Analyzing and Improving the Image Quality of StyleGAN[1]

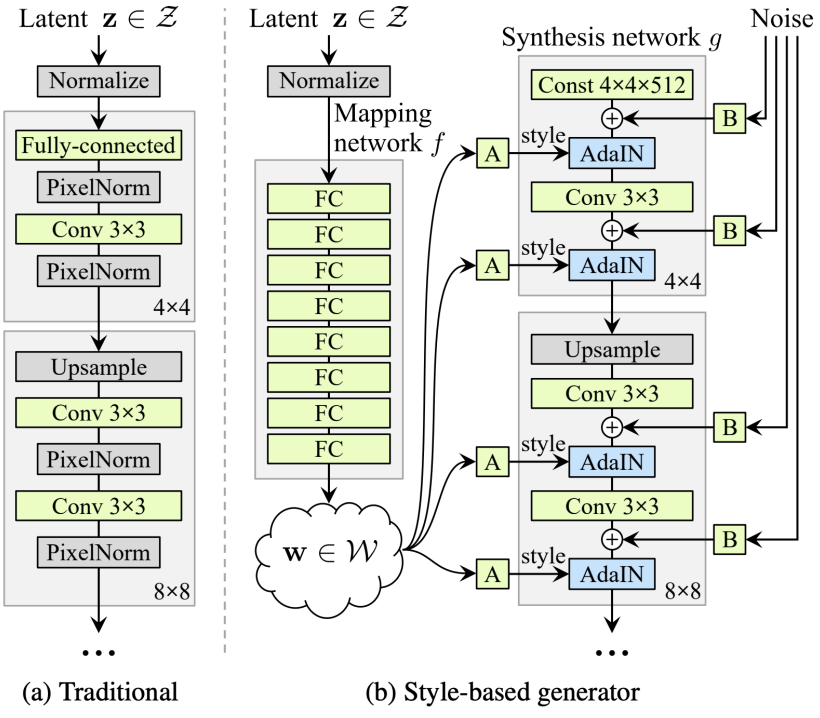
August 19, 2020

[1] "Analyzing and Improving the Image Quality of StyleGAN" by Karras et al., 2020

Overview

- Authors improve several flaws of StyleGAN2
  - Remove *droplet artifacts* by replacing AdaIN
  - Remove *phase artifacts* by replacing progressive growing with multi-scale gradients
  - Make latent space nicer by introducing *path length regularization*
- Overall, scores are improved by up to 30% in terms of quantitative metrics

StyleGAN architecture



(a) Traditional (b) Style-based generator

- Two sources of noise: latent noise z and many additional channel-wise noises at each resolution
- z is transformed into latent code w through a mapping network
- Latent vector w is passed through AdaIN layers
- Uses progressive growing

Adaptive Instance Normalization (AdaIN)

- AdaIN is similar to batch normalization, but operates on an instance-wise basis
- It replaces statistics $\mu_i(x_i), \sigma_i(x_i)$ of filter map x_i with the externally provided ones $y_{b,i}, y_{s,i}$:

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

- StyleGAN passes w into G via AdaIN instead of feed-forward layers as is usually done

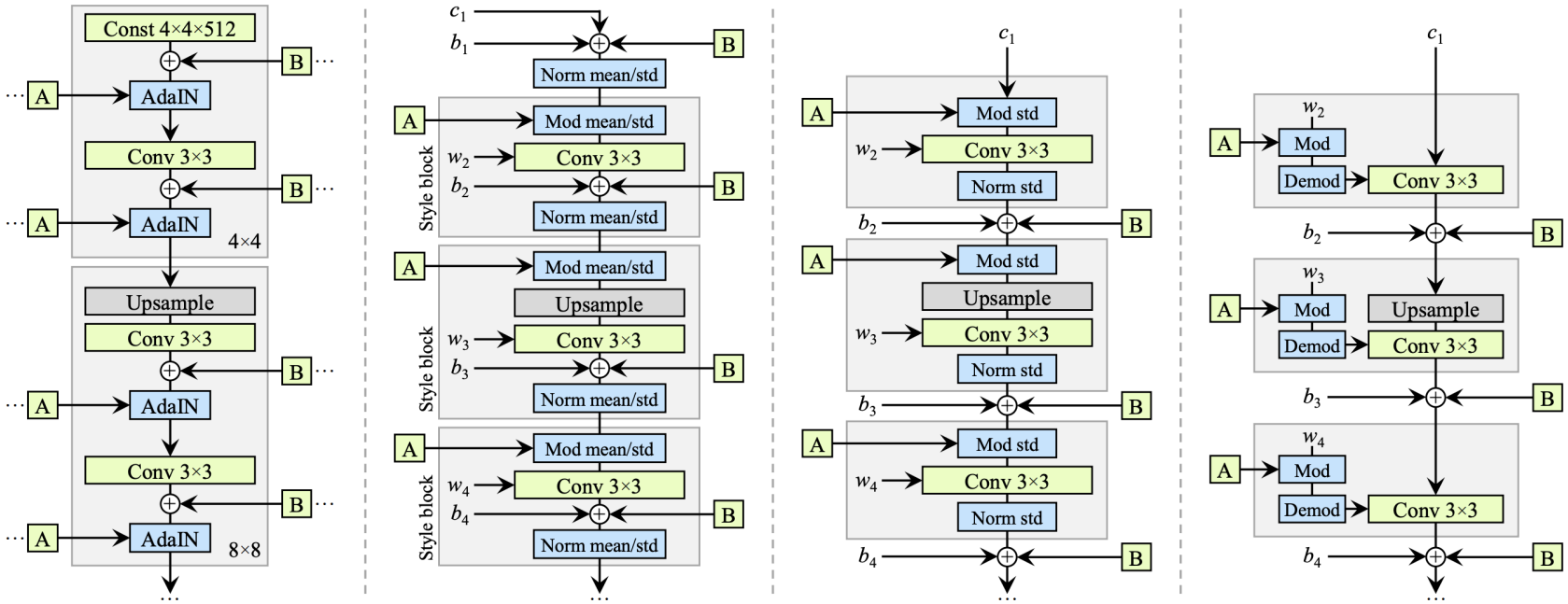
Droplet artifact

AdaIN causes droplet artifacts:

- Imagine that G carries some important information in activations magnitudes
- Generally, normalization procedure erases this information
- A dirty workaround for G to preent the erasion is to make some activations *huge*
- This allows it to set mean/std for the remaining activations to the desired value

Weights demodulation

- Authors noticed that it is not necessary to normalize the mean
- Then they noticed that updating the std is equivalent to rescaling convolutional weights



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

Perceptual Path Length

StyleGANv1 paper proposed PPL metric to measure the latent space quality

$$\ell_{\mathcal{Z}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d\left(G\left(\text{slerp}\left(\mathbf{z}_1, \mathbf{z}_2; t\right)\right), G\left(\text{slerp}\left(\mathbf{z}_1, \mathbf{z}_2; t+\epsilon\right)\right)\right)\right] \tag{2}$$

where d is the *perceptual distance*, i.e. a L_2 distance between VGG16 embeddings.

Path length regularization

Authors noticed that PPL correlates very well with image quality and propose a way to optimize it during training:

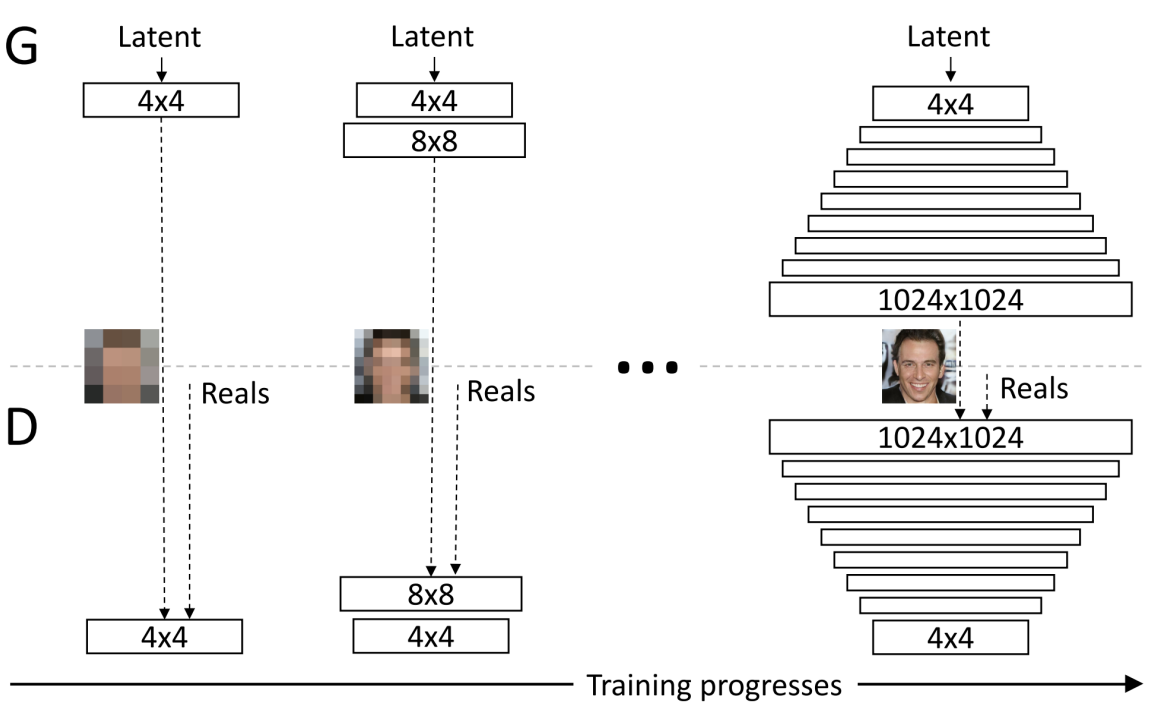
$$\mathbb{E}_{\mathbf{w},\mathbf{y}\sim\mathcal{N}(0,\mathbf{I})}\left(\left\|\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y}\right\|_{2}-a\right)^{2} \tag{3}$$

where:

- J_w is the Jacobian of G with respect to latent code $\boldsymbol{w}$
- $\boldsymbol{y}$ is a random image of size $512 \times 512 \times 3$
- a is computed as EMA of $\|\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y}\|_{2}$
- The motivation is to make G change its output by a constant value when w is changed by a constant value
- They prove that it is minimized when J_w is orthogonal
  - i.e. output is changing at the same speed as the input

Progressive growing

ProGAN gently adds new layers during training increasing the resolution:



After adding a layer we use a weighted sum for a new output resolution:

$$x_{\text{out}} = \alpha \cdot x_{\text{new}} + (1 - \alpha) \cdot \text{upsample}(x_{\text{old}}) \tag{4}$$

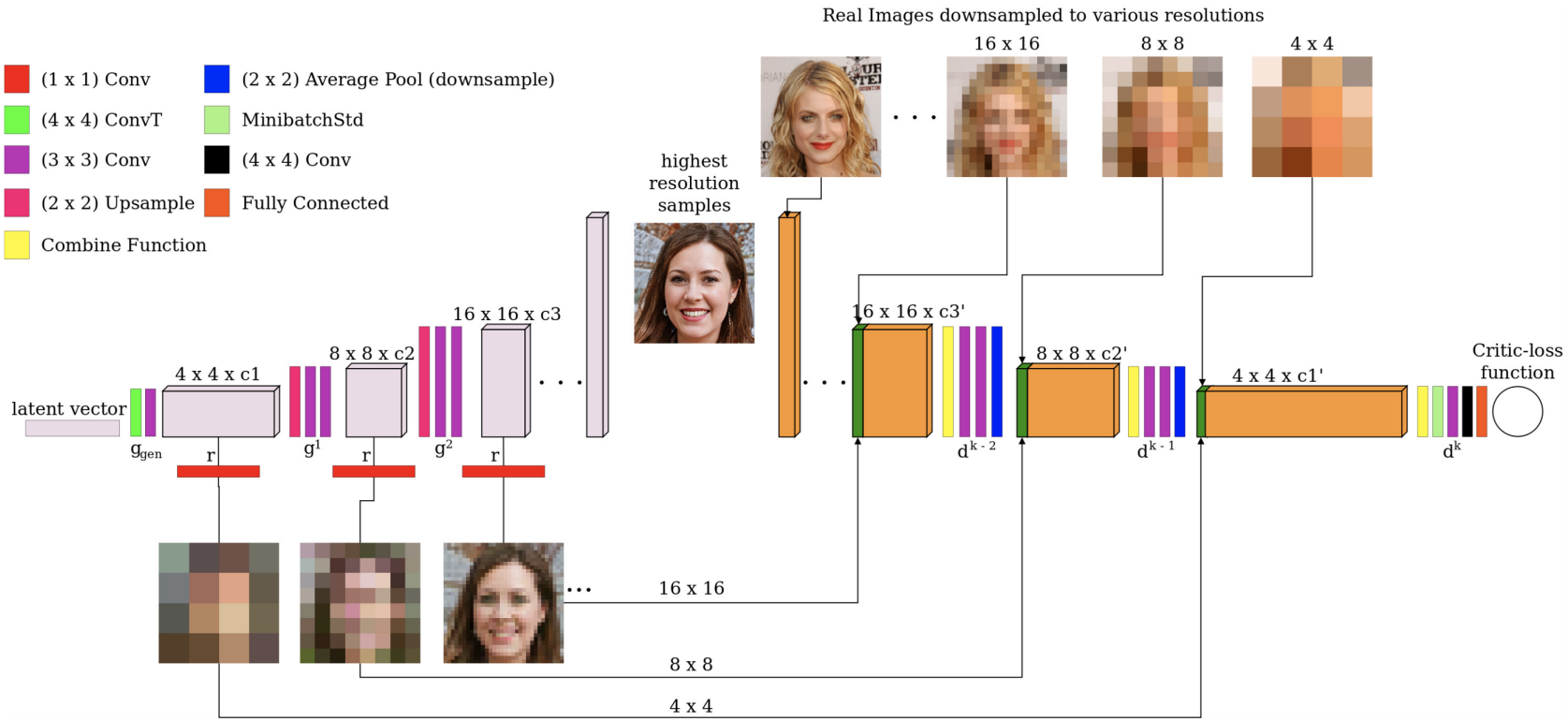
and α is gradually increased from 0 to 1.

Phase artifact

- We rotate a face, but teeth are not rotated
- Authors blame progressive growing for this:
  - During training, x_{old} is used to produce an image of twice higher resolution than it is suitable for
  - This forces G to generate excessively high frequencies for lower resolutions
  - This forces G to decide on important details too early during a forward pass
  - This "compromises shift invariance"[2]

[2] I didn't get that too.

Multi-scale gradient



Authors incorporate MSG in StyleGAN2 and remove progressing growing.[3]

[3] They modify it a lot but it's out of scope

Lazy regularization

- Instead of WGAN-GP penalty, they use R1-penalty:

$$R_1 := \frac{\gamma}{2} \mathrm{E}_{p_{\mathcal{D}}(x)} \left[\|\nabla D(x)\|^2 \right] \tag{5}$$

- They compute it only once in 16 iterations which makes training faster
- They also compute PLR regularization once in 8 iterations
- This requires tweaking Adam optimizer

Final scores

	Configuration	FFHQ, 1024×1024				LSUN Car, 512×384			
		FID ↓	Path length ↓	Precision ↑	Recall ↑	FID ↓	Path length ↓	Precision ↑	Recall ↑
A	Baseline StyleGAN [24]	4.40	212.1	**0.721**	0.399	3.27	1484.5	**0.701**	0.435
B	+ Weight demodulation	4.39	175.4	0.702	0.425	3.04	862.4	0.685	0.488
C	+ Lazy regularization	4.38	158.0	0.719	0.427	2.83	981.6	0.688	0.493
D	+ Path length regularization	4.34	**122.5**	0.715	0.418	3.43	651.2	0.697	0.452
E	+ No growing, new G & D arch.	3.31	124.5	0.705	0.449	3.19	471.2	0.690	0.454
F	+ Large networks (StyleGAN2)	**2.84**	145.0	0.689	**0.492**	**2.32**	**415.5**	0.678	**0.514**
	Config A with large networks	3.98	199.2	0.716	0.422	–	–	–	–

Cherry-picked samples

Non-cherrypicked samples 1/2

Non-cherrypicked samples 2/2

Final thoughts

- Strong SotA for image generation
- The ablation study is good
- A lot of interesting gritty details are being discussed
- They also observed a good improvement in quality by simply increasing a model size
- They spent 51 GPU-years (v100-like) for the entire project
  - If it took 3 months, then it was equivalent to a *constant* usage of 204 GPUs