Stable diffusion class models: development and application

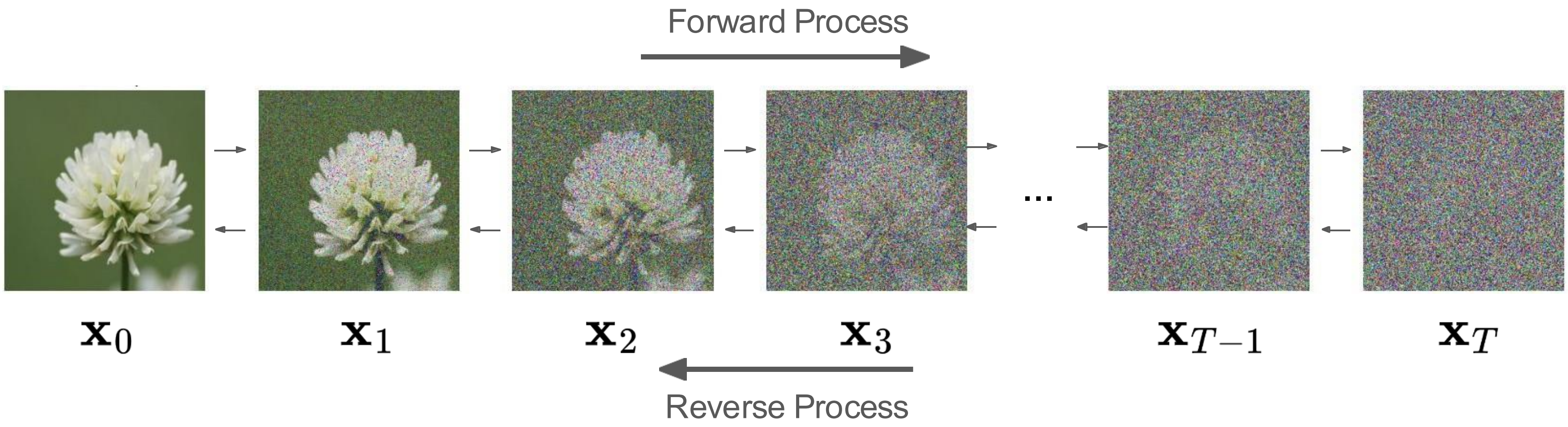
Blagodarniy Artyom

stability.ai

Denoising Diffusion Models

Denoising diffusion models consist of two processes:

- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising

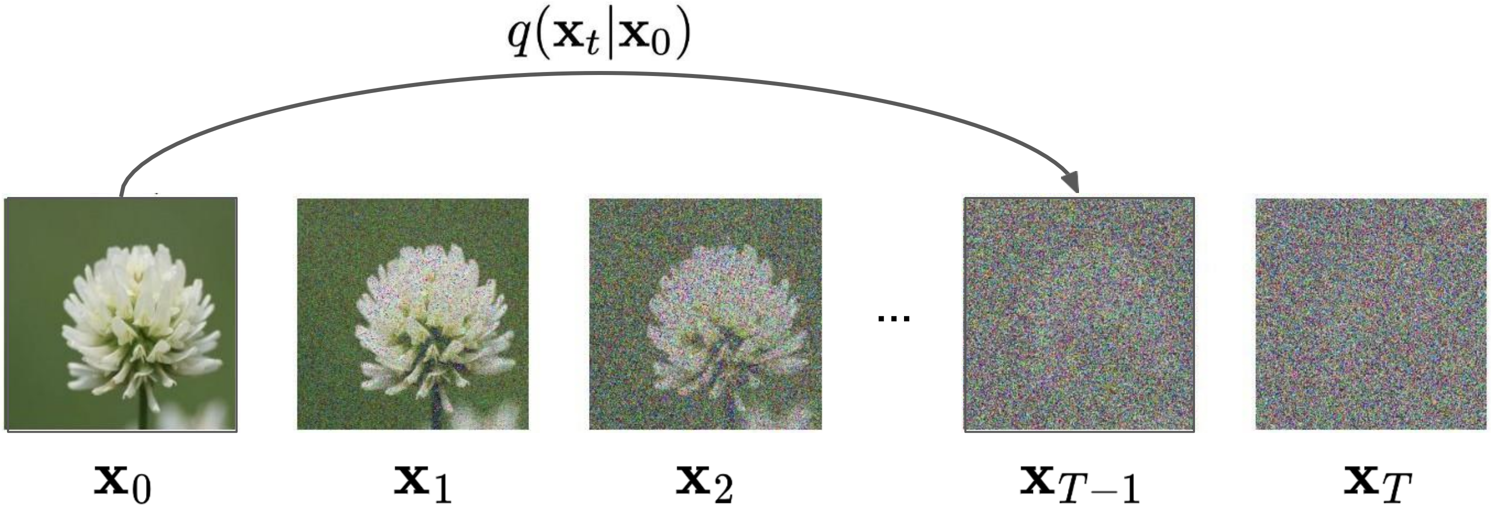


Details: Forward Process

Can sample $\mathbf{x}_t$ in closed-form as $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \bar{\alpha}_t \in (0,1)$$



Aside: Noise Schedules

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \bar{\alpha}_t \in (0,1)$$

- Define the noise schedule in terms of $\bar{\alpha}_t \in (0,1)$
  - Some monotonically decreasing function from 1 to 0
- Cosine Noise schedule:

$$\bar{\alpha}_t = \cos(.5\pi t/T)^2$$

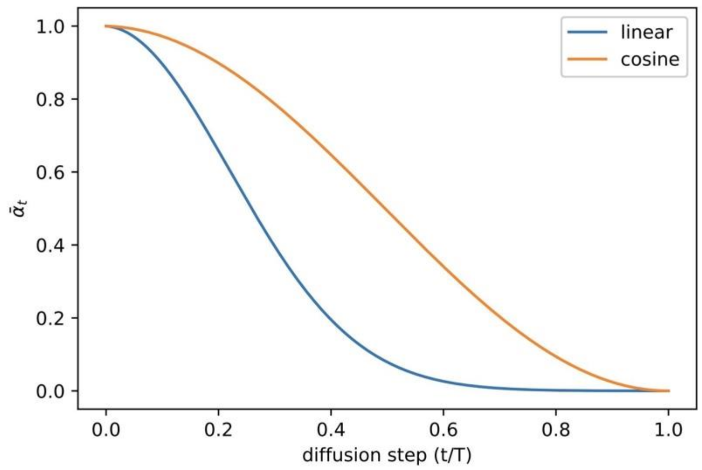


Figure 5. $\bar{\alpha}_t$ throughout diffusion in the linear schedule and our proposed cosine schedule.

Nichol, Alexander Quinn, and Prafulla Dhariwal. "Improved denoising diffusion probabilistic models." International conference on machine learning. PMLR, 2021.

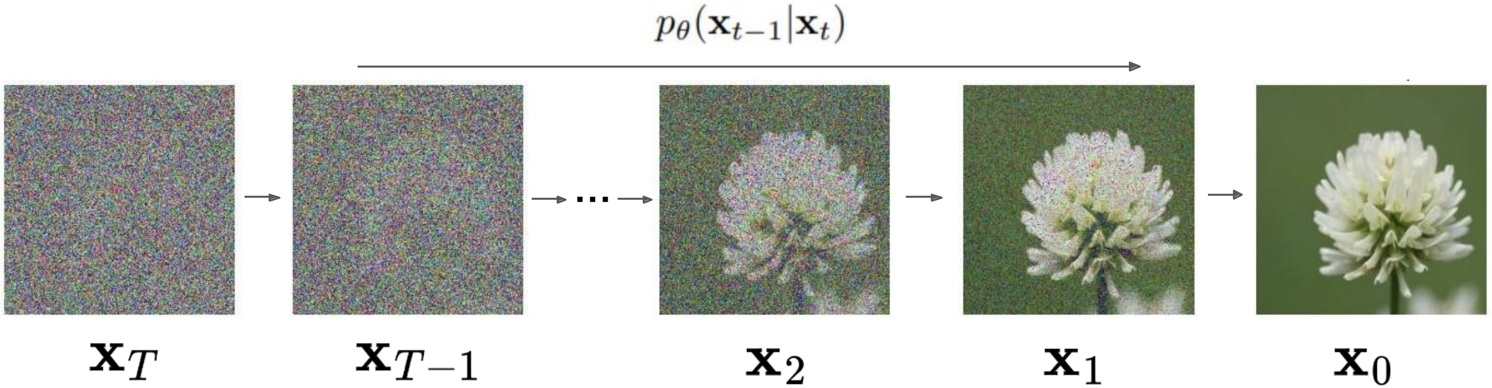
Cornell Bowers C·IS

Key Idea

We introduce a generative model to approximate the reverse process:

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$



Training Objective

- Bound the likelihood with the ELBO
  - Exactly like VAEs

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right]$$

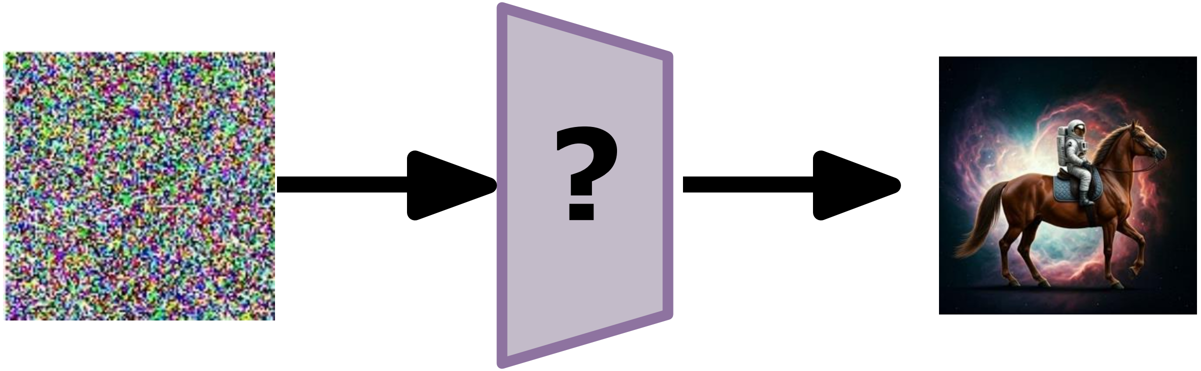
$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}_{\text{denoising matching term}}$$

http://cs231n.stanford.edu/slides/2023/lecture_15.pdf

Diffusion Models

[Sohl-Dickstein et al. 2015]

- Draw a sample of Gaussian noise
- Transform it to obtain a natural image



The U-Net

[Ronneberger et al. 2015]

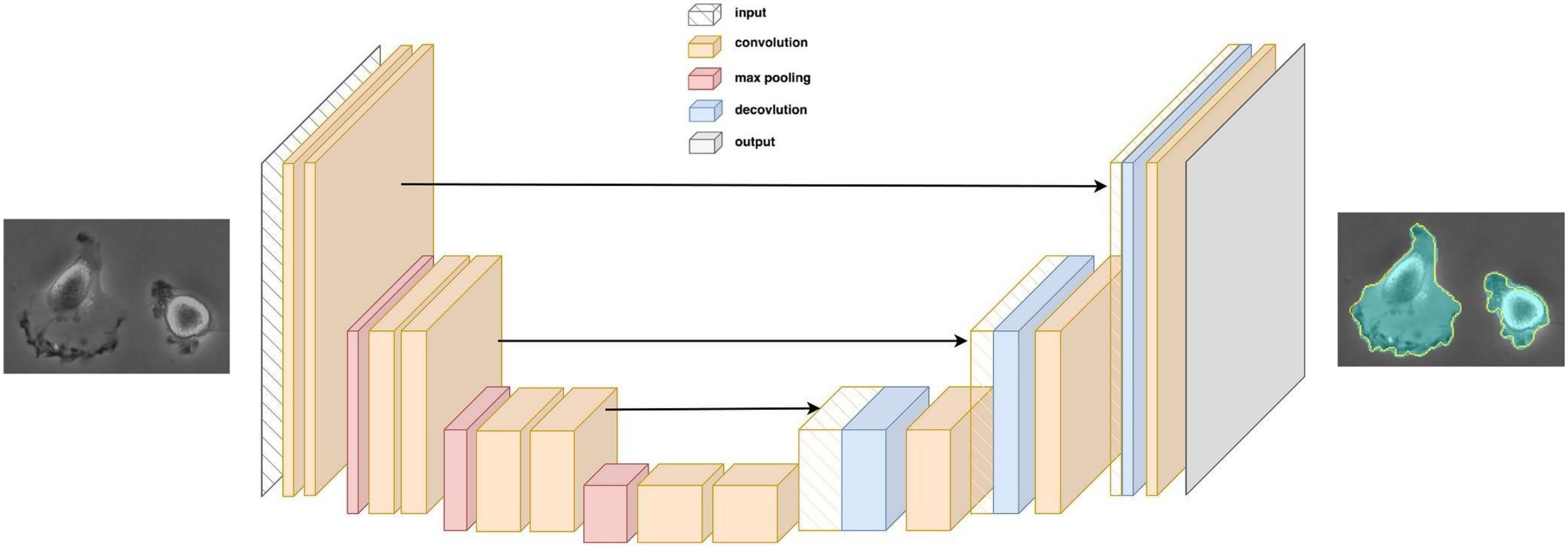


Image Source:
https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

The U-Net

[Ronneberger et al. 2015]

$\sigma = 0.75$

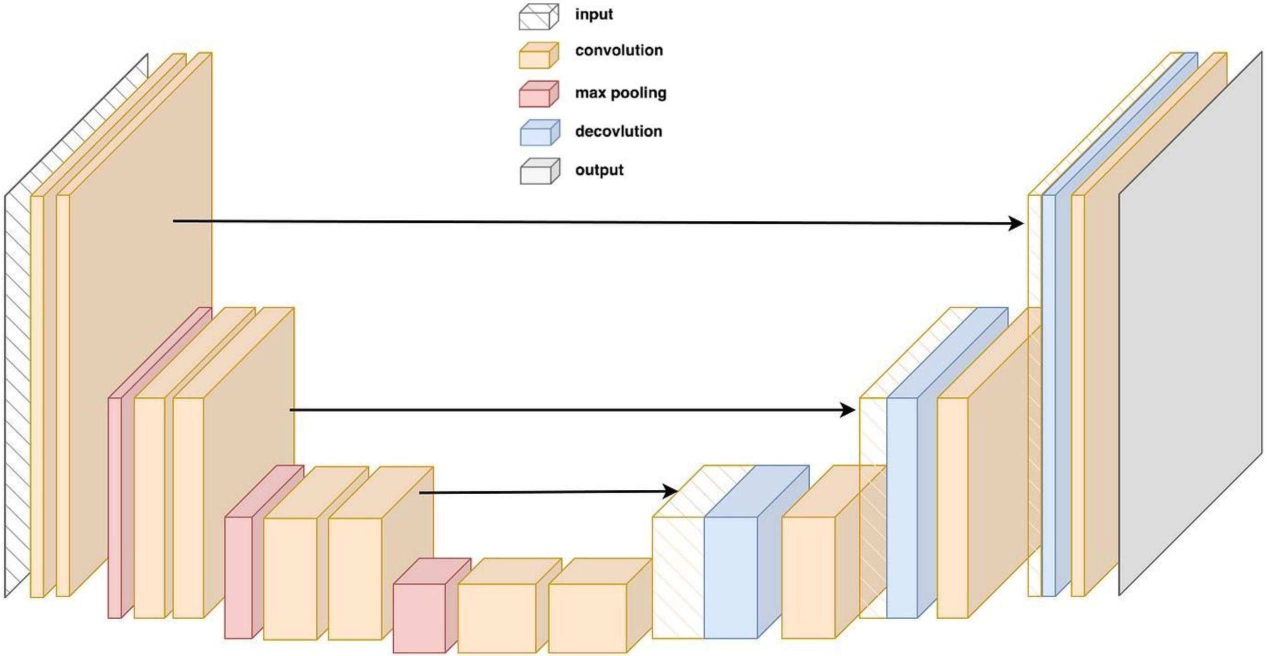


Image Source:
https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

The U-Net

[Ronneberger et al. 2015]

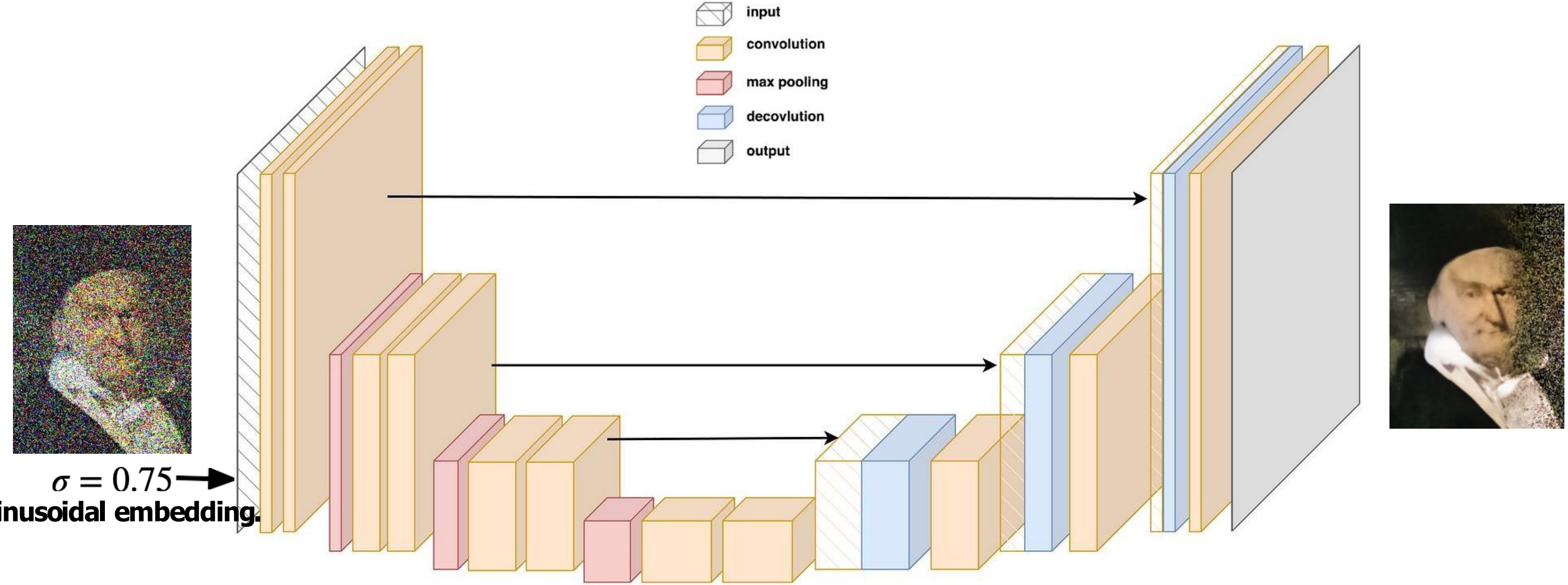
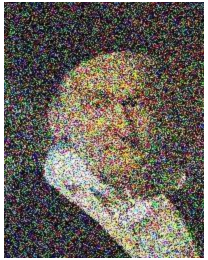


Image Source:
https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wcHHFp_fQZQ.png

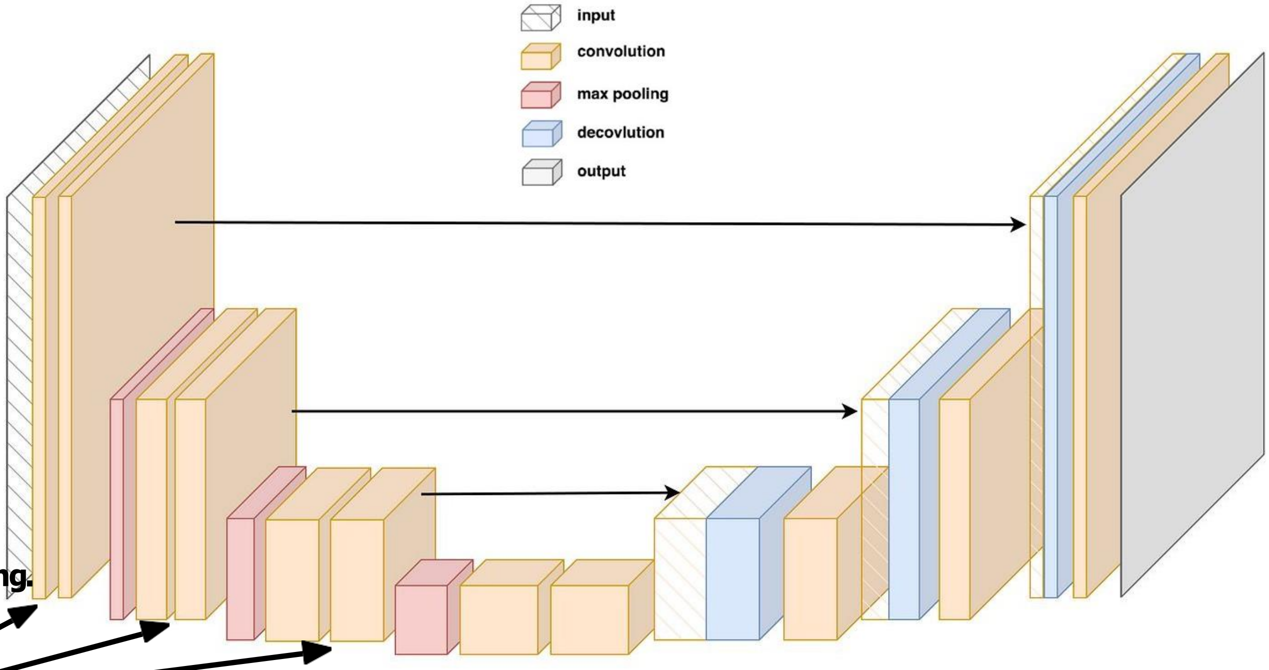
The U-Net

[Ronneberger et al. 2015]



$\sigma = 0.75$

Sinusoidal embedding.

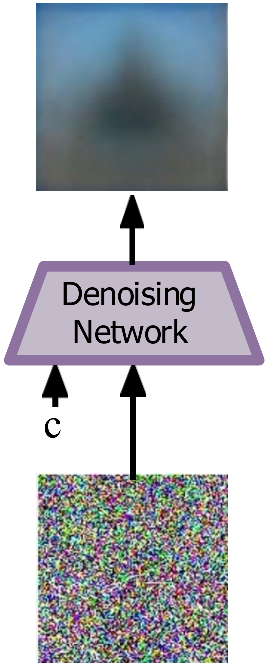


c="An oil painting of a old white man with a white shirt, black hat, and gray side burns."

Image Source:
https://miro.medium.com/v2/resize:fit:1400/1*VUS2cCaPB45wdHHEp_fOZQ.png

c="An astronaut riding a horse"

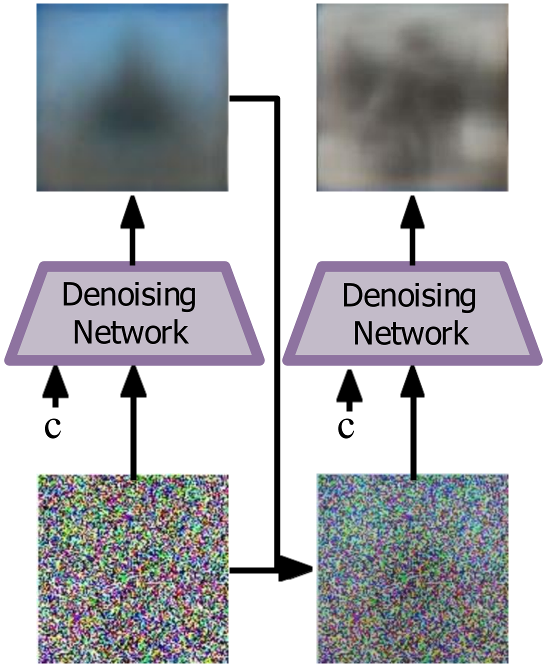
Diffusion Sampling



$$\mathbf{x}_T = \sigma_T \epsilon_T$$

c=“An astronaut riding a horse”

Diffusion Sampling

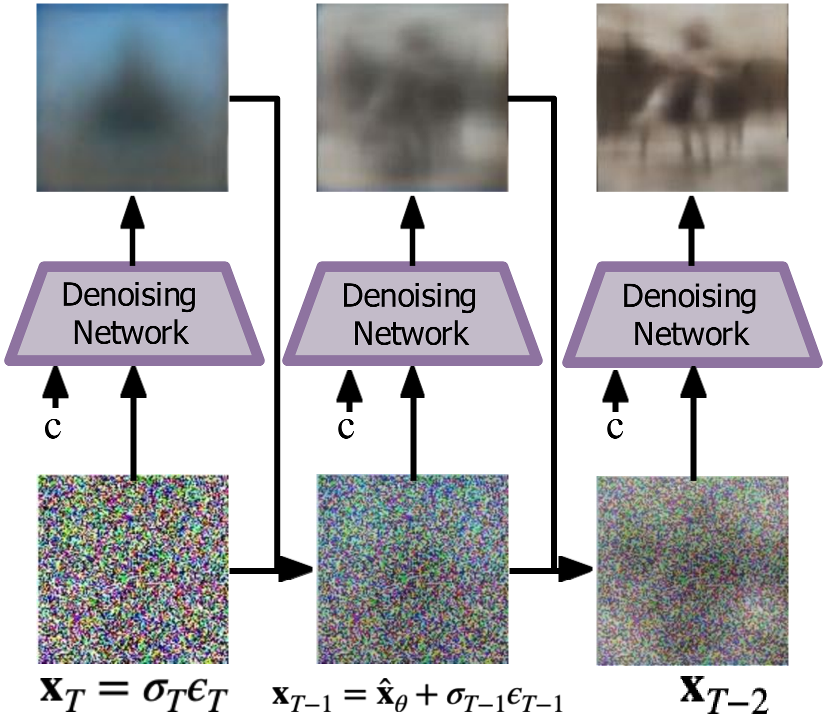


$\mathbf{x}_T = \sigma_T \epsilon_T$ $\mathbf{x}_{T-1} = \hat{\mathbf{x}}_\theta + \sigma_{T-1}\epsilon_{T-1}$

$$\boxed{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{x}_\theta(\mathbf{x}_t, t)) \approx q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{x}_0)}$$

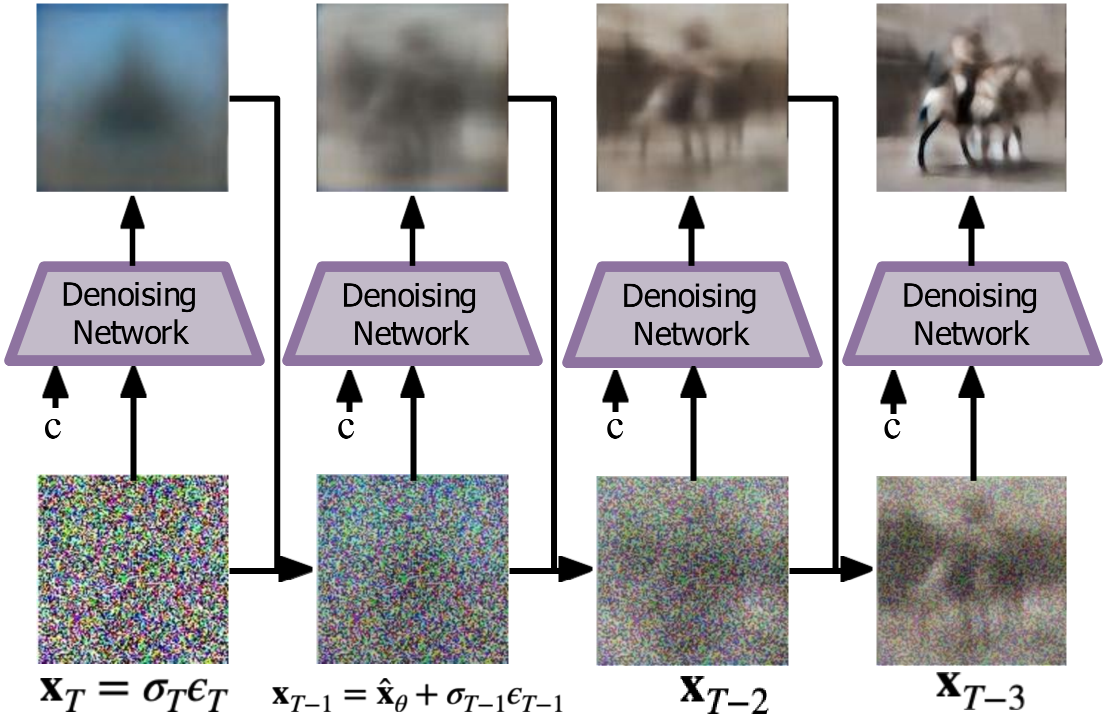
c="An astronaut riding a horse"

Diffusion Sampling



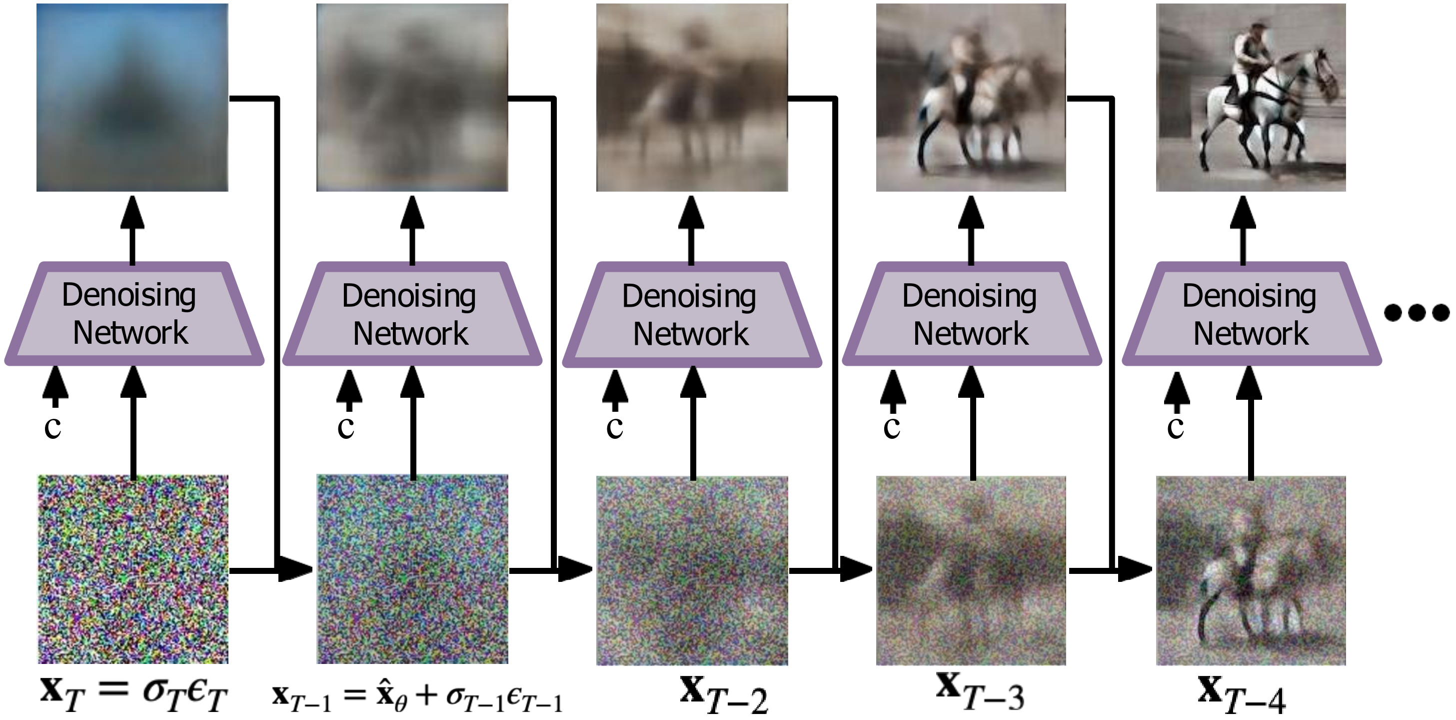
c="An astronaut riding a horse"

Diffusion Sampling



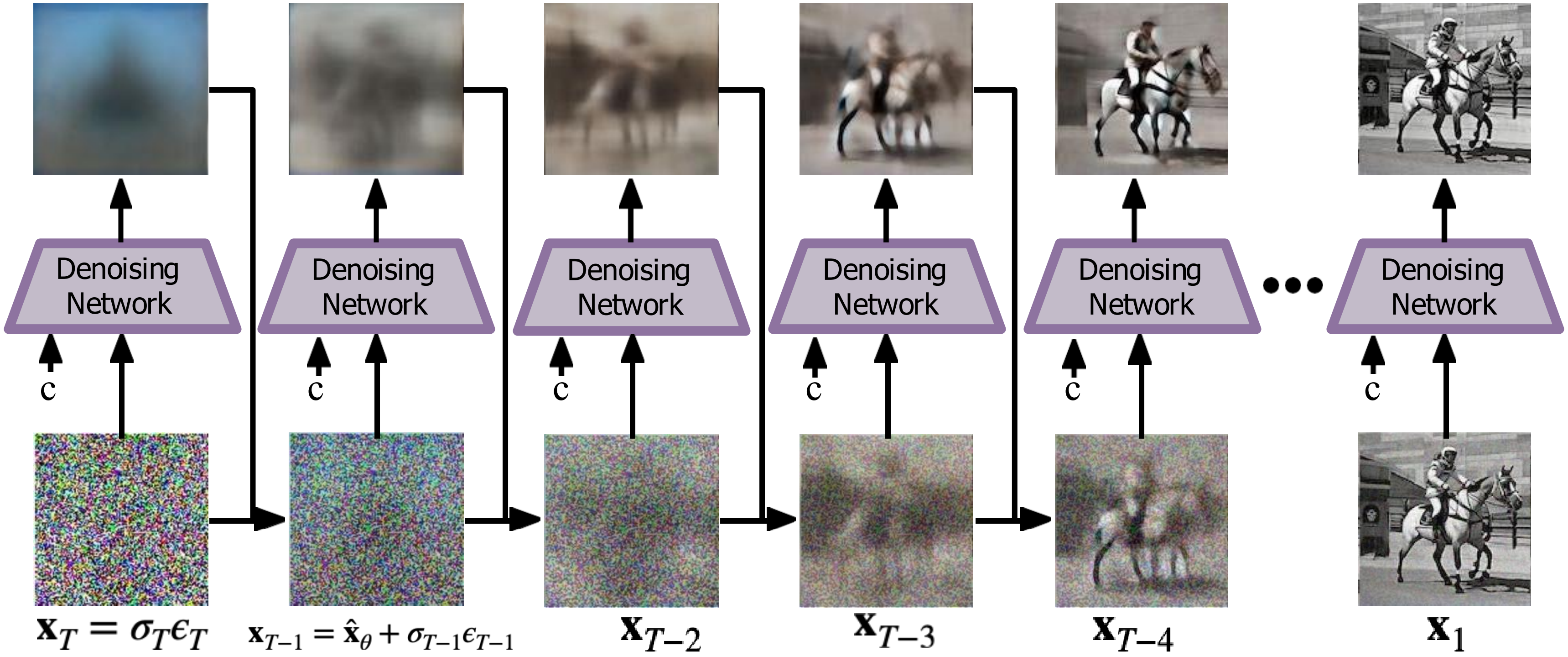
c="An astronaut riding a horse"

Diffusion Sampling



c="An astronaut riding a horse"

Diffusion Sampling



Diffusion Models

An astronaut riding a horse

Photo Credit: Lin, Shanchuan, et al. "Common diffusion noise schedules and sample steps are flawed." *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2024.

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Training Algorithm

Repeat until convergence

1. $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ ← Sample original image from image distribution
2. $t \sim U\{1, 2, \ldots, T\}$ ← Sample random time step uniformly
3. $\epsilon \sim \mathcal{N}(0, 1)$ ← Sample Gaussian noise
4. Optimizer step on $L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2]$

← Model predicts noise applied at time step t and calculate loss

Inference Sampling Algorithm

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ← Sample pure Gaussian noise

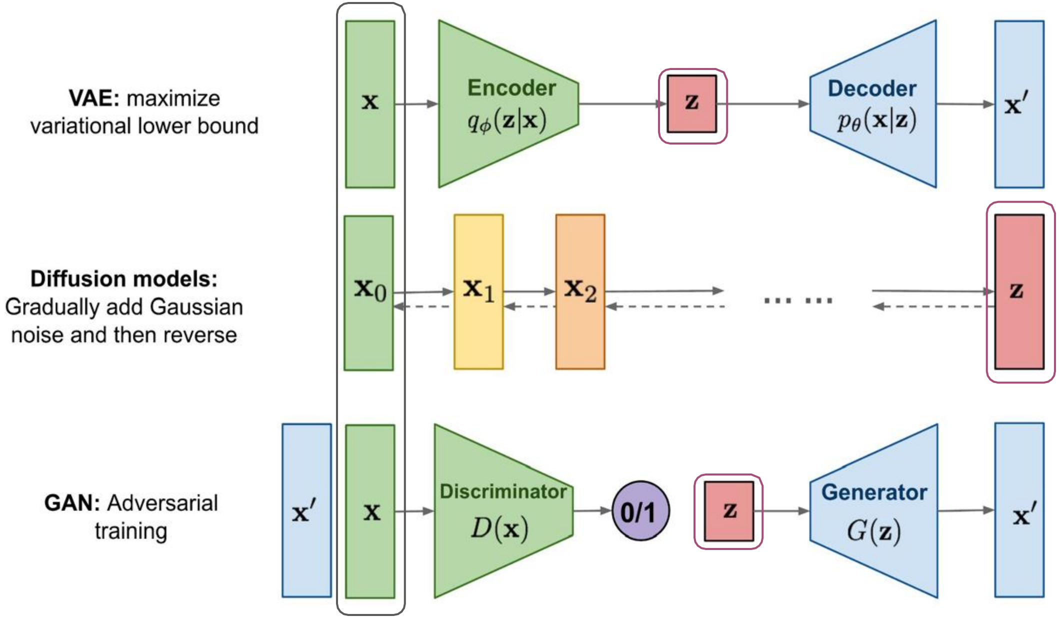
For $t = T, T-1 \ldots, 1$

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$ ← Sample Gaussian noise to apply to image

$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$ ←Predict noise applied to image and remove that noise

Return $\mathbf{x}_0$

Generative Modeling



VAEs typically have a smaller latent dimension, while diffusion models do not

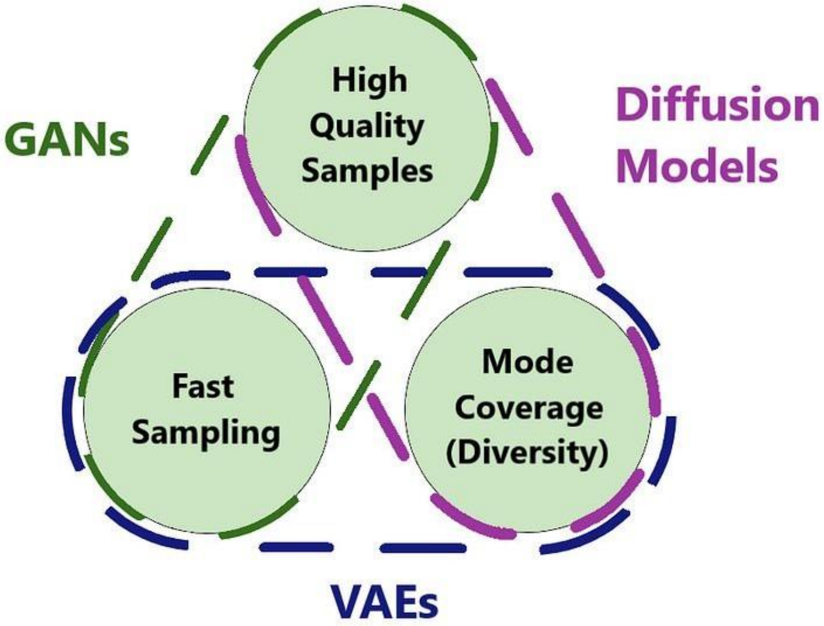
Latent Distribution

Target Distribution

Image Source

Diffusion Models vs. VAEs vs. GAN

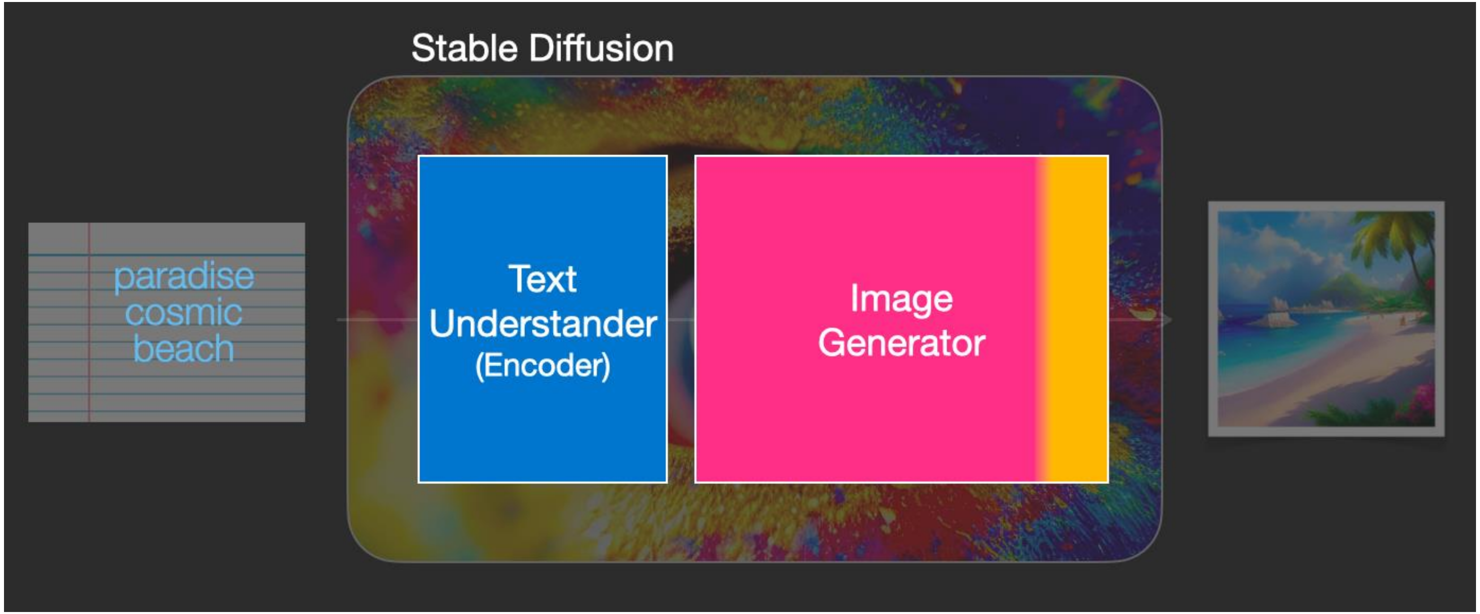
Source



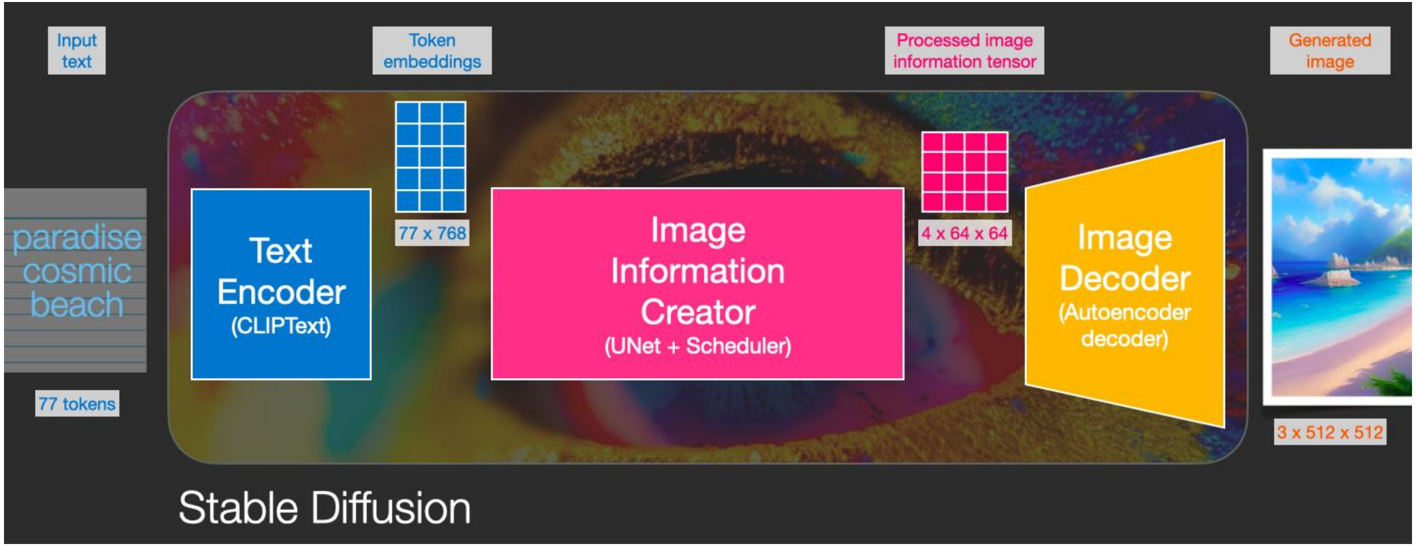
Stable Diffusion

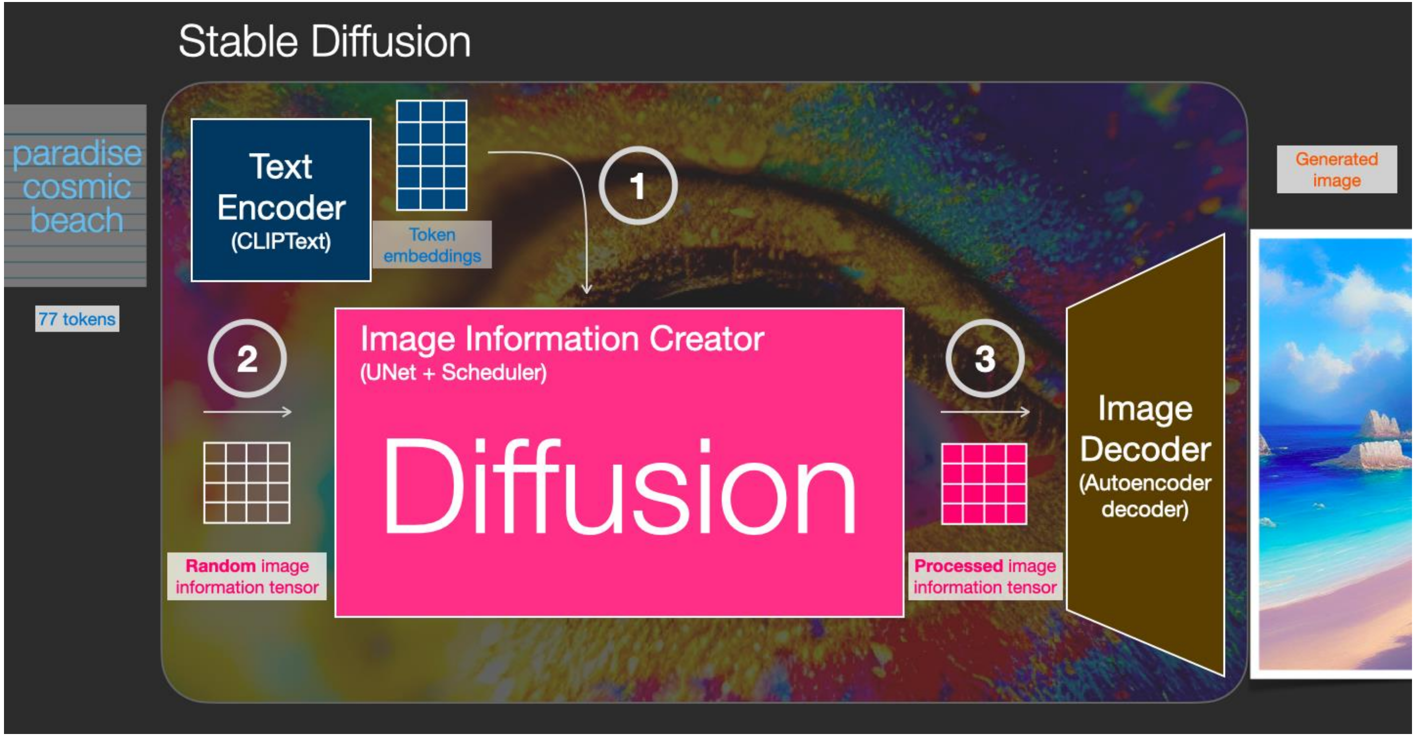
Stable Diffusion is a deep learning, text-to-image model released in 2022 based on diffusion techniques

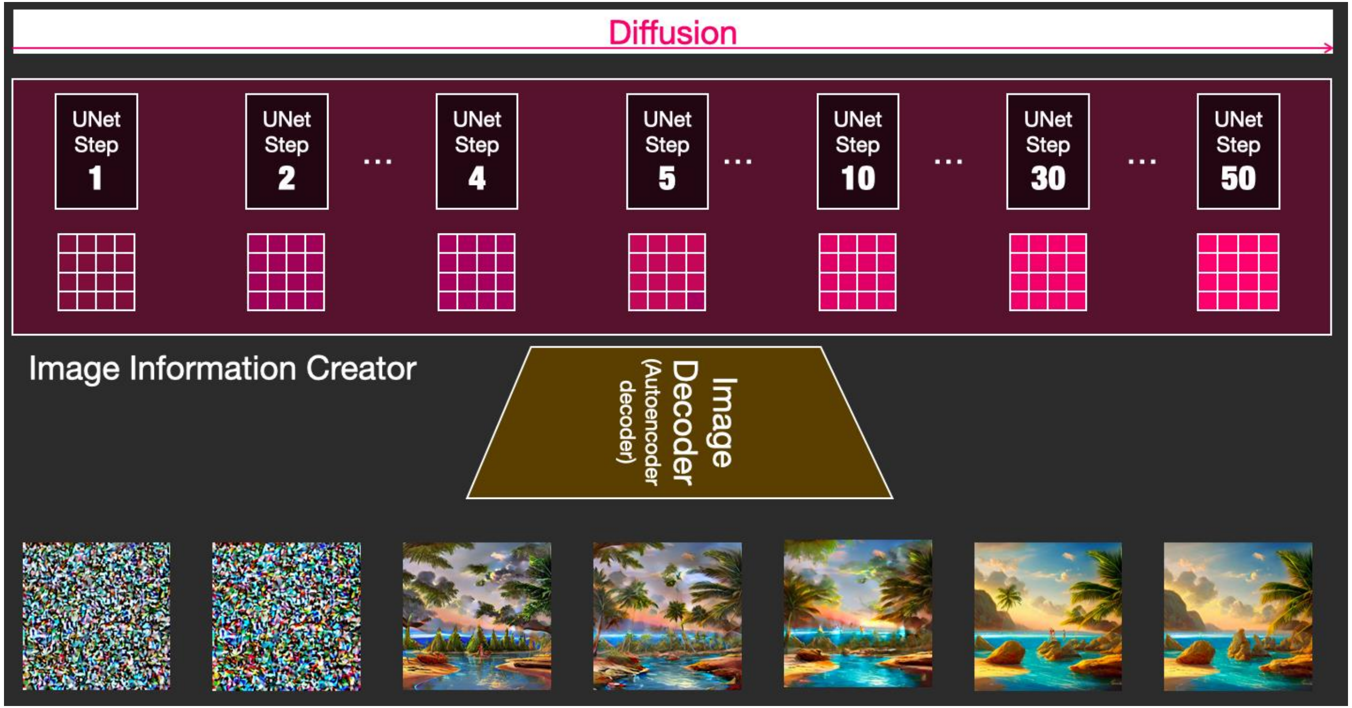
Text encoder is a special Transformer language model (CLIP model)



ClipText for text encoding.
UNet + Scheduler to gradually process/diffuse information in the information (latent) space.
Autoencoder Decoder that paints the final image using the processed information array.




Stable Diffusion
paradise
cosmic
beach
77 tokens
Text
Encoder
(CLIPText)
Token
embeddings
1
2
Image Information Creator
(UNet + Scheduler)
Diffusion
3
Image
Decoder
(Autoencoder
decoder)
Generated
image
Random image
information tensor
Processed image
information tensor


Diffusion
UNet Step 1
UNet Step 2
UNet Step 4
UNet Step 5
UNet Step 10
UNet Step 30
UNet Step 50
Image Information Creator
Image Decoder (Autoencoder decoder)

Application Stable Diffusion

Sample input: "messi as a real madrid player"

Application Stable Diffusion

Prompt: 3D animation of a small, round, fluffy creature with big, expressive eyes explores a vibrant, enchanted forest.

Prompt: A cat waking up its sleeping owner demanding breakfast.