Fundamentals of Generative Diffusion Models

Form DDPM to Score-based Models

Fancheng Li （李凡成）

School of Physics and Technology

Wuhan University

Wuhan, China

武漢大學

Outline

- 01 Core Concept of Bayesian Statistics
- 02 A Unified Perspective of Diffusion Models
- 03 Energy-based Models and Guidance
- 04 Stochastic Process and Diffusion Models

Outline

- **01 Core Concept of Bayesian Statistics**
- 02 A Unified Perspective of Diffusion Models
- 03 Energy-based Models and Guidance
- 04 Stochastic Process and Diffusion Models

Bayes' Rule and Its Validity

Bayes formula: $p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$

before the fact

$p(\theta)$

prior probability

$p(x)$

observations (evidence)

$p(x|\theta)$

update our belief

after the fact

$p(\theta|x)$

posterior probability

Our purpose is to learn the posterior probability and likelihood

Cromwell's Rule

"I beseech you,
in the bowels of Christ,
think it possible that you may be mistaken."

I think the moon might be made of cheese

Bernstein-von Mises Theorem

For some case,when n is enough large:

$$||P(\theta|x_1,\ldots x_n) - \mathcal{N}(\hat{\theta}_n, n^{-1}\mathcal{I}(\theta_0)^{-1})||_{\mathrm{TV}} \xrightarrow{P_{\theta_0}} = 0$$

They are the starting point and the end point of Bayesian inference

https://en.wikipedia.org/wiki/Bayesian_statistics

Tweedie's Formula and Evidence Lower Bound

From x to estimate parameters by MSE

$$L = \mathbb{E}[(\hat{\theta}(x) - \theta)^2]$$

from the assumption

$$p(x|\theta) = \mathcal{N}(\theta, \sigma^2)$$

$$p(x) = \int_{-\infty}^{\infty} p(x|\theta)p(\theta)\mathrm{d}\theta$$

Tweedie's Estimator

$$\mathbb{E}[\theta|x] = x + \sigma^2 \frac{\mathrm{d}}{\mathrm{d}x} \log p(x)$$

$$\hat{\theta}^{TE} = x + \sigma^2 \frac{\mathrm{d}}{\mathrm{d}x} \log p(x)$$

Variational Bayesian inference

ELBO is a good Loss Function

$$\begin{aligned}
\log p(\boldsymbol{x}) &= \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}) \mathrm{d}\boldsymbol{z} \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] + D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})) \\
&= \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{重建项}} - \underbrace{D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{先验匹配项}} + \underbrace{D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x}))}_{\text{一致性项}}
\end{aligned}$$

$\log p(x)$ *is alike energy function in statistical physics and related to score-based or energy-based models*

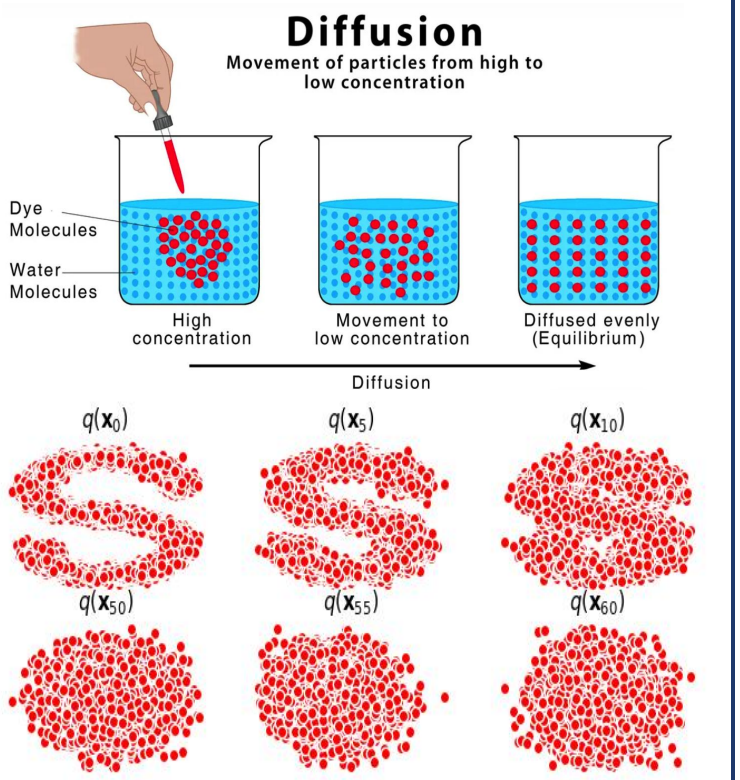
https://en.wikipedia.org/wiki/Bayesian_statistics

Outline

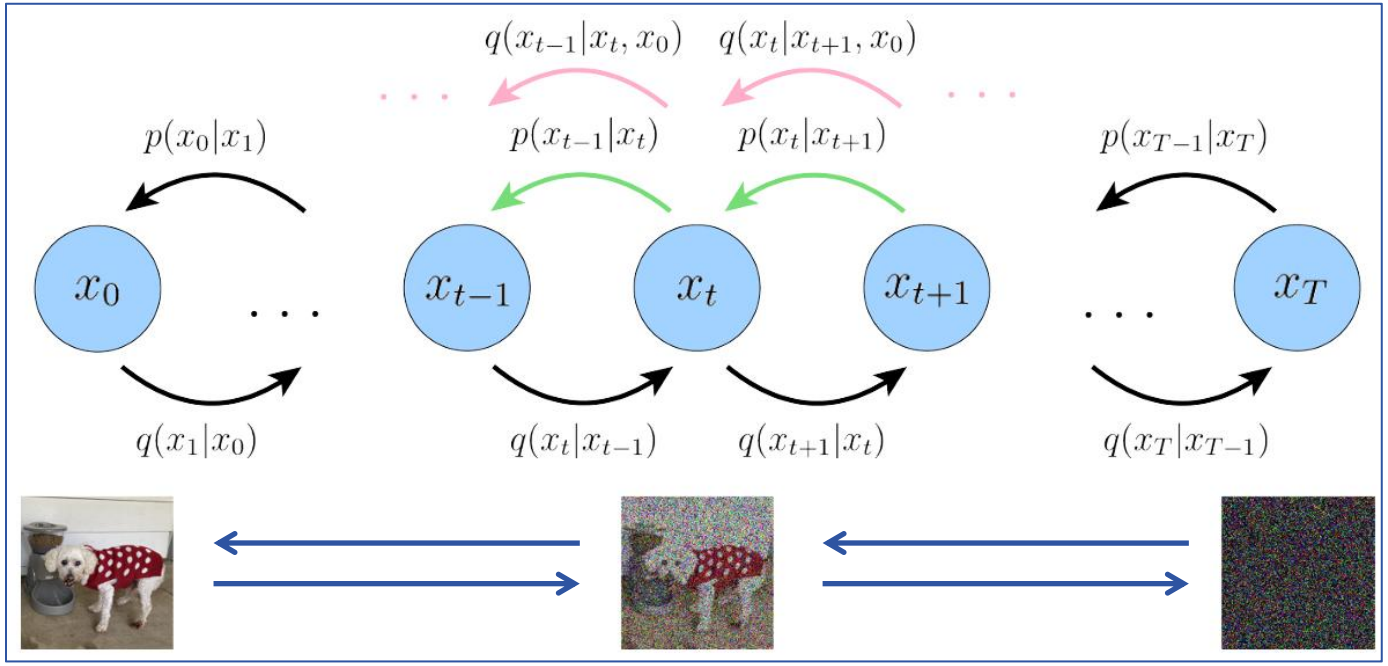
- 01 Core Concept of Bayesian Statistics
- **02 A Unified Perspective of Diffusion Models**
- 03 Energy-based Models and Guidance
- 04 Stochastic Process and Diffusion Models

Framework of Diffusion Models

(Forward) Perturbation	$q(\mathbf{x}_{1:T}\|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t\|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t\|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$
(Reverse) Denoising	$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}\|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}\|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$



Forward and Reverse Diffusion



Calvin Luo., arXiv:2208.11970 (2022)

Forward Process

Forward Process Adds Noise to Images

Perturbation of every time step using reparameterization

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

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{x}_t; 0, \boldsymbol{I})$$

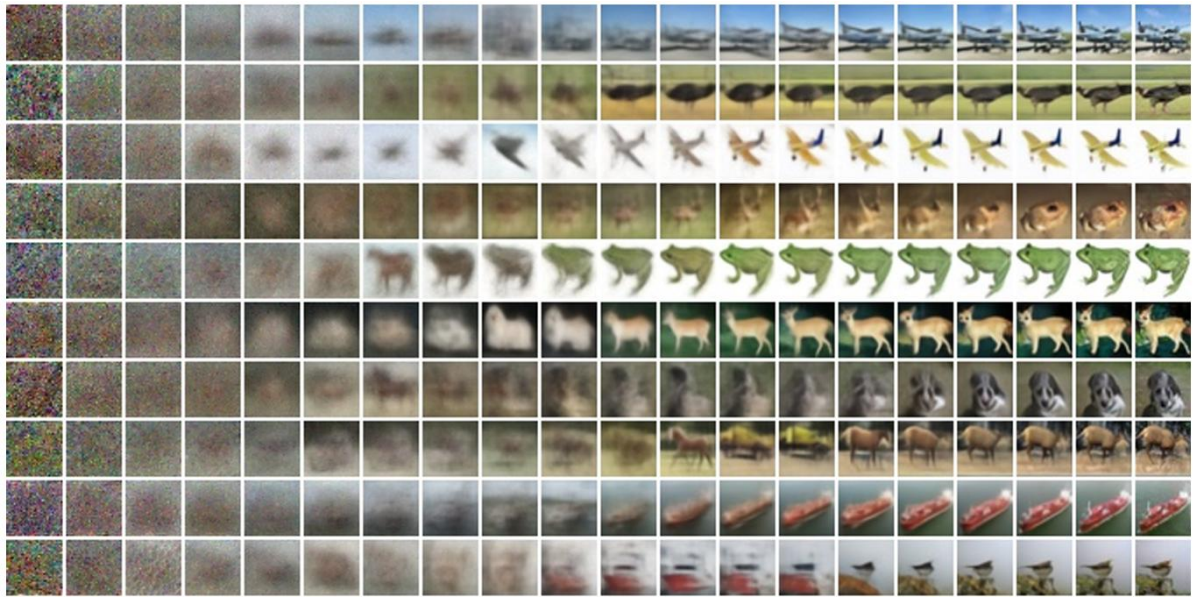
Perturbation from the original image by recursion

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I})$$

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

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



Converting from images to Gaussian noise

Abbeel, P et al., NeurIPS, 33, 6084 (2020)

Reverse Process

Reverse Process Denoises the Gaussian Noise

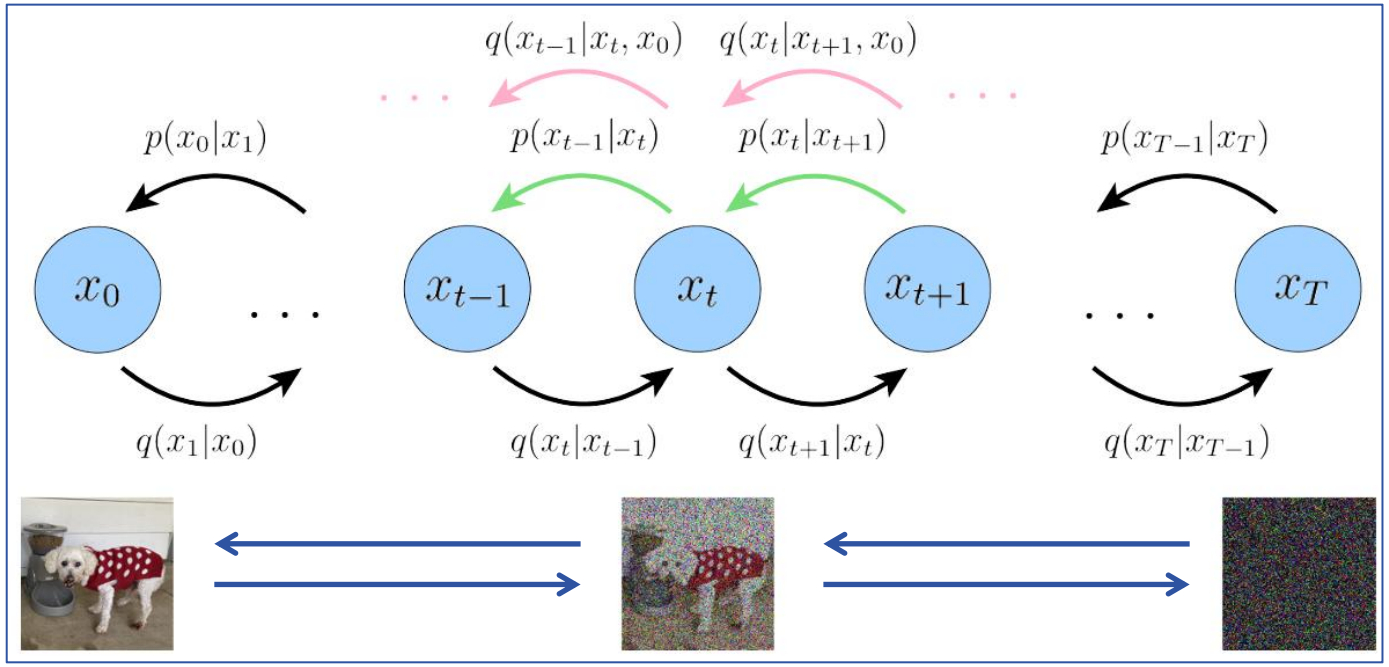
Generating new images from Gaussian noise
by predicting the reverse samples in every time step

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

$$\mathcal{N}(\boldsymbol{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}}_{\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{I}}_{\boldsymbol{\sigma}_q(t)})$$

$$p(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

We need to match the posterior probability and predicted probability



Abbeel, P et al., NeurIPS, 33, 6084 (2020)
Calvin Luo., arXiv:2208.11970 (2022)

Evidence Lower Bound (ELBO)

The formula of ELBO

$$\begin{aligned}\log p(\boldsymbol{x}) &= \log \int \frac{p(\boldsymbol{x}_{0:T})q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \mathrm{d}\boldsymbol{x}_{1:T} \\ &= \log \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right] \\ &\geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \right]\end{aligned}$$

The model is trained

by maximizing ELBO

So we need to further learn about ELBO

ELBO can be written as three terms:

$$\begin{aligned}&= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} \left[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1) \right]}_{\text{重构项}} \\ &- \underbrace{D_{KL}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{先验匹配项}} \\ &- \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} \left[D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \right]}_{\text{去噪匹配项}}\end{aligned}$$

The first two terms are determined by forward process.

So we only need to optimize the last term.

Calvin Luo., arXiv:2208.11970 (2022)

Several Equivalent Optimization Function

For KL divergence of Gaussian distribution

$$D_{KL}(\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_x,\boldsymbol{\Sigma}_x)\|\mathcal{N}(\boldsymbol{y};\boldsymbol{\mu}_y,\boldsymbol{\Sigma}_y))$$
$$=\frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_y|}{|\boldsymbol{\Sigma}_x|}+tr(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_x)+(\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)^{\mathrm{T}}\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\mu}_y-\boldsymbol{\mu}_x)-n\right]$$

$$\arg\min_{\theta} D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))$$

$$\arg\min_{\theta}\frac{1}{2\sigma_q^2(t)}\left[\|\boldsymbol{\mu}_\theta-\boldsymbol{\mu}_q\|_2^2\right]$$

So we get the first **optimization function**

*we need to predict the **mean value** by neural networks*

Predict samples by neural networks

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

$$\arg\min_{\theta}\frac{1}{2\boldsymbol{\sigma}_q^2(t)}\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\|(\hat{\boldsymbol{x}}_\theta(\boldsymbol{x}_t,t)-\boldsymbol{x}_0)\|_2^2\right]$$

Predict noise by neural networks

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t,t)=\frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)$$

$$\arg\min_{\theta}\frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t}\left[\|(\boldsymbol{\epsilon}-\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t))\|_2^2\right]$$

Calvin Luo., arXiv:2208.11970 (2022)

Several Equivalent Optimization Function

Tweedie's Estimator

$$\mathbb{E}[\theta|x] = x + \sigma^2 \frac{\mathrm{d}}{\mathrm{d}x} \log p(x)$$

$$\mathbb{E}[\boldsymbol{\mu}_{\boldsymbol{x}_t}|\boldsymbol{x}_t] = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

$$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

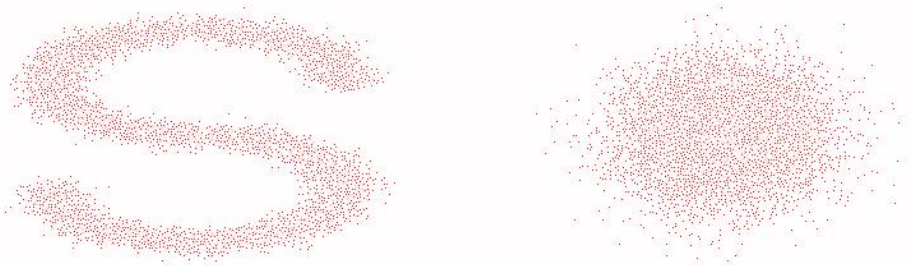
$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{\alpha_t} \left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p(\boldsymbol{x}_t)\|_2^2\right]$$

Relation between noise and score function

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_0$$

The score function measures how the data is move to maximize, it clear that the opposite direction is noisy and as can be seen from the above formula.



Calvin Luo., arXiv:2208.11970 (2022)

Outline

- 01 Core Concept of Bayesian Statistics
- 02 A Unified Persperctive of Diffusion Models
- **03 Energy-based Models and Guidance**
- 04 Stochastic Process and Diffusion Models

Energy-based Models

Why the last formula can lead to energy-based model

$$p_\theta(\boldsymbol{x}) = \frac{1}{z_\theta} e^{-f_\theta(\boldsymbol{x})}$$

$$\begin{aligned}
\nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} \log\left(\frac{1}{\boldsymbol{z_\theta}} e^{-f_\theta(\boldsymbol{x})}\right) \\
&= \nabla_x \log \frac{1}{z_\theta} + \nabla_x \log e^{-f_\theta(\boldsymbol{x})} \\
&= -\nabla_x f_\theta(\boldsymbol{x}) \\
&\approx \boldsymbol{s_\theta}(\boldsymbol{x})
\end{aligned}$$

$$\mathbb{E}_{p(\boldsymbol{x})}\left[\|\boldsymbol{s_\theta}(\boldsymbol{x}) - \nabla \log p(\boldsymbol{x})\|_2^2\right]$$

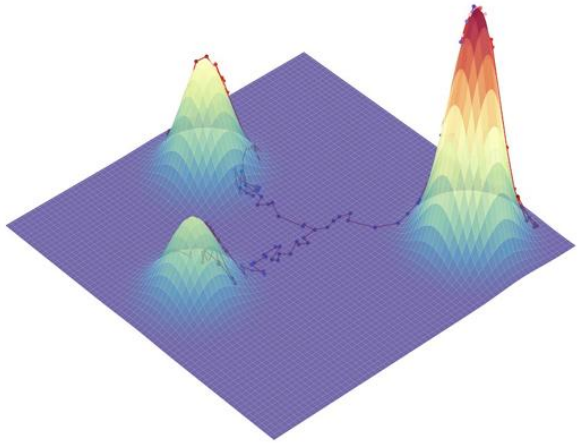
We can further see the relationship between the diffusion models and statistical physics.

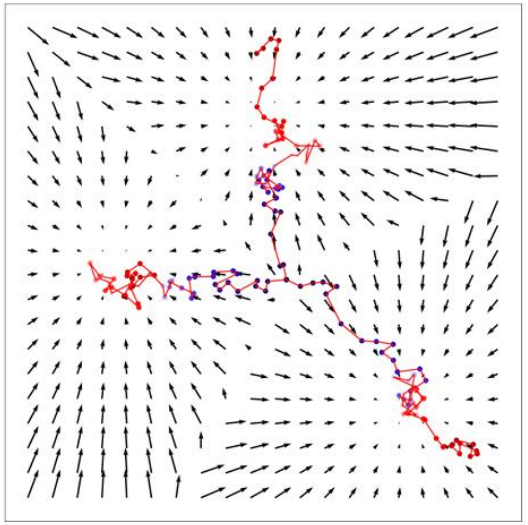
Langevin dynamics

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + c\nabla \log p(\boldsymbol{x}_i) + \sqrt{2c}\boldsymbol{\epsilon}$$

The score function represents a move on the manifold

Noise protection against local optimality





Calvin Luo., arXiv:2208.11970 (2022)

Guidance

Classifier Guidance

$$\begin{aligned}\nabla \log p(\boldsymbol{x}_t|y) &= \nabla \log \left(\frac{p(\boldsymbol{x}_t)p(y|\boldsymbol{x}_t)}{p(y)} \right) \\ &= \nabla \log p(\boldsymbol{x}_t) + \nabla \log p(y|\boldsymbol{x}_t) - \nabla \log p(y) \\ &= \underbrace{\nabla \log p(\boldsymbol{x}_t)}_{\text{无条件得分}} + \underbrace{\nabla \log p(y|\boldsymbol{x}_t)}_{\text{对抗梯度}}\end{aligned}$$

$$\nabla \log p(\boldsymbol{x}_t|y) = \nabla \log p(\boldsymbol{x}_t) + \gamma \nabla \log p(y|\boldsymbol{x}_t)$$

Classifer guidance need to train two different diffusion models

So training is ***very expensive.***

Classifier-Free Guidance

$$\nabla \log p(y|\boldsymbol{x}_t) = \nabla \log p(\boldsymbol{x}_t|y) - \nabla \log p(\boldsymbol{x}_t)$$

$$\begin{aligned}\nabla \log p(\boldsymbol{x}_t|y) &= \nabla \log p(\boldsymbol{x}_t) + \gamma(\nabla \log p(\boldsymbol{x}_t|y) - \nabla \log p(\boldsymbol{x}_t)) \\ &= \underbrace{\gamma \nabla \log p(\boldsymbol{x}_t|y)}_{\text{条件得分}} + \underbrace{(1-\gamma)\nabla \log p(\boldsymbol{x}_t)}_{\text{无条件得分}}\end{aligned}$$

Classifer-Free guidance can train the diffusion models with guidance with ***more convenience.***

We can get the output of diffusion models with ***more requirements.***

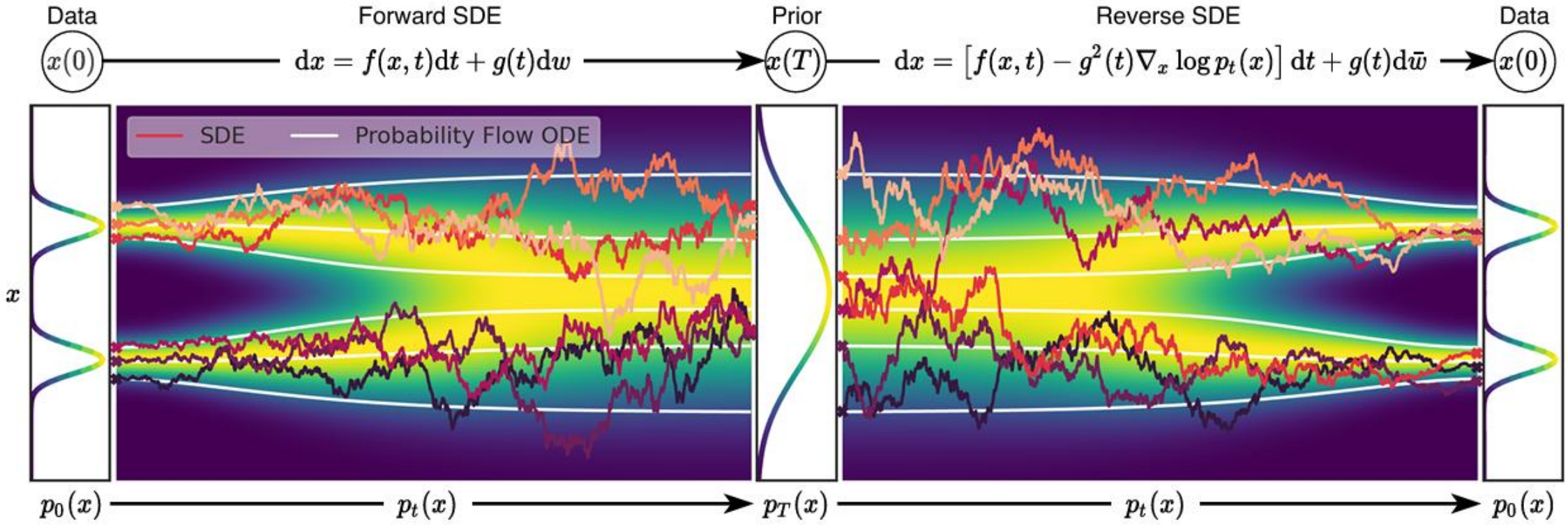
Abbeel, P et al., NeurIPS, 33, 6084 (2020)

Outline

- 01 Core Concept of Bayesian Statistics
- 02 A Unified Persperctive of Diffusion Models
- 03 Energy-based Models and Guidance
- **04 Stochastic Process and Diffusion Models**

Stochastic Differential Equations

Real diffusion process can be describe by stochastic differential equations(SDE) and Anderson prove the forward SDE have a reverse formula.
So SDE might be useful in generative diffusion models.



https://yang-song.net/blog/2021/score/

From SDE to DDPM

It is easy to show that DDPM can be seen as a special case of SDE.

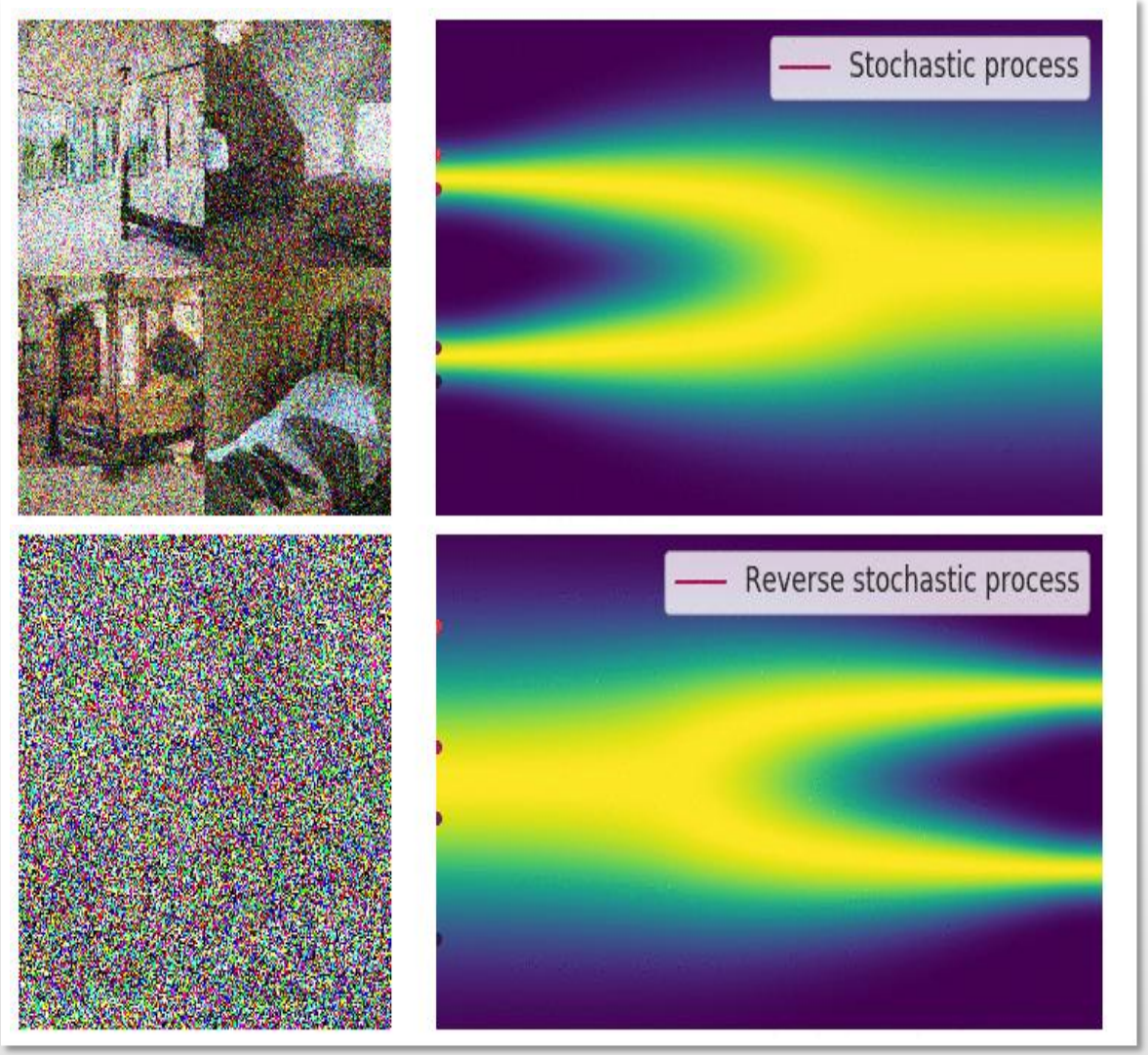
$$\boldsymbol{x}(t+\Delta t)=\sqrt{1-\beta(t+\Delta t)\Delta t}\boldsymbol{x}(t)+\sqrt{\beta(t+\Delta t)\Delta t}\boldsymbol{\epsilon}(t)$$

$$\boldsymbol{x}(t+\Delta t)\approx\left[1-\frac{\beta(t+\Delta t)\Delta t}{2}\right]\boldsymbol{x}(t)+\sqrt{\beta(t+\Delta t)\Delta t}\boldsymbol{\epsilon}(t)$$

$$\mathrm{d}\boldsymbol{x}(t)\approx-\frac{\beta(t)\boldsymbol{x}(t)}{2}\mathrm{d}t+\sqrt{\beta(t)}\sqrt{\mathrm{d}t}\boldsymbol{\epsilon}(t)$$

$$\mathrm{d}\boldsymbol{x}=-\frac{\beta(t)\boldsymbol{x}}{2}\mathrm{d}t+\sqrt{\beta(t)}\mathrm{d}\boldsymbol{w}$$

SDE provides a lot of convenience for design and give an understanding of diffusion model of stochastic thermodynamics



Song, Y et al.,ICLR, 2020

谢谢大家
敬请批评指正

Thanks for listening

Fancheng Li （李凡成）