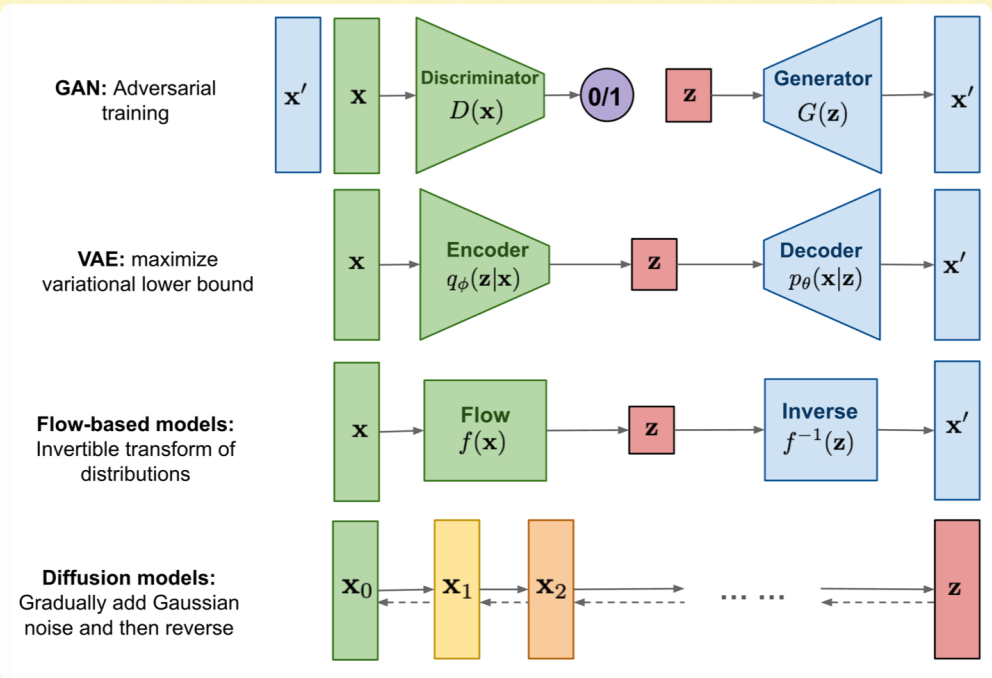
GAN, VAE, Flow-based generative model은 각각 단점이 있다.

GAN은 불안정하고, VAE는 surrogate loss의 의존하고,

Flow-based는 reversible 해야하기 때문에 architecture의 제약이 있다.

Diffusion model은 markov chain을 이용하여 noise를 조금씩 더하고, reverse process를 복원 학습한다



- forward forward process

small Gaussian noise를 더하는 것을 T steps를 수행할 때, $\mathbf{x}_t$의 noise sample step size를 Variance $\beta_t \in [0,1]$ 의 따라 달라진다.

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

forward가 진행될수록 input feature가 지닌 distinguish한 특징은 사라지고

isotropic gaussian distribution으로 변한다.

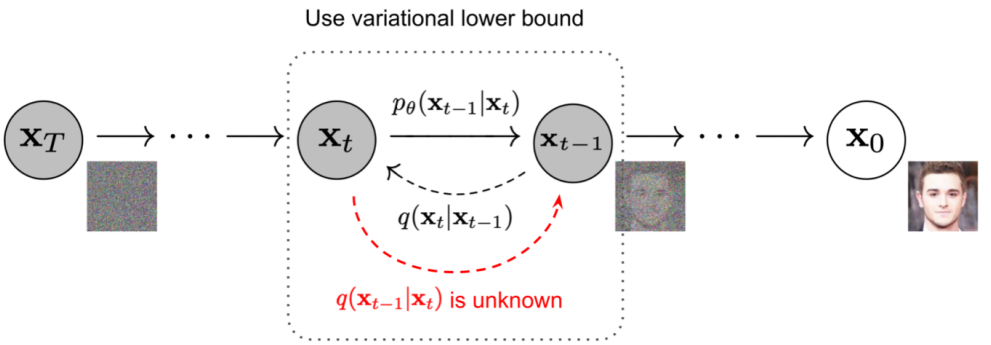


Fig. 2. The Markov chain of forward (reverse) diffusion process of generating a sample by slowly adding (removing) noise. (Image source: Ho et al. 2020 with a few additional annotations)

⇒ reparameterization trick을 이용하여, 임의의 단계에서의 sample x_t를 샘플링할 수 있다. $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$ 일때,

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, z_{t-1}$$

$$= \sqrt{\alpha_t \alpha_{t-1}}\, x_{t-2} + \sqrt{1-\alpha_t \alpha_{t-1}}\, z_{t-2}$$

$$\vdots$$

$$= \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, z$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t)I)$$

대개, noise가 점점 더 많은 양을 적용할 수 있는 noise 강도가 커지기 때문에

$\beta_1 < \beta_2 < \cdots < \beta_T$, $\bar{\alpha}_1 > \bar{\alpha}_2 > \cdots > \bar{\alpha}_T$

Markov chain update 에서 $\nabla_x \log p(x)$ 만을 이용하여 확률 밀도 $p(x)$ 에서 sampling 을 할 수 있는 방식이

SGLD (stocastic Gradient Langevin Dynamics) 라.

https://towardsdatascience.com/langevin-dynamics-29bbb9407b47

$$x_t = x_{t-1} + \frac{\epsilon}{2} \nabla_x p(x_{t-1}) + \sqrt{\epsilon}\, z_t, \quad z_t \sim \mathcal{N}(0,1), \quad \epsilon = \text{step size}$$

↳ local minimize collapse 방지를 위해 Gaussian noise를

update parameter의 없다.

- Reverse Diffusion process

$q(x_{t-1}|x_t)$를 이용하여 Gaussian Noise $\mathcal{N}(0, I)$로부터 true sample을 얻을 수 없다.

하지만 $q(x_{t-1}|x_t)$를 추정하는 것은 target space에 대한 모든 sample의 확률적인 관계로 어렵다. 따라서 $p_\theta(x_{t-1}|x_t)$를 이용하여 근사한다.

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

x_0의 대해서 reverse condition probability는 tractable하다.

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

$$\begin{aligned} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\ &\propto \exp\Big(-\frac{1}{2}\Big(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\Big)\Big) \\ &= \exp\Big(-\frac{1}{2}\Big((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\Big)\Big) \end{aligned}$$

$$\begin{aligned} \tilde{\beta}_t &= 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t \\ \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\mathbf{x}_0)/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 \end{aligned}$$

$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\,z_t)$

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t) \\ &= \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\Big) \end{aligned}$$

- Loss term

VLB (Variational lower bound)

$$
\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\le -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \| p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}\right] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)\right] \\
&= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\
\text{Let } L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \ge -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)
\end{aligned}
$$

CE (Cross-entropy)

$$
\begin{aligned}
L_{\mathrm{CE}} &= -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \right) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} \right) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right) \\
&\le -\mathbb{E}_{q(\mathbf{x}_{0:T})} \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] = L_{\mathrm{VLB}}
\end{aligned}
$$

Analytically computable

https://arxiv.org/pdf/1503.03585.pdf

$$
\begin{aligned}
L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\
&= \mathbb{E}_q\left[\log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\right) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\right] \\
&= \mathbb{E}_q\left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] \\
&= \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \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}]
\end{aligned}
$$

$$
\begin{aligned}
L_{\text{VLB}} &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where } L_T &= D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T)) \\
L_t &= D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{ for } 1 \le t \le T-1 \\
L_0 &= -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)
\end{aligned}
$$

L_t는 두 확률 분포의 비교, closed form이다.

L_T는 q는 learnable parameter가 없고, X_T는 Gaussian Noise 이기 때문에 constant다.

- Parameterization L_t for training Loss

$P_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t,t), \Sigma_\theta(x_t,t))$

↳ $\mu_\theta \Rightarrow \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_t\right)$

x_t는 훈련시, input으로 취급이 가능하기 때문에, x_t로부터 z_t를 예측하도록, reparameterize 할 수 있다.

$$
\begin{aligned}
\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right) \\
\text{Thus } \mathbf{x}_{t-1} &= \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))
\end{aligned}
$$

$\tilde{\mu}$의 차이를 minimize 하기 위한 loss term L_t

$$
\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t,t)\right)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\mathbf{z}_t - \mathbf{z}_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\mathbf{z}_t - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t, t)\|^2\right]
\end{aligned}
$$

* Simplification

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

$$
L_{\text{simple}} = L_t^{\text{simple}} + C \quad \leftarrow \text{constant}
$$

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
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

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

Fig. 4. The training and sampling algorithms in DDPM (Image source: Ho et al. 2020)

- Connection with noise-conditioned score Networks (NCSN)

score-based generative model은 Langevin dynamics에 의한 sampling과 score matching을 이용한 방식이다.

각 sample x의 pdf에 대한 score는 $\nabla_x \log p(x)$로 정의된다.

score 함수는 $s_\theta : \mathbb{R}^D \to \mathbb{R}^D$의 형태로 이루어진다.

sampling된 data가 고차원처럼 보여도 대부분 저차원에 밀집되어 있다.

따라서 밀도가 낮은 곳에서는 정확도가 떨어지기 때문에 여러 측정의 noise에 대한 score를 측정한다.

- Parameterization of β_t

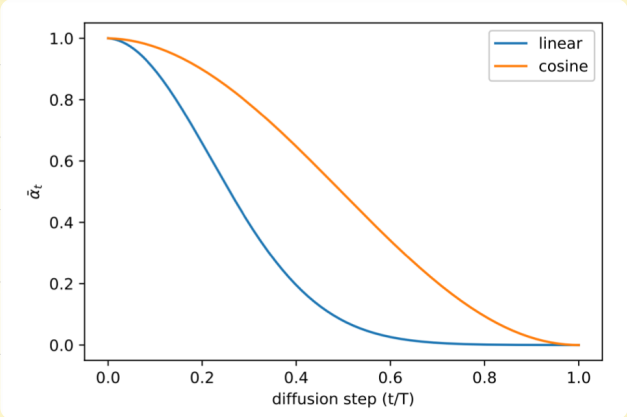
$\beta_1 = 10^{-4}$, $\beta_T = 0.02$로 forward 시에는 설정 (DDPM)

↳ But, 성능이 조금 떨림

cosine-based variance schedule (improved DDPM)

$$\beta_t = \text{clip}(1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, 0.999) \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \quad \text{where } f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)$$

⇒ DDPM보다 선형적이고 임의적이다.



- Parameterization of Reverse process variance Σ_θ

DDPM은 $\Sigma_\theta(x_t, t) = \sigma_t^2 I$ 이며 σ_t와 β_t는 learnable 하지 않지만

$\beta_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$ 로 설정했다

⇒ learnable한 Σ_θ가 불안정하기 때문

improved DDPM은 Σ_θ를 β_t와 $\tilde{\beta}_t$ 사이의 interpolation으로 설정함

$$\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \exp(\mathbf{v} \log \beta_t + (1 - \mathbf{v}) \log \tilde{\beta}_t)$$

↳ mixing vector

하지만 L_{simple}이 Σ_θ에 대한 param이 없기 때문에

$L_{hybrid} = L_{simple} + \lambda L_{VLB}$

↳ Σ_θ 학습할땐, μ_θ에 대한 학습을 멈출

↳ gradient의 noise 때문에 optimal 어려워, time average smoothing

- Speed up diffusion sampling

Markov chain을 이용한 sampling으로 수천 step에 대한 sampling은 시간이 오래 걸린다.

⇒ 32x32 의 5만개 sampling 에 20시간

1. sampling 을 [T]s] step 에서 하는것. $T > S$

2.

For another approach, let's rewrite $q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ to be parameterized by a desired standard deviation σ_t according to the nice property:

$$\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\mathbf{z}_{t-1} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\mathbf{z}_t + \sigma_t\mathbf{z} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t\mathbf{z} \\
q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I})
\end{aligned}$$

Recall that in $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I})$, therefore we have:

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

Let $\sigma_t^2 = \eta \cdot \tilde{\beta}_t$ such that we can adjust $\eta \in \mathbb{R}^+$ as a hyperparameter to control the sampling stochasticity. The special case of $\eta = 0$ makes the sampling process *deterministic*. Such a model is named the *denoising diffusion implicit model* (**DDIM**; Song et al., 2020). DDIM has the same marginal noise distribution but deterministically maps noise back to the original data samples.

During generation, we only sample a subset of S diffusion steps $\{\tau_1, \dots, \tau_S\}$ and the inference process becomes:

$$q_{\sigma,\tau}(\mathbf{x}_{\tau_{i-1}}|\mathbf{x}_{\tau_t}, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{\tau_{i-1}}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_{\tau_i} - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I})$$

⇒ DDIM ($\eta=0$), DDPM ($\eta=1$) ← S가 커야함

↳ small S 에서 좋은 성능

Compared to DDPM, DDIM is able to:

1. Generate higher-quality samples using a much fewer number of steps.
2. Have "consistency" property since the generative process is deterministic, meaning that multiple samples conditioned on the same latent variable should have similar high-level features.
3. Because of the consistency, DDIM can do semantically meaningful interpolation in the latent variable.

- Conditioned Generation

ImageNet 같은 경우 class에 따라 이미지를 생성해야한다.

명시적인 class 정보를 갖는 diffusion model을 위해,

improved DDPM은 noisy image x_t에서 $f_\phi(y|x_t, t)$를 학습시키고,

target class y sampling을 위해 $\nabla_x \log f_\phi(y|x_t, t)$를 사용한다.

ADM (ablated diffusion model), ADM-G (additional classifier Guidance)

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $f_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log f_\phi(y|x_t), \Sigma)$
end for
return x_0

Algorithm 2 Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, classifier $f_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log f_\phi(y|x_t)$
  $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return x_0

Fig. 8. The algorithms use guidance from a classifier to run conditioned generation with DDPM and DDIM. (Image source: Dhariwal & Nichol, 2021])

improved DDPM의 구조는 DDPM과 비슷, depth / width 가 더 크고
더 많은 attention head 와, multi-resolution attention,
BigGAN의 residual, U-net의 residual connection rescale, AdaGN

- Summary
  - pros

    tractability와 flexibility는 원래 상충되는 성능이다.

    tractable 한 분포를 쓰면, flexible (다양한 generative) 하지 않지만,

    diffusion model은 두 성능 모두 좋다

  - cons

    sample generate 이 markov chain 에 의존하는데

    이 과정은 시간과 계산이 많이 든다.

    IDDPM도 GAN보단 많은 시간과 비용이 든다

* paper

* Introduction.

energy-based model이 최근 Generative 분야에서 큰 성과

Diffusion probabilistic model을 제시. (이하 DM)

DM은 유한한 시간 후의 data sample과 일치하는 sample을

뽑기 위해 VI를 사용하여 훈련된 markov chain model 이다.

VI는 $p(x|z) = \frac{p(z|x)p(z)}{p(x)}$ 에서 posterior가 알기 힘들 때

더 쉬운 확률 분포 $q(z)$로 근사하기 위한 방법이다.

Chain은 sampling의 반대되는 방향으로, Gaussian noise를 추가하는 것을 data가 파괴될 때까지

하는 Markov chain인 DM의 reverse를 학습한다

diffusion에서 Gaussian noise가 작다면, markov chain sampling은

conditional Gaussian으로 설정해도되고, NN으로 parameterize가 가능하다

게다가, DM의 parameterize는 multi level noising에서 denoising score match와

같다는 걸 알수 있다. ↓ 부연 내용

* Back ground.

DM은 $p_\theta(x_0) = \int p_\theta(x_{0:T})\, dx_{1:T}$ 의 형태의 latent variable model 이다.

$x_0 \sim q(x_0)$ 이며, $x_1 \sim x_T$는 같은 dimension의 latent 다.

$p_\theta(x_{0:T})$는 $p(x_T) = \mathcal{N}(x_T; 0, I)$ 부터 시작해서 Gaussian transition을 학습하는

Markov chain으로, reverse process 라고 한다.

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I) \quad \cdots (1)$$

(variance 위에 ~ 표시)

학습은 VLB(Variation lower bound)를 이용한 NLL을 optim 하여 수행된다.

$$L : E[-\log p_\theta(x_0)] \le E_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] = E_q\left[-\log p(x_T) - \sum_{t\ge 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] \quad \cdots (3)$$

variance β_t는 trainable 하게 할 수도 있고, constant로 할 수도 있다.

↳ reverse process는 $p_\theta(x_{t-1}|x_t)$로 표현할 수 있다.

↳ β_t가 작으면 $q = p$

또한 forward process 에서 step t 에서의 sample은 closed form으로 표현할 수 있다

$\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ 일때

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I) \quad \cdots (4)$$

(new, mean, var)

∴ L을 SGD 방으로써, training이 이뤄지고, L은 다시 표현하면

$$L : E_q\left[\underbrace{D_{KL}(q(x_T|x_0) \,\|\, p(x_T))}_{L_T} + \underbrace{\sum_{t\ge 1} D_{KL}(q(x_{t-1}|x_t, x_0) \,\|\, p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \underbrace{- \log p_\theta(x_0|x_1)}_{L_0}\right] \quad \cdots (5)$$

x_0 에서 tractable한 $q(x_{t-1}|x_t, x_0)$ 와 p_θ의 분포로 같도록 학습

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta} I), \quad \leftarrow \text{sampling} \quad \cdots (6)$$

↳ $\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t, \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \quad \cdots (7)$

↳ Rao-Blackwell theorem 으로 closed form 에서 계산

— Rao - Blackwell theorem

모수 측정시, 충분한 data에서 conditional expectation을 취함으로써,

estimator의 효율성을 향상시킨다.

예를들어, $\theta = (\theta_1, \cdots, \theta_q)$가 unknown vector 일 때, $\chi = (\chi_1, \cdots, \chi_n)$는

확률분포 $p(\chi, \theta)$에서의 random sample이라고 할 때,

θ에 대한 estimator $t(\chi) = (t_1(\chi), \cdots, t_q(\chi))$, $C(t) = (C_{i,j})$, $i,j = 1 \sim q$

↳ $C_{ij} = E[(t_i(\chi) - \theta_i)(t_j(\chi) - \theta_j)]$

$E(t|S) = T(\chi)$가 θ와 독립적이도록 S를 vector valued statistic이라 한다.

이때, Rao - Blackwell theorem에 따르면

$C(t) - C(T)$는 PD matrix다.

S의 포본이 충분히 크다면, $T(\chi)$가 θ에 독립적이라는 것은 충족된다.

따라서, $E(t_r - \theta_r)^2 \geq E(T_r - \theta_r)^2$이고, t_r을 T_r로 대체하면 MSE 측면에서

효율성이 향상된다.

또한 ϕ가 convex function 이라면

$E[\phi(t_r - \theta_r)] \geq E[\phi(T_r - \theta_r)]$, $r = 1 \sim q$가 성립한다.

※ estimator가 충분통계량이 아닐 경우, 충분통계량의 함수인 다른 estimator가 있고

이 estimator는 MSE의 관점에서 더 효율적이다.

* Diffusion model and denoising autoencoder

DM 구현에는 β_t와 reverse architecture란 Gaussian distribution의 parameterization이 필요하다.

이를 위해 DM과 denoising score matching을 명시적으로 연결한다.

- Forward process and L_T

β_t는 learnable하게 reparameterization 할 수 있지만, paper에서는 fixed constant로 설정.

$\therefore L_T$도 constant하다. (noise가 없는 분포로 끝난다)

- Reverse process and $L_{1:T-1}$

$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t,t), \Sigma_\theta(x_t,t))$, $1 < t \le T$ 인 이유

1. $\Sigma_\theta(x_t,t) = \sigma_t^2 I$ 를 훈련되지 않는 time dependent constant로 설정한다.

실험적으로, $\sigma_t^2 = \beta_t$ 일 때와 $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ 일 때의 결과는 유사했다.

($\sigma_t^2 = \beta_t$: $x_0 \sim \mathcal{N}(0, I)$ 일 때 optimal. $\sigma_t^2 = \tilde{\beta}_t$: 한 sample 에 대해 optimal)

이는 coordinate-wise unit variance (좌표별로 독립적으로 계산 적용됨) data에 대한 reverse process entropy에서, upper, lower bound에 해당

극단적인 선택이다.

2. $\mu_\theta(x_t,t)$는 L_t로부터, 특정 parameterization을 통해, 표현된다.

$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t,t), \sigma_t^2 I)$ 에서

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)\|^2\right] + C, \quad C\text{는 } \theta\text{와 independent한 constant} \quad \cdots (8)$$

로 나타낼 수 있다.

따라서 μ_θ를 가장 잘 parameterization을 하는 방법은 $\tilde{\mu}_t$를 예측하는 것이다.

Eq.8은 eq.4를 $x_t(x_0,\epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, $\epsilon \sim \mathcal{N}(0,I)$로 reparameterization 하고 eq.7을 적용함으로서 확장할 수 있다.

$$L_{t-1} - C = E_{x_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}_t\left(x_t(x_0,\epsilon), \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t(x_0,\epsilon) - \sqrt{1-\bar{\alpha}_t}\epsilon)\right) - \mu_\theta(x_t(x_0,\epsilon), t)\right\|^2\right] \cdots (9)$$

$$= E_{x_0,\epsilon}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(x_t(x_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) - \mu_\theta(x_t(x_0,\epsilon), t)\right\|^2\right] \cdots (10)$$

↳ μ_θ는 $\frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$를 예측해야한다.

x_t는 model의 input이 될 수 있기 때문에

$\mu_\theta(x_t, t) = \tilde{\mu}_t\left(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(x_t))\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$,

ϵ_θ는 x_t로부터의 ϵ를 예측한 function

로 parameterization을 할 수 있다.

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

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

$x_{t-1} \sim p_\theta(x_{t-1}|x_t)$ 에서의 sampling은 $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$로 계산할 수 있다.

sampling process인 Algorithm 2는 데이터의 밀도의 학습된 gradient로 ϵ_θ를 사용하는

Langevin dynamics와 유사하다.

Eq.11을 이용하여 eq.10을 simplify할 수 있다.

$$\Rightarrow E_{x_0,\epsilon}\left[\frac{\beta_t}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right] \cdots (12)$$

↳ multi noise level t에서 denoising score matching과 비슷하다.

Eq.12는 eq.11의 variational bound와 유사하다.

즉, denoising score matching과 유사한 복잡함수의 optimization은 → Algorithm 2

Langevin dynamics와 비슷한 finite-time sample chain의 marginal을 맞추기 위해

Variational Inference를 사용하는 것과 같다.

즉, μ_θ가 $\tilde{\mu}_t$를 근사하도록 reverse process를 훈련시킬 수 있고,

parameterization을 변경해서 ϵ를 근사하도록 훈련시킬 수도 있다.

ϵ-parameterization은 효율적이지만 아직 $p_\theta(x_{t-1}|x_t)$의 또다른 parameterization 이기 때문에

Section 4에서 ϵ과 $\tilde{\mu}_t$를 비교한다.

- Data scaling, reverse process decoder, and L_0

image는 0~255를 -1~1로 scaling 하여 사용한다.

↙ 각 pixel 별 이라는 뜻인듯...?

discrete log likelihood를 위해 $\mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I)$로부터

$$p_\theta(x_0|x_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2) dx$$

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

D는 data dimension

→ Sampling의 끝에 $\mu_\theta(x_1, 1)$를 display

- Simplified training objective

위에서 정의된 reverse process와 decoder로 eq 12, 13으로 구성된 variational bound를

명확히 미분할 수 있고, 훈련할 수 있다.

하지만 아래와 같이 하는게 더 간단함.

$$L_{simple}(\theta) = E_{t, x_0, \epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right], \quad t = 1 \sim T \text{ 사이}$$

$t=1$ 일때는 eq 13과 비슷하고, $2 \sim T-1$은 eq. 12과 같다. T는 fixed 이기 때문에 고려X

정확한 동작은 Algorithm 1과 같다.

Eq.14는 L에 비해 coefficient가 없기 때문에 기존 VI의 비례식은

weighted VI라 볼 수 있다.

이건 낮은 t에 대해 가중치가 낮아지는데 t가 높아질수록 어려운 task이기 때문에

어려운 task에 더 집중할 수 있게 하여, 더 좋은 성능을 얻을 수 있다.

★ Experiments.

$T = 1000$, $\beta_t = 10^{-4} \sim 0.02$ 까지 linear하게 증가

↳ forward와 reverse가 서로 동일하도록 작게 설정

reverse process에서, unmasked PixelCNN++의 UNet backbone에 Group normalize를 한다.

network는 transformer의 position encoding 같이 시간에 따라 공유된다.

↳ Appendix B

- Sample quality

code length?

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

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87 ± 0.12	25.32	
SNGAN [39]	8.22 ± 0.05	21.7	
SNGAN-DDLS [4]	9.09 ± 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	$\mathbf{9.74} \pm 0.05$	3.26	
Ours (L, fixed isotropic Σ)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	**3.17**	≤ 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
$\tilde{\mu}$ prediction (baseline)		
L, learned diagonal Σ	7.28 ± 0.10	23.69
L, fixed isotropic Σ	8.06 ± 0.09	13.22
$\|\tilde{\mu} - \tilde{\mu}_\theta\|^2$	–	–
ϵ prediction (ours)		
L, learned diagonal Σ	–	–
L, fixed isotropic Σ	7.67 ± 0.13	13.51
$\|\tilde{\epsilon} - \epsilon_\theta\|^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	**3.17**

- reverse process parameterization and training objective ablation

learned variance는 불안정하다.

ϵ 예측하면, 앞과 비슷한 성능을 내지만, simplified objective로 훈련할 때, 성능이 좋다.

- Progressive coding

과적합 여부 관련: Appendix D.

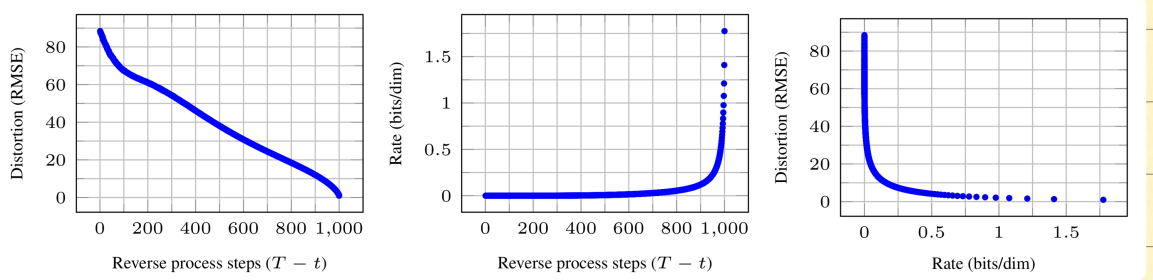
- Progressive lossy compression

Algorithm 3 Sending $\mathbf{x}_0$

1: Send $\mathbf{x}_T \sim q(\mathbf{x}_T|\mathbf{x}_0)$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 2, 1$ **do**
3: Send $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0)$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: Send $\mathbf{x}_0$ using $p_\theta(\mathbf{x}_0|\mathbf{x}_1)$

Algorithm 4 Receiving

1: Receive $\mathbf{x}_T$ using $p(\mathbf{x}_T)$
2: **for** $t = T-1, \ldots, 1, 0$ **do**
3: Receive $\mathbf{x}_t$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
4: **end for**
5: **return** $\mathbf{x}_0$



- Progressive generation

random bit로부터 progressive decomposition에 의해,

progressive unconditional generation process를 실행한다.

Fig 6.10은 Algorithm 2를 이용한 progressive generation을 나타내고,

Fig 7은 다양한 t에 대한 prediction $x_0 \sim p_\theta(x_0|x_t)$를 의미한다.

t가 크면 전체적인 feature만 유지되지만, 작으면 detail한 부분도 보존된다.

- Connection to Autoregressive decoding

Variational bound eq.5는 다음과 같이 표현할 수 있다.

$$L = D_{KL}(q(x_T)\|p(x_T)) + E_q\left[\sum_{t\geq 1} D_{KL}(q(x_{t-1}|x_t)\|p(x_{t-1}|x_t))\right] + H_0(x_0) \quad \cdots (16)$$

DM을 data 차원을 재배치하는 것으로는 표현할 수 없는 일반화된 bit 순서로

autoregressive 한다고 해석한다.

- Interpolation

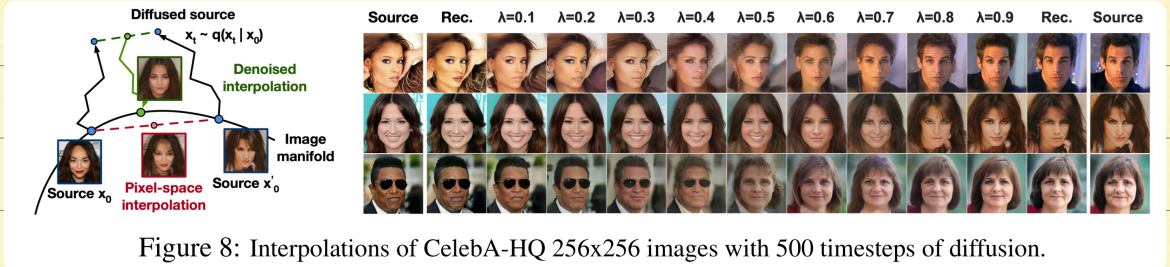


Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

⇒ 두 이미지의 중간으로 생기는 artifact를 x_T 에서 Interpolation 함으로써 구할 수 있다.

* Conclusion

확산모델은 고품질 이미지 생성을 제안했고, Markov chain 훈련, denoising score matching,

annealed Langevin dynamics, autoregressive, progressive lossy compression을 위한

VI와 DM간의 연관성을 찾았다

* Appendix A.

$$L = E_q\left[D_{KL}(q(x_T|x_0) \| p(x_T)) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)\right]$$

$$L = E_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] = E_q\left[-\log p(x_T) - \sum_{t\geq 1} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$$

$$= E_q\left[-\log p(x_T) - \sum_{t>1} \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \cdot \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} - \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$= E_q\left[-\log \frac{p(x_T)}{q(x_T|x_0)} - \sum_{t>1} \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} - \log p_\theta(x_0|x_1)\right]$$

$$= E_q\left[D_{KL}(q(x_T|x_0) \| p(x_T)) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) - \log p_\theta(x_0|x_1)\right]$$

→ progressive

$$L = E_q\left[-\log p(x_T) - \sum_{t\geq 1} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$$

$$= E_q\left[-\log p(x_T) - \sum_{t\geq 1} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \cdot \frac{q(x_{t-1})}{q(x_t)}\right]$$

$$= E_q\left[-\log \frac{p(x_T)}{q(x_T)} - \sum_{t>1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t)} - \log q(x_0)\right]$$

$$= D_{KL}(q(x_T) \| p(x_T)) + E_q\left[\sum_{t>1} D_{KL}(q(x_{t-1}|x_t) \| p_\theta(x_{t-1}|x_t))\right] + H(x_0)$$

*Appendix B

Backbone 으로는 wide resnet 을 기반으로한 UNet 인 PixelCNN++ 을 사용했다

Weight Normalization 을 group normalization 으로 교체

32x32 는 32x32 ~ 4x4 resolution, 256 은 6level resolution

resolution 마다 2level residual conv block, 16x16 에는 conv-block 마다 self-attention

각 resblock 에 time 은 transformer 의 sinusoidal PE 적용

EMA 의 decay 0.9999 로 사용

+ Denoising score matching

score matching은 분할 함수(변수에 관한 함수값들)가 다루기 힘든 비정규화 확률 밀도 모델에 적합한 maximum log-likelihood의 대안!

⇒ Fisher Divergence는 score matching, Kullback-Leibler Divergence는 ML을 도출함

Score matching이 찾는 parameter는 훈련 데이터의 small-noise perturbation에 robust하다.

Score matching은 generalized score matching과 minimum probability flow의 보다 general framwork에서 이산 분포에 대한 score matching의 generalization을 허용한다. minimum probability flow는 다른 alternative 방식들로 재구성할 수 있다.

↳ data 분포와 예측 사이의 D_{KL}을 최소화 시킴으로써

이전 연구에서는 극소 variance의 Gaussian noise에 대한 optimal denoising에 SM을 연결하고, SM으로 Gaussian binary RBM을 훈련하는 것이 추가 regularization을 사용하여, 일반 autoencoder를 훈련하는 것과 동일하고, SM과 Contrastive Divergence 간의 연결을 연구했다

논문은 regularized score matching의 형식으로 DAE의 훈련 재구성

denoising은 continuous value 값을 가지기 때문에 CE가 아닌 MSE로 훈련

- Expilict Score matching (ESM)

Score matching intractable한 partition function $Z(\theta)$를 probability density model $p(x;\theta)$의 θ 학습을 위해 제안됨

$$p(x;\theta) = \frac{1}{Z(\theta)} \exp(-E(x;\theta))$$

(E(x;θ) → energy function)

$$\text{score}: \psi(x;\theta) = \frac{\partial \log p(x;\theta)}{\partial x}$$

⇒ score는 param의 Gradient를 의미.

$$J_{ESM_q}(\theta) = E_{q(x)}\left[\frac{1}{2}\left\|\psi(x;\theta) - \frac{\partial \log q(x)}{\partial x}\right\|^2\right]$$ ← GT

↳ but q를 모르면 쓸 수 없음

- Implict score matching

$$\underbrace{E_{q(x)}\left[\frac{1}{2}\left\|\psi(x;\theta) - \frac{\partial \log q(x)}{\partial x}\right\|^2\right]}_{J_{ESM_q}(\theta)} = \underbrace{E_{q(x)}\left[\frac{1}{2}\left\|\psi(x;\theta)\right\|^2 + \sum_{i=1}^{d}\frac{\partial \psi_i(x;\theta)}{\partial x_i}\right]}_{J_{ISM_q}(\theta)} + C_1$$

$$\psi_i(x;\theta) = \psi(x;\theta)_i = \frac{\partial \log p(x;\theta)}{\partial x_i}$$

- Finite Sample Version of Implict score matching

실제로는 D_n 개의 sample만 있을때 (머신러닝의 data)

이에 대한 optimize를 해야

$$J_{ISM_{q_0}}(\theta) = E_{q_0(x)}\left[\frac{1}{2}\left\|\psi(x;\theta)\right\|^2 + \sum_{i=1}^{d}\frac{\partial \psi_i(x;\theta)}{\partial x_i}\right]$$

$$= \frac{1}{n}\sum_{t=1}^{n}\left(\frac{1}{2}\left\|\psi(x;\theta)\right\|^2 + \sum_{i=1}^{d}\frac{\partial \psi_i(x;\theta)}{\partial x_i}\right)$$

$$J_{ESM_q} = J_{ISM_q} = \lim_{n\to\infty} J_{ISM_{q_0}}$$

Criterion의 안정성 향상을 위해 regularizer 추가.

$$J_{ISM_{reg}}(\theta) = J_{ISM_{q_0}}(\theta) + \lambda\sum_{i=1}^{d}\left(\frac{\partial \psi_i(x^{(t)};\theta)}{\partial x_i}\right)^2$$

- Linking Score Matching to the Denoising Autoencoder Objective

- Matching the score of a Non-Parametric Estimator

ESM with Parzen window density estimator $q_\sigma(\tilde{x})$

$$J_{ESMq_\sigma}(\theta) = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta) - \frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$$

$\sigma > 0$ 이고, 미분가능하면, $E_{q_\sigma}\left[\left\|\frac{\partial \log q_\sigma(\tilde{x})}{\partial \tilde{x}}\right\|^2\right]$ 는 유한하다.

$\Rightarrow J_{ESMq_\sigma} \frown J_{ISMq_\sigma}$

$\lim_{\sigma \to 0}$ 이면 $\bigwedge$ 꼴, 미분 불가능하기 때문에 정의 X J_{ESM} 계산이 안됨

- Denoising score Matching

(clean, corrupted) = $(x, \tilde{x})$, $q_\sigma(\tilde{x}, x) = q_\sigma(\tilde{x}|x)q_0(x)$

$$J_{DSMq_\sigma}(\theta) = \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta) - \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$$

$\tilde{x} \to x$ 로 학습 $\quad \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} = \frac{1}{\sigma^2}(\mathbf{x} - \tilde{\mathbf{x}}).$

$J_{ESMq_\sigma} \frown J_{DSMq_\sigma}$

$$\begin{aligned}
J_{DSMq_\sigma}(\theta) &= \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta) - \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\|^2\right] \\
&= \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\frac{1}{\sigma^2}\left(\mathbf{W}^T \operatorname{sigmoid}(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b}) + \mathbf{c} - \tilde{\mathbf{x}}\right) - \frac{1}{\sigma^2}(\mathbf{x} - \tilde{\mathbf{x}})\right\|^2\right] \\
&= \frac{1}{2\sigma^4}\mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\left\|\mathbf{W}^T \operatorname{sigmoid}(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b}) + \mathbf{c} - \mathbf{x}\right\|^2\right] \\
&= \frac{1}{2\sigma^4} J_{DAE_\sigma}(\theta).
\end{aligned}$$

We have thus shown that

$$J_{DSMq_\sigma} \smile J_{DAE_\sigma} \tag{14}$$

Proof that $J_{ESMq_\sigma} \backsim J_{DSMq_\sigma}$ (11)

The explicit score matching criterion using the Parzen density estimator is defined in Eq. 7 as

$$J_{ESMq_\sigma}(\theta) = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta) - \frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$$

which we can develop as

$$J_{ESMq_\sigma}(\theta) \;=\; \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta)\right\|^2\right] - S(\theta) + C_2 \tag{16}$$

where $C_2 = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$ is a constant that does not depend on θ, and

$$\begin{aligned}
S(\theta) &= \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\rangle\right] \\
&= \int_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\frac{\partial}{\partial \tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})}{q_\sigma(\tilde{\mathbf{x}})}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial}{\partial \tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial}{\partial \tilde{\mathbf{x}}} \int_{\mathbf{x}} q_0(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) d\mathbf{x}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \int_{\mathbf{x}} q_0(\mathbf{x}) \frac{\partial q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} d\mathbf{x}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \int_{\mathbf{x}} q_0(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} d\mathbf{x}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q_0(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle d\mathbf{x} d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q_\sigma(\tilde{\mathbf{x}}, \mathbf{x}) \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle d\mathbf{x} d\tilde{\mathbf{x}} \\
&= \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}},\mathbf{x})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle\right].
\end{aligned}$$

Substituting this expression for $S(\theta)$ in Eq. 16 yields

$$\begin{aligned}
J_{ESMq_\sigma}(\theta) \;=\; & \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta)\right\|^2\right] \\
& - \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle\right] + C_2.
\end{aligned} \tag{17}$$

We also have defined in Eq. 9,

$$J_{DSMq_\sigma}(\theta) = \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta) - \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\|^2\right],$$

which we can develop as

$$\begin{aligned} J_{DSMq_\sigma}(\theta) &= \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta)\right\|^2\right] \\ &\quad -\mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle\right] + C_3 \end{aligned} \tag{18}$$

where $C_3 = \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$ is a constant that does not depend on θ.

Looking at equations 17 and 18 we see that $J_{ESMq_\sigma}(\theta) = J_{DSMq_\sigma}(\theta) + C_2 - C_3$. We have thus shown that the two optimization objectives are equivalent.

Reference paper : Nonequilibrium

2. Algorithm

목적은 forward process의 복잡한 distribution을 간단하고 tractable한 분포로 바꾸고

Generative model로 reverse를 학습하는 것

2.1 forward trajectory

초기 데이터는 $\sim q(x_0)$

훈련도메인 $\sim \pi(y)$와 같은 것이 목표

↳ Markov chain kernel $T_\pi(y|y';\beta)$를 반복적으로 수행

$$\pi(y) = \int dy' T_\pi(y|y';\beta)\pi(y')$$

$$q(x_t|x_{t-1}) = T_\pi(x_t|x_{t-1};\beta_t)$$

⇓ $0 \sim T$ 적용

$$q(x_{T:0}) = q(x_0)\prod_{t=1}^{T} q(x_t|x_{t-1})$$

2.2 reverse trajectory

$$p(x_T) = \pi(x_T)$$

$$p(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t)$$

$\beta_1, \beta_2, \beta_3 \sim \cdots \beta_T$

$\sqrt{\alpha_4}\left[\sqrt{\alpha_3}\left(\sqrt{\alpha_2}\left(\sqrt{\alpha_1}x_0 + \beta_1 z\right) + \beta_2 z\right) + \beta_3 z\right) + \beta_4 z$

$\sqrt{\alpha_2}x_0 + \sqrt{\alpha_2}\beta_1 z + \beta_2 z$

$\sqrt{\alpha_3}x_0 + \sqrt{\alpha_3}\sqrt{\alpha_2}\beta_1 z + \sqrt{\alpha_3}\beta_2 z + \beta_3 z$

$\sqrt{\alpha_4}x_0 + \sqrt{\alpha_4}\sqrt{\alpha_3}\sqrt{\alpha_2}\beta_1 z + \sqrt{\alpha_4}\sqrt{\alpha_3}\beta_2 z + \sqrt{\alpha_4}\beta_3 z + \beta_4 z$

$z\left(\sqrt{\alpha_4}\sqrt{\alpha_3}\sqrt{\alpha_2}\beta_1 + \sqrt{\alpha_4}\sqrt{\alpha_3}\beta_2 + \sqrt{\alpha_4}\beta_3 + \beta_4\right)$

$\alpha_4\alpha_3\alpha_2(1-\alpha_1) + \alpha_4\alpha_3(1-\alpha_2) + \alpha_4(1-\alpha_3) + (1-\alpha_4)$

$1 - \alpha_1\alpha_2\alpha_3\cdots\alpha_T$

$\alpha_4\alpha_3\alpha_2 - \alpha_1\alpha_2\alpha_3\alpha_4 - \alpha_4\alpha_3\alpha_2 + \alpha_4\alpha_3 - \alpha_3\alpha_4 + \alpha_4 + 1 - \alpha_4$

$t-1 \to t : N\left(\sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$

$0 \to t : N\left(\sqrt{\bar{\alpha}_t}x_{t-1}, (1-\bar{\alpha}_t)I\right) \to$ 0~1이 더해지는 것

$t-1 \to t : N\left(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$

$L_3 = \beta_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$

$q(x_{t-1}|x_t, x_0)$ $\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$

$q(x_t|x_{t-1}) = \sqrt{1-\beta_t}x_{t-1} + \beta_t I$

$q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + (1-\bar{\alpha}_t)I$

$\sqrt{1-\beta_t}\left(\sqrt{\bar{\alpha}_{t-1}}x_0 + (1-\bar{\alpha}_{t-1})\right) + \beta_t = x_t$

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + (1-\bar{\alpha}_t) = \sqrt{1-\beta_t}\, x_{t-1} + \beta_t$$

$$x_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\left(\sqrt{\bar{\alpha}_t}\, x_0 + (1-\bar{\alpha}_t)\right)$$

$$= \frac{\alpha_t(1-\bar{\alpha}_{t-1})\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t} x_0 + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})$$

$$\Rightarrow \frac{(\alpha_t - \bar{\alpha}_t + \beta_t)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t} x_0 + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \quad (1-\alpha_t)$$

$$= \sqrt{\bar{\alpha}_{t-1}}\, x_0 + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) + \frac{1-\bar{\alpha}_{t-1}-\alpha_t+\bar{\alpha}_t}{1-\bar{\alpha}_t}$$

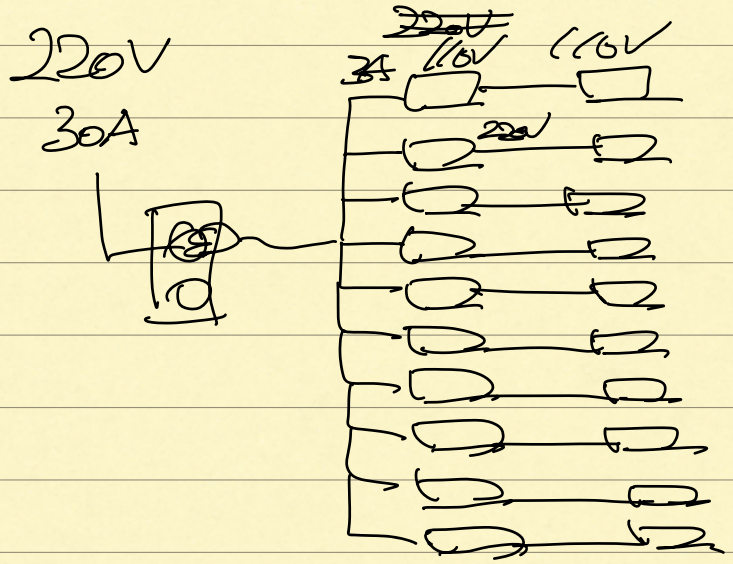
$$= \sqrt{\bar{\alpha}_{t-1}}\, x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})(1-\alpha_t) + 1-\bar{\alpha}_{t-1}-\alpha_t+\bar{\alpha}_t}{1-\bar{\alpha}_t}$$

$$\frac{\sqrt{\alpha_t}(1-\alpha_t-\bar{\alpha}_{t-1}+\bar{\alpha}_t\bar{\alpha}_{t-1}) + 1-\bar{\alpha}_{t-1}-\alpha_t+\bar{\alpha}_t}{1-\bar{\alpha}_t}$$

$$\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})(1-\alpha_t) + (1+\bar{\alpha}_t-\bar{\alpha}_{t-1}-\alpha_t)}{1-\bar{\alpha}_t} = 1-\bar{\alpha}_{t-1}$$

$$\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) + \frac{-1+\bar{\alpha}_t+1-\bar{\alpha}_{t-1}-\alpha_t}{1-\bar{\alpha}_t}$$

$$= \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) - 1 + \frac{1-\alpha_t+1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$$


220V
30A
220V
110V
110V
220V