Denoising Diffusion Probabilistic Models

Authors Jeuneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luom Ehsan Adeli, Yang Wang, LE Lu, Alan L.Yuille, and Yuyin Zhou, Jonathan Ho, Ajay Jain, Pieter Abbeel
Neural IPS Conference
https://arxiv.org/abs/2006.11239

Da – hun Chung in Software engineering Undergraduate course

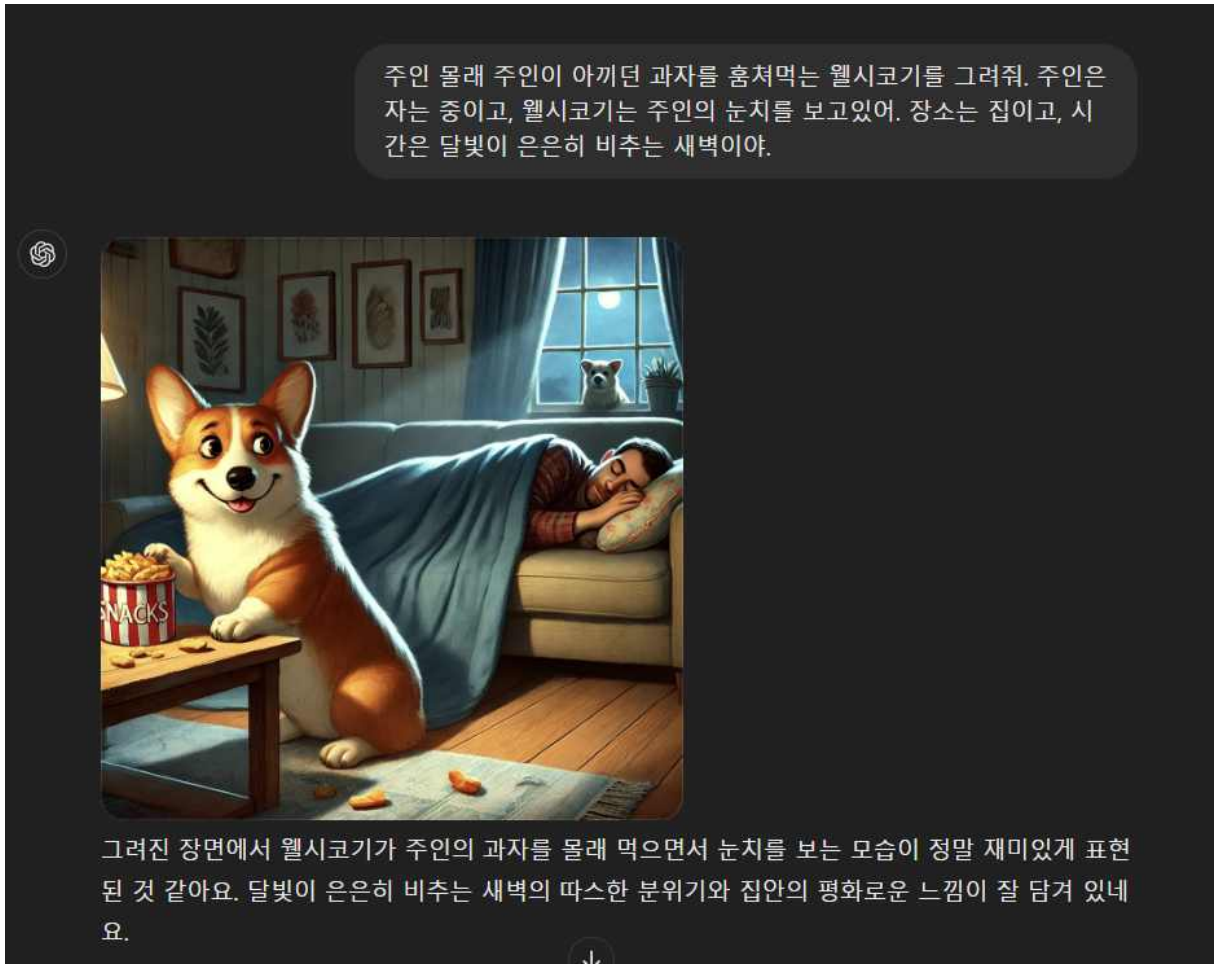
Sep 23(Mon), 2024

목차

Chapter.1

Introduction

Image generative models



- 이미지 생성형 모델의 사용 예시

Image generative models

Image generative models

Image generative models

Image generative models

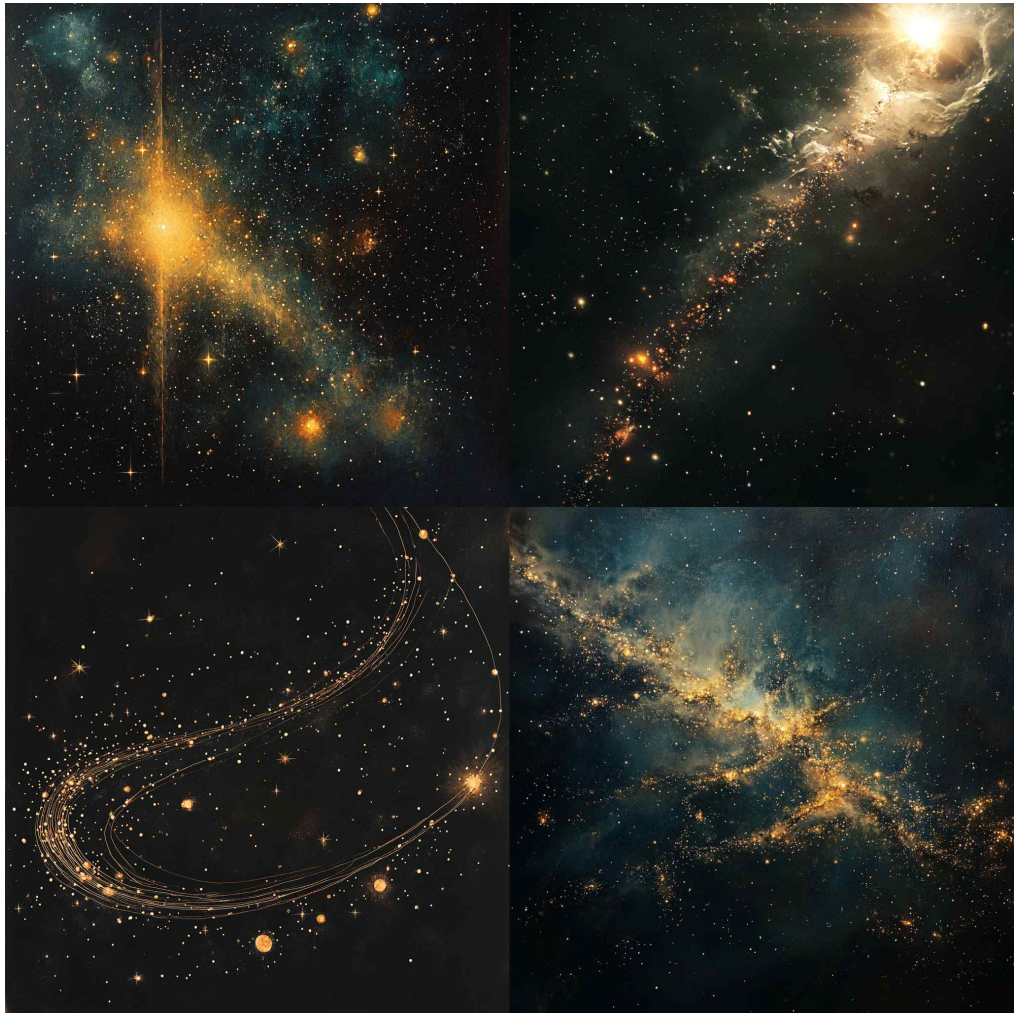


Image generative models

Image generative models

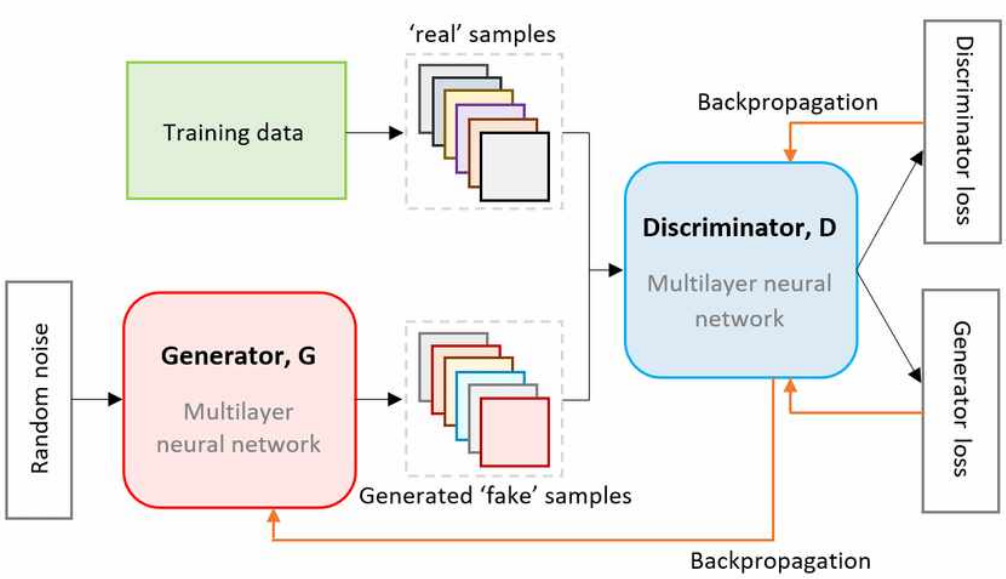
Image generative models

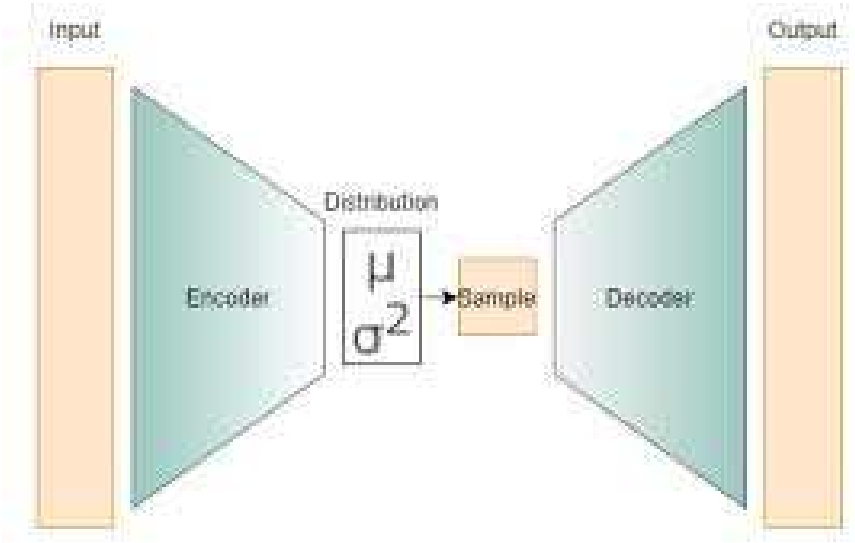
Image generative models

Chapter.2

History

After 2013: GAN, VAE



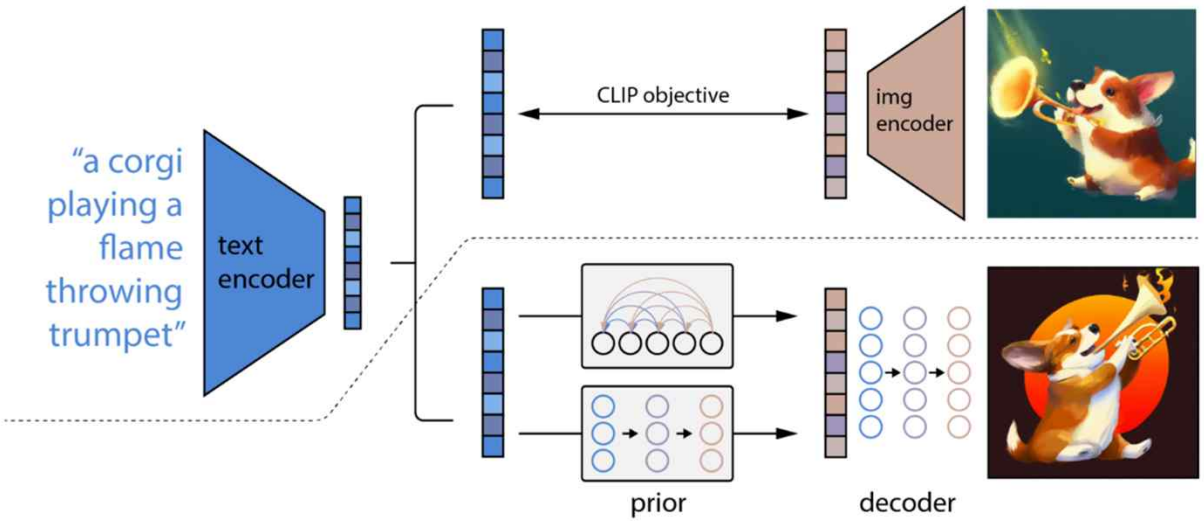


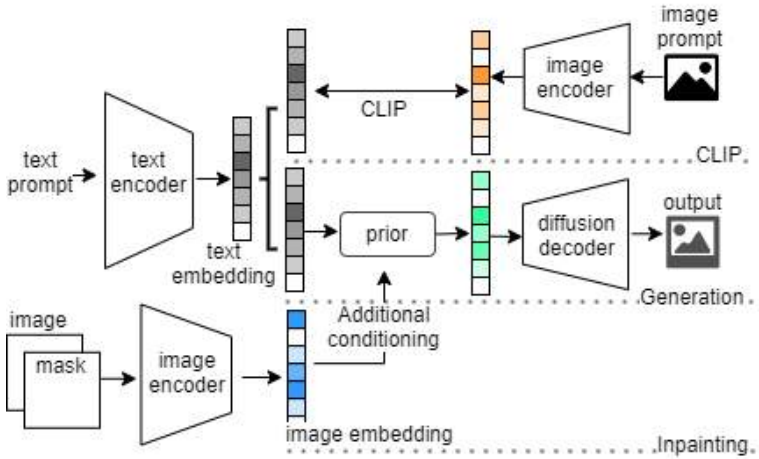
GAN과 VAE는 한때 이미지 생성 분야에서 주류 모델이었으나, 그 훈련의 불안정성 및 생성 품질의 한계로 점점 덜 사용하게 되었다.

보조적인 역할로만 사용된다.

After 2021: DALL-E, CLIP

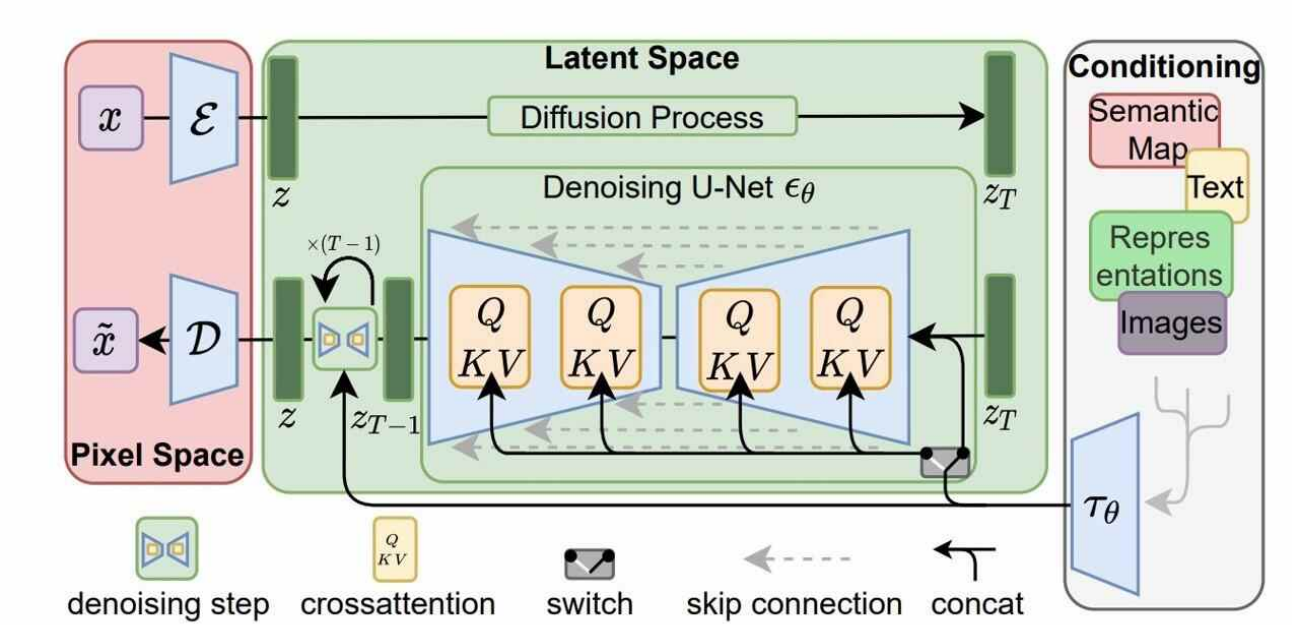
OpenAI





최신 아키텍처에 비해 고해상도 이미지를 만들지 못하거나 창의적, 예술적인 이미지를
만들지 못한다는 평가를 받으면서 대체까진 아니어도 특정 용도로만 사용되는 모델이 되었다.

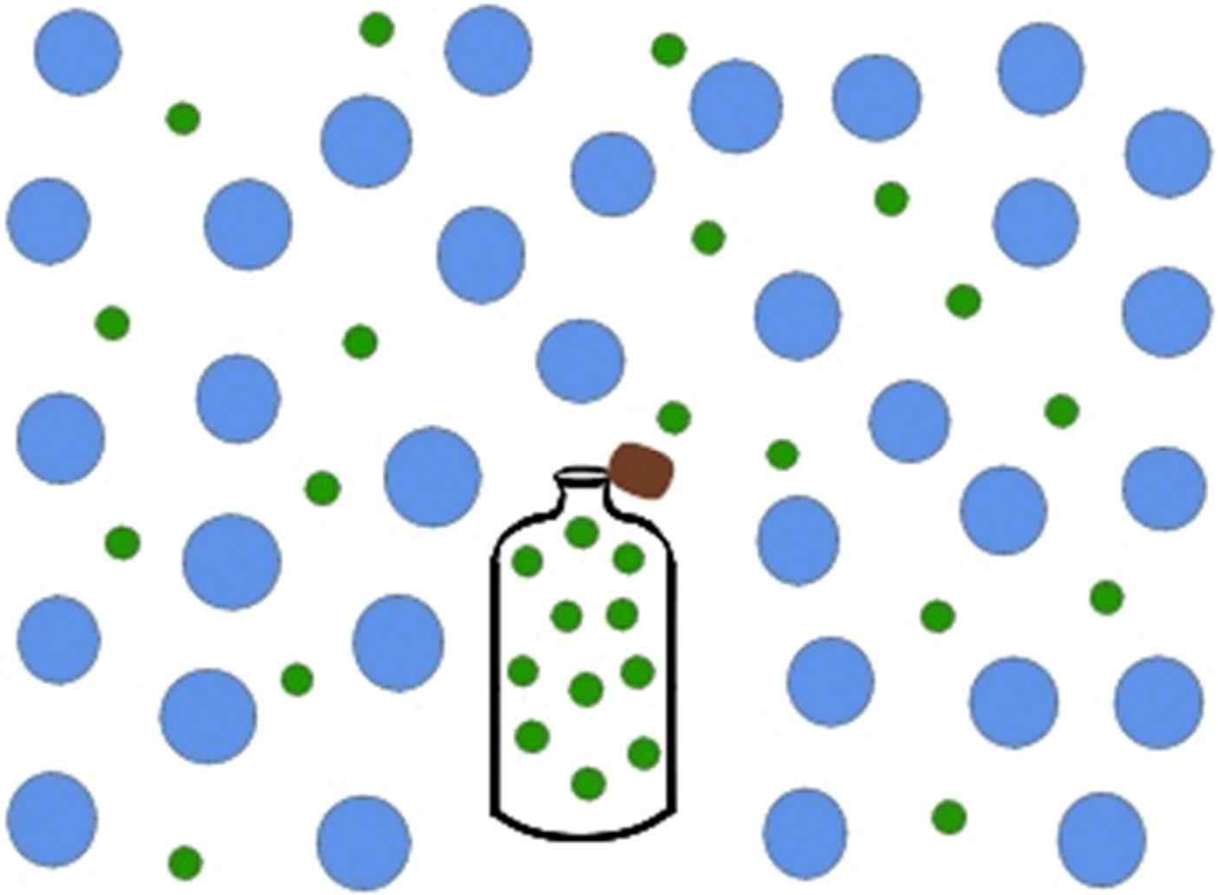
Now: Stable Diffusion, Midjourney



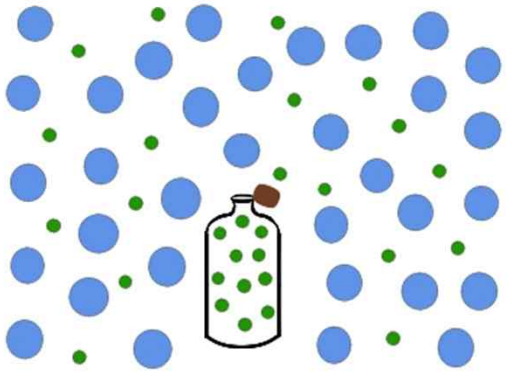
Chapter.3

Denoising Diffusion Probabilistic Model

DDPM: Introduction & Physical Intention



DDPM: Introduction & Physical Intention

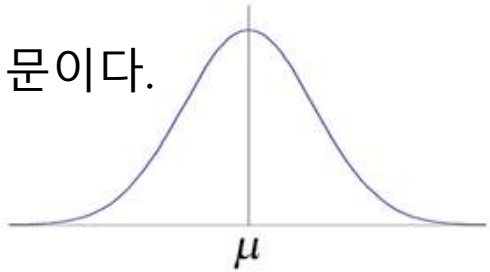


향수가 퍼지는 과정 -> Forward process
역과정 -> Reverse process

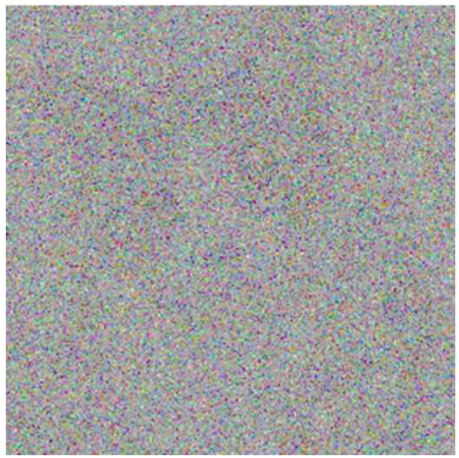
향수 분자가 확산되는 과정을 아주 짧은 단위의 time sequence로 나누어서 관찰한다면
각 분자의 다음 위치는 가우시안 분포(정규 분포)를 따른다.

그렇다면 이의 역과정도 가우시안 분포를 따른다.

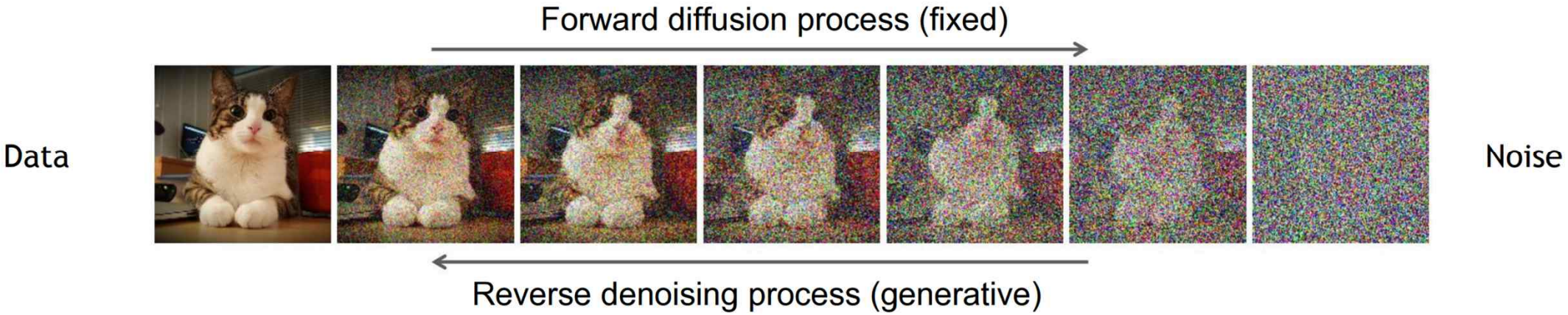
가우시안 분포를 따르는 이유는 각 분자의 다음 위치가 무작위로 결정되기 때문이다.
이 무작위성은 일반적으로 가우시안 분포를 따른다.



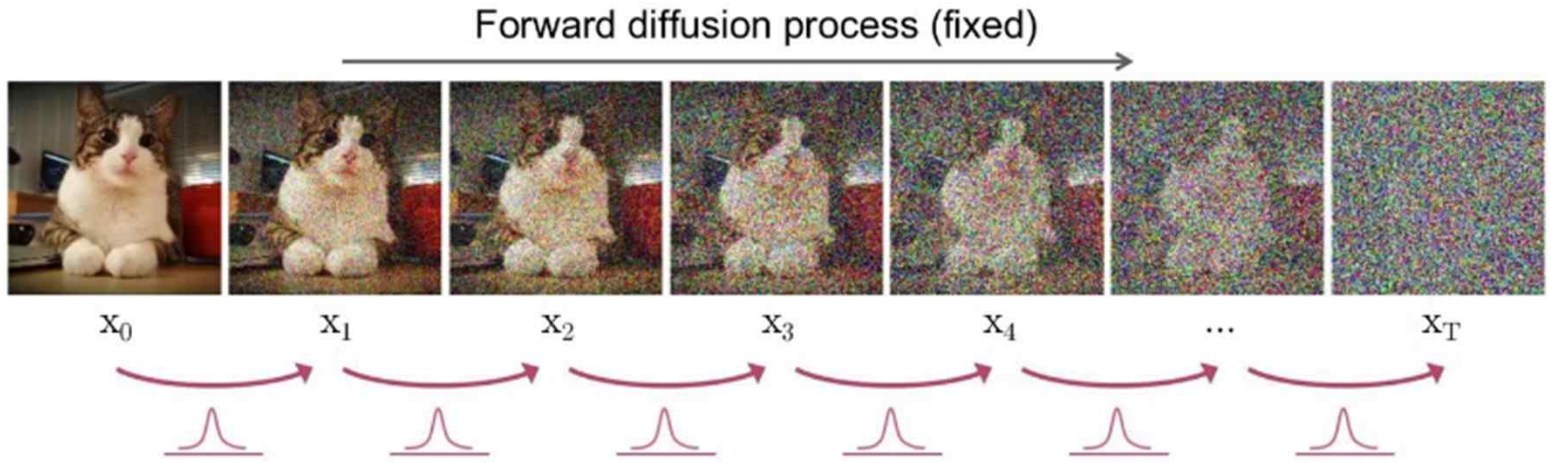
DDPM: Introduction & Physical Intention



Imaging the Model's Structure



Forward Process

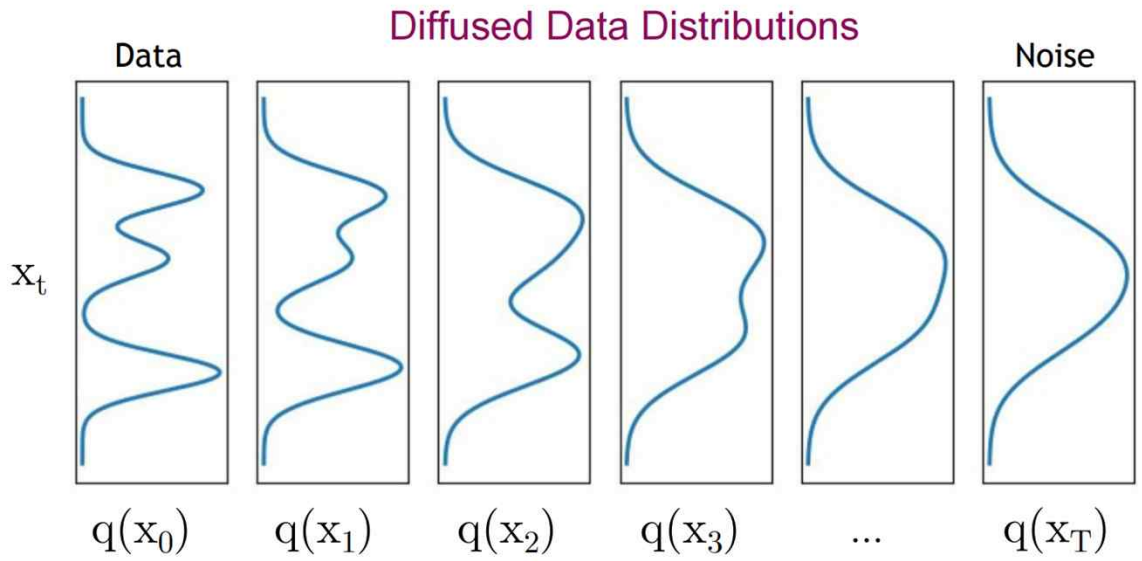


$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

노이즈 추가 과정은 단순히 이미지의 분포가 표준 정규 분포를 향하게 한다.

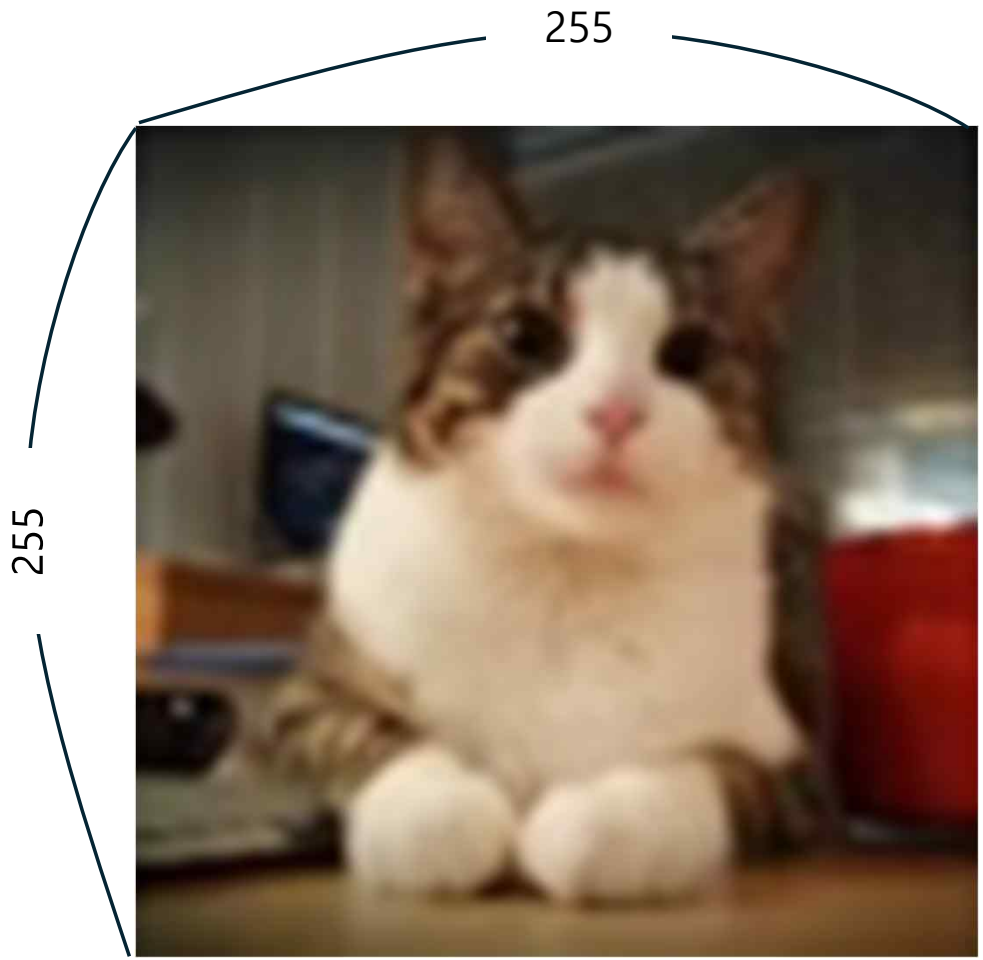
이 때 β_t는 매우 작은 양수 값이다.(예를 들어 0.00001)

Forward Process



이미지에서 말하는 '분포'란?
픽셀의 RGB값 각각의 분포이다.
픽셀은 [R, G, B]로 구성되고, 각 요소는 [0,255] 구간에서 값을 가진다.

Forward Process



Forward Process

1. 원래 상태

X_0 pixel 1:

[150, 80, 238]

2. 정규화

X_0 pixel 1:

[0.17, -0.37, 0.87]

3. 가우시안 분포 추가

X_1 pixel 1:

[$0.17+\alpha_1$, $-0.37+\alpha_2$, $0.87+\alpha_3$]

Forward Process

4. 반복

X_2 pixel 1:

$[0.17 + \alpha_1 + \beta_1,\ -0.37 + \alpha_2 + \beta_2,\ 0.87 + \alpha_3 + \beta_3]$

...

Forward Process

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

β_t는 매우 작은 양수이다. (ex. 0.0001)

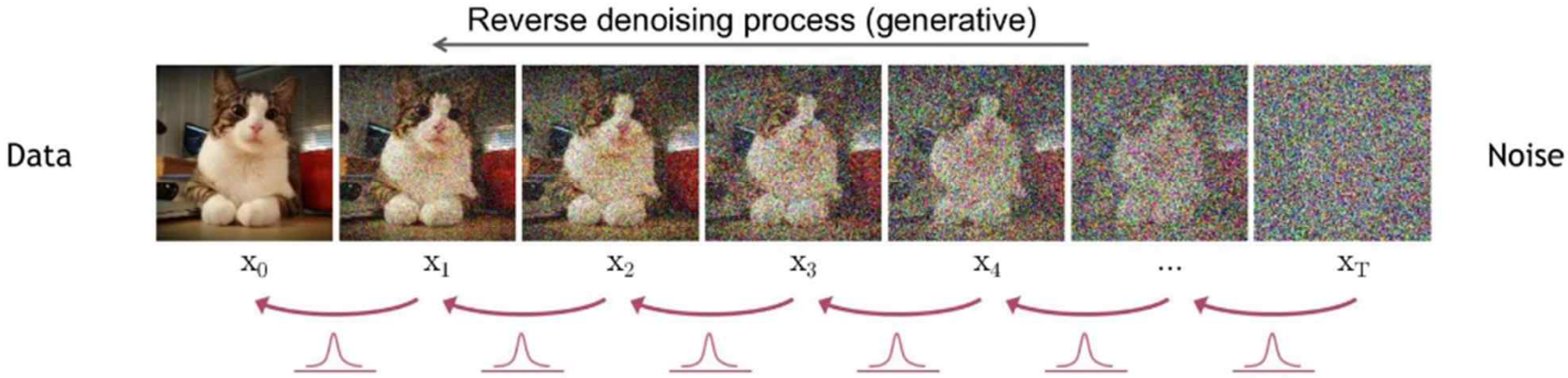
X_0의 평균부터 시작해서 X_T의 평균은 0을 향한다.

분산은 1을 유지한다.

$$N(X_{t-1};\, X_{t-1}, I) \rightarrow N\left(X_t; \sqrt{1-\beta_t}\, X_{t-1}, \beta_t I\right)$$

분산: 1 -> $1 - \beta_t + \beta_t = 1$

Reverse Process



마찬가지로 가우시안 분포로 확산의 역과정을 진행한다.

이 때 가우시안 분포의 평균과 분산은 어떻게 설정해야하는가? -> 이것이 Neural net을 통해 조정

특정 Time step에서 Forward process에서 추가한 노이즈와 Reverse process에서 제거한 노이즈의 차이를 로스함수가 비교하여 최소화한다.

Reverse Process

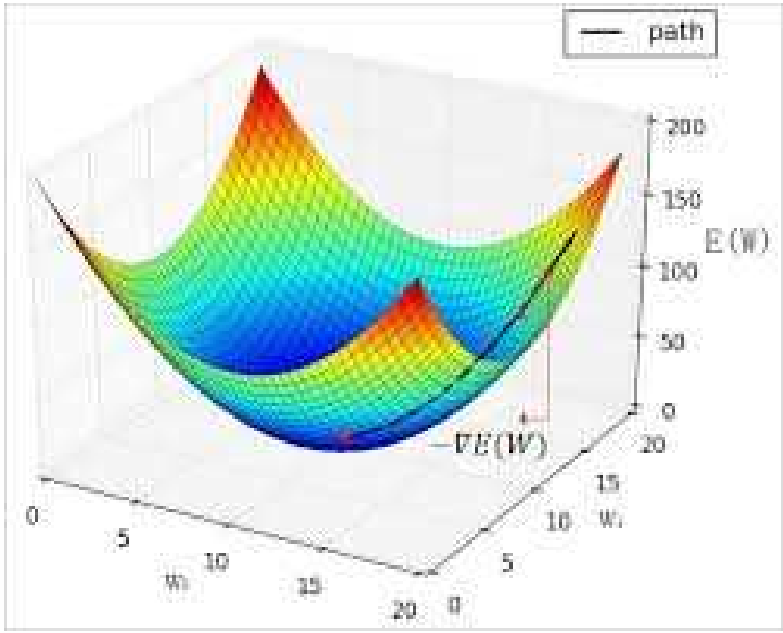
$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \qquad (1)$$

Reverse process, 즉 노이즈를 걷어내는 p_θ에서 θ가 붙어있음을 통해 수식에서
이 과정이 뉴럴 넷임을 친절히 알려준다.

또한 $\mu, \sum$에 θ가 붙어있음을 통해 가우시안 분포의 평균과 분산을 학습함을 알 수 있다.

즉 각 Time step의 Reverse process에서 어떤 가우시안 분포를 사용해야 하는지를 학습한다.

Loss Function



Loss 함수는 최적의 파라미터, DDPM에서는 각 Time step에서의 가우시안 분포의 평균과 분산을 찾도록 하는 함수이다.

Loss 함수를 최소화 하는 것이 Neural Net 훈련의 목표이다.

Loss 함수를 논리적으로 구현하여 사용하는 것이 Neural net의 핵심이다.

Loss Function

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

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

첫 번째 term(L_t)는 가우시안 분포끼리의 분포를 비교하는 것이라 의미가 없어 사용되지 않는다.

두 번째 term(L_{t-1})과 세 번째 term(L_0)은 이후 설명

Loss Function: L_{t-1}

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

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

(8) 수식에서 Time step이 t-1일 때 Loss 함수, 즉 neural net의 목표는 Time step이 t일 때의 평균(μ)이다.

이는 X_t와 X_{t-1}의 관계, 즉 이미지에 집중한다.

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

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

Loss Function: L_{t-1}

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

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

X_0에서 X_t를 만들고, 다시 X_0와 합쳐서 X_{t-1}를 만들고…

위와 같은 식 정리 과정을 거친다.

Loss Function: L_{t-1}

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

where $\boldsymbol{\epsilon}_\theta$ is a function approximator intended to predict $\boldsymbol{\epsilon}$ from $\mathbf{x}_t$. To sample $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is to compute $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The complete sampling procedure, Algorithm 2, resembles Langevin dynamics with $\boldsymbol{\epsilon}_\theta$ as a learned gradient of the data density. Furthermore, with the parameterization (11), Eq. (10) simplifies to:

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

식 정리를 마친 수식 (12)에서의 neural net의 목표는 ϵ, 즉 노이즈이다.

글자는 달라도 결국, t에서 t-1로 갈 때의 노이즈를 걷어내는 과정이라는 점은 같다.

Loss Function: L_0

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

3.3 Data scaling, reverse process decoder, and L_0

We assume that image data consists of integers in $\{0, 1, \ldots, 255\}$ scaled linearly to $[-1, 1]$. This ensures that the neural network reverse process operates on consistently scaled inputs starting from the standard normal prior $p(\mathbf{x}_T)$. To obtain discrete log likelihoods, we set the last term of the reverse process to an independent discrete decoder derived from the Gaussian $\mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_\theta(\mathbf{x}_1, 1), \sigma_1^2 \mathbf{I})$:

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

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

이미지는 0에서 255 사이의 정수 픽셀값을 가져야 하지만 노이즈는 float이므로

마지막 Loss 함수는 확률적으로 위 수식과 같이 설정해야 한다.

Summary

Forward Process: 각 타임 스텝에 대한 가우시안 노이즈 추가 및 추가한 노이즈 기억

Reverse Process: 각 타임 스텝에서 가우시안 노이즈 제거 및 Neural Net 학습

Loss Function: 노이즈에 대한 피드백으로 Forward proces와
유사한 Reverse process를 수행하도록 Neural Net을 학습하게 함

Summary

Q. 노이즈를 제거하는 reverse proces가 DDPM의 핵심으로 보이고, 노이즈를 추가하는 Forward Process는 필요한 과정이 아닌 것 같은데 단순히 표준 정규 분포를 따르는 노이즈 이미지에서 reverse proces만 진행해서 Neural net을 학습하면 되지 않나요?

A. Forward process를 통해 생성된 노이즈와 원본 이미지 간의 관계는 로스 함수의 기반이 됩니다.
노이즈 이미지가 Forward process를 통해 생성되기 때문에, reverse process 중 모델이 예측한 노이즈와 실제 노이즈 간의 차이를 계산하는 것이 가능합니다.

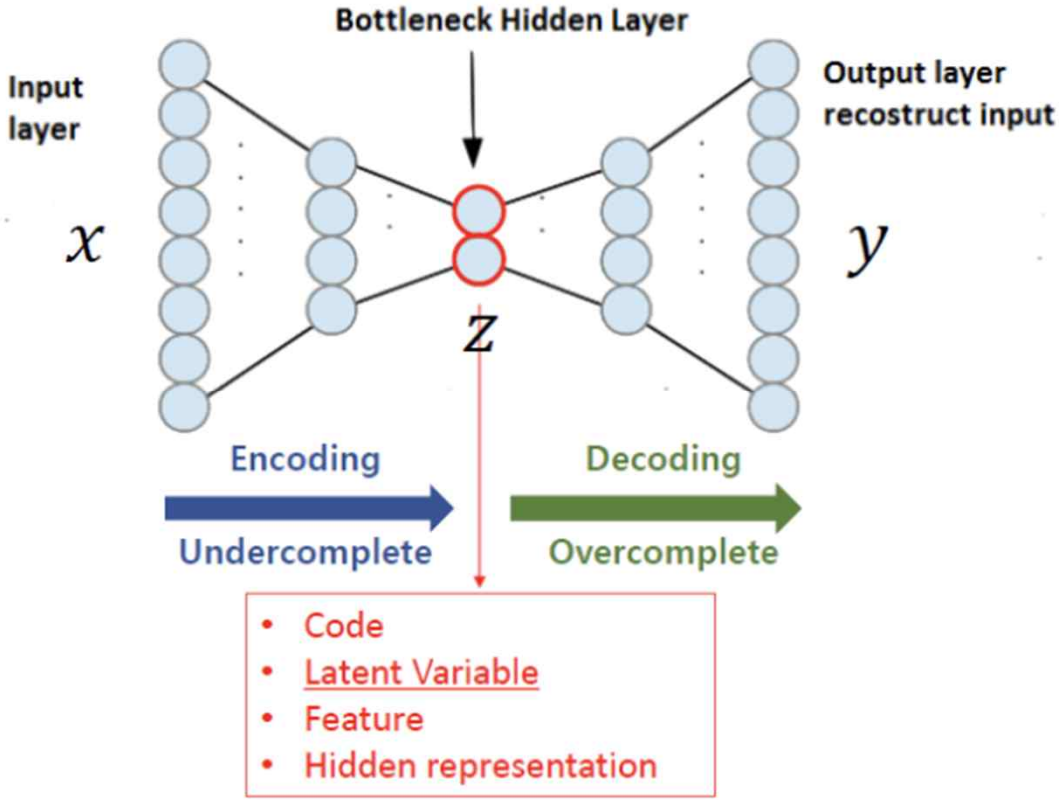
Chapter.4

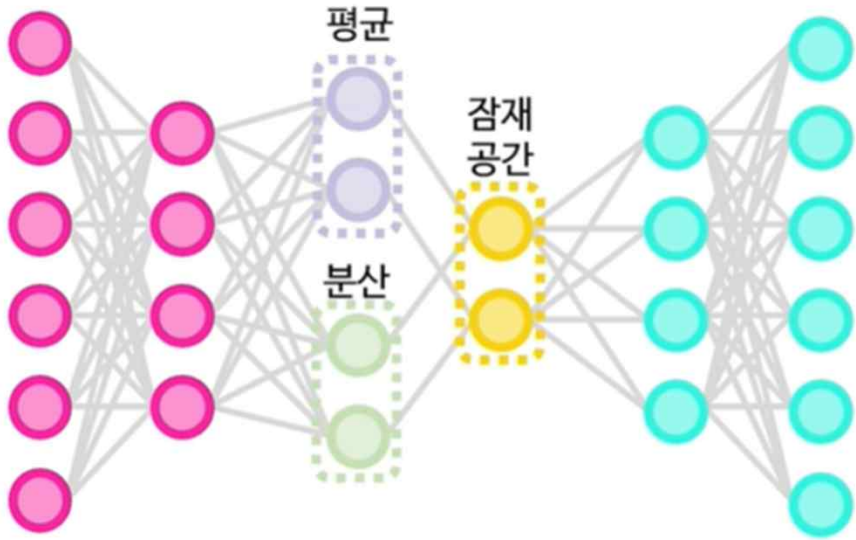
Introduction of Improved Models

Improved models from DDPM

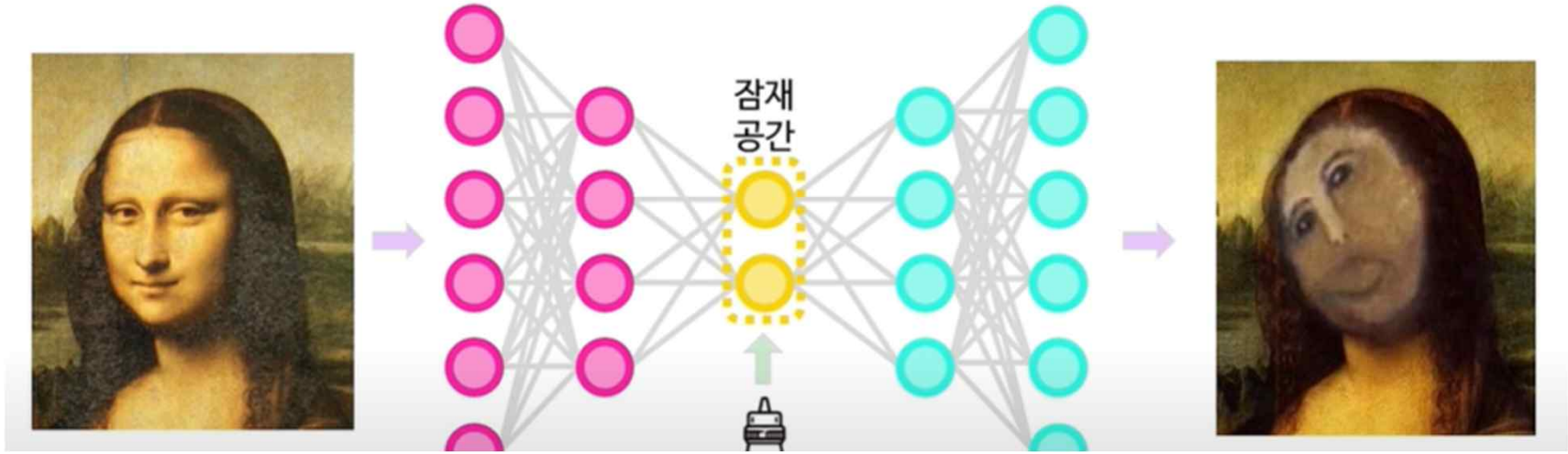
특징	DDPM	DDIM	Score-based Diffusion Models
모델 유형	확률적 모델 (Probabilistic)	결정론적 모델 (Deterministic)	점수 기반 모델 (Score-based Model)
샘플링 과정	Gaussian 노이즈를 단계적으로 제거하여 샘플링	간소화된 샘플링 과정으로, 보다 빠른 이미지 생성이 가능	점수 함수에 기반한 노이즈 제거
샘플링 속도	비교적 느림	빠름	데이터의 점수 함수(gradient)를 직접적으로 학습
노이즈 추가 방식	확률적으로 추가	고정된 방식으로 노이즈를 제거	점수 추정기를 훈련하는데, MLE(최대 가능성 추정) 사용
디퓨전 과정	확률적 정의 기반	결정론적으로 사전 정의된 경로를 따름	일반적으로 DDPM보다 더 빠름
복잡성	모델 복잡도 증가 시 성능 향상 가능, 트레이딩 오프 존재	모델 복잡도를 줄이고 효율성을 늘리는 경향	다양한 응용 사례에서 효율성과 품질 강점 제공
응용 가능성	고해상도 이미지 생성 및 다양한 작업에 광범위하게 사용	빠르게 이미지 생성이 필요한 응용에 유리	점수 기반 접근이므로 더 유연하게 일관된 샘플 생성 가능

Auto Encoder, VAE

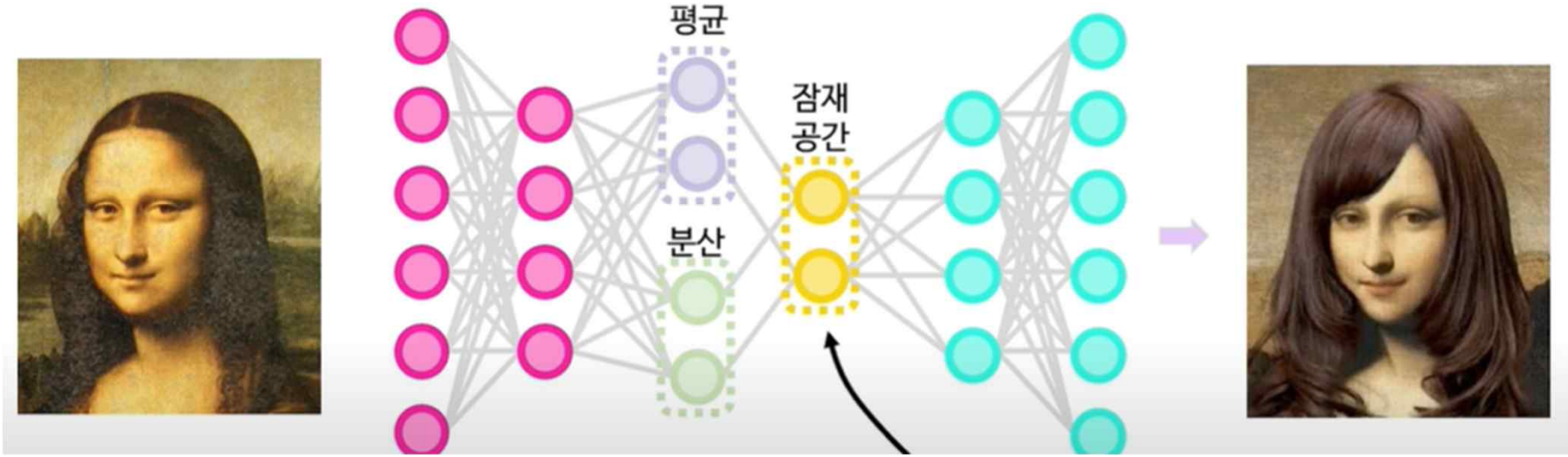




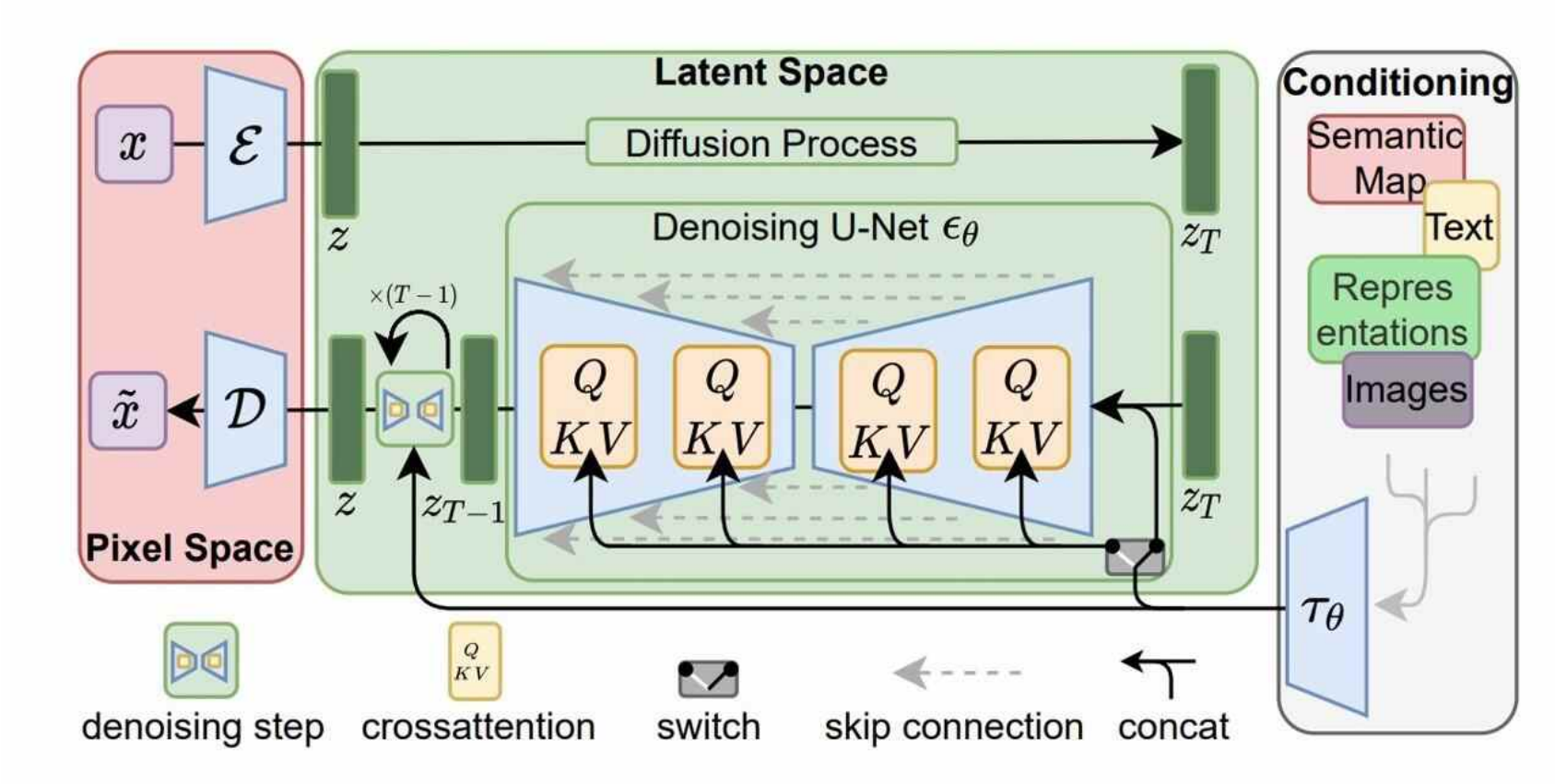
Auto Encoder



VAE



Now: Stable Diffusion, Midjourney



Q&A

Impact of Image generative models and examples