上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Institute of Media,
Information, and Network

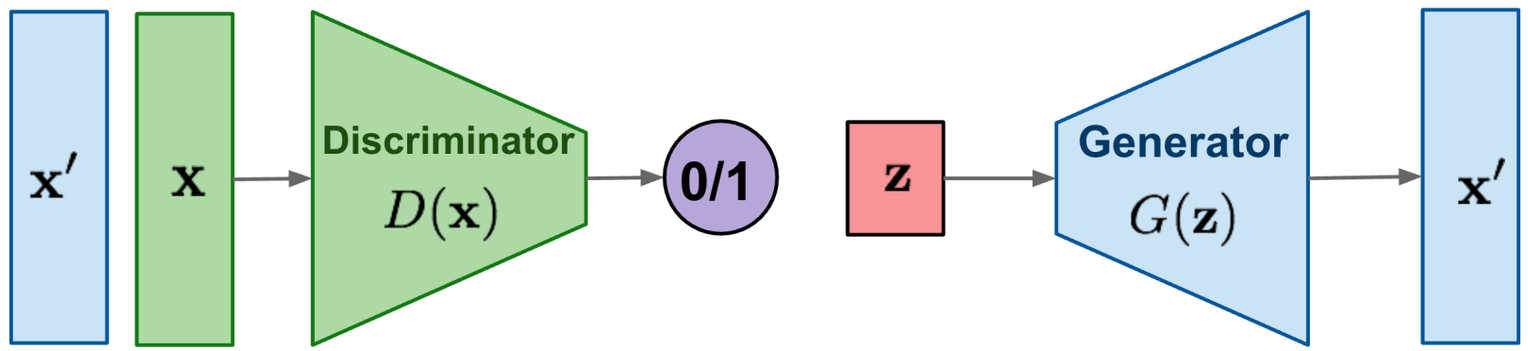
A Tutorial on Generative Models

Dept. of Computer Science and Engineering
Shanghai Jiao Tong University

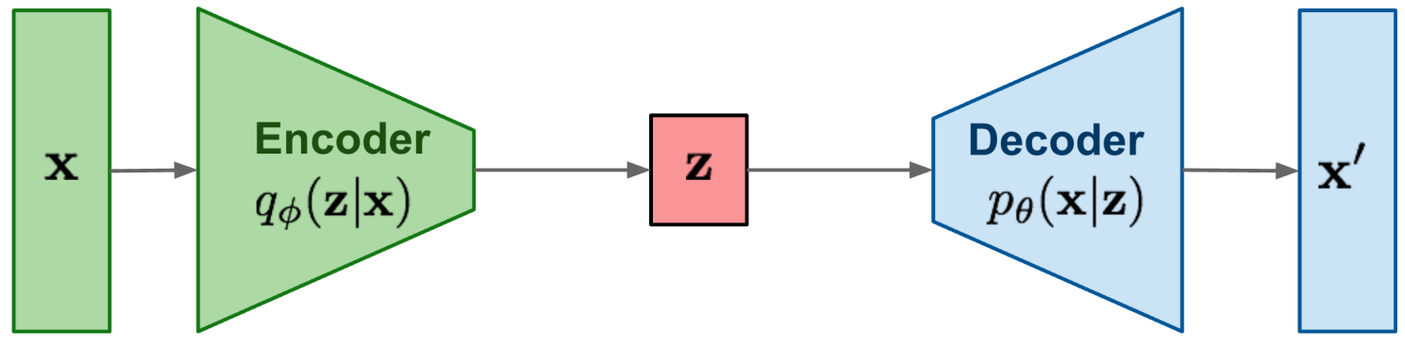
2024.06.07

Generative Models

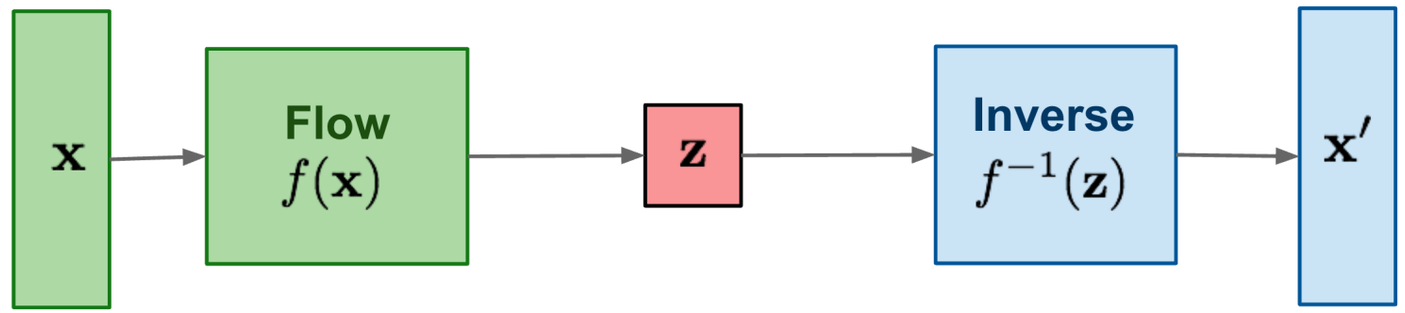
GAN: Adversarial training



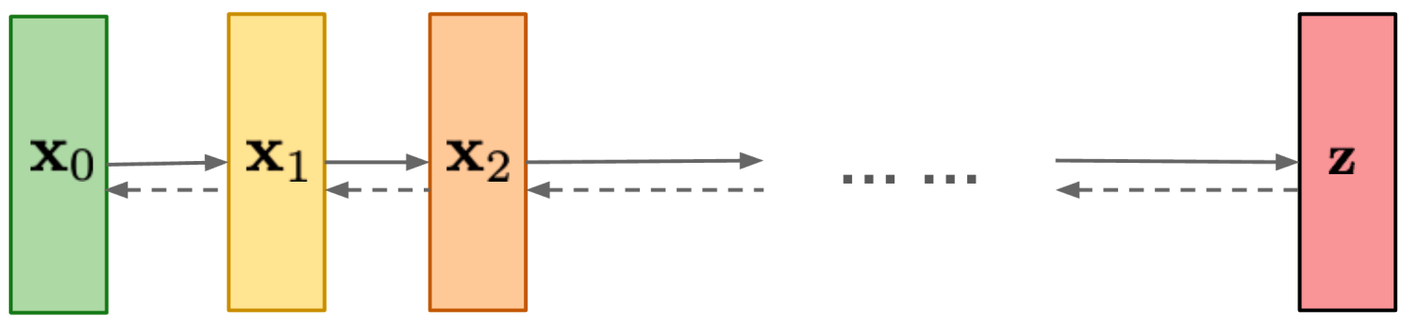
VAE: maximize variational lower bound



Flow-based models: Invertible transform of distributions



Diffusion models: Gradually add Gaussian noise and then reverse



Review of latest Score Based Generative Modeling papers:
https://scorebasedgenerativemodeling.github.io

Preliminary for Normalizing Flows

- Given a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, the **Jacobian matrix** of $\mathbf{f}$ is defined as:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

- The **determinant** of a $n \times n$ matrix $\mathbf{M}$ is:

$$\det M = \det \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \sum_{j_1 j_2 \dots j_n} (-1)^{\tau(j_1 j_2 \dots j_n)} a_{1j_1} a_{2j_2} \dots a_{nj_n}$$

If $det(M) = 0$, M is not invertible; otherwise, M is invertible.

$$det(AB) = det(A)det(B)$$

Preliminary for Normalizing Flows

- **Change of Variable Theorem**

A random variable z, its known pdf $z \sim \pi(z)$,

1-1 mapping function $x = f(z)$, f is invertible so $z = f^{-1}(x)$,

How to infer the unknown pdf $p(x)$?

Single variable case:

$\int p(x)dx = \int \pi(z)dz = 1$ **; Definition of probability distribution.**

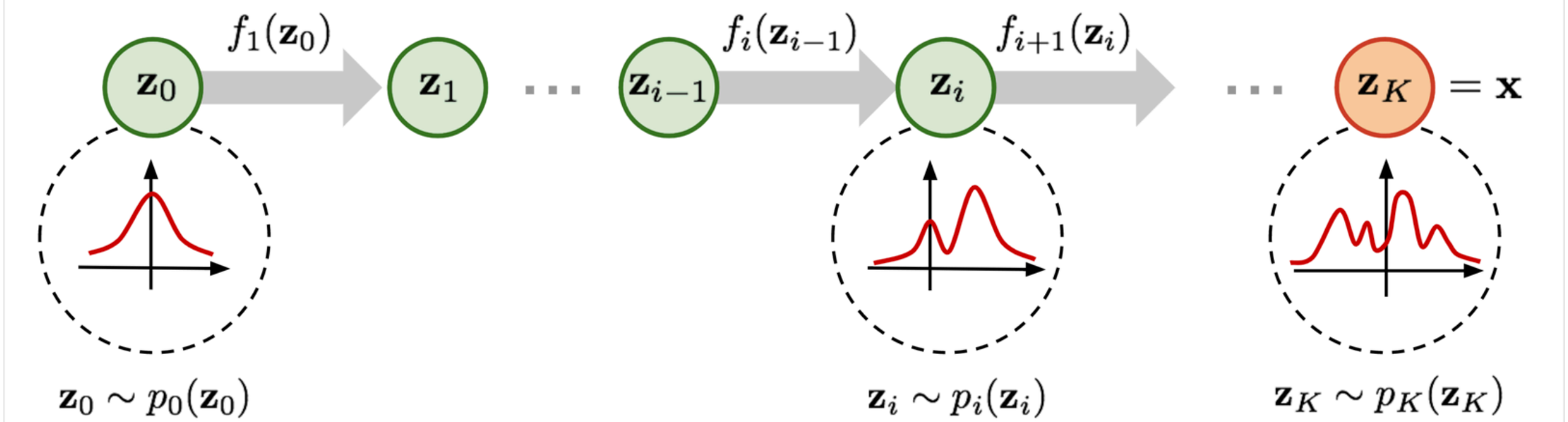
$$p(x) = \pi(z)\left|\frac{dz}{dx}\right| = \pi(f^{-1}(x))\left|\frac{df^{-1}}{dx}\right| = \pi(f^{-1}(x))\,|\,(f^{-1})'(x)\,|$$

Multivariable case:

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

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

Normalizing Flows



$$\mathbf{z}_{i-1} \sim p_{i-1}(\mathbf{z}_{i-1})$$

$$\mathbf{z}_i = f_i(\mathbf{z}_{i-1}), \text{ thus } \mathbf{z}_{i-1} = f_i^{-1}(\mathbf{z}_i)$$

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

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

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

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

If $y = f(x)$ and $x = f^{-1}(y)$, we have

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

⇧

; According to the inverse func theorem.

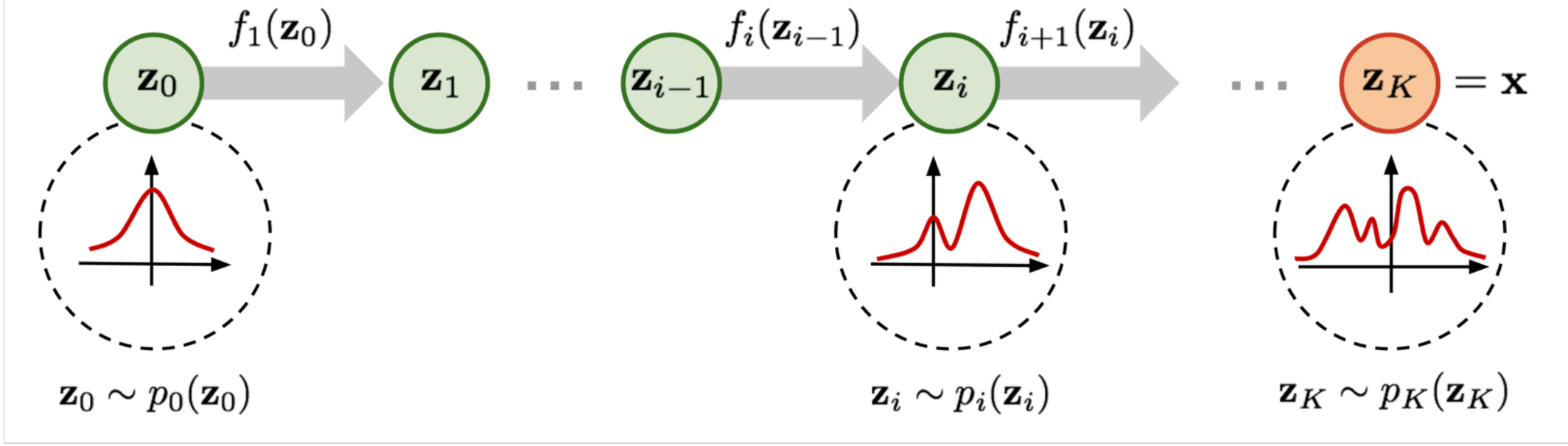
; According to a property of Jacobians of invertible func.

⇩

We have $\det(M^{-1}) = (\det(M))^{-1}$, because

$\det(M)\det(M^{-1}) = \det(M \cdot M^{-1}) = \det(I) = 1$

Normalizing Flows



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

The full chain:

$$\mathbf{x} = \mathbf{z}_K = f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z}_0)$$

$$\log p(\mathbf{x}) = \log p_K(\mathbf{z}_K) = \log p_{K-1}(\mathbf{z}_{K-1}) - \log \left| \det \frac{df_K}{d\mathbf{z}_{K-1}} \right|$$

$$= \log p_{K-2}(\mathbf{z}_{K-2}) - \log \left| \det \frac{df_{K-1}}{d\mathbf{z}_{K-2}} \right| - \log \left| \det \frac{df_K}{d\mathbf{z}_{K-1}} \right|$$

$$= \ldots$$

$$= \log p_0(\mathbf{z}_0) - \sum_{i=1}^{K} \log \left| \det \frac{df_i}{d\mathbf{z}_{i-1}} \right|$$

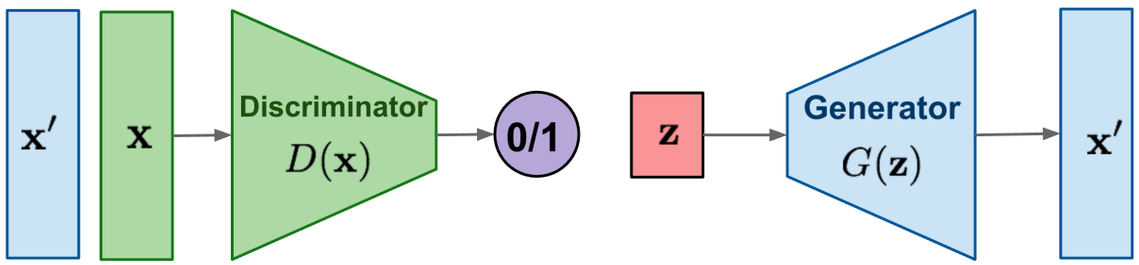
Each f_i should satisfy two properties:

- It is easily invertible.
- Its Jacobian determinant is easy to compute.

Loss: $L_D = -\frac{1}{|D|} \sum_{\mathbf{x} \in D} \log p(\mathbf{x})$

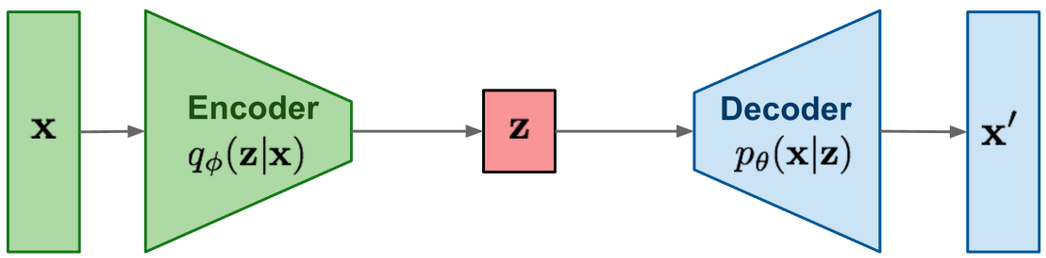
Why Diffusion Models?

GAN: Adversarial training



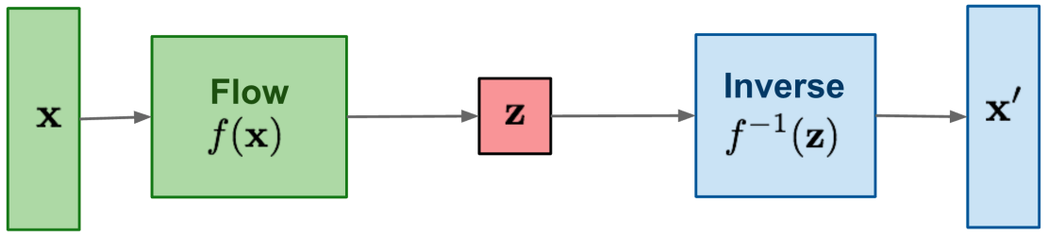
- Potentially unstable training
- Less diversity in generation due to adversarial training nature

VAE: maximize variational lower bound



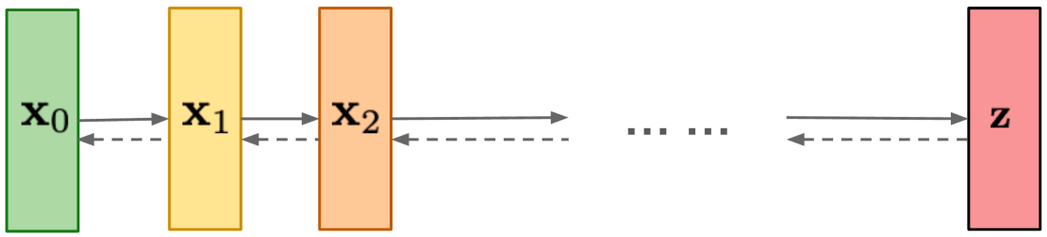
- Choice of the posterior
- Simultaneous optimization of encoder and decoder

Flow-based models: Invertible transform of distributions



- Specialized architectures to construct reversible transform

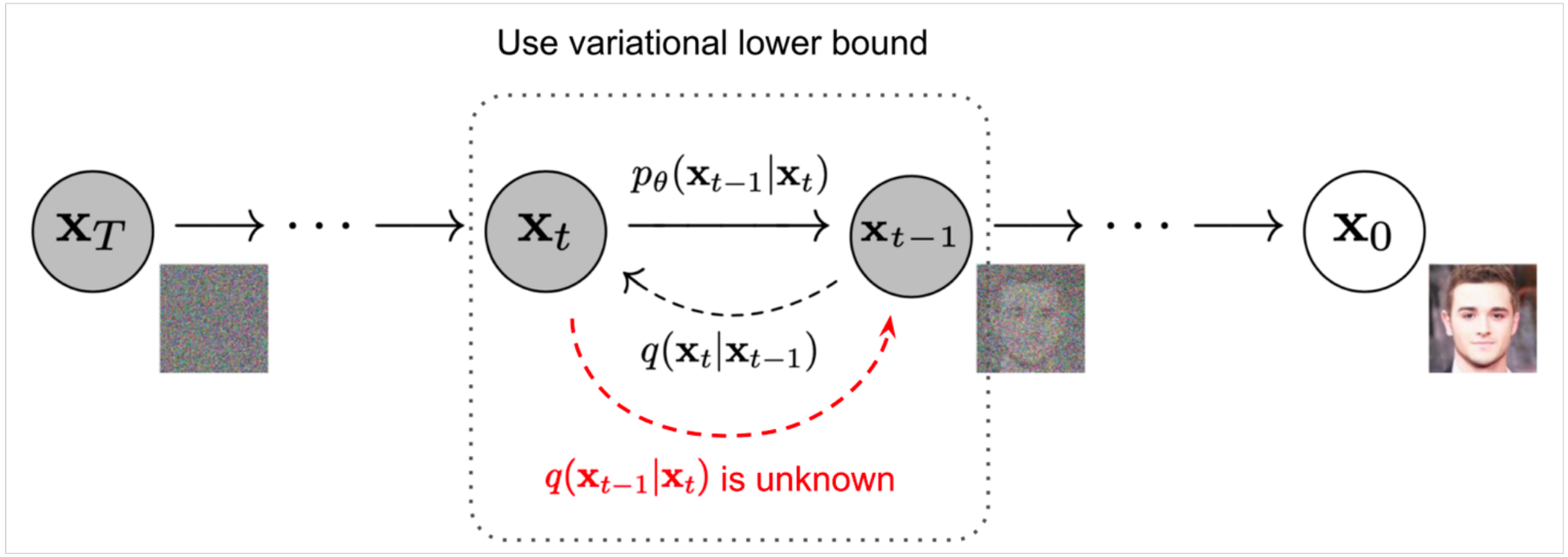
Diffusion models: Gradually add Gaussian noise and then reverse



- Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.
- Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution."Advances in neural information processing systems 32 (2019).
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Denoising Diffusion Probabilistic Models

Forward diffusion process:



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1} \quad \boldsymbol{\epsilon}_{t-1} \sim N(\mathbf{0}, \mathbf{I})$$

$\beta_t \in (0,1)$

Let $\alpha_t = 1 - \beta_t$, we have

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$

usually $\beta_1 < \beta_2 < \ldots < \beta_T, \alpha_1 > \alpha_2 > \ldots > \alpha_T$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

$$q(\mathbf{x}_2|\mathbf{x}_1)q(\mathbf{x}_1|\mathbf{x}_0) = q(\mathbf{x}_2|\mathbf{x}_1, \mathbf{x}_0)q(\mathbf{x}_1|\mathbf{x}_0)$$

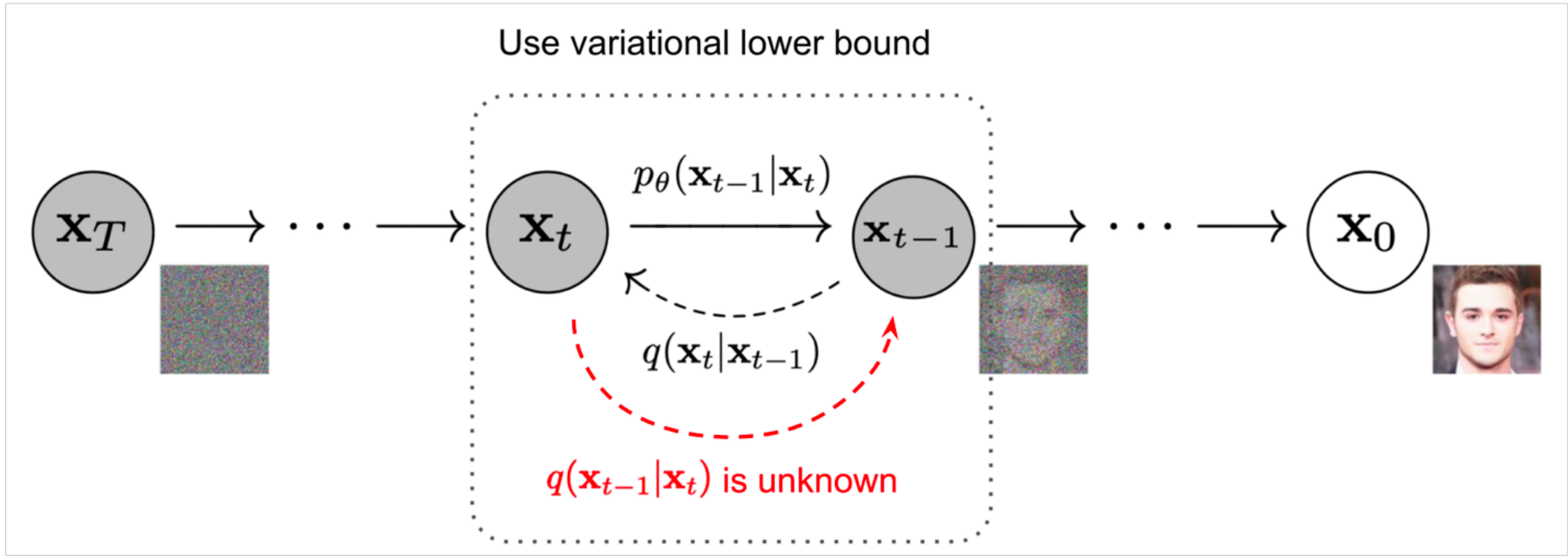
$$= q(\mathbf{x}_2|\mathbf{x}_1, \mathbf{x}_0)\frac{q(\mathbf{x}_1, \mathbf{x}_0)}{q(\mathbf{x}_0)}$$

$$= \frac{q(\mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0)}{q(\mathbf{x}_0)} = q(\mathbf{x}_{1:2}|\mathbf{x}_0)$$

☐ Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Denoising Diffusion Probabilistic Models

Forward diffusion process:



$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$
$$= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$
$$= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2}$$
$$= \dots$$
$$= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

*Merge two Gaussians $N(0,\sigma_1^2\mathbf{I})$ and $N(0,\sigma_2^2\mathbf{I})$, new distribution is $N(0,(\sigma_1^2+\sigma_2^2)\mathbf{I})$

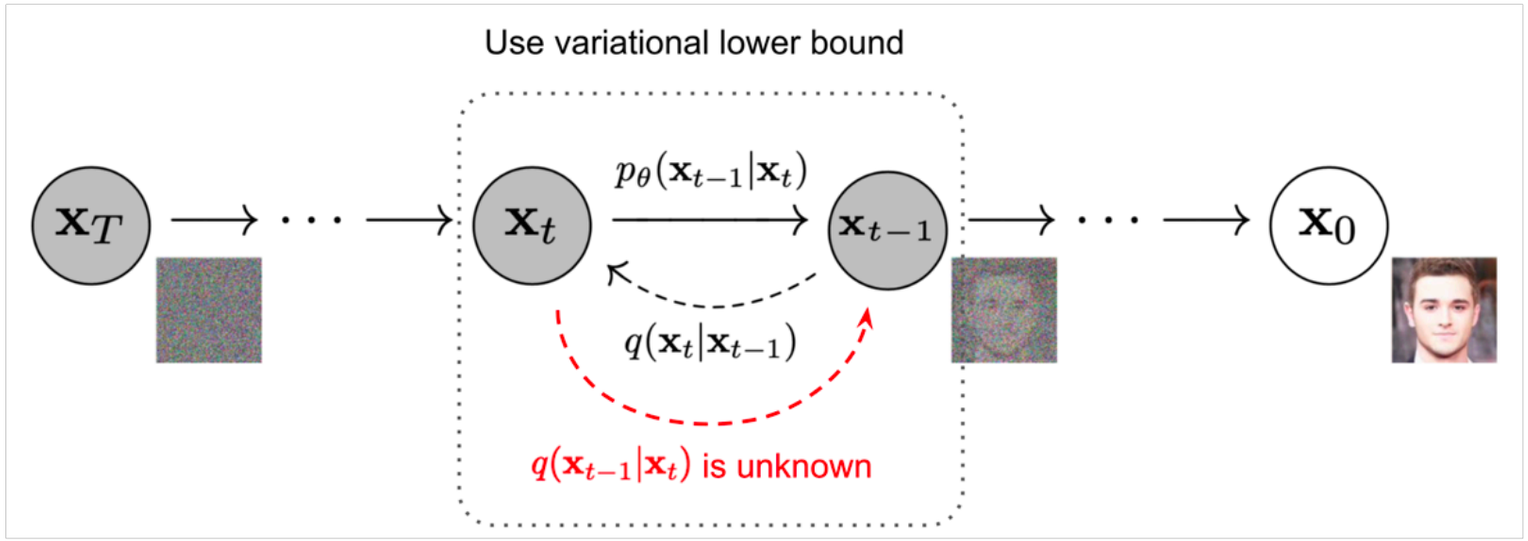
$$* \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \quad \bar{\alpha}_1 > \dots > \bar{\alpha}_{T-1} > \bar{\alpha}_T$$

$q(\mathbf{x}_t|\mathbf{x}_0) = N(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$; when $t \to \infty$, $\mathbf{x}_t \to$ isotropic Gaussian distribution .

☐ Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Denoising Diffusion Probabilistic Models

Reverse diffusion process:



$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

Unknown $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$

If β_t is small enough, $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ will also be Gaussian.

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) = q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) \exp\left(\log q_{t-1}\left(\mathbf{x}_{t-1}\right) - \log q_t\left(\mathbf{x}_t\right)\right)$$

*Because $q(A \mid B) = q(B \mid A)q(A)/q(B)$

$$\approx q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) \exp\left(\log q_t\left(\mathbf{x}_{t-1}\right) - \log q_t\left(\mathbf{x}_t\right)\right)$$

*Because $q_t(\cdot) \approx q_{t-1}(\cdot)$ as β_t is small enough

$$\approx q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) \exp\left(\left(\mathbf{x}_{t-1} - \mathbf{x}_t\right) \nabla_x \log q_t\left(\mathbf{x}_t\right)\right)$$

*The first-order Taylor approximation

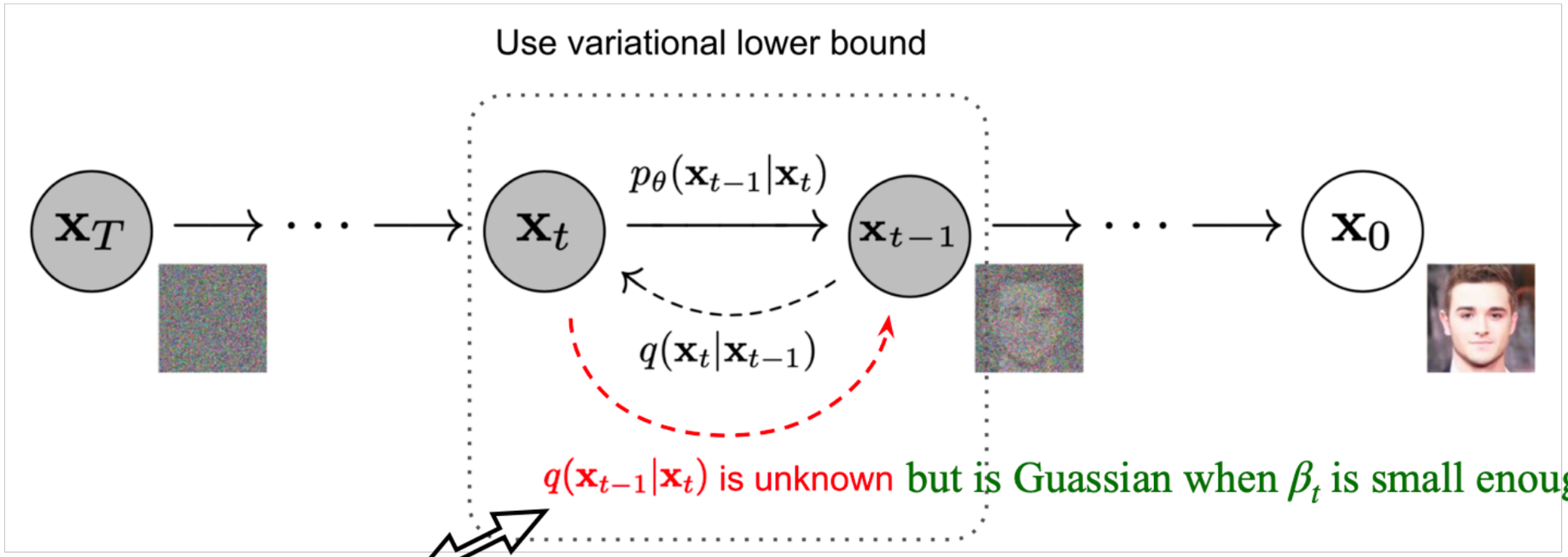
$$= \frac{1}{\sqrt{2\pi}\beta_t} \exp\left(\frac{\left\| \mathbf{x}_t - \sqrt{1-\beta_t}\mathbf{x}_{t-1} \right\|_2^2}{2\beta_t^2} + \left(\mathbf{x}_{t-1} - \mathbf{x}_t\right) \nabla_{\mathbf{x}} \log q_t\left(\mathbf{x}_t\right)\right)$$

* $\nabla_{\mathbf{x}} \log q_t\left(\mathbf{x}_t\right)$: score function

- ☐ Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.
- ☐ Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." ICLR (2021, Outstanding Paper Award).
- ☐ Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." Advances in neural information processing systems 32 (2019).

Denoising Diffusion Probabilistic Models

Reverse diffusion process:



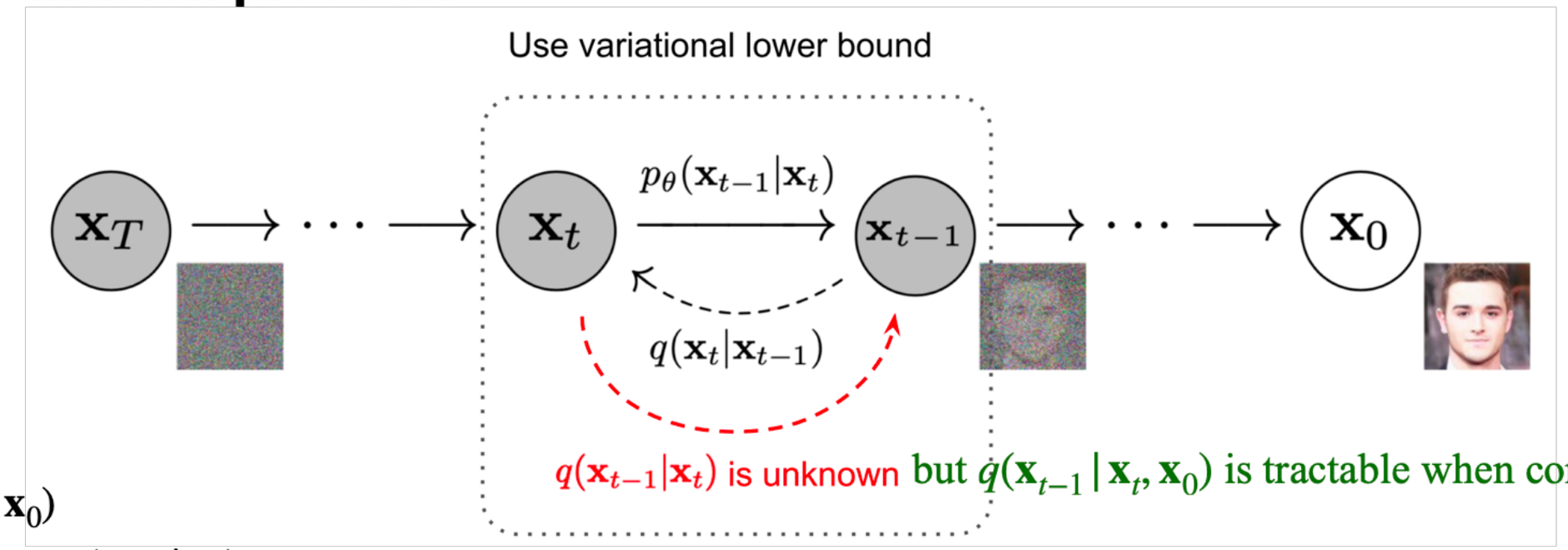
$$p_\theta(\mathbf{x}_{t-1}\,|\,\mathbf{x}_t) = N(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}) &= p_\theta(\mathbf{x}_T\,|\,\mathbf{x}_{T-1}) \ldots p_\theta(\mathbf{x}_2\,|\,\mathbf{x}_1) p_\theta(\mathbf{x}_1\,|\,\mathbf{x}_0) p_\theta(\mathbf{x}_0) \\ &= p_\theta(\mathbf{x}_T\,|\,\mathbf{x}_{T-1}) \ldots p_\theta(\mathbf{x}_2\,|\,\mathbf{x}_1) p_\theta(\mathbf{x}_1) p_\theta(\mathbf{x}_0\,|\,\mathbf{x}_1) \\ &= p_\theta(\mathbf{x}_T\,|\,\mathbf{x}_{T-1}) \ldots p_\theta(\mathbf{x}_2) p_\theta(\mathbf{x}_1\,|\,\mathbf{x}_2) p_\theta(\mathbf{x}_0\,|\,\mathbf{x}_1) \\ &= p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}\,|\,\mathbf{x}_t) \end{aligned}$$

☐ Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Denoising Diffusion Probabilistic Models

Reverse diffusion process:



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

$$= q(\mathbf{x}_t\,|\,\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}\,|\,\mathbf{x}_0)}{q(\mathbf{x}_t\,|\,\mathbf{x}_0)} \qquad *\text{Because } q(A\,|\,B) = q(B\,|\,A)q(A)/q(B)$$

$$\propto \exp\Big(-\frac{1}{2}\Big(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\Big)\Big) \qquad *\underline{\text{Because}}\ q(\mathbf{x}_t\,|\,\mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$
$$\underline{\text{and}}\ q(\mathbf{x}_t\,|\,\mathbf{x}_0) = N(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

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

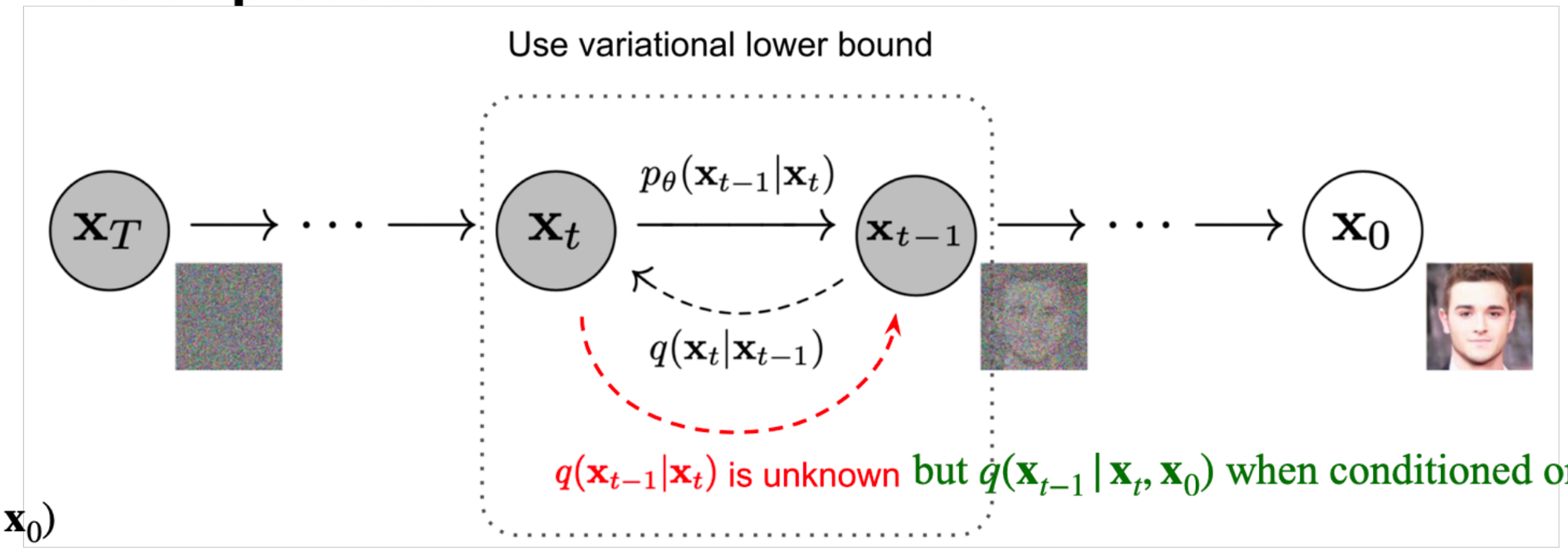
$$= \exp\Big(-\frac{1}{2}\Big(\Big(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\Big)\mathbf{x}_{t-1}^2 - \Big(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\Big)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\Big)\Big) \qquad * \ A\mathbf{x}^2 - B\mathbf{x} + \frac{B^2}{4A} = A(\mathbf{x} - \frac{B}{2A})^2$$

$$= \exp\Big(-\frac{\big(\mathbf{x}_{t-1} - \hat{\boldsymbol{\mu}}_{t-1}(\mathbf{x}_t, \mathbf{x}_0)\big)^2}{2\hat{\boldsymbol{\sigma}}^2}\Big) \qquad \text{Gaussian distribution } \frac{(\mathbf{x}-\mu)^2}{\sigma^2}$$

☐ Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Denoising Diffusion Probabilistic Models

Reverse diffusion process:



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

$$\hat{\boldsymbol{\mu}}_{t-1}(\mathbf{x}_t,\mathbf{x}_0) = (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)/(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})$$

$$= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t$$

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

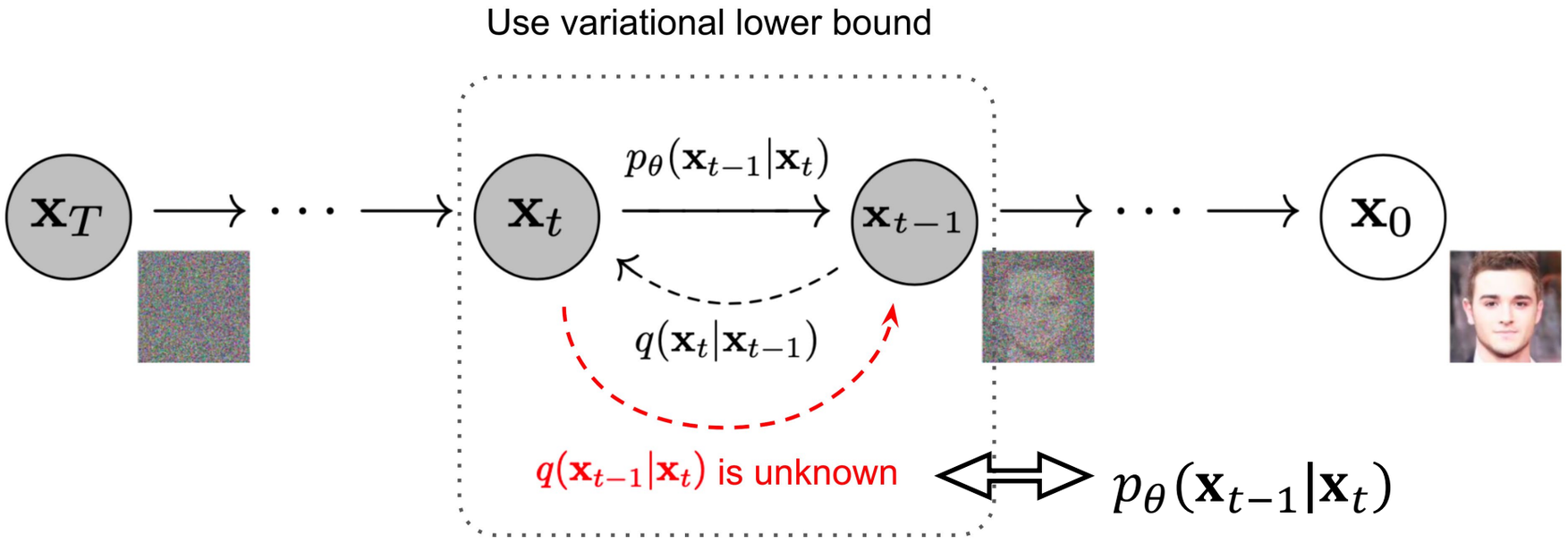
$$= \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\Big)$$

$$\hat{\sigma}_{t-1}^2 = 1/(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t$$

*Because $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon})$

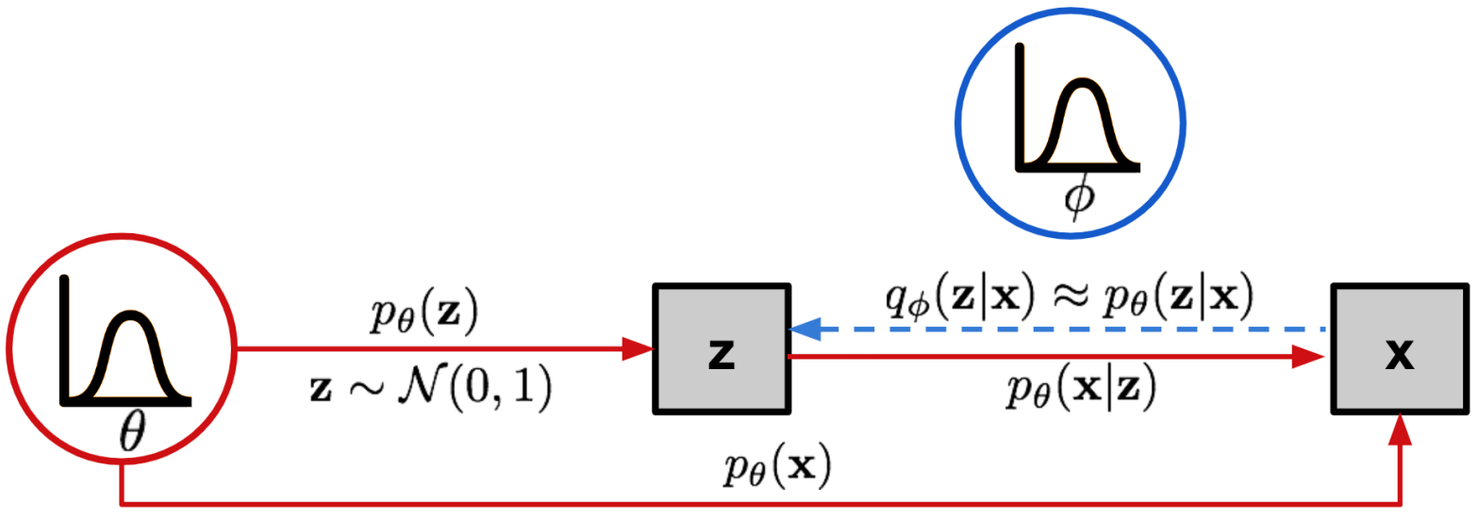
Denoising Diffusion Probabilistic Models

DDPM:



$$L_{DDPM} = -\log p_\theta(\mathbf{x}_0) + D_{KL}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0))$$

VAE:



$$L_{VAE} = -\log p_\theta(\mathbf{x}) + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x}))$$

Denoising Diffusion Probabilistic Models

$$
\begin{aligned}
L_{DDPM} &= -\log p_\theta(\mathbf{x}_0) + D_{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] = \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_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{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}
$$

*Because $D_{KL}(p\|q) = \mathbb{E}_p\left[\log \frac{p(x)}{q(x)}\right]$ and $p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0) = p_\theta(\mathbf{x}_{1:T},\mathbf{x}_0)/p_\theta(\mathbf{x}_0)$

*Slide 14 and slide 17

*Slide 18

Denoising Diffusion Probabilistic Models

$$L_{DDPM} = \mathbb{E}_q[\underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

Slide 15: $q(\mathbf{x}_t|\mathbf{x}_0) = N(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$

$\mathbf{x}_T$ is a Gaussian noise.

Slide 19: $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = N(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right), \hat{\sigma}_{t-1}^2 I)$

Slide 17: $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = N(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$

For $N_0 = N(\mu_0, \Sigma_0)$ and $N_1 = N(\mu_1, \Sigma_1)$, the KL divergence is

$$D_{KL}\left(N_0 \| N_1\right) = \frac{1}{2}\left(\operatorname{tr}\left(\Sigma_1^{-1}\Sigma_0\right) - k + \left(\mu_1 - \mu_0\right)^\top \Sigma_1^{-1}\left(\mu_1 - \mu_0\right) + \ln\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right)$$

Assume that $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)$ and $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \hat{\sigma}_{t-1}^2 I$

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

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

- Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Denoising Diffusion Probabilistic Models

$$L_{DDPM} = \mathbb{E}_q[\underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2\right] \Leftrightarrow \mathbb{E}_q[-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] = \mathbb{E}_q[\log \frac{q(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}] = D_{KL}(q(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_0)|p_\theta(\mathbf{x}_0|\mathbf{x}_1))$$

Denoising diffusion probabilistic models (2020)

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

Why $std = 0$ when $t = 1$: $D_{KL}(q(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_0)|p_\theta(\mathbf{x}_0|\mathbf{x}_1))$

$N(\mathbf{x}_0; \mathbf{x}_0, 0)$

$MSE(\mathbf{x}_0, \mathbf{x}_\theta(\mathbf{x}_1))$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = N(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) \text{ and } \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \hat{\sigma}_{t-1}^2 I$$

$$\hat{\sigma}_{t-1} = 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t \qquad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

☐ Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.

Score-based Diffusion Models

- Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.
- Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." Advances in neural information processing systems 32 (2019).
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851.
- Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." ICLR (2021, Outstanding Paper Award).

DDPM

Score-based Diffusion Models

True samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \sim p(\mathbf{x})$ Unknown distribution

Let $f_\theta(\mathbf{x}) \in \mathbb{R}$ be a real-valued function parameterized by a learnable parameter θ, we define $p_\theta(\mathbf{x}) = \dfrac{e^{-f_\theta(\mathbf{x})}}{Z_\theta}$

where Z_θ is a normalizing constant only dependent on θ, such that $\int p_\theta(\mathbf{x}) d\mathbf{x} = 1$

Max the log-likelihood of the data: $\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i)$ **BUT** Z_θ is intractable

Score function: $\mathbf{s}_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = -\nabla_\mathbf{x} f_\theta(\mathbf{x}) - \underbrace{\nabla_\mathbf{x} \log Z_\theta}_{=0} = -\nabla_\mathbf{x} f_\theta(\mathbf{x})$.

⇕ **Optimization**

Fisher divergence: $\mathbb{E}_{p(\mathbf{x})}[\|\nabla_\mathbf{x} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$

Score Matching

True samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

Score-based Diffusion Models

Score function: $\mathbf{s}_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z_\theta}_{=0} = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x})\,.$

⇕ **Optimization**

Fisher divergence: $\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$

Score Matching

True samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

☐ A. Hyvarinen. "Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research", Vol 6(Apr), pp. 695--709. 2005. ***(Score Matching)***

☐ P. Vincent. "A connection between score matching and denoising autoencoders". Neural computation, Vol 23(7), pp. 1661--1674. MIT Press. 2011. ***(Denoising Score Matching)***

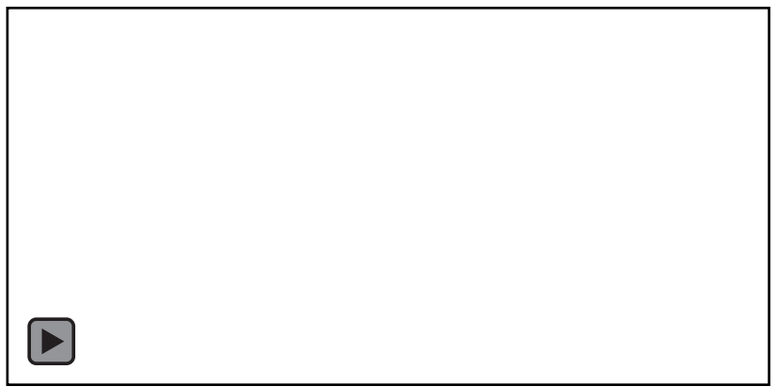
☐ Y. Song, S. Garg, J. Shi, S. Ermon. "Sliced score matching: A scalable approach to density and score estimation". Uncertainty in Artificial Intelligence, pp. 574--584. 2020. ***(Sliced Score Matching)***

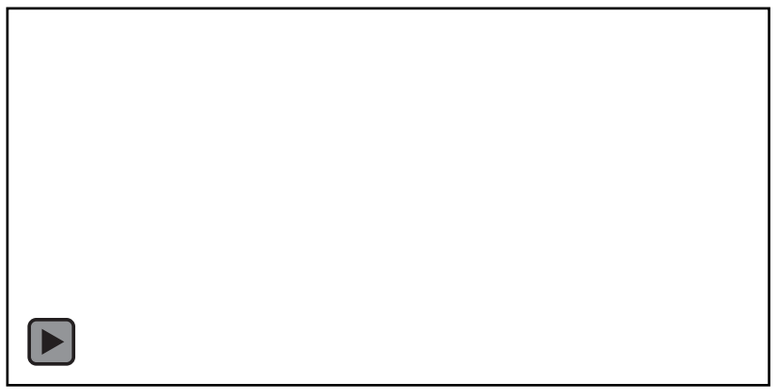
Once trained a $s_\theta(\mathbf{x})$, use **Langevin dynamics** to draw samples from it

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\ \mathbf{z}_i, \quad i = 0,1,\cdots,K,$$

where $\mathbf{z}_i \sim N(0,I)$, $\mathbf{x}_0$ is obtained from an arbitrary distribution.

When $\epsilon \to 0$ and $K \to \infty$, $\mathbf{x}_K$ converges to a sample of $p(\mathbf{x})$.

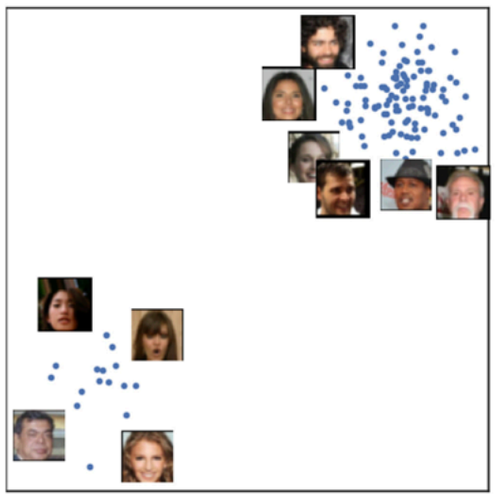




☐ G. Parisi. "Correlation functions and computer simulations". Nuclear Physics B, Vol 180(3), pp. 378--384. Elsevier. 1981.

☐ U. Grenander, M.I. Miller. "Representations of knowledge in complex systems". Journal of the Royal Statistical Society: Series B (Methodological), Vol 56(4), pp. 549--581. Wiley Online Library. 1994.

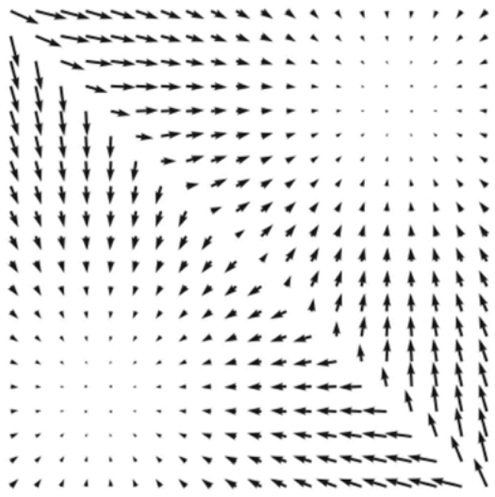
Score-based Diffusion Models



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

score matching



Scores

$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Langevin dynamics

New samples

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\ \mathbf{z}_i, \quad i = 0,1,\cdots,K,$$

Forward diffusion:

$$p_{\sigma_i}(\mathbf{x}_i \mid \mathbf{x}_0) = N(\mathbf{x}_i; \mathbf{x}_0, \sigma_i^2 \mathbf{I})$$

$$p(\mathbf{x}_i \mid \mathbf{x}_{i-1}) = N\left(\mathbf{x}_{i-1}; (\sigma_i^2 - \sigma_{i-1}^2)\,\mathbf{I}\right)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\,\mathbf{z}_{i-1}, \quad i = 1,\ldots,N$$

Reverse diffusion:

$$\mathbf{x}_i^m = \mathbf{x}_i^{m-1} + \epsilon_i s_\theta\left(\mathbf{x}_i^{m-1}, \sigma_i\right) + \sqrt{2\epsilon_i}\,\mathbf{z}_i^m, \quad m = 1,\ldots,M$$

Score Matching:

$$L = \sum_{i=1}^{N} \sigma_i^2 \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\mathbb{E}_{p_{\sigma_i}(\tilde{\mathbf{x}}|\mathbf{x})}\left[\left\| s_\theta\left(\tilde{\mathbf{x}}, \sigma_i\right) - \nabla_{\tilde{\mathbf{x}}} \log p_{\sigma_i}(\tilde{\mathbf{x}} \mid \mathbf{x}) \right\|^2\right]\right]$$

DDPM Forward diffusion:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = N(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$

DDPM Reverse diffusion: Slide 16

DDPM ELBO:

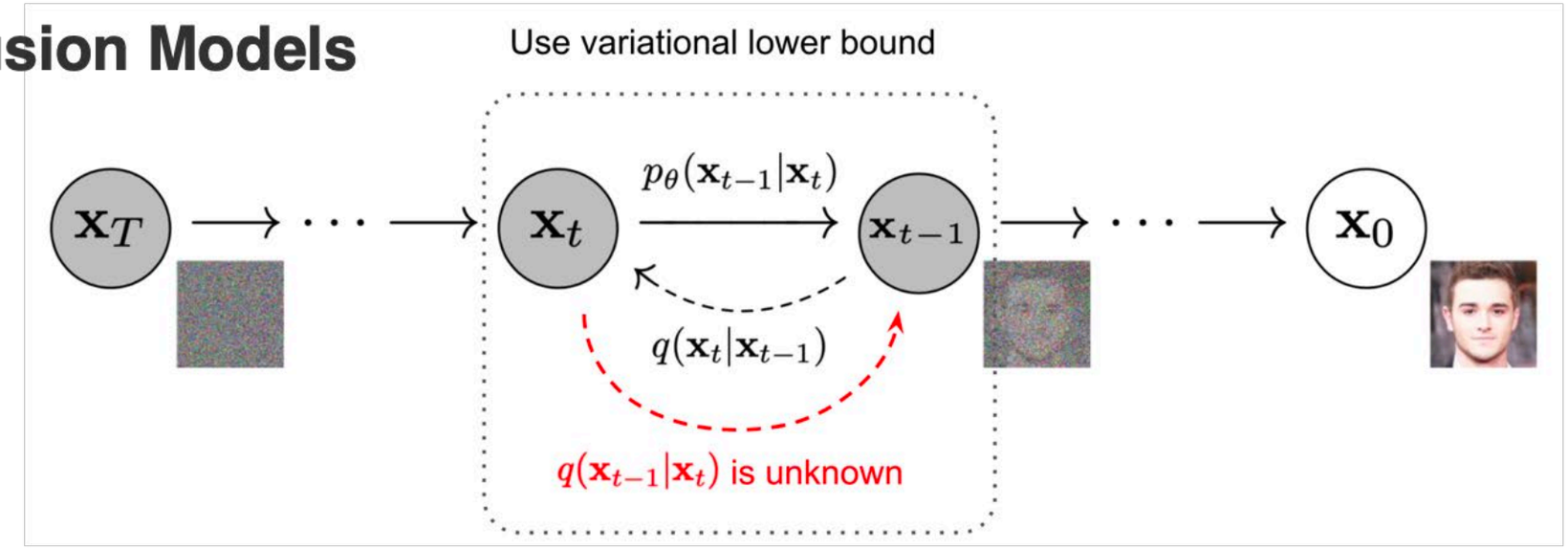
$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2\right]$$

☐ Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." Advances in neural information processing systems 32 (2019).

Continuous Diffusion Models

DDPM, Score-based Diffusion Models

discrete



Stochastic Differential Equations

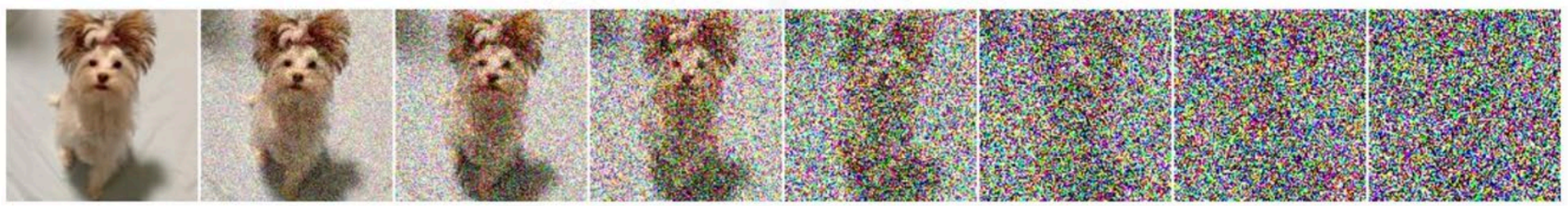
Continuous

$d\mathbf{w}$: white noise particle

Forward SDE (data → noise)

$\mathbf{x}(0)$ — $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ → $\mathbf{x}(T)$

Drift coefficient — Diffusion coefficient



score function

$\mathbf{x}(0)$ ← $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ — $\mathbf{x}(T)$

Reverse SDE (noise → data)

Any SDE has a corresponding reverse SDE

- Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." ICLR (2021, Outstanding Paper Award).
- B.D. Anderson. "Reverse-time diffusion equation models." Stochastic Processes and their Applications, Vol 12(3), pp. 313--326. Elsevier. 1982.

Continuous Diffusion Models

DDPM, Score-based Diffusion Models (Discrete)

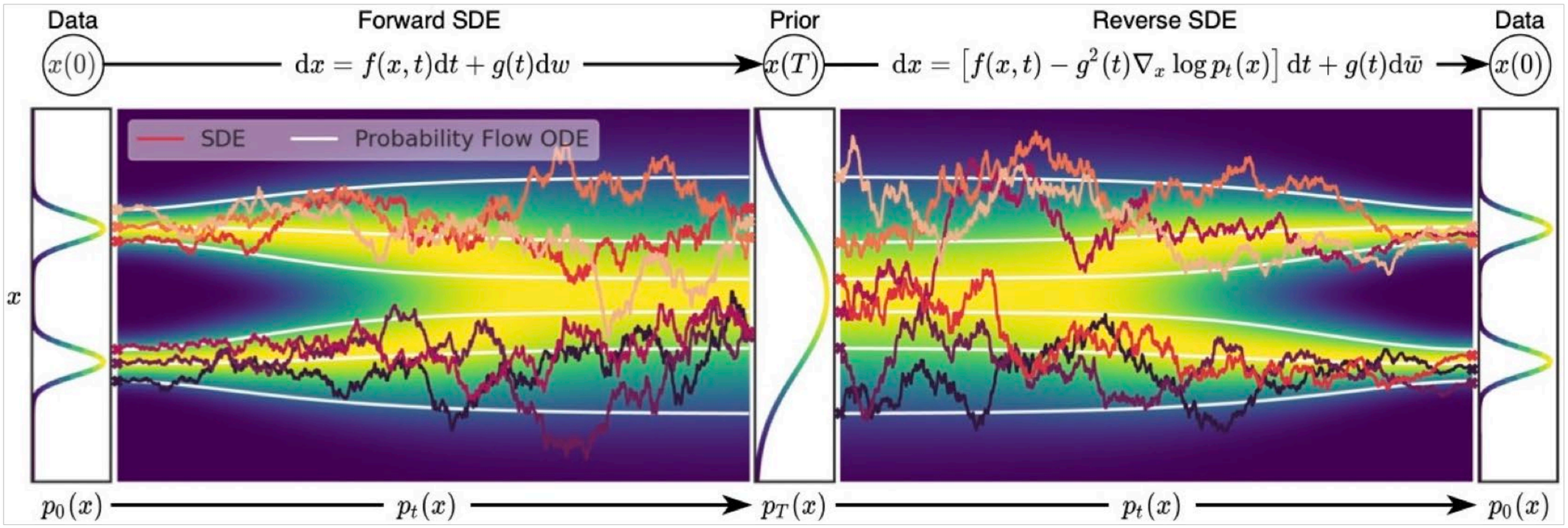
⇧ Comprise

Stochastic Differential Equations (Continuous)

any SDE has a corresponding ODE ⇩

⇩ How to solve the reverse SDE

Predictor-Corrector samplers



Solved with numerical ODE solvers

Forward ODE:

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g^2(t)\, \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t$$

***When $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is replaced by its approximation $s_\theta(\mathbf{x}, t)$, the ODE becomes a special case of a neural ODE.**

☐ Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." ICLR (2021, Outstanding Paper Award).
☐ T.Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud. “Neural Ordinary Differential Equations.” NeurIPS 2018
☐ Karras, Tero, et al. “Elucidating the Design Space of Diffusion-Based Generative Models.” NeurIPS 2022.

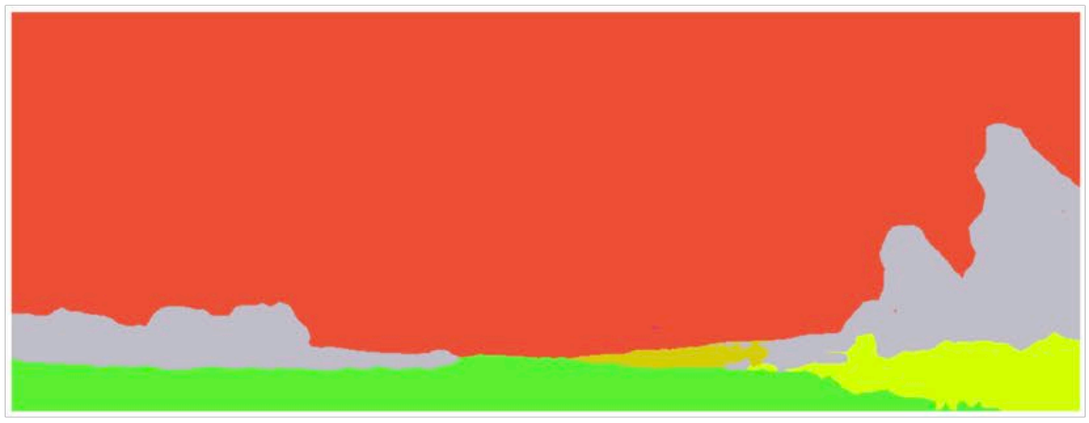
Diffusion Models: Research Interests

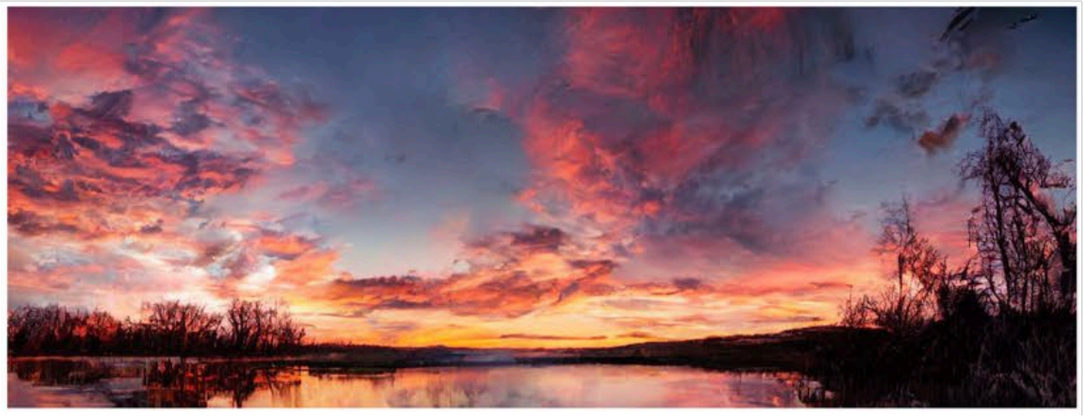
Speed up:

- ☐ Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in Neural Information Processing Systems 33 (2020): 6840-6851. ***(T=1000)***
- ☐ Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models." International Conference on Learning Representations 2021. ***(T=100)***
- ☐ Bao, Fan, et al. "Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models." International Conference on Learning Representations 2022 (Outstanding Paper Award). ***(T=40~60)***
- ☐ Lu, Cheng, et al. "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps." Advances in Neural Information Processing Systems 2022. ***(T=10)***

Vision tasks (Image synthesis):

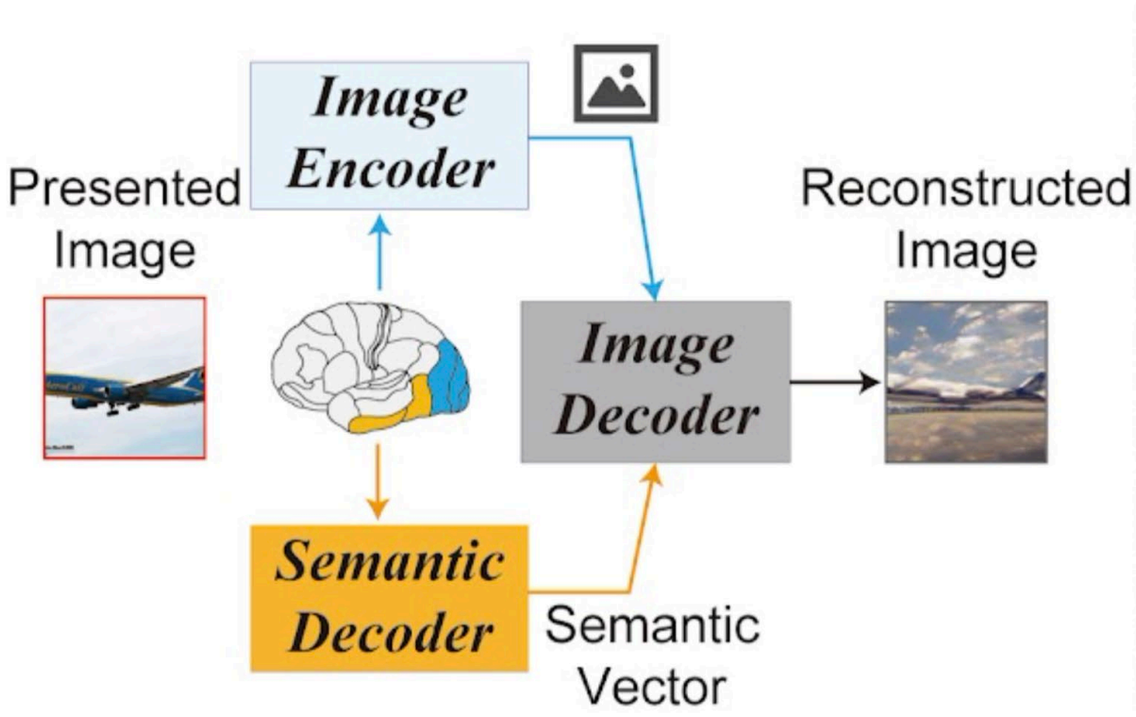
- ☐ Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.



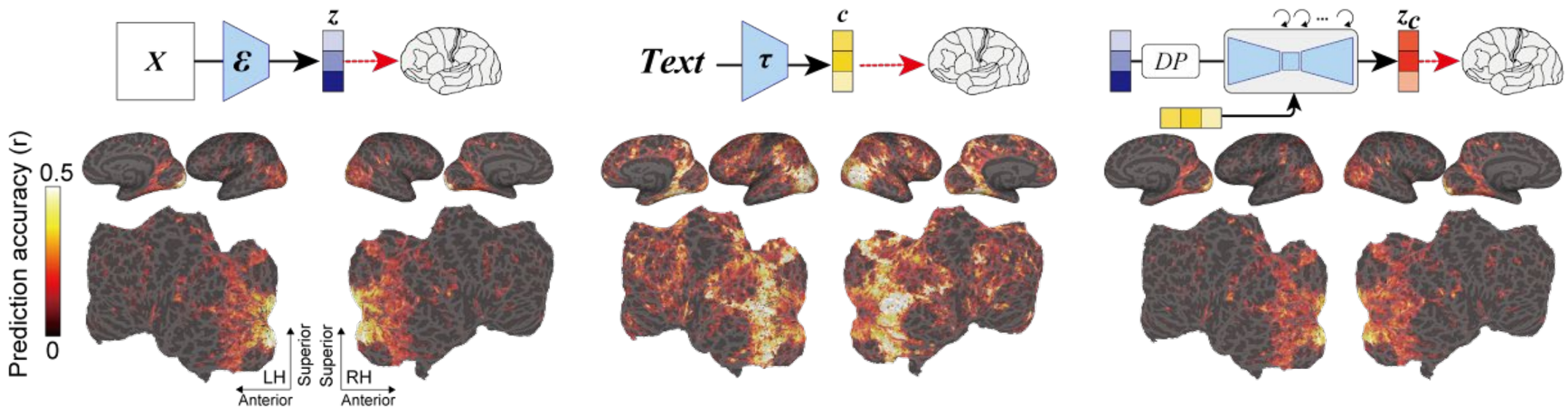



Diffusion Models: Research Interests

Human brain activity:



functional magnetic resonance imaging (fMRI)



- Takagi, Yu, and Shinji Nishimoto. “High-resolution image reconstruction with latent diffusion models from human brain activity.” CVPR 2023.

Diffusion Models: Research Interests

Classification and regression:

☐ Han, Xizewen, Huangjie Zheng, and Mingyuan Zhou. "CARD: Classification and Regression Diffusion Models." Advances in Neural Information Processing Systems 2022.

Diffusion with non-Gaussian:

☐ Song, Jiaming, et al. "Pseudoinverse-Guided Diffusion Models for Inverse Problems." *International Conference on Learning Representations*. 2023.

Diffusion with discrete state:

☐ Campbell, Andrew, et al. "A Continuous Time Framework for Discrete Denoising Models." Advances in Neural Information Processing Systems 2022.

Combinatorial problems:

☐ Graikos, Alexandros, et al. "Diffusion Models as Plug-and-Play Priors." *Advances in Neural Information Processing Systems 2022*.

Inverse problems:

☐ Kawar, Bahjat, et al. "Denoising Diffusion Restoration Models." Advances in Neural Information Processing Systems 2022.

☐ Chung, Hyungjin, Byeongsu Sim, and Jong Chul Ye. "Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

☐ Song, Yang, et al. "Solving Inverse Problems in Medical Imaging with Score-Based Generative Models." International Conference on Learning Representations 2021.

.....

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Institute of Media,
Information, and Network

Q & A

Thanks !