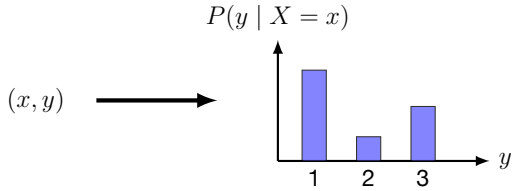
Diffusion Models

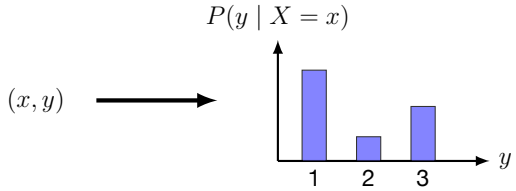
Review of Supervised Learning

- Classification

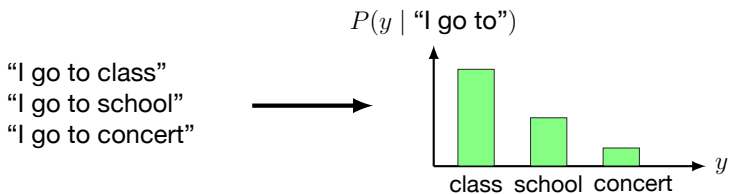


Review of Supervised Learning

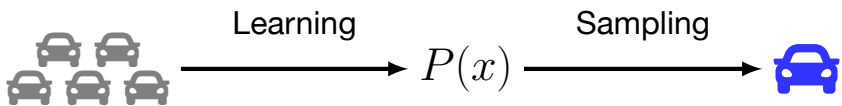
- Classification



- Text generation

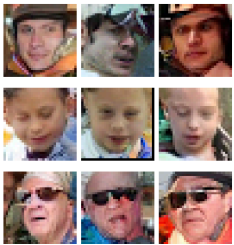


Generative Models



Progress in Image Generation

PixelCNN
(2016)



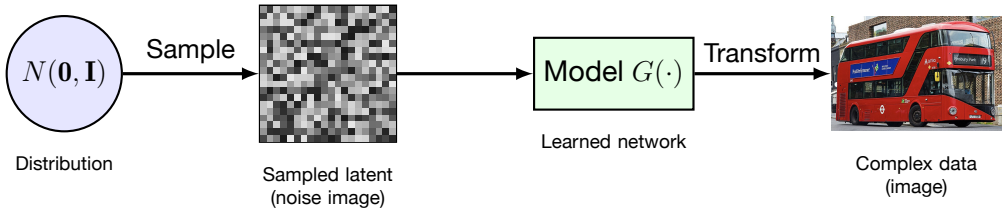
StyleGAN
(2019)

Imagen
(2022)

Stable Diffusion
(2022)

'A painting of a squirrel eating a burger'

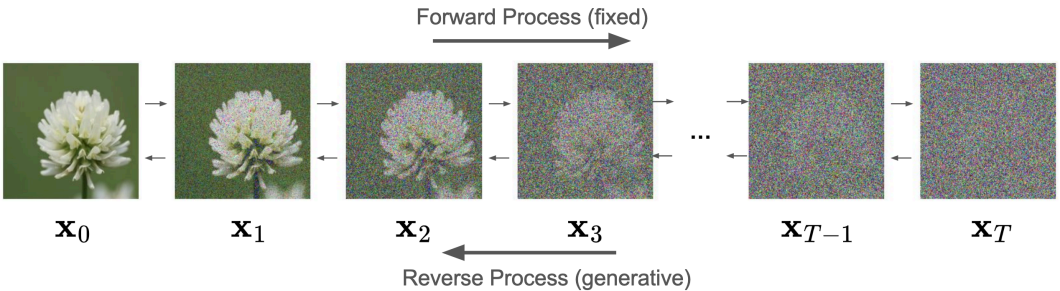
Generation as Transformation



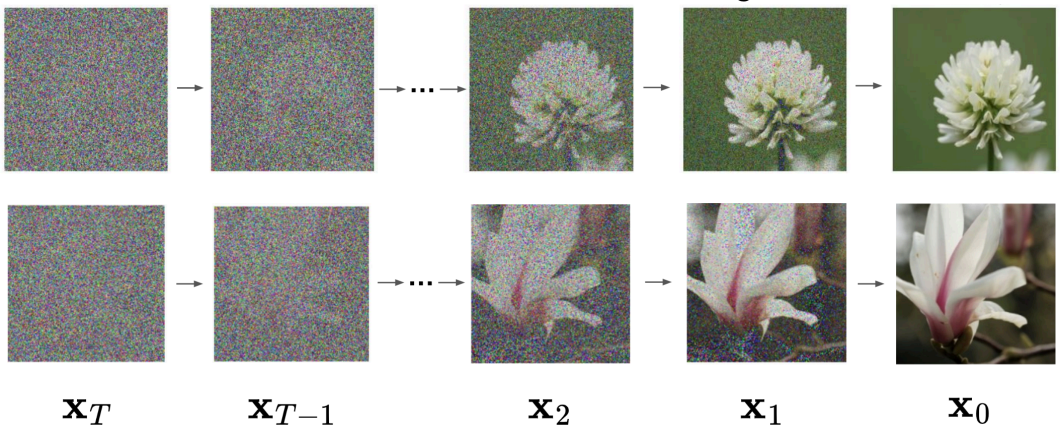
Generative models by learning a transformation from random noises.

Denoising Diffusion Probabilistic Models

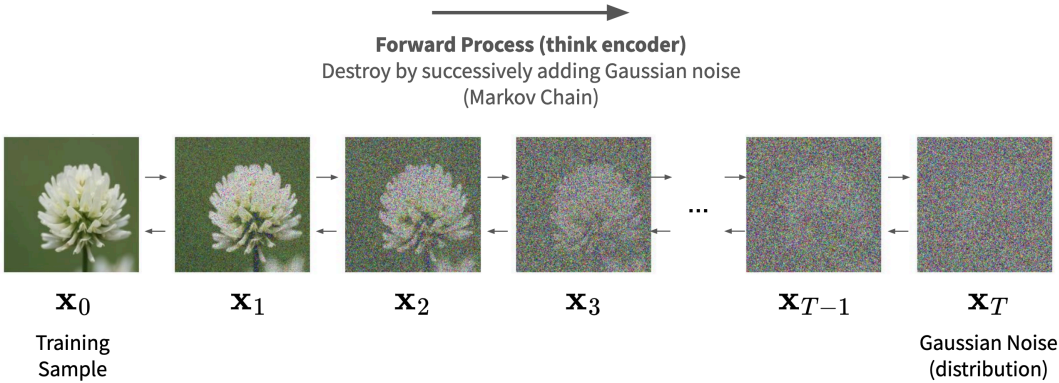
Ho, Jain & Abbeel (2020)



Denoising Diffusion Probabilistic Models



Forward Process

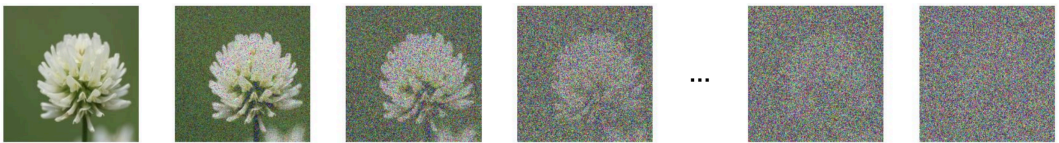


No training yet!

Forward Process

Sample $\boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1}$$

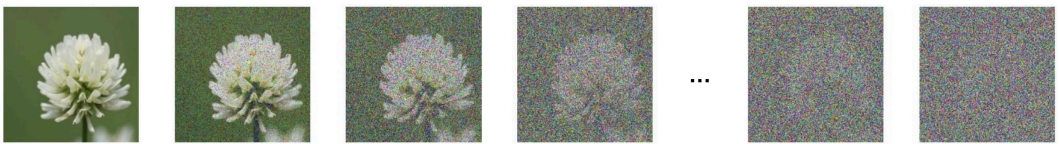


Forward Process

Let $\alpha_t = 1 - \beta_t$

Sample $\boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

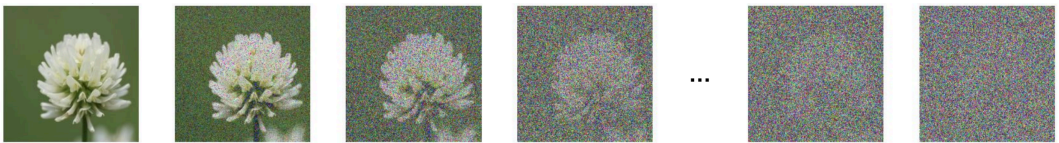
$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$



Forward Process

Sample $\boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

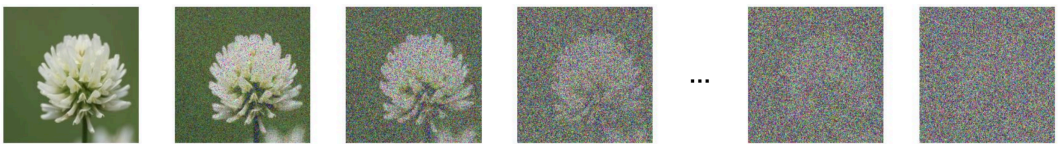
$$\mathbf{x}_t = \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$



Forward Process

Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

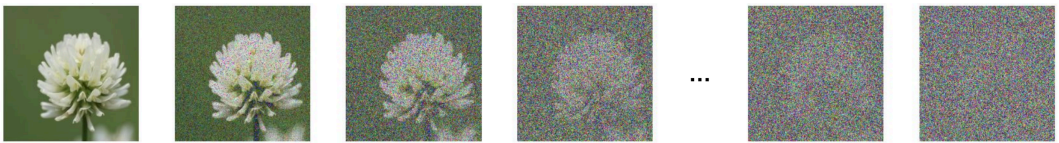
$$\mathbf{x}_t = \sqrt{\alpha_t \alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}}\boldsymbol{\epsilon}$$



Forward Process

Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\mathbf{x}_t = \sqrt{\alpha_1 \dots \alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_1 \dots \alpha_t}\boldsymbol{\epsilon}$$

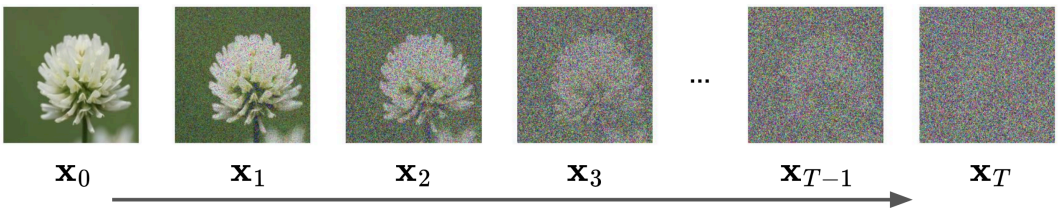


Forward Process

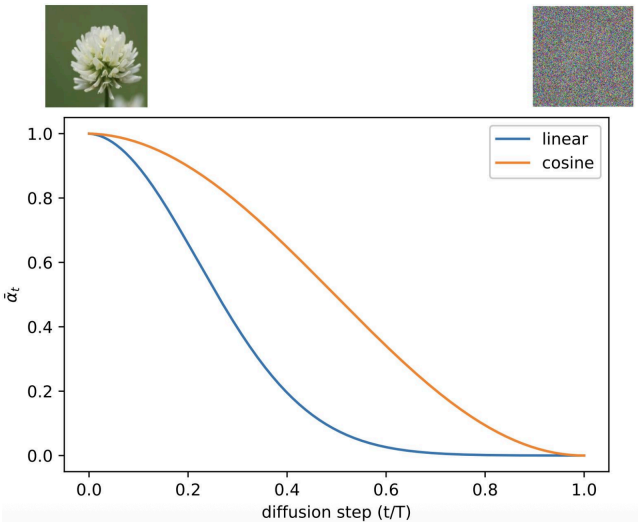
Let $\bar{\alpha}_t = \alpha_1 \dots \alpha_t$

Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$$

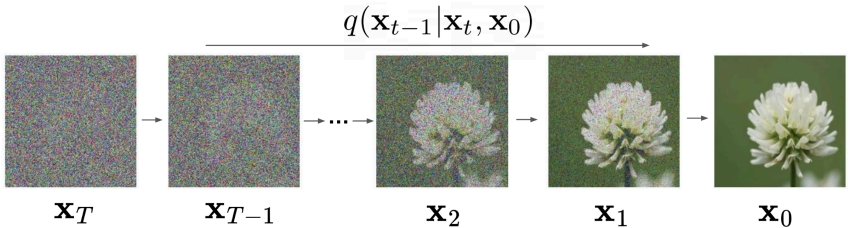


Noise Schedule



The Reverse Process

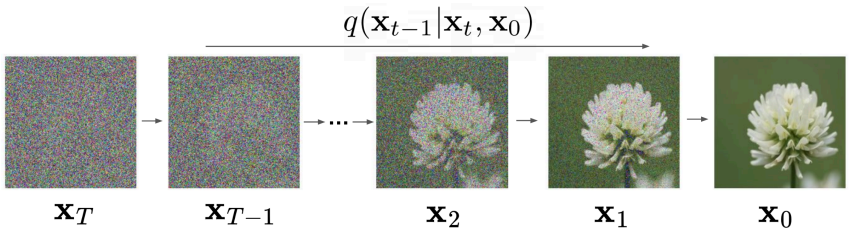
Can we compute $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$?



The Reverse Process

Yes! If we know the original image $\mathbf{x}_0$, we can perfectly reverse the diffusion process

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$



Deriving $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ (Part 1)

Let's substitute our known forward distributions into Bayes' Rule:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

Recall from the forward process:

$$\begin{aligned} q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}) \\ q(\mathbf{x}_{t-1}|\mathbf{x}_0) &= \mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}) \\ q(\mathbf{x}_t|\mathbf{x}_0) &= \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \end{aligned}$$

Since all three terms are Gaussian, their product and quotient yield another Gaussian!

Deriving $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ (Part 2)

By expanding the exponential terms of these Gaussians and completing the square for $\mathbf{x}_{t-1}$, we find that:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \sigma_t^2 \mathbf{I}\right)$$

where the mean $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0)$ and the variance σ_t^2 can be written as:

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$$

$$\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}(1-\alpha_t)$$

What's Next?

- We have shown that $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ can be easily computed
- **However**, the probability requires $\mathbf{x}_0$. During inference (when generating a new image), we only have pure noise $\mathbf{x}_T$. We don't have the original data!
- **The Solution:** Train a neural network $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$.

Approximating the Reverse Process

We introduce a learned generative model p_θ to approximate the true (but intractable) reverse process $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

- Start at pure noise:

$$p(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \mathbf{0}, \mathbf{I})$$

- Learned reverse transition:

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

Here, we only train the network to predict the mean $\mu_\theta(\mathbf{x}_t, t)$.

Parameterizing the Mean

Let's look at the true mean equation:

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right)$$

Since $\mathbf{x}_t$ is known during training, the only unknown variable in this equation is the noise $\boldsymbol{\epsilon}$ that was added. Therefore, we design our neural network to predict the noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right)$$

The Simplified Loss

Matching the true noise $\boldsymbol{\epsilon}$ and the prediction from the neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$,

We arrive at a final loss function:

$$\mathcal{L}_{\text{DDPM}}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\right]$$

The Training Algorithm

Repeat until convergence:

1. Sample an original image: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
2. Sample a random timestep: $t \sim U\{1, 2, \ldots, T\}$
3. Sample pure Gaussian noise: $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
4. Compute the noisy image at step t:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$$

5. Take a gradient descent step on the MSE loss:

$$\theta \longleftarrow \theta - \eta \nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right\|^2$$

The Sampling Algorithm

1. Sample noise: $\mathbf{x}_t \sim \mathcal{N}(0, \mathbf{I})$
2. **For** $t = T, T-1, \ldots, 1$:
   - Sample $z \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $z = 0$
   - Predict the noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, calculate the mean $\mu_\theta(\mathbf{x}_t, t)$ and add the noise:

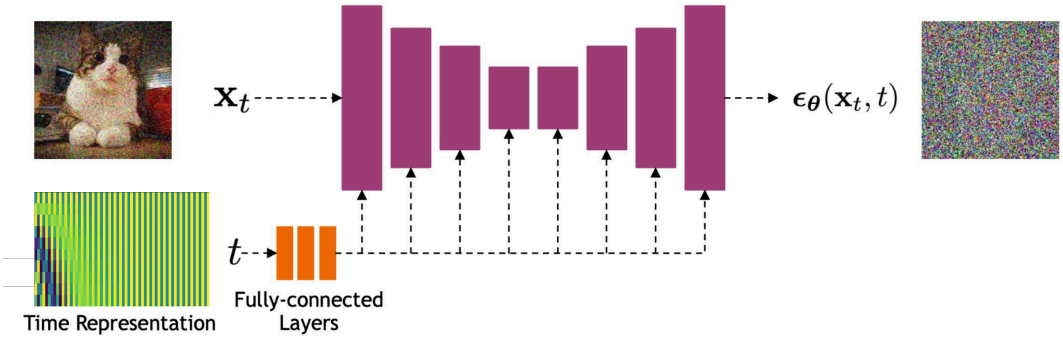
$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t z$$

3. **Return** $\mathbf{x}_0$

Network Architecture

How do we build $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$?

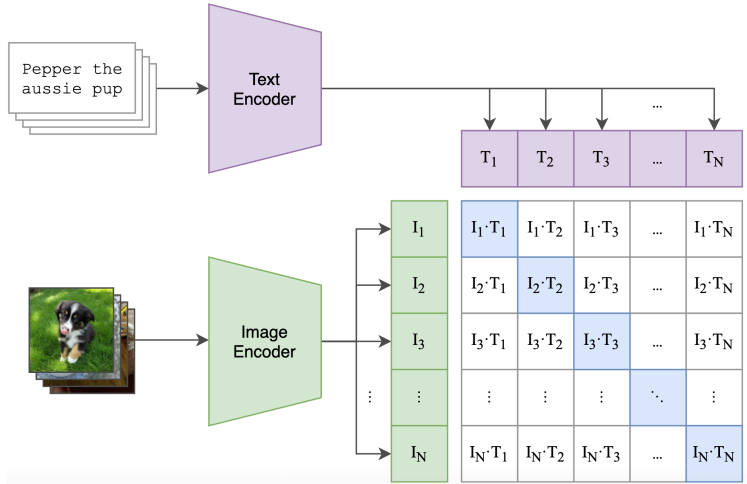
U-Net with Time Embedding



Generating Images from Texts

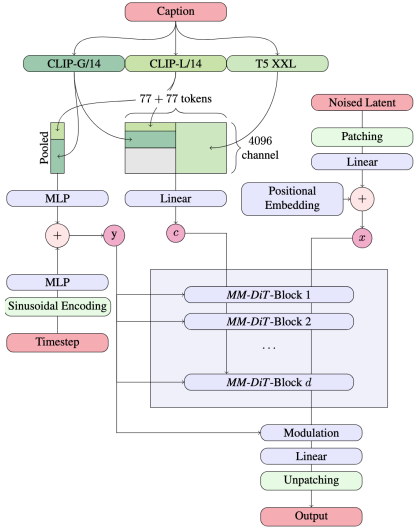
“An astronaut riding a horse”

Contrastive Language-Image Pre-training (CLIP)

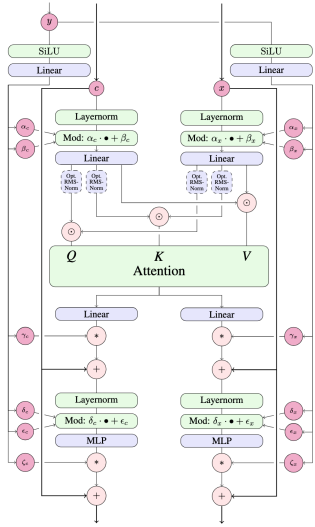


Learning Transferable Visual Models From Natural Language Supervision (Radford et al. 2021)

Stable Diffusion 3



(a) Overview of all components.

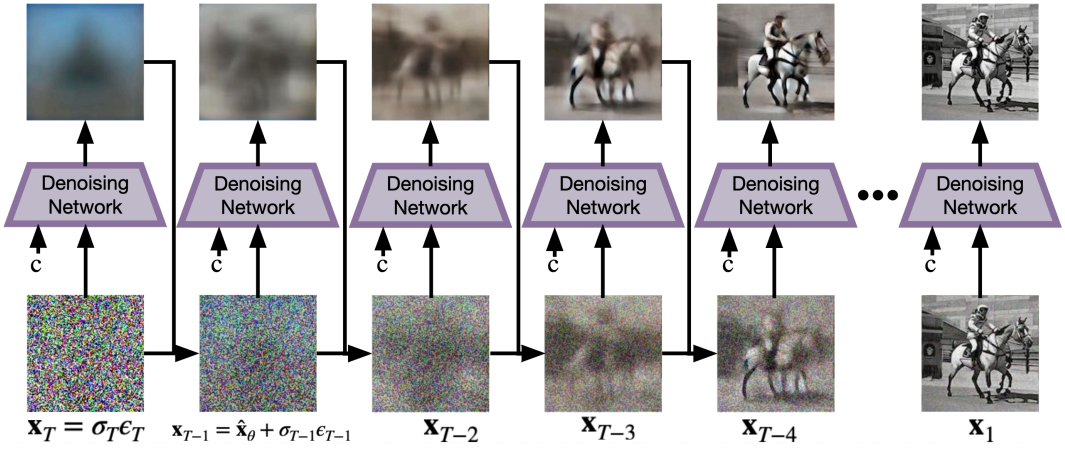


(b) One *MM-DiT* block

Scaling Rectified Flow Transformers for High-Resolution Image Synthesis (Esser et al., 2024)

Example

$c =$ “An astronaut riding a horse”



Try the Demo!

https://huggingface.co/spaces/black-forest-labs/FLUX.2-klein-9B

“A Corgi astronaut floating in a vast empty space”