Analyzing and Improving the Image Quality of StyleGAN

Arxiv

2019.12.23

발표자 박성현

DAVIAN
Data and Visual Analytics Lab

KOREA UNIVERSITY

1 Introduction

Progressive GAN

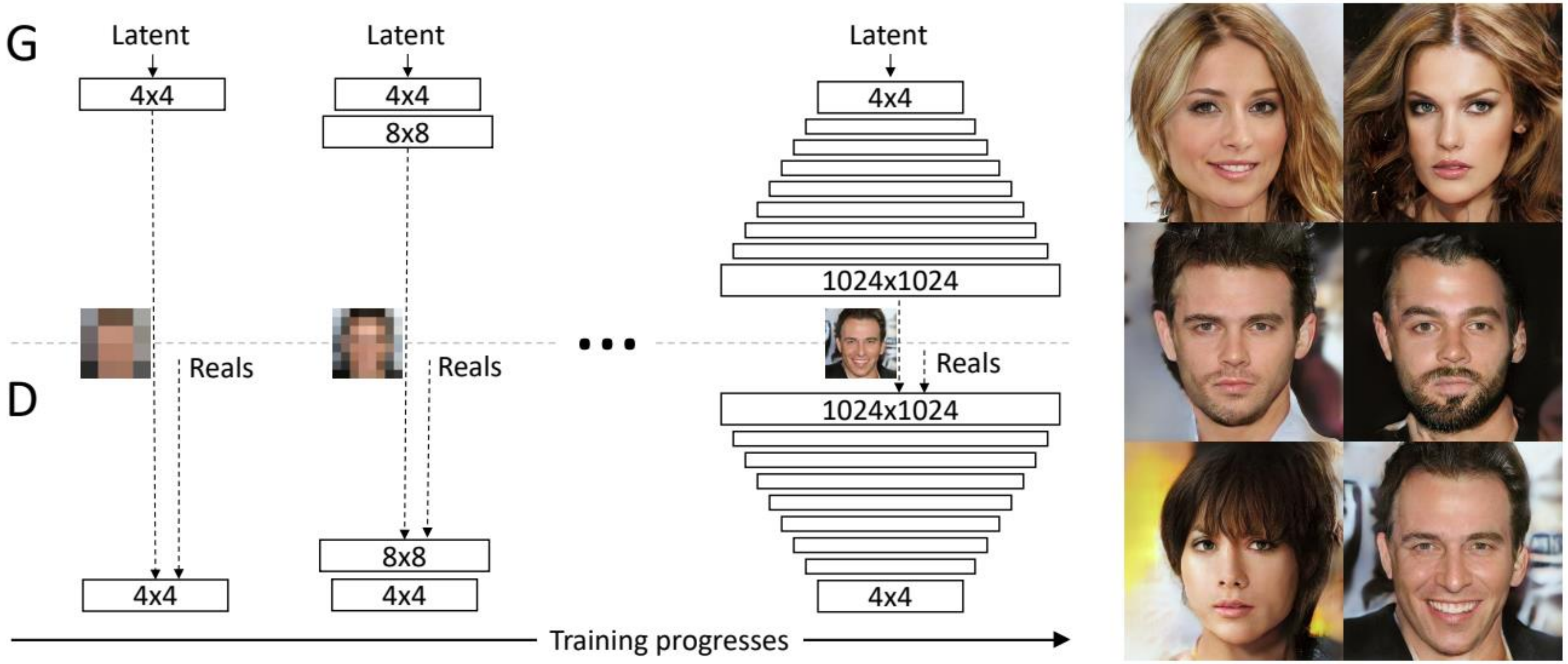


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout 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 Growing of GANs for Improved Quality, Stability, and Variation (ICLR 2018)

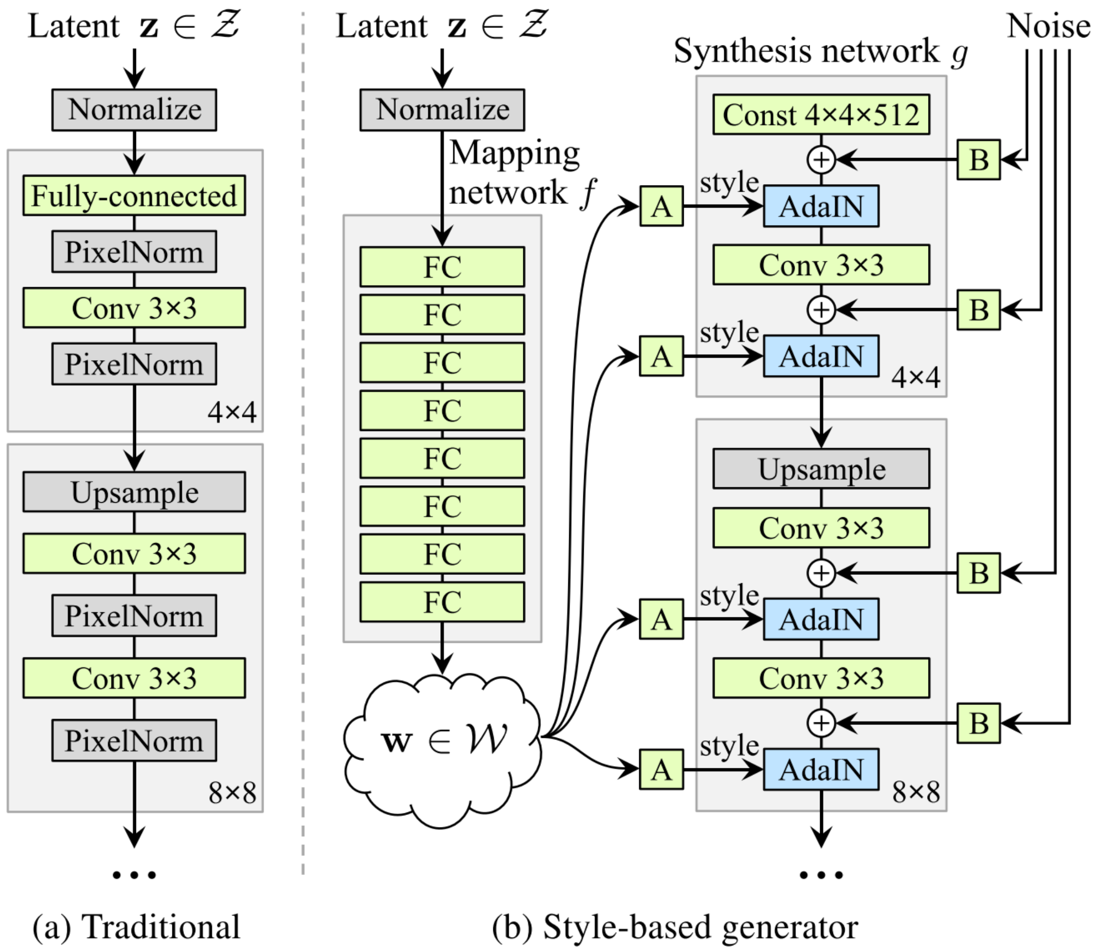
1 Introduction

Progressive GAN

Progressive Growing of GANs for Improved Quality, Stability, and Variation (ICLR 2018)

1 Introduction

StyleGAN - Overview of the model

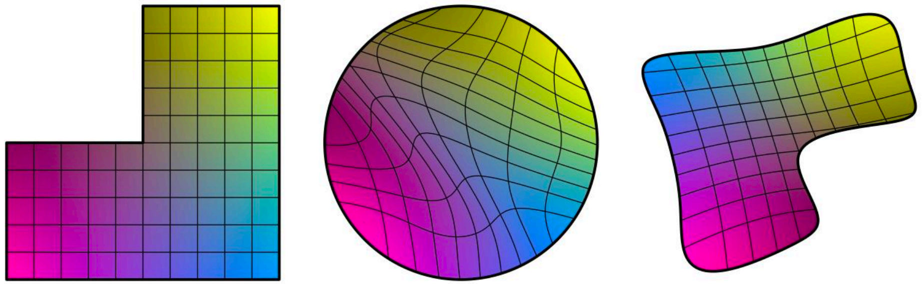


A Style-Based Generator Architecture for Generative Adversarial Networks (CVPR 2019)

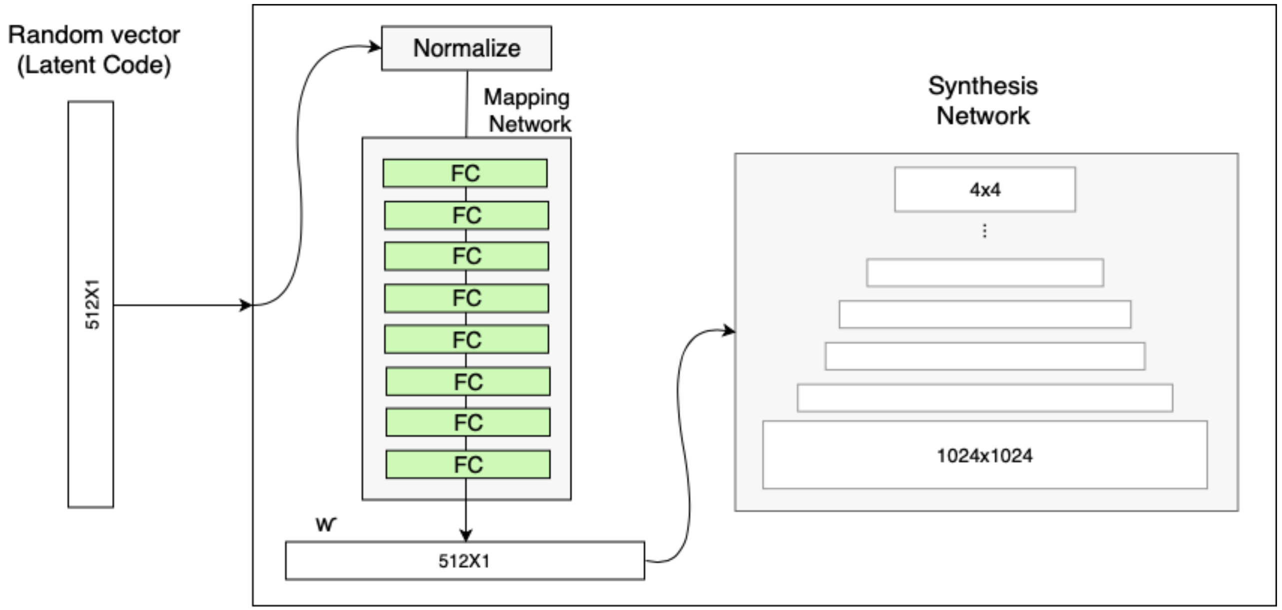
1 Introduction

StyleGAN - Mapping Network

- Input vector(z)로부터 직접 이미지를 생성하는 것이 아니라 mapping network를 거쳐 intermediate vector(w)로 먼저 변환한 후 이미지를 생성한다.
- Mapping network를 사용할 경우 w는 고정된 distribution을 따를 필요가 없어지기 때문에, 학습 데이터를 훨씬 유동적인 공간에 mapping할 수 있고 w를 이용하여 visual attribute를 조절하기 훨씬 용이해 진다.



(a) Distribution of features in training set (b) Mapping from $\mathcal{Z}$ to features (c) Mapping from $\mathcal{W}$ to features

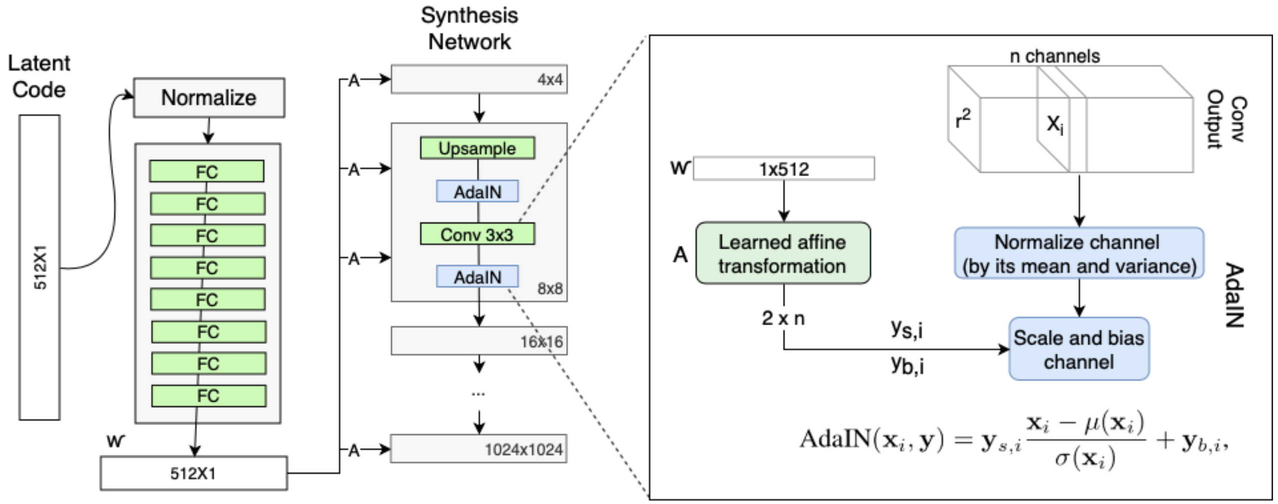


A Style-Based Generator Architecture for Generative Adversarial Networks (CVPR 2019)

1 Introduction

StyleGAN - Style Module

- Synthesis network의 매 layer마다 AdaIN을 통해 style을 normalize한 후 새로운 style을 입히게 되므로, 특정 layer에서 입혀진 style은 바로 다음 Conv에만 영향을 끼친다. 따라서 각 layer의 style이 특정한 visual attribute만 담당하는 것이 용이하다.
- Style을 조정한다는 것은 이미지의 global한 정보를 통째로 조정한다는 것을 의미한다. 이로 인해 항상 spatially-consistent한 이미지를 얻게 되고, 기존의 generator보다 안정적으로 이미지를 얻을 수 있다.

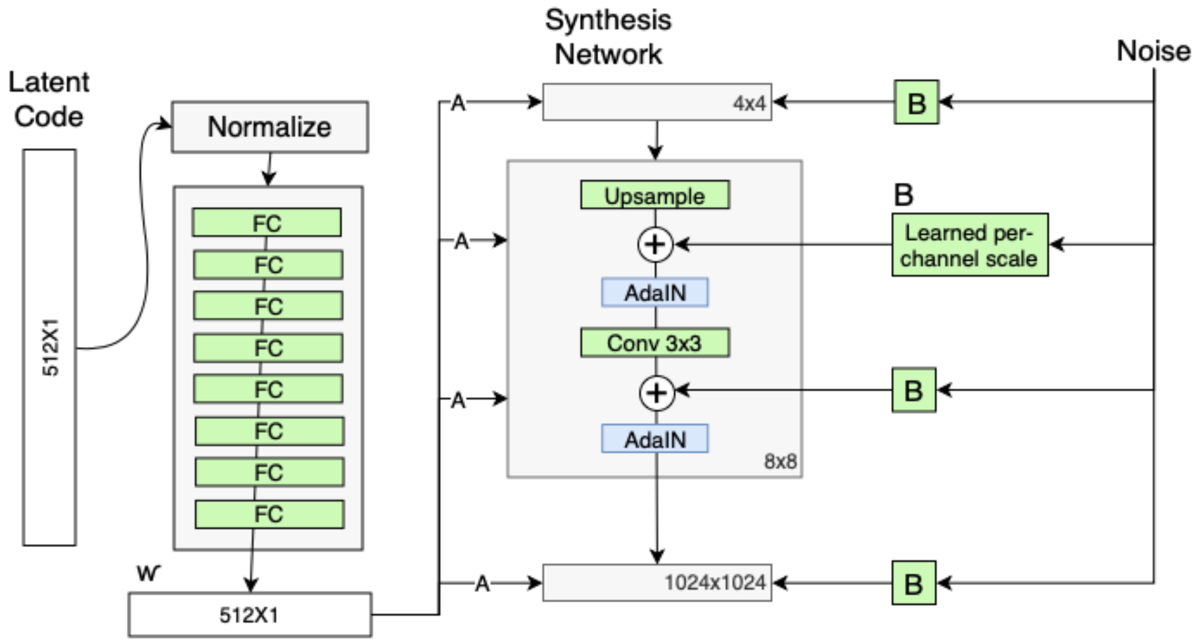


A Style-Based Generator Architecture for Generative Adversarial Networks (CVPR 2019)

1 Introduction

StyleGAN - Stochastic Variation

- 같은 사람에 대한 이미지라 하더라도 수염, 주름 등 stochastic하다고 볼 수 있는 요소가 있는데, 이를 위해 각 layer마다 random noise를 추가하였다. 이렇게 stochastic한 정보를 따로 추가하면 더욱 사실적인 이미지를 생성하게 되고, input latent vector는 이미지의 중요한 정보를 표현하는 데에만 집중할 수 있게 된다.



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

1

Introduction

StyleGAN

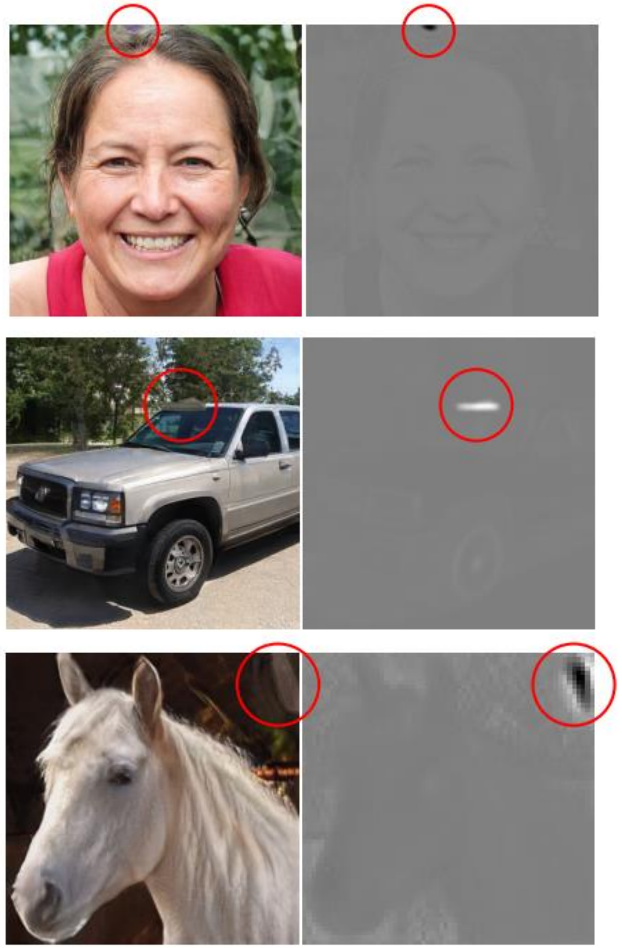
A Style-Based Generator Architecture for Generative Adversarial Networks (CVPR 2019)

2

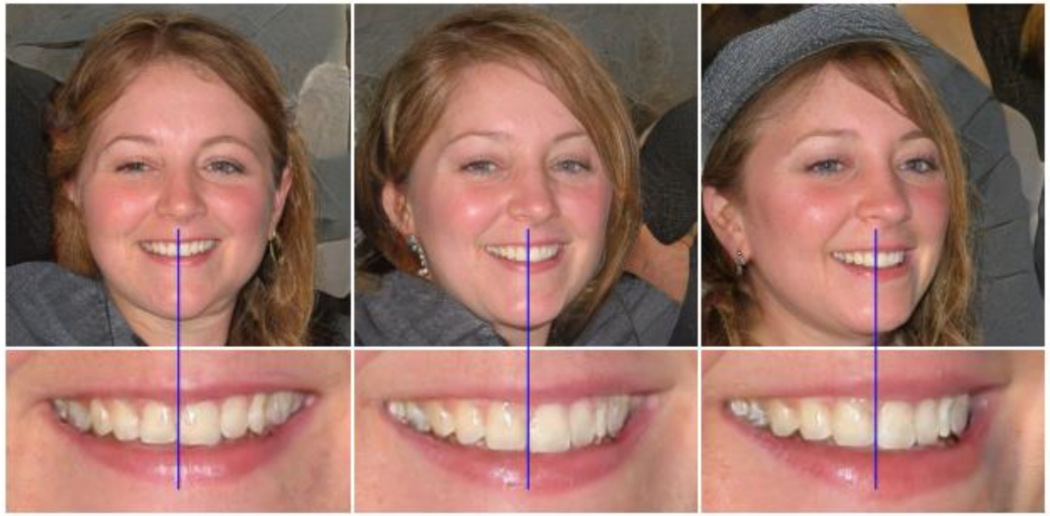
Motivation

StyleGAN's Problem

[Droplet-like Artifacts]



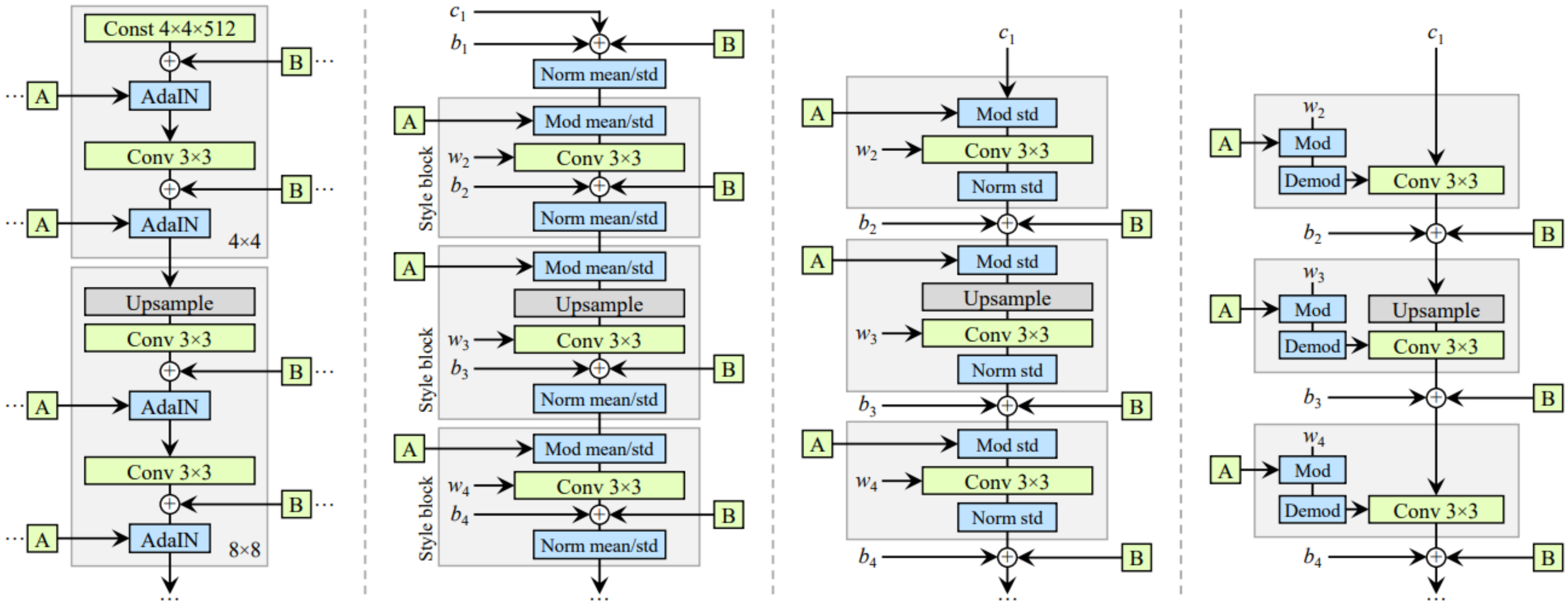
[Phase Artifacts]



3

Removing normalization artifacts

Redesigned architecture of the StyleGAN



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

3

Removing normalization artifacts

Redesigned architecture of the StyleGAN

[Modulation]

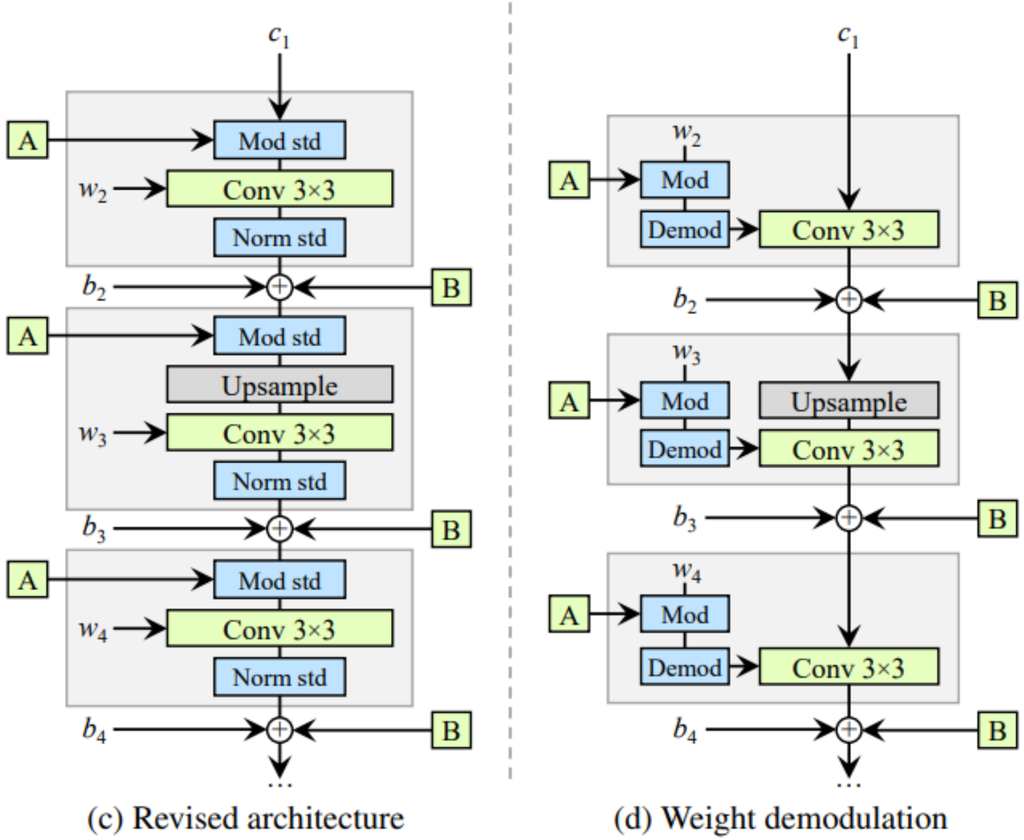
$$w'_{ijk} = s_i \cdot w_{ijk}$$

[After modulation and convolution]

$$\sigma_j = \sqrt{\sum_{i,k} {w'_{ijk}}^2}$$

[Demodulation]

$$w''_{ijk} = w'_{ijk} \Big/ \sqrt{\sum_{i,k} {w'_{ijk}}^2 + \epsilon}$$



3 Removing normalization artifacts

Removing artifacts with demodulation

Figure 3. Replacing normalization with demodulation removes the characteristic artifacts from images and activations.

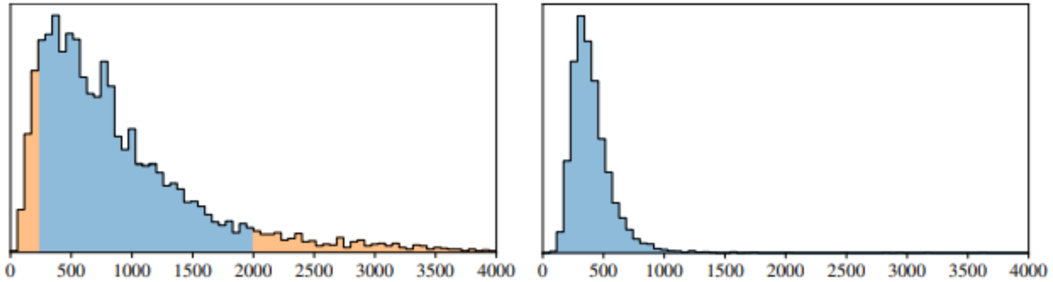
4 Image quality and generator smoothness

Perceptual Path Length

***PPL(Perceptual Path Length)** – a metric that was originally introduced for quantifying the smoothness of the mapping from a latent space to the output image by **measuring average LPIPS distances** between generated images under small perturbations in latent space.*

(a) Low PPL scores (b) High PPL scores

Figure 4. Connection between perceptual path length and image quality using baseline StyleGAN (config A in Table 1). (a) Random examples with low PPL ($\leq 10^{\text{th}}$ percentile). (b) Examples with high PPL ($\geq 90^{\text{th}}$ percentile). There is a clear correlation between PPL scores and semantic consistency of the images.



(a) Baseline StyleGAN (b) Our method (config F)

Figure 5. (a) Distribution of PPL scores of individual images generated using a baseline StyleGAN (config A in Table 1, FID = 8.53, PPL = 924). The percentile ranges corresponding to Figure 4 are highlighted in orange. (b) Our method (config F) improves the PPL distribution considerably (showing a snapshot with the same FID = 8.53, PPL = 387).

4 Image quality and generator smoothness

Lazy regularization & Path length regularization

[Lazy Regularization]

Observe that typically the regularization terms(e.g. R_1) can be computed much less frequently than the main loss function(e.g. logistic loss), thus greatly diminishing their computational cost and the overall memory usage. Table 1, row C shows that no harm is caused when ***R_1 regularization is performed only once every 16 minibatches****, and adopt the same strategy for new regularizer as well.*

[Path Length Regularization]

They consider a generator mapping from the latent space to image space to be well-conditioned if, at each point in latent space, small displacements yield changes of equal magnitude in image space regardless of the direction of perturbation. Motivated by the desire to preserve the expected lengths of vectors regardless of the direction, formulate regularizer as

$$\mathbb{E}_{\mathbf{w},\mathbf{y}\sim\mathcal{N}(0,\mathbf{I})}\left(\left\|\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y}\right\|_2 - a\right)^2$$

Configuration	FFHQ, 1024×1024 FID	Path length	Precision	Recall	LSUN Car, 512×384 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**

Table 1. Main results. For each training run, we selected the training snapshot with the lowest FID. We computed each metric 10 times with different random seeds and report their average. The "path length" column corresponds to the PPL metric, computed based on path endpoints in $\mathcal{W}$ [24]. For LSUN datasets, we report path lengths without the center crop that was originally proposed for FFHQ. The FFHQ dataset contains 70k images, and we showed the discriminator 25M images during training. For LSUN CAR the corresponding numbers were 893k and 57M.

5

Progressive growing revisited

Multi-scale gradient GAN (MSG-GAN)

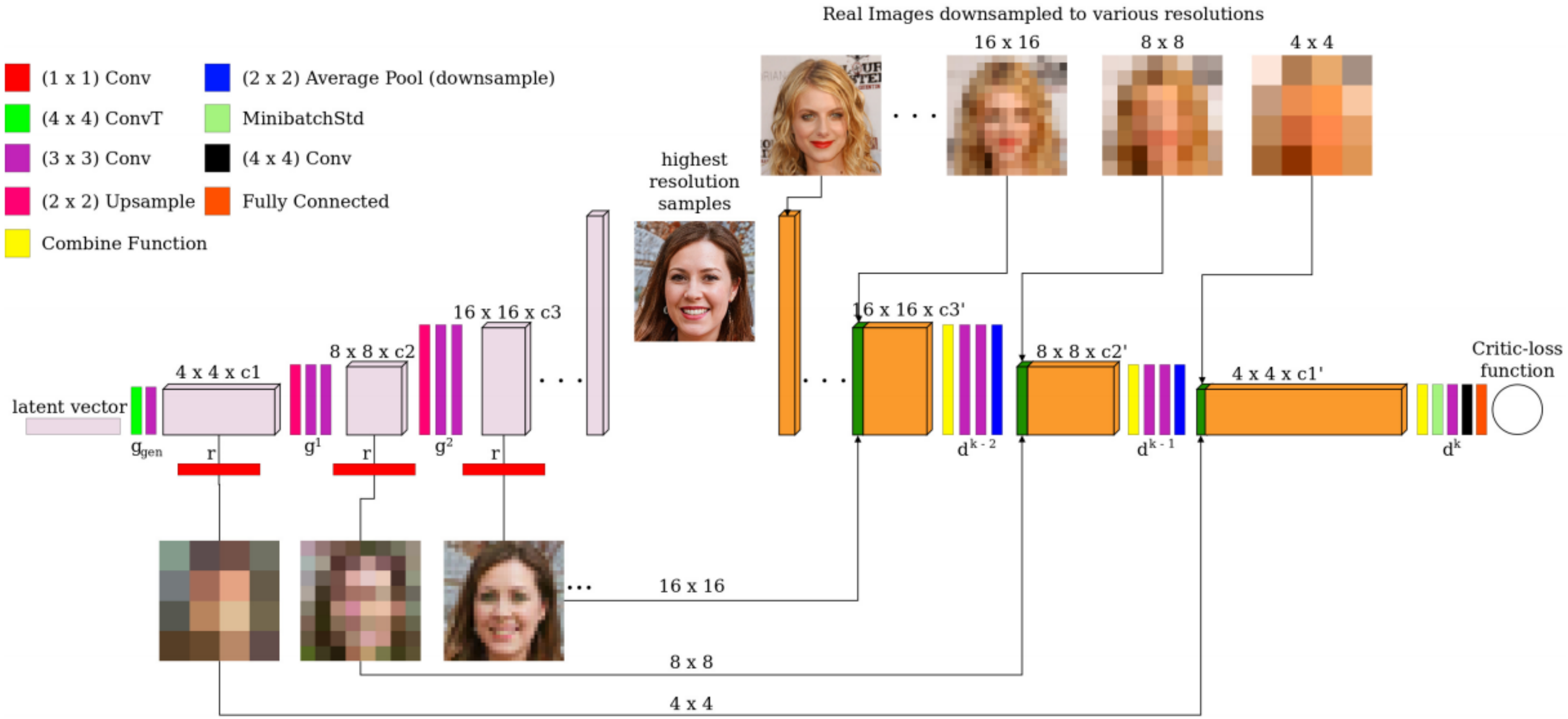
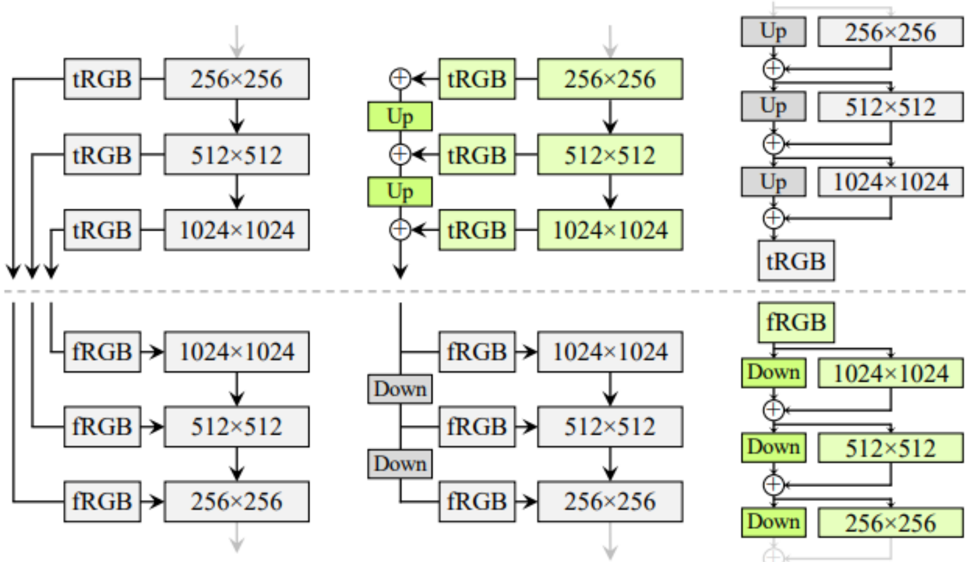


Figure 2: Architecture of MSG-GAN, shown here on the base model proposed in ProGANs [13]. Our architecture includes connections from the intermediate layers of the generator to the intermediate layers of the discriminator. Multi-scale images sent to the discriminator are concatenated with the corresponding activation volumes obtained from the main path of convolutional layers followed by a combine function (shown in yellow).

5 Progressive growing revisited

Alternative network architectures



(a) MSG-GAN (b) Input/output skips (c) Residual nets

Figure 7. Three generator (above the dashed line) and discriminator architectures. Up and Down denote bilinear up and downsampling, respectively. In residual networks these also include 1×1 convolutions to adjust the number of feature maps. tRGB and fRGB convert between RGB and high-dimensional per-pixel data. Architectures used in configurations E and F are highlighted in green.

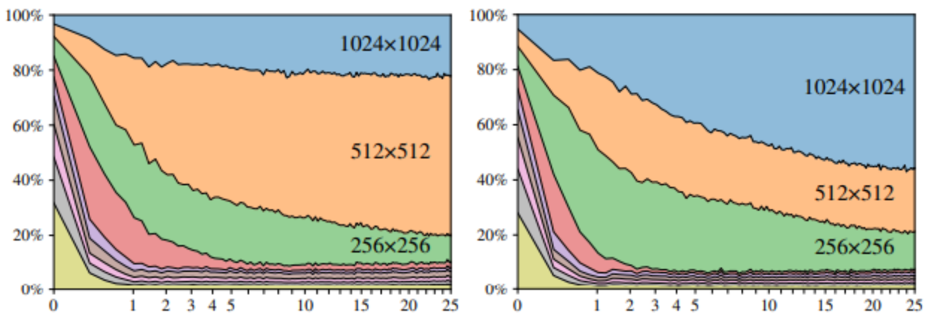
FFHQ	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	4.32	237	4.18	207	3.58	238
G output skips	4.33	149	3.77	**116**	**3.31**	117
G residual	4.35	187	3.96	201	3.79	203

LSUN Car	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	3.75	905	3.23	758	3.25	802
G output skips	3.77	544	3.86	**316**	3.19	471
G residual	3.93	981	3.40	667	**2.66**	645

Table 2. Comparison of generator and discriminator architectures without progressive growing. The combination of generator with output skips and residual discriminator corresponds to configuration E in the main result table.

5 Progressive growing revisited

Resolution usage



(a) StyleGAN-sized (config E) (b) Large networks (config F)

Figure 8. Contribution of each resolution to the output of the generator as a function of training time. The vertical axis shows a breakdown of the relative standard deviations of different resolutions, and the horizontal axis corresponds to training progress, measured in millions of training images shown to the discriminator. We can see that in the beginning the network focuses on low-resolution images and progressively shifts its focus on larger resolutions as training progresses. In (a) the generator basically outputs a 512^2 image with some minor sharpening for 1024^2, while in (b) the larger network focuses more on the high-resolution details.

Dataset	Resolution	StyleGAN (A)		Ours (F)	
		FID	PPL	FID	PPL
LSUN CAR	512×384	3.27	1485	**2.32**	**416**
LSUN CAT	256×256	8.53	924	**6.93**	**439**
LSUN CHURCH	256×256	4.21	742	**3.86**	**342**
LSUN HORSE	256×256	3.83	1405	**3.43**	**338**

Table 3. Improvement in LSUN datasets measured using FID and PPL. We trained CAR for 57M images, CAT for 88M, CHURCH for 48M, and HORSE for 100M images.