DATA
SCIENCE LAB

Probabilistic Generative Models And Diffusion

23.03.30 / 7기 박지호

0. Intro

DALL.E 2

"a teddy bear on a skateboard in times square"

Imagen

A group of teddy bears in suit in a corporate office celebrating the birthday of their friend. There is a pizza cake on the desk.

0. Intro


A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.
A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat.
A photo of a raccoon wearing an astronaut helmet, looking out of the window at night.
A robot couple fine dining with Eiffel Tower in the background.
An extremely angry bird.
A small cactus wearing a straw hat and neon sunglasses in the Sahara desert.
Teddy bears swimming at the Olympics 400m Butterfly event
An alien octopus floats through a portal reading a newspaper.

CONTENTS

1. Basics

1. Basics

Generative Model의 Task(Goal)?

Data Distribution을 알아내는 것

Training samples ~$p_{data}(x)$ Generated samples ~$p_{model}(x)$

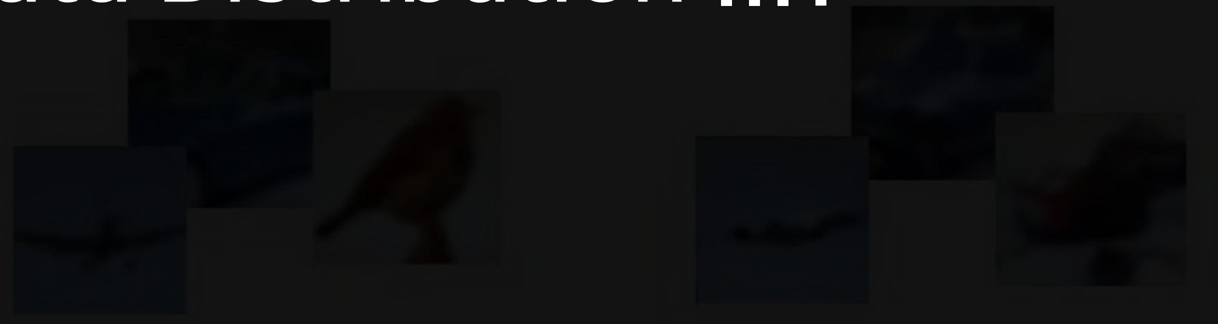
Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

1. Basics

Generative Model의 Task(Goal)?

Data Distribution을 알아내는 것

Data Distribution …?



Training samples $\sim p_{data}(x)$ Generated samples $\sim p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

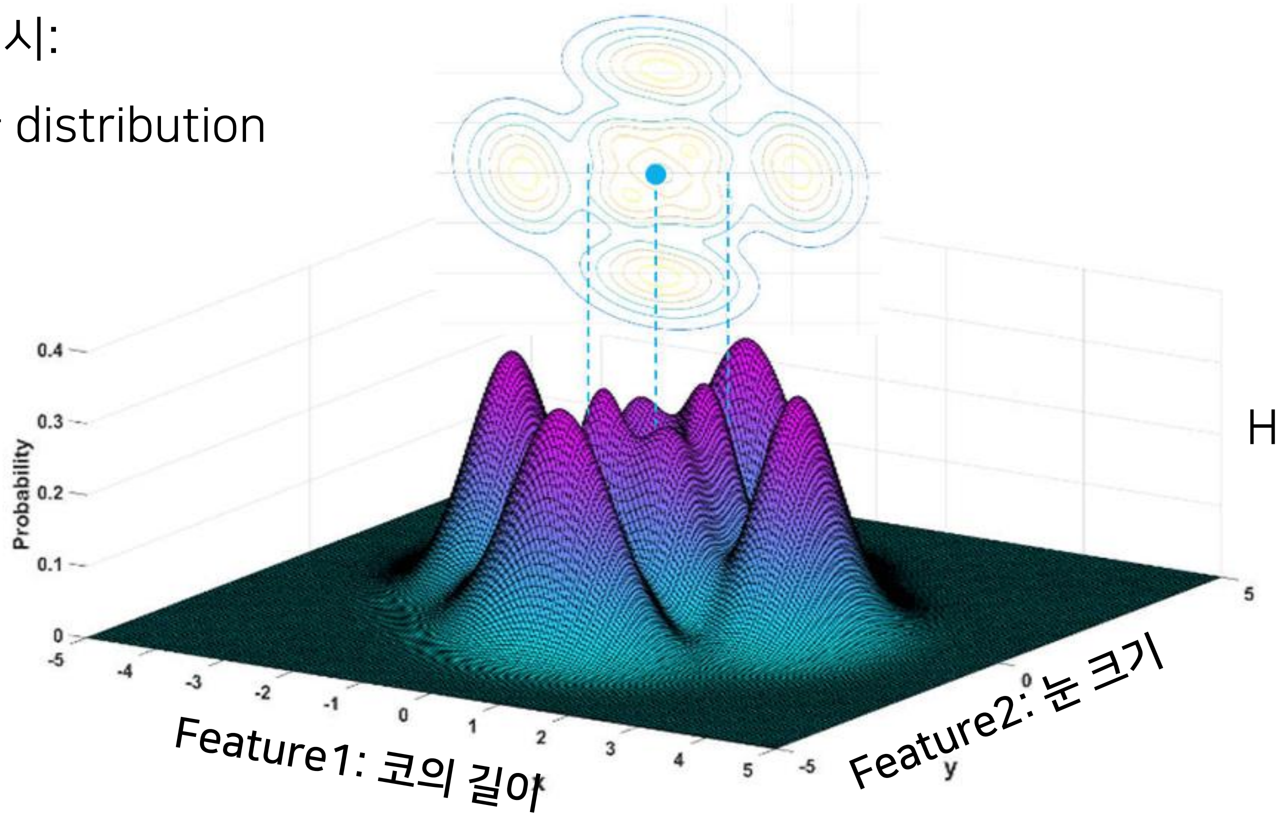
1. Basics

1) Data Distribution

Data Points on Feature HyperPlane→ PDF/PMF on HyperCube ⇒ Data Distribution

비유적 예시:

사람 얼굴 distribution



High Probability

Low Probability

1. Basics

2) Basic Probability Equations

Marginalization

$$p(x) = \int p(x, z)dz$$

(Chain rule: $p(x) = \frac{p(x,z)}{p(z|x)}$)

Bayesian

$$\boldsymbol{p(H|e) = \frac{p(e|H)p(H)}{p(e)}}$$

H: Hypothesis

e: evidence

Expectation

$$E_{q(z)}[p(x|z)] = \int q(z)p(x|z)dz$$

$$= \sum_i^{\infty} q(z)p(x|z)$$

1. Basics

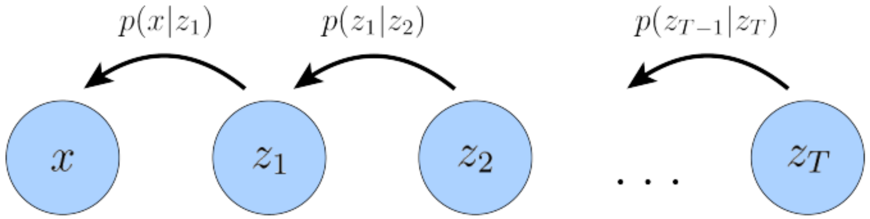
2) Basic Probability Equations

Monte Carlo Approximation

$$E[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Markov Process

$$p(x|z_1, z_2 \dots, z_T) = p(x|z_1)$$



1. Basics

3) KL-Divergence

- Distribution의 다른(divergence) 정도
- 2 가지 수식 유도/해석 : 1) 정보이론 entropy 관점, 2) 단순한 확률 값 차이의 평균

$$D_{KL}(p(x)||q(x)) = \int p(x)\log\frac{p(x)}{q(x)}dx$$

$$= E_{p(x)}[\log\frac{p(x)}{q(x)}]$$

1. Basics

3) KL-Divergence

수식 유도(2번 관점)

Step1. 각 데이터의 확률 값(log)의 차이

$$\log p(x_i) - \log q(x_i) = \log \frac{p(x_i)}{q(x_i)}$$

Step2. 데이터의 등장 빈도 반영 하여 평균

$$\sum_i p(x_i) \log \frac{p(x_i)}{q(x_i)}$$

Monte Carlo Approximation

$$D_{KL}(p(x)||q(x)) = \sum_i^{\infty} p(x_i) \log \frac{p(x_i)}{q(x_i)} = \int p(x) \log \frac{p(x)}{q(x)} dx = E_{p(x)}[\log \frac{p(x)}{q(x)}]$$

2. VAE Review

2. VAE Review

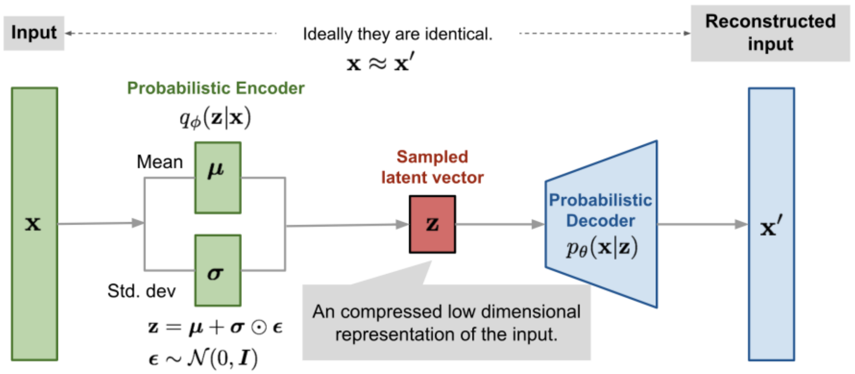
1) Maximizing ELBO → Maximizing $p_\theta(x)$

$$\log p_\theta(x) \geq ELBO$$

2) Inside ELBO

$$ELBO = E_{q_\phi(z|x)}[p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$$

3) Actual Implementation



2. VAE Review

1) Maximizing ELBO → Maximizing $p_\theta(x)$

원래 Maximizing ELBO의 의미 = posterior $p(z|x)$ 의 근사! Not maximizing $p(x)$

2. VAE Review

1) Maximizing ELBO → Maximizing $p_\theta(x)$

$p_\theta(x)$를 maximizing 하는 것이 무슨 의미?

$$\log p_\theta(x) \geq ELBO$$

p_θ: Generator가 만들어내는 data의 Distribution

x: 실제 Data

$p_\theta(x)$: Generator가 만들어내는 Distribution에 대한 실제 data의 확률

2. VAE Review

2) Inside ELBO

Variational inference를 위한 approximation class 중 선택

원 데이터에 대한 likelihood 선택

다루기 쉬운 확률 분포 중 선택

$$L_i(\phi, \theta, x_i) = \underbrace{-\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p_\theta(x_i|z)\right)\right]}_{\text{Reconstruction Error}} + \underbrace{KL\left(q_\phi(z|x_i)||p(z)\right)}_{\text{Regularization}}$$

Reconstruction Error

- 현재 샘플된 z에 대한 negative log likelihood
- x_i에 대한 복원 오차 (AutoEncoder 관점)

Regularization

- 현재 샘플된 z에 대한 대한 추가 조건
- 샘플링되는 z들에 대한 통제성을 prior를 통해 부여, Variational distribution q(z|x)가 p(z)와 유사해야 한다는 조건을 부여

2. VAE Review

3) Actual Implementation

Maximize ELBO: $E_{q_\phi(z|x)}[p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$

실제 Loss

```
def loss_function(recon_x, x, mu, logvar):
    reconstruction = F.mse_loss(recon_x,x)
    #reconstruction = F.binary_cross_entropy(recon_x, x.view(-1, 784))
    prior_matching = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())
    return reconstruction + prior_matching
```

2. VAE Review

3) Actual Implementation

Maximize ELBO: $E_{q_\phi(z|x)}[p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$

ELBO식을 계산하기 위해서는,

또 다른 가정(inductive bias)이 필요하다.

실제 Loss

```
def loss_function(recon_x, x, mu, logvar):
    reconstruction = F.mse_loss(recon_x,x)
    #reconstruction = F.binary_cross_entropy(recon_x, x.view(-1, 784))
    prior_matching = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())
    return reconstruction + prior_matching
```

2. VAE Review

3) Actual Implementation

Modeling Process

Step1. 확률 모델링

Step2. 적절한 가정을 추가하여 만든 모델을 계산

3. Diffusion Intro

3. Diffusion Introduction

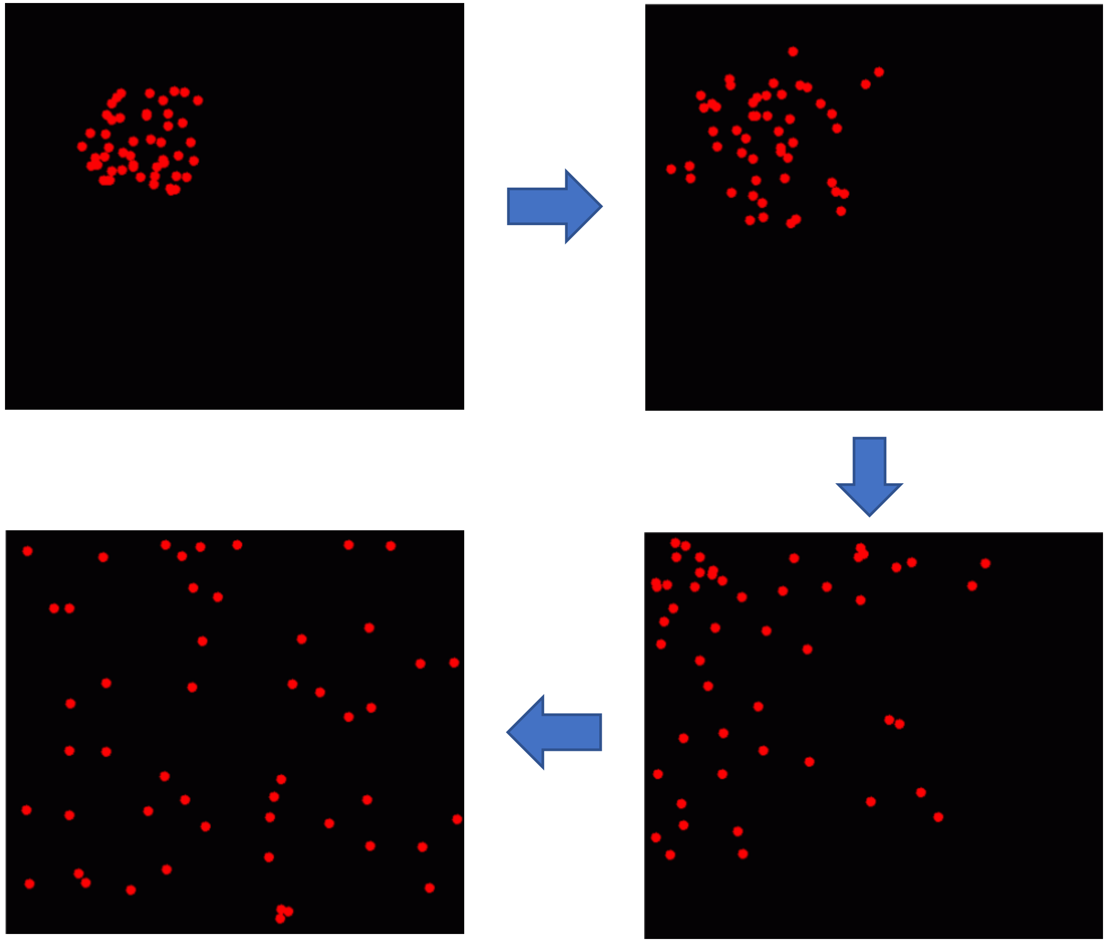
1) Intuition of Diffusion
2) Hierarchical VAE
3) Paper Timeline

3. Diffusion Introduction

1) Intuition: 확산!

Data Distribution:
x

Ideal Latent:
z

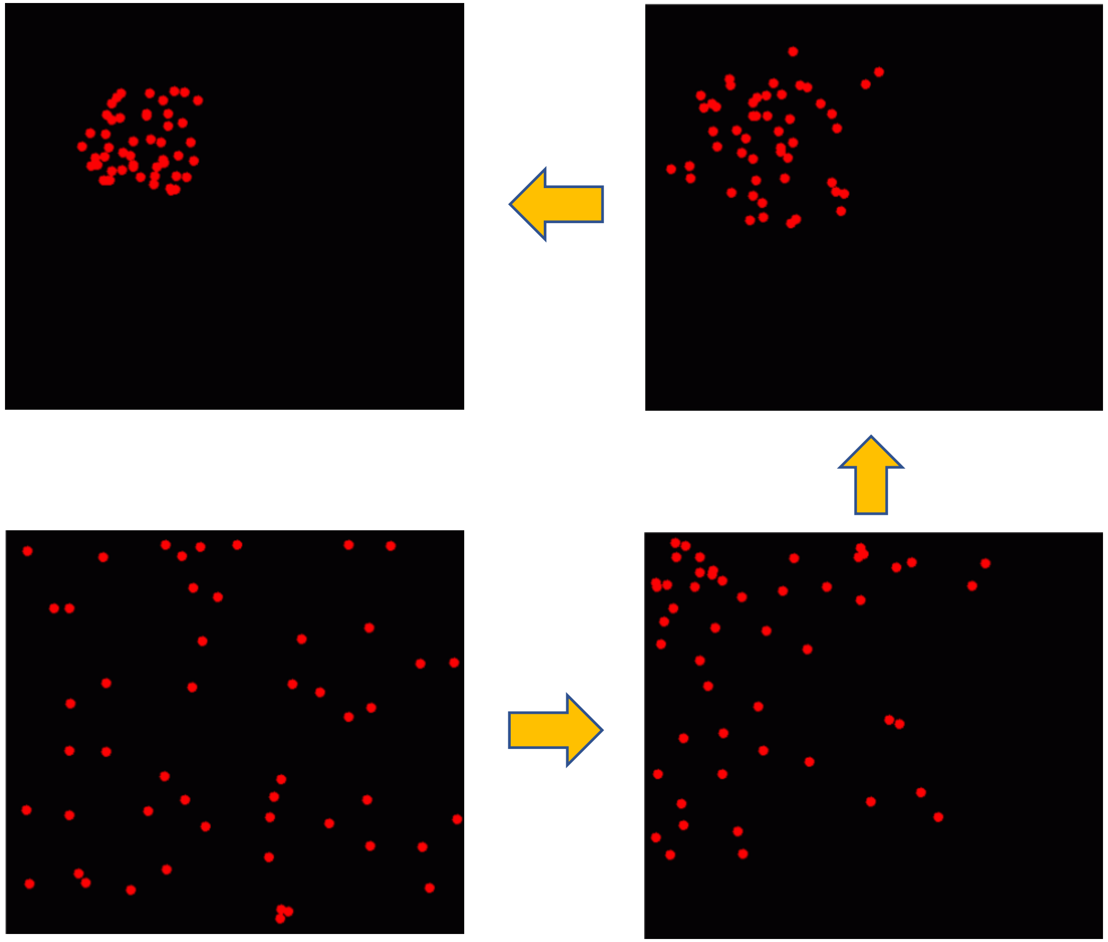


3. Diffusion Introduction

1) Intuition: 확산 과정의 역은 Generation?

Data Distribution:
x

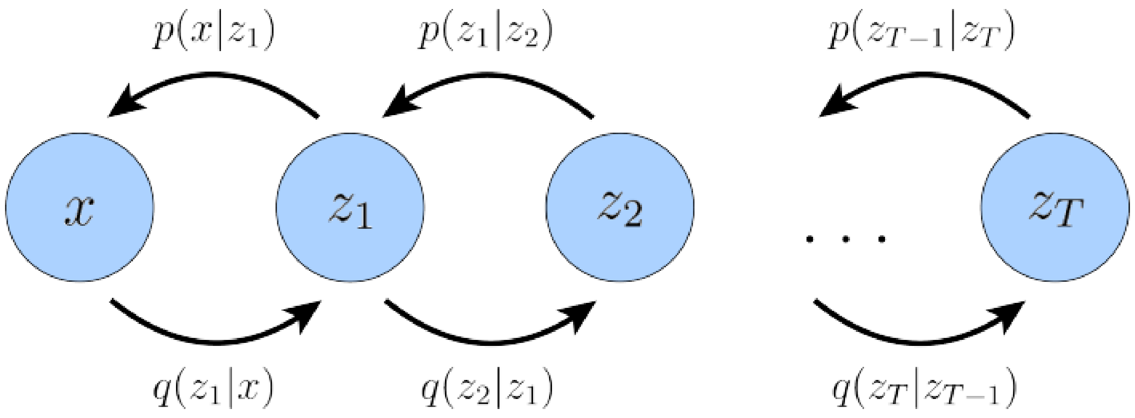
Ideal Latent:
z



3. Diffusion Introduction

2) Hierarchical VAE

계층적 latent $z_1, z_2, \ldots z_T$



원래 ELBO

$$\log p_\theta(x) \geq ELBO = E_{q_\phi(z|x)}[\log \frac{p(x,z)}{q_\phi(z|x)}]$$

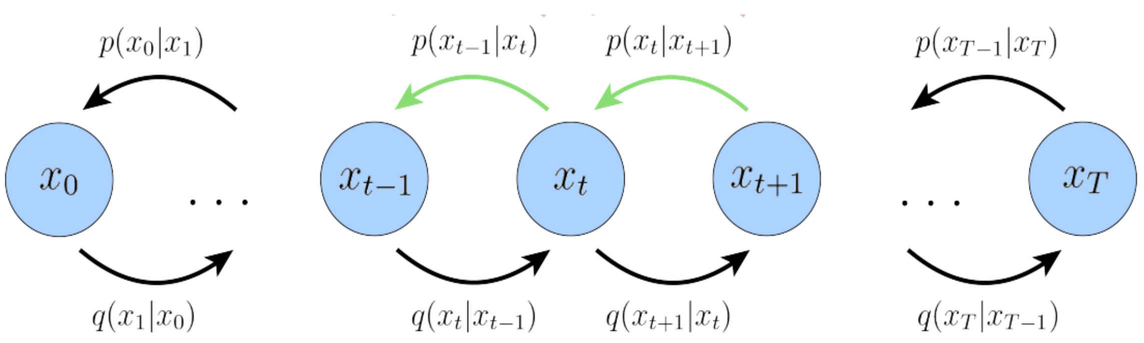
HVAE ELBO

$$\log p_\theta(x) \geq ELBO = E_{q_\phi(z_{1:T}|x)}[\log \frac{p(x,z_{1:T})}{q_\phi(z_{1:T}|x)}]$$

(Joint Probability Notation: $p(z_{1:T}) = p(z_1, z_2, \ldots z_T)$)

3. Diffusion Introduction

2) Hierarchical VAE



$$\begin{aligned}\log p(\boldsymbol{x}) &\geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right] \\ &= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\ &= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right]\end{aligned}$$

Markov Process: $q(x_1,\dots,x_T\,|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$

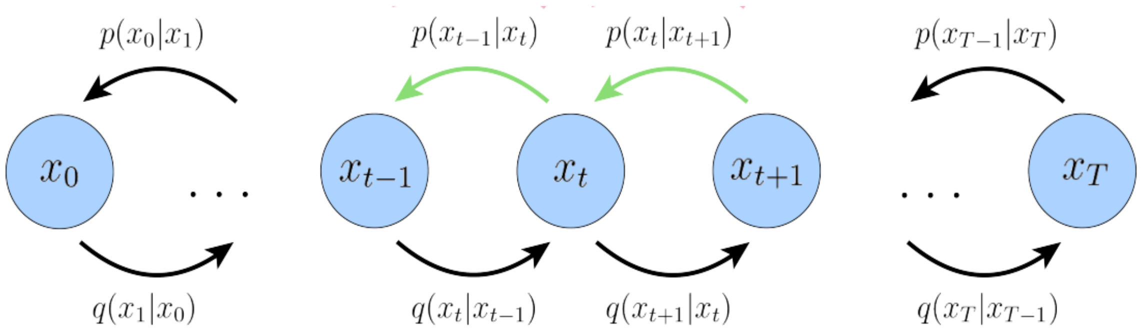
$$\underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)\parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T}\underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsym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

3. Diffusion Introduction

Modeling Process
Step1. 확률 모델링
Step2. 적절한 가정을 추가하여 만든 모델을 계산

2) Hierarchical VAE

Diffusion Model은 $q(x_t|x_{t-1})$ 가 Gaussian process 라는 가정하에 Hierarchical VAE를 풀어낸 것



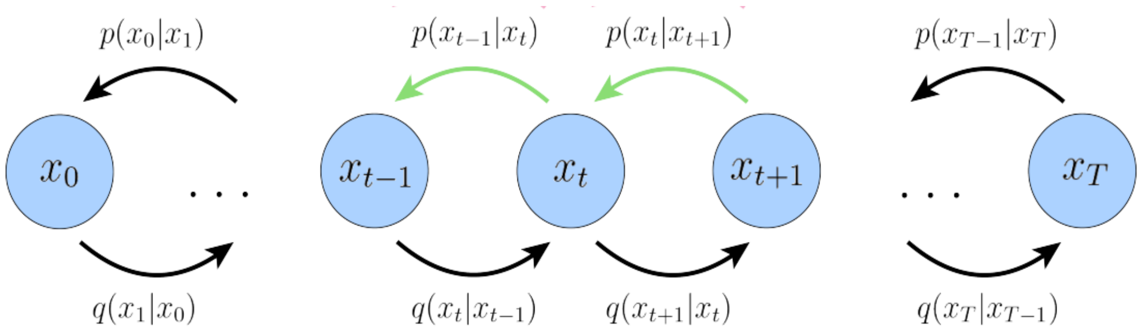
$$q(x_t|x_{t-1}) = N(x_t; \alpha x_{t-1}, \beta I)$$

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

3. Diffusion Introduction

2) Hierarchical VAE

Diffusion Model은 $q(x_t|x_{t-1})$ 가 Gaussian process 라는 가정하에 Hierarchical VAE를 풀어낸 것



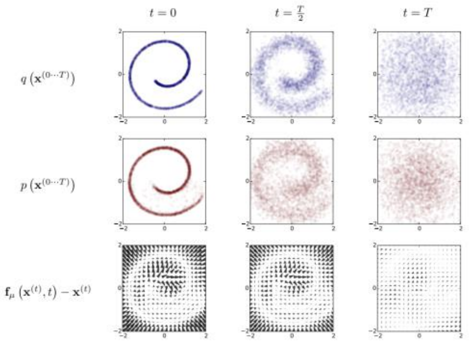
$$q(x_t|x_{t-1}) = N(x_t; \alpha x_{t-1}, \sigma^2 I)$$

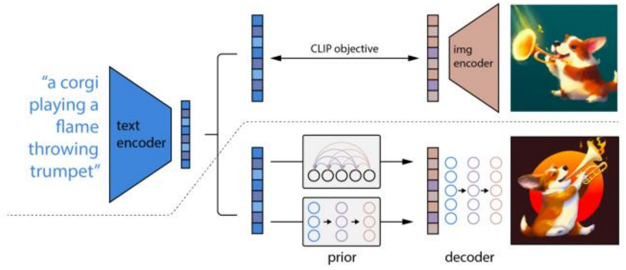
Not enough!
아래 ELBO 식을 계산할 수 없음

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

3. Diffusion Introduction

3) Paper TimeLine





DDPM
좋은 성능을 내도록 개선

Diffusion Models Beat GAN
Sota 성능 검증

DallE.2(OpenAI)
Text-to-Image

2015

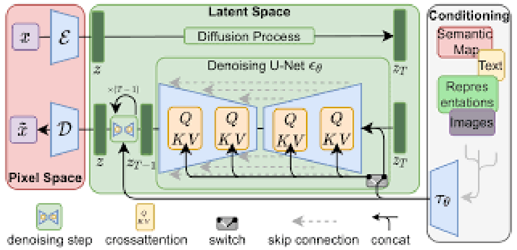
2020

2021

2022

Diffusion 제안
Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Stable Diffusion
MultiModal Performance!
(e.g. Text-to-Image)



Imagen(Google)
Text-to-Image

4. DDPM

4. DDPM

Denoising Diffusion Probabilistic Model

0) What DDPM did?

1) Forward Diffusion(Noising) Process

2) Reverse Denoising Process

3) Loss

4) Network Architecture

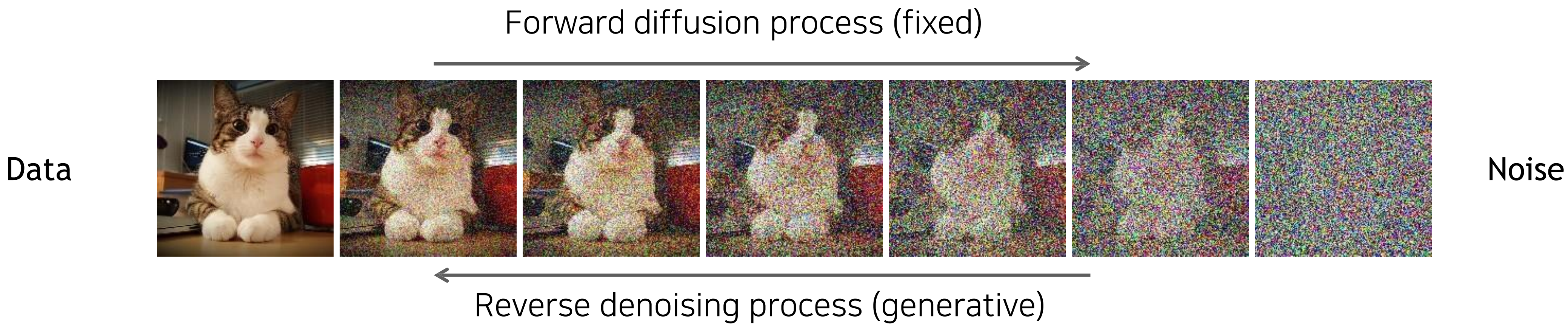
4. DDPM

Denoising Diffusion Probabilistic Model

What DDPM did?

1. Forward Diffusion Process: One-step Noising
2. Reverse Denoising Process: Gaussian임을 증명 + Loss 계산 간략화

⇒ 위 두 개를 바탕으로 Diffusion을 실질적 생성 모델로 쓸 수 있음을 보여줌



4. DDPM

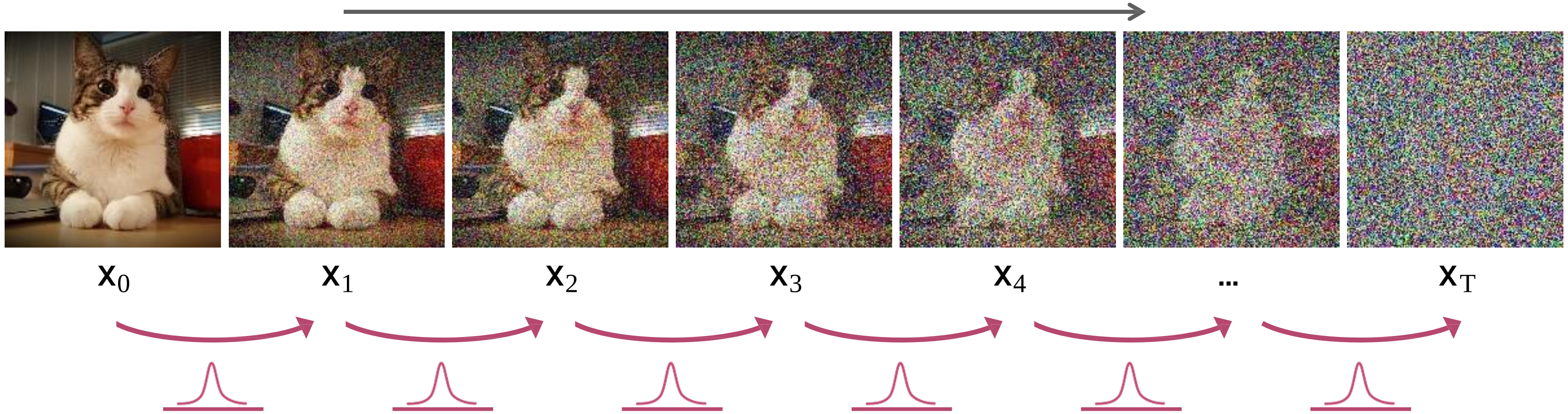
1) Forward Diffusion Process: one-step noising

Forward Process(Encoding)를 고정! 학습 필요 X

β_t는 Hyper Parameter

Forward diffusion process (fixed)

Data



Noise

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

4. DDPM

1) Forward Diffusion Process: one-step noising

$q(x_{t:1}|x_0) = q(x_t|x_{t-1})q(x_{t-1}|x_{t-2}) \dots q(x_1|x_0)$

기존에는, x_0로부터 t step 이후의 x_t 를 얻기 위해서 t 번 Gaussian Sampling을 계산 했어야 했다.

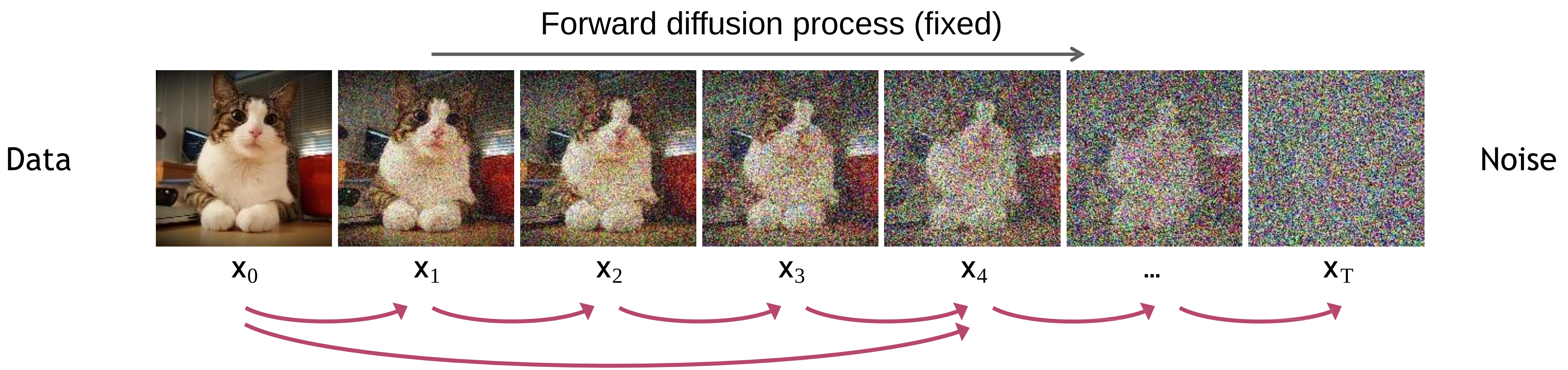
DDPM 에서는 mean = $\sqrt{1-\beta}\, x_{t-1}$, std = $\sqrt{\beta I}$ 으로 설정하여 one – step noising $(q(x_t|x_0))$ 가능하게 함

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

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

4. DDPM

1) Forward Diffusion Process: one-step noising



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

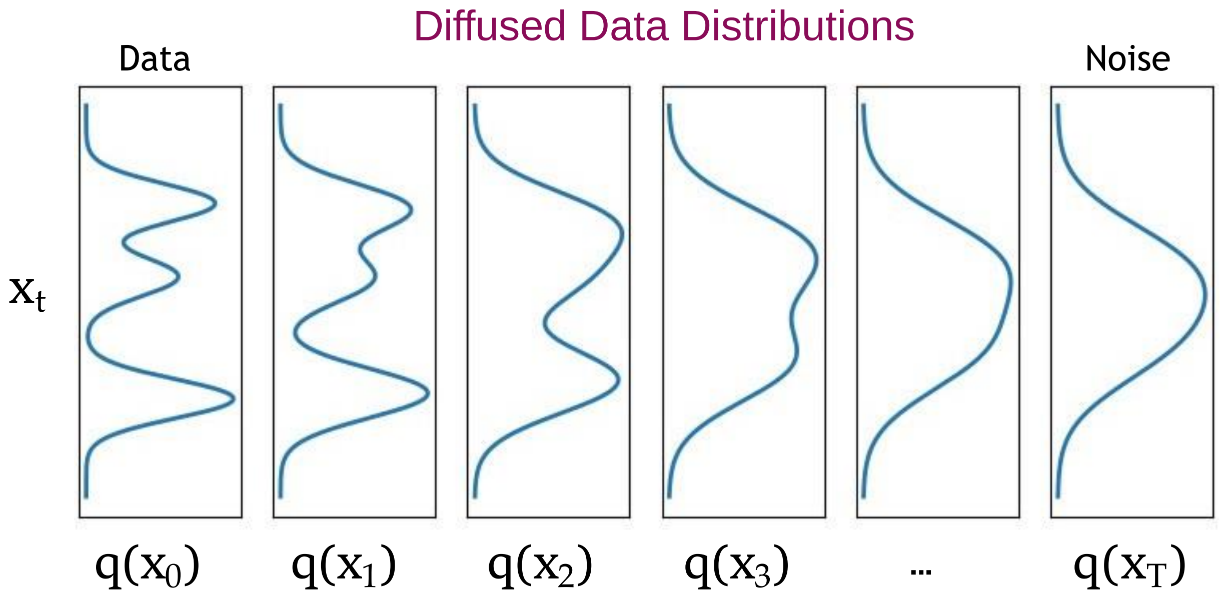
β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

4. DDPM

What happens to a distribution in step t?

So far, we discussed the diffusion kernel $q(\mathbf{x}_t|\mathbf{x}_0)$ but what about $q(\mathbf{x}_t)$?

$$\underbrace{q(\mathbf{x}_t)}_{\text{Diffused data dist.}} = \int \underbrace{q(\mathbf{x}_0, \mathbf{x}_t)}_{\text{Joint dist.}} \, d\mathbf{x}_0 = \int \underbrace{q(\mathbf{x}_0)}_{\text{Input data dist}} \, \underbrace{q(\mathbf{x}_t|\mathbf{x}_0)}_{\text{Diffusion kernel}} \, d\mathbf{x}_0$$



The diffusion kernel is Gaussian convolution.

We can sample $\mathbf{x}_t \sim q(\mathbf{x}_t)$ by first sampling $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ and then sampling $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$ (i.e., ancestral sampling).

2) Reverse Denoising Process

Denoising Process $q(x_{-1}|x_t, x_0)$ is Gaussian!

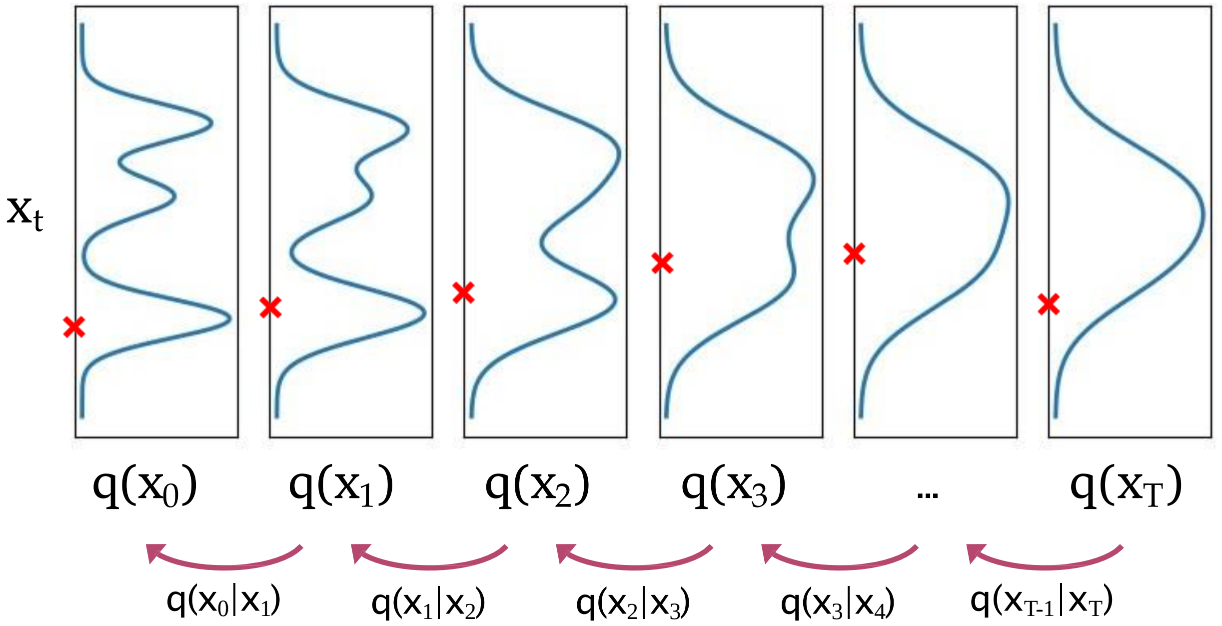
In general, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is intractable.

But!

$$q(x_t|x_{t-1}, x_0) \propto \mathcal{N}(\boldsymbol{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}}_{\mu_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}}_{\boldsymbol{\Sigma}_q(t)})$$

Why? → Appendix

Diffused Data Distributions

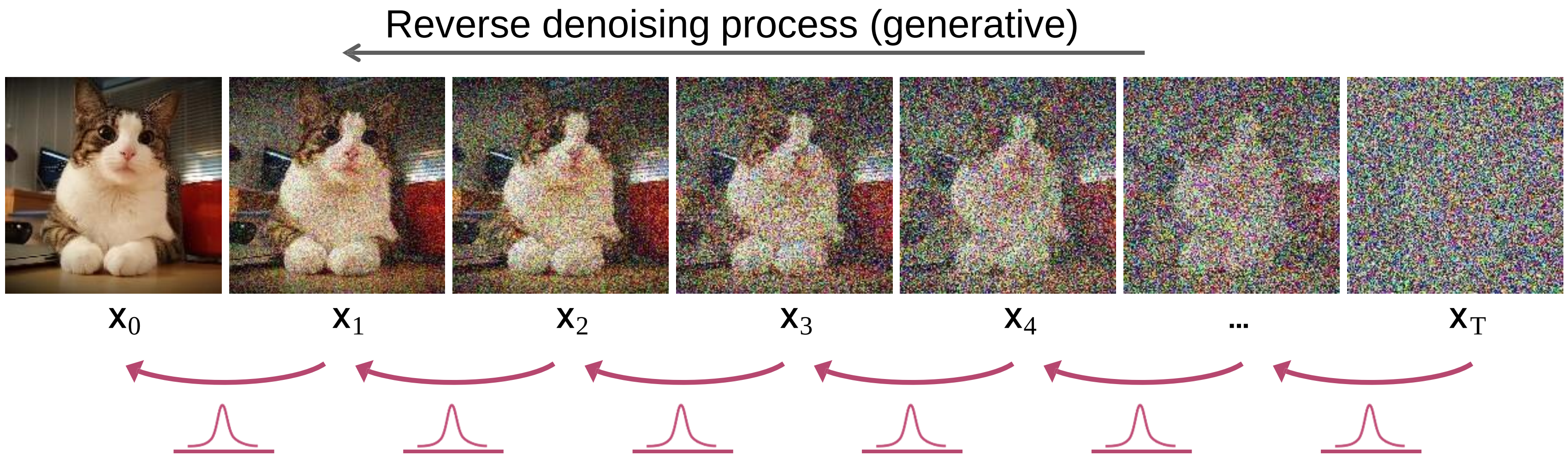


4. DDPM

2) Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:

Data



Noise

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{\substack{\text{Trainable network} \\ \text{(U-net, Denoising Autoencoder)}}}, \sigma_t^2\mathbf{I})$$

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

4. DDPM

3) Loss

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

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

Recall that $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\ \epsilon$

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

x_0를 x_t, ϵ으로 표현하면, noise ϵ 만 예측하는 것으로 task가 바뀐다.

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

With this parameterization

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

4. DDPM

3) Loss, Generating(Sampling)

결국, step 마다 첨가되었던 random noise 'ϵ' 을 예측하는 것 → 수식전개의 결과가 직관에 더 가까움

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

Algorithm 1 Training

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

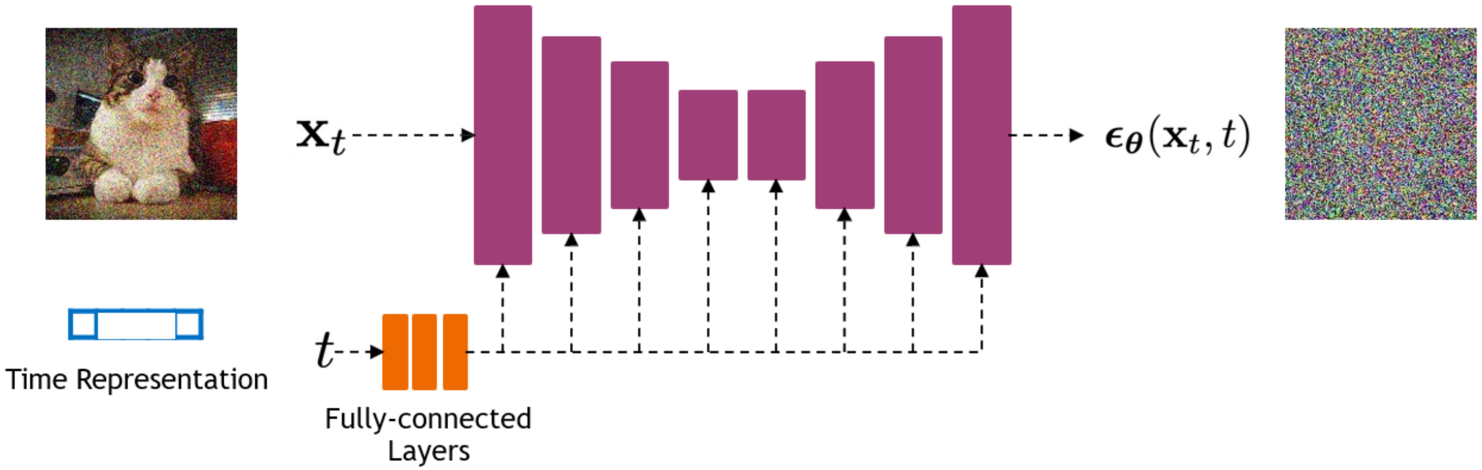
Algorithm 2 Sampling

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

4) Network Architecture

Input: step t, 노이즈 낀 이미지 x_t

Output: 첨가되었던 noise → x_{t-1}로의 Reverse Gaussian Process의 평균 계산



Diffusion models often use U-Net architectures with ResNet blocks and self-attention layers to represent $\epsilon_\theta(\mathbf{x}_t, t)$

Time representation: sinusoidal positional embeddings or random Fourier features.

Time features are fed to the residual blocks using either simple spatial addition or using adaptive group normalization layers.
(see Dharivwal and Nichol NeurIPS 2021)

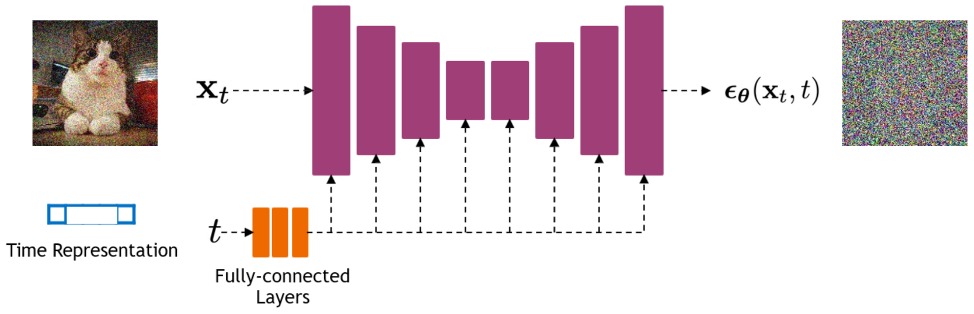
4. DDPM

4) Network Architecture

Generation Process

1) Step 마다 첨가되었던 noise를 예측

2) 예측된 noise로 Reverse Gaussian Process의 평균을 계산하여 Gaussian Process 진행



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

4. DDPM

DDPM Summary

1) Forward Diffusion Process

- Gaussian Process를 적절히 가정하여 One-step noising $q(x_t|x_0)$ 을 가능하게 함

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

2) Reverse Denoising Process

- Reverse Process가 Gaussian 임을 계산

3) Loss

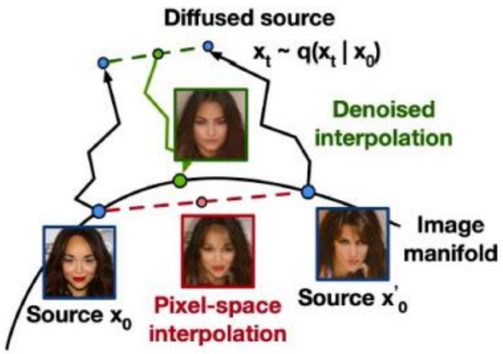
- ELBO Loss term을 첨가되었던 random noise를 예측하는 것으로 단순화

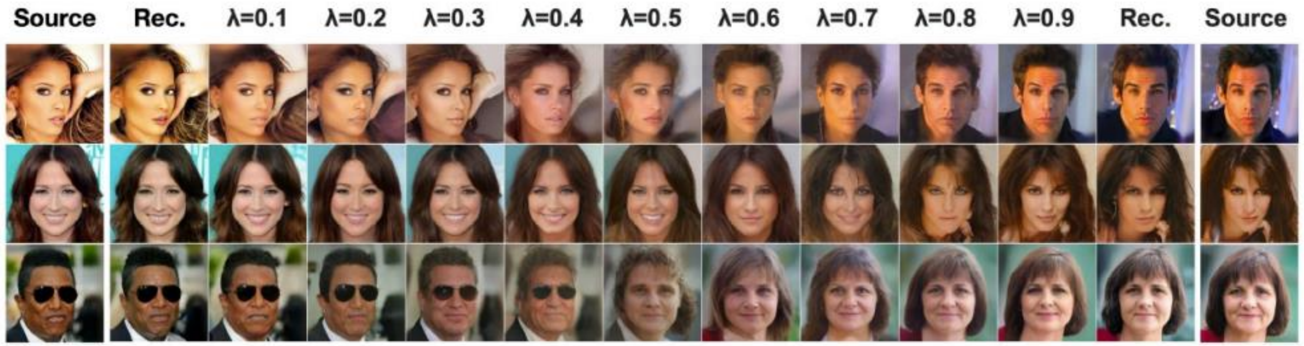
4) Network Architecture

- 노이즈가 첨가된 결과인 x_t로부터, 첨가되었던 Random Noise 를 예측하는 U-Net

4. DDPM

Generation Results





5. Score-Based Generative Model

5. Score-Based Generative Model

With Stochastic Differential Equation

Consider the limit of **many small steps:**

Forward diffusion process (fixed)

Data  Noise

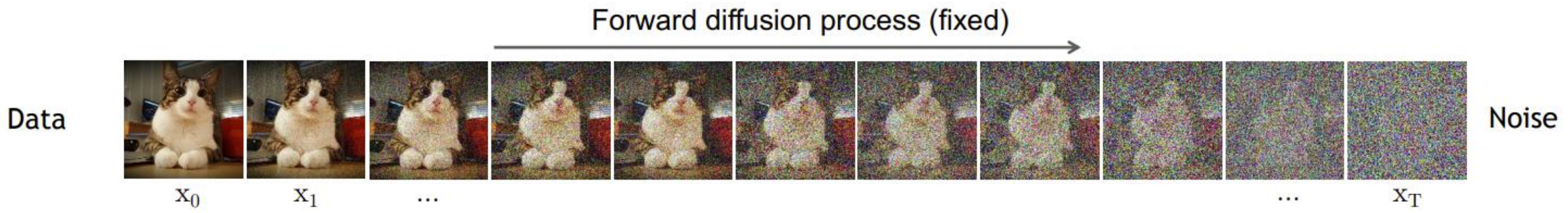
x_0 x_1 $\cdots$ $\cdots$ x_T

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

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \\
&= \sqrt{1-\beta(t)\Delta t}\,\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && (\beta_t := \beta(t)\Delta t) \\
\Rightarrow \quad &\approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && \text{(Taylor expansion)}
\end{aligned}$$

With Stochastic Differential Equation

Consider the limit of **many small steps:**

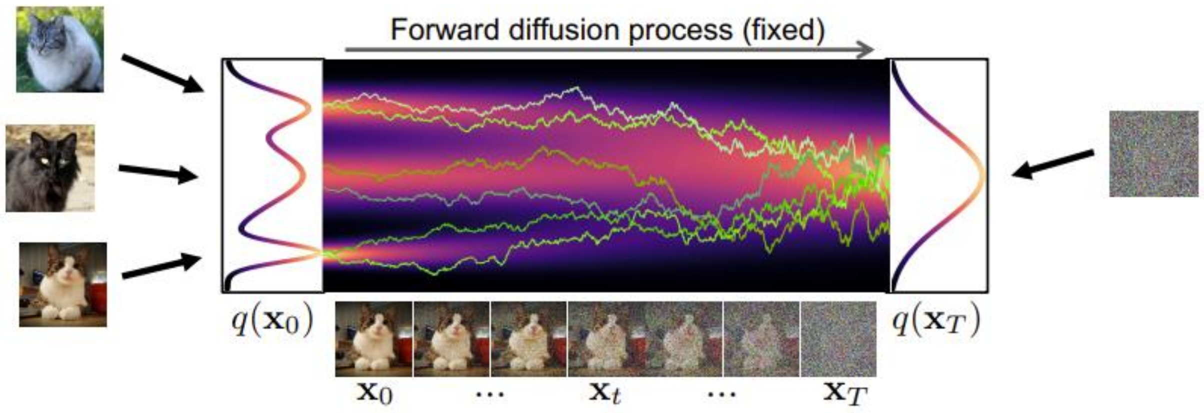


$$\mathbf{x}_t \approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Stochastic Differential Equation (SDE)
describing the diffusion in infinitesimal limit

With Stochastic Differential Equation



Forward Diffusion SDE:

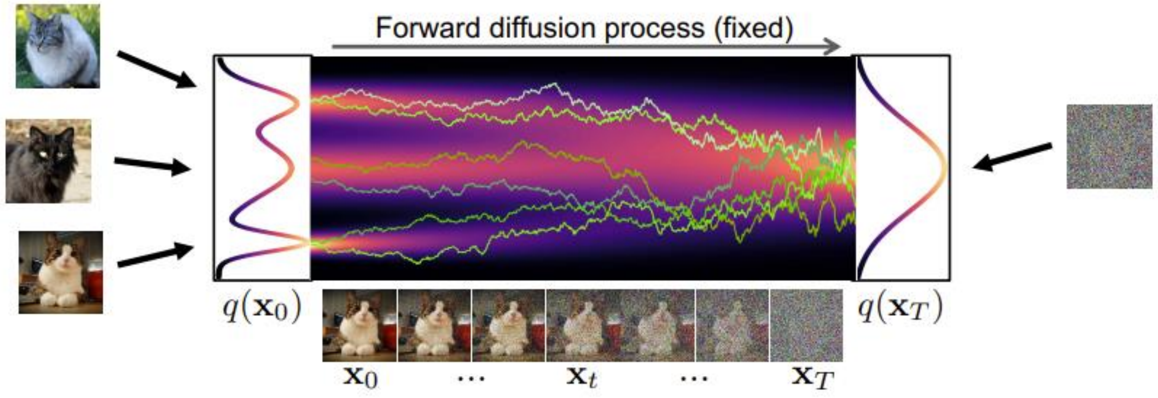
$$\mathrm{d}\mathbf{x}_t = \underbrace{-\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t}_{\substack{\text{drift term} \\ \text{(pulls towards mode)}}} + \underbrace{\sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t}_{\substack{\text{diffusion term} \\ \text{(injects noise)}}}$$

Special case of more general SDEs used in generative diffusion models:

$$\mathrm{d}\mathbf{x}_t = f(t)\mathbf{x}_t\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\omega}_t$$

5. Score-Based Generative Model

With Stochastic Differential Equation



Forward Diffusion SDE:

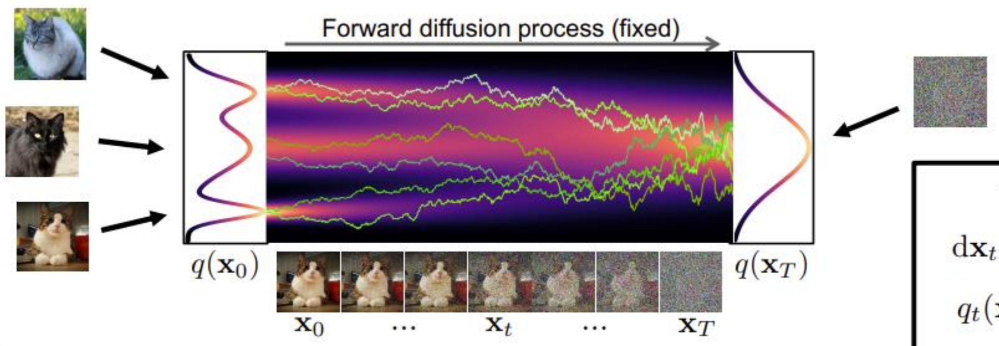
$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,dt + \sqrt{\beta(t)}\,d\boldsymbol{\omega}_t$$

Reverse Generative Diffusion SDE:

$$d\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}} dt + \overbrace{\sqrt{\beta(t)}\,d\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

With Stochastic Differential Equation

Stochastic 미분방정식으로 Diffusion 과정을 정의하고, 풀어냈더니 DDPM과 같은 Loss 식 유도됨!



"Variance Preserving" SDE:

$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,dt + \sqrt{\beta(t)}\,d\boldsymbol{\omega}_t$$

$$q_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \gamma_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$$

$$\gamma_t = e^{-\frac{1}{2}\int_0^t \beta(s)ds}$$

$$\sigma_t^2 = 1 - e^{-\int_0^t \beta(s)ds}$$

- **Denoising Score Matching:**

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)}||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\mathbf{x}_0)||_2^2$$

  - Re-parametrized sampling: $\mathbf{x}_t = \gamma_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
  - Score function: $\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\mathbf{x}_0) = -\nabla_{\mathbf{x}_t}\frac{(\mathbf{x}_t-\gamma_t\mathbf{x}_0)^2}{2\sigma_t^2} = -\frac{\mathbf{x}_t-\gamma_t\mathbf{x}_0}{\sigma_t^2} = -\frac{\gamma_t\mathbf{x}_0+\sigma_t\boldsymbol{\epsilon}-\gamma_t\mathbf{x}_0}{\sigma_t^2} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$
  - Neural network model: $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) := -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sigma_t}$

$$\Rightarrow \min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\frac{1}{\sigma_t^2}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)||_2^2$$

DATA
SCIENCE LAB

6. Guidance/Conditional Diffusion

6. Guidance/Conditional Diffusion

1) Classifier Guidance Diffusion

2) Classifier-free Guidance Diffusion

$$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}}\mathrm{d}t + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

둘 다 결론은 직관적이지만, 유도는 Score function 으로부터

6. Guidance/Conditional Diffusion

1) Classifier Guidance Diffusion

Idea: 각 Step 마다 Classifier, 그에 대한 loss 계산 후 gradient 반영

Idea의 수학적 배경:

$$\mathrm{d}\mathbf{x} = \left[\boldsymbol{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \mathbf{y})}\right] \mathrm{d}t + g(t)\, \mathrm{d}\mathbf{w}$$

$$\mathrm{d}\mathbf{x} = \left[\boldsymbol{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})} - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{y} \mid \mathbf{x})}\right] \mathrm{d}t + g(t)\, \mathrm{d}\mathbf{w}$$

$$\begin{aligned}
\nabla \log p(\boldsymbol{x}_t|y) &= \nabla \log \left(\frac{p(\boldsymbol{x}_t)p(y|\boldsymbol{x}_t)}{p(y)}\right) \\
&= \nabla \log p(\boldsymbol{x}_t) + \nabla \log p(y|\boldsymbol{x}_t) - \nabla \log p(y) \\
&= \underbrace{\nabla \log p(\boldsymbol{x}_t)}_{\text{unconditional score}} + \underbrace{\nabla \log p(y|\boldsymbol{x}_t)}_{\text{adversarial gradient}}
\end{aligned}$$

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

```
Input: class label y, gradient scale s
x_T ← sample from N(0, I)
for all t from T to 1 do
    μ, Σ ← μ_θ(x_t), Σ_θ(x_t)
    x_{t−1} ← sample from N(μ + sΣ ∇_{x_t} log p_φ(y|x_t), Σ)
end for
return x_0
```

Score model

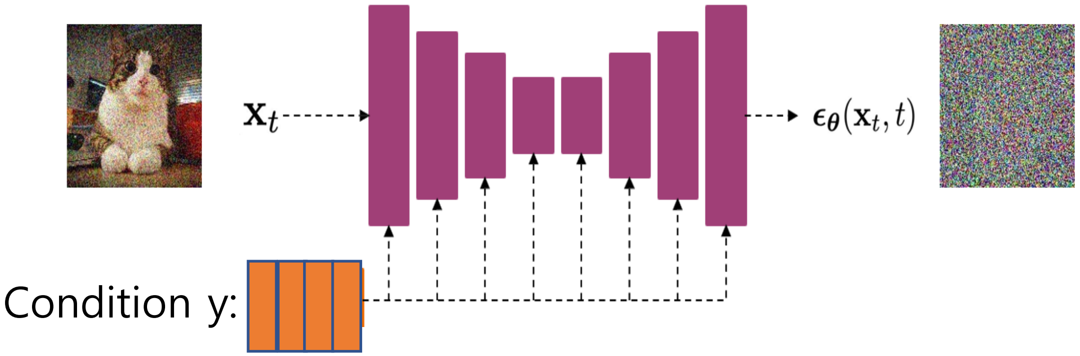
Classifier gradient

6. Guidance/Conditional Diffusion

2) Classifier-free Guidance Diffusion

Idea $\epsilon_\theta(x_t, c)$: Noise 예측 network에 Condition을 input에 같이 넣어준다!

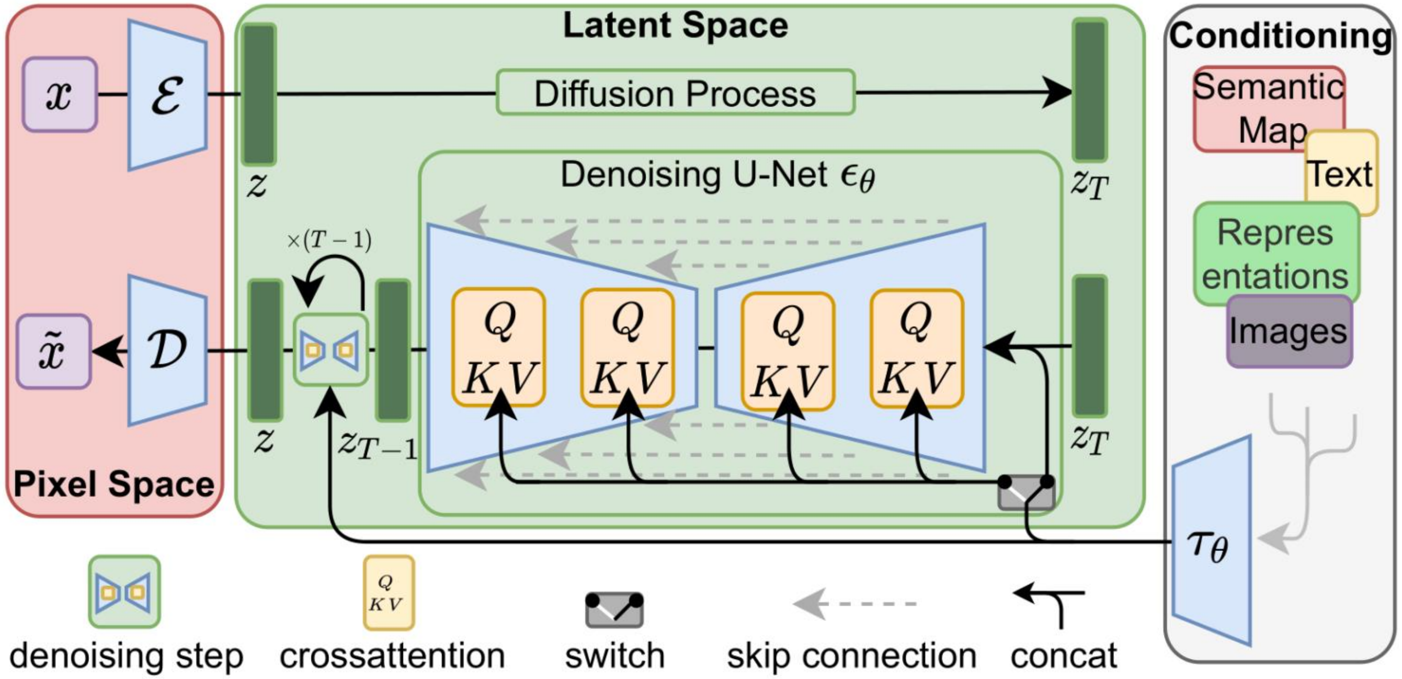
핵심: 이렇게 Condition을 주는 것과, Classifier Guidance 방식이 수학적으로 동일!



2) Classifier-free Guidance Diffusion

Idea $\epsilon_\theta(x_t, c)$: Noise 예측 network에 Condition을 input에 같이 넣어준다!

Stable Diffusion



Reference

[Tutorial]

- CVPR Diffusion Tutorial: https://cvpr2022-tutorial-diffusion-models.github.io/

  (Diffusion Slide는 CVPR Diffusion Tutorial의 slide를 활용하였습니다.)

[Paper]

- Understanding Diffusion Models: A Unified Perspective
- Denoising diffusion probabilistic models (DDPM)

[Youtube]

- KL-Divergence: https://youtu.be/9_eZHt2qJs4
- '권민기' 님의 Diffusion 발표영상: https://youtu.be/uFoGaIVHfoE

Appendix

Hierarchical VAE: ELBO 식 전개

$$
\begin{aligned}
\log p(\boldsymbol{x}) &\geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}\right]
\end{aligned}
$$

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

Appendix

DDPM: t-step noising 식 유도 (where $\mu = \sqrt{\alpha_t} x_{t-1},\ \sigma = \sqrt{1-\alpha_t I}$)

Then, the form of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ can be recursively derived through repeated applications of the reparameterization trick. Suppose that we have access to $2T$ random noise variables $\{\boldsymbol{\epsilon}_t^*, \boldsymbol{\epsilon}_t\}_{t=0}^{T} \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$. Then, for an arbitrary sample $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$, we can rewrite it as:

$$
\begin{aligned}
\boldsymbol{x}_t &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}^* && (61)\\
&= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}^*\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}^* && (62)\\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}^* + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}^* && (63)\\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}^2 + \sqrt{1-\alpha_t}^2}\boldsymbol{\epsilon}_{t-2} && (64)\\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1} + 1 - \alpha_t}\boldsymbol{\epsilon}_{t-2} && (65)\\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} && (66)\\
&= \ldots && (67)\\
&= \sqrt{\prod_{i=1}^{t}\alpha_i}\boldsymbol{x}_0 + \sqrt{1 - \prod_{i=1}^{t}\alpha_i}\boldsymbol{\epsilon}_0 && (68)\\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0 && (69)\\
&\sim \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) && (70)
\end{aligned}
$$

where in Equation 64 we have utilized the fact that the sum of two independent Gaussian random variables remains a Gaussian with mean being the sum of the two means, and variance being the sum of the two variances. Interpreting $\sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}^*$ as a sample from Gaussian $\mathcal{N}(\mathbf{0}, (1-\alpha_t)\mathbf{I})$, and $\sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}^*$ as a sample from Gaussian $\mathcal{N}(\mathbf{0}, (\alpha_t - \alpha_t\alpha_{t-1})\mathbf{I})$, we can then treat their sum as a random variable sampled from Gaussian $\mathcal{N}(\mathbf{0}, (1 - \alpha_t + \alpha_t - \alpha_t\alpha_{t-1})\mathbf{I}) = \mathcal{N}(\mathbf{0}, (1-\alpha_t\alpha_{t-1})\mathbf{I})$. A sample from this distribution can then be represented using the reparameterization trick as $\sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}$, as in Equation 66.

Appendix

DDPM:

Reverse Denoising Process가

Gaussian 임을 증명

$$
\begin{aligned}
q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) &= \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} && (71)\\
&= \frac{\mathcal{N}(\boldsymbol{x}_t;\sqrt{\alpha_t}\boldsymbol{x}_{t-1},(1-\alpha_t)\mathbf{I})\mathcal{N}(\boldsymbol{x}_{t-1};\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0,(1-\bar\alpha_{t-1})\mathbf{I})}{\mathcal{N}(\boldsymbol{x}_t;\sqrt{\bar\alpha_t}\boldsymbol{x}_0,(1-\bar\alpha_t)\mathbf{I})} && (72)\\
&\propto \exp\left\{-\left[\frac{(\boldsymbol{x}_t-\sqrt{\alpha_t}\boldsymbol{x}_{t-1})^2}{2(1-\alpha_t)}+\frac{(\boldsymbol{x}_{t-1}-\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0)^2}{2(1-\bar\alpha_{t-1})}-\frac{(\boldsymbol{x}_t-\sqrt{\bar\alpha_t}\boldsymbol{x}_0)^2}{2(1-\bar\alpha_t)}\right]\right\} && (73)\\
&= \exp\left\{-\frac{1}{2}\left[\frac{(\boldsymbol{x}_t-\sqrt{\alpha_t}\boldsymbol{x}_{t-1})^2}{1-\alpha_t}+\frac{(\boldsymbol{x}_{t-1}-\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0)^2}{1-\bar\alpha_{t-1}}-\frac{(\boldsymbol{x}_t-\sqrt{\bar\alpha_t}\boldsymbol{x}_0)^2}{1-\bar\alpha_t}\right]\right\} && (74)\\
&= \exp\left\{-\frac{1}{2}\left[\frac{(-2\sqrt{\alpha_t}\boldsymbol{x}_t\boldsymbol{x}_{t-1}+\alpha_t\boldsymbol{x}_{t-1}^2)}{1-\alpha_t}+\frac{(\boldsymbol{x}_{t-1}^2-2\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_{t-1}\boldsymbol{x}_0)}{1-\bar\alpha_{t-1}}+C(\boldsymbol{x}_t,\boldsymbol{x}_0)\right]\right\} && (75)\\
&\propto \exp\left\{-\frac{1}{2}\left[-\frac{2\sqrt{\alpha_t}\boldsymbol{x}_t\boldsymbol{x}_{t-1}}{1-\alpha_t}+\frac{\alpha_t\boldsymbol{x}_{t-1}^2}{1-\alpha_t}+\frac{\boldsymbol{x}_{t-1}^2}{1-\bar\alpha_{t-1}}-\frac{2\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_{t-1}\boldsymbol{x}_0}{1-\bar\alpha_{t-1}}\right]\right\} && (76)\\
&= \exp\left\{-\frac{1}{2}\left[(\frac{\alpha_t}{1-\alpha_t}+\frac{1}{1-\bar\alpha_{t-1}})\boldsymbol{x}_{t-1}^2-2\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0}{1-\bar\alpha_{t-1}}\right)\boldsymbol{x}_{t-1}\right]\right\} && (77)\\
&= \exp\left\{-\frac{1}{2}\left[\frac{\alpha_t(1-\bar\alpha_{t-1})+1-\alpha_t}{(1-\alpha_t)(1-\bar\alpha_{t-1})}\boldsymbol{x}_{t-1}^2-2\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0}{1-\bar\alpha_{t-1}}\right)\boldsymbol{x}_{t-1}\right]\right\} && (78)\\
&= \exp\left\{-\frac{1}{2}\left[\frac{\alpha_t-\bar\alpha_t+1-\alpha_t}{(1-\alpha_t)(1-\bar\alpha_{t-1})}\boldsymbol{x}_{t-1}^2-2\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0}{1-\bar\alpha_{t-1}}\right)\boldsymbol{x}_{t-1}\right]\right\} && (79)\\
&= \exp\left\{-\frac{1}{2}\left[\frac{1-\bar\alpha_t}{(1-\alpha_t)(1-\bar\alpha_{t-1})}\boldsymbol{x}_{t-1}^2-2\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0}{1-\bar\alpha_{t-1}}\right)\boldsymbol{x}_{t-1}\right]\right\} && (80)\\
&= \exp\left\{-\frac{1}{2}\left(\frac{1-\bar\alpha_t}{(1-\alpha_t)(1-\bar\alpha_{t-1})}\right)\left[\boldsymbol{x}_{t-1}^2-2\frac{\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0}{1-\bar\alpha_{t-1}}\right)}{\frac{1-\bar\alpha_t}{(1-\alpha_t)(1-\bar\alpha_{t-1})}}\boldsymbol{x}_{t-1}\right]\right\} && (81)\\
&= \exp\left\{-\frac{1}{2}\left(\frac{1-\bar\alpha_t}{(1-\alpha_t)(1-\bar\alpha_{t-1})}\right)\left[\boldsymbol{x}_{t-1}^2-2\frac{\left(\frac{\sqrt{\alpha_t}\boldsymbol{x}_t}{1-\alpha_t}+\frac{\sqrt{\bar\alpha_{t-1}}\boldsymbol{x}_0}{1-\bar\alpha_{t-1}}\right)(1-\alpha_t)(1-\bar\alpha_{t-1})}{1-\bar\alpha_t}\boldsymbol{x}_{t-1}\right]\right\} && (82)\\
&= \exp\left\{-\frac{1}{2}\left(\frac{1}{\frac{(1-\alpha_t)(1-\bar\alpha_{t-1})}{1-\bar\alpha_t}}\right)\left[\boldsymbol{x}_{t-1}^2-2\frac{\sqrt{\alpha_t}(1-\bar\alpha_{t-1})\boldsymbol{x}_t+\sqrt{\bar\alpha_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar\alpha_t}\boldsymbol{x}_{t-1}\right]\right\} && (83)\\
&\propto \mathcal{N}(\boldsymbol{x}_{t-1};\underbrace{\frac{\sqrt{\alpha_t}(1-\bar\alpha_{t-1})\boldsymbol{x}_t+\sqrt{\bar\alpha_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar\alpha_t}}_{\mu_q(\boldsymbol{x}_t,\boldsymbol{x}_0)},\underbrace{\frac{(1-\alpha_t)(1-\bar\alpha_{t-1})}{1-\bar\alpha_t}\mathbf{I}}_{\boldsymbol{\Sigma}_q(t)}) && (84)
\end{aligned}
$$

DATA
SCIENCE LAB

발표자 박지호
E-mail: qkrwlgh0314@yonsei.ac.kr