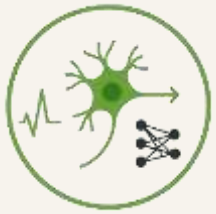
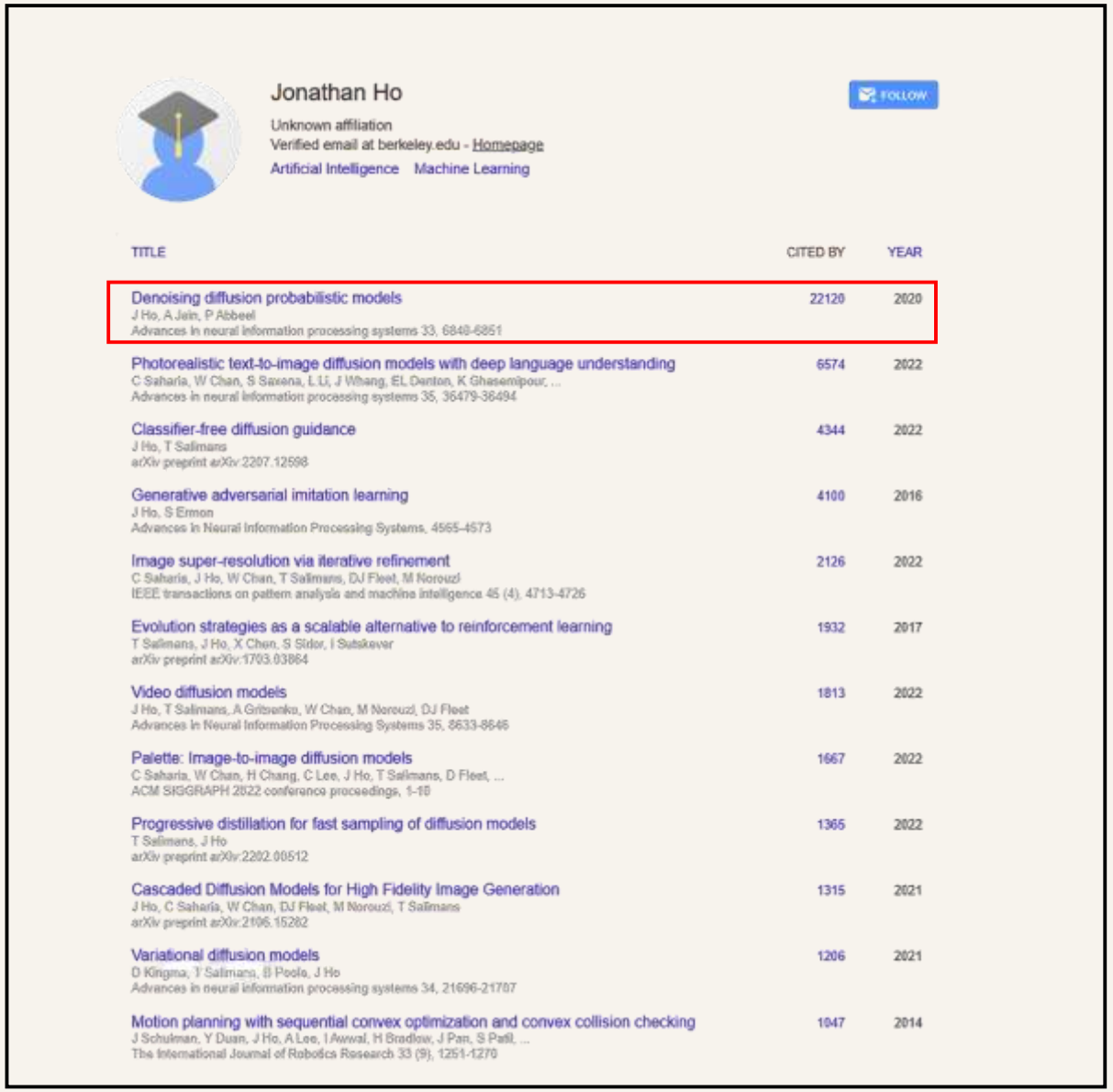
BrAIn Lab.
Brain and Artificial Intelligence Lab.



Denoising Diffusion Probabilistic Models

(2020-NeurIPS)



Jonathan Ho

FOLLOW

Unknown affiliation
Verified email at berkeley.edu - Homepage
Artificial Intelligence Machine Learning

TITLE	CITED BY	YEAR
Denoising diffusion probabilistic models J Ho, A Jain, P Abbeel Advances in neural information processing systems 33, 6840-6851	22120	2020
Photorealistic text-to-image diffusion models with deep language understanding C Saharia, W Chan, S Saxena, L Li, J Whang, EL Denton, K Ghasemipour, ... Advances in neural information processing systems 35, 36479-36494	6574	2022
Classifier-free diffusion guidance J Ho, T Salimans arXiv preprint arXiv:2207.12598	4344	2022
Generative adversarial imitation learning J Ho, S Ermon Advances in Neural Information Processing Systems, 4565-4573	4100	2016
Image super-resolution via iterative refinement C Saharia, J Ho, W Chan, T Salimans, DJ Fleet, M Norouzi IEEE transactions on pattern analysis and machine intelligence 45 (4), 4713-4726	2126	2022
Evolution strategies as a scalable alternative to reinforcement learning T Salimans, J Ho, X Chen, S Sidor, I Sutskever arXiv preprint arXiv:1703.03864	1932	2017
Video diffusion models J Ho, T Salimans, A Gritsenko, W Chan, M Norouzi, DJ Fleet Advances in Neural Information Processing Systems 35, 8633-8646	1813	2022
Palette: Image-to-image diffusion models C Saharia, W Chan, H Chang, C Lee, J Ho, T Salimans, D Fleet, ... ACM SIGGRAPH 2022 conference proceedings, 1-10	1667	2022
Progressive distillation for fast sampling of diffusion models T Salimans, J Ho arXiv preprint arXiv:2202.00512	1365	2022
Cascaded Diffusion Models for High Fidelity Image Generation J Ho, C Saharia, W Chan, DJ Fleet, M Norouzi, T Salimans arXiv preprint arXiv:2106.15282	1315	2021
Variational diffusion models D Kingma, T Salimans, B Poole, J Ho Advances in neural information processing systems 34, 21696-21707	1206	2021
Motion planning with sequential convex optimization and convex collision checking J Schulman, Y Duan, J Ho, A Lee, I Awwal, H Bradlow, J Pan, S Patil, ... The International Journal of Robotics Research 33 (9), 1251-1270	1047	2014

Contents.

BrAIn Lab.
Brain and Artificial Intelligence Lab.

1 Background

Background

Markov Chain

- 시간에 따라서 상태가 변하는 시스템을 모델링하는 방법
- Markov Property : 특정 시점 t+1 에서의 상태는 오직 바로 직전시점 t 상태에만 의존하고 이전 시점과는 독립적이다.

$$P(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}) = P(X_{t+1} = x_{t+1} | X_t = x_t)$$

KL-Divergence

- 두 확률분포 P, Q가 얼마나 다른지를 측정하는 척도

$$KL(P||Q) = E_p[log\frac{P(x)}{Q(x)}] = -\sum P(x) log\frac{P(x)}{Q(x)} = H(P,Q) - H(P)$$

→ P, Q의 Cross-Entropy

MLE(Maximum Likelihood Estimation)

- Likelihood : 주어진 데이터 X가 특정 확률분포로부터 나왔을 가능성을 나타내는 값
- MLE : 관측된 데이터 X를 가장 잘 설명하는(Likelihood를 최대로 만드는) θ를 찾는 방법

Background

Bayes Rule

- Bayes Rule : 주어진 정보를 바탕으로 사전 확률을 업데이트하여 사후 확률을 얻는 방법

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- $P(A)$: 사전확률 (사건 B에 대한 정보 없이 사건 A가 발생할 확률)
- $P(B)$: 사건 B가 발생할 확률
- $P(B|A)$: likelihood
- $P(A|B)$: 사후 확률

→ 순방향 과정을 바탕으로 역방향을 추론한다.

확률의 연쇄법칙

- 확률의 변수가 여러 개일때 확률을 조건부 확률의 곱으로 분해하는 방법

$$P(X_1, X_2, \ldots, X_n) = P(X1) * P(X2|X1) * P(X_3|X_1, X_2) * \cdots * P(X_n|X_{1,\ldots}X_{n-1})$$

2 Diffusion

Diffusion

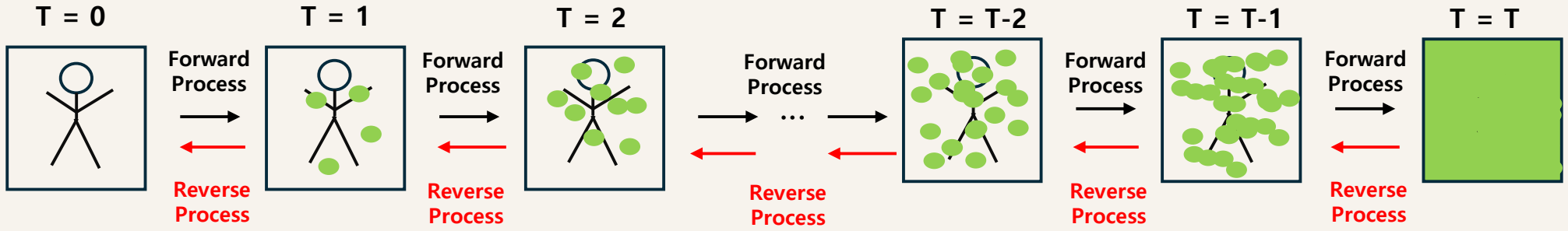
Overview

- 잉크 분자들은 시간이 지남에 따라서 물 전체에 고르게 퍼져(diffusion) 나감 (Uniform 분포)
- 잉크 분자들이 움직이는 단계를 확대해서 보면 움직임의 확률 분포는 Gaussian 분포를 따르는것으로 근사할 수 있음
- 잉크 분자들의 움직임의 확률분포를 알 수 있다면 다시 되돌리는것도 가능함
- 따라서 움직임의 확률분포를 Neural Net으로 근사하여 해당 확률분포를 학습하는것이 Diffusion의 목표

Diffusion

Overview

- 패턴을 학습하기 위해 고의적으로 패턴을 붕괴 (Nosing), 이를 다시 복원하도록(reverse) Neurl Network를 학습
- Nosing 과정: Forward(Diffusion) Process (가우시안 노이즈를 추가)
- Denosing 과정 : Reverse Process

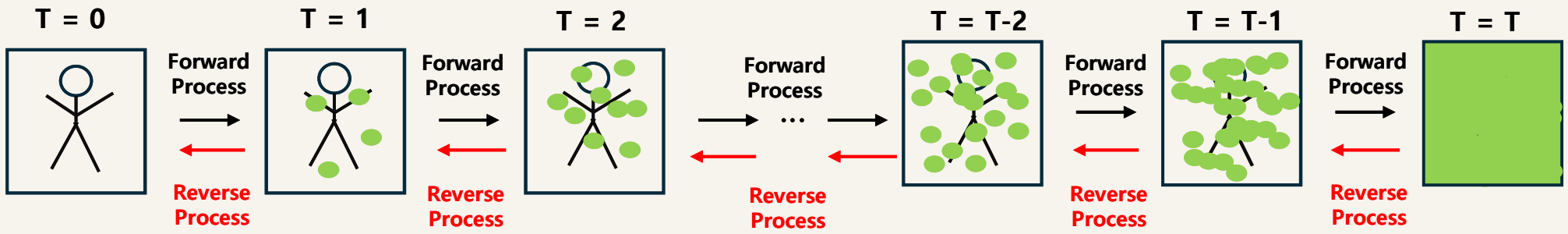


- Forward process : $q(X_{1:T}|X_o) \coloneqq \prod_{t=1}^{T} q(X_t|X_{t-1})$ → 가우시안 노이즈를 추가하는 과정
- Reverse process : $P_\theta(X_{0:T}) \coloneqq P(X_T) \prod_{t=1}^{T} P_\theta(X_{t-1}|X_t)$ → 노이즈를 제거하여 원본을 복원하는 과정

Goal : 확률 분포 q에서 관측한 값으로 P_θ의 likelihood를 구했을때 그 likelihood가 최대가 되도록 하는 θ를 찾는것

3 Diffusion Process

Diffusion Process



- Forward process : $q(X_{1:T}|X_o) \coloneqq \boxed{\prod_{t=1}^{T} q(X_t|X_{t-1})}$ → 가우시안 노이즈를 추가하는 과정

  → 노이즈를 추가하는 과정(Forward Process)은 모델 X 따라서 학습을 진행하지 않는다.

- Reverse process : $P_\theta(X_{0:T}) \coloneqq \boxed{P(X_T)\prod_{t=1}^{T} P_\theta(X_{t-1}|X_t)}$ → 노이즈를 제거하여 원본을 복원하는 과정

  → 노이즈를 제거하는 과정(Reverse Process)을 Neurl Net으로 모델링하여 학습을 진행

Forward Process

Forward Process

- 원본 이미지 X_o에 아주 조금씩 여러단계(T) 에 걸쳐 가우시안 노이즈를 추가하여 완전한 가우시안 노이즈 (X_T)로 만드는 과정
- 해당 과정은 Markov Chain에 정의되며, 각 단계의 상태 X_T 는 오직 X_{T-1} 단계에만 의존한다
- $q(X_{1:T}|X_0) = \prod_{t=1}^{T} q(X_t|X_{t-1})$
- 네트워크 정의 X -> 훈련 파라미터가 존재하지 않는다.

수식 표현

$$q(X_t|X_{t-1}) \coloneqq N(x_t; \sqrt{1-B_t}X_{t-1}, B_tI)$$

- $N(x_t; \sqrt{1-B_t}X_{t-1}, B_tI)$: 평균이 $\sqrt{1-B_t}X_{t-1}$, 분산이 B_tI 인 정규분포
- B_t : 분산 스케줄 (t 단계에서 얼마나 많은 노이즈를 추가할지 결정하는 파라미터, 보통 t가 클수록 B_t 도 커지게 설정)
- B_tI : 추가되는 노이즈의 분산

$$X_t = \sqrt{1-B_t}X_{t-1} + \sqrt{B_t}z_{t-1}$$

- 이전 시점 (t-1) 에서 현재 시점(t) 이미지(노이즈를 섞은) 를 만드는 과정
- z_{t-1} : $N(0, I)$에서 샘플링 된 가우시안 노이즈

Forward Process

Forward Process

$$X_t = \sqrt{1 - B_t} X_{t-1} + \sqrt{B_t} z_{t-1}$$

$$X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1 - \alpha_t} z_{t-1} \quad (\alpha_t := 1 - B_t)$$

- 이전 시점 (t-1) 에서 현재 시점(t) 이미지(노이즈를 섞은) 를 만드는 과정
- z_{t-1} : $N(0, I)$에서 샘플링 된 노이즈
- 이렇게 진행할 시 0부터 T 까지 모든 스텝을 진행해야하므로 샘플링 시간이 매우 오래걸림
- 따라서 DDPM은 중간 단계를 거치지 않고 원본이미지 X_0로부터 임의의 시점 t의 X_t를 한 번에 샘플링 할 수 있게 함(학습의 효율)

Forward Process

증명

1. $\alpha_t := 1 - B_t \ (0 < B_t < 1)$, $\bar{a}_t := \prod_{s=1}^{t} \alpha_s$
2. $X_t = \sqrt{\alpha_t} X_{t-1} + \sqrt{1-\alpha_t} z_{t-1}$
3. $X_1 = \sqrt{\alpha_1} X_0 + \sqrt{1-\alpha_1} z_0$
4. $X_2 = \sqrt{\alpha_2} X_1 + \sqrt{1-\alpha_2} z_1$
5. X_1를 X_2에 대입 : $X_2 = \sqrt{\alpha_2 \alpha_1} X_0 + \sqrt{\alpha_2(1-\alpha_0)} z_0 + \sqrt{1-\alpha_2} z_1$

   각 가우시안 분포 z_0, z_1에서 샘플링된 노이즈
6. $X_2 = \sqrt{\alpha_2 \alpha_1} X_0 + \sqrt{1-\bar{a}_2} \bar{z}_1$
7. $X_t = \sqrt{\bar{a}_t} X_0 + \sqrt{1-\bar{a}_t} \bar{z}_t$

   - X_0에서 바로 X_t 를 샘플링 가능 (샘플링 시간 단축)
   - t가 커질수록 점점 원본 이미지 X_0는 0에 가까워져 노이즈만 남는다

두 가우시안 분포 합치기

$z_0 \sim N(0, I)$, $z_1 \sim N(0, I)$

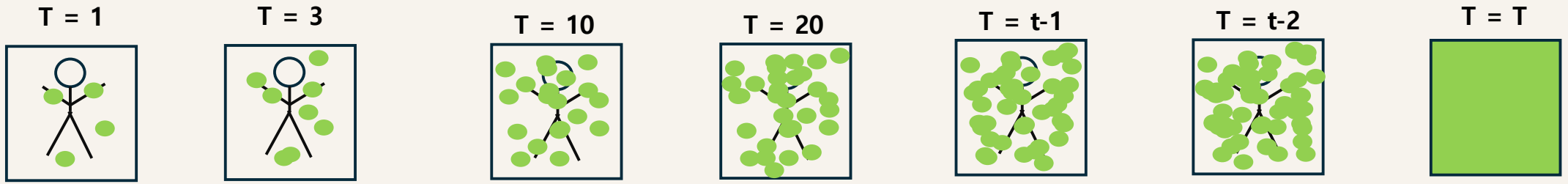
-> $a_0 z_0 + b_0 z_1$은 평균이 0 이고 분산이 $(a^2 + b^2)I$ 인 가우시안 분포를 따른다.

$$
\begin{aligned}
a^2 + b^2 &= \alpha_2(1-\alpha_0) + 1 - \alpha_2 \\
&= \alpha_2 - \alpha_2\alpha_1 + 1 - \alpha_2 \\
&= 1 - \alpha_2\alpha_1 \\
&= 1 - \bar{\alpha}_2
\end{aligned}
$$

Forward Process

실제 모델내부 동작과정

$$X_t = \sqrt{\bar{a}_t} X_0 + \sqrt{1 - \bar{a}_t} \bar{z}_t$$



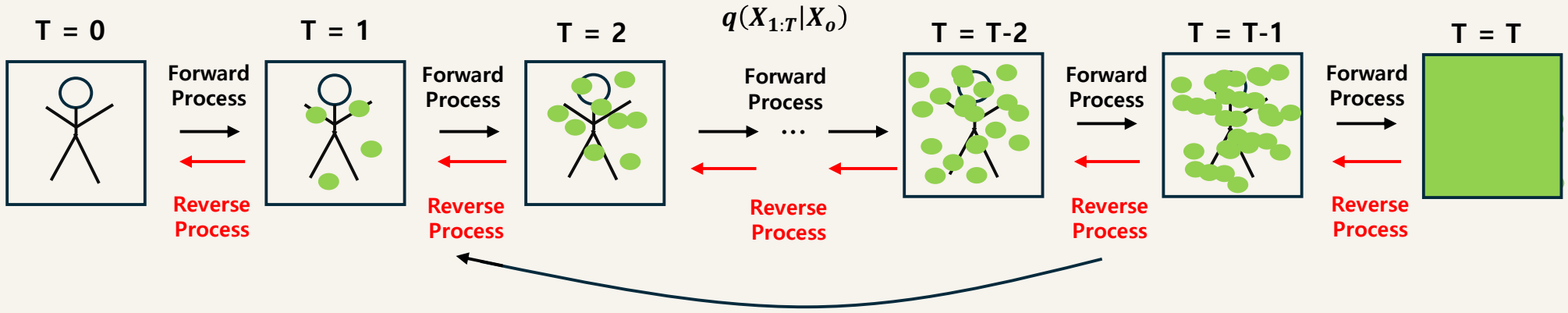
→ 1. 1~T 까지의 다양한 t 값을 랜덤하게 뽑아 X_t 를 만든다

2. 만들어진 X_t 로 신경망은 노이즈를 제거하는 패턴을 학습한다 (Reverse process)

Reverse Process

Reverse Process

- 노이즈(X_t) 를 이미지(X_0) 로 만드는 과정 (Denosing)



$q(X_{t-1}|X_t)$

"Intractable" → 원본 데이터의 분포를 알 수 없음

$$P_\theta(X_{t-1}|X_t) := N(X_{t-1};\ \mu_\theta(x_t,t), \sum_\theta(x_t,t))$$

Neural Net으로 모델링하여 디노이징 과정을 학습한다

Goal : 확률 분포 q에서 관측한 값으로 P_θ의 likelihood를 구했을때 그 likelihood가 최대가 되도록 하는 θ를 찾는것

Reverse Process

DDPM Loss

$$E_{q(X_t|X_0)}[-\log(p_\theta(X_0)] = \int(-\log p_\theta(X_0)) * q(X_t|X_0)dX_T$$

$$= \int(-\log\frac{p_\theta(X_0,X_t)}{p_\theta(X_t|X_0)}) * q(X_t|X_0)dX_T$$

$$= \int(-\log\frac{p_\theta(X_0,X_t)}{p_\theta(X_t|X_0)} * \frac{q(X_t|X_0)}{q(X_t|X_0)}) * q(X_t|X_0)dX_T$$

$$= \int(-\log\frac{p_\theta(X_0,X_t)}{q(X_t|X_0)} * \frac{q(X_t|X_0)}{p_\theta(X_t|X_0)}) * q(X_t|X_0)dX_T$$

$$= \int(-[\log\frac{p_\theta(X_0,X_t)}{q(X_t|X_0)} * q(X_t|X_0) + log\frac{q(X_t|X_0)}{p_\theta(X_t|X_0)}) * q(X_t|X_0)]dX_T$$

$$= \int(-\log\frac{p_\theta(X_0,X_t)}{q(X_t|X_0)}) * q(X_t|X_0)dX_T - D_{KL}[q||p]$$

$$\boxed{\leq \int(-\log\frac{p_\theta(X_0,X_t)}{q(X_t|X_0)}) * q(X_t|X_0))dX_T}$$

$$= \int(-\log\frac{p_\theta(X_0|X_t)}{q(X_t|X_0)} * p_\theta(X_t)) * q(X_t|X_0))dX_T$$

$$= \int(-\log\frac{p_\theta(X_0|X_t)}{q(X_t|X_0)}) * q(X_t|X_0)dX_T + \int(-logp_\theta(X_t))\, q(X_t|X_0))dX_T = E_{q(X_t|X_0)}[-logp_\theta(X_t)] + E_{q(X_t|X_0)}[-log\frac{p_\theta(X_0|X_t)}{\boxed{q(X_t|X_0)}}]$$

Reverse Process

- 노이즈(X_t) 를 이미지(X_0) 로 만드는 과정 (Denosing)
- Gaussian Markov chain의 모수인 μ_θ 를 학습하고자 함

Reverse Process

DDPM Loss

$$E_{q(X_t|X_0)}[-\log(p_\theta(X_0)] \quad \boxed{\leq \int(-\log\frac{p_\theta(X_0,X_t)}{q(X_t|X_0)}) * q(X_t|X_0))dX_T}$$

$$= \int(-\log\frac{p_\theta(X_0|X_t)}{q(X_t|X_0)} * p_\theta(X_t)) * q(X_t|X_0))dX_T$$

$$= \int(-\log\frac{p_\theta(X_0|X_t)}{q(X_t|X_0)}) * q(X_t|X_0)dX_T + \int(-logp_\theta(X_t))\, q(X_t|X_0))dX_T$$

$$= E_{q(X_t|X_0)}[-logp_\theta(X_t)] + E_{q(X_t|X_0)}[-log\frac{p_\theta(X_0|X_t)}{q(X_t|X_0)}]$$

이대로 학습을 진행하면 P_θ 는 q에 따라서 노이즈를 추가하는 방향으로 학습이 진행된다.

따라서 $q(X_t|X_0)$ 를 $q(X_0|X_t)$ 로 바꾸어 학습을 진행해야함

Reverse Process

DDPM Loss

$$
\begin{aligned}
E_{q(X_{1:T}|X_0)}[-\log(p_\theta(X_0)] &= E_{q(X_{1:T}|X_0)}[-\log\frac{p_\theta(X_0,X_1,\dots,X_T)}{p_\theta(X_1,X_2,\dots,X_T|X_0)}] \\
&= E_{q(X_{1:T}|X_0)}[-\log\frac{p_\theta(X_0,X_1,\dots,X_T)}{p_\theta(X_1,X_2,\dots,X_T|X_0)} * \frac{q(X_{1:T}|X_0)}{q(X_{1:T}|X_0)}] \\
&= E_{q(X_{1:T}|X_0)}[-\log\frac{p_\theta(X_0,X_1,\dots,X_T)}{q(X_{1:T}|X_0)} * \frac{q(X_{1:T}|X_0)}{p_\theta(X_1,X_2,\dots,X_T|X_0)}] \\
&= E_{q(X_{1:T}|X_0)}\left[-\log\frac{p_\theta(X_0,X_1,\dots,X_T)}{q(X_{1:T}|X_0)}\right] - D_{KL}((q_{X_1:T}|X_0)||p_\theta(X_1,X_2,\dots,X_t|X_0)) \\
&\leq E_{q(X_{1:T}|X_0)}\left[-\log\frac{p_\theta(X_0,X_1,\dots,X_T)}{q(X_{1:T}|X_0)}\right] \\
&= E_{q(X_{1:T}|X_0)}\left[-\log\frac{p_\theta(X_T)\prod_{t=1}^{T}p_\theta(X_{t-1}|X_t)}{q(X_{1:T}|X_0)}\right] \longrightarrow \text{Markov chain \& Baye`s} \\
&= E_{q(X_{1:T}|X_0)}\left[-log p_\theta(X_T) - \sum_{t=1}^{T} log\frac{p_\theta(X_{t-1}|X_t)}{q(X_t|X_{t-1})}\right] \\
&= E_{q(X_{1:T}|X_0)}\left[-log p_\theta(X_T) - \sum_{t=2}^{T} log\frac{p_\theta(X_{t-1}|X_t)}{q(X_t|X_{t-1})} - log\frac{p_\theta(X_0|X_1)}{q(X_1|X_0)}\right] \longrightarrow \text{t = 1 항을 분리하기}
\end{aligned}
$$

Reverse Process

DDPM Loss

$$E_{q(X_T|X_0)}[-\log(p_\theta(X_0)] \leq E_{q(X_T|X_0)}\left[-\log\frac{p_\theta(X_0,X_1,\dots,X_T)}{q(X_{1:T}|X_0)}\right]$$

$$= E_{q(X_{1:T}|X_0)}\left[-\log\frac{p_\theta(X_T)\prod_{t=1}^{T}p_\theta(X_{t-1}|X_t)}{q(X_{1:T}|X_0)}\right]$$

→ Markov chain

$$= E_{q(X_{1:T}|X_0)}\left[-logp_\theta(X_T)-\sum_{t=1}^{T}log\frac{p_\theta(X_{t-1}|X_t)}{q(X_t|X_{t-1})}\right]$$

$$= E_{q(X_{1:T}|X_0)}\left[-logp_\theta(X_T)-\sum_{t=2}^{T}log\frac{p_\theta(X_{t-1}|X_t)}{q(X_t|X_{t-1})}-log\frac{p_\theta(X_0|X_1)}{q(X_1|X_0)}\right]$$

→ t = 1 항을 분리하기

$$\vdots$$

$$= E_{q(X_{1:T}|X_0)}[-log\frac{p_\theta(X_T)}{q(X_T|X_0)}-\sum_{t=2}^{T}log\frac{p_\theta(X_{t-1}|X_t)}{q(X_{t-1}|X_t,X_0)}-logp_\theta(X_0|X_1)]$$

→ 이전에 유도했던 Loss와 다르게 reverse process 방향으로 진행

↳ 최종 디퓨전 Loss

Reverse Process

DDPM Loss

$$E_{q(X_{1:T}|X_0)}[-log\frac{p_\theta(X_T)}{q(X_T|X_0)} - \sum_{t=2}^{T} log\frac{p_\theta(X_{t-1}|X_t)}{q(X_{t-1}|X_t, X_0)} - logp_\theta(X_0|X_1)]$$

① ② ③

① $-log\frac{p_\theta(X_T)}{q(X_T|X_0)}$: 학습을 진행하는 파라미터 θ에 대해서 무관하기때문에 무시

② $\sum_{t=2}^{T} log\frac{p_\theta(X_{t-1}|X_t)}{q(X_{t-1}|X_t, X_0)}$: 주된 학습 목표 (reverse process)

③ $logp_\theta(X_0|X_1)$: 수많은 T(Time-step)에서 극히 일부분이므로 값이 매우 작음(학습과정에서 무시)

결국 ② 를 minimize 하는 문제

Reverse Process

DDPM Loss

② $E_{q(X_{1:T}|X_0)} \sum_{t=2}^{T} -log \frac{p_\theta(X_{t-1}|X_t)}{q(X_{t-1}|X_t, X_0)}$: 주된 학습 목표 (reverse process)

$$\int p(x)logp(x)dx = -\frac{1}{2}(1 + \log(2\pi\sigma_1^2))$$

$$\int p(x)logq(x)dx = -\frac{1}{2}\log(2\pi\sigma_2^2) - (\frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2})$$

→ 분포가 가우시안(p,q)

$$D_{KL}(p||q) = \int p(x)logp(x)dx - \int p(x)logq(x)dx$$

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

→ DDPM에서는 분산이 고정값이므로 $\frac{\sigma_2}{\sigma_1}$ = 1

$$= E_q\left[\frac{1}{2\sigma_t^2}||\mu_1 - \mu_2||^2\right] + C$$

Reverse Process

DDPM Loss

$$②E_{q(X_{1:T}|X_0)}\sum_{t=2}^{T}-log\frac{p_\theta(X_{t-1}|X_t)}{q(X_{t-1}|X_t,X_0)}$$ **: 주된 학습 목표 (reverse process)** $D_{KL}(p||q)=E_q\left[\frac{1}{2\sigma_t^2}||\mu_1-\mu_2||^2\right]+C$

$$=E_q\left[\frac{1}{2\sigma_t^2}||\underbrace{\tilde{\mu}_t(X_t,X_0)}_{\text{q}}-\underbrace{\mu_\theta(X_t,t)}_{\text{p}}||^2\right]+C$$

⟶ q, p 분포의 평균값($\tilde{\mu}_t, \mu_\theta$)만 구하면 로스 계산이 가능

Reverse Process

DDPM Loss

$$E_q\left[\frac{1}{2\sigma_t^2}\left|\left|\underbrace{\tilde{\mu}_t(X_t, X_0)}_{\text{q}} - \underbrace{\mu_\theta(X_t, t)}_{\text{p}}\right|\right|^2\right] + C$$

→ **Goal : q, p 의 평균값을 구하기**

1. $q(X_t|X_0)$

$q(X_t|X_{t-1}) = N(X_t; A = \sqrt{1-\beta_t}X_{t-1}, \beta_t I)$ → **Reparameterization trick 사용하여 표현**

$\alpha_t \coloneqq 1 - \beta_t \; and \; \bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s$

→ $\boldsymbol{X_t = \sqrt{\alpha_t}X_{t-1} + \sqrt{1-\alpha_t}\epsilon}$

⋮ X_{t-1}을 **같은 방식으로 구하여 대입 후 정리**

→ $\boldsymbol{X_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon}$

→ **즉 $q(X_t|X_0)$ 는 평균이 $\sqrt{\bar{\alpha}_t}$, 표준편차가 $\sqrt{1-\bar{\alpha}_t}$ 인 분포**

$q(X_t|X_{t-1}) = N(X_t; A = \sqrt{1-\beta_t}X_{t-1}, \beta_t I)$

$X \sim N(\mu, \sigma^2) \rightarrow x = \mu + \sigma\epsilon, \quad \epsilon \sim N(0, I)$

1. $\alpha_t \coloneqq 1 - B_t \; (0 < B_t < 1)$, $\bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s$
2. $X_t = \sqrt{\alpha_t}X_{t-1} + \sqrt{1-\alpha_t}z_{t-1}$
3. $X_1 = \sqrt{\alpha_1}X_0 + \sqrt{1-\alpha_1}z_0$
4. $X_2 = \sqrt{\alpha_2}X_1 + \sqrt{1-\alpha_2}z_1$
5. X_1를 X_2에 대입 : $X_2 = \sqrt{\alpha_2\alpha_1}X_0 + \sqrt{\alpha_2(1-\alpha_0)}z_0 + \sqrt{1-\alpha_2}z_1$
6. $X_2 = \sqrt{\alpha_2\alpha_1}X_0 + \sqrt{1-\bar{\alpha}_2}\bar{z}_1$
7. $X_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\bar{z}_t$

Reverse Process

DDPM Loss

② $E_{q(X_{1:T}|X_0)} \sum_{t=2}^{T} -log \frac{p_\theta(X_{t-1}|X_t)}{q(X_{t-1}|X_t, X_0)} = E_q \left[\frac{1}{2\sigma_t^2} \left|\left| \tilde{\mu}_t(X_t, X_0) - \mu_\theta(X_t, t) \right|\right|^2 \right] + C$

1. $q(X_{t-1}|X_t, X_0)$

$$q(X_{t-1}|X_t, X_0) = q(X_t|X_{t-1}, X_0) \frac{q(X_{t-1}|X_0)}{q(X_t|X_0)}$$

→ $q(X_t|X_0) = N(X_t; \sqrt{\bar{\alpha}_t} X_0, (1-\bar{\alpha}_t) I)$ 대입

⋮ $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ 에 대입 후 계산

→ $\tilde{u}_t(x_t, x_0) \coloneqq \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} X_0 + \frac{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} X_t \, , \tilde{\beta}_t \coloneqq \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$

⋮ X_T에 대해서 정리

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left(X_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t \right)$$

2. $p_\theta(X_{t-1}|X_t)$

→ 근본적으로 q에 근사를 해야한다.
다만, θ (학습 가능한 파라미터) 를 도입하여 학습 가능하도록 설정한다

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

Reverse Process

DDPM Loss

② $E_{q(X_{1:T}|X_0)} \sum_{t=2}^{T} -log \frac{p_\theta(X_{t-1}|X_t)}{q(X_{t-1}|X_t, X_0)} = E_q \left[\frac{1}{2\sigma_t^2} \left|\left| \tilde{\mu}_t(X_t, X_0) - \mu_\theta(X_t, t) \right|\right|^2 \right] + C$

$\longrightarrow$ $\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} (X_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t)$ $\quad$ $\mu_\theta(X_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(X_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(X_t, t) \right)$

⋮ 위 두 식을 대입하여 정리

최종 $Diffusion\ Loss = E_{t,X_0,\epsilon}\left[\left|\left| \epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right) \right|\right|^2 \right]$

$\longrightarrow$ Neural Network는 원본 노이즈 ϵ 과 시점 t에서 모델이 예측한 노이즈의 L2 Norm을 최소화 하는 방향으로 학습하게 된다(노이즈 패턴의 학습)

4 Result & Summary

Result & Summary

평가지표 : FID score, IS score

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87 ± 0.12	25.32	
SNGAN [39]	8.22 ± 0.05	21.7	
SNGAN-DDLS [4]	9.09 ± 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	$\mathbf{9.74} \pm 0.05$	3.26	
Ours (L, fixed isotropic $\boldsymbol{\Sigma}$)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	**3.17**	≤ 3.75 (3.72)

FID Score : 실제 이미지와 생성된 이미지를 Pre-trained 된 InceptionNet 을 이용하여 피처맵을 추출한 뒤 추출된 두개의 피처맵의 분포 차이로 계산

IS Score: 실제 이미지와 생성된 이미지를 Pre-trained 된 InceptionNet 을 이용하여 분류한 뒤 분류 확률값을 이용하여 KL-divergence 를 계산

NLL Test: 모델이 데이터의 분포를 얼마나 효율적으로 학습했는지를 보여주는 지표, 한 픽셀을 표현할때 얼마나 많은 비트가 사용되는지

Result & Summary

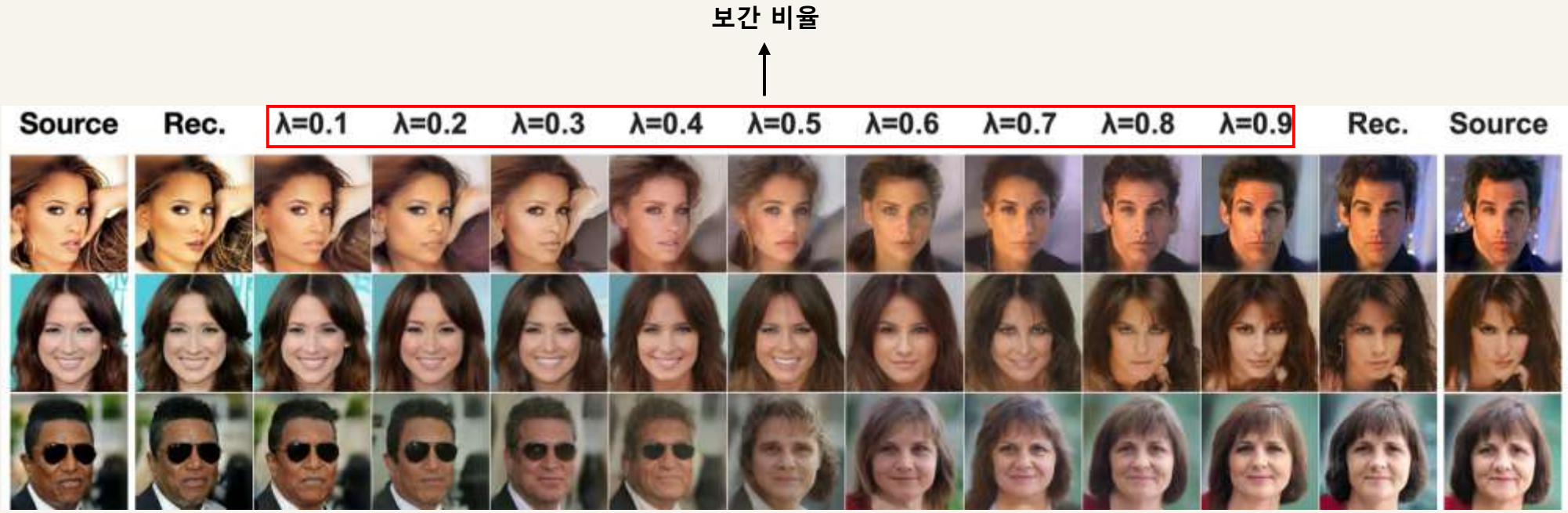
생성 샘플

LSUN(Church)，FID : 7.89

LSUN(Bedroom)，FID : 4.90

Result & Summary

서로 다른 두 이미지 보간



Result & Summary

Summary

- Diffusion은 Forward Process와 Reverse Process로 나뉜다.
- Forward Process는 원본 이미지에 Gaussian noise를 주입하여 완전한 Gaussian noise로 만든다
- Reverse Process는 Denoise 방향으로 학습이 진행된다.

5 Code

Expreiment

Simple diffusion process code

```
class GaussianDiffusionTrainer(nn.Module):
    def __init__(self, model,beta_1 , beta_T, T):
        super().__init__()

        self.mode = model
        self.T = T

          Ctrl+I to chat, Ctrl+K to generate
        # 1. Beta schedule 정의
        self.register_buffer(
            'betas', torch.linspace(beta_1, beta_T, T).double())

        # 2 alpha, alpha_bar 계산
        alphas = 1. -self.betas
        alphas_bar = torch.cumprod(alphas , dim = 0)

        # 3. x_t 계산에 필요한 계수들 계산
        self.register_buffer(
            'sqrt_alphas_bar' , torch.sqrt(alphas_bar)
        )
        self.register_buffer(
            'sqrt_one_minus_alphas_bar', torch.sqrt(1-alphas_bar)
        )

    def forward(self, x_0):

        # 1. 랜덤한 타임스텝 t 선택
        t = torch.randint(self.T ,size=(x_0.shape[0], ), device=x_0.device)


        # 2. 표준 정규분포에서 노이즈 샘플링
        noise = torch.randn_like(x_0)

        # 3. x_t 생성 (Diffusion forward process)
        x_t = (
            extract(self.sqrt_alphas_bar, t, x_0.shape) * x_0 +
            extract(self.sqrt_one_minus_alphas_bar, t, x_0.shape) * noise

        )

        # 4. Loss 계산
        loss = F.mse_loss(self.model(x_t , t), noise , reduction='none')

        return loss
```

Noise 강도 β 선택

x_t를 샘플링 하기 위해서 $\alpha, \bar{\alpha}$ 계산

x_t를 샘플링 하기 위해서 $\sqrt{\alpha}, \sqrt{1-\bar{\alpha}}$ 계산

$x_t = \sqrt{\alpha} x_0 + \sqrt{1-\bar{\alpha}} z$ 계산

$E_{t,X_0,\epsilon}[\left|\left|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} X_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right|\right|^2$ 계산