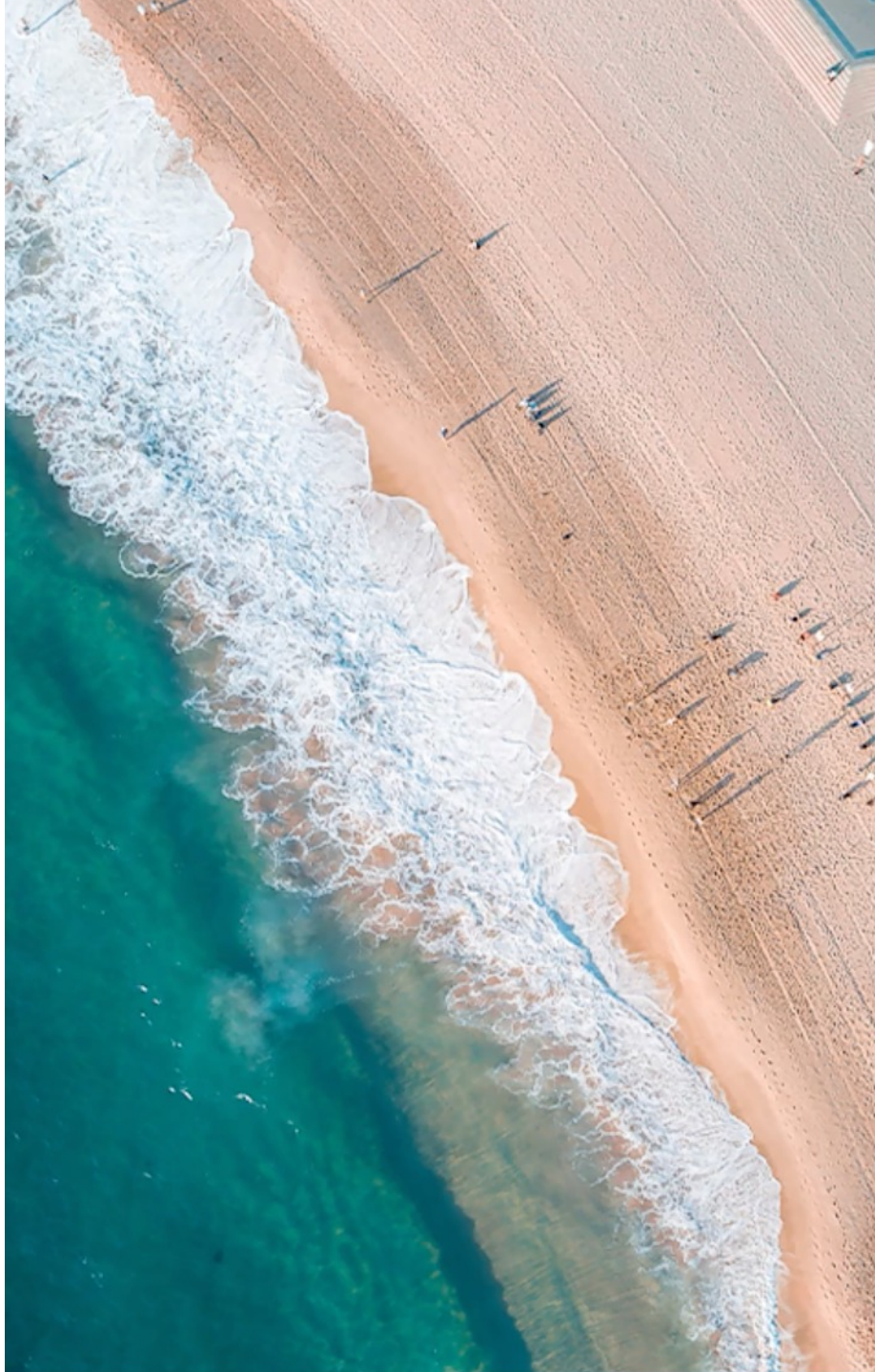
Denoising Diffusion Probabilistic Model

Jonathan Ho, Ajay Jain, Pieter Abbeel

UC Berkeley 2020

2022/06 CUHKSZ



Contents

Disco Diffusion: 强大的AI辅助作图工具

檀州乐

明 陈子龙

山头嵯峨烽火绝，

此时胡雏窥汉月。

明驼快马凌风雪，

帐前健儿沙中血，

回首繁华灯未灭。

Word to Graph

Disco Diffusion: 强大的AI辅助作图工具

Prompt: A beautiful detailed landscape matte painting of blue ocean, by Caspar David Friedrich

Prompt: A digital painting of cyberpunk city by beeple, mist, V-Ray.

Disco Diffusion: 强大的AI辅助作图工具

生成过程：初始画面模糊 -> 逐渐清晰
相同的初始画面和prompt可能生成相似但风格迥异的多种画面

Prompt: 星空下的向日葵花海，Disco Diffusion 生成过程

DDPM: Disco Diffusion 的理论基础

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

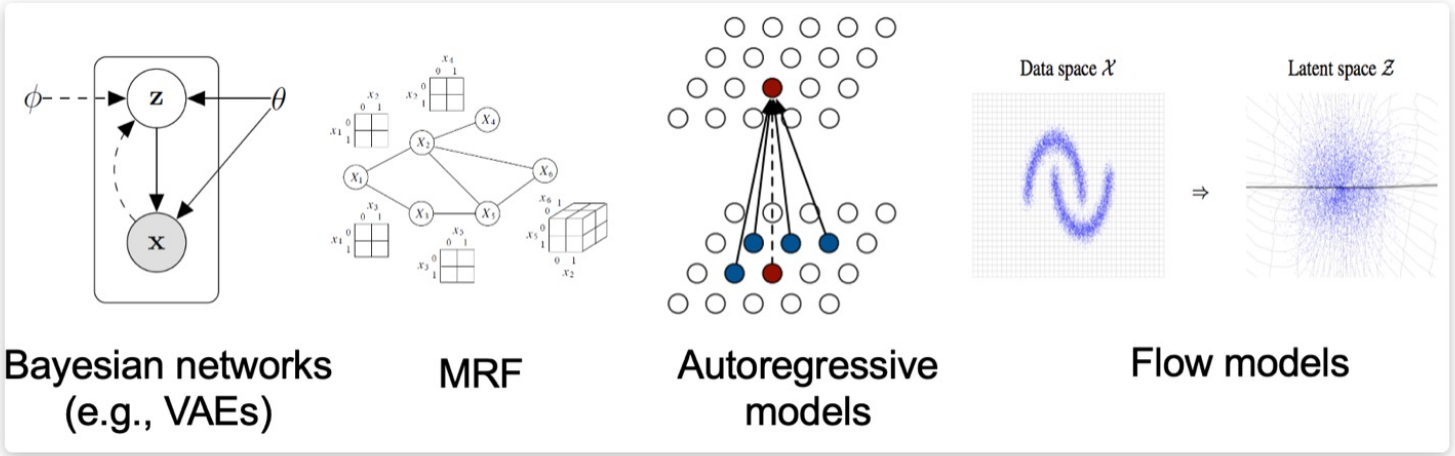
Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at https://github.com/hojonathanho/diffusion.

Key Words
1.高质量图像生成
2.扩散概率模型
3.Langevin 采样

Background: Generative Models

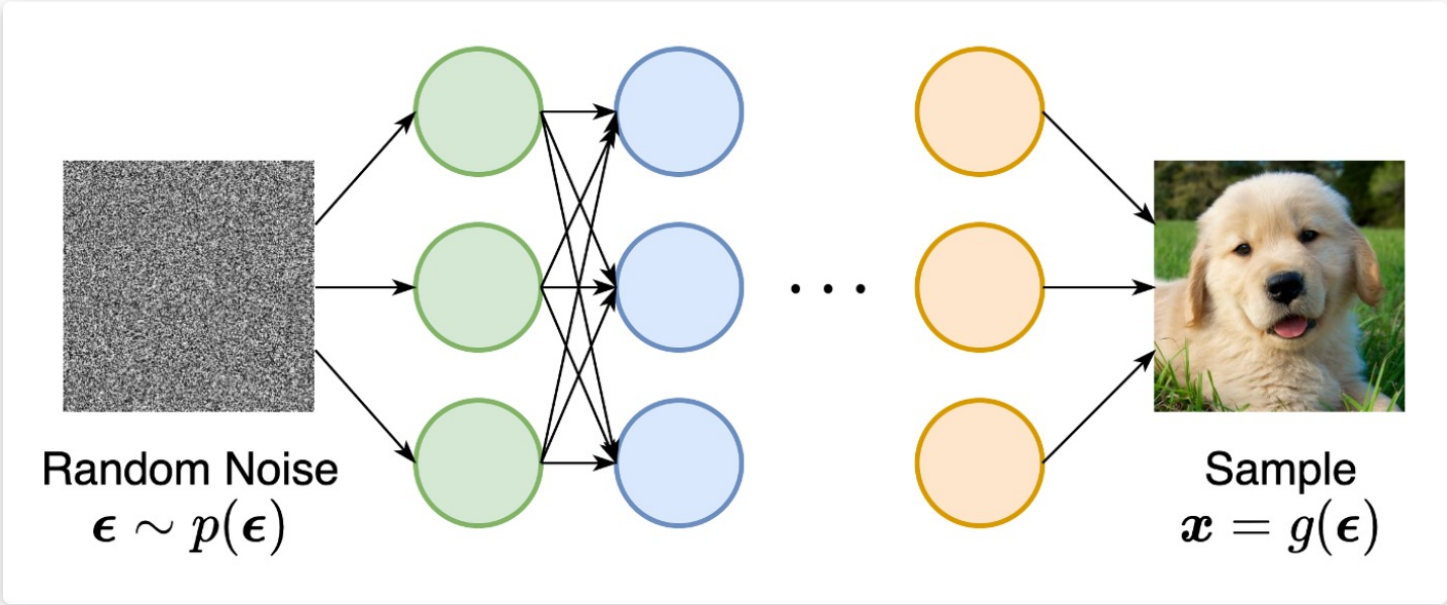
- **基于似然的模型**，通过（近似）最大似然直接学习分布的概率密度（或质量）函数。典型的基于似然的模型包括自回归模型、归一化流模型、基于能量的模型（EBM）和变分自动编码器（VAE）。



贝叶斯网络、马尔可夫随机场（MRF）、自回归模型和归一化流模型都是基于似然的模型的例子。所有这些模型都表示分布的概率密度或质量函数。

- **隐式生成模型**，其中概率分布由其采样过程的模型隐式表示。最突出的例子是生成对抗网络（GAN），其中通过将随机高斯矢量转换为神经网络来合成数据分布中的新样本。

Diffusion Model 属于隐式生成模型，但区别于GAN，Diffusion 具有更好的解释性，可以转化成显式的 score matching 和 Langevin 采样



GAN是隐式模型的一个例子。它隐式表示对生成器网络可以生成的所有对象的分布。

Background: Diffusion Process

首次将热动力学扩散引入机器学习，性能表现不佳
Deep unsupervised learning using nonequilibrium thermodynamics. J. Sohl-Dickstein. ICML 15.

Score-based Model + 注入噪音机制
Generative modeling by estimating gradients of the data distribution. Yang Song. NIPS 19.

基于以上两者，提出高斯噪音注入机制的扩散概率模型
Denoising Diffusion Probabilistic Models. Jonathan Ho. NIPS 20.

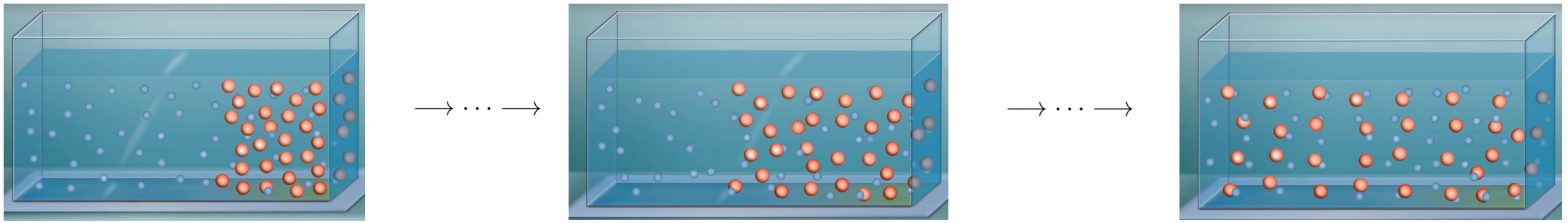
Background: Diffusion Process

扩散（Diffusion）在热力学中指细小颗粒从高密度区域扩散至低密度区域，在统计领域，扩散则指将复杂的分布转换为一个简单的分布的过程。 Diffusion模型定义了一个概率分布转换模型$\mathcal{T}$，能将原始数据x_0构成的复杂分布p_{complex}转换为一个简单的已知参数的先验分布p_{prior}：

$$\mathbf{x}_0 \sim p_{\text{complex}} \implies \mathcal{T}(\mathbf{x}_0) \sim p_{\text{prior}}$$

统计

热力学



Atkins物理化学 | 扩散 Diffusion
https://www.bilibili.com/video/BV1t34y1s7uL?share_source=copy_web

Forward Diffusion: Injecting Noise

受到物理领域的热动力学相关知识启发，Diffusion模型提出可以用马尔科夫链(Markov Chain)来构造T，即定义一系列条件概率分布$q(\mathbf{x}_t|\mathbf{x}_{t-1})$，　$t \in \{1, 2, 3...T\}$，将$\mathbf{x}_0$依次转换为$\mathbf{x}_1$，$\mathbf{x}_2$，…，$\mathbf{x}_T$，希望当$T \to \inf$时，$\mathbf{x}_T \sim p_{\text{prior}}$。

能满足$\mathbf{x}_\infty \sim p_{\text{prior}}$这个期望的$\{q, p_{\text{prior}}\}$组合选择有很多，最简洁有效的选择就是正态分布，即：

非学习参数，人为指定

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

$$q(\mathbf{x}_T) = p_{\text{prior}}(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathrm{I}) \quad where\ T \to \inf$$

即已知$\mathbf{x}_{t-1}$时，$\mathbf{x}_t$的概率分布是一个平均值为$\sqrt{1-\beta_t}\mathbf{x}_{t-1}$，协方差为$\beta_t\mathbf{I}$的正态分布。

根据重参数化技巧可得：　解决随机采样离散样本不能反传梯度问题

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_{t-1} \quad where\ \mathbf{z}_{t-1} \in \mathcal{N}(0, \mathbf{I}) \tag{2}$$

这一过程可以视作$\mathbf{x}_{t-1}$与标准正态分布噪声$\mathbf{z}$混合，扩散率系数β_t控制融合$\mathbf{x}_{t-1}$分布和标准正态分布的混合比例。从原始数据分布$\mathbf{x}_0$到$\mathbf{x}_T$，这一过程可以视作是在重复地给原始数据分布添加噪声，直到变为一个简单固定的分布为止。

问题：\sqrt{1-β}是否为构造 x_T 高斯分布的唯一解？
使用其他形式噪音是否可以？（可以，但高斯噪声包含信息量最大）

Forward Diffusion: Injecting Noise

扩散率β_t到底是什么呢？数据分布混合噪声分布时的比例为什么要设计成$\sqrt{1-\beta_t}$和$\sqrt{\beta_t}$呢?

假设$\alpha_t = 1 - \beta_t$，$\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$，那么：

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} && ;\text{where } \mathbf{z}_{t-1}, \mathbf{z}_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\mathbf{z}_{t-2} && ;\text{where } \bar{\mathbf{z}}_{t-2}, \bar{\mathbf{z}}_{t-3}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (3) \\
&= \cdots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}
\end{aligned}
$$

$z_{t-1}, z_{t-2}, \ldots$ 同属于正态分布

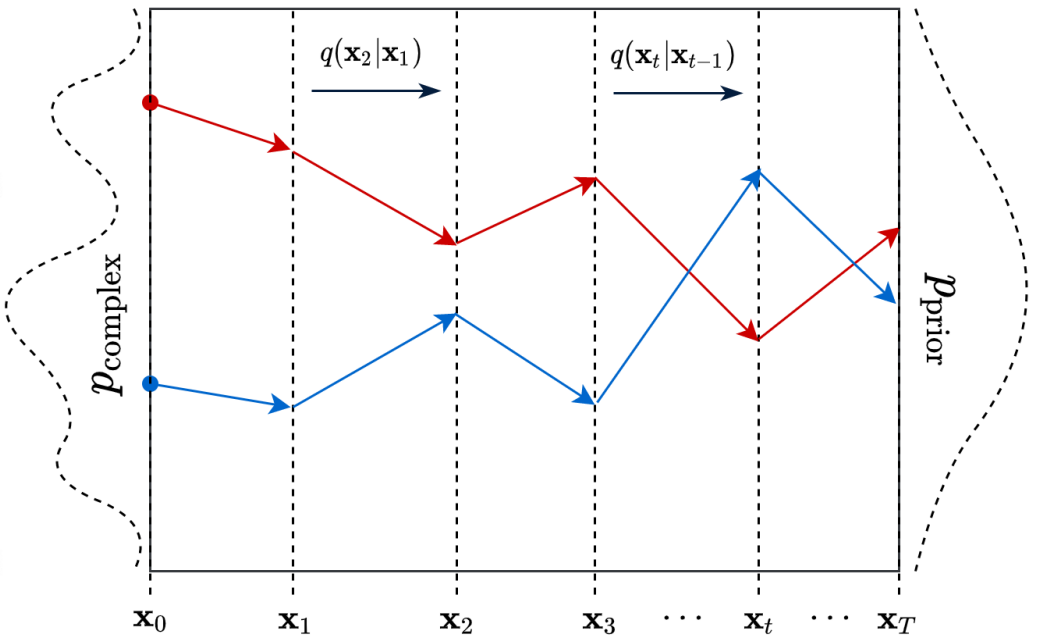
$\left(\sqrt{\alpha_t(1-\alpha_{t-1})}\right)^2 + \left(\sqrt{1-\alpha_t}\right)^2 = 1 - \alpha_t\alpha_{t-1}$

公式中第二行到第三行的转换利用了一个性质：两个正态分布$\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$和$\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$相加，新分布为$\mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$。

将公式3写为条件概率形式，可以得到：

$$
q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \quad (4)
$$

由于$\beta_t \in (0,1)$，那么$\alpha_t \in (0,1)$。$t \to \inf$时，$\bar{\alpha}_t \to 0$。可以看出，$\sqrt{1-\beta_t}$和$\sqrt{\beta_t}$作为系数保证了当$T \to \inf$时，$q(\mathbf{x}_T) = p_{\text{prior}}(\mathbf{x}_T) = \mathbf{N}(0, \mathbf{I})$。实际上，只要$T$取一个足够大的值，不需要无限次迭代，得到的分布就已经很接近于标准正态分布了。



初始复杂分布中采样的两个样本点(红/蓝)，在正向注入噪音过程中的 diffusion trajectory

Forward Diffusion: Injecting Noise

特性1：重参数（reparameterization trick）

重参数技巧在很多工作（gumbel softmax, VAE）中有所引用。如果我们要从某个分布中随机采样(高斯分布)一个样本，这个过程是无法反传梯度的。而这个通过高斯噪声采样得到 x_t 的过程在diffusion中到处都是，因此我们需要通过重参数技巧来使得他可微。最通常的做法是吧随机性通过一个独立的随机变量(ϵ)引导过去。举个例子，如果要从高斯分布 $z \sim \mathcal{N}(z; \mu_\theta, \sigma_\theta^2 \mathbf{I})$ 采样一个z，我们可以写成:

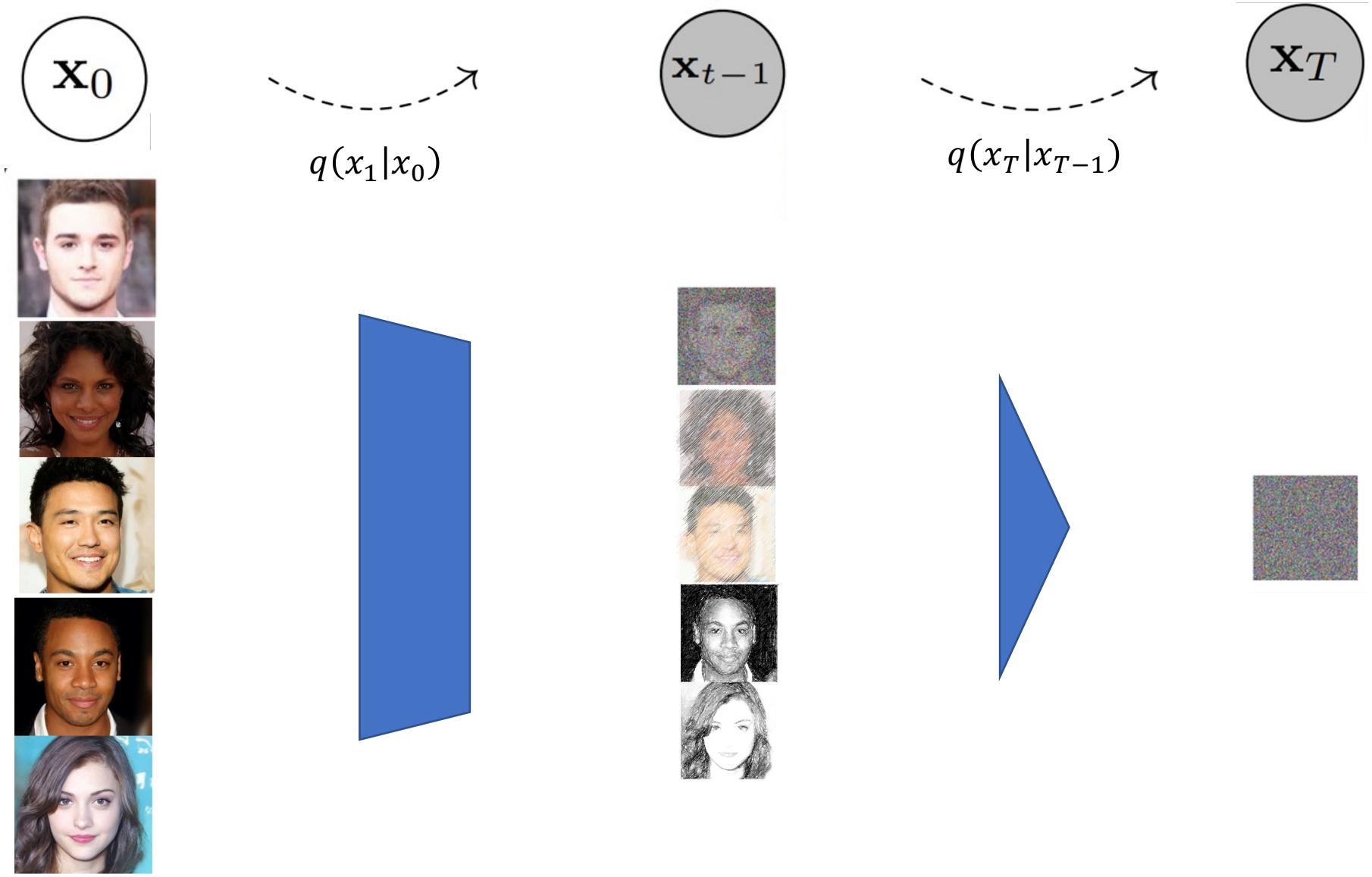
$$z = \mu_\theta + \sigma_\theta \odot \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

上式的z依旧是有随机性的， 且满足均值为 μ_θ 方差为 σ_θ^2 的高斯分布。这里的 μ_θ ， σ_θ^2 可以是由参数 θ 的神经网络推断得到的。整个“采样”过程依旧梯度可导，随机性被转嫁到了 ϵ 上。

通过多次重复单步操作“添加一个噪声”，将复杂的问题解耦为多个简单问题的堆叠，这是 diffusion 简洁有效的核心思想；

正向注入噪音的过程保证了无论什么样的初始输入 x_0 都可以通过这种机制来归一化到相同的最终结果 x_T ~ N(0,I)；

之后只需要将注入噪音过程的 reverse process 表示出来，即可从任意一个正态分布采样结果反向操作得到 x_0，即图片生成。



Reverse Diffusion: Denoising Process

和GAN类似，Diffusion模型的最终目标是从p_{prior}中采样一个样本，将其转换为原始数据分布中的一个样本。显然，如果逆转上一节提到的Diffusion过程，依次从$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$，　$t \in \{T, T-1, T-2 \ldots 0\}$中采样，Diffusion模型就可以实现从$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$到数据分布$p_{\text{complex}}$的转换。即图像生成，只是初始状态并非白板而是随机采样高斯分布

现在问题来了，$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$到底是什么样的分布呢？ Feller等人在1949年证明连续扩散过程的逆转具有与正向过程相同的分布形式。即当扩散率β_t足够小，扩散次数足够多时，离散扩散过程接近于连续扩散过程，$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$的分布形式同$q(\mathbf{x}_t|\mathbf{x}_{t-1})$一致，同样是高斯分布。但是很难直接写出$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$的分布参数。为此，可以用分布$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$来近似$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$：

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{5}$$

其中$\boldsymbol{\mu}_\theta$和$\boldsymbol{\Sigma}_\theta$都是要学习的函数，接受$\mathbf{x}_t, t$作为参数。

学习目标：参数化 reverse Gaussian process 的 期望 和 方差

逆过程与正向扩散过程均为 Gaussian 分布

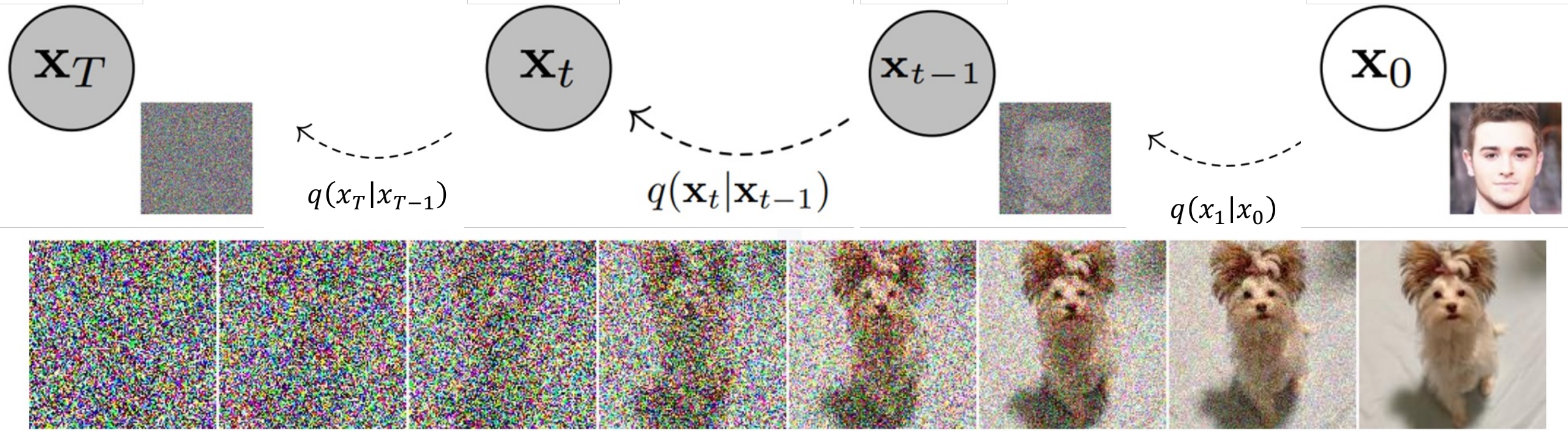
重点来了：**这样的前向过程的每一小步的逆过程都可以近似为高斯分布！**准确地讲，如果我们选取比较合适的 β_t ，使得 x_t 与 x_{t-1} 对应的分布变化非常微小时，我们可以用高斯分布来近似它们的逆过程 $q(x_{t-1}|x_t)$ 。精确的推导需要借助随机微分方程（SDE）的相关知识，可以参考Yang Song发表在ICLR 2021的outstanding paper：Score-Based Generative Modeling through Stochastic Differential Equations。以下我们从不太严谨的直观角度来推导：

$$
\begin{aligned}
q(x_{t-1}|x_t) &= q(x_t|x_{t-1})\exp(\log p_{t-1}(x_{t-1}) - \log p_t(x_t)) \\
&\approx q(x_t|x_{t-1})\exp(\log p_t(x_{t-1}) - \log p_t(x_t)) \\
&\approx q(x_t|x_{t-1})\exp((x_{t-1}-x_t)\nabla_x \log p_t(x_t)) \\
&= \frac{1}{\sqrt{2\pi}\beta_t}\exp\left(\frac{\|x_t - \sqrt{1-\beta_t}x_{t-1}\|_2^2}{2\beta_t^2} + (x_{t-1}-x_t)\nabla_x \log p_t(x_t)\right)
\end{aligned}
$$

其中第一个近似是 $q_t(\cdot) \approx q_{t-1}(\cdot)$ （假设前向过程每一步的变化非常微小），第二个近似是对 $\log p_t(\cdot)$ 做一阶泰勒展开。在给定 x_t 后，括号里的公式可以看成一个关于 x_{t-1} 的二次函数，我们可以通过简单的配方来把它配方成 $\exp\left(\frac{\|x_{t-1}-\cdot\|_2^2}{\cdot}\right)$ 的形式，也就得到了一个高斯分布的形式。这也是为什么diffusion model里经常会见到用一个参数化的高斯分布 $p_\theta(x_{t-1}|x_t)$ 来近似逆向过程。更重要的是，这个高斯分布的均值大部分是解析的，唯一不知道的是 $\nabla_x \log p_t(x_t)$ ，这个函数也被称为“score function”。这也是为什么diffusion model还有个名字是score-based generative model，就是因为逆过程里有score function。具体的推导也可以看Score-Based Generative Modeling through Stochastic Differential Equations。

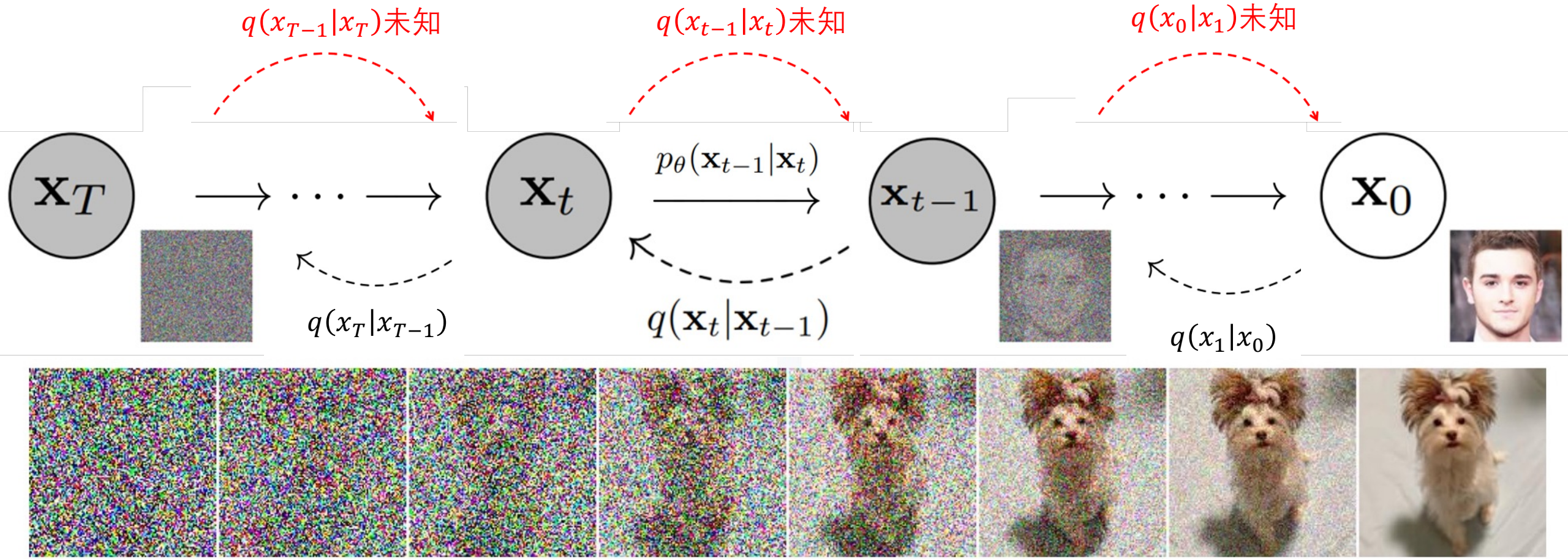
Reverse Diffusion: Denoising Process

正向 diffusion 过程: 不断注入噪音将原始图片转化为纯高斯分布噪音图片



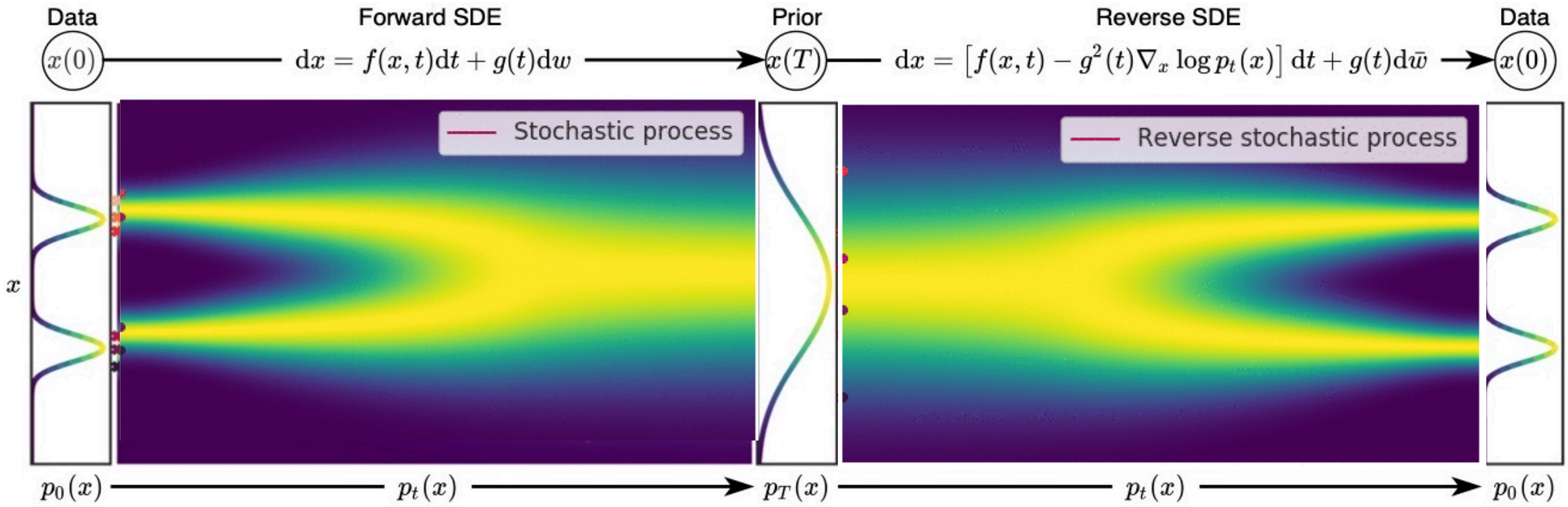
Reverse Diffusion: Denoising Process

逆向 diffusion 过程: 不断去除注入的噪音，将纯高斯分布采样的样本恢复成原始图片，学习参数化的 p 来代替未知逆过程 $q(x_{t-1}|x_t)$



首先从$\mathcal{N}(\mathbf{0},\mathbf{I})$中采样得到$\mathbf{x}_T$，然后在以$\boldsymbol{\mu}_\theta(\mathbf{x}_T,T)$为均值，$\boldsymbol{\Sigma}_\theta(\mathbf{x}_T,T)$为方差的正态分布中采样得到$\mathbf{x}_{T-1}$。
依次重复这个过程，直到得到最终结果$\mathbf{x}_0$。由于$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$未知，所以在逆转Diffusion过程中，用学习到的
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$代替它。

SDE 视角下的完整扩散+去噪过程



另一个例子

前向扩散过程从初始分布添加噪声得到纯高斯分布

逆向扩散过程从高斯分布去除噪声恢复原图像

The forward trajectory

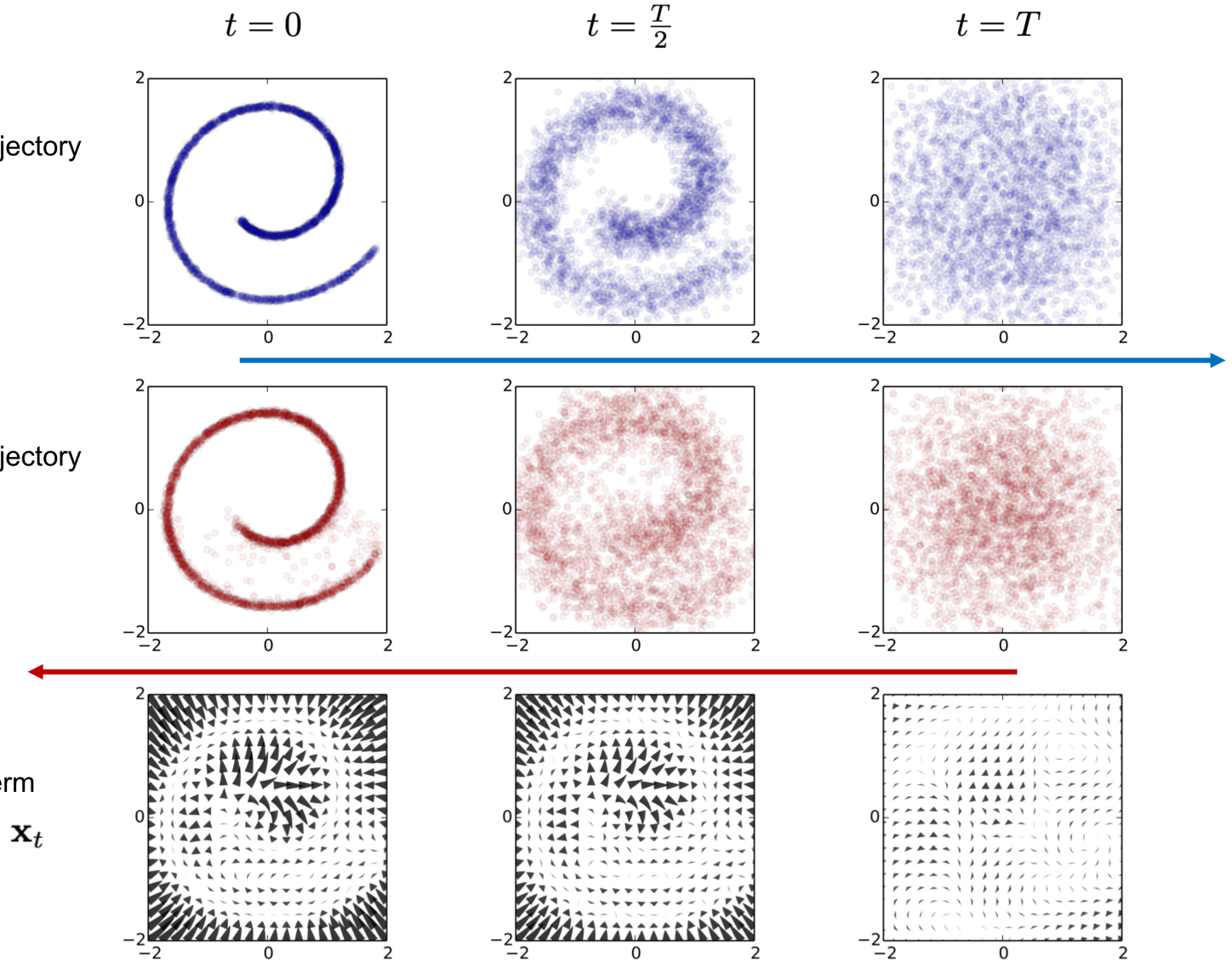
$q(\mathbf{x}_{0:T})$

The reverse trajectory

$p_\theta(\mathbf{x}_{0:T})$

The drifting term

$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t$



An example of training a diffusion model for modeling a 2D swiss roll data.
(Image source: Sohl-Dickstein et al., 2015)

优化目标

现在只剩下最后一个问题，究竟怎么优化得到理想的$\boldsymbol{\mu}_\theta$和$\boldsymbol{\Sigma}_\theta$？类似于其它生成模型，可以最小化在真实数据期望下，模型预测分布的负对数似然，即最小化预测$p_{\text{data}} = q(\mathbf{x}_0)$和$p_\theta(\mathbf{x}_0)$的交叉熵：

即比较模型使用当前参数化操作逆向还原的**生成图片结果 p(x_0)** 与**真实图片 q(x_0)** 之间的差异

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0)} \left[-\log p_\theta(\mathbf{x}_0) \right]$$

优化目标

$$
\begin{aligned}
\mathcal{L} &= \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0)} \big[-\log p_\theta(\mathbf{x}_0) \big] \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \right) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} \right) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right) \\
&\leq -\mathbb{E}_{q(\mathbf{x}_{0:T})} \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right]
\end{aligned}
$$

Jensen 不等式

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

涉及概率密度函数的形式

假设Ω是实线的可测子集，f（x）是一个非负函数

$$\int_{-\infty}^{\infty} f(x)\,dx = 1$$

在概率语言中，f是概率密度函数。

然后Jensen的不等式变成了关于凸积分的下面的陈述：

如果g是任何实值可测函数且ϕ在g的范围内是凸的，那么：

$$\varphi\left(\int_{-\infty}^{\infty} g(x) f(x)\,dx\right) \leqslant \int_{-\infty}^{\infty} \varphi\left[g(x)\right] f(x)\,dx$$

如果g（x）=x，那么这种不等式的形式可以简化为一个常用的特例：

$$\varphi\left(\int_{-\infty}^{\infty} x\, f(x)\,dx\right) \leq \int_{-\infty}^{\infty} \varphi(x)\, f(x)\,dx.$$

优化目标转为最小化交叉熵损失上界 L_{VLB}

$$
\begin{aligned}
L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\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{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}] + \sum_{t=2}^{T}\underbrace{D_{\mathrm{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{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_T))}_{L_T}
\end{aligned}
$$

考虑到第一个步骤是从离散样本x_0到连续分布x_1，单独拆分 $q(x_1|x_0)$

中间项两两相消

整合

上式中蓝色部分直接的变换实际上利用了贝叶斯公式：

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

注意由马尔科夫链的性质，有$q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1})$。

另一种推导思路

$$\mathcal{L} = \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)].$$ ★

这个过程很像VAE，即可以使用变分下限(VLB)来优化负对数似然。由于KL散度非负，可得到：

$$\begin{aligned}
-\log p_\theta(x_0) &\leq -\log p_\theta(x_0) + D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0)) \\
&= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\right]; \qquad \text{where} \quad p_\theta(x_{1:T}|x_0) = \frac{p_\theta(x_{0:T})}{p_\theta(x_0)} \\
&= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \underbrace{\log p_\theta(x_0)}_{\text{与}q\text{无关}}\right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]. \quad ☺
\end{aligned}$$

对(☺)左右取期望 $\mathbb{E}_{q(x_0)}$ ，利用到重积分中的Fubini定理：

$$\mathcal{L}_{VLB} = \underbrace{\mathbb{E}_{q(x_0)}\left(\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]\right) = \mathbb{E}_{q(x_{0:T})}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]}_{Fubini\text{定理}} \geq \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)].$$

能够最小化 $\mathcal{L}_{VLB}$ 即可最小化我们的目标损失 ★

优化目标: 交叉熵等价于参数化逆过程与真实未知逆过程的 KL 散度

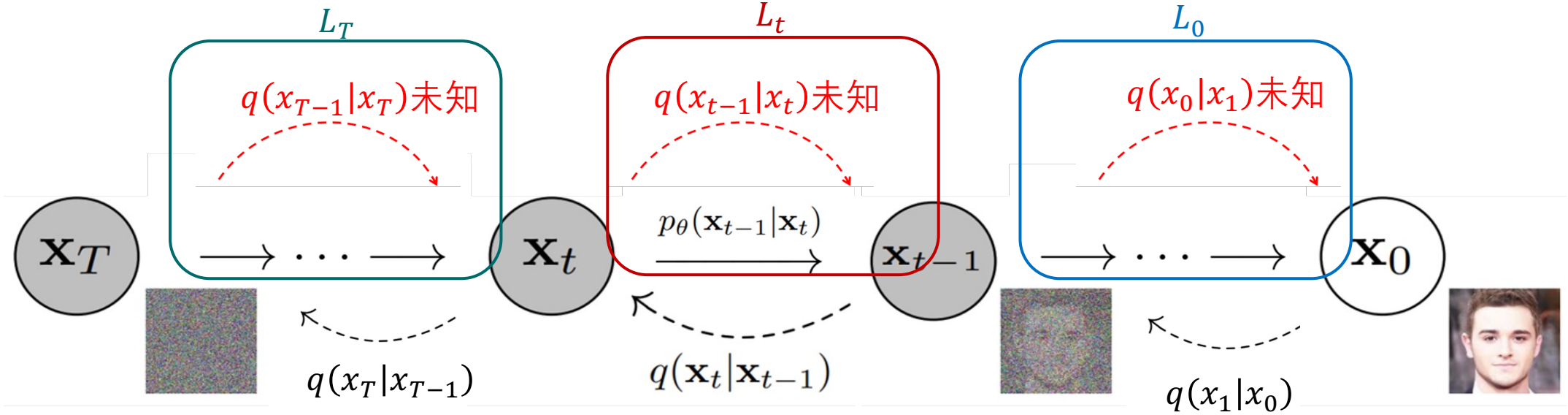
$$L_{\mathrm{VLB}} = \mathbb{E}_q[\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}] + \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} + \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)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

再重新看公式10的最后一行，可以看出L_{VLB}实际上由一个熵（L_0），以及多个KL散度($L_t, t \in \{1,2,3,\cdots,T\}$)构成。其中$L_T$中$\mathbf{x}_T$和$\mathbf{x}_0$一个是先验分布，一个是数据分布，都是固定的，故$L_T$是一个常数，最小化$L_{VLB}$时可以忽略。可以只去研究$L_0$和$L_t, t \in \{1,2,3,\cdots,T-1\}$。

分布$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$和分布$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$之间的KL散度

根据公式5，分布$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$是一个高斯分布，其平均值和方差由Diffusion模型网络预测产生。

实际上就是将复杂的生成问题解耦为优化学习到的逆过程与真实逆过程之间的 KL 散度！



简化目标 L_t

而分布$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$可以根据贝叶斯定律，

$$
\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&= q(\mathbf{x}_t|\mathbf{x}_{t-1})\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&\propto \exp\Big(-\frac{1}{2}\big(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\big)\Big)
\end{aligned}
\tag{12}
$$

继续推导下去，可以发现$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$同样是一个高斯分布。假设：

$$
q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \tag{13}
$$

那么，由公式12，公式13中的两个新变量：

$$
\begin{aligned}
\tilde{\beta}_t &= 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t \\
\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0
\end{aligned}
\tag{14}
$$

简化目标 L_t

因此，最小化L_t这个KL损失实际上目标就是拉近下面这两个高斯分布的距离：

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}) \longleftrightarrow p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{15}$$

多元正态分布之间的KL散度可以直接根据分布参数计算出来。

$$L_t = \mathbb{E}_q \Big[\frac{1}{2 \| \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) \|_2^2} \| \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \|^2 \Big] + C \tag{16}$$

上式中C是一个不依赖于θ的常数。为了模型简单，可以令$\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$，其中$\sigma_t^2$可以设置为$\beta_t$或$\tilde{\beta}_t$，论文说这两个选择效果差不多。实际上，当$\sigma_t^2 = \tilde{\beta}_t$时，公式15中的两个分布的方差就一样了。这一选择是为了简化计算，并不是唯一的。

从公式16来看，只需要定义一个网络$\mu_\theta(\mathbf{x}_t, t)$，使用L2损失约束其预测值同$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)$一致即可。具体来说，可以定义一个接受$x_t$和$t$作为参数的网络，从原数据分布中采样一个数据$x_0$，通过公式3计算得到$x_t$，然后利用公式14计算得到$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)$，将$x_t$和$t$送入网络得到$\mu_\theta(\mathbf{x}_t, t)$。使用L2损失约束两个样本一致，并优化网络。

$$L_t = \mathbb{E}_q\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0)-\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right]+C \tag{16}$$

但DDPM并没有停止于此，继续分析化简公式16。$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0)$的输入有$\mathbf{x}_t,\mathbf{x}_0$，而$\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)$以$x_t$作为输入。借助公式3，可以得到$\mathbf{x}_0=\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\mathbf{z}_t)$。将其代入，有：

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\mathbf{z}_t)\\ &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right)\end{aligned} \tag{17}$$

根据公式16和公式17，$\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)$在给定$\mathbf{x}_t$的情况下，需要预测出$\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right)$。为了降低学习的难度，可以直接定义：

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t,t)\right) \tag{18}$$

这样，公式16可以继续简化：

$$\begin{aligned}L_t-C &= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{1}{2\sigma_t^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0)-\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\right]\\ &= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{1}{2\sigma_t^2}\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right)-\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t,t)\right)\|^2\right]\\ &= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\sigma_t^2}\|\mathbf{z}_t-\mathbf{z}_\theta(\mathbf{x}_t,t)\|^2\right]\\ &= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\sigma_t^2}\|\mathbf{z}_t-\mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\mathbf{z}_t,t)\|^2\right]\end{aligned} \tag{19}$$

公式19表示在优化时，采样$\mathbf{x}_0\sim\mathbf{p}_{data}$和$\mathbf{z}_t\in\mathcal{N}(0,\mathbf{I})$，后计算$\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\mathbf{z}_t$，然后联合时间$t$，送入$\mathbf{z}_\theta$，得到预测值，约束其与$\mathbf{z}_t$一致。

计算目标 L_0

已知$L_0 = -\mathbb{E}_{\mathbf{x}_0,\mathbf{x}_1} \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$，而$p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_1,1), \sigma_1^2\mathbf{I})$。所以$L_0$实际上是一个多元高斯分布的负对数似然的期望，即其熵。多元高斯分布的熵仅与其协方差有关，即L_0仅与$\sigma_1^2\mathbf{I}$有关，L_0是一个常数。

然而，论文DDPM指出，一般而言，$\mathbf{x}_0$的分布实际上是离散的，而不是连续的。比如图片数据，像素值取值必须是整数，归一化到$[-1,1]$后，依然是离散的点。Diffusion前向的第一步实际上是为离散数据添加噪声。那么，逆Diffusion的最后一步，即从$\mathbf{x}_1$到$\mathbf{x}_0$，也不能被简单地看作从$\mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_1,1), \sigma_1^2\mathbf{I})$中采样，而是在从$\mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_1,1), \sigma_1^2\mathbf{I})$采样的基础上再加上离散化操作。$L_0$也不再是一个常数，而是一个与$\boldsymbol{\mu}_\theta(\mathbf{x}_1,1)$相关的积分，其具体表达式可以参考DDPM的Sec3.3。在忽略σ_1^2和边缘效应后，L_0的取值可以被$\mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_1,1), \sigma_1^2\mathbf{I})$的密度函数与离散时的分块大小(bin width)相乘所拟合。

另外值得一提的是，逆Diffusion的最后一步，DDPM直接取$\boldsymbol{\mu}_\theta(\mathbf{x}_1,1)$作为$\mathbf{x}_0$。

进一步简化 Loss

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

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

L_0仅与$\sigma_1^2\mathbf{I}$有关，L_0是一个常数。

L_T中$\mathbf{x}_T$和$\mathbf{x}_0$一个是先验分布，一个是数据分布，都是固定的，是一个常数。

上文已经分别描述了$L_t, t \in \{0,1,2,3,\cdots,T-1\}$的计算过程，最终可以按照公式10，最小化$L_0 + \sum_{t=1}^{T-1} L_t$来优化网络。论文DDPM发现，去除$L_t$中的加权系数$\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\sigma_t^2}$，得到简化的训练目标如下：

$$L_{\mathrm{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\epsilon_t}\left[\|\epsilon_t - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\|^2\right] \tag{20}$$

公式中t从$\{1,2,\cdots,T\}$中均匀采样。$t=1$时对应于L_0的一个近似，$t>1$时对应于去除了加权系数的公式19。

相对于直接计算L_{VLB}，L_{simple}实现起来更加简单，t较小时的L_t权重被减少，t较大时的权重被增加。这样网络能更专注于t较大，图片中噪声更多时，更难更复杂的噪声预测任务。

训练过程

可以将上文描述的Diffusion模型的训练采样过程分别总结如下：

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

训练时，分别从$q(\mathbf{x}_0)$、$Uniform(1, \cdots, T)$、$\mathcal{N}(\mathbf{0}, \mathbf{I})$中采样得到$x_0$，$t$和$\epsilon$，利用公式3计算得到$x_t$，将$x_t$和$t$送入网络，预测得到一个噪声。最小化预测噪声和真实采样的ϵ之间的距离。重复这一过程直到网络收敛。

Diffusion模型的逆转采样每个时刻主要包含以下三步：

1. 将x_t和t送入网络，预测得到噪声ϵ
2. 利用估计的噪声ϵ和x_t，计算$\mu_\theta = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right)$
3. 如果$t > 1$，需要从$\mathcal{N}(\mu_\theta, \sigma_t^2 \mathbf{I})$中采样得到$x_{t-1}$，利用重参数化技巧，可以将采样过程转换为首先采样$z \in \mathcal{N}(\mathbf{0}, \mathbf{I})$，然后计算$x_{t-1} = \mu_\theta + \sigma_t z$。如果$t = 1$，直接令$x_0 = \mu_\theta$

Connections with Score-based Models

GAN不稳定，缺少多样性，由于天然的对抗训练性质

GAN: Adversarial training

VAE 依赖于 surrogate loss

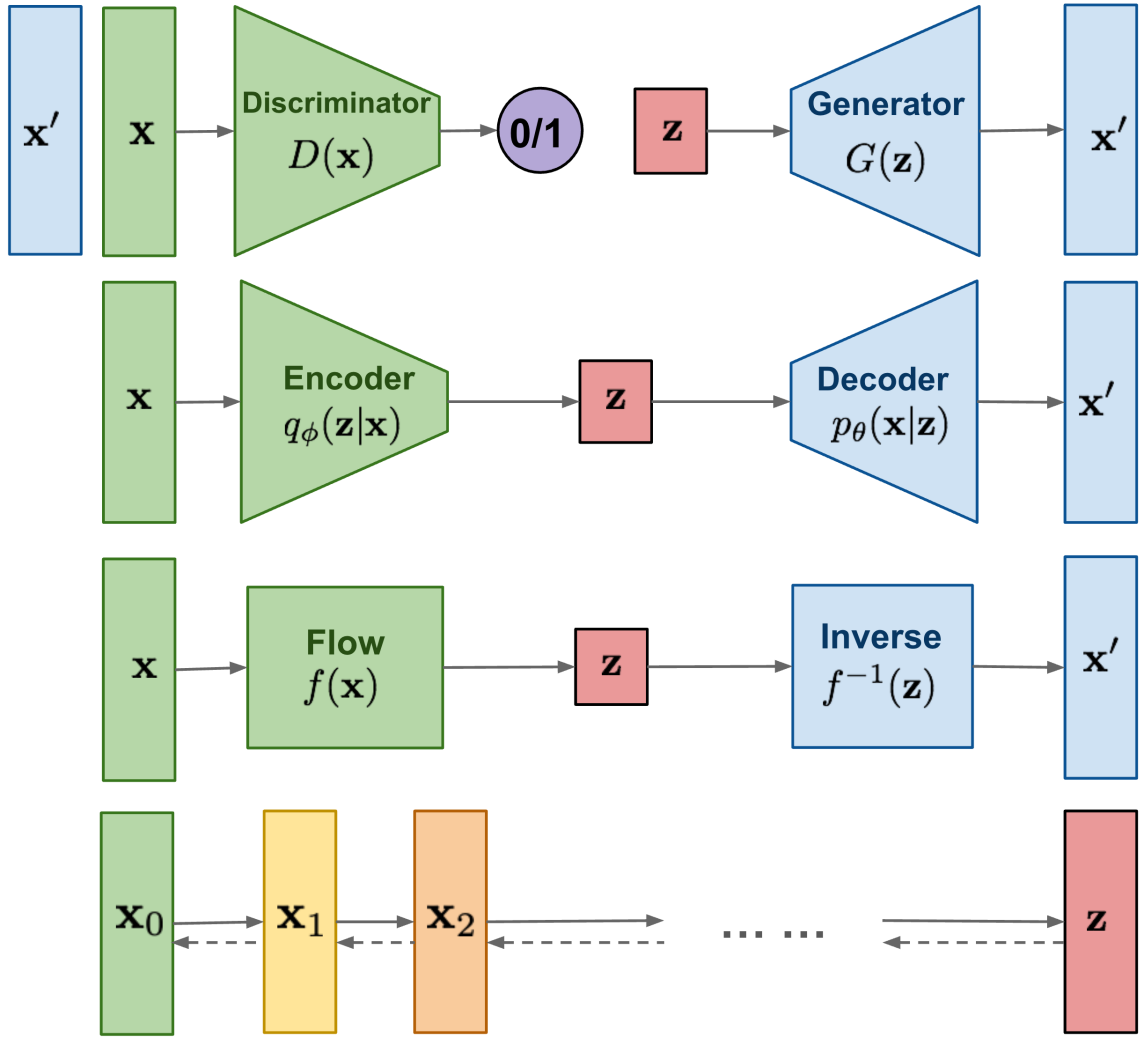
VAE: maximize variational lower bound

Flow model 需要使用特殊结构来构建可逆的转换过程 f(x)

Flow-based models: Invertible transform of distributions

而diffusion使用多步添加噪音的扩散-逆扩散过程，具有 fixed procedure，并且 latent variable 具有高 dimensionality, same as origin data

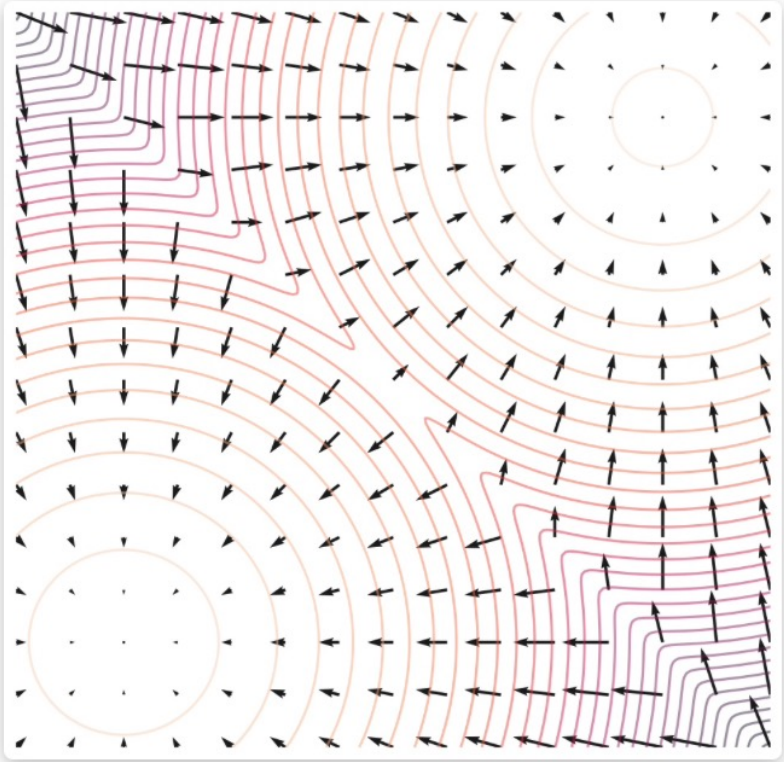
Diffusion models: Gradually add Gaussian noise and then reverse



Score based Generative Models

Diffusion Model可以避免以上限制，关键想法是建模 log probability 的梯度，也被称为 Stein score function，这类 score-based model 不需要 tractable normalizing constant，并且可以直接通过 score matching 学习。

并且score based model 与 normalized flow models 相关，允许精确计算 likelihood 和特征学习。同时建模和估计 score 可以促进 reverse 过程求解，在色彩填充，压缩感知，医疗影像 MRI 重构上有很大应用。



Score function (the vector field) and density function (contours) of a mixture of two Gaussians.

Score based Generative Models

假设我们有一个数据集 $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ 每个样本都是从底层分布中独立采样
生成模型的目标是拟合该数据分布模型，以此来生成新的服从该分布的数据点。
通常采用一个参数化函数 $f_\theta(\mathbf{x}) \in \mathbb{R}$
并进行归一化来表示 pdf

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta},$$

where $Z_\theta > 0$ is a normalizing constant dependent on θ, such that $\int p_\theta(\mathbf{x})\mathbf{dx} = 1$.

函数 f 就是未归一化概率模型，或称为 energy-based model，通常最大化 log-likelihood 来训练 p

$$\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i).$$

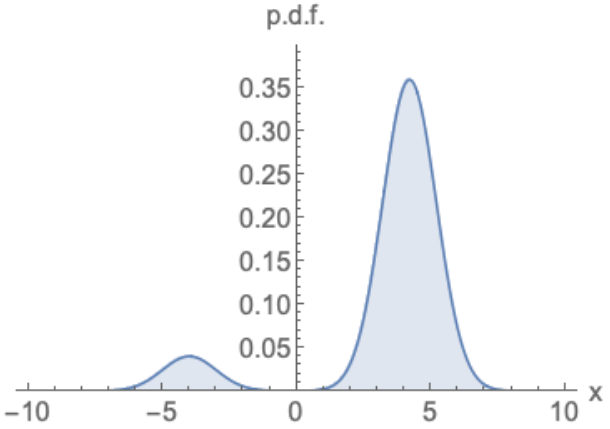
观察式子发现这里的 p 是归一化后的 pdf，想要计算 p 的话必须计算归一化常数 Z, 通常来讲是 intractable 的。为此，likelihood 方法必须限制模型的结构来实现 Z tractable（自回归模型的 causal convolutions，流模型的可逆网络），或者花费大量计算逼近归一化常数（VAE里的 variational inference，contrastive divergence 的 MCMC采样）

Score based Generative Models

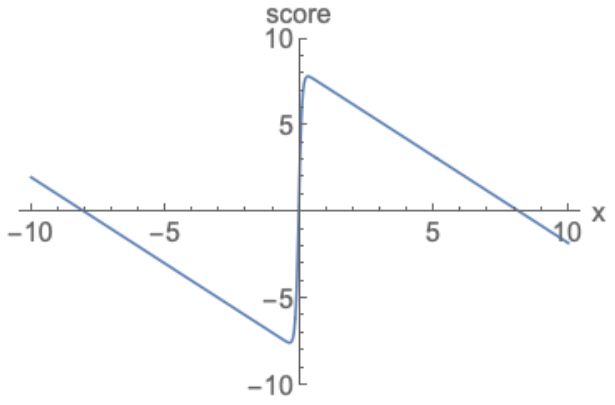
那么如何绕开这个归一化常数（分母）呢？一个自然的想法是对分式取 log 拆分 f 和 Z，然后对 Z 的无关变量求偏导消去 Z。
因此，定义分布的 score function

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

这样我们无需计算 Z，极大地提高了可使用的模型种类，无需任何特定结构来满足归一化常数 tractable



Parameterizing probability density functions. No matter how you change the model family and parameters, it has to be normalized (area under the curve must integrate to one).



Parameterizing score functions. No need to worry about normalization.

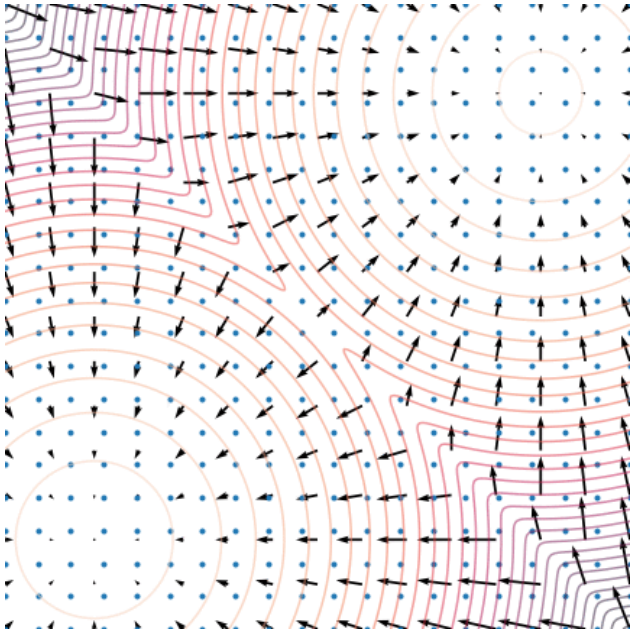
然后我们最小化 Fisher-divergence，比较 squared l2 距离，$\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$
由于 data score 未知无法计算，我们使用 score matching 最小化而无需知道 ground truth data

Langevin Dynamics

获得了 score-based model 后，我们可以使用 iterative procedure 称为 朗之万动力学 Langevin dynamics 进行采样
这是一个MCMC过程，只使用分数函数，初始化了一个从任意初始分布开始的 chain，迭代的采样

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

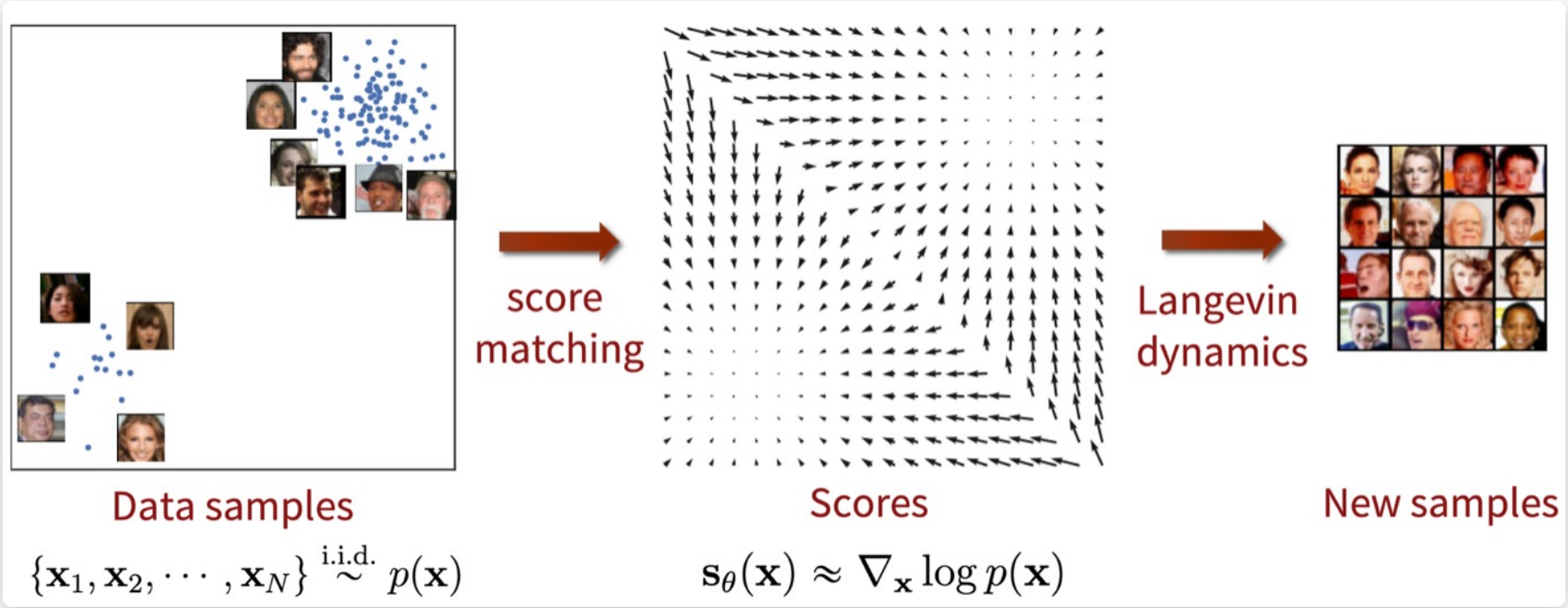
Z是高斯分布，当K无穷大而epsilon=0，x_K 收敛到分布在某些 regularity conditions 下的采样，



Using Langevin dynamics to sample from a mixture of two Gaussians

与标准SGD相比，随机梯度 Langevin 动力学将高斯噪声注入参数更新，以避免陷入局部最小值。

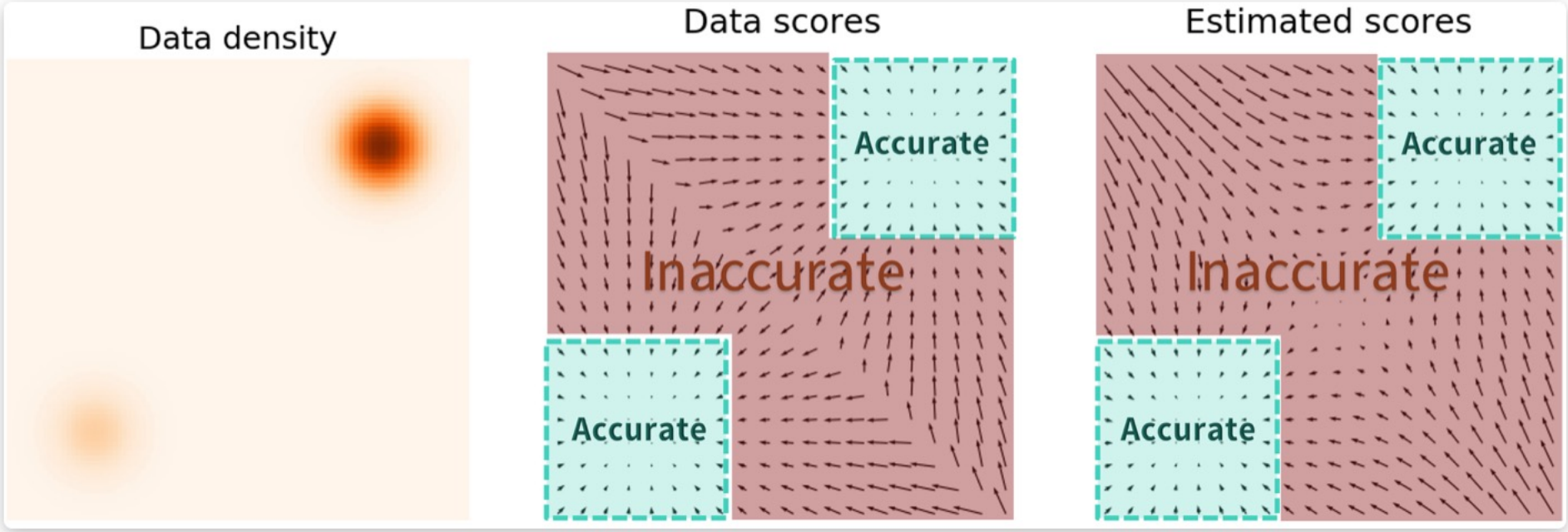
Naïve Score Based Model



Score-based generative modeling with score matching + Langevin dynamics.

那么图像生成问题就分解为两步，首先通过 score matching 估计出分布的 score manifold，然后利用 Langevin dynamics 采样得到新的图像服从该分布。

Pitfall in Naïve Score Based Model



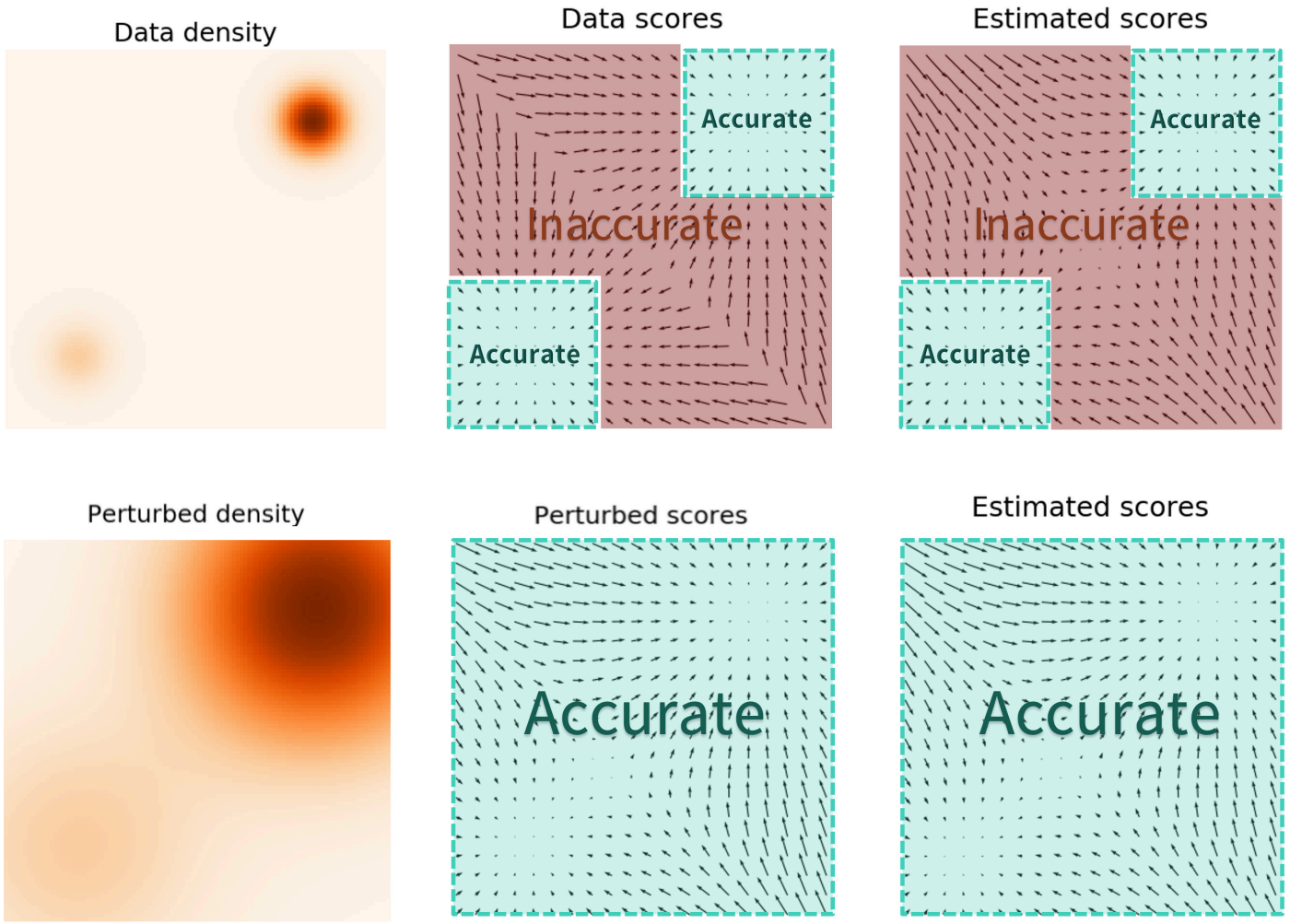
Estimated scores are only accurate in high density regions.

但是这里有个陷阱，score function model 通常不准，尤其是样本数据少的 low density regions，又因为计算 fisher divergence 时要与分布加权，低密度区域的估计误差被进一步忽略了，但是 Langevin dynamic 极度受到初始采样正确性的影响，而初始采样大概率在低密度 inaccurate 区域，反过来导致了低质量的图像生成。

$$\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2] = \int p(\mathbf{x})\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2 \mathrm{d}\mathbf{x}.$$

Score-based Models with multiple noise perturbations

既然原始数据在低密度区样本少难以准确训练，一个自然的想法是：注入噪音人为增加样本，当噪音尺度足够大，可以提升低密度区估计 scores 的准确性



Score-based Models with multiple noise perturbations

这很有效，但面临另一个问题：如何选择合适的噪声尺度？
- 大的噪声可以 cover 低密度区获得更好的 score 估计，但是覆盖了数据的原始分布信息；
- 小的噪声带来更少的覆盖，但没有很有效的覆盖低密度区

一个自然的想法是：同时使用多个尺度噪音扰动，这就是 NSCN 19的核心想法。
假设我们总是使用 isotropic 各向同性高斯噪声，设定 L 档逐渐增加的高斯噪声方差 $\sigma_1 < \sigma_2 < \cdots < \sigma_L$.
分别得到多个 noise-perturbed distribution

$$p_{\sigma_i}(\mathbf{x}) = \int p(\mathbf{y}) \mathcal{N}(\mathbf{x}; \mathbf{y}, \sigma_i^2 I) \mathrm{d}\mathbf{y}.$$

Note that we can easily draw samples from $p_{\sigma_i}(\mathbf{x})$ by sampling $\mathbf{x} \sim p(\mathbf{x})$ and computing $\mathbf{x} + \sigma_i \mathbf{z}$, with $\mathbf{z} \sim \mathcal{N}(0, I)$.

现在我们可以在噪音扰动的分布上训练一个 noise conditional score-based model s_theta(x,i)，也称为NCSC

$$\mathbf{s}_\theta(\mathbf{x}, i) \approx \nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}) \text{ for all } i = 1, 2, \cdots, L.$$

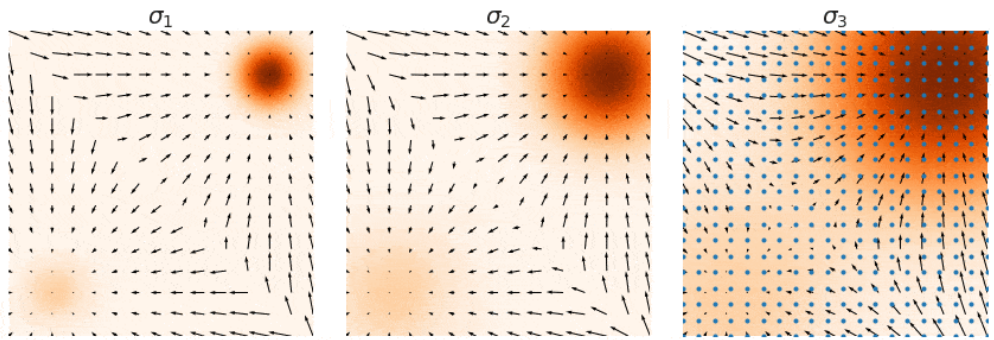
Generative modeling by estimating gradients of the data distribution. Yang Song. NIPS 19.

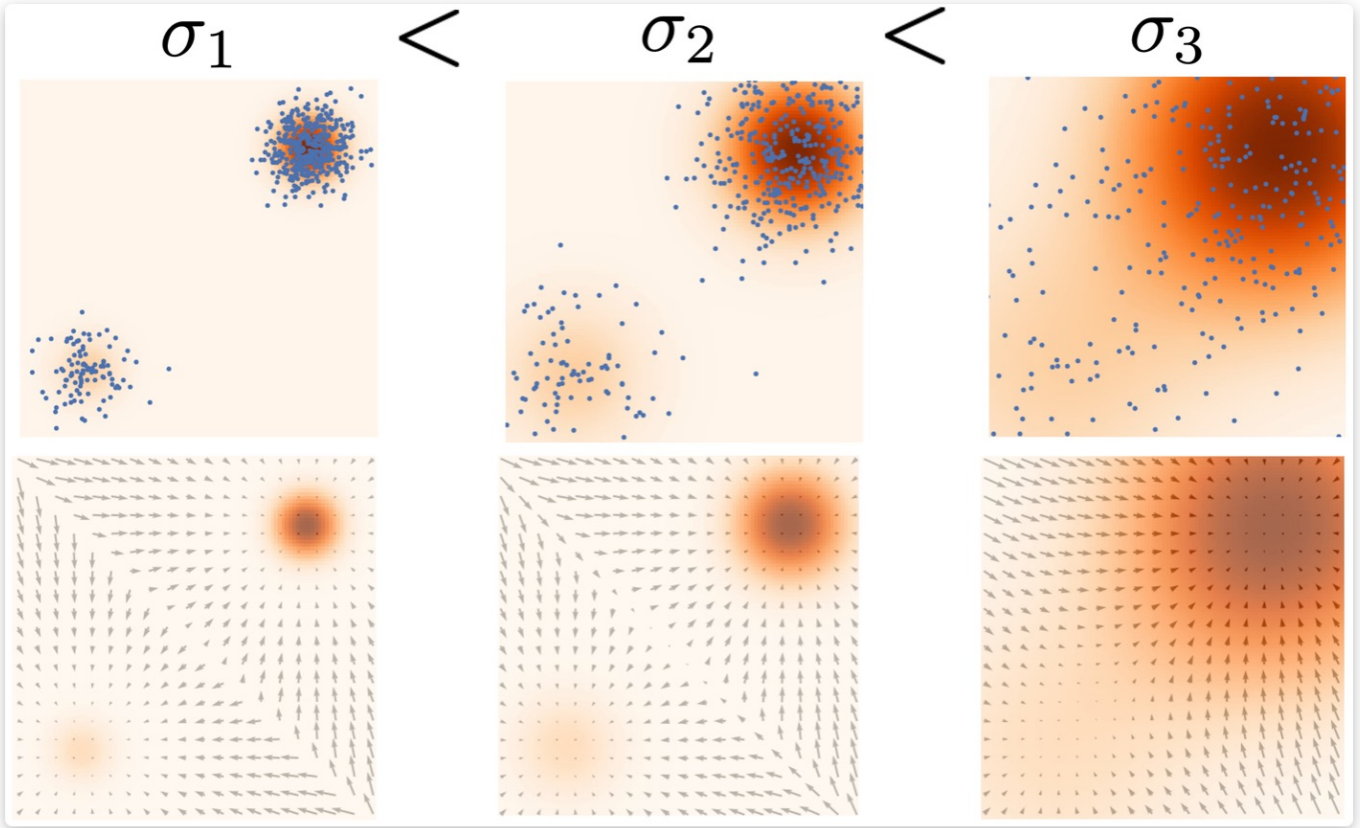
因此训练目标是 fisher divergences 的加权和，可以写作

$$\sum_{i=1}^{L} \lambda(i) \mathbb{E}_{p_{\sigma_i}(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, i)\|_2^2],$$

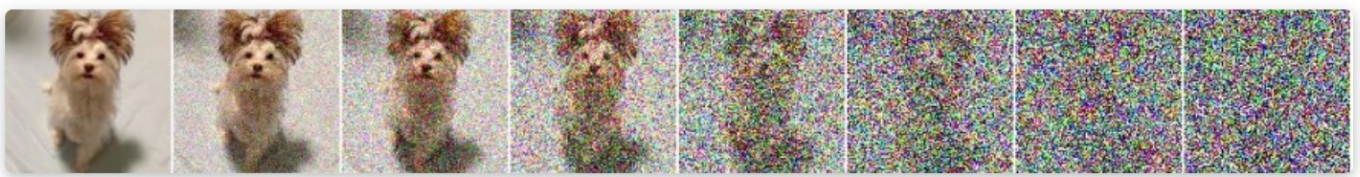
where $\lambda(i) \in \mathbb{R}_{>0}$ is a positive weighting function, often chosen to be $\lambda(i) = \sigma_i^2$.

在训练了 NCSC 得到更精确的 score model后，我们可以按照从 L 到 1 的序列来运行 Langevin dynamics，也称为 annealed Langevin dynamics，因为噪声尺度随时间逐渐降低 anneal





We apply multiple scales of Gaussian noise to perturb the data distribution (**first row**), and jointly estimate the score functions for all of them (**second row**).



Perturbing an image with multiple scales of Gaussian noise.

与 NCSN 的异同

C Discussion on related work

Our model architecture, forward process definition, and prior differ from NCSN [55, 56] in subtle but important ways that improve sample quality, and, notably, we directly train our sampler as a latent variable model rather than adding it after training post-hoc. In greater detail:

1. We use a U-Net with self-attention; NCSN uses a RefineNet with dilated convolutions. We condition all layers on t by adding in the Transformer sinusoidal position embedding, rather than only in normalization layers (NCSNv1) or only at the output (v2).
2. Diffusion models scale down the data with each forward process step (by a $\sqrt{1-\beta_t}$ factor) so that variance does not grow when adding noise, thus providing consistently scaled inputs to the neural net reverse process. NCSN omits this scaling factor.
3. Unlike NCSN, our forward process destroys signal ($D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})) \approx 0$), ensuring a close match between the prior and aggregate posterior of $\mathbf{x}_T$. Also unlike NCSN, our β_t are very small, which ensures that the forward process is reversible by a Markov chain with conditional Gaussians. Both of these factors prevent distribution shift when sampling.
4. Our Langevin-like sampler has coefficients (learning rate, noise scale, etc.) derived rigorously from β_t in the forward process. Thus, our training procedure directly trains our sampler to match the data distribution after T steps: it trains the sampler as a latent variable model using variational inference. In contrast, NCSN's sampler coefficients are set by hand post-hoc, and their training procedure is not guaranteed to directly optimize a quality metric of their sampler.

总结

NCSN 是一种 score-based 生成模型，采样由 Langevin dynamics 生成，使用 score matching 估计的数据分布的梯度，每个采样x的密度概率的 score 由其梯度定义，训练一个 score network 来估计。为了使得在深度学习高维数据上 scalable，提出了 either denoising score matching（添加一个预先指定的小噪音到数据中） or sliced score matching （使用随机算子projections）

然而，由于 manifold hypothesis，大多数数据集中于一个低维 manifold，即使观测到的数据可能看起来任意高维，这带来一个 negative effect on score estimation，因为数据点不能覆盖整个空间。在数据密度低的区域，score estimation 更不可靠。

然而，通过添加小的 Gaussian 噪音来使得扰动的数据分布覆盖整个空间，训练的 score estimator network 变得更稳定。

NCSN 通过使用不同等级的噪音扰动数据，并且训练一个 noise-conditioned score network 来联合估计所有被不同噪声等级扰动的数据的 scores。

不断增加噪声等级类似于 forward diffusion

总结

可牵引性和灵活性是生成建模中两个相互冲突的目标。可追溯模型可以进行分析评估，并廉价地拟合数据（例如通过高斯或拉普拉斯），但它们不容易描述丰富数据集中的结构。灵活的模型可以适合数据中的任意结构，但从这些模型中评估、训练或取样通常很昂贵。扩散模型在分析上既易于处理，又灵活

扩散模型依靠长的马尔可夫扩散步骤链来生成样本，因此在时间和计算方面可能相当昂贵。已经提出了使该过程更快的新方法，但采样速度仍然比GAN慢。

Following Works

参数优化

加速采样: DDIM

条件生成

从离散到连续: 与SDE的关系

SDE的解析解: Analytic-DPM

SDE的高阶解析解: DPM-Solver

Following Works

论文中，随着扩散过程中噪音越来越多，因此可以增大扩散率，论文采用 0.0001 到 0.02 的插值，这相对于归一化的像素仍然是较小的。Diffusion model 展示了高质量的采样，但是无法实现可比的 model log-likelihood as other 生成模型

Nichol 21 做了些改进获得更低的NLL，使用 cosine-based variance schedule，可以是任意的，只要在训练中期提供了近线形的drop，并且在开始和结束 subtle changes

$$\beta_t = \text{clip}(1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, 0.999) \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \quad \text{where } f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)$$

where the small offset s is to prevent β_t from being too small when close to $t = 0$.

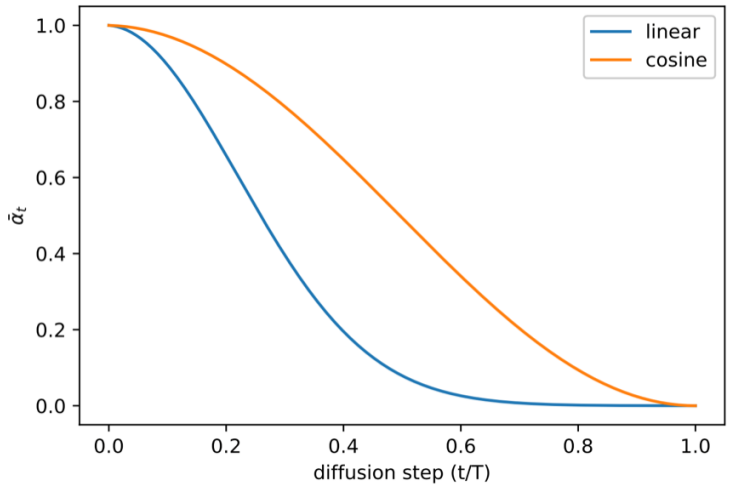


Fig. 5. Comparison of linear and cosine-based scheduling of β_t during training. (Image source: Nichol & Dhariwal, 2021)

Alex Nichol & Prafulla Dhariwal. “ Improved denoising diffusion probabilistic models” arxiv Preprint arxiv:2102.09672 (2021).

逆过程方差的参数选择

DDPM 选择固定扩散率系数，不可学习，并设定

$$\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I} \text{ , where } \sigma_t \text{ is not learned but set to } \beta_t \text{ or } \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t.$$

因为发现 diagonal variance 会导致训练不稳定，以及更差的采样质量

同样在 Nichol 21文章中，提出学习方差作为一个插值，建模预测一个混合向量

$$\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \exp(\mathbf{v} \log \beta_t + (1 - \mathbf{v}) \log \tilde{\beta}_t)$$

然而简化的目标函数 L_simple 与方差无关，位了使得与方差有关，构造了一个混合目标函数

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{VLB}} \text{ where } \lambda = 0.001 \text{ is small}$$

并且截断了 L_VLB 关于期望的梯度，只 guide 方差，然而基于噪声梯度非常难以优化 L_VLB，因此提出使用 time-averaging smoothed version L_VLB 使用重要性采样。

Alex Nichol & Prafulla Dhariwal. “ Improved denoising diffusion probabilistic models” arxiv Preprint arxiv:2102.09672 (2021).

加速采样: DDIM

DDPM 扩散步骤上千，从逆过程 Markov Chain 采样十分耗时，大约是 GAN 的 1200 倍
一种简单方法是多步采样一次，在 nichol 21文章中提到

另一种方法是将逆过程重写为

$$
\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\mathbf{z}_{t-1} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\mathbf{z}_t + \sigma_t\mathbf{z} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t\mathbf{z} \\
q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I})
\end{aligned}
$$

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

$$
\tilde{\beta}_t = \sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t
$$

Let $\sigma_t^2 = \eta\cdot\tilde{\beta}_t$ such that we can adjust $\eta\in\mathbb{R}^+$ as a hyperparameter to control the sampling stochasticity. The special case of $\eta=0$ makes the sampling process *deterministic*. Such a model is

这个方法称为DDIM，DDIM具有相同的边际噪声分布，但确定性地将噪声映射回原始数据样本。

加速采样: DDIM

While all the models are trained with $T = 1000$ diffusion steps in the experiments, they observed that DDIM ($\eta = 0$) can produce the best quality samples when S is small, while DDPM ($\eta = 1$) performs much worse on small S. DDPM does perform better when we can afford to run the full reverse Markov diffusion steps ($S = T = 1000$). With DDIM, it is possible to train the diffusion model up to any arbitrary number of forward steps but only sample from a subset of steps in the generative process.

		CIFAR10 (32×32)					CelebA (64×64)				
	S	10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
	$\hat{\sigma}$	367.43	133.37	32.72	9.99	**3.17**	299.71	183.83	71.71	45.20	**3.26**

Fig. 7. FID scores on CIFAR10 and CelebA datasets by diffusion models of different settings, including DDIM ($\eta = 0$) and DDPM ($\hat{\sigma}$). (Image source: Song et al., 2020)

Compared to DDPM, DDIM is able to:

1. Generate higher-quality samples using a much fewer number of steps.
2. Have "consistency" property since the generative process is deterministic, meaning that multiple samples conditioned on the same latent variable should have similar high-level features.
3. Because of the consistency, DDIM can do semantically meaningful interpolation in the latent variable.

条件生成

另一方面，图像生成通常需要基于标签分类，Dharial 21 显式地训练了一个噪音图片x_t上的分类器，并且使用分类器梯度来引导diffusion采样过程到目标分类，这种 *ablated diffusion model* (**ADM**) 和 the one with additional classifier guidance (**ADM-G**) 可以取得比sota更好的效果，如BigGAN

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $f_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log f_\phi(y|x_t), \Sigma)$
end for
return x_0

Algorithm 2 Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, classifier $f_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log f_\phi(y|x_t)$
  $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return x_0

Fig. 8. The algorithms use guidance from a classifier to run conditioned generation with DDPM and DDIM. (Image source: Dhariwal & Nichol, 2021])

Prafula Dhariwal & Alex Nichol. "Diffusion Models Beat GANs on Image Synthesis." arxiv Preprint arxiv:2105.05233 (2021).

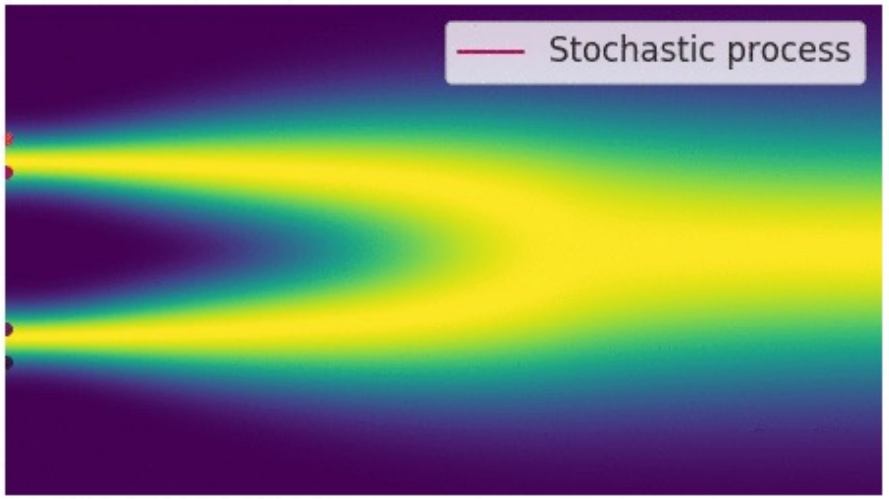
从离散到连续：DDPM 与 SDE 的关系

基于NCSN，添加多种尺度噪音对于score based生成模型至关重要，如果我们把噪声尺度扩展到 infinity，不仅可以得到更高质量的采样，也可以得到 exact log-likelihood computation，以及 controllable generation for inverse problem solving

当噪声尺度变为无限的，那么就得到了连续上升的噪音等级，这样噪音扰动就转化为了连续时间随机过程，可以用SDE来描述，形式为

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{8}$$

where $\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued function called the drift coefficient, $g(t) \in \mathbb{R}$ is a real-valued function called the diffusion coefficient, $\mathbf{w}$ denotes a standard Brownian motion, and $\mathbf{dw}$ can be viewed as infinitesimal white noise. The solution of a stochastic differential



Score-Based Generative Modeling through Stochastic Differential Equations。Yang Song 21 ICLR

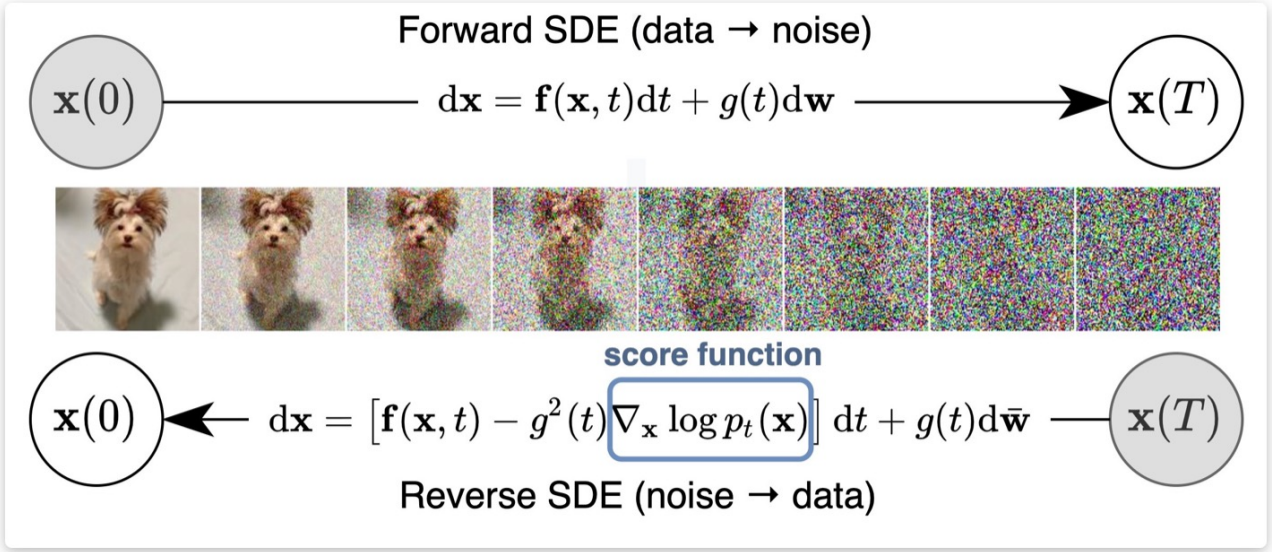
从离散到连续：DDPM 与 SDE 的关系

在之前使用有限数量噪声尺度时，可以通过 annealed Langevin dynamics 来生成逆扰动过程的采样，对于无限噪声尺度，我们可以类似的逆采样，使用 reverse SDE

Importantly, any SDE has a corresponding reverse SDE [34], whose closed form is given by

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{10}$$

Here $\mathbf{d}t$ represents a negative infinitesimal time step, since the SDE (10) needs to be solved backwards in time (from $t = T$ to $t = 0$). In order to compute the reverse SDE, we need to estimate $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, which is exactly the **score function** of $p_t(\mathbf{x})$.



Solving a reverse SDE yields a score-based generative model. Transforming data to a simple noise distribution can be accomplished with an SDE. It can be reversed to generate samples from noise if we know the score of the distribution at each intermediate time step.

从离散到连续：DDPM 与 SDE 的关系

为了求解 reverse SDE，需要知道两点：最终分布p_T和 score function：前者可以指定 prior 分布，后者通过训练一个 time-dependent score-based model 来估计，$\mathbf{s}_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

等价于 NCSC 中的 condition-score function $\mathbf{s}_\theta(\mathbf{x}, i) \approx \nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x})$.

Our training objective for $\mathbf{s}_\theta(\mathbf{x}, t)$ is a continuous weighted combination of Fisher divergences, given by

$$\mathbb{E}_{t\in\mathcal{U}(0,T)}\mathbb{E}_{p_t(\mathbf{x})}[\lambda(t)\|\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, t)\|_2^2], \tag{11}$$

where $\mathcal{U}(0, T)$ denotes a uniform distribution over the time interval $[0, T]$, and $\lambda : \mathbb{R} \to \mathbb{R}_{>0}$ is a positive weighting function. Typically we use $\lambda(t) \propto 1/\mathbb{E}[\|\nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t) \mid \mathbf{x}(0))\|_2^2]$ to balance the magnitude of different score matching losses across time.

求解权重 Fisher divergences 可以使用高效求解方法，比如 denoising score matching 和 sliced score matching，

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\mathbf{s}_\theta(\mathbf{x}, t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}.$$

如果 $\lambda(t) = g^2(t)$，，那么和 KL 散度有一个重要的联系 $\mathrm{KL}(p_0(\mathbf{x})\|p_\theta(\mathbf{x})) \le \frac{T}{2}\mathbb{E}_{t\in\mathcal{U}(0,T)}\mathbb{E}_{p_t(\mathbf{x})}[\lambda(t)\|\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, t)\|_2^2]$

$$+ \mathrm{KL}(p_T \,\|\, \pi). \tag{13}$$

因此称这个条件为 likelihood weighting function，可以用来训练 score based 生成模型实现高似然，接近自回归模型sota

Sliced score matching: A scalable approach to density and score estimation. Yang Song 20 UAI
Maximum Likelihood Training of Score-Based Diffusion Models。Yang Song 21 NIPS

Summary of Yang Song's Work

而求解 reverse SDE 可以使用 Euler-Maruyama 方法，离散了 SDE 使用很小的 time steps 和 Gaussian noise，与 Langevin 动力学 十分相似。
同时，SDE可以转化为ODE求解，并且更为普适化和平滑

2019 开始，make score matching scalable for training deep energy-based models on 高维数据，
第一个工作是 sliced score matching，**Sliced score matching: A scalable approach to density and score estimation 2020 UAI，**实现了 scalable 但是发现 Langevin 采样无法生成合理的样本，发现了三个可以显著提升的方法：

I. 扰动数据，多尺度噪声，训练 score based model for each noise scale；
II. 使用 U-Net 结构而不是 RefineNet 结构
III. 应用 Langevin MCMC 到每个噪声尺度并且把他们 chain 起来，因此得到了 **Generative Modeling by Estimating Gradients of the Data Distribution** 2019 nips 和改进版本 **Improved Techniques for Training Score-Based Generative Models** NIPS 20

然而添加噪音并非专属 score based 生成，在 2015 扩散概率模型也有，但是 score 方法从另一视角出发，前者通过 evidence lower bound ELBO 训练和学习到的 decoder 采样，而后者利用 score matching 训练并从 Langevin 采样。

DDPM首次显著提升了经验性能，并且解释了更深层次的联系与score based model， ELBO 其实等价于权重组合分数匹配目标，并且参数化 decoder 为一个 U-Net 网络下的序列 score-based model，同样可以生成优于GAN的结果。

进一步在 **Score-Based Generative Modeling through Stochastic Differential Equations ICLR 21** 讨论了 diffusion model 和 score-based generative model 之间的关系，diffusion probabilistic model 的采样方法可以被融合为 annealed Langevin dynamics in score-based models，来生成一个统一形式的 Predictor-Corrector sampler.
通过生成无限尺度噪音，进一步证明两者都可以看做 score function 决定的 SDE 方程的离散化形式，建立了统一形式框架，类似于波粒二象性的统一。

这个 score matching 视角允许精确计算 log-likelihoods，直接与 energy-based models, schrodinger bridges 和 最优传输问题相关

Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling NIPS 21,

Analytic-DPM: SDE 离散化

4.1. DDPM与其解析形式Analytic-DPM：对diffusion SDE的离散化

DDPM提出要用一个特殊参数化的高斯分布来建模逆过程 $p_\theta(x_{t-1}|x_t)$ ，这个高斯分布的均值特殊地取成了 $q(x_{t-1}|x_t,x_0)$ 的均值，并把 x_0 换成了一个参数化模型（即上文提到的三种等价参数化模型中的“去噪”模型）：

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)=\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t,\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t))\right)=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\right)$$

离散时间diffusion model的逆过程的均值

具体的推导基于最小化前向过程和逆向过程的联合分布的KL。然而，DDPM原文中的推导非常不严谨，有一点“硬凑”的感觉，并且也没有给出怎么确定高斯分布的方差，也是强行凑了一个方差。在这之后原作者又做了个后续工作叫Improved DDPM，使用了一个参数化网络去学习高斯分布的协方差矩阵，但优化起来也非常麻烦。

在这之后，Yang Song在Score-Based Generative Modeling through Stochastic Differential Equations的附录中证明了，DDPM的采样过程（基于 $p_\theta(x_{t-1}|x_t)$ 的祖先采样（ancestral sampling））实际上等价于diffusion SDE的一阶离散化，这就统一了diffusion SDE的离散与连续。然而，实际上针对SDE的一阶离散化有无数种等价形式（只需要保证离散化误差的阶是1阶即可）。有一个非常重要的问题还没有被解决：**DDPM这种看似有点巧妙的“硬凑”出来的均值到底对应了怎样的SDE离散化？有没有更好的方差建模方式？**

接着在ICLR 2022有两篇独立研究的工作出现，一篇是我实验室同届的同学鲍凡 @Baof 提出的Analytic-DPM（Oral，outstanding paper award），另一篇是Diffusion-Based Voice Conversion with Fast Maximum Likelihood Sampling Scheme（Oral），这两篇结合起来证明了这件事：**DDPM的均值实际上是对diffusion SDE的maximum likelihood SDE solver，并且最优方差有解析形式，且可以被score function唯一确定。**

具体而言，Analytic-DPM从严谨的数学角度证明了DDPM中最小化KL得到的 $p_\theta(x_{t-1}|x_t)$ **的均值和方差都是有解析形式的！**这直接颠覆了以前搞DDPM那些人的观念（我猜内心os：合着我费劲用神经网络学了半天的方差其实根本不用学？）。具体而言，最优的均值和方差长这样：

Theorem 1. *(Score representation of the optimal solution to Eq. (4), proof in Appendix A.2)*

The optimal solution $\boldsymbol{\mu}_n^*(\boldsymbol{x}_n)$ and σ_n^{*2} to Eq. (4) are

$$\boldsymbol{\mu}_n^*(\boldsymbol{x}_n)=\tilde{\boldsymbol{\mu}}_n\left(\boldsymbol{x}_n,\frac{1}{\sqrt{\bar{\alpha}_n}}(\boldsymbol{x}_n+\bar{\beta}_n\nabla_{\boldsymbol{x}_n}\log q_n(\boldsymbol{x}_n))\right),\tag{6}$$

$$\sigma_n^{*2}=\lambda_n^2+\left(\sqrt{\frac{\bar{\beta}_n}{\alpha_n}}-\sqrt{\bar{\beta}_{n-1}-\lambda_n^2}\right)^2\left(1-\bar{\beta}_n\mathbb{E}_{q_n(\boldsymbol{x}_n)}\frac{||\nabla_{\boldsymbol{x}_n}\log q_n(\boldsymbol{x}_n)||^2}{d}\right),\tag{7}$$

where $q_n(\boldsymbol{x}_n)$ is the marginal distribution of the forward process at timestep n, and d is the dimension of the data.

Analytic-DPM中的最优均值与方差的解析形式

Analytic-DPM在附录中证明了，这里的最优均值等价于DDPM中的均值的参数化（把score function替换成score model s_θ ，再把score model等价转换成噪声预测模型 ϵ_θ ），并且由于方差里只有score function是未知的，也可以直接替换成我们预训练的score model（或者等价的噪声预测模型）。这是一个非常漂亮的理论：**不但从数学角度严谨支持了DDPM原来选择的均值的数学背景，还证明了最优方差也可以用score model来近似！**基于这个发现，Analytic-DPM极大地提高了原始DDPM的likelihood和sample速度，在几十步就可以达到和原来1000步可比的效果。在鲍凡后续发表在ICML 2022的工作Estimating the Optimal Covariance with Imperfect Mean in Diffusion Probabilistic Models中，把Analytic-DPM扩展到了对角协方差矩阵的形式，并且考虑了均值网络的训练误差来进一步优化协方差矩阵，在原始的Analytic-DPM的基础上把效果又提高了一些。

此外，上文提到的那篇同时独立的工作证明了DDPM其实是对diffusion SDE的maximum likelihood SDE solver（其实就是Analytic-DPM中说的“最小化每一小步的KL”），但这篇工作并没有给出最优方差，就不多做介绍了。

综上所述，离散时间的DDPM其实基本被研究清楚了：**DDPM对应了diffusion SDE的maximum likelihood SDE solver，并且最优方差由Analytic-DPM来解析地给出。**

DPM-Solver: 高阶解析解

4.2. DDIM及其高阶形式DPM-Solver：对diffusion ODE的离散化

Jiaming Song在ICLR 2021提出了DDIM，对应的是另一种离散时间的diffusion model。在那时，diffusion model的连续形式还没有被提出，因此DDIM的原始推导也像DDPM那样“硬凑”了一个逆向过程的均值与方差，并且当方差设置为0时变成了一个implicit generative model。巧的是，这种方差为0的模型由于没有噪声的干扰，可以在几十步到100步内就可以达到原始DDPM接近1000步的采样质量。这是针对DDPM的加速采样的开山之作。

然而，当diffusion model的连续版本提出之时，DDIM到底对应了一种怎样的连续版本，一直是一个未解之谜，直到ICLR 2022中，DDPM的原作者们提出了基于DDIM的蒸馏来为diffusion model做加速：Progressive Distillation for Fast Sampling of Diffusion Models，在这篇文章的附录中证明了DDIM是diffusion ODE在换元到log-SNR后的一阶ODE solver。然而，一直以来大家都发现使用那些general的ODE solver（例如RK45）在步数少的时候采样质量远远不如DDIM，而这篇文章的证明里并没有能给出一个直观的解释：为什么DDIM能比普通的ODE solver好?

这个问题直到最近才被我和实验室同学提出的DPM-Solver所解决：我们证明了，DDIM其实是基于diffusion ODE的半线性结构（semi-linear ODE）的一阶ODE solver，并且给出了对应的高阶solver形式。具体地，我们给出了**diffusion ODE的解的解析形式：**

Proposition 3.1 (Exact solution of diffusion ODEs). *Given an initial value $\boldsymbol{x}_s$ at time $s > 0$, the solution $\boldsymbol{x}_t$ at time $t \in [0, s]$ of diffusion ODEs in Eq. (2.7) is:*

$$\boldsymbol{x}_t = \frac{\alpha_t}{\alpha_s}\boldsymbol{x}_s - \alpha_t \int_{\lambda_s}^{\lambda_t} e^{-\lambda}\hat{\boldsymbol{\epsilon}}_\theta(\hat{\boldsymbol{x}}_\lambda, \lambda)\mathrm{d}\lambda. \tag{3.4}$$

Diffusion ODE的解析解

我们可以看到，diffusion ODE的离散化其实就是对这个解析解的积分做离散化近似。一个最简单的方法就是直接让 $\epsilon_\theta(\tilde{x}_\lambda, \lambda) \approx \epsilon_\theta(\tilde{x}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}})$ （零阶泰勒展开），那么可以得到：

$$\begin{aligned}\boldsymbol{x}_{t_{i-1}\to t_i} &= \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\tilde{\boldsymbol{x}}_{t_{i-1}} - \alpha_{t_i}\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1})\int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda}\mathrm{d}\lambda + \mathcal{O}(h_i^2)\\ &= \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}}\tilde{\boldsymbol{x}}_{t_{i-1}} - \sigma_{t_i}(e^{h_i}-1)\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1}) + \mathcal{O}(h_i^2).\end{aligned}$$

DDIM是DPM-Solver的1阶形式

如果忽略后面的误差项，这其实就是DDIM！基于这个观察，我们进一步做了更高阶的ODE solver（类似数值分析中Runge-Kutta法的推导，这里就不展开了），我们把他们统一称为DPM-Solver。

综上所述，离散时间的DDIM其实也基本被研究清楚了：**DDIM对应了diffusion ODE的1阶ODE solver，它的加速效果好是因为它考虑了ODE的半线性结构，而DPM-Solver给出了对应的更高阶的solver，可以让10步左右的采样达到与DDPM的1000步的采样相当。**

Related Source

Blog：Song Yang | Stanford USA | Generative Modeling by Estimating Gradients of the Data Distribution

https://yang-song.github.io/blog/2021/score/#concluding-remarks

Blog: Ayan Das | Surrey UK | An introduction to Diffusion Probabilistic Models

https://ayandas.me/blog-tut/2021/12/04/diffusion-prob-models.html

Blog: Lilian Weng | OpenAI USA | What are Diffusion Models?

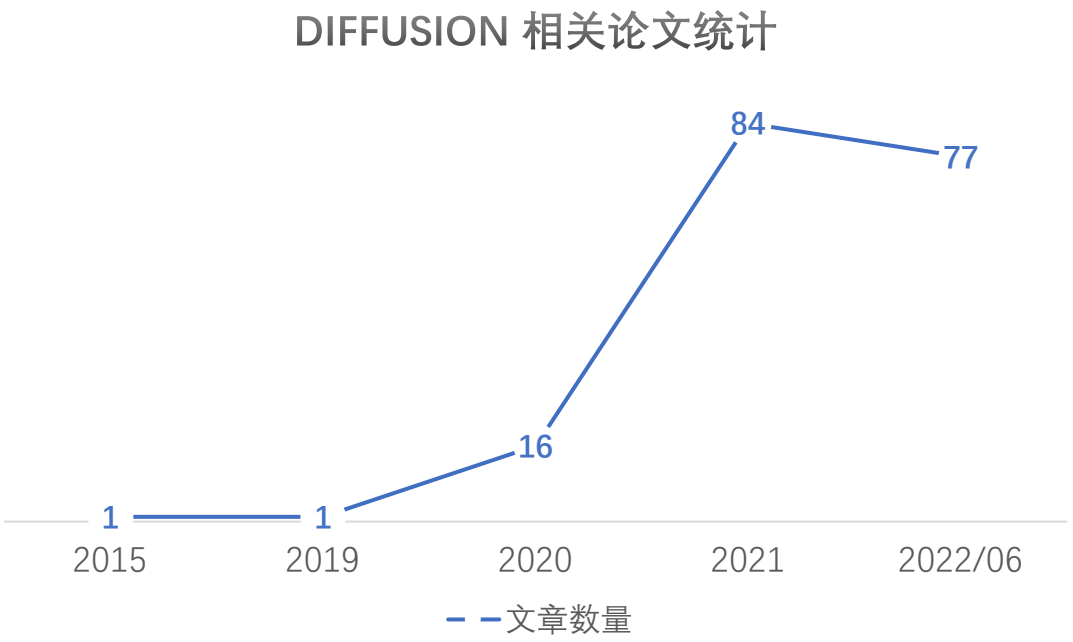
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Slides: Fan Bao | Tsinghua CN | Diffusion Probabilistic Models: Theory and Applications

Related Source

Paper List

https://scorebasedgenerativemodeling.github.io



What's the score?

Review of latest Score Based Generative Modeling papers.

About

- James Thornton
- Valentin De Bortoli

Select Categories: Nothing selected Select Authors: Nothing selected

Select Year: Nothing selected

Convergence for score-based generative modeling with polynomial complexity

June 13, 2022 | Holden Lee, Jianfeng Lu, Yixin Tan

cs.LG, math.PR, math.ST, stat.ML, stat.TH

Paper

Multi-instrument Music Synthesis with Spectrogram Diffusion

June 11, 2022 | Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, Jesse Engel

cs.SD, cs.LG, eess.AS

Paper

How Much is Enough? A Study on Diffusion Times in Score-based Generative Models

June 10, 2022 | Giulio Franzese, Simone Rossi, Lixuan Yang, Alessandro Finamore, Dario Rossi, Maurizio Filippone, Pietro Michiardi

stat.ML, cs.LG

Paper

Image Generation with Multimodal Priors using Denoising Diffusion Probabilistic Models

June 10, 2022 | Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, Vishal M Patel

cs.CV

Paper

Probability flow solution of the Fokker-Planck equation

June 09, 2022 | Nicholas M. Boffi, Eric Vanden-Eijnden

cs.LG, cond-mat.dis-nn, cond-mat.stat-mech, cs.NA, math.NA, math.PR

Paper

Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem

June 08, 2022 | Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, Tommi Jaakkola

q-bio.BM, cs.LG, stat.ML

Paper

Accelerating Score-based Generative Models for High-Resolution Image

Related Source

Disco Diffusion Tutorial

https://blog.csdn.net/xiqiao_ce/article/details/125175930

https://zhuanlan.zhihu.com/p/526242977

Disco Diffusion v5.4 Google Colab

https://colab.research.google.com/github/alembics/disco-diffusion/blob/main/Disco_Diffusion.ipynb

5. Diffusion model的优势与热点研究问题

Diffusion model最大的优势是训练简单。例如“预测噪声”，其实就是用一个二范数来训练。Diffusion model借助了图像分割领域的UNet，训练loss稳定，模型效果非常也好。相比于GAN需要和判别器对抗训练或者VAE需要变分后验，diffusion model的loss真的是太简单了。究其本质，其实就是我在1中提到的，diffusion model只需要“模仿”一个非常简单的前向过程对应的逆过程即可。这样简单高效的训练也使得diffusion model在许多任务中的表现都非常好，甚至超过了GAN。

然而，目前diffusion model的理论研究中最大的问题就是它的采样速度较慢。目前有许多针对这个问题的研究，开山之作是Jiaming Song提出的DDIM，包括鲍凡的Analytic-DPM的主要应用也是加速DDPM的采样。我们最新的研究提出了一个叫DPM-Solver的加速采样算法，证明了DDIM是diffusion ODE的一阶ODE solver，并且提出了二阶、三阶solver，**可以做到10步采样质量很不错，20步几乎收敛。我们的方法不需要任何额外训练，任给一个pretrained model都可以直接用。据我所知，这应该是目前training-free sampler里最快的了。**

深度生成模型除了可以生成数据以外，还有一类核心任务是估计数据的概率密度（可以用模型计算的数据似然（likelihood）来刻画）。GAN被诟病的一点就是无法计算似然，因为它是隐式生成模型（implicit generative model），这导致GAN无法被用来数据压缩等领域。而VAE只能计算数据似然的一个下界，也不太令人满意。在Diffusion model提出之前，人们通常都是用autoregressive model或者normalizing flow来计算数据的精确似然，然而这类模型的表现都不够好。我博一时候与组里另一位博士后学长合作的VFlow就是为了提高normalizing flow的似然，达到了当时CIFAR-10最好的似然估计水平。然而，这类模型非常难训练，模型的参数量也巨大，很不美观。

DDPM与上文提到的连续版本的reverse SDE可以称为diffusion SDE（也叫Score-based SDE），它们都是基于SDE（离散版本为条件高斯分布）定义的生成模型。然而，这类模型与VAE一样，无法计算精确的数据似然（likelihood），而只能计算ELBO。Diffusion model的另一个优势是，只要训练了score model（上文提到的三种等价形式都可以转换为score model），那么就可以导出一个Neural ODE，它是一类continuous normalizing flow，可以精确计算数据的似然。这个推导也是由Yang Song在Score-Based Generative Modeling through Stochastic Differential Equations中首次提出（所以说大佬就是大佬，直接开创一个领域...）。这里就不展开详细的推导了，只是可以理解为，**diffusion model同时可以定义一个Neural ODE，它可以被用来做密度估计和似然计算。**基于diffusion ODE，diffusion model在密度估计领域里也达到了SOTA，吊打了以往所有的normalizing flow。

目前大部分应用领域的文章都是基于diffusion SDE（或者它的离散版本，DDPM）来发的，这是因为这类模型的生成效果往往比diffusion ODE好。然而，diffusion SDE有一个致命问题是采样速度极慢，因为SDE的离散化比ODE困难得多。目前基于diffusion SDE的最快的采样方法应该还是鲍凡 @Baof 的Analytic-DPM和它的扩展版本，可以做到25-50步就达到比较好的效果。然而，步数再低，采样效果就急剧下降了。

在这个月之前，diffusion ODE的生成效果一直都不如diffusion SDE。然而，这个问题在这个月被大名鼎鼎的StyleGAN的作者给解决了。在最近挂arxiv的Elucidating the Design Space of Diffusion-Based Generative Models中，大佬们直接把diffusion ODE的生成效果调到比diffusion SDE还要好，这给diffusion ODE的后续下游应用也带来了非常大的希望。