Alias-Free Generative Adversarial Networks

> Karras, Tero, et al. "Alias-free generative adversarial networks." *Advances in Neural Information Processing Systems* 34 (2021).

Abstract

이전 계층적인 convolution을 통한 이미지 합성에서는 객체의 스타일이 객체의 표면에 있는 것이 아닌 이미지 좌표에 의존한다는 문제점이 있음

참고 영상, 3:04~

1. Introduction

Texture sticking

실제 영상의 경우는 객체의 움직임에 따라 객체 내부의 요소들이 같이 움직이거나, 크기가 변한다.

하지만, gan의 경우 객체가 움직임에도 특정 요소들이 픽셀에 고정되어 있는 현상이 발생한다.

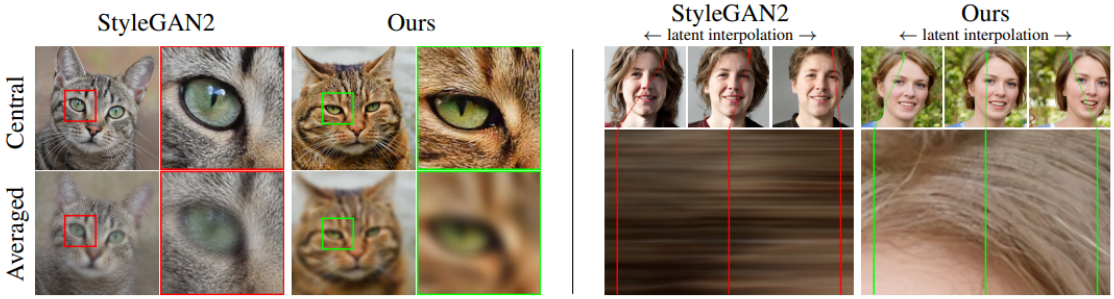


Figure 1: Examples of "texture sticking". **Left:** The average of images generated from a small neighborhood around a central latent (top row). The intended result is uniformly blurry because all details should move together. However, with StyleGAN2 many details (e.g., fur) stick to the same pixel coordinates, showing unwanted sharpness. **Right:** From a latent space interpolation (top row), we extract a short vertical segment of pixels from each generated image and stack them horizontally (bottom). The desired result is hairs moving in animation, creating a time-varying field. With StyleGAN2 the hairs mostly stick to the same coordinates, creating horizontal streaks instead.

이에 대한 예시로, 이미지를 약간씩 움직이면서 생성한 이미지들의 평균을 본 결과, 이전 stylegan2에선 눈이 blur 되고 주변의 털과 같은 texture들은 선명하게 있음을 확인 가능하다. 이는 눈 주변의 털과 같은 경우 움직이지 않음을 의미한다.

이유는 generator의 convolution이 의도치 않게 positional reference를 참조하기 때문

Unintentional positional reference

1. image borders
   Convolution 연산 시 모서리 padding을 함으로 network가 모서리의 위치를 파악함.
   이러한 점은 간단히 이미지를 크게 만들고 crop하는 방식으로 해결 가능
2. per-pixel noise inputs
3. positional encoding > 위치 자체에 대한 인코딩을 사용
4. aliasing > 위치 변환 후 upsampling 할 때, 특정 부분만 aliasing 현상이 발생하여 sync가 맞지 않음.

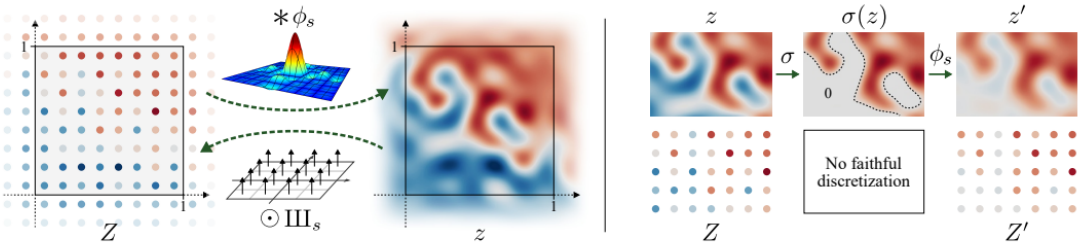
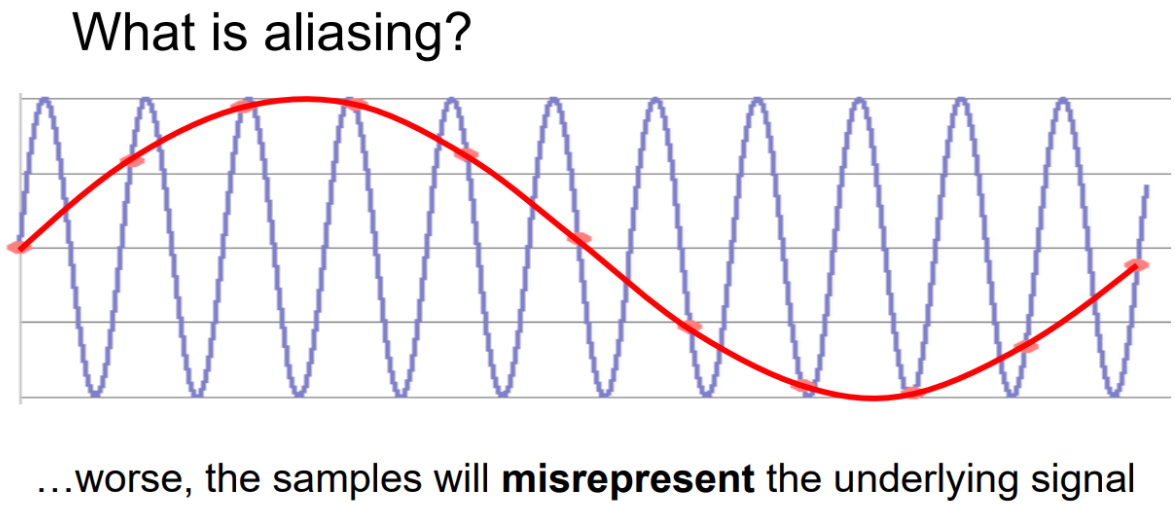


Figure 2: **Left:** Discrete representation Z and continuous representation z are related to each other via convolution with ideal interpolation filter ϕ_s and pointwise multiplication with Dirac comb Ш_s. **Right:** Nonlinearity σ, ReLU in this example, may produce arbitrarily high frequencies in the continuous-domain $\sigma(z)$. Low-pass filtering via ϕ_s is necessary to ensure that Z' captures the result.

upsampling filter의 interpolation 방식, ReLU나 swish의 픽셀별 비선형성
=> 여러 scale에서 texture의 패턴을 그리고 결합하면서 약간의 aliasing 현상 발생
이는 bandlimited function을 활용하여 이산적인 sample grid에 대한 표현보다 연속적인 도메인으로 변환

aliasing



aliasing은 특정 frequency를 sampling 할 때 적당하지 않은 sampling rate로 sampling하면 원래의 frequency를 복원 못하는 현상

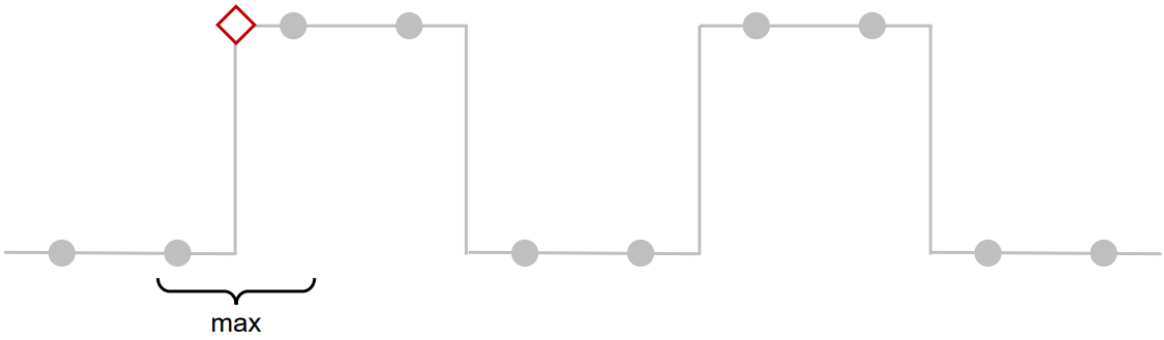
[Making Convolutional Networks Shift-Invariant Again]

> Zhang, Richard. "Making convolutional networks shift-invariant again." *International conference on machine learning*. PMLR, 2019.

이처럼 이미지가 이동함에도 같은 결과를 내야하는 특징(equivariance)을 가져야하지만 실제로 다른 결과를 내보인다.

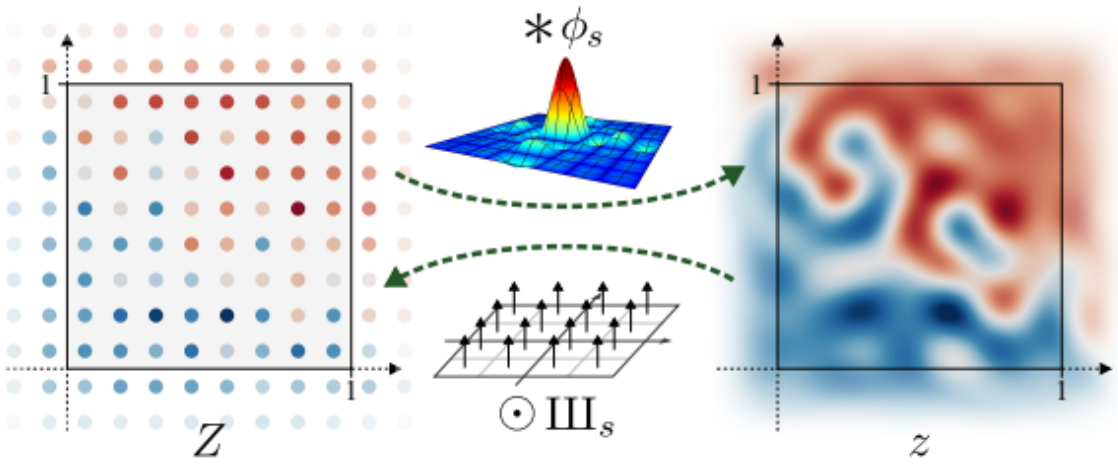
이 문제는 convolution이 아닌 pooling에 이유가 있음

Re-examining Max-Pooling



이러한 shift equivariance에 대해선 stride를 1로 maxpooling-> blur kernel -> sub sampling을 통해 개선한 선행 연구가 있음.

2. Equivariance via continuous signal interpretation



$$\mathbf{f}(z) = \phi_{s'} * \mathbf{F}(\text{Ш}_s \odot z), \qquad \mathbf{F}(Z) = \text{Ш}_{s'} \odot \mathbf{f}(\phi_s * Z), \tag{1}$$

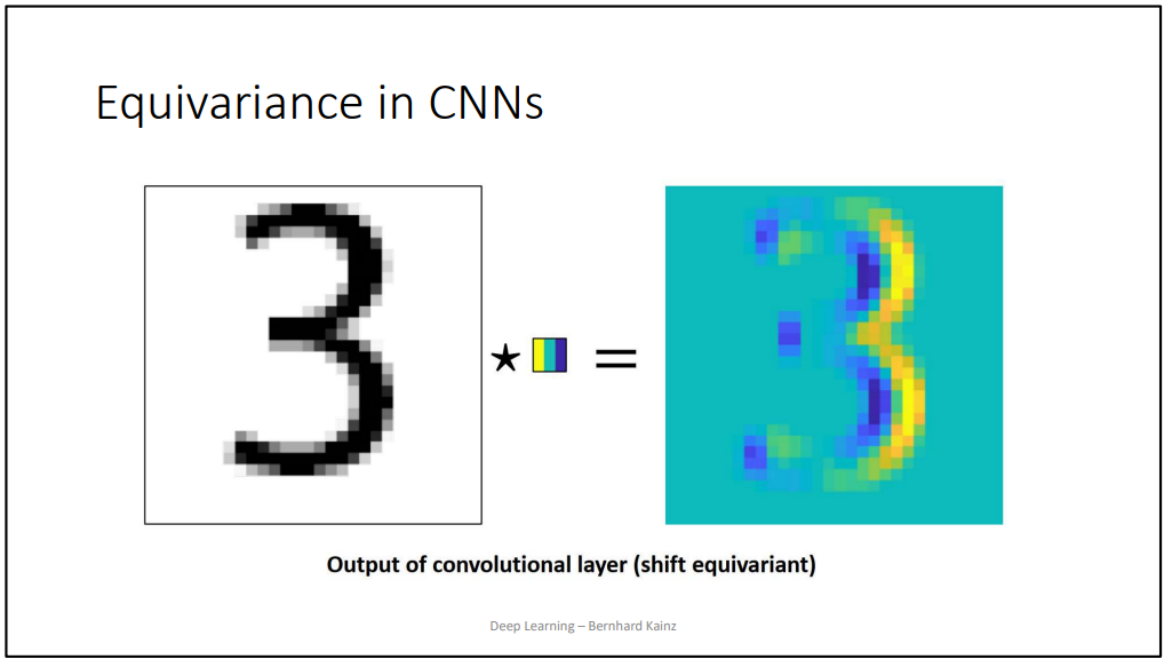
이처럼 sampling과 reconstruction을 하는 filter를 잘 설계하면 equivariance를 만족 할 수 있다.

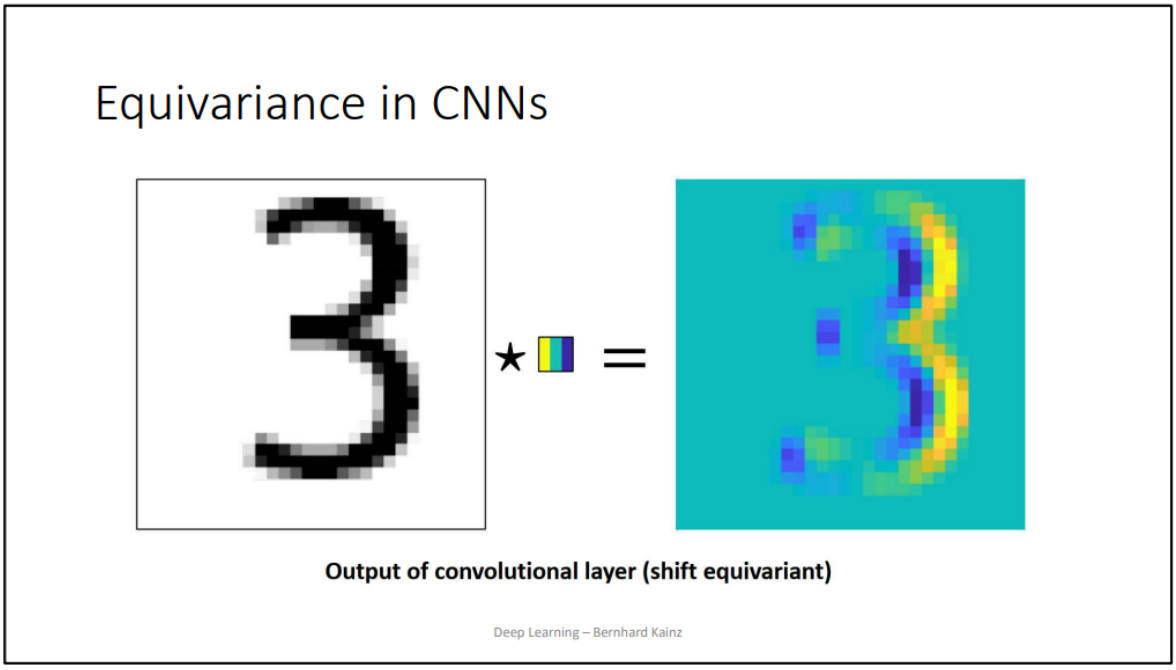
2.1 Equivariant network layers

$$f \circ t = t \circ f$$

위 식에서 f는 generator, t는 translation일 때, 연산 순서가 서로 바뀌어도 같은 결과가 나오는지 확인해야한다.

Convolution





직관적으로 convolution 연산은 kernel이 sliding 하면서 연산을 하기 때문에, translation에 equivariance특성을 가진다고 생각할 수 있다.

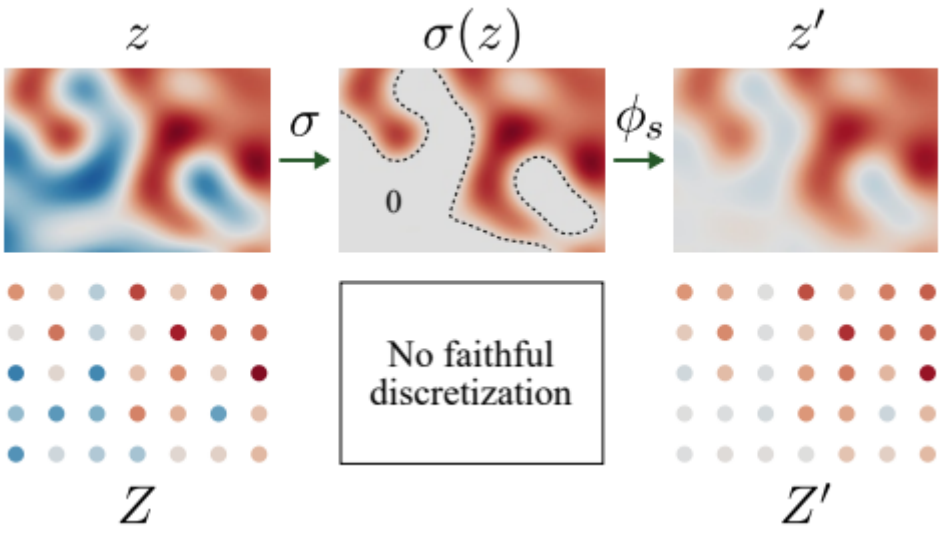
$$\mathbf{f}_{\text{conv}}(z) = \phi_s * \left(K * (\text{Ш}_s \odot z)\right) = K * \left(\phi_s * (\text{Ш}_s \odot z)\right) = K * z \tag{2}$$

rotation?

Upsampling and Downsampling

이상적인 upsampling은 Shannon-Nyquist에 맞는 sampling rate가 된다면 연속 공간 내에서 표현을 바꾸지 않지만,

Nonlinearity



위처럼 ReLU activation의 경우 0 기준으로 nonlinear하기 때문에, 위 그림에서 가운데 실제 출력 부분과 같이 엣지가 생기게 된다. 이러한 엣지는 continuous domain에서 결과에서 표현될 수 없는 높은 frequency를 가지게 된다는 것이다. 하지만 discrete domain Z에서는 바로 직접적으로 연산이 불가함으로 z로 upsampling 한 후에 z' 이후 다시 downsampling 함으로 근사시킬 수 있다.

이는 low-pass filter를 적용함으로 z'과 같이 high frequency를 제거 가능하다.

3. Practical application to generator network

위와 같은 방법을 적용한다면, generator는 w,rotation,translation에 대해 equivariant 특성을 가질 수 있을 것이다.

$$z_0 : \mathbf{g}(\mathbf{t}[z_0]; w) = \mathbf{t}[\mathbf{g}(z_0; w)]$$

이에 대해서 평가하기 위해서 PSNR을 사용하여 평가할 수 있다.

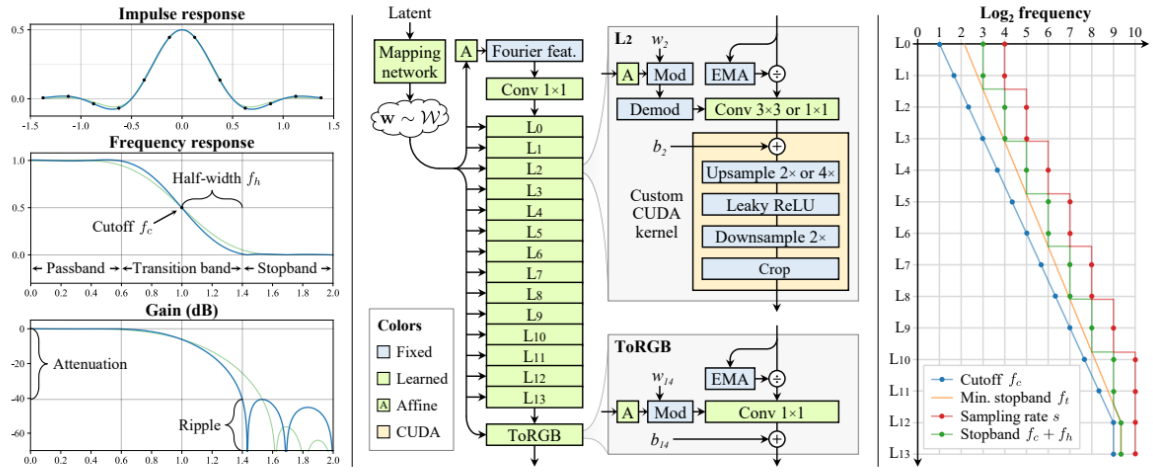
$$PSNR = 10 \cdot log_{10}\left(\frac{MAX_i^2}{MSE}\right)$$

$$EQ-T = 10 \cdot log_{10}\left(I_{max}^2 / \mathbb{E}_{\mathbf{w}\sim\mathcal{W},x\sim\mathcal{X}^2,p\sim\mathcal{V},c\sim\mathcal{C}}\left[\left(\mathbf{g}(\mathbf{t}_x[z_0]; \mathbf{w})_c(p) - \mathbf{t}_x[\mathbf{g}(z_0; \mathbf{w})]_c(p)\right)^2\right]\right)$$

	Configuration	FID↓	EQ-T↑	EQ-R↑
A	StyleGAN2	5.14	–	–
B	+ Fourier features	4.79	16.23	10.81
C	+ No noise inputs	4.54	15.81	10.84
D	+ Simplified generator	5.21	19.47	10.41
E	+ Boundaries & upsampling	6.02	24.62	10.97
F	+ Filtered nonlinearities	6.35	30.60	10.81
G	+ Non-critical sampling	4.78	43.90	10.84
H	+ Transformed Fourier features	4.64	45.20	10.61
T	+ Flexible layers (StyleGAN3-T)	4.62	63.01	13.12
R	+ Rotation equiv. (StyleGAN3-R)	**4.50**	**66.65**	**40.48**

	Parameter	FID↓	EQ-T↑	EQ-R↑	Time	Mem.
	Filter size $n = 4$	4.72	57.49	39.70	**0.84**×	**0.99**×
*	Filter size $n = 6$	**4.50**	**66.65**	40.48	1.00×	1.00×
	Filter size $n = 8$	4.66	65.57	**42.09**	1.18×	1.01×
	Upsampling $m = 1$	**4.38**	39.96	36.42	**0.65**×	**0.87**×
*	Upsampling $m = 2$	4.50	66.65	40.48	1.00×	1.00×
	Upsampling $m = 4$	4.57	**74.21**	**40.97**	2.31×	1.62×
	Stopband $f_{t,0} = 2^{1.5}$	4.62	51.10	29.14	**0.86**×	**0.90**×
*	Stopband $f_{t,0} = 2^{2.1}$	**4.50**	66.65	40.48	1.00×	1.00×
	Stopband $f_{t,0} = 2^{3.1}$	4.68	**73.13**	**41.63**	1.36×	1.25×

Figure 3: Results for FFHQ-U (unaligned FFHQ) at 256^2. **Left:** Training configurations. FID is computed between 50k generated images and all training images [26, 32]; lower is better. EQ-T and EQ-R are our equivariance metrics in decibels (dB); higher is better. **Right:** Parameter ablations using our final configuration (R) for the filter's support, magnification around nonlinearities, and the minimum stopband frequency at the first layer. * indicates our default choices.



(a) Filter design concepts (b) Our alias-free StyleGAN3 generator architecture (c) Flexible layers

Figure 4: **(a)** 1D example of a 2× upsampling filter with $n = 6$, $s = 2$, $f_c = 1$, and $f_h = 0.4$ (blue). Setting $f_h = 0.6$ makes the transition band wider (green), which reduces the unwanted stopband ripple and thus leads to stronger attenuation. **(b)** Our alias-free generator, corresponding to configs T and R in Figure 3. The main datapath consists of Fourier features and normalization (Section 3.1), modulated convolutions [34], and filtered nonlinearities (Section 3.2). **(c)** Flexible layer specifications (config T) with $N = 14$ and $s_N = 1024$. Cutoff f_c (blue) and minimum acceptable stopband frequency f_t (orange) obey geometric progression over the layers; sampling rate s (red) and actual stopband $f_c + f_h$ (green) are computed according to our design constraints.

Fourier features (B)

이전 stylegan2 는 입력 벡터를 4x4x512로 사용했지만, 입력 벡터를 fourier features로 바꾸게 되었다.

이는 frequency domain이기 때문에 크기가 정해져 있는게 아닌 무한한 크기의 맵을 정의해줄 수 있다고 한다.

No noise inputs (C)

이는 이전의 per-pixel noise input은 translation과 상관이 없기 때문에 제거하였다고 한다.

Simplified generator (D)

이전 stylegan2에서 mapping network depth, mixing regularization, path length regularization을 줄이거나 사용하지 않았다.
그 이유로는 위의 방법들은 gradient에 대해서 FID를 높이기 위해서 사용했던 방법으로, 간단히 convolution 전에 nomalization을 함으로 간단히 해결하였다고 한다. 그렇기 때문에 상대적으로 FID가 높아짐을 확인할 수 있다.

Boundaries and upsampling (E)

image borders에서 모서리를 통해 위치 정보를 알 수 있었으니, 더 큰 크기의 feature map을 연산 후 crop함.

이전의 bilinear upsampling 대신 kaiser window 기반의 filter를 만들어 convolution연산 함으로 upsamping한다.

Filtered nonlinearities (F)

ReLU와 같은 non-linear activation으로 인한 high frequency 문제에 대해서
m x upsampling -> ReLU -> m x downsampling 과 같은 방식으로 사용한다.

하지만 upsampling 하는 layer의 경우,
2m x upsampling -> ReLU -> m x downsampling 을 사용하여 upsampling을 진행한다.

이와 같은 block의 경우 custom cuda kernel를 따로 구현하여 학습 속도를 10배 빠르게 했다고 한다.

Non-critical sampling (G)

low-pass filter의 경우 Frequency response 그림을 보다시피 cut off frequency를 낮춘다.

$$f_c = s/2 - f_h$$

이는 s/2보다 큰 frequency를 제거해버리게 되는데 이렇게 한다면 aliasing을 확실히 줄일 수 있다. 하지만 generator의 resolution이 높은 layer의 경우 명확한 이미지, 즉 high frequency에 대해서 생성해야 함으로 마지가장 큰 resolution을 가지는 layer를 제외하고 cut off frequency를 낮추었다고 한다.

Transformed Fourier features (H)

이는 추가적인 affine layer를 사용해서 w를 기준으로 (r_cos, r_sin, t_x, t_y)를 생성하도록 하여 rotation, translation과 같은 기하적 변환을 만들어 내는 것이다.

이를 통해서 style에 해당했던 w와 fourier features간의 domain을 맞추는 것과 비슷한 역할을 한 것

Flexible layer specifications (T)

config G 와 같은 방법으로 equivariance quality에 대한 향상이 있었지만, 아직도 artifacts들이 존재했다고 한다.

그래서 낮은 resolution일 수록 aliasing을 없에기 위해 낮은 cut off를 적용해야 했으며,
반대로, 높은 resolution일 수록 high frequency로부터 나오는 detail을 위해 높은 cut off를 적용해야 한다.

그래서 위의 그림 C와 같이 각 layer에서 f_c,f_h,f_t를 자동적으로 결정할 수 있게 한다.

Rotation equivariance (R)

rotation equivariance를 위해서 모든 convolution kernel의 크기를 3x3 에서 1x1로 감소시키고 feature map 수를 2배로 증가시켰다고 한다.

그 이유로는 3x3 kernel의 경우는 kernel 내에서 방향이 나오기 때문에 방향성이 없는 1x1을 사용한 것이다. 이로 인한 capacity 감소가 있기 때문에 feature map을 2배 증가시킨다.
이는 fid나 eq-t 를 크게 해치지 않으면서 각 layer당 56%정도의 파라미터 감소를 만들 수 있다고 한다.

추가로 downsampling filter를 radially symmetric하게 만들어 어느 방향으로든 대칭이 될 수 있게 만들었다고 한다.

4. Results

Dataset	Config	FID↓	EQ-T↑	EQ-R↑
FFHQ-U 70000 img, 1024^2 Train from scratch	StyleGAN2	3.79	15.89	10.79
	StyleGAN3-T (ours)	3.67	61.69	13.95
	StyleGAN3-R (ours)	**3.66**	**64.78**	**47.64**
FFHQ 70000 img, 1024^2 Train from scratch	StyleGAN2	**2.70**	13.58	10.22
	StyleGAN3-T (ours)	2.79	61.21	13.82
	StyleGAN3-R (ours)	3.07	**64.76**	**46.62**
METFACES-U 1336 img, 1024^2 ADA, from FFHQ-U	StyleGAN2	18.98	18.77	13.19
	StyleGAN3-T (ours)	**18.75**	64.11	16.63
	StyleGAN3-R (ours)	**18.75**	**66.34**	**48.57**
METFACES 1336 img, 1024^2 ADA, from FFHQ	StyleGAN2	15.22	16.39	12.89
	StyleGAN3-T (ours)	**15.11**	**65.23**	16.82
	StyleGAN3-R (ours)	15.33	64.86	**46.81**
AFHQV2 15803 img, 512^2 ADA, from scratch	StyleGAN2	4.62	13.83	11.50
	StyleGAN3-T (ours)	**4.04**	60.15	13.51
	StyleGAN3-R (ours)	4.40	**64.89**	**40.34**
BEACHES 20155 img, 512^2 ADA, from scratch	StyleGAN2	5.03	15.73	12.69
	StyleGAN3-T (ours)	**4.32**	59.33	15.88
	StyleGAN3-R (ours)	4.57	**63.66**	**37.42**

Ablation	Translation eq.		+ Rotation eq.		
	FID↓	EQ-T↑	FID↓	EQ-T↑	EQ-R↑
* Main configuration	4.62	63.01	**4.50**	66.65	40.48
With mixing reg.	**4.60**	63.48	4.67	63.59	40.90
With noise inputs	4.96	24.46	5.79	26.71	26.80
Without flexible layers	4.64	45.20	4.65	44.74	22.52
Fixed Fourier features	5.93	64.57	6.48	66.20	41.77
With path length reg.	5.00	**68.36**	5.98	**71.64**	**42.18**
0.5× capacity	7.43	63.14	6.52	63.08	39.89
* 1.0× capacity	4.62	63.01	4.50	66.65	40.48
2.0× capacity	**3.80**	**66.61**	**4.18**	**70.06**	**42.51**
* Kaiser filter, $n = 6$	**4.62**	**63.01**	4.50	**66.65**	**40.48**
Lanczos filter, $a = 2$	4.69	51.93	**4.44**	57.70	25.25
Gaussian filter, $\sigma = 0.4$	5.91	56.89	5.73	59.53	39.43

G-CNN comparison	FID↓	EQ-T↑	EQ-R↑	Params	Time
* StyleGAN3-T (ours)	4.62	63.01	13.12	23.3M	**1.00**×
+ $p4$ symmetry [16]	4.69	61.90	17.07	21.8M	2.48×
* StyleGAN3-R (ours)	**4.50**	**66.65**	**40.48**	15.8M	1.37×

Figure 5: **Left:** Results for six datasets. We use adaptive discriminator augmentation (ADA) [32] for the smaller datasets. "StyleGAN2" corresponds to our baseline config B with Fourier features. **Right:** Ablations and comparisons for FFHQ-U (unaligned FFHQ) at 256^2. * indicates our default choices.

5. Limitations, discussion, and future work

generator 뿐만 아니라 discriminator도 equivariant하게 만들면 더욱 좋을 것이다.

noise input과 path length regularization을 equivariant를 고려하여 설계