Diffusion Probabilistic Generative Models

Demin Yu

2023.04

Contents

Background

Background——
Generative Model

- 给定数据样本X，如何表达概率分布？
- 两种生成模型分类：
  - 基于似然估计（Likelihood-based models）：直接通过最大似然估计，建模目标分布的概率密度，如Autogressive models、flow models以及VAE系列
  - 隐式生成模型（Implicit generative models）：从已知分布采样，对数据采样过程建模，将样本映射到目标分布，如GAN等

$$x \in \{x^1, x^2, \ldots, x^N\} \sim p(x)$$

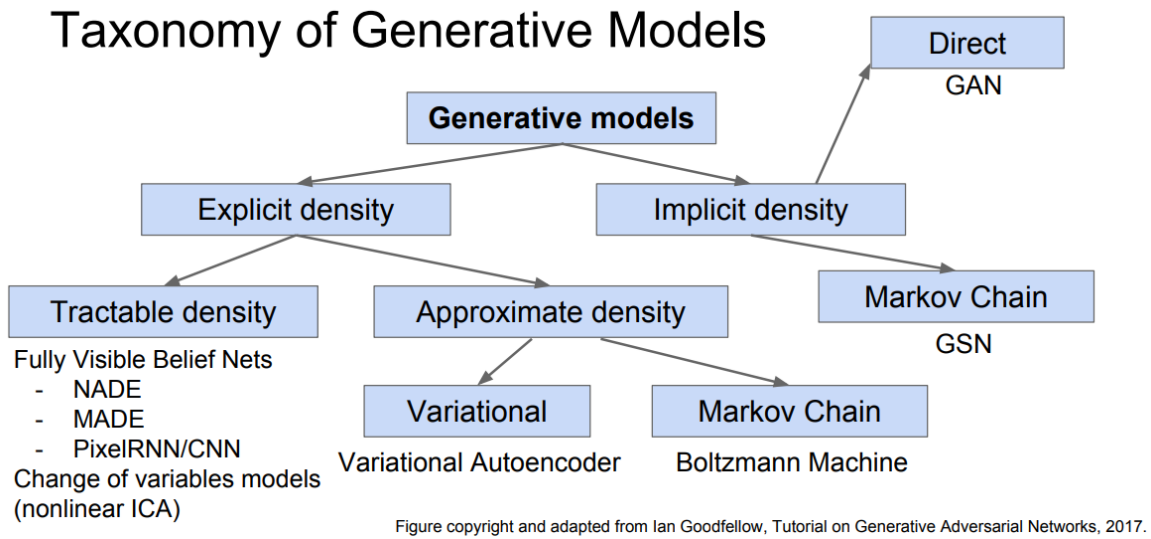
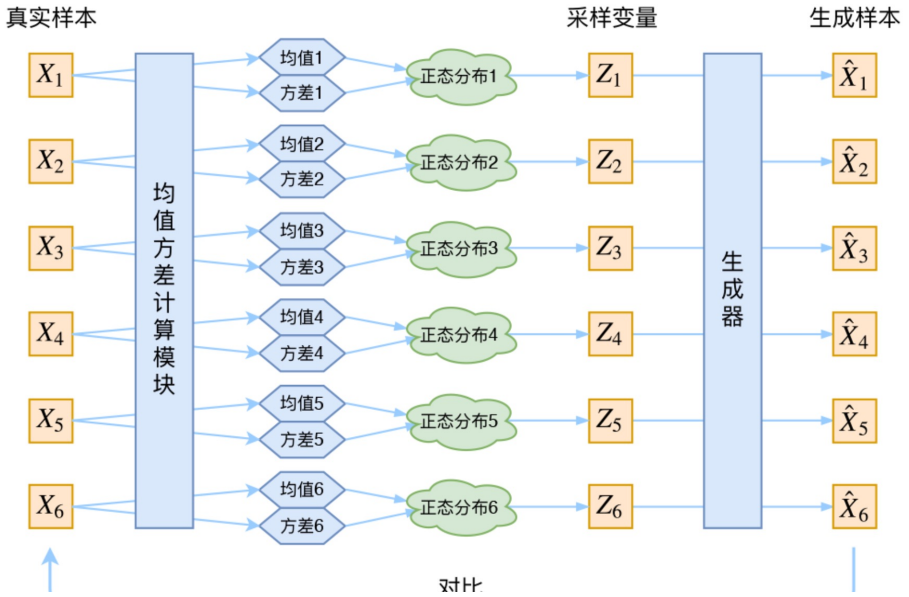


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

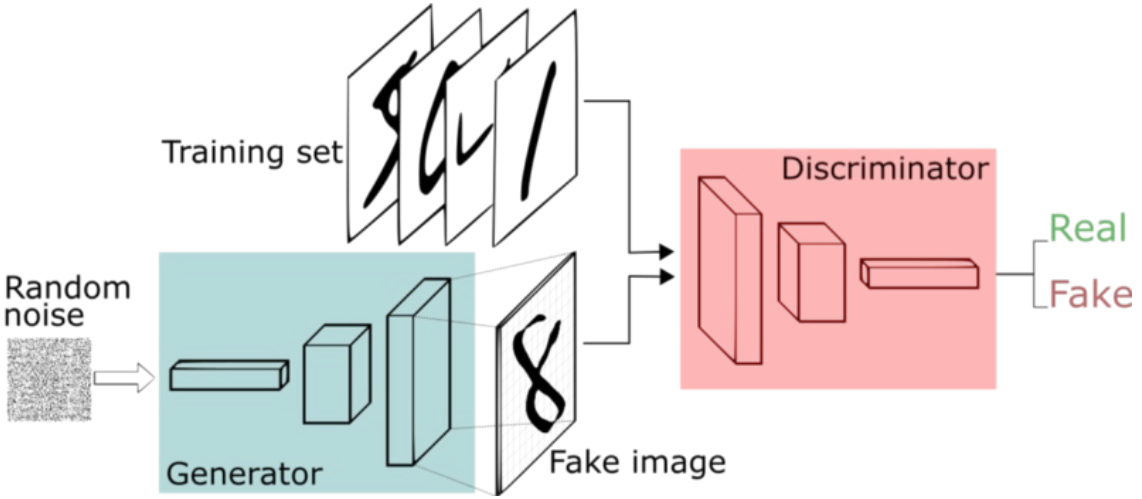
1. Generative Modeling by Estimating Gradients of the Data Distribution | Yang Song (yang-song.net)

Background——
Generative Model

- 变分自编码器（VAE）与对抗生成模型（GAN）



$$p(z) = \sum_{x_i \in X} p(z|x_i)p(x_i) = \sum_{X_i \in X} \mathcal{N}(0,1)p(x_i) = \mathcal{N}(0,1) \sum_{x_i \in X} p(x_i) = \mathcal{N}(0$$



Training: $D^* = arg \max_D V(D,G)$

Objective Function for D

$$V(G,D) = E_{y \sim P_{data}}[logD(y)] + E_{y \sim P_G}[log(1 - D(y))]$$

1. Generative Modeling by Estimating Gradients of the Data Distribution | Yang Song (yang-song.net)

Score-based Generative Model

Score-based Generative Model

- 存在问题
  - 基于似然估计（Likelihood-based models）：在模型结构上加入强约束来保证归一化常量作最大似然的计算
  - 隐式生成模型（Implicit generative models）：不稳定的对抗训练

- NCSN（Noise Conditional Score Network）
  - 不直接去建模概率密度函数, 而是建模 the gradient of the log probability density function：

$$\nabla_x \log p(x)$$

  - 也称为得分函数Score function。

directly model p.d.f: $p_\theta(x) = \frac{e^{-f_\theta(x)}}{Z_\theta}$

The Z_θ is to make sure: $\int p_\theta(x)dx = 1$

Training goal: $\max x_\theta \sum_{i=1}^{N} \log p_\theta(x_i)$

这种形式的概率密度函数是生成模型的一种表示范式

1. Generative Modeling by Estimating Gradients of the Data Distribution | Yang Song (yang-song.net)
2. Song Y, Ermon S. Generative modeling by estimating gradients of the data distribution[J]. Advances in neural information processing systems, 2019, 32.

Score-based Generative Model

- 弊端的原因根本上是由于Z_θ与f_θ互相绑定，不方便计算
- 使用Score function的好处：
  - 建模得分函数的模型$s_\theta(x)$，使得 $s_\theta(x) \approx \nabla_x \log p(x)$
  - 如此就可以直接忽略掉归一化常量：

$$\begin{aligned} s_\theta(x) &\approx \nabla_x \log p(x) \\ &= \nabla_x \log \frac{e^{-f_\theta(x)}}{Z_\theta} \\ &= -\nabla_x f_\theta(x) - \underbrace{\nabla_x \log Z_\theta}_{=0} \\ &= -\nabla_x f_\theta(x) \end{aligned}$$

  - 训练目标：

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

directly model p.d.f: $p_\theta(x) = \frac{e^{-f_\theta(x)}}{Z_\theta}$

The Z_θ is to make sure: $\int p_\theta(x)dx = 1$

Training goal: $\max x_\theta \sum_{i=1}^{N} \log p_\theta(x_i)$

**这种形式的概率密度函数可视为生成模型的一种表示范式*

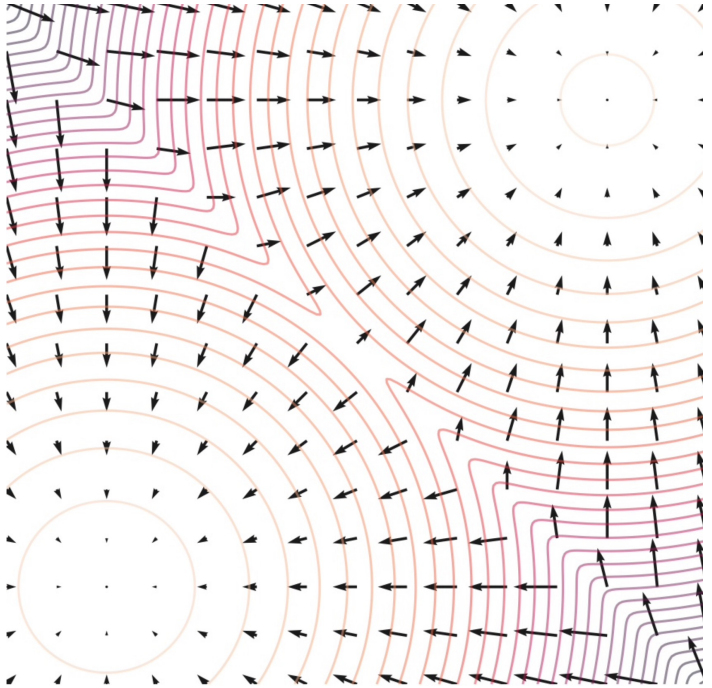
Score function $\nabla_x \log p(x)$

1. Song Y, Ermon S. Generative modeling by estimating gradients of the data distribution[J]. Advances in neural information processing systems, 2019, 32.

Score-based Generative Model——

How to understand Score Function?

- 几何角度
  - 高维空间中，一个分布到另一个分布的变化方向。
  - 类比到梯度下降，我们可以通过很多步的微小移动，逐步将已知先验分布过度到目标分布
- 数学角度
  - 从分布的定义出发，得分函数应当是一个“矢量场”
  - 矢量就有方向，方向为：对于输入数据（样本），其对数概率密度增长最快的方向



$$\nabla_x \log p(x)$$

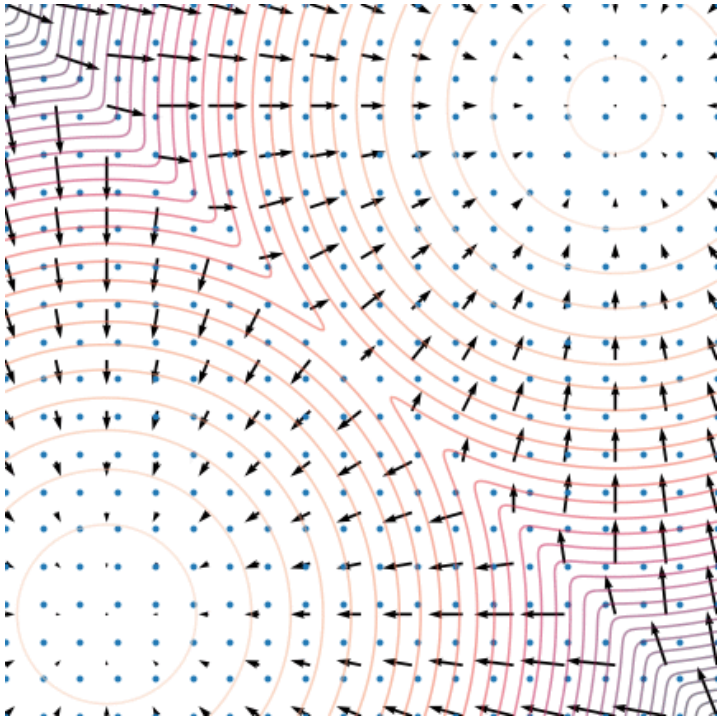
Note: 每一步的移动不是值，而是分布，如何表达分布间的移动？
隐式采样

1. Liu Q, Lee J, Jordan M. A kernelized Stein discrepancy for goodness-of-fit tests[C]//International conference on machine learning. PMLR, 2016: 276-284.

Score-based Generative Model——

Key to use Score Function

- 训练目标：Fisher divergence $\mathbb{E}_{p(x)}[\|\nabla_x \log p(x) - s_\theta(x)\|_2^2]$
- 如何获得目标式中的未知$\mathrm{p}(x)$：Score Matching
- 如何使用用训练好的得分模型
  - 目标是从当前分布转移到目标分布
  - 方式：郎之万动力学采样（**Langevin dynamics**）
- 如何避免得分陷阱：加噪



$\nabla_x \log p(x)$

Score-based Generative Model——

Score Matching for training s(x)

- 我们需要一个网络$s_\theta(x)$去估算得分函数，使用Fisher散度作为训练目标 $\mathbb{E}_{p(x)}[\|\nabla_x \log p(x) - s_\theta(x)\|_2^2]$
- 给定数据样本x，该样本的分布p(x)无法直接表示，利用Score Matching技巧可以避免计算p(x)而直接最小化目标：
  - denoising score matching
  - sliced score matching

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{p(x)}[\|\nabla_x \log p(x) - s_\theta(x)\|_2^2] \\ &= \int p(x)[\|\nabla_x \log p(x)\|^2 + \|s_\theta(x)\|^2 - 2(\nabla_x \log p(x))^T s_\theta(x)]dx \\ &= \cdots\cdots \\ &= \mathbb{E}_{p(x)}[\|s_\theta(x)\|^2 + 2tr(\nabla_x s_\theta(x))]\end{aligned}$$

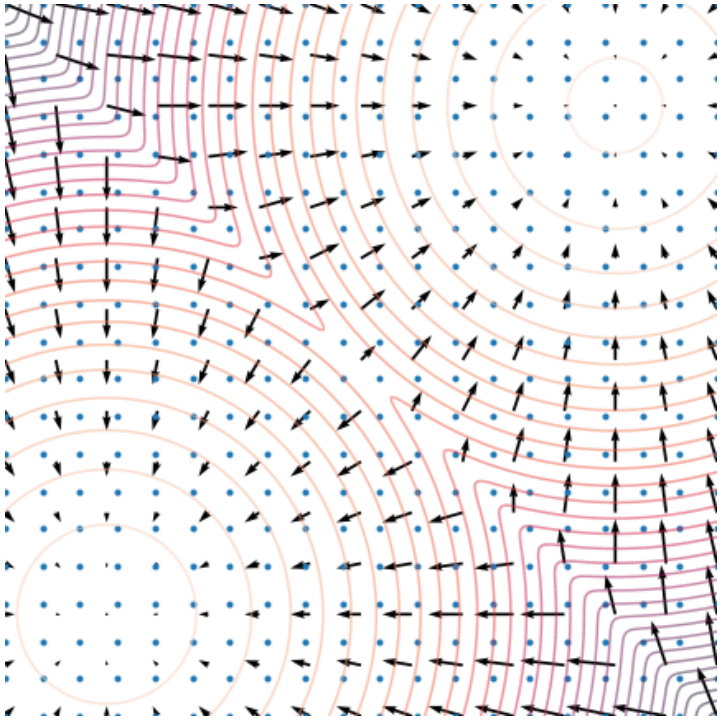
1. Score-based Generative Model - OCSS的文章 - 知乎 https://zhuanlan.zhihu.com/p/471570934
2. Estimation of non-normalized statistical models by score matching A. Hyvarinen. Journal of Machine Learning Research, Vol 6(Apr), pp. 695--709. 2005.
3. A connection between score matching and denoising autoencoders P. Vincent. Neural computation, Vol 23(7), pp. 1661--1674. MIT Press. 2011.
4. Sliced score matching: A scalable approach to density and score estimation Y. Song, S. Garg, J. Shi, S. Ermon. Uncertainty in Artificial Intelligence, pp. 574--584. 2020.

Score-based Generative Model——

Generate sample with score function

- 我们有得分模型 $s_\theta(x_i) \approx \nabla_{x_i} \log p(x_i)$
- 假设先验分布为高斯分布$\mathrm{x_T} = \mathrm{z} \sim \pi(x)$，采样$x_T$利用得分模型转变为目标分布的样本$x_0 \sim p(x)$。
- 注意，我们的模型描述的分布变化的方向
- 使用郎之万采样，通过多步迭代来获得目标分的采样数据，采样过程只用到分布的梯度表示

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p(x) + \sqrt{2\epsilon}\mathbf{z_i}$$
$$\mathbf{z_i} \in \mathcal{N}(0,1); i = T, \ldots, 1, 0$$



$\nabla_x \log p(x)$

Score-based Generative Model——

Score Trap

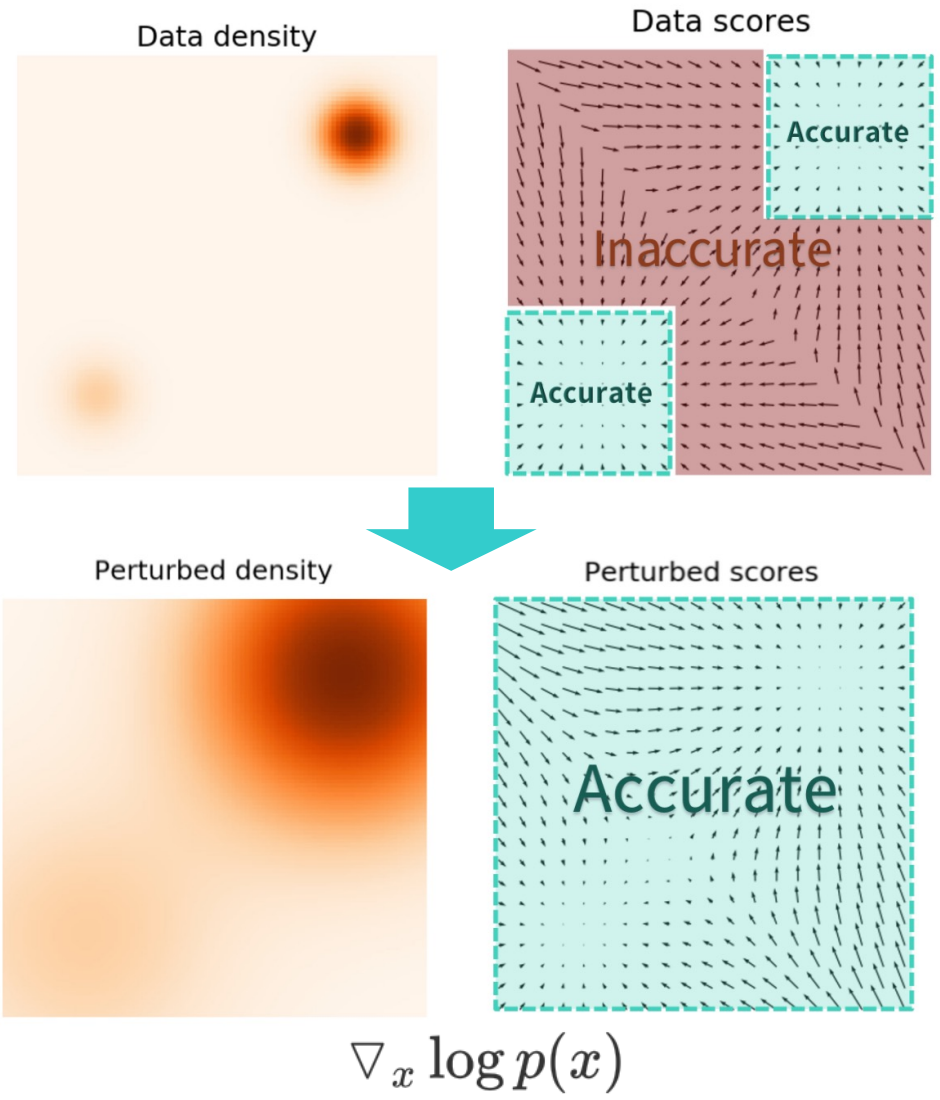
- 回顾：
  - 我们有得分模型： $s_\theta(x_i) \approx \nabla_{x_i} \log p(x_i)$
  - 模型通过Fisher散度作为优化目标：

$$\mathbb{E}_{p(x)}[\|\nabla_x \log p(x) - s_\theta(x)\|_2^2] = \int p(x)\|\nabla_x \log p(x) - s_\theta(x)\|_2^2 dx$$

  - 通过模型，借助郎之万采样生成目标数据

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p(x) + \sqrt{2\epsilon}\mathbf{z_i}$$
$$\mathbf{z_i} \in \mathcal{N}(0,1); i = T, \ldots, 1, 0$$

- 存在问题：得分函数代表分布密度，采样时会有低密度区。
- 解决方法：训练时对真实数据加入多尺度噪声扰动



Score-based Generative Model——

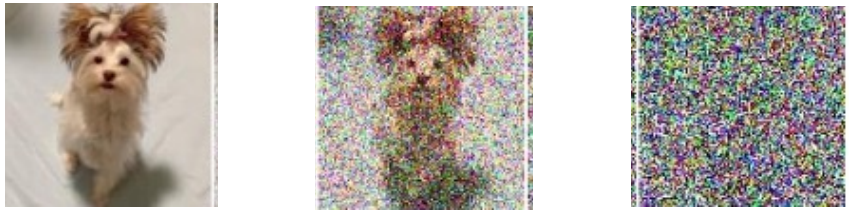
How to avoid Score Trap

- 回顾：当前的训练过程没有多步扩散过程。采样过程包含郎之万采样的多步迭代采样
- 如何填补低密度分布区域？加入噪声扰动，增大目标分布的范围
- 加多少噪声？过大噪声导致采样质量下降。
  - 我们希望借助多步采样，在采样不同阶段设计多尺度加噪。
  - 采样从高斯分布开始，可以在这个阶段使用更大噪声、
  - 因此在训练时，也引入 `时间步`；每个时间步加入不同噪声，分别学习不同时间步的得分函数

$$s_\theta(x,i) \approx \nabla_x \log p_{\sigma_i}(x), \sum_{i=1}^{T} \lambda(i)\mathbb{E}_{p_{\sigma_i}(x)}[\|\nabla_x \log p_{\sigma_i}(x) - s_\theta(x,i)\|_2^2]$$

$$x_i = x_{i-1} + \sigma_i \mathbf{z}, \mathbf{z} \sim \mathcal{N}(0,1)$$

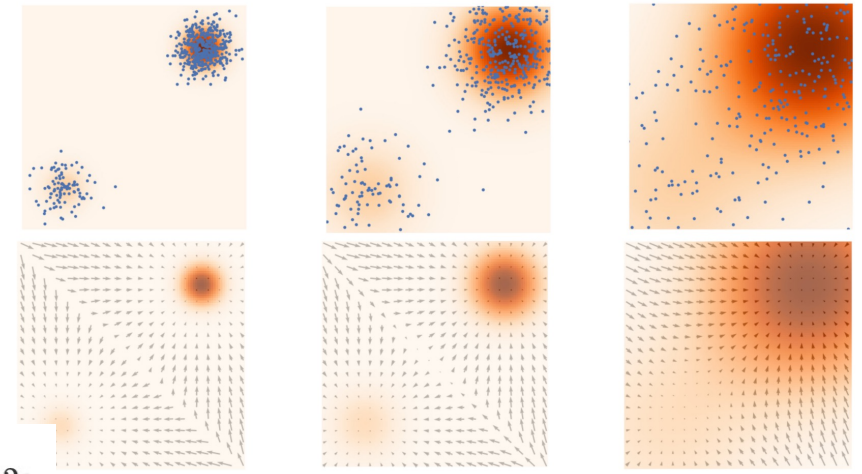
前向加噪



反向采样

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p_{\sigma_i}(x) + \sqrt{2\epsilon}\mathbf{z_i}$$

$\sigma_1 < \sigma_2 < \sigma_3$



$\nabla_x \log p(x)$

Diffusion Models

Background——
（1）diffusion probabilistic models（2015）

- 扩散在热力学指粒子从高密度区域扩散至低密度区域；在统计领域指将复杂分布转换为简单分布的过程，即

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

- 对于扩散模型，受 `非平衡统计物理学[2]` 启发，提出可以使用Markov过程构造 $\mathcal{T}$，这就可以通过一系列（T步）条件概率分布过渡到先验分布。
- 对于先验分布，最简洁有效的就是高斯分布
- 基于想法：平滑的目标分布，可以通过有限步的微小扰动，过渡到另一分布
- 构建了一个`破坏-恢复`过程的分布学习框架：
  - 通过迭代前向扩散过程系统地、缓慢地破坏数据分布中的结构。然后，学习一个反向扩散过程，该过程可以恢复数据中的结构

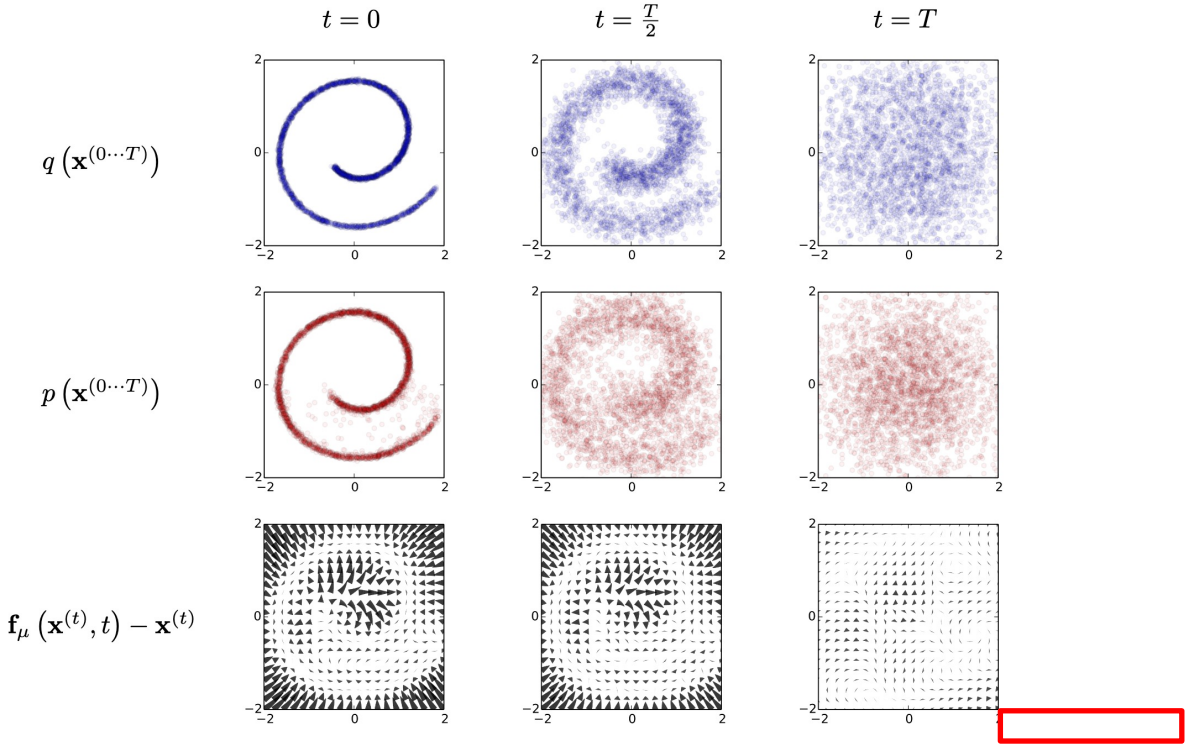


Figure 1. The proposed modeling framework trained on 2-d swiss roll data. The top row shows time slices from the forward trajectory $q\left(\mathbf{x}^{(0\cdots T)}\right)$. The data distribution (left) undergoes Gaussian diffusion, which gradually transforms it into an identity-covariance Gaussian (right). The middle row shows the corresponding time slices from the trained reverse trajectory $p\left(\mathbf{x}^{(0\cdots T)}\right)$. An identity-covariance Gaussian (right) undergoes a Gaussian diffusion process with learned mean and covariance functions, and is gradually transformed back into the data distribution (left). The bottom row shows the drift term, $\mathbf{f}_\mu\left(\mathbf{x}^{(t)}, t\right) - \mathbf{x}^{(t)}$, for the same reverse diffusion process.

1. Sohl-Dickstein J, Weiss E, Maheswaranathan N, et al. Deep unsupervised learning using nonequilibrium thermodynamics[C]//International Conference on Machine Learning. PMLR, 2015: 2256-2265.
2. Jarzynski C. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach[J]. Physical Review E, 1997, 56(5): 5018.=

Background——
（2）Hierarchical VAE（2015、2020等）

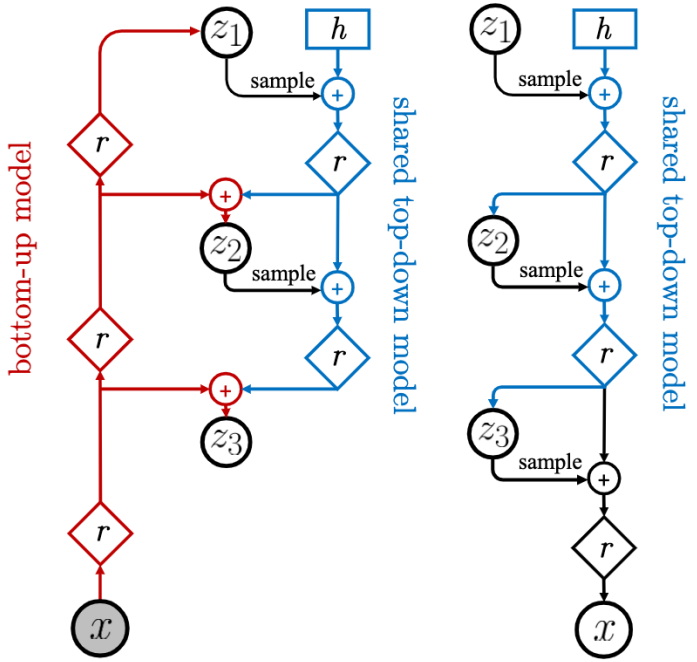
- VAE的主要缺陷在于：编码后验分布$p(z|x)$、$q(x|z)$的独立高斯分布不具备足够的拟合任意分布的能力
- HVAE从架构角度优化VAE。将潜在变量z分成不相交的多组高斯分布，不同组之间呈自回归模型的关系
- 即令：$z = \{z_1, z_2, \ldots, z_L\}$

$$q(z) = \prod_{l=1}^{L} q(z_l|z_{<l}), \quad p(z|x) = \prod_{l=1}^{L} p(z_l|z_{<l}, x)$$

$$q(z_l|z_{<l}) = \mathcal{N}\left(z_l; \mu(z_{<l}), \sigma^2(z_{<l})\right)$$

$$p(z_l|z_{<l}, x) = \mathcal{N}\left(z_l; \mu(z_{<l}) + \Delta\mu(z_{<l}, x), \sigma^2(z_{<l}) \otimes \Delta\sigma^2(z_{<l}, x)\right)$$

- 与扩散模型相比较，每一步的潜在变量仍然需要神经网络建模，拟合能力大大增强，但也不容易训练。HVAE具备了多状态隐编码的特点



(a) Bidirectional Encoder (b) Generative Model

Figure 2: The neural networks implementing an encoder $q(\boldsymbol{z}|\boldsymbol{x})$ and generative model $p(\boldsymbol{x}, \boldsymbol{z})$ for a 3-group hierarchical VAE. ◇r denotes residual neural networks, ⊕ denotes feature combination (e.g., concatenation), and [h] is a trainable parameter.

1. Vahdat A, Kautz J. NVAE: A deep hierarchical variational autoencoder[J]. Advances in neural information processing systems, 2020, 33: 19667-19679.
2. 强大的NVAE：以后再也不能说VAE生成的图像模糊了 - 科学空间|Scientific Spaces (kexue.fm)
3. Kingma D P, Salimans T, Jozefowicz R, et al. Improved variational inference with inverse autoregressive flow[J]. Advances in neural information processing systems, 2016, 29.

Diffusion Models

- Denoising Diffusion Probabilistic Models（DDPM）的出发点

  是结合NCSN对Original-Diffusion的改进

> (which we will call a "diffusion model" for brevity) is a parameterized Markov chain trained using variational inference to produce samples matching the data after finite time. Transitions of this chain are learned to reverse a diffusion process, which is a Markov chain that gradually adds noise to the data in the opposite direction of sampling until signal is destroyed.

- 数学角度推理：
  - 前向编码——加噪
  - 反向解码——去噪
  - 优化目标
  - 训练与采样

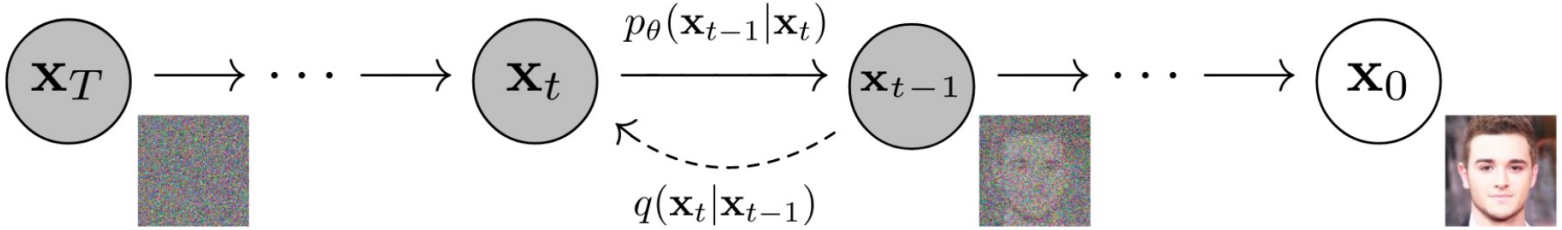
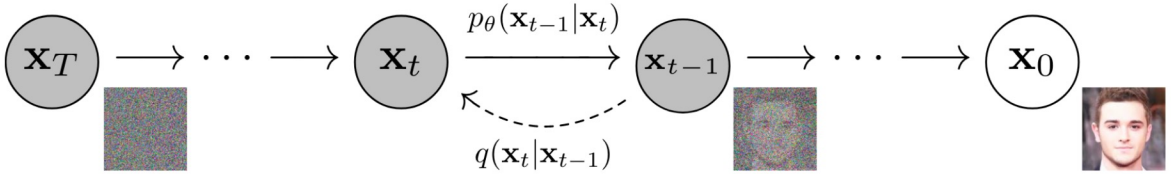


Figure 2: The directed graphical model considered in this work.

1. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.

Diffusion Models
——Forward Progress



- 给定目标数据分布 x~q(x), 生成模型要做的是将数据映射到高斯分布 z~π(x)=N(0,1)。然后从高斯分布采样z，将z解码到目标分布
- 对于DDPM实现分布的映射，其通过构造一系列，同样建模为高斯分布(VAE)的中间隐状态，借助辅助状态的互相转移，逐步到达高斯分布 。

$$q(x_{1:T}|x_0) = \prod_{i=1}^{T} q(x_t|x_{t-1})$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \quad , \beta_t \in (0,1)$$
$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_t \quad , \epsilon_t \sim \mathcal{N}(0,1)$$

- 具体地，
  - DDPM构建Markov的加噪过程
  - 引入隐变量 $\{x_i\}_{i=1}^{T}$ 作为Markov的各个状态。T为扩散的步长
  - 每一步的隐状态都基于前一状态转换，由条件概率分布q $(x_t|x_{t-1})$来实现
  - 重参数技巧以直接构建 x0→xt 映射
  - 加噪过程是固定的，不可学习的；系数α、β保证T步时达到先验分布

$$令\alpha_t = 1-\beta_t, \bar{\alpha}_t = \prod_{t=1}^{T}\alpha_i$$

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\mathbf{z}}_{t-2} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z} \\
q(x_t|x_0) &= \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I)
\end{aligned}$$

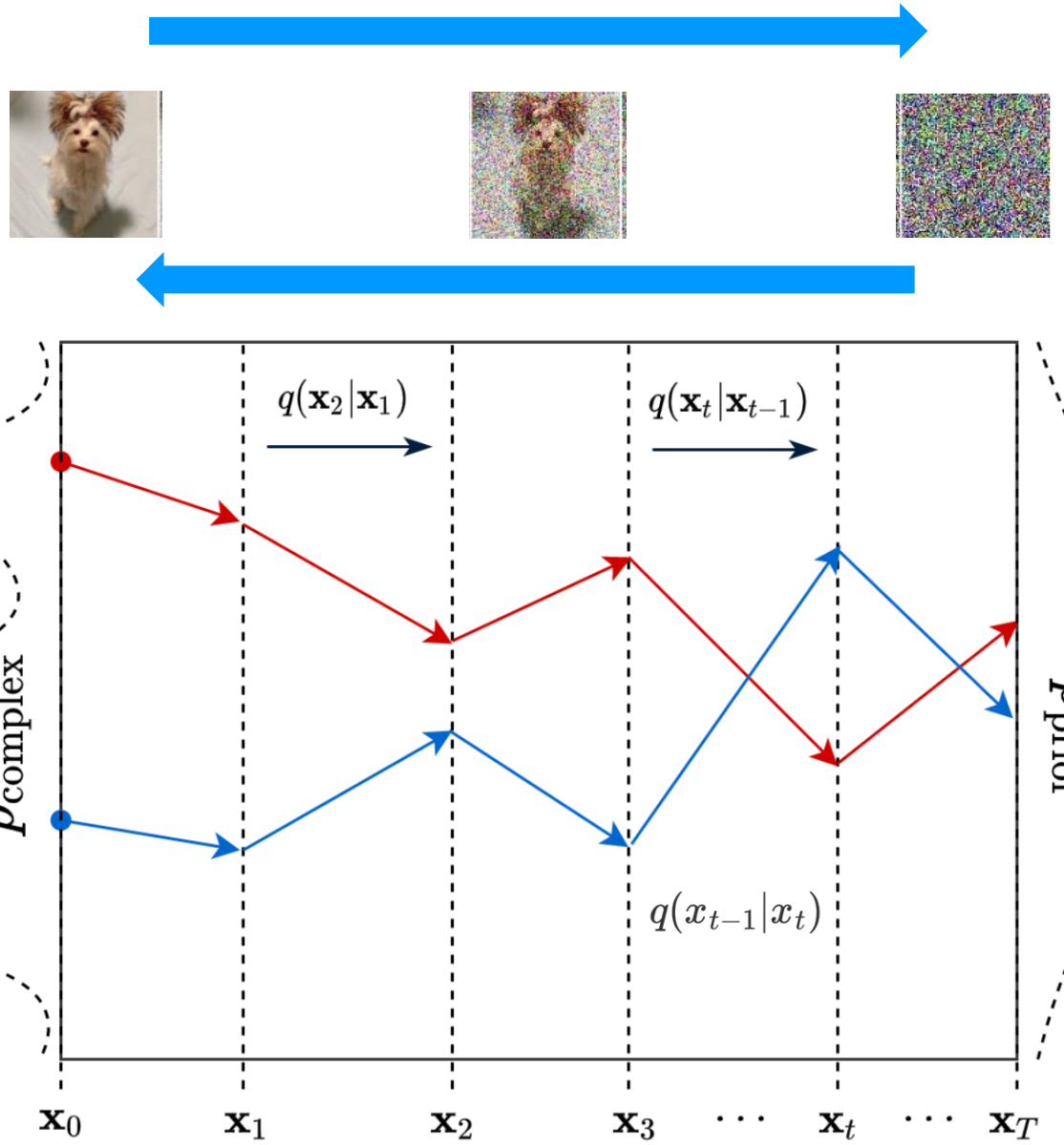
1. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.
2. What are Diffusion Models? | Lil'Log (lilianweng.github.io)

Diffusion Models
——Reverse Progress

- DDPM要做的，是根据上述手工设计的加噪过程 ，建模反向去噪

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

- 从先验分布中采样x_T，就可以利用$q(x_{t-1}|x_t)$多步去噪，获得目标样本x0



1. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Reverse Progress

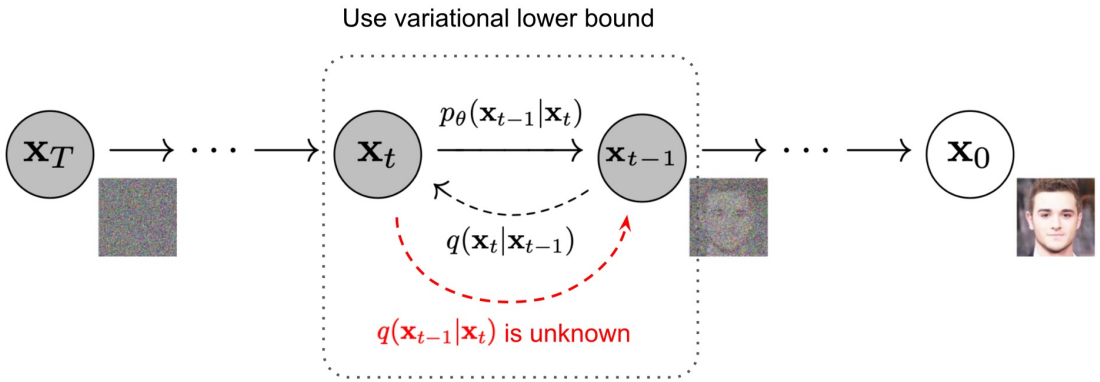
- Key: 反向转移分布 $q(x_{t-1}|x_t)$ 是什么样子？
  - 定理：连续扩散过程的逆转具有与正向过程相同的分布形式 [1]
  - 利用Bayes:

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

  - 回退，根据马尔科夫链性质，考虑在已知x0的条件下使用Bayes：

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

  - 该条件下，每一项都有解析式，因此等式可获得解析解
  - 可知 $q(x_{t-1}|x_t, x_0)$ 仍满足高斯分布
    - 方差项不包含未知数



$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t I)$$
$$q(x_{t-1}|x_0) = \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0, \sqrt{1-\bar{\alpha}_{t-1}}I)$$
$$q(x_t|x_0) = \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}I)$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_0, x_t), \tilde{\sigma}_t^2 I)$$
$$\tilde{\mu}_t(x_0, x_t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0$$
$$\tilde{\sigma}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

1. Feller W. On the theory of stochastic processes, with particular reference to applications[M]//Selected Papers I. Springer, Cham, 2015: 769-798.
2. 生成扩散模型漫谈（三）：DDPM = 贝叶斯 + 去噪 - 科学空间|Scientific Spaces (kexue.fm)
3. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Optimization Target

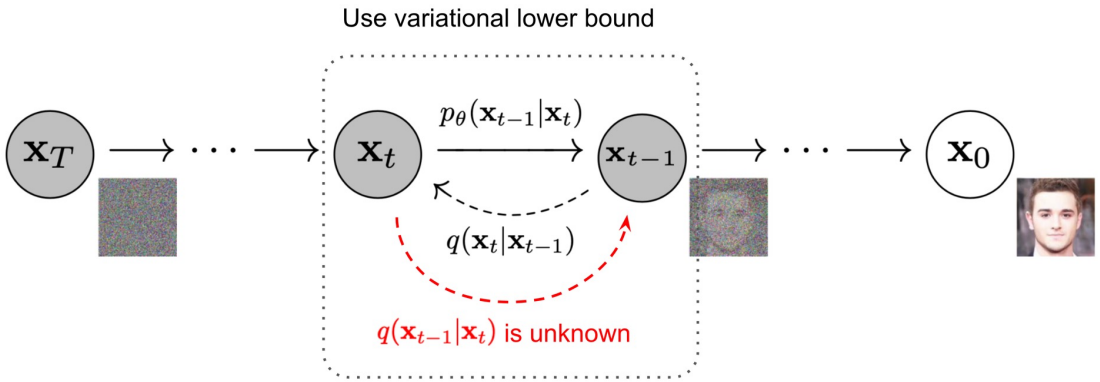
- 虽然有了解析式，但由于参数x0的存在，无法直接用来采样。
- 构建相似分布 $p_\theta(x_{t-1}|x_t) \approx q(x_{t-1}|x_t, x_0)$
- 只以xt为参数，且符合高斯分布，令

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t))$$

- 优化目标则最小化两个分布p、q的KL散度
  - 两个多元正态分布的KL散度可直接根据分布参数获得：

$$L = \mathbb{E}_q[\frac{1}{2\|\sigma_\theta(x_t, t)\|_2^2}\|\tilde{\mu}_t(x_0, x_t) - \mu_\theta(x_t, t)\|^2] + C$$

  - 这里由于方差项 $\tilde{\sigma}_t$ 都为常数，可以直接等价



$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_0, x_t), \tilde{\sigma}_t^2 I)$$

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

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

优化目标1：均值loss

$$\|\tilde{\mu}_t(x_0, x_t) - \mu_\theta(x_t, t)\|^2$$

1. 生成扩散模型漫谈（三）：DDPM = 贝叶斯 + 去噪 - 科学空间|Scientific Spaces (kexue.fm)
2. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Recall Optimization

- 从模型角度回顾优化过程：
  - 我们使用神经网络 $\mu_\theta(x_t, t)$,参数为（xt，t），预测均值 $\tilde{\mu}_t(x_0, x_t)$ 在训练时，采样样本x0和t，根据q(xt|x0)获得xt, 利用右式 $\tilde{\mu}_t(x_0, x_t)$ ，结合x0,xt获得ground truth；将(xt, t)送入网络获得输出，与ground truth做L2损失优化

- 与其他生成模型类似，最根本的优化目标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q(x_0)和$p_\theta(x_0)$的交叉熵：

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

  - 具体推导过程可参考[2]

优化目标1：均值loss

$$\|\tilde{\mu}_t(x_0, x_t) - \mu_\theta(x_t, t)\|^2$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_0, x_t), \tilde{\sigma}_t^2 I)$$

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

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

$$L = \mathbb{E}_q\left[\frac{1}{2\|\sigma_\theta(x_t, t)\|_2^2}\|\tilde{\mu}_t(x_0, x_t) - \mu_\theta(x_t, t)\|^2\right] + C$$

1. 生成扩散模型漫谈（三）：DDPM = 贝叶斯 + 去噪 - 科学空间|Scientific Spaces (kexue.fm)
2. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Optimization Target

- 回想VAE对均值建模，其表达能力是有限的。DDPM做了进一步简化
- 考虑下式，x0是作为未知数出现的，那能否直接通过xt预测x0？

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

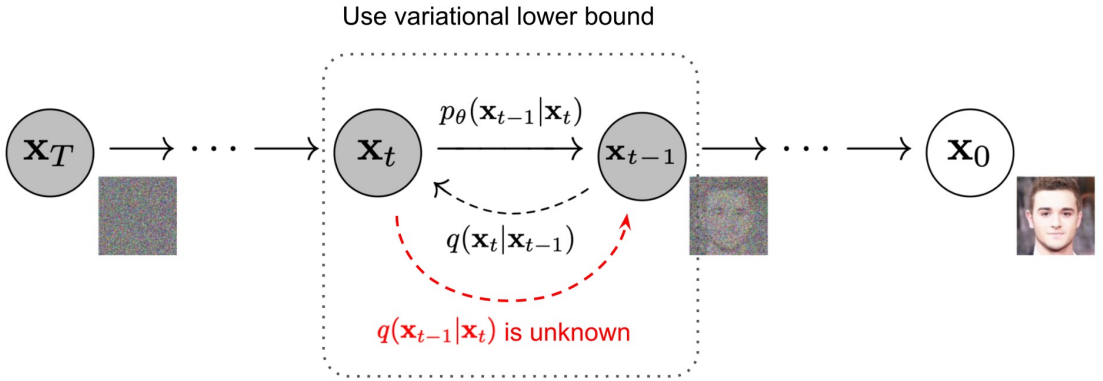
- 尝试使用神经网络 $x_\theta(x_t, t)$ 预测x0。带入到上式可得：

$$\mu_\theta(x_0, x_t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{x}_\theta(x_t, t)$$

- 这种情况下，直接以神经网络的输出和x0作L2 运算即可优化模型。

$$\left\| x_0 - \hat{x}_\theta(x_t, t) \right\|^2$$

- 从这个角度讲，x0为原始图像，xt为加噪图像，这个神经网络的本质就是去噪模型，也就是DDPM第一个`D`(Denosing)的含义



优化目标2：图像loss

$$\left\| x_0 - \hat{x}_\theta(x_t, t) \right\|^2$$

1. 生成扩散模型漫谈（三）：DDPM = 贝叶斯 + 去噪 - 科学空间|Scientific Spaces (kexue.fm)
2. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Optimization Target

- 然而不同阶段xt的加噪程度是不同的，在接近T时刻，xt接近高斯分布，从xt直接预测x0有困难
- 再次简化x0。根据q(xt|x0) ，可以得到x0与xt的转变关系：

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\bar{\epsilon}_t$$

$$\Rightarrow \quad x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\bar{\epsilon}_t}{\sqrt{\bar{\alpha}_t}}$$

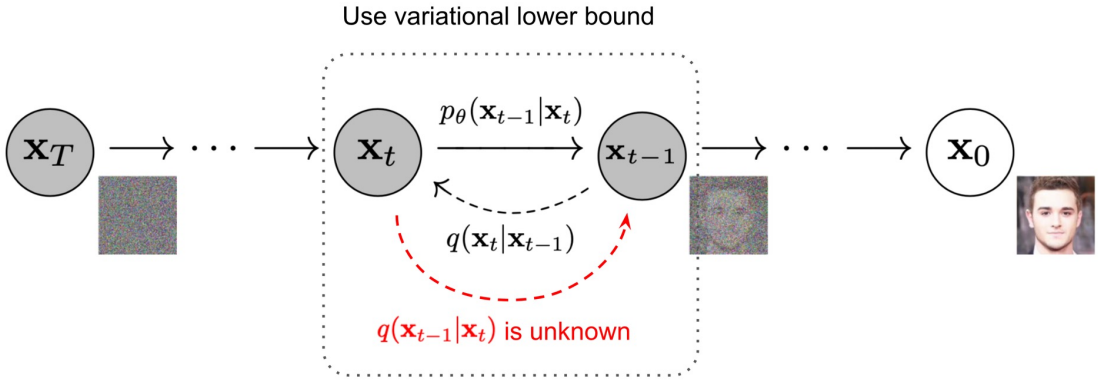
- 将x0带入右式 $\tilde{\mu}_t(x_0, x_t)$ 得：

$$\tilde{\mu}_t(x_0, x_t) = \frac{1}{\sqrt{\alpha_t}}x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\epsilon$$

- 这里的未知数ϵ代表噪声。这样我们就可以用神经网络$\hat{\epsilon}_\theta(x_t, t)$去拟合噪声

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\right]$$

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



$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_0, x_t), \tilde{\sigma}_t^2 I)$$

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

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

优化目标3：噪声loss

$$\|\epsilon - \hat{\epsilon}_\theta(x_t, t)\|^2$$

1. What are Diffusion Models? | Lil'Log (lilianweng.github.io)
2. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Predictor-Corrector

- 在构建相似分布来预测下一步xt时

$$p_\theta(x_{t-1}|x_t) \approx q(x_{t-1}|x_t, x_0)$$

- 我们曾试图使用神经网络 $x_\theta(x_t, t)$预测x0

$$\mu_\theta(x_0, x_t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{x}_\theta(x_t, t)$$

- 如果能直接预测准确x0，直接一步到位而不必多步迭代采样？
  - 首先当然是在初始阶段无法完全准确去噪；
  - 之所以预测x0，是起到前瞻性的预估作用，得到去噪的大体方向；
  - 然后利用 $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 前向调整一小步
  - 即在大方向上粗略预估，跟鱼粗略的预估方向精确调节一小步。

1. What are Diffusion Models? | Lil'Log (lilianweng.github.io)
2. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Training and Sampling

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$x_0 = \mu_\theta$

1. What are Diffusion Models? | Lil'Log (lilianweng.github.io)
2. 什么是Diffusion模型？ - wrong.wang

Diffusion Models
——Code Example

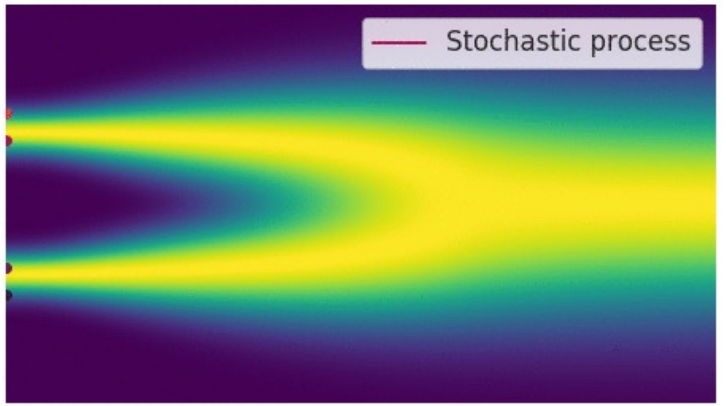
```python
betas = torch.linspace(start=0.0001, end=0.02, steps=1000)
alphas = 1 - betas
# cumprod 相当于为每个时间步 t 计算一个数组 alphas 的前缀乘结果
# https://pytorch.org/docs/stable/generated/torch.cumprod.html
alphas_cum = torch.cumprod(alphas, 0)
alphas_cum_s = torch.sqrt(alphas_cum)
alphas_cum_sm = torch.sqrt(1 - alphas_cum)

def diffusion_loss(unet_model, x0, t, noise):
    # 根据公式计算 xt
    xt = alphas_cum_s[t] * x0 + alphas_cum_sm[t] * noise
    # 模型预测噪声
    predicted_noise = unet_model(xt, t)
    # 计算Loss
    return mse_loss(predicted_noise, noise)

for i in len(data_loader):
    # 从数据集读取一个 batch 的真实图片
    x0 = next(data_loader)
    # 采样时间步
    t = torch.randint(0, 1000, (batch_size,))
    # 生成高斯噪声
    noise = torch.randn_like(x_0)
    loss = diffusion_loss(model, x0, t, noise)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

1. 一文弄懂 Diffusion Model

Unified Perspective

Unified Perspective——

Score stochastic differential equations (Score SDE)



- Score-based GM 和 Diffusion Model具有密切联系，从不同角度说明了加噪-去噪的过程
- Score SDE从随机微分方程角度对两者进行统一[1]：
  - 从目标分布到高斯分布过程，我们设置有穷的T步(T=1000)作为扩散过程
  - 当扩散步T→∞, 噪声扰动过程可以视为连续时间的随机过程
  - 如何描述一个随机过程（扩散过程）？随机微分方程(SDE)

$$\mathrm{d}x = f(x,t)\mathrm{d}t + g(t)\mathrm{d}w$$
$$f(\cdot,t) : \mathbb{R}^d \Rightarrow \mathbb{R}^d, g(t) \in \mathbb{R}$$

$$x_{t+\triangle t} - x_t = f(x_t,t) \triangle t + g(t)\sqrt{\triangle t}\epsilon$$
$$,\epsilon \in \mathcal{N}(0,1)$$

$$p(x_{t+\triangle t}|x_t) = \mathcal{N}(x_{t+\triangle t}; x_t + f(x_t,t) \triangle t, g(t)^2 \triangle t\mathrm{I})$$

1. Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-Based Generative Modeling through Stochastic Differential Equations[C]//International Conference on Learning Representations.
2. 生成扩散模型漫谈（五）：一般框架之SDE篇 - 科学空间|Scientific Spaces (kexue.fm)
3. Understanding Diffusion Models: A Unified Perspective (calvinyluo.com)

Unified Perspective——

stochastic differential equations (SDE)

$$\mathrm{d}x = f(x,t)\mathrm{d}t + g(t)\mathrm{d}w$$
$$f(\cdot,t):\mathbb{R}^d \Rightarrow \mathbb{R}^d, g(t) \in \mathbb{R}$$

- 随机微分方程SDE描述连续时间变化上的随机性扩散过程
  - f(·,t)称为漂移系数，在xt原数据空间做映射
  - g(t)称为扩散系数，控制随机性的大小
  - w代表一个标准布朗运动，表达随机性。dw可视为无限小的随机噪声
- 随机微分方程的解是一系列的随机状态集合$\{x(t)\}_{t\in[0,T]}$ 代表上文中建立的T个加噪状态
- 我们令p(xt)描述x(t)状态的概率密度函数，那么p(x0)、p(xT)分别代表了目标分布与先验分布
- NCSN 和 DDPM都有对应的SDE表示

NCSN

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}$$
$$\Rightarrow \mathrm{d}x = \sqrt{\frac{\mathrm{d}[\sigma^2(t)]}{\mathrm{d}t}}\mathrm{d}w$$

DDPM

$$x_i = \sqrt{1-\beta_i}x_{i-1} + \sqrt{\beta_i} z_{i-1}$$
$$\Rightarrow \mathrm{d}x = -\frac{1}{2}\beta(t)x\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}w$$

$$, z_{i-1} \sim \mathcal{N}(0,1), i = 1,2,\cdots,N$$

1. Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-Based Generative Modeling through Stochastic Differential Equations[C]//International Conference on Learning Representations.
2. 生成扩散模型漫谈（五）：一般框架之SDE篇 - 科学空间|Scientific Spaces (kexue.fm)
3. Understanding Diffusion Models: A Unified Perspective (calvinyluo.com)

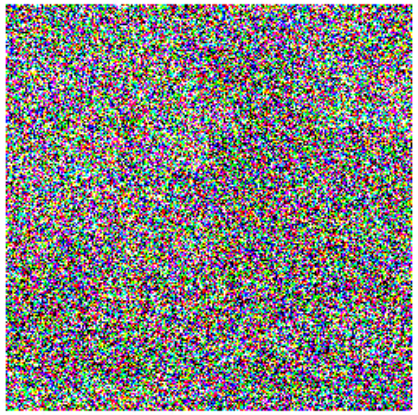
Unified Perspective——

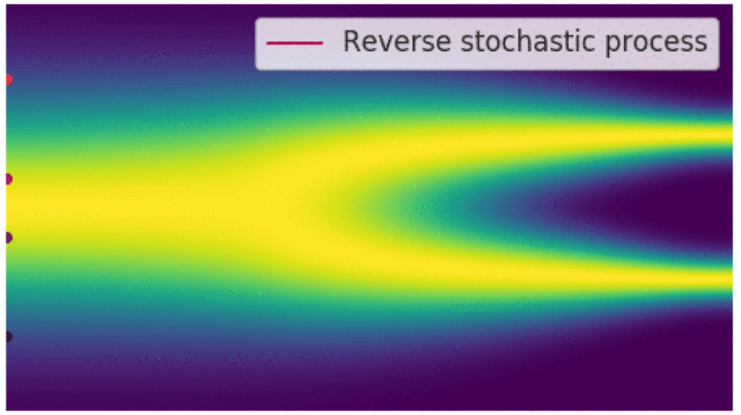
Reverse SDE

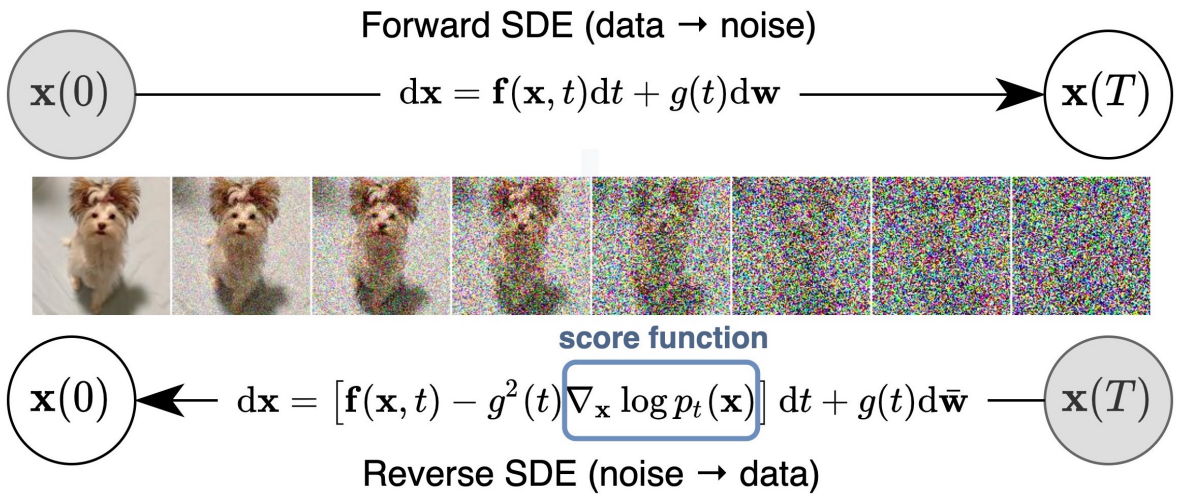
- 回顾：对于离散时间的Score-based Model，我们可以使用郎之万采样从每个噪声状态中顺序采样
- 对于连续时间的无限噪声，我们需要求解反向SDE（任意SDE都有其对应逆向时间的SDE [2]）

$$\mathrm{d}x = [\mathrm{f}(x,t) - g^2(t)\nabla_x \log p_t(x)]\mathrm{d}t + g(t)\mathrm{d}w$$

  - 这里反向的含义指时间t为T→0
- 在反向SDE中，重新出现了得分函数 $\nabla_x \log p_t(x)$ 我们需要估计得分函数以方便建模反向SDE







1. Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-Based Generative Modeling through Stochastic Differential Equations[C]//International Conference on Learning Representations.
2. Anderson B D O. Reverse-time diffusion equation models[J]. Stochastic Processes and their Applications, 1982, 12(3): 313-326.
3. 生成扩散模型漫谈（五）：一般框架之SDE篇 - 科学空间|Scientific Spaces (kexue.fm)

Unified Perspective——

Estimating the Reverse SDE

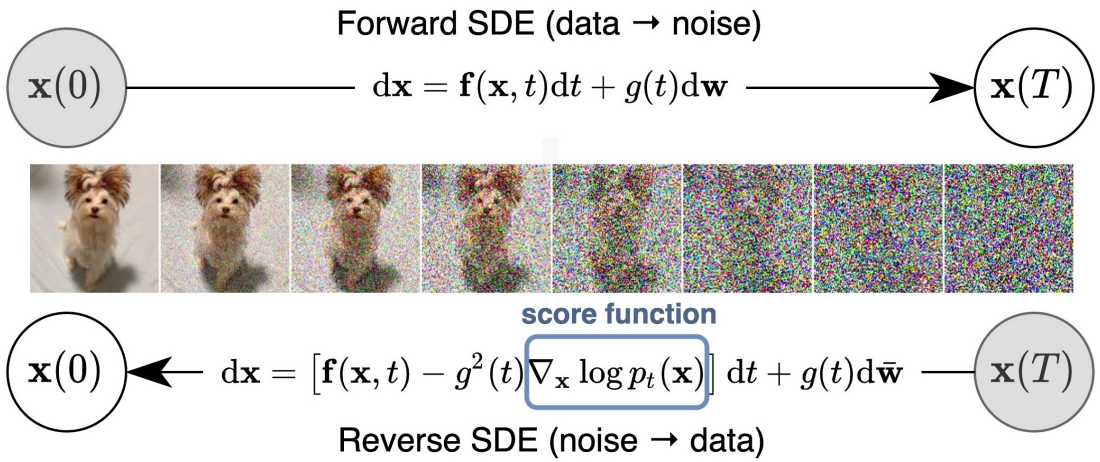
- 求解反向SDE最主要的就是估计得分函数。与NCSN同样的思路，用神经网络拟合得分函数并用Score Matching优化：

$$s_\theta(x,t) \approx \nabla_x \log p_t(x)$$

$$\mathbb{E}_{t\in\mu(0,T)}\mathbb{E}_{p_t(x)}[\lambda(t)\|\nabla_x \log p_t(x) - s_\theta(x,t)\|_2^2]$$

$$\lambda(t) \propto 1/\mathbb{E}[\|\nabla_{x(t)} \log p_t(x_t|x_0)\|]$$

  - 这里μ代表时间间隔内的连续分布
  - $\lambda(t)$作为一个加权平均项，抵消无关项，调节Loss权重
- 与先前相同，模型训练可以借助Score Matching技巧简化Loss的形式。
- 使用训练好后的网络替代得分函数，就得到反向SDE的表达式



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

1. Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-Based Generative Modeling through Stochastic Differential Equations[C]//International Conference on Learning Representations.
2. Anderson B D O. Reverse-time diffusion equation models[J]. Stochastic Processes and their Applications, 1982, 12(3): 313-326.
3. 生成扩散模型漫谈（五）：一般框架之SDE篇 - 科学空间|Scientific Spaces (kexue.fm)

Unified Perspective——

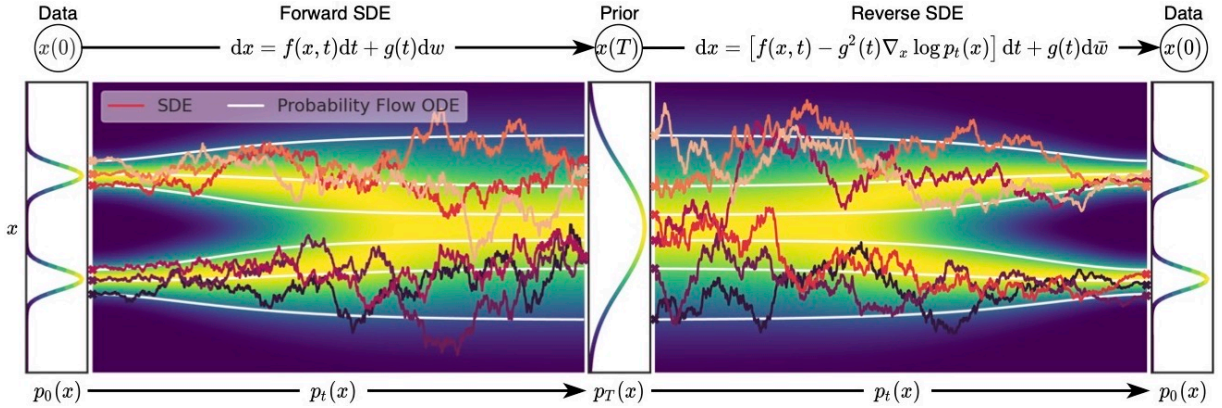
Solving Reverse SDE to sampling

- Reverse SDE包含了从先验高斯分布到目标分布的连续时间状态。
- 给定先验分布的采样xT，设定一个时间间隔Δt，我们就可以根据 Reverse SDE 求解器 获得状态$x_{T+\Delta t}$。迭代足够步数，即可以获得t=0状态下属于目标分布的样本x_0

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

Euler-Maruyama method SDE Solver：

$$x_t - x_{t+\triangle t} = -[f_{t+\triangle t}(x_{t+\triangle t}) - g^2_{t+\triangle t}\nabla_{x_{t+\triangle t}}\log p_t(x_{t+\triangle t})]\ \triangle\, t + g_{t+\triangle t}\sqrt{\triangle t}\epsilon$$
$$,\epsilon \sim \mathcal{N}(0,1)$$



- 概率流ODE 采样
  - [1]中证明任何SDE都可以在不改变边缘分布p(xt)的条件下转换为对应的常微分方程ODE，进而利用ODE做确定性采样

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

  - 大部分围绕加速采样的工作都基于ODE Solver加速计算
    - 龙格 - 库塔自适应步长ODE求解器[1]
    - 高阶自适应步长SDE求解器[2]
    - 重参数化的，平滑ODE[3]
    - 使用线性多步的高阶ODE求解器[4]
    - 指数ODE积分器[5,6]
    - 使用Heun方法的高阶ODE求解器[7]

1. Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, ICLR, 2021
2. Martineau et al., “Gotta Go Fast When Generating Data with Score-Based Models”, arXiv, 2021
3. Song et al., “Denoising Diffusion Implicit Models”, ICLR, 2021
4. Liu et al., “Pseudo Numerical Methods for Diffusion Models on Manifolds”, ICLR, 2022
5. Zhang and Chen, “Fast Sampling of Diffusion Models with Exponential Integrator”, arXiv, 2022
6. Lu et al., “DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps”, arXiv, 2022
7. Karras et al., “Elucidating the Design Space of Diffusion-Based Generative Models”, arXiv, 2022

Unified Perspective——

How to design our own Diffusion Process

- 前向过程：根据SDE(或其离散表示)定义前向过程
- 设计f(x,t)形式，保证q(xt|x0)是可以获得解析式的
- 得分匹配：通过损失函数训练网络 $s_\theta(x,t)$
- 反向过程：通过离散化反向SDE反向采样

$$
\begin{aligned}
\mathrm{d}x &= f(x,t)\mathrm{d}t + g(t)\mathrm{d}w \\
x_{t+\triangle t} - x_t &= f(x_t,t)\ \triangle\, t + g(t)\sqrt{\triangle t}\epsilon \\
&\Downarrow \\
p(x_t|x_0) &= \mathcal{N}(x_t;\overline{\alpha}_t x_0, \overline{\beta}_t^2 \mathrm{I}) \\
&\Downarrow \\
s_\theta(x,t) &\approx \nabla_x \log p_t(x) \\
&\Downarrow \\
\mathrm{d}x &= [\mathrm{f}(x,t) - g^2(t)s_\theta(x,t)]\mathrm{d}t + g(t)\mathrm{d}w \\
x_t - x_{t+\triangle t} &= -[f_{t+\triangle t}(x_{t+\triangle t}) - g^2_{t+\triangle t}s_\theta(x_{t+\triangle t}, t+\triangle t)]\ \triangle\, t + g_{t+\triangle t}\sqrt{\triangle t}\epsilon
\end{aligned}
$$

- ** p(xt|x0)的解析式约束主要保证得分函估计时的加权平均（？）*

$$p_t(\boldsymbol{x}_t) = \int p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)p_0(\boldsymbol{x}_0)d\boldsymbol{x}_0 = \mathbb{E}_{\boldsymbol{x}_0\sim p_0(\boldsymbol{x}_0)}\left[p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\right]$$

1. Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-Based Generative Modeling through Stochastic Differential Equations[C]//International Conference on Learning Representations.
2. Anderson B D O. Reverse-time diffusion equation models[J]. Stochastic Processes and their Applications, 1982, 12(3): 313-326.
3. 生成扩散模型漫谈（五）：一般框架之SDE篇 - 科学空间|Scientific Spaces (kexue.fm)

Conditional Guidance

Conditional Guidance

- 生成任务：构建目标分布p(x)；对于条件生成任务，给定条件输入y（类别、文本或图像），构建条件引导的目标分布p(x|y)
- 从得分匹配的角度，建模条件分布的得分函数 $\nabla_{x_t} \log p(x_t|y)$
  - 能否直接构造条件神经网络，使得

$$s_\theta(x_t, t, y) \approx \nabla_{x_t} \log p(x_t|y)$$

  - 网络会忽略或淡化条件信息，需要更加明确的控制模型的条件信息权重
- 借助贝叶斯公式，我们可以得到：
  - 条件得分函数可以简单视为非条件得分和条件项的和
  - 条件项 $\nabla_x \log p(y|x)$ 并非得分函数，因为是对x求梯度而非y

$$p(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p(x_{t-1}|x_t)$$

$$p(x_{0:T}|y) = p(x_T) \prod_{t=1}^{T} p(x_{t-1}|x_t, y)$$

$$p(x_t|y) = \frac{p(y|x_t) \cdot p(x_t)}{p(y)}$$

$$\Rightarrow \quad \log p(x_t|y) = \log p(y|x_t) + \log p(x_t) - \log p(y)$$

$$\Rightarrow \quad \nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(y|x_t) + \nabla_{x_t} \log p(x_t) - 0$$

1. Ho J, Salimans T. Classifier-free diffusion guidance[J]. arXiv preprint arXiv:2207.12598, 2022.
2. Understanding Diffusion Models: A Unified Perspective (calvinyluo.com)
3. Guidance: a cheat code for diffusion models – Sander Dieleman

Conditional Guidance——

Classifier Guidance

- 根据得分函数的条件展开，想到添加一个预测条件信息的分类器，来指导整体的得分
  - 有点类似gan的adversarial gradient classifier概念
  - 在拉格朗日采样生成时，通过计算原始的非条件得分和噪声分类器的梯度来指导
- 加入权重来控制条件引导的程度

$$p(x_t|y) = \frac{p(y|x_t) \cdot p(x_t)}{p(y)}$$

$$\Rightarrow \quad \log p(x_t|y) = \log p(y|x_t) + \log p(x_t) - \log p(y)$$

$$\Rightarrow \quad \nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p(y|x_t) + \nabla_{x_t} \log p(x_t) - 0$$

$$\nabla_{x_t} \log p(x_t|y) = \gamma \nabla_{x_t} \log p(y|x_t) + \nabla_{x_t} \log p(x_t)$$

1. Ho J, Salimans T. Classifier-free diffusion guidance[J]. arXiv preprint arXiv:2207.12598, 2022.
2. Guidance: a cheat code for diffusion models – Sander Dieleman

Conditional Guidance——

Classifier-free Guidance

$$\nabla_{x_t} \log p(x_t|y) = \gamma \nabla_{x_t} \log p(y|x_t) + \nabla_{x_t} \log p(x_t)$$

- Classifier Guidance存在先天的缺陷：
  - 分类器除了需要对原始图像分类，还需要学习对大部分的噪声图像xt分类
  - 必须对预训练好的分类器对噪声图像做微调
  - 同样由于噪声图像，在接近T时与原图像相似性很低，采样初期无法有效指导

$$\nabla_{x_t} \log p(y|x_t) = \nabla_{x_t} \log p(x_t|y) - \nabla_{x_t} \log p(x_t)$$

- 对分类得分项再次用贝叶斯展开：
- 构建条件得分与非条件得分的统一表示

$$\begin{aligned}\nabla_{x_t} \log p(x_t|y) &= \gamma \nabla_{x_t} \log p(y|x_t) + \nabla_{x_t} \log p(x_t) \\ &= \gamma(\nabla_{x_t} \log p(x_t|y) - \nabla_{x_t} \log p(x_t)) + \nabla_{x_t} \log p(x_t) \\ &= \underbrace{\gamma \nabla_{x_t} \log p(x_t|y)} + \underbrace{(1-\gamma) \nabla_{x_t} \log p(x_t)}\end{aligned}$$

1. Ho J, Salimans T. Classifier-free diffusion guidance[J]. arXiv preprint arXiv:2207.12598, 2022.
2. Guidance: a cheat code for diffusion models – Sander Dieleman

Conditional Guidance——

Classifier-free Guidance

- 如先前提到的，构造两个模型，分别建模条件得分和非条件得分。
  - 为了简化，统一使用一个模型（参数共享），模型输入为(x,y)。对于无条件生成时令y=0即可。
- 有效性在于，当$\gamma > 1$时，模型不近有限考虑条件分数项，而且会远离无条件得分函数的移动方向
- 通过负分项的无条件似然，降低了样本的无条件似然性，同时增加了条件似然性

$$\begin{aligned}\nabla_{x_t}\log p(x_t|y) &= \gamma\nabla_{x_t}\log p(y|x_t) + \nabla_{x_t}\log p(x_t)\\ &= \gamma(\nabla_{x_t}\log p(x_t|y) - \nabla_{x_t}\log p(x_t)) + \nabla_{x_t}\log p(x_t)\\ &= \underbrace{\gamma\nabla_{x_t}\log p(x_t|y)}_{\text{conditional score}} + \underbrace{(1-\gamma)\nabla_{x_t}\log p(x_t)}_{\text{unconditional score}}\end{aligned}$$

1. Ho J, Salimans T. Classifier-free diffusion guidance[J]. arXiv preprint arXiv:2207.12598, 2022.
2. Guidance: a cheat code for diffusion models – Sander Dieleman

Temporospatial Application of Diffusion

Temporospatial Application
《Video Diffusion models》(2022)

- Main Contribution:
  - Jointly models entire videos (blocks of frames) using a 3D video architecture (Conv3d) with interleaved spatial and temporal attention
  - Conditional Sampling method for video consistency: Adding conditional guidance for target frames sampling：

$$\tilde{\mathbf{x}}_\theta^b(\mathbf{z}_t) = \hat{\mathbf{x}}_\theta^b(\mathbf{z}_t) - \frac{w_r \alpha_t}{2} \nabla_{\mathbf{z}_t^b} \|\mathbf{x}^a - \hat{\mathbf{x}}_\theta^a(\mathbf{z}_t)\|_2^2 .$$

- Experiment:

Task	Benchmark	Resolution	Conditional Frames	Target Frames
Unconditional Video Generation:	Soomro	64x64	0	16
"Video Prediction"	BAIR Robot Pushing	64x64	1	15
	Kinetics-600	64x64	5	11

1. Cao H, Tan C, Gao Z, et al. A survey on generative diffusion model[J]. arXiv preprint arXiv:2209.02646, 2022.
2. Ho J, Salimans T, Gritsenko A A, et al. Video Diffusion Models[C]//Advances in Neural Information Processing Systems.

Temporospatial Application

《MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation》(2022)

- Main Contribution:
  - Design the mask manner with randomly mask past/future frames for multiple video tasks by a single model.
  - Separately concatenate the past/future conditional frames and the noisy current frames in the channel dimension for video diffusion with Conv2d

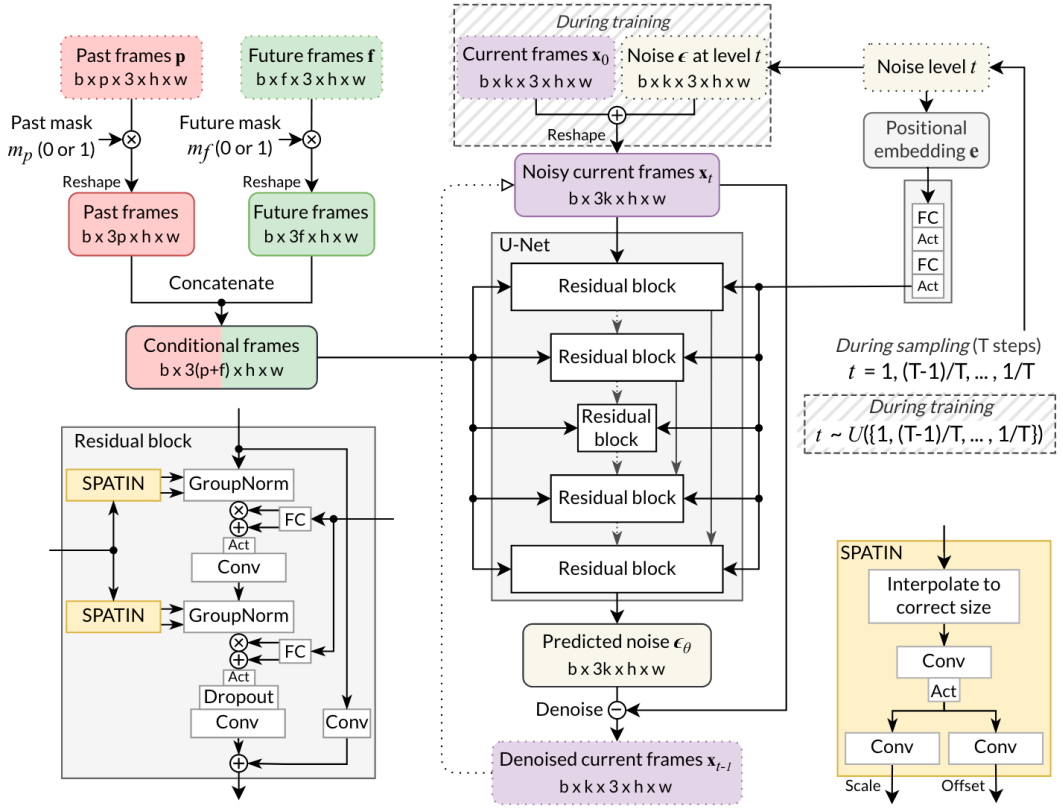


Figure 3: We give noisy current frames to a U-Net whose residual blocks receive conditional information from past/future frames and noise-level. The output is the predicted noise in the current frames, which we use to denoise the current frames. At test time, we start from pure noise.

1. Voleti V, Jolicoeur-Martineau A, Pal C. MCVD-Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation[C]//Advances in Neural Information Processing Systems..

Temporospatial Application

《MCVD》(2022)

- Experiment:

Task	Benchmark	Description	Resolution	Conditional Frames	Target Frames
Unconditional Video Generation:	BAIR	color, multiple objects, simple scene	64x64	0	5/16
	UCF101	color, 101 categories of natural scenes	64x64	0	4/16
"Video Prediction"	Moving MNIST	black-and-white digits	64x64	5	5/10
	KTH	grayscale single-humans	64x64	10	5/30
	BAIR		64x64	1	5/15
	Cityscapes	Color, natural complex natural driving scene	128x128	2	5/28
Interpolation	Moving MNIST, KTH, and BAIR		64x64	5+5	5/10

1. Voleti V, Jolicoeur-Martineau A, Pal C. MCVD-Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation[C]//Advances in Neural Information Processing Systems..

Temporospatial Application

《Video Probabilistic Diffusion Models in Projected Latent Space》(2023)

- Main Contribution:
  - Learning video representation (3D→ 2D) in 3 discrete latent codes and applying diffusion model on such factorized latent space.

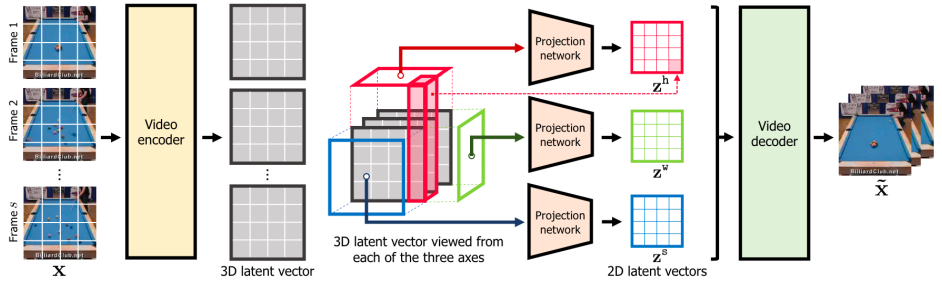


Figure 2. Detailed illustration of our autoencoder architecture in PVDM framework ((a) in Figure 1).

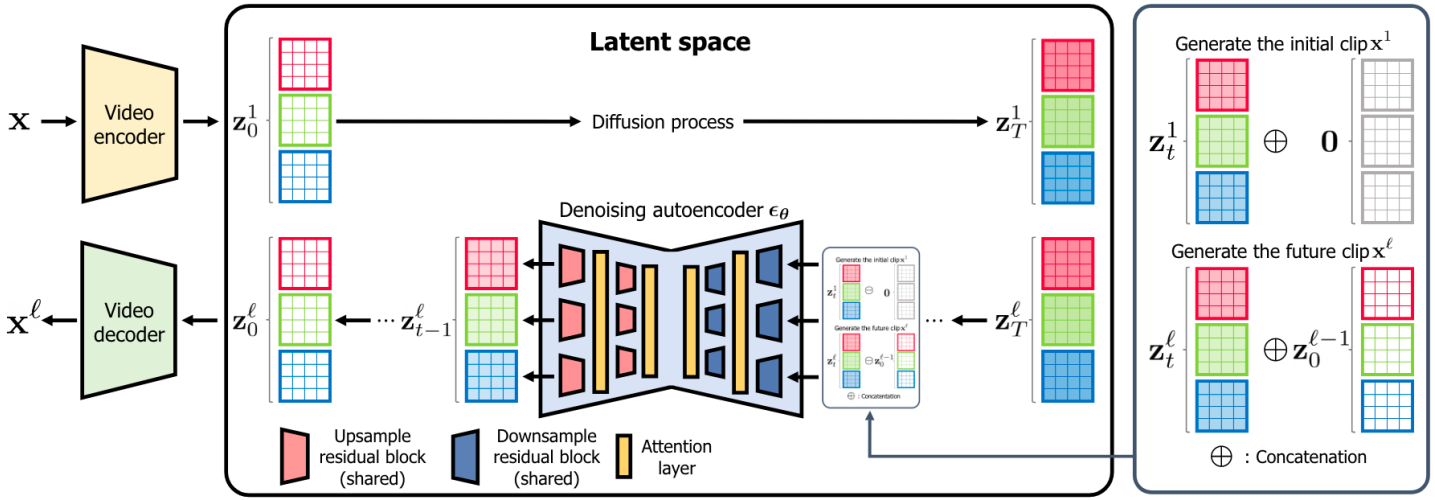


Figure 1. Overall illustration of our projected latent video diffusion model (PVDM) framework. PVDM is composed of two components: (a) (left) an autoencoder that maps a video into 2D image-like latent space (b) (right) a diffusion model operates in this latent space.

1. Yu S, Sohn K, Kim S, et al. Video Probabilistic Diffusion Models in Projected Latent Space[J]. arXiv preprint arXiv:2302.07685, 2023. (CVPR 2023)

Temporospatial Application
《Video Probabilistic Diffusion Models in Projected Latent Space》(2023)

$$\mathbb{E}_{(\mathbf{x}_0^1,\mathbf{x}_0^2),\epsilon,t}\Big[\lambda||\epsilon-\epsilon_\theta(\mathbf{z}_t^2,\mathbf{z}_0^1,t)||_2^2 + (1-\lambda)||\epsilon-\epsilon_\theta(\mathbf{z}_t^2,\mathbf{0},t)||_2^2\Big],$$

- Video Generation Task:
  - Like Classifier-free Condition to train autoregressive generative model.
  - Sample an initial video clip x1 ~ pθ(x), and repeat generating next clip x`+1 ~ pθ(x`+1|x`) conditioned on the previous clip.
- Experiment:
  - Video Generation task

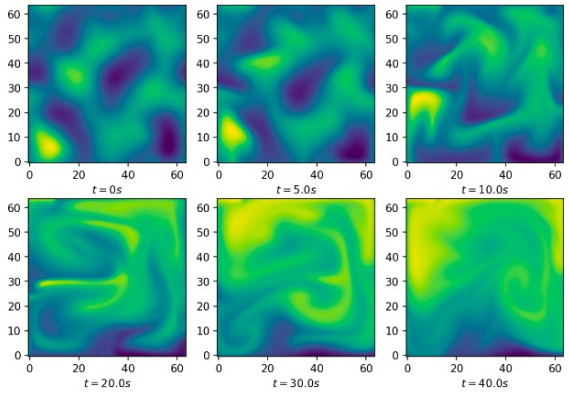
Task	Benchmark	Resolution	Target Frames
Unconditional Video Generation:	UCF101	256x256	16&128 per clip
	Sky-Timelapse	256x256	16&128 per clip

1. Yu S, Sohn K, Kim S, et al. Video Probabilistic Diffusion Models in Projected Latent Space[J]. arXiv preprint arXiv:2302.07685, 2023. (CVPR 2023)

Temporospatial Application

《A Denoising Diffusion Model for Fluid Field Prediction》(2023)

- Main Contribution:
  - Model the fluid field with $p(x|\tau, \mathbf{c})$; The c can be represented as grid data and τ represent the future time and is extended to be a matrix
- Experiment
  - 2D 64x64 random floating-smoke case



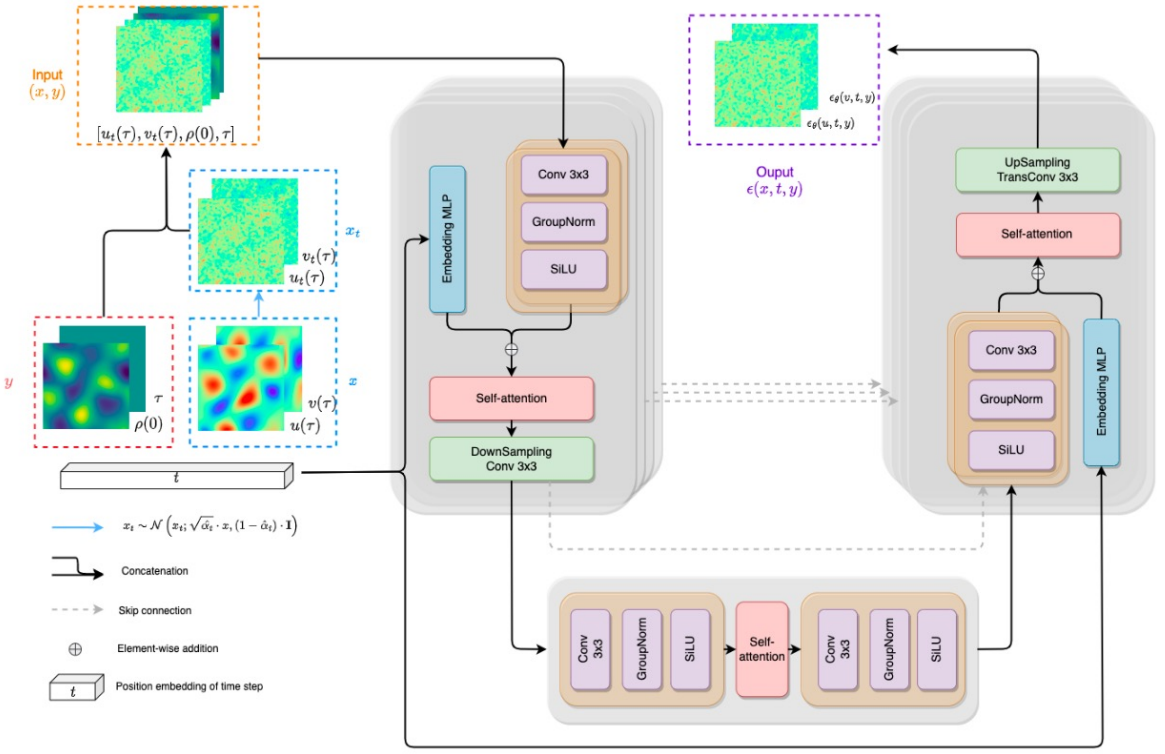


Figure 4. An overview of our proposed model, where the fluid field data samples x are corrupted via diffusion process (5) to be x_t, while initial condition and desired predicting time are ascribed as y. Then x_t and y are concatenated as the input of the network, together with the diffusion time step embedding t through (9). The network is trained to predict the noise ϵ added on x given t, y and further used in sampling process, which is not shown here

1. Yang G, Sommer S. A Denoising Diffusion Model for Fluid Field Prediction[J]. arXiv e-prints, 2023: arXiv: 2301.11661.)

Temporospatial Application

《A Denoising Diffusion Model for Fluid Field Prediction》(2023)

- Experiment
  - 2D 64x64 random floating-smoke case

MODEL	MAE	RMSE
CGAN	0.4030	0.5749
PINN	0.1324	0.1767
U-NET	0.3894	0.5603
FLUIDDIFF	0.1975	0.3137

- Three drawbacks of diffusion models on Prediction
  - low sampling speed
  - absence of physics constraint
  - "spatial inaccuracy, since diffusion-based model is originally developed for image-like spatial invariant data, which is contradicted with the accurate fluid predictions."

1. Yang G, Sommer S. A Denoising Diffusion Model for Fluid Field Prediction[J]. arXiv e-prints, 2023: arXiv: 2301.11661.)

Closing

- 对于扩散模型的场景，由于其引入随机性，本身就是“一对多”的任务，即每次生成不一样的结果就是合理的、正常的特点。对于无条件生成，生成结果是在所有目标中随机sample；而对于有条件生成，其在一定“子集”中生成，这个子集中所有样本具有共性，实现一种泛在的可控生成
- 具体到预测任务(超分辨率)，提高条件的约束性（降低目标方差）

1. Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-Based Generative Modeling through Stochastic Differential Equations[C]//International Conference on Learning Representations.
2. Anderson B D O. Reverse-time diffusion equation models[J]. Stochastic Processes and their Applications, 1982, 12(3): 313-326.
3. 生成扩散模型漫谈（五）：一般框架之SDE篇 - 科学空间|Scientific Spaces (kexue.fm)

Thanks