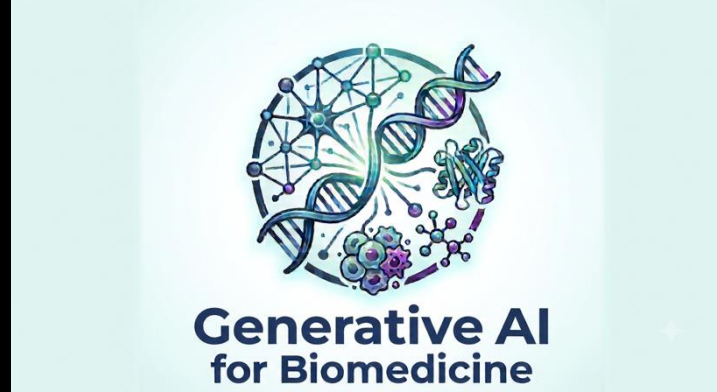




Diffusion Model

Lei Li

Carnegie Mellon University

Language Technologies Institute

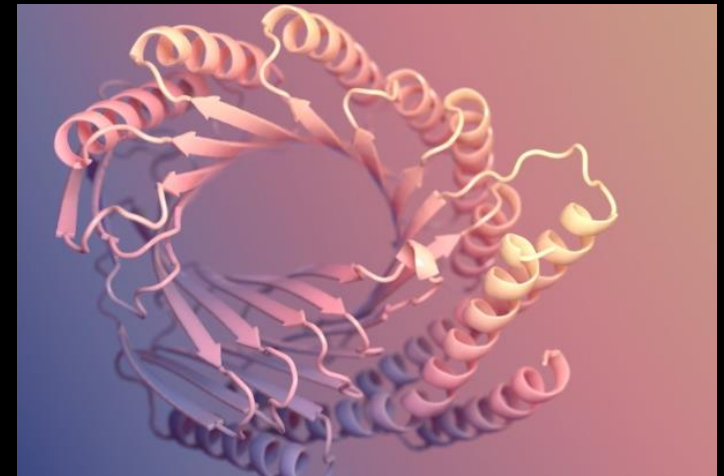
Image and Video Generation

generate an image of a
colorful deer riding on the
walking into the sky at cmu
campus



Diffusion Models are state-of-the-art models for (Continuous) Data Generation

- Time series data
- Audio/speech
- 3D Objects (coordinates)
 - Molecule structures



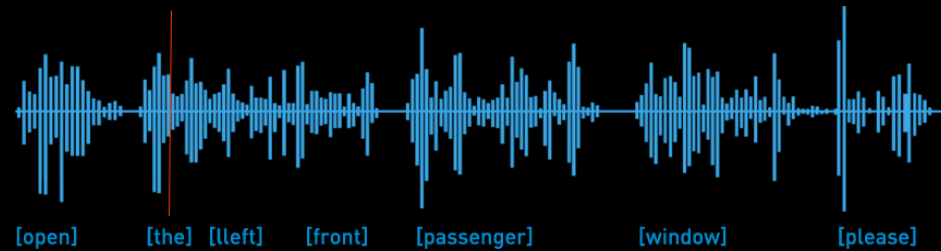
Data Representation

Image



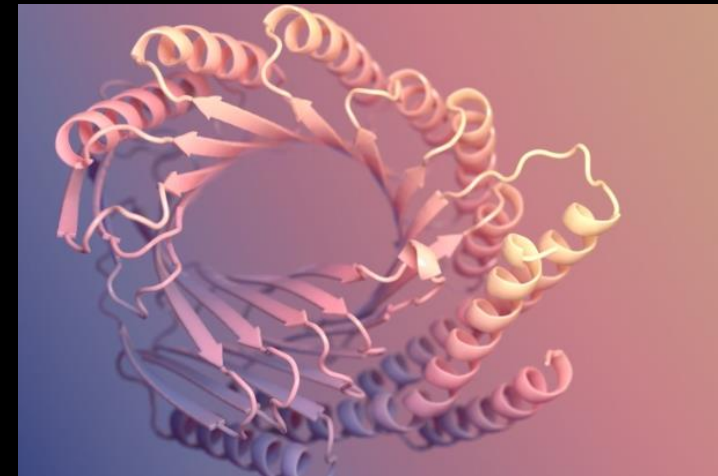
$$x \in \mathbb{R}^{w \times h}$$

Speech



$$x \in \mathbb{R}^t$$

Molecule Structure



$$x \in \mathbb{R}^{N \times 3}$$

Probabilistic Model for Data

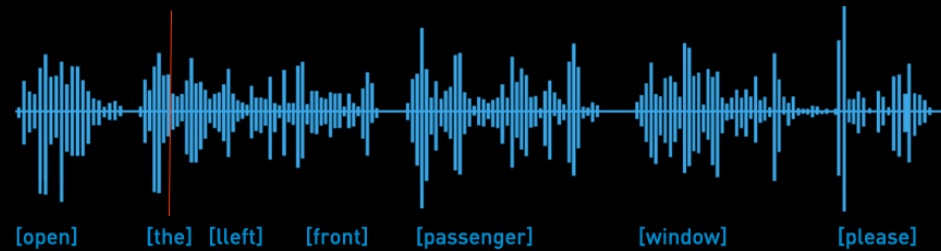
Find a probabilistic model $p(x)$ to fit the data and to generate sample

Image



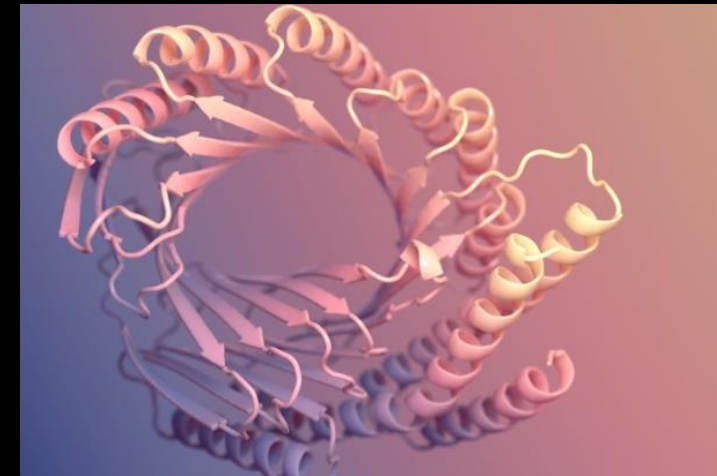
$$x \in \mathbb{R}^{w \times h}$$

Speech



$$x \in \mathbb{R}^t$$

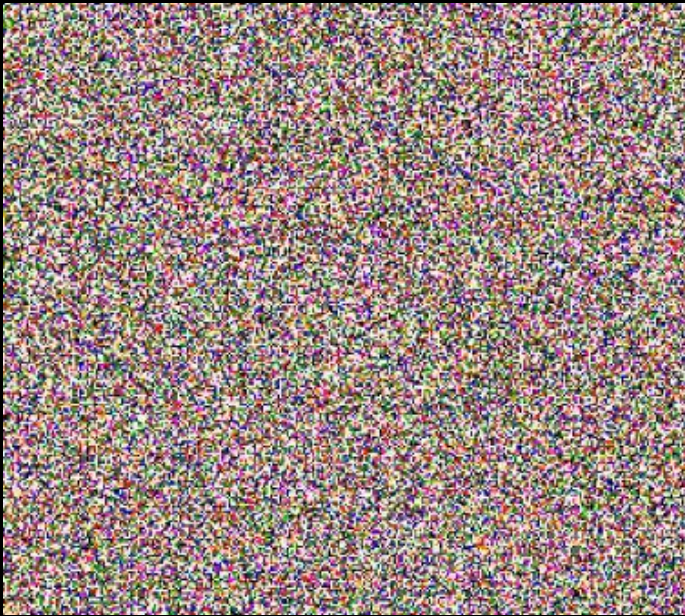
Molecule Structure



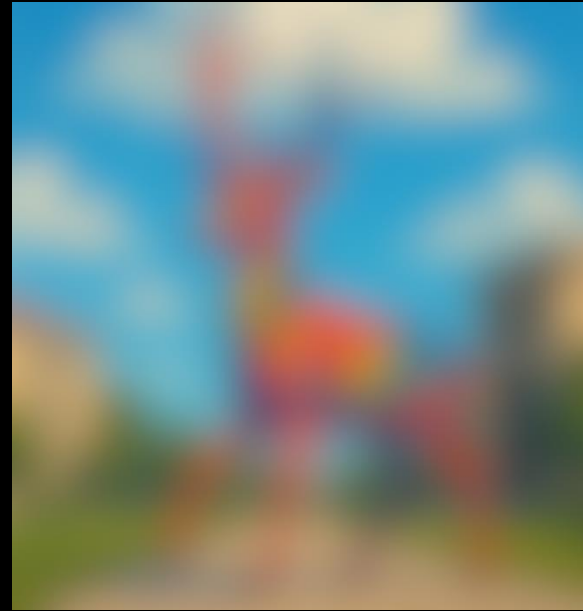
$$x \in \mathbb{R}^{N \times 3}$$

Probabilistic Generative Process

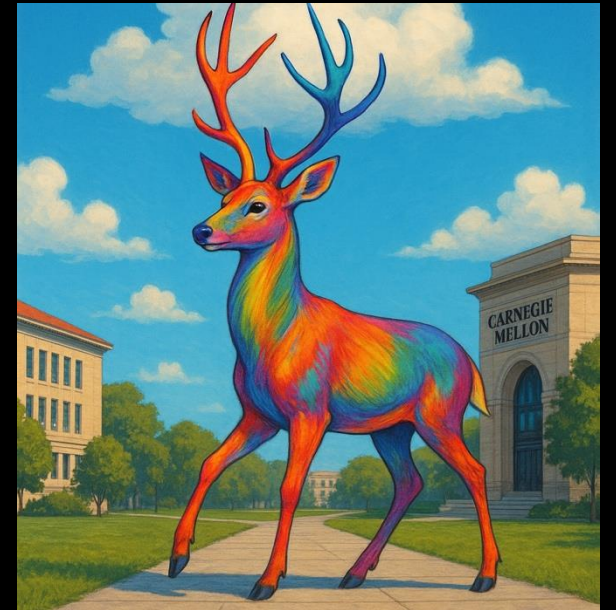
Start from initial
distribution $p_T(x_T)$
e.g. Gaussian $N(0, I)$



Generate slightly
improved data
 $x_{t-1} \sim p_{t-1}(x_{t-1} | x_t)$



final data
 $x_0 \sim p_0(x_0 | x_1)$



The generation process is a Markov chain

Learning the generative model
= learning the parameters for
each $p_{t-1}(x_{t-1}|x_t)$

It is difficult to directly construct a series of probability distributions

Diffusion Model

1. Forward noising process
 - Synthesize data for training by repeatedly adding noise each step
2. Reverse generation process
 - Start from complete noise, generate clean data
 - Also known as denoising

(Forward) Noising Process

- adding Gaussian noise at each time step t $x_t \sim q(x_t | x_{t-1}) = N(\sqrt{\alpha_t}x_{t-1}, \beta_t I)$, $\alpha_t = 1 - \beta_t$

equivalently $x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{\beta_t}\epsilon$, $\epsilon \sim N(0, I)$

$t=0$

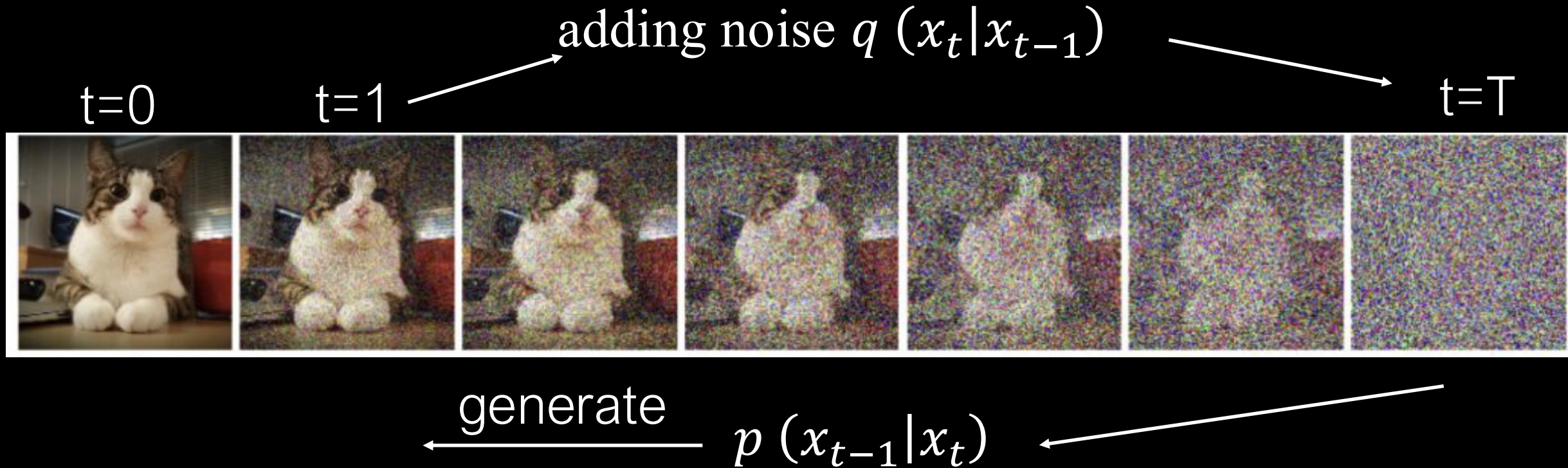
$t=1$

$t=T$



Consequence: with sufficient large T , the result will be Gaussian noise

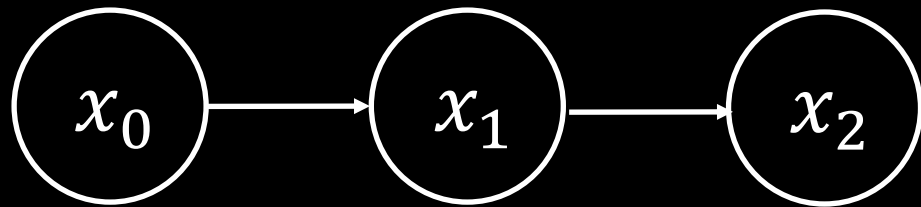
(Reverse) Generation/Denoising Process



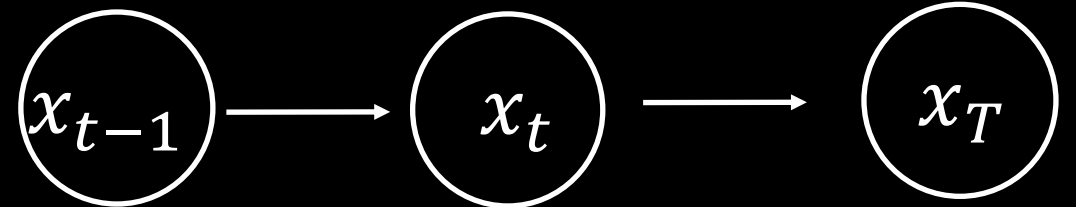
Learning problem: how to find parameters of p to match the sequence of noised images?

Graphical Model for Diffusion

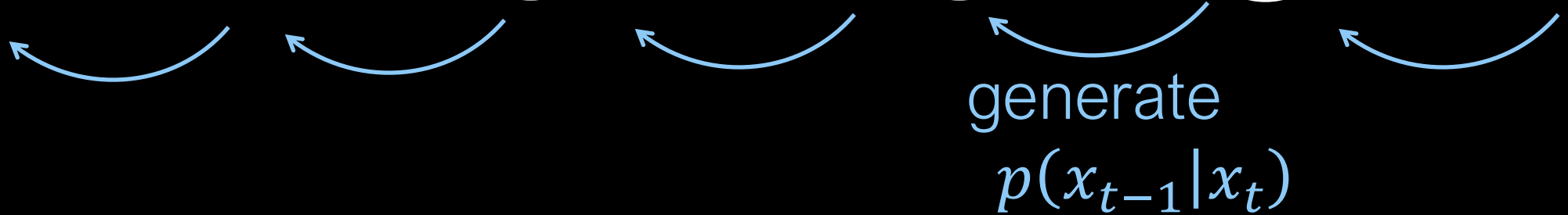
target data
 $t=0$



adding noise
 $q(x_t|x_{t-1})$



noise
 $t=T$



Learning problem: how to find parameters of p to match the sequence of noised images?

Denoising Diffusion Probabilistic Models (DDPM)

- Reverse Process (Generation), aka denoising

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

← predicted by NN

- Forward Process (Diffusion), aka noising

$$q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1})$$
$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I)$$

noising schedule: β_t in $[0, 1]$, and follows a predefined schedule

DDPM Training

For each data sample x_0 (e.g. image), random pick step t

1. sample a noise from standard Gaussian $N(0, I) \Rightarrow \epsilon$
2. noise data at step t : $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$, $\bar{\alpha}_t = \prod_{s=1}^t(1 - \beta_s)$
3. estimate noise using Neural network: $\hat{\epsilon}_\theta = \text{NN}_{\text{forward}}(x_t, t)$
4. compute MSE loss: $L = \|\epsilon - \hat{\epsilon}_\theta\|^2$
5. compute gradient via backward through NN
6. update model parameters using optimization alg (e.g. Adam)

DDPM training is essentially predicting the noise based on image

DDPM Inference

1. randomly sample from $\text{Gaussian}(0, I) \rightarrow x_T$
2. For step $t = T$ to 1
 - 1) estimate noise using Neural network: $\hat{\epsilon}_\theta = \text{NN}_{\text{forward}}(x_t, t)$
 - 2) generate noise from $N(0, I) \rightarrow \eta$
 - 3) estimate data mean: $\tilde{\mu}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \hat{\epsilon}_\theta \right)$
 - 4) update data: $x_{t-1} = \tilde{\mu}_{t-1} + \sqrt{\beta_t} \eta$

DDPM inference is essentially iteratively removing predicted noise

Intuition and Theory

- Training objective:

$$\begin{aligned} L &= -\log p_{\theta}(x_0) \\ &\leq -\mathbb{E} \left[\log \frac{p_{\theta}(x_{0:T})}{q(x_{1:T}|x_0)} \right] = L_{elbo} \end{aligned} \quad \text{ELBO loss}$$

$$\begin{aligned} L_{elbo} &= \sum_{t=1}^T \text{KL}(q(x_{t-1}|x_t, x_0) \parallel p_{\theta}(x_{t-1}|x_t)) + \text{KL}(q(x_T|x_0) \parallel p_{\theta}(x_T)) \\ &\quad - \mathbb{E} \log p_{\theta}(x_0|x_1) \end{aligned}$$

Hint: Jensen's inequality, posterior for linear Gaussian distribution

$$q(x_t|x_0) = N(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$$

$$\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$$

$$q(x_{t-1}|x_t, x_0) = N(x_{t-1}; \tilde{\mu}_t, \tilde{\beta}_t I)$$

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right)$$

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}$$

What about $p_{\theta}(x_{t-1}|x_t)$?

just set

$$p_{\theta}(x_{t-1}|x_t) = N(\mu_{\theta}(x_t, t), \sigma_t^2 I)$$

$$\text{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))$$

becomes

$$\text{KL}\left(N(x_{t-1}; \tilde{\mu}_t, \tilde{\beta}_t I) \parallel N(\mu_\theta(x_t, t), \sigma_t^2 I)\right)$$

let $\sigma_t^2 = \tilde{\beta}_t$, and let

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon}_\theta(x_t, t) \right)$$

$$\min \|\tilde{\mu}_t - \mu_\theta(x_t, t)\|^2$$

is equivalent to

$$\min \frac{\beta_t^2}{\alpha_t(1 - \bar{\alpha}_t)} \|\epsilon - \hat{\epsilon}_\theta(x_t, t)\|^2$$

Essentially, DDPM is using NN to predict noise given noised data

Summary

- Diffusion Model
 - is a probabilistic generative model by iteratively removing noise
 - construct a sequence of corrupted data by applying Gaussian noises
 - use a neural network to estimate the noise given the corrupted data (without knowing the original clean data)
 - once estimate the noise, use posterior estimation to remove noise
 - can use any suitable NN (e.g. UNet)
 - much better than VAE, but much slower