模式识别
Pattern Recognition

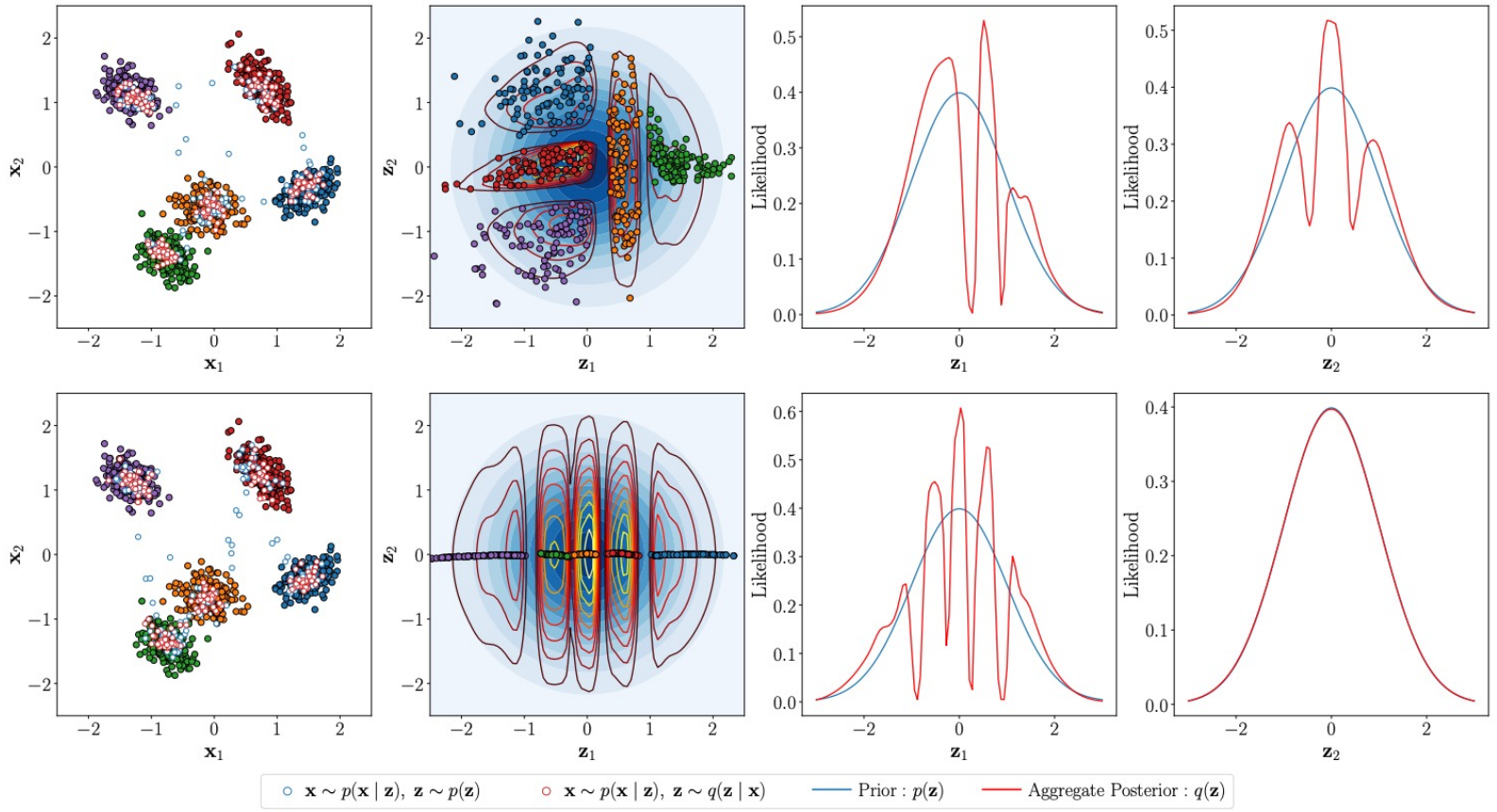
李泽榉，复旦大学 生物医学工程与技术创新学院

课件内容参考MIT 6.S978 Deep Generative Models

目录

VAE

- VAE的空洞问题：没采样到的样本很难生成

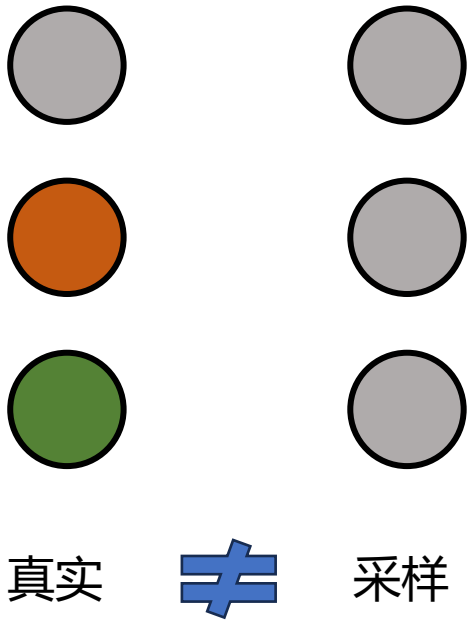


Demystifying Variational Diffusion Models

建模条件分布

VAE的空洞问题：

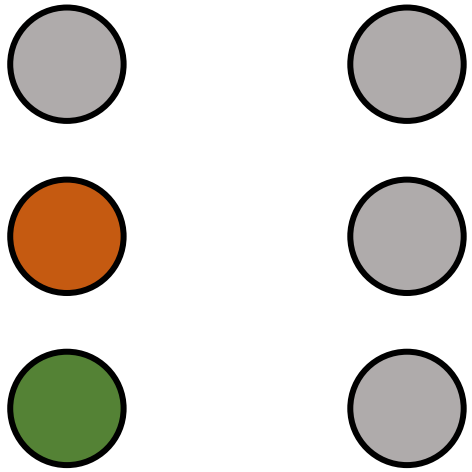
- 表征缺少相关性
- 难以建模没采样到的点



真实 ≠ 采样

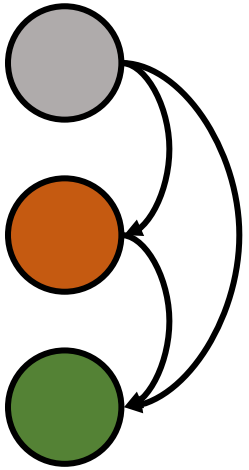
建模条件分布

VAE的空洞问题：
- 表征缺少相关性
- 难以建模没采样到的点

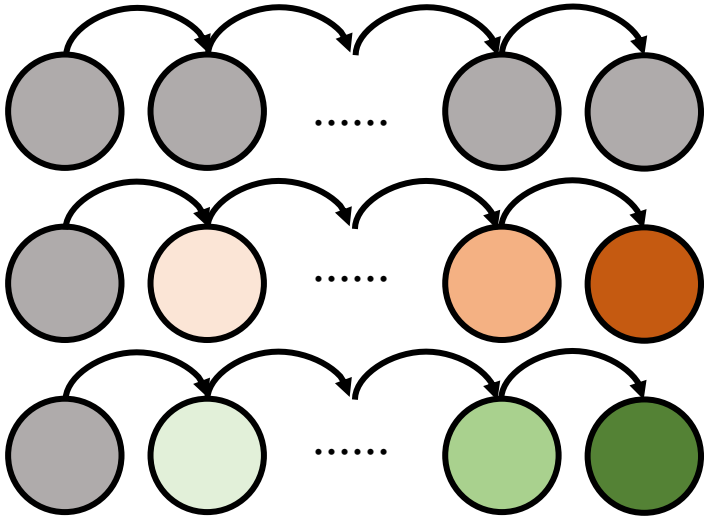


真实 ≠ 采样

自回归模型：
- 基于时间建模
- 条件依赖，顺序建模

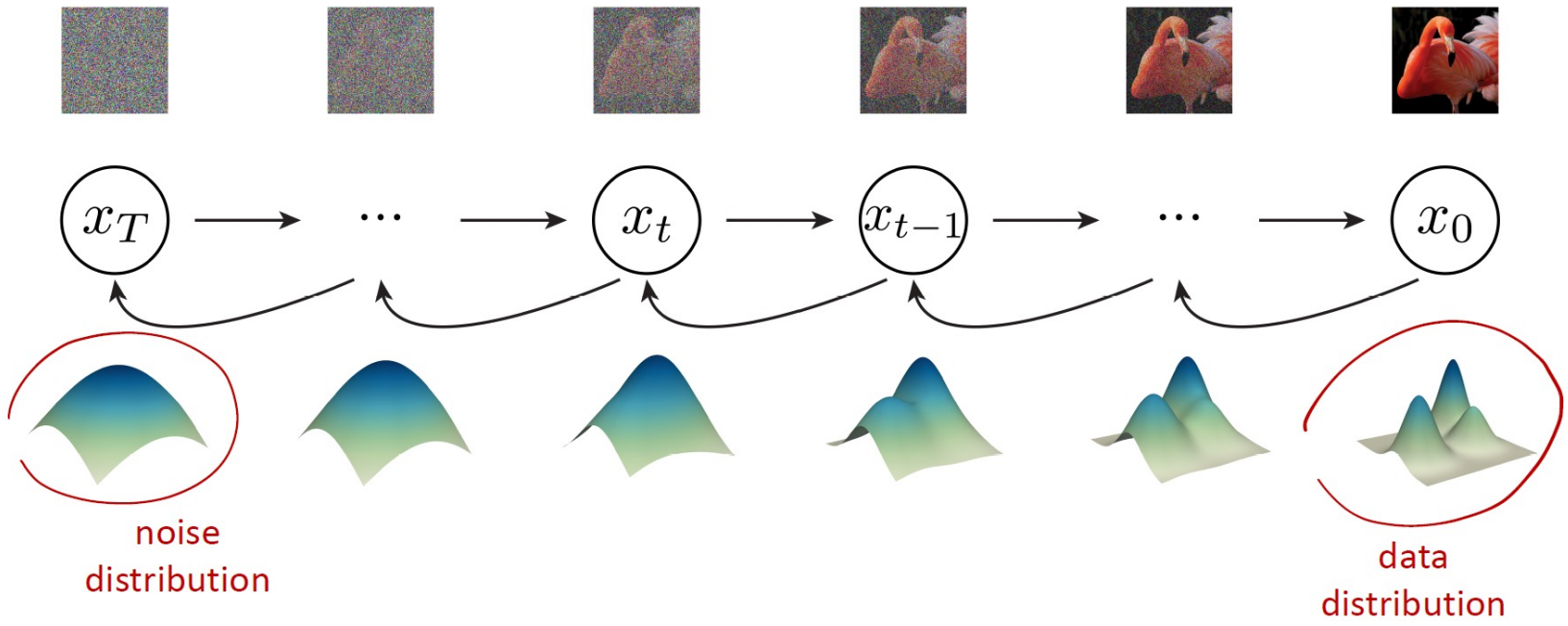


扩散模型：
- 基于空间建模
- 迭代去噪，精炼相关性



扩散模型

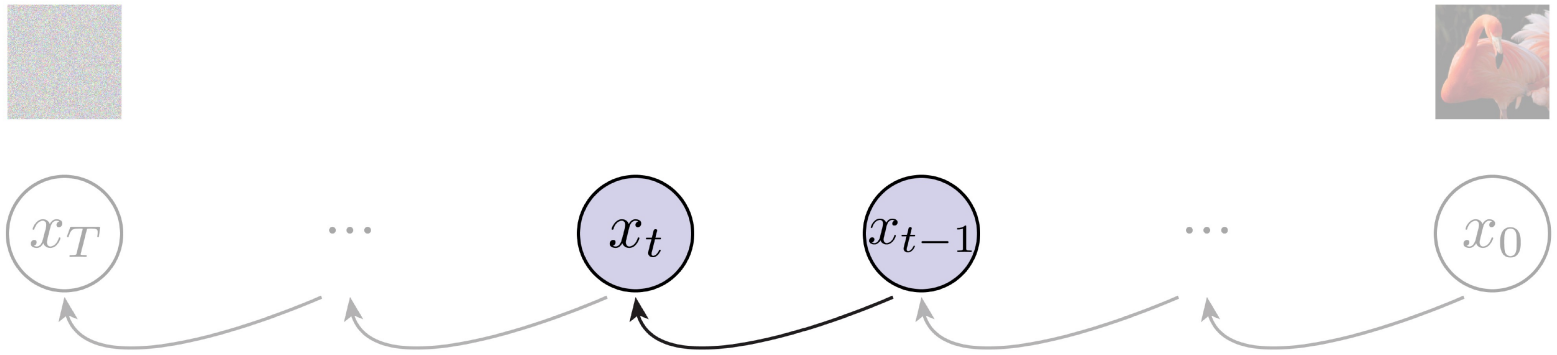
- 扩散模型通过构造，将**聚合后验**定义为等于先验来解决空洞问题
- 但因为后验都遵循加噪过程，其牺牲了学习有意义表征的能力



目录

前向过程

- 基于马尔可夫链的加入高斯噪声
- 为了保证图像可控，一般噪声会遵循Variance Preserving (VP)

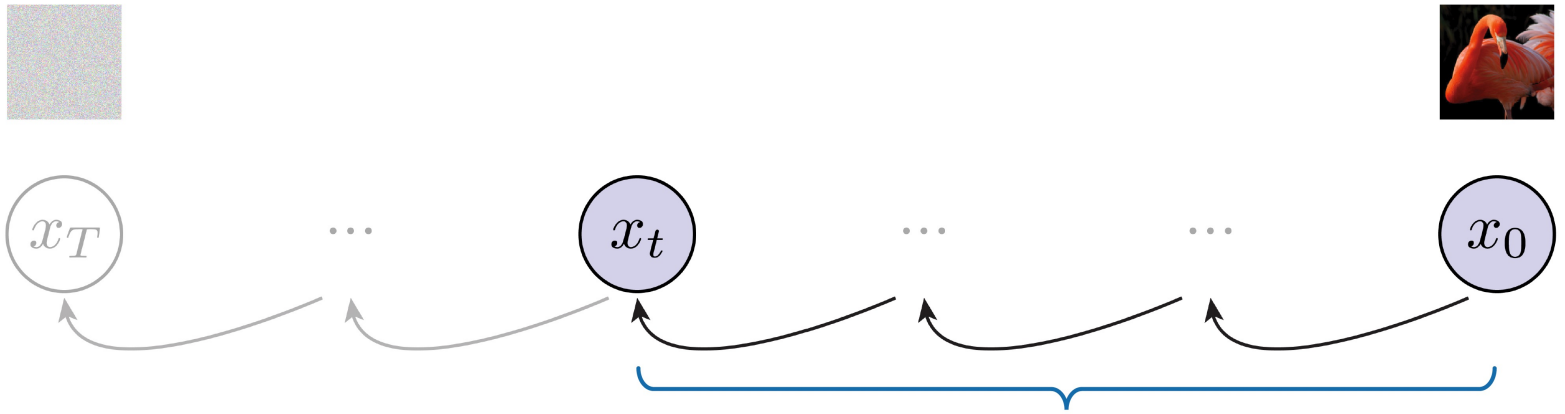


$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

coefficients:
variance preserving

前向过程

- 高斯分布再生性：多个独立高斯随机变量的和仍然是高斯分布
- 所以可以训练的时候随机取一步



- sampling without simulation
- x_t from x_0 in closed form

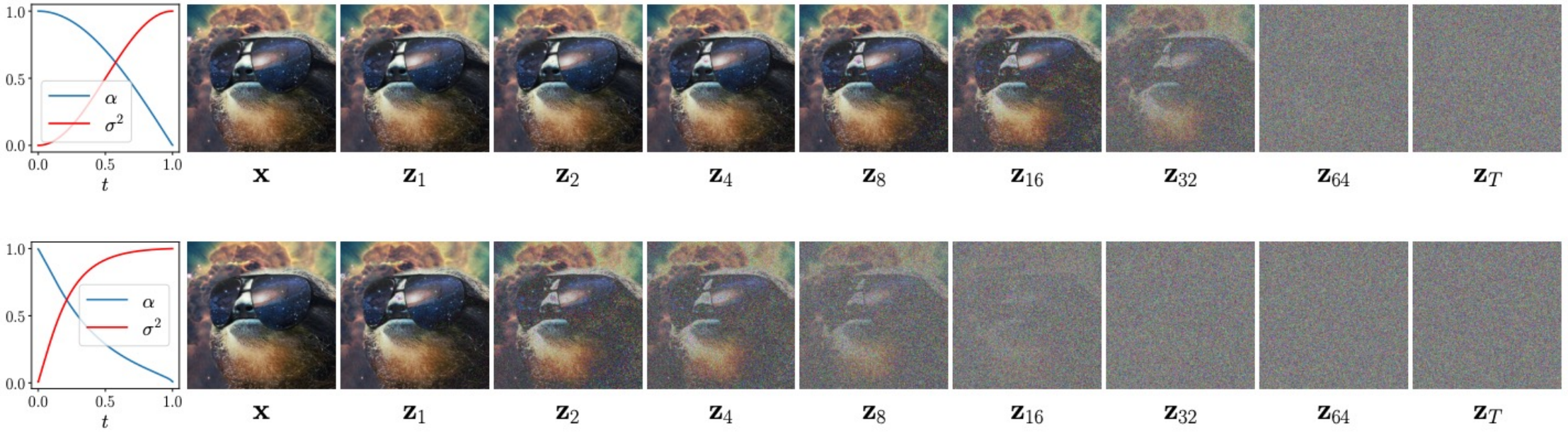
$$q(x_t \mid x_0) = \mathcal{N}(x_t \mid \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

coefficients given by β

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

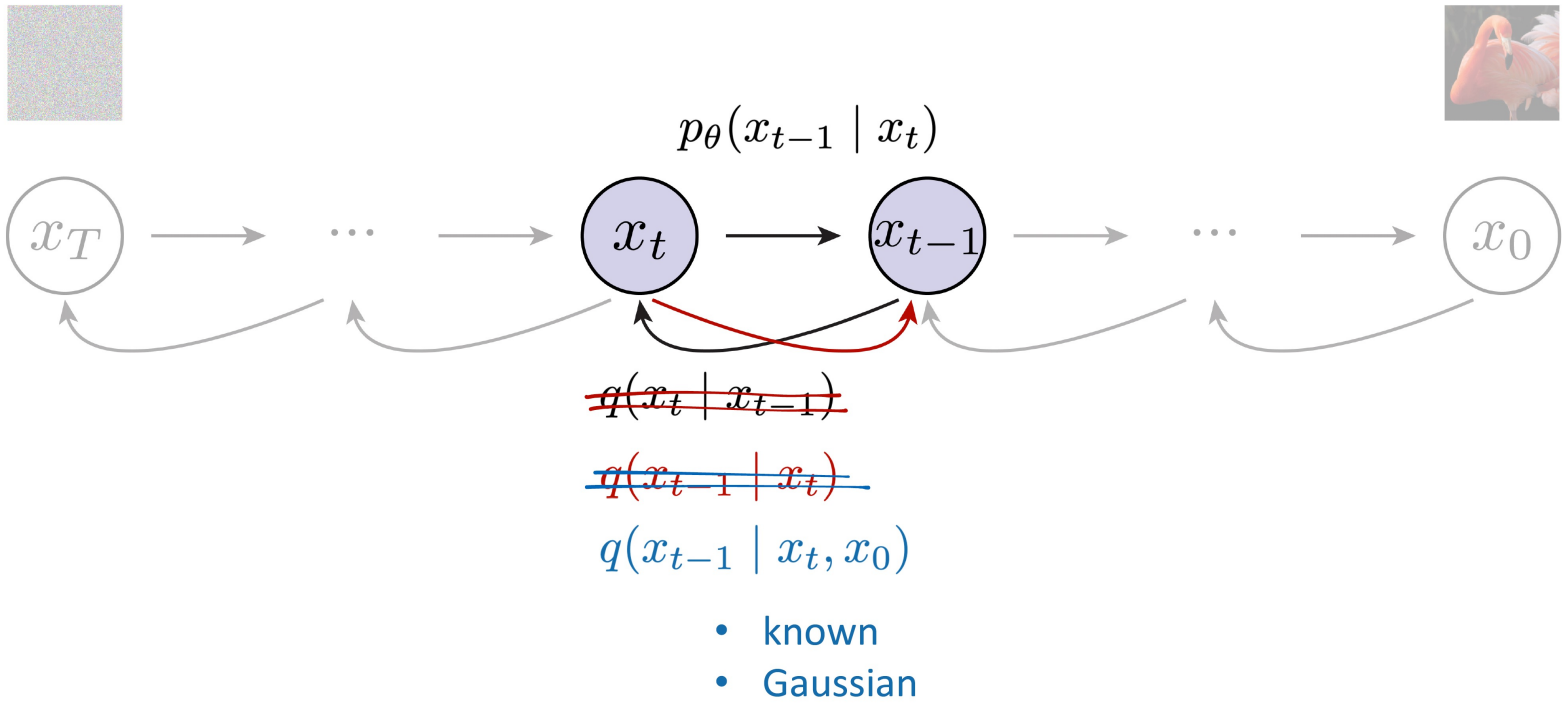
前向过程

- 噪声在干净步骤均值最接近1，方差最小
- 噪声在噪声步骤均值最接近0，方差最大



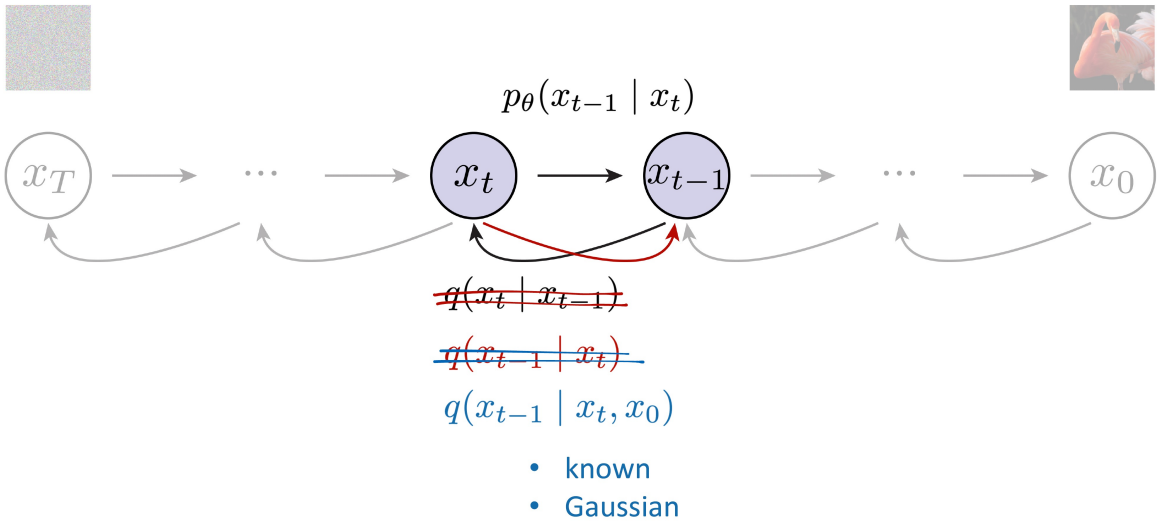
反向过程

- 在**同时观察到** x_t 和知道 x_0 的情况下，对 x_{t-1} 的估计



反向过程

- 在**同时观察到** x_t 和知道 x_0 的情况下，对 x_{t-1} 的估计



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

马尔可夫性⟹

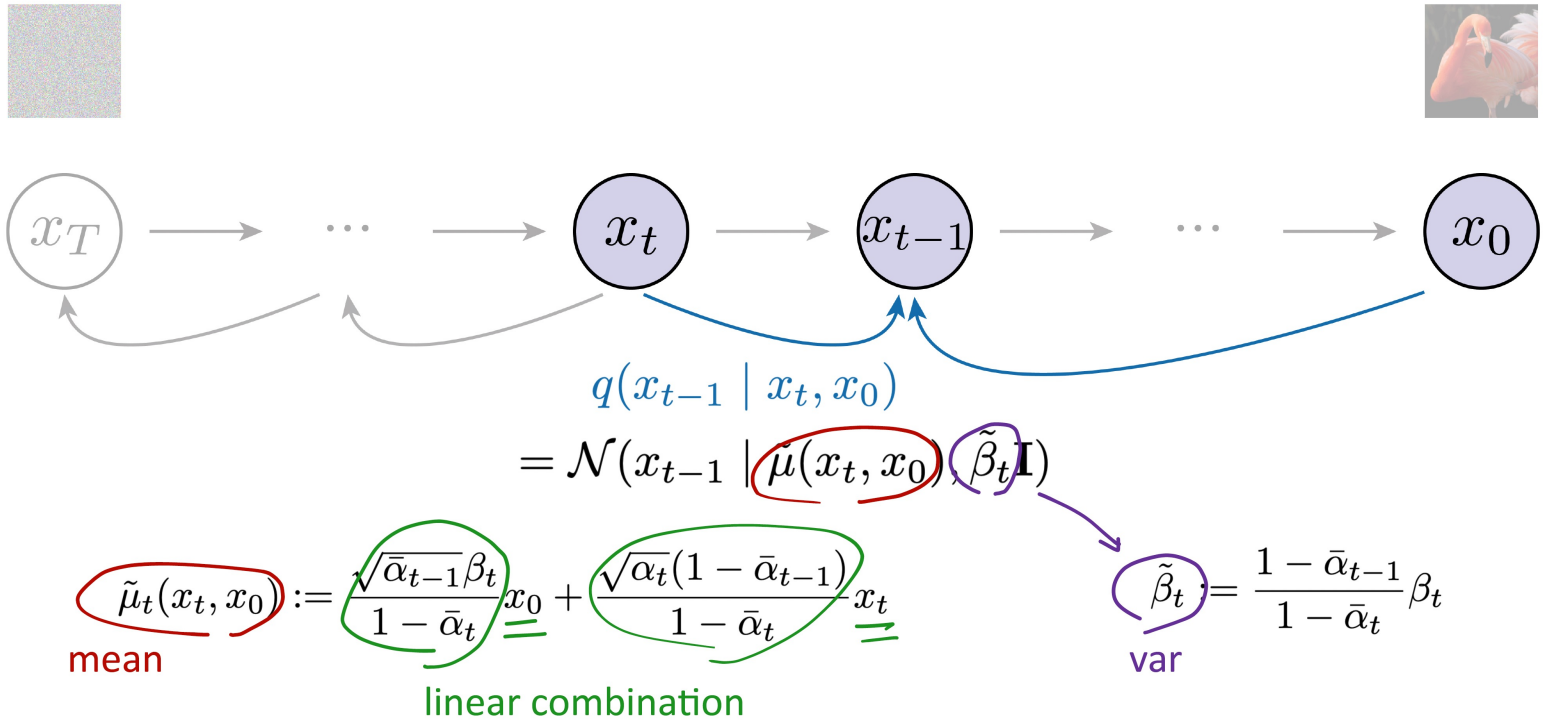
$$q(x_{t-1}|x_t, x_0) = \boldsymbol{q(x_t|x_{t-1})}\frac{\boldsymbol{q(x_{t-1}|x_0)}}{q(x_t|x_0)}$$

代入高斯分布，忽略与x_{t-1}无关常数 ⟹

$$q(x_{t-1}|x_t, x_0) \propto \exp\left(-\frac{1}{2}\left[\frac{\left\|x_t - \sqrt{1-\beta_t}\boldsymbol{x_{t-1}}\right\|^2}{\beta_t} + \frac{\left\|\boldsymbol{x_{t-1}} - \sqrt{\bar{\alpha}_{t-1}}x_0\right\|^2}{\bar{\beta}_{t-1}}\right]\right)$$

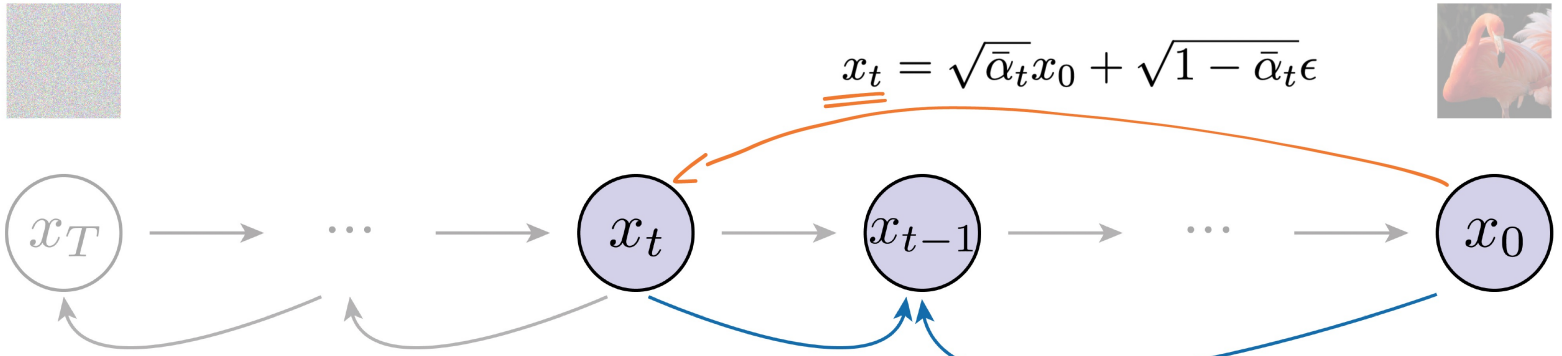
反向过程

- 在x_0已知时，反向转移也是已知的高斯分布



反向过程

- 将x_0到 x_t的过程整理成统一的噪声ϵ



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

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

mean

$$= \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$$

“noise”

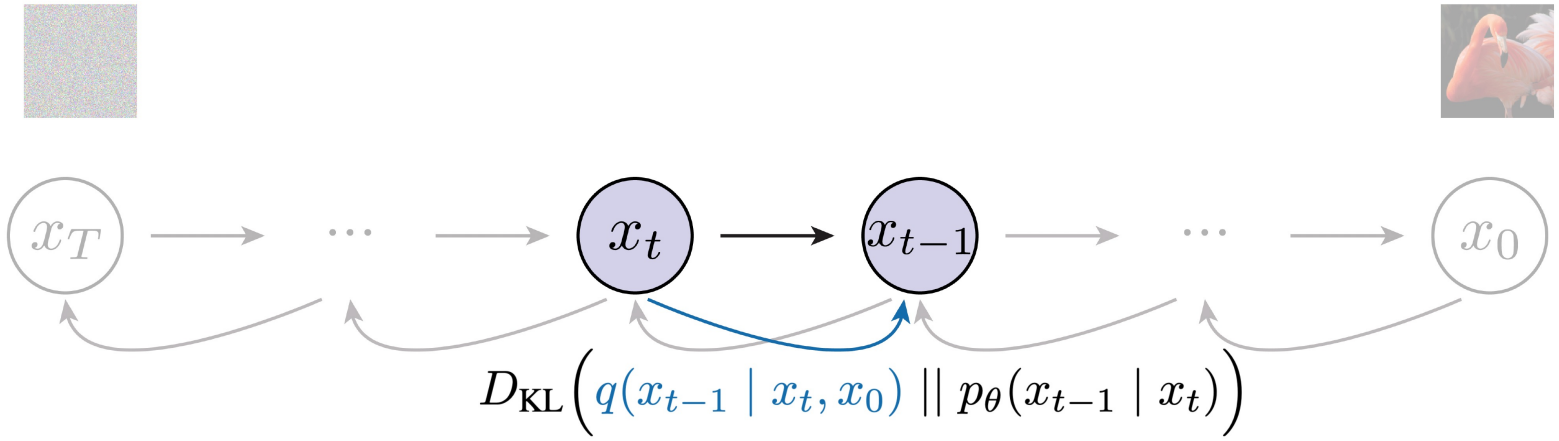
$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

var

注意，这里的噪声是加在x_0上，用于得到x_t的

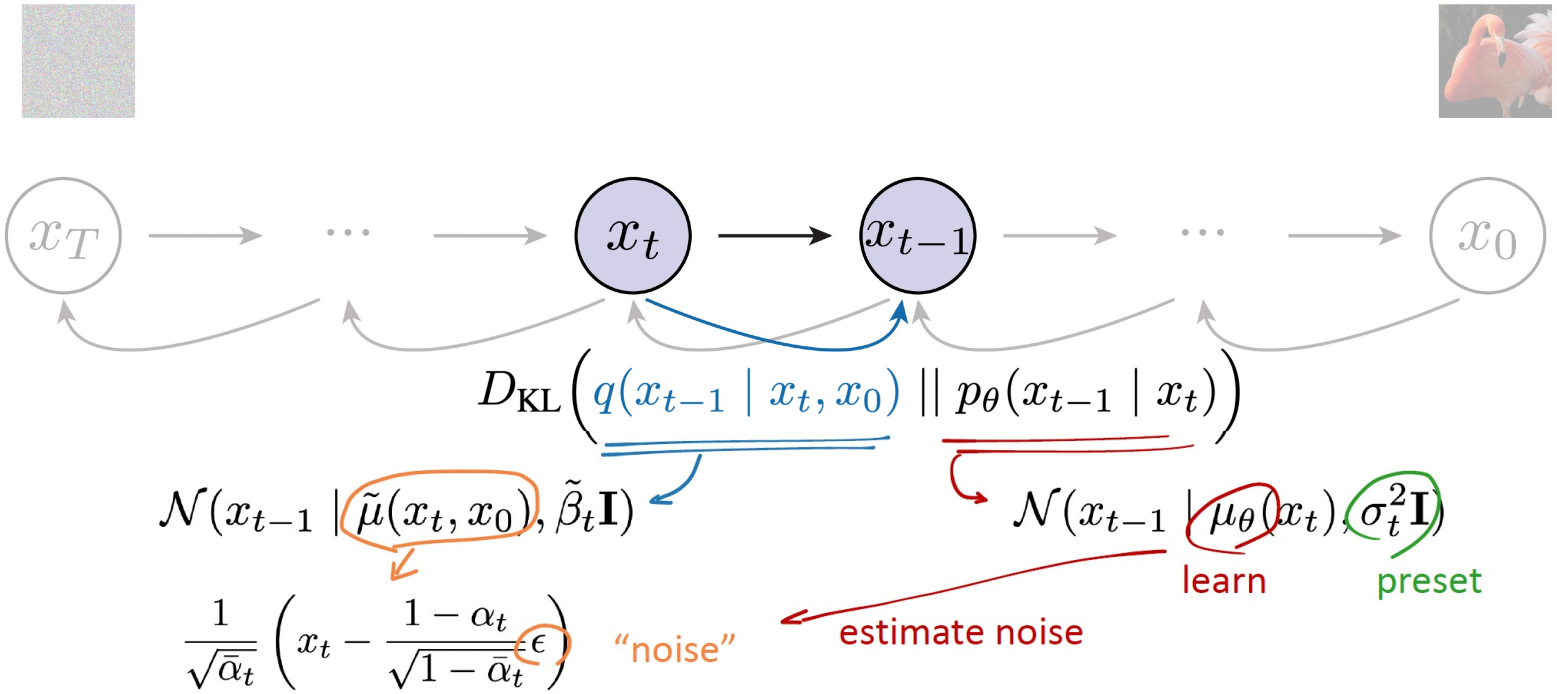
反向过程

- 最小化KL散度



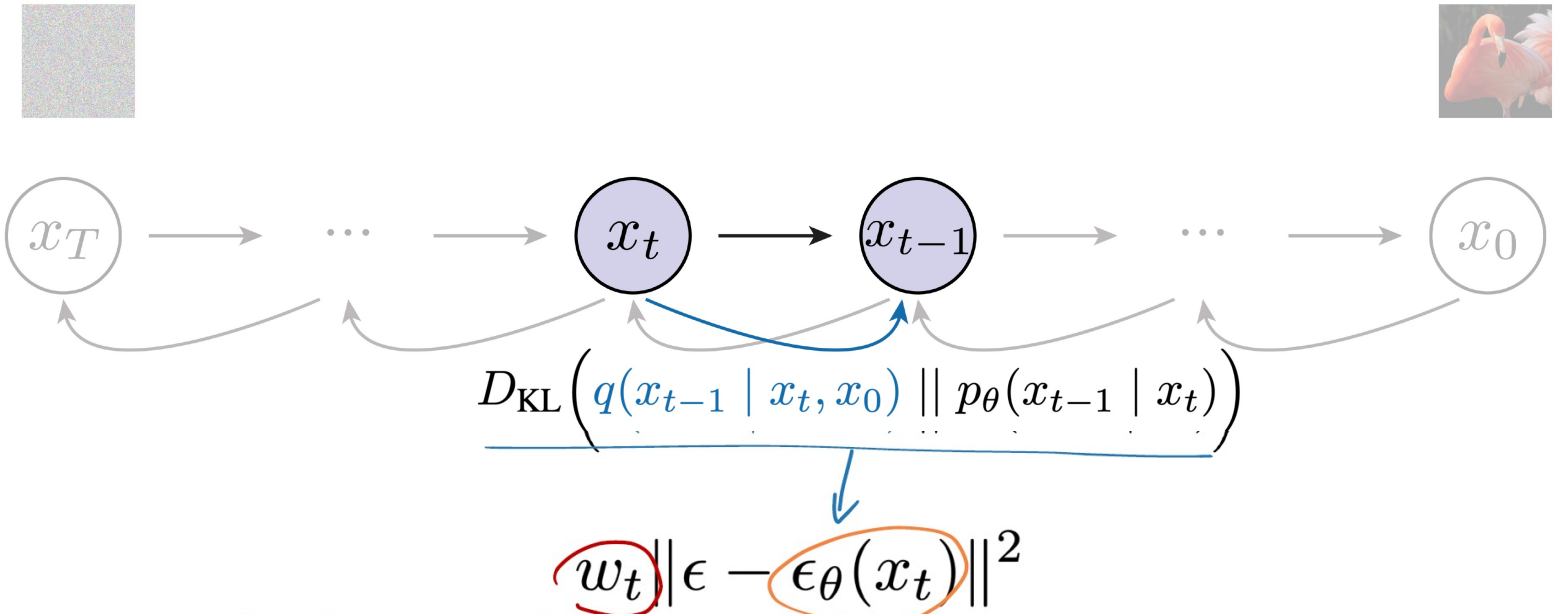
反向过程

- KL散度简化成噪声预测



反向过程

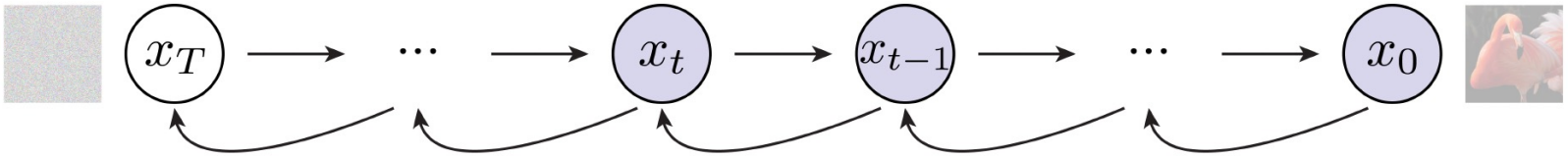
- 噪声预测的L2损失



目录

训练目标

- 所有时间步加和，通过引入近似后验分布来逼近真实后验
- 是一种ELBO衍生的损失



- variational lower bound
- like ELBO

$$\mathcal{L}_{\text{VLB}} := \mathcal{L}_T + \mathcal{L}_{T-1} + \ldots + \mathcal{L}_0$$

$$\mathcal{L}_T := D_{\text{KL}}\left(q(x_T \mid x_0) \,||\, p_\theta(x_T)\right)$$

只是为了完整，不需要训练

$$\mathcal{L}_{t-1} := D_{\text{KL}}\left(q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)\right)$$

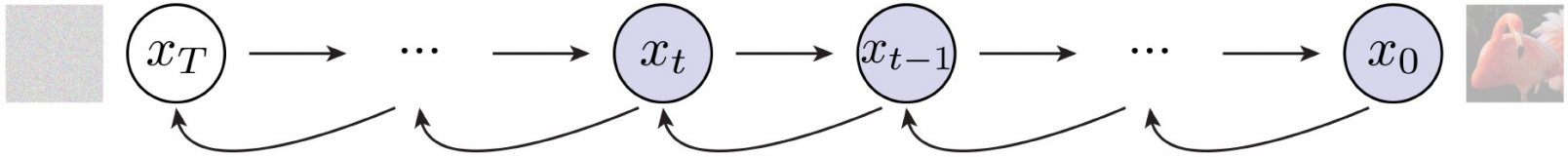
学习如何一步步去噪（核心学习任务）

$$\mathcal{L}_0 := -\log p_\theta(x_0 \mid x_1)$$

最终的重构质量，被隐式忽略

训练目标

- 可以直接简化成去噪任务



$$\mathcal{L} = \mathbb{E}_{x_0,t,\epsilon}\left[w_t \|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

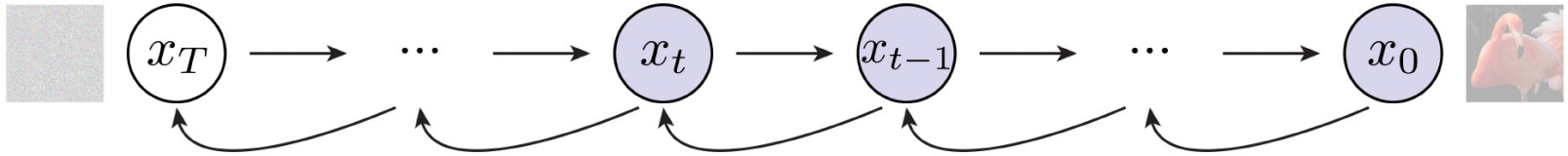
over p_{data}

over $[1, T]$

over $\mathcal{N}(0, \mathbf{I})$

训练目标

- 理论上$w_t = \frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)}$在信噪比变化最快时间步权重高，高噪声水平权重小
- **理论最优 ≠ 实践最佳**，实际上发现$w_t = 1$表现更好



$$\mathcal{L} = \mathbb{E}_{x_0,t,\epsilon}\left[w_t \|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

set as 1 **(critical)**

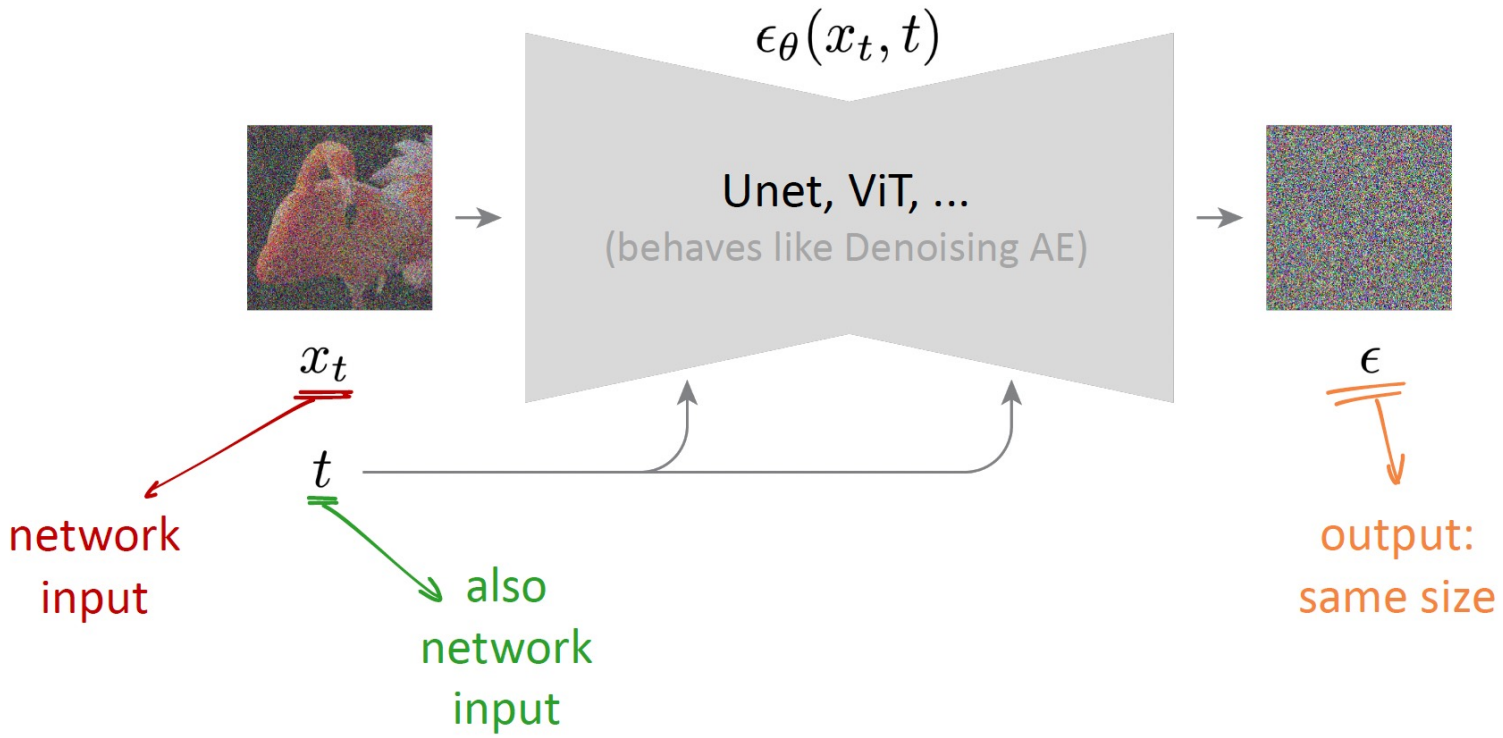
Objective	IS	FID
L, learned diagonal $\boldsymbol{\Sigma}$	–	–
L, fixed isotropic $\boldsymbol{\Sigma}$	7.67 ± 0.13	13.51
$\|\tilde{\epsilon} - \epsilon_\theta\|^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	**3.17**

[Ho et al. 2020]; see more in [Salimans & Ho, 2022]

目录

噪声条件网络

- 利用同一个网络来预测所有时间步噪声 $p_\theta(x_{t-1}|x_t)$



噪声条件网络

- 基础扩散模型的训练和采样过程

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

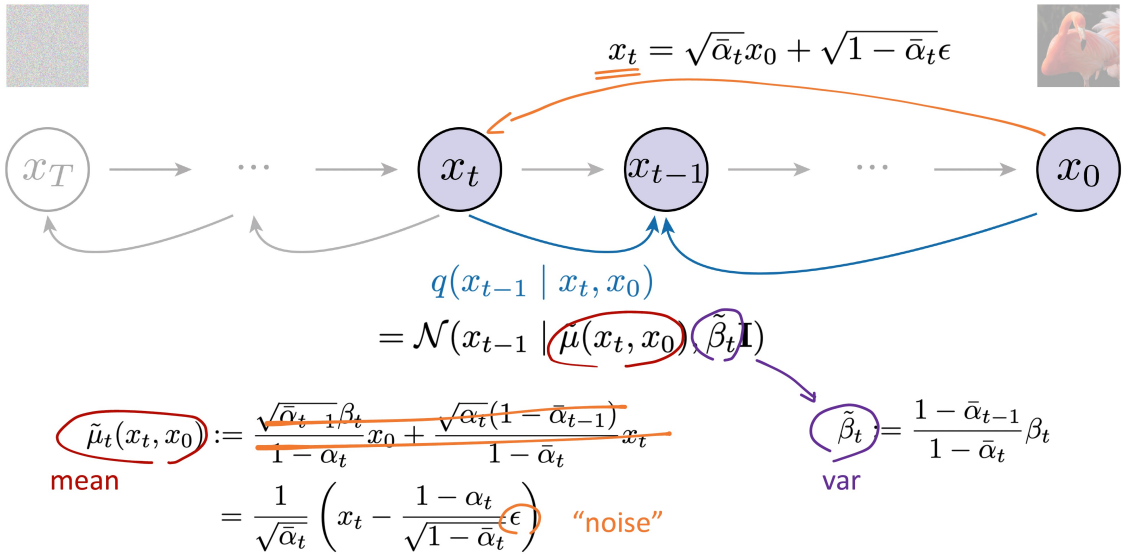
x_t

Algorithm 2 Sampling

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

estimated μ

sampling from
estimated distribution



扩散模型的设计空间

- 扩散模型的成功很大程度归功于大规模的工程实践与集体技术攻坚

Table 1: Specific design choices employed by different model families. N is the number of ODE solver iterations that we wish to execute during sampling. The corresponding sequence of time steps is $\{t_0, t_1, \ldots, t_N\}$, where $t_N = 0$. If the model was originally trained for specific choices of N and $\{t_i\}$, the originals are denoted by M and $\{u_j\}$, respectively. The denoiser is defined as $D_\theta(\boldsymbol{x};\sigma) = c_{\text{skip}}(\sigma)\boldsymbol{x} + c_{\text{out}}(\sigma)F_\theta\big(c_{\text{in}}(\sigma)\boldsymbol{x}; c_{\text{noise}}(\sigma)\big)$; F_θ represents the raw neural network layers.

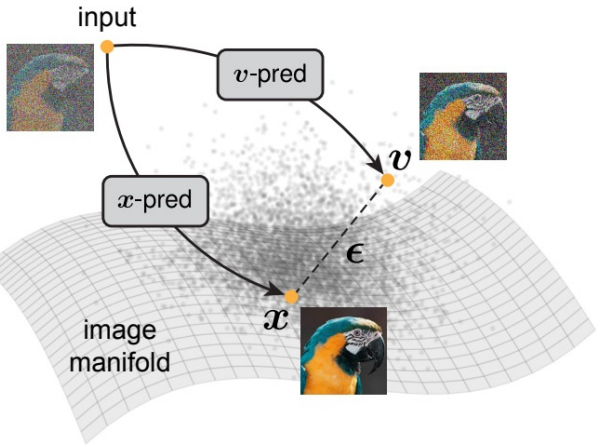
		VP [49]	**VE [49]**	**iDDPM [37] + DDIM [47]**	**Ours ("EDM")**
Sampling (Section 3)					
ODE solver		Euler	Euler	Euler	2nd order Heun
Time steps	$t_{i<N}$	$1 + \frac{i}{N-1}(\epsilon_s - 1)$	$\sigma_{\max}^2 \left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0 + \frac{M-1-j_0}{N-1} i + \frac{1}{2} \rfloor}$, where $u_M = 0$, $u_{j-1} = \sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j, C_1)} - 1}$	$\left(\sigma_{\max}^{\frac{1}{\rho}} + \frac{i}{N-1}(\sigma_{\min}^{\frac{1}{\rho}} - \sigma_{\max}^{\frac{1}{\rho}})\right)^\rho$
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2 + \beta_{\min} t} - 1}$	$\sqrt{t}$	t	t
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2 + \beta_{\min} t}}$	1	1	1
Network and preconditioning (Section 5)					
Architecture of F_θ		DDPM++	NCSN++	DDPM	(any)
Skip scaling	$c_{\text{skip}}(\sigma)$	1	1	1	$\sigma_{\text{data}}^2/\left(\sigma^2 + \sigma_{\text{data}}^2\right)$
Output scaling	$c_{\text{out}}(\sigma)$	$-\sigma$	σ	$-\sigma$	$\sigma \cdot \sigma_{\text{data}}/\sqrt{\sigma_{\text{data}}^2 + \sigma^2}$
Input scaling	$c_{\text{in}}(\sigma)$	$1/\sqrt{\sigma^2+1}$	1	$1/\sqrt{\sigma^2+1}$	$1/\sqrt{\sigma^2 + \sigma_{\text{data}}^2}$
Noise cond.	$c_{\text{noise}}(\sigma)$	$(M-1)\,\sigma^{-1}(\sigma)$	$\ln(\frac{1}{2}\sigma)$	$M - 1 - \arg\min_j \lvert u_j - \sigma \rvert$	$\frac{1}{4}\ln(\sigma)$
Training (Section 5)					
Noise distribution		$\sigma^{-1}(\sigma) \sim \mathcal{U}(\epsilon_t, 1)$	$\ln(\sigma) \sim \mathcal{U}(\ln(\sigma_{\min}), \ln(\sigma_{\max}))$	$\sigma = u_j,\ \ j \sim \mathcal{U}\{0, M-1\}$	$\ln(\sigma) \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$
Loss weighting	$\lambda(\sigma)$	$1/\sigma^2$	$1/\sigma^2$	$1/\sigma^2$ (note: *)	$\left(\sigma^2 + \sigma_{\text{data}}^2\right)/(\sigma \cdot \sigma_{\text{data}})^2$
Parameters		$\beta_d = 19.9, \beta_{\min} = 0.1$ $\epsilon_s = 10^{-3}, \epsilon_t = 10^{-5}$ $M = 1000$	$\sigma_{\min} = 0.02$ $\sigma_{\max} = 100$	$\bar{\alpha}_j = \sin^2(\frac{\pi}{2}\frac{j}{M(C_2+1)})$ $C_1 = 0.001, C_2 = 0.008$ $M = 1000, j_0 = 8^\dagger$	$\sigma_{\min} = 0.002, \sigma_{\max} = 80$ $\sigma_{\text{data}} = 0.5, \rho = 7$ $P_{\text{mean}} = -1.2, P_{\text{std}} = 1.2$

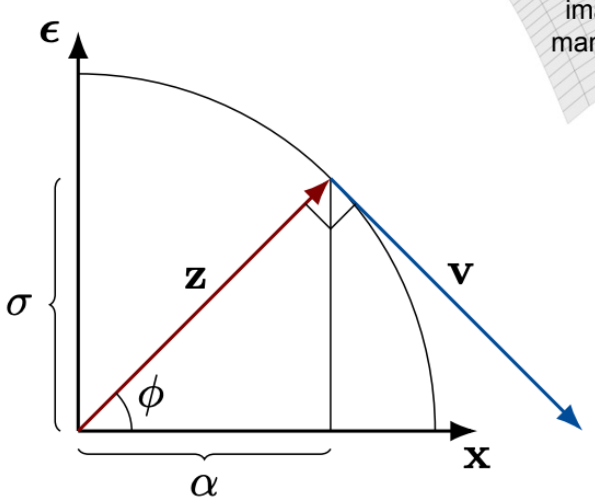
* iDDPM also employs a second loss term L_{vlb} † In our tests, $j_0 = 8$ yielded better FID than $j_0 = 0$ used by iDDPM

Elucidating the Design Space of Diffusion-Based Generative Models

多种参数化方法

- 不一定预测噪声，可以不同成分建模
- 去噪过程略有不同





Model	Posterior Mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t; s, t)$	
Image Denoising $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	$\frac{\alpha_{t\|s}\sigma_s^2}{\sigma_t^2}\mathbf{z}_t + \frac{\alpha_s\sigma_{t\|s}^2}{\sigma_t^2}\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	预测原图
Noise Prediction $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	$\frac{1}{\alpha_{t\|s}}\mathbf{z}_t - \frac{\sigma_{t\|s}^2}{\alpha_{t\|s}\sigma_t}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	预测噪声
Score-based $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	$\frac{1}{\alpha_{t\|s}}\mathbf{z}_t + \frac{\sigma_{t\|s}^2}{\alpha_{t\|s}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	
Energy-based $E_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	$\frac{1}{\alpha_{t\|s}}\mathbf{z}_t - \frac{\sigma_{t\|s}^2}{\alpha_{t\|s}}\nabla_{\mathbf{z}_t}E_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	
Velocity Prediction $\hat{\mathbf{v}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	$\frac{1-\sigma_{t\|s}^2}{\alpha_{t\|s}}\mathbf{z}_t - \frac{\sigma_{t\|s}^2\alpha_s}{\sigma_t}\hat{\mathbf{v}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	预测速度
Flow-based $\hat{\mathbf{u}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	$\frac{\sigma_t-\sigma_{t\|s}^2}{\alpha_{t\|s}\sigma_t}\mathbf{z}_t - \frac{\sigma_{t\|s}^2\alpha_s}{\sigma_t}\hat{\mathbf{u}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$	

注意，这里的定义与之前稍有不同 $\mathrm{q}(z_t|x) = \mathcal{N}(z_t|\alpha_t x, {\sigma_t}^2\mathbf{I})$

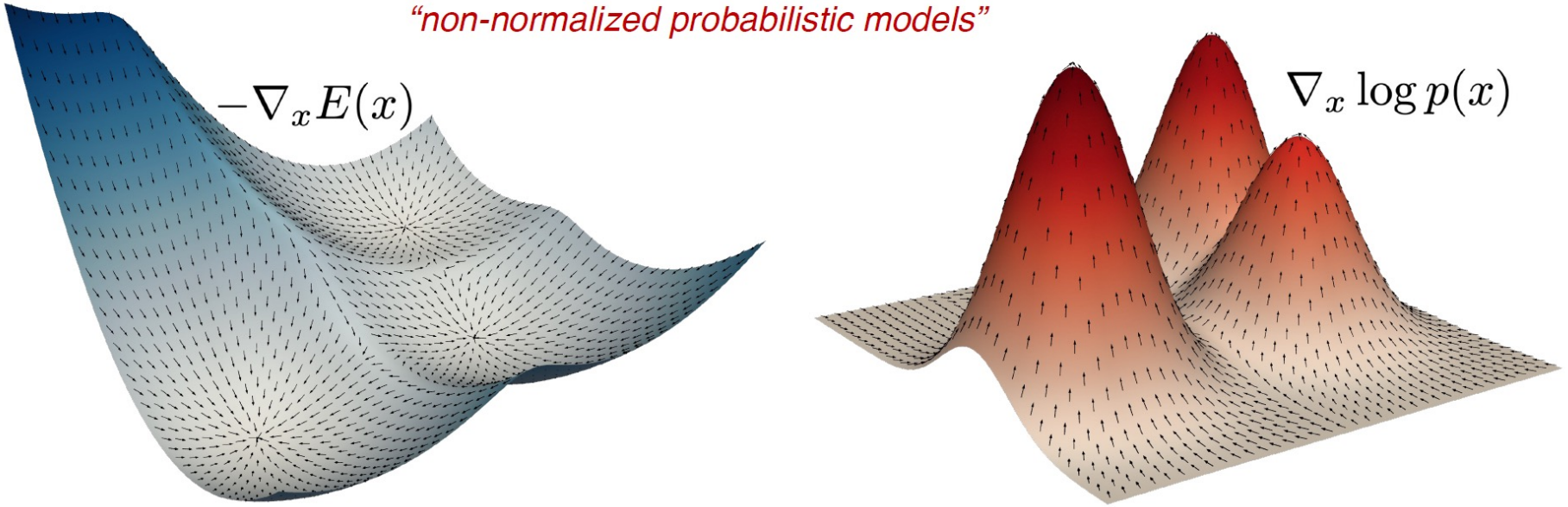
Demystifying Variational Diffusion Models

得分匹配

- 得分函数：概率的对数梯度
- 得分指向能量下降最快的方向

$$\nabla_x \log p(x) = -\nabla_x E(x)$$

“non-normalized probabilistic models”

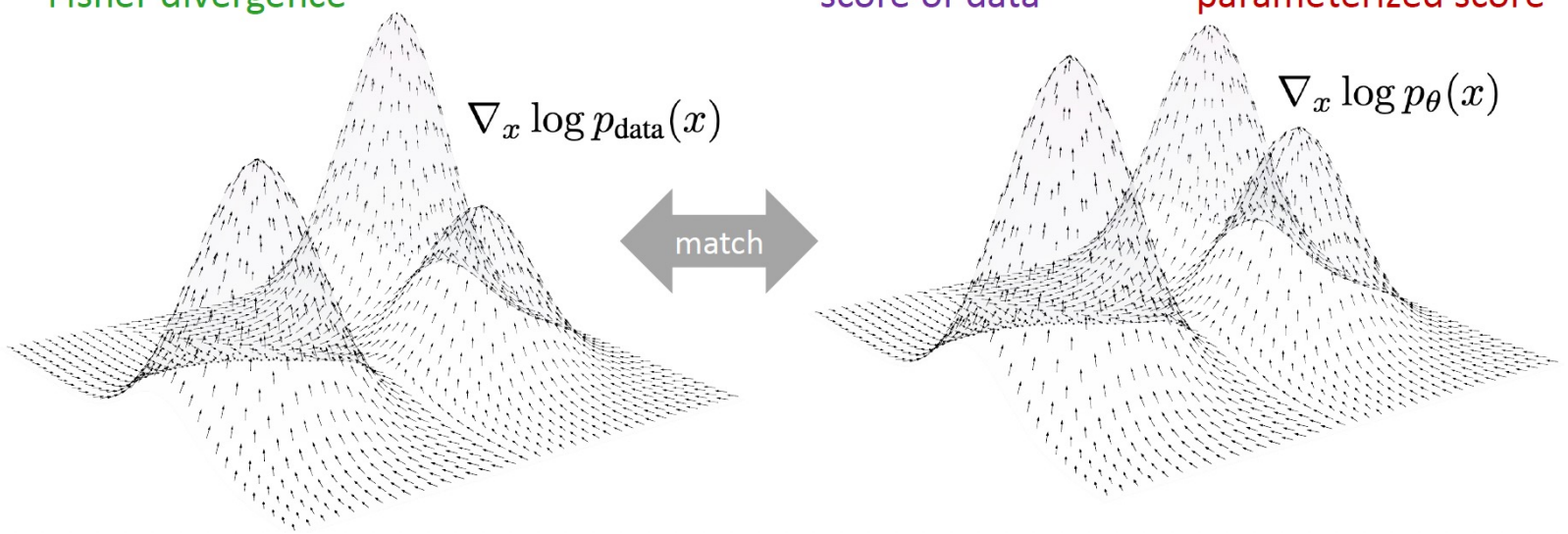


得分匹配

- 建模真实分布难，直接对得分函数进行建模

$$D_F(p_{\text{data}}(x) \,\|\, p_\theta(x)) = \mathbb{E}_{p_{\text{data}}(x)}\left[\frac{1}{2}\|\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x)\|^2\right]$$

Fisher divergence　　score of data　　parameterized score



得分匹配

- 若数据是加噪的形式

$$D_F(q(\tilde{x}) \parallel p_\theta(\tilde{x})) = \mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\left\|\nabla_{\tilde{x}} \log q(\tilde{x} \mid x) - \nabla_{\tilde{x}} \log p_\theta(\tilde{x})\right\|^2\right] + \text{constant}$$

Fisher divergence of noised data

joint distribution

score of conditional

parameterized score

- 如果是高斯噪声，其等价于预测噪声

$$D_F(q(\tilde{x}) \parallel p_\theta(\tilde{x})) = \mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\left\|\nabla_{\tilde{x}} \log q(\tilde{x} \mid x) - \nabla_{\tilde{x}} \log p_\theta(\tilde{x})\right\|^2\right] + \text{constant}$$

a network to predict (negative) noise

Gaussian noise:

$$= \frac{1}{\sigma^2}(x - \tilde{x})$$

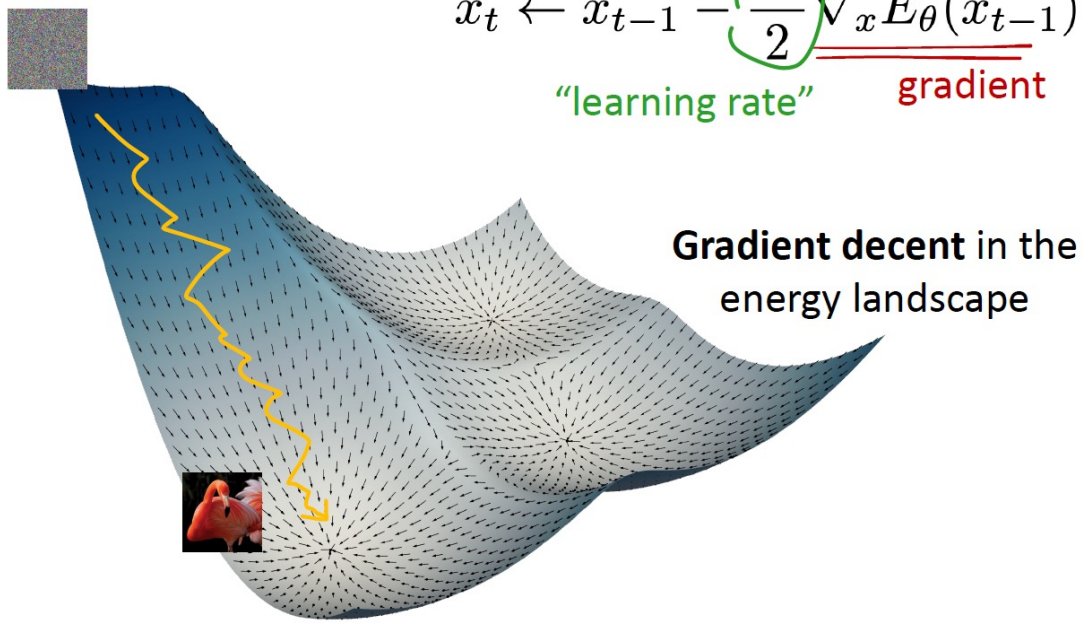
$$-\epsilon$$

得分匹配

- 学习到得分函数之后，可以对样本进行迭代采样
- 提供理论解释：预测噪声 ⇨ 学习得分函数 ⇨ 近似真实数据分布的对数梯度
- 基于得分函数的朗之万动力学：采样时能量最小

$$x_t \leftarrow x_{t-1} - \frac{\sigma^2}{2}\nabla_x E_\theta(x_{t-1}) + \sigma z_t$$

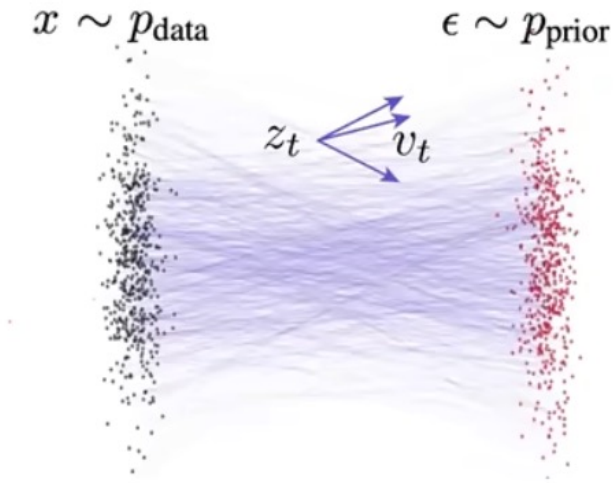
"learning rate" gradient perturbation

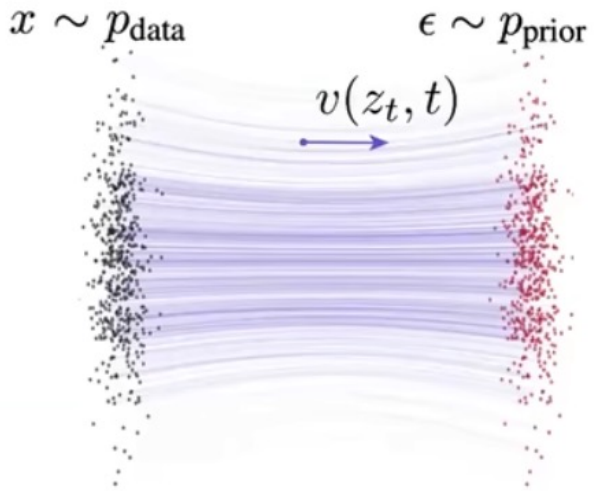


Gradient decent in the energy landscape

流匹配

- 直接学习两个分布之间插值的速度场
- 学习到的是最近路径，并且采样更快





$$z_t = (1-t)x + t\epsilon$$

conditional velocity: $v_t = \epsilon - x$

marginal velocity: $v(z_t, t) \triangleq \mathbb{E}_{p_t(v_t|z_t)}[v_t]$

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}\|v_\theta(z_t, t) - v_t\|^2$$

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\|v_\theta(z_t, t) - v(z_t, t)\|^2$$

多种参数化方法

- 不同参数化方法，是可以相互转换的

Model	Equivalent Reparameterization				
	$\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\hat{\mathbf{v}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\hat{\mathbf{u}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$
Image Denoising $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	-	$\frac{\mathbf{z}_t-\sigma_t\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}}{\alpha_t}$	$\frac{\mathbf{z}_t+\sigma_t^2\mathbf{s}_{\boldsymbol{\theta}}}{\alpha_t}$	$\alpha_t\mathbf{z}_t-\sigma_t\hat{\mathbf{v}}_{\boldsymbol{\theta}}$	$\mathbf{z}_t-\sigma_t\hat{\mathbf{u}}_{\boldsymbol{\theta}}$
Noise Prediction $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\frac{\mathbf{z}_t-\alpha_t\hat{\mathbf{x}}_{\boldsymbol{\theta}}}{\sigma_t}$	-	$-\sigma_t\mathbf{s}_{\boldsymbol{\theta}}$	$\sigma_t\mathbf{z}_t+\alpha_t\hat{\mathbf{v}}_{\boldsymbol{\theta}}$	$\alpha_t\hat{\mathbf{u}}_{\boldsymbol{\theta}}+\mathbf{z}_t$
Score-based $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\frac{\alpha_t\hat{\mathbf{x}}_{\boldsymbol{\theta}}-\mathbf{z}_t)}{\sigma_t^2}$	$\frac{-\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}}{\sigma_t}$	-	$\frac{-\sigma_t\mathbf{z}_t+\alpha_t\hat{\mathbf{v}}_{\boldsymbol{\theta}}}{\sigma_t}$	$\frac{-\alpha_t\hat{\mathbf{u}}_{\boldsymbol{\theta}}+\mathbf{z}_t}{\sigma_t}$
Velocity Prediction $\hat{\mathbf{v}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\frac{\alpha_t\mathbf{z}_t-\hat{\mathbf{x}}_{\boldsymbol{\theta}}}{\sigma_t}$	$\frac{\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}-\sigma_t\mathbf{z}_t)}{\alpha_t}$	$\frac{-\sigma_t(\mathbf{z}_t+\mathbf{s}_{\boldsymbol{\theta}})}{\alpha_t}$	-	$\hat{\mathbf{u}}_{\boldsymbol{\theta}}-\mathbf{z}_t$
Flow-Based $\hat{\mathbf{u}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$	$\frac{\mathbf{z}_t-\hat{\mathbf{x}}_{\boldsymbol{\theta}}}{\sigma_t}$	$\frac{\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\alpha_t-\sigma_t)-\mathbf{z}_t}{\alpha_t}$	$\frac{\sigma_t\mathbf{s}_{\boldsymbol{\theta}}(\sigma_t-\alpha_t)-\mathbf{z}_t}{\alpha_t}$	$\mathbf{z}_t+\hat{\mathbf{v}}_{\boldsymbol{\theta}}$	-

Demystifying Variational Diffusion Models

多种参数化方法

- 不同损失参数化之间本质是样本加权方式不同

Loss	Image Denoising $\|\mathbf{x}-\hat{\mathbf{x}}_\theta(\mathbf{z}_t;t)\|_2^2$	Noise Prediction $\|\boldsymbol{\epsilon}-\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t;t)\|_2^2$	Velocity Prediction $\|\mathbf{v}-\hat{\mathbf{v}}_\theta(\mathbf{z}_t;t)\|_2^2$
$\|\mathbf{x}-\hat{\mathbf{x}}_\theta(\mathbf{z}_t;t)\|_2^2$	1	σ_t^2/α_t^2	σ_t^2
$\|\boldsymbol{\epsilon}-\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t;t)\|_2^2$	α_t^2/σ_t^2	1	$1/\alpha_t^2$
$\|\mathbf{v}-\hat{\mathbf{v}}_\theta(\mathbf{z}_t;t)\|_2^2$	$\sigma_t^2\left(\frac{\alpha_t^2}{\sigma_t^2}+1\right)^2$	$\alpha_t^2\left(\frac{\sigma_t^2}{\alpha_t^2}+1\right)^2$	1

未来研究方向

- 可解释性表征：让表征更加有意义

- 可解释性过程：确定性前向过程建模的因果表征，e.g. 概率流

小结

- 扩散模型自回归模型殊途同归，最终都为了在 latent space 中创造出符合真实数据分布的、内部高度相关的、有意义的结构
- 相比自回归模型，扩散模型灵活性高，易于通过引导控制
- 扩散模型的前向和反向过程为对称的加噪和去噪
- 扩散模型的训练目标也是一种ELBO的变体
- 受益于高斯再生性，不同的扩散模型参数化是可以进行转换的