DIFFUSION MODEL

Generative Model

CHR_Diffusion Team

01

Context of the project

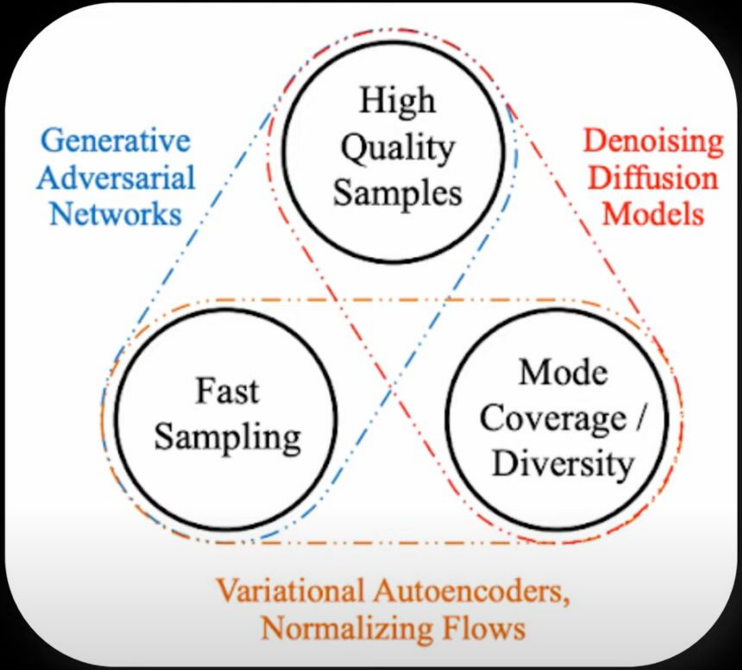
Diffusion Model

- Inspired by non-equilibrium thermodynamics
- They define a Markov chain of diffusion steps
- The latent variable has high dimensionality
- Fixed procedure

Diffusion Model has two main process

- Forward diffusion process
- Reverse diffusion process

Comparison between GMs



Source: Nvidia Developer Blog

Diffusion Probabilistic model 02

Conception and implementation of a first DPM

Forward Diffusion Process

Make images noisily with Gaussian Distribution.

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

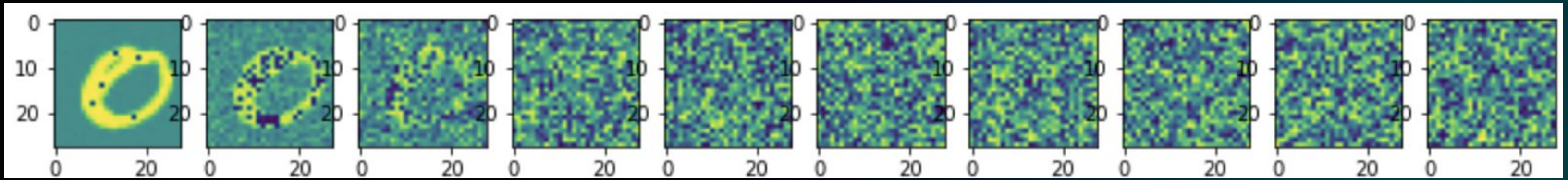
Reverse Diffusion Process

- Unet

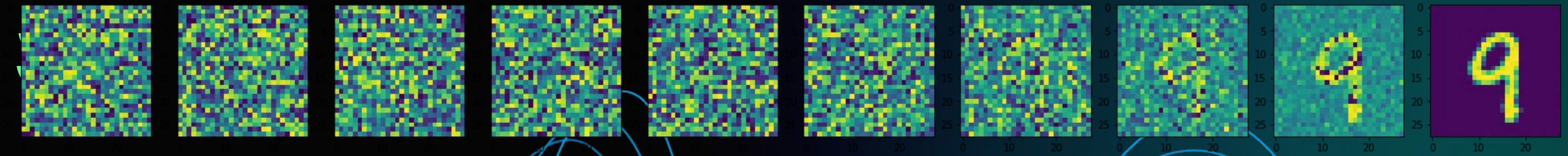
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

Forward Diffusion Process



Reverse Diffusion Process

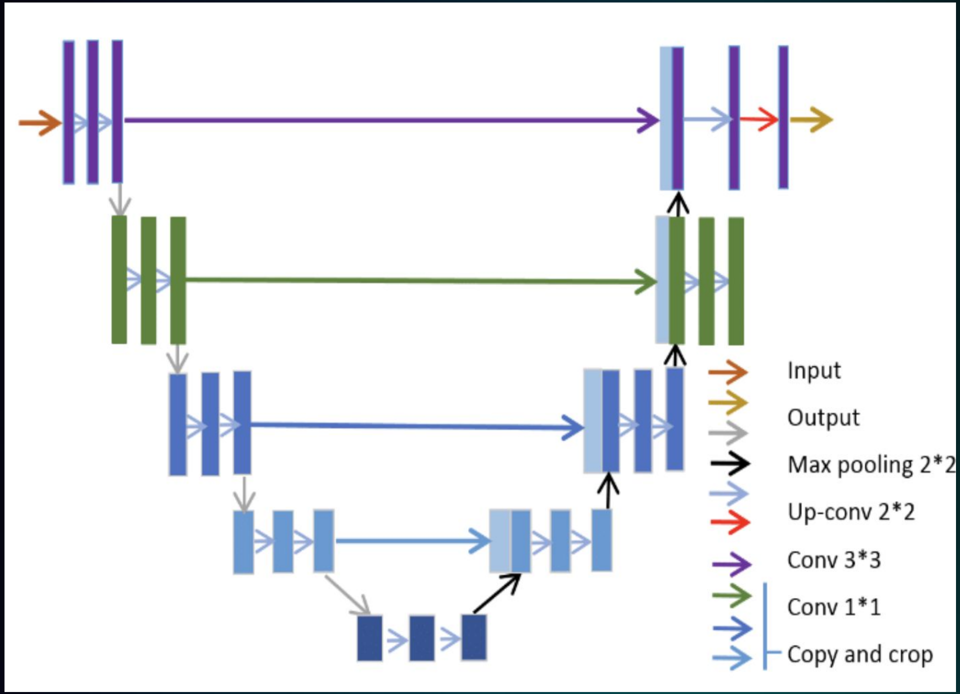


UNet Model

UNet Contain Convolution Layers for extracted features from images :

- Convolution Layers
  - ➔ Down-sampling (Conv)
  - ➔ Up-sampling (Up-Conv)

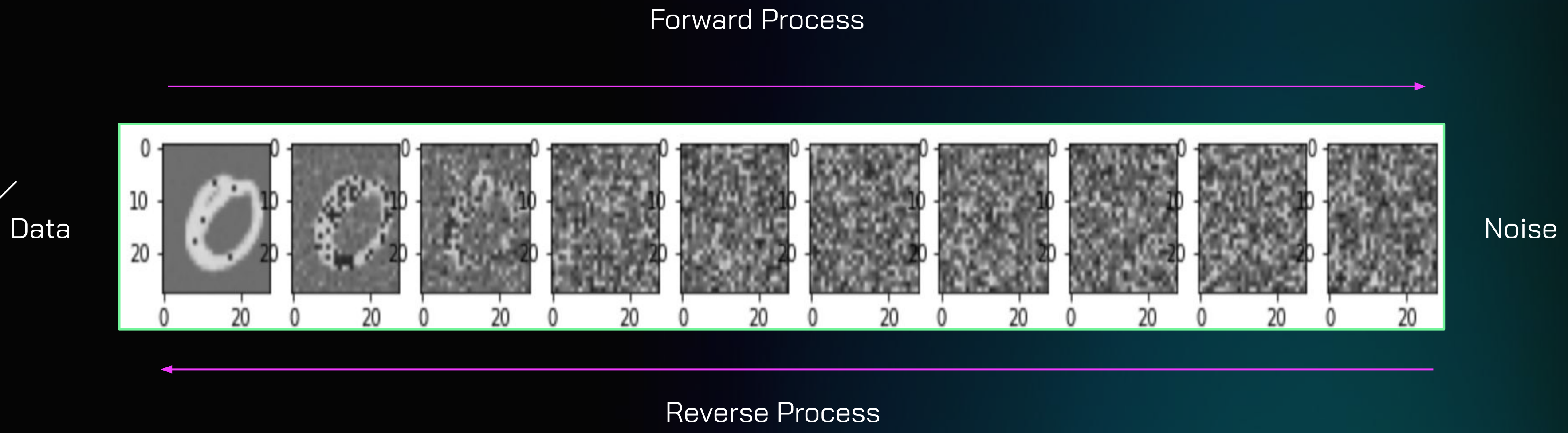
- Batch Normalization
- Skip Connection between
- Down-sampling and Up-sampling.
- Pooling for reduce size of images after extracted or conv images.
- Attention layer
- Positional embedding



03

Improvement with SDE

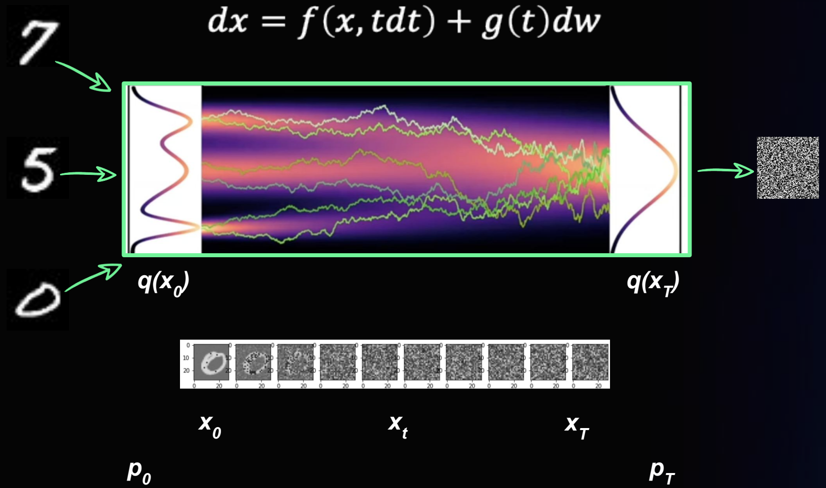
Diffusion model improvements



Differential Diffusion model

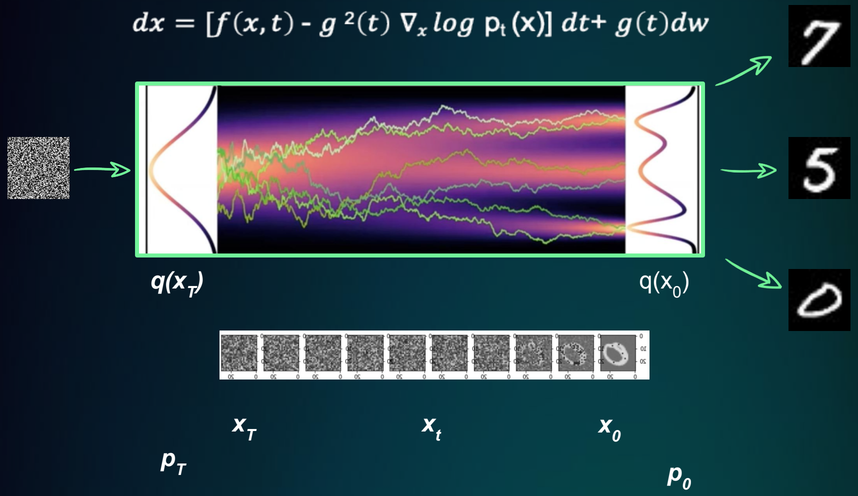
Forward Diffusion Process

$$dx = f(x, tdt) + g(t)dw$$

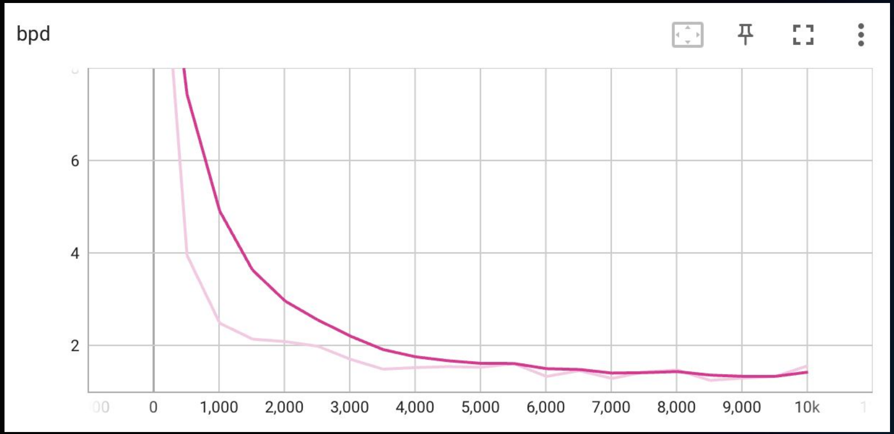


Reverse Diffusion Process

$$dx = [f(x,t) - g^2(t)\,\nabla_x \log p_t(x)]\,dt + g(t)dw$$



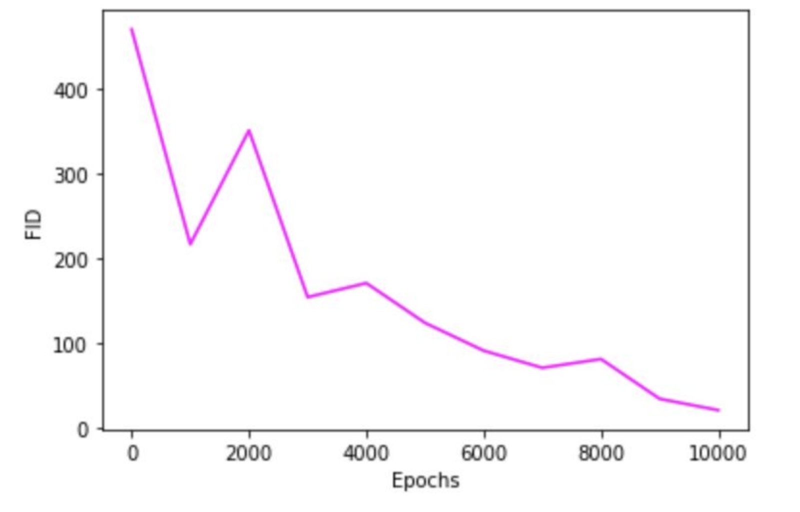
Bits per dimension



BDP

The negative log_likelihood lower value better

Fréchet Inception Distance



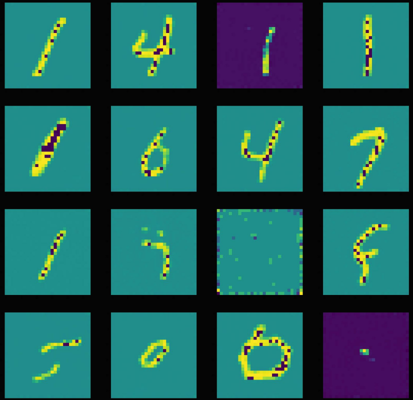
FID

Over epochs
FID = 25

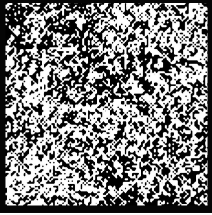
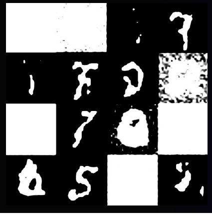
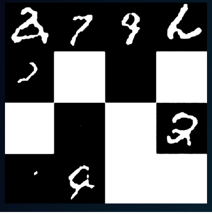
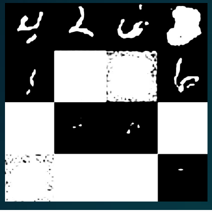
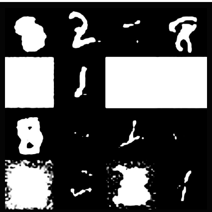
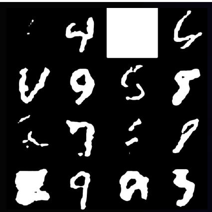
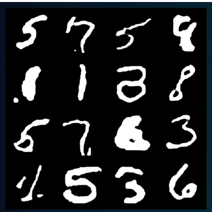
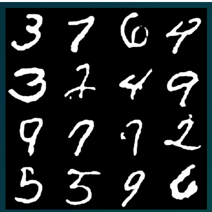
Comparison of results 04

Loss, Time and FID

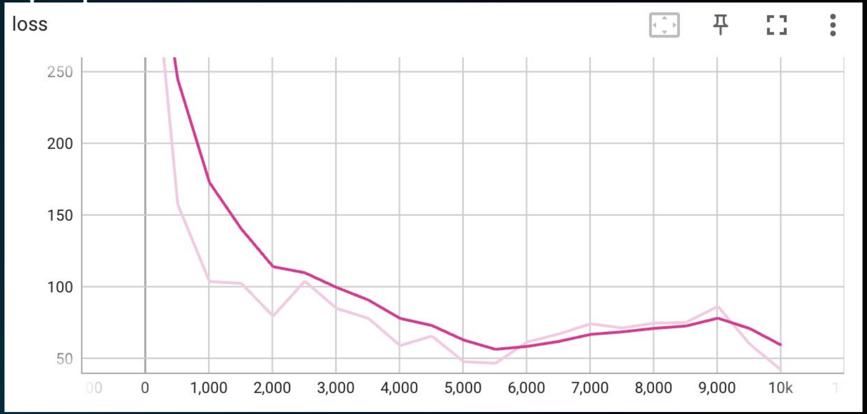
Samples of results
Basic Unet

100 epoch	200 epoch	300 epoch
		

Samples of results
SDM

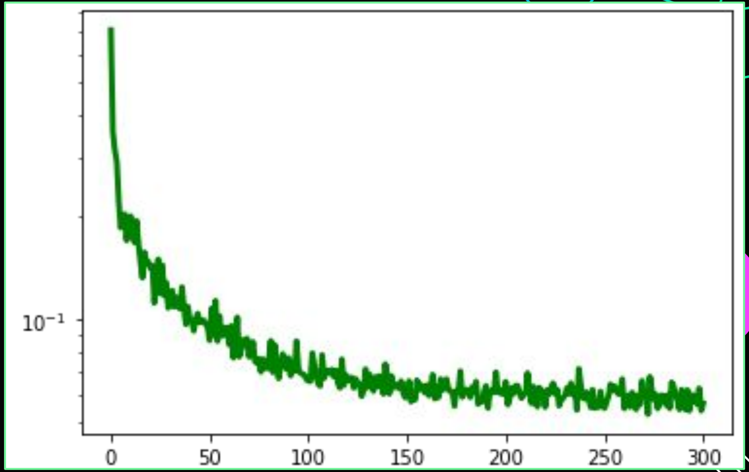
500 epoch	1K epoch	2K epoch	3K epoch
			
4K epoch	5K epoch	6K epoch	7K epoch
			

Loss and FID comparaison



SDM Unet

5500 epochs :
loss : 0.05
time : 660 s



Basic Unet

300 epochs :
loss : 0.056
time : 13122 s

Thanks

Resources

Papers

[Paper 1] https://arxiv.org/pdf/1503.03585.pdf
[Paper 2] https://arxiv.org/pdf/2006.11239.pdf
[Paper 3] https://arxiv.org/pdf/2105.05233.pdf
[Paper 4] https://arxiv.org/pdf/2102.09672.pdf
[Paper 5] https://arxiv.org/pdf/2209.00796.pdf
[Paper 6] https://arxiv.org/pdf/2206.02262.pdf

Article

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
https://towardsai.net/p/l/gan-is-diffusion-all-you-need

Videos

https://www.youtube.com/watch?v=a4Yfz2FxXiY
https://www.youtube.com/watch?v=cS6JQpEY9cs

Paper Explained

https://www.youtube.com/watch?v=W-O7AZNzbzQ

Math Explained

https://www.youtube.com/watch?v=HoKDTa5jHvg

Jeremy Howard

https://www.youtube.com/watch?v=_7rMfsA24Ls&feature=youtu.be
https://www.youtube.com/watch?v=0_BBRNYInx8
https://www.youtube.com/watch?v=mYpjmM7O-30

Code

shttps://github.com/lucidrains/denoising-diffusion-pytorc
hhttps://colab.research.google.com/drive/1DPrTwr3ZnUxBHAUiYBKP
9m9EFxVxHhR8
https://github.com/NVlabs/stylegan3