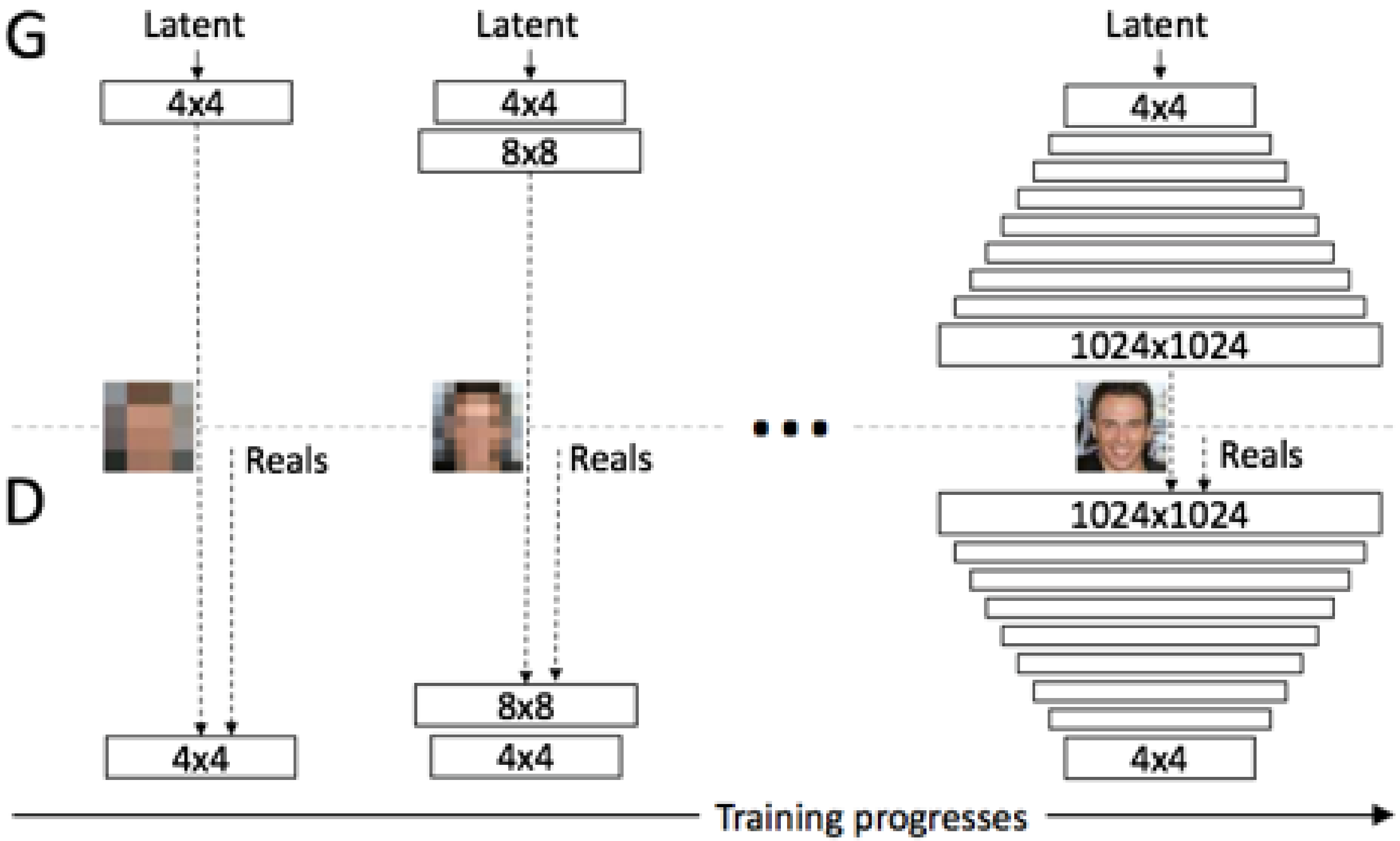
Style Generative Adversarial Networks

진짜 같은 고화질 가짜 이미지 생성하기 (StyleGAN v1)

SMARCLE 신도현

Contents

What is StyleGAN ?



Baseline: PGGAN

- Progressive Growing

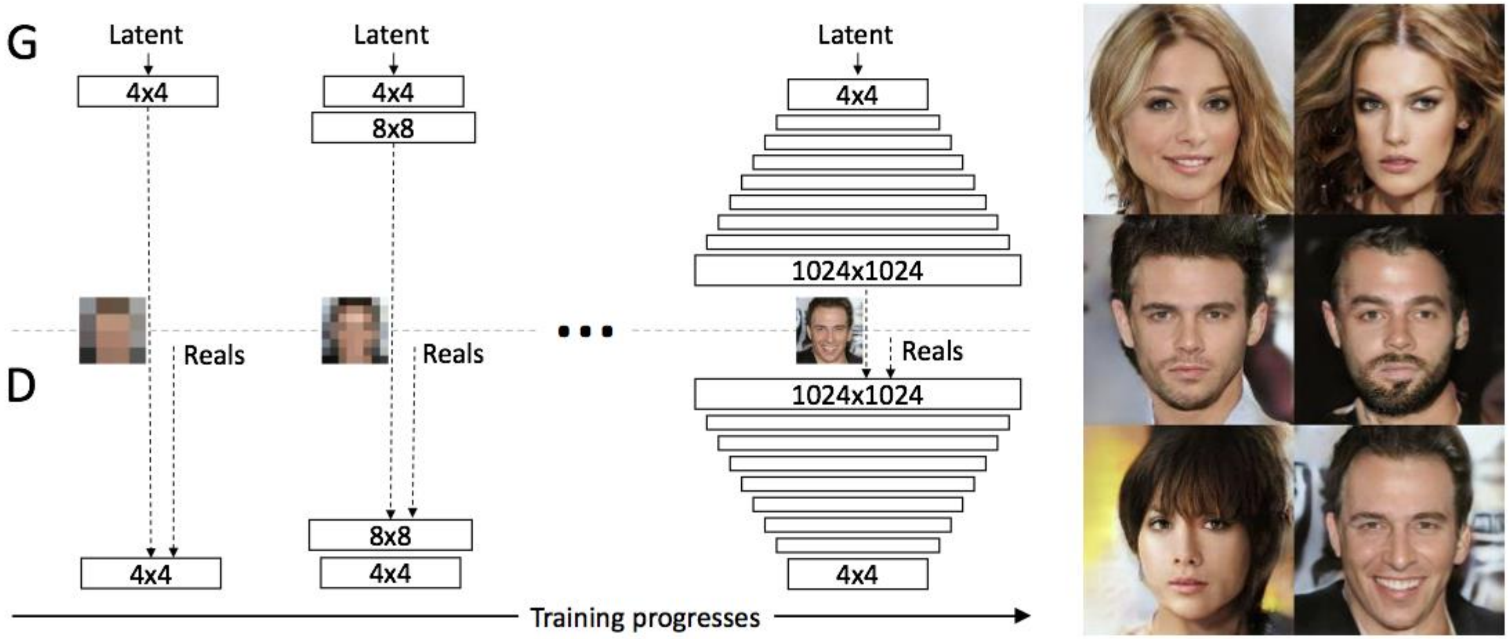
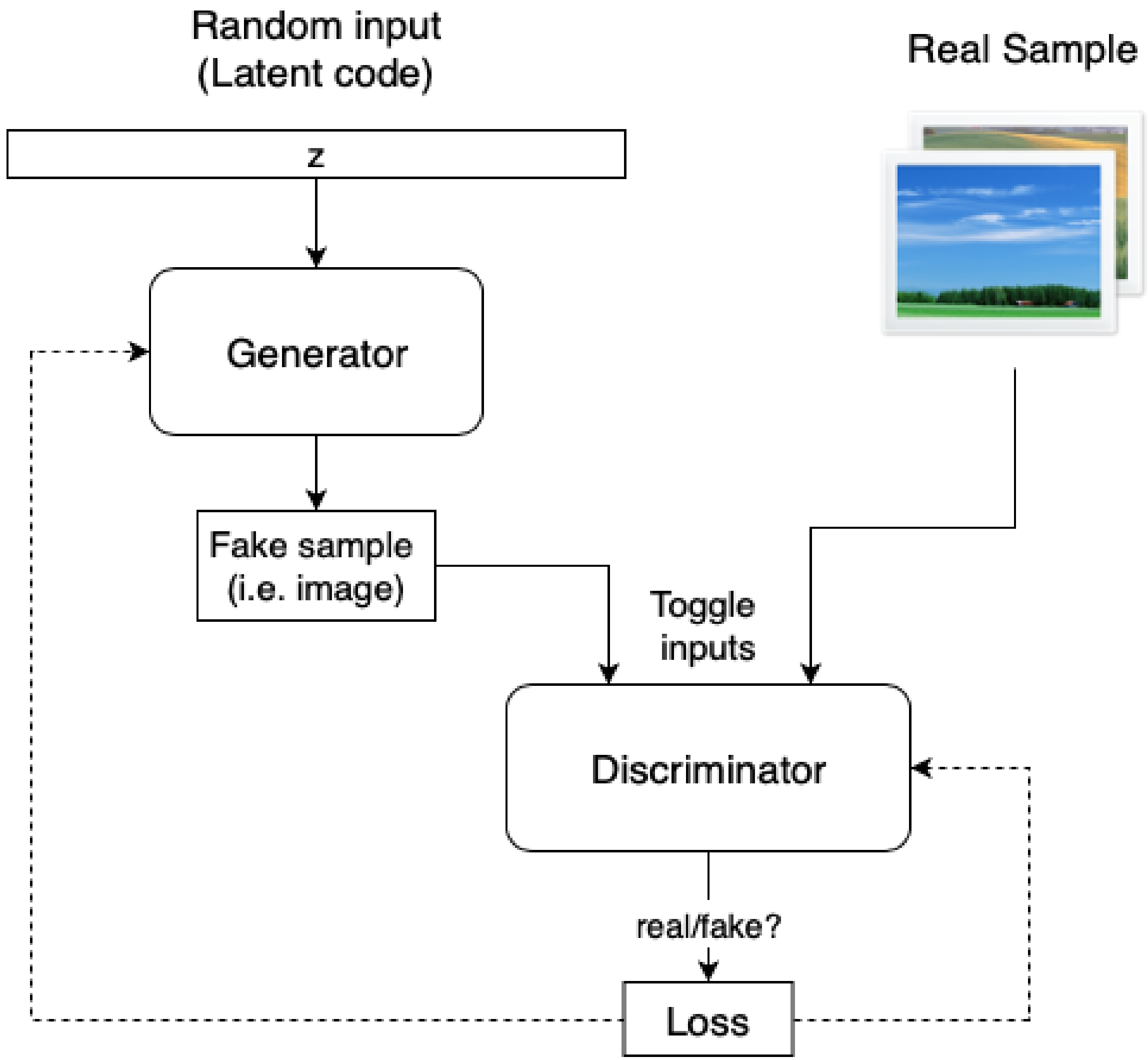
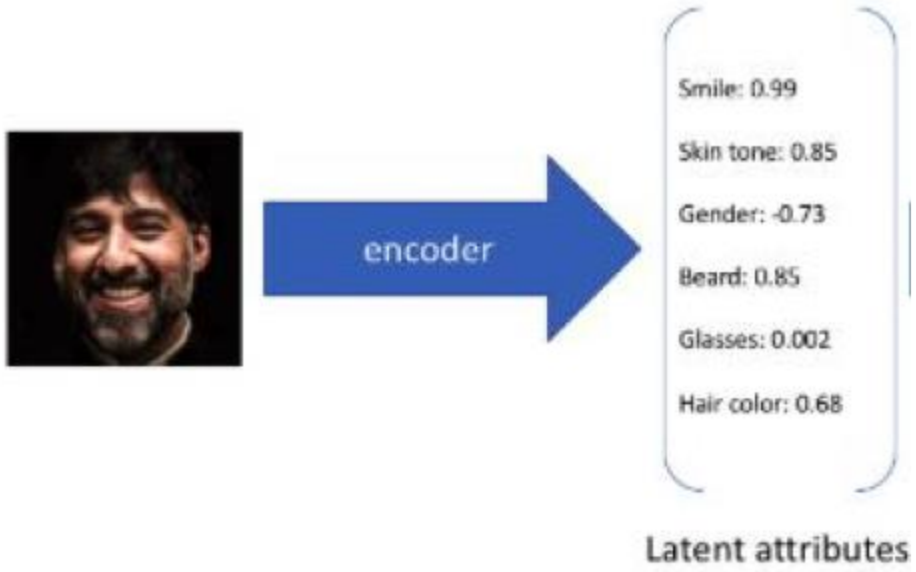
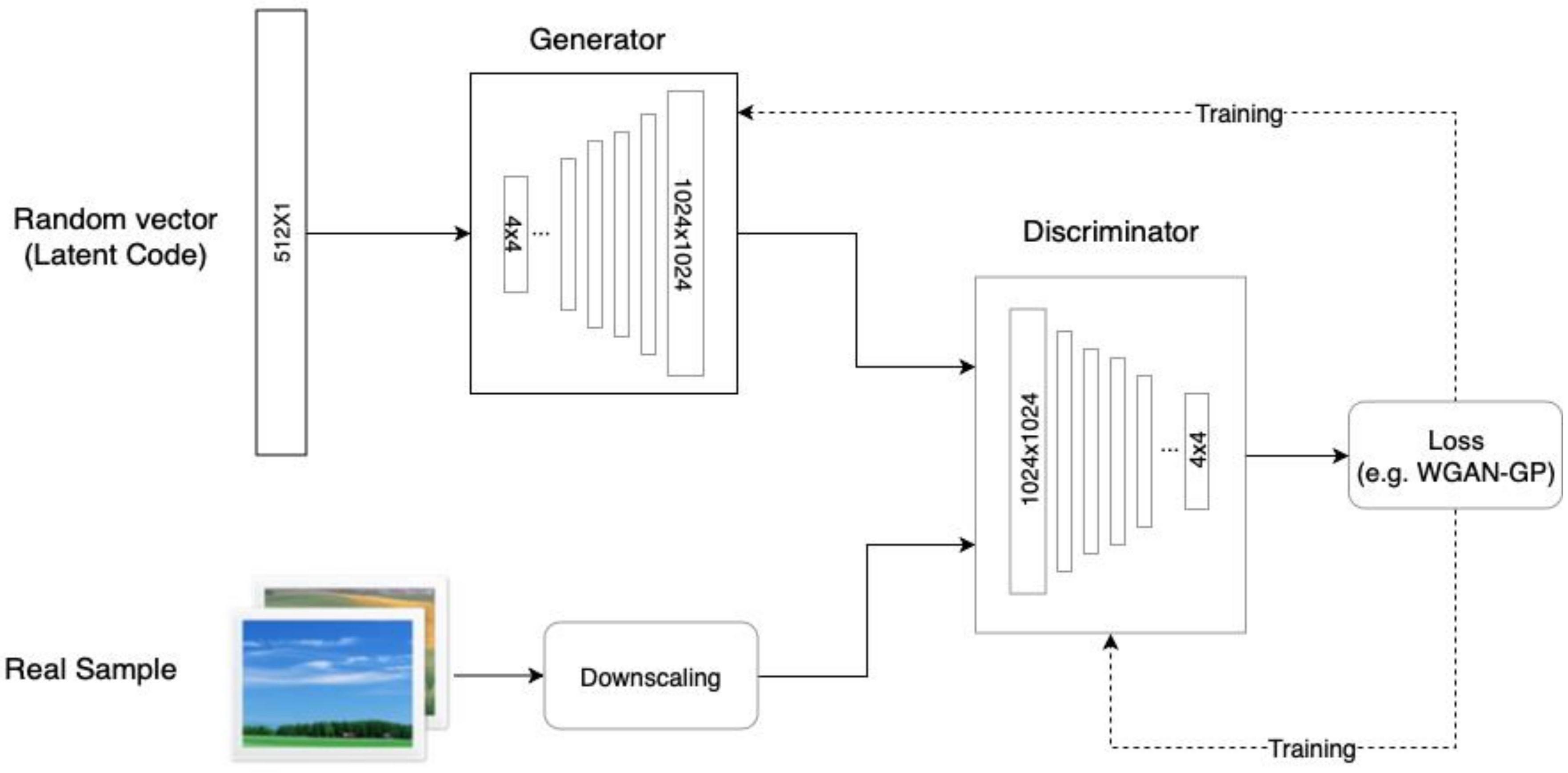


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout 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.

Background:


Random input
(Latent code)
z
Generator
Fake sample
(i.e. image)
Real Sample
Toggle
inputs
Discriminator
real/fake?
Loss


encoder
Smile: 0.99
Skin tone: 0.85
Gender: -0.73
Beard: 0.85
Glasses: 0.002
Hair color: 0.68
Latent attributes



- ~ 2018: **고화질**의 큰 이미지를 만들기 어려웠음

- ProGAN's keyword: 'Progressive Training'

StyleGAN: How it works?

- Upgrade version of ProGAN (image G)

1. Coarse(굵직한 특징)
- 4 ~ 8 해상도까지 (4 x 4 ~ 8 x 8 layer)
- ex. 포즈, 일반적인 헤어스타일, 얼굴형 등에 영향

2. Middle(중간 특징)
- 16 ~ 32 해상도까지 (16 x 16 ~ 32 x 32 layer)
- ex. 자세한 얼굴 특징, 헤어스타일, 눈 뜨거나/감음 등에 영향

3. Fine(자세한 특징)
- 64 ~ 1024 해상도까지 (64 x 64 ~ 1024 x 1024 layer)
- 눈, 머리, 피부 등의 색 조합 / 미세한 특징 등에 영향

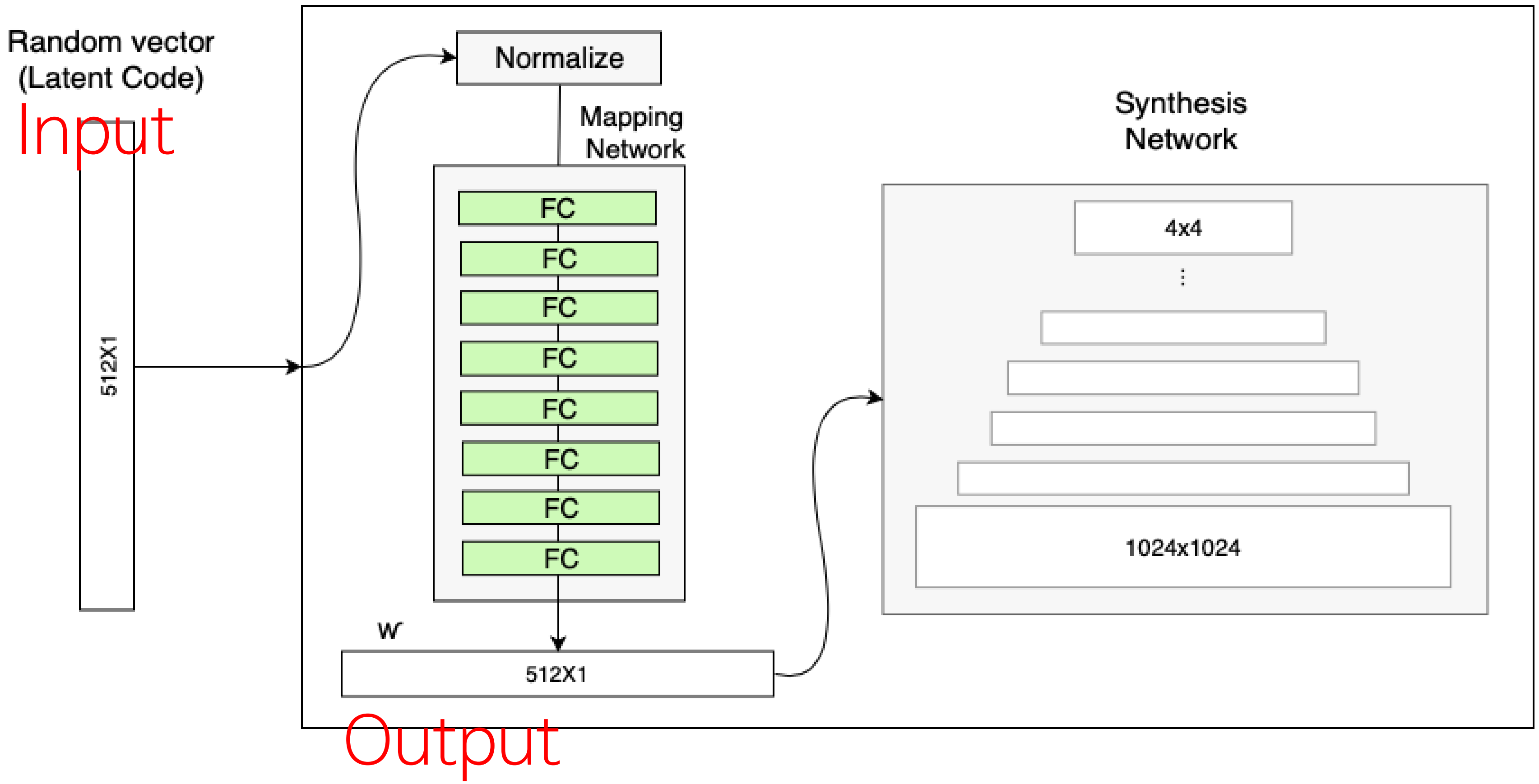
StyleGAN: Mapping Network

- Goals : Input vector -〉 intermediate vector(중간 벡터)로 인코딩 하는 것

- entangled vector (기존 GAN)

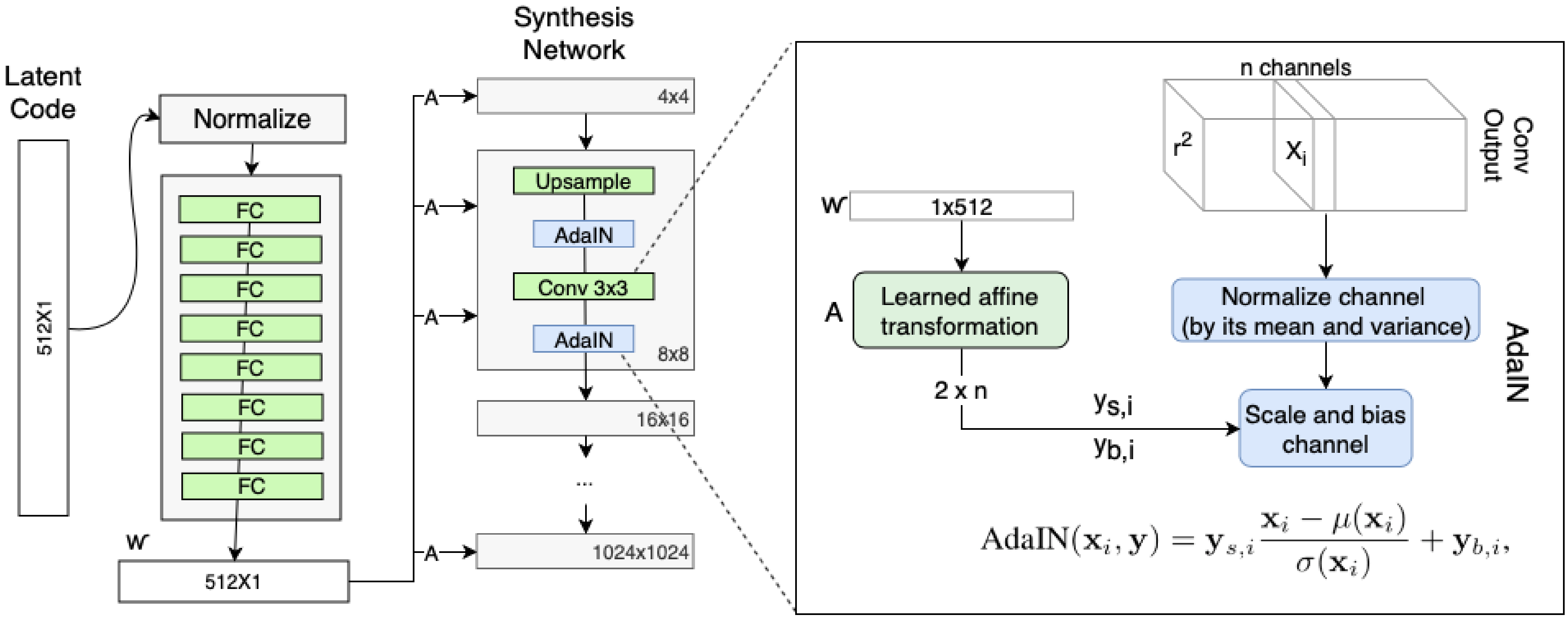
- disentangled vector

Mapping Network



- FC : fully connected Layer(=Dense Layer)

AdaIN (Adaptive Instance Normalization)



Style Mixing

Source A
: feature of low level

Source B
: feature of high level

- https://www.youtube.com/watch?v=kSLJriaOumA&t=1s

Results

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [26]	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**

The performance(FID score) of the model in the different configurations compared to ProGAN.

The lower score the better the model.

⇒SOTA score (best)

Figure 10. Uncurated set of images produced by our style-based generator (config F) with the LSUN BEDROOM dataset at 256^2. FID computed for 50K images was 2.65.

Figure 11. Uncurated set of images produced by our style-based generator (config F) with the LSUN CAR dataset at 512×384. FID computed for 50K images was 3.27.

Conclusion

5. Conclusion

Based on both our results and parallel work by Chen et al. [6], it is becoming clear that the traditional GAN generator architecture is in every way inferior to a style-based design. This is true in terms of established quality metrics, and we further believe that our investigations to the separation of high-level attributes and stochastic effects, as well as the linearity of the intermediate latent space will prove fruitful in improving the understanding and controllability of GAN synthesis.

We note that our average path length metric could easily be used as a regularizer during training, and perhaps some variant of the linear separability metric could act as one, too. In general, we expect that methods for directly shaping the intermediate latent space during training will provide interesting avenues for future work.