Diffusion Probabilistic Models

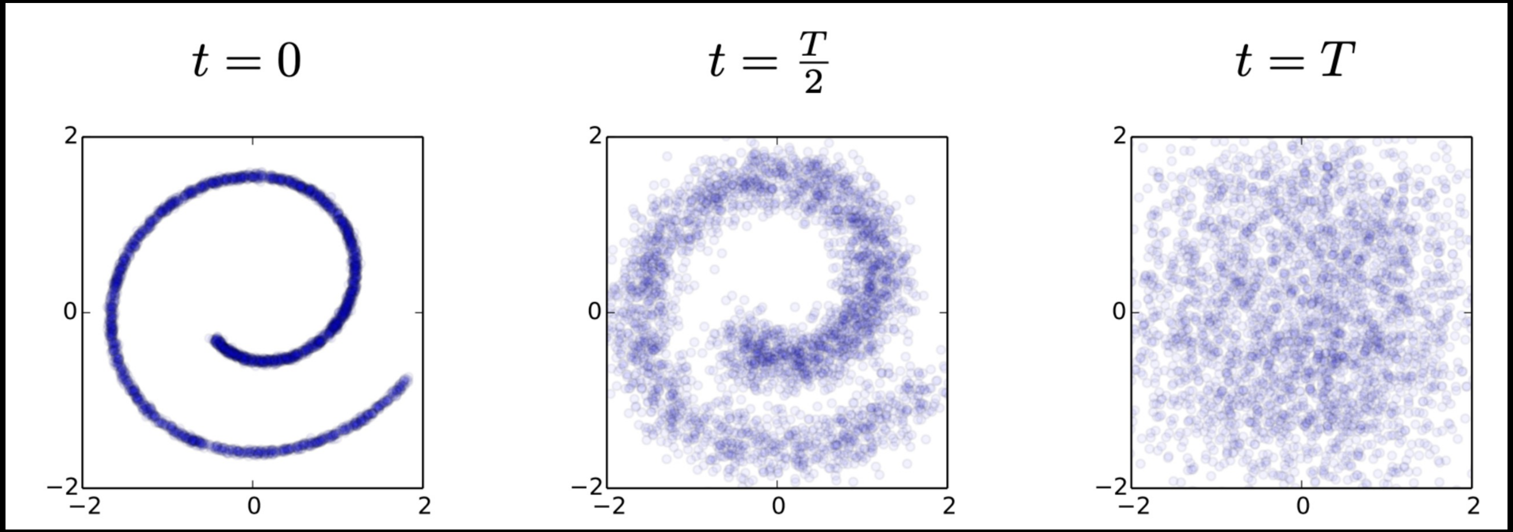
논문 리뷰

Diffusion Models

Deep Unsupervised Learning using Nonequilibrium Thermodynamics (2015)

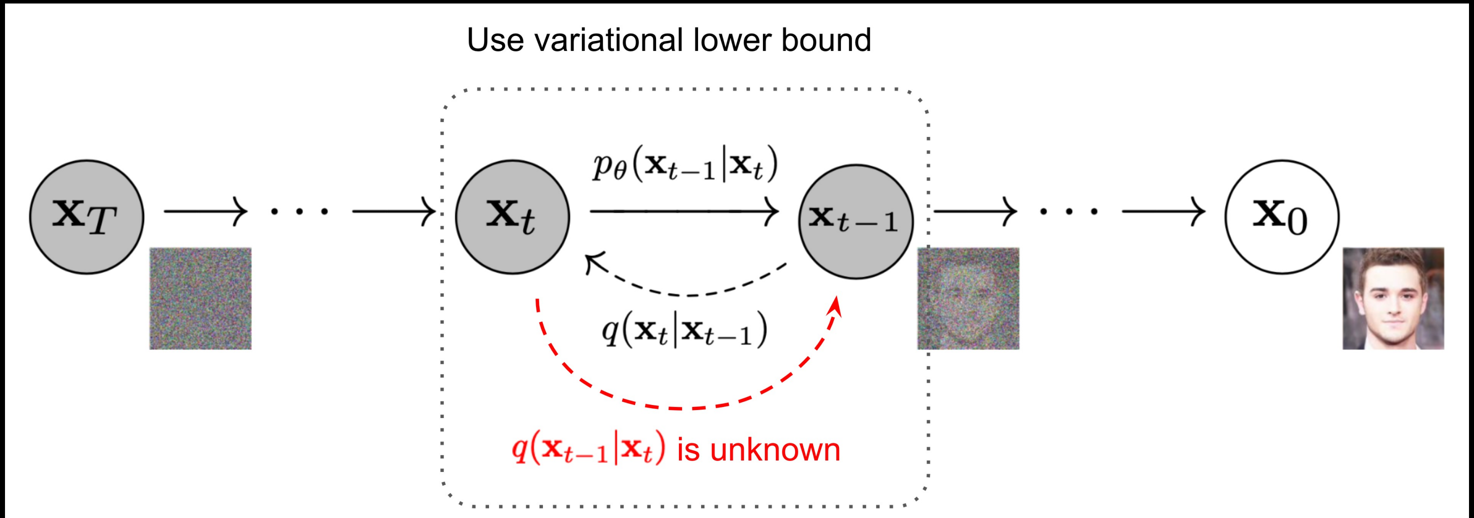
- 이 논문에서 물리에서 확산의 원리를 unsupervised learning에 사용하자는 방법론이 제시

Diffusion Models



- 시간이 지날수록 데이터의 형태가 퍼져 규칙을 잃어버리게 만드는 과정
- 최종적으로 Gaussian Noise 형태가 되게 된다.

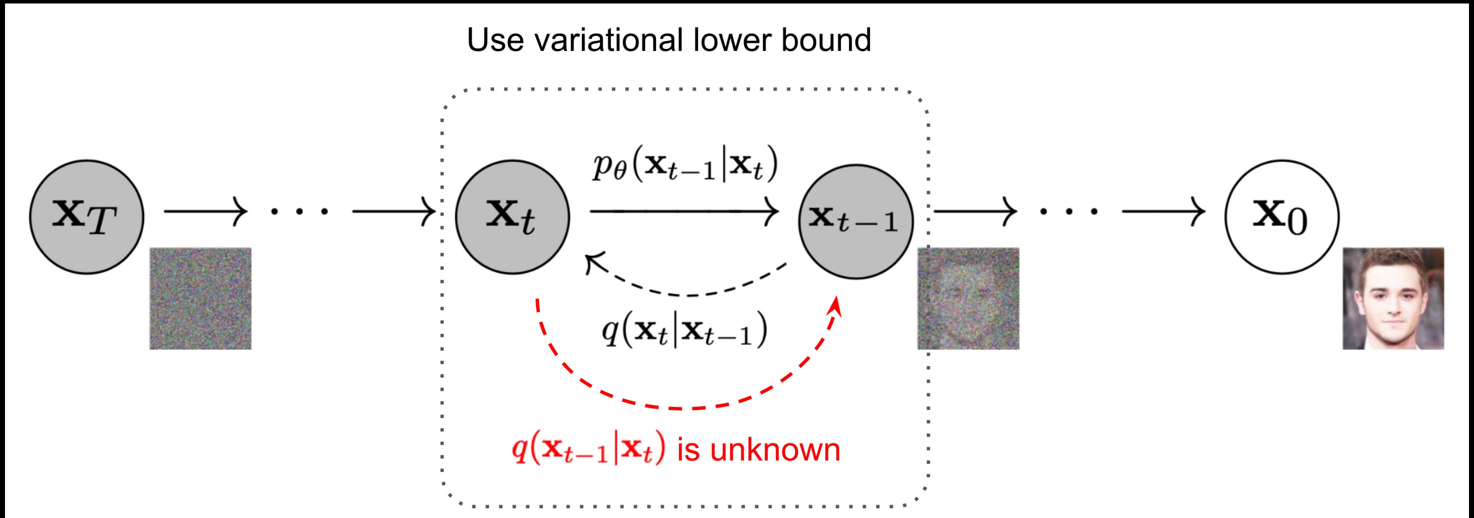
Diffusion Models



- Forward Process (Noising)
  - X_0에서 X_T로 가는 과정은 데이터에 노이즈를 추가 하는 과정
- Reverse Process (Denoising)
  - X_T에서 X_0로 가는 과정은 데이터를 생성하는 과정

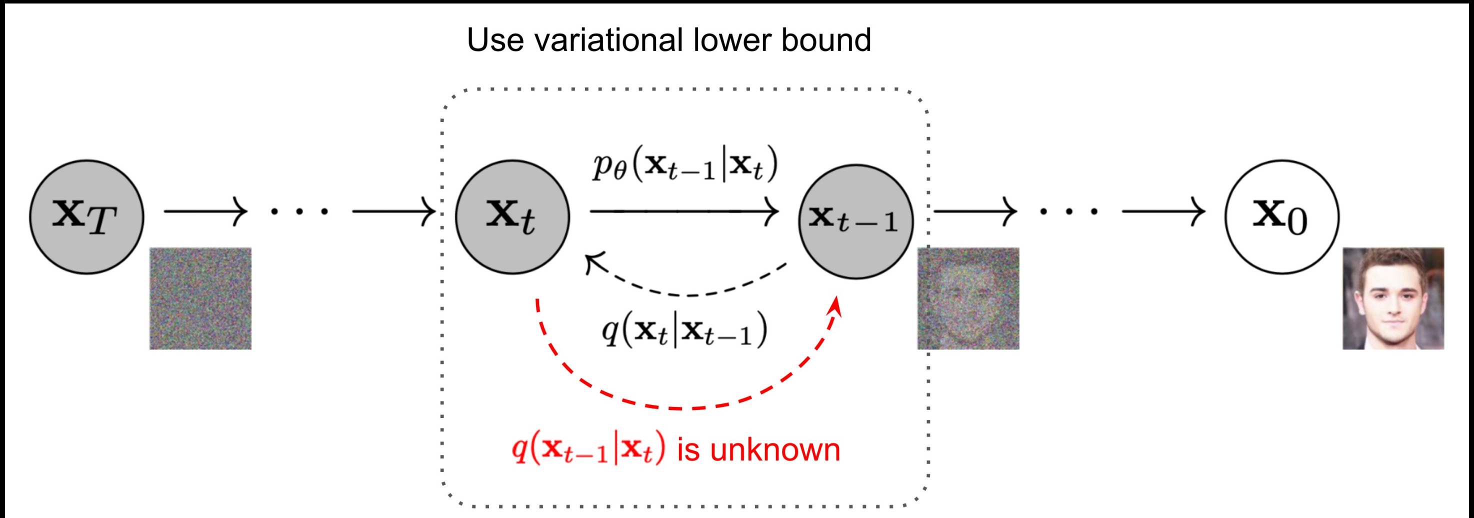
데이터의 **패턴을 없애고(Forward)** **복원하는 과정(Reverse)**을 학습

Diffusion Models



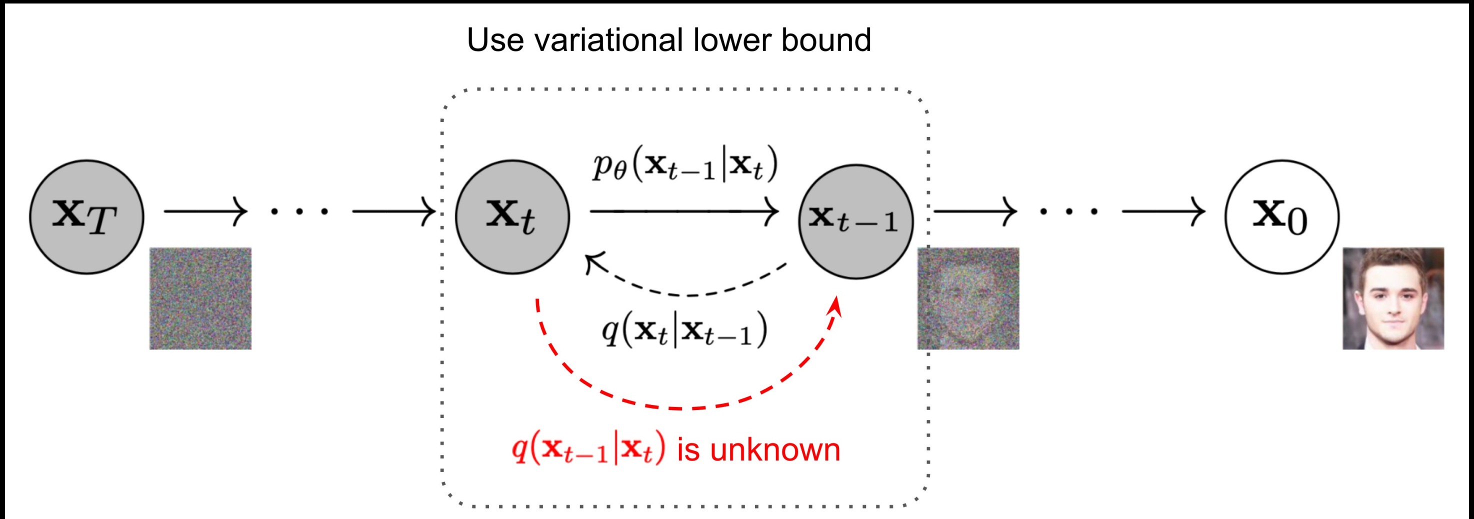
- Forward 과정은 학습이 필요하지 X
  - $q(x_t|x_{t-1})$는 원본 데이터에 노이즈를 추가하는 과정
- Reverse 과정에서의 확률 값은 알 수 없다.
  - $q(x_t|x_{t-1})$을 통해 $q(x_{t-1}|x_t)$을 알아낼 수 있는 방법이 없다.
  - 따라서 이 값을 학습으로 찾아내는 것이 목표

Diffusion Models



- 하지만 **Forward**에서 $q(x_t|x_{t-1})$가 **Gaussian 분포**를 따르게 된다면 **Reverse**에서 $q(x_{t-1}|x_t)$도 **Gaussian 분포**를 따른다는 연구 결과가 존재 (물리학에서 증명)
  (단, β_t가 아주 작을 때만 성립이 가능)
- $p_\theta(x_{t-1}|x_t) \approx q(x_{t-1}|x_t)$ 이 될 수 있게 학습을 진행

Diffusion Models



- X_T에서 X_0로 한번에 오게 하는 과정은 굉장히 어렵고 낮은 성능을 보인다.
- Diffusion 논문에서는 이 과정을 **1000번**으로 나누어 실행하게 된다.
- 이 과정들은 **Markov Chain**으로 구성되게 된다.

  (이전 단계를 조건으로 이후 단계를 얻어내는 과정)

Diffusion Models

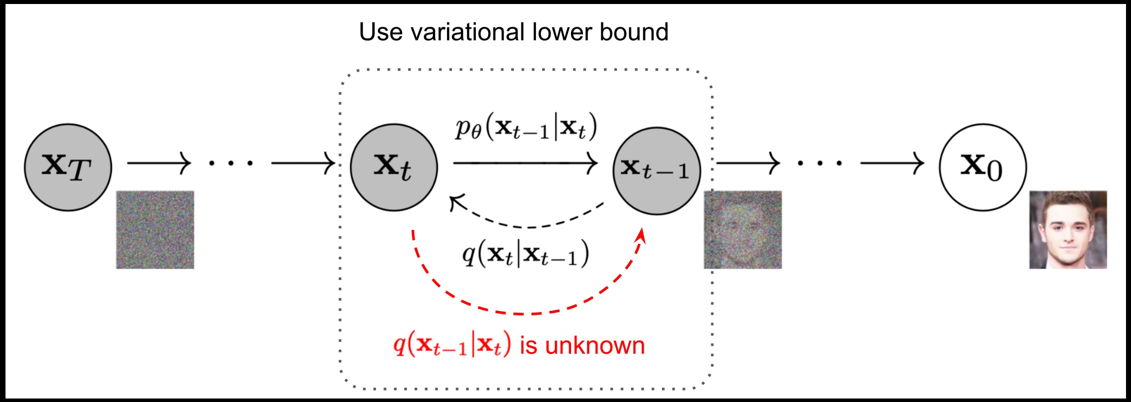
- Markov 성질을 갖는 이산 확률과정
  - ✓ Markov 성질 : "특정 상태의 확률(t+1)은 오직 현재(t)의 상태에 의존한다"
  - ✓ 이산 확률과정 : 이산적인 시간(0초, 1초, 2초, ..) 속에서의 확률적 현상

$$P[s_{t+1}|s_t] = \mathrm{P}[s_{t+1}|s_1,..,s_t]$$

$$Ex) \quad P(s_{10}) \quad P(s_9) \quad P(s_8) \quad P(s_7) \quad P(s_6) \quad \ldots \quad P(s_1)$$

✓ 예 : "내일의 날씨는 오늘의 날씨만 보고 알 수 있다." (내일의 날씨는 오로지 오늘의 날씨 만을 조건부로 하는 확률적 과정)

Forward Process

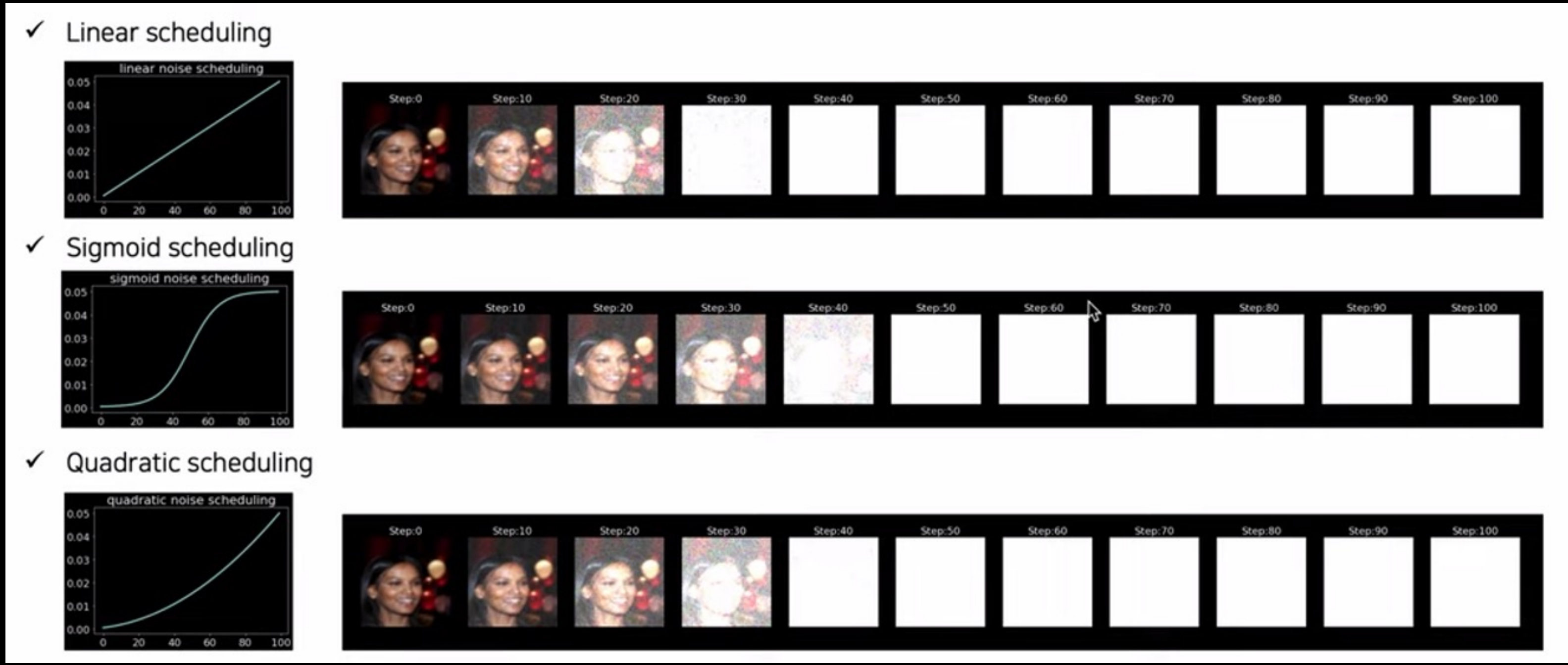


$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

- **β_t는 Gaussian Noise를 추가하는 정도**를 나타내는 변수
- 이 변수를 학습시켜서 사용할 수도 있고 사전 정의하여 Hyper parameter로 사용할 수도 있다.
- 주입되는 노이즈는 **점진적으로** 커지게 설계가 되어 있다. **$(0.0001 = \beta_1 < \beta_2 < \cdots < \beta_T = 0.02)$**

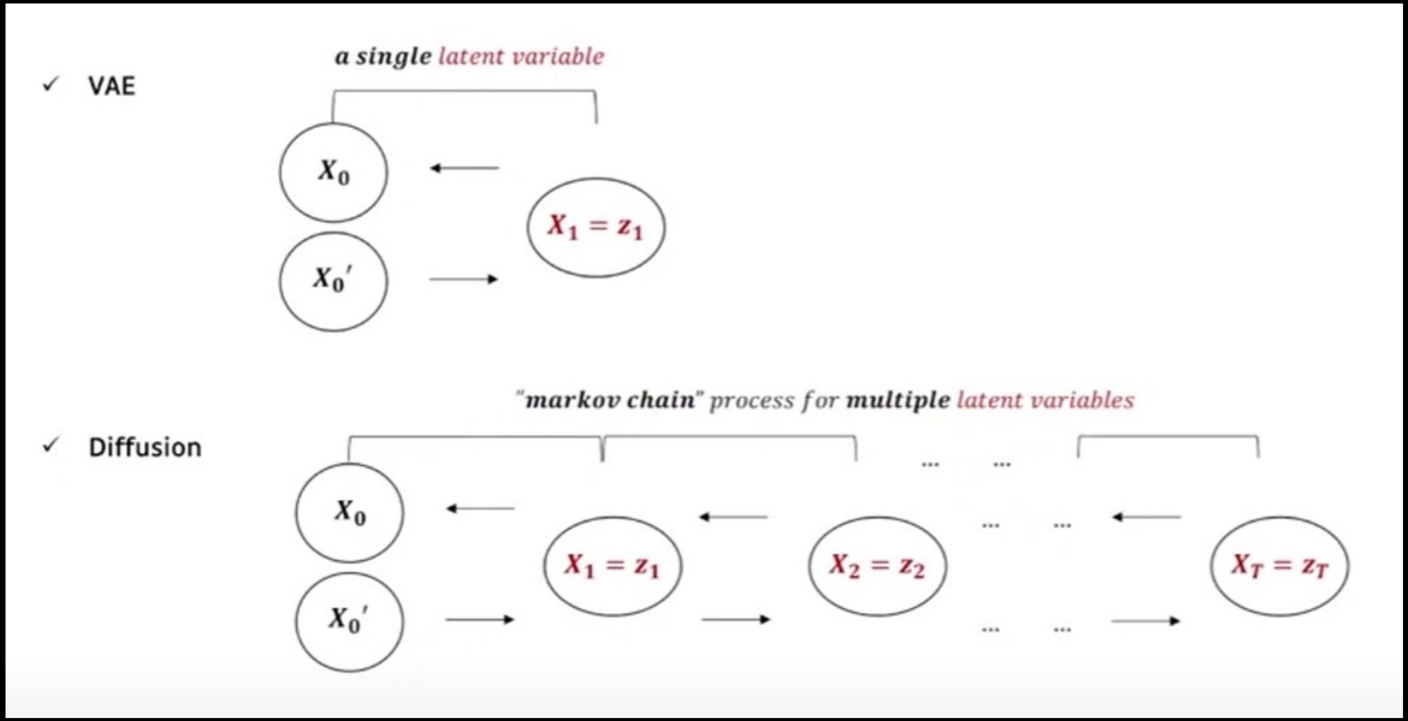
  (Linear schedule, Quad schedule, Sigmoid schedule, Cosine schedule)

Forward Process



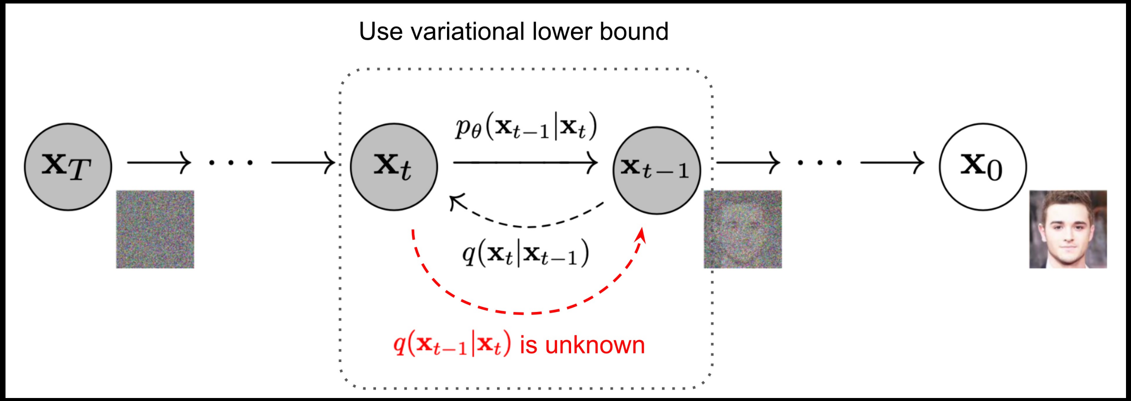
- β_t가 커지는 기울기에 따라 Gaussian 분포에 도달하는 속도가 다른 것을 확인할 수 있다.
- 초반에는 Noise를 적게 넣어야 더 많은 학습을 진행할 수 있을 것으로 판단
- 실험 결과 최종적으로 **Cosine scheduling**을 사용

Forward Process



- VAE는 Encoding과 Decoding을 통해 **하나의 latent variable**을 얻어내는 과정
- Diffusion은 사실상 **여러 개의 latent variable**을 얻어내는 과정 (Hierarchical VAE와 유사)
- 가장 마지막 latent variable은 pure isotropic gaussian을 얻을 수 있다.

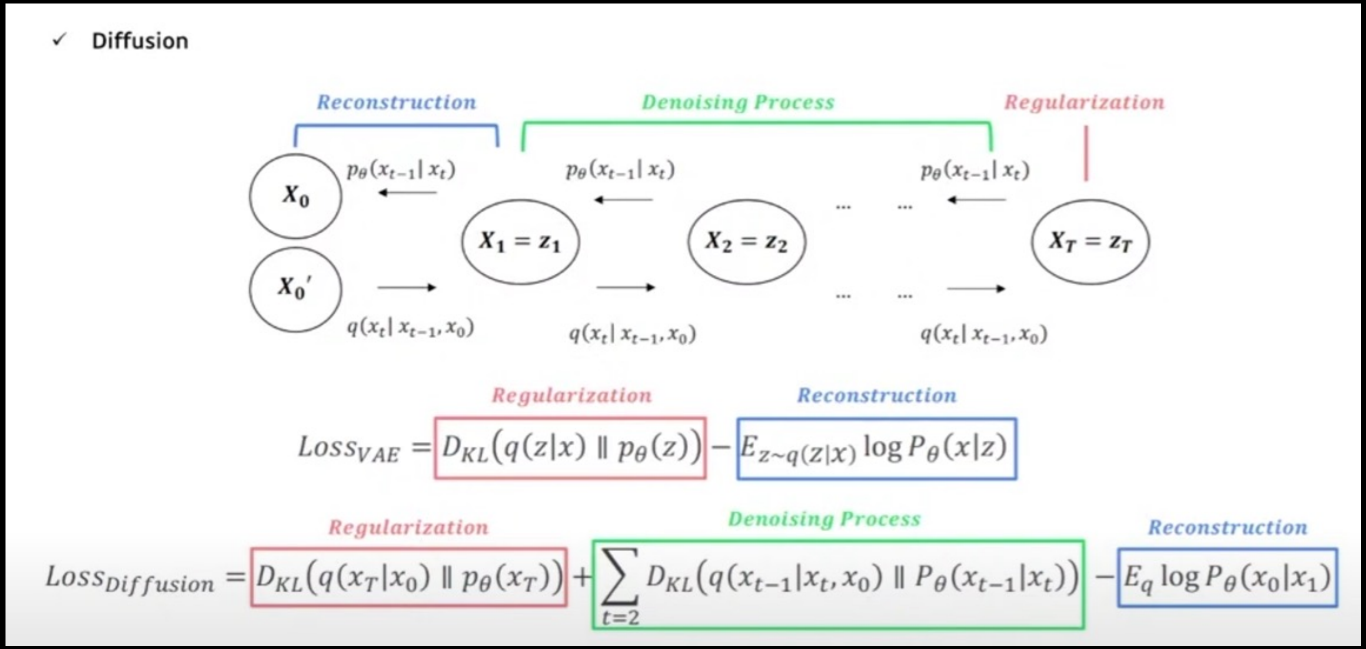
Reverse Process



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

- Reverse Process의 확률 분포는 구할 수 없기 때문에 학습을 통해서 얻어낸다.
- $\mu_\theta(x_t, t)$, $\Sigma_\theta(x_t, t)$ 는 Reverse Process에서 학습 대상 (mean과 variance를 학습)

Reverse Process



- Diffusion은 VAE와 과정이 비슷하기 때문에 Loss function의 모습도 비슷하다.
- 중간에 Denoising Process만 다른 것을 확인할 수 있다.

DDPM

1. Residual Estimation

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

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

- $\boldsymbol{\mu}_{\boldsymbol{\theta}}$가 실제로는 $\boldsymbol{x}_{\boldsymbol{t}}$와 굉장히 유사한데 원래 Diffusion model에서는 한번에 $\boldsymbol{\mu}_{\boldsymbol{\theta}}$를 예측
- DDPM에서는 $\boldsymbol{\mu}_{\boldsymbol{\theta}}$를 한번에 예측하는 것이 아닌 $\boldsymbol{\mu}_{\boldsymbol{\theta}}$과 $\boldsymbol{x}_{\boldsymbol{t}}$의 Residual인 $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$만 학습하게 했다.

DDPM

2. Loss Simplification

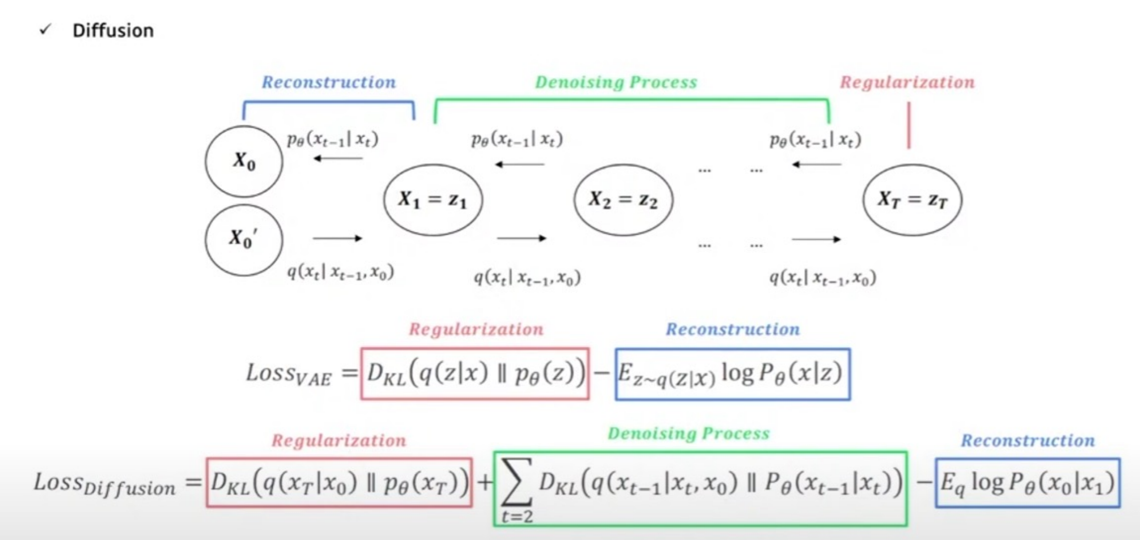
$$\mathbb{E}_q\left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T))}_{L_T}+\sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\right]$$

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

- Loss function을 간단하게 만들어 MSE 형태의 식으로 만들어 이를 활용
- 이 loss의 모습이 Denoising 알고리즘과 모습이 비슷해 Denoising을 Diffusion model 앞에 붙임

DDPM

2. Loss Simplification



- Regularization 부분을 없애준다.
- Regularization 파트는 β_t를 학습하기 위해 필요한 부분
- β_t를 Cosine scheduling 해주었기 때문에 상수가 되어 학습이 필요하지 않게 되었다.

DDPM

2. Loss Simplification

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

Now we discuss our choices in $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$ for $1 < t \leq T$. First, we set $\underline{\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I}}$ to untrained time dependent constants. Experimentally, both $\sigma_t^2 = \beta_t$ and $\underline{\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}$ had similar results. The first choice is optimal for $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the

- 분산을 학습 대상에서 제외시킨다.
- 알고 있는 β_t를 활용해 분산을 구해서 사용한다.

DDPM

2. Loss Simplification

$$\sum_{t=2} D_{kL}(q(x_{t-1} \mid x_t, x_0) \| P_\theta(x_{t-1} \mid x_t))$$

$$q(x_{t-1} \mid x_t, x_0) = N(x_{t-1} ; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \cdot I)$$

$$p_\theta(x_{t-1} \mid x_t) = N(x_{t-1} ; \mu_\theta(x_t, t), \tilde{\beta}_t \cdot I)$$

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

Denoising Process

- KL Divergence를 두 mean간의 차이로 바꾸어 사용할 수 있다.

DDPM

2. Loss Simplification

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

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

- 최종적으로 loss function을 간단하게 표현할 있게 된다.

DDPM

DDPM이 제시하는 점

- 기존 Diffusion model에서 Residual만 계산
- Loss function을 단순화 시킴

=> β_t를 학습X, variance 학습x

=> Diffusion model에게 학습시키게 만드는 부분들을 사람의 **inductive bias**를 주입하여 학습량을 줄여 성능을 높이게 되었다.

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

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

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

DDPM

- CIFAR10에서 이미지 생성을 잘 하는 것을 확인할 수 있다.

DDPM