Mahidol University
Faculty of Medicine Ramathibodi Hospital
Department of Clinical Epidemiology and Biostatistics

Journal club : May 20,2022

GAN & StyleGAN

Ekapob Sangariyavanich
Ph.D. student in Data Science for Healthcare and Clinical Informatics program
Year 2020

Topic

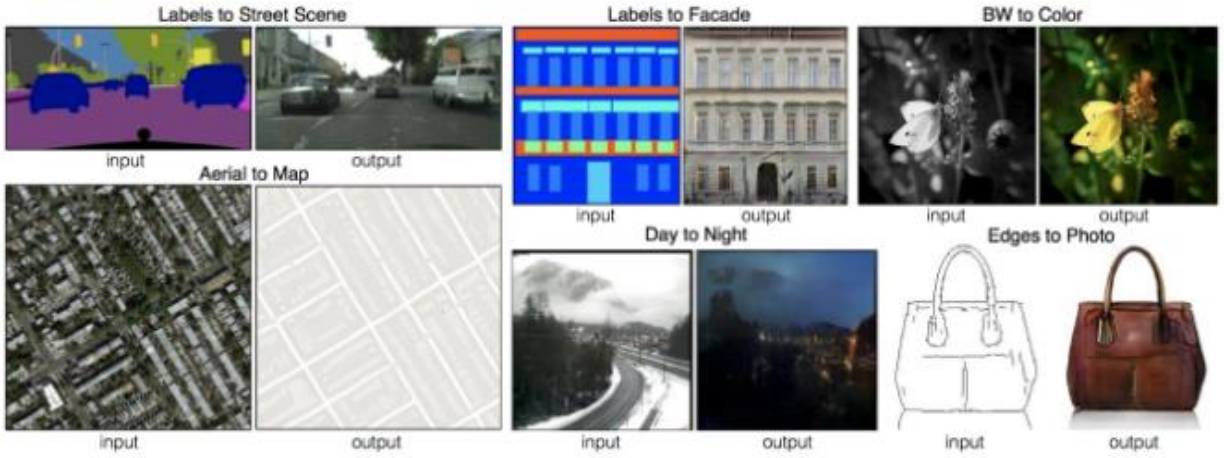
- GAN
  - GAN structure
  - Application
  - Common problem
  - GAN evaluation
- StyleGAN
- StyleGAN2

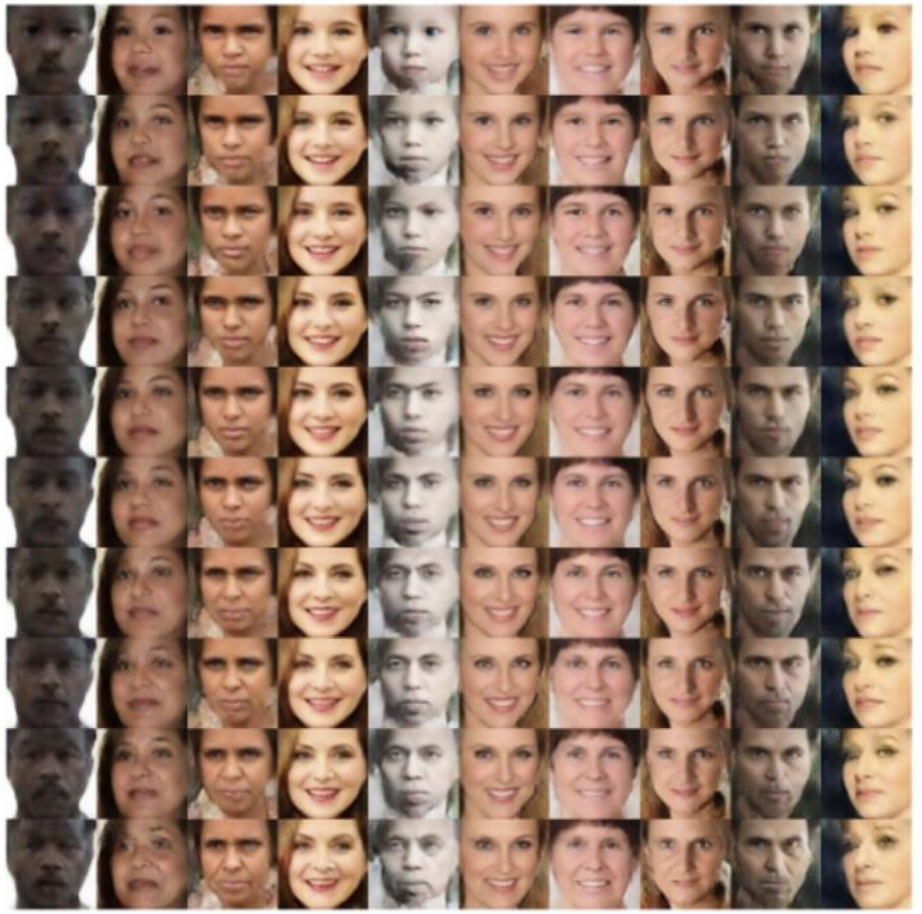
https://thispersondoesnotexist.com/

Comparing original and deepfake videos of Russian president Vladimir Putin. Photograph: Alexandra Robinson/AFP via Getty Images

DeepFakes !

Generative Adversarial Networks (GAN)



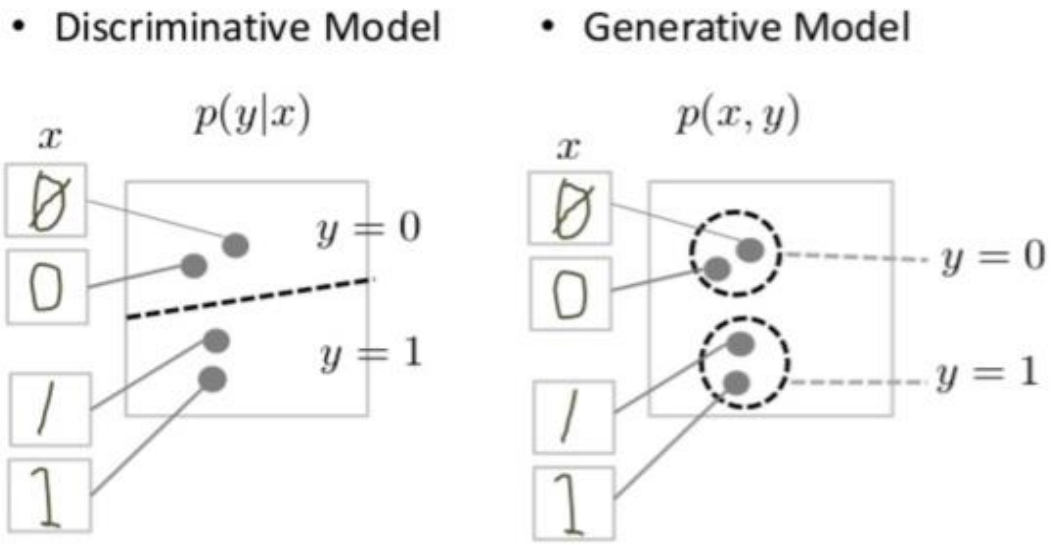


Source : https://towardsdatascience.com/generative-adversarial-networks-explained-34472718707a

Generative model VS Discriminative model

- **Generative** models can generate new data instances.
- **Discriminative** models discriminate between different kinds of data instances.

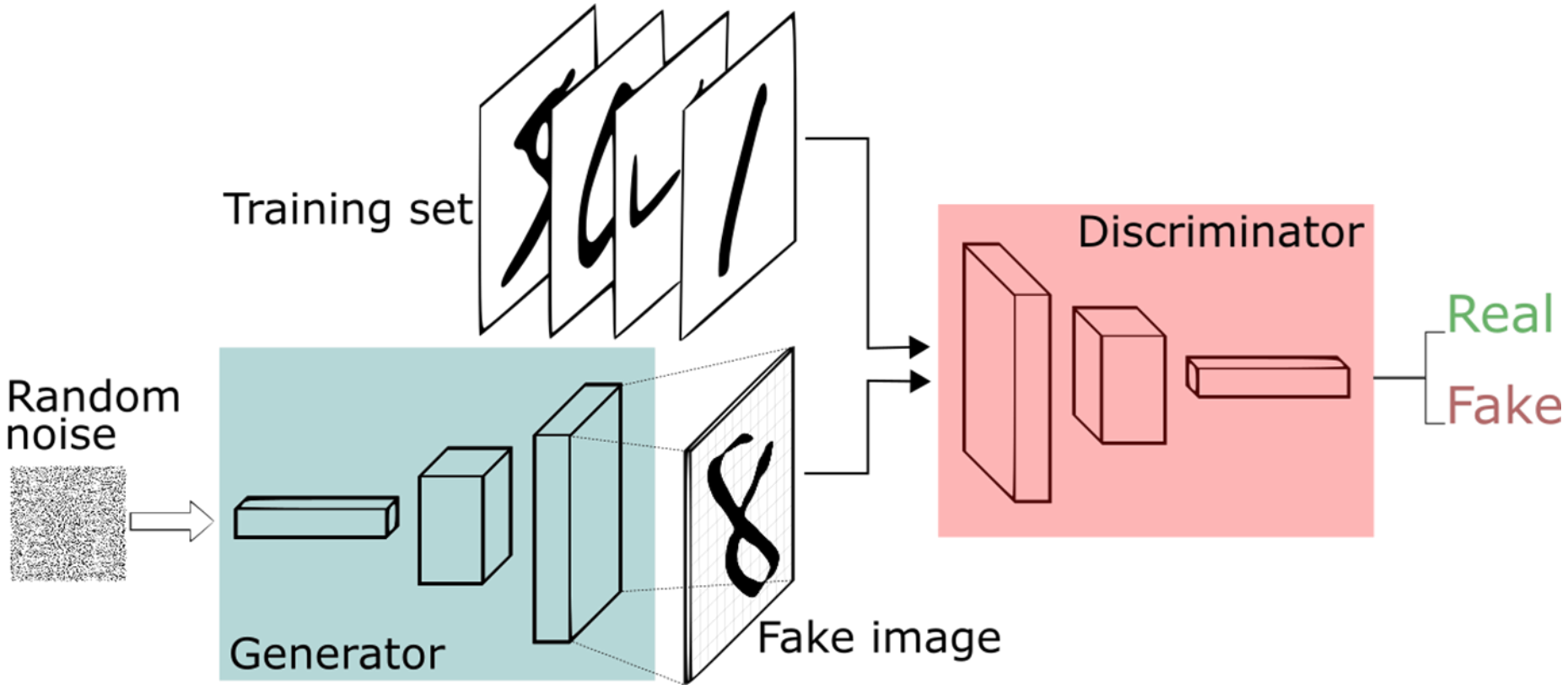
Tell the difference between handwritten 0's and 1's



Produce convincing 1's and 0's by generating digits that fall close to their real counterparts in the data space

https://developers.google.com/machine-learning/gan/generative#:~:text=A%20generative%20model%20includes%20the,to%20a%20sequence%20of%20words

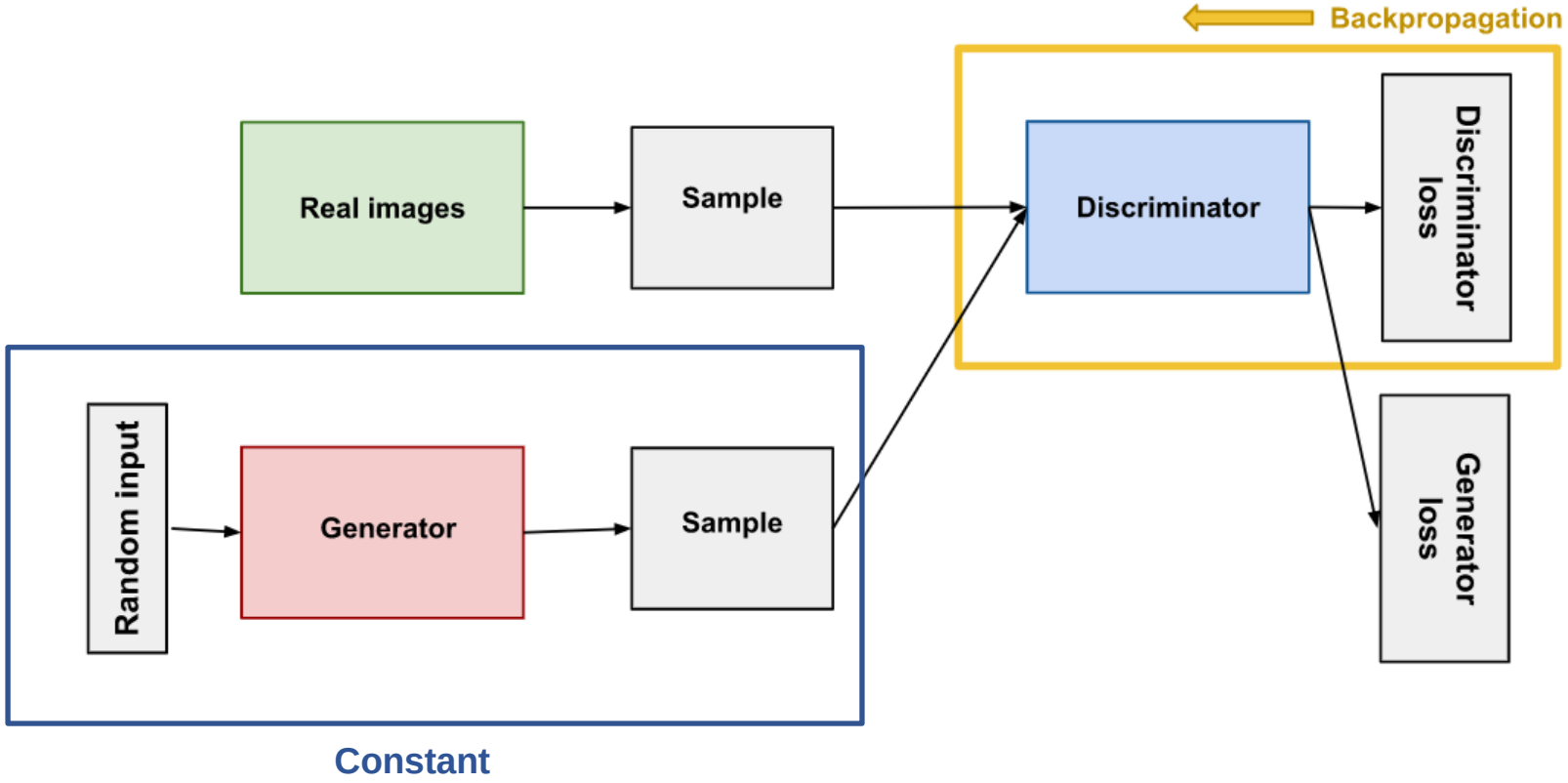
GAN structure



https://towardsdatascience.com/generative-adversarial-networks-explained-34472718707a

Generator Win !

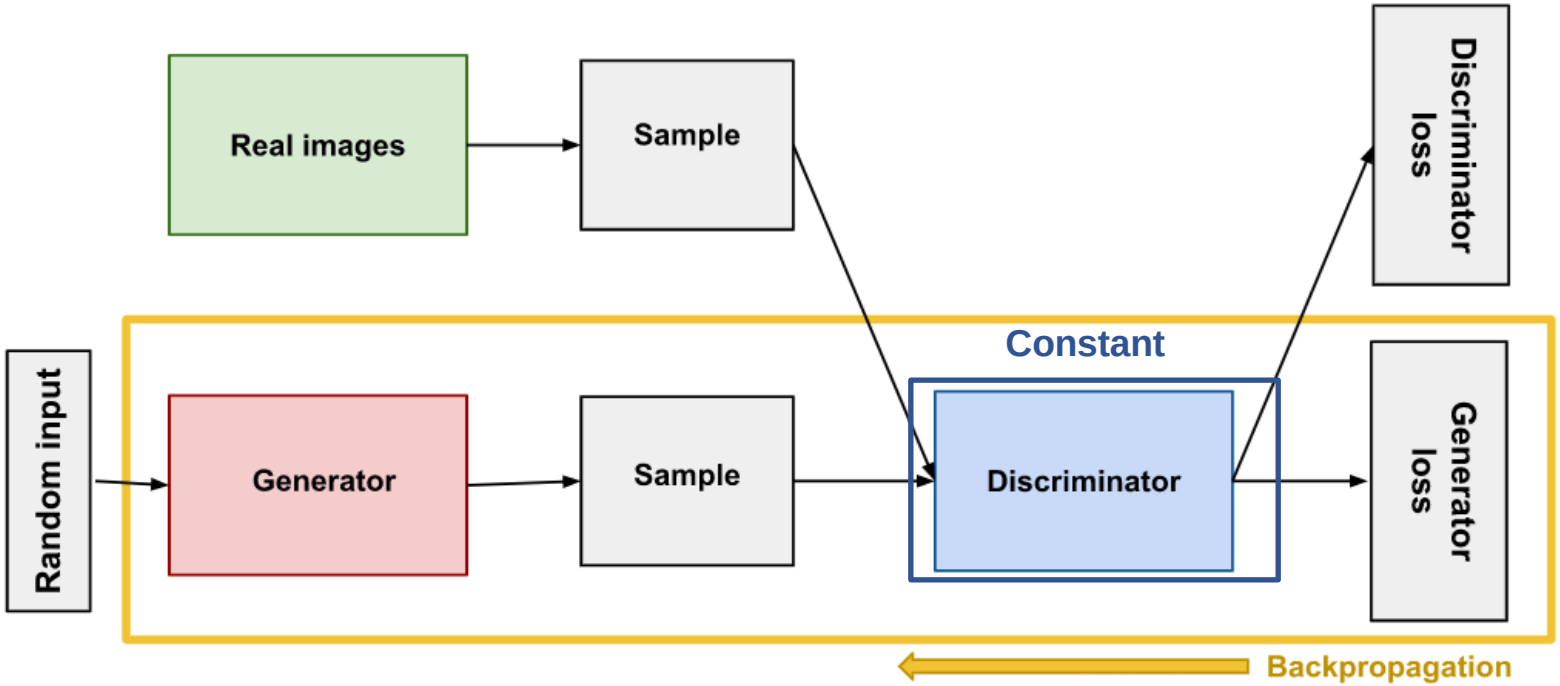
Discriminator loss : predict fake sample from generator as real sample



https://developers.google.com/machine-learning/gan/discriminator

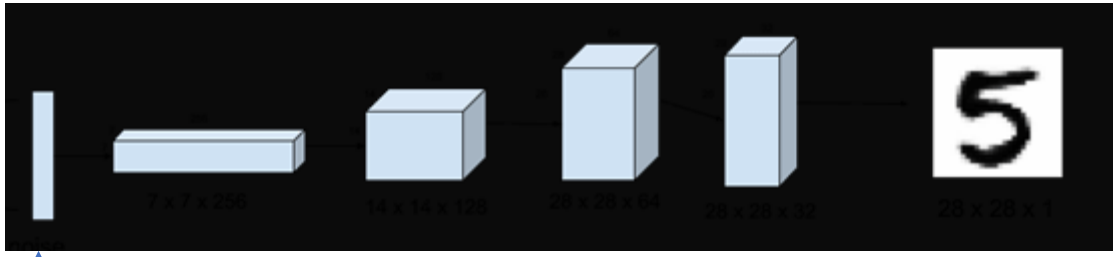
Discriminator Win !

Generator loss : predict fake sample from generator as fake sample



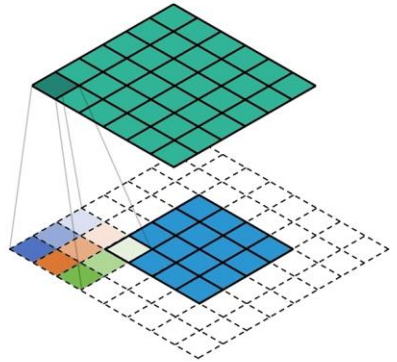
https://developers.google.com/machine-learning/gan/generator

Generator



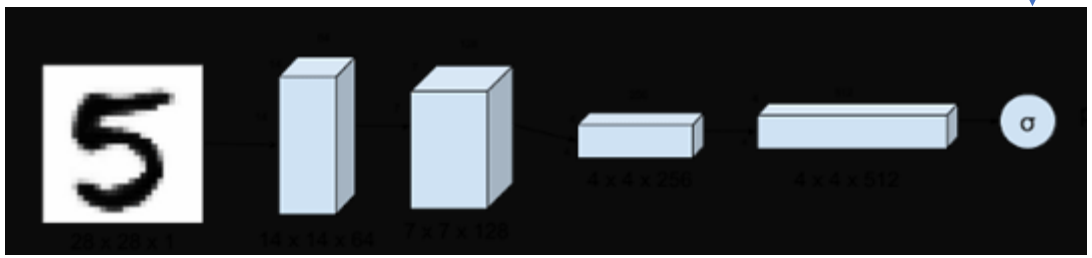
Several Transpose convolutional layer
(Deconvolution)

Latent vector

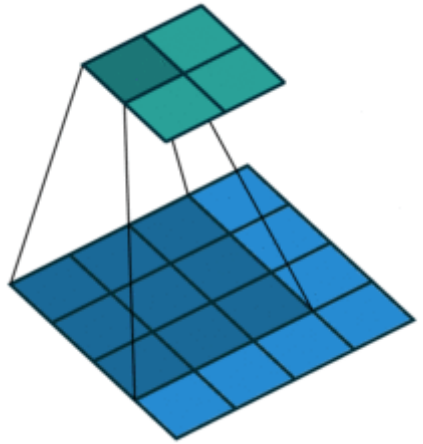


Prediction

Discriminator
(Classifier)



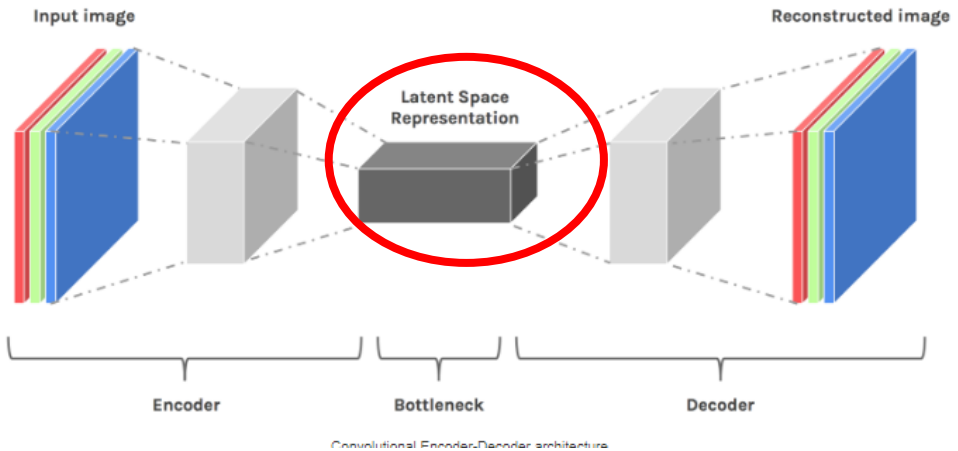
(Convolution)



https://affine.ai/progan-stylegan-stylegan2-exploring-nvidias-breakthroughs/

Latent space

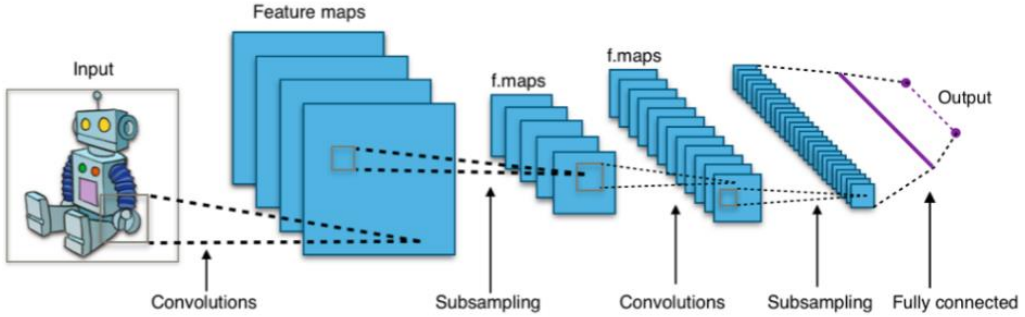
- An *abstract multi-dimensional space* containing feature values that we cannot interpret directly, but which encodes a meaningful internal representation of externally observed events.
- Latent space in GANS : a Gaussian distribution with mean 0 and standard deviation 1. which means, the numbers range from -1 to 1.
- The input RGB image will be remapped from 0–255 to -1 to 1.

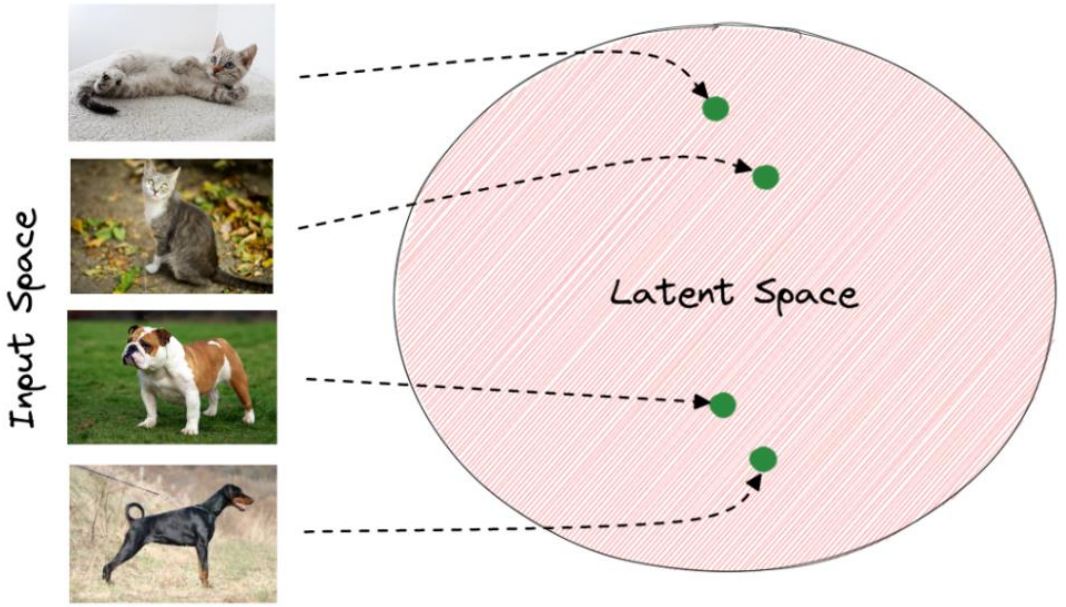


Convolutional Encoder-Decoder architecture

https://hackernoon.com/latent-space-visualization-deep-learning-bits-2-bd09a46920df

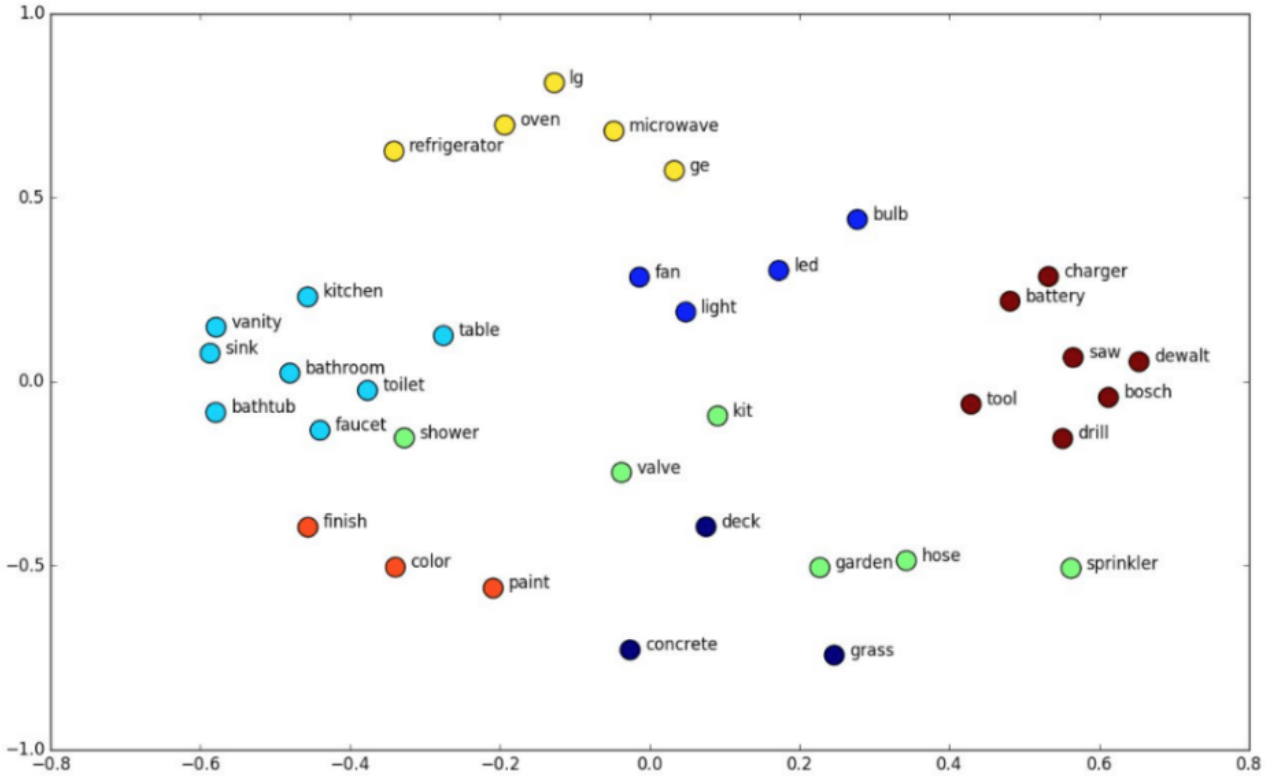
- Latent space in image



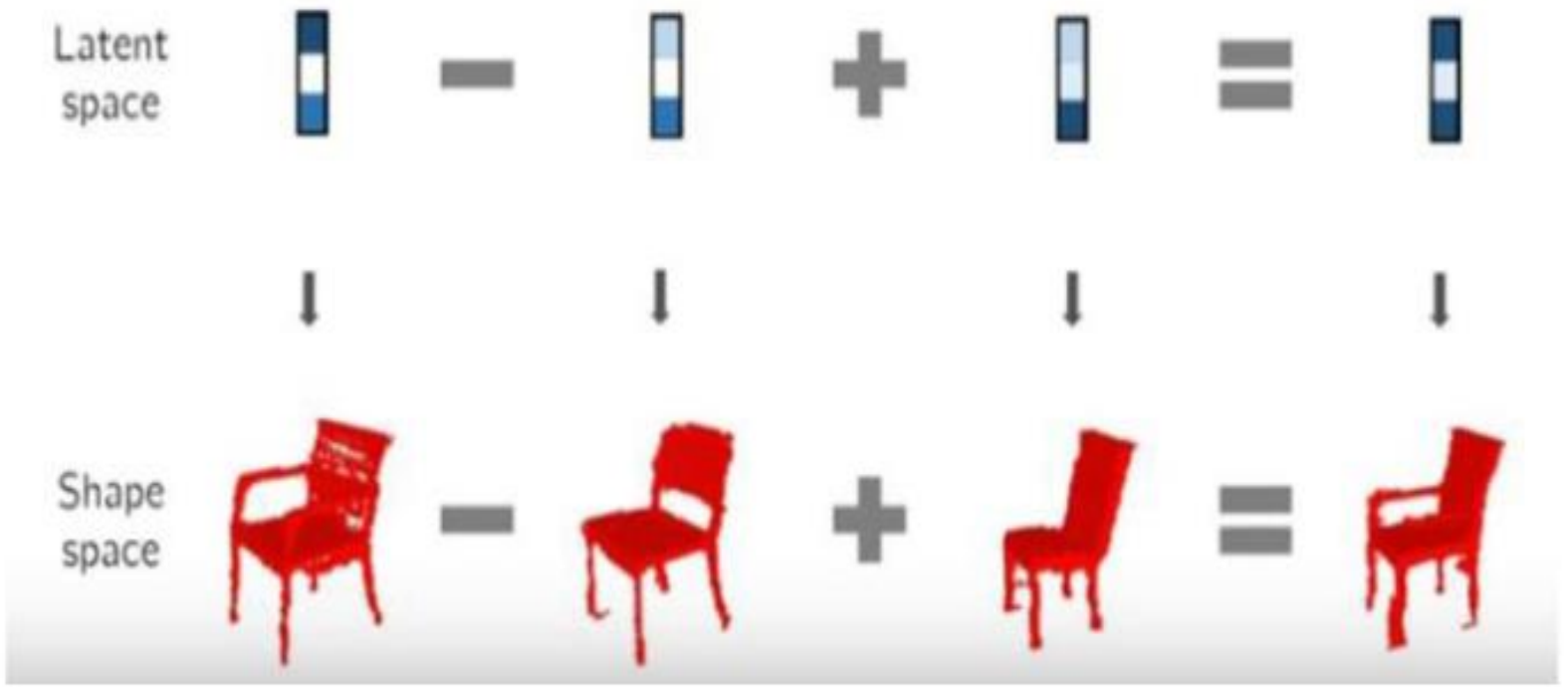


https://www.baeldung.com/cs/dl-latent-space

- Word embedding space



https://www.baeldung.com/cs/dl-latent-space



Latent space can be added and subtracted to produce intermediate results

Source: https://medium.com/@jain.yasha/gan-latent-space-1b32cd34cfda

LATENT SPACE WALK

While walking between two latent spaces, we can generate images from all intermediate points of latent space

Source: https://medium.com/@jain.yasha/gan-latent-space-1b32cd34cfda

GAN loss function

- Zero sum games
- Loss function : Minimax loss

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

- Generator tries to minimize while Discriminator tries to maximize this function

Common problem in GAN training

- Non-convergence
  - Generator dominates
  - Discriminator dominates

- Mode collapse :

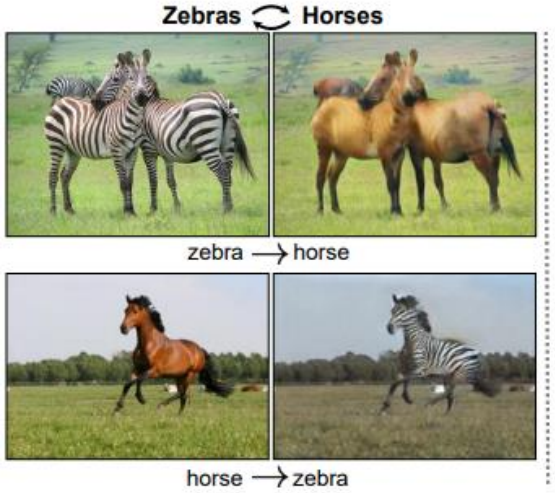
  Generator produce limited variety of samples

- Vanishing gradient :

  an optimal discriminator doesn't provide enough information for the generator to make progress.

Application of GAN

- Generate examples for Image Datasets
- Image-to-Image Translation
- Text-to-Image Translation
- Face Aging
- Video Prediction
- Etc.



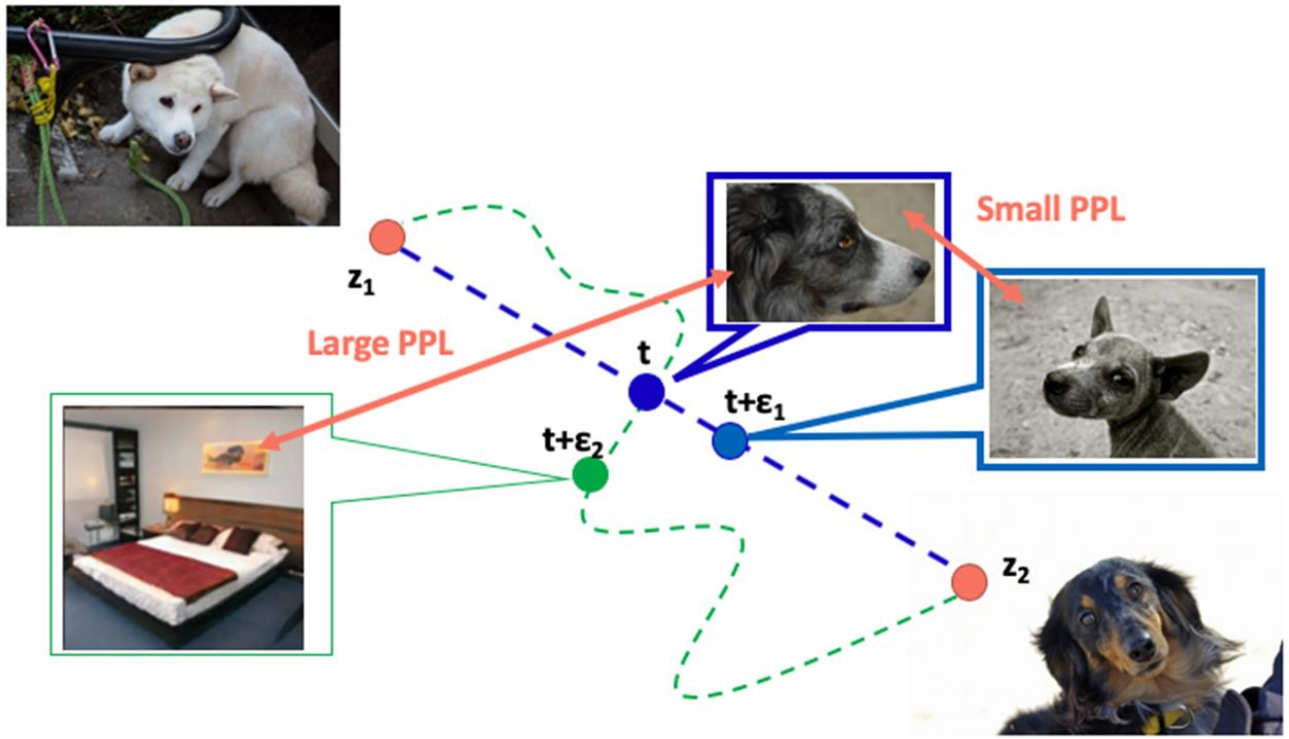
Metric for evaluate GAN

- **Frechet Inception Distance (FID)**

  - one of the most common automated metrics used to evaluate images.
  - Comparing the activations of inception V3 model (the last pooling layer) statistics (Mean & covariance) on real and generated images.
  - Lower is better

- **Perceptual Path length (PPL)**

- Perceptual Path Length (PPL) is an indicator of whether the image changes smoothly in “perceptual”.



(a) Low PPL scores

(b) High PPL scores

https://medium.com/analytics-vidhya/from-gan-basic-to-stylegan2-680add7abe82

StyleGAN

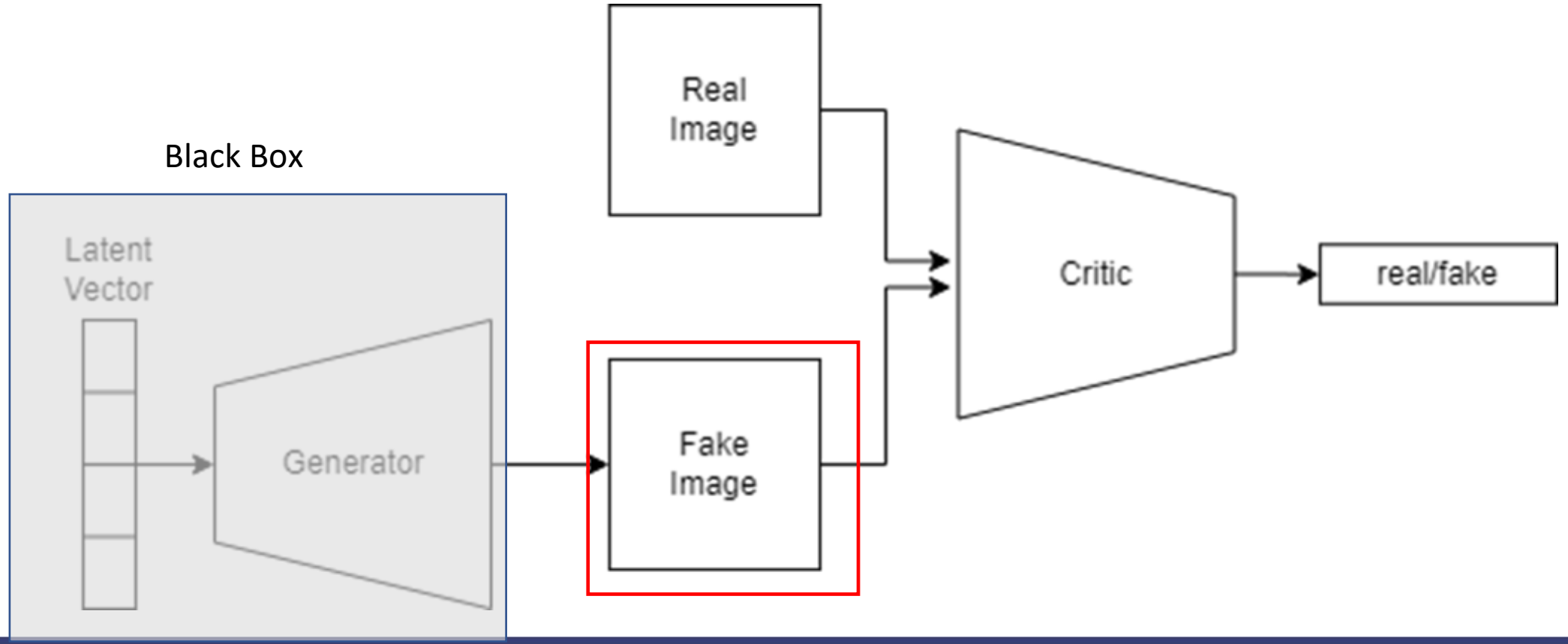
- Style GAN

Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2019 (pp. 4401-4410).

- Style GAN 2 :

Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. Analyzing and improving the image quality of stylegan. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020 (pp. 8110-8119).

How to control generated images ?



StyleGAN (2019)

A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras
NVIDIA
tkarras@nvidia.com

Samuli Laine
NVIDIA
slaine@nvidia.com

Timo Aila
NVIDIA
taila@nvidia.com

- Generate photorealistic high-quality photos
- Offer controls over the style of the generated images

StyleGAN 1

1. Progressive growing GANS
2. Mapping Network and Synthetic network (AdaIN) (Control style)
3. Addition of gaussian noise

1.Progressive growing GANs – high resolution images

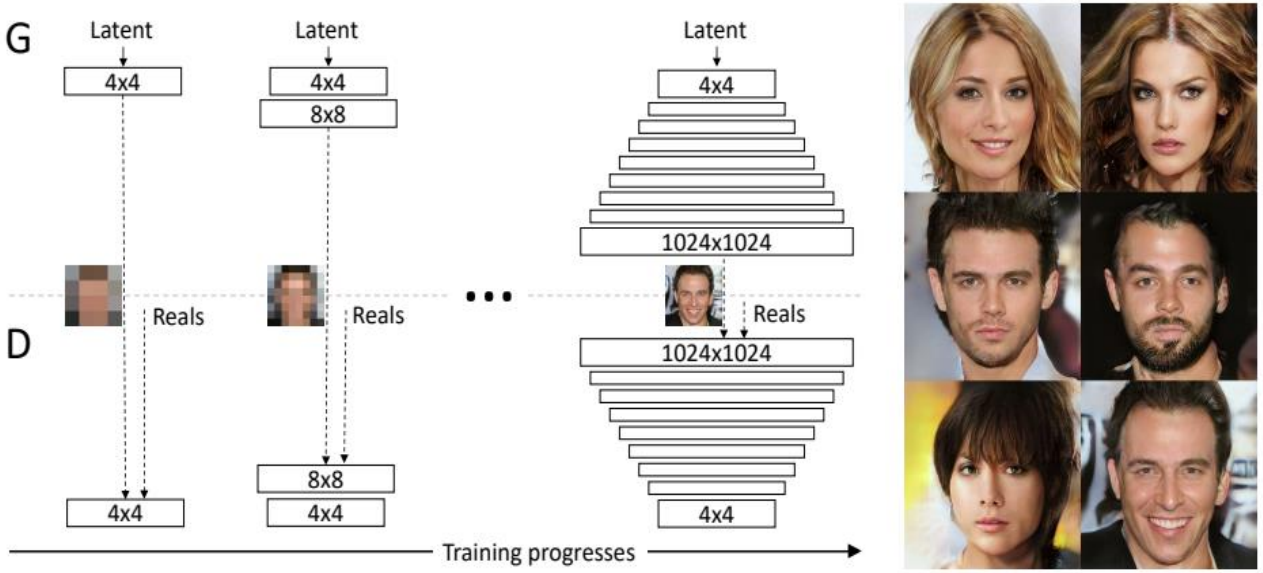
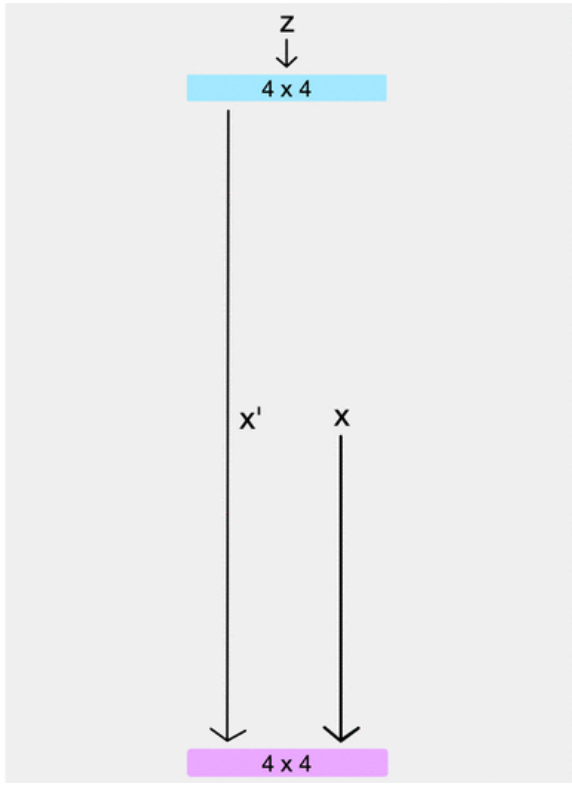
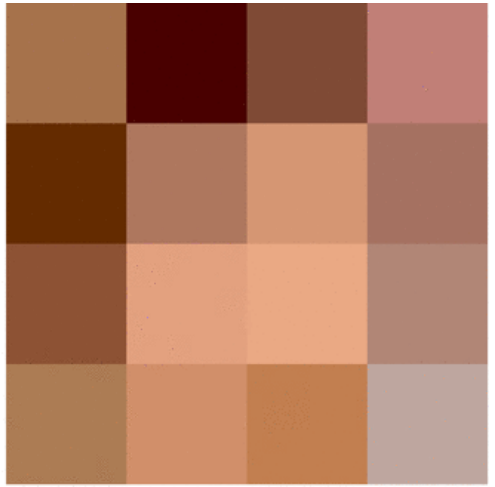


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout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





Training time: 0 days
4x4 resolution

Generator
Discriminator

z = random code
x = real image
x' = generated image

Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196. 2017 Oct 27.

2. Mapping Network and Synthesis network

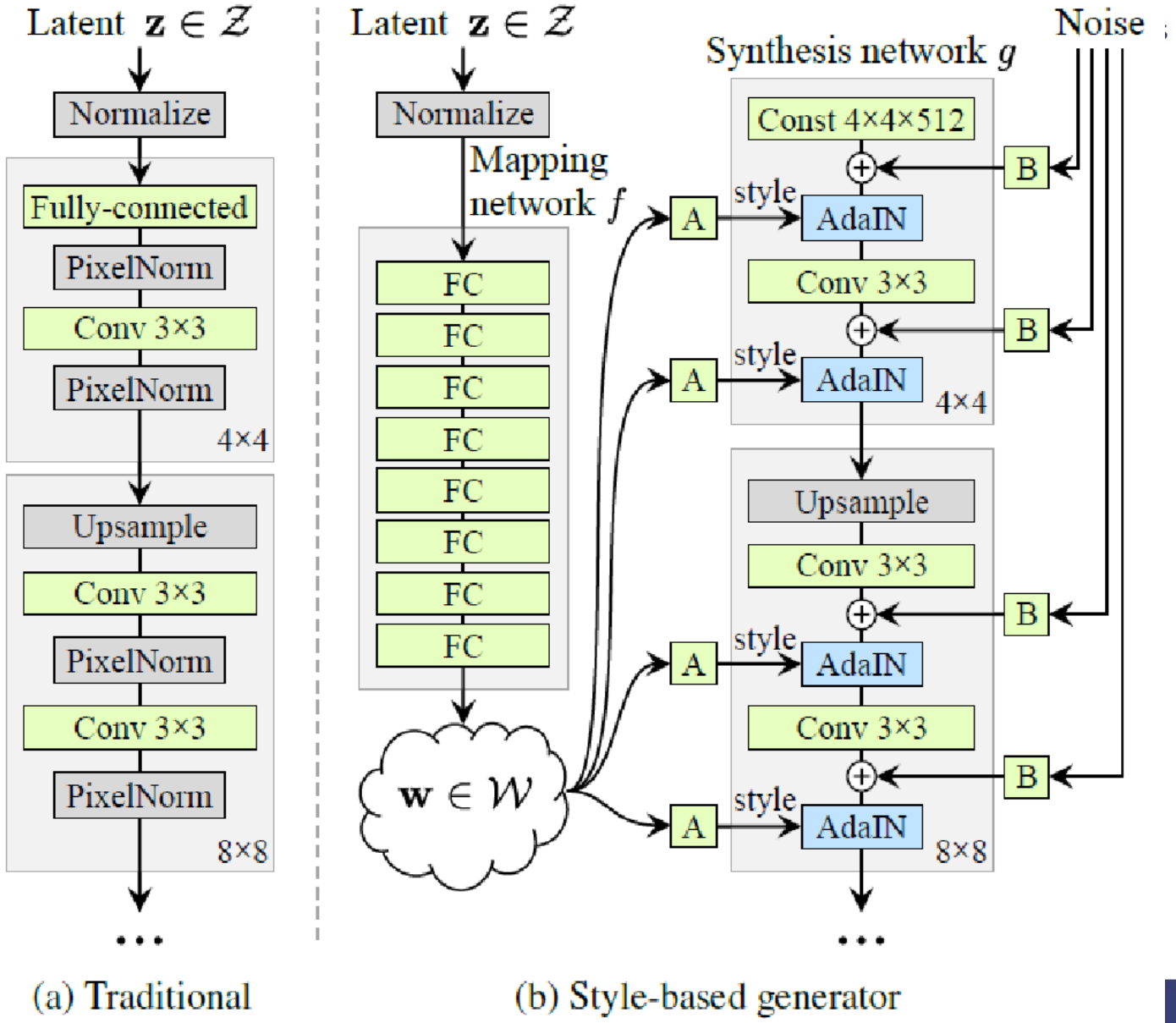
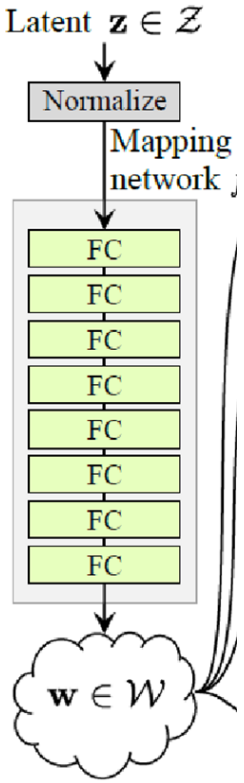


Figure 1. While a traditional generator [30] feeds the latent code though the input layer only, we first map the input to an intermediate latent space $\mathcal{W}$, which then controls the generator through adaptive instance normalization (AdaIN) at each convolution layer. Gaussian noise is added after each convolution, before evaluating the nonlinearity. Here "A" stands for a learned affine transform, and "B" applies learned per-channel scaling factors to the noise input. The mapping network f consists of 8 layers and the synthesis network g consists of 18 layers — two for each resolution ($4^2 - 1024^2$). The output of the last layer is converted to RGB using a separate 1×1 convolution, similar to Karras et al. [30]. Our generator has a total of 26.2M trainable parameters, compared to 23.1M in the traditional generator.

z= 512-dimension vector



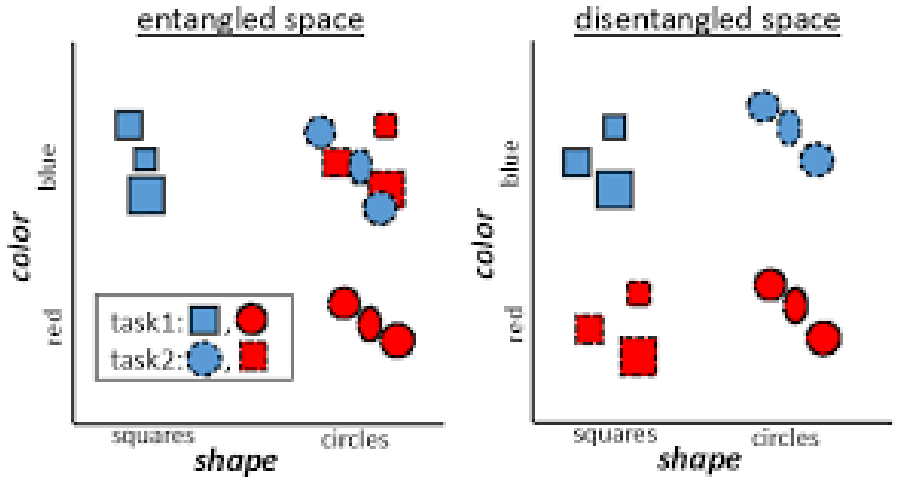
8 layers MLP

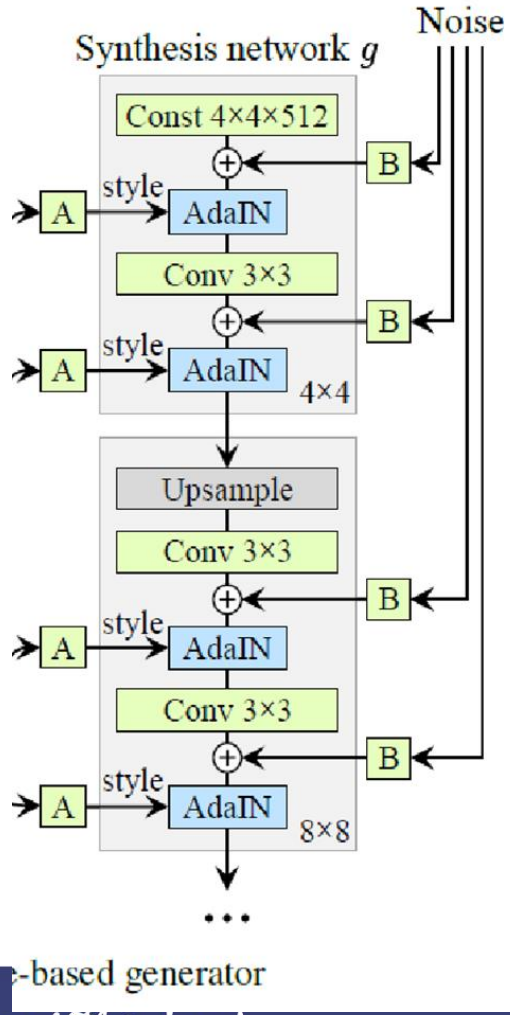
w= 512-dimension vector

(b) Sty

- Mapping network

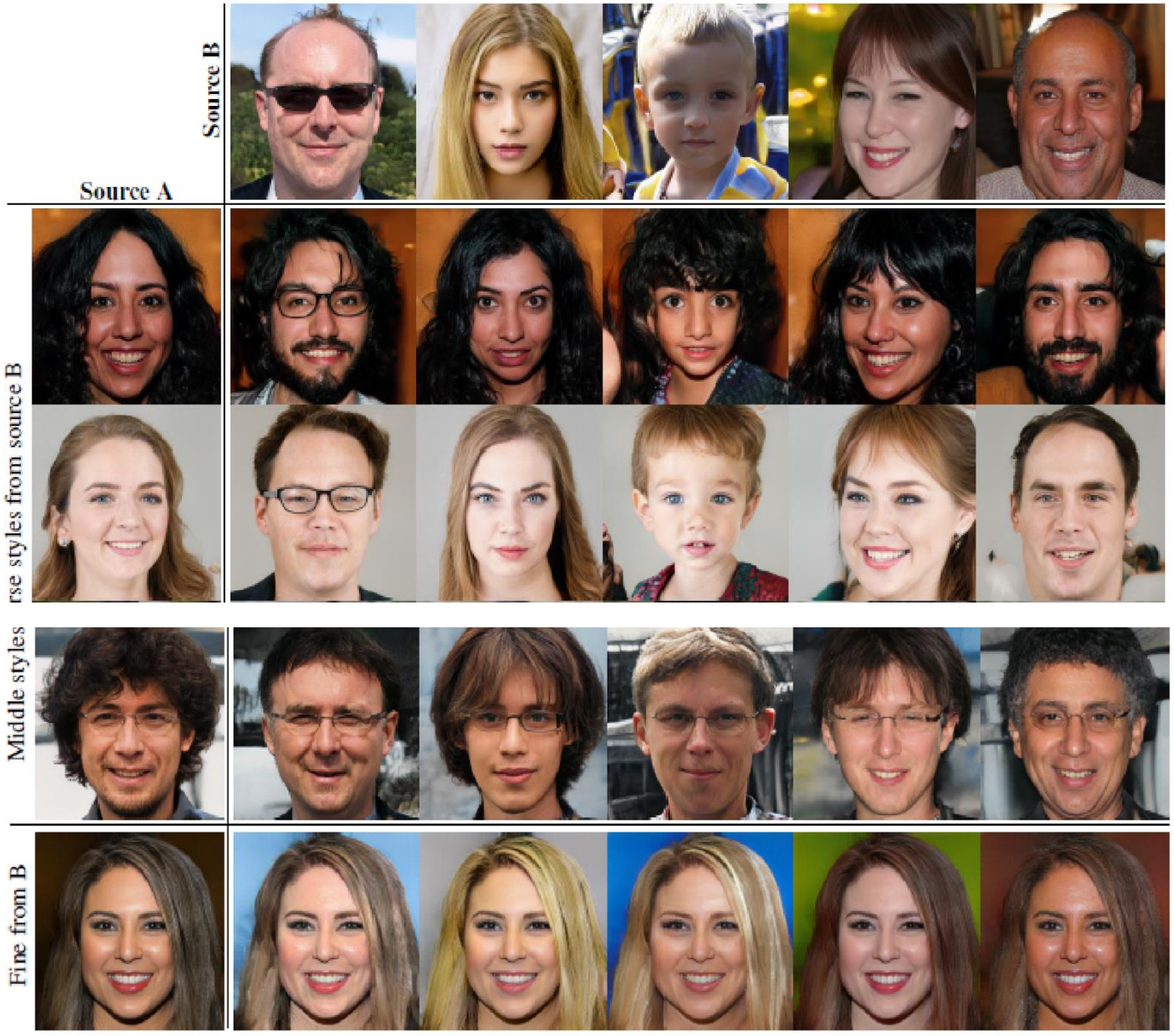
- enforces a sort of disentangled representation of latent vector





-based generator

- Synthesis network
  - 18 layers consisted of 2 of each resolution
  - 4x4,8x8,16x16,……,1024x1024
  - The learned intermediate latent space(A) that corresponds to style is injected.
  - AdaIN : adaptive instance normalization



Control styles

Course styles (4^2-8^2)

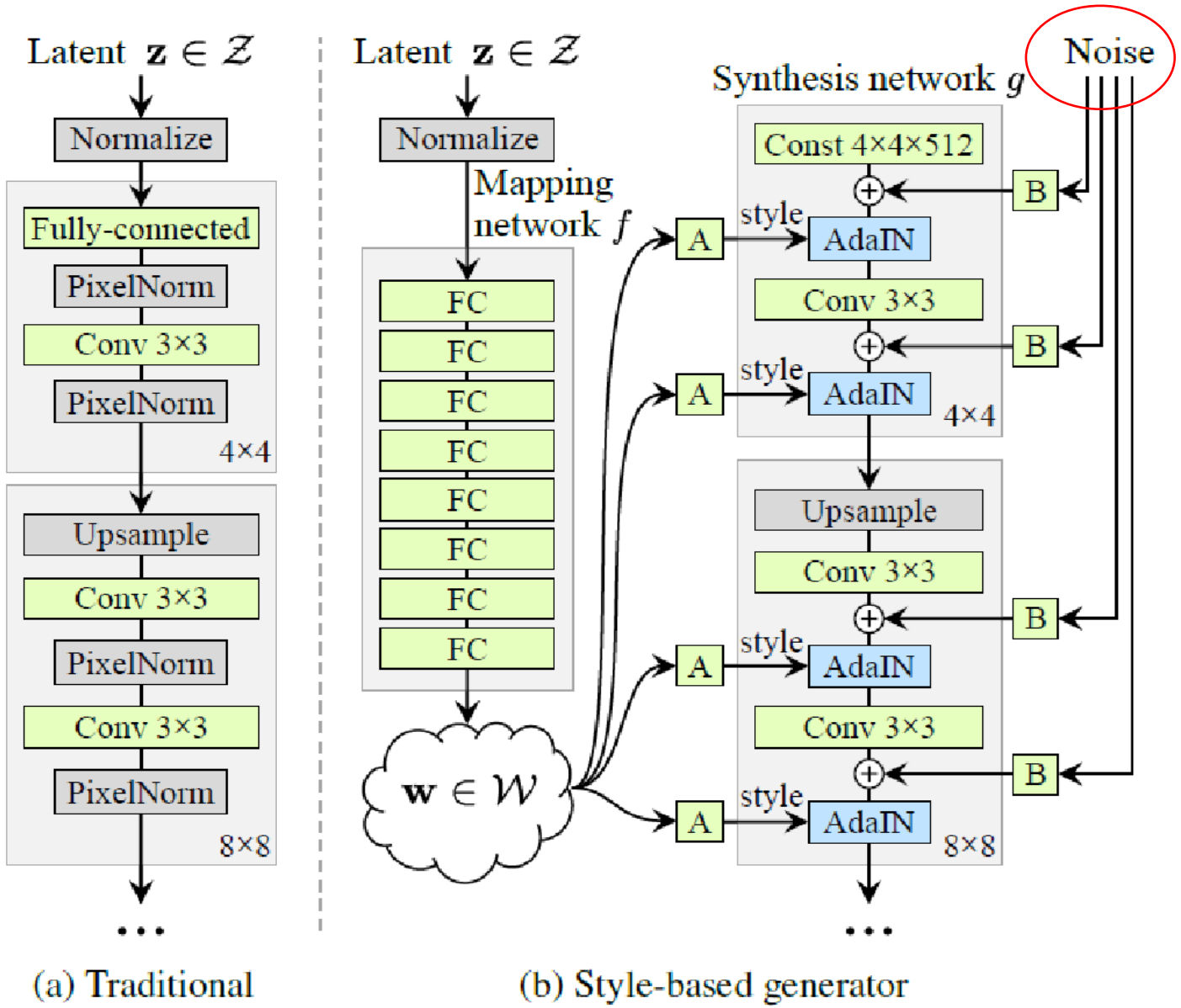
--pose , hair , face , shape

Middle styles (16^2-32^2)

-- facial features , eye

Fine styles (64^2-1024^2)

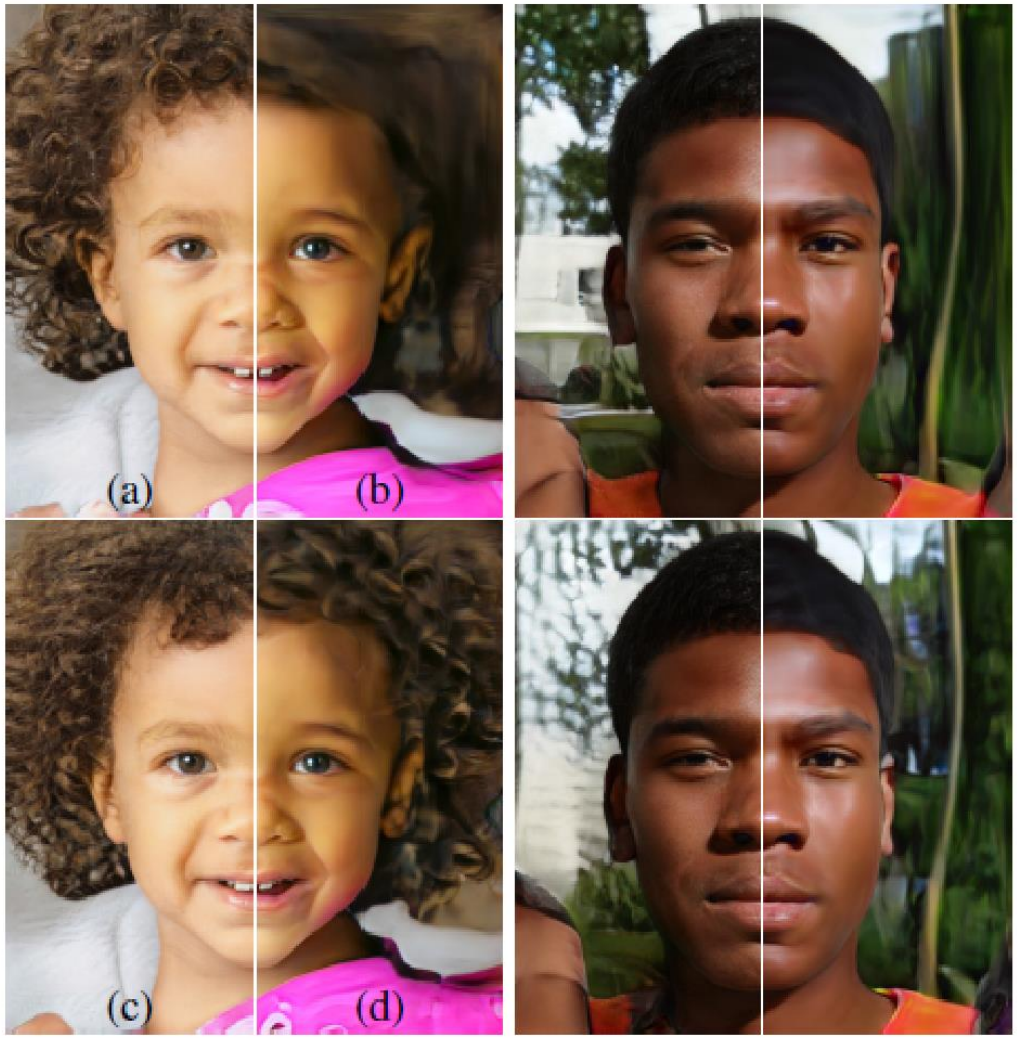
– color scheme



(a) Traditional (b) Style-based generator

3. Addition of gaussian noise

Figure 1. While a traditional generator [30] feeds the latent code though the input layer only, we first map the input to an intermediate latent space $\mathcal{W}$, which then controls the generator through adaptive instance normalization (AdaIN) at each convolution layer. Gaussian noise is added after each convolution, before evaluating the nonlinearity. Here "A" stands for a learned affine transform, and "B" applies learned per-channel scaling factors to the noise input. The mapping network f consists of 8 layers and the synthesis network g consists of 18 layers — two for each resolution ($4^2 - 1024^2$). The output of the last layer is converted to RGB using a separate 1×1 convolution, similar to Karras et al. [30]. Our generator has a total of 26.2M trainable parameters, compared to 23.1M in the traditional generator.



(a) Noise in all layer

(b) No noise

(c) Noise in fine layers (64^2 – 1024^2)

(d) Noise in course layers (4^2 – 32^2)

Additionally, our generator automatically separates inconsequential variation from high-level attributes (pose, identity, etc.)

- Coarse noise ⟶ large-scale curling of hair
- Fine noise ⟶ finer details, texture
- No noise ⟶ featureless “painterly” look

Analyzing and Improving the Image Quality of StyleGAN

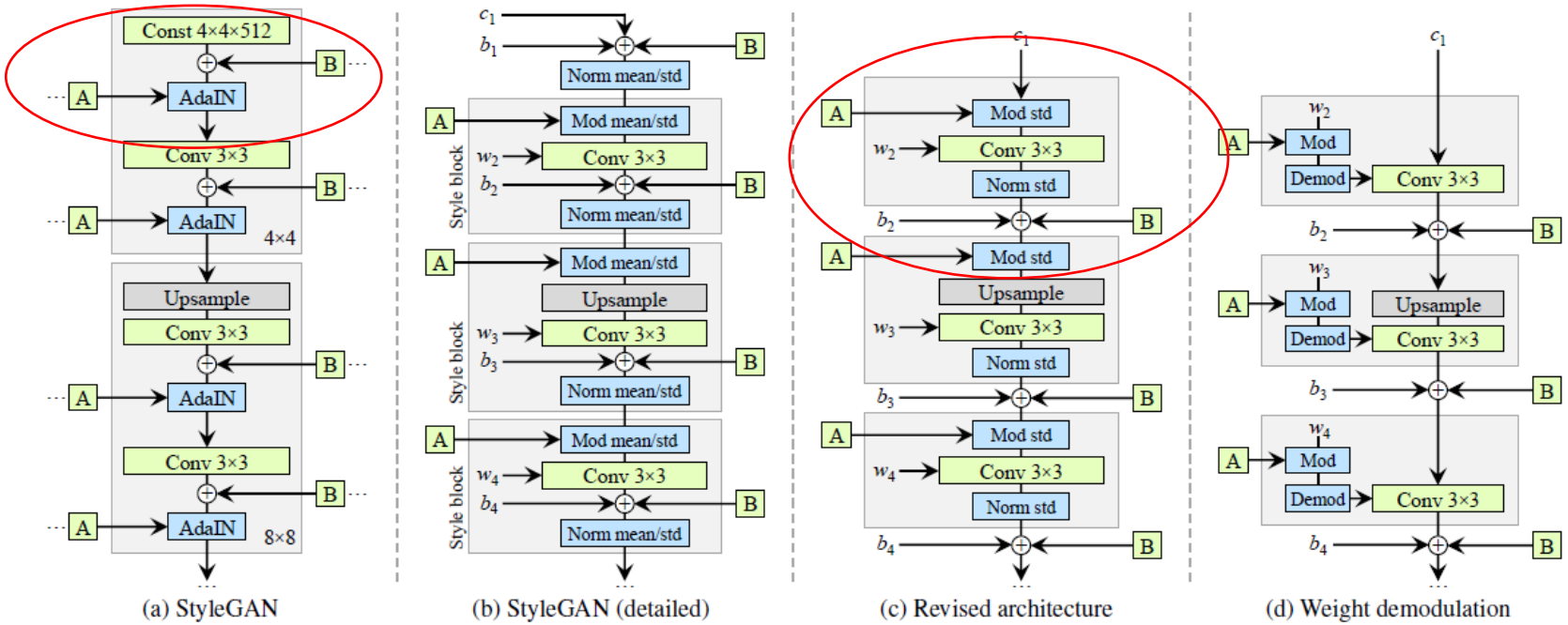
Tero Karras NVIDIA
Samuli Laine NVIDIA
Miika Aittala NVIDIA
Janne Hellsten NVIDIA
Jaakko Lehtinen NVIDIA and Aalto University
Timo Aila NVIDIA

- StyleGAN2 (2020)

  - Address styleGAN1 weakness → the artifacts starting from 64x64 resolution

StyleGAN2 : Improvement from StyleGAN

1.Weight demodulation



- Replaced the AdaIN with the Modulation and Normalization.
- Addition Bias and noise outside of the block

- StyleGAN1

- StyleGAN2 : demodulation

2. Remove progressive growing
   - Progressive growing lead to phase artifact
   - Implements a new network (MSG-GAN) design based on skip connection/residual nature like ResNet to produce high resolution images.

Figure 6. Progressive growing leads to "phase" artifacts. In this example the teeth do not follow the pose but stay aligned to the camera, as indicated by the blue line.

3. New type of regularization : lazy and path length regularization

Result

	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	–	–	–	–

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. *Path length* corresponds to the PPL metric, computed based on path endpoints in $\mathcal{W}$ [24], without the central crop used by Karras et al. [24]. The FFHQ dataset contains 70k images, and the discriminator saw 25M images during training. For LSUN CAR the numbers were 893k and 57M. ↑ indicates that higher is better, and ↓ that lower is better.

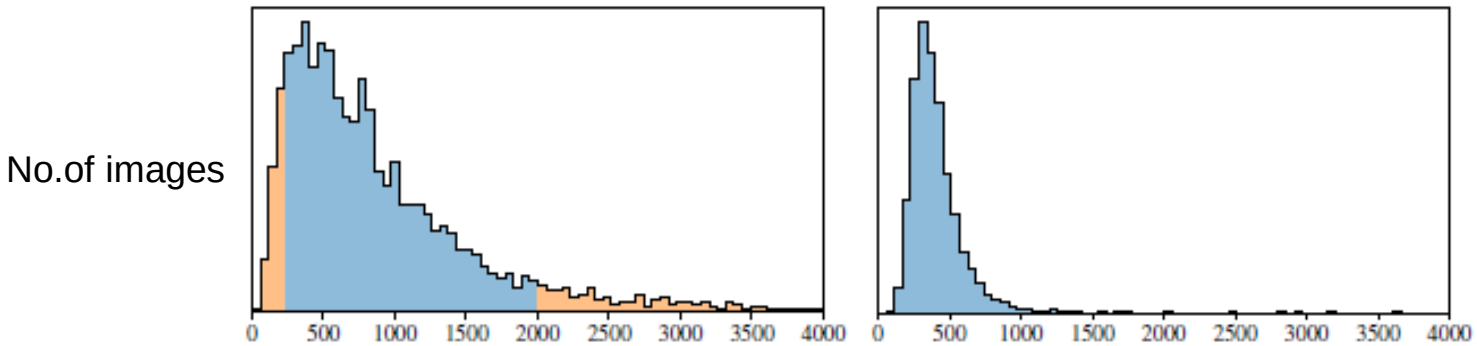


Figure 5. (a) Distribution of PPL scores of individual images generated using baseline StyleGAN (config A) with LSUN CAT (FID = 8.53, PPL = 924). The percentile ranges corresponding to Figure 4 are highlighted in orange. (b) StyleGAN2 (config F) improves the PPL distribution considerably (showing a snapshot with the same FID = 8.53, PPL = 387).

Summary

- GAN
- StyleGAN 1 & StyleGAN2 :

  extract and control individual attributes of generated images