Generative modeling

: use the dataset to learn a model $p_\theta(x)$ for generating new samples from $p_d(x)$

(Stein) Score matching

: learn an unnormalized density $\tilde{p}_d(x)$ s.t. $p_d(x) = \frac{\tilde{p}_d(x)}{\int \tilde{p}_d(x)\,dx} \to Z$

⇒ score ($s(x)$): $\nabla_x \log p_d(x) = \nabla_x \log \tilde{p}_d(x) - \cancel{\nabla_x Z}$

→ no need to take care of the intractable partition function Z

⇒ score function ($s_\theta(x)$): $\nabla_x \log p_\theta(x)$

→ $\min_\theta L(\theta) = \frac{1}{2} \mathbb{E}_{p_d(x)} \left[\| s_\theta(x) - s(x) \|_2^2 \right]$

(f-divergence between $p_d(x)$ and $p_\theta(x)$)

→ $\min_\theta J(\theta) = \mathbb{E}_{p_d(x)} \left[\underbrace{\mathrm{tr}(\nabla_x s_\theta(x))}_{\text{computational burden}} + \frac{1}{2} \| s_\theta(x) \|_2^2 \right]$

⇒ Sliced score matching : faster computation

→ $\min_\theta L(\theta, p_v) = \frac{1}{2} \mathbb{E}_{p_v} \mathbb{E}_{p_d(x)} \left[(v^T s_\theta(x) - v^T s(x))^2 \right]$

(projecting score (function) to random direction $v \sim p_v$ s.t. $\mathbb{E}_{p_v}[vv^T] \succ 0$, $\mathbb{E}_{p_v}[\|v\|_2^2] < \infty$)

→ $\min_\theta J(\theta, p_v) = \mathbb{E}_{p_v} \mathbb{E}_{p_d(x)} \left[v^T \nabla_x s_\theta(x) v + \frac{1}{2} (v^T s_\theta(x))^2 \right]$

$= \mathbb{E}_{p_d(x)} \left[\underbrace{\mathbb{E}_{p_v} \left[v^T \nabla_x s_\theta(x) v \right]}_{\text{less computation } (\because \nabla_x v^T s_\theta(x) = v^T \nabla_x s_\theta(x))} + \frac{1}{2} \| s_\theta(x) \|_2^2 \right]$

⇒ Denoised score-matching : learn denoising direction

→ $q_\sigma(\tilde{x}|x) = \mathcal{N}(\tilde{x}|x, \sigma^2 I)$ s.t. $\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) = -\frac{\tilde{x} - x}{\sigma^2}$

→ $\min_\theta l(\theta, \sigma) = \frac{1}{2} \mathbb{E}_{p_d(x)} \mathbb{E}_{q(\tilde{x}|x)} \left[\| s_\theta(\tilde{x}, \sigma) + \frac{\tilde{x} - x}{\sigma^2} \|_2^2 \right]$

Score-based generative modeling

: sample data via Langevin dynamics using the estimated score function

⇒ Langevin dynamics

→ $\tilde{x}_t = \tilde{x}_{t-1} + \frac{\epsilon}{2} s_\theta(\tilde{x}_{t-1}) + \sqrt{\epsilon}\, z_t$ where $z_t \sim N(0, I)$

→ the distribution of $\tilde{x}_T$ equals $p_d(x)$ when $\epsilon \to 0$, $T \to \infty$

⇒ challenges { inaccurate $s_\theta(x)$ due to manifold hypothesis
slow mixing of Langevin dynamics due to low data density region

⇒ Noise Conditional Score Networks (NCSN)

① perturb data with random Gaussian noise for better score-matching

(use multiple noise level and train a single conditioned score function $s_\theta(x, \sigma)$
s.t. $\frac{\sigma_1}{\sigma_2} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$

→ $\min_\theta L(\theta, \{\sigma_i\}_{i=1}^L) = \frac{1}{L} \sum_{i=1}^{L} \lambda(\sigma_i)\, \ell(\theta, \sigma_i)$ ↷ denoising score-matching loss.

where $\ell(\theta, \sigma_i) = \frac{1}{2} \mathbb{E}_{p_d(x)} \mathbb{E}_{g(\tilde{x}|x)} \left[\left\| s_\theta(\tilde{x}, \sigma_i) + \frac{\tilde{x} - x}{\sigma_i^2} \right\|_2^2 \right]$ and $\lambda(\sigma_i) = \sigma_i^2$

② initially use scores with large noise and gradually scale down the noise

→ $\tilde{x}_t = \tilde{x}_{t-1} + \frac{\alpha_i}{2} s_\theta(\tilde{x}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, z_t$ where $\alpha_i = \epsilon \cdot \frac{\sigma_i^2}{\sigma_L^2}$, $z_t \sim N(0, I)$

Denoising Diffusion Probabilistic Models (DDPM)

: sample data via a parameterized Markov Chain trained to reverse the diffusion process

⇒ diffusion process

→ given x_0, forward pass gradually adds noise until signal is destroyed

⇒ $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$ where $q(x_t|x_{t-1}) = \mathcal{N}(x_t|\sqrt{1-\beta_t}\,x_{t-1} + \beta_t I)$

→ allow sampling x_t at an arbitrary time-step t in closed-form

⇒ $q(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\bar{\alpha}_t}\,x_0, (1-\bar{\alpha}_t)I)$ where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$

→ given x_0, backward pass can be estimated in closed-form

⇒ $$q(x_{t-1}|x_t, x_0) = \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} q(x_t|x_{t-1})$$

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

$$= \mathcal{N}\left(x_{t-1}\middle| \underbrace{\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t}_{\tilde{\mu}_t(x_t, x_0)}, \underbrace{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I}_{\tilde{\Sigma}_t(x_t, x_0)}\right)$$

⇒ reverse process

→ $p(x_{0:T}) = p(x_T) \cdot \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$ where $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}|\mu_\theta(x_t,t), \Sigma_\theta(x_t,t))$

→ $\mu_\theta(x_t,t)$ and $\Sigma_\theta(x_t,t)$ can be formulated according to $q(x_{t-1}|x_t,x_0)$

⇒ $\mu_\theta(x_t,t) = \tilde{\mu}_t\left(x_t, \underbrace{\frac{1}{\sqrt{\alpha_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon_\theta(x_t,t)\right)}_{x_0}\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t,t)\right)$

⇒ $\Sigma_\theta(x_t,t) = \tilde{\Sigma}_t(x_t,x_0) = \sigma_t^2 I$

→ Training with variational inference

⇒ $\min_\theta \mathbb{E}_{p_d(x_0)}[-\log p_\theta(x_0)] = \mathbb{E}_{p_d(x_0)}\left[\int -\log q(x_{1:T}|x_0)\cdot\frac{p(x_{0:T})}{q(x_{1:T}|x_0)}dx_{1:T}\right]$

$\le \mathbb{E}_{q(x_{1:T}|x_0)p_d(x_0)}\left[-\log\frac{p(x_{0:T})}{q(x_{1:T}|x_0)}\right]$

$= \mathbb{E}_{q(x_{1:T}|x_0)p_d(x_0)}\left[-\log p(x_T) - \sum_{t\ge1}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$

$= \mathbb{E}_{q(x_{1:T}|x_0)p_d(x_0)}\left[-\log p(x_T) - \sum_{t>1}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - \log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$

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

$= \mathbb{E}_{q(x_{1:T}|x_0)p_d(x_0)}\left[\underbrace{KL(q(x_T|x_0)\|p(x_T))}_{0} + \sum_{t>1}\underbrace{KL(q(x_{t-1}|x_t,x_0)\|p_\theta(x_{t-1}|x_t))}_{L(\theta,t)} - \log p(x_0|x_1)\right]$

⇒ $\min_\theta L(\theta,t) = \mathbb{E}_{q(x_{1:T}|x_0)p_d(x_0)}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)\|_2^2\right] + C$ predict $\mu_\theta(x_t,t)$

$= \mathbb{E}_{\varepsilon p_d(x_0)}\left[\underbrace{\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}}_{\text{remove for simplicity}}\|\varepsilon - \varepsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon}_{x_t}, t)\|_2^2\right]$ or predict $\varepsilon_\theta(x_t,t)$

↳ remove for simplicity ⇒ down-weight for small t

⇒ more focus on large t for large noise

⇒ inference

Step 1: $x_T \sim \mathcal{N}(0,I)$

Step 2: for $t = T,\cdots,1$, $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$

i) $z_t \sim \mathcal{N}(0,I)$

ii) $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t,t)\right) + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t z_t$ ≈ Langevin dynamics

Denoising Diffusion Implicit Models (DDIM)

: use deterministic non-Markovian diffusion process
- enable consistency
- enable latent traverse
- enable short generative process
- enable non-Gaussian diffusion process (Categorical)

⇒ non-Markovian forward process

→ $q(x_{1:T}|x_0) = q(x_T|x_0) \cdot \prod_{t=2}^{T} q(x_{t-1}|x_t, x_0)$

where $q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1} \middle| \sqrt{\bar{\alpha}_{t-1}}\, x_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \frac{x_t - \sqrt{\bar{\alpha}_t}\, x_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2 I\right)$

s.t. $q(x_t|x_0) = \mathcal{N}\left(x_t \middle| \sqrt{\bar{\alpha}_t}\, x_0, \sqrt{1-\bar{\alpha}_t}\, I\right)$

⇒ reverse process

: given x_t, predict x_0, and estimate x_{t-1}

→ $p_\theta(x_{t-1}|x_t) = \begin{cases} \mathcal{N}(x_{t-1}|f_\theta(x_1, 1), \sigma_1^2 I) & \text{if } t=1 \\ q(x_{t-1}|x_t, f_\theta(x_t, t)) & \text{otherwise} \end{cases}$

where $f_\theta(x_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t} \cdot \varepsilon_\theta(x_t, t))$ is the predicted x_0 from $q(x_t|x_0)$

⇒ inference

Step 1: $x_T \sim \mathcal{N}(0, I)$

Step 2: for $t = T, \cdots, 1$, $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$

i) $z_t \sim \mathcal{N}(0, I)$

ii) $x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot f_\theta(x_t, t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \varepsilon_\theta(x_t, t) + \sigma_t z_t$

$\varepsilon_\theta(x_t, t) \to \frac{x_t - \sqrt{\bar{\alpha}_t} f_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}}$

(if $\sigma_t^2 = 0$ for all t, it becomes an implicit model)

(a subset of time steps can be used if $\sum_{t \notin s} KL(q(x_t|x_0) \| p_\theta(x_0|x_t))$ is added to the loss)

$p_\theta(x_0|x_t) \to \mathcal{N}(x_0|f_\theta(x_t, t), \sigma_t^2 I)$

Improved Denoising Diffusion Probabilistic Models

: achieve competitive log-likelihood with high sample quality

⇒ learn the reverse process variances and use hybrid objective

→ $\Sigma_\theta(x_t, t) = \exp\left(v_\theta(x_t, t) \log \beta_t + (1 - v_\theta(x_t, t)) \log \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t\right)$

→ $L_{hybrid} = L_{simple}$ (for $\mu_\theta(x_t, t)$) $+ L_{vlb}$ (for $\Sigma_\theta(x_t, t)$)

⇒ improve the noise schedule

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

⇒ reduce gradient noise of L_{vlb}

→ $L_{vlb} = \mathbb{E}_{p_t}\left[\frac{L_t}{p_t}\right]$ where $p_t \propto \sqrt{\mathbb{E}[L_t^2]}$ and $\sum p_t = 1$

↳ maintain a history of the previous 10 values

Diffusion Models Beat GAN on Image Synthesis

⇒ architecture improvement

★ U-Net
- stack of residual layers & downsampling convolutions
- stack of residual layers & upsampling convolutions
- skip connection connecting layers with the same sampling size
- global attention layer at 16x16 resolution with a single head
- projection of the timestep embedding into each residual block

(△) ⇒ increase depth versus width, holding model size relatively constant

(○) ⇒ increase the number of attention heads

(○) ⇒ use attention at 32x32, 16x16, and 8x8 resolutions rather than not only 16x16

(○) ⇒ use the BigGAN residual block for upsampling and downsampling

(X) ⇒ rescale residual connections with $\frac{1}{\sqrt{2}}$

(○) ⇒ more heads or fewer channels per head

(○) ⇒ use adaptive group normalization to incorporate the timestep and class embedding into residual block

$$\left(\mathrm{AdaGN}(h, y) = y_{\text{time}}\,\mathrm{GN}(h) + y_{\text{class}} \quad \text{s.t.} \quad y = [y_{\text{time}}, y_{\text{class}}] \right)$$

⇒ classifier guided sampling

: train a classifier $p_\phi(y|x_{t-1})$ and use $\nabla_{x_{t-1}} \log p_\phi(y|x_{t-1})$ to further guide inference

→ conditional diffusion process

(Suppose $\hat{q}(y|x_0)$ is given. Then, $\hat{q}$ behaves like q)

① $\hat{q}(x_t|x_{t-1}, y) := q(x_t|x_{t-1}) \Longrightarrow \hat{q}(x_t|x_{t-1}) = q(x_t|x_{t-1})$

② $\hat{q}(x_{1:T}|x_0, y) := \prod_{t=1}^{T} \hat{q}(x_t|x_{t-1}, y) \Longrightarrow \hat{q}(x_{1:T}|x_0) = q(x_{1:T}|x_0)$

$\therefore$ $\hat{q}(x_t) = q(x_t)$ and $\hat{q}(x_{t-1}|x_t) = q(x_{t-1}|x_t)$ and $\hat{q}(y|x_{t-1}, x_t) = \hat{q}(y|x_t)$

→ conditional reverse process

$$\Rightarrow \hat{q}(x_{t-1}|x_t, y) = \frac{q(x_{t-1}|x_t)\,\hat{q}(y|x_{t-1}, x_t)}{\hat{q}(y|x_t)} = \frac{q(x_{t-1}|x_t)\,\hat{q}(y|x_{t-1})}{\hat{q}(y|x_t)}$$

$$\propto \underset{\overset{\text{ss}}{p_\theta(x_{t-1}|x_t)}}{q(x_{t-1}|x_t)}\ \underset{\overset{\text{ss}}{p_\phi(y|x_{t-1})}}{\hat{q}(y|x_{t-1})}$$

$$\Rightarrow \nabla_{x_{t-1}} \log \hat{q}(x_{t-1}|x_t, y) = \nabla_{x_{t-1}} \log p_\theta(x_{t-1}|x_t) + \nabla_{x_{t-1}} \log p_\phi(y|x_{t-1})$$

→ $\log p_\theta(x_{t-1}|x_t) = \log \mathcal{N}(\mu, \Sigma) = -\frac{1}{2}(x_{t-1}-\mu)^T \Sigma^{-1}(x_t-\mu) + C$

→ $\log p_\phi(y|x_{t-1}) \approx \log p_\phi(y|x_{t-1})\big|_{x_{t-1}=\mu} + (x_{t-1}-\mu)\nabla_{x_{t-1}} \log p_\phi(y|x_{t-1})\big|_{x_{t-1}=\mu}$

$$= (x_{t-1}-\mu)\,g + C_1$$

→ $\log p_\theta(x_{t-1}|x_t)\,p_\phi(y|x_{t-1}) \approx -\frac{1}{2}(x_t-\mu-\Sigma g)^T \Sigma^{-1}(x_t-\mu-\Sigma g) + C_2$

$$= \log \mathcal{N}(\mu + \underset{\text{shifting mean}}{\underbrace{\Sigma g}}, \Sigma) + C_3$$

⇒ inference

Step 1: $x_T \sim N(0, I)$

Step 2: for $t = T, \cdots, 1$,

"option 1": $x_{t-1} \sim \hat{q}(x_{t-1} | x_t, y)$

i) estimate $p_\theta(x_{t-1}|x_t) = N(x_{t-1} | \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$

ii) $x_{t-1} \sim N(\mu_\theta(x_t, t) + s \Sigma_\theta(x_t, t) \nabla_{x_t} \log p_\phi(y|x_t), \Sigma_\theta(x_t, t))$

s introduces trade-off b/t fidelity and diversity
s needs to be set to larger than 1.

"option 2": DDIM sampling with calibrated noise prediction

i) $\hat{\epsilon} = \epsilon_\theta(x_t, t) - \sqrt{1-\bar{\alpha}_t} \nabla_{x_t} \log p_\phi(y|x_t) \Longrightarrow$ score function for $p(x_t, y) = p(x_t) \cdot p(y|x_t)$

$\left(\because \nabla_{x_t} \log p_\theta(x_t) + \nabla_{x_t} \log p_\phi(y|x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) + \nabla_{x_t} \log p_\phi(y|x_t) \right)$

ii) $x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot \frac{x_t - \sqrt{1-\bar{\alpha}_t}\, \hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \hat{\epsilon}$