Conditional Diffusion Models

Jong Hyun Lee

2023.06.09

Introduction

발표자 소개

❖ **이종현 (Jong Hyun Lee)**

- 고려대학교 산업경영공학과 석사 과정 (2022.09 ~ Present)
- Data Mining & Quality Analytics Lab

❖ **Research Interests**

- Deep Generative Models
- Diffusion Models

❖ **Contact**

- E-mail: tomtom1103@korea.ac.kr

Introduction

Generative Models

❖ 생성모델 (Generative Models)

- Text 를 입력하면 그림을 그려주는 AI 로 큰 화재
- 엄청난 속도로 발전하는 Domain

2021

2023

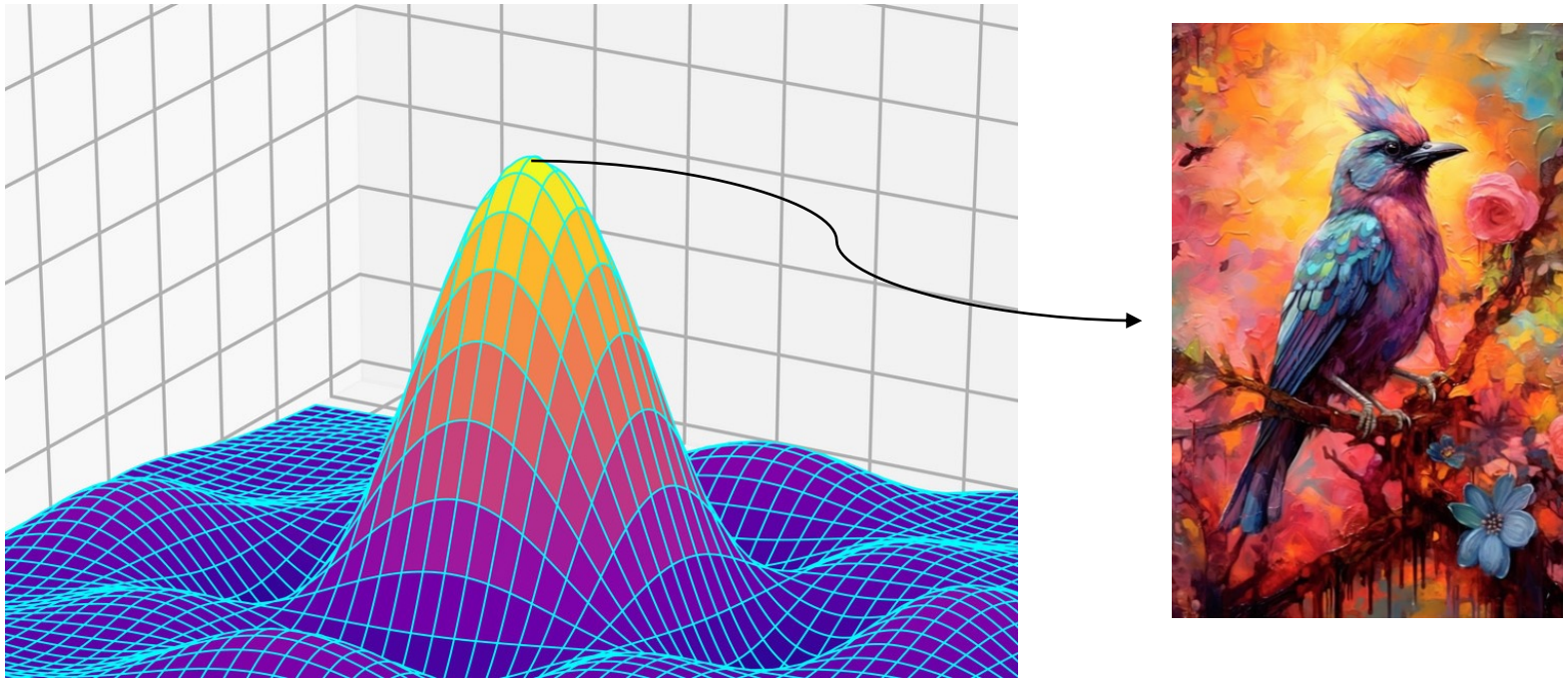
Frans, Kevin, Lisa Soros, and Olaf Witkowski. "Clipdraw: Exploring text-to-drawing synthesis through language-image encoders." Advances in Neural Information Processing Systems 35 (2022): 5207-5218
https://www.midjourney.com/showcase/recent/

Introduction

Generative Models

❖ 생성모델 (Generative Models)

- 생성모델은 본질적으로 데이터의 분포를 학습함
- 분포 학습 후, 해당 분포에서 Sampling 하는 것을 "생성" 이라고 칭함



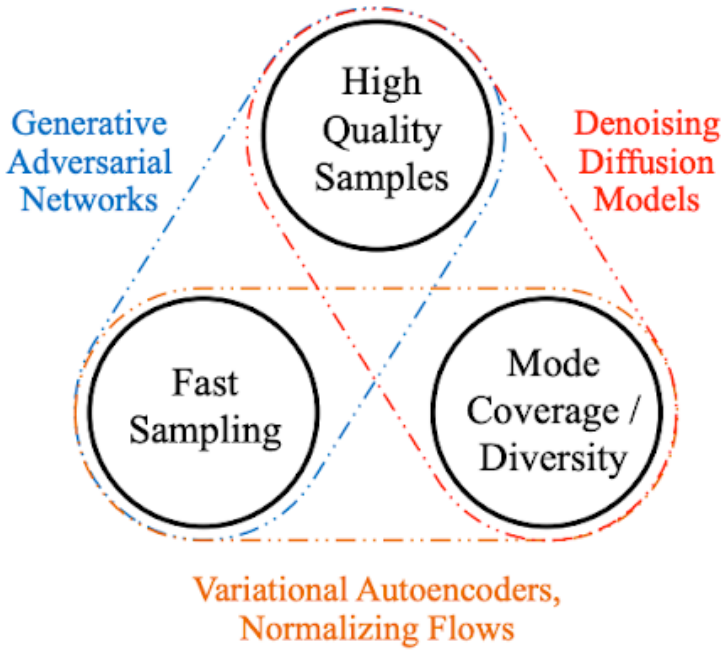
https://www.vectorstock.com/royalty-free-vector/big-data-3d-graph-vector-33923859

Introduction

Generative Models

❖ 생성모델 (Generative Models)

- The Generative Model Trilemma



❖ High Quality Samples: 모델이 높은 퀄리티의 이미지를 생성

❖ Mode Coverage/Diversity: 다양한 이미지를 생성

❖ Fast Sampling: 빠른 생성

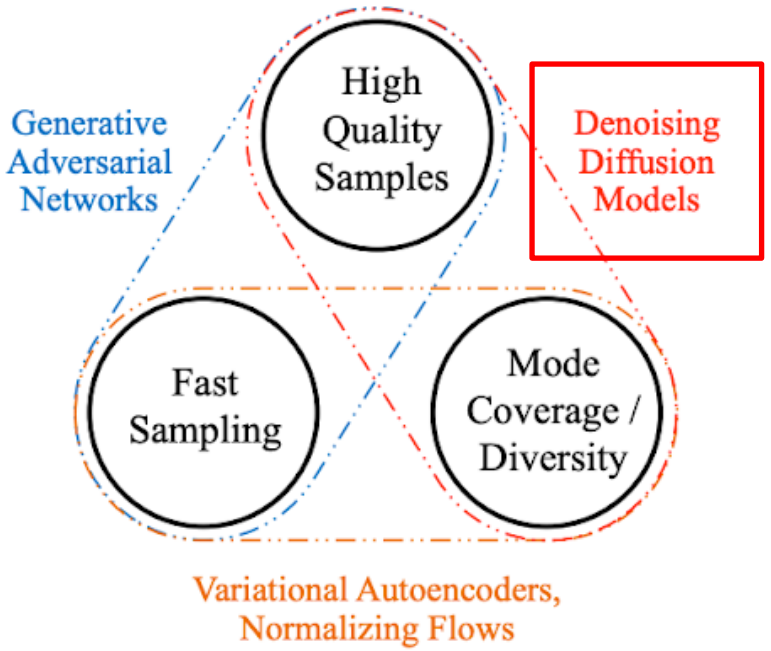
https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Introduction

Generative Models

❖ 생성모델 (Generative Models)

- The Generative Model Trilemma



❖ **High Quality Samples:** 모델이 높은 퀄리티의 이미지를 생성

❖ **Mode Coverage/Diversity:** 다양한 이미지를 생성

❖ Fast Sampling: 빠른 생성

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Introduction

Generative Models

❖ Diffusion Models

- Text 를 Image 로 바꿔 주는 Diffusion Model → Conditional Diffusion Models

DDPM

Denoising Diffusion Probabilistic Models

CFG

Classifier Free Guidance

LDM

Latent Diffusion Models

Introduction

Generative Models

❖ Diffusion Models

- Text 를 Image 로 바꿔 주는 Diffusion Model → Conditional Diffusion Models

DDPM

Denoising Diffusion Probabilistic Models

CFG

Classifier Free Guidance

LDM

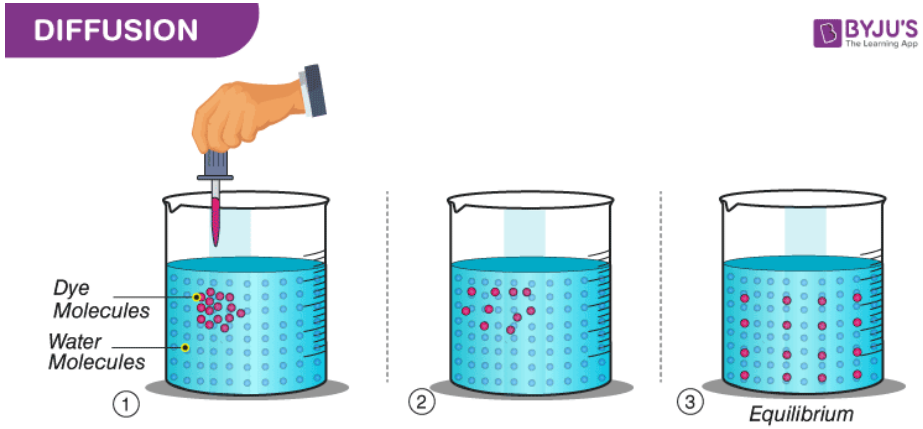
Latent Diffusion Models

Introduction

Diffusion

❖ Diffusion (확산): 액체나 기체에 다른 물질이 섞이고, 그것이 조금씩 번져가다 마지막엔 일률적인 농도로 바뀌는 현상

- 비정형 열역학 (Nonequilibrium Thermodynamics) 에 기초, 2015년 논문 " Deep unsupervised learning using nonequilibrium thermodynamics" 에서 Diffusion 을 최초로 딥러닝으로 모델링함



Deep unsupervised learning using nonequilibrium thermodynamics

J Sohl-Dickstein, E Weiss… - International …, 2015 - proceedings.mlr.press

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion …

☆ 저장 인용 932회 인용 관련 학술자료 전체 6개의 버전 »

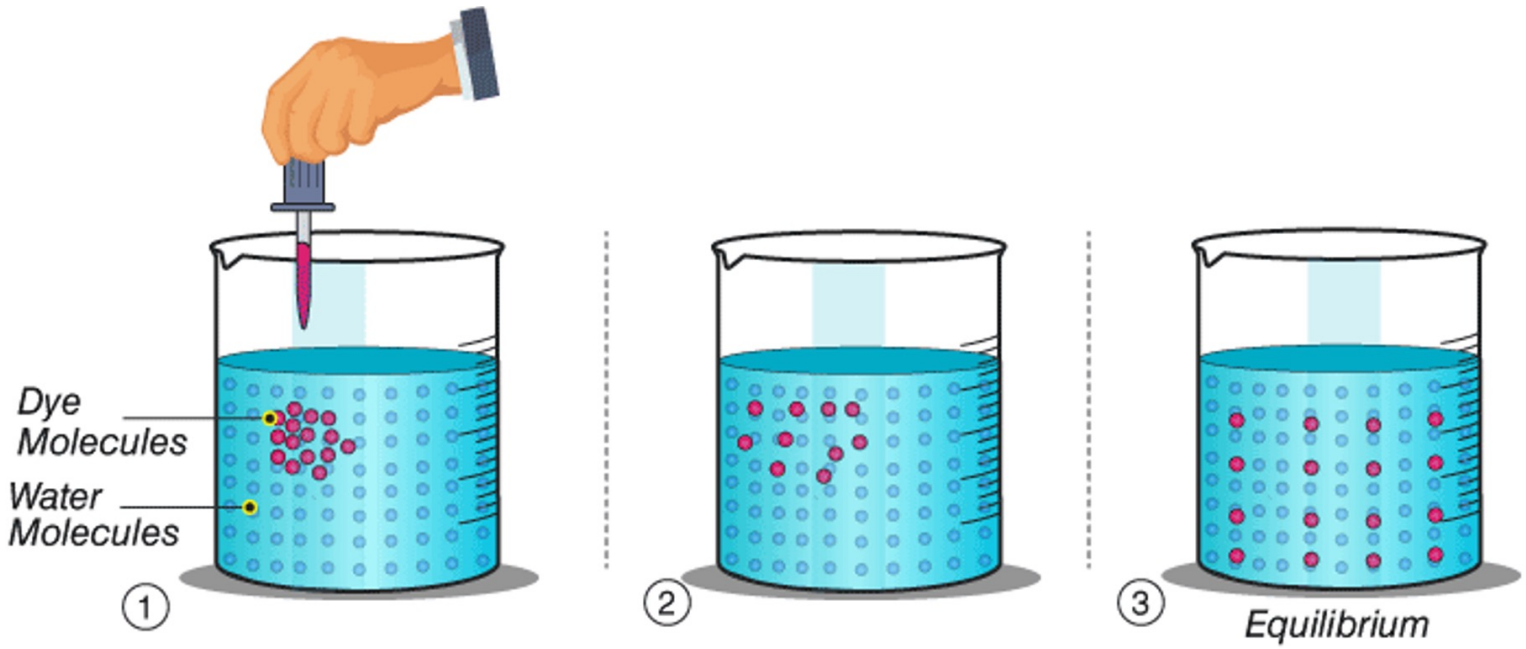
Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015

DDPM

Denoising Diffusion Probabilistic Models (DDPM)

❖ 2020년 "Denoising Diffusion Probabilistic Models" 에서 딥러닝으로 모델링 된 Diffusion 을 이미지 Domain 에 적용

- 핵심 Idea: Neural Net 에게 점진적으로 물질이 퍼져 나가는 과정을 학습시키면 반대 과정도 수행 할 수 있지 않을까?



Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

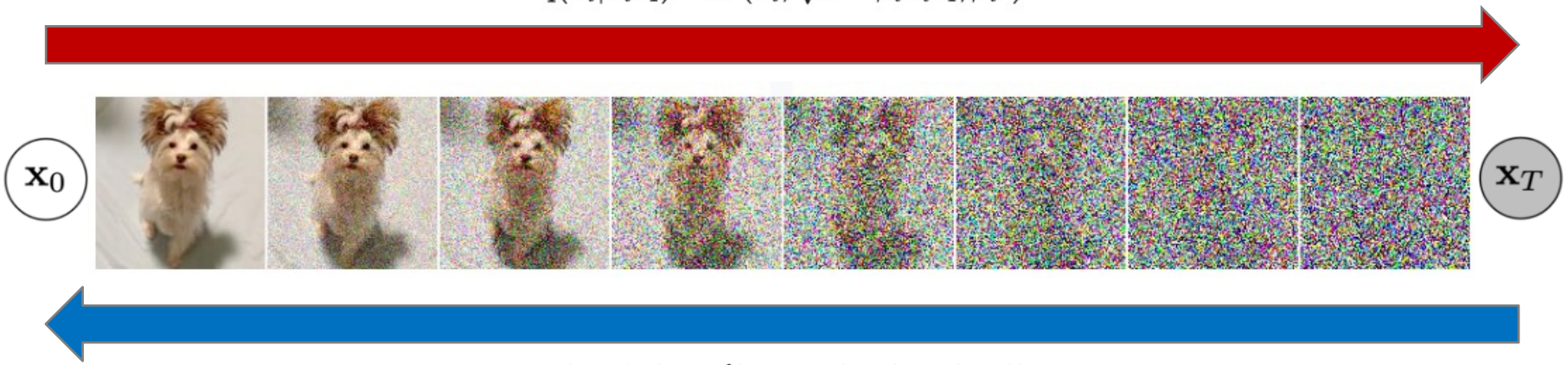
DDPM

Forward & Reverse Process

❖ Forward Process, Reverse Process

- Forward Process: 이미지 (x_0) 가 완전한 Gaussian Noise (x_T) 가 될 때까지 Gaussian Noise 를 점진적으로 추가하는 Markov Process
- Reverse Process: Gaussian Noise (x_T) 에서 점진적으로 Gaussian Noise 를 제거하여 이미지 (x_0) 를 복원하는 Markov Process

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



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

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

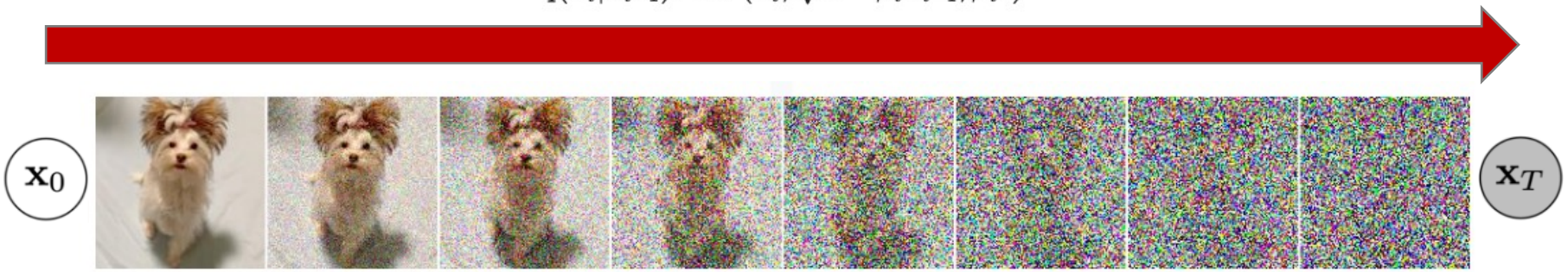
DDPM

Forward & Reverse Process

❖ Forward Process

- 이미지 (x_0) 가 완전한 Gaussian Noise (x_T) 가 될 때까지 Gaussian Noise 를 점진적으로 추가하는 Markov Process
- 현재 이미지가 주어졌을 때 1초 뒤 이미지는 평균이 $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$, 분산이 $\beta_t\mathbf{I}$ 인 Gaussian 의 분포를 따른다

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



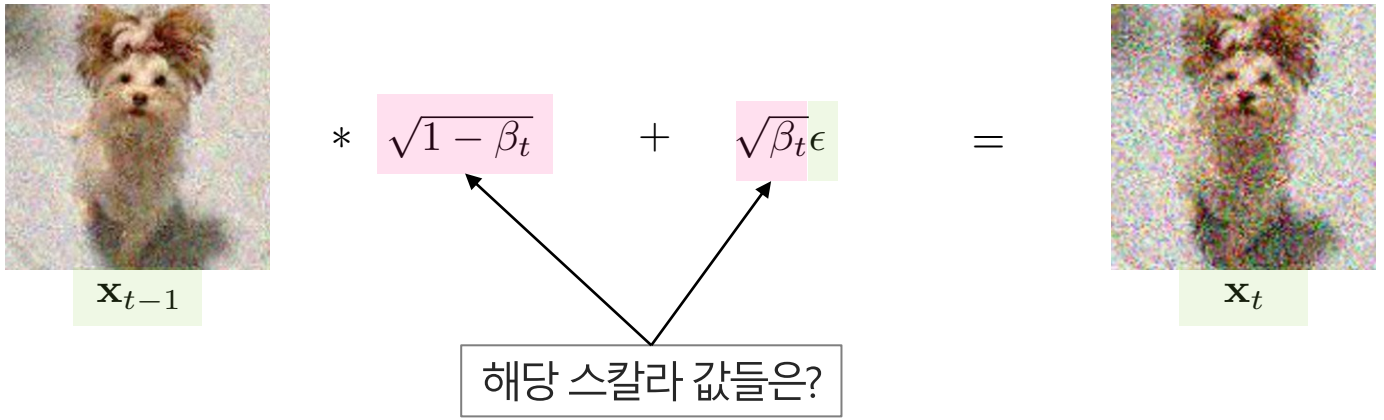
Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851
Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013)

DDPM

Forward & Reverse Process

❖ Forward Process

- 이미지 (x_0) 가 완전한 Gaussian Noise (x_T) 가 될 때까지 Gaussian Noise 를 점진적으로 추가하는 Markov Process
- 현재 이미지가 주어졌을 때 1초 뒤 이미지는 평균이 $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$, 분산이 $\beta_t\mathbf{I}$ 인 Gaussian 의 분포를 따른다
- VAE (Kingma, et.al.) 에서 제안한 Reparameterization Trick 으로 구현
- $\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon, \epsilon \sim \mathcal{N}(0, I)$



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

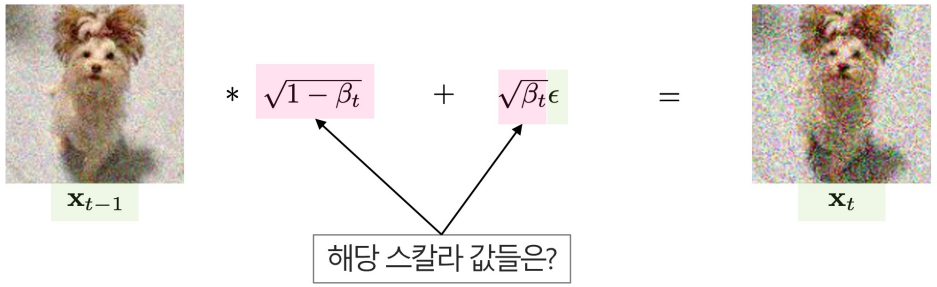
Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851
Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013)

DDPM

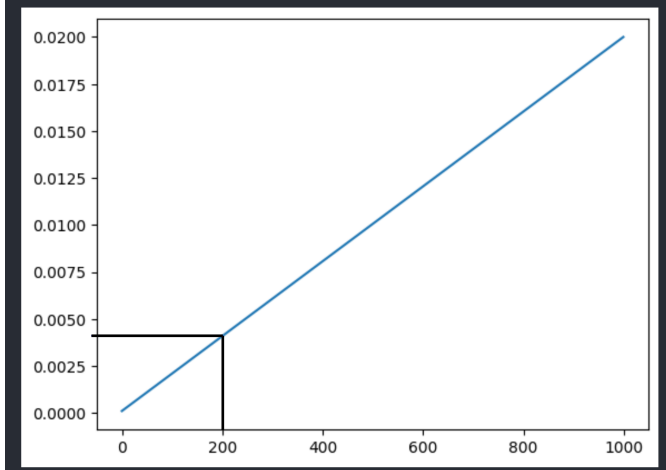
Forward & Reverse Process

❖ Forward Process

- Diffusion: 사전 정의된 Timestep 만큼 T 번 Forward process 를 수행하면 이미지는 Isotropic Gaussian (완전한 노이즈) 로 수렴해야 한다
- 이를 만족하기 위해 Noise Schedule 이란 것을 정의
- Noise Schedule: 베타값의 집합




```
1 T = 1000
2 betas = torch.linspace(0.0001, 0.02, T)
3 plt.plot(betas);
✓ 0.1s
```



❖ 현 시점이 200일 때 한번 Forward process 적용:

1. 이미지의 픽셀값에 $\sqrt{1-\beta_{200}}$ 를 Element wise multiplication
2. 이미지와 같은 차원의 랜덤 가우시안에서 샘플 추출
3. 랜덤 가우시안과 $\sqrt{\beta_{200}}$ 를 Element wise multiplication
4. 더해주면 201 시점의 노이즈가 더 추가된 이미지 생성

DDPM 논문에선 $T = 1000, \beta_0 = 0.0001, \beta_{1000} = 0.02$ 로 정의

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

Forward & Reverse Process

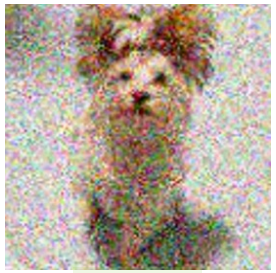
❖ Forward Process

- T=0 의 깨끗한 이미지로부터 T=500까지 가려면 Forward Process 500번 수행 → 비효율적
- Noise Schedule 을 수식적으로 다시 정리하면 한번에 수행 가능

$$\alpha_t = 1 - \beta_t$$
$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$
$$\begin{aligned}\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \\ &= \dots \\ &= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon\end{aligned}$$
$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right)$$

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

$\mathbf{x}_0$ $\quad * \quad \sqrt{\bar{\alpha}_t} \quad + \quad \sqrt{1-\bar{\alpha}_t}\epsilon \quad =$



$\mathbf{x}_t$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

Forward & Reverse Process

❖ Forward Process

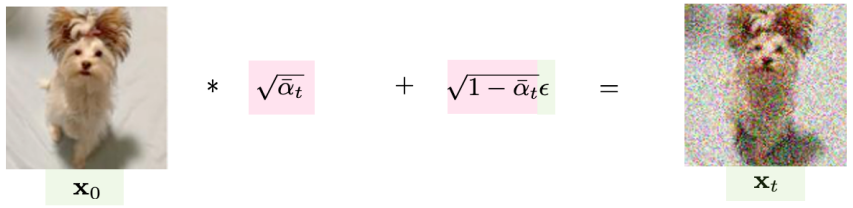
- Noise Schedule 을 수식적으로 다시 정리하면 한번에 수행 가능
- Noise Schedule 의 의미: Forward process 를 진행할 수록

  1. 이미지 분포의 평균은 0에 근접하도록

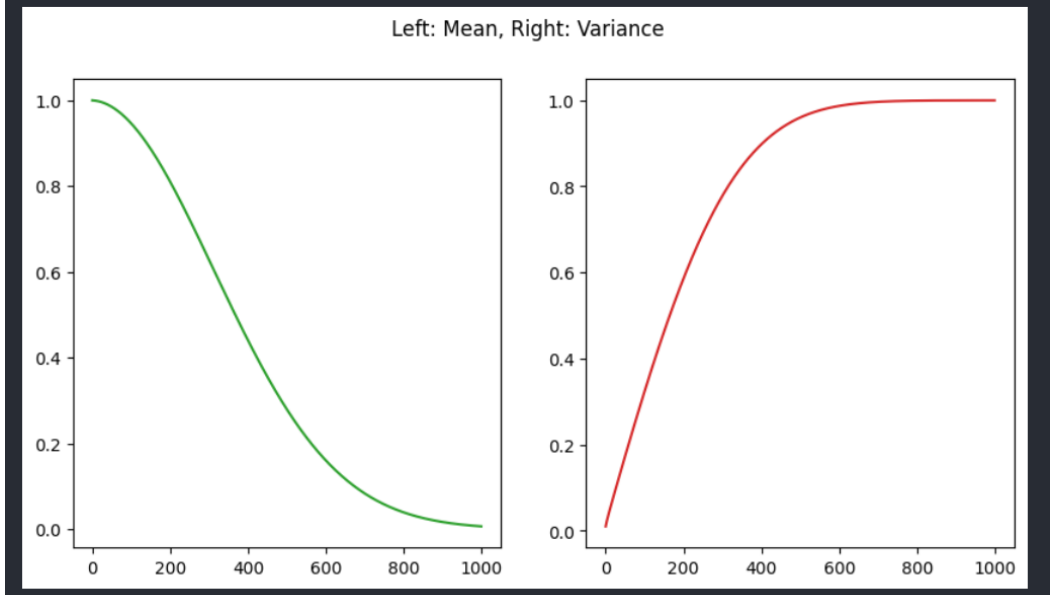
  2. 이미지 분포의 분산은 1에 근접하도록

  → Normal Gaussian 이 되도록

$$\alpha_t = 1 - \beta_t$$
$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$
$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, I)$$




```
1  T = 1000
2  betas = torch.linspace(0.0001, 0.02, T)
3  alphas = 1. - betas
4  alphas_bar = torch.cumprod(alphas, axis=0)
5  sqrt_alphas_bar = torch.sqrt(alphas_bar)
6  sqrt_one_minus_alphas_bar = torch.sqrt(1-alphas_bar)
7
8  fig, (ax1, ax2) = plt.subplots(1,2, figsize=(10,5))
9  fig.suptitle('Left: Mean, Right: Variance')
10 ax1.plot(sqrt_alphas_bar, 'tab:green');
11 ax2.plot(sqrt_one_minus_alphas_bar, 'tab:red');
✓ 0.1s
```



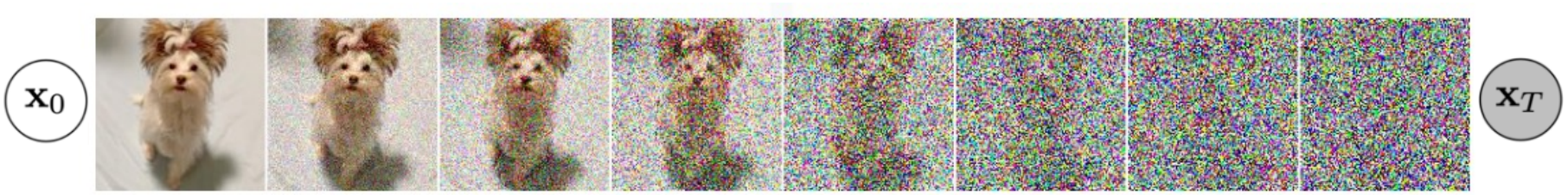
Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

Forward & Reverse Process

❖ Reverse Process

- Reverse Process: Gaussian Noise (x_T) 에서 점진적으로 Gaussian Noise 를 제거하여 이미지 (x_0) 를 복원하는 Markov Process
- DDPM에서는 Reverse Process 의 분산을 고정, 평균을 학습



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

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

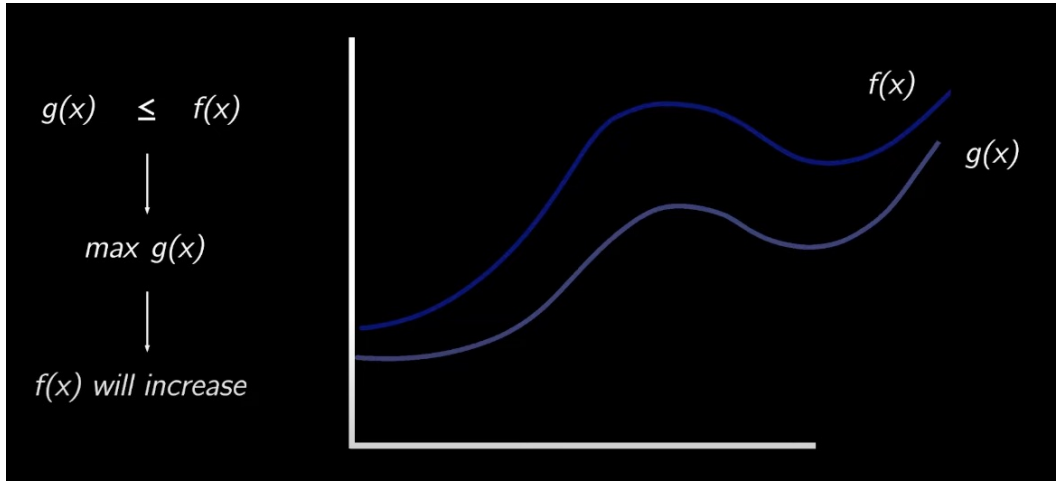
DDPM

Forward & Reverse Process

❖ Reverse Process

- 생성모델의 추상적인 Loss Function: $\max.\log(p_\theta(\mathbf{x}_0))$

  Maximize log likelihood → Untractable
- Proxy Loss: Variational Lower Bound 사용
- VLB 는 모든 구간에 대해서 실제 Likelihood 보다 작도록 설정



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

Prior Matching Term | Denoising Term | Reconstruction Term

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851
https://www.youtube.com/watch?v=HoKDTa5jHvg&t=571s&ab_channel=Outlier

DDPM

Forward & Reverse Process

❖ Reverse Process

- **Prior Matching Term**
  - T 까지 Forward Process 를 수행 할 때, 이미지의 사전에 정의한 분포 (Prior) 와 유사하도록 Minimize KL-Divergence
  - Forward Process 자체를 가우시안으로 정의 했기 때문에 **상수취급 가능**

$$\mathbb{E}_q \Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T}$$

Prior Matching Term

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

Forward & Reverse Process

❖ Reverse Process

- **Prior Matching Term**
  - T 까지 Forward Process 를 수행 할 때, 이미지의 사전에 정의한 분포 (Prior) 와 유사하도록 Minimize KL-Divergence
  - Forward Process 자체를 가우시안으로 정의 했기 때문에 **상수취급 가능**
- **Reconstruction Term**
  - 마지막 완전한 이미지로 갈 때 Maximize log likelihood
  - 전체 스텝 중 한번만 계산되기 때문에 비중이 아주 작음 → **상수취급 가능**

$$\mathbb{E}_q \Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T}$$

Prior Matching Term

$$\underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Big]$$

Reconstruction Term

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

Forward & Reverse Process

❖ Reverse Process

- **Prior Matching Term**
  - T 까지 Forward Process 를 수행 할 때, 이미지의 사전에 정의한 분포 (Prior) 와 유사하도록 Minimize KL-Divergence
  - Forward Process 자체를 가우시안으로 정의 했기 때문에 **상수취급 가능**
- **Reconstruction Term**
  - 마지막 완전한 이미지로 갈 때 Maximize log likelihood
  - 전체 스텝 중 한번만 계산되기 때문에 비중이 아주 작음 → **상수취급 가능**
- **Denoising Term**
  - DDPM 의 핵심

$$\mathbb{E}_q \Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T}$$

Prior Matching Term

$$\underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \Big]$$

Reconstruction Term

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}$$

Denoising Term

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

Loss Function

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}$$

Denoising Term

❖ Loss Function

- 실제로 minimize 하고 싶은 것: $D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)$
- $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ 를 모른다는 문제가 있지만, $\mathbf{x}_0$ 를 컨디셔닝 해도 Reverse Process 의 Markov 정의가 깨지지 않음
- 공교롭게도 $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)$ 는 Bayes Rule 로 실제 값을 계산 가능

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

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

$$\propto \exp\Big(-\frac{1}{2}\big(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big)$$

$$= \exp\Big(-\frac{1}{2}\big(\frac{\mathbf{x}_t^2 - 2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1} + \alpha_t\mathbf{x}_{t-1}^2}{\beta_t} + \frac{\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1} + \bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big)$$

$$= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\big)\Big)$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

DDPM

Loss Function

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}$$

Denoising Term

❖ Loss Function

- 실제로 minimize 하고 싶은 것: $D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)$
- $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ 를 모른다는 문제가 있지만, $\mathbf{x}_0$ 를 컨디셔닝 해도 Forward Process 의 Markov 정의가 깨지지 않음
- 공교롭게도 $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)$ 는 Bayes Rule 로 실제 값을 계산 가능

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

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

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

즉 원본 이미지 $\mathbf{x}_0$ 와 현 시점 이미지 $\mathbf{x}_t$ 가 있을 때,
$\mathbf{x}_{t-1}$ 은 평균이 $\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)$,
분산이 $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$ 인 가우시안을 따른다

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

DDPM

Loss Function

❖ Loss Function

- 정답 분포 $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)$ 와 모델이 예측하는 분포 $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$) 가 최대한 비슷하도록 학습
- 정답 분포의 평균 $\tilde{\boldsymbol{\mu}}_t=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_t\right)$과 분산 $\tilde{\beta}_t=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$ 중 미지수는 ϵ_t
- 모델은 ϵ_t, 즉 해당 시점의 노이즈만 예측하면 된다

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

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}}$$

Denoising Term

$$\min .\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right\|^2=\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{\left(1-\bar{\alpha}_t\right)} \boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

DDPM

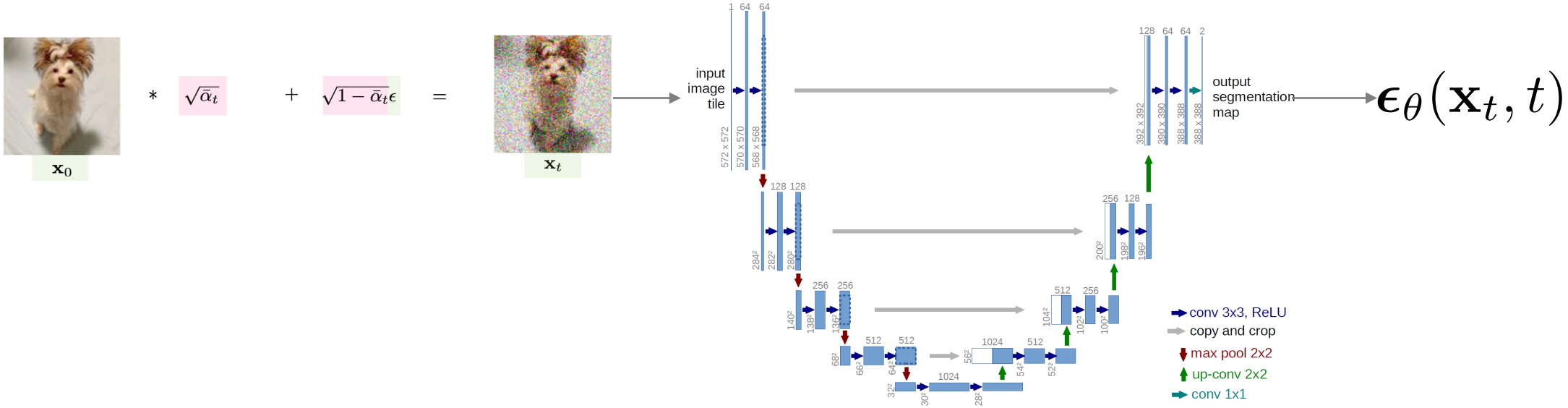
Training

- Model Architecture: U-Net
  - Input 과 Output 이 같은 차원 (크기) 인 모델 구조
  - Residual Blocks, Downsampling Blocks,

    Attention Layers, Skip Connection 으로 구성

Algorithm 1 Training

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



Intuition: $\mathbf{x}_0$ 를 $\mathbf{x}_t$ 로 만드는데 추가된 노이즈를 예측하는 모델

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

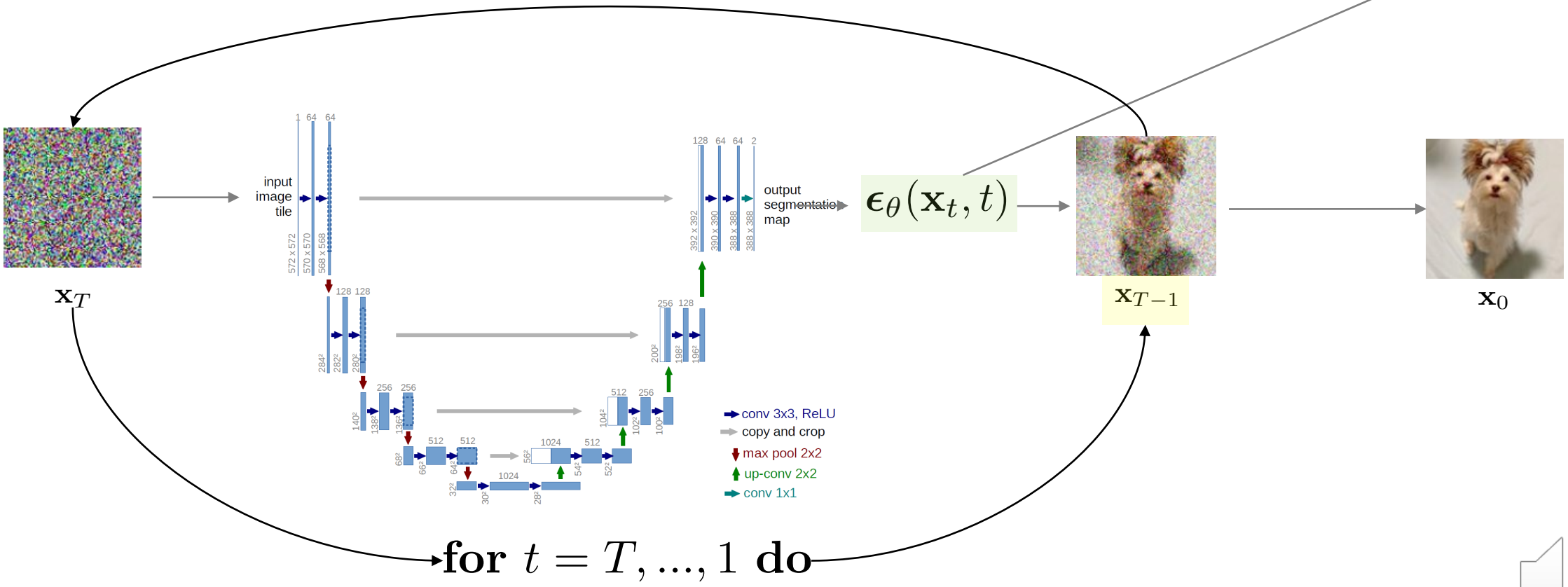
Sampling

❖ Sampling: Gaussian Noise 에서 시작,

각 Timestep 에 따라서 점진적으로 예측된 Noise 제거

Algorithm 2 Sampling

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



Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

DDPM

Results

❖ Results

- 학습에 사용된 데이터셋 속 이미지와 유사한 샘플들 생성하는데 성공
- 이때 까지만 해도 아직 GAN 보다 성능이 좋지 못함
- 그럼에도 불구하고 생성모델의 새로운 Paradigm 개척 – 현재까지 인용 수 약 1,950
- DDPM 은 Unconditional Model: $\mathbf{x} \sim p(\mathbf{x})$ 가능, $\mathbf{x} \sim p(\mathbf{x} \mid \mathbf{y})$ 불가능

Figure 3: LSUN Church samples. FID=7.89

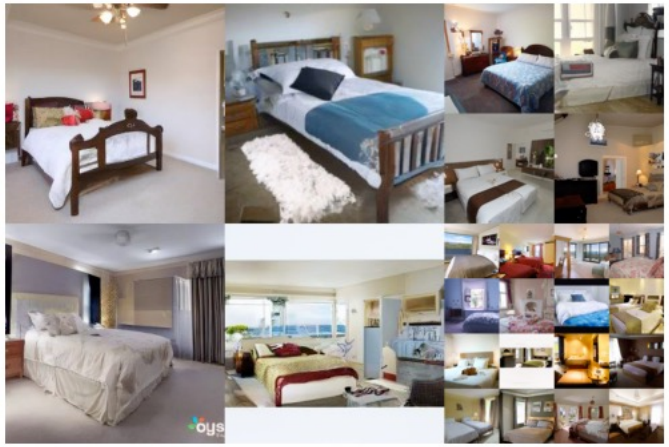


Figure 4: LSUN Bedroom samples. FID=4.90

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in Neural Information Processing Systems* 33 (2020): 6840-6851

Diffusion Models

Generative Models

DDPM

Denoising Diffusion Probabilistic Models

CFG

Classifier Free Guidance

LDM

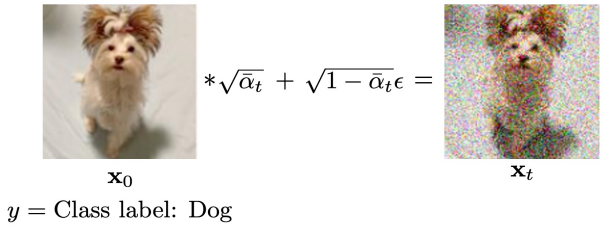
Latent Diffusion Models

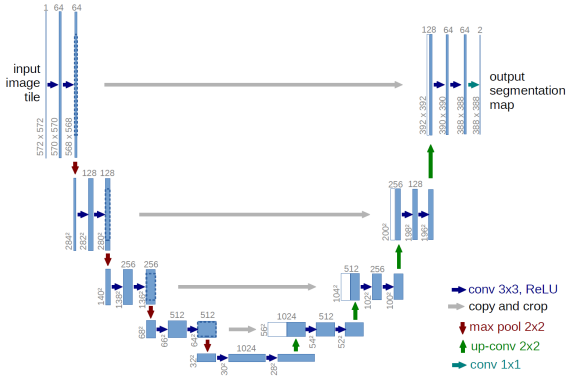
CFG

Classifier Free Guidance

❖ $\mathbf{x} \sim p(\mathbf{x} \mid \mathbf{y})$

- Diffusion Model 에 Conditioning information 주는 법: $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}))$
- 학습 과정에서 $\mathbf{x}_t$ 와 Class label/Encoded text $\mathbf{y}$ 를 U-Net 에 넣어 DDPM 학습하는 방식으로 가능
- **하지만 사실상 이런 방식은 효과적이지 않다는 게 실험적으로 밝혀짐**
- 이에 "Diffusion Models Beat GANs on Image Synthesis" (이하 ADM) 에서 **Classifier Guidance** 를 제안





$$\epsilon_\theta(\mathbf{x}_t, t, y)$$

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

Abstract

We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128, 4.59 on ImageNet 256×256, and 7.72 on ImageNet 512×512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512. We release our code at https://github.com/openai/guided-diffusion.

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." *Advances in Neural Information Processing Systems* 34 (2021): 8780-8794

CFG

Classifier Free Guidance

❖ Classifier Guidance

- Score Function: $\nabla \log p\left(\mathbf{x}_t\right) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t\right)$
- Score: 특정 시점에서 샘플이 주어졌을 때 Log – likelihood 의 Gradient → 해당 방향으로 이동하면 우도가 높아진다
- Bayes Rule 로 전개: Unconditional Model 을 Classifier 하나로 Conditional Model 로 바꿀 수 있다

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

$$
\hat{\epsilon}(\mathbf{x}_t) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t) - \gamma\sqrt{1-\bar{\alpha}_t}\nabla \log p(y \mid \mathbf{x}_t)
$$

Intuition: Sampling 할 때 Classifier 의 Gradient 만으로도 Conditional Sampling 이 가능하다

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." *Advances in Neural Information Processing Systems* 34 (2021): 8780-8794

CFG

Classifier Free Guidance

❖ Classifier Guidance

- 단점 1: Diffusion Model 만으로 Sampling 을 할 수 없고, 추가적인 Classifier Model 이 필요
- 단점 2: 모든 Noise level 에 대해서 학습이 된 Classifier 이 필요하기 때문에, 기존 Pretrained Model 사용 불가
- **Diffusion Model 만으로 Conditional Sampling 을 할 수 있지 않을까?**

Figure 3: Samples from an unconditional diffusion model with classifier guidance to condition on the class "Pembroke Welsh corgi". Using classifier scale 1.0 (left; FID: 33.0) does not produce convincing samples in this class, whereas classifier scale 10.0 (right; FID: 12.0) produces much more class-consistent images.

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." *Advances in Neural Information Processing Systems* 34 (2021): 8780-8794

CFG

Classifier Free Guidance

❖ Classifier Free Guidance

- Classifier Guidance 의 전개식을 다르게 전개
- CFG 가 의미하는 것: 추가적인 Classifier 없이 Gamma 로 Condition 이 반영되는 것을 조절 할 수 있다

$$\nabla \log p\left(\mathbf{x}_t \mid y\right) = \underbrace{\nabla \log p\left(\mathbf{x}_t\right)}_{\text{unconditional score}} + \gamma \underbrace{\nabla \log p\left(y \mid \mathbf{x}_t\right)}_{\text{adversarial gradient}}$$

1. Unconditional Score 을 좌변으로 이항

2 Adversarial Gradient 대입

$$\gamma \underbrace{\nabla \log p\left(y \mid \mathbf{x}_t\right)}_{\text{adversarial gradient}} = \nabla \log p\left(\mathbf{x}_t \mid y\right) - \underbrace{\nabla \log p\left(\mathbf{x}_t\right)}_{\text{unconditional score}}$$

$$\begin{aligned}
\nabla \log p\left(\mathbf{x}_t \mid y\right) &= \nabla \log p\left(\mathbf{x}_t\right) + \gamma\left(\nabla \log p\left(\mathbf{x}_t \mid y\right) - \nabla \log p\left(\mathbf{x}_t\right)\right) \\
&= \nabla \log p\left(\mathbf{x}_t\right) + \gamma \nabla \log p\left(\mathbf{x}_t \mid y\right) - \gamma \nabla \log p\left(\mathbf{x}_t\right) \\
&= \underbrace{\gamma \nabla \log p\left(\mathbf{x}_t \mid y\right)}_{\text{conditional score}} + \underbrace{(1-\gamma) \nabla \log p\left(\mathbf{x}_t\right)}_{\text{unconditional score}}
\end{aligned}$$

Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." *arXiv preprint arXiv:2207.12598* (2022)

Diffusion Models

Generative Models

DDPM

Denoising Diffusion Probabilistic Models

CFG

Classifier Free Guidance

LDM

Latent Diffusion Models

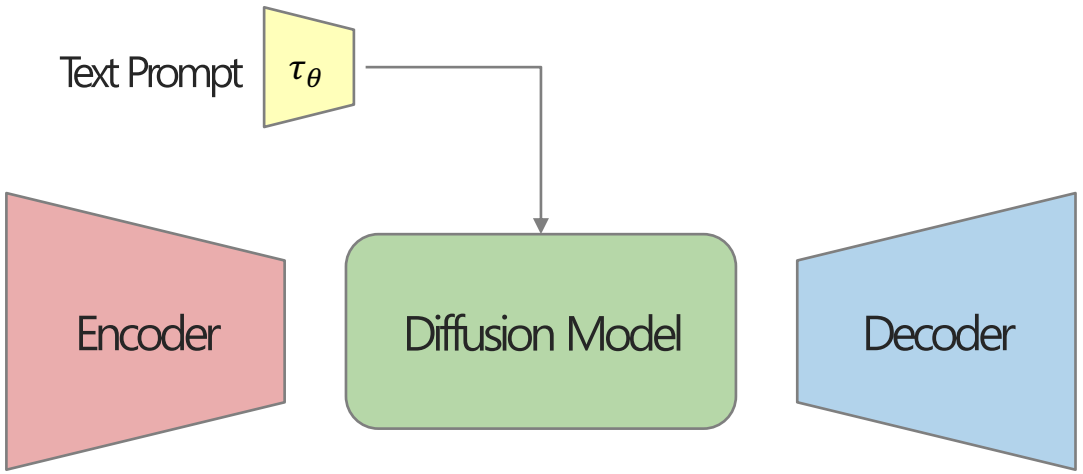
LDM

Stable Diffusion

❖ Latent Diffusion Models

- LDM 의 특정 모델이 Stable Diffusion → 학습하는데 약 7억원, 그럼에도 Open source 로 모델 공개
- 생성모델의 대중화를 선도, LDM 기반 다양한 연구 진행
- Diffusion Model 앞/뒤에 Autoencoder 을 추가함으로써 샘플링 시 Computational Cost 를 대폭 줄임

"AI Art" 의 Google Trends.
Stable Diffusion 공개일 기준으로 급격히 상승



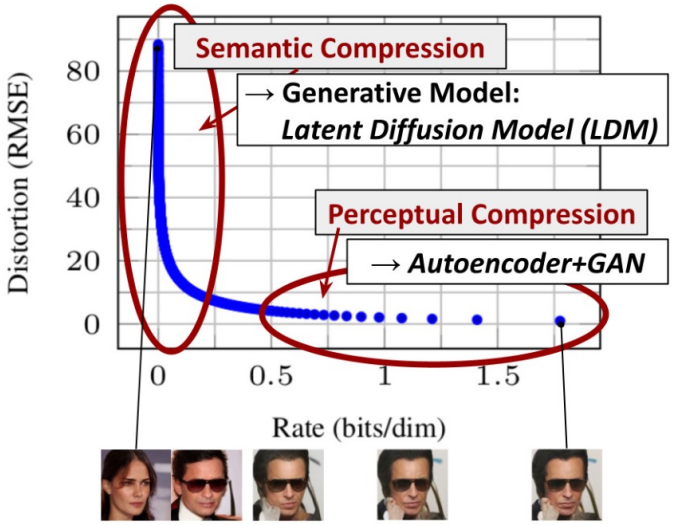
Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022

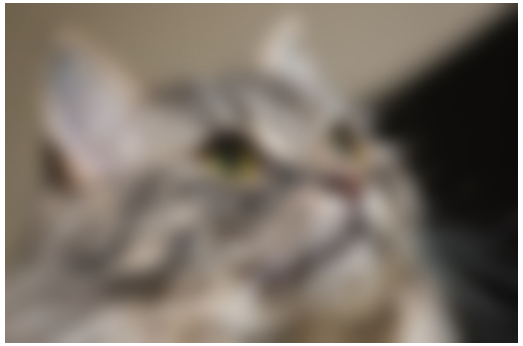
LDM

Stable Diffusion

❖ Perceptual and Semantic Compression

- 일반적인 생성모델은 이미지를 두 단계로 학습: Perceptual Compression, Semantic Compression
- **Perceptual Compression:** High Frequency Detail 들이 사라지지만 Semantic 은 유지되는 구간
- **Semantic Compression:** 실제 데이터의 본질이 Abstract 하게 학습되는 구간





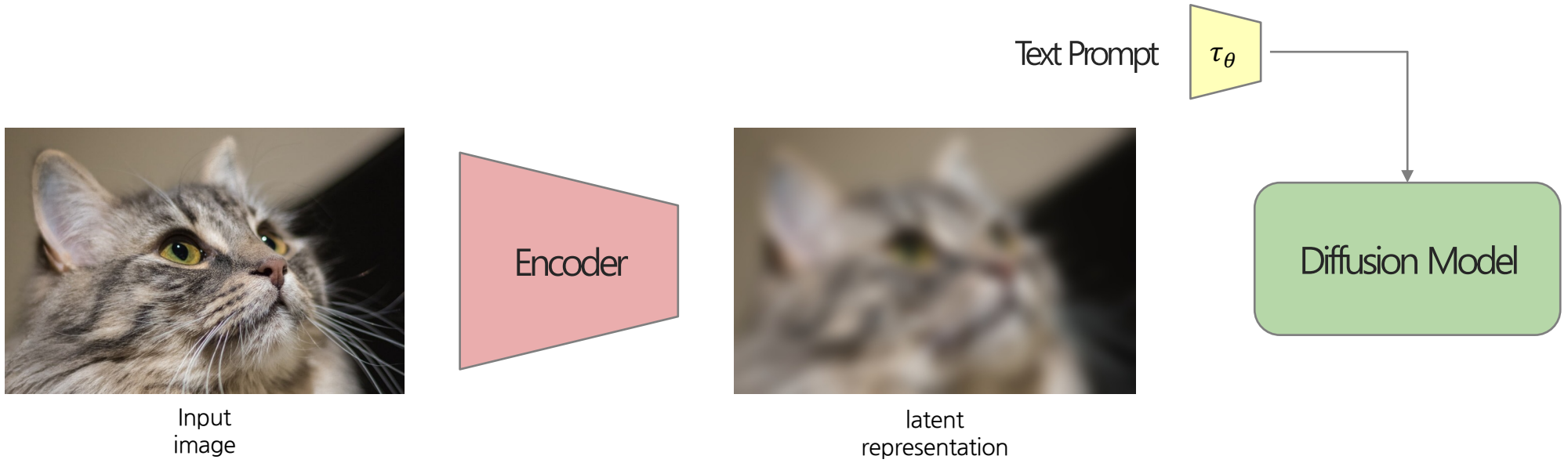
Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022

LDM

Stable Diffusion

❖ Perceptual and Semantic Compression

- 일반적으로 Diffusion Model 은 Perceptual Compression 단계에서 과도하게 오래 걸림
- 저자들은 **오토인코더로 Perceptual Compression** 을 진행한 뒤, **Semantic Compression 을 Diffusion** 의 task 로 넘김
- 오토인코더가 Perceptual Compression 만 진행하고 Semantics 를 남겨두므로 이미지의 Inductive Bias 그대로 사용 가능
- 이에 일반적인 모델을 **Pixel Space Model,** Compression 을 진행한 모델을 **Latent Space Model** 이라 함



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022

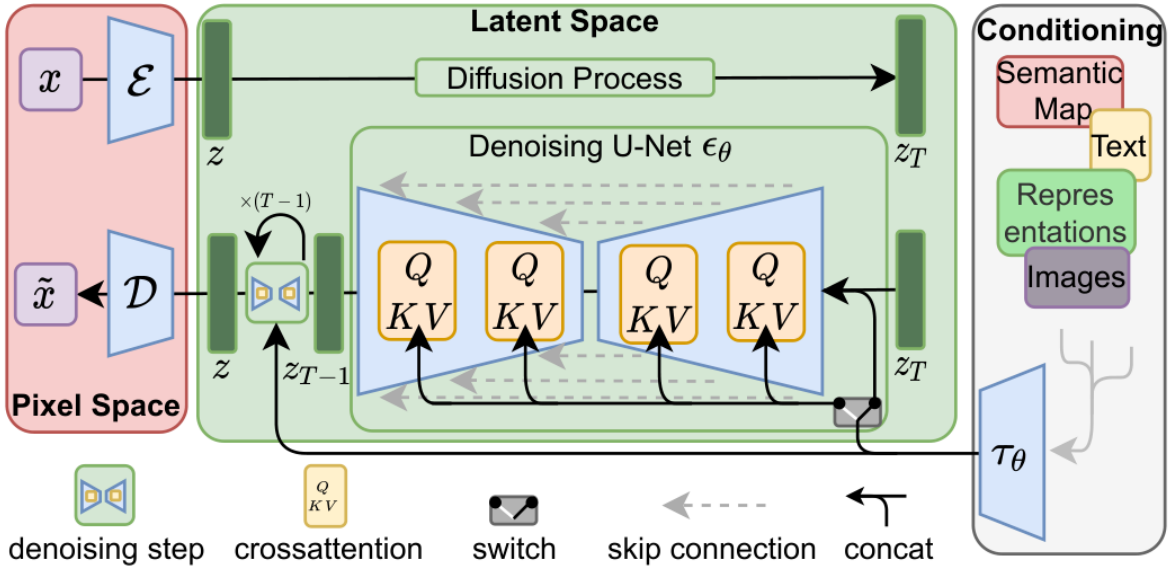
LDM

Stable Diffusion

❖ Architecture

- 트랜스포머의 텍스트 인코더 블럭 사용, U-Net 에 Cross Attention 추가하여 Condition 주입
- Stable Diffusion 은 텍스트 인코더를 CLIP 의 텍스트 인코더로 사용

$$x \in \mathbb{R}^{H \times W \times 3}, z \in \mathbb{R}^{h \times w \times c}$$
$$z = \mathcal{E}(x)$$
$$\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$$



Query: z_t
Key: $\tau_\theta(y)$
Value: $\tau_\theta(y)$

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, \tau_\theta(y)\right)\|_2^2\right]$$

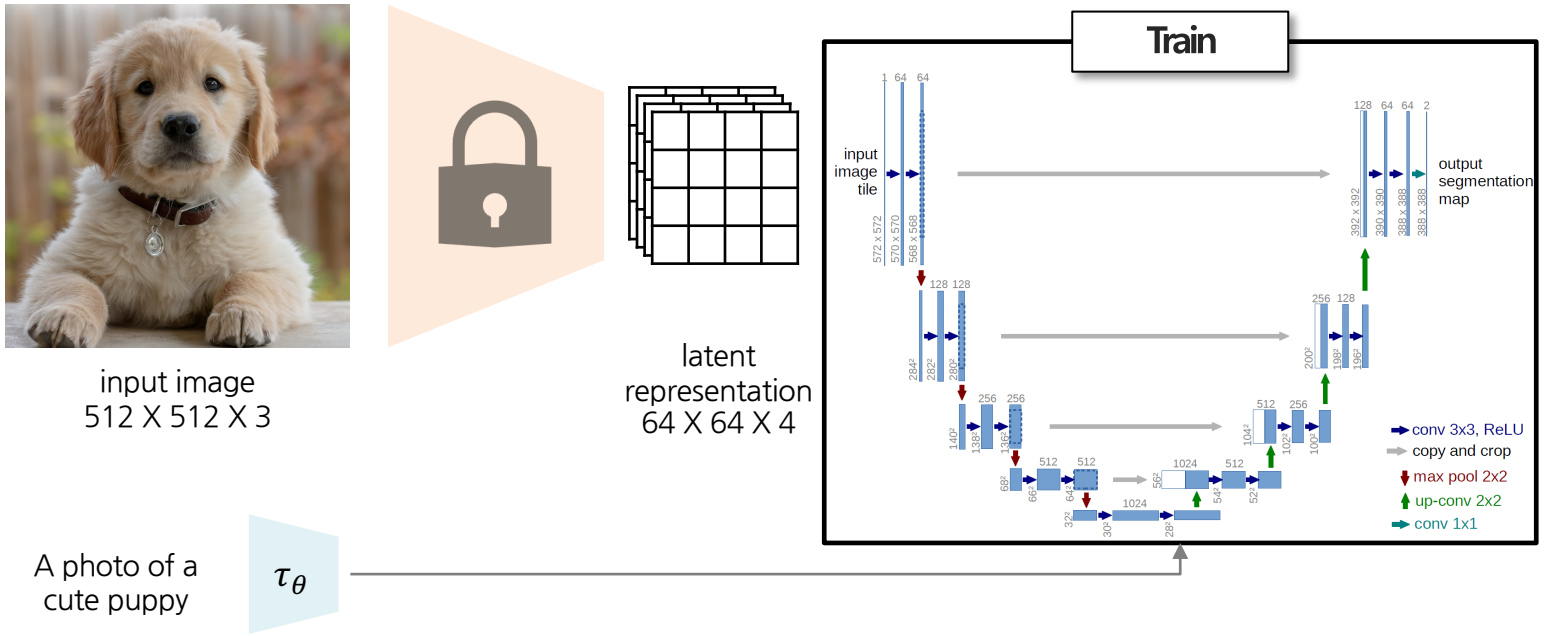
Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022

LDM

Stable Diffusion

❖ Training

- 학습된 autoencoder를 사용하여, 이미지에 대한 **latent representation 획득**
- Latent representation을 이용하여 Diffusion Model 을 학습



$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, \tau_\theta(y)\right)\|_2^2\right]$$

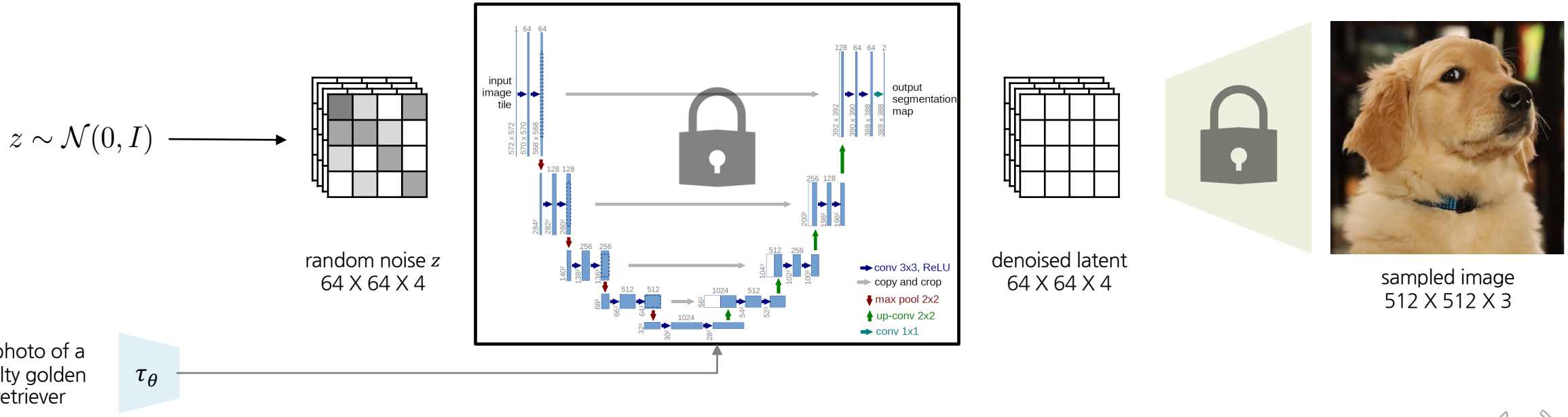
Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022

LDM

Stable Diffusion

❖ Sampling

- 가우시안에서 latent와 같은 사이즈의 random noise 획득, Text prompt 를 Text 인코더에 입력
- 획득한 random noise로 **sampling을 진행하고, sampling 결과에 대한 decoding을 진행하여 이미지 생성**
- **이때 CFG 의 gamma 로 얼만큼 Text 를 반영할지 결정 (base: 7.5)**



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022

LDM

Stable Diffusion

❖ Other Conditions

- Text prompt 이외에도 다양한 컨디션에 모델을 학습하여 여러가지 Task 수행
- E.g. Semantic synthesis, Super Resolution…
- 학습 시 $L_{LDM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, \tau_\theta(y)\right)\|_2^2\right]$, 컨디션 $\tau_\theta(y)$ 과 컨디셔닝 방법을 변경

	Conditioning Method
Text	Text 인코더, U-Net 의 Cross-Attn. 에 주입
Semantic Maps	Downsampling 된 semantic map 을 latent z 와 concat (channel dim.)
Super Resolution	Semantic maps 와 동일

Figure 9. A *LDM* trained on 256^2 resolution can generalize to larger resolution (here: 512×1024) for spatially conditioned tasks such as semantic synthesis of landscape images. See Sec. 4.3.2.

Figure 10. ImageNet 64→256 super-resolution on ImageNet-Val. *LDM-SR* has advantages at rendering realistic textures but SR3 can synthesize more coherent fine structures. See appendix for additional samples and cropouts. SR3 results from [72].

Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022

Conclusion

Diffusion Models

❖ DDPM

- Diffusion Model 의 시초, Forward & Reverse Process 와 Loss function
- Model Architecture: U-Net

❖ CFG

- 컨디션을 부여하여 샘플링 하는 방법론
- Classifier Guidance, Classifier Free Guidance

❖ LDM

- Stable Diffusion 논문, Diffusion Model 앞/뒤에 오토인코더 추가
- Text conditional generation, 이외의 다른 conditional generation

Prompt: A cat wearing a bachelor's cap, holding a sign that reads "thank you!".

Stable Diffusion 은 Pixel space model (Imagen, DeepFloyd IF) 과 다르게
text in image generation 을 잘 못합니다.