A gentle tutorial on diffusion probabilistic model

Diffusion Probabilistic Model

nonezero@kumoh.ac.kr
고 재 필

Last updated: Jan. 10, 2023

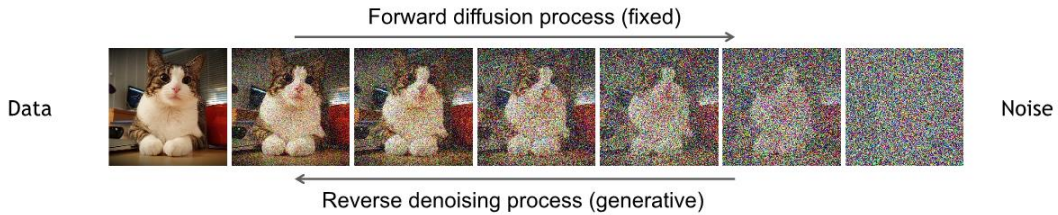
https://cvpr2022-tutorial-diffusion-models.github.io/

Denoising Diffusion Models 개요

Learning to generate by denoising

Denoising diffusion models consist of two processes:

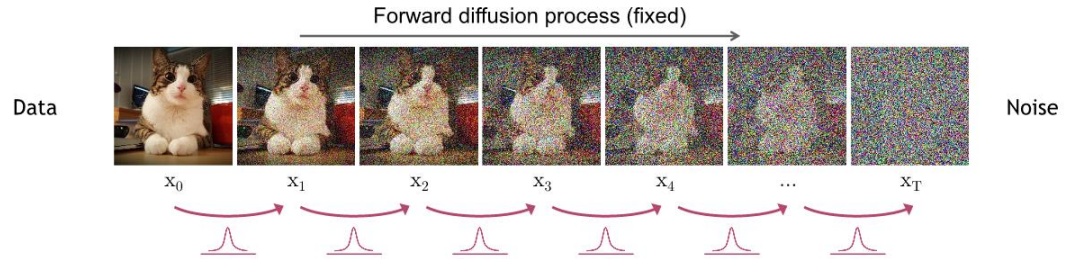
- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising



Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Forward Diffusion Process Forward

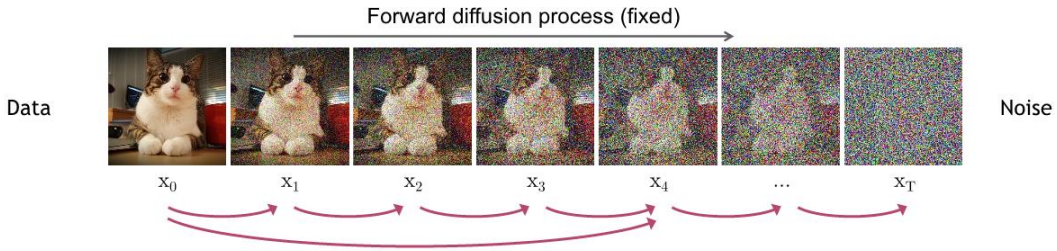
The formal definition of the forward process in T steps:



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

Diffusion Kernel Forward

x_0 로 바로 계산



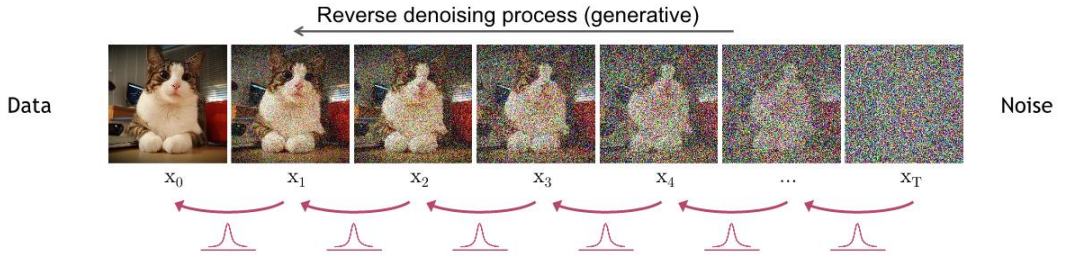
Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

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

β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

Reverse Denoising Process Reverse

Formal definition of forward and reverse processes in T steps:



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t), \sigma_t^2}_{\text{Trainable network}}\mathbf{I}) \quad \Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

(U-net, Denoising Autoencoder)

Learning Denoising Model 훈련 목적식

Variational upper bound

For training, we can form variational upper bound that is commonly used for training variational autoencoders:

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

Sohl-Dickstein et al. ICML 2015 and Ho et al. NeurIPS 2020 show that:

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

where $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ is the tractable posterior distribution:

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

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

24

Parameterizing the Denoising Model

목적식 구체화: 파라미터릭 표현

Since both $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ are Normal distributions, the KL divergence has a simple form:

$$L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}||\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) - \mu_\theta(\mathbf{x}_t,t)||^2\right] + C$$

Recall that $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$. Ho et al. NeurIPS 2020 observe that:

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

They propose to represent the mean of the denoising model using a *noise-prediction* network:

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

With this parameterization

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon}_{\mathbf{x}_t},t)||^2\right] + C$$

25

Training Objective Weighting 훈련 잘 되도록

Trading likelihood for perceptual quality

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\underbrace{\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}}_{\lambda_t}||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,t)||^2\right]$$

The time dependent λ_t ensures that the training objective is weighted properly for the maximum data likelihood training.

However, this weight is often very large for small t's.

Ho et al. NeurIPS 2020 observe that simply setting $\lambda_t = 1$ improves sample quality. So, they propose to use:

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(1,T)}\left[||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon}_{\mathbf{x}_t},t)||^2\right]$$

For more advanced weighting see Choi et al., Perception Prioritized Training of Diffusion Models, CVPR 2022.

26

Summary

Training and Sample Generation

훈련, 생성 알고리즘

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: $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
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** $\mathbf{x}_0$

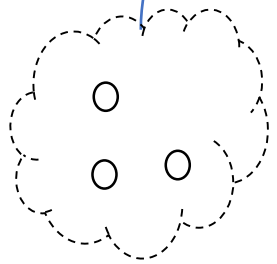
27

선수지식

$x \sim p\,(x)$

generation

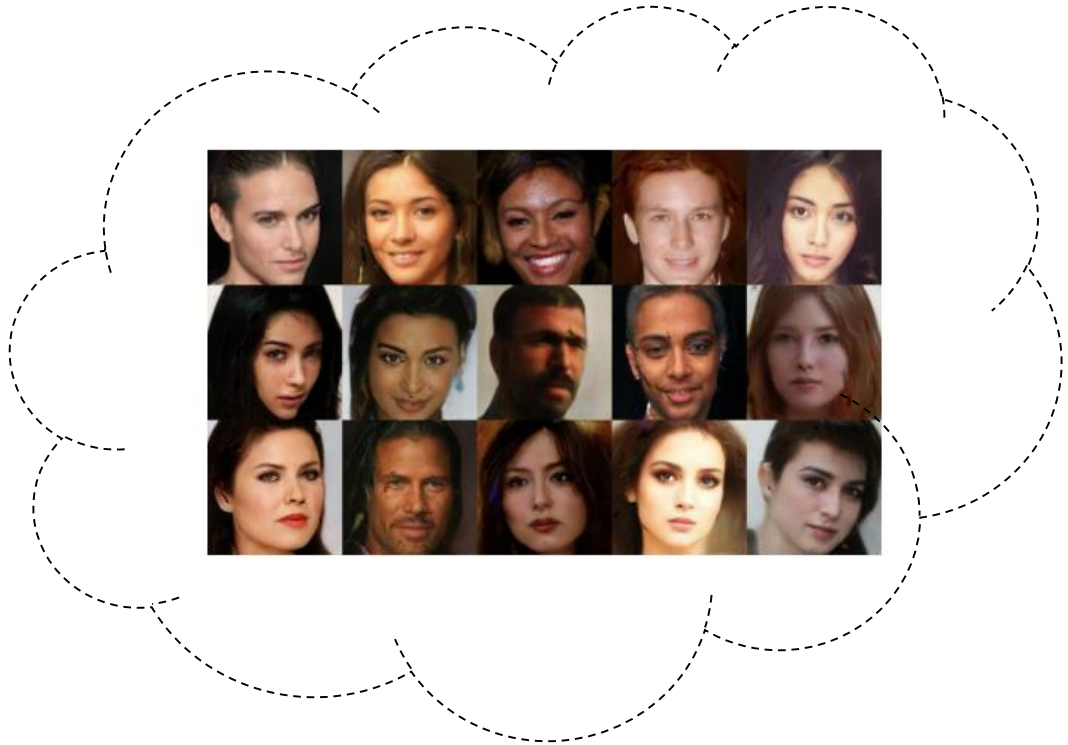
sampling



$p(x)$

분포함수

distribution



$p(x) \leftrightarrow p_X(x) \leftrightarrow p(X = x)$

random variable

information $-log^{p(x)}$

expectation $E[X] = \sum_x xp(x)$

$$E[f(X)] = \sum_x f(x)p(x)$$

entropy $E\left[-log^{p(X)}\right] = \sum_x -log^{p(x)}p(x)$

cross-entropy $E_{q(x)}\left[-log^{p(X)}\right] = \sum_x -log^{p(x)}q(x)$

Bayes' rule $p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(y)p(x|y)}{\sum_y p(y)p(x|y)}$

$$p(x,y) = p(x)p(y|x) = p(y)p(x|y)$$

$$p(x,y,z) = p(x)p(y|x)p(z|x,y)$$

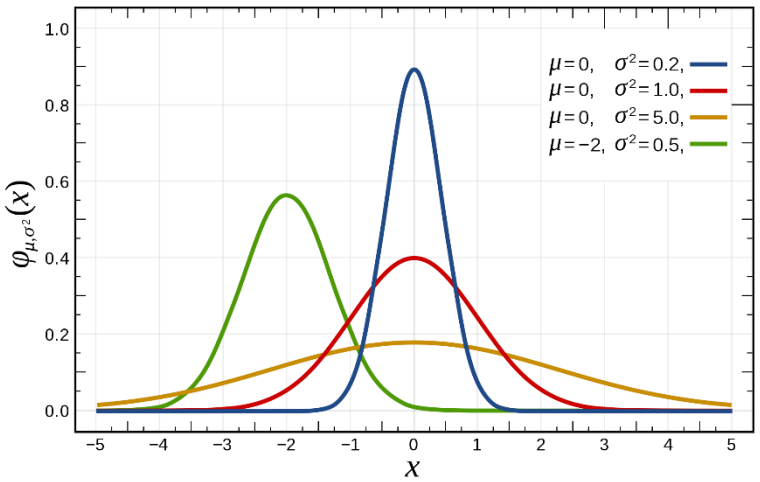
$$p(x) = \sum_y p(x,y) = \sum_y p(y)p(x|y)$$

KL divergence

$$KL(p_1(x)||p_2(x))$$
$$= -\int p_1(x)logp_2(x)dx - \left(-\int p_1(x)logp_1(x)dx\right)$$
$$= -\int p_1(x)log\frac{p_2(x)}{p_1(x)}dx$$
$$= E_{p_1(x)}\left[-log\frac{p_2(x)}{p_1(x)}\right]$$

random process $x_1 \rightarrow x_2 \rightarrow x_3$

Markov random process $p(x_3|x_2,x_1) = p(x_3|x_2)$



$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}((x-\mu)/\sigma)^2}$$

$p(x; u, \sigma)$

$p_\theta(x) \quad , \theta = (u, \sigma)$

reparameterization trick

$$p(x) = N(u, \sigma^2)$$

$$x \sim p(x)$$

$$x = u + \sigma\epsilon, \epsilon \sim N(0, 1)$$

KL divergence between two Gaussian distributions

$$KL(p, q) = \frac{1}{2}\log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

$$p(x) = N(\mu_1, \sigma_1^2)$$

$$q(x) = N(\mu_2, \sigma_2^2)$$

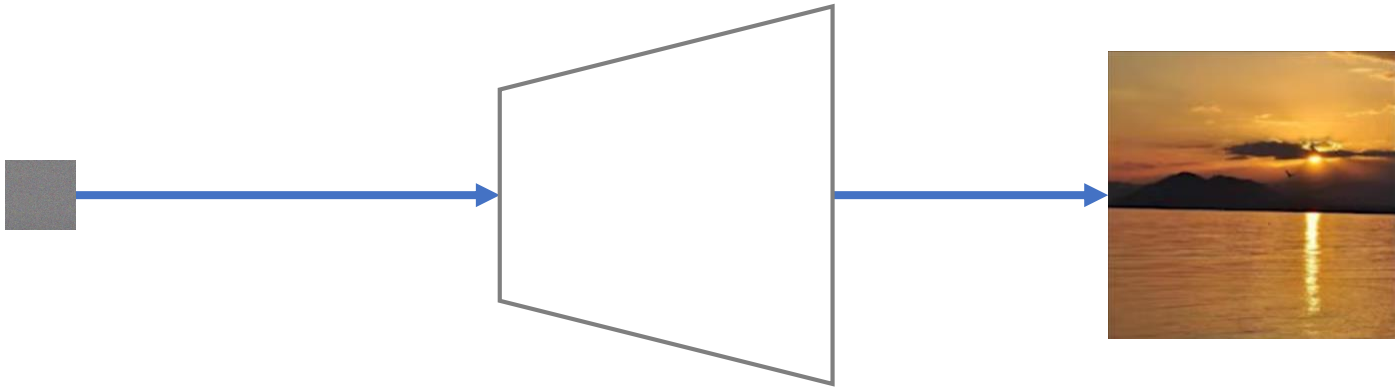
$$N(\mu_1, \sigma_1^2) + N(\mu_2, \sigma_2^2) = N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$

직관으로 이해하기

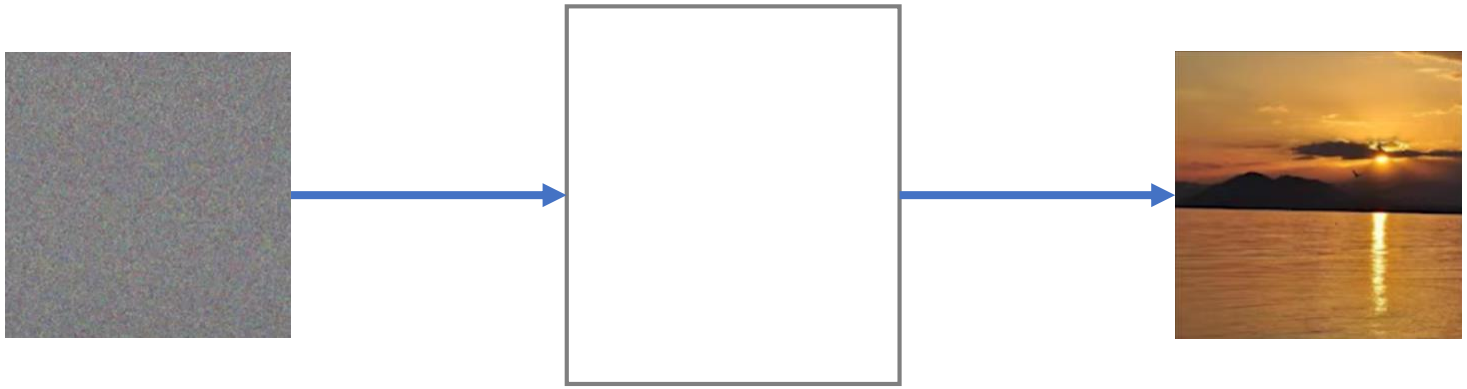
신경망은,

신경망구조 + 손실함수

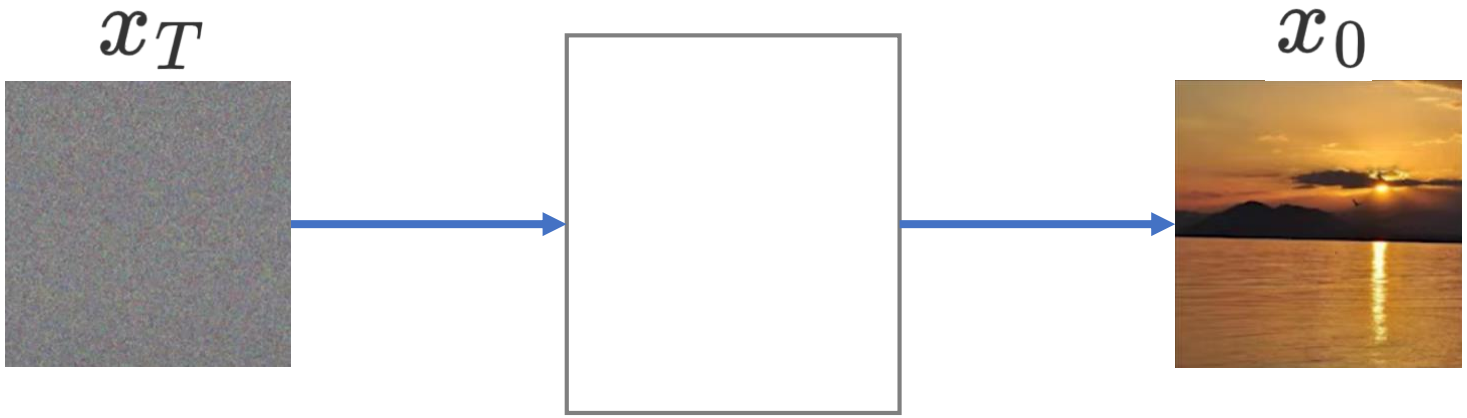
Conventional Generative Model



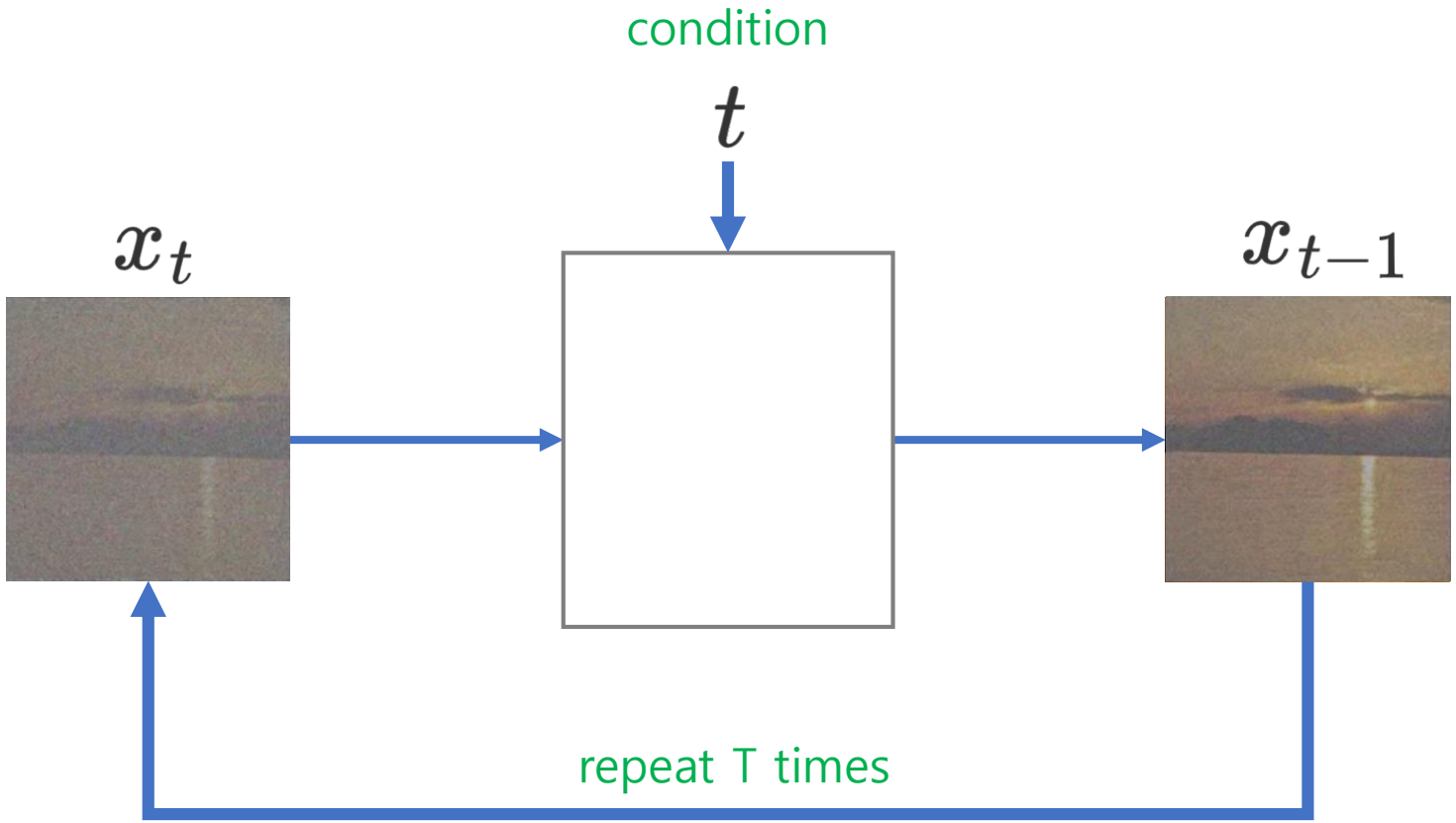
Diffusion Probabilistic Model



출력크기와 같다.



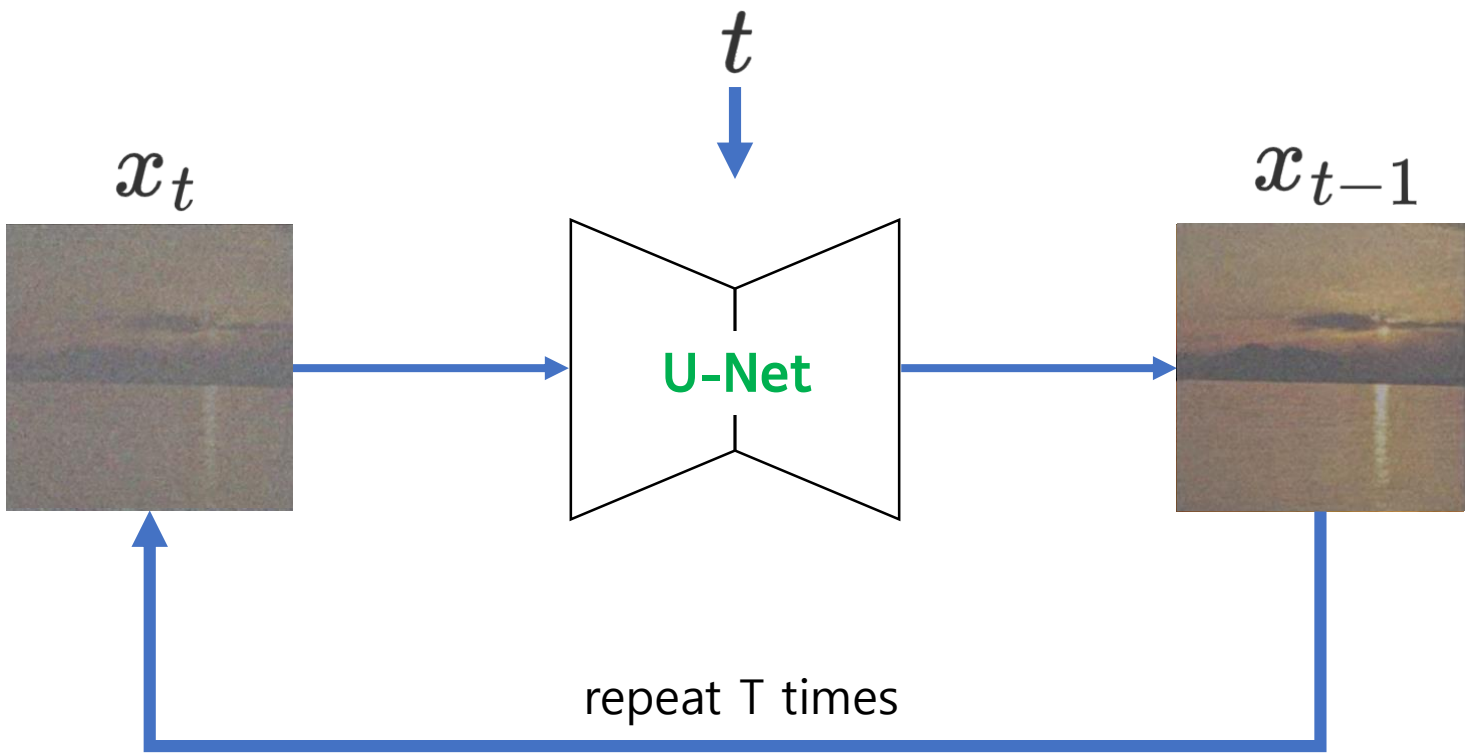
바로 변환은 어렵다.


condition
t
x_t
x_{t-1}
repeat T times

DPM 신경망구조

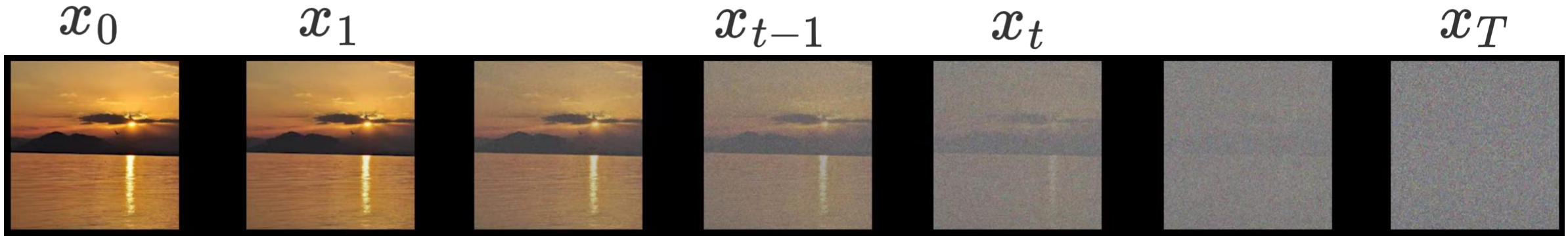
U-Net이 좋다.
(Pix2Pix)

- ✓ Detail을 살려야 한다.
- ✓ Scene의 구조가 바뀌지 않는다.



신경망 훈련

가우시안 노이즈를 점진적으로 추가 (영상처리)

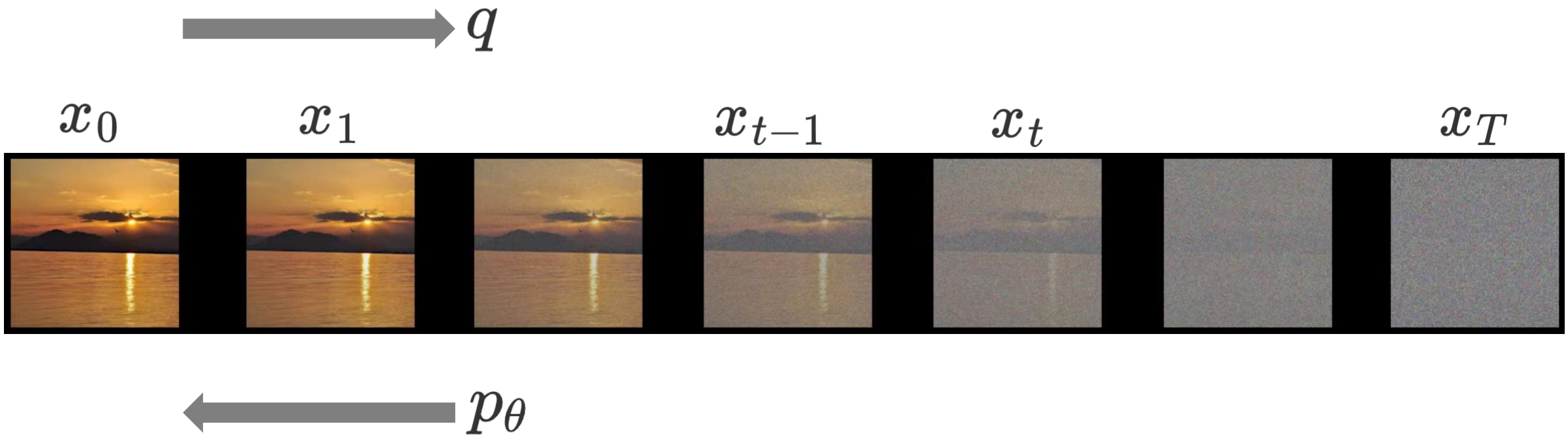


$T = 1{,}000$

신경망

Image credits:
https://www.youtube.com/watch?v=HoKDTa5jHvg

분포함수로 표현



분포함수로 표현

q

$q(x_T)$

x_0 x_1 x_{t-1} x_t x_T

p_θ

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

x_0 x_1 x_{t-1} x_t x_T

p_θ $p_\theta(x_{t-1}|x_t)$ $p_\theta(x_T)$

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

x_0 x_1 x_{t-1} x_t x_T

p_θ $p_\theta(x_{t-1}|x_t)$ $p(x_T)$

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

x_0 x_1 x_{t-1} x_t x_T

p_θ $p_\theta(x_{t-1}|x_t)$ $p(x_T)$

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

tractable ?

output

input

q

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

$q(x_T)$

x_0 x_1 x_{t-1} x_t x_T

p_θ

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

$p(x_T)$

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

q $q(x_t|x_{t-1}, x_0)$ $q(x_T|x_0)$

x_0 x_1 x_{t-1} x_t x_T

p_θ $p_\theta(x_{t-1}|x_t)$ $p(x_T)$

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

Loss L_T

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

$$q(x_T|x_0)$$

x_0 x_1 x_{t-1} x_t x_T

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

$$p(x_T)$$

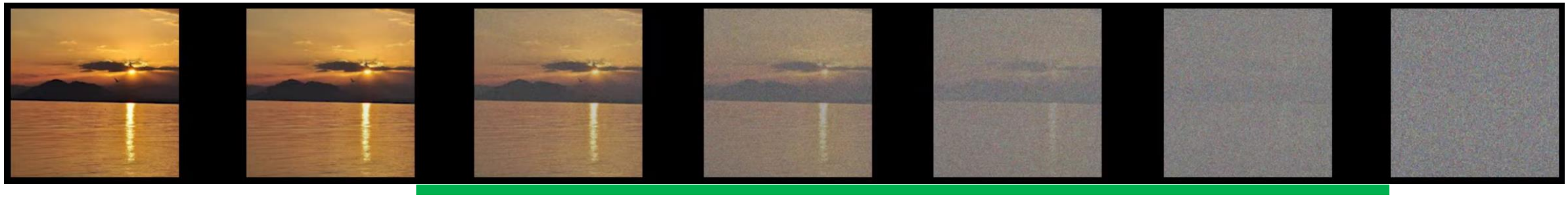
$$q(x_{t-1}|x_t, x_0)$$

$$D_{KL}(q(x_T|x_0)||p(x_T)$$

Loss L_{t-1}

$q(x_t|x_{t-1}, x_0)$ $q(x_T|x_0)$

x_0 x_1 x_{t-1} x_t x_T



$p_\theta(x_{t-1}|x_t)$ $p(x_T)$

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

$$\sum_{t=2}^{T} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$$

Loss L_0

$q(x_t|x_{t-1}, x_0)$ $q(x_T|x_0)$

x_0 x_1 x_{t-1} x_t x_T

$p_\theta(x_{t-1}|x_t)$ $p(x_T)$

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

$$-log p_\theta(x_0|x_1)$$

$E_q[-log p_\theta(x_0|x_1)]$

cross-entropy / mse

DPM 손실함수

$$E_{q(x_0)}[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

regularization **reconstruction**

Forward Diffusion Process

Bayes' rule: $p(x, y) =$

Bayes' rule: $p(x, y) = p(x)p(y|x)$

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$p(x, y, z) =$

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$$p(x, y, z) = p(x)p(y|x)p(z|y, x)$$

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$$p(x, y, z) = p(x)p(y|x)p(z|y, x)$$

$x_1 \rightarrow x_2 \rightarrow x_3$ $p(x_3|x_2, x_1) =$

Process

$$q(x_{1:T}|x_0) = \prod_1^T q(x_t|x_{t-1})$$

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$$p(x, y, z) = p(x)p(y|x)p(z|y, x)$$

$x_1 \rightarrow x_2 \rightarrow x_3$ $p(x_3|x_2, x_1) = p(x_3|x_2)$

x_1의 정보가 반영되어 있다.

first order **Markov Process (chain)**

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$$p(x, y, z) = p(x)p(y|x)p(z|y, x)$$

$x_1 \rightarrow x_2 \rightarrow x_3$ $\quad p(x_3|x_2, x_1) = p(x_3|x_2)$

Markov Process (chain)

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)p(x_4|x_3, x_2, x_1)$$

$$p(x_1, x_2, x_3, x_4) =$$

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$$p(x, y, z) = p(x)p(y|x)p(z|y, x)$$

$x_1 \rightarrow x_2 \rightarrow x_3$ $\quad p(x_3|x_2, x_1) = p(x_3|x_2)$

Markov Process (chain)

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)p(x_4|x_3, x_2, x_1)$$

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3)$$

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$$p(x, y, z) = p(x)p(y|x)p(z|y, x)$$

$x_1 \rightarrow x_2 \rightarrow x_3$ $\quad p(x_3|x_2, x_1) = p(x_3|x_2)$

Markov Process (chain)

$$q(x_1, x_2, \ldots, x_T|x_0) = q(x_1|x_0)q(x_2|x_1, x_0)q(x_3|x_2, x_1, x_0) \ldots q(x_T|x_{T-1}, \ldots, x_0)$$

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

Bayes' rule: $p(x, y) = p(x)p(y|x)$

$$p(x, y, z) = p(x)p(y|x)p(z|y, x)$$

$x_1 \rightarrow x_2 \rightarrow x_3$ $\quad p(x_3|x_2, x_1) = p(x_3|x_2)$

Markov Process (chain)

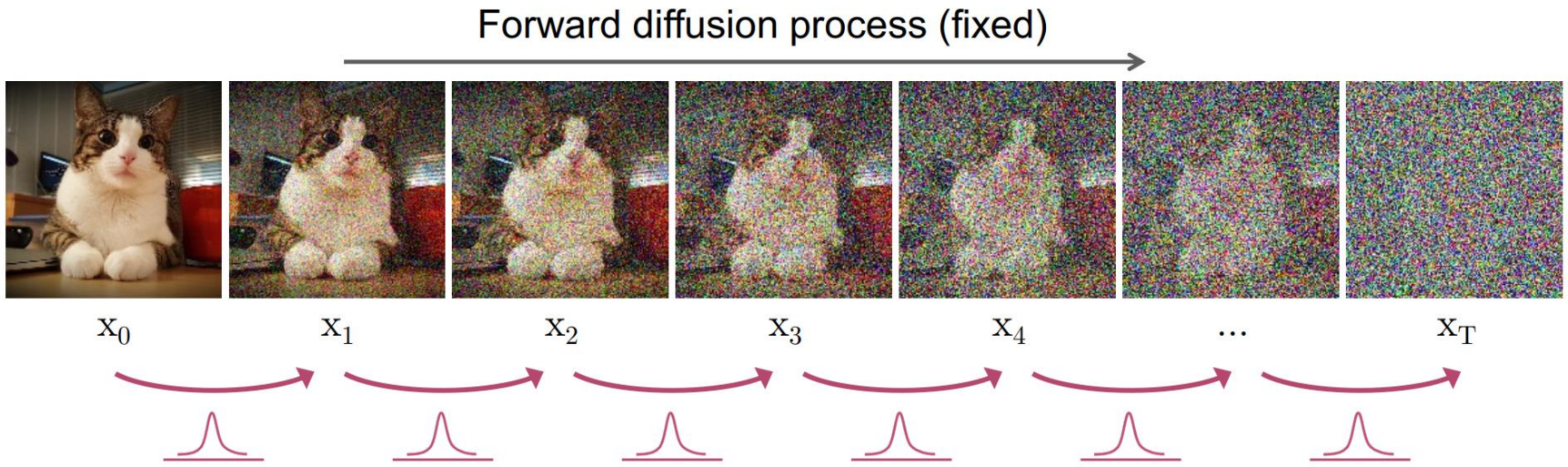
$$q(x_1, x_2, \ldots, x_T|x_0) = q(x_1|x_0)q(x_2|x_1, x_0)q(x_3|x_2, x_1, x_0) \ldots q(x_T|x_{T-1}, \ldots, x_0)$$

$$q(x_1, x_2, \ldots, x_T|x_0) = q(x_1|x_0)q(x_2|x_1)q(x_3|x_2) \ldots q(x_T|x_{T-1})$$

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

Forward Diffusion Process

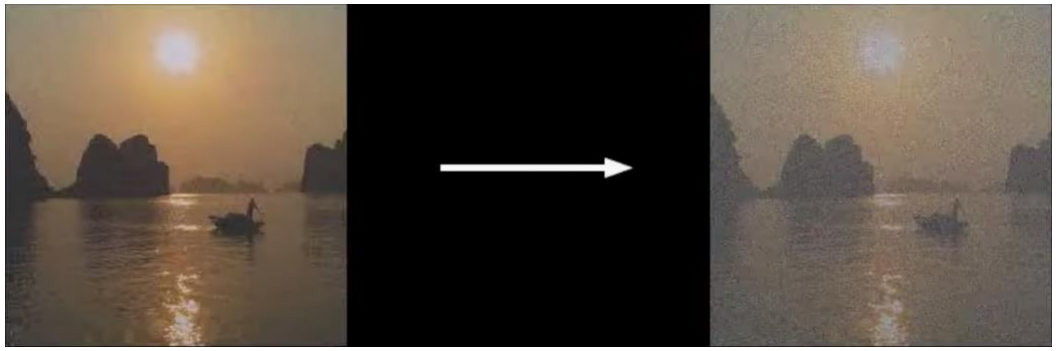
The formal definition of the forward process in T steps:

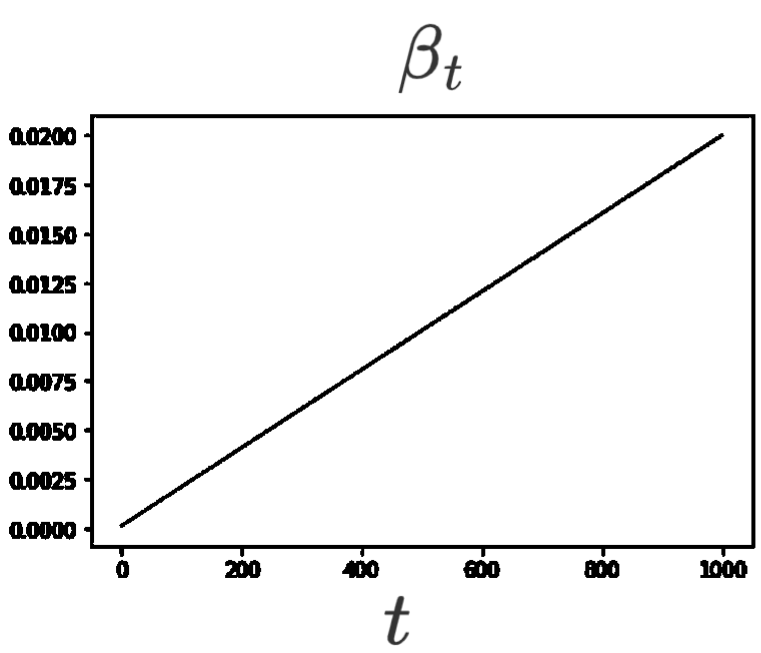


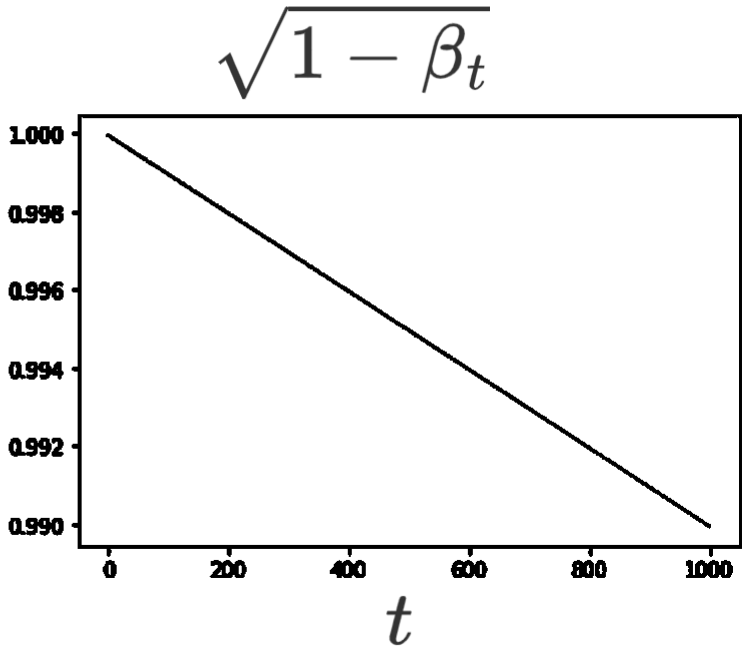
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

Add Gaussian Noise

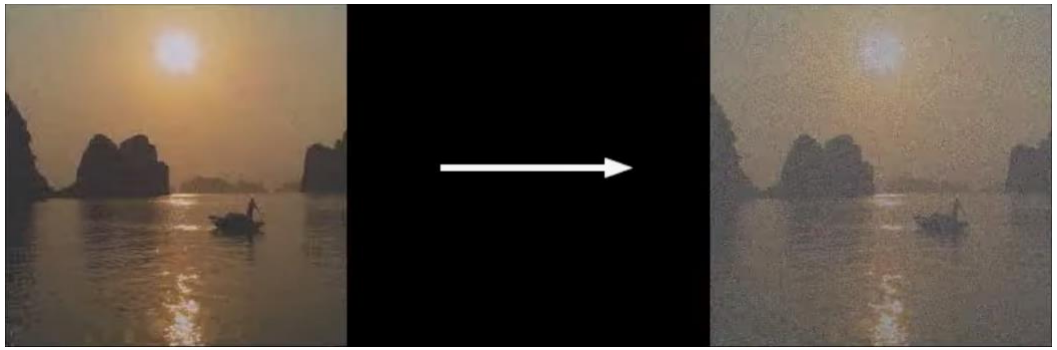
x_{t-1} x_t




β_t
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
200
400
600
800
1000
t

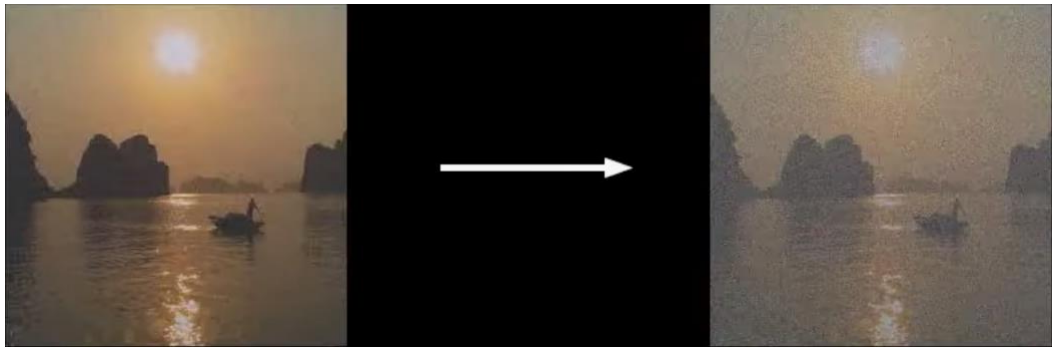

$\sqrt{1-\beta_t}$
1.000
0.998
0.996
0.994
0.992
0.990
0
200
400
600
800
1000
t

x_{t-1} x_t



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

x_{t-1} x_t



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

이전 픽셀 값을 거의 유지

약간의 변형

$$Var(x_t) = Var\left(\sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon\right)$$

$$= Var\left(\sqrt{1-\beta_t}x_{t-1}\right) + Var\left(\sqrt{\beta_t}\epsilon\right) \qquad x_{t-1} \quad \epsilon \quad \text{독립}$$

$$= (1-\beta_t)Var\left(x_{t-1}\right) + \beta_t Var\left(\epsilon\right)$$

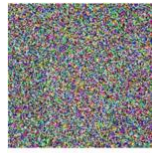
$$= 1 - \beta_t + \beta_t$$

$$= 1$$

$Var(x_T) = 1$

$Var(x_t) = 1$

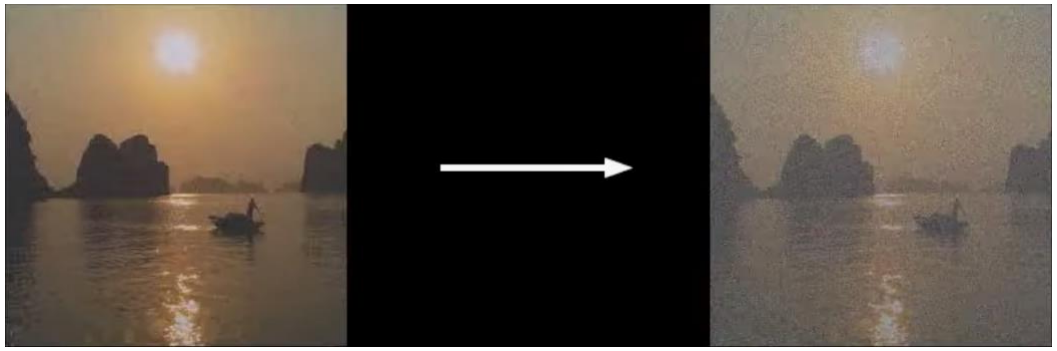
$Var(x_{t-1}) = 1$



x_T

같은 식으로 x_T부터 쭉 이어져 왔다면...

x_{t-1} x_t



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

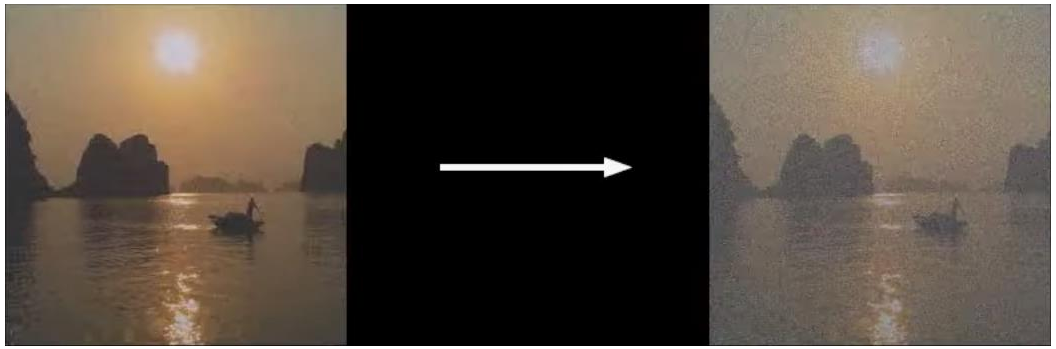
이전 픽셀 값을 거의 유지

약간의 변형

$$x_t \sim N(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

Reparameterization trick

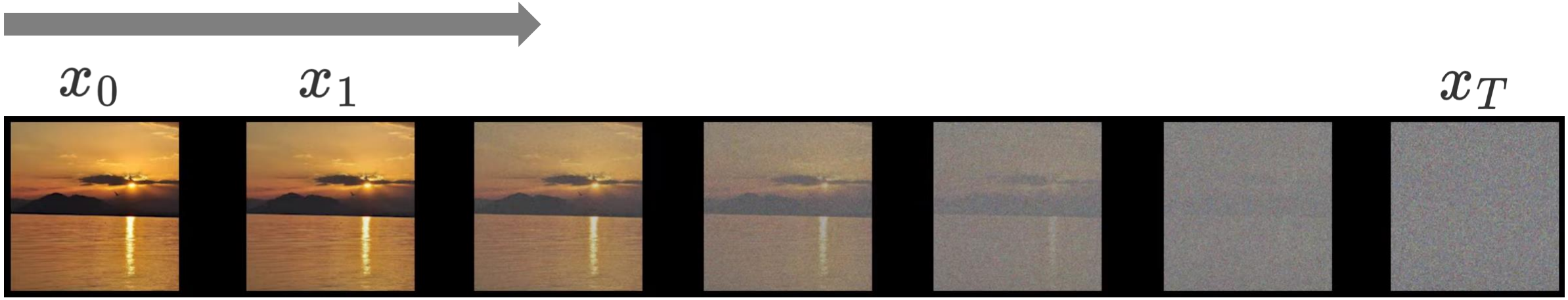
$$q(x_t|x_{t-1})$$

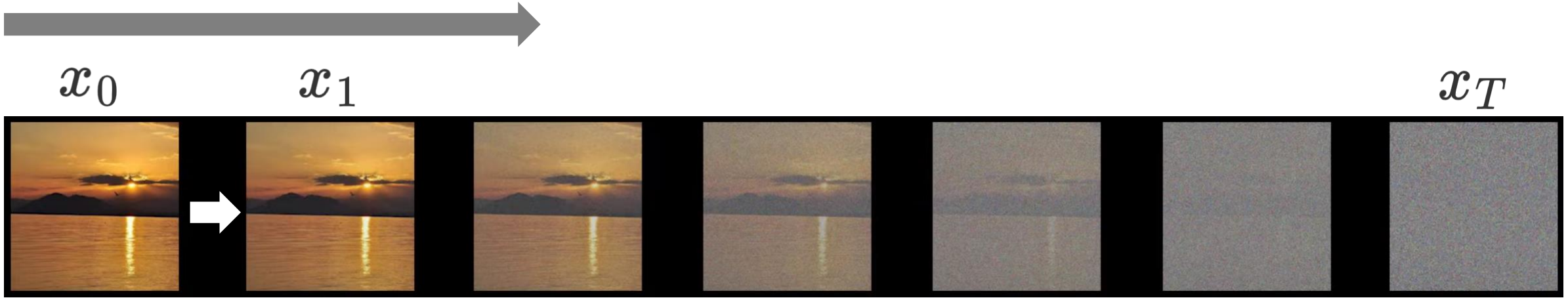


x_{t-1} x_t

$$\Leftrightarrow N(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

Forward Process

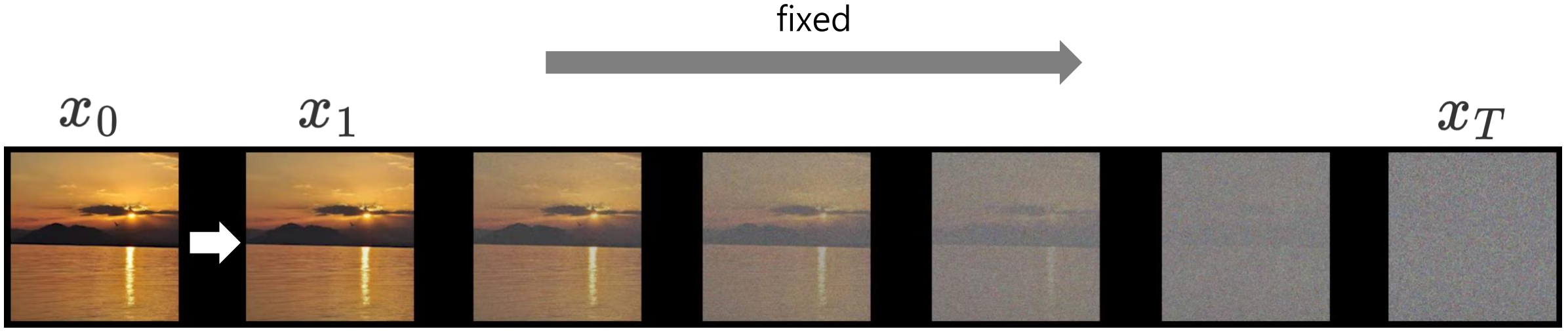



x_0
x_1
x_T

$q(x_1, x_2, \ldots, x_T | x_0)$

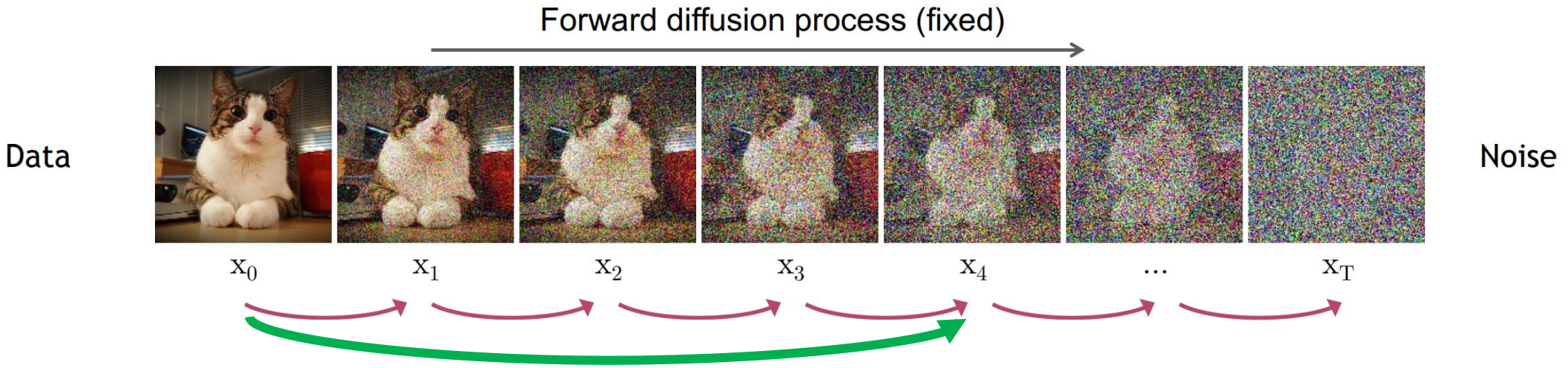
x_0 x_1 x_T

$q(x_{1:T} | x_0)$



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

Diffusion Kernel



Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

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

β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

$$\alpha_t = 1 - \beta_t$$

$$\bar{\alpha}_t = \prod_{s=1}^{t} a_s$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon$$

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \epsilon$$

$$= \sqrt{\alpha_t\alpha_{t-1}}\, x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\, \epsilon$$

$$= \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\, x_{t-3} + \sqrt{1-\alpha_t\alpha_{t-1}\alpha_{t-2}}\, \epsilon$$

$$= \sqrt{\alpha_t\alpha_{t-1}\ldots\alpha_1\alpha_0}\, x_0 + \sqrt{1-\alpha_t\alpha_{t-1}\ldots\alpha_1\alpha_0}\, \epsilon$$

x_t의 분포(함수) 표현

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

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

이렇게 유도될 것 같기도 하고...

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

x_t의 함수표현 $x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$

$$x_1 = \sqrt{\alpha_1} x_0 + \sqrt{1-\alpha_1}\epsilon_1$$

$$x_2 = \sqrt{\alpha_2} x_1 + \sqrt{1-\alpha_2}\epsilon_2$$

$$= \sqrt{\alpha_2}(\sqrt{\alpha_1} x_0 + \sqrt{1-\alpha_1}\epsilon_1) + \sqrt{1-\alpha_2}\epsilon_2$$

$$= \sqrt{\alpha_2\alpha_1} x_0 + \sqrt{\alpha_2(1-\alpha_1)}\epsilon_1 + \sqrt{1-\alpha_2}\epsilon_2$$

$$= \sqrt{\alpha_2\alpha_1} x_0 + \sqrt{\alpha_2(1-\alpha_1) + (1-\alpha_2)}\epsilon$$

$$= \sqrt{\alpha_2\alpha_1} x_0 + \sqrt{\alpha_2 - \alpha_2\alpha_1 + 1 - \alpha_2)}\epsilon$$

$$= \sqrt{\alpha_2\alpha_1} x_0 + \sqrt{1-\alpha_2\alpha_1}\epsilon$$

$$N(\mu_1, \sigma_1^2) + N(\mu_2, \sigma_2^2) = N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$

$$\epsilon_1 = \epsilon_2 = \ldots \epsilon = N(0, I)$$

Reverse Diffusion Process

$q(x_1, x_2, \ldots, x_T | x_0)$

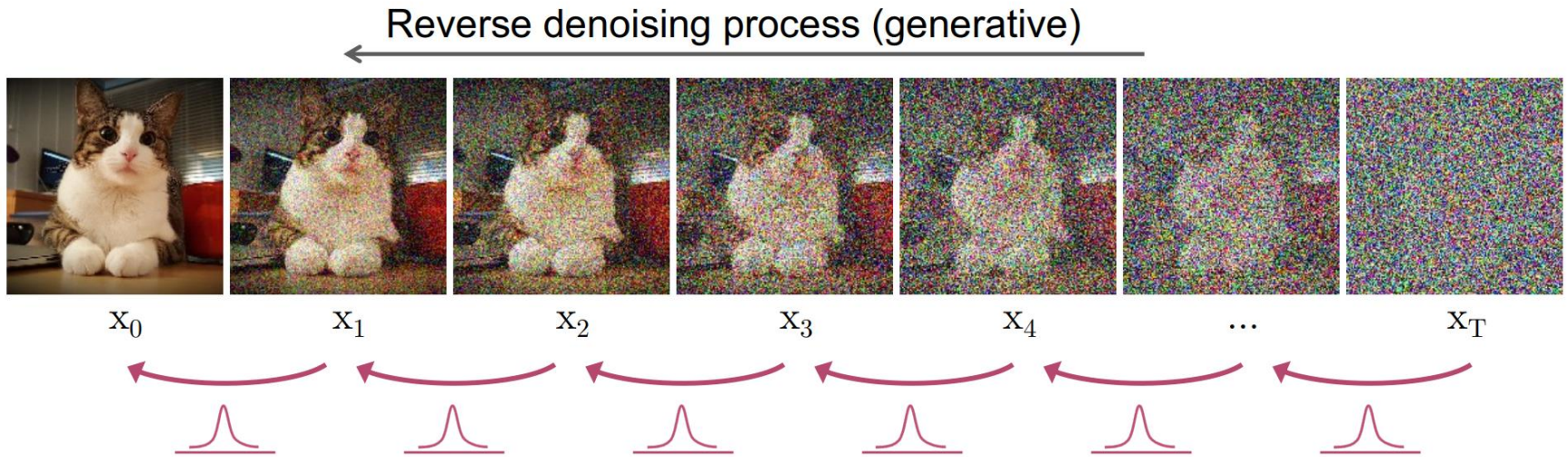
x_0 x_1 x_T

$T = 1{,}000$

$p_\theta(x_0, x_1, \ldots, x_T)$

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:



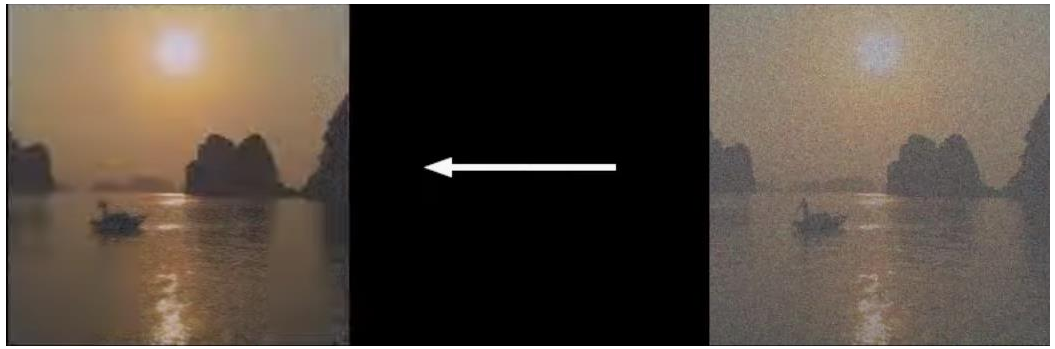
$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}}_{\text{Trainable network (U-net, Denoising Autoencoder)}})$$

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

Remove Gaussian Noise

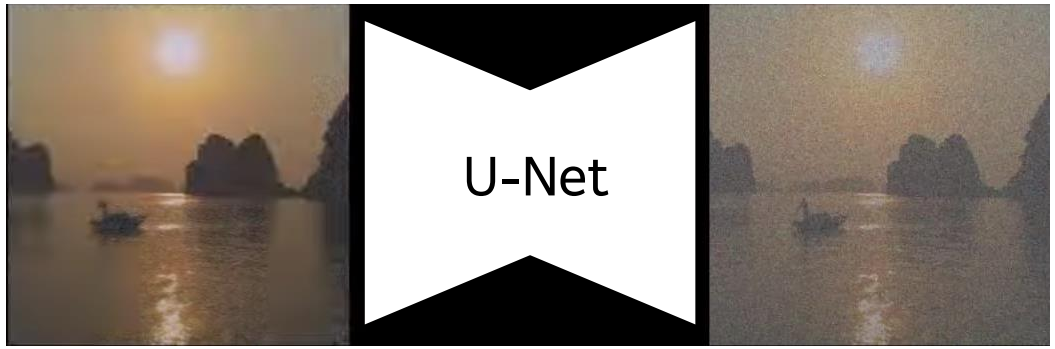
x_{t-1} $p(x_{t-1}|x_t)$ x_t



가우시안 노이즈를 넣을 수는 있지만,
그 반대는 어렵지

어려운 것은 신경망한테 맡기자

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

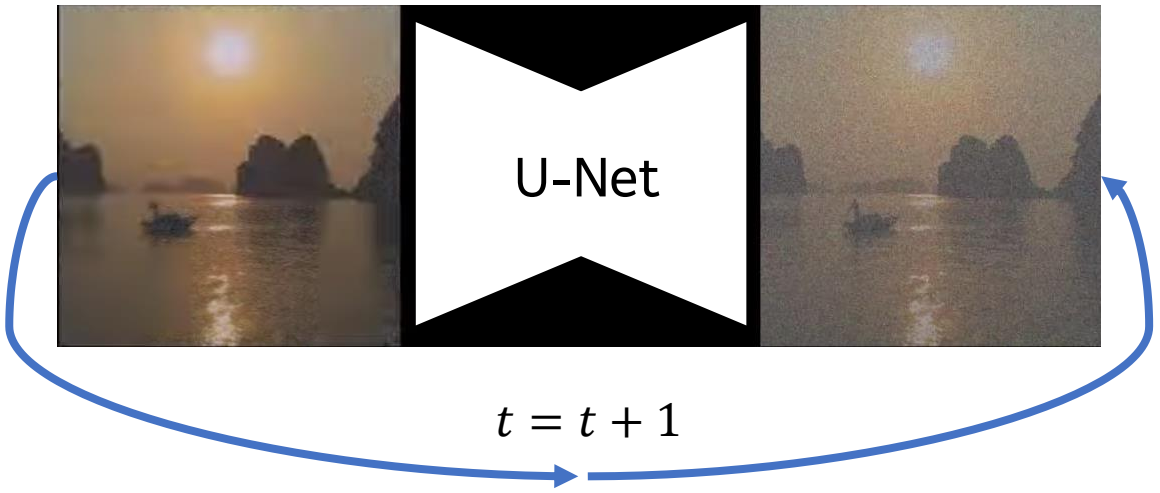


U-Net을 T개 만들기는 좀 그렇지...

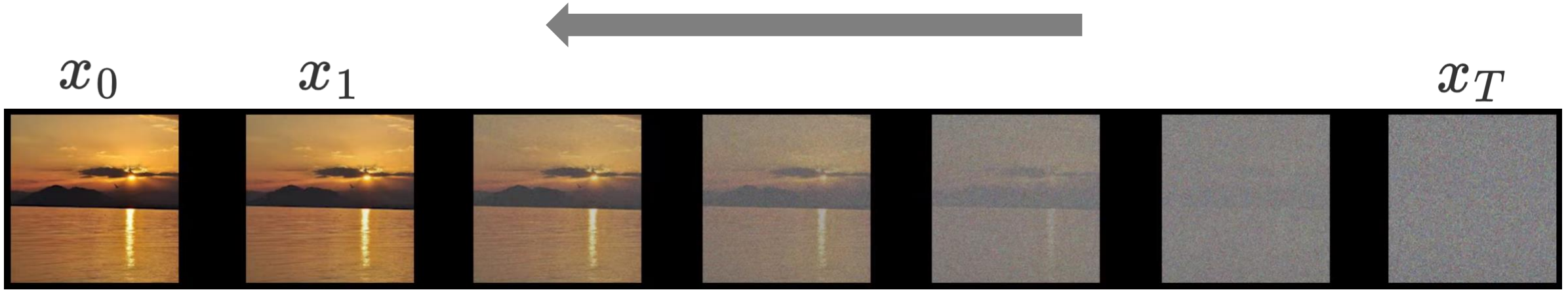
$$p_\theta(x_{t-1}|x_t)$$

$$\mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

t를 파라미터로 갖는
U-Net으로.



시간 정보를 주고, 반복!



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

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

결국, 신경망 훈련을 통해 구하는 것

공분산은 0이라고 간주하자.
각 영상에 낀 노이즈끼리 독립!

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

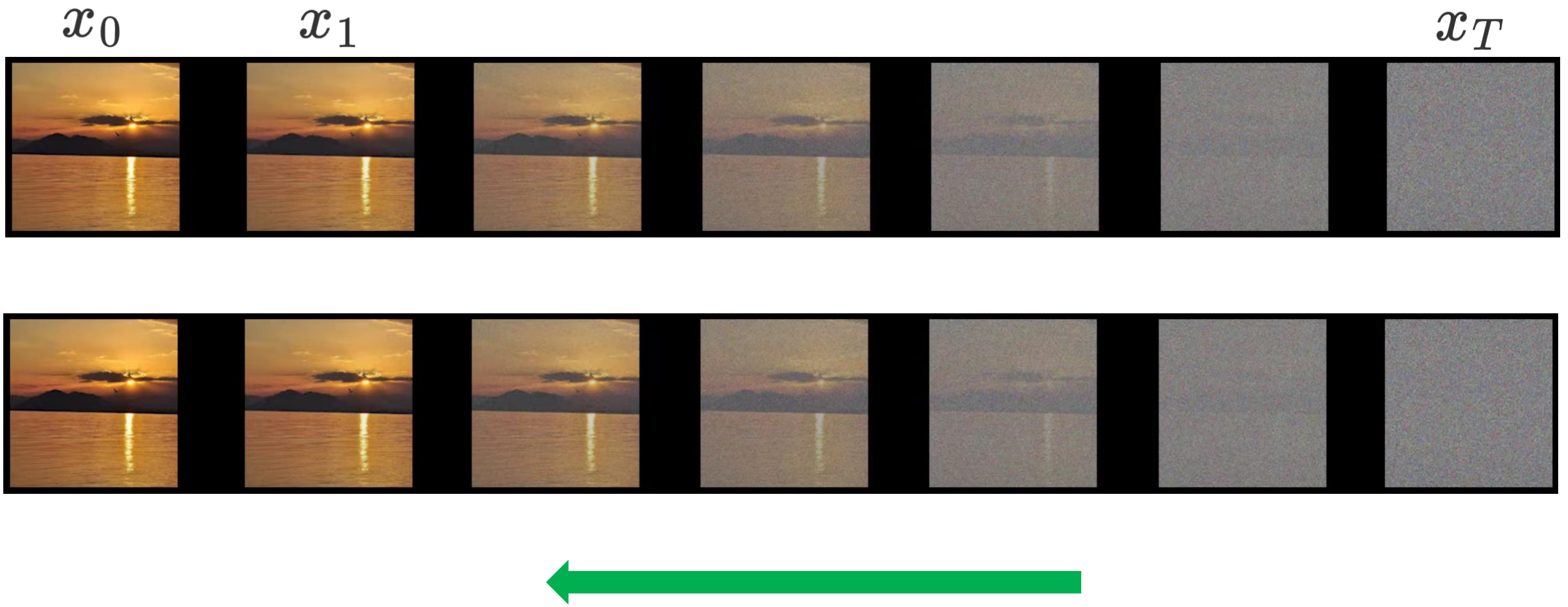
손실함수 유도

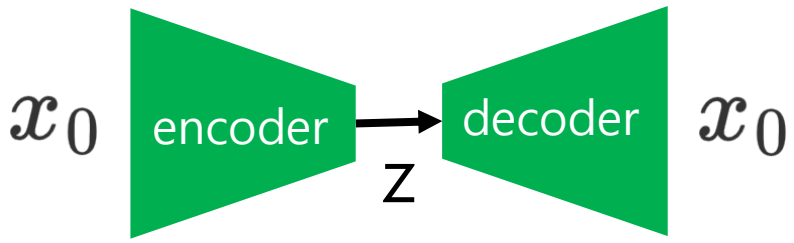
Recall: DPM 손실함수

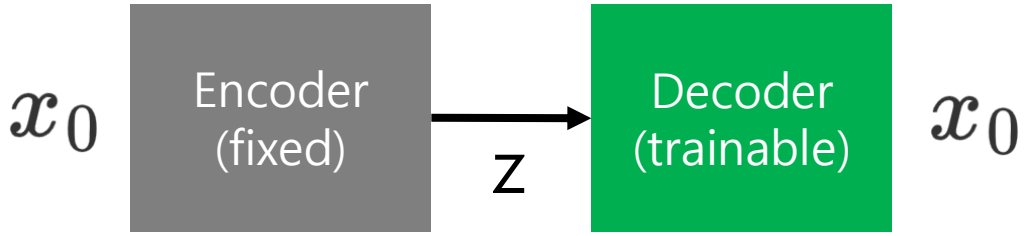
$$E_{q(x_0)}[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

regularization **reconstruction**

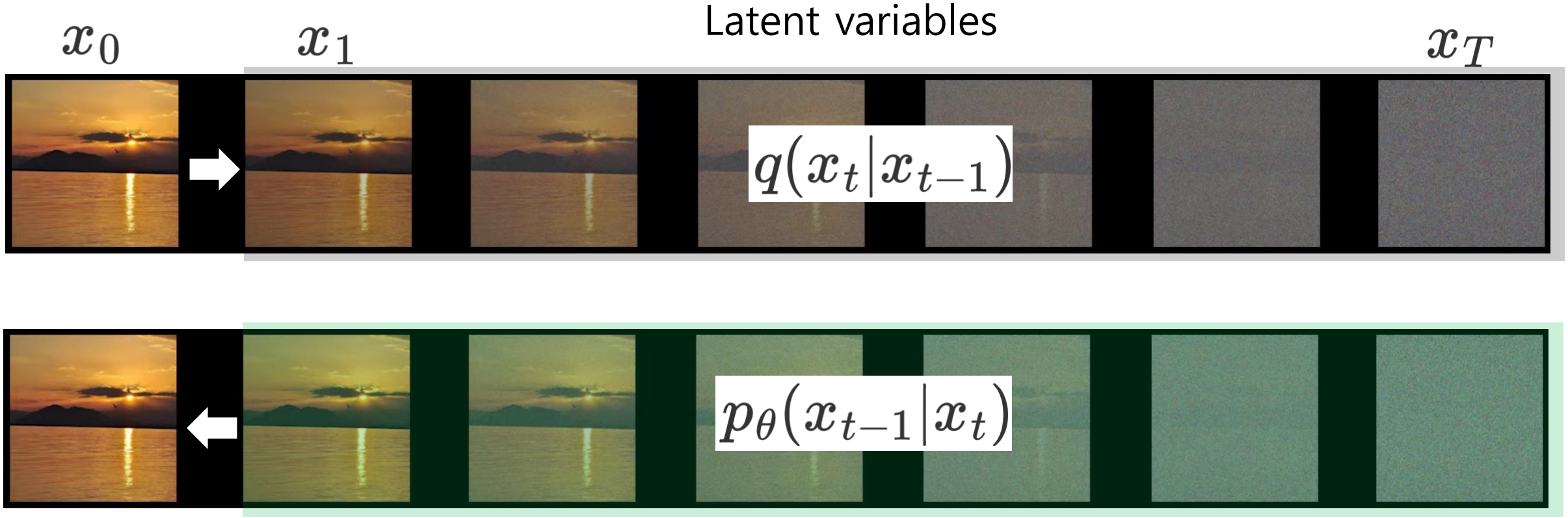
어떻게 훈련시킬까?

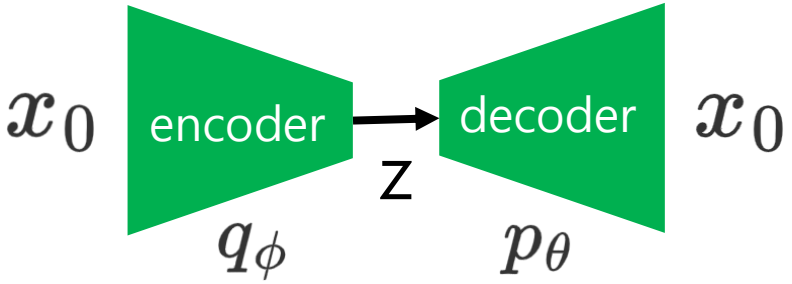


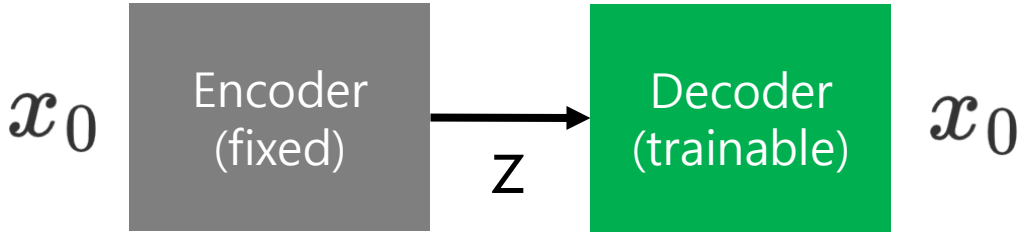

x_0
encoder
z
decoder
x_0


x_0
Encoder
(fixed)
z
Decoder
(trainable)
x_0

어떻게 훈련시킬까?

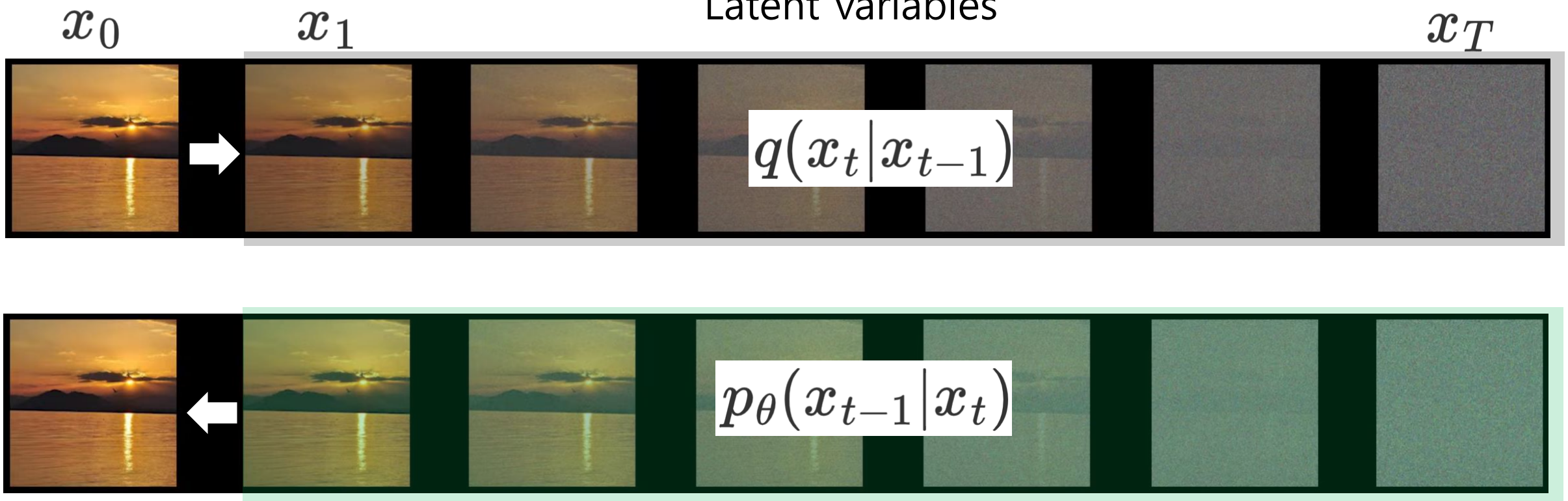

Latent variables
x_0
x_1
x_T
$q(x_t|x_{t-1})$
$p_\theta(x_{t-1}|x_t)$

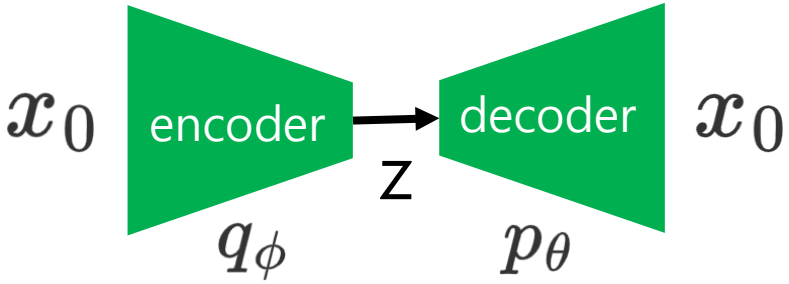

x_0
encoder
z
decoder
x_0
q_ϕ
p_θ

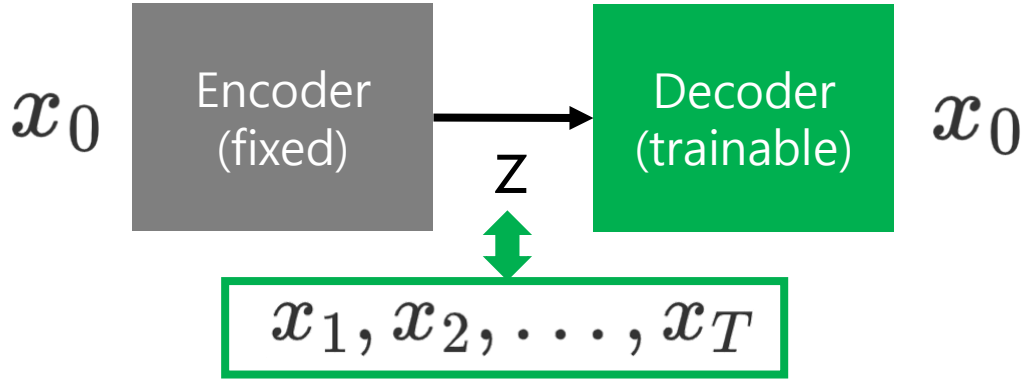

x_0
Encoder
(fixed)
z
Decoder
(trainable)
x_0

어떻게 훈련시킬까?

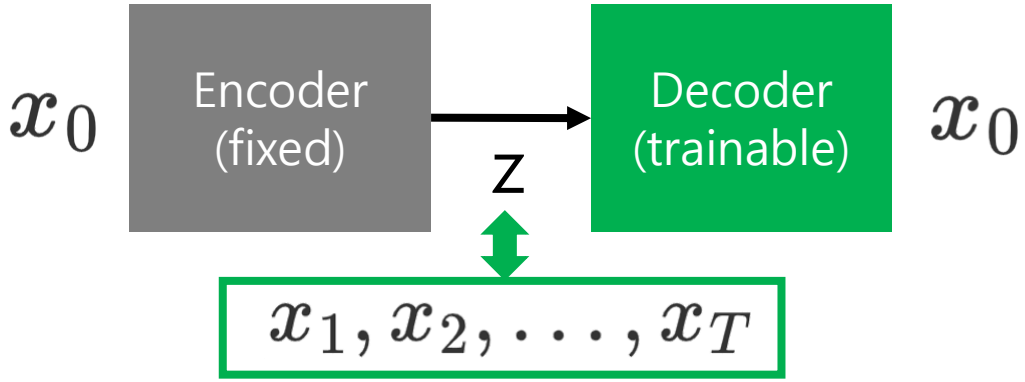
Latent variables



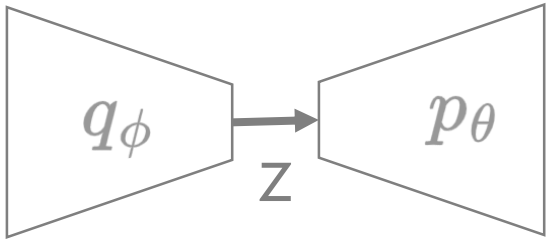




VAE에서 사용했던 테크닉을 사용하자!



디코더만 훈련하면 되니까 더 학습이 수월할지도...



$$log(p_\theta(x)) \geq \underline{\text{evidence lower bound}}$$

ELBO

Variational Inference

Posterior 구하기

꿩 **대신** 닭

↓

변이

$p(z|x)$ 구하기 어렵다.

$q(z|x)$ 우리가 알고 있는 쉬운 분포; 정규분포로 대치하자.

대체(변이) 사후확률 구함

조건: $p(z|x) \approx q(z|x)$

$$\min_{q(z|x)} \mathrm{KL}\big(q(z|x)||p(z|x)\big)$$

KL 최소화

Reverse KL을 선호

VI : biased, low variance

MC: unbiased, high variance

꿩 대신 닭

$$KL\big(q(z|x)||p(z|x)\big) = -\sum q(z|x)log\left(\frac{p(z|x)}{q(z|x)}\right)$$

$$= -\sum q(z|x)log\left(\frac{\frac{p(x,z)}{p(x)}}{\frac{q(z|x)}{1}}\right)$$

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\frac{1}{p(x)}\right)$$

$$= -\sum q(z|x)\left(log\left(\frac{p(x,z)}{q(z|x)}\right) + log\left(\frac{1}{p(x)}\right)\right)$$

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right) + \sum_z q(z|x)log\big(p(x)\big)$$

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right) + log\big(p(x)\big)\sum_z q(z|x)$$

($\sum_z q(z|x) = 1$)

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right) + log\big(p(x)\big)$$

x in black: a random variable
x in gray: a fixed value of the random variable x

x is given, then it is fixed

$$log\big(p(x)\big) = \underbrace{KL\big(q(z|x)||p(z|x)\big)}_{\geq 0} + \sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right)$$

최소화 해야함 → 최대화 하면 된다.

$$log\big(p(x)\big) \geq \sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right)$$

https://www.youtube.com/watch?v=uaaqyVS9-rM

x in black: a random variable
x in gray: a fixed value of the random variable x

꿩 대신 닭

$$KL\big(q(z|x)||p(z|x)\big) = -\sum q(z|x)log\left(\frac{p(z|x)}{q(z|x)}\right)$$

$$= -\sum q(z|x)log\left(\frac{\frac{p(x,z)}{p(x)}}{\frac{q(z|x)}{1}}\right)$$

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\frac{1}{p(x)}\right)$$

$$= -\sum q(z|x)\left(log\left(\frac{p(x,z)}{q(z|x)}\right) + log\left(\frac{1}{p(x)}\right)\right)$$

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right) + \sum_z q(z|x)log\big(p(x)\big)$$

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right) + log\big(p(x)\big)\sum_z q(z|x)$$ (1)

$$= -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right) + log\big(p(x)\big)$$

x is given, then it is fixed

$$log\big(p(x)\big) = \underbrace{KL\big(q(z|x)||p(z|x)\big)}_{\geq 0} + \sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right)$$

최소화 해야함 → 최대화 하면 된다.

$log\big(p(x)\big) \geq$ variational **lower bound**

Log-Likelihood 크게 하는 것

https://www.youtube.com/watch?v=uaaqyVS9-rM

Lower Bound 정리하면…

$$log(p(x)) \geq \sum q(z|x) log\left(\frac{p(x,z)}{q(z|x)}\right)$$

Log-Likelihood
크게 하는 것

$$= \sum q(z|x) log\left(\frac{p(x|z)p(z)}{q(z|x)}\right) \qquad \cancel{p(z|x)p(x)}$$

$$= \sum q(z|x)\left(log(p(x|z)) + log\left(\frac{p(z)}{q(z|x)}\right)\right)$$

$$= \sum q(z|x) log(p(x|z)) + \sum q(z|x) log\left(\frac{p(z)}{q(z|x)}\right)$$

$$= E_{q(z|x)}[log(p(x|z))] - KL(q(z|x)||p(z))$$

Evidence Lower BOund (ELBO)

KL divergence로 표현

$$log(p(x)) \geq \sum q(z|x) log\left(\frac{p(x,z)}{q(z|x)}\right) \Leftrightarrow -KL\left(q(z|x)||p(x,z)\right)$$

Log-Likelihood
크게 하는 것

$$= \sum q(z|x) log\left(\frac{p(x|z)p(z)}{q(z|x)}\right) \qquad \cancel{p(z|x)p(x)}$$

$$= \sum q(z|x)\left(log(p(x|z)) + log\left(\frac{p(z)}{q(z|x)}\right)\right)$$

$$= \sum q(z|x) log(p(x|z)) + \sum q(z|x) log\left(\frac{p(z)}{q(z|x)}\right)$$

$$= E_{q(z|x)}[log(p(x|z))] - KL(q(z|x)||p(z))$$

Evidence Lower BOund (ELBO)

손실함수로...

$$-log(p(x)) \leq -\sum q(z|x)log\left(\frac{p(x,z)}{q(z|x)}\right) \Leftrightarrow KL\left(q(z|x)||p(x,z)\right)$$

- Log-Likelihood 작게 하는 것

$$= -\sum q(z|x)log\left(\frac{p(x|z)p(z)}{q(z|x)}\right) \quad \cancel{p(z|x)p(x)}$$

$$= -\sum q(z|x)\left(log(p(x|z)) + log\left(\frac{p(z)}{q(z|x)}\right)\right)$$

$$= -\sum q(z|x)log(p(x|z)) - \sum q(z|x)log\left(\frac{p(z)}{q(z|x)}\right)$$

$$= -E_{q(z|x)}[log(p(x|z))] + KL(q(z|x)||p(z))$$

Evidence Lower BOund (ELBO)

Learning Objective of DM

VAE

$$-log(p(x)) \leq -\sum q(z|x) log\left(\frac{p(x,z)}{q(z|x)}\right) \Leftrightarrow KL\left(q(z|x)||p(x,z)\right)$$

- Log-Likelihood 작게 하는 것

Diffusion Model

$$-log(p(x_0)) \leq KL\left(q(x_{1:T}|x_0)||p(x_{0:T})\right)$$

Learning Objective of DM

VAE

$$-log(p(x)) \leq -\sum q(z|x) log\left(\frac{p(x,z)}{q(z|x)}\right) \Leftrightarrow KL\left(q(z|x)||p(x,z)\right)$$

- Log-Likelihood 작게 하는 것

Diffusion Model

$$-log(p(x_0)) \leq KL\left(q(x_{1:T}|x_0)||p(x_{0:T})\right)$$

$$E_{q(x_0)}[-log(p(x_0))] \leq E_{q(x_0)}[KL(q||p)]$$

$$E_{q(x_{1:T}|x_0)}[-log(\frac{p(x_{0:T})}{q(x_{1:T}|x_0)})]$$

Learning Denoising Model

Variational upper bound

For training, we can form variational upper bound that is commonly used for training variational autoencoders:

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

1

Sohl-Dickstein et al. ICML 2015 and Ho et al. NeurIPS 2020 show that:

$$L = \mathbb{E}_{q(x_0)\cancel{q(x_{1:T}|x_0)}}\left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1))}_{L_0}\right]$$

where $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ is the tractable posterior distribution:

2

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

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

1

$$E_{q(x_{0:T})}[-log(\frac{p(x_{0:T})}{q(x_{1:T}|x_0)})]$$

$$E_q[-log\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}]$$

$$q(x_{0:T}) = q(x_0)q(x_{1:T}|x_0)$$

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

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

$$E_q\left[-logp(x_T) - log\frac{\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}\right]$$

$$E_q\left[-logp(x_T) - \sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

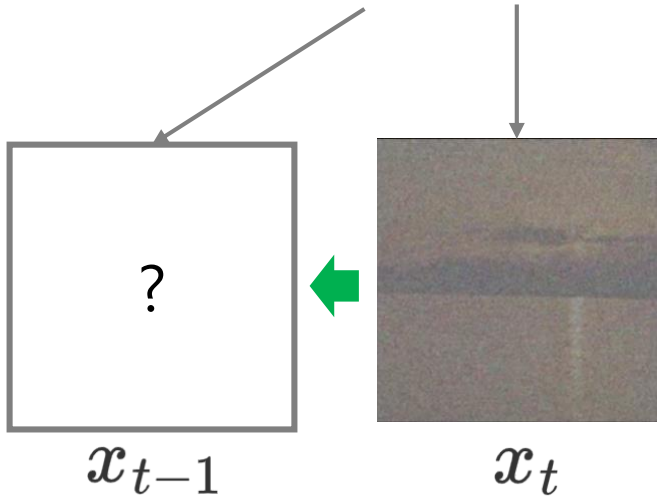
https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/

1 $E_q\left[-logp(x_T) - \sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$

2

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

2 $$-\sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t)}\frac{q(x_{t-1})}{q(x_t)}$$



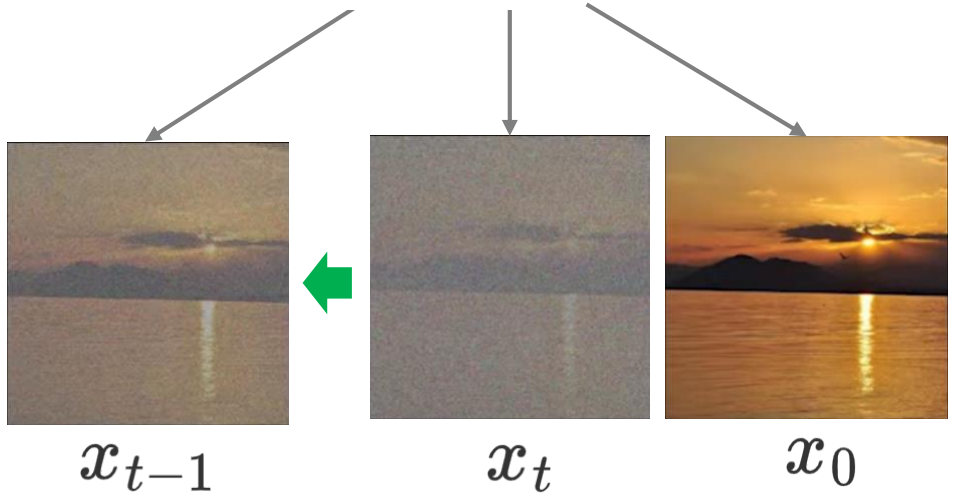
https://www.youtube.com/watch?v=HoKDTa5jHvg

1 $$E_q\left[-logp(x_T)-\sum_{t=2}^{T}log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}-log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

2

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

2 $$-\sum_{t=2}^{T}log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}$$



https://www.youtube.com/watch?v=HoKDTa5jHvg

2

$$-\sum_{t=2}^{T} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$= -\sum_{t=2}^{T} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - \sum_{t=2}^{T} log \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

3

$$= -\sum_{t=2}^{T} logq(x_{t-1}|x_0) + \sum_{t=2}^{T} logq(x_t|x_0)$$

3

$$= -\sum_{t=1}^{T-1} logq(x_t|x_0) + \sum_{t=2}^{T} logq(x_t|x_0)$$

$$= -logq(x_1|x_0) - \sum_{t=2}^{T-1} logq(x_t|x_0) + \sum_{t=2}^{T-1} logq(x_t|x_0) + logq(x_T|x_0)$$

$$= -logq(x_1|x_0) + logq(x_T|x_0)$$

https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/

2 $$-\sum_{t=2}^{T} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$= -\sum_{t=2}^{T} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} -logq(x_1|x_0) + logq(x_T|x_0)$$

3

$$= -\sum_{t=2}^{T} logq(x_{t-1}|x_0) + \sum_{t=2}^{T} logq(x_t|x_0)$$ 3

$$= -\sum_{t=1}^{T-1} logq(x_t|x_0) + \sum_{t=2}^{T} logq(x_t|x_0)$$

$$= -logq(x_1|x_0) - \sum_{t=2}^{T-1} logq(x_t|x_0) + \sum_{t=2}^{T-1} logq(x_t|x_0) + logq(x_T|x_0)$$

$$= -logq(x_1|x_0) + logq(x_T|x_0)$$

1 $E_q \left[-logp(x_T) - \sum_{t=2}^{T} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)} \right]$

2

$$- \sum_{t=2}^{T} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} - logq(x_1|x_0) + logq(x_T|x_0)$$

1 $$E_q\left[-logp(x_T)-\sum_{t=2}^{T}log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}-log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

2 $$-\sum_{t=2}^{T}log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}-logq(x_1|x_0)+logq(x_T|x_0)$$

$$E_q\left[-log\frac{p(x_T)}{q(x_T|x_0)}-\sum_{t=2}^{T}log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}-logp_\theta(x_0|x_1)\right]$$

1

$$E_q\left[-logp(x_T) - \sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

2

$$-\sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - logq(x_1|x_0) + logq(x_T|x_0)$$

$$E_q\left[-log\frac{p(x_T)}{q(x_T|x_0)} - \sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - logp_\theta(x_0|x_1)\right]$$

1

$$E_q\left[-logp(x_T) - \sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

2

$$-\sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - logq(x_1|x_0) + logq(x_T|x_0)$$

$$E_q\left[-log\frac{p(x_T)}{q(x_T|x_0)} - \sum_{t=2}^{T} log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - logp_\theta(x_0|x_1)\right]$$

$$E_q \left[-log \frac{p(x_T)}{q(x_T|x_0)} - \sum_{t=2}^{T} log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} - log p_\theta(x_0|x_1) \right]$$

$$E_q\left[-log\frac{p(x_T)}{q(x_T|x_0)}-\sum_{t=2}^{T}log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}-logp_\theta(x_0|x_1)\right]$$

$$D_{KL}(p_1(x)||p_2(x))=-\int p_1(x)log\frac{p_2(x)}{p_1(x)}dx=E_{p_1(x)}\left[-log\frac{p_2(x)}{p_1(x)}\right]$$

$$\begin{aligned}=&D_{KL}(q(x_T|x_0)||p(x_T))\\&+\sum_{t=2}^{T}D_{KL}(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t))\\&-logp_\theta(x_0|x_1)\end{aligned}$$

★ ★★

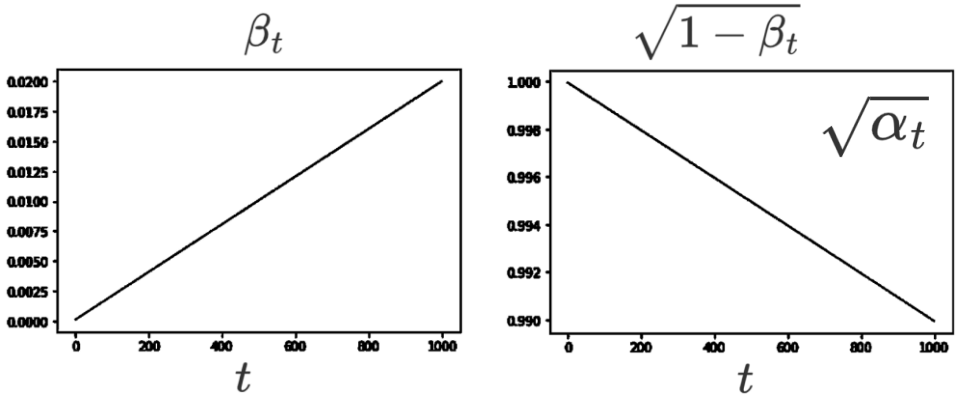
$E_q[-logp_\theta(x_0|x_1)]$ cross-entropy / mse

$$E_{q(x_0)}[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_T))}_{L_T}+\sum_{t=2}^{T}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

2

$$x_t \sim N(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

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



diffusion kernel ⬇ α_t 대신 $\bar{\alpha}_t$

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

$$\beta_1 = 10^-4, \ldots, \beta_T = 0.02$$

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

$$x_t \sim N\left(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right)$$

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

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

$\bar{\alpha}_t$ 점점 0으로 근접,

$$q(x_T|x_0) \sim N(\underbrace{0.00635}_{\approx 0} \cdot x_0, \underbrace{0.9999}_{\approx 1} \cdot I) \approx N(0, I)$$

$$x_T \sim N(0, I)$$
$$p(x_T) = N(0, I)$$

L_T $D_{KL}(q(x_T|x_0)||p(x_T)) \approx Constant$

Parameterizing the Denoising Model

Since both $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ are Normal distributions, the KL divergence has a simple form:

$$L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}||\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)||^2\right] + C$$

Recall that $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$. Ho et al. NeurIPS 2020 observe that:

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

They propose to represent the mean of the denoising model using a *noise-prediction* network:

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

With this parameterization

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon}_{\mathbf{x}_t}, t)||^2\right] + C$$

★

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

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

가우시안 분포라고 가정

mean

variance

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

by Bayes' rule

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

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

$$\propto \quad exp\left(-\frac{1}{2}\left(\frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t-\sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \quad exp\left(-\frac{1}{2}\left(\frac{x_t^2-2\sqrt{\alpha_t}x_t x_{t-1}+\alpha_t x_{t-1}^2}{\beta_t} + \frac{x_{t-1}^2-2\sqrt{\bar{\alpha}_{t-1}}x_0 x_{t-1}+\bar{\alpha}_{t-1}x_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t-\sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \quad exp\left(-\frac{1}{2}\left(\underbrace{\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)}_{a}x_{t-1}^2 - \underbrace{\left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)}_{b}x_{t-1} + C(x_t, x_0)\right)\right)$$

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}((x-\mu)/\sigma)^2} \dashrightarrow \frac{(x-\mu)^2}{\sigma^2} = \frac{x^2}{\sigma^2} - \frac{2x\mu}{\sigma^2} + \frac{\mu^2}{\sigma^2}$$

$$a = \frac{1}{\sigma^2}, \sigma^2 = \frac{1}{a} \qquad b = \frac{2\mu}{\sigma^2}, \mu = \frac{b\sigma^2}{2} = \frac{b/a}{2} = \frac{b}{2a}$$

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

$\tilde{\beta}_t$

$$a = \left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right) \qquad a = \frac{1}{\sigma^2}, \sigma^2 = \frac{1}{a}$$

$$b = \left(\frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0\right) \qquad b = \frac{2\mu}{\sigma^2}, \mu = \frac{b\sigma^2}{2} = \frac{b/a}{2} = \frac{b}{2a}$$

$\tilde{\mu}$

$$\tilde{\beta}_t = \frac{1}{a} = \frac{1}{(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})} = \frac{1}{\frac{\alpha_t - \alpha_t\bar{\alpha}_{t-1} + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}} = \frac{1}{\frac{\alpha_t - \bar{\alpha}_t + 1 - \alpha_t}{\beta_t(1-\bar{\alpha}_{t-1})}} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\tilde{\mu}(x_t, x_0) = \frac{b}{2a} = \frac{\left(\frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0\right)}{2\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)} = \left(\frac{\sqrt{\alpha_t}}{\beta_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0\right)\left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right)$$

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

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

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

x_0 를 없애자.

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

$$= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon\right) \quad \epsilon \sim N(0, I)$$

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

x_t 에 대한 함수가 되었다.

$$\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon\right)$$

$$\left(\frac{\alpha_t(1-\bar{\alpha}_{t-1})}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)} + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}\right)x_t - \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}\sqrt{1-\bar{\alpha}_t}\epsilon$$

$$\frac{\sqrt{\bar{\alpha}_t}\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})+\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$\frac{\sqrt{\bar{\alpha}_t}\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})+\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$\frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{\alpha_t}\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})+\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$\frac{\sqrt{\bar{\alpha}_{t-1}}\alpha_t-\sqrt{\bar{\alpha}_{t-1}}\alpha_t\bar{\alpha}_{t-1}+\sqrt{\bar{\alpha}_{t-1}}-\sqrt{\bar{\alpha}_{t-1}}\alpha_t}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$\frac{-\sqrt{\bar{\alpha}_{t-1}}\bar{\alpha}_t+\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$\frac{\sqrt{\bar{\alpha}_{t-1}}(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$\frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\alpha_t}\sqrt{\bar{\alpha}_{t-1}}}$$

$$\frac{1}{\sqrt{\alpha_t}}$$

$$\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\bar{\alpha}_t}\sqrt{1-\bar{\alpha}_t}}$$

$$\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\alpha_t}\sqrt{\bar{\alpha}_{t-1}}\sqrt{1-\bar{\alpha}_t}}$$

$$\frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}$$

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

$$p_\theta(x_{t-1}|x_t) = N\left(\mu_\theta(x_t, t), \sum_\theta(x_t, t)\right)$$

가우시안 분포라고 가정

mean

variance

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

output input

$$p_\theta(x_{t-1}|x_t) = N\left(\mu_\theta(x_t, t), \sum_\theta(x_t, t)\right)$$

output input

신경망이 훈련하도록 해도 되겠지만,

- $\sigma_t^2 I$ t가 크면 이 값이 크도록 설정해야 한다.
- β_t t가 크면 이 값이 크다.
- $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ t가 크면 이 값이 크다.

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

$$p_\theta(x_{t-1}|x_t) = N\left(\mu_\theta(x_t, t), \sum_\theta(x_t, t)\right)$$

$$\tilde{\mu}(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) \quad \epsilon \sim N(0, I)$$

Proposed by Ho et al.
Denoising Diffusion Probabilistic Models (DDPM) - NeurIPS 2020

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

L_{t-1} $\sum_{t=2}^{T} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$

$$q(x_{t-1}|x_t, x_0) = N\left(\tilde{\mu}(x_t, x_0), \sigma_t^2 I\right)$$
$$p_\theta(x_{t-1}, x_t) = N\left(\mu_\theta(x_t, t), \sigma_t^2 I\right)$$

KL divergence between two Gaussian distributions

$$KL(p, q) = \frac{1}{2}\log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

$$p(x) = N(\mu_1, \sigma_1^2)$$
$$q(x) = N(\mu_2, \sigma_2^2)$$

$$KL(p,q) = -\int p(x)\log q(x)dx + \int p(x)\log p(x)dx$$

$$p(x) = N(\mu_1,\sigma_1) \text{ and } q(x) = N(\mu_2,\sigma_2)$$

$$\int \left[\log(p(x)) - \log(q(x))\right] p(x)dx$$

$$= \int \left[-\tfrac{1}{2}\log(2\pi) - \tfrac{1}{2}\log(\sigma_1) - \tfrac{1}{2}\left(\tfrac{x-\mu_1}{\sigma_1}\right)^2 + \tfrac{1}{2}\log(2\pi) + \log(\sigma_2) + \tfrac{1}{2}\left(\tfrac{x-\mu_2}{\sigma_2}\right)^2\right]$$

$$\times \tfrac{1}{\sqrt{2\pi}\sigma_1}\exp\left[-\tfrac{1}{2}\left(\tfrac{x-\mu_1}{\sigma_1}\right)^2\right]dx$$

$$= \int \left\{\tfrac{1}{2}\log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2}\left[\left(\tfrac{x-\mu_2}{\sigma_2}\right)^2 - \left(\tfrac{x-\mu_1}{\sigma_1}\right)^2\right]\right\} \times \tfrac{1}{\sqrt{2\pi}\sigma_1}\exp\left[-\tfrac{1}{2}\left(\tfrac{x-\mu_1}{\sigma_1}\right)^2\right]dx$$

$$= E_1\left\{\tfrac{1}{2}\log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2}\left[\left(\tfrac{x-\mu_2}{\sigma_2}\right)^2 - \left(\tfrac{x-\mu_1}{\sigma_1}\right)^2\right]\right\}$$

$$= \tfrac{1}{2}\log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2\sigma_2^2}E_1\left\{(X-\mu_2)^2\right\} - \tfrac{1}{2\sigma_1^2}E_1\left\{(X-\mu_1)^2\right\}$$

$$= \tfrac{1}{2}\log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2\sigma_2^2}E_1\left\{(X-\mu_2)^2\right\} - \tfrac{1}{2}$$

(Now note that

$$(X-\mu_2)^2 = (X-\mu_1+\mu_1-\mu_2)^2 = (X-\mu_1)^2 + 2(X-\mu_1)(\mu_1-\mu_2) + (\mu_1-\mu_2)^2)$$

$$= \tfrac{1}{2}\log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{1}{2\sigma_2^2}\left[E_1\left\{(X-\mu_1)^2\right\} + 2(\mu_1-\mu_2)E_1\left\{X-\mu_1\right\} + (\mu_1-\mu_2)^2\right] - \tfrac{1}{2}$$

$$= \tfrac{1}{2}\log\left(\tfrac{\sigma_2}{\sigma_1}\right) + \tfrac{\sigma_1^2+(\mu_1-\mu_2)^2}{2\sigma_2^2} - \tfrac{1}{2}$$

https://stats.stackexchange.com/questions/7440/kl-divergence-between-two-univariate-gaussians

$$L_{t-1} \quad \sum_{t=2}^{T} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$$

$$q(x_{t-1}|x_t, x_0) = N\left(\tilde{\mu}(x_t, x_0), \sigma_t^2 I\right)$$
$$p_\theta(x_{t-1}, x_t) = N\left(\mu_\theta(x_t, t), \sigma_t^2 I\right)$$

$$E_{q(x_0)}[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

KL divergence between two Gaussian distributions

$$KL(p, q) = \frac{1}{2}\log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

$p(x) = N(\mu_1, \sigma_1^2)$
$q(x) = N(\mu_2, \sigma_2^2)$

$$E_{q(x_0)}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\right\|^2\right] + C$$

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

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

$$E_{q(x_0)}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\right\|^2\right] + C$$

for MCMC $x_0 \sim q(x_0)$

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

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

$$\epsilon \sim N(0, I)$$

$$E_{x_0 \sim q(x_0), \epsilon \sim N(0,I)}\left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta(x_t, t)\right\|^2\right]$$

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

noise-prediction network
(denoising)

$$E_{x_0 \sim q(x_0), \epsilon \sim N(0,I)} \left[\underbrace{\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)}}_{\lambda_t = 1} \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \right\|^2 \right]$$

$$L_{simple} = E_{t \sim U(1,T), x_0 \sim q(x_0), \epsilon \sim N(0,I)} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \right\|^2 \right]$$

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** $\mathbf{x}_0$

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

Training Objective Weighting

Trading likelihood for perceptual quality

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \Big[\underbrace{\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}}_{\lambda_t} ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, t)||^2 \Big]$$

The time dependent λ_t ensures that the training objective is weighted properly for the maximum data likelihood training.

However, this weight is often very large for small t's.

Ho et al. NeurIPS 2020 observe that simply setting $\lambda_t = 1$ improves sample quality. So, they propose to use:

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(1,T)} \Big[||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon}_{\mathbf{x}_t}, t)||^2 \Big]$$

For more advanced weighting see Choi et al., Perception Prioritized Training of Diffusion Models, CVPR 2022.

Summary

Training and Sample Generation

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: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
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** $\mathbf{x}_0$

References

- [**Paper**] Jascha Sohl-Dickstein et al., "**Deep Unsupervised Learning using Nonequilibrium Thermodynamics**", ICML, 2015.
- [**Paper**] Jonathan Ho et al., "**Denoising diffusion probabilistic models**", arXiv:2006.11239, 2020.
- [**Tutorial**] Karsten Kreis et al., "**Denoising Diffusion-based Generative Modeling: Foundations and Applications**", June 19, CVPR 2022.
- [**Tutorial**] (한글) Injung Kim, "**Diffusion Probabilistic Models**", July 1, KCC 2022.
- [**Tutorial**] (한글) Jaepil Ko, "**Tutorial on VAE**", slideshare, https://www.slideshare.net/jaepilko10/variational-autoencoder-vae-255270887
- [**Blog**] **What are Diffusion Models**, https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- [**Blog**] Introduction to Diffusion Models for Machine Learning, https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/
- [**Blog**] (한글) https://developers-shack.tistory.com/8
- [**YouTube**] https://www.youtube.com/watch?v=HoKDTa5jHvg