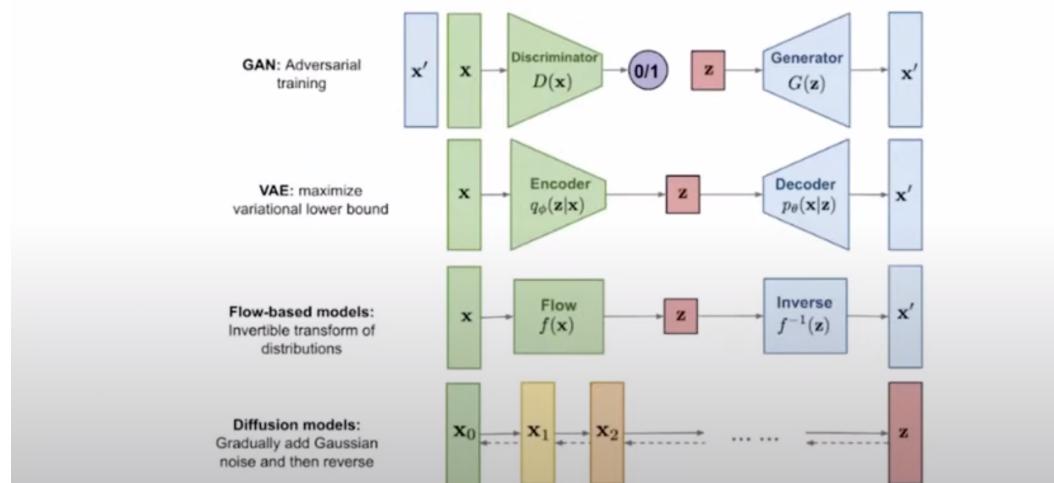


# Denoising Diffusion Probabilistic Models

Generative Models의 목표 Data의 Manifold를 찾는 것!

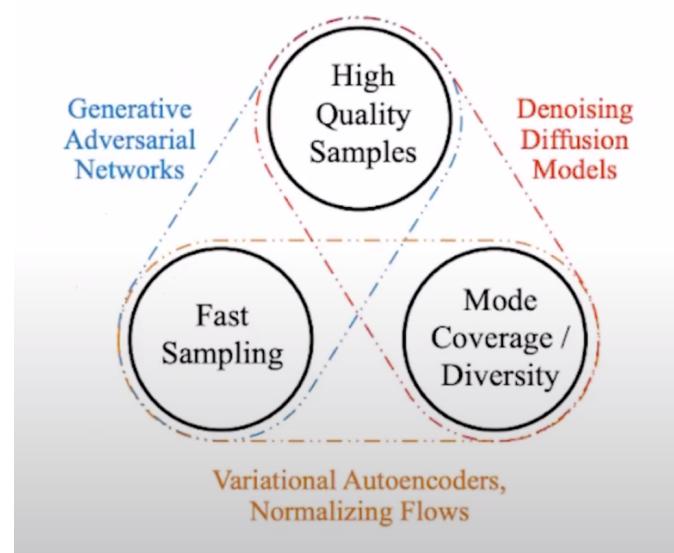
내가 찾고 싶은 이미지가 살고 있는 마을을 찾는 도구를 만들어 내는 것!

## ◆ Generative Models



출처 [https://www.youtube.com/watch?v=1j0W\\_lu55nc](https://www.youtube.com/watch?v=1j0W_lu55nc)

Diffusion Models가 VAE랑 다른 점은 Encoder의 구조가 사람이 만들었으며 학습을 진행하지 않는다. 고정되어 있다!



[https://www.youtube.com/watch?v=1j0W\\_lu55nc](https://www.youtube.com/watch?v=1j0W_lu55nc)

생성 모델마다 특화 분야가 다르고, 용도에 맞춰 사용하면 된다.

**Denoising Diffusion Probabilistic Models** 은 기존의 2015년에 제안된 **Diffusion**을 실용적으로 변경하는 논문이다.

## Abstract

우리는 확률 분포 모델을 사용하여 높은 퀄리티의 이미지를 생성해낸다. 우리의 결과는 확률 분포 모델과 노이즈 제거 방식에 따른 가중치 변형에 대한 훈련을 통해 얻을 수 있습니다. CIFAR 10 데이터 셋에서 우리는 9.46 Inception score, 3.17 FID score를 얻습니다.

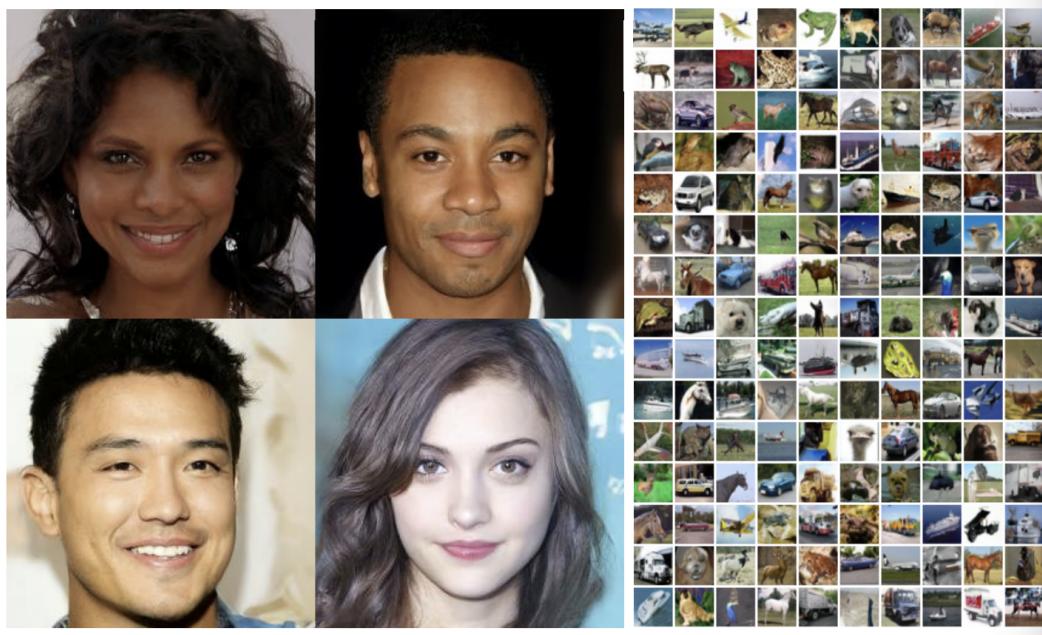


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

## Introduction

최근에 모든 딥러닝 생성 모델은 다양한 방식을 통해 높은 결과를냅니다. GAN, autoregressive models, flows, VAEs는 뛰어난 이미지와 오디오 샘플을 만들고 energy 기반 모델링과 스코어 매칭은 GAN과 비교할만한 뛰어난 발전이 있었습니다.

이 논문은 확률 분포 모델의 과정을 보여줍니다. 확률 분포 모델은 샘플 매칭 데이터를 생산하기 위한 다양한 추론을 사용하는 마르코브 체인 기반입니다. 마르코브 체인은 데이터의 정보가 파괴될 때까지 노이즈를 주입하게 되며 이러한 process를 반대로 진행시키기 위해 학습을 진행합니다. 가우시안 노이즈의 분포로 구성했을 경우, 조건부에 따라 체인을 설정할 수 있으며 이는 쉽게 신경망에 적용할 수 있습니다.

디퓨전 모델은 정의가 간단하며 학습시키기 효율적입니다. 그러나 우리가 우리가 아는 한, 이 모델이 높은 품질의 이미지를 만들 수 있는가에 대한 증명은 하기 어렵습니다. 디퓨전 모델은 높은 이미지를 생성하는 것을 보여주며 때때로 GAN 모델 보다 더 퀄리티 있는 이미지를 보여줄 수 있습니다. 게다가 우리는 디퓨전 모델의 파라미터가 스코어 매칭 방식과 동등함을 보입니다.

우리는 좋은 품질의 이미지를 갖지만, log-likelihood 기법의 다른 모델과 비교했을 경우 경쟁적이지는 않습니다.

## Background

디퓨전 모델은  $x_1, x_2, \dots, x_T$ 에서의 잠재 변수 모델입니다.  $x_0 - q(x_0)$ 은 같은 차원을 가집니다. 그리고  $p_\theta : T$ 는 반대 과정으로 가우시안 변환 학습을 하는 마르코브 체인으로 정의됩니다.

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

디퓨전 모델이 다른 잠재 변수 모델과 구별되는 점은 디퓨전 과정의 전진 과정에서 사후분포가  $B_1, \dots, B_T$  스케줄에 따라 가우시안 노이즈가 점점 더해지는 마르코브 체인으로 수정된 점입니다.

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}), \quad q(\mathbf{x}_t | \mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad (2)$$

학습은 네거티브 log likelihood에서 분산을 최적화함에 따라 수행되어집니다.

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[ -\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \mathbb{E}_q \left[ -\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] =: L \quad (3)$$

전진 과정의 variance  $\beta_t$ 는 파라미터나 하이퍼파라미터의 고정 상수로 이용이 가능합니다. 그리고 반대 과정에서 또한 가우시안 노이즈를 사용합니다. 전진 프로세스는 sampling  $x_t$ 를  $t$ 시점에서 이용합니다.

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

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (4)$$

위를 통하여 확률적 경사 하강법에 의해 최적화 학습이 가능하다. 우리는 아래와 같이 쓰인 Loss방식에서 부터 개선을 시작한다.

$$\mathbb{E}_q \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right] \quad (5)$$

위의 식의 경우 KL divergence를  $p_\theta(x(t-1)|x(t))$ 에 대해 전진 사후 분포로 사용한다.  $x_0$ 가 다음과 같이 정의 될 때

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

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

결국 KL divergences는 5번째 공식 안에 들어있다. 그리고 5번 공식은 가우시안과 비교되며 분산이 높은 몬테 카를로 추정을 대신하여 좀 덜한 추정이 필요한 Rao-Blackwellized fashion으로 계산되어진다.

## Diffusion models and denoising autoencoders

디퓨전 모델은 확률 변수 모델의 제한적인 클래스를 보일 수 있습니다. 그러나 실행에 있어서 많은 다양한 생성이 가능합니다. 우리는 전진 과정에서의 변수  $B_t$ 와 모델 구조 그리고 역 과정에서의 가우시안 학습을 선택해야 합니다. 우리는 간단하고 가중치 분산 목적 함수를 통해 디퓨전 모델과 디노이즈 스코어 매칭 사이의 새로운 연결을 만들었습니다. 궁극적으로 우리는 모델을 단순하게 정의하였습니다. 우리는 위의 5번째 공식에 대해 정리합니다.

### Forward process and $L_t$

우리는 전진 과정의 학습 가능한 변수  $\beta_t$ 를 고정 상수로 변경합니다. 이에 따라 posterior  $q$ 는 더 이상 학습할 파라미터가 없고 결과적으로 5번 공식에서의  $L_T$ 는 학습 과정에서 필요 없습니다.

### Reverse process and $L_1 : T - 1$

이제 우리가 얘기할 부분은 역 과정입니다.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \text{ for } 1 < t \leq T. \quad \Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$$

$\sigma_t^2 I$ 는 학습이 필요하지 않습니다. 실험적으로,  $\sigma_t^2 = \beta_t$ 는 같으며 아래의 식도 같습니다.

$$\sigma_t^2 = \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$$

그리고 첫번째 선택은  $x_0 > N(0, I)$ 에 대한 최적화입니다. 그리고  $x_0$ 이 한 지점으로 가는 것에 대한 최적화입니다. 이 두 가지 선택은 역 과정 데이터에 대한 엔트로피의 상한선과 하한선에 해당합니다.

두 번째로, mean  $u_\theta(x_t, t)$ 를 나타내기 위해 우리는  $L_t$ 를 분석해 특정 공식을 제한 합니다.

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

위와 같은 식은 아래와 같이 정의할 수 있습니다.

$$L_{t-1} = \mathbb{E}_q \left[ \frac{1}{2\sigma_t^2} \|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2 \right] + C \quad (8)$$

$C$ 는  $\theta$ 에 의존하지 않는 고정 상수입니다. 그래서 우리는  $u_\theta$ 는 전진 과정의 사후 평균인  $u_t$ 를 예측하는 것임을 알 수 있습니다. 그러나, 우리는 8번 공식을 확장 시켜

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

에 따라 아래와 같이 적을 수 있습니다.

$$L_{t-1} - C = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[ \frac{1}{2\sigma_t^2} \left\| \tilde{\boldsymbol{\mu}}_t \left( \mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}) \right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t) \right\|^2 \right] \quad (9)$$

$$= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[ \frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right) - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t) \right\|^2 \right] \quad (10)$$

공식 10은  $u_\theta$ 가  $x_t$ 에 따라 아래의 식을 예측 할 수 있음을 보입니다.

$$\frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right)$$

$x_t$ 의 경우 모델에서 주어지기 때문에 우리는 공식을 아래와 같이 선택할 수 있습니다.

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \tilde{\boldsymbol{\mu}}_t \left( \mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t)) \right) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) \quad (11)$$

$\boldsymbol{\epsilon}_\theta$ 는  $x_t$ 로 부터  $\boldsymbol{\epsilon}$ 을 추측할 수 있게 하는 근사치입니다.  $X_t$  주어졌을 때  $\mathbf{x}(t-1)$ 을 예측하는 사후분포는 아래와 같이 정의됩니다.

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}, \text{ where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

샘플링 하는 과정은 학습 가능한 분산  $\epsilon_\theta$ 로 Langevin dynamics과 유사합니다. 게다가 11번째 공식을 이용해서 10번째의 공식을 아래와 같이 단순화 시킬 수 있습니다.

$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[ \frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t) \right\|^2 \right] \quad (12)$$

이러한 것은 denoising 스코어 매칭과 유사합니다. 공식 12는 Langevin-like의 역과정과 유사합니다. 요약하자면, 우리는 평균 근사치  $\mu_t$ 로 역 과정을 학습시킬 수 있으며 파라미터를 수정하여  $\boldsymbol{\epsilon}$ 를 예측할 수 있습니다.  $\boldsymbol{\epsilon}$ 를 예측하는 것은 Langevin dynamics과 유사하며 디퓨전 모델의 변수를 단순화 시킬 수 있습니다. 하지만 이는 ( $u_t$  대신  $\boldsymbol{\epsilon}$ 을 예측하는) 기준과는 다른 학습 방식이며 유용성을 증명해야 합니다.

### Data scaling, reverse process decoder, and $L_0$

우리는 이미지의 값을 0~255를 [-1, 1]의 범위로 스케일링을 진행합니다. 이러한 것은 데이터의 시작이 정규분포에 따름을 알 수 있습니다. 개별적인 log likelihood를 얻기 위해 우리는 역 과정의 마지막 구간을

$$\text{Gaussian } \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_\theta(\mathbf{x}_1, 1), \sigma_1^2 \mathbf{I})$$

를 따르는 독립적인 디코더로 구성합니다.

$$p_\theta(\mathbf{x}_0 | \mathbf{x}_1) = \prod_{i=1}^D \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x}_1, 1), \sigma_1^2) dx \quad (13)$$

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases} \quad \delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

$D$ 는 데이터의 차원입니다. 이산 연속 분포를 사용하는 VAE decoder와 autoregressive models과 유사하지만 우리는 여기서 개별 데이터의 길이가 손실이 없고 noise를 추가적으로 더할 필요도 없습니다.

### Simplified training objective

역과정과 디코더의 정의에 따라 공식 12와 13에 의해 유도된 잠재 구간은  $\theta$ 에 의해 미분 가능하며 학습이 가능하다. 우리는 다음과 같은 Loss를 사용하여 학습하는 것이 뛰어남을 발견했다.

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[ \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \right\|^2 \right] \quad (14)$$

$t$ 는 1~T 사이에 균일 분포이다. 이러한 로스는  $t$ 가 작을 때도  $t$ 가 클 때도 학습이 잘 이루어진다.

## Experiments

우리는  $T = 1000$ 을 모든 실험해서 설정해서 진행한다.  $\beta_1 = 10^{-4}$  to  $\beta_T = 0.02$ 로 선형적으로 증가한다.

### Sample quality

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

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

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
<b><math>\tilde{\mu}</math> prediction (baseline)</b>		
$L$ , learned diagonal $\Sigma$	$7.28 \pm 0.10$	23.69
$L$ , fixed isotropic $\Sigma$	$8.06 \pm 0.09$	13.22
$\ \tilde{\mu} - \tilde{\mu}_\theta\ ^2$	-	-
<b><math>\epsilon</math> prediction (ours)</b>		
$L$ , learned diagonal $\Sigma$	-	-
$L$ , fixed isotropic $\Sigma$	$7.67 \pm 0.13$	13.51
$\ \tilde{\epsilon} - \epsilon_\theta\ ^2$ ( $L_{\text{simple}}$ )	<b><math>9.46 \pm 0.11</math></b>	<b>3.17</b>

## Reverse process parameterization and training objective ablation

표2를 보면 variance를 고정시키고  $u$ 를 학습시킬 때 성능이 올라갈 수 있다. 하지만 목적 함수를 간단히 만들었을 경우 학습이 제대로 되지 않는다. fixed variance와 목적 함수 간단화를 사용할 경우 성능이 올라간다.

## Conclusion

우리는 diffusion model을 사용하여 좋은 품질의 이미지를 만든다. 그리고 학습에서 디퓨전 모델과 다양한 생성 모델과의 연결을 발견했다. diffusion model은 이미지 데이터에 대한 좋은 inductive bias가 있기 때문에 우리는 이에 대한 다양한 활용을 기대한다.

### DDPM의 중요 포인트

- encoder에서  $\beta$ 에 대해서 학습하지 않고 고정 상수로 변경 시킴.
  - 분산을 학습시키는 것이 아닌,  $\epsilon$ 을 학습 시켜 좀 더 쉬운 것을 학습하도록 변경 → 전체적인 로스의 간단화
  - residual connection 과 같이 전의 결과를 통해 이후의 이미지를 도출해낸다. 바로 이미지를 생성하는 것이 아니라 이전 이미지에서 노이즈가 얼마나 추가되었는지를 예측해 빼는 방식의 로스 계산을 사용한다.
- 기존의 복잡한 방식의 inductive bias를 주입해 모델이 잘 학습하도록 이미지가 잘 만들어지도록 유도한다.

후기.

기존의 다른 논문과 다르게 논문에서 처음보는 여러 용어들이 있어 말로 잘 풀어지지 않은 것 같다.

논문의 기본적인 흐름과 모델에 대한 전반적인 이해는 되었으나, 수식의 구체적인 부분에 대한 이해는 아직 부족하게 느낀다. 이후 추가적인 공부가 필요.

참고 : [https://www.youtube.com/watch?v=1j0W\\_lu55nc](https://www.youtube.com/watch?v=1j0W_lu55nc)