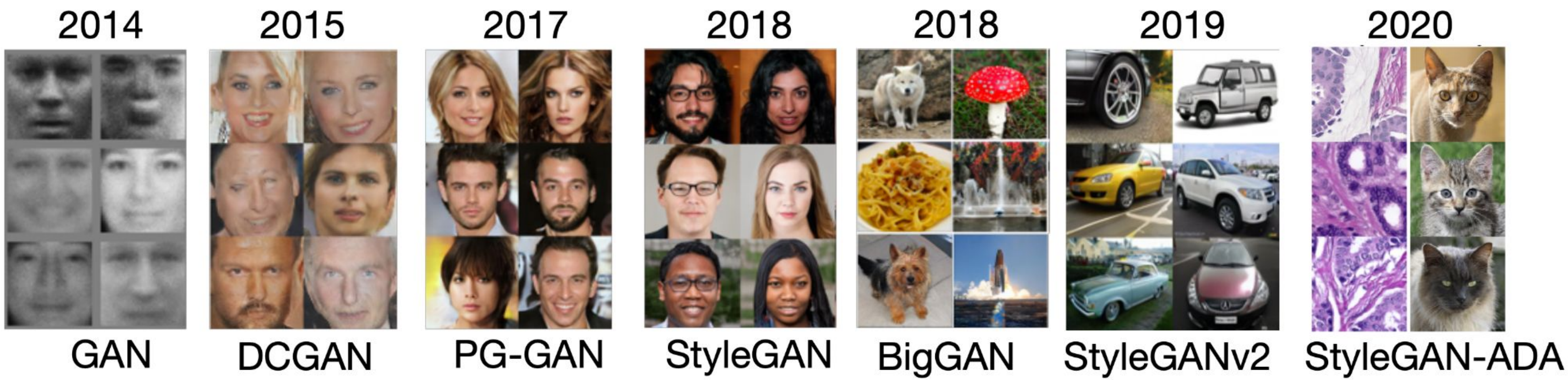
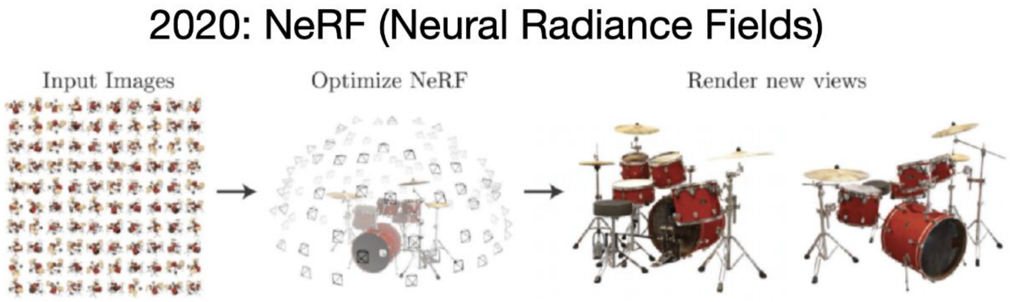
1. GANs and StyleGANs

2. AutoEncoders

Chen Wei, 11/29/2022 @PMVC

Progress for Image Generation





Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

Progress for Image Generation

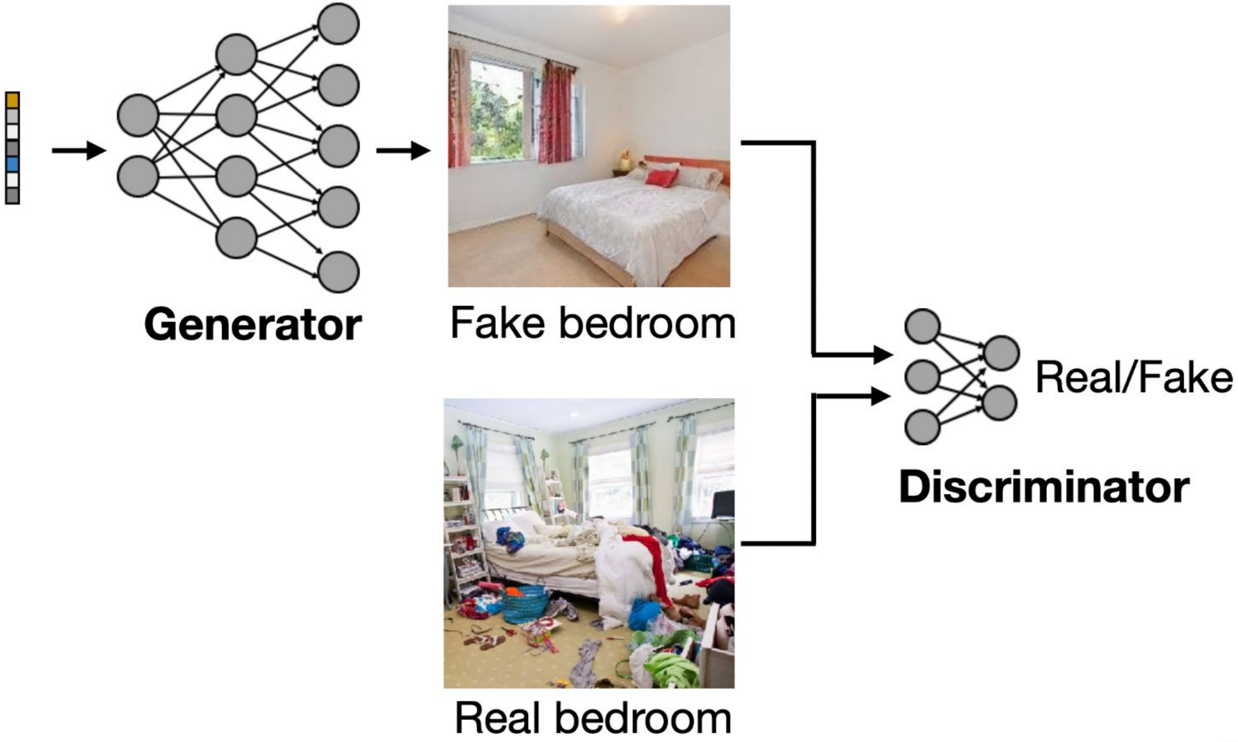
2020? 2022?

An astronaut riding a horse in a photorealistic style.

Diffusion Models

OpenAI DALL-E2 https://openai.com/dall-e-2/
Google Imagen https://imagen.research.google/
Meta Make-A-Video https://makeavideo.studio/
GPT-4 ???

Generative Adversarial Networks (GANs)

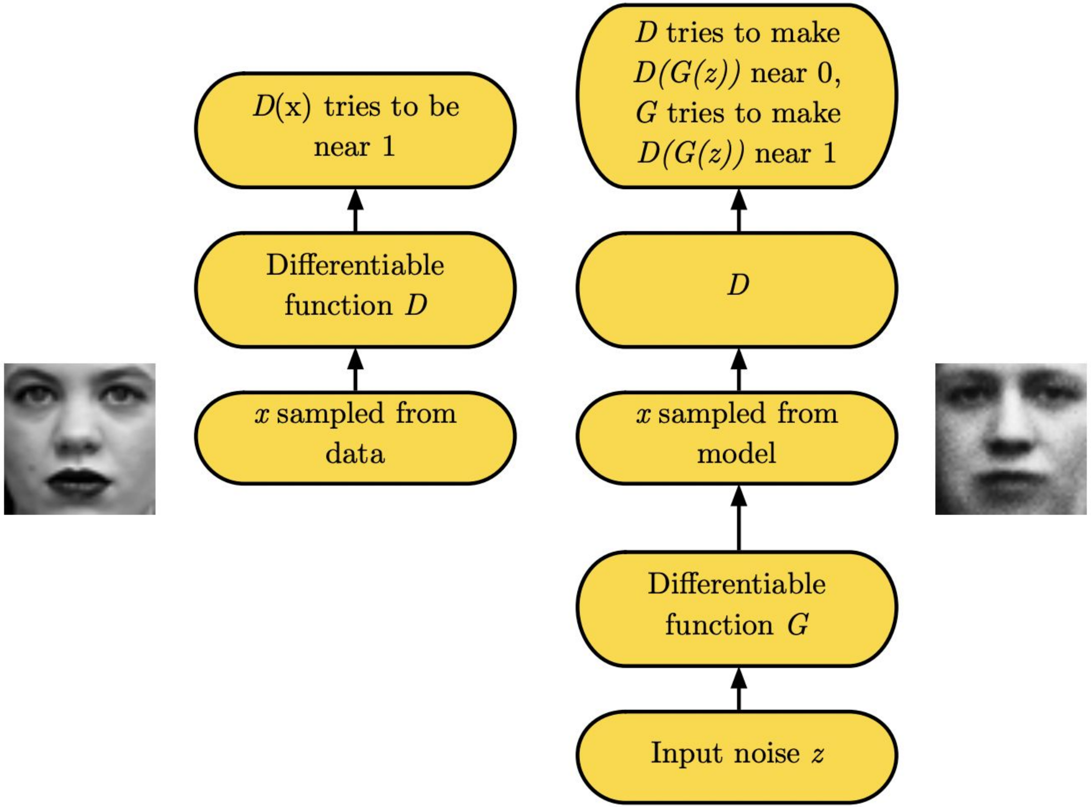


Goodfellow et al. NeurIPS'14

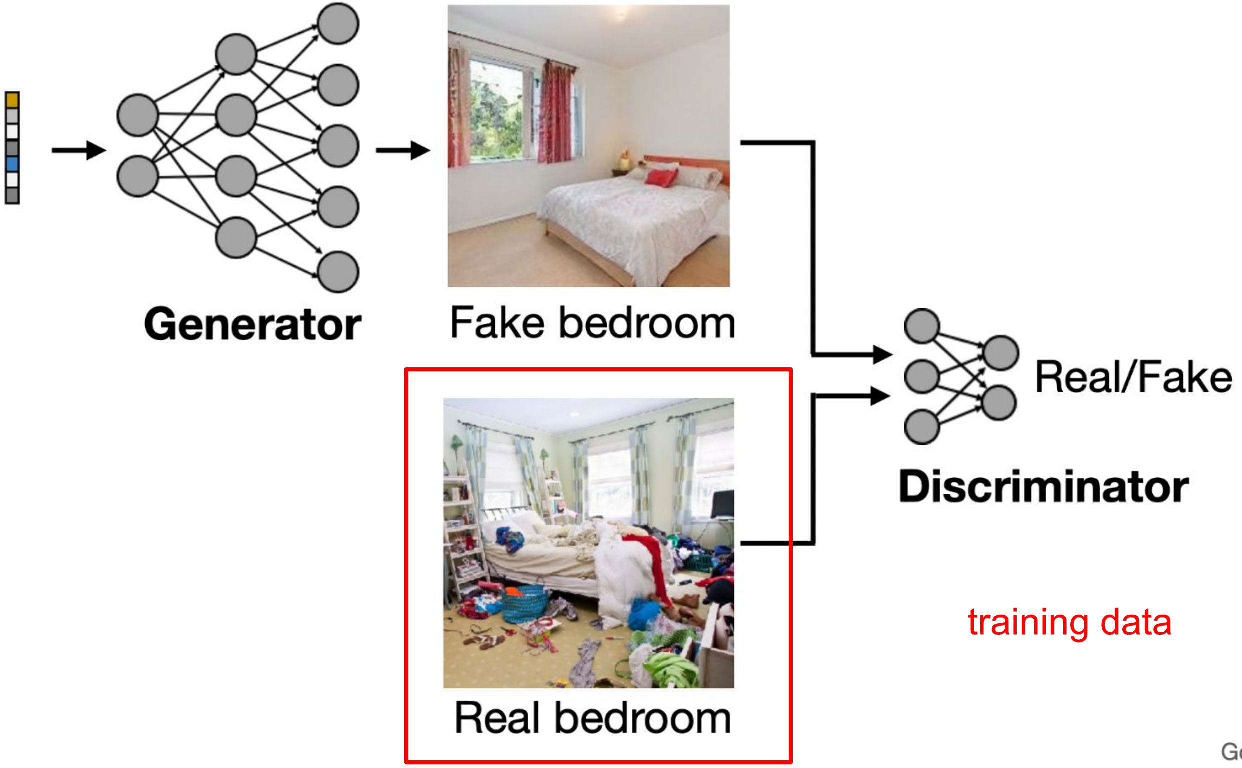
Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

Generative Adversarial Networks (GANs)

- The basic idea of GANs is to set up a game between two players.
  - Generator
    - Creates samples that are intended to come from the same distribution as the training data
    - The counterfeiter: Trained to fool the discriminator
  - Discriminator
    - Examines samples to determine whether they are real or fake
    - The police: Trained to distinguish between the generated or the real (training data)
- Formally, GANs are a structured probabilistic model containing latent variables $\mathbf{z}$ and observed variables $\mathbf{x}$.


D(x) tries to be near 1
Differentiable function D
x sampled from data
D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1
D
x sampled from model
Differentiable function G
Input noise z

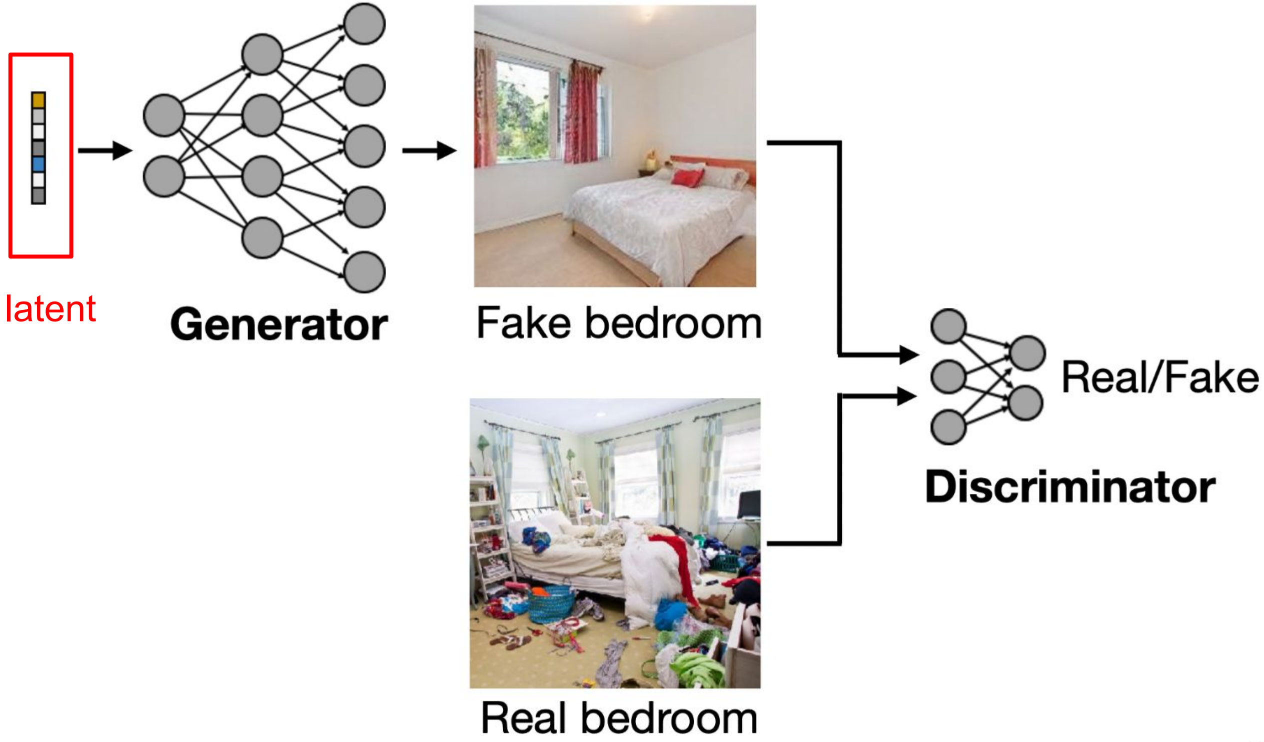
Generative Adversarial Networks (GANs)



Goodfellow et al. NeurIPS’14

Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

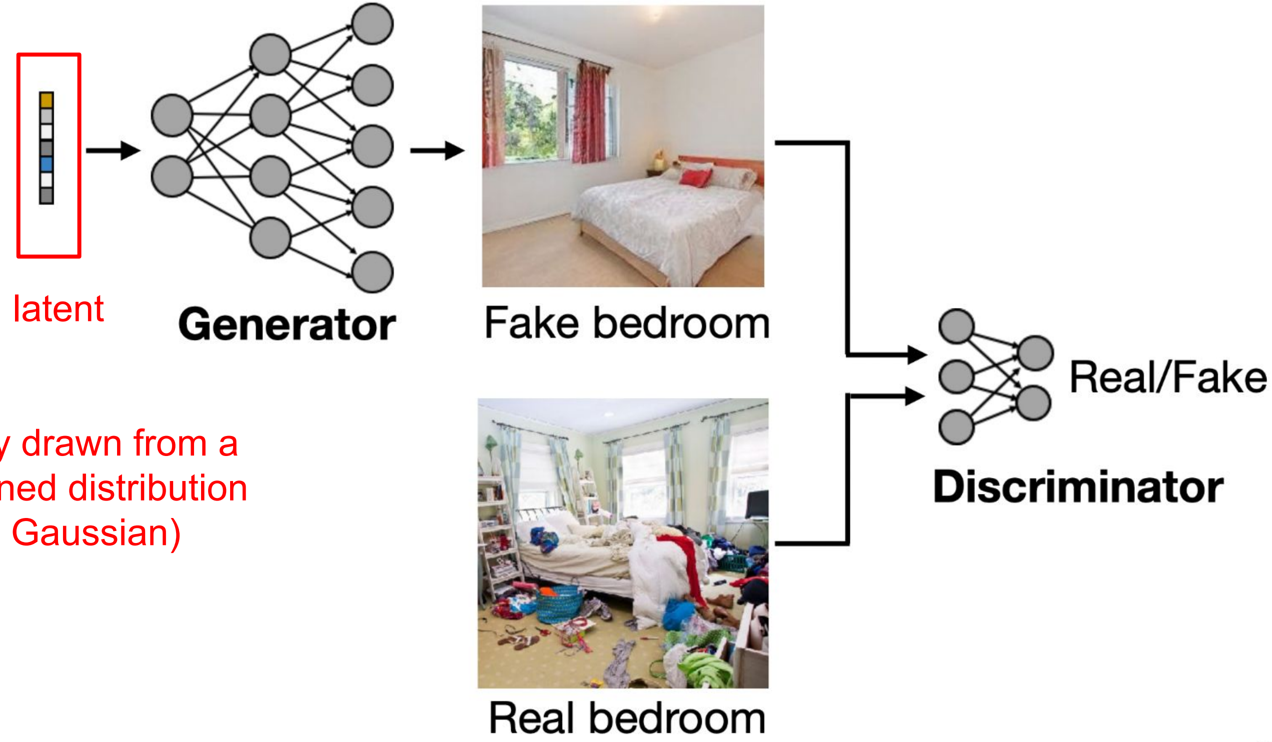
Generative Adversarial Networks (GANs)



Goodfellow et al. NeurIPS'14

Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

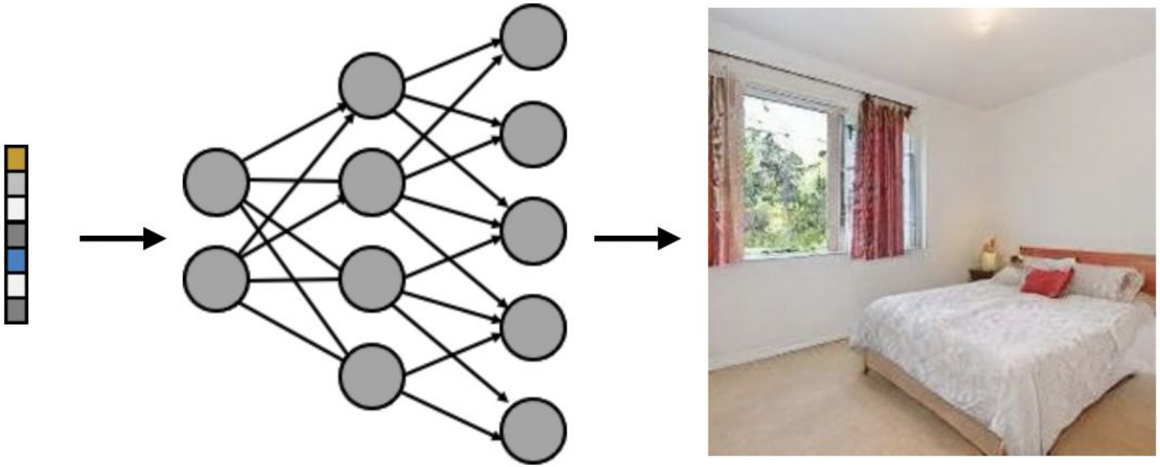
Generative Adversarial Networks (GANs)



typically drawn from a pre-defined distribution (eg. Gaussian)

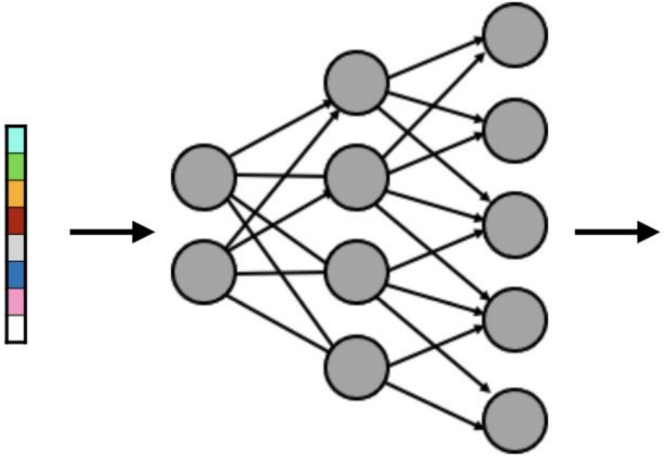
Goodfellow et al. NeurIPS'14

Generative Adversarial Networks (GANs)



Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

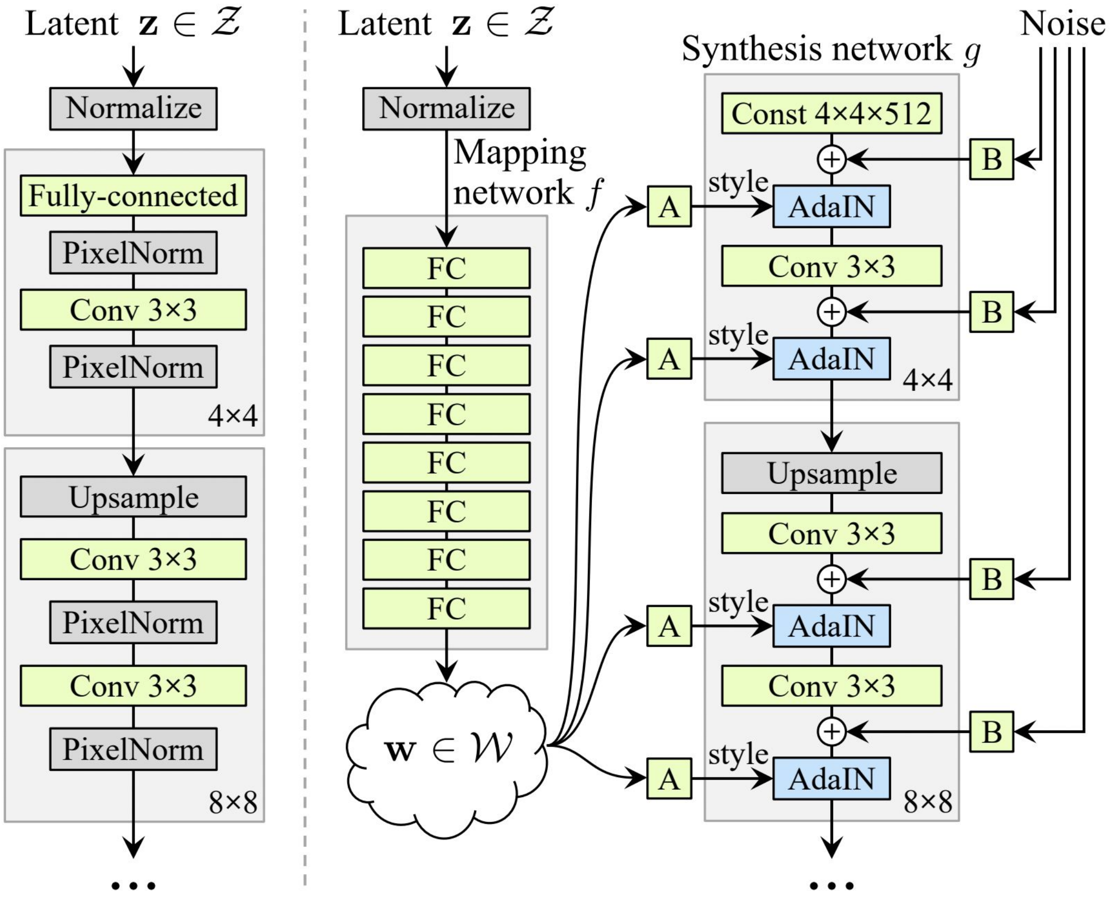
Generative Adversarial Networks (GANs)



Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

StyleGANs

Generator



(a) Traditional (b) Style-based generator

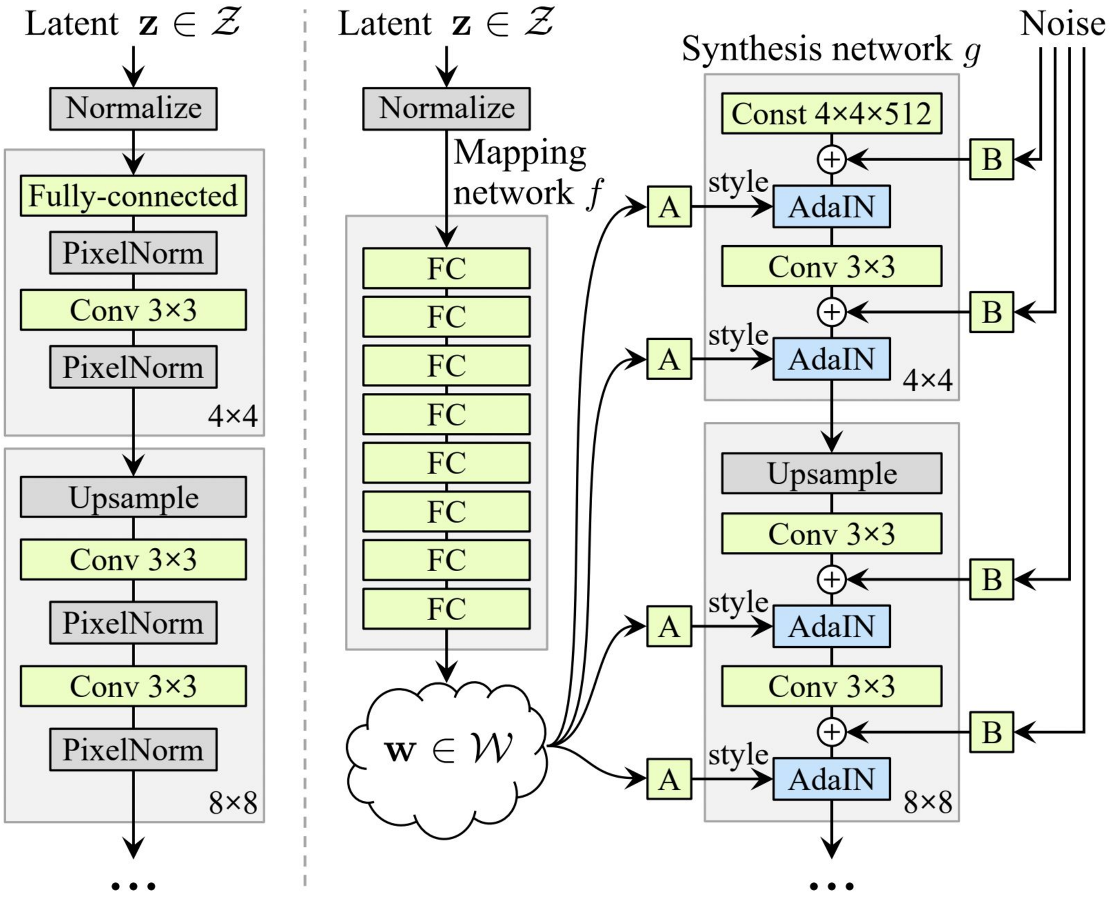
Tero Karras et al., A Style-Based Generator Architecture for Generative Adversarial Networks, CVPR 2019

StyleGANs

AdaIN

adaptive instance normalization

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

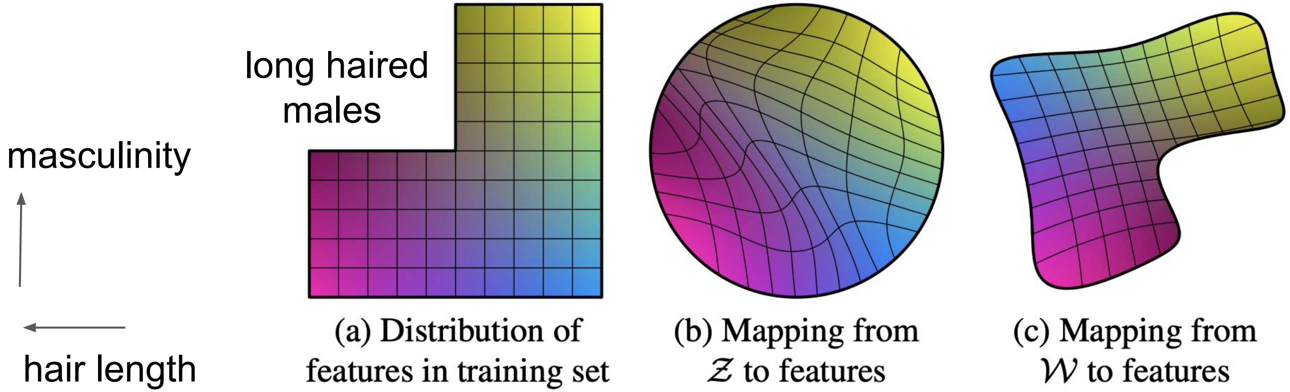


(a) Traditional (b) Style-based generator

Tero Karras et al., A Style-Based Generator Architecture for Generative Adversarial Networks, CVPR 2019

StyleGANs

- StyleGAN embeds the input latent code **z** into an intermediate latent space **w**
  - **w** = F(**z**)
- Now it is **w**, not **z**, that controls the *style* of the generated images



(a) Distribution of features in training set

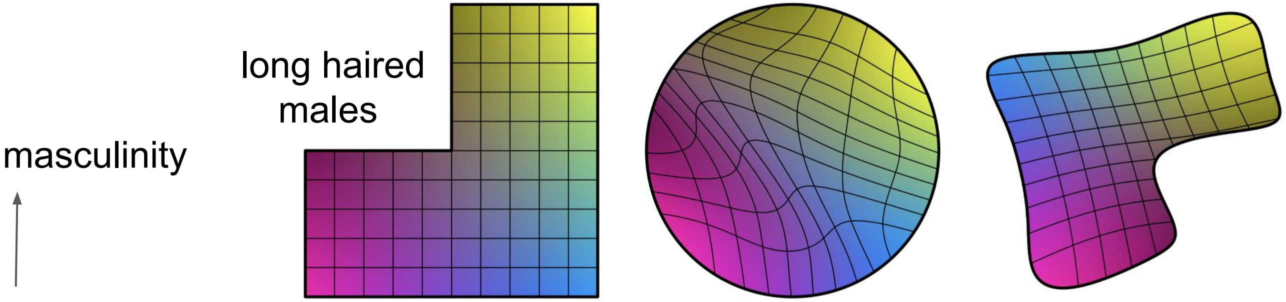
(b) Mapping from $\mathcal{Z}$ to features

(c) Mapping from $\mathcal{W}$ to features

Tero Karras et al., A Style-Based Generator Architecture for Generative Adversarial Networks, CVPR 2019

StyleGANs

- StyleGAN embeds the input latent code **z** into an intermediate latent space **w**
  - **w** = **F**(**z**)
- Now it is **w**, not **z**, that controls the *style* of the generated images
  - The mapping **F** can "unwarp" **z** to **w** so that the factors of variation become more linear.
  - Style: factors of variation of the domain of interest



(a) Distribution of features in training set

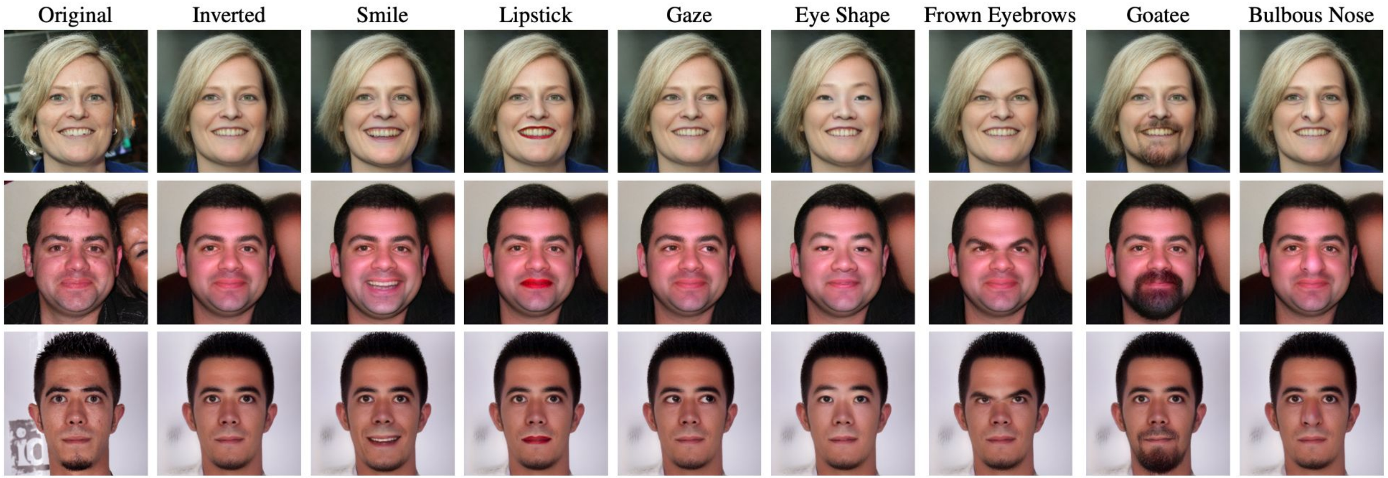
(b) Mapping from $\mathcal{Z}$ to features

(c) Mapping from $\mathcal{W}$ to features

Tero Karras et al., A Style-Based Generator Architecture for Generative Adversarial Networks, CVPR 2019

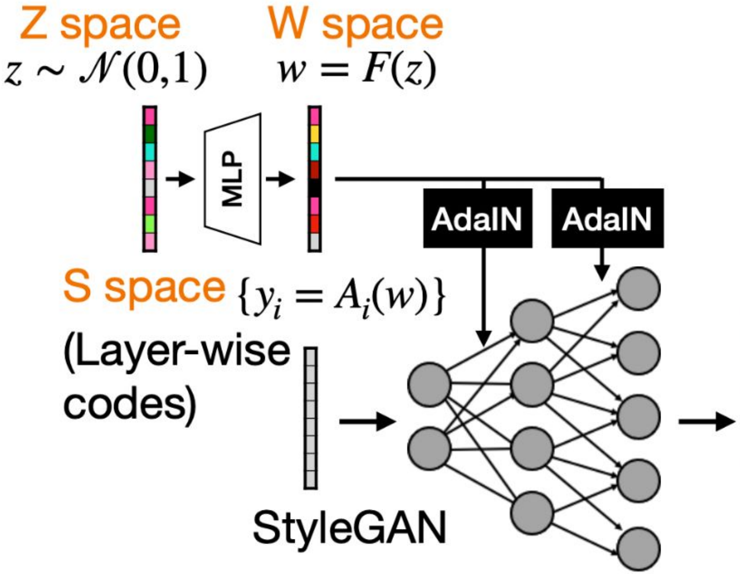
StyleGANs - The Disentangled Latent Space

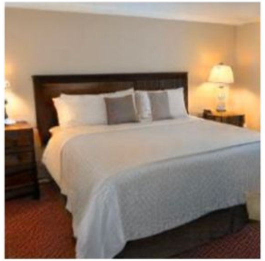
- There are various definitions for disentanglement.
- A common goal is a latent space that consists of linear subspaces, each of which controls one factor of variation.



Zongze Wu et al., StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation, CVPR 2021

Which latent space is more disentangled?





Reconstruction Error (Xu et al. CVPR'21)

Space	MSE	FID
W space	0.0601	22.24
S space	**0.0464**	**18.48**

Disentanglement (Wu et al. CVPR'21)

Space	Disentanglement
Z space	0.31
W space	0.54
S space	**0.75**

Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

GAN Inversion

Applying the pretrained GAN model to image processing tasks

GAN inversion:

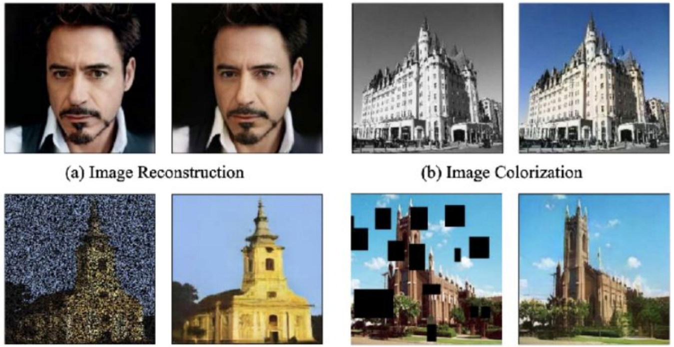
$$x^* = argmin_x || G(x) - I ||$$

Colorization:

$$x^* = argmin_x || \text{rgb2gray}(G(x)) - I_{gray} ||$$

Super-resolution:

$$x^* = argmin_x || \text{down}(G(x)) - I_{small} ||$$



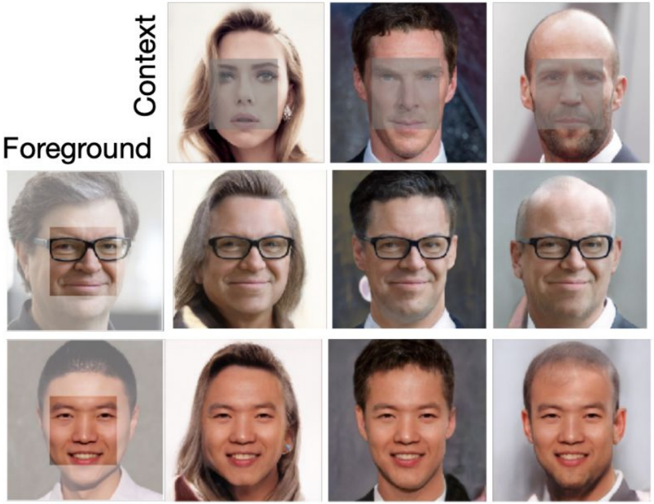
(a) Image Reconstruction (b) Image Colorization

(d) Image Denoising (e) Image Inpainting

Gu, Shen, Zhou. Image Processing Using Multi-Code GAN Prior. CVPR'20

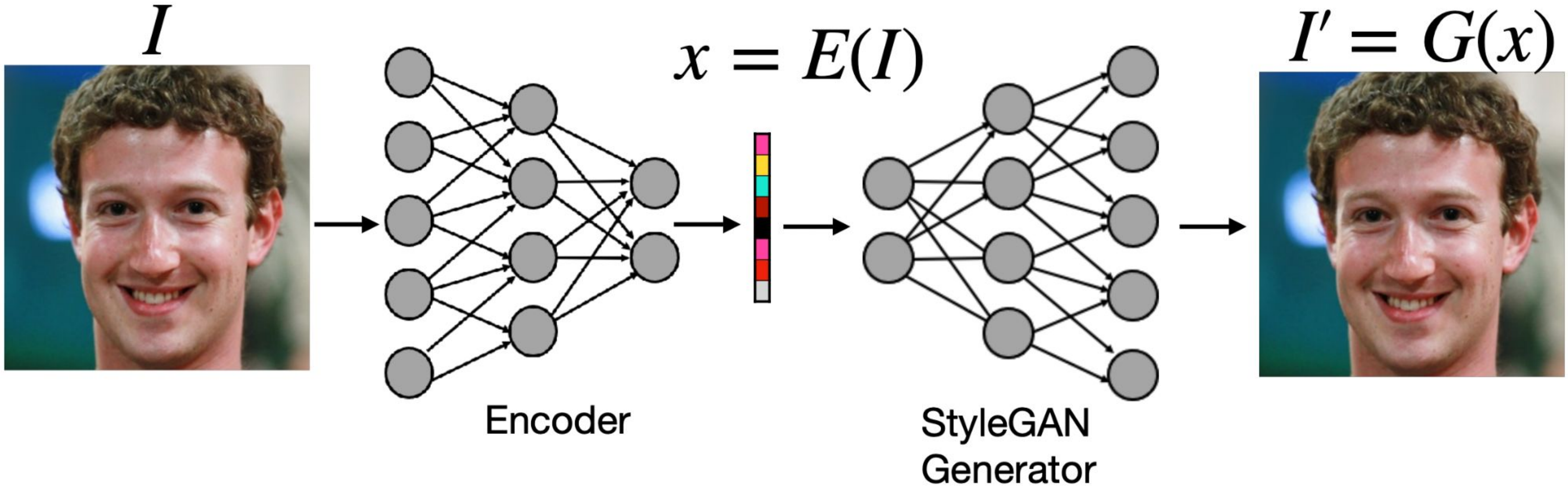
Masked optimization

$$x^* = argmin_x || m \cdot G(x) - m \cdot I_{context} ||$$



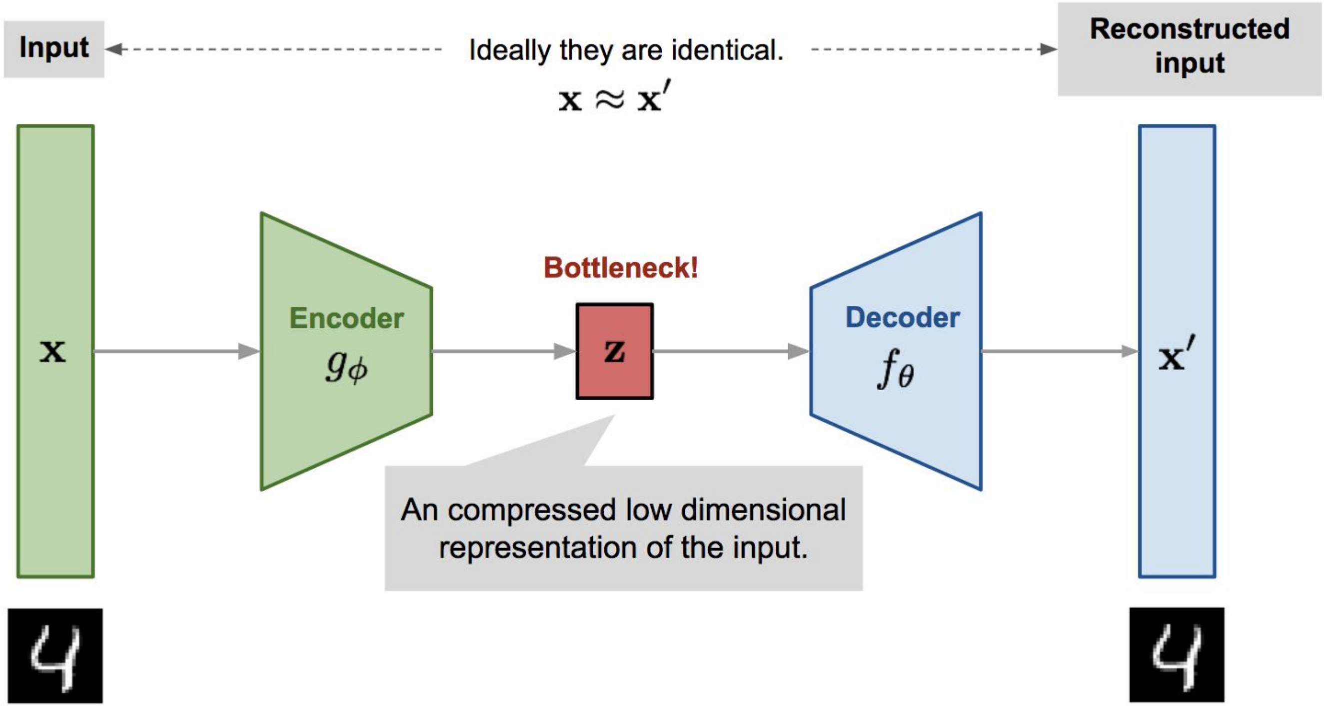
Zhu, Shen, Zhao, Zhou. In-domain GAN Inversion. ECCV'20

Encoding Real Image into StyleGAN space



Bolei, Zhou, Tutorial on Interpretable Machine Learning for Computer Vision at CVPR 2021

AutoEncoders



https://lilianweng.github.io/posts/2018-08-12-vae/

AutoEncoders

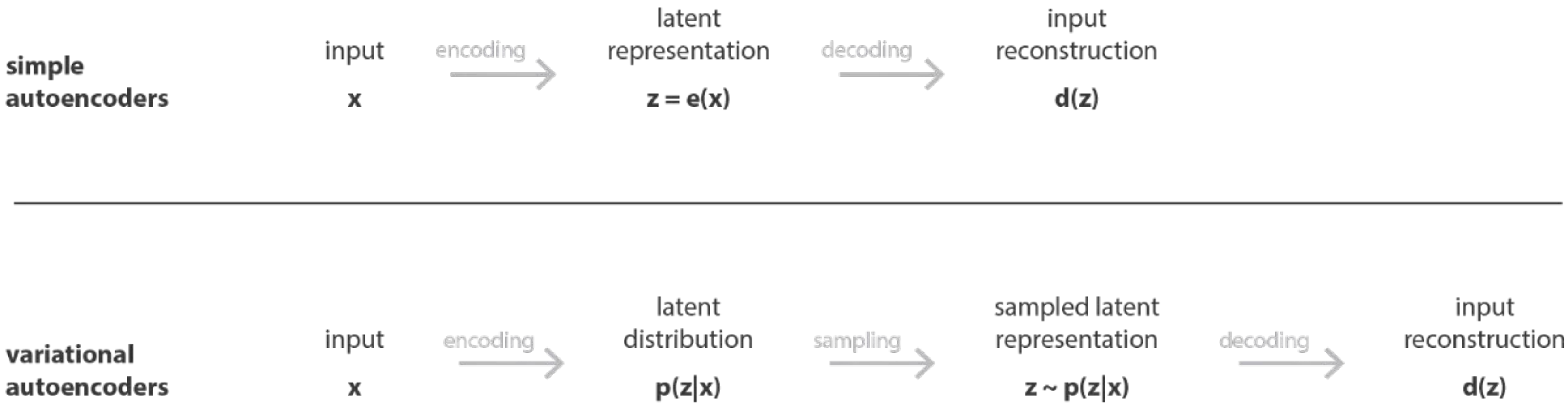
- Encoder
  - Transforms the original high-dimension input (eg., images) into the low-dimensional latent.
  - Hopefully lossless
- Decoder
  - Recovers the high-dimensional data from the encoded low-dimensional latents
- Dimensionality Reduction
  - Links to PCA

$$L_{\mathrm{AE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$

Variational AutoEncoders

- Vanilla autoencoder’s latent space is NOT well-organized/structured to be sampled from
  - Because there is no force for the latent space to do so
- Variational AutoEncoders
  - Autoencoders whose latent space is regularized to a structured distribution (eg., Gaussian distribution)
  - The latent is now a *distribution*

Variational AutoEncoders



$$
\begin{aligned}
L_{\mathrm{VAE}}(\theta, \phi) &= -\log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) \\
&= -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))
\end{aligned}
$$

https://miro.medium.com/max/1400/1*ejNnusxYrn1NRDZf4Kg2lw@2x.webp

Variational AutoEncoders

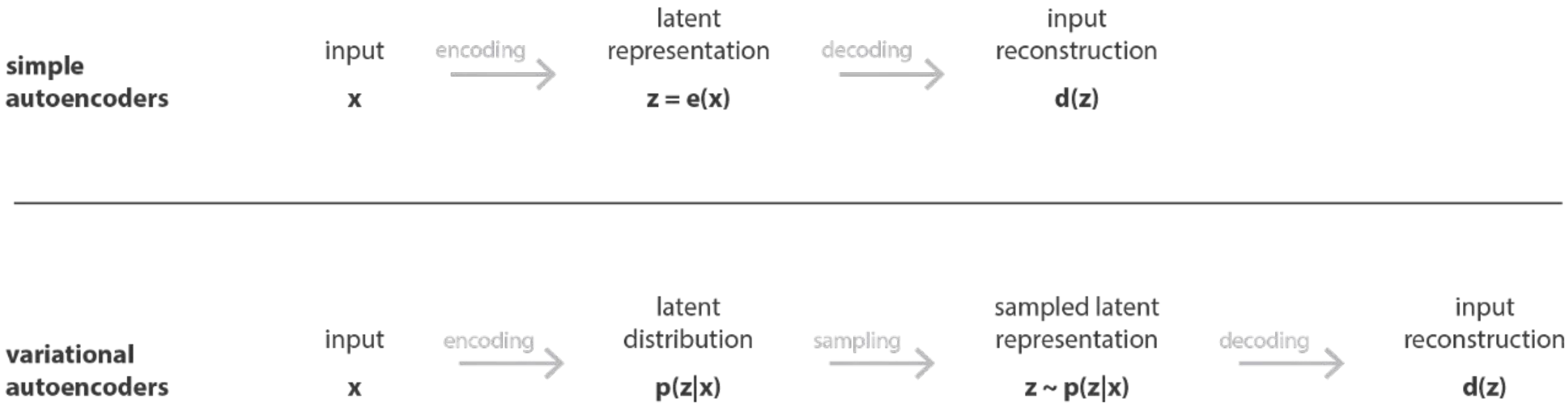
simple autoencoders	input **x**	encoding →	latent representation **z = e(x)**	decoding →	input reconstruction **d(z)**			
variational autoencoders	input **x**	encoding →	latent distribution **p(z|x)**	sampling →	sampled latent representation **z ~ p(z|x)**	decoding →	input reconstruction **d(z)**

$$
\begin{aligned}
L_{\mathrm{VAE}}(\theta, \phi) &= -\log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) \\
&= \boxed{-\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})} + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))
\end{aligned}
$$

the reconstruction term:
negative log-likelihood

https://miro.medium.com/max/1400/1*ejNnusxYrn1NRDZf4Kg2lw@2x.webp

Variational AutoEncoders



$$L_{\mathrm{VAE}}(\theta, \phi) = -\log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x}))$$

the regularization term:
KL divergence to the prior distribution

$$= -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + \boxed{D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))}$$

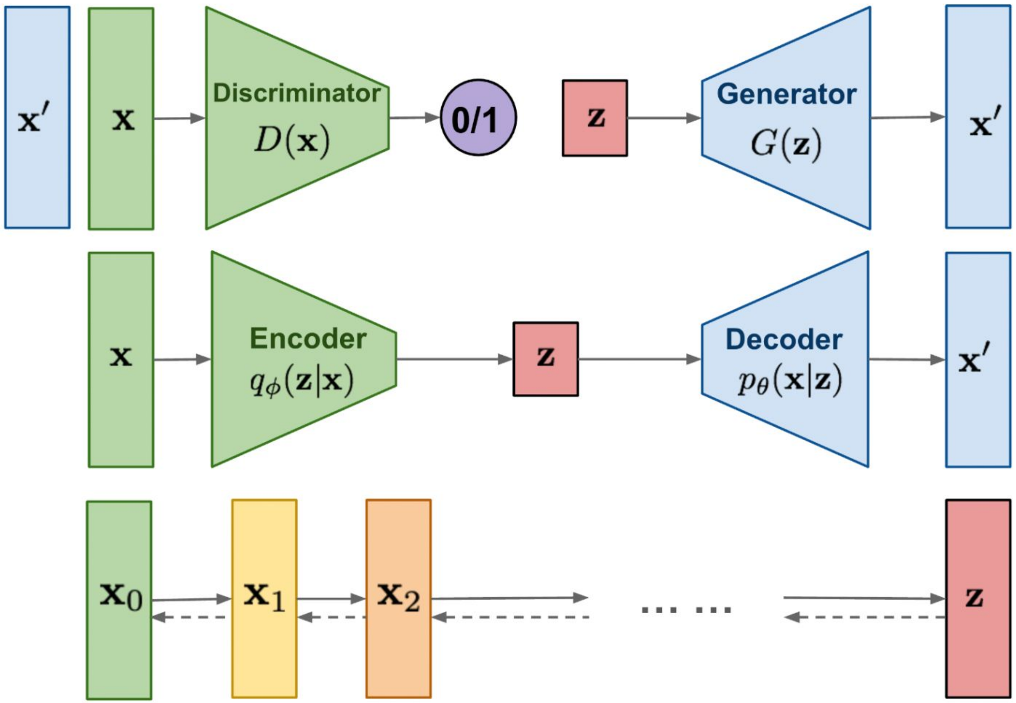
https://miro.medium.com/max/1400/1*ejNnusxYrn1NRDZf4Kg2lw@2x.webp

Summary

GAN: Adversarial training

VAE: maximize variational lower bound

Diffusion models: Gradually add Gaussian noise and then reverse



https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Recall Fidler's 3D Neural Rendering Approach

$$I \mapsto W \mapsto z \mapsto I$$

- The whole system is an autoencoder
- I → W: An autoencoder approach where the decoder is a differentiable renderer I=F(W;α) and $P_\varphi(W \mid I)$ is the encoder.
- z → I: A learned StyleGAN generative model $P_\theta(I \mid z)$
- W → z: Learn $f_\psi(z \mid W)$ using an autoencoder reconstruction loss
  - This can be done by the autoencoder because the latent variables of styleGANs are fairly interpretable and so the function $f_\psi(z \mid W)$ cannot be too complicated