

Lecture 5 - Advanced Image Processing

Advanced Machine Learning

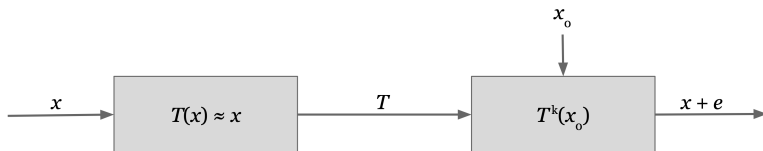
Alexander Kovalenko, Miroslav Čepek, Zdeněk Buk, Rodrigo
Da Silva Alves, Vojtěch Rybář, Petr Šimánek

FIT CTU

2. 3. 2023

Fractal Image Compression

- A very good blogpost: [link](#)
- O. Zmeskal et al, Entropy of fractal systems, 2013
- Y. Fisher, Fractal Image Compression, 1994



$$x = \lim_{k \rightarrow \infty} T^k(x_0) + e$$

Fractal Image Compression (2)

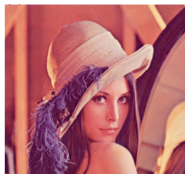


Original Lena image

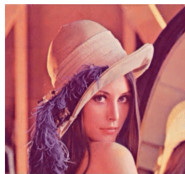


Self-similar portions of the image

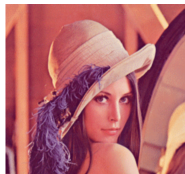
Fractal Image Compression (3)



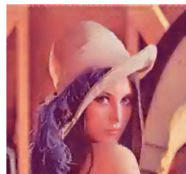
Original Lena image (184,320 bytes)



FIF-max. quality (30,368)
comp. ratio: **6.07:1**

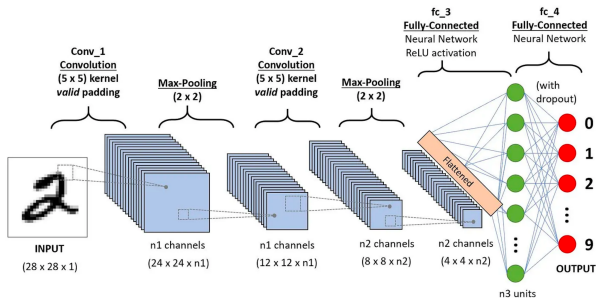


JPEG-min. quality (8,247)
comp. ratio: **22.35:1**



FIF-min. quality (3,924)
comp. ratio: **46.97:1**

A Quick Recap on CNNs



Y LeCun, L Bottou, Y Bengio, P Haffner
Proceedings of the IEEE 86 (11), 1998

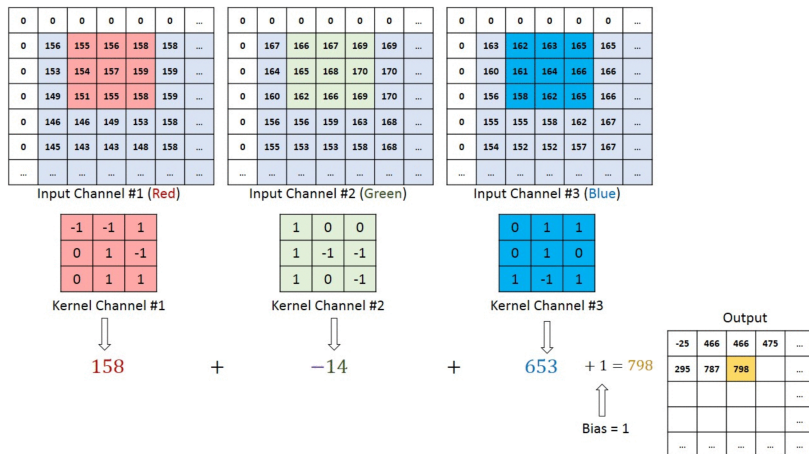
1	1	1	0	0
0	1 _{c1}	1 _{c1}	1 _{c1}	0
0	0 _{c2}	1 _{c2}	1 _{c2}	1
0	0 _{c3}	1 _{c3}	1 _{c3}	0
0	1	1	0	0

Image

4	3	4
2	4	

Convolved Feature

Quick recap on CNNs (2)



Generative Models

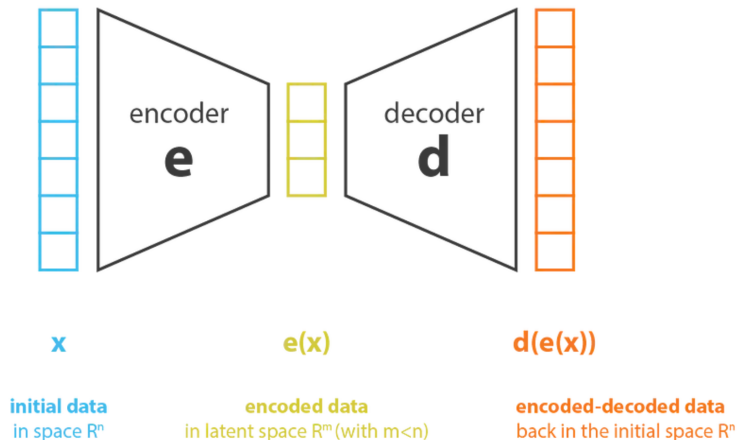
- Attempt to model the underlying distribution of the data
- Learn how the data was generated
- Can be used to generate new data points similar to the original data
- Useful when there is a need to generate new data points

Discriminative Models

- Focus on learning the decision boundary
- Do not try to model the distribution of the data
- Directly learn the mapping from input to output labels
- Useful when the goal is to classify new data points

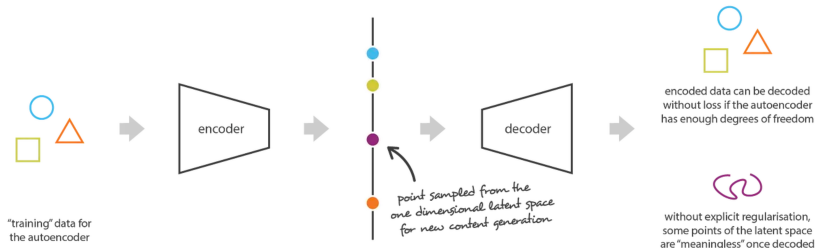
- VAE – CVAE, VampPrior VAE ...
- GAN – DCGAN, CGAN, CycleGAN ...
- Transformers – TransGAN, MaskGIT ...
- Diffusion – DiffAE, stable diffusion, Paella ...

Autoencoders vs. Variational Autoencoders



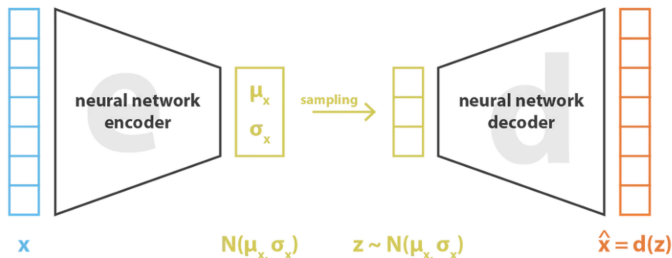
<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

Autoencoders vs. Variational Autoencoders (2)



<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

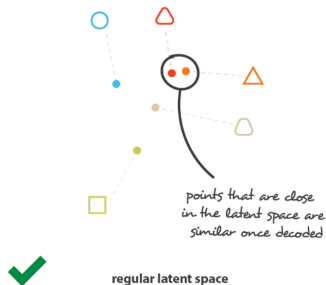
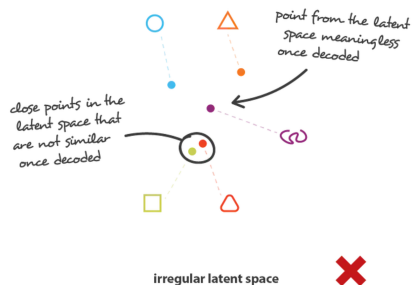
Autoencoders vs. Variational Autoencoders (3)



$$\text{loss} = ||x - \hat{x}||^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = ||x - d(z)||^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)]$$

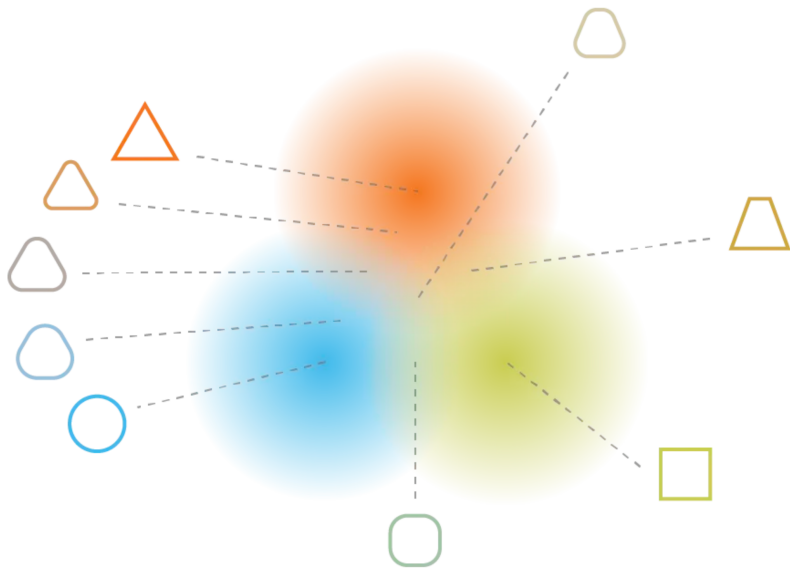
<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

Autoencoders vs. Variational Autoencoders (4)

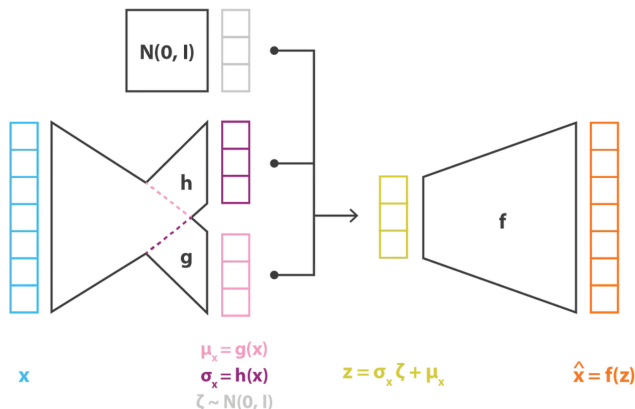


<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

Autoencoders vs. Variational Autoencoders (5)



Autoencoders vs. Variational Autoencoders (6)



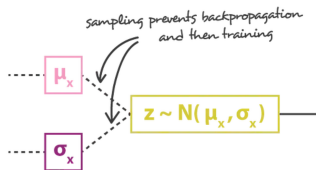
$$\text{loss} = C \|x - \hat{x}\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = C \|x - f(z)\|^2 + \text{KL}[N(g(x), h(x)), N(0, I)]$$

<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

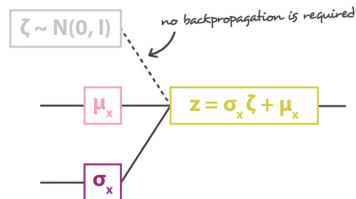
Autoencoders vs. Variational Autoencoders (7)

—— no problem for backpropagation

..... backpropagation is not possible due to sampling



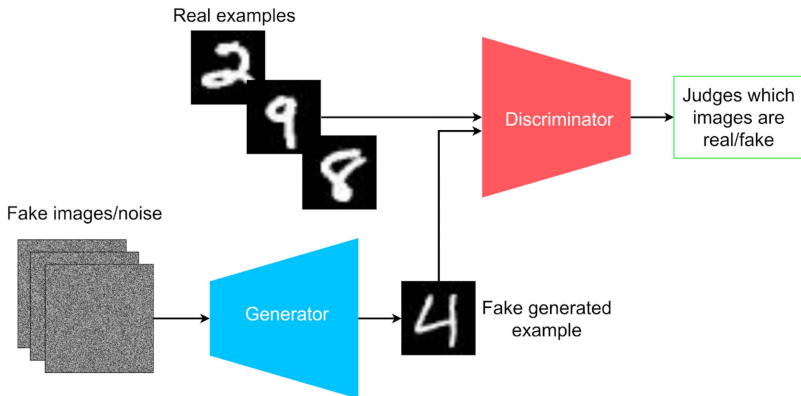
sampling without reparametrisation trick



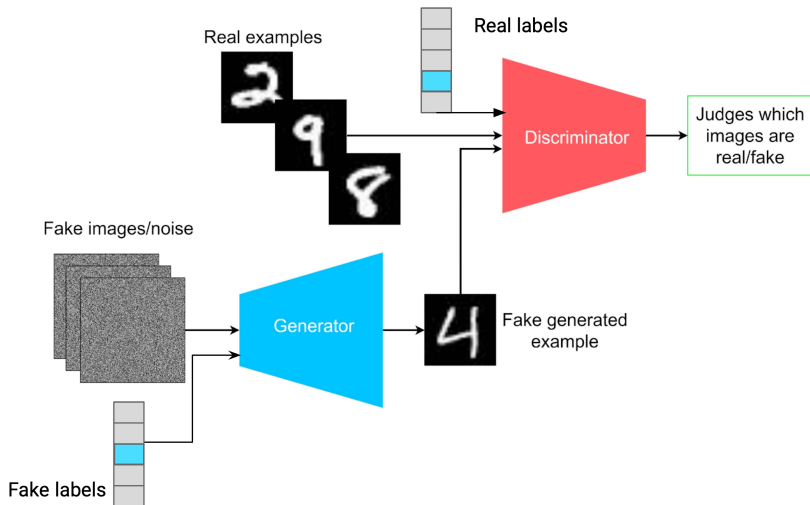
sampling with reparametrisation trick

<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

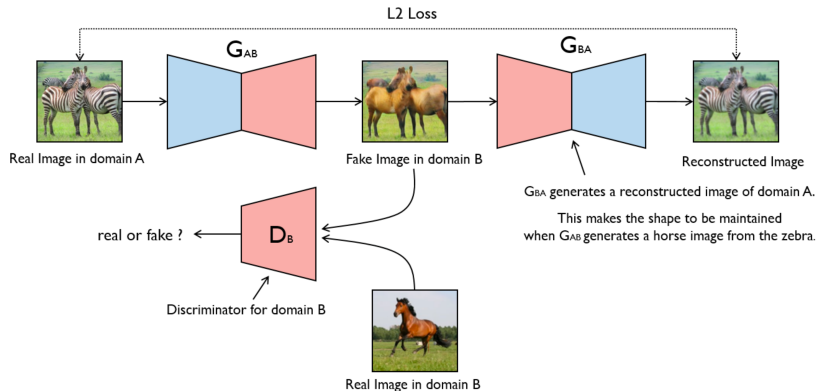
Classical GAN



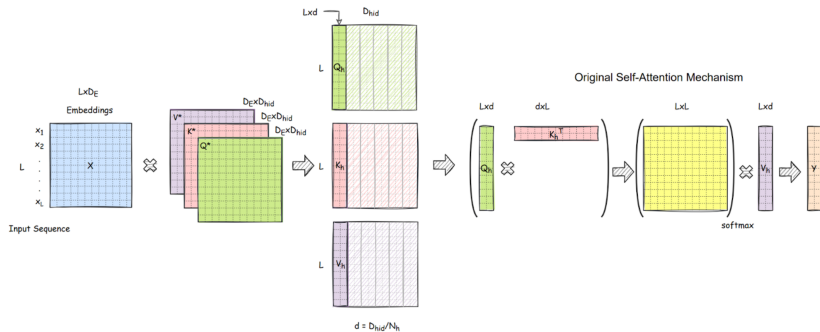
Conditional GAN



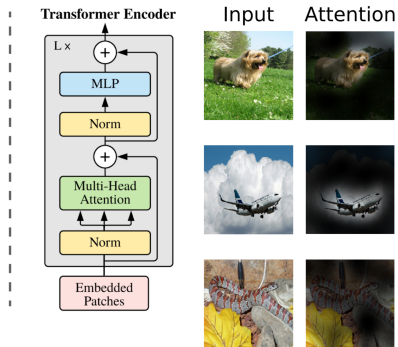
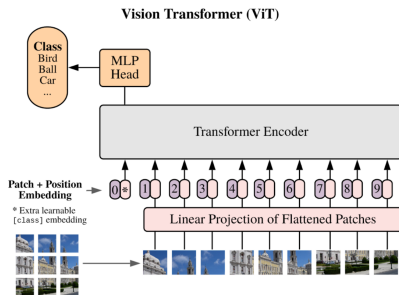
Generative Adversarial Networks



Vision Transformer (1)

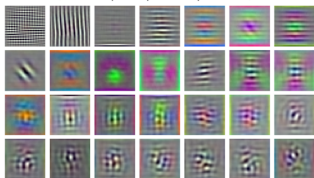


Vision Transformer (2)

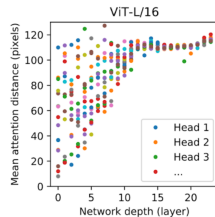
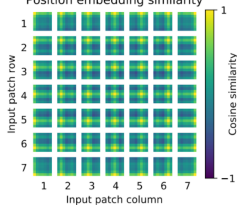


Vision Transformer (3)

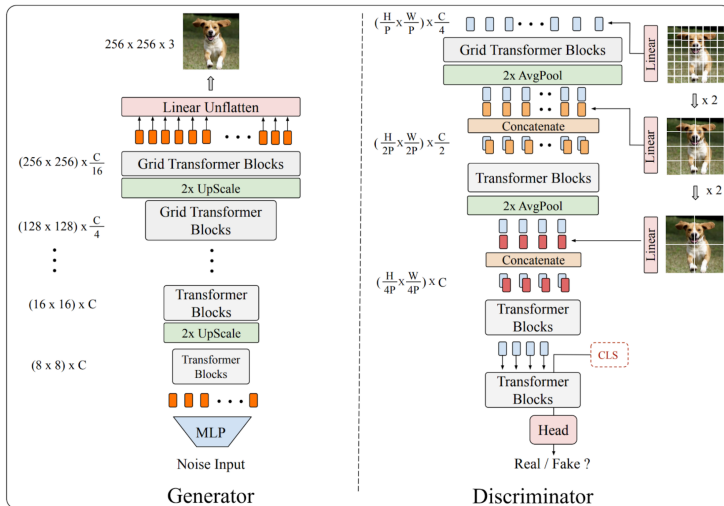
RGB embedding filters
(first 28 principal components)



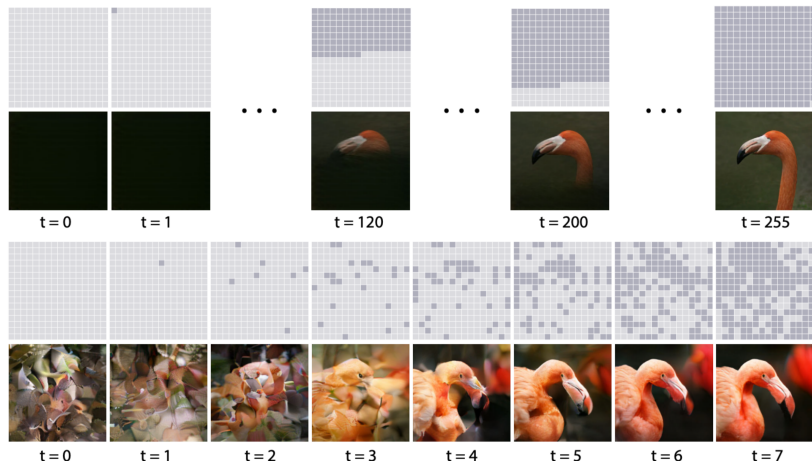
Position embedding similarity



Trans GAN

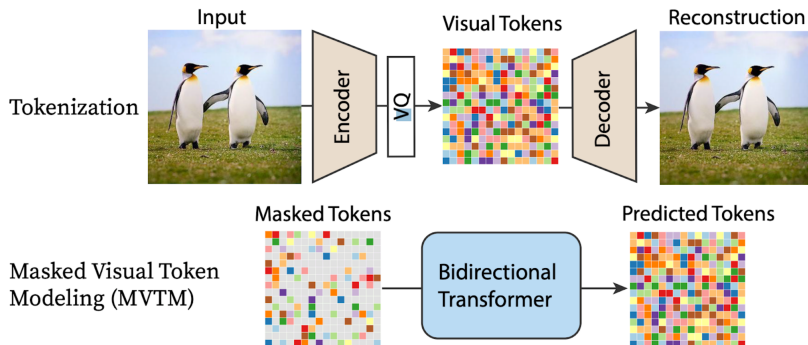


maskgit: Masked Generative Image Transformer



<https://arxiv.org/pdf/2202.04200.pdf>

maskgit: Masked Generative Image Transformer (2)

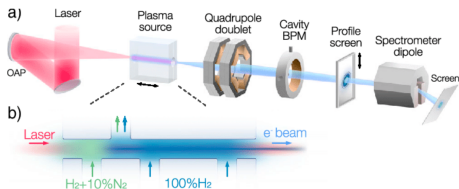
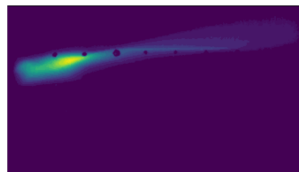


<https://arxiv.org/pdf/2202.04200.pdf>

Key papers:

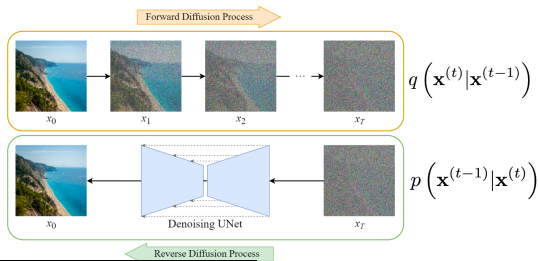
- J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015, [ArXiv Paper](#) ([Github](#))
- Y. Song, S. Ermon, Generative Modeling by Estimating Gradients of the Data Distribution, 2020, [ArXiv Paper](#)
- J. Ho, A. Jain, P. Abbeel, Denoising Diffusion Probabilistic Models, 2020, [ArXiv Paper](#)
- A. Nichol, P. Dhariwal, Improved Denoising Diffusion Probabilistic Models, 2021, [ArXiv Paper](#) ([Github](#))
- P. Dhariwal, A. Nichol, Diffusion Models Beat GANs on Image Synthesis, [ArXiv Paper](#), ([Github](#))

Commercial Break!

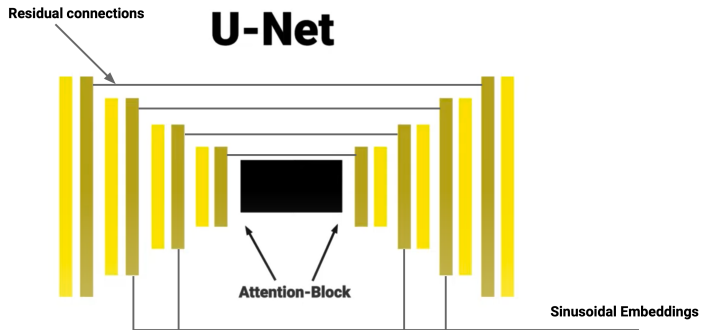


Diffusion Models (2)

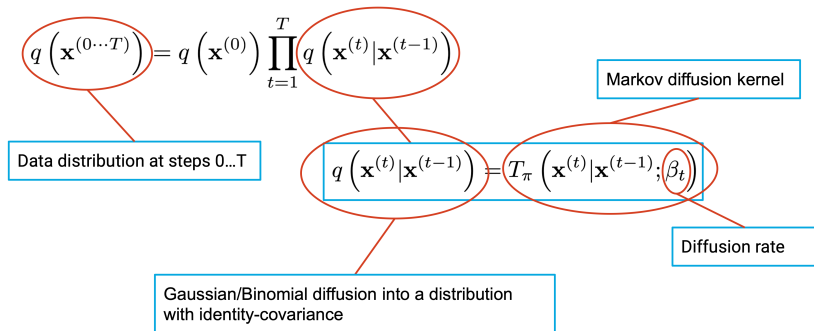
- To create samples using score-based models, we must first take into account a diffusion process that gradually transforms the data into random noise. By reversing this diffusion process, we can generate samples with scores.
- A diffusion process can be likened to the movement of a particle in a fluid, akin to Brownian motion. The particle moves unpredictably due to chance collisions with other particles along its path.



Diffusion Models (3)



Forward Diffusion



Forward Diffusion (2)

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t, \sqrt{1 - \beta_t}x_{t-1}, \beta_t I)$$

Diagram illustrating the forward diffusion process equation:

- $\mathcal{N}$: Normal distribution
- x_t : Mean
- $\sqrt{1 - \beta_t}x_{t-1}$: Variance



Forward Diffusion (3)

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t, x_{t-1}\sqrt{1-\beta}, \beta_t\mathbf{I})$$

$$\beta = 0.001 \dots 0.02$$

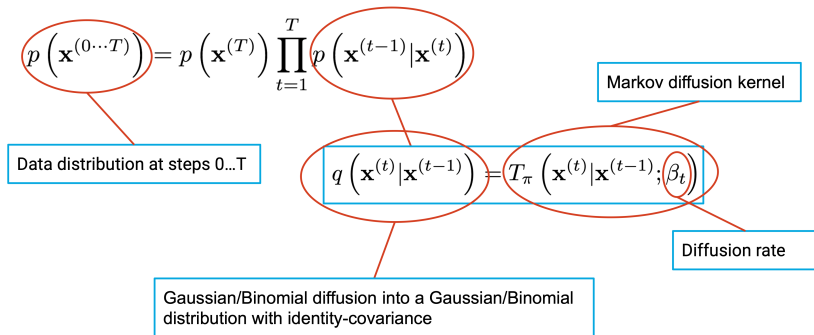
$$\alpha_t = \prod_{s=1}^t \alpha_s$$

$$\begin{aligned} q(x_t|x_{t-1}) &= \mathcal{N}(x_t, x_{t-1}\sqrt{1-\beta}, \beta_t\mathbf{I}) \\ &= x_{t-1}\sqrt{1-\beta} + \epsilon\sqrt{\beta_t} \\ &= x_{t-1}\sqrt{\alpha_t} + \epsilon\sqrt{1-\alpha_t} \end{aligned}$$

$$q(x_t|x_{t-2}) = x_{t-2}\sqrt{\alpha_t\alpha_{t-1}} + \epsilon\sqrt{1-\alpha_t\alpha_{t-1}}$$

$$q(x_t|x_0) = x_0\sqrt{\alpha_t\alpha_{t-1}\dots\alpha_1\alpha_0} + \epsilon\sqrt{1-\alpha_t\alpha_{t-1}\dots\alpha_1\alpha_0}$$

Reverse Diffusion



Reverse Diffusion

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_{\theta}(x_t, t), \Sigma_{\theta}(x_t, t))$$

Mean

Variance (fixed)

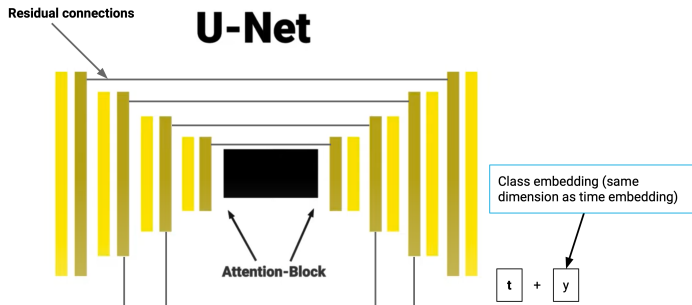
$$L = -\log(p_{\theta}(x_0))$$

The full math can be found in:

- <https://arxiv.org/pdf/1503.03585.pdf>, Appendix B,
- or in the blogpost: <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>

However, the authors have finally given up on it and approximated the formula:

$$L_{simple} = \mathbb{E}_{t, x_0, \epsilon} [\|\epsilon - \epsilon_{\theta}(x_t, t)\|^2]$$



Guided Diffusion (2)

- J. Ho & T. Salimans, Classifier Free Diffusion Guidance, 2022, <https://arxiv.org/pdf/2207.12598.pdf>
- Given the differentiable discriminative model that estimates:

$$p(y|x)$$

- we can obtain:

$$\nabla_x \log p(y|x)$$

- Guided diffusion by scaling the conditioning term:

$$\nabla_x \log p_\gamma(x|y) = \nabla_x \log p(x) + \gamma \nabla_x \log p(y|x)$$

More of a good stuff here:

<https://sander.ai/2022/05/26/guidance.html#fn:cf>

Guided Diffusion (3)

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$$\implies \log p(y|x) = \log p(x|y) + \log p(y) - \log p(x)$$

$$\implies \nabla_x \log p(y|x) = \nabla_x \log p(x|y) - \nabla_x \log p(x)$$

Substitute into the previous formula for guidance:

$$\nabla_x \log p_\gamma(y|x) = \nabla_x \log p(x) + \gamma(\nabla_x \log p(x|y) - \nabla_x \log p(x))$$

Or, equivalently:

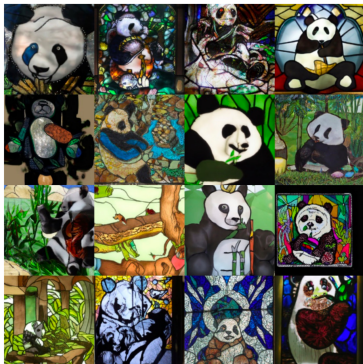
$$\nabla_x \log p_\gamma(y|x) = (1 - \gamma)\nabla_x \log p(x) + \gamma\nabla_x \log p(x|y)$$

More of a good stuff here:

<https://sander.ai/2022/05/26/guidance.html#fn:cf>

Guided Diffusion (4)

$\gamma = 1$

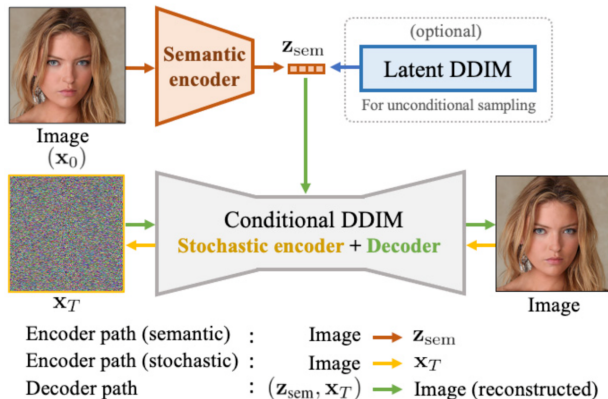


$\gamma = 3$



Diffusion Autoencoders

K. Preechakul, N. Chatthee, S. Wizadwongsa, S. Suwajanakorn,
Diffusion Autoencoders: Toward a Meaningful and Decodable
Representation, [ArXiv Paper](#), ([Github](#))

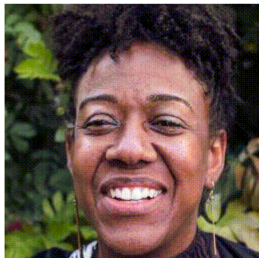
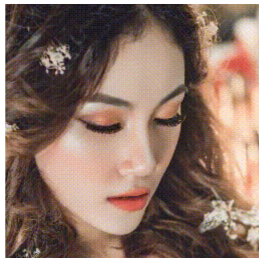
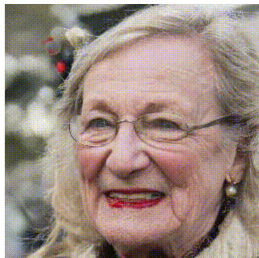
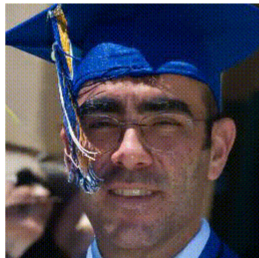


<https://diff-ae.github.io/>

Diffusion Autoencoders (2)



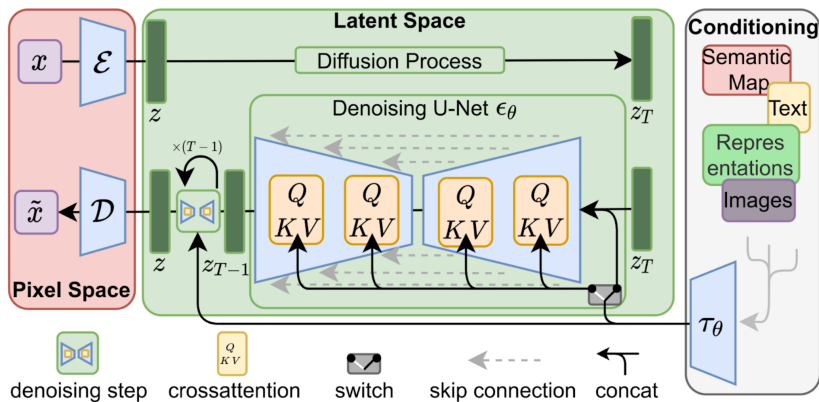
Diffusion Autoencoders (3)



R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer,
High-Resolution Image Synthesis with Latent Diffusion Models,
2022, [ArXiv Paper](#) ([Github](#))



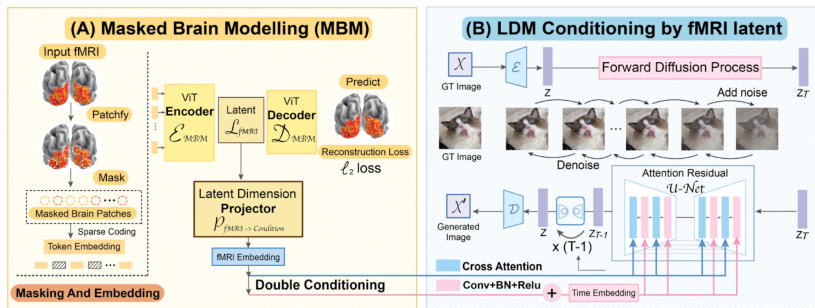
Stable Diffusion



<https://arxiv.org/pdf/2112.10752.pdf>

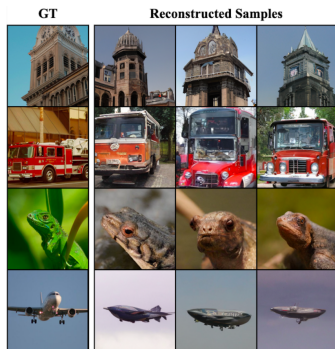
Seeing Beyond the Brain

J. Chen, et al., Seeing Beyond the Brain: Conditional Diffusion Model with Sparse Masked Modeling for Vision Decoding, 2022,
<https://mind-vis.github.io/>



<https://arxiv.org/abs/2211.06956>

Seeing Beyond the Brain



<https://arxiv.org/abs/2211.06956>