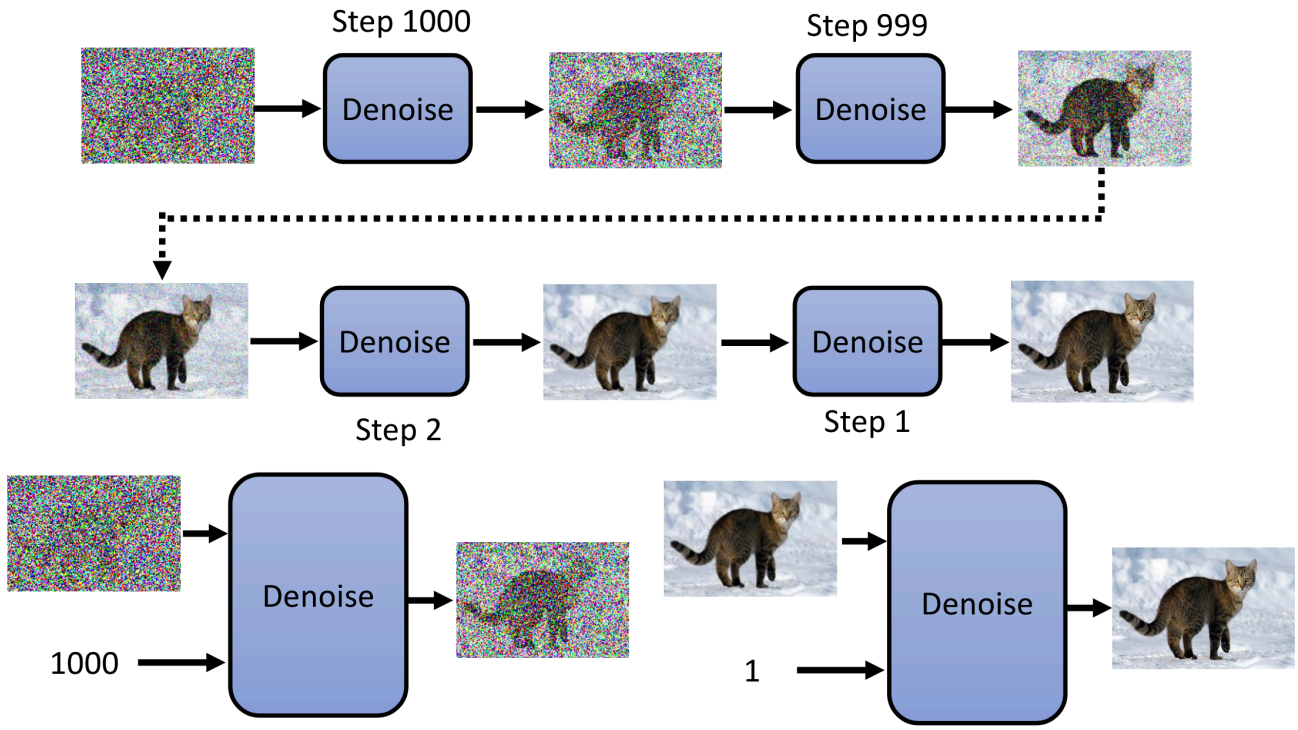
1.General Understanding

1.1 What is Diffusion?

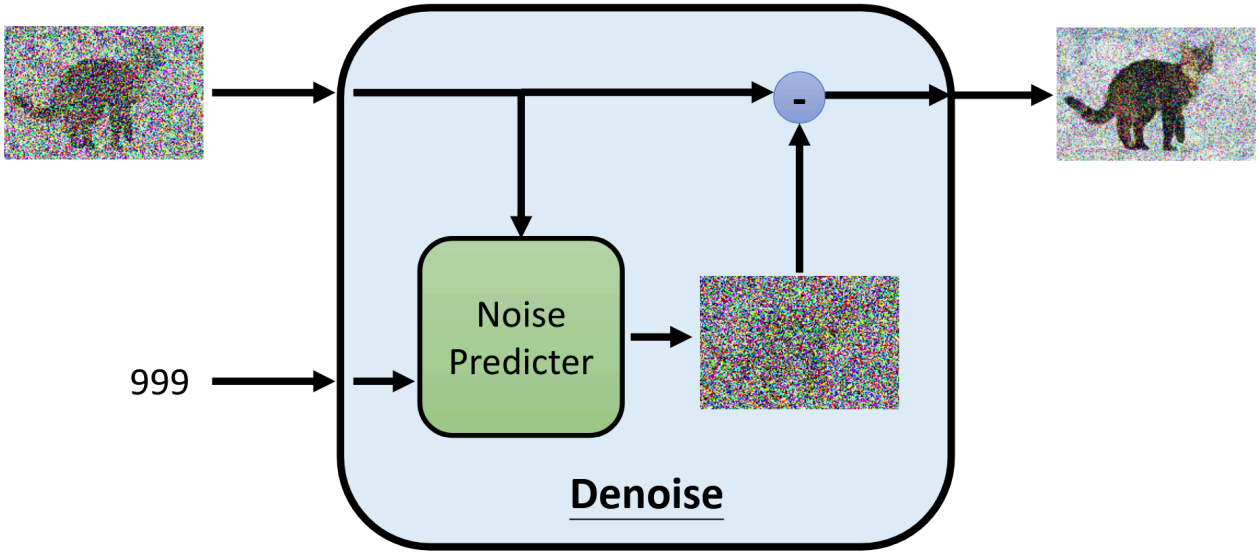
We hope to **sample a noise from a Gaussian distribution and take the noise with step as input to denoise a clear iamge**
the process shown here isn't the paper's method but it's easier to understand the core value of Diffusion:
"The sculpture is already complete within the marble block, before I start my work. It is already there, I just have to chisel away the superfluous material." - Michelangelo



1.2 What is Denoise and how to train it?

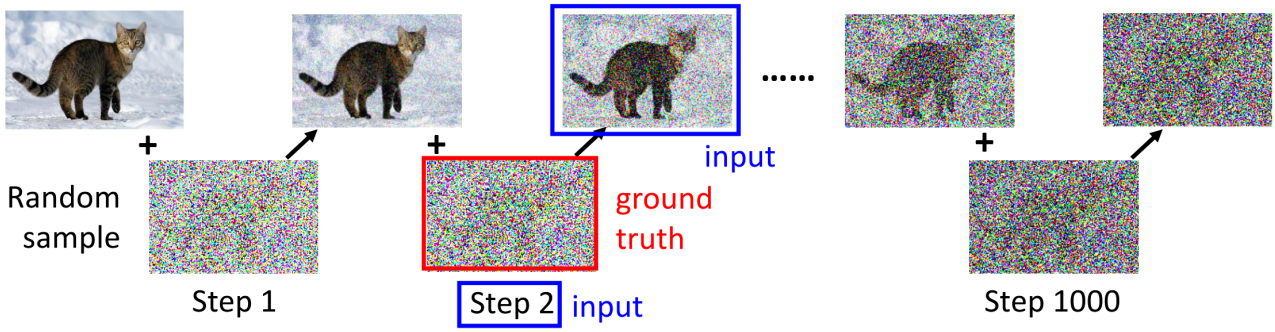
Instead of using an end-to-end model which can directly take noisy image with step as input and output the denoise image, Diffusion uses Noise Predicter to predict noise and substract the

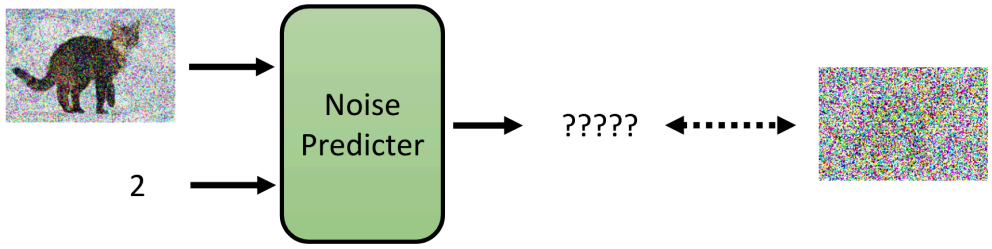
predicted noise from the origin image to give an output



If we want to train the Denoise part, we need

1. noisy image
2. noise added on the image
3. step
   So we use the origin and clear image as begin, manually add noise which sampled from a distribution. This process's name is **Forward Process**

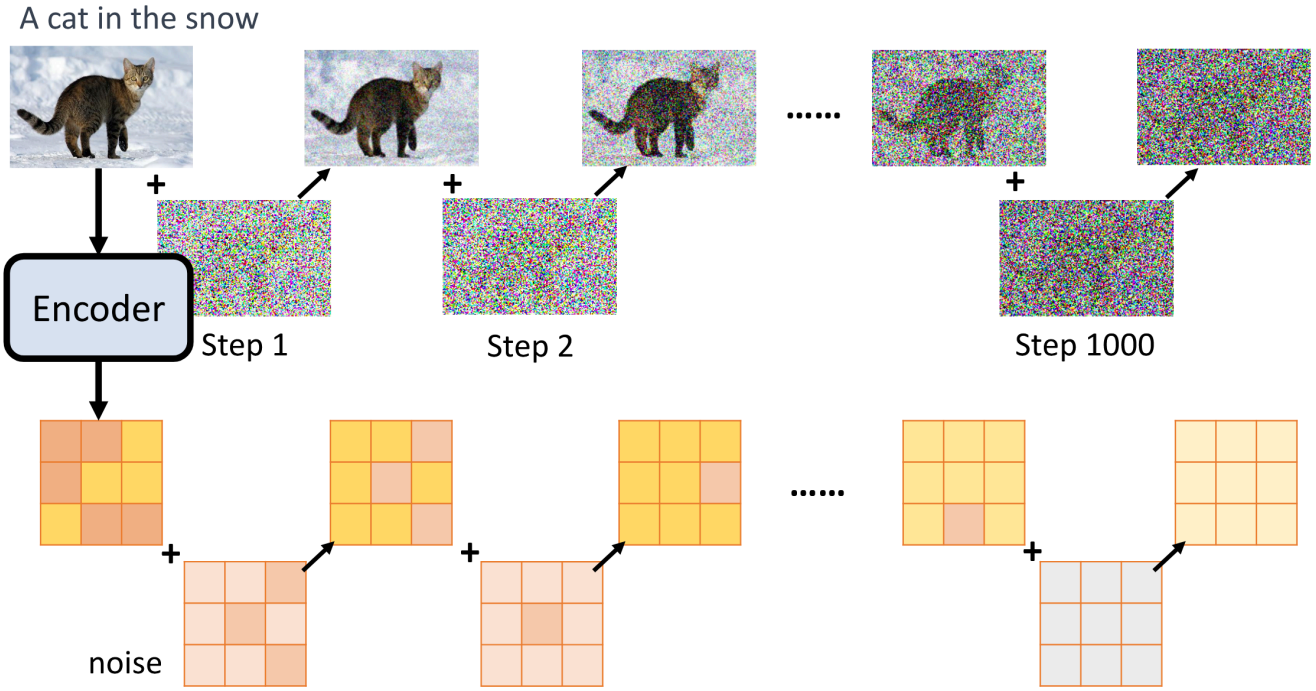




1.3 Diffusion with latent representation

Different from the image input to diffusion a new and clear image, we sometimes need text-to-image Diffusion Model, which will need encoder, decoder and latent representation.

the Diffusion of latent representation is similar to image Diffusion.
The only diiferent point here is :
the noisy image somehow we can distinguish what it is.
the latent vector can only be decoded by the decoder but not interpreted by human.



2. Algorithm & Math Principle

2.1 Diffusion Algorithm

This algorithm includes **training and sampling**

Algorithm 1 Training

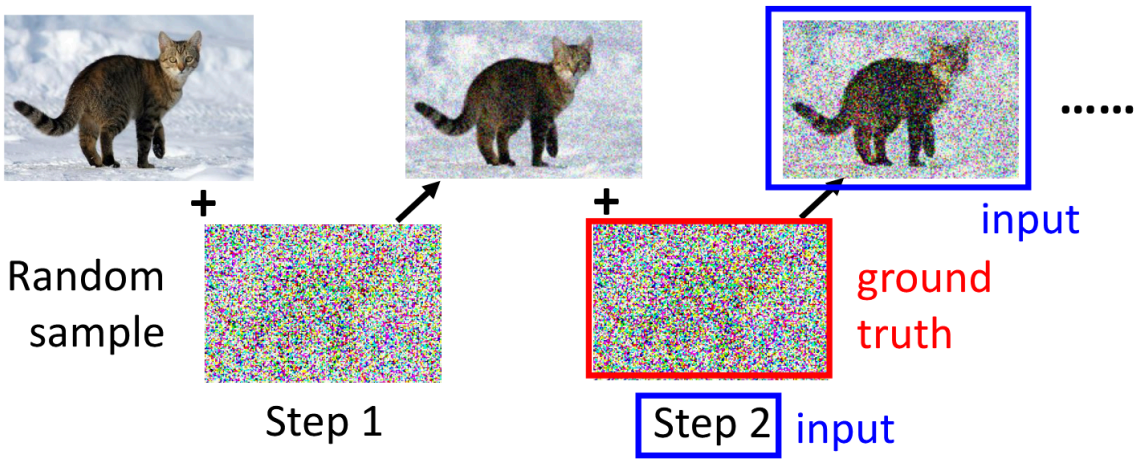
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

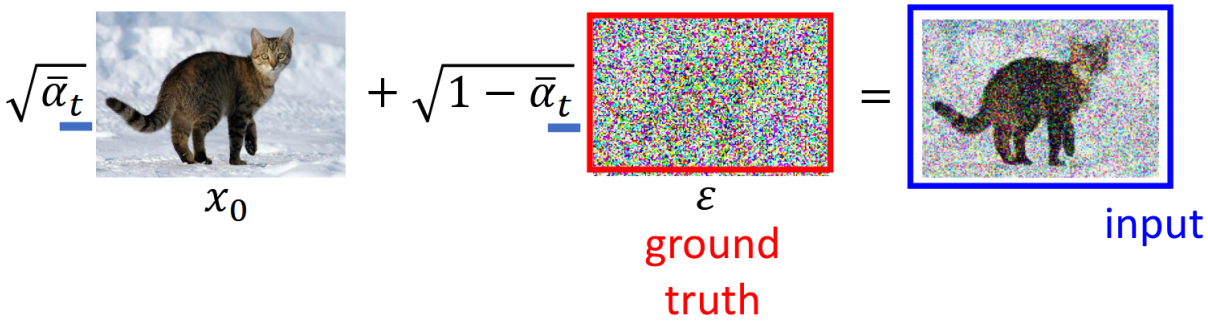
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

This algorithm is slightly different from the general understanding(the reason is in the math principle part). In the training part, we thought that we add noise step by step. But in reality, we use single-step weighted sum to add noise (the weight is $\bar{\alpha}_t$).
For example , we take x_0 as clear image and ϵ as noise.
if $\bar{\alpha}_t$ is smaller , then this image will be more noisy.
To represent the step t concept, we manually define $\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3, \cdots, \bar{\alpha}_t$, where $\bar{\alpha}_i$ decreases as

the index i increases.

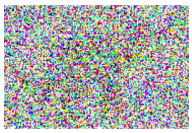




2.1.1 Training

Training

x_0: clean image



ε: noise

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ ⟵ sample clean image
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ⟵ sample a noise
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Noisy image

$\bar{\alpha}_1, \bar{\alpha}_2, \dots \bar{\alpha}_T$ → smaller

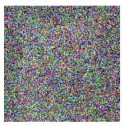
Target Noise

Noise predictor

As we discuss above , we train a Noise predictor to predict the noise , which is ϵ_θ .

2.1.2 Sampling

Inference



x_T

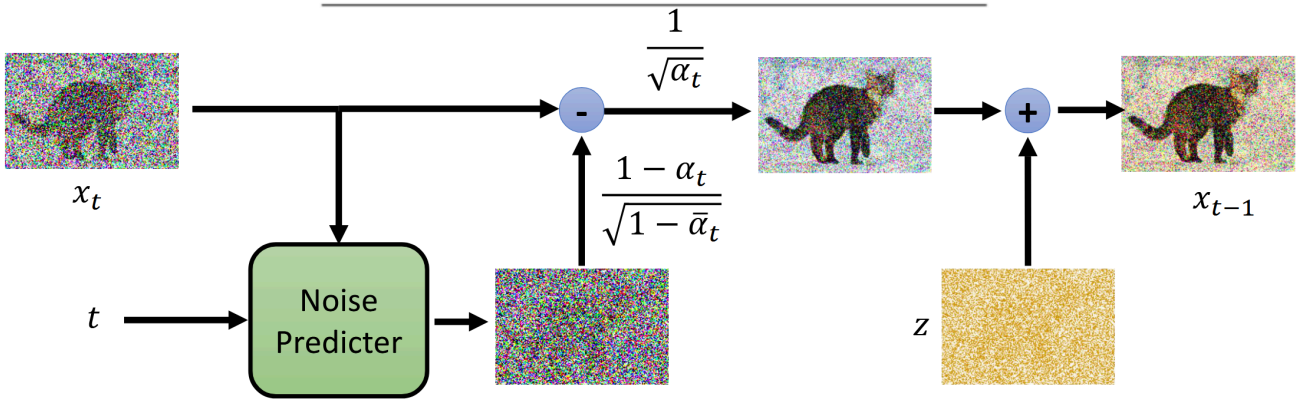
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

sample a noise?!

$\bar{\alpha}_1, \bar{\alpha}_2, \dots \bar{\alpha}_T$

$\alpha_1, \alpha_2, \dots \alpha_T$



There is a question : why we sample a new noise $\mathbf{z}$ here and add to the cleaner image $\mathbf{x}_{t-1}$? the answer is after math principle.

2.2 Math Principle

First , we want our model θ can produce images which are similar to reality distribution. In formal , we want our model to learn the true distribution $P_{data}(x)$ as $P_{\theta^*}(x)$. According to the maximum likelihood estimation, we should be able to compute $P_\theta(x)$, sample $\{x_1, x_2, \cdots, x_m\}$ from $P_{data}(x)$ and finally choose $\theta^\star$ by

$$\theta^\star = \arg\max_\theta \prod_{i=1}^{m} P_\theta(x_i)$$

The following content is using variational inference to optimize $P_\theta(x)$ (equal to $\log P_\theta(x)$)

2.2.1 VAE as a introduction

Just like in CS285/CS285笔记.md Variation Inference Chapter, we use a latent variable $\mathbf{z}$ here, to maximum the lower bound of $\log P_\theta(x)$

VAE: Lower bound of $logP(x)$

$$logP_\theta(x) = \int_z q(z|x) logP(x) dz$$ $q(z|x)$ can be any distribution

$$= \int_z q(z|x) log\left(\frac{P(z,x)}{P(z|x)}\right) dz = \int_z q(z|x) log\left(\frac{P(z,x)}{q(z|x)}\frac{q(z|x)}{P(z|x)}\right) dz$$

$$= \int_z q(z|x) log\left(\frac{P(z,x)}{q(z|x)}\right) dz + \underbrace{\int_z q(z|x) log\left(\frac{q(z|x)}{P(z|x)}\right) dz}_{KL(q(z|x)||P(z|x))} \geq 0$$

$$\geq \int_z q(z|x) log\left(\frac{P(z,x)}{q(z|x)}\right) dz = \mathrm{E}_{q(z|x)}[log\left(\frac{P(x,z)}{q(z|x)}\right)]$$ *lower bound*

Encoder

2.2.2 Diffusion‘s Math Principle

To use maximum likelihood estimation , we need to comput $P_\theta(x)$ or use variation inference just like VAE to maximum the lower bound of $\log P_\theta(x)$.
So, what the $P_\theta(x_0)$ itself is ?

$$P_\theta(x_0) = \int_{x_1:\cdots:x_T} P(x_T) P_\theta(x_{T-1}|x_T) \dots P_\theta(x_{t-1}|x_t) \dots P_\theta(x_0|x_1)\, dx_1 : \cdots : dx_T$$

similar to VAE, we can also lower bound of $\log P(x)$ in DDPM:

DDPM: Lower bound of $logP(x)$

VAE Maximize $logP_\theta(x)$ ⟶ Maximize $\mathrm{E}_{q(z|x)}[log\left(\frac{P(x,z)}{q(z|x)}\right)]$

Encoder

DDPM Maximize $logP_\theta(x_0)$ ⟶ Maximize $\mathrm{E}_{q(x_1:x_T|x_0)}[log\left(\frac{P(x_0:x_T)}{q(x_1:x_T|x_0)}\right)]$

Forward Process
(Diffusion Process)

$$q(x_1:x_T|x_0) = q(x_1|x_0)q(x_2|x_1) \dots q(x_T|x_{T-1})$$

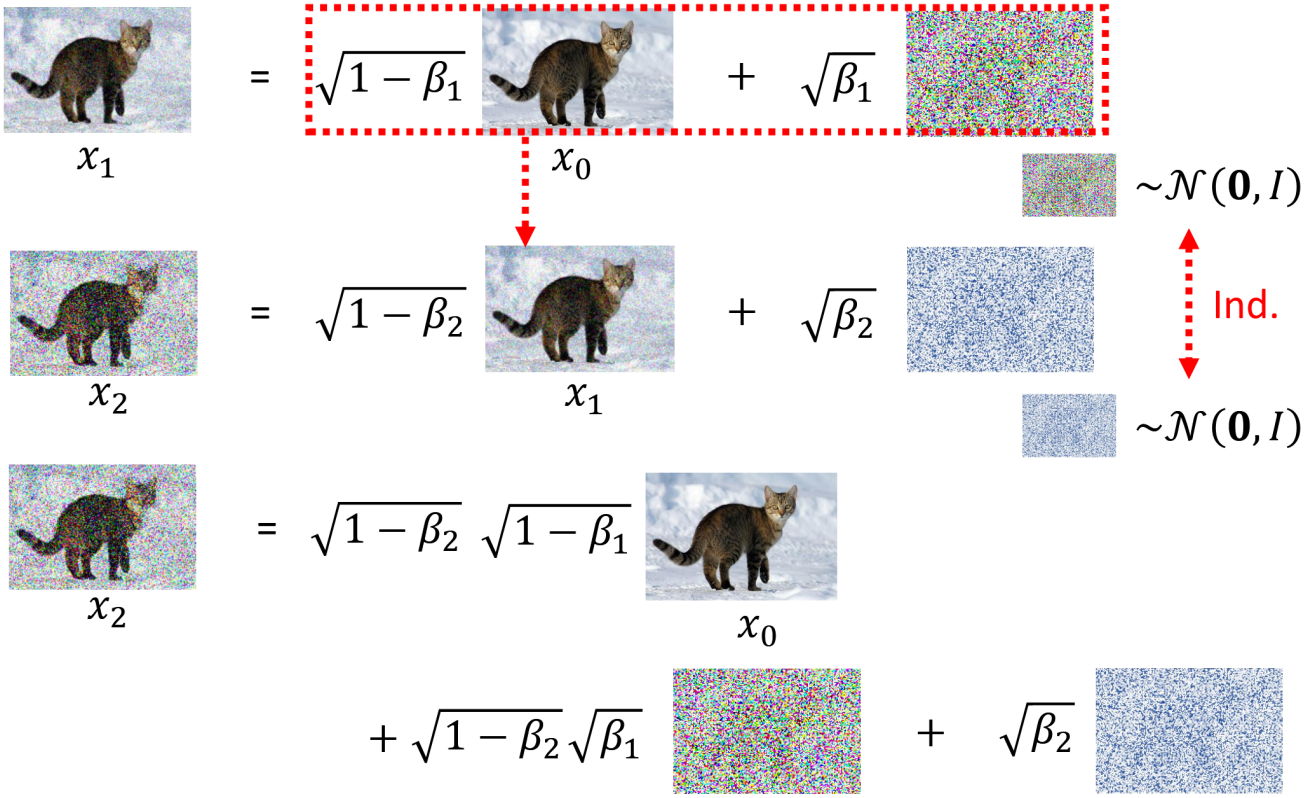
the forward process $q(x_1 : x_T|x_0)$ according to the markov property, can be written as

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

then our goal is to compute $q(x_t|x_{t-1})$:
First, we are given a series of hyperparameters β_1 to β_T.
the connection between x_t and x_{t-1} is

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}noise_t \quad noise_t \sim \mathcal{N}(0, I)$$

now we can use this iterated equation to compute $q(x_t|x_0)$, for example:



according to the Gaussian DIstribution's property, we can combine $\sqrt{1-\beta_2}\sqrt{\beta_1}noise_1 + \sqrt{\beta_2}noise_2$ as a union Gaussian Distribution

$$mean_{union} = mean_1 + mean_2, var_{union} = var_1 + var_2$$

So ,

$$\sqrt{1-\beta_2}\sqrt{\beta_1}noise_1 + \sqrt{\beta_2}noise_2 = \sqrt{1-(1-\beta_2)(1-\beta_1)}noise_{\text{union}} \quad noise_{\text{union}} \sim \mathcal{N}(0, I)$$

keep doing this iteration , we have:

$$x_t = \sqrt{(1-\beta_1)(1-\beta_2)\cdots(1-\beta_t)}x_0 + \sqrt{1-(1-\beta_1)(1-\beta_2)\cdots(1-\beta_t)}noise \quad noise \sim \mathcal{N}(0, I)$$

let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \alpha_1\alpha_2\cdots\alpha_t$, we have:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} noise \qquad noise \sim \mathcal{N}(0, I)$$

where are we?

our goal is

$$\text{Maximize } \mathbb{E}_{q(x_1:x_T|x_0)} \left[\log \left(\frac{p(x_0 : x_T)}{q(x_1 : x_T \mid x_0)} \right) \right]$$

and according to the calculation above, we have $q(x_1 : x_T|x_0)$, so we can optimize the goal:

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right] \tag{47}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})} \right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1) \prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0) \prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})} \right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1) \prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0) \prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)} \right] \tag{50}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)} \right] \tag{51}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}} \right] \tag{52}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}} \right] \tag{53}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{\cancel{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}} + \log \frac{\cancel{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)} \right] \tag{54}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} + \sum_{t=2}^{T} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)} \right] \tag{55}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1) \right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} \right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)} \right] \tag{56}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1) \right] + \mathbb{E}_{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} \right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)} \right] \tag{57}$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1) \right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \right]}_{\text{denoising matching term}} \tag{58}$$

$$\text{Maximize } \mathrm{E}_{q(x_1:x_T|x_0)}[log \left(\frac{P(x_0 : x_T)}{q(x_1 : x_T|x_0)} \right)]$$

Understanding Diffusion Models: A Unified Perspective

https://arxiv.org/pdf/2208.11970.pdf

finally, our goal is maximize :

$$\mathbb{E}_{q(x_1|x_0)}[\log p(x_0 \mid x_1)] - \mathrm{KL}\big(q(x_T \mid x_0) \,\big\|\, p(x_T)\big) - \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)} \big[\mathrm{KL}\big(q(x_{t-1} \mid x_t, x_0) \,\big\|\, p(x_{t-1} \mid x_t)\big)\big]$$

But! the second term $\mathrm{KL}\big(q(x_T \mid x_0) \,\big\|\, p(x_T)\big)$ doesn't depend on the network θ ,so we just optimize the first term and the third term. These two term are similar and we use optimizing $\sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)} \big[\mathrm{KL}\big(q(x_{t-1} \mid x_t, x_0) \,\big\|\, p(x_{t-1} \mid x_t)\big)\big]$ for example.

our goal: calculate $\sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)} \big[\mathrm{KL}\big(q(x_{t-1} \mid x_t, x_0) \,\big\|\, p(x_{t-1} \mid x_t)\big)\big]$
we are calculating KL Divergence of two distribution:

1. $q(x_{t-1} \mid x_t, x_0)$, which is independent of θ .
2. $p(x_{t-1} \mid x_t)$, which is related to θ .
   we have $q(x_t|x_0), q(x_{t-1}|x_0), q(x_t|x_{t-1})$, according to the Bayes' theorem, the posterior distribution $q(x_{t-1} \mid x_t, x_0)$ can be written as

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_{t-1}, x_t, x_0)}{q(x_t, x_0)} = \frac{q(x_t|x_{t-1}) q(x_{t-1}|x_0) q(x_0)}{q(x_t|x_0) q(x_0)} = \frac{q(x_t|x_{t-1}) q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

the ultimate terms $q(x_t|x_{t-1}), q(x_{t-1}|x_0), q(x_t|x_0)$ are all known Gaussian distribution. after a complex math derivation, we can prove that $q(x_{t-1}|x_t, x_0)$ is also a Gaussian distribution.

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1};\ \mu_q(x_t, x_0),\ \Sigma_q(t))$$

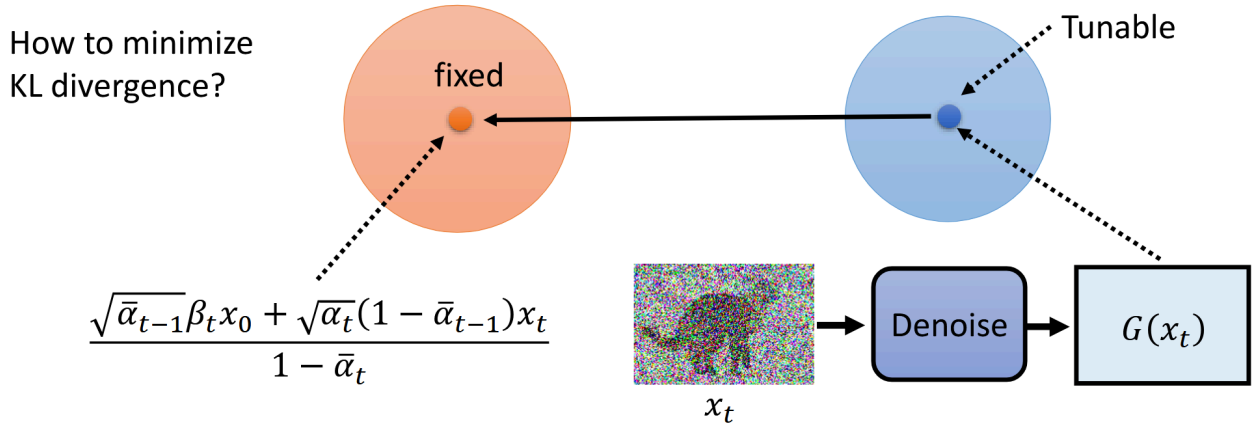
$$\mu_q(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$$

$$\Sigma_q(t) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathbf{I}$$

Recall that the KL Divergence between two Gaussian distributions is :

$$D_{\mathrm{KL}}\big(\mathcal{N}(x;\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)\,\|\,\mathcal{N}(y;\boldsymbol{\mu}_y,\boldsymbol{\Sigma}_y)\big) = \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_y|}{|\boldsymbol{\Sigma}_x|} - d + \mathrm{tr}\big(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_x\big) + (\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)^T\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)\right]$$

this KL DIvergence is compute both by means and variances from two distribution.
this paper assumes that the Denoise output only a Gaussian mean and the variance is fixed manually, future paper named "Variance-Preserving Diffusion Models" https://arxiv.org/abs/2306.08527 use model learn both mean and var.
In DDPM, the var is fixed, which means we don't compute KL Divergence by both mean and var but only use mean is fine. In this figure, the center point represents the mean, and the size of the light-colored circle represents the variance. if the var of θ is fixed, we just need to minimize mean of p and θ .



and this equal to compute the MSE loss of mean of q and θ just as below:

$$\mathbb{E}_{q(x_t|x_0)}\left[\mathrm{KL}\left(q(x_{t-1}|x_t,x_0)\,\|\,p_\theta(x_{t-1}|x_t)\right)\right] \propto \mathbb{E}_{q(x_t|x_0)}\left[\|\mu_q(x_t,x_0) - \mu_\theta(x_t,t)\|^2\right]$$

we already have the posterior distribution's mean: $\mu_q(x_t, x_0)$:

$$\mu_q(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t x_0 + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t}{1-\bar{\alpha}_t}$$

but directly using a network to predict this mean $\mu_q(x_t, x_0)$ is complex, we need to simplify this equation by using:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$

therefore,

$$x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\, \epsilon}{\sqrt{\bar{\alpha}_t}}$$

Substitute this expression back into the original equation,

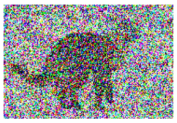
$$\mu_q(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right)$$

in this equation, we only need network to predict the noise ϵ because the other coefficients and x_t are constant. Let's use ϵ_θ as the predicted noise.
So, the ϵ Gaussian distribution's mean is:

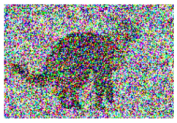
$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

This is consistent with the sampling algorithm.

x_0



x_t



Sample x_t

t

Denoise → ? ⇢ $\frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon \right)$

實際需要 network predict 的部分

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \varepsilon$$

$$x_t - \sqrt{1 - \bar{\alpha}_t} \varepsilon = \sqrt{\bar{\alpha}_t} x_0$$

$$\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \varepsilon}{\sqrt{\bar{\alpha}_t}} = x_0$$

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

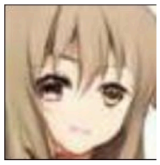
Now we discuss why we need the added noise $\mathbf{z}$ (link to 2.1.2)

1. if we doing experiment such as set $\sigma_t = 0$ which means no added noise, we got poor performance.

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

σ_t as paper



$\sigma_t = 0$

感謝伏宇寬助教提供結果

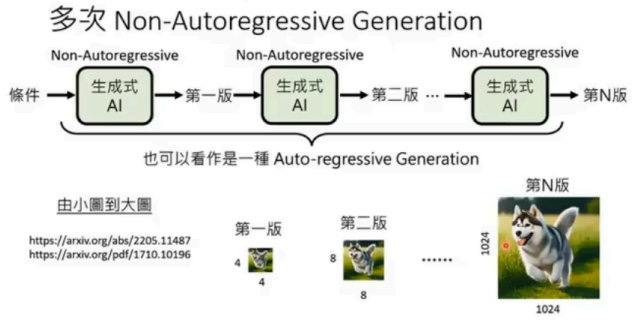
2. just like we choose the next token, we usually don't choose the highest probability one, but sample from a next token distribution.
   According to https://www.bilibili.com/video/BV1MtXHYUE6M/?spm_id_from=333.337.search-card.all.click&vd_source=d7cc9749aae49952734872a9708a74f2 , a more detailed explaination.

   为什么不预测原图像，而是预测噪声?

   因为预测噪声的难度远远小于预测原始图像，预测图像会导致效果变差

   为什么在生成的过程中，要加入噪声

   本质上diffusion model是一种结合非自回归生成和自回归生成，加入噪声才能保证比较好的生成效果



Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

3.Reference

Math slide:DDPM (v7).pdf
slide and video from:https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php