

DDIM

: Denoising Diffusion Implicit Models

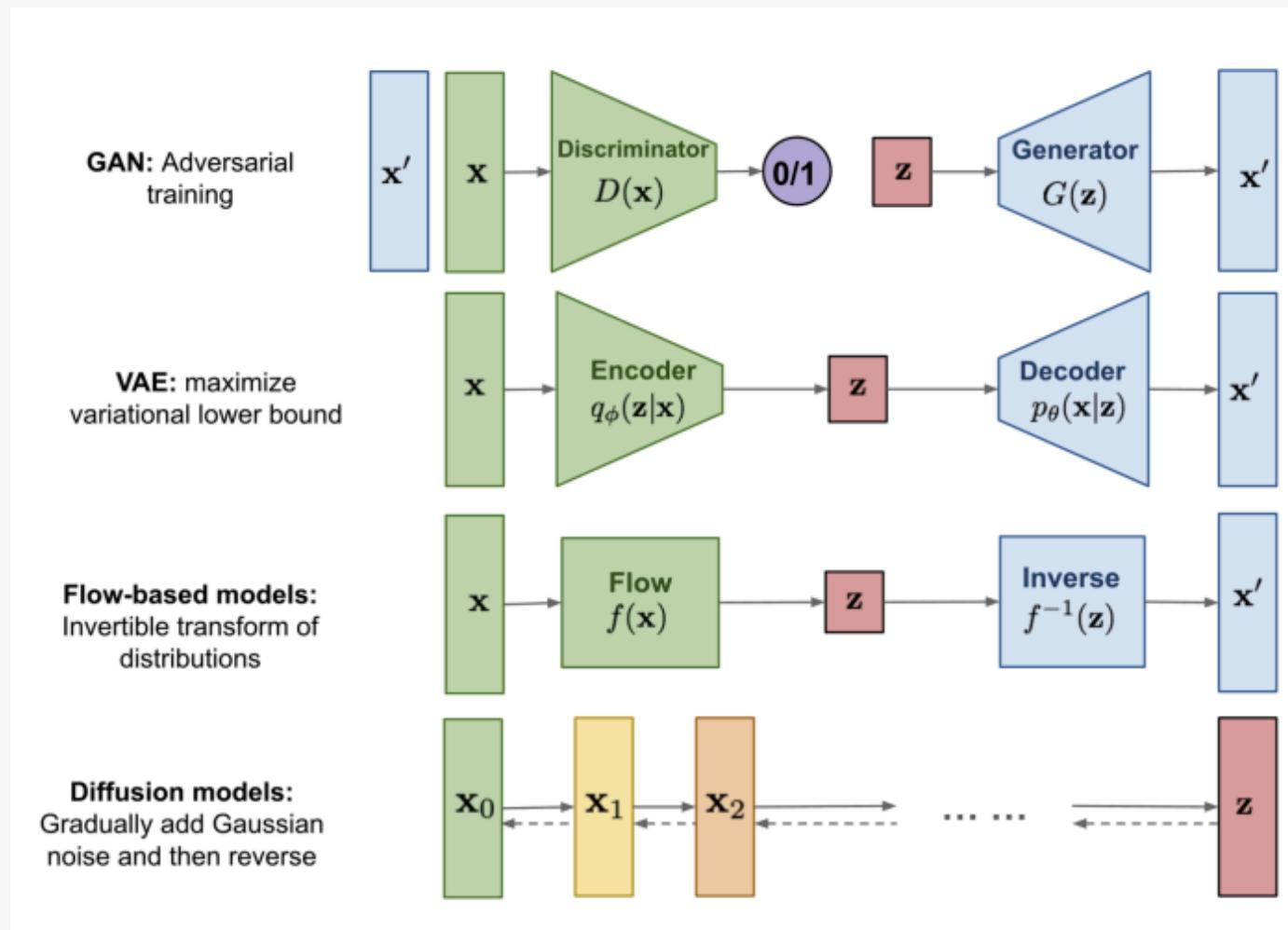
황영서

1

INTRODUCTION

1. Introduction

Generative Model과 GAN



- **GAN >>> VAE , autogressive model (자귀회귀모델) , normalizing flows (정규화 흐름)**
- **GAN의 단점 :**
 1. **adversarial training** 을 하여 균형을 이루어야함
 2. 성능이 아키텍쳐에 큰 영향을 받음
 3. 다양한 sample을 생성하지 못하는 **Mode Collapse** 문제가 있음

1. Introduction

DDPM의 도입 & 특징

- DDPM은 adversarial learning 을 하지않고 GAN에 견줄 만한 sample을 생성
- Gaussain noise 에 의해 손상된 sample의 noise를 제거하도록 , denoising autoencoding 모델 train
- sample은 Markov-chain의해 생성되며, foward process를 역으로 수행하며 이루어짐

DDPM의 단점

- DDPM의 generation process는 foaward process의 역을 근사하기 때문에 모든 단계를 반복해야, 하나의 sample을 생성할 수 있다는 단점
- 네트워크를 한 번 통과하면 되는 GAN과 비교할 때 훨씬 느림

ex) 같은 GPU를 사용할때, 50k 32x32 크기 이미지를 sampling 하는데
DDPM으로 약 20시간이 걸리지만,
GAN으로 1분이내 가능

더큰 이미지를 sampling 할경우, 50k 개의 256x256 이미지를 sampling하는데 1000시간이 걸림

1. Introduction

DDIM 도입

DDPM과 GAN간의 효율성 격차를 줄이기 위해 DDIM을 제안함!!!

DDIM?

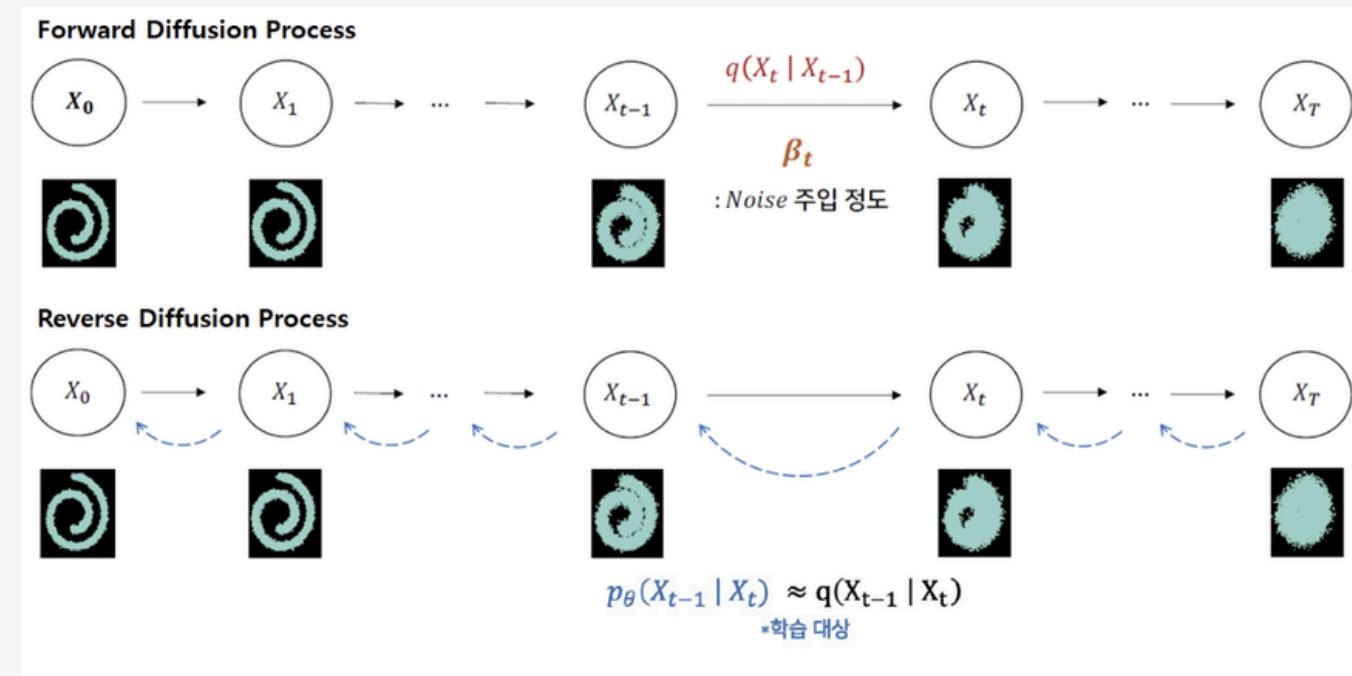
- DDPM의 forward process를 non-Markov chain으로 변경
- variational training objectives는 DDPM을 train할 때 사용했던 식과 일치
- short generative chain을 활용하여, sampling 속도를 높임

2

BACKGROUND (DDPM)

2. DDPM

1. Forward Process



- forward process (diffusion process) 에서는 점진적으로 노이즈를 추가하면서 변형된다. 특정 단계에서의 상태 x_t 가 x_{t-1} 에 의존하여 어떻게 분포하는지 나타내며, 이 과정은 Gaussian 분포를 따른다.

- 특정 시점 t 에서의 상태 x_t 가 x_0 에 조건부로 어떻게 분포되는지?

$$q(\mathbf{x}_t | \mathbf{x}_0) := \int q(\mathbf{x}_{1:t} | \mathbf{x}_0) d\mathbf{x}_{1:(t-1)} = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_0, (1 - \alpha_t) \mathbf{I});$$

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

(*DDPM 논문에서는 α_t 의 누적곱을 $\bar{\alpha}_t$ 로 표시하지만 DDIM 논문에서는 그냥 α_t 로 표시하는 것 같다)

- Approximate Posterior : 전체 forward process 과정을 나타내며, x_0 로부터 $x_1, x_2, x_3, \dots, x_T$ 모든 latent vector 의 joint distribution 나타낸다.

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

- Reparameterization Trick을 활용하여 다음과 같이 표현할 수 있음

$$\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

2. DDPM

2. Reverse Process

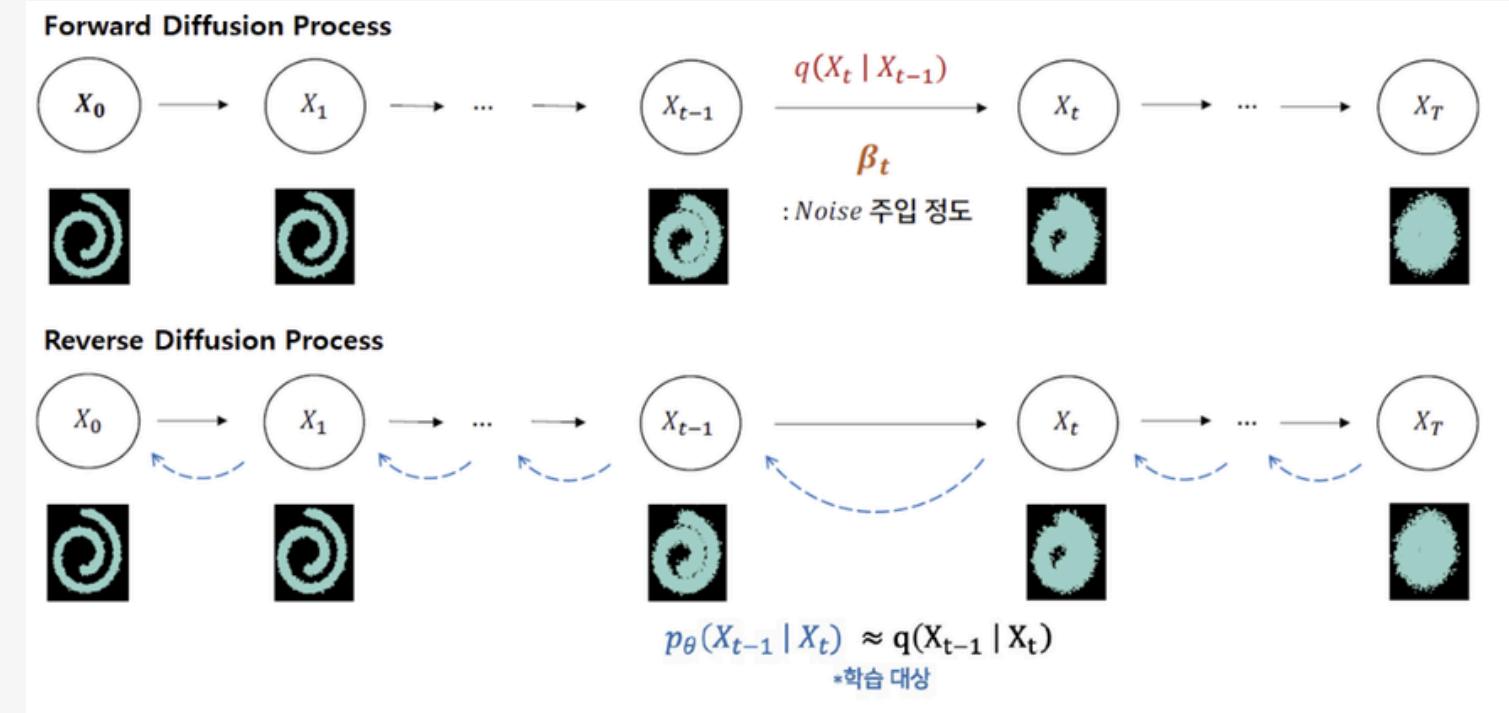
알고 싶은 것은 $q(x_{t-1}|x_t)$ 이나, 이를 알기는 어려워서 $p_\theta(x_{t-1}|x_t)$ 의 parameter를 예측 및 학습

- Reverse process :

$p_\theta(x_{t-1}|x_t)$ 는 x_t 에서 x_{t-1} 로 가는 조건부 분포를 나타내며, 이는 Gaussian 분포를 따르며, 평균과 공분산은 학습 parameter θ 에 의해 결정된다.

$p_\theta(x_{1:T}|x_0)$ 는 x_T 부터 x_0 까지의 latent vector의 joint distribution을 나타낸다.

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (1)$$



- $q(x_{t-1}|x_t)$ 는 알 수 없지만, Bayes Rules에 의해 $q(x_{t-1}|x_t, x_0)$ 는 다음과 같이 알 수 있다,

$$\textcircled{1} \quad q(x_{t-1}|x_t, x_0) = q(x_t|x_{t-1}, x_0) \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \quad \because \text{bayes rule}$$

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

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

2. DDPM

3. 학습 목표

- $\mathbb{E}_{q(\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0)]$ 를 최대화하는 것이 목표이며, 이를 직접 최대화하는 것이 어려워 ELBO (변분 하한) 을 활용하여, ELBO를 최대화 하여 이를 근사한다.

$$\max_{\theta} \mathbb{E}_{q(\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_0)] \leq \max_{\theta} \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T)} [\log p_\theta(\mathbf{x}_{0:T}) - \log q(\mathbf{x}_{1:T} | \mathbf{x}_0)] \quad (2)$$

- ELBO를 negative log likelihood로 표현하여, 최소화 문제로 변환하였고, 이를 간단히 정리하면 다음과 같다.

최종 Loss 함수라고 생각하면 된다. 예측된 노이즈와 실제 노이즈 ϵ_t 간의 차이의 제곱 오차를 최소화하는 것이 목표이다.

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

$$L_\gamma(\epsilon_\theta) := \sum_{t=1}^T \gamma_t \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \epsilon_\theta^{(t)} (\sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1-\alpha_t} \epsilon_t) - \epsilon_t \right\|_2^2 \right] \quad (5)$$

4. Sampling

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:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 
```

3

DDIM

**Forward Process,
Generation Process,
Variational Inference Objective**

3. DDIM

3.1 Non-Markovian Forward Processes

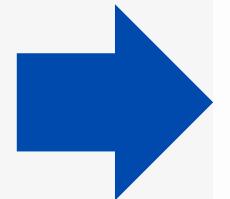
marginal distribution $q(x_t|x_0)$ 를 그대로 사용하되,

$$q(\mathbf{x}_t|\mathbf{x}_0) := \int q(\mathbf{x}_{1:t}|\mathbf{x}_0) d\mathbf{x}_{1:(t-1)} = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I});$$

joint distribution을 다음과 같이 바꾼다

- DDPM에서는 x_t 가 바로 이전 step x_{t-1} 에 의해 결정되는 markovian chain이다

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



- DDIM에서는 x_t 가 바로 이전 step x_{t-1} 과 x_0 에 의해 결정되는 non-markovian chain으로 바꾼다

$$q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) := q_\sigma(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^T q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$$

3. DDIM

3.1 Non-Markovian Forward Processes

DDPM에 따르면 다음과 같은 식을 만족하고,

$$q_\sigma(\mathbf{x}_T | \mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_T} \mathbf{x}_0, (1 - \alpha_T) \mathbf{I}) \text{ and for all } t > 1.$$

이를 만족하기 위해서는,

$$q_\sigma(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}} \mathbf{x}_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \mathbf{I}\right)$$

reverse process는 다음과 같이 정리되어야 한다.

Forward Process

Forward process는 Bayes Rules에 의해 다음과 같이 표현할 수 있으며, 이는 gaussian distribution을 따른다.

$$q_\sigma(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q_\sigma(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) q_\sigma(\mathbf{x}_t | \mathbf{x}_0)}{q_\sigma(\mathbf{x}_{t-1} | \mathbf{x}_0)}, \quad (8)$$

- DDIM의 Forward Process는 non-markovian이다
- σ 는 forward process가 얼마나 stochastic 한지를 결정하며, 이 값이 0에 가까워질수록 무작위성이 없어져 x_0 와 x_t 가 알려진 경우 x_{t-1} 이 고정된다. 즉, deterministic 해진다

<https://junia3.github.io/blog/ddim>

Bishop의 pattern recognition 책의 2절 115번 공식으로 증명할 수 있다고 한다.

3. DDIM

3.2 Generative process and Unified Variational Inference Objective

Generative Process

학습가능한 generative process $p_\theta(x_{0:T})$ 를 정의해본다.

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

$$q(\mathbf{x}_t|\mathbf{x}_0) := \int q(\mathbf{x}_{1:t}|\mathbf{x}_0) d\mathbf{x}_{1:(t-1)} = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I});$$

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (4)$$

를 활용하여

$p_\theta(\mathbf{x}_T) = N(\mathbf{0}, \mathbf{I})$ 와 함께, 다음과 같이 정의할 수 있다.

$$f_\theta^{(t)}(\mathbf{x}_t) := (\mathbf{x}_t - \sqrt{1 - \alpha_t} \cdot \epsilon_\theta^{(t)}(\mathbf{x}_t)) / \sqrt{\alpha_t}. \quad (9)$$

$$p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \begin{cases} \mathcal{N}(f_\theta^{(1)}(\mathbf{x}_1), \sigma_1^2 \mathbf{I}) & \text{if } t = 1 \\ q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, f_\theta^{(t)}(\mathbf{x}_t)) & \text{otherwise,} \end{cases} \quad (10)$$

위와 같이 식을 변형할 수 있고, x_t 를 이용하여 x_0 을 예측할 수 있다.

3. DDIM

3.2 Generative process and Unified Variational Inference Objective

Variational Inference Objective

$\log p_\theta(x_0)$ 를 최대화 하는것 즉, $E[-\log p_\theta(x_0)]$ 를 최소화 하는 것이 목적이다

$$\begin{aligned} & E_{x_T \sim q(x_T|x_0)} [-\log p_\theta(x_0)] \\ &= E_{x_T \sim q(x_T|x_0)} \left[-\log \frac{p_\theta(x_0, x_1, x_2, \dots, x_T)}{p_\theta(x_1, x_2, x_3, \dots, x_T|x_0)} \right] \quad \text{..... ①} \\ &= E_{x_T \sim q(x_T|x_0)} \left[-\log \frac{p_\theta(x_0, x_1, x_2, \dots, x_T)}{p_\theta(x_1, x_2, x_3, \dots, x_T|x_0)} \cdot \frac{q(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} \right] \quad \text{..... ②} \\ &\leq E_{x_T \sim q(x_T|x_0)} \left[-\log \frac{p_\theta(x_0, x_1, x_2, \dots, x_T)}{q(x_{1:T}|x_0)} \right] \quad \text{..... ③} \\ &= E_{x_T \sim q(x_T|x_0)} \left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right] \quad \text{..... ④} \end{aligned}$$

$$J_\sigma(\epsilon_\theta) := \mathbb{E}_{\mathbf{x}_{0:T} \sim q_\sigma(\mathbf{x}_{0:T})} [\log q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) - \log p_\theta(\mathbf{x}_{0:T})] \quad (11)$$

$$q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) := q_\sigma(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^T q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \quad (6)$$

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

$$= \mathbb{E}_{\mathbf{x}_{0:T} \sim q_\sigma(\mathbf{x}_{0:T})} \left[\log q_\sigma(\mathbf{x}_T|\mathbf{x}_0) + \sum_{t=2}^T \log q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) - \sum_{t=1}^T \log p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) - \log p_\theta(\mathbf{x}_T) \right]$$

3. DDIM

3.2 Generative process and Unified Variational Inference Objective

DDIM의 Variational Inference Objective 인 J_σ 에 대해서

$$\begin{aligned} J_\sigma(\epsilon_\theta) &:= \mathbb{E}_{\mathbf{x}_{0:T} \sim q_\sigma(\mathbf{x}_{0:T})} [\log q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) - \log p_\theta(\mathbf{x}_{0:T})] \\ &= \mathbb{E}_{\mathbf{x}_{0:T} \sim q_\sigma(\mathbf{x}_{0:T})} \left[\log q_\sigma(\mathbf{x}_T|\mathbf{x}_0) + \sum_{t=2}^T \log q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) - \sum_{t=1}^T \log p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) - \log p_\theta(\mathbf{x}_T) \right] \end{aligned} \quad (11)$$

σ 값이 달라질 때마다 다른 Variational Inference Objective J_σ 를 가지므로,
이론적으로는 각 σ 에 대해 다른 모델을 훈련해야 할 것 같지만,

모든 $\sigma > 0$ 에 대해서 $J_\sigma = L_\gamma + C$ 가 존재한다.

Theorem 1. For all $\sigma > 0$, there exists $\gamma \in \mathbb{R}_{>0}^T$ and $C \in \mathbb{R}$, such that $J_\sigma = L_\gamma + C$.

- L_γ 의 특성으로 인해, 모델의 파라미터가 각 시간 단계 t 마다 공유되지 않는 경우, 가중치 γ 에 의존하지 않는다는 특성을 가진다
- L_γ 의 특성 덕분에, L_1 목표를 사용하여 J_σ 를 최적화할 수 있다 (여기서 L_1 인경우는 DDPM의 Variational Inference Objective를 simple하게 한 경우 같음)

-> 결론적으로 DDIM의 variational Inference Objective는 DDPM과 같다

DDPM의 Variational Inference Objective L_γ 을 활용하여

$$L_\gamma(\epsilon_\theta) := \sum_{t=1}^T \gamma_t \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \epsilon_\theta^{(t)} (\sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon_t) - \epsilon_t \right\|_2^2 \right] \quad (5)$$

4

DDIM

Sampling From Generalized Generative Processes

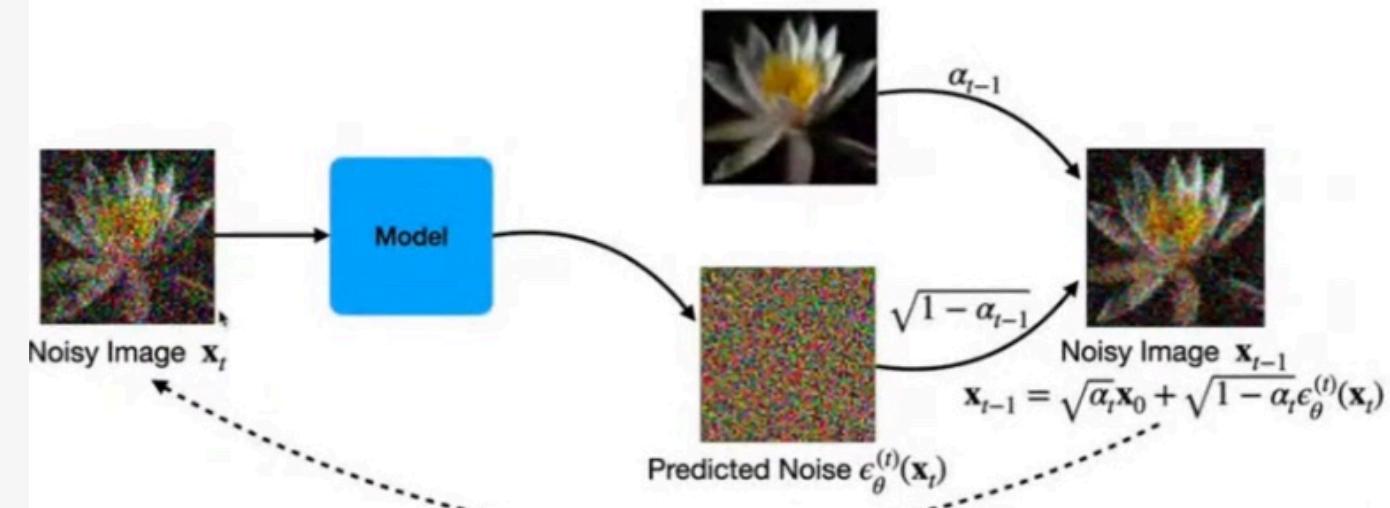
4. DDIM

4.1 Denoising Diffusion Implicit Models

$$p_{\theta}^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \begin{cases} \mathcal{N}(f_{\theta}^{(1)}(\mathbf{x}_1), \sigma_1^2 \mathbf{I}) & \text{if } t = 1 \\ q_{\sigma}(\mathbf{x}_{t-1}|\mathbf{x}_t, f_{\theta}^{(t)}(\mathbf{x}_t)) & \text{otherwise,} \end{cases} \quad (10)$$

$$q_\sigma(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N} \left(\sqrt{\alpha_{t-1}} \mathbf{x}_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \mathbf{I} \right).$$

$$\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_{\theta}^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}}$$



$$f_\theta^{(t)}(\mathbf{x}_t) := (\mathbf{x}_t - \sqrt{1 - \alpha_t} \cdot \epsilon_\theta^{(t)}(\mathbf{x}_t)) / \sqrt{\alpha_t}. \quad (9)$$

$$(\mathbf{x}_t - \sqrt{1 - \alpha_t} \cdot \epsilon_{\theta}^{(t)}(\mathbf{x}_t)) / \sqrt{\alpha_t}$$

$$\sqrt{\alpha_{t-1}} \overset{\downarrow}{x_0} + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t} x_0}{\sqrt{1 - \alpha_t}}$$

$$+ \sigma_t \epsilon_t$$

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \left(\underbrace{\frac{\mathbf{x}_t - \sqrt{1-\alpha_t} \epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}}}_{\text{“predicted } \mathbf{x}_0\text{”}} \right) + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{“direction pointing to } \mathbf{x}_t\text{”}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}} \quad (12)$$

4. DDIM

4.1 Denoising Diffusion Implicit Models

σ 의 역할

- ϵ_t 는 x_t 와 독립적인 gaussian noise이며 $\alpha = 1$ 로 정의한다

1)

$$\sigma_t = \sqrt{(1 - \alpha_{t-1})/(1 - \alpha_t)} \sqrt{1 - \alpha_t/\alpha_{t-1}}$$

일때, forward pass는 Markov-chain을 이루게 된다. DDPM 방식으로 generate한다.

2)

$\sigma_t = 0$ 이면

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N} \left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 I \right).$$

$x_{t-1} = \sqrt{\alpha_{t-1}}x_0 + \sqrt{1 - \alpha_{t-1}} \cdot \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1 - \alpha_t}}$ 와 같은 식이되고,

$$x_{t-1} = \underbrace{\sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1 - \alpha_t} \epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{"predicted } x_0\text{"}} + \underbrace{\sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{"direction pointing to } x_t\text{"}} + \cancel{\sigma_t \epsilon_t}$$

random noise

random noise의 영향이 없어지기 때문에 implicit probabilites model이 된다.
즉, $t \geq 1$ 에 대해 x_{t-1} 이 x_t 와 예측된 x_0 가 주어지면,
deterministic (stochastic x) 방식으로 데이터를 생성한다.

-> implicit 확률 모델은 명시적으로 확률 분포를 정의하지 않으며, 샘플 생성 과정을 통해 확률을 implicit(암묵적)으로 정의한다.

-> 즉, DDPM의 objective로 학습된 implicit probabilistic 모델이기 때문에 DDIM(Denoising Diffusion Implicit Model)로 이름 붙였다.

4. DDIM

4.2 Accelerated Generation Processes

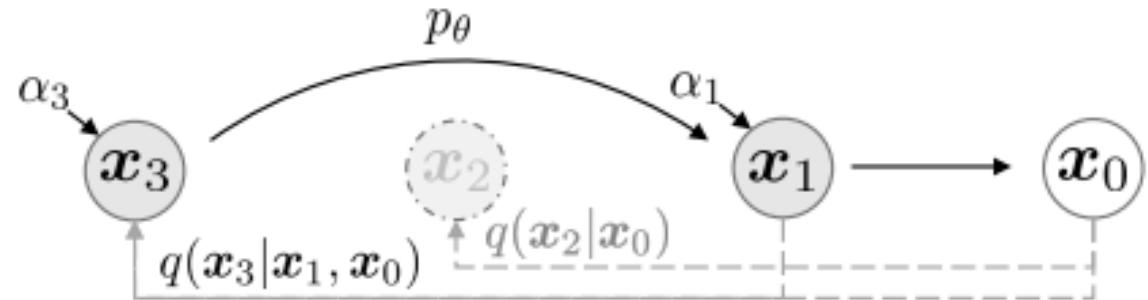


Figure 2: Graphical model for accelerated generation, where $\tau = [1, 3]$.

generative process는 reverse process 와 매우 유사하다. 따라서 foward process가 T step을 거친다면 일반적으로 generative process도 T step 을 가져야한다.

$q_\sigma(x_t|x_0)$ 가 고정된다면($\sigma=0$) forward process도 T보다 짧은 step으로 줄일 수 있기 때문에, generative process도 가속화 될 수 있다.

forward pass를 모든 latenet vector $x_{1:T}$ 에 대해 정의하는 대신 $x_{\tau_1}, \dots, x_{\tau_s}$ 의 subset에 대해서만 foward 와 generative process 가 진행되어, 계산량이 크게 줄어들게 된다.

ex)

T=1000

S=100

$\tau=\{100, 200, 300, \dots, 1000\}$

4. DDIM

4.3 Relevance to Neural ODEs

DDIM은 오일러 근사 추정이 가능한 sampling의 형태를 보여준다.

$$\mathbf{x}_{t-1} = \underbrace{\sqrt{\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t} \epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{"predicted } \mathbf{x}_0\text{"}} + \underbrace{\sqrt{1-\alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{"direction pointing to } \mathbf{x}_t\text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}} \quad (12)$$

앞서 말했던 것 처럼 $\sigma = 0$ 으로 두고, 시간간격을 Δ 로 바꿔주면 다음과 같은 식으로 바꿀 수 있다.

$$\frac{\mathbf{x}_{t-\Delta t}}{\sqrt{\alpha_{t-\Delta t}}} = \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} + \left(\sqrt{\frac{1-\alpha_{t-\Delta t}}{\alpha_{t-\Delta t}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}} \right) \epsilon_\theta^{(t)}(\mathbf{x}_t) \quad (13)$$

$$\begin{aligned} \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} &= \bar{\mathbf{x}}_t \\ \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} &= \sigma \text{ 라고 두고, 식을 정리해보면} \end{aligned}$$

최종적으로 아래와 같은 식이 나오며 미분방정식이 됨을 확인할 수 있다.

$$d\bar{\mathbf{x}}(t) = \epsilon_\theta^{(t)} \left(\frac{\bar{\mathbf{x}}(t)}{\sqrt{\sigma^2 + 1}} \right) d\sigma(t), \quad (14)$$

초기조건은 매우 큰 $\sigma(T)$ 에서 $x(T)$ 가 정규분포 $N(0, \sigma(T))$ 를 따르고, 충분한 이산화 단계까 있으면 생성과정을 역으로 수행할 수 있다.

DDIM이 SDE가 아니라 ODE를 사용함으로써, 시간 dt 가 아니라 $\sigma(t)$ 라는 변수를 사용하여 데이터를 업데이트 하여, sampling 단계의 수가 적을 때 효과적이다.

5

Experiments

5. Experiments

5.1 Sample Quality And Efficiency

$$\sigma_{\tau_i}(\eta) = \eta \sqrt{(1 - \alpha_{\tau_{i-1}})/(1 - \alpha_{\tau_i})} \sqrt{1 - \alpha_{\tau_i}/\alpha_{\tau_{i-1}}},$$

- $\eta = 0$ (DDIM)
- $\eta = 1$ (DDPM)

$$\hat{\sigma}: \hat{\sigma}_{\tau_i} = \sqrt{1 - \alpha_{\tau_i}/\alpha_{\tau_{i-1}}} > \sigma(1)$$

FID : 생성된 품질을 나타내는 지표 (낮을수록 좋음)
S: 샘플링을 거치는 단계

Table 1: CIFAR10 and CelebA image generation measured in FID. $\eta = 1.0$ and $\hat{\sigma}$ are cases of DDPM (although Ho et al. (2020) only considered $T = 1000$ steps, and $S < T$ can be seen as simulating DDPMs trained with S steps), and $\eta = 0.0$ indicates DDIM.

S	CIFAR10 (32×32)					CelebA (64×64)				
	10	20	50	100	1000	10	20	50	100	1000
η	0.0	13.36	6.84	4.67	4.16	4.04	17.33	13.73	9.17	6.53
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93
$\hat{\sigma}$	367.43	133.37	32.72	9.99	3.17	299.71	183.83	71.71	45.20	3.26

- time step이 증가할수록 sample의 품질이 높아진다
- DDIM의 경우, 작은 time step에서도 높은 sample 품질을 생성해낸다
- DDPM인 경우, 동일한 time step에서 덜 stochastic 한 모델이 sample 품질이 더 좋다

5. Experiments

5.1 Sample Quality And Efficiency

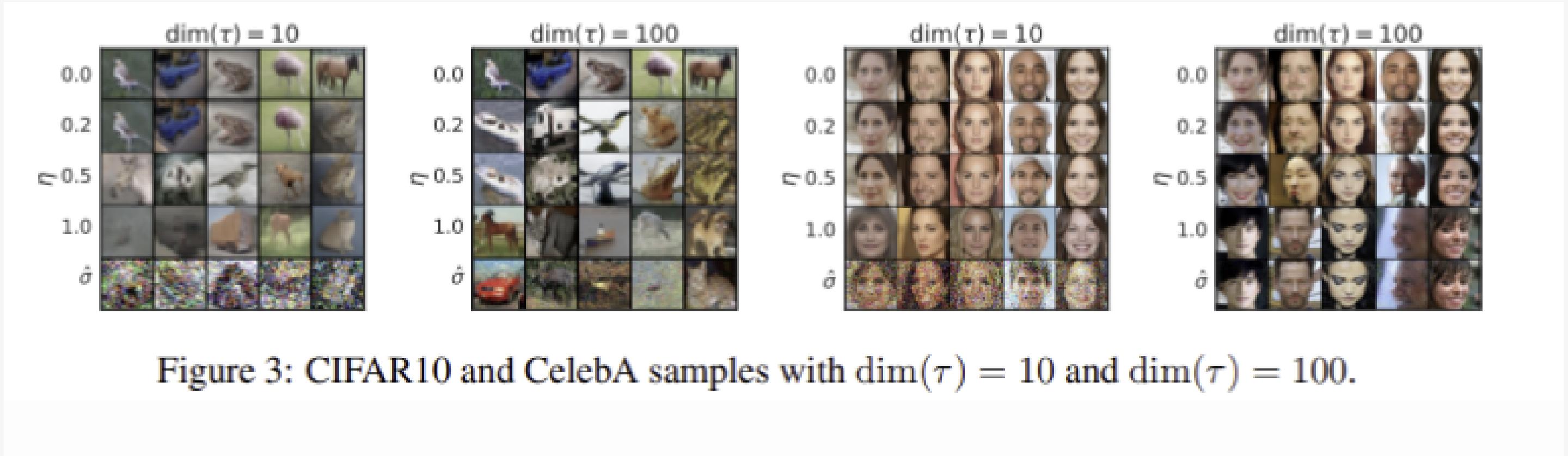


Figure 3: CIFAR10 and CelebA samples with $\text{dim}(\tau) = 10$ and $\text{dim}(\tau) = 100$.

**time step이 동일할 때, 다양한 σ 를 적용하여 실험한 결과,
DDPM의 경우 time step이 작을때, sample 품질이 좋지 않다.**

5. Experiments

5.1 Sample Quality And Efficiency

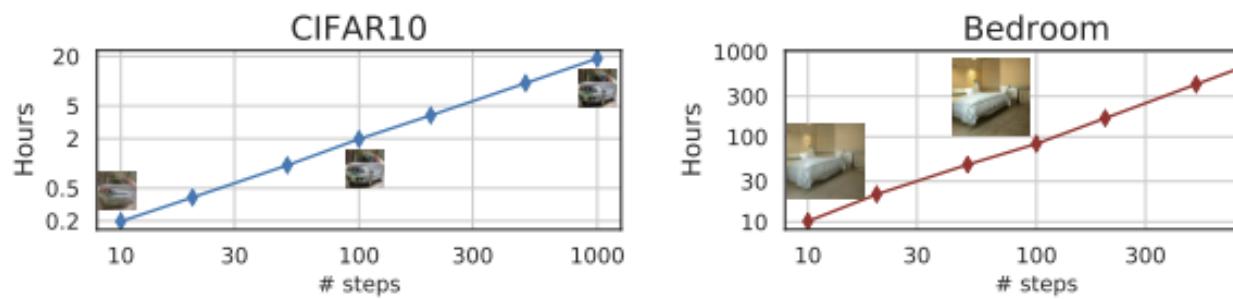


Figure 4: Hours to sample 50k images with one Nvidia 2080 Ti GPU and samples at different steps.

Table 1: CIFAR10 and CelebA image generation measured in FID. $\eta = 1.0$ and $\hat{\sigma}$ are cases of DDPM (although Ho et al. (2020) only considered $T = 1000$ steps, and $S < T$ can be seen as simulating DDPMs trained with S steps), and $\eta = 0.0$ indicates DDIM.

S	CIFAR10 (32×32)					CelebA (64×64)				
	10	20	50	100	1000	10	20	50	100	1000
η	0.0	13.36	6.84	4.67	4.16	4.04	17.33	13.73	9.17	6.53
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93
$\hat{\sigma}$	367.43	133.37	32.72	9.99	3.17	299.71	183.83	71.71	45.20	3.26

CelebA 데이터셋에서 DDPM의 100스텝 FID 점수 = DDIM의 20 스텝 FID 점수
-> DDIM은 20에서 100스텝 내에 DDPM의 1000스텝 모델과 비교할 수 있는 품질을 생성할 수 있다.
이는 DDPM에 비해 10배에서 50배 빠르다.

5. Experiments

5.2 Sample Consistency in DDIMs



Figure 5: Samples from DDIM with the same random x_T and different number of steps.

DDIM의 경우 **generate** 과정이 **deterministic** 이기 때문에,
동일한 초기 x_T 에서 시작한경우, **high level** 특징이 유사한 **sample**을 생성한다.

5. Experiments

5.3 Interpolation in Deterministic Generative Processes

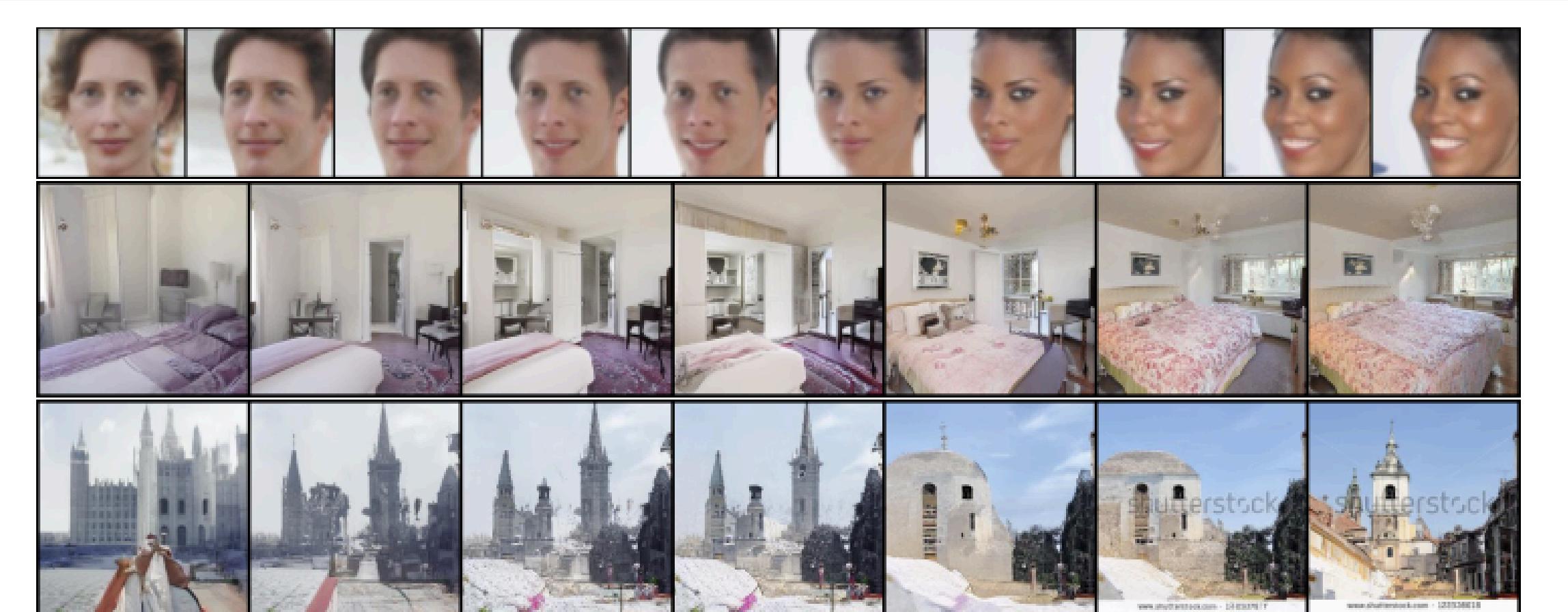


Figure 6: Interpolation of samples from DDIM with $\dim(\tau) = 50$.

DDIM은 deterministic 하기 때문에,
단순한 interpolation 과정이 두 샘플 간의 semantic interpolation을 생성할 수 있다.
(semantic interpolation : 단순히 픽셀 단위의 interpolation이 아니라,
이미지가 자연스럽게 전환됨)

5. Experiments

5.4 Reconstruction from Latenet Space

Table 2: Reconstruction error with DDIM on CIFAR-10 test set, rounded to 10^{-4} .

S	10	20	50	100	200	500	1000
Error	0.014	0.0065	0.0023	0.0009	0.0004	0.0001	0.0001

S 는 Encoding 및 Decoding 과정에서 사용된 단계로,

S 가 증가할수록 x_T 에서 x_0 으로 재구성하는 reconstruction error 가 줄어든다

-> 이는 Neural ODE와 유사한 속성을 가짐

(Neural ODE : 시간에 따른 상태 변화를 모델링하여 초기상태에서 최종상태로 연속적으로 변환)

