Image Generative Model

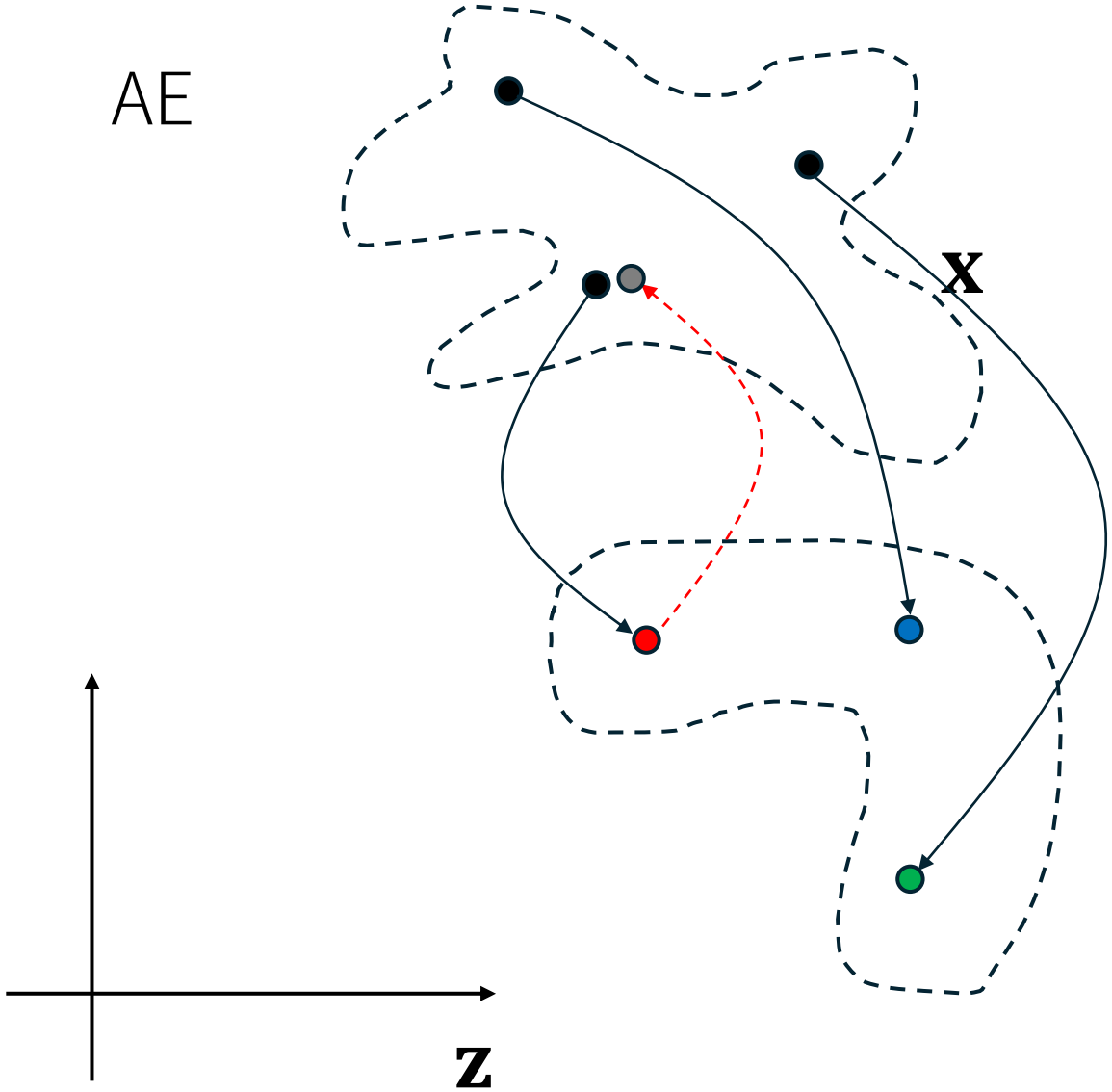
Auto Encoder, VAE and Diffusion Model

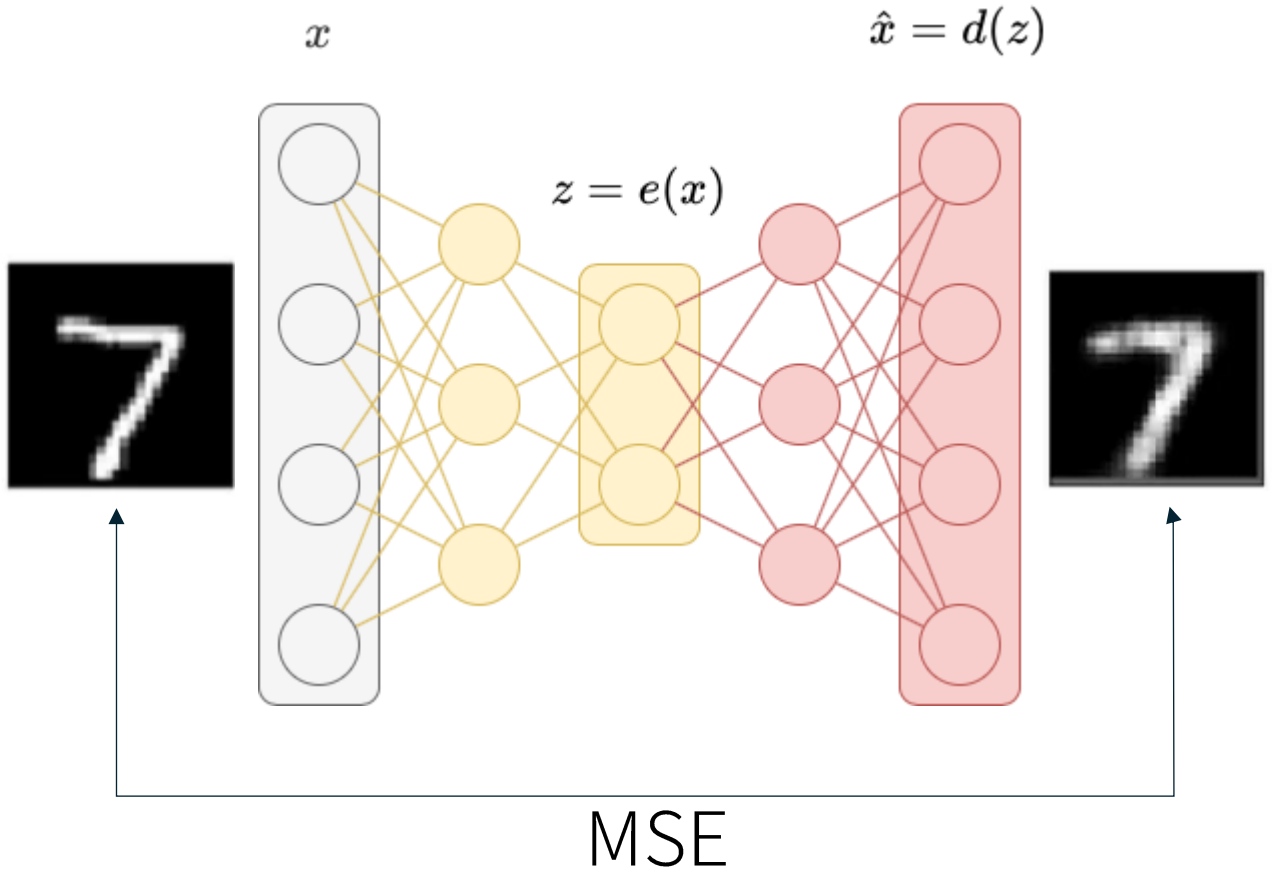
조준우
metamath@gmail.com

Auto Encoder

Auto Encoder

AE





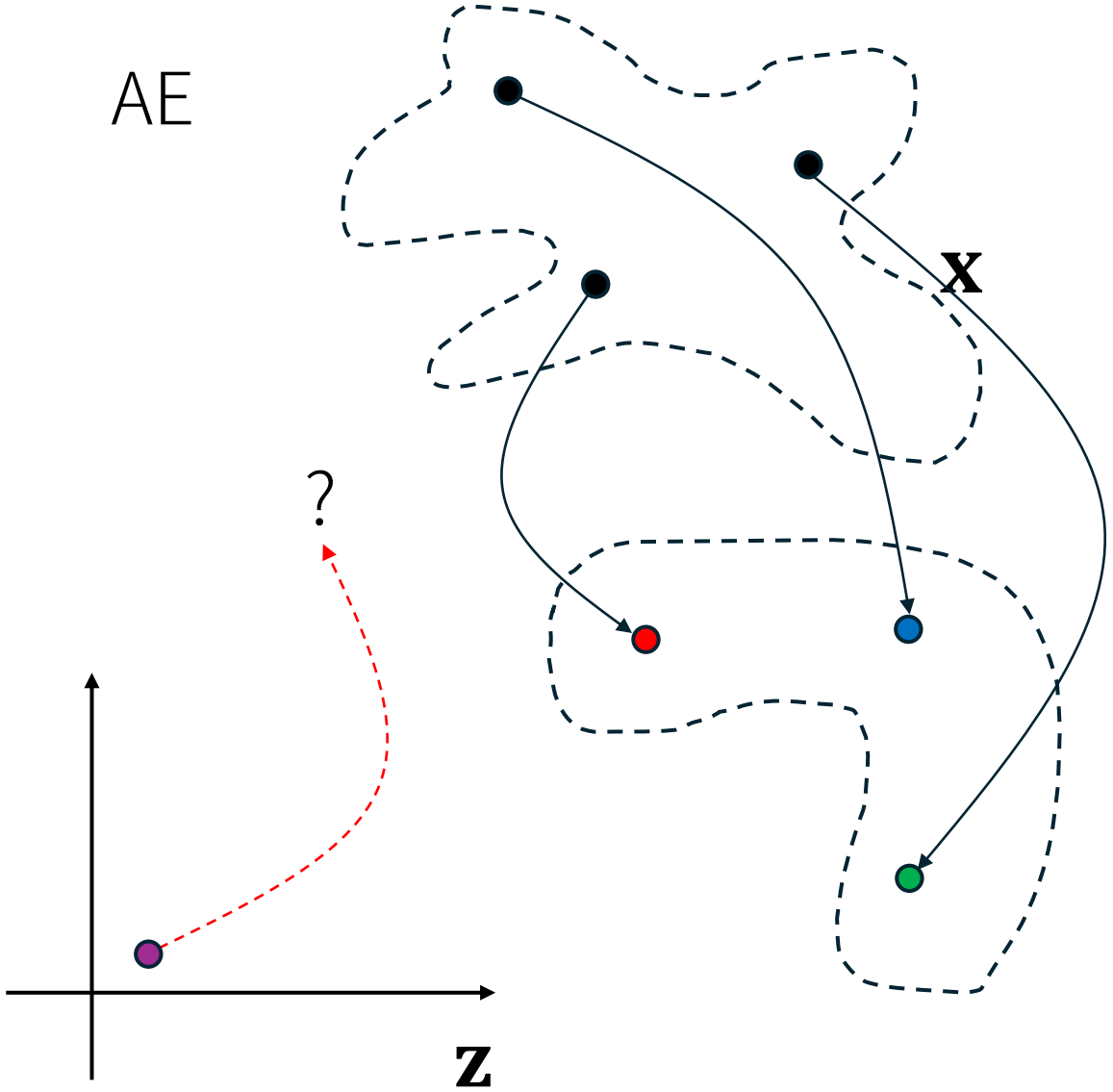
https://avandekleut.github.io/vae/

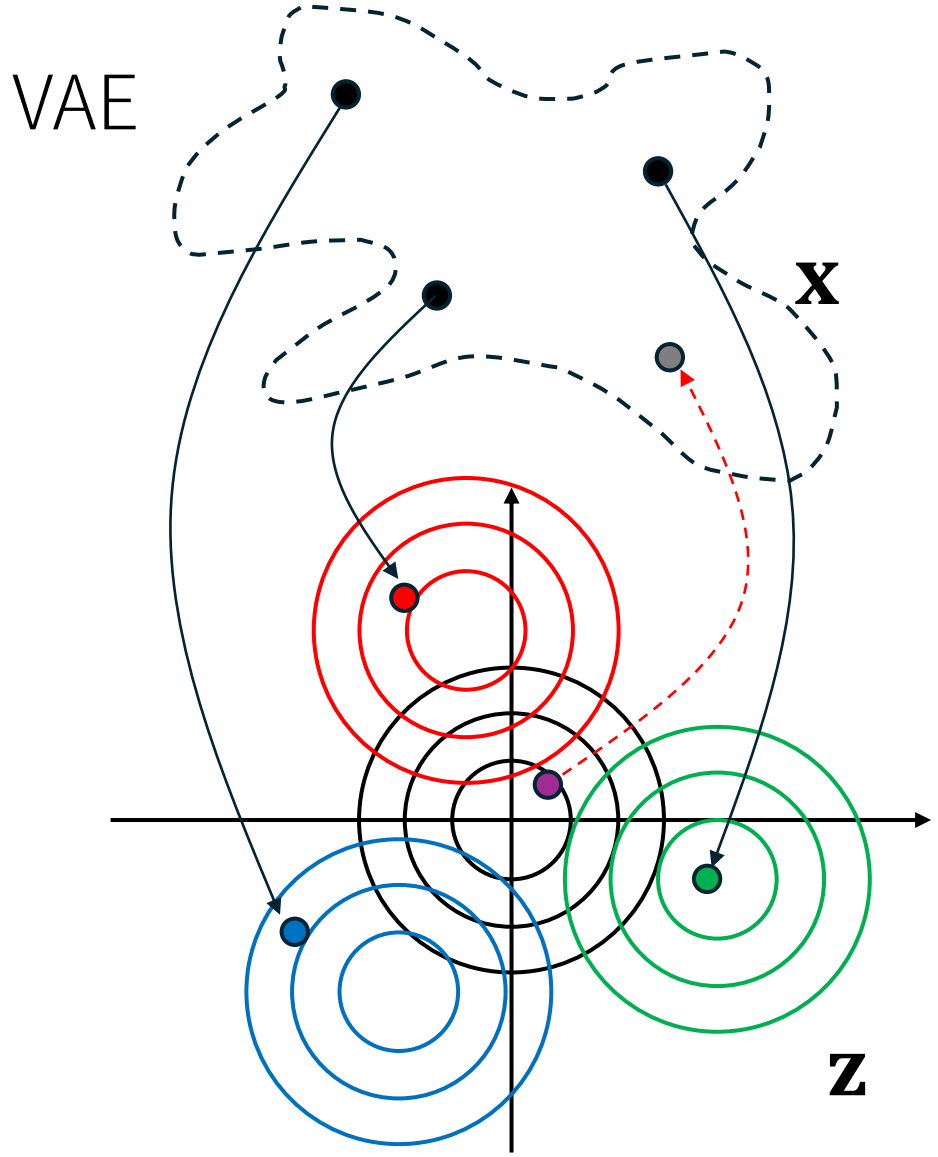
Variational Auto Encoder

Auto-Encoding Variational Bayes (2013)

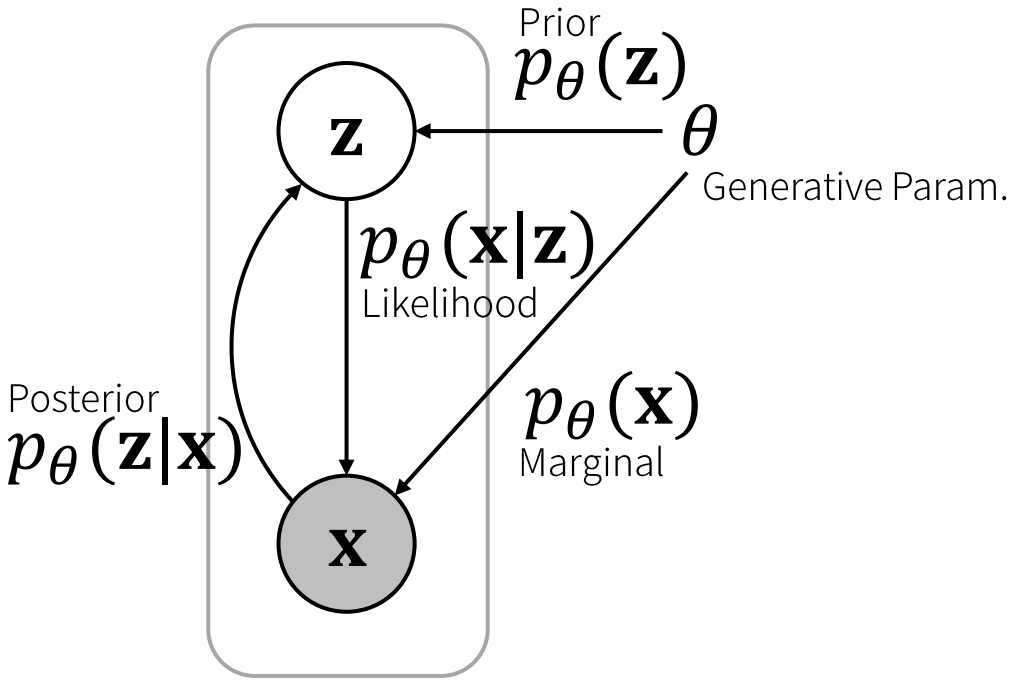
Diederik P. Kingma and Max Welling, University of Amsterdam

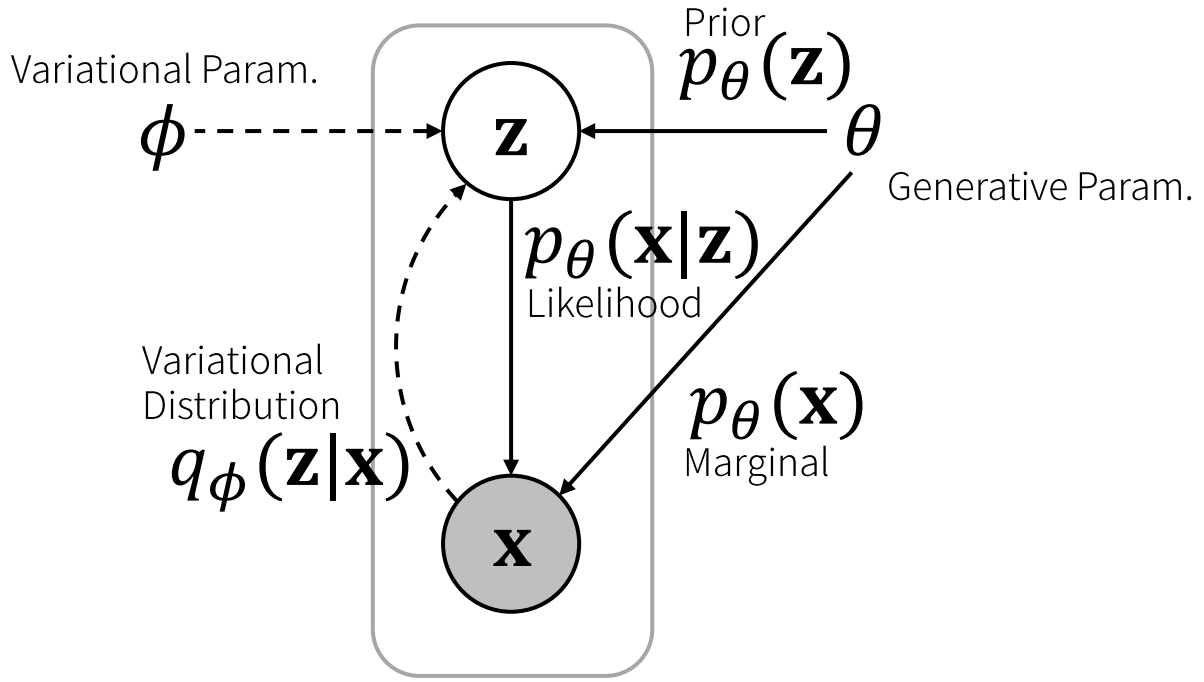
The Concept of VAE



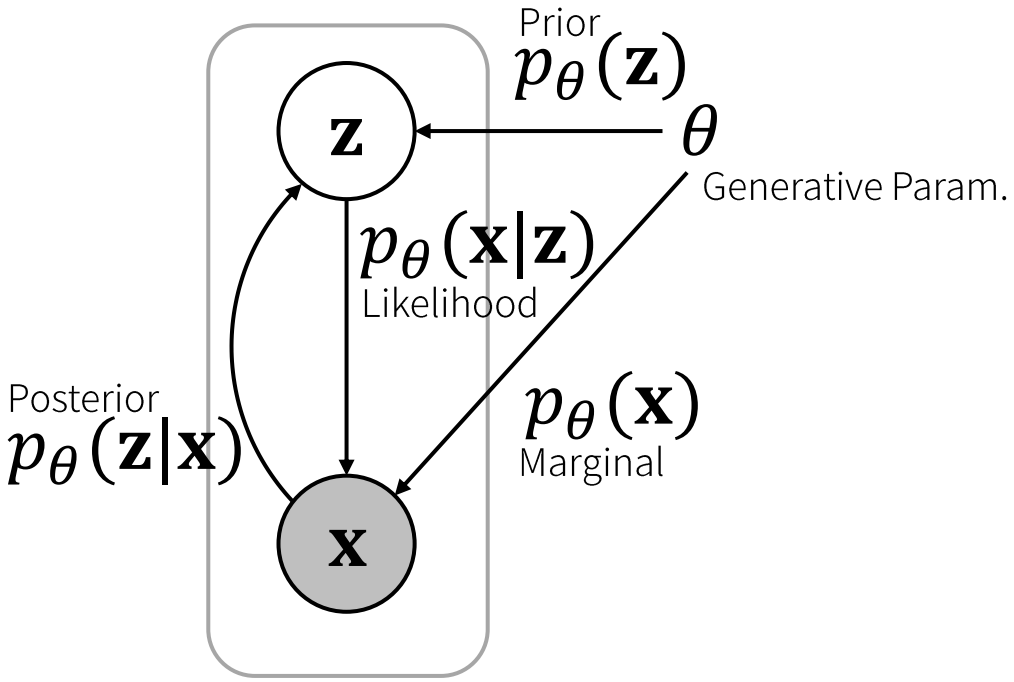


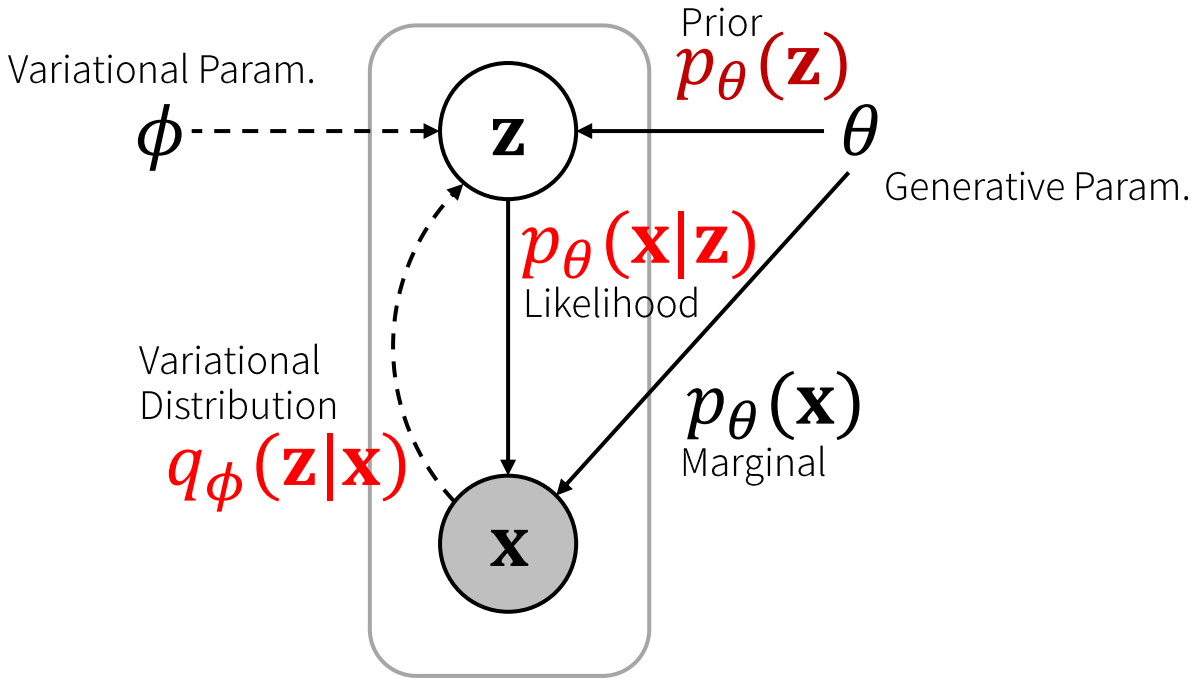
The Graphical Model in VAE



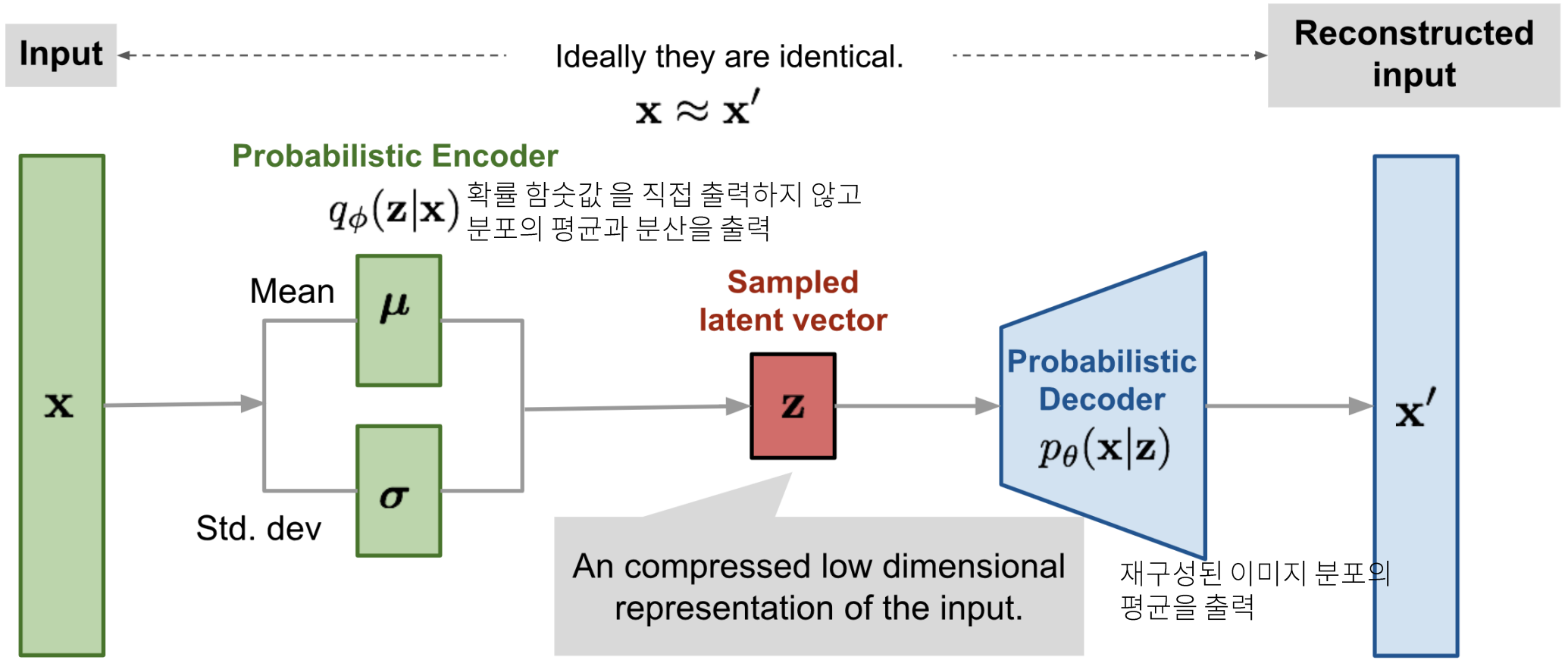


The Graphical Model in VAE





VAE Model



https://lilianweng.github.io/posts/2018-08-12-vae/

Variational Low Bound

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right]$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]$$

$$= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})\ (\text{ELBO})} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))}$$

$$E_{q_\phi(z|x)}[\log p_\theta(x)] = \int q_\phi(z|x)\log p_\theta(x)dz$$

$$= \log p_\theta(x)\int q_\phi(z|x)\,dz$$

$$= \log p_\theta(x)$$

Diederik P. Kingma and Max Welling (2019), “An Introduction to Variational Autoencoders”

Variational Low Bound

데이터의 로그 가능도 ⇨ 최대가 되길…

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})} = \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}} = \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}$$

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})\ (\text{ELBO})} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))\ \geq 0}
\end{aligned}$$

Diederik P. Kingma and Max Welling (2019), “An Introduction to Variational Autoencoders”

Variational Low Bound

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})\ (\text{ELBO})} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))}$$

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right] \tag{2.10}$$

$$= \log p_{\boldsymbol{\theta}}(\mathbf{x}) - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})) \tag{2.11}$$

By looking at equation 2.11, it can be understood that maximization of the ELBO $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})$ w.r.t. the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, will concurrently optimize the two things we care about:

1. It will approximately maximize the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x})$. This means that our generative model will become better.

2. It will minimize the KL divergence of the approximation $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ from the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, so $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ becomes better.

Diederik P. Kingma and Max Welling (2019), “An Introduction to Variational Autoencoders”

Variational Loss for VAE

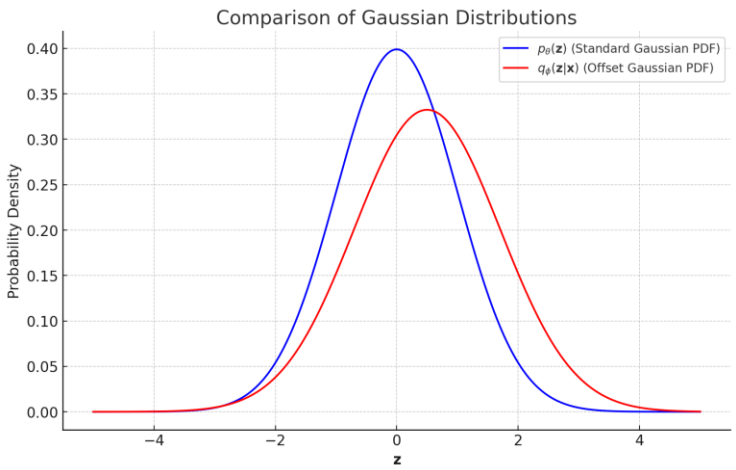
In Kingma and Welling (2014), following eq.(2) is the same equation as eq. 2.10

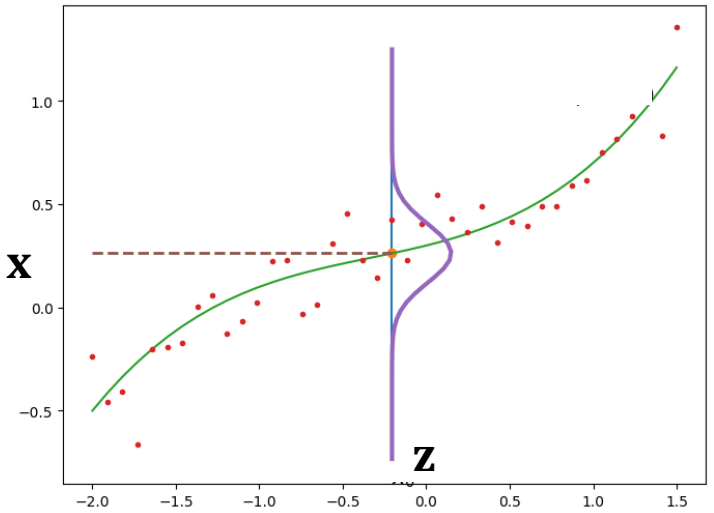
$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \right] \qquad (2)$$

좀 복잡한 계산 후.....

Regularization Regression Term(MSE)

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}) || p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) \right] \qquad (3)$$



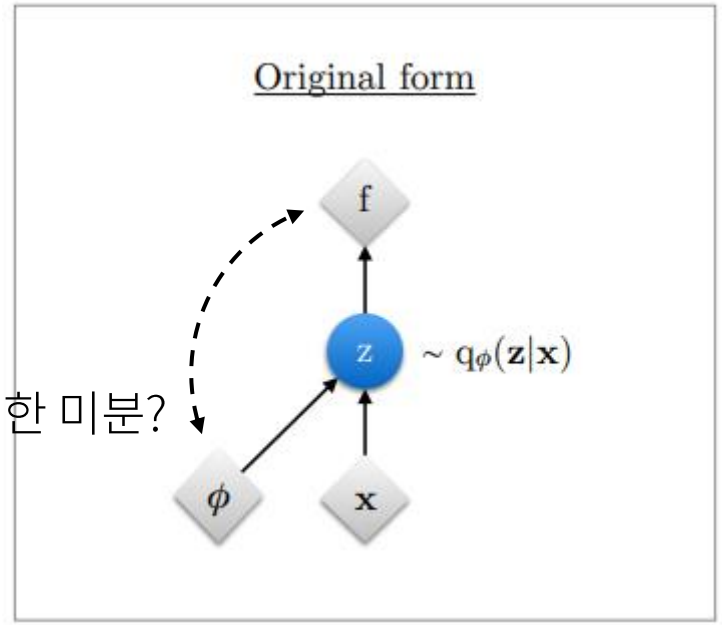


Kingma and Welling (2014). “Auto-Encoding Variational Bayes”. International Conference on Learning Representations.

An Optimization Problem

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\underbrace{\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right]}_{\text{Function of } \mathbf{z}} \quad (3)$$

$\mathcal{L}$의 $\boldsymbol{\phi}$에 대한 미분?

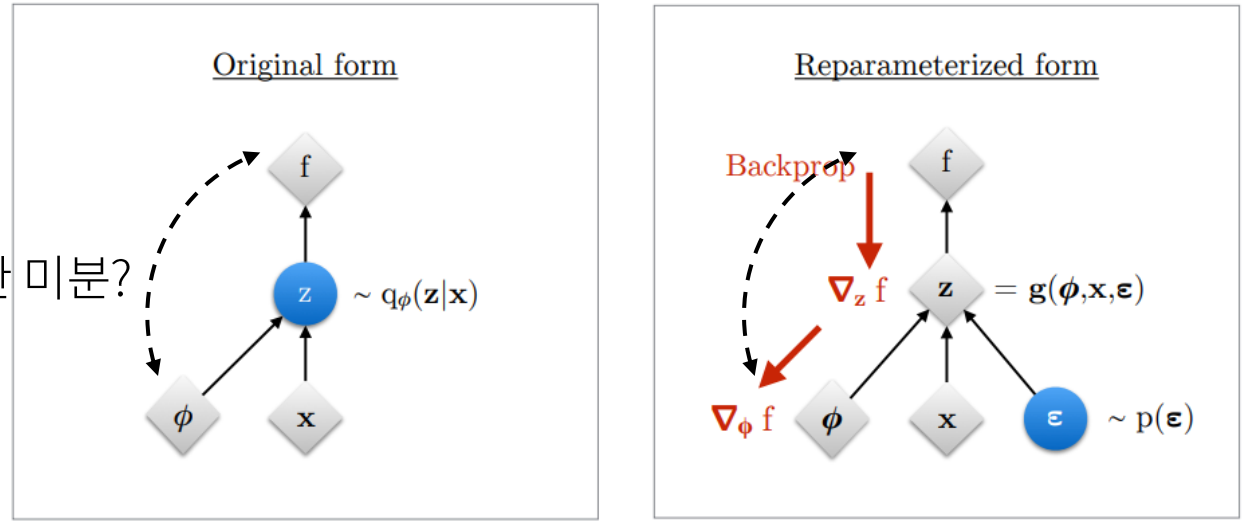


Diederik P. Kingma and Max Welling (2019), "An Introduction to Variational Autoencoders"

Re-parameterization Technique

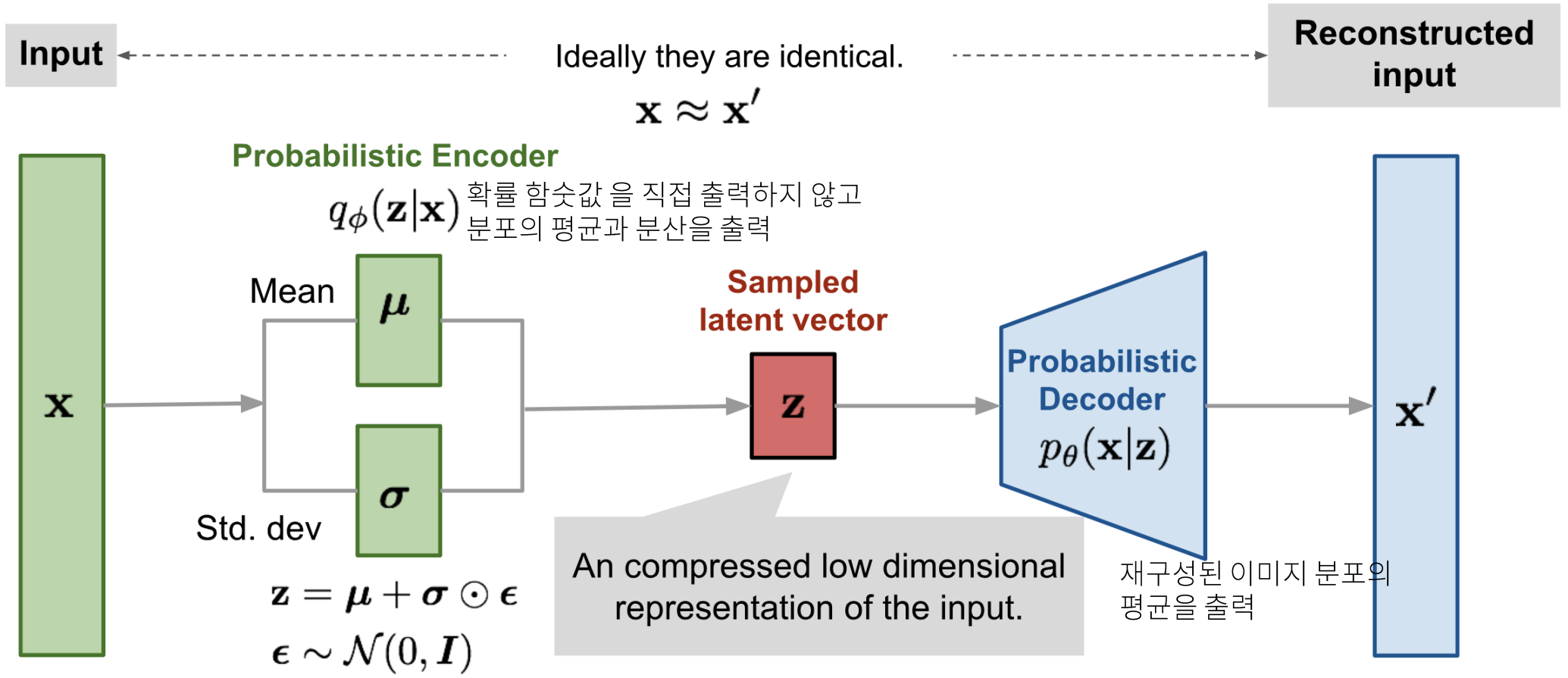
$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\underbrace{\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right]}_{\text{Function of } \mathbf{z}} \quad (3)$$

$\mathcal{L}$의 $\boldsymbol{\phi}$에 대한 미분?



Diederik P. Kingma and Max Welling (2019), "An Introduction to Variational Autoencoders"

VAE Model

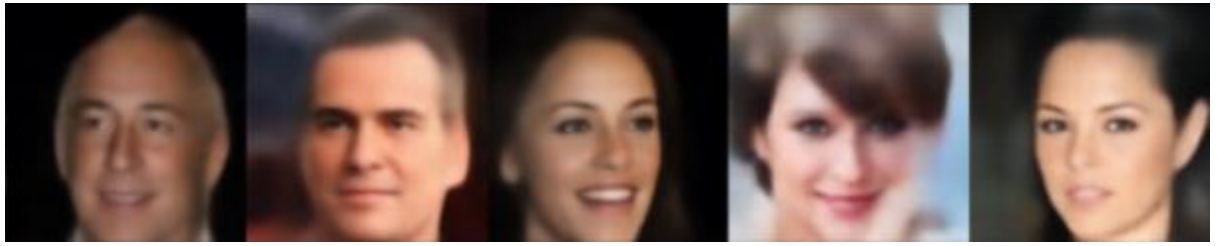


https://lilianweng.github.io/posts/2018-08-12-vae/

CelebA Results

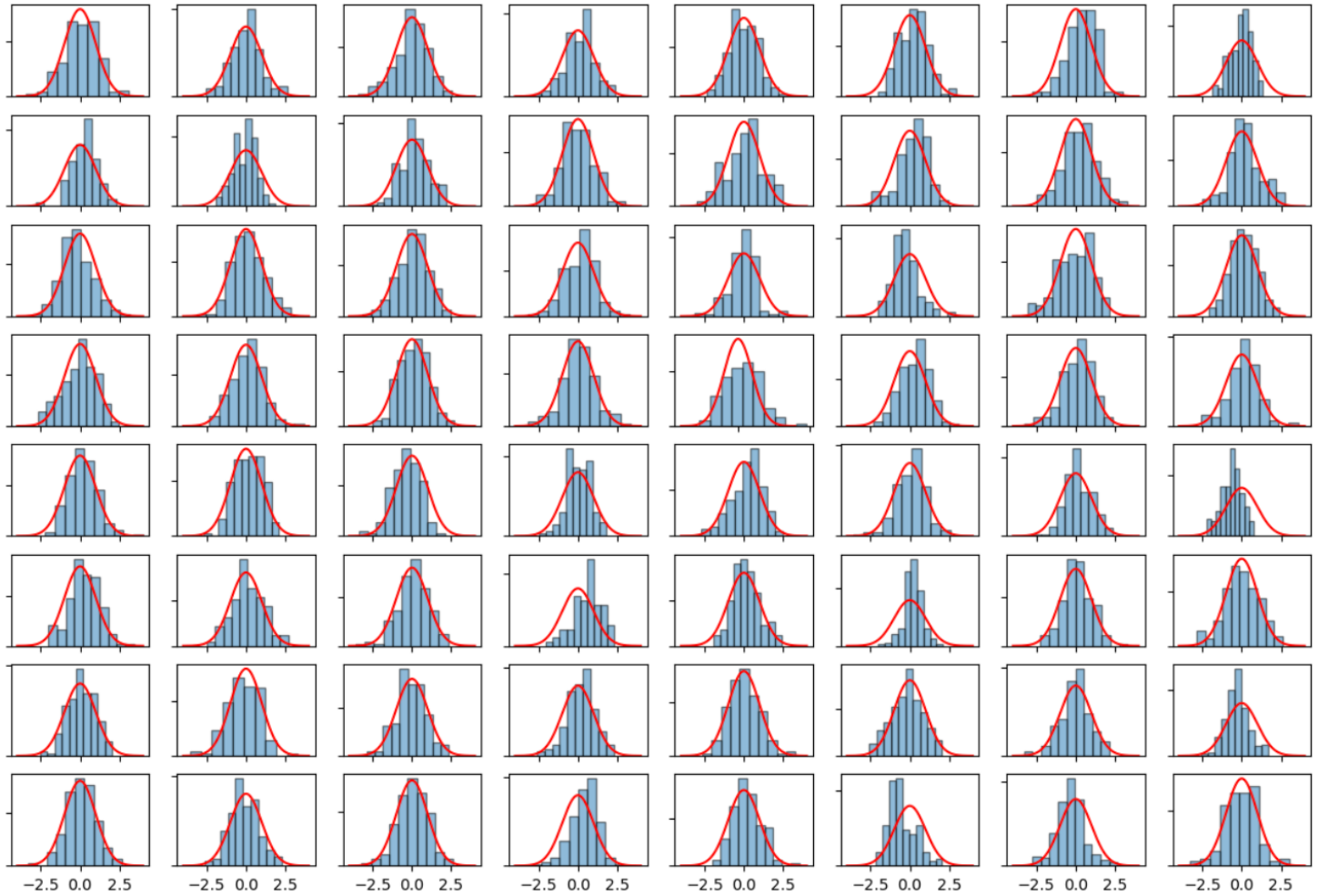
Truth Image

Reconstructed Image

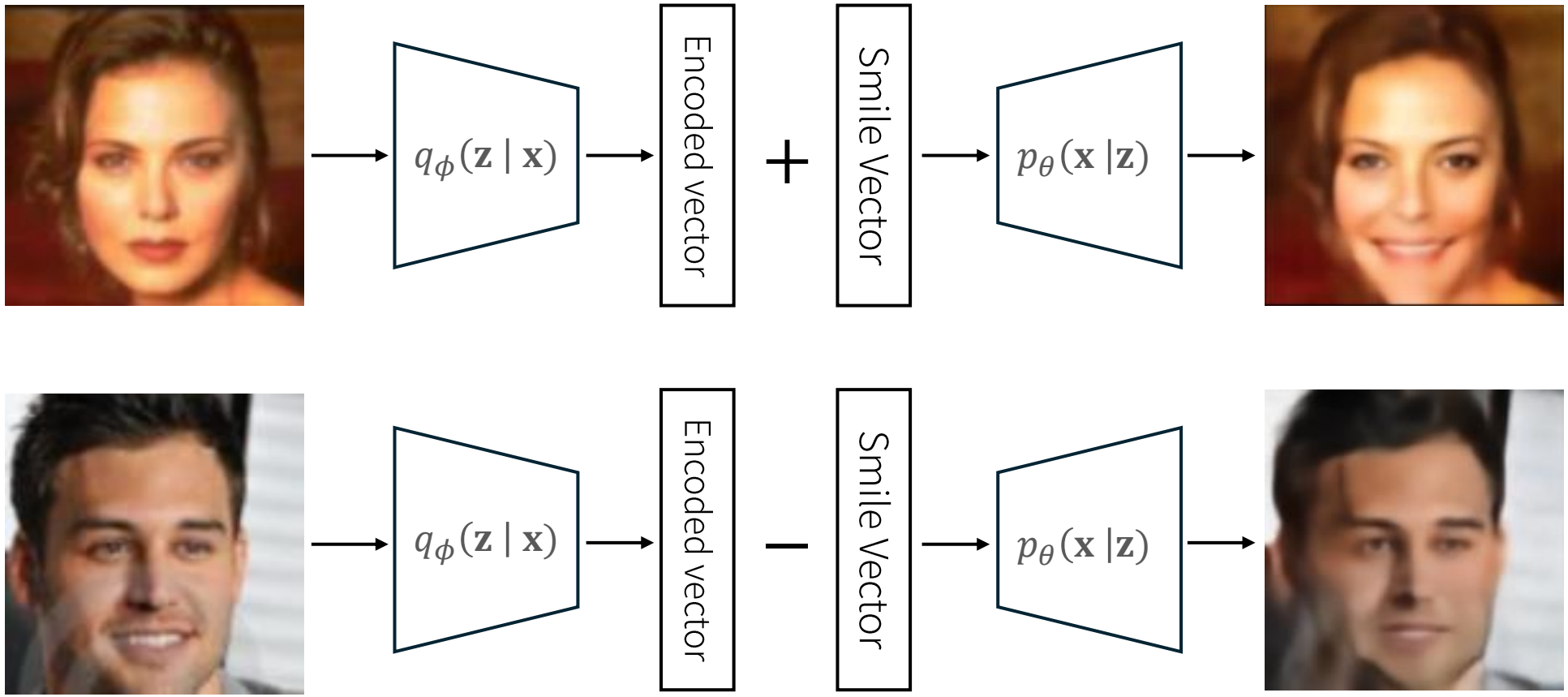


Generative Image from Random Noise

The Dist. of 64 pixels encoded by $q_\phi(\mathbf{z} \mid \mathbf{x})$



The Dist. of 64 pixels encoded by $q_\phi(\mathbf{z} \mid \mathbf{x})$



Diffusion Model

Deep Unsupervised Learning using Nonequilibrium Thermodynamics (2015)

Jascha Sohl-Dickstein el al., Stanford Univ., UC Berkeley

Diffusion Process

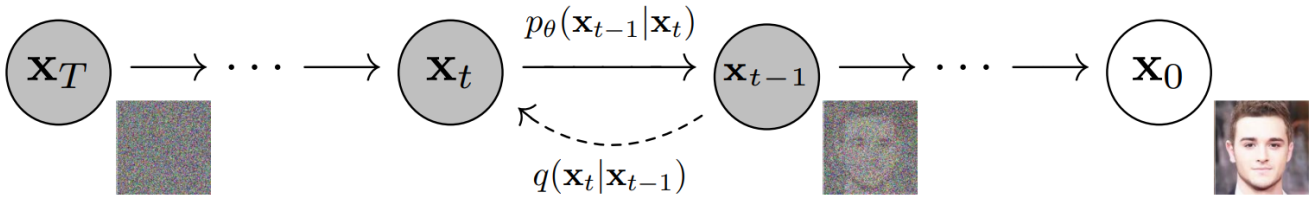


Figure 2: The directed graphical model considered in this work.

Forward Process , Diffusion Process, Inference Process

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}),$$

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

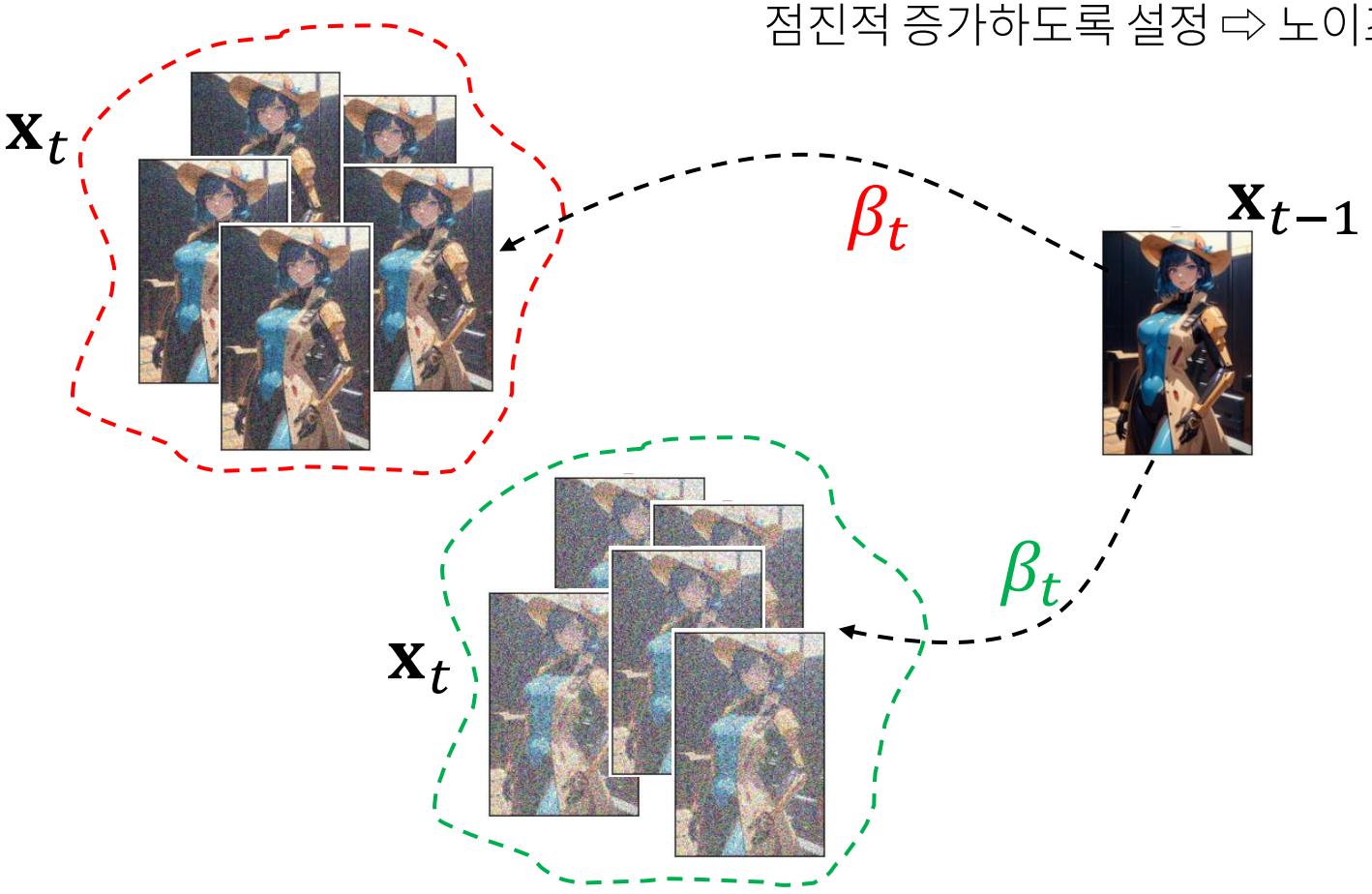
점진적 증가하도록 설정 ⇨ 노이즈 스케쥴

Diffusion Process

Forward Process , Diffusion Process, Inference Process

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

점진적 증가하도록 설정 ⇨ 노이즈 스케쥴



Generative Process

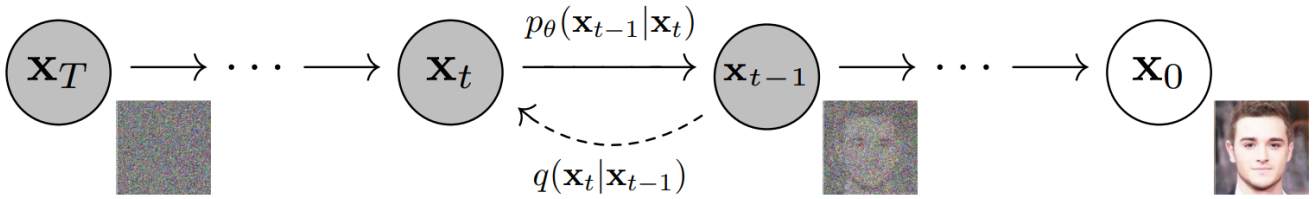


Figure 2: The directed graphical model considered in this work.

Forward Process, Diffusion Process, Inference Process

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}),$$

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

점진적 증가하도록 설정 ⇨ 노이즈 스케쥴

Reverse Process, Generative Process

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

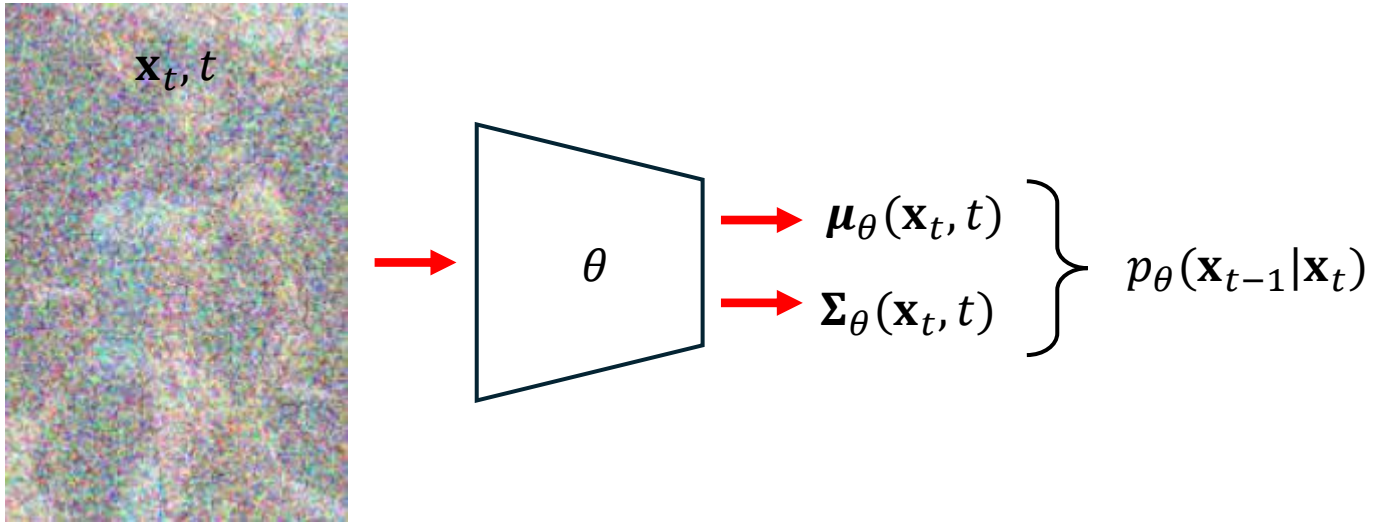
$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right),$$

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

Generative Process

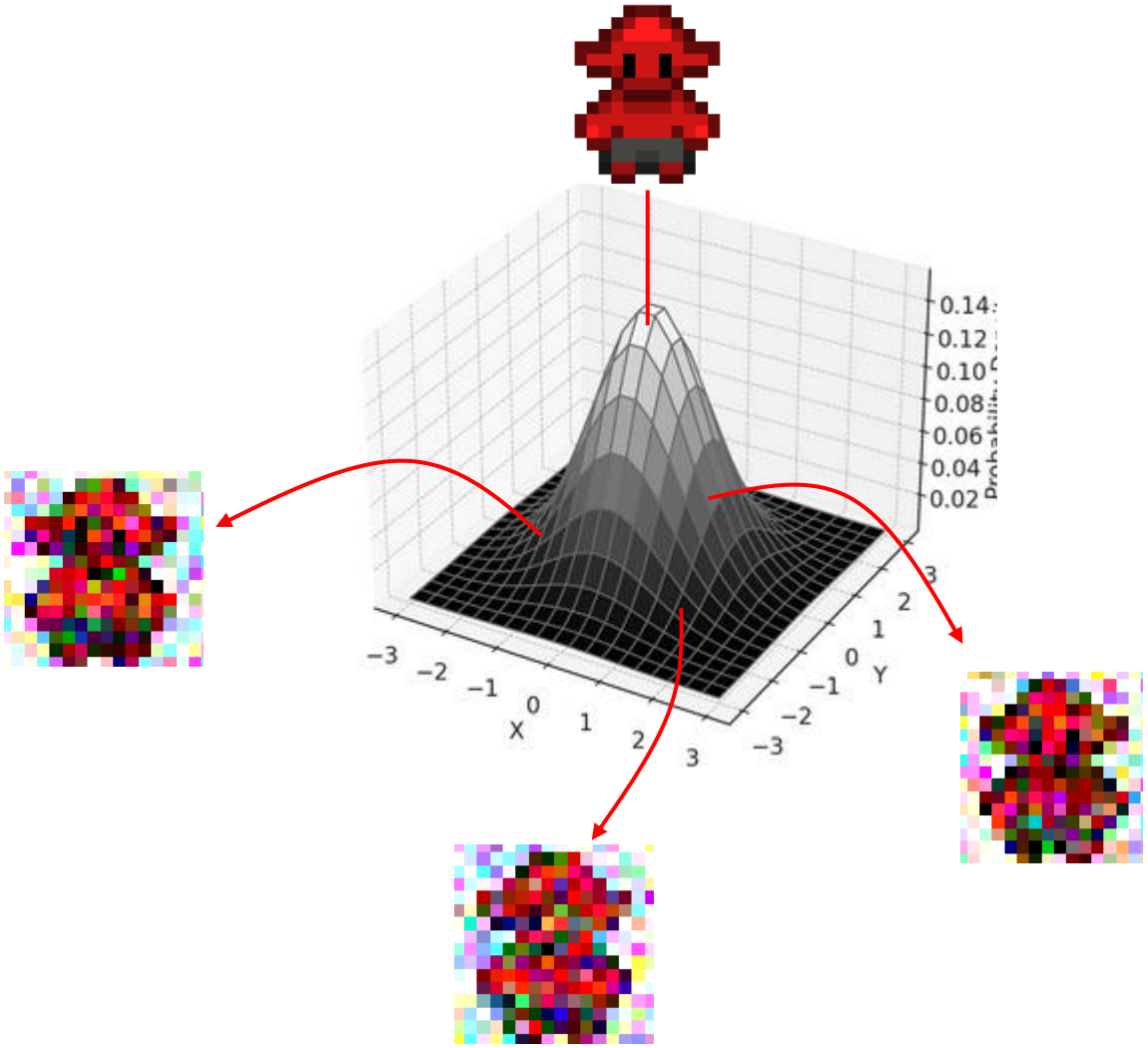
Reverse Process, Generative Process

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right),$$

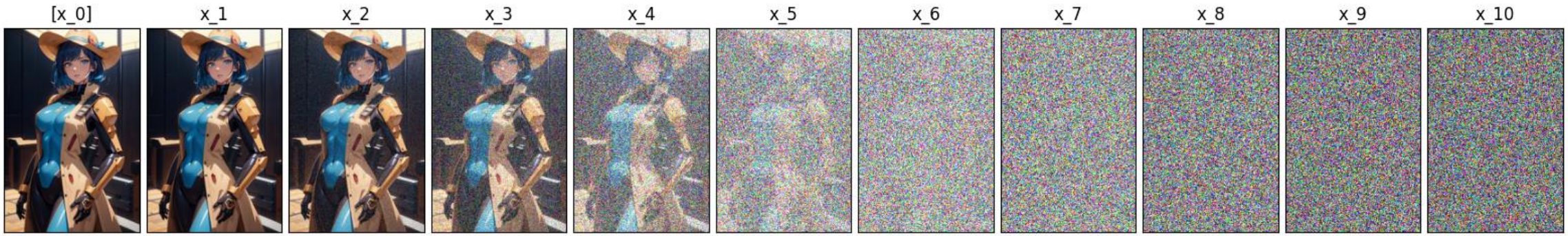


Sampling from Multi Var. Dist.

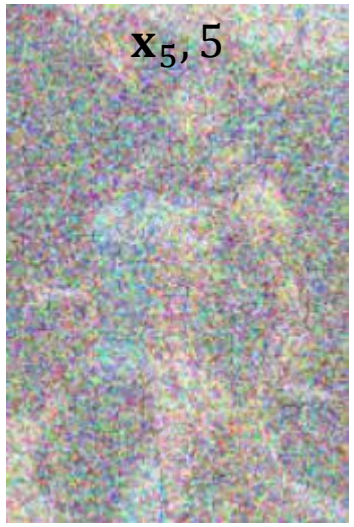
$$\mathcal{N}(\mu = \text{  }, \sigma^2)$$



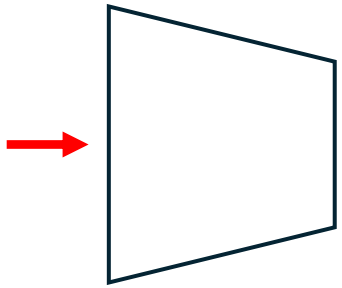
Diffusion and Generative Process



$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$



$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right),$$



$\boldsymbol{\mu}_\theta(\mathbf{x}_5, 5)$

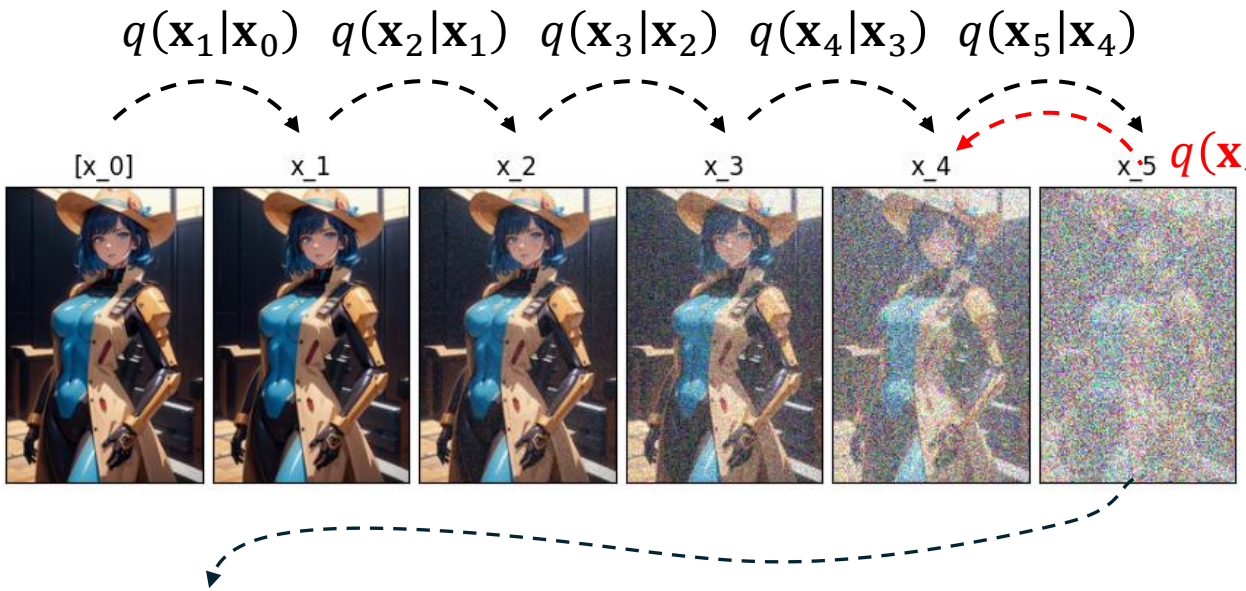
$\boldsymbol{\Sigma}_\theta(\mathbf{x}_5, 5)$

Sampling

$\mathcal{N}(\mathbf{x}_4 | \boldsymbol{\mu}_\theta(\mathbf{x}_5, 5), \boldsymbol{\Sigma}_\theta(\mathbf{x}_5, t))$

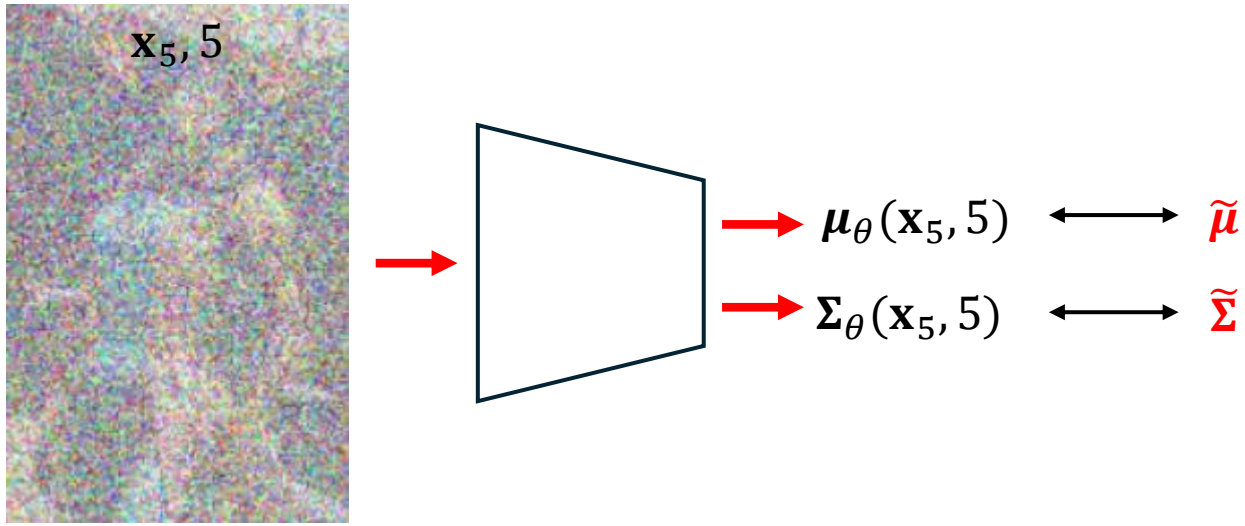
4 단계 노이지 이미지들이
모여 있는 분포

Two problems for training



Degenerating image $\mathbf{x}_t$

이미지에 노이즈를 추가하는 단계가 매우 길면(많으면)
시간 오래 걸림 ⇨ 낮은 효율

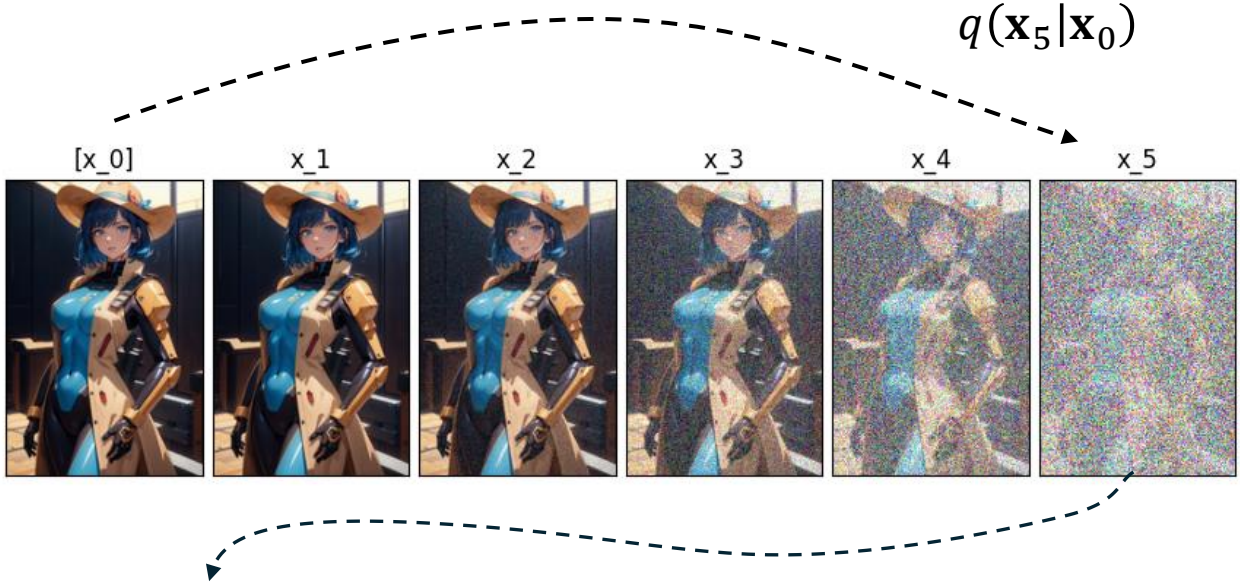


Target

$q(\mathbf{x}_4|\mathbf{x}_5)$

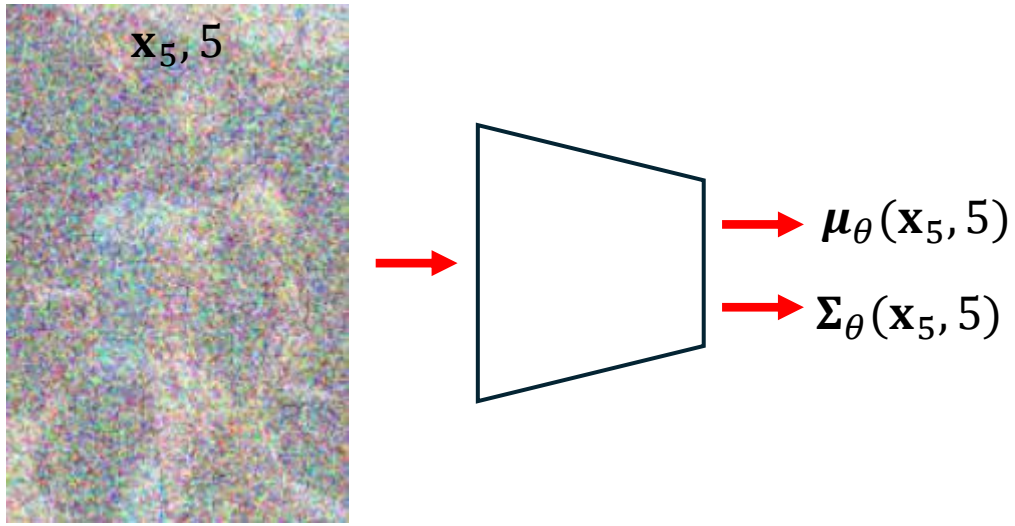
비교할 타겟이 필요

Notable Property for Diffusion Process



$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\alpha_t := 1 - \beta_t \qquad \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$$

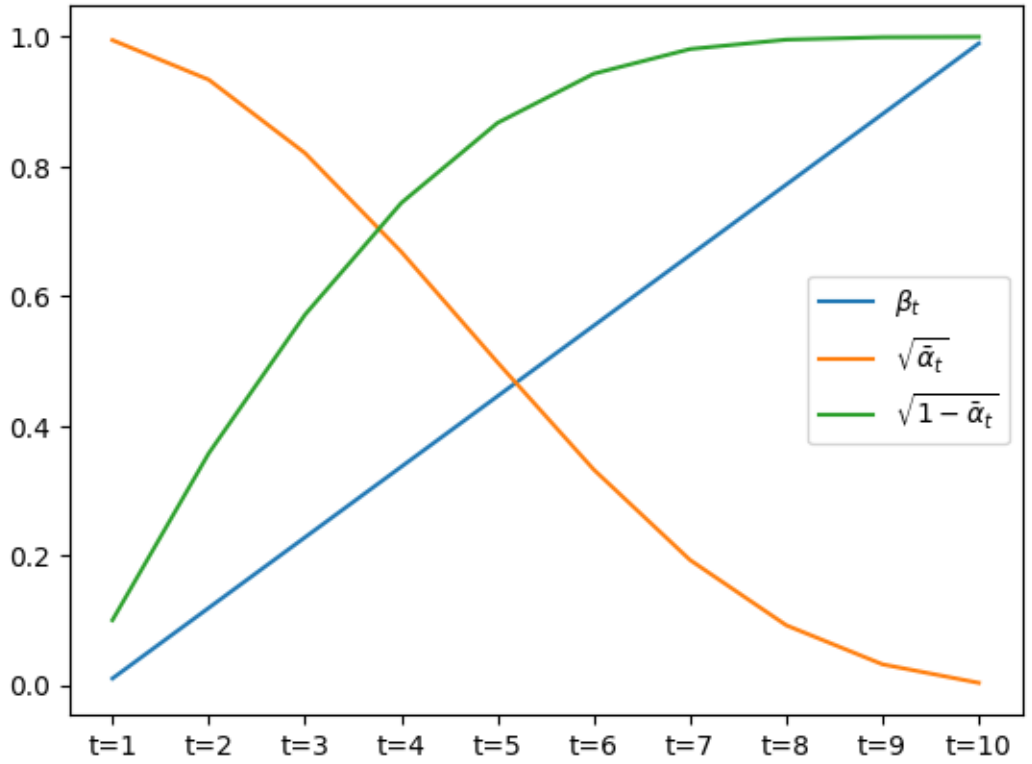


Standardization

$$Z = \frac{X - \mu}{\sigma} \qquad X = \mu + \sigma Z$$

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

Noise Schedule



$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\alpha_t := 1 - \beta_t \qquad \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$$

Target for Training

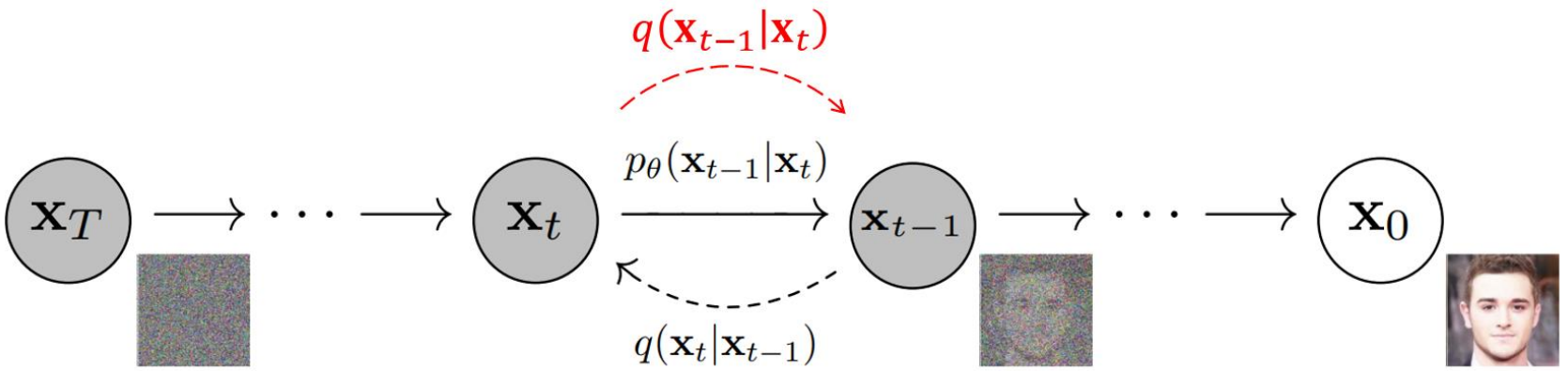


Figure 2: The directed graphical model considered in this work.

Tractable Posterior

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \qquad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right),$$

https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

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

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Forward Process　Notable Property

Tractable Posterior

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

Bayes Theorem　　Markov Assumption　　Notable Property

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

좀 복잡한 계산 후.....

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \qquad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

Variational Loss

왜 갑자기 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ 등장? $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 는 계산 못해서?

Variational Loss

$$-\log p_\theta(\mathbf{x}_0) = -\log\left(\int p_\theta(\mathbf{x}_{0:T})d\mathbf{x}_{1:T}\right)$$

깨끗한 이미지에 대한 음의 로그 가능도NLL

$$= -\log\left(\int q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}d\mathbf{x}_{1:T}\right)$$

$$= -\log\left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right]\right)$$

$$\leq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log\left(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right)\right]$$

$f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$

Jensen's Inequality

모든 $\mathbf{x}_0$에 대한 평균 음의 로그 가능도

$$\mathbb{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[-\log\left(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right)\right]$$ 음의 로그 가능도에 대한 상한

https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

Variational Loss

왜 갑자기 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ 등장? $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 는 계산 못해서?

Variational Loss

$$\mathbb{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[-\log\left(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}\mid\mathbf{x}_0)}\right)\right]$$

$$= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^{T}\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t)}{q(\mathbf{x}_t\mid\mathbf{x}_{t-1})}\right] := L$$

이 항은 계산 가능하고
이를 $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_0)$ 의 손실로 사용

곱하기를 더하기로 변환

$$-\log\left(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}\mid\mathbf{x}_0)}\right) = -\log\left(\frac{p(\mathbf{x}_T)\prod_{t=1}^{T}p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t)}{\prod_{t=1}^{T}q(\mathbf{x}_t\mid\mathbf{x}_{t-1})}\right)$$

$$= -\log\left(p(\mathbf{x}_T)\cdot\frac{\prod_{t=1}^{T}p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t)}{\prod_{t=1}^{T}q(\mathbf{x}_t\mid\mathbf{x}_{t-1})}\right)$$

$$= -\log p(\mathbf{x}_T) - \log\prod_{t=1}^{T}\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t)}{q(\mathbf{x}_t\mid\mathbf{x}_{t-1})}$$

$$= -\log p(\mathbf{x}_T) - \sum_{t=1}^{T}\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t)}{q(\mathbf{x}_t\mid\mathbf{x}_{t-1})}$$

https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

Variational Loss

왜 갑자기 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ 등장? $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 는 계산 못해서?

Variational Loss

$$L = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[-\log\left(\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}\mid\mathbf{x}_0)}\right)\right]$$

$$= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^{T}\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t)}{q(\mathbf{x}_t\mid\mathbf{x}_{t-1})}\right]$$

좀 복잡한 계산 후.....

$$= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\underbrace{D_{KL}(q(\mathbf{x}_T\mid\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T}\underbrace{D_{KL}(q(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1)}_{L_0}\right]$$

Posterior와 네트워크가 만드는 분포의 차이

https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

Training Loss Function

Training Loss

Gaussian?

Gaussian Implicitly

$$L = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\underbrace{D_{KL}(q(\mathbf{x}_T \mid \mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T}\underbrace{D_{KL}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}_{L_0}\right]$$

$$D_{KL}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)) = D_{KL}(\mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \,\|\, \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}))$$

$$= \frac{1}{2}\left((\boldsymbol{\mu}_\theta - \tilde{\boldsymbol{\mu}}_t)^T(\sigma_t^2\mathbf{I})^{-1}(\boldsymbol{\mu}_\theta - \tilde{\boldsymbol{\mu}}_t) + \underbrace{\operatorname{tr}\left((\sigma_t^2\mathbf{I})^{-1}\tilde{\beta}_t\mathbf{I}\right) - d + \log\frac{\det\sigma_t^2\mathbf{I}}{\det\tilde{\beta}\mathbf{I}}}_{C}\right)$$

$$= \frac{1}{2\sigma_t^2}\|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)\|_2^2 + C$$

두 분포의 평균을 비교하는 형태

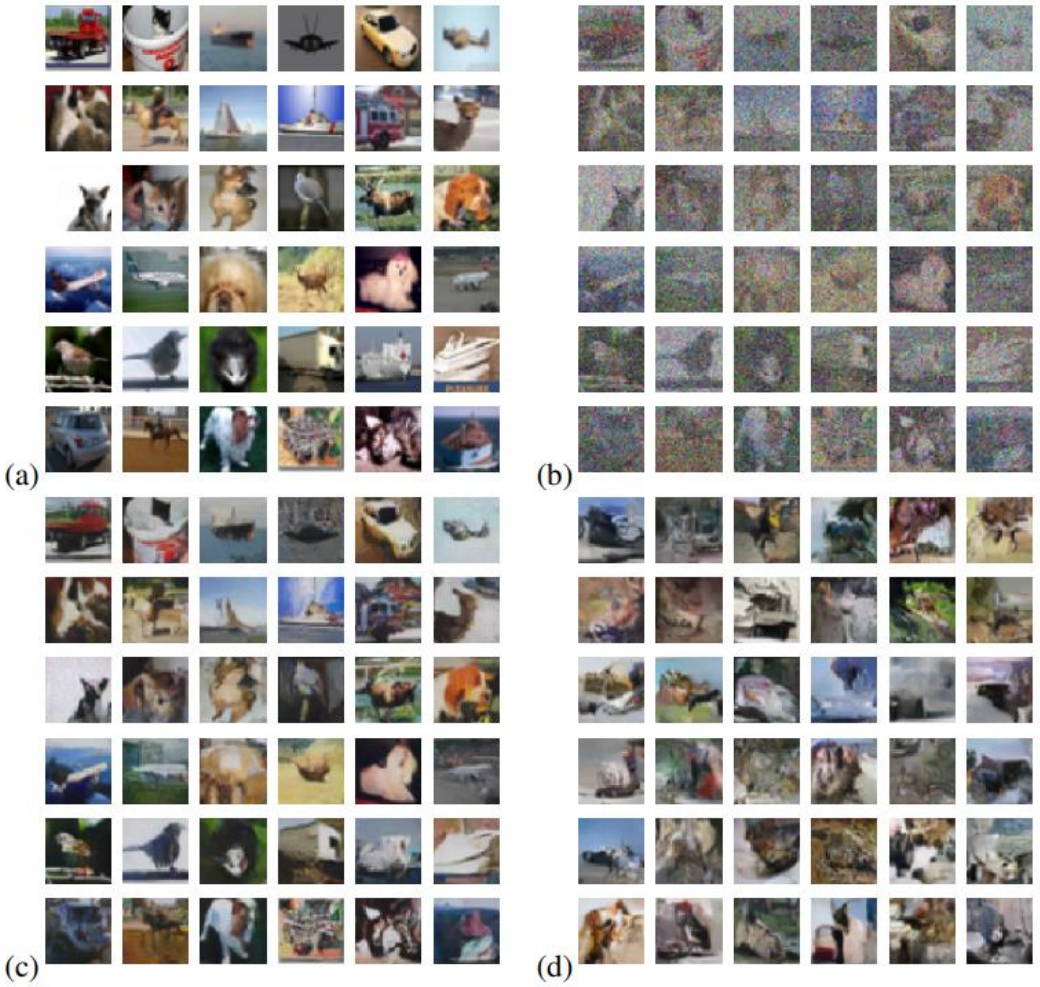
https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \tag{6}$$

$$\text{where} \quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \quad \text{and} \quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{7}$$

원본 이미지

노이즈 추가



원본 $\mathbf{x_0}$, 노이지 $\mathbf{x}_t$를 사용해서 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x_0})$를 이용해서 샘플링

$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x_0})$ 의 $\tilde{\boldsymbol{\mu}}_t$ 를 예측하는 디퓨전 모델에서 샘플링

Figure 3. The proposed framework trained on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. *(a)* Example holdout data (similar to training data). *(b)* Holdout data corrupted with Gaussian noise of variance 1 (SNR = 1). *(c)* Denoised images, generated by sampling from the posterior distribution over denoised images conditioned on the images in (b). *(d)* Samples generated by the diffusion model.

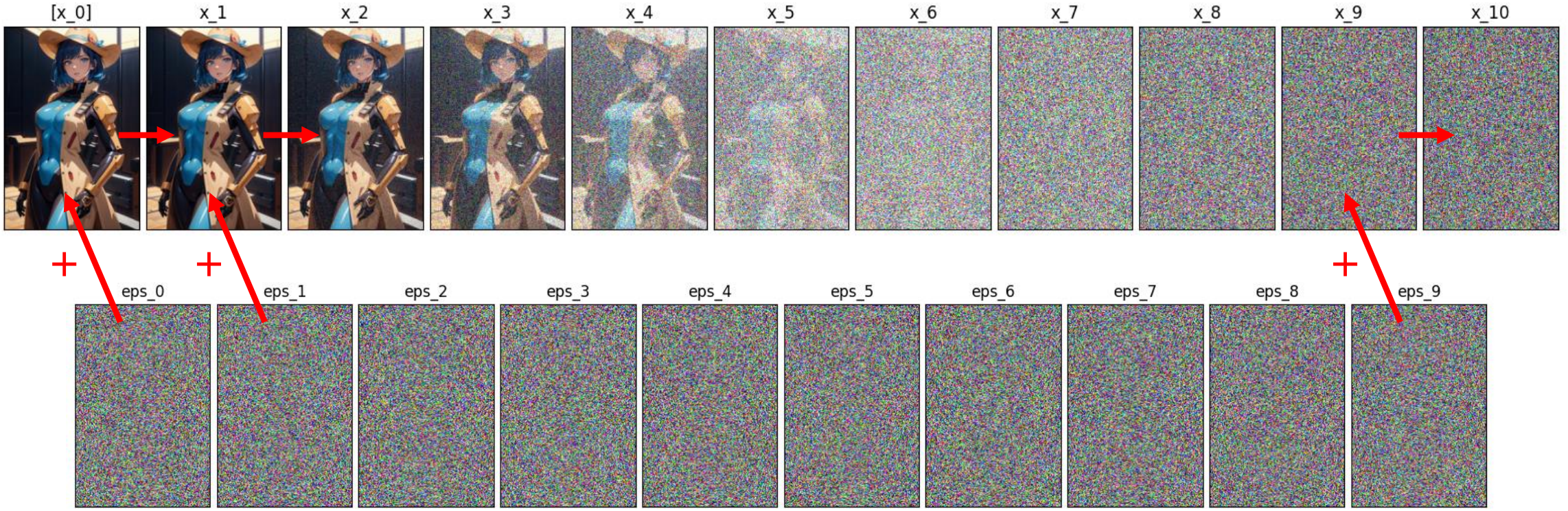
DDPM

Denoising Diffusion Probabilistic Models (2020)

Jonathan Ho el al., UC Berkeley

Diffusion Process

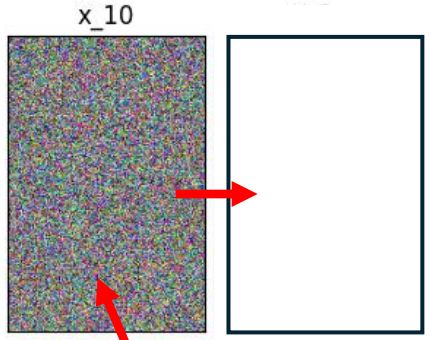
Random Gaussian Noise



$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

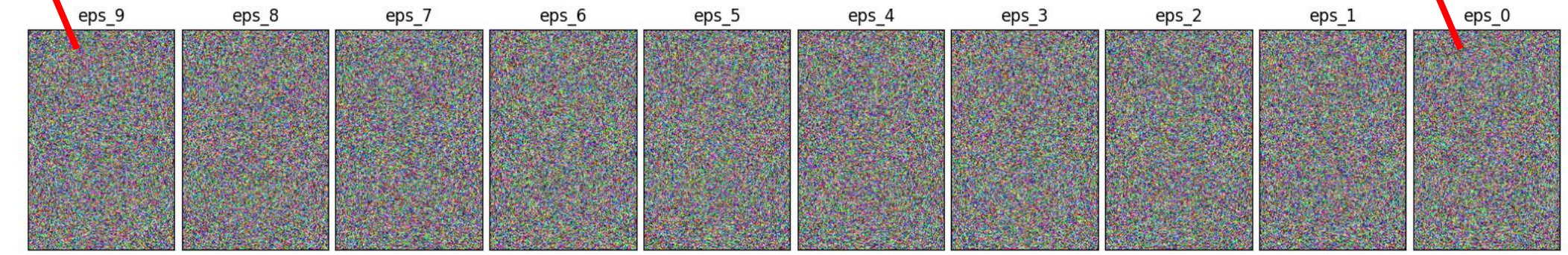
$$\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \text{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Reverse Process



$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right),$$

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



noise_experiment.ipynb

Predict Noise

$$= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\underbrace{D_{KL}(q(\mathbf{x}_T \mid \mathbf{x}_0) || p(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{KL}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) || p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}_{L_0} \right]$$

$$= \frac{1}{2\sigma_t^2} \| \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) \|_2^2 + C$$

두 분포의 평균을 비교하는 형태

$$L_{t-1} - C = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{1}{2\sigma_t^2} \cdot \frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)} \| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}_{\mathbf{x}_t}, t) \|_2^2 \right]$$

노이즈를 비교하는 형태

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|_2^2 \right]$$

https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

DDPM: Algorithm

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ ← 원본 이미지
3: $t \sim \text{Uniform}(\{1, \dots, T\})$ ← 임의의 시간단계
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ← 임의의 노이즈
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

모델의 출력 노이즈 추가된 이미지 $\mathbf{x}_t$

Algorithm 2 Sampling

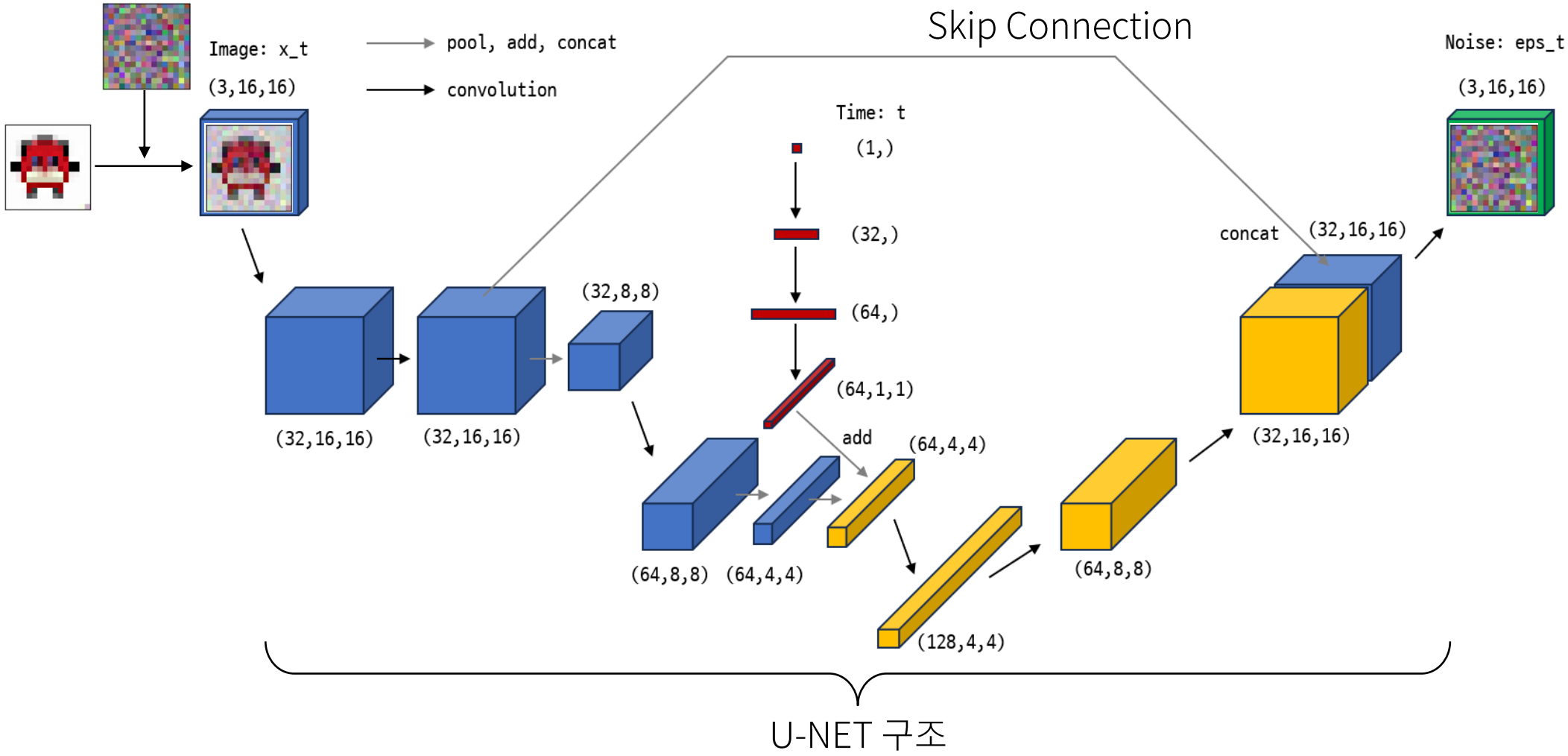
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}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

예측된 노이즈로 부터 계산된 t-1 단계의 노이즈가 조금 줄어든 이미지 분포의 평균

https://metamath1.github.io/blog/posts/diffusion/ddpm_part1.html

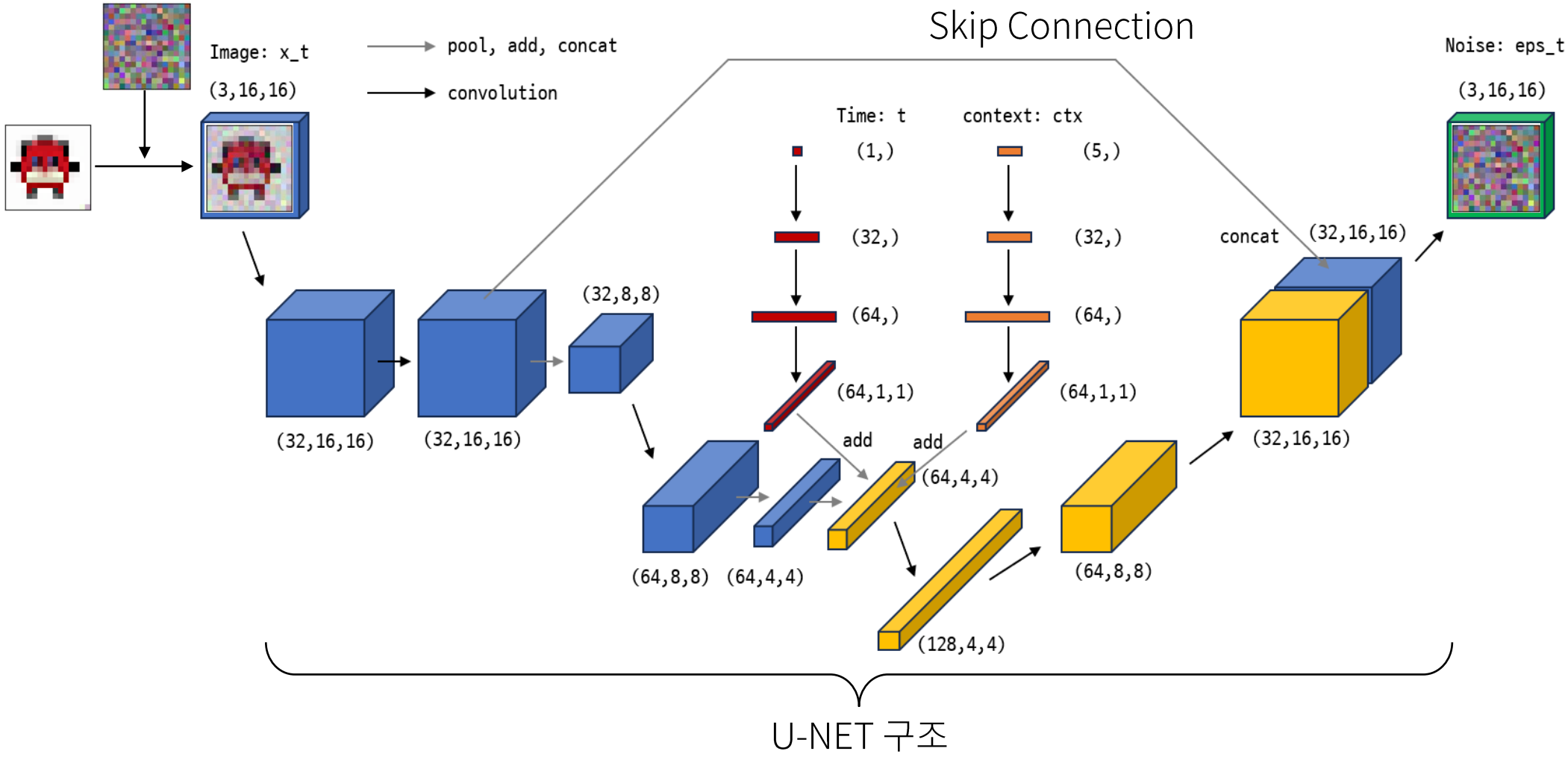
DDPM: Proof of Concept

Unconditional Generation



DDPM: Proof of Concept

Conditional Generation

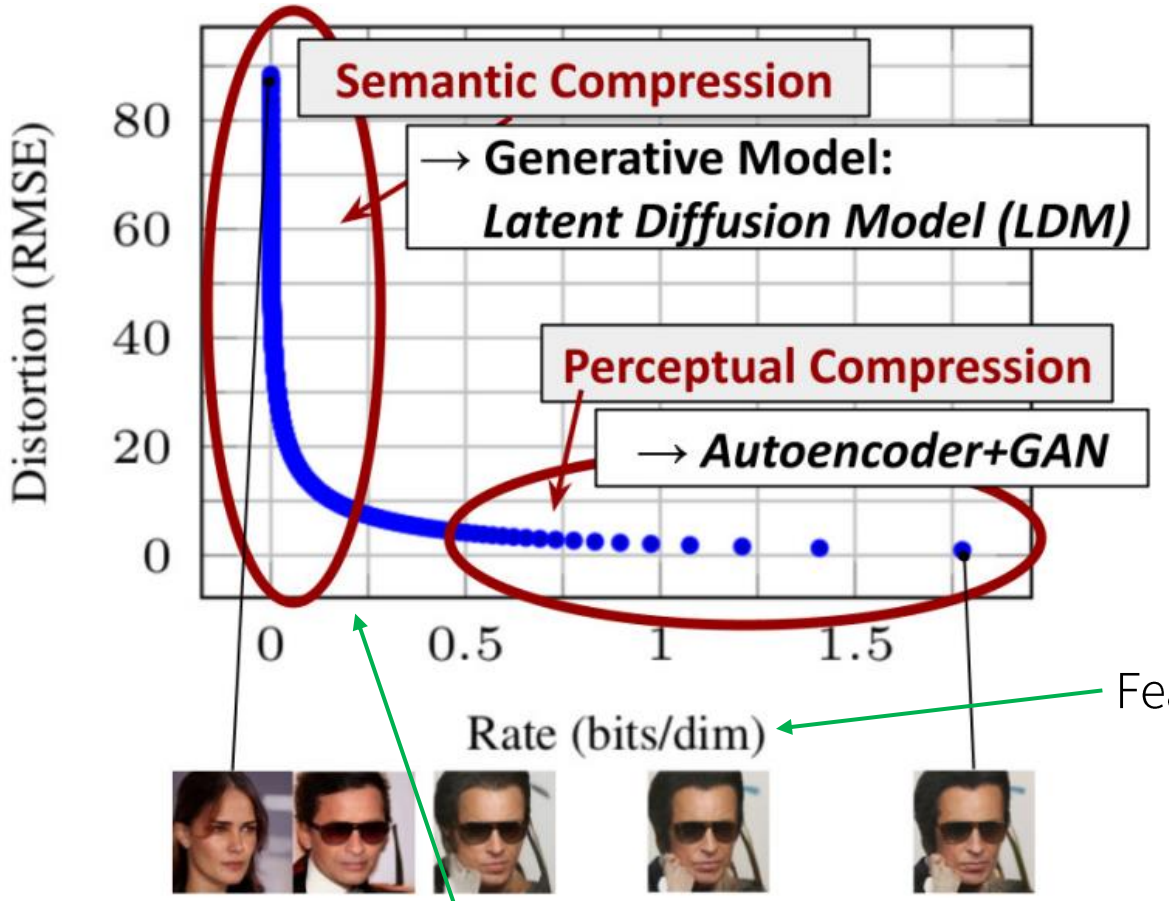


LDM(Stable Diffusion)

High-Resolution Image Synthesis with Latent Diffusion Models (2022)

Robin Rombach el al., Runway ML

Motivation



Feature의 dim당 bit 수

Rate가 줄어도 의미적 정보는 사라지지 않음

LDM 구조

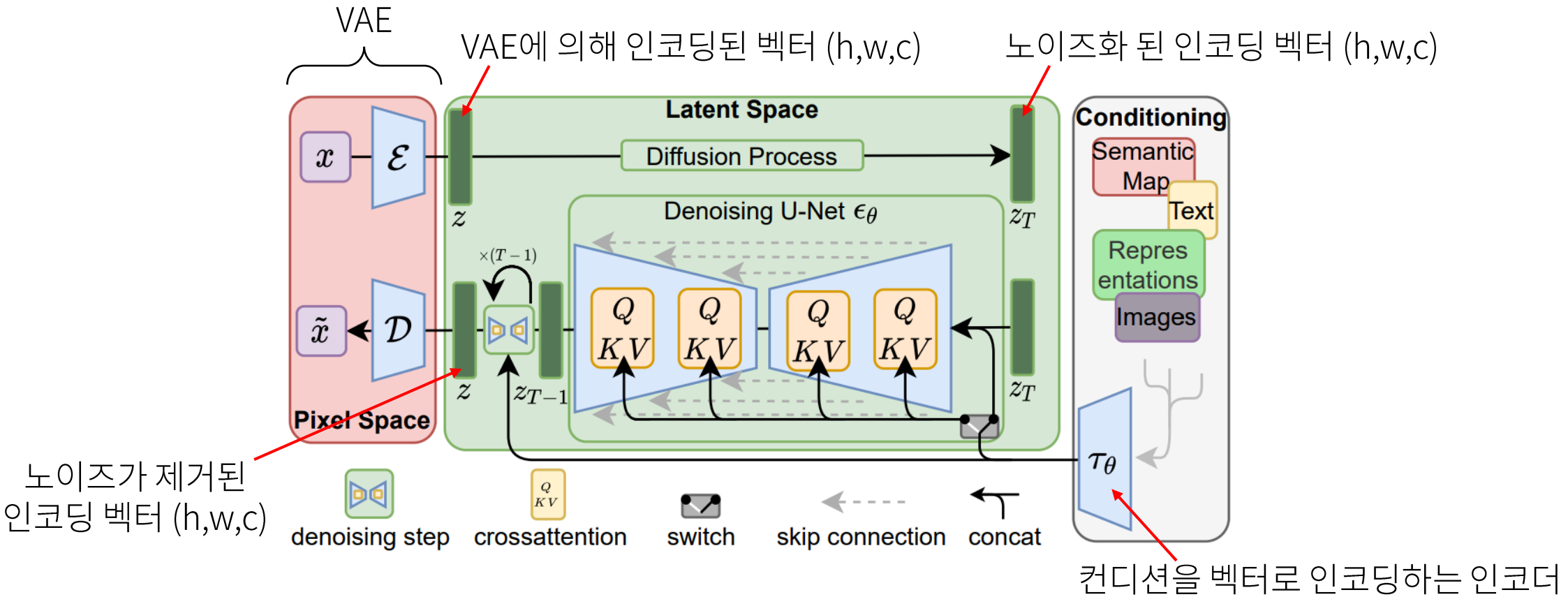
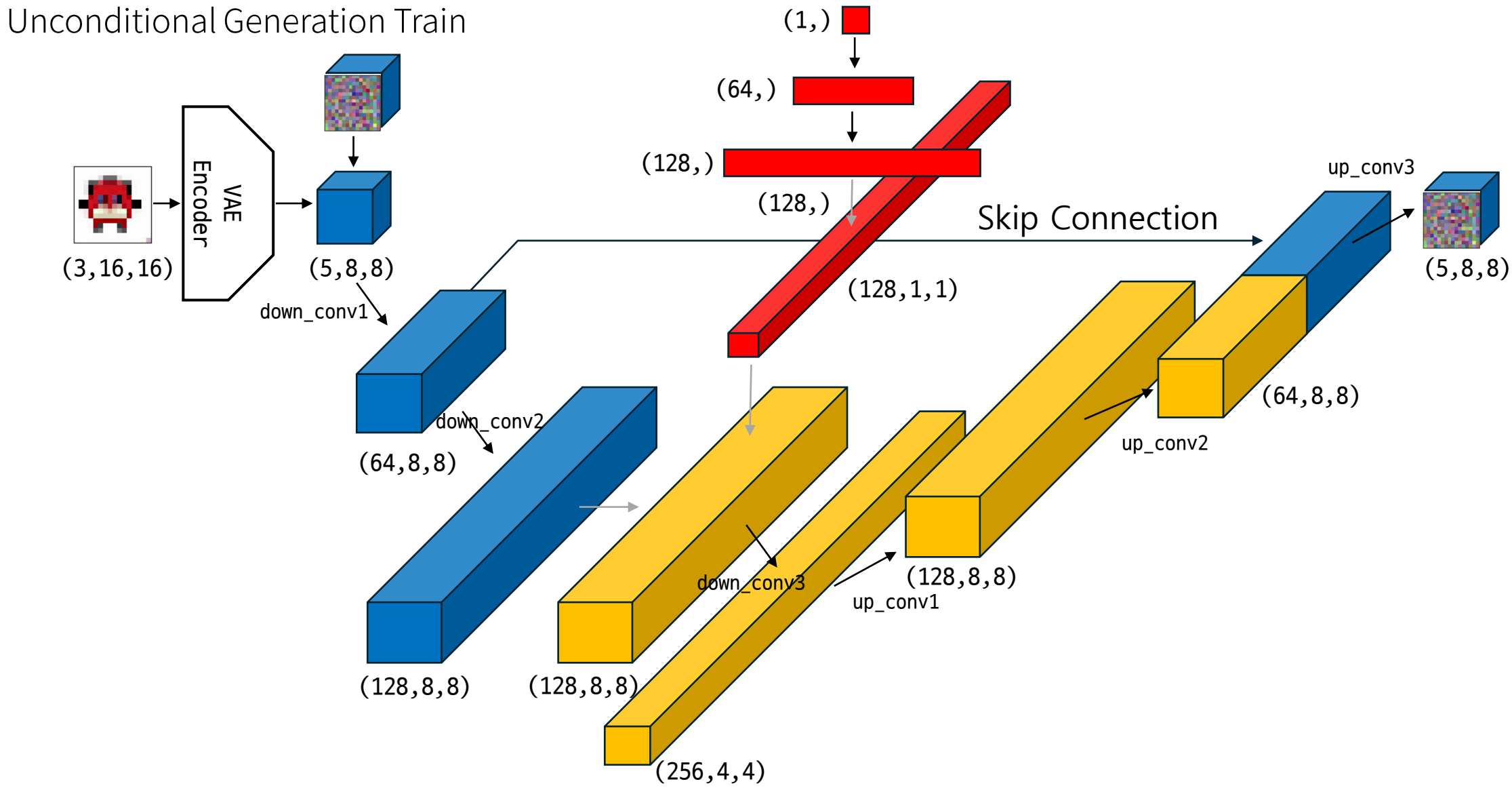


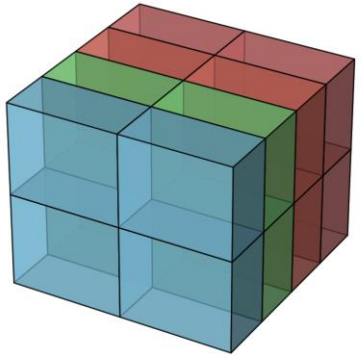
Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism. See Sec. 3.3

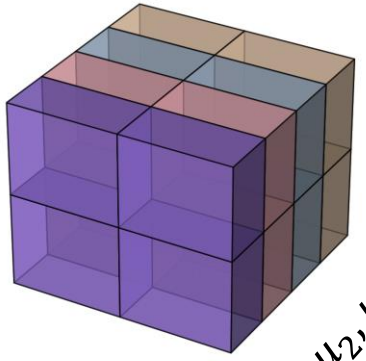
LDM: Proof of Concept



Group Normalization

x.shape: (2, 4, 2, 2)



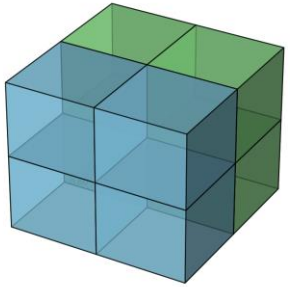


BatchNorm(4)

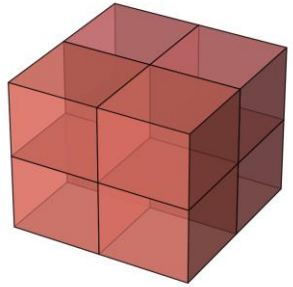
$\mu_1, \mu_2, \mu_3, \mu_4$

GroupNorm(2, 4)

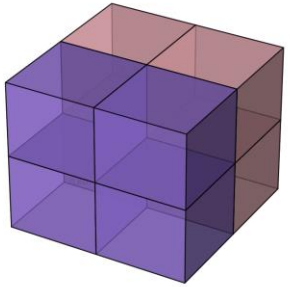
(batch, # groups, ch/group, h, w) = (2, 2, 2, 2, 2)



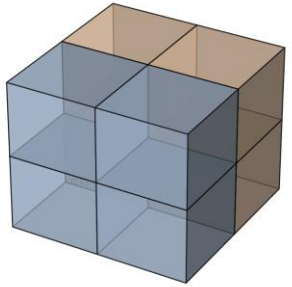
μ_1



μ_2



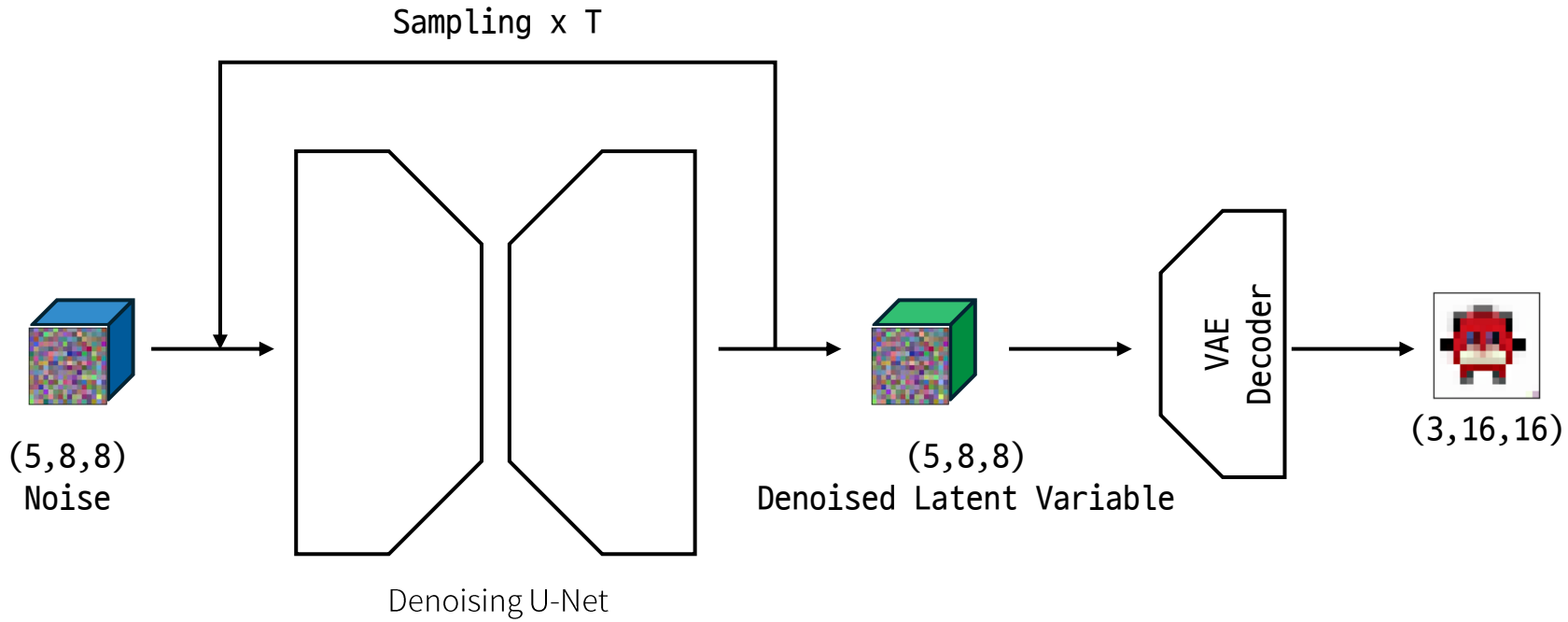
μ_3



μ_4

LDM: Proof of Concept

Generation



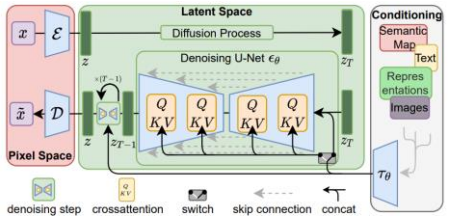

Diffusers

🤗 Diffusers

🤗 Diffusers 라이브러리 소개

Noise: [1, 4, 64, 64]

"a photograph of an astronaut riding a horse"



핵심 기능

- **최신 확산 모델 제공**: 이미지, 오디오, 3D 구조 생성
- **모듈식 도구 상자**: 추론 및 모델 훈련 모두 지원

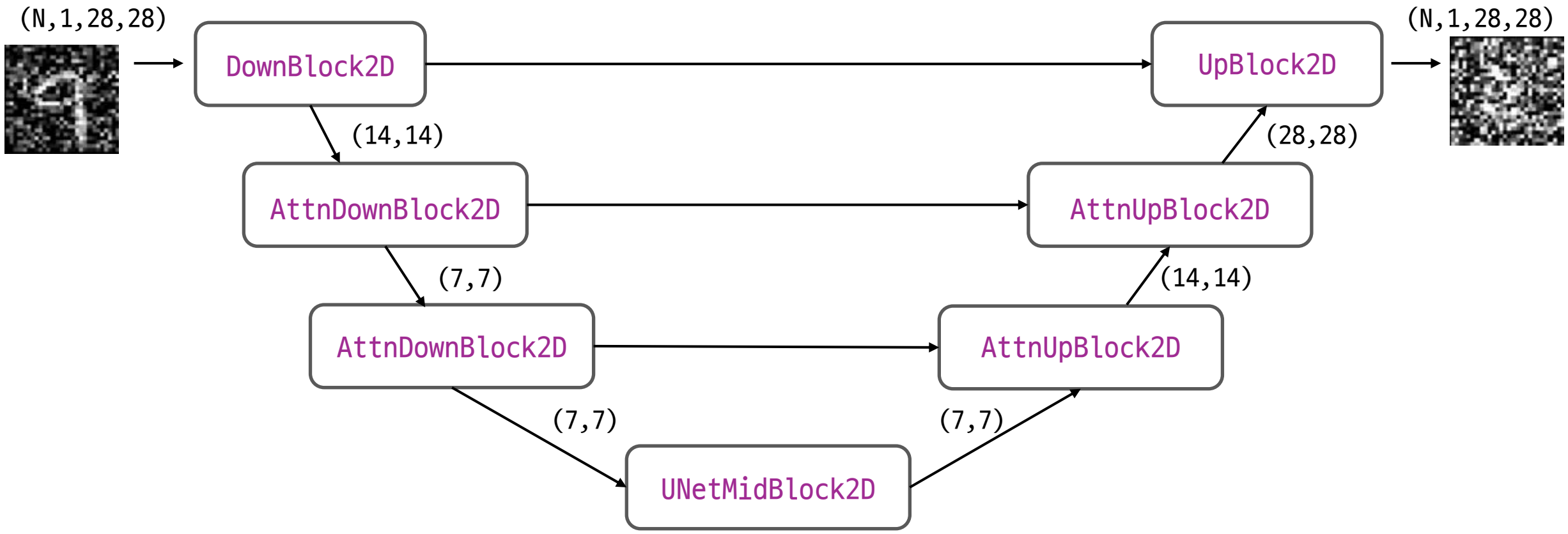
설계 철학

- **사용성 중시**: 성능보다 사용성을 우선
- **단순함 중시**: 복잡한 추상화보다 단순함 강조

주요 구성 요소

- **확산 파이프라인**: 간단한 코드로 추론 가능
- **교체 가능한 노이즈 스케줄러**: 생성 속도와 품질 조정
- **사전 학습된 모델**: end to end 확산 시스템 구축 가능

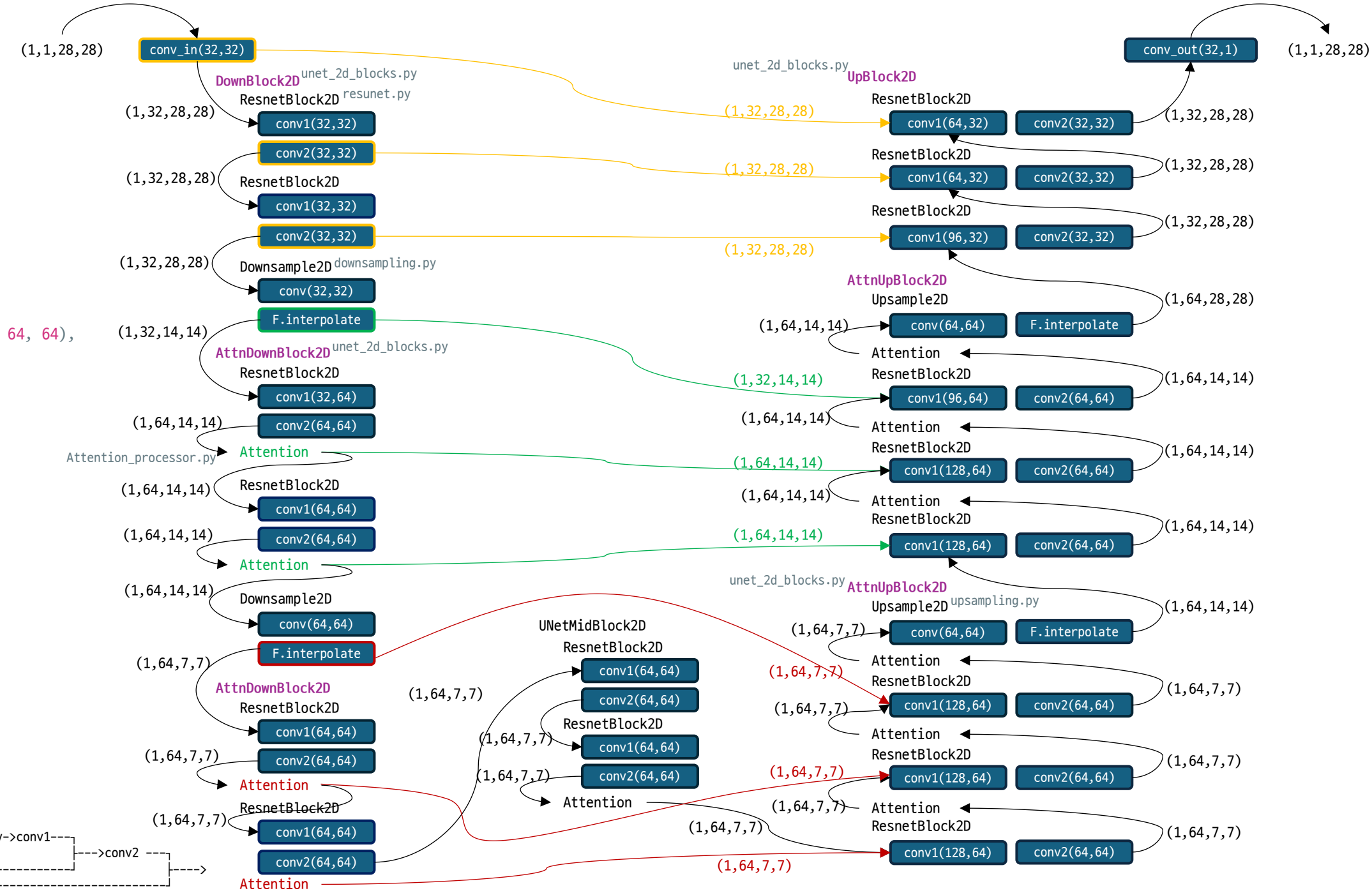
UNet2DModel



UNet2DModel

unet_2d.py

```
model = UNet2DModel(
  sample_size=28,
  in_channels=1 ,
  out_channels=1,
  layers_per_block=2,
  block_out_channels=(32, 64, 64),
  down_block_types=(
    "DownBlock2D",
    "AttnDownBlock2D",
    "AttnDownBlock2D",
  ),
  up_block_types=(
    "AttnUpBlock2D",
    "AttnUpBlock2D",
    "UpBlock2D",
  )
)
```

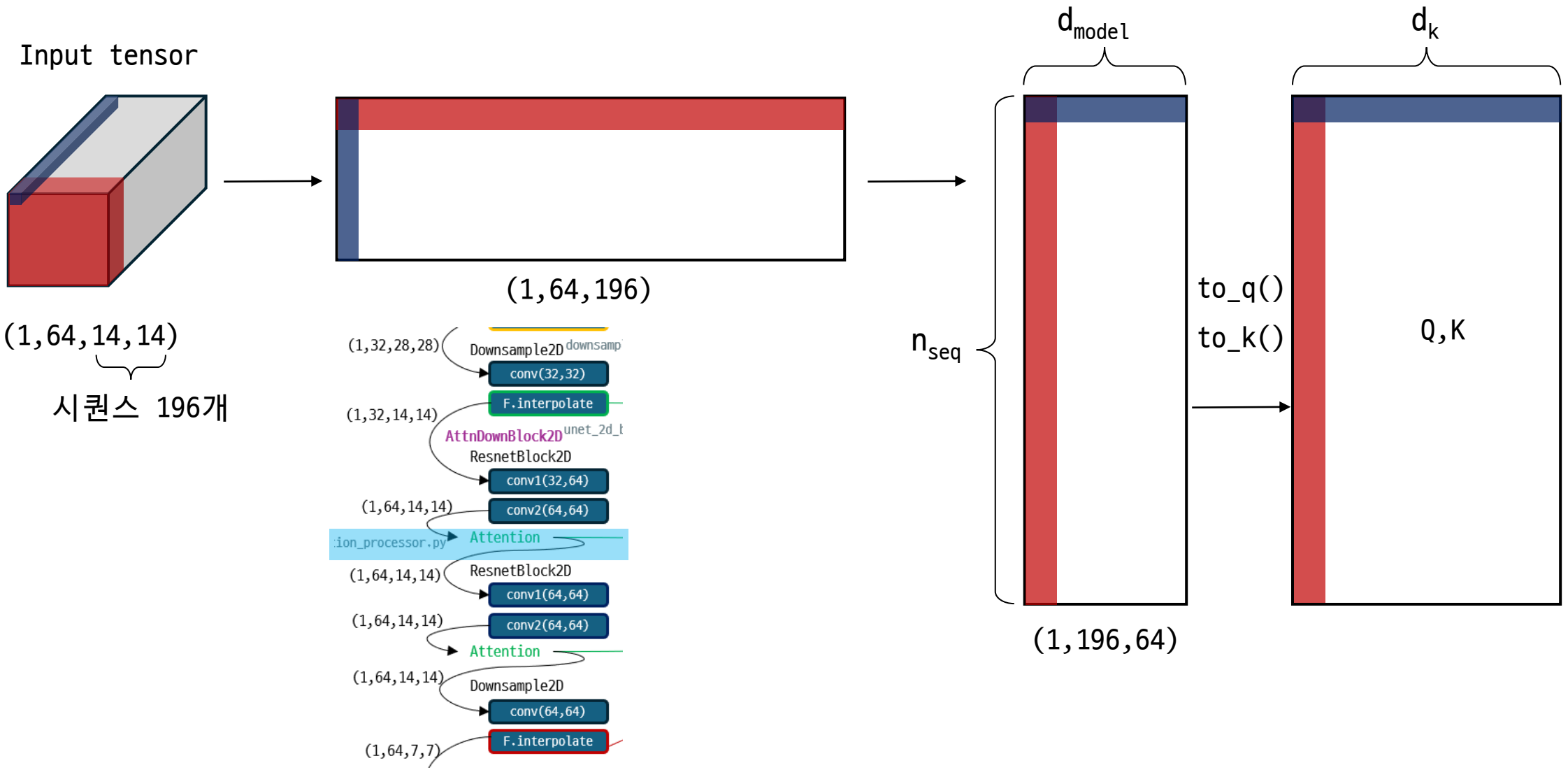


ResnetBlock2D의 기본 동작 방식

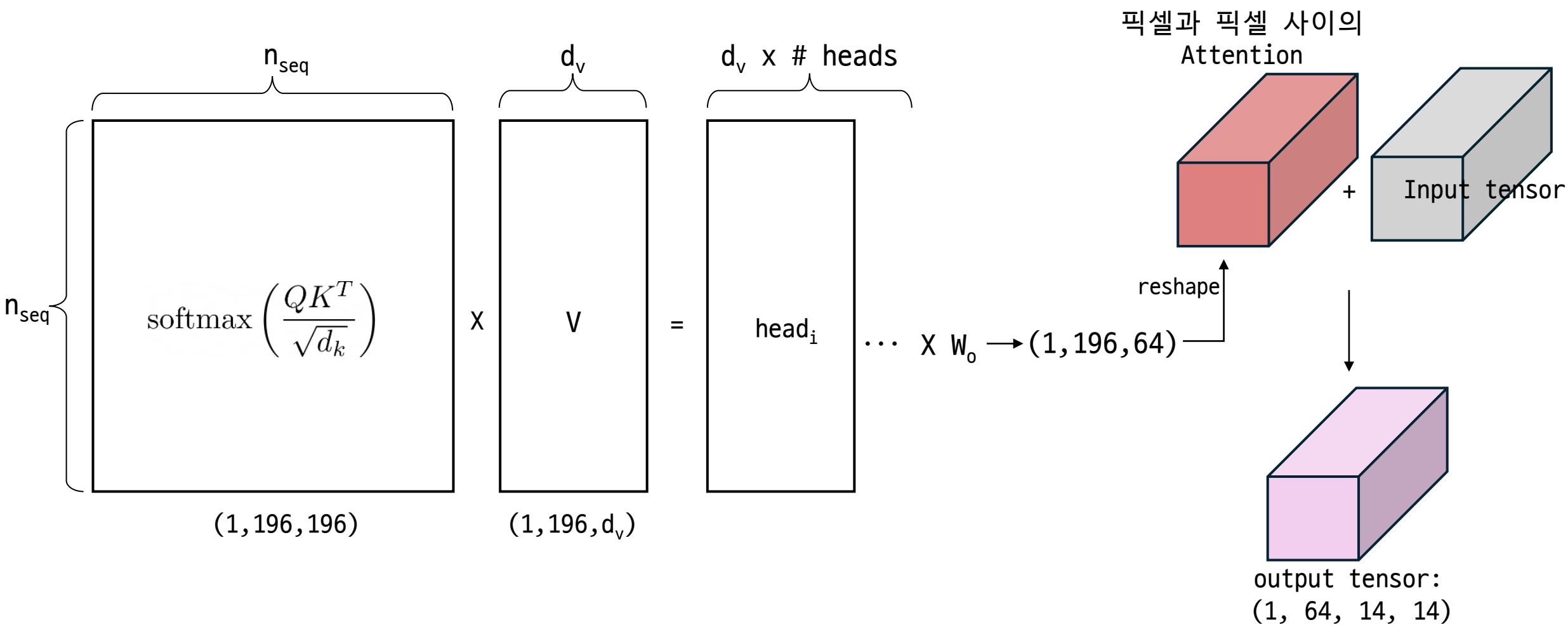
입력: input_tensor, temb

```
input_tensor->norm1->nonlinearity->conv1---┐
                                           ├--->conv2 ---┐
temb->time_emb_proj ----------------------┘              ├---->
input_tensor->conv_shortcut-------------------------------┘
```

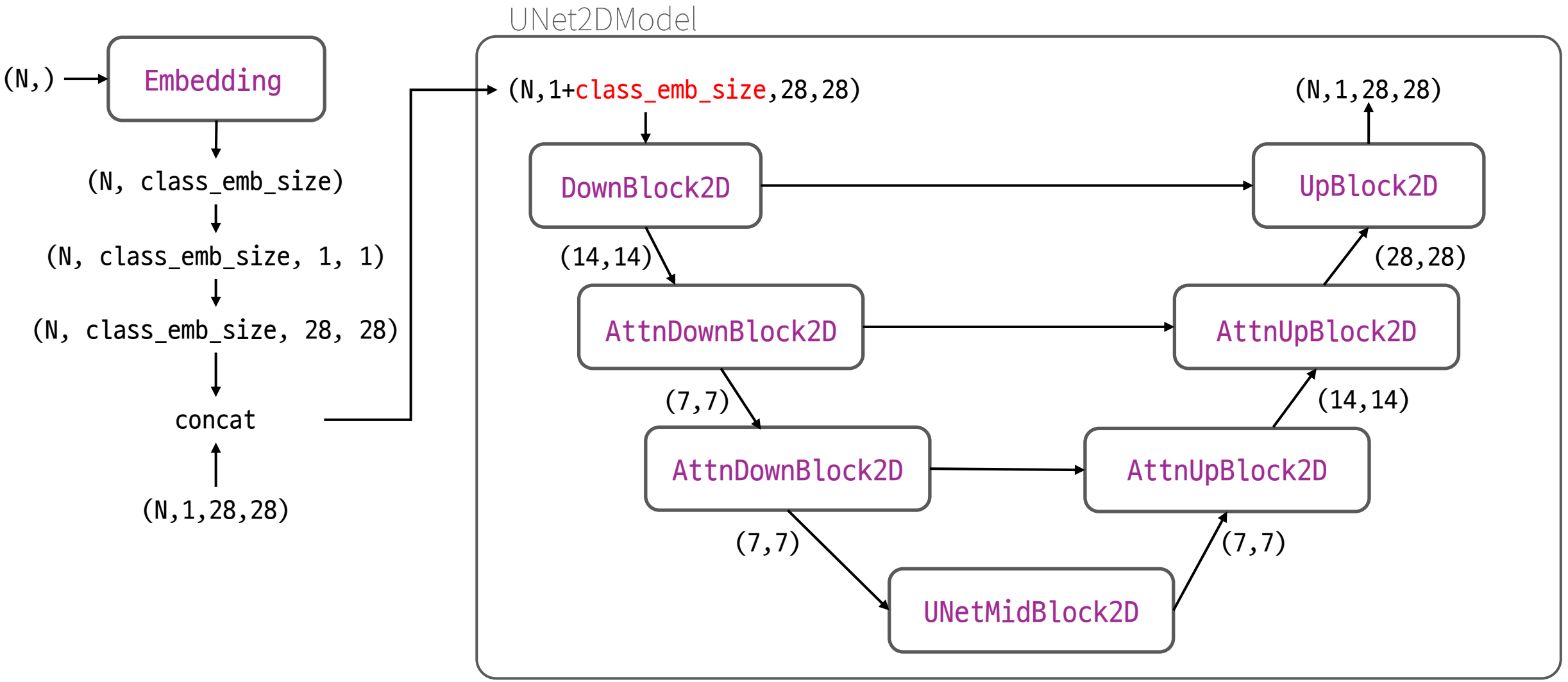
Self-Attention for 4D Feature Map



Self-Attention for 4D Feature Map

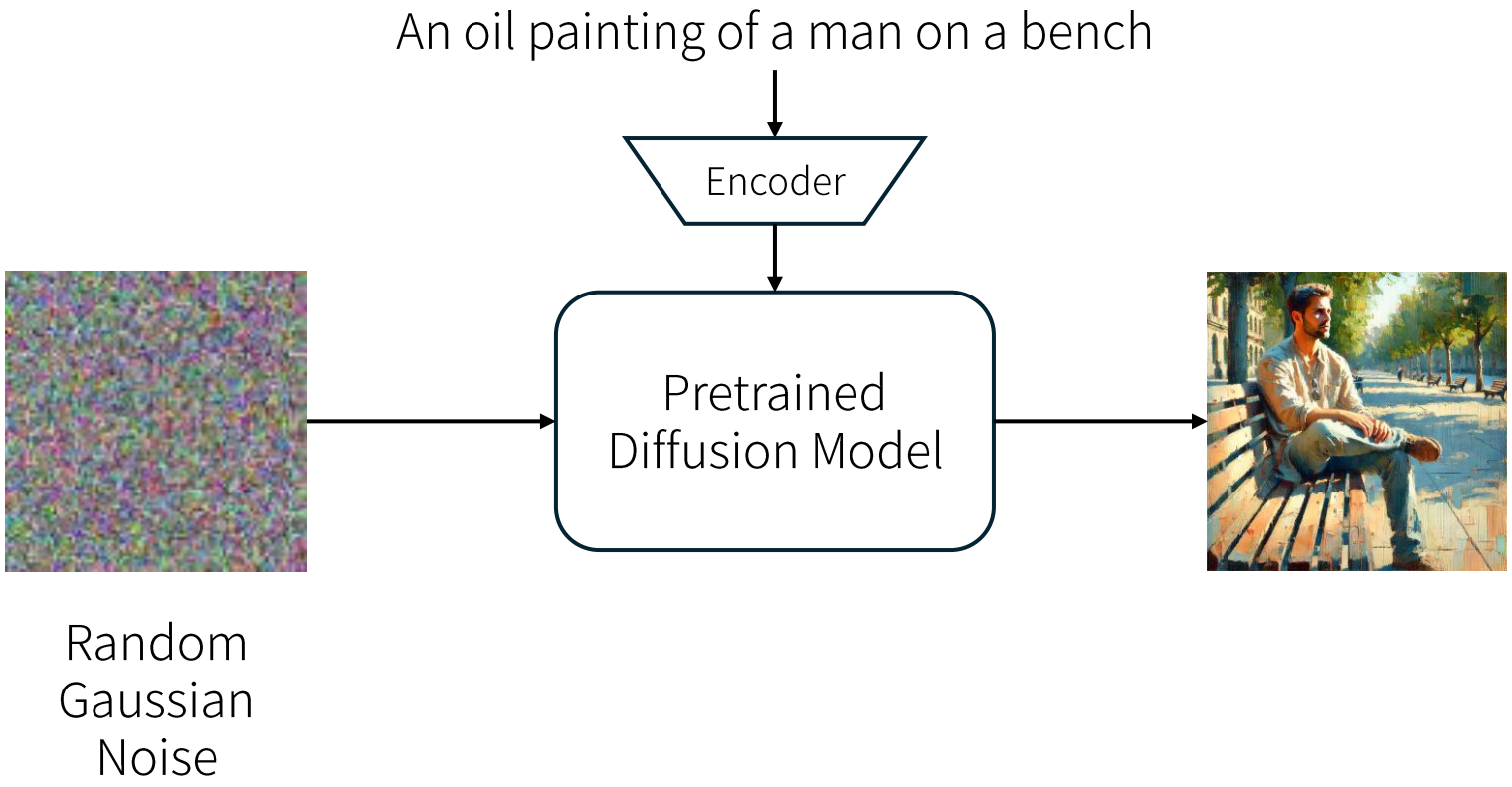


UNet2DModel (class conditioned)



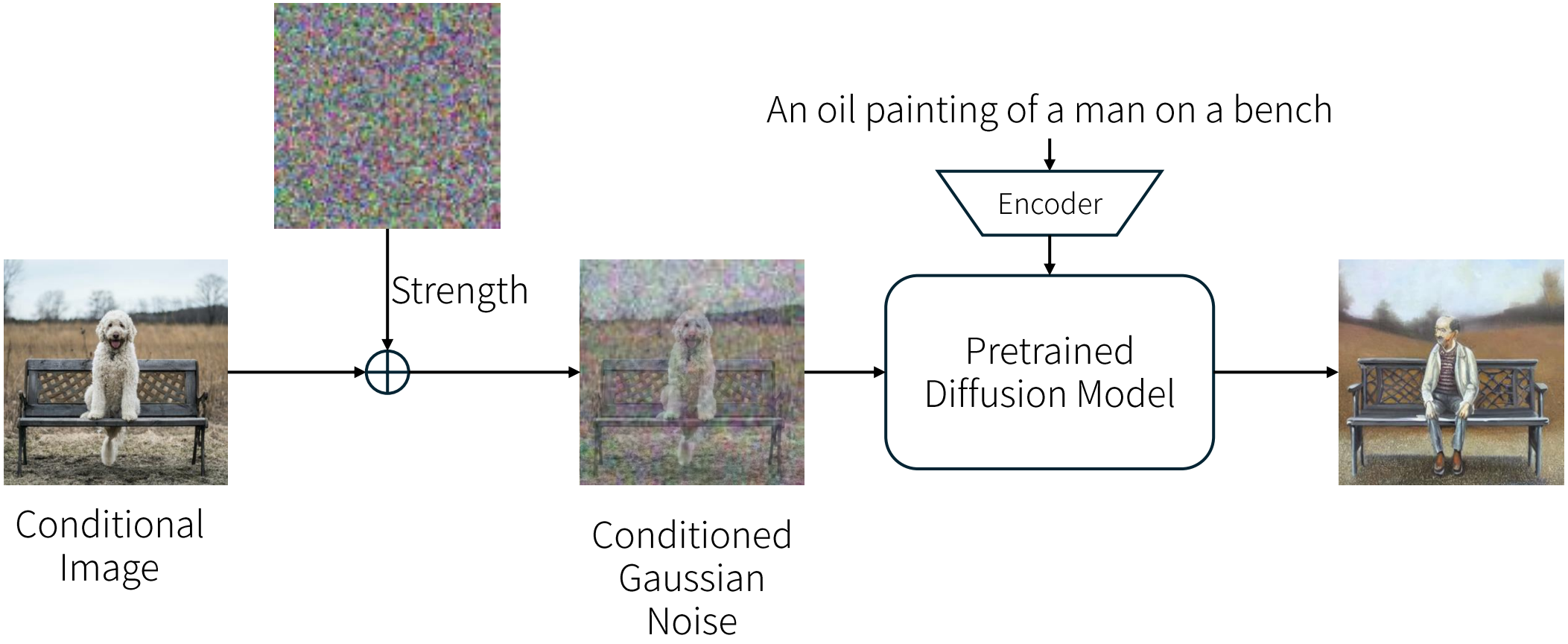
응용 태스크: Image to Image

기본 프로세스



응용 태스크: Image to Image

Image to Image



응용 태스크:
기본 모델을 이용한 Image Inpaint 생성

원본 이미지

img encoder

원본 이미지의
잠재변수

t 스탭 수준의
노이즈 추가

마스킹

x T

정규 분포
노이즈

condition

text encoder

A small robot, high resolution, sitting on a park bench

Denoising U-Net

t 스탭 노이즈
제거 잠재변수

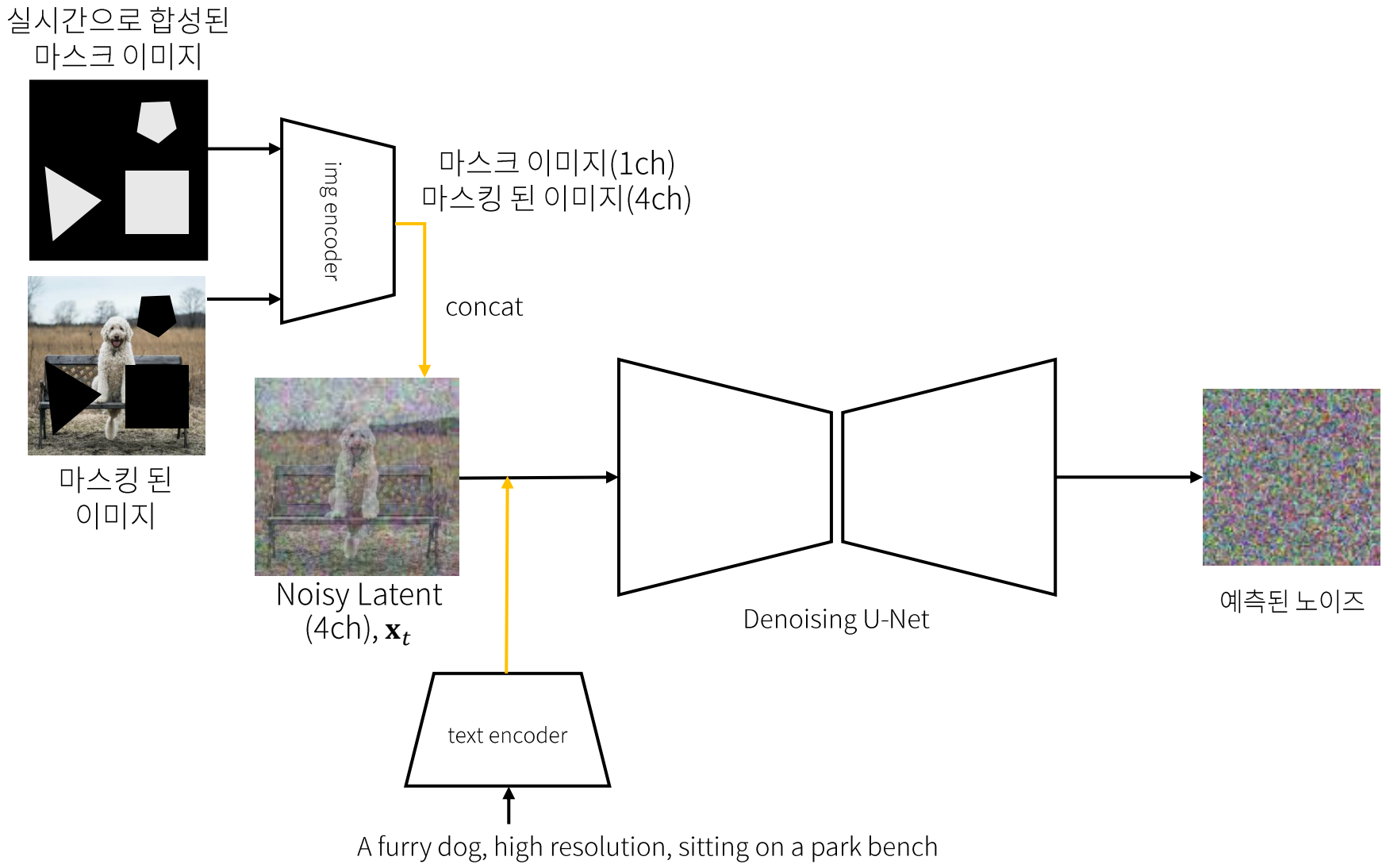
[주의] 로봇 이미지는 참고로 나타낸 것
으로 잠재변수 상에서 보이지는 않음

마스킹

노이즈 제거 완료된
잠재변수

Img decoder

응용 태스크:
Image Inpaint 기능 학습



응용 태스크:
Image Inpaint 기능학습한 모델을 이용한 생성

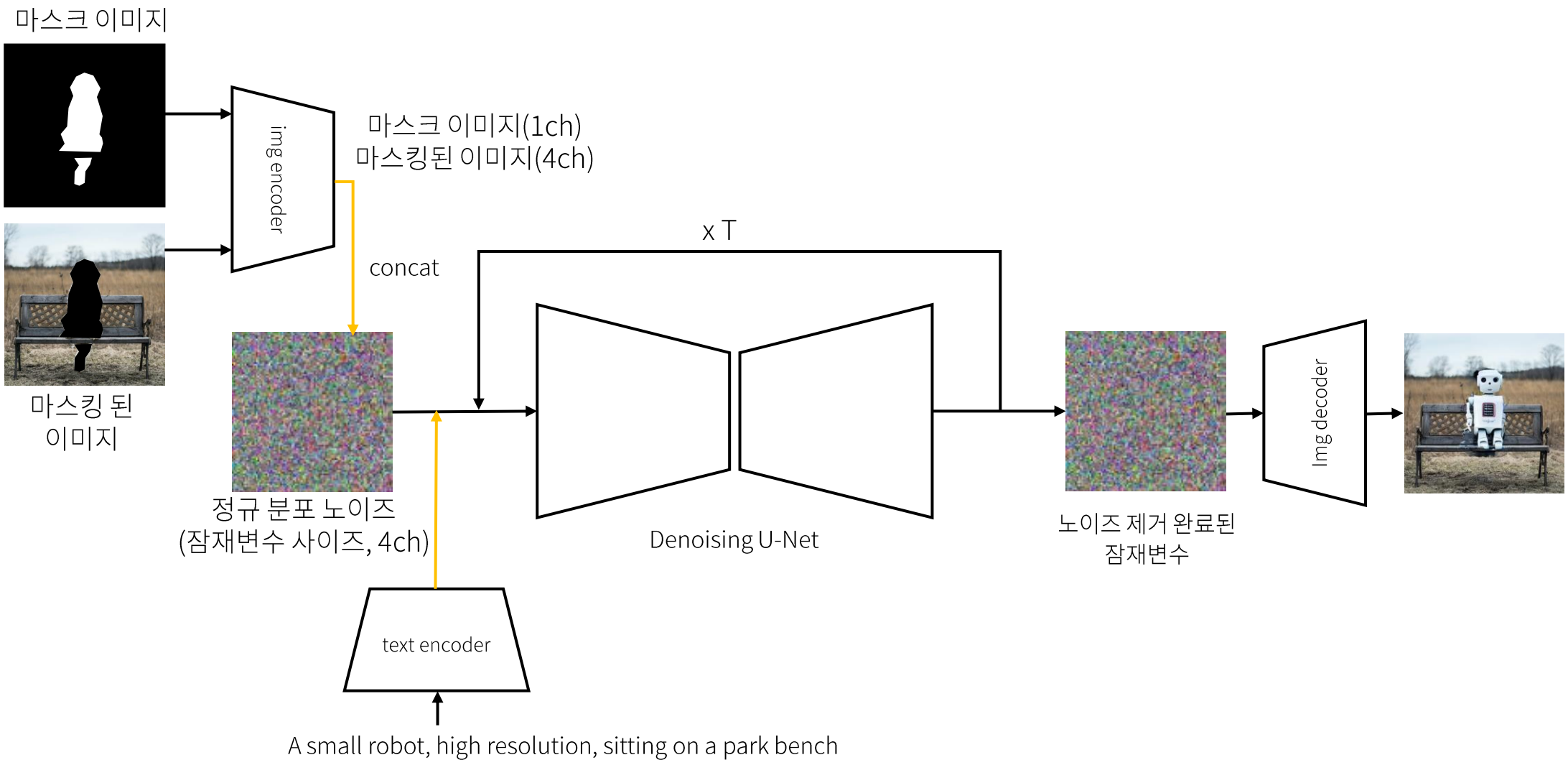
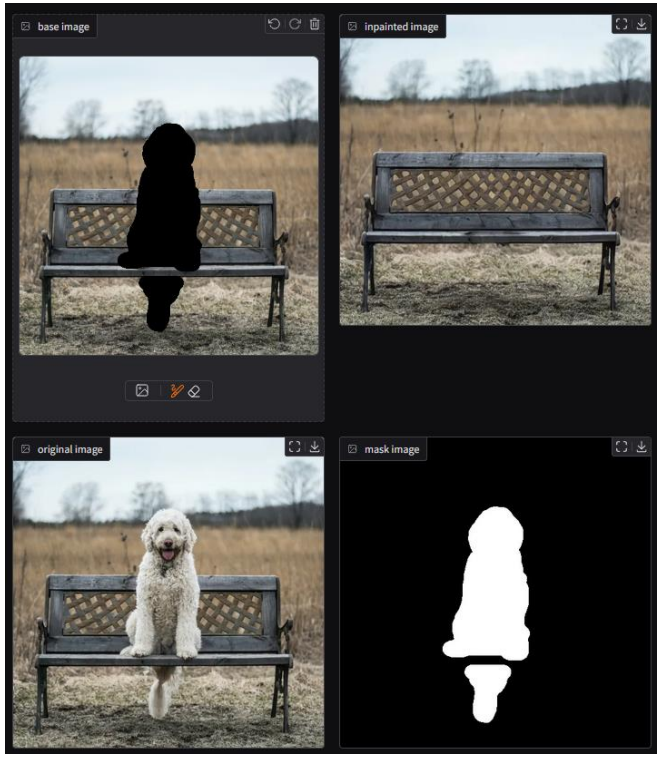
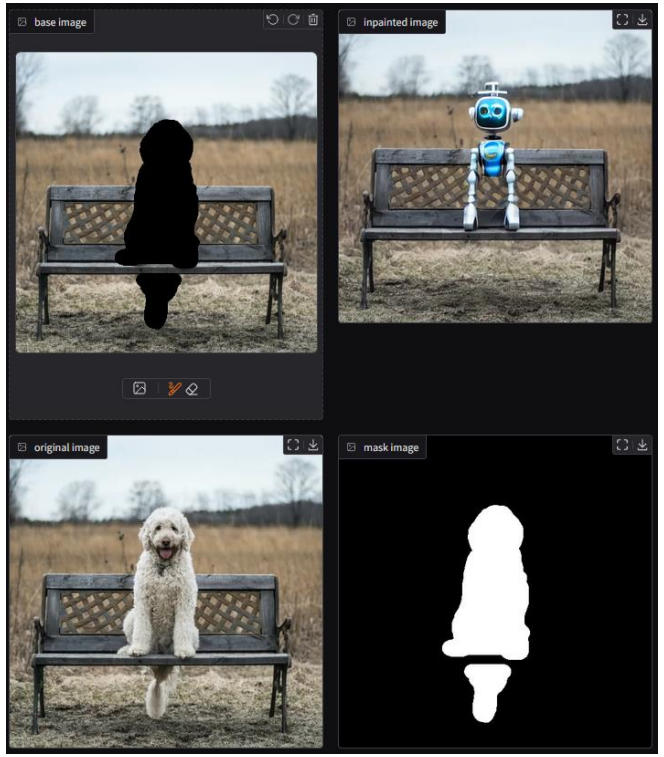


Image Inpaint Project

- 🤗 Diffusers 라이브러리와 Gradio를 이용한 Image Inpaint 앱 구현
  - Huggingface Diffusers AutoPipelineForInpainting 사용
  - 모델: stable-diffusion-inpainting



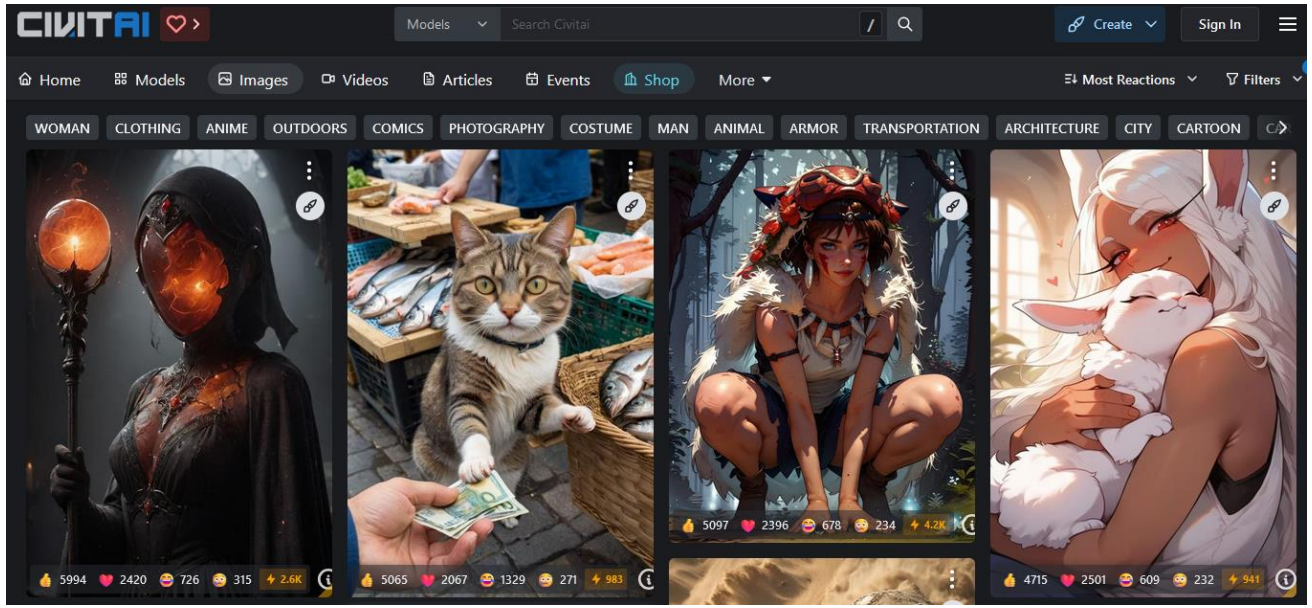
Prompt: null

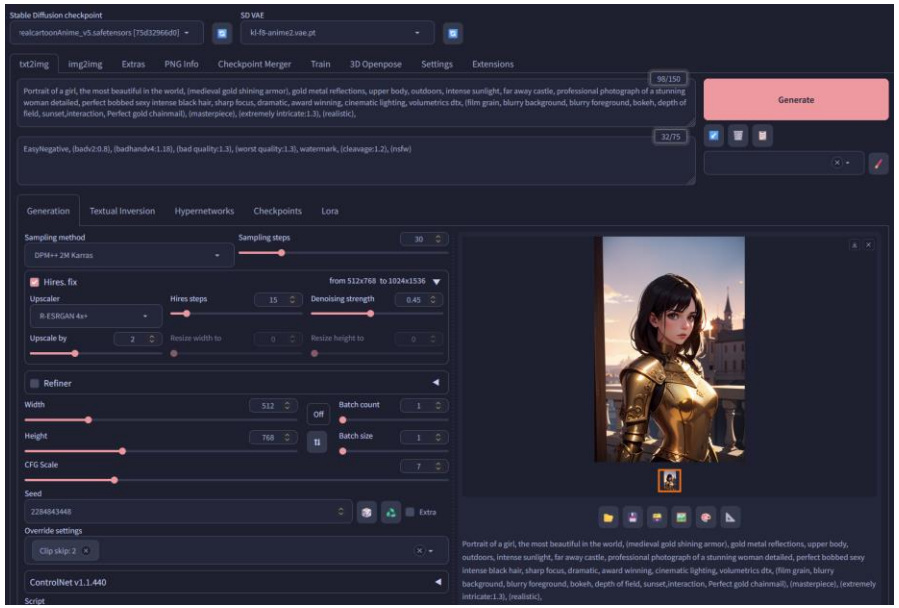


Prompt: A small robot, high resolution, sitting on a park bench

Community pipelines

- https://civitai.com 에 공유되는 체크포인트 파일을 🤗 Diffusers 라이브러리와 함께 사용하기
  - 체크포인트: CAT-Citron Anime Tresure, https://civitai.com/models/131986/cat-citron-anime-treasure-illustrious-and-sdxl-and-sd15?modelVersionId=1082124
  - VAE: PPPAnimix VAE, https://civitai.com/models/285852/pppanimix-vae
  - LoRA: Pixel Art XL, https://civitai.com/models/120096/pixel-art-xl





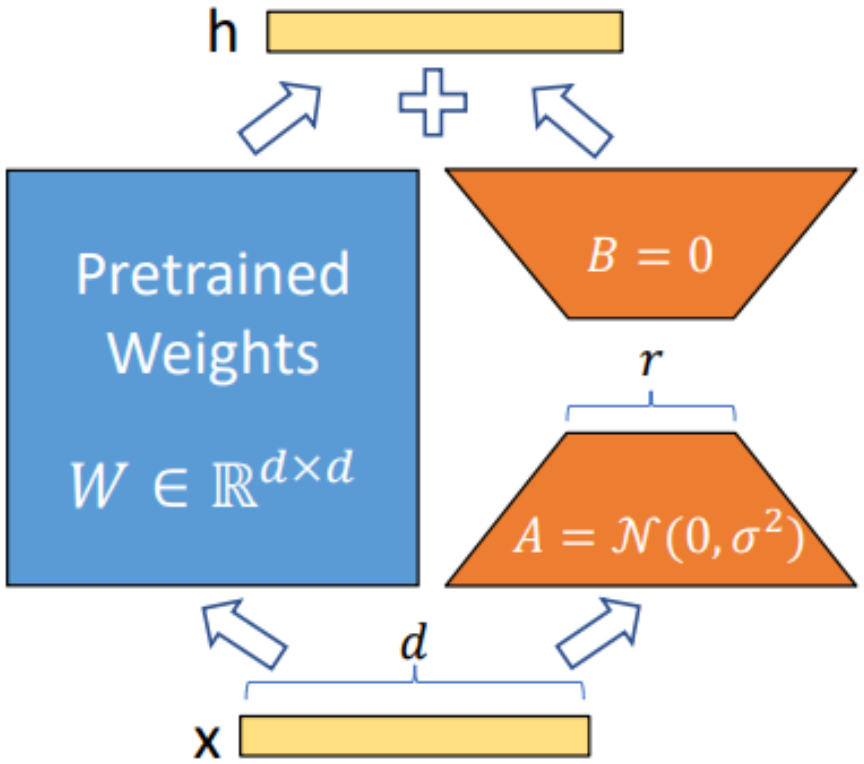
https://github.com/AUTOMATIC1111/stable-diffusion-webui

https://civitai.com/images/2471915

LoRA: Low-Rank Adaptation

미세 조정(fine-tuning) 방법으로, 기존 모델의 모든 가중치를 업데이트하지 않고 **저차원(rank reduction) 파라미터를 추가**해 효율적으로 모델을 조정하는 방식

- 모델의 주요 구조는 그대로 유지되며 추가되는 파라미터만 학습
- 메모리 및 저장 공간 절약: 기존 미세 조정 방법에 비해 훨씬 적은 메모리와 저장 공간을 사용
- 효율성: 대규모 모델에서 모든 파라미터를 업데이트하지 않으므로 훈련 속도가 향상
- 성능 유지: 원본 모델의 성능을 유지하면서 새로운 정보만 반영하여 과적합 위험 감소



Community pipelines

catCitronAnime sdxl

DPM++ Scheduler

pppanimixVAE_XL

LoRA: pixel-art-xl

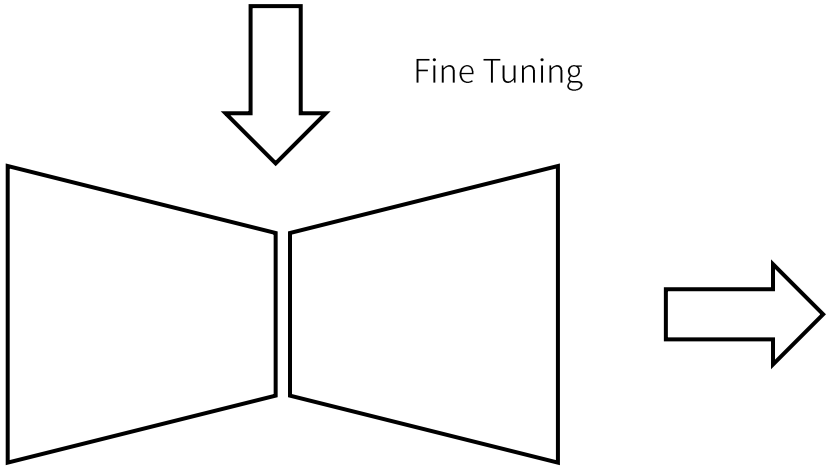
SD 1.5

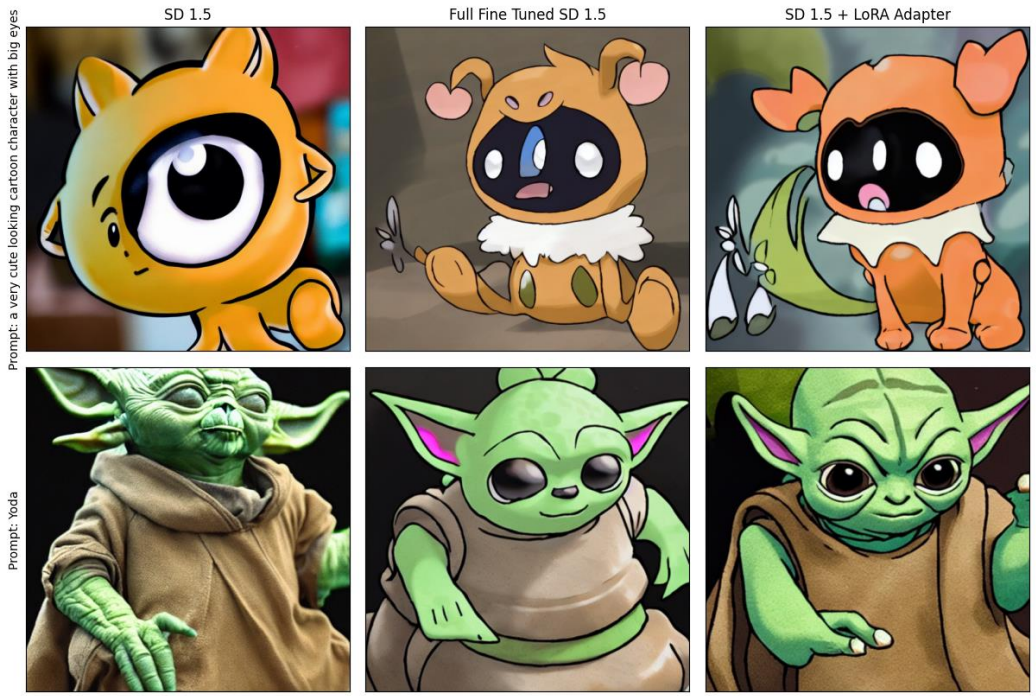
"a very cute looking cartoon character with big eyes"

"Yoda"

SD 1.5 Fine Tuning

Pokémon dataset





Using LoRA for Efficient Stable Diffusion Fine-Tuning
https://huggingface.co/blog/lora, Pedro Cuenca, Sayak Paul