ESE 6510

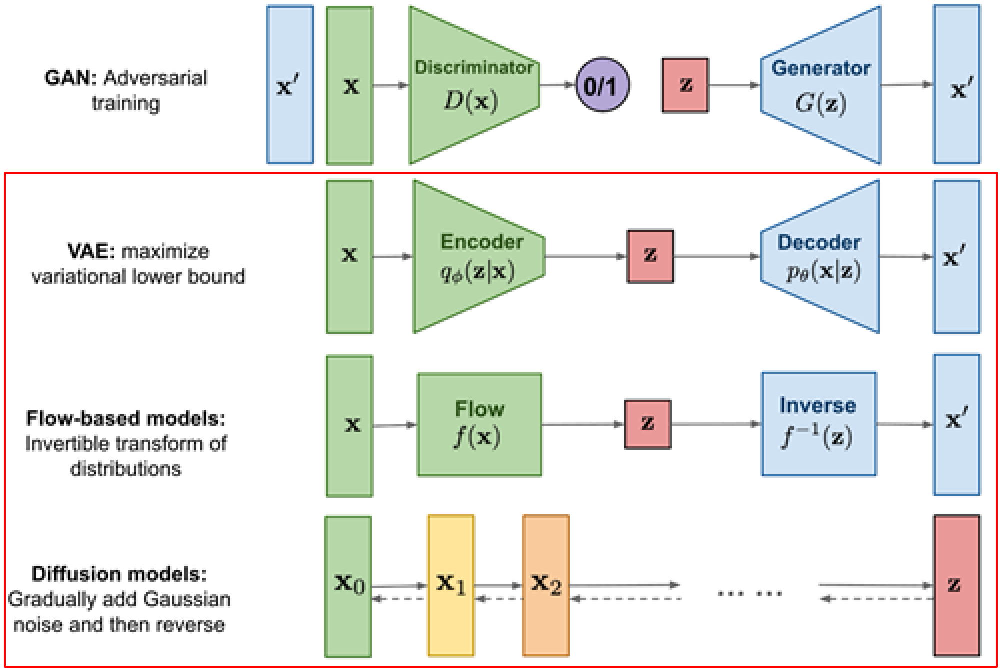
Generative Models

Presented by: Chunwei Xing

Vincent van Gogh, 1889

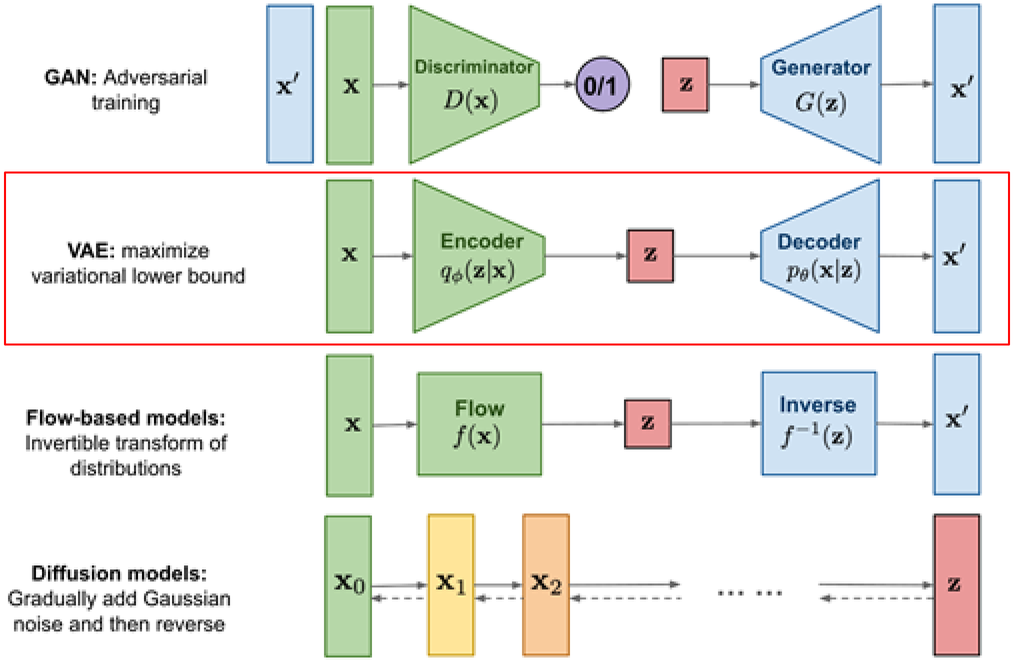
Generative Models

- ❑ Variational Autoencoder
- ❑ Diffusion Models
- ❑ Flow-based Models



Generative Models

- ❑ Variational Autoencoder
- ❑ Diffusion Models
- ❑ Flow-based Models



Autoencoder - A Brief History

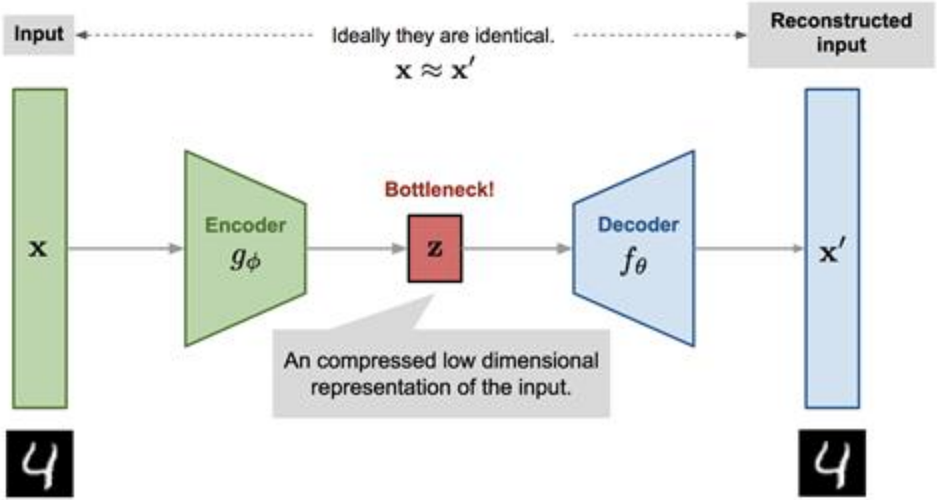
- 1982 — **PCA**: Oja showed PCA is equivalent to a 1-hidden-layer linear neural net
- 1989–1991 — **Nonlinear PCA**: Baldi & Hornik (1989) and Kramer (1991) generalized PCA to neural "autoassociative" networks
- Mid-late 1980s — **Auto-association**: The idea to run a neural net in "auto-association mode" (1986) was implemented for speech (1987–88) and images (1987).
- Early 1990s — **Applications**: dimensionality reduction/feature learning
- 2006 — **Deep revival via pretraining**: Hinton & Salakhutdinov popularized deep autoencoders using layer-wise pretraining
- Nowadays: **generative modeling** using VAE for large-scale generative AI

Autoencoders

- Compressed representation
- Unsupervised learning
- Encoder network: $\mathbf{z} = g_\phi(\mathbf{x})$
- Decoder network: $\mathbf{x}' = f_\theta(g_\phi(\mathbf{x}))$
- Reconstruction loss:

$$L_{\text{AE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$

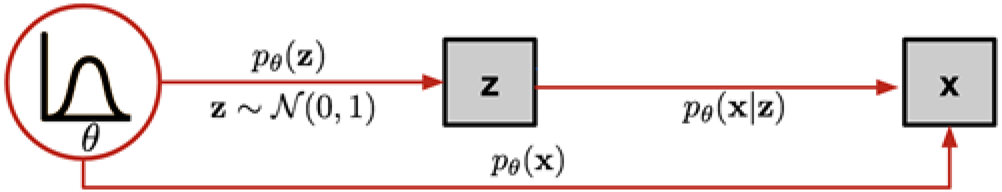
- But it's not generative!



Variational Autoencoder

- ❑ Prior $p_\theta(z)$
- ❑ Likelihood $p_\theta(x|z)$
- ❑ Maximize the log-likelihood:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p_\theta(\mathbf{x}^{(i)})$$



- ❑ Compute $\log p_\theta(x^{(i)}) = \log \int p_\theta(x^{(i)} \mid z)p(z)\mathrm{d}z$
- ❑ What's the issue?
  - ❑ No closed-form expression for general neural network parameterizations
  - ❑ Expensive to approximate the integral over many latents for each data point

Variational Inference

- **Theorem:** the likelihood can be written as $\log p_\theta(x) = \max_{q(\cdot|x):q(\cdot|x)\geq 0, \int q(z|x)\mathrm{d}z=1} \int q(z \mid x) \log \frac{p_\theta(x,z)}{q(z \mid x)} \mathrm{d}z.$

  and the maximizing distribution is given by $q^*(z \mid x) = p_\theta(z \mid x)$

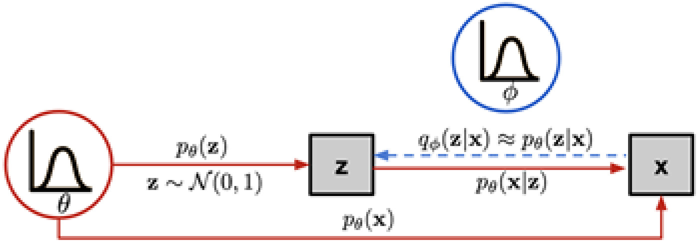
- Therefore, the new objective is given by

$$\max_\theta \sum_{i=1}^n \log p_\theta(x^{(i)}) = \max_\theta \max_{q(\cdot|x^{(i)}),\forall i} \sum_{i=1}^n \int q(z \mid x^{(i)}) \log \frac{p_\theta(x^{(i)},z)}{q(z \mid x^{(i)})} \mathrm{d}z$$

- Approximate the posterior with neural networks parameterized by $q_\phi(z|x)$

$$\max_\theta \max_\phi \sum_{i=1}^n \int q_\phi(z|x^{(i)}) \log \frac{p_\theta(x^{(i)},z)}{q_\phi(z|x^{(i)})} \mathrm{d}z$$

- Is the new objective tractable now?



Proof - VI Theorem

$$\log p_\theta(x) = \int q(z|x) \log p_\theta(x) \mathrm{d}z$$
$$= \int q(z|x) \log \frac{p_\theta(x,z)}{p_\theta(z|x)} \mathrm{d}z$$
$$= \int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)} \cdot \frac{q(z|x)}{p_\theta(z|x)} \mathrm{d}z$$
$$= \underbrace{\int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)} \mathrm{d}z}_{\text{Evidence Lower Bound (ELBO)}} + \underbrace{\int q(z|x) \log \frac{q(z|x)}{p_\theta(z|x)} \mathrm{d}z}_{\mathrm{KL}[q(z|x)\|p_\theta(z|x)]}$$

- ❑ Given that $\int q(z|x) dz = 1$

$$\max_{q(\cdot|x)} \int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)} \mathrm{d}z = \max_{q(\cdot|x)} \log p_\theta(x) - \mathrm{KL}[q(z|x)\|p_\theta(z|x)]$$
$$= \log p_\theta(x) - \min_{q(\cdot|x)} \mathrm{KL}[q(z|x)\|p_\theta(z|x)]$$
$$= \log p_\theta(x)$$

- ❑ Given that KL divergence non-negative and

$$\mathrm{KL}(q\|p) = 0 \text{ iff } p = q$$

Variational Autoencoder

$$\mathcal{L}(\theta,\phi;\boldsymbol{x}) = \underbrace{\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{\mathrm{KL}(q_\phi(\boldsymbol{z}\mid\boldsymbol{x})\,\|\,p(\boldsymbol{z}))}_{\text{regularization to prior}}$$

- Learning objective: maximize the ELBO
  - Maximize the likelihood of generating real data (decoder)
  - Minimize the difference between the prior and posterior distributions (encoder)
- An example
  - Encoder: $q_\phi(\boldsymbol{z}\mid\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z};\, \boldsymbol{\mu}_\phi(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}))).$
  - Prior: $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$
  - Decoder: $p_\theta(\boldsymbol{x}\mid\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x};\, \boldsymbol{\mu}_\theta(\boldsymbol{z}),\, \eta\,\boldsymbol{I}).$
  - Reconstruction term: $\mathbb{E}_{q_\phi}\left[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z})\right] \approx -\frac{1}{2\eta M}\sum_{m=1}^{M}\|\boldsymbol{x}-\boldsymbol{\mu}_\theta(\boldsymbol{z}^{(m)})\|_2^2 + \text{const.}$
  - Regularization term: $\mathrm{KL}(\mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}\,\boldsymbol{\sigma}^2)\,\|\,\mathcal{N}(\boldsymbol{0},\boldsymbol{I})) = \frac{1}{2}\sum_{j=1}^{d}\left(\sigma_j^2 + \mu_j^2 - 1 - \log\sigma_j^2\right).$

VAE - Reparameterization Tricks

- We can estimate gradients wrt. θ using MC estimation

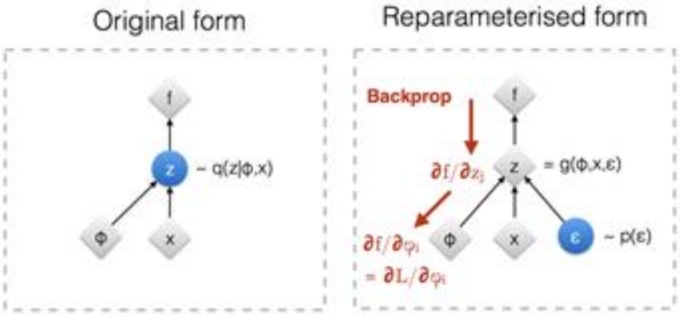
$$\nabla_\theta \mathcal{L}(\theta, \phi; \boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_\theta \log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] \approx \frac{1}{M}\sum_{m=1}^{M} \nabla_\theta \log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}^{(m)}).$$

- But not wrt. ϕ

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)] \neq \mathbb{E}_{z\sim q_\phi(z|x)} \nabla_\phi [\log p_\theta(x,z) - \log q_\phi(z|x)]$$

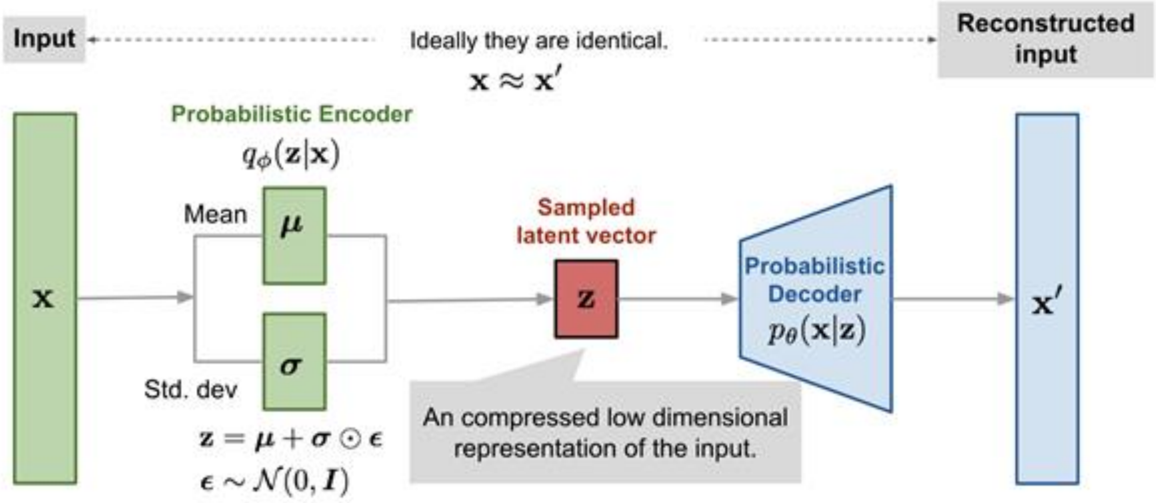
- Reparameterization and sample $z_\phi = g(\epsilon, \phi, x) = \mu_\phi(x) + \sigma_\phi(x)\epsilon \quad \text{for } \epsilon \sim \mathcal{N}(0, I)$
- Then we can estimate the gradients

$$\begin{aligned}\nabla_\phi \mathcal{L}_{\theta,\phi}(x) &= \nabla_\phi \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)] \\ &= \nabla_\phi \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}[\log p_\theta(x,z_\phi) - \log q_\phi(z_\phi|x)]\end{aligned}$$



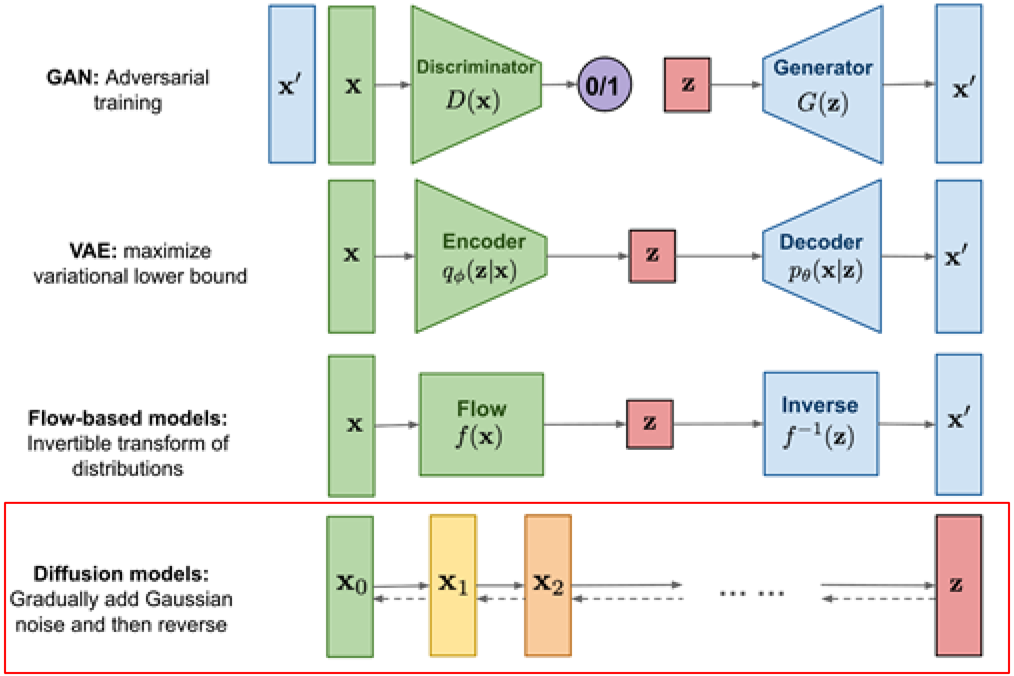
Variational Autoencoder

- Beta-VAE
- Joint-VAE
- VQ-VAE
- ...

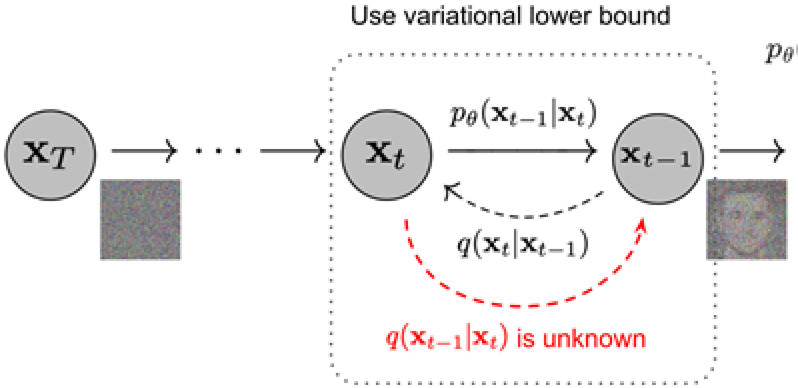


Generative Models

- ❑ Variational Autoencoder
- ❑ **Diffusion Models**
- ❑ Flow-based Models



Diffusion Models



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

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

- ❑ Forward diffusion process: define a **Markov chain** of diffusion steps to slowly add random noise to data
- ❑ Reverse diffusion process: construct desired data samples from the noise
- ❑ Connections to the VAE? Encoder? Decoder? Latents? Prior? Posterior?

Forward Diffusion Process

- Compute $q(x_t|x_0)$
- Given: $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ $\quad \alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} && \text{reparameterization trick: } \boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} && \text{merges two Gaussians} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}
\end{aligned}$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

- Enable sampling at any time t
- $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$?

Reverse Diffusion Process

- Compute $q(x_{t-1}|x_t, x_0)$

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

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

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

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

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

Reverse Diffusion Process

- ❑ Compute $\tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t$

$$\tilde{\beta}_t = 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = 1/(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$$

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

ELBO

- Recall the ELBO from VAE: $\int q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)} \mathrm{d}z$
- For the diffusion process:
- Joint distribution: $p_\theta(\mathbf{x}_0, \mathbf{x}_{1:T}) = p_\theta(\mathbf{x}_T) p_\theta(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
- Posterior: $q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0) = q_\phi(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1})$
- Substitute to get the ELBO for diffusion:

$$\mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:T})}{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_T) p_\theta(\mathbf{x}_0|\mathbf{x}_1) \prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_\phi(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1})} \right]$$

ELBO Derivation

- Compute $q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1}) = q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) q_\phi(\mathbf{x}_t|\mathbf{x}_0)}{q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_0)}$
- Substitute into the ELBO:

$$
\begin{aligned}
\log p(\mathbf{x}) &\geq \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_\phi(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^{T} q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_\phi(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^{T} q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q_\phi(\mathbf{x}_t|\mathbf{x}_0)/q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_0)}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)q_\phi(\mathbf{x}_1|\mathbf{x}_0)}{q_\phi(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^{T} q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q_\phi(\mathbf{x}_T|\mathbf{x}_0)}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)q_\phi(\mathbf{x}_1|\mathbf{x}_0)}{q_\phi(\mathbf{x}_1|\mathbf{x}_0)q_\phi(\mathbf{x}_T|\mathbf{x}_0)}\prod_{t=2}^{T}\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}\right]
\end{aligned}
$$

ELBO Derivation

- Reorganize into three terms

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_T)p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q_\phi(\mathbf{x}_T|\mathbf{x}_0)} \prod_{t=2}^{T} \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}\right]$$

$$= \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] + \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_T)}{q_\phi(\mathbf{x}_T|\mathbf{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q_\phi(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}\right]$$

$$= \mathbb{E}_{q_\phi(\mathbf{x}_1|\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] + \mathbb{E}_{q_\phi(\mathbf{x}_T|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_T)}{q_\phi(\mathbf{x}_T|\mathbf{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q_\phi(\mathbf{x}_t|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}\right]$$

$$= \underbrace{\mathbb{E}_{q_\phi(\mathbf{x}_1|\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right]}_{\text{Reconstruction term}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(\mathbf{x}_T|\mathbf{x}_0) \| p_\theta(\mathbf{x}_T)\right)}_{\text{Prior matching term}} - \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q_\phi(\mathbf{x}_t|\mathbf{x}_0)}\left[D_{\mathrm{KL}}\left(q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)\right]}_{\text{Score matching term}}$$

$$= \underbrace{\mathbb{E}_{q_\phi(\mathbf{x}_1|\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right]}_{L_0} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(\mathbf{x}_T|\mathbf{x}_0) \| p_\theta(\mathbf{x}_T)\right)}_{L_T} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q_\phi(\mathbf{x}_t|\mathbf{x}_0)}\left[D_{\mathrm{KL}}\left(q_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)\right]}_{L_{t-1}}$$

ELBO Derivation

- Recall $D_{\mathrm{KL}}(\mathcal{N}_0 \| \mathcal{N}_1) = \frac{1}{2}\left[\mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) - k + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \ln\left(\frac{\det \boldsymbol{\Sigma}_1}{\det \boldsymbol{\Sigma}_0}\right)\right]$
- The score matching term at timestep t in [2, T]:

$$
\begin{aligned}
L_{t-1} &= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{1}{2\|\Sigma_\theta(\mathbf{x}_t,t)\|_2^2}\|\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) - \mu_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{1}{2\|\Sigma_\theta\|_2^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t,t)\right)\right\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta\|_2^2}\|\epsilon_t - \epsilon_\theta(\mathbf{x}_t,t)\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta\|_2^2}\left\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\right\|^2\right]
\end{aligned}
$$

- What about the other two terms? L_T, L_0

DDPM algorithm

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

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \ldots, T\})$
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

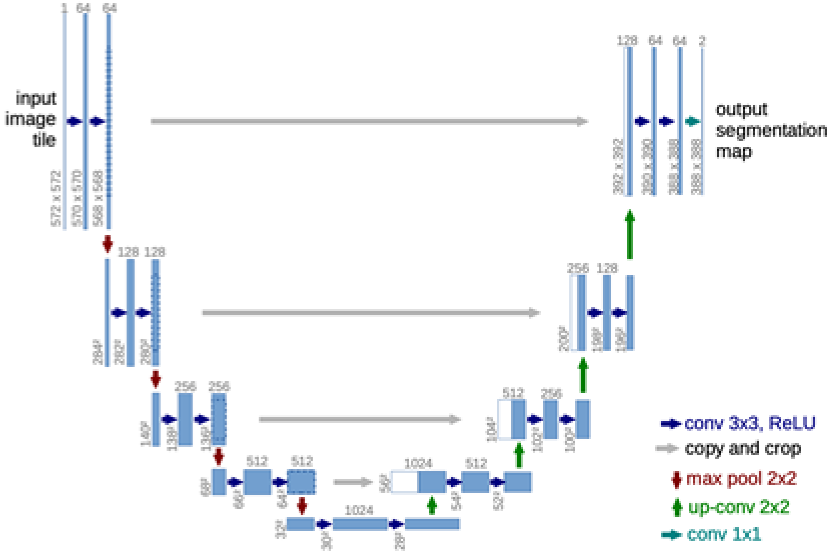
Algorithm 2 Sampling

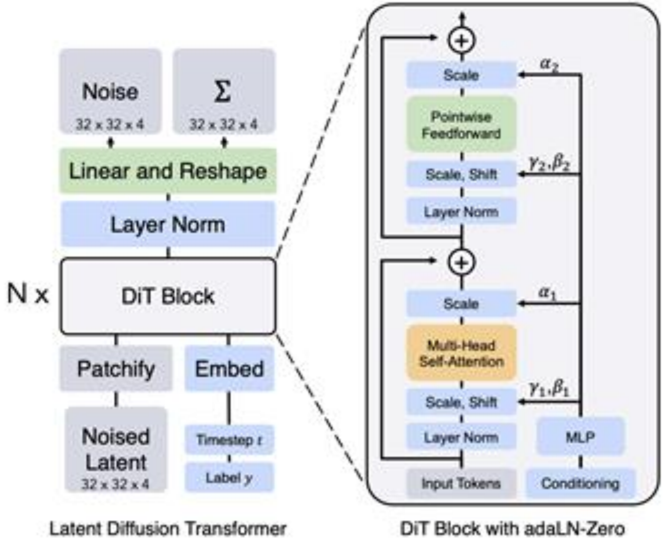
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,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

- We usually choose T to be a large number, e.g. 1k, 2k, 4k to have better performance
- Sampling is expensive. DDIM, consistency models, distillation...

Backbones - UNet & DiT

❑ Conditioning methods: FiLM, AdaLN





Classifier-Guided Diffusion

- Score function: $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{1-\bar{\alpha}_t} = -\frac{\epsilon_t}{\sqrt{1-\bar{\alpha}_t}}$
- Joint distribution of data samples and class labels: $q(\mathbf{x}_t, y)$
- Score function for the joint distribution:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t, y) &= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(y|\mathbf{x}_t) \\
&\approx -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log \boxed{f_\phi(y|\mathbf{x}_t)} \\
&= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(\epsilon_\theta(\mathbf{x}_t, t) - \sqrt{1-\bar{\alpha}_t}\nabla_{\mathbf{x}_t} \log f_\phi(y|\mathbf{x}_t)\right)
\end{aligned}$$

Trained classifier

- New classifier-guided noise predictor:

$$\bar{\epsilon}_\theta(\mathbf{x}_t, t) = \epsilon_\theta(\mathbf{x}_t, t) - \sqrt{1-\bar{\alpha}_t}\, w \nabla_{\mathbf{x}_t} \log f_\phi(y|\mathbf{x}_t) \longleftarrow$$

Ablated diffusion model (ADM)

Classifier-Free-Guided Diffusion

- What if there's no trained classifier?
- Consider the conditional distribution using Bayes rule:

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}(\epsilon_\theta(\mathbf{x}_t, t, y) - \epsilon_\theta(\mathbf{x}_t, t, y=\emptyset))\end{aligned}$$

- Then we have the noise predictor with class labels guidance:

$$\begin{aligned}\bar{\epsilon}_\theta(\mathbf{x}_t, t, y) &= \epsilon_\theta(\mathbf{x}_t, t, y) - \sqrt{1-\bar{\alpha}_t}\, w \nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) \\ &= \epsilon_\theta(\mathbf{x}_t, t, y) + w(\epsilon_\theta(\mathbf{x}_t, t, y) - \epsilon_\theta(\mathbf{x}_t, t)) \\ &= (w+1)\epsilon_\theta(\mathbf{x}_t, t, y) - w\epsilon_\theta(\mathbf{x}_t, t)\end{aligned}$$

Imitation Learning as Conditional Generation

- Conditional sampling

Learn $p_\theta(x \mid c)$ to sample x given class labels

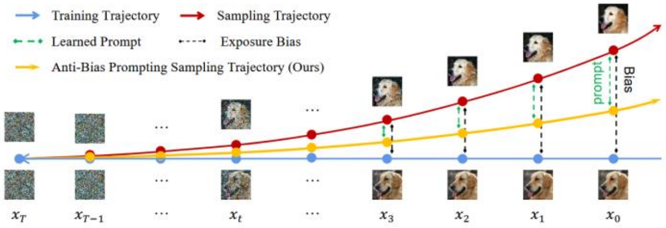
- Maximize the likelihood

$$\max_\theta \mathbb{E}_{(x_0,c)\sim\mathcal{D}}\big[\log p_\theta(x_0 \mid c)\big]$$

- Classifier-free guidance

$$p_\theta(x_0 \mid c)$$

- Exposure bias (diffusion models)



- Conditional sampling

Learn $\pi_\theta(a \mid s)$ to sample action given states

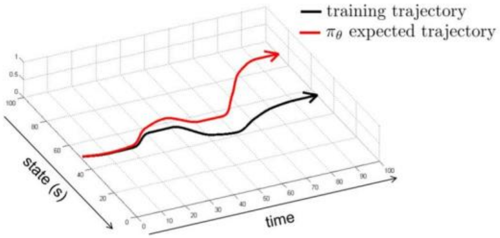
- Maximize the likelihood

$$\max_\theta \mathbb{E}_{(s,a)\sim\mathcal{D}}\big[\log \pi_\theta(a \mid s)\big]$$

- Goal-conditioned BC

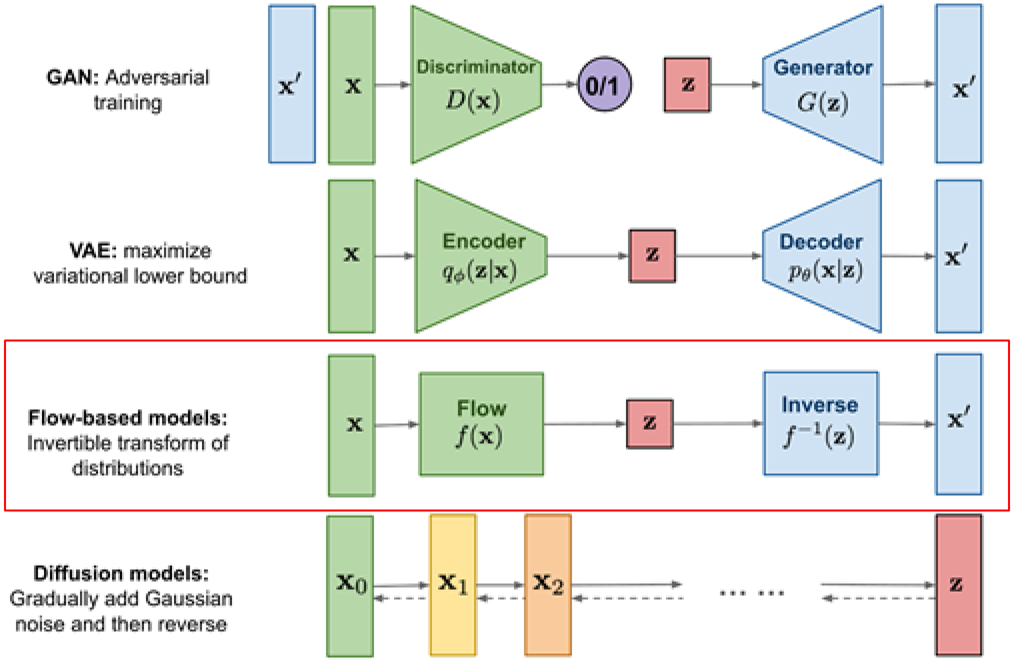
$$\pi_\theta(a \mid s, g)$$

- Distribution shift

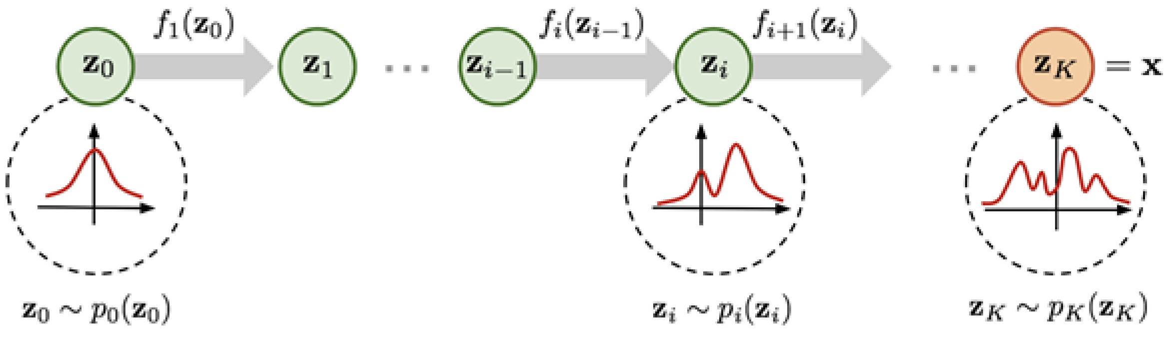


Generative Models

- ❑ Variational Autoencoder
- ❑ Diffusion Models
- ❑ **Flow-based Models**



Normalizing Flow



- Definition: $\mathbf{z}_{i-1} \sim \pi(\mathbf{z}_{i-1}), \mathbf{z}_i = f_i(\mathbf{z}_{i-1}), \mathbf{z}_{i-1} = f_i^{-1}(\mathbf{z}_i)$
- Compute the data distribution: $\mathbf{x} = \mathbf{z}_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{z}_0)$
- Learn by maximizing the log-likelihood
- How to compute the log-likelihood? $\log p(\mathbf{x}) = \log \pi_K(\mathbf{z}_K)$

Normalizing Flow

- Preliminary: given $z \sim \pi(z)$, construct new variable $x = f(z)$, f is invertible, then we have

$$\int p(x)dx = \int \pi(z)dz = 1 \qquad p(x) = \pi(z)\left|\frac{dz}{dx}\right| = \pi(f^{-1}(x))\left|\frac{df^{-1}}{dx}\right| = \pi(f^{-1}(x))|(f^{-1})'(x)|$$

- For multivariate case:

$$\mathbf{z} \sim \pi(\mathbf{z}), \mathbf{x} = f(\mathbf{z}), \mathbf{z} = f^{-1}(\mathbf{x})$$

$$p(\mathbf{x}) = \pi(\mathbf{z})\left|\det \frac{d\mathbf{z}}{d\mathbf{x}}\right| = \pi(f^{-1}(\mathbf{x}))\left|\det \frac{df^{-1}}{d\mathbf{x}}\right|$$

- Inverse function theorem: given $y = f(x) \text{ and } x = f^{-1}(y)$, then we have

$$\frac{df^{-1}(y)}{dy} = \frac{dx}{dy} = \left(\frac{dy}{dx}\right)^{-1} = \left(\frac{df(x)}{dx}\right)^{-1}$$

Normalizing Flow

- Given $\mathbf{z}_{i-1} \sim p_{i-1}(\mathbf{z}_{i-1}), \mathbf{z}_i = f_i(\mathbf{z}_{i-1}), \mathbf{z}_{i-1} = f_i^{-1}(\mathbf{z}_i)$
- Compute $\log p(\mathbf{x}) = \log \pi_K(\mathbf{z}_K)$
- From the change of variable theorem: $p_i(\mathbf{z}_i) = p_{i-1}(f_i^{-1}(\mathbf{z}_i)) \left|\det\left(\frac{df_i^{-1}}{d\mathbf{z}_i}\right)\right|$

$$= p_{i-1}(\mathbf{z}_{i-1}) \left|\det\left(\left(\frac{df_i}{d\mathbf{z}_{i-1}}\right)^{-1}\right)\right| \quad = p_{i-1}(\mathbf{z}_{i-1}) \left|\left(\det\left(\frac{df_i}{d\mathbf{z}_{i-1}}\right)\right)^{-1}\right| \quad = p_{i-1}(\mathbf{z}_{i-1}) \frac{1}{\left|\det\left(\frac{df_i}{d\mathbf{z}_{i-1}}\right)\right|}$$

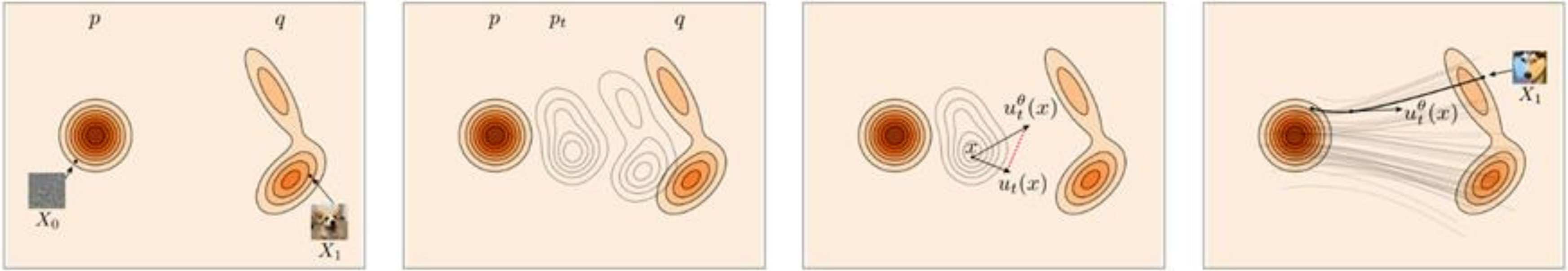
- We get $\log p_i(\mathbf{z}_i) = \log p_{i-1}(\mathbf{z}_{i-1}) - \log \left|\det\left(\frac{df_i}{d\mathbf{z}_{i-1}}\right)\right|$
- And $\log p(\mathbf{x}) = \log \pi_K(\mathbf{z}_K)$

$$= \log \pi_{K-1}(\mathbf{z}_{K-1}) - \log \left|\det \frac{df_K}{d\mathbf{z}_{K-1}}\right|$$

$$= \log \pi_{K-2}(\mathbf{z}_{K-2}) - \log \left|\det \frac{df_{K-1}}{d\mathbf{z}_{K-2}}\right| - \log \left|\det \frac{df_K}{d\mathbf{z}_{K-1}}\right| \quad = \log \pi_0(\mathbf{z}_0) - \sum_{i=1}^{K} \log \left|\det \frac{df_i}{d\mathbf{z}_{i-1}}\right|$$

- Requirements on f?
  - 1. Easily invertible. 2. Easy to compute jacobians

Flow-Matching Methods



(a) Data. **(b)** Path design. **(c)** Training. **(d)** Sampling.

- Training:
  - Build a probability path $(p_t)_{0\leq t\leq 1}$ from a **known** source distribution **p** to a **target** distribution **q**
  - regression on the **vector field** used to convert distributions along the prob path
- Sampling (from the target distribution):
  - Sample from the source distribution $X_0 \sim p$
  - Solve an ODE determined by the vector field to get $X_1 \sim q$

Flow-Matching Methods

- Vector field: $u : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, flow: $\psi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$
- ODE: $\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = u_t(\psi_t(x))$ where $\psi_t := \psi(t,x)$ and $\psi_0(x) = x$
- u generates the prob. path if $X_t := \psi_t(X_0) \sim p_t$ for $X_0 \sim p_0$
- **Learning objective:** learn a vector field that can generates the prob. path p_t
- A simple probability path? $X_t = tX_1 + (1-t)X_0 \sim p_t$
- **Flow matching loss:** $\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,X_t} \left\| u_t^\theta(X_t) - u_t(X_t) \right\|^2$, where $t \sim \mathcal{U}[0,1]$ and $X_t \sim p_t$
- What's the issue?
    - We cannot compute the target distribution p_1

Flow-Matching Methods

- Conditional random variables: $X_{t|1} = tx_1 + (1-t)X_0 \quad \sim \quad p_{t|1}(\cdot|x_1) = \mathcal{N}(\cdot \mid tx_1, (1-t)^2 I)$
- Solving for the cond. vector field: $\frac{\mathrm{d}}{\mathrm{d}t} X_{t|1} = u_t(X_{t|1}|x_1) \longrightarrow u_t(x|x_1) = \dfrac{x_1 - x}{1-t}$
- **Conditional flow matching loss:**

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,X_t,X_1} \| u_t^\theta(X_t) - u_t(X_t|X_1) \|^2, \text{ where } t \sim U[0,1],\ X_0 \sim p,\ X_1 \sim q$$

$$\downarrow$$

$$\mathcal{L}_{\mathrm{CFM}}^{\mathrm{OT,Gauss}}(\theta) = \mathbb{E}_{t,X_0,X_1} \| u_t^\theta(X_t) - (X_1 - X_0) \|^2, \text{ where } t \sim U[0,1],\ X_0 \sim \mathcal{N}(0,I),\ X_1 \sim q$$

References

- Weng, Lilian. (Jul 2021). What are diffusion models? Lil'Log. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.
- Lipman, Yaron, et al. "Flow matching guide and code." arXiv preprint arXiv:2412.06264 (2024).
- Jonathan Ho et al. "Denoising diffusion probabilistic models." arxiv Preprint arxiv:2006.11239 (2020).