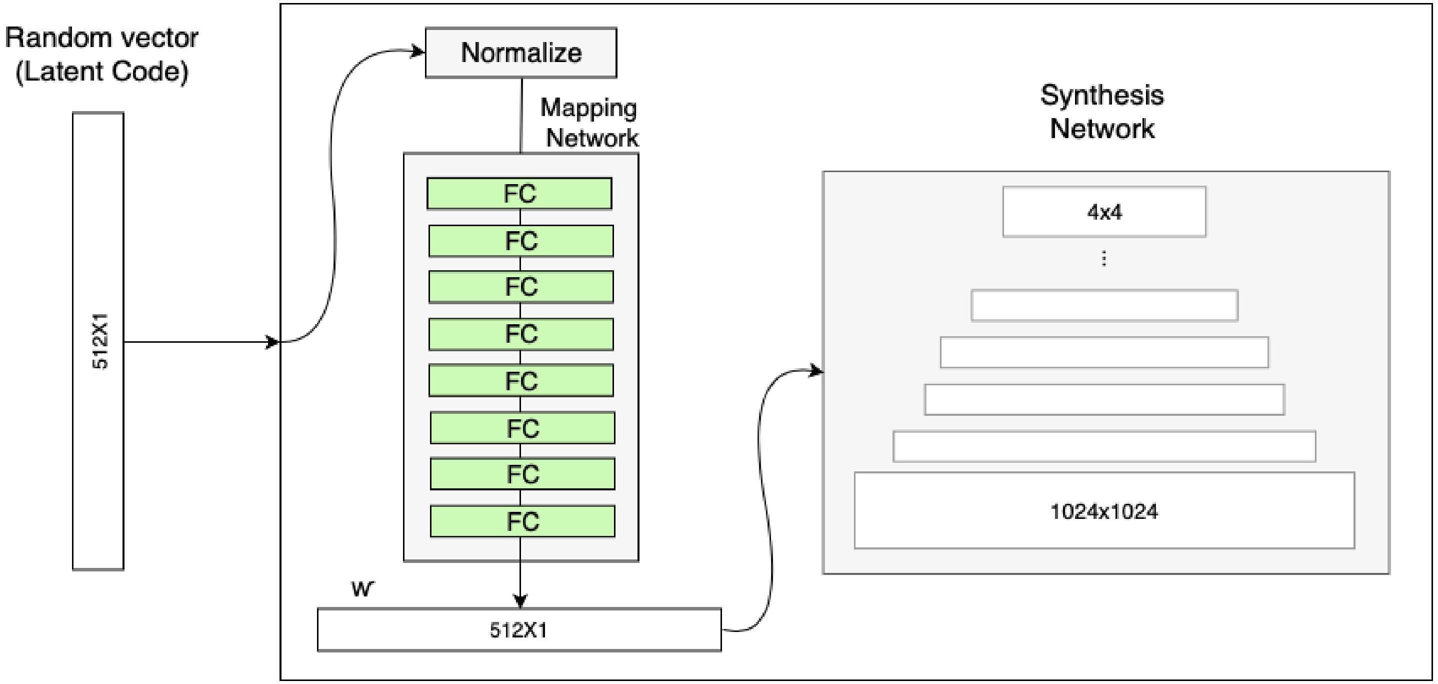
Traditional

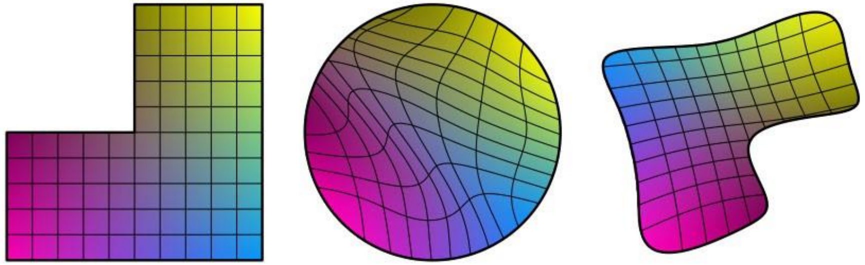


StyleGAN



(1)STYLEGAN MAPPING NETWORK





(a) Distribution of features in training set

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

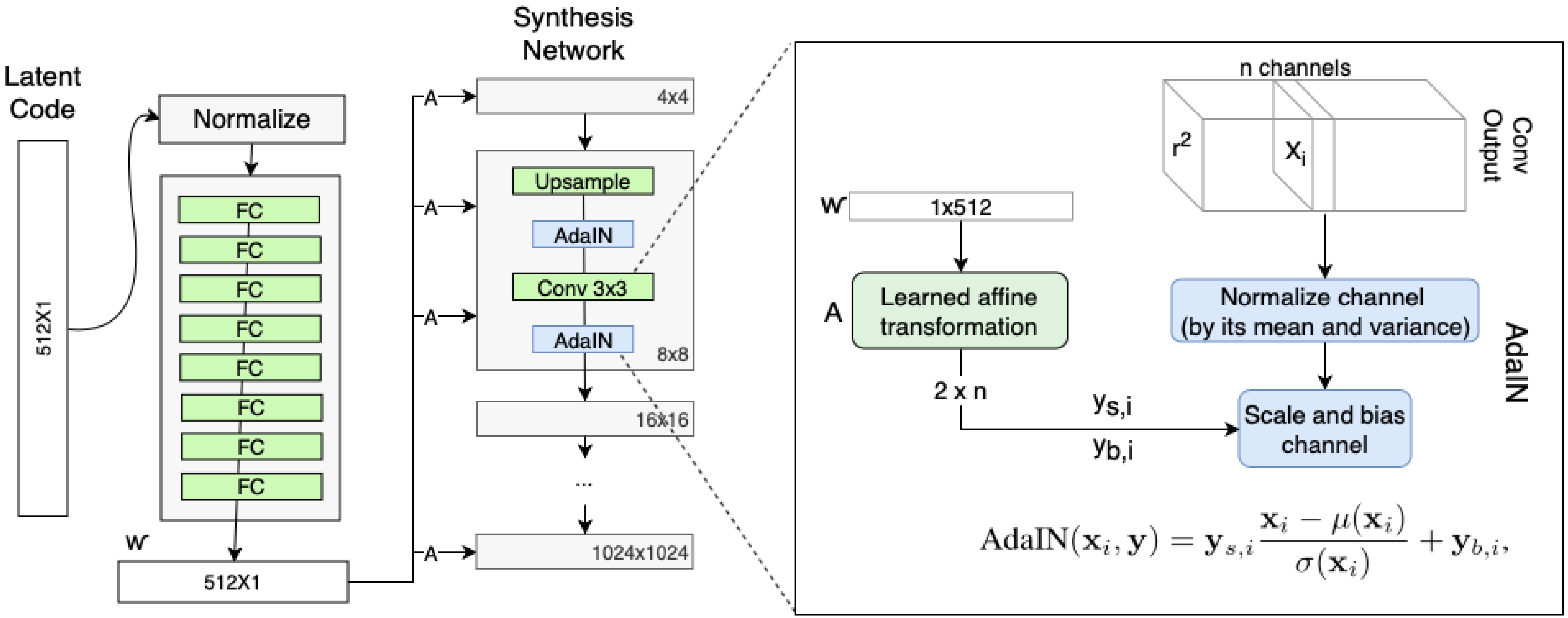
(c) Mapping from $\mathcal{W}$ to features

Feature Entanglement:
Train set에 많은 비중을 차지하는 feature에 많이 매핑 되어서 모델이 새로운 입력을 feature로 매핑하는 capability가 부족해지는 현상

Mapping network 목표:
생성기에 의해 만들어지는 얽히지 않은 특징들을 만들고 훈련 데이터셋에 나타나지 않은 특징 혼합을 피하기 위하여

(2)ADAIN
→ ADAPTIVE INSTANCE NORMALIZATION

Z → W 로 매핑한 후 A라는 affine 변환을 이용하여 style을 입힘



feature map i

the mean & standard deviation of the feature map i

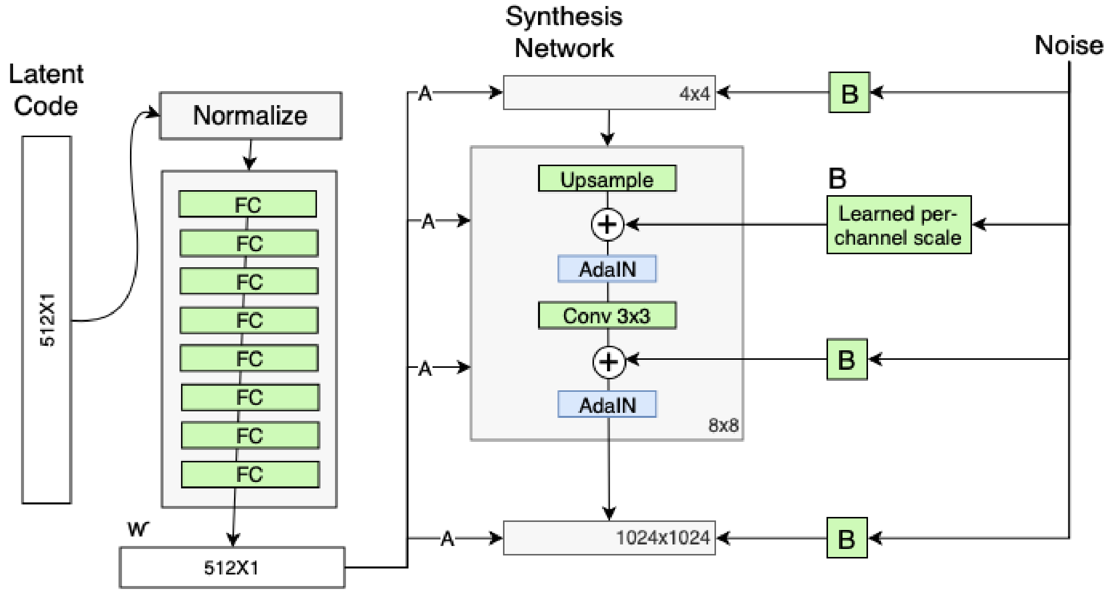
$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \underbrace{\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)}} + \mathbf{y}_{b,i}$$

normalize the feature map value
(instance normalization)

CycleGAN에서 사용한 Instance Normalization
이후에 y scale, bias를 곱하고 더하고

(3) STOCHASTIC VARIATION 을 위한 NOISE INJECTION(PER PIXEL)

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

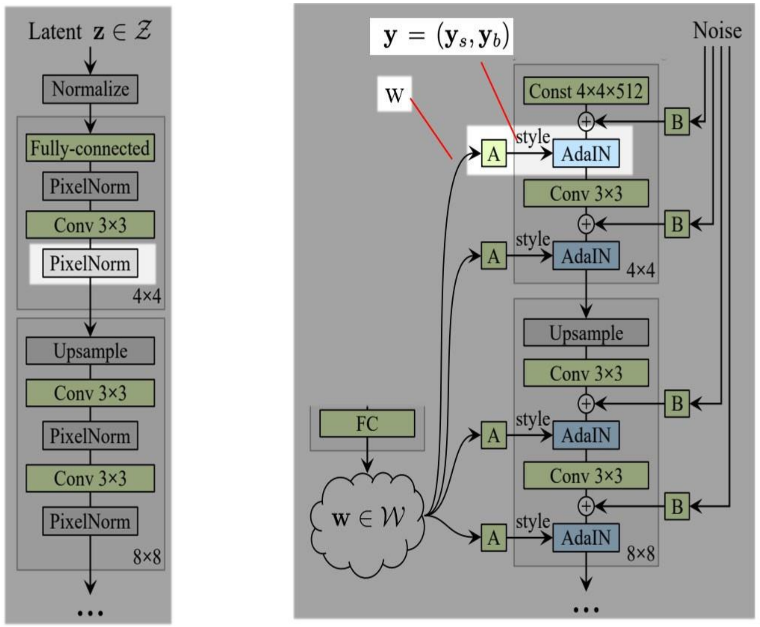


논문 내의 수치

FID 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**

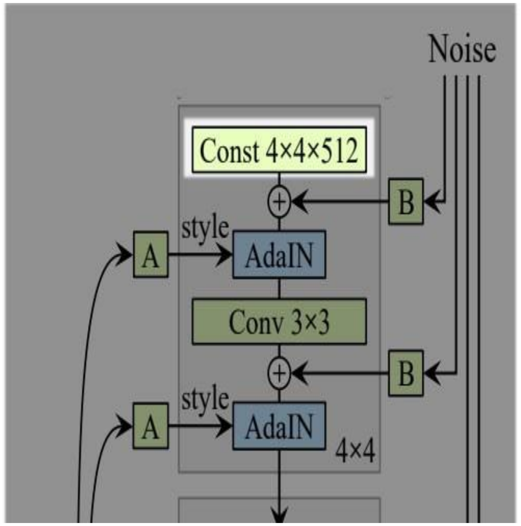
A. PROGAN 그대로
B. 원래 Nearest Neighbor 작업(up/down)을 Bilinear Interpolation으로 교체시
C. PixelNorm 작업을 AdaIN으로 교체
D. 첫 Layer를 4X4X512 고정 크기 learned constant 매트릭스로 교체(경험적으로 입력 layer에 Variable을 넣는 이점이 없다는 것을 알기 때문에(z))
E. Stochastic Variation을 위한 Noise injection

C

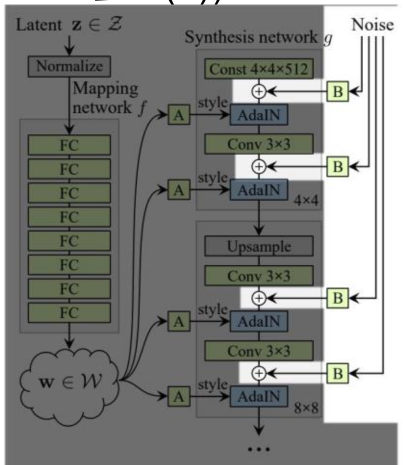


(a) Traditional
(b) Style-based generator

D

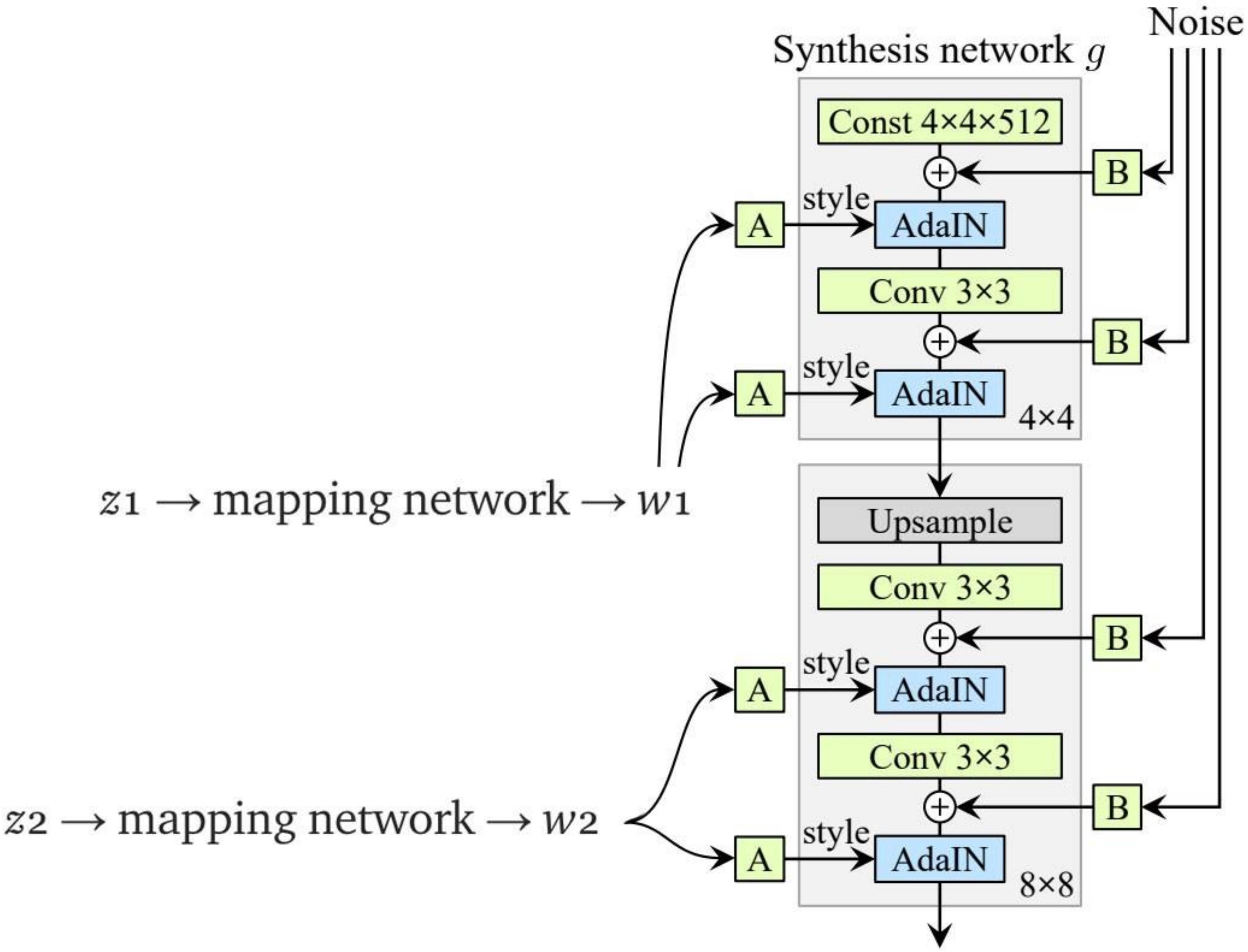


E



(b) Style-based generator

(F) MIXING REGULARIZATION

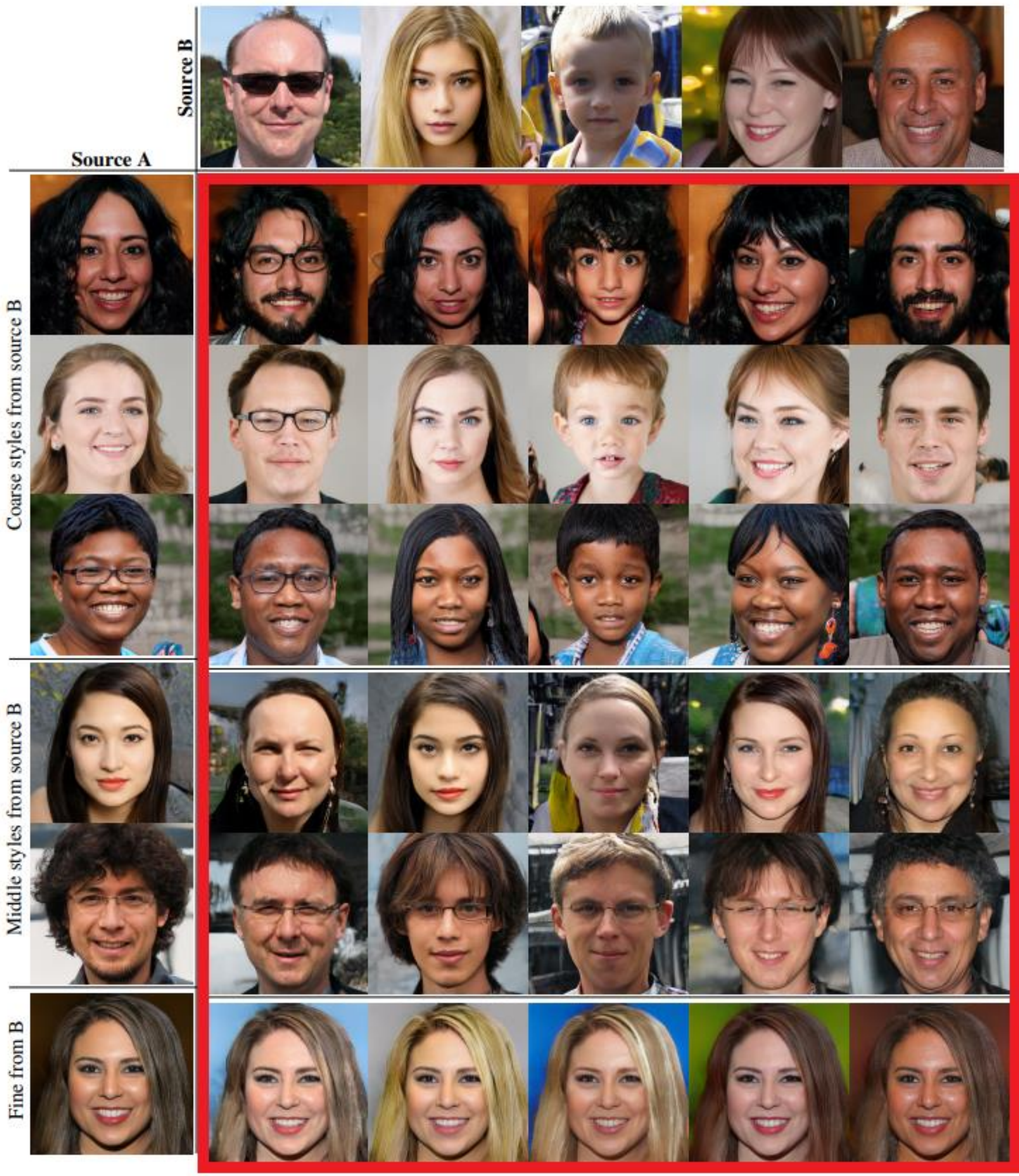


MIXING
COARSE (4^2 – 8^2)
MIDDLE (16^2- 32^2)
FINE (64^2-1024^2) STYLE

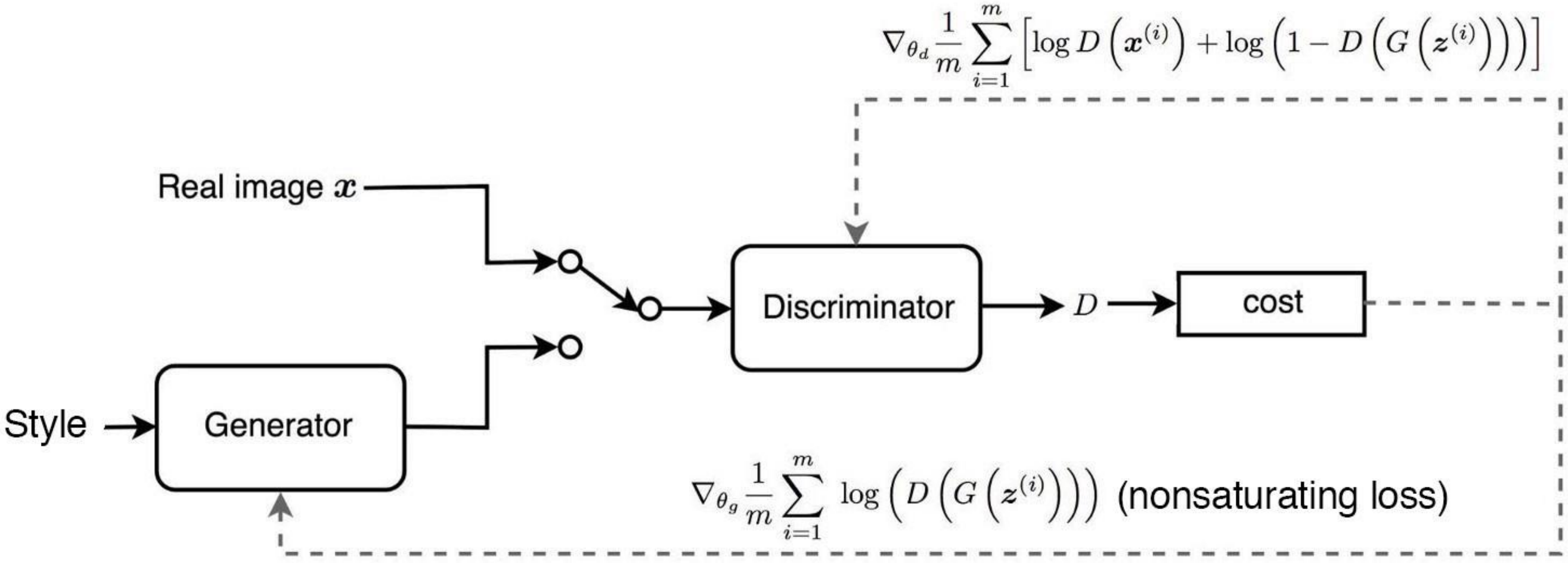
Coarse → hair, face shape, eyeglasses

Middle → facial Feature

Fine → color of eye, etc



TRAINING



$$R_1(\psi) := \frac{\gamma}{2} \mathrm{E}_{p_{\mathcal{D}}(x)} \left[\|\nabla D_\psi(x)\|^2\right]$$ (Regularization)

FFHQ(Flickr-Faces-HQ)를 훈련할 때 사용한 방식

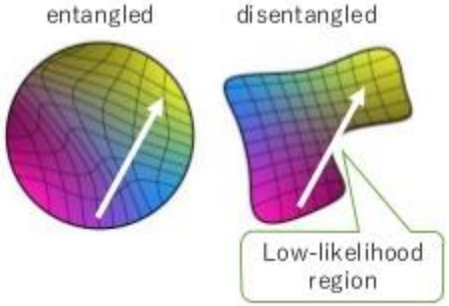
CelebA-HQ와 달리 WGAN-GP가 아닌 GAN LOSS+ Regularization 사용

논문 내에서 FEATURE DISENTANGLEMENT를 측정하기 위한 방법 제시

Measure of Entanglement

Perceptual path length (curvature)

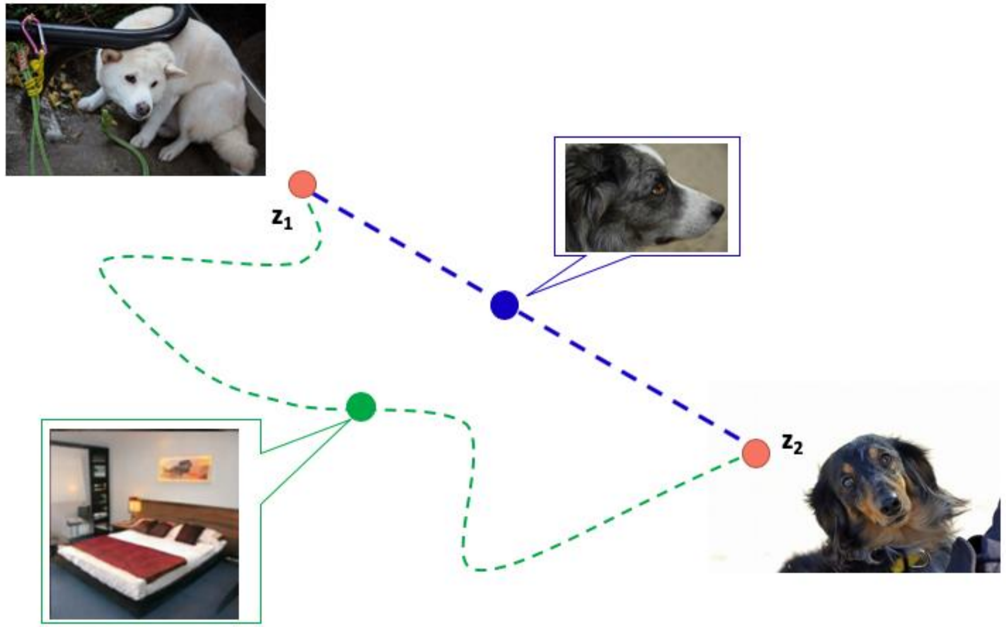
- If features are disentangled, interpolation between $\mathbf{z}_1$, $\mathbf{z}_2$ gives smooth image transition
- CNN features (e.g., VGG) can be used to calculate the perceptual distance

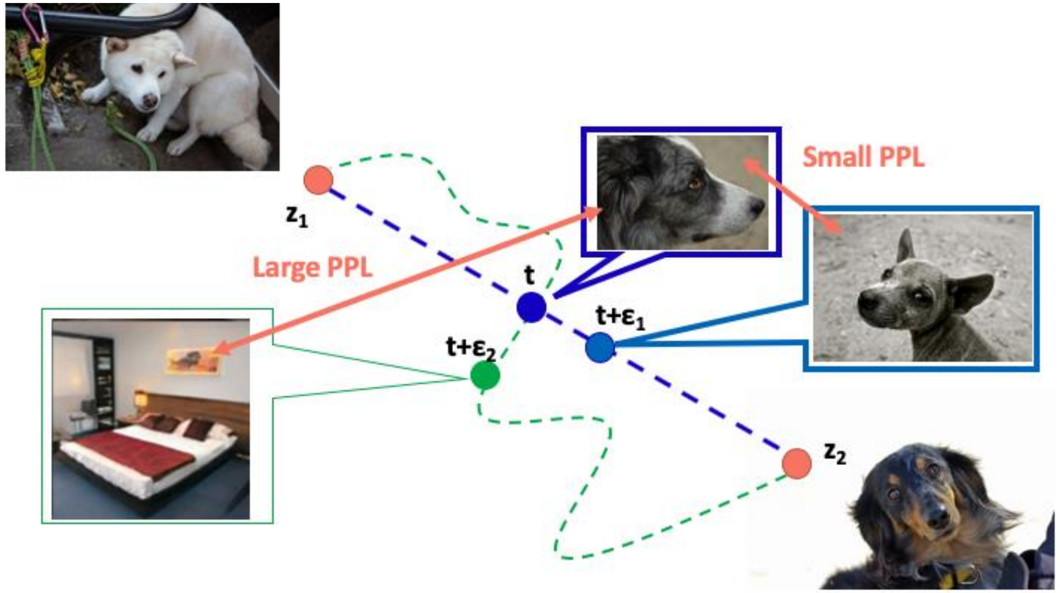


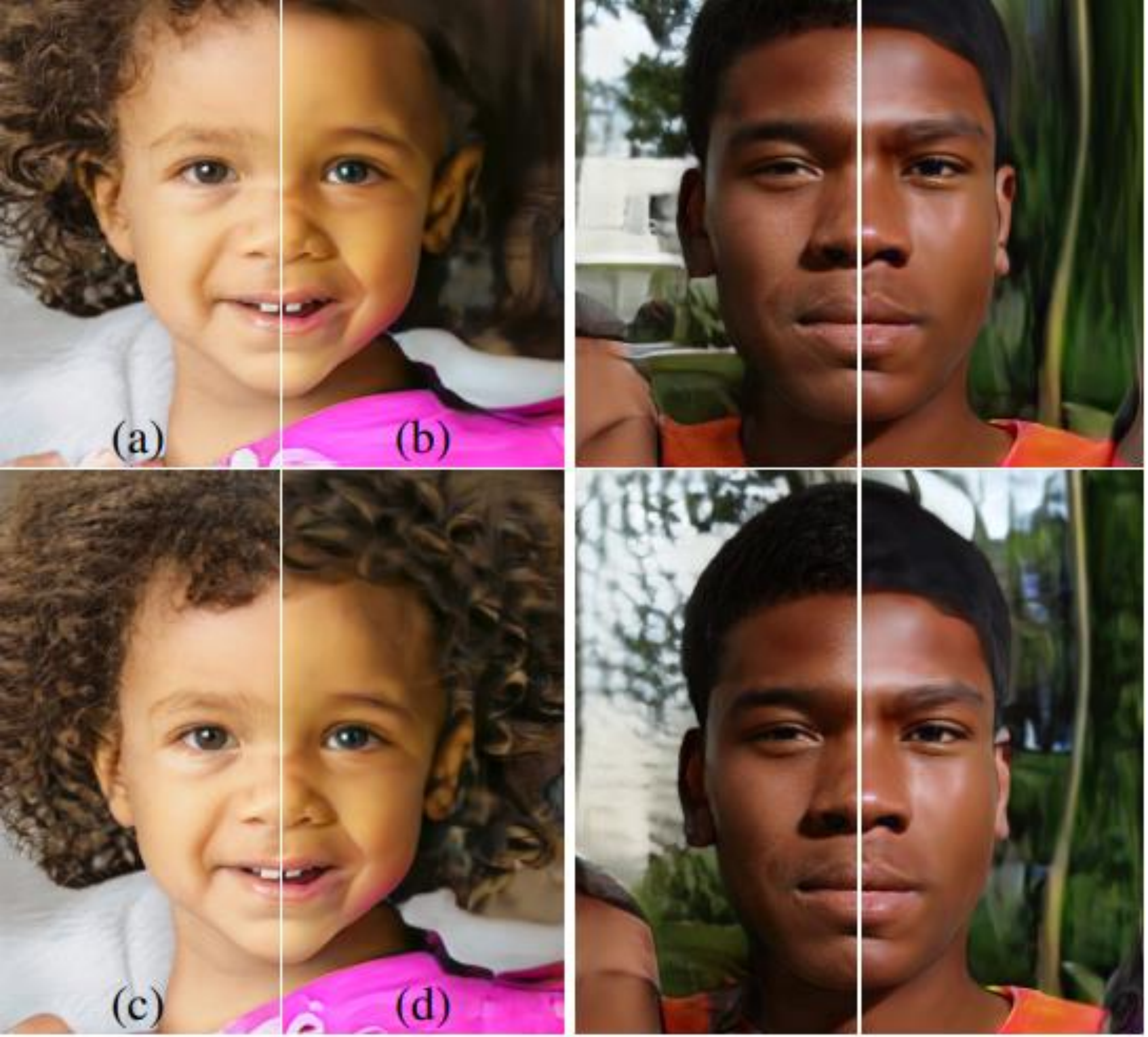
Linear separability

- SVM infers binary attributes from **z**
- Disentanglement makes classification easier

Method			Path length full	Path length end	Separa-bility
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



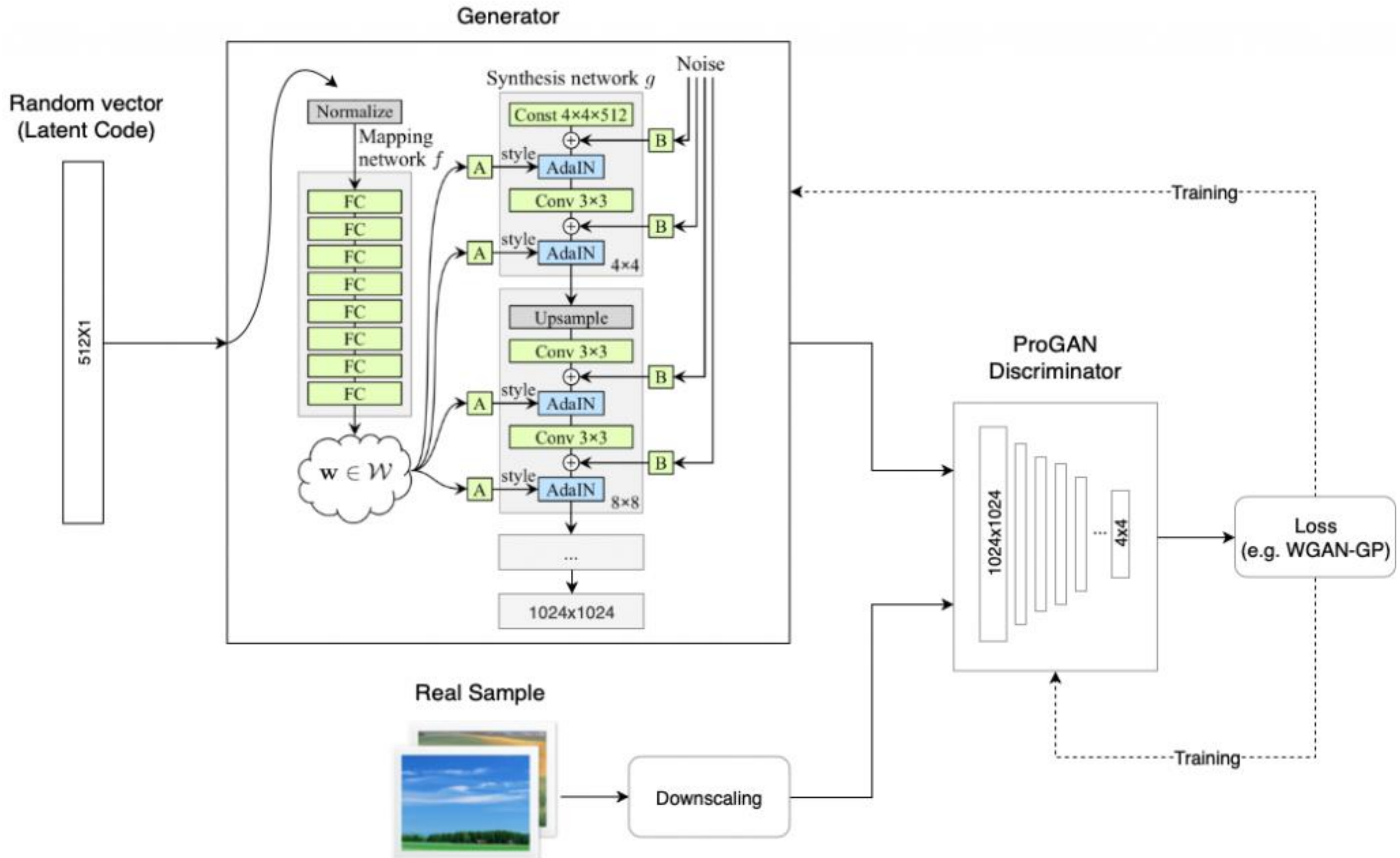




EFFECT OF NOISE

- (a) → noise applied to all layers
- (b) → no noise
- (c) → noise only in 64^2 - 1024^2
- (d) → noise only in 4^2 - 32^2

STYLEGAN OVERALL ARCHITECTURE 정리

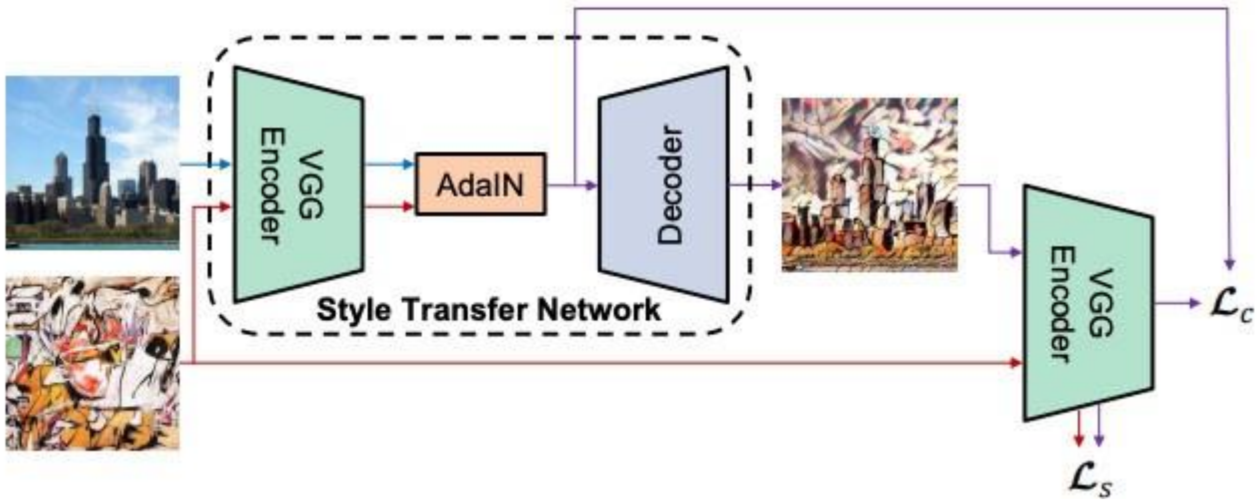


하지만 문제가 있었으니 → 이미지 중간에 ABNORMAL한 패턴이 나온다!

이러한 문제점을 없애기 위해

STYLEGANv2(A.K.A STYLEGAN2)에서는 이러한 문제점을 tracing하여 문제가 AdaIN 에서 사용하는 Instance Normalization에 의해 야기된다는 것을 알게 됨

BLOB 발생 이유
→ADAIN 자체가 STYLE TRANSFER에서 하나의 이미지 스타일을 다른 이미지로 바꾸는 것을 진행할 때 사용하는데 입력의 중요 내용들이 사라질 수 있기 때문

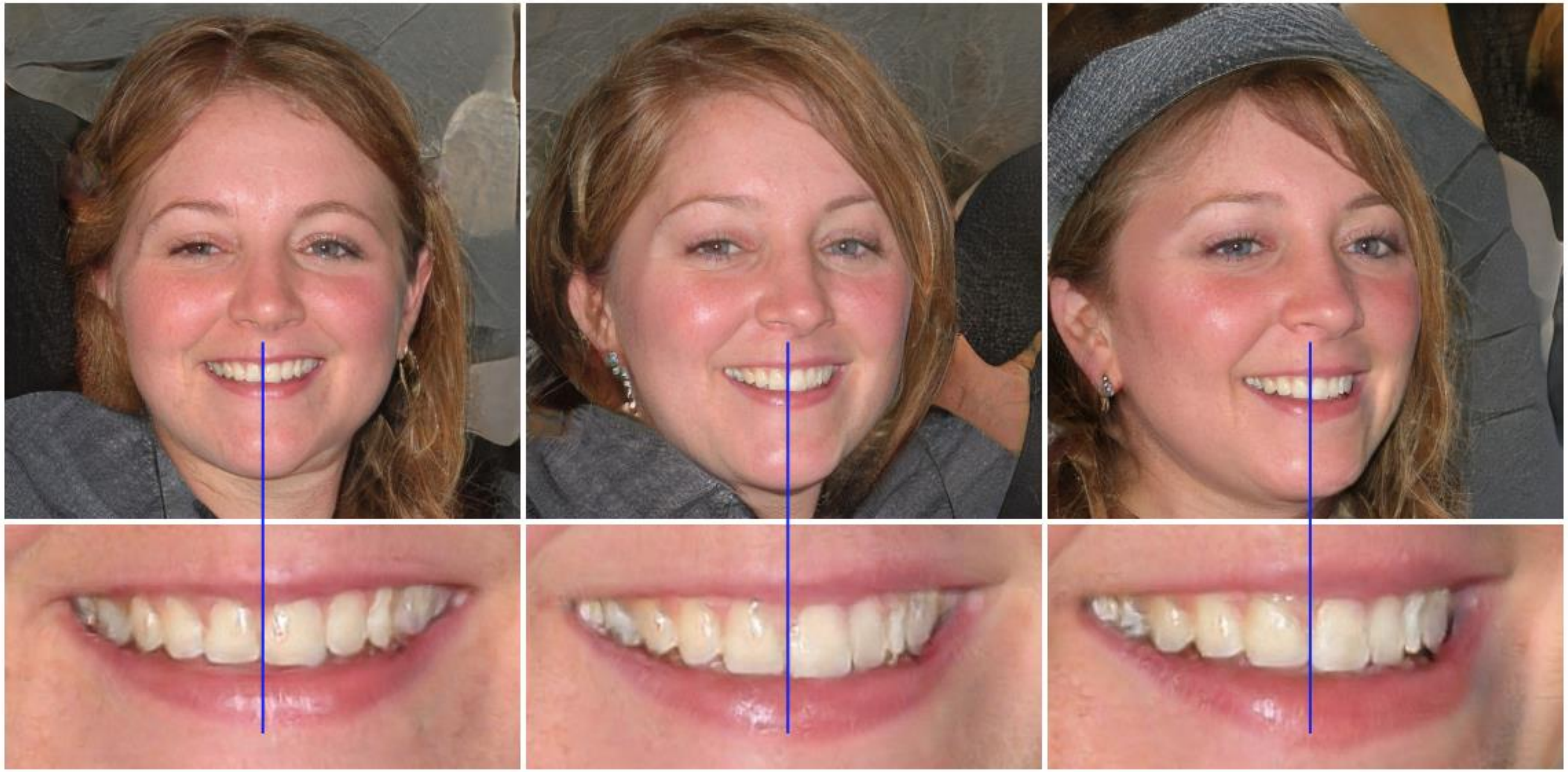


Style Transfer에서 발췌

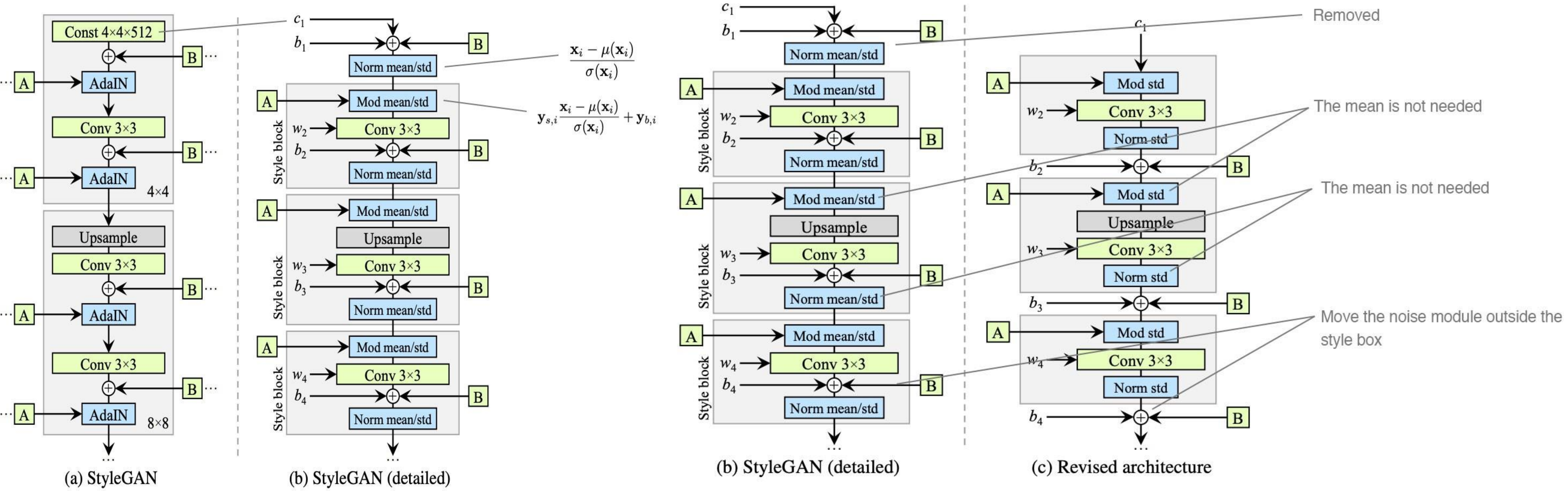
STYLEGAN2 논문 내용 발췌

> *We pinpoint the problem to the AdaIN operation that normalizes the mean and variance of each feature map separately, thereby potentially destroying any information found in the magnitudes of the features relative to each other. We hypothesize that the droplet artifact is a result of the generator intentionally sneaking signal strength information past instance normalization: by creating a strong, localized spike that dominates the statistics, the generator can effectively scale the signal as it likes elsewhere*

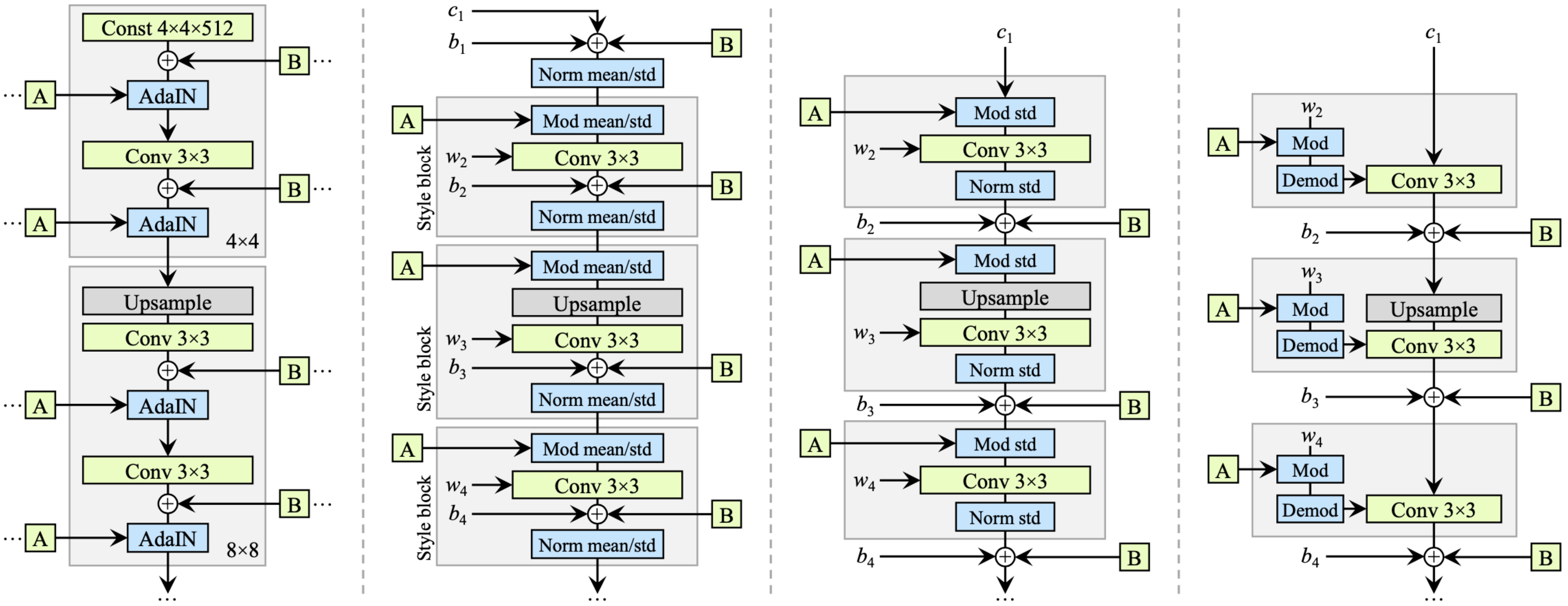
또다른 문제점 → 얼굴의 방향이 바뀌어도 이빨의 중심점이 변하지 않는 문제 (PROGAN에 의한 문제점)



이러한 문제점들을 보완하기 위한 STYLEGAN2



위 내용을 정리하면



(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

논문 내의 수치들(STYLEGAN2)

	Configuration	FFHQ, 1024×1024				LSUN Car, 512×384			
		FID	Path length	Precision	Recall	FID	Path length	Precision	Recall
A	Baseline StyleGAN [24]	4.40	195.9	**0.721**	0.399	3.27	1484.5	**0.701**	0.435
B	+ Weight demodulation	4.39	173.8	0.702	0.425	3.04	862.4	0.685	0.488
C	+ Lazy regularization	4.38	167.2	0.719	0.427	2.83	981.6	0.688	0.493
D	+ Path length regularization	4.34	139.2	0.715	0.418	3.43	651.2	0.697	0.452
E	+ No growing, new G & D arch.	3.31	**116.7**	0.705	0.449	3.19	471.2	0.690	0.454
F	+ Large networks	**2.84**	129.4	0.689	**0.492**	**2.32**	**415.5**	0.678	**0.514**

[HTTPS://THISXDOESNOTEXIST.COM/](https://thisxdoesnotexist.com/)

STYLEGAN을 이용한 THIS X DOES NOT EXIST APPLICATION

This Person Does Not Exist

The site that started it all, with the name that says it all. Created using a style-based generative adversarial network (StyleGAN), this website had the tech community buzzing with excitement and intrigue and inspired many more sites.

Created by Phillip Wang.

This Cat Does Not Exist

These purr-fect GAN-made cats will freshen your feeline-gs and make you wish you could reach through your screen and cuddle them. Once in a while the cats have visual deformities due to imperfections in the model – beware, they can cause nightmares.

Created by Ryan Hoover.

This Rental Does Not Exist

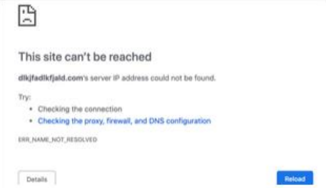
Why bother trying to look for the perfect home when you can create one instead? Just find a listing you like, buy some land, build it, and then enjoy the rest of your life.

Created by Christopher Schmidt.

This Waifu Does Not Exist

This one is a little stranger than the others, but I bet you'll read more of the plot summary than you expect. It's engrossing.

Created by Gwern Branwen.



This URL Does Not Exist

It doesn't, but at the same time it kind of does. Does that sound too cryptic? Click on it and you'll understand (and look at the URL). A simple but effective satire into the entire hype, but as far as I know doesn't utilize any form of deep learning.

Created by Preston Richey.

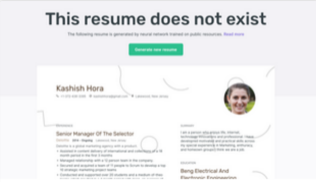
This Startup Does Not Exist

If your IP address is in San Francisco, this site actually just redirects to a random portfolio company from an AI-, blockchain-, or crypto-focused venture capital firm. Can you spot the difference? (Disclaimer: that was a joke).

This Question Does Not Exist

By generating seemingly realistic Stack Overflow questions, you can keep this open on your computer to seem like you're actually doing work. The most interesting aspect is the "Other Random Questions" section on the right side of the page, which shows other equally synthetic questions.

Created by Brad Dwyer.



This Resume Does Not Exist

To showcase their resume template builder, this company went as far as to incorporate TextgenRNN – not quite a GAN but rather a recurrent neural network (RNN) specializing in text generation. Bonus points – the headshots used in each resume are from thispersondoesnotexist.com.

Created by Enhancv.

This Emotion Does Not Exist

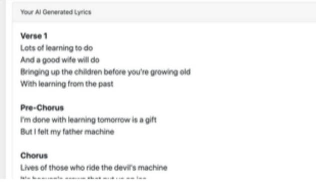
While it doesn't necessarily use Deep Learning, this site showcases the power of a relatively simple model's ability to interpret emotions from facial expressions. The results may shock you. Or delight you. Or anger you. Regardless, the site can identify the emotion.

Created by Jan Schneider.

This Vessel Does Not Exist

Drawing a beautiful parallel between Generators and Discriminators in GANs and apprentices and masters in ceramics, this site demonstrates the beautiful ability of neural networks to replicate the mastery of professionals at yet another craft.

Created by Derek Philip Au.



These Lyrics Do Not Exist

Now we can generate lyrics for a song given a theme or topic. If only we combined this with text-to-speech and a melody-generating model to create completely original songs. Then the music industry would take GANs seriously.

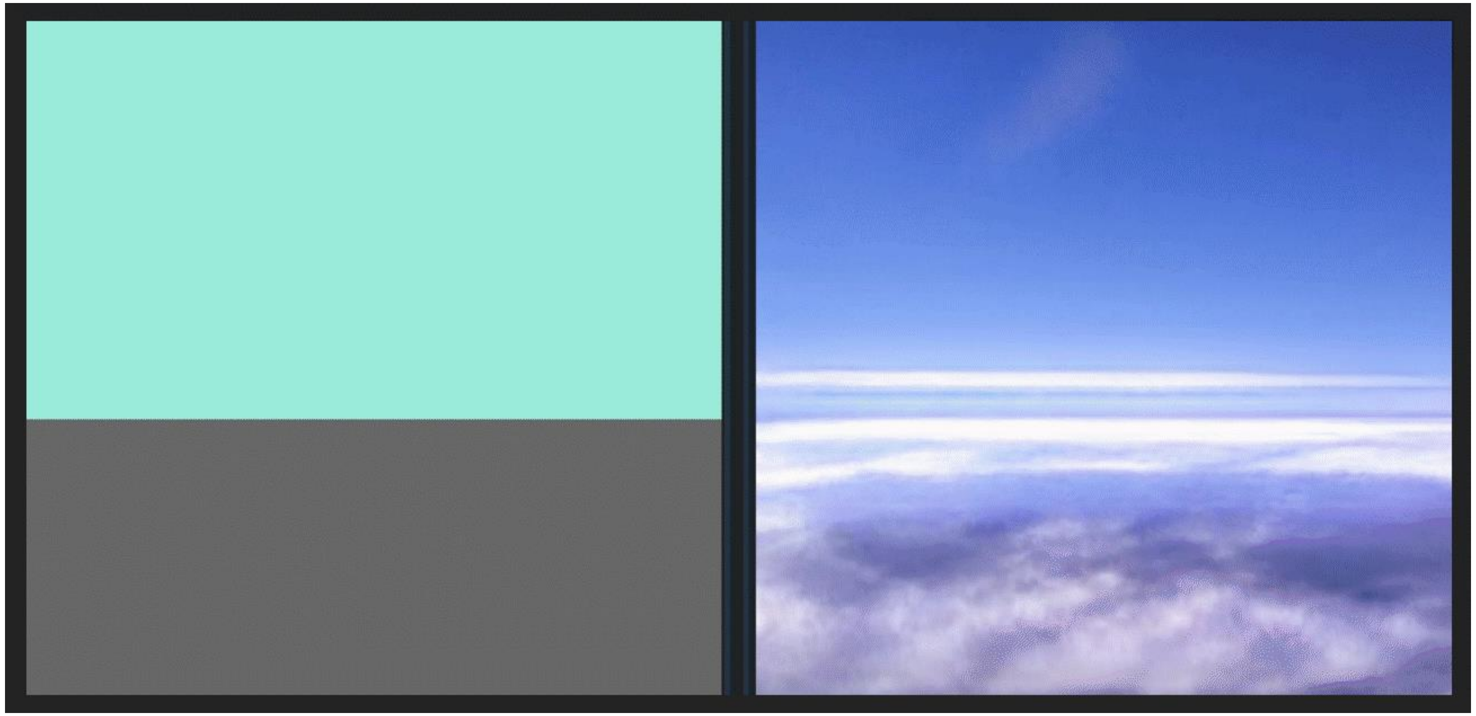
Created by Peter Ranieri.

This Snack Does Not Exist

For when you're feeling especially hungry and creative, this website will remind you that you're more hungry and less creative than you thought.

Created by Ariel Levi, Koby Ofek, et al.

STYLE TRANSFER 관련 다른 GAN



GauGAN(based on Spatial Adaptive-Normalization)

REFERENCE

- https://arxiv.org/abs/1812.04948
- https://www.youtube.com/watch?v=TWzEbMrH59o
- https://www.youtube.com/watch?v=2W1e7u41HvU
- https://www.youtube.com/watch?v=AQBti_wN414 (Stylegan)
- https://www.youtube.com/watch?v=u8qPvzk0AfY (Styleganv2)
- https://medium.com/@jonathan_hui/gan-stylegan-stylegan2-479bdf256299
- https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490