

MIT Class 6.S184 - Lecture 1 & Lecture 2

MIT Class 6.S184: *Generative AI With Stochastic Differential Equations*, 2025

- An Introduction to Flow Matching and Diffusion Models

1. Introduction

- *Creating noise from data is easy; creating data from noise is generative modeling.*

Generative Modeling As Sampling 生成即采样

- 将生成的对象（图像、视频、蛋白质等）识别为向量 $z \in \mathbb{R}^d$
 - 图像表示为 $z \in \mathbb{R}^{H \times W \times 3}$, 视频表示为 $z \in \mathbb{R}^{T \times H \times W \times 3}$, 分子结构表示为 $z \in \mathbb{R}^{N \times 3}$
 - 文本数据通过自回归语言模型建模为离散对象；虽然也已有相关技术将flow、diffusion用于离散数据
- 生成一个对象 z 即为从未知的数据分布 $z \sim p_{\text{data}}$ 中采样，训练数据集涵盖真实分布的有限数量样本 $z_1, \dots, z_N \sim p_{\text{data}}$
- 根据条件变量 y （如文本提示词）进行**条件生成**，即从条件数据分布 $z \sim p_{\text{data}}(\cdot | y)$ 中采样
 - 训练数据集涵盖样本对 $(z_1, y_1), \dots, (z_N, y_N)$
 - 无条件生成很容易推广到条件生成

From Noise to Data 从噪声到数据

- 生成模型的目标是学习一种变换，将简单初始分布 p_{init} （通常取高斯噪声 $p_{\text{init}} = \mathcal{N}(0, I_d)$ ）迭代转换为复杂的目标数据分布 p_{data}
- 模拟常微分方程 (ODE) 或随机微分方程 (SDE) 来实现这种从噪声到数据的演化
- 流匹配和去噪扩散模型是目前最广泛使用的算法，能够使用深度神经网络大规模构建、训练和模拟此类 ODE/SDE

Summary 2 (Generation as Sampling)

We summarize the findings of this section:

1. In this class, we consider the task of generating objects that are represented as vectors $z \in \mathbb{R}^d$ such as images, videos, or molecular structures.
2. Generation is the task of generating samples from a probability distribution p_{data} having access to a dataset of samples $z_1, \dots, z_N \sim p_{\text{data}}$ during training.
3. Conditional generation assumes that we condition the distribution on a label y and we want to sample from $p_{\text{data}}(\cdot | y)$ having access to data set of pairs $(z_1, y), \dots, (z_N, y)$ during training.
4. Our goal is to train a generative model to transform samples from a simple distribution p_{init} (e.g. a Gaussian) into samples from p_{data} .

2. Flow and Diffusion Models

2.1 Flow Models

- 常微分方程 (ODE) 的解由轨迹定义

$$X : [0, 1] \rightarrow \mathbb{R}^d, \quad t \mapsto X_t$$

- ODE由向量场 (vector field) u 定义

$$u : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d, \quad (x, t) \mapsto u_t(x)$$

- 定义ODE的解为轨迹 X

$$\begin{aligned} X_0 &= x_0 && \blacktriangleright \text{initial conditions} \\ \frac{d}{dt} X_t &= u_t(X_t) && \blacktriangleright \text{ODE} \end{aligned}$$

- 向量场 u_t (vector fields) (连续可微且导数有界 \subset Lipschitz利普希茨连续) 定义的ODE的唯一解是flow流 ψ_t (平滑可逆的微分同胚)

$$\begin{aligned} \psi : \mathbb{R}^d \times [0, 1] &\rightarrow \mathbb{R}^d, \quad (x_0, t) \mapsto \psi_t(x_0) \\ \frac{d}{dt} \psi_t(x_0) &= u_t(\psi_t(x_0)) && \blacktriangleright \text{flow ODE} \\ \psi_0(x_0) &= x_0 && \blacktriangleright \text{flow initial conditions} \end{aligned}$$

- 确定初始状态 $X_0 = x_0$ 可以得到ODE的轨迹 $X_t = \psi_t(X_0)$

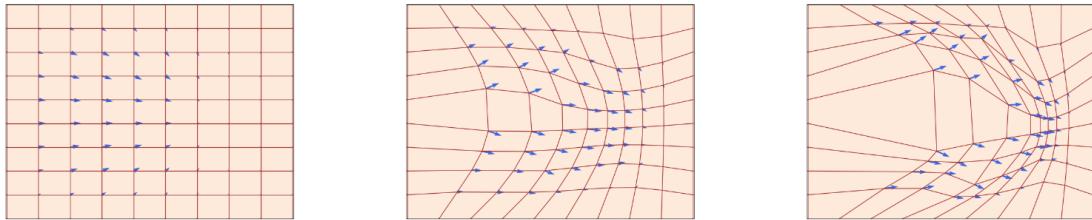


Figure 1: A flow $\psi_t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ (red square grid) is defined by a velocity field $u_t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ (visualized with blue arrows) that prescribes its instantaneous movements at all locations (here, $d = 2$). We show three different times t . As one can see, a flow is a diffeomorphism that "warps" space. Figure from [15].

- 例如

$$\begin{aligned} u_t(x) &= -\theta x \quad \text{for } \theta > 0 \\ \psi_t(x_0) &= \exp(-\theta t)x_0 \\ \psi_0(x_0) &= x_0 \\ \frac{d}{dt} \psi_t(x_0) &= \frac{d}{dt} (\exp(-\theta t)x_0) = -\theta \exp(-\theta t)x_0 = -\theta \psi_t(x_0) = u_t(\psi_t(x_0)) \end{aligned}$$

- 非线性 u_t 无法显式计算 ψ_t , 需要用数值方法模拟ODE, 如使用Euler欧拉法

$$X_0 = x_0, X_{t+h} \approx X_t + h u_t(X_t) \quad (t = 0, h, 2h, 3h, \dots, 1-h), \quad h = \frac{1}{n}, \quad n \in \mathbb{N}$$

- 通过ODE构建生成式flow模型，将简单分布 p_{init} 转换为复杂分布 p_{data} ，其中向量场 $u_t^\theta : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$ 是参数为 θ 的神经网络（参数化的是向量场而非流本身）

$$\begin{aligned} X_0 &\sim p_{\text{init}} && \blacktriangleright \text{random initialization} \\ \frac{d}{dt} X_t &= u_t^\theta(X_t) && \blacktriangleright \text{ODE} \end{aligned}$$

- 使得轨迹的终点 X_1 属于目标数据的分布，其中 ψ_t^θ 由 u_t^θ 得到

$$X_1 \sim p_{\text{data}} \iff \psi_1^\theta(X_0) \sim p_{\text{data}} \iff p_{\text{data}} = (\psi_1^\theta)_\# p_{\text{init}}$$

- 通过欧拉法在流模型中采样

Algorithm Sampling from a Flow Model with Euler method

Require: Neural network vector field u_t^θ , number of steps n

```

Set  $t = 0$ 
Set step size  $h = \frac{1}{n}$ 
Draw a sample  $X_0 \sim p_{\text{init}}$ 
for  $i = 1, \dots, n$  do
     $X_{t+h} = X_t + h u_t^\theta(X_t)$ 
    Update  $t \leftarrow t + h$ 
end for
return  $X_1$ 

```

2.2 Diffusion Models

- 随机微分方程 (SDE)** 将 ODE 的确定性轨迹扩展为随机轨迹（随机过程 $(X_t)_{0 \leq t \leq 1}$ ）。即使初始点相同，由于动力学中引入了随机性，每次模拟的结果可能不同
- 布朗运动 (Brownian Motion, W_t) / 维纳过程 (Wiener Process)** 可视为连续随机游走。
 - 性质:** $W_0 = 0$ ；轨迹连续；**独立增量** $W_{t_1} - W_{t_0}, \dots, W_{t_n} - W_{t_{n-1}}$ ；**正态增量** $W_t - W_s \sim \mathcal{N}(0, (t-s)I_d)$
 - 数值模拟:** 使用步长 h 进行近似更新：

$$W_{t+h} = W_t + \sqrt{h}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I_d), \quad t = 0, h, 2h, \dots, 1-h$$
- 对于ODE，轨迹 X_t 是光滑的，其导数存在，可以通过泰勒展开将导数定义转化为微小增量形式，其中 $R_t(h)$ 代表高阶无穷小项（通常为 $O(h)$ ），使得整体误差为 $O(h^2)$ ：

$$\begin{aligned} \frac{d}{dt} X_t &= u_t(X_t) && \blacktriangleright \text{standard derivative definition} \\ \iff \lim_{h \rightarrow 0} \frac{X_{t+h} - X_t}{h} &= u_t(X_t) \\ \iff X_{t+h} &= X_t + h u_t(X_t) + h R_t(h) && \blacktriangleright \text{discrete update with error } \lim_{h \rightarrow 0} R_t(h) = 0 \end{aligned}$$

- SDE** 在 ODE 的确定性动力学基础上引入布朗运动 W_t 时，其增量 $\Delta W_t = W_{t+h} - W_t$ 服从 $\mathcal{N}(0, hI_d)$ ，即量级为 \sqrt{h} 。若尝试计算导数 $\frac{dX}{dt}$ ，随机项的比值表现为：

$$\frac{\Delta W_t}{h} \sim \frac{\sqrt{h}}{h} = \frac{1}{\sqrt{h}} \xrightarrow{h \rightarrow 0} \infty$$

- 由于 $\frac{1}{\sqrt{h}}$ 发散, **SDE 的轨迹处处连续但处处不可微 (Nowhere Differentiable)**。因此, SDE 不能写成 $\frac{dX}{dt} = \dots$ 的形式, 必须使用无穷小增量或积分方程来定义。
- SDE 的轨迹 $(X_t)_{0 \leq t \leq 1}$ 可以被视为在每个时间步 h 上, 由确定性漂移 (Drift, $O(h)$) 和随机扩散 (Diffusion, $O(\sqrt{h})$) 共同驱动的过程:

$$X_{t+h} = X_t + \underbrace{h u_t(X_t)}_{\text{drift (deterministic)}} + \underbrace{\sigma_t(W_{t+h} - W_t)}_{\text{diffusion (stochastic)}} + \underbrace{h R_t(h)}_{\text{approximation error}}$$

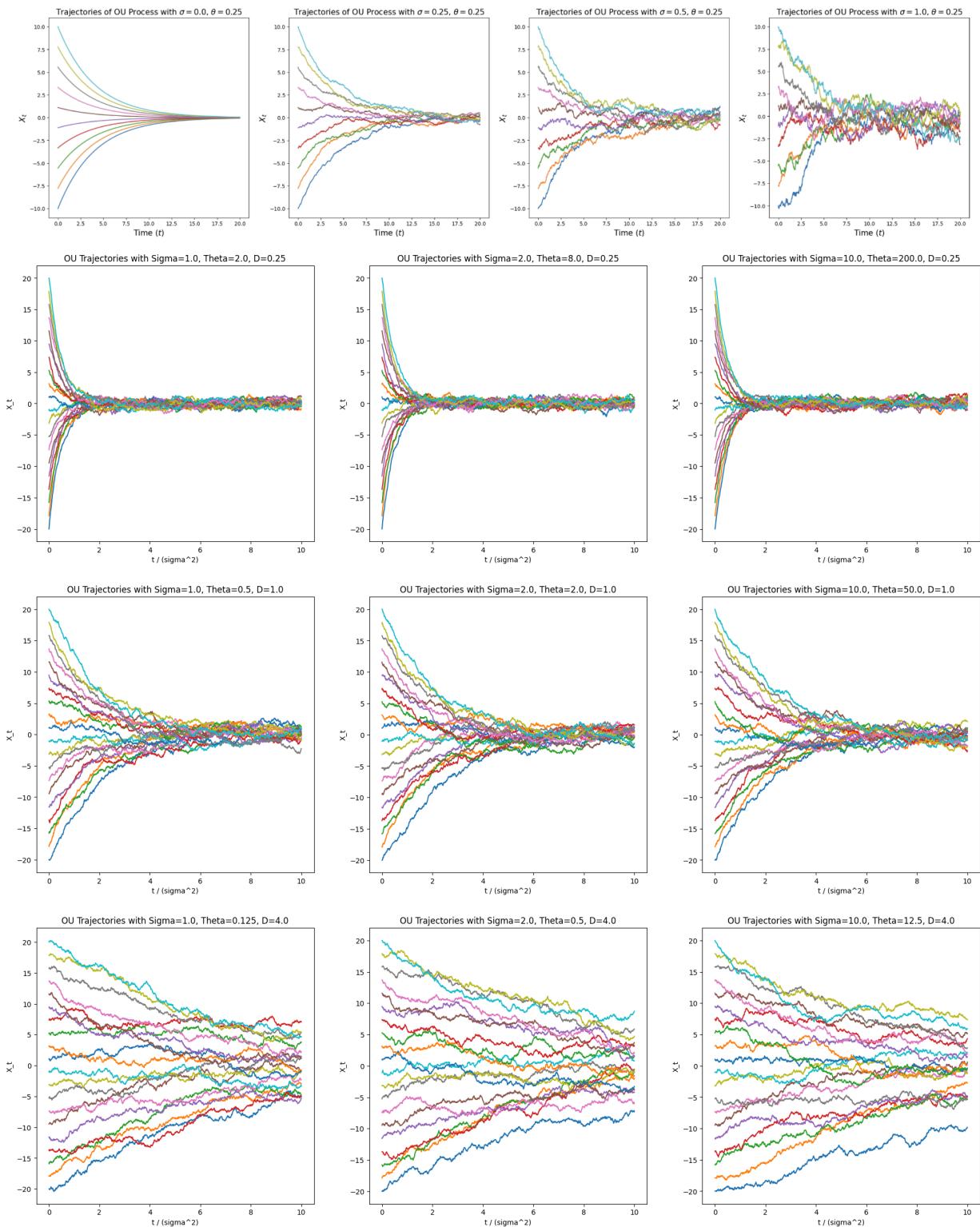
- 注: 在短时间内, 随机项的量级 $O(\sqrt{h})$ 远大于确定性项的量级 $O(h)$ (因为 $\lim_{h \rightarrow 0} \frac{\sqrt{h}}{h} = \lim_{h \rightarrow 0} \frac{1}{\sqrt{h}} = \infty$, 因此微观轨迹主要由噪声主导)。
- 取极限 $h \rightarrow 0$, 丢弃高阶误差项, 定义SDE:

$$\begin{aligned} X_0 &= x_0 && \blacktriangleright \text{ initial condition} \\ \underbrace{dX_t}_{\text{infinitesimal change}} &= \underbrace{u_t(X_t)dt}_{\text{Drift Term}} + \underbrace{\sigma_t dW_t}_{\text{Diffusion Term}} && \blacktriangleright \text{ SDE} \end{aligned}$$

- 这等价于积分形式:

$$X_t = X_0 + \int_0^t u_s(X_s)ds + \int_0^t \sigma_s dW_s$$

- SDE中 X_t 无法完全由 $X_0 \sim p_{\text{init}}$ 确定, 不存在确定性的流映射 (Flow Map) ψ_t 。当 $\sigma_t = 0$ 时, 随机项消失, **SDE 平滑退化为 ODE (流模型)**
- 若 u 连续可微且导数有界 (Lipschitz 连续), 且 σ_t 连续, 则 SDE 存在唯一的随机过程解 $(X_t)_{0 \leq t \leq 1}$
- 例如Ornstein-Uhlenbeck (OU) 过程
 - 设定线性漂移 $u_t(x) = -\theta x$ ($\theta > 0$) 和恒定扩散 $\sigma_t = \sigma \geq 0$: $dX_t = -\theta X_t dt + \sigma dW_t$
 - 向量场 $-\theta x$ 将过程推向中心 0, 而 σ 持续添加噪声。当 $t \rightarrow \infty$ 时, 该过程收敛于高斯分布 $\mathcal{N}(0, \frac{\sigma^2}{2\theta} I_d)$



- ODE 欧拉法在 SDE 上的扩展：Euler-Maruyama 欧拉-丸山法进行数值模拟

$$X_0 = x_0, X_{t+h} = X_t + h u_t(X_t) + \sigma_t \sqrt{h} \epsilon_t, \epsilon_t \sim \mathcal{N}(0, I_d), h = \frac{1}{n}, n \in \mathbb{N}$$

- 通过SDE构建生成式diffusion模型，其中向量场 $u_t^\theta : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$ 是参数为 θ 的神经网络，扩散系数 σ_t 通常是预设固定的

$$\begin{array}{ll} X_0 \sim p_{\text{init}} & \blacktriangleright \text{random initialization} \\ dX_t = u_t^\theta(X_t)dt + \sigma_t dW_t & \blacktriangleright \text{SDE} \end{array}$$

- 通过欧拉-丸山法在扩散模型中采样

Algorithm Sampling from a Diffusion Model (Euler-Maruyama method)

Require: Neural network u_t^θ , number of steps n , diffusion coefficient σ_t

```

Set  $t = 0$ 
Set step size  $h = \frac{1}{n}$ 
Draw a sample  $X_0 \sim p_{\text{init}}$ 
for  $i = 1, \dots, n$  do
    Draw a sample  $\epsilon_i \sim \mathcal{N}(0, I_d)$ 
     $X_{t+h} = X_t + hu_t^\theta(X_t) + \sigma_t \sqrt{h} \epsilon_i$ 
    Update  $t \leftarrow t + h$ 
end for
return  $X_1$ 

```

Summary 7 (SDE generative model)

Throughout this document, a **diffusion model** consists of a neural network u_t^θ with parameters θ that parameterize a vector field and a fixed diffusion coefficient σ_t :

Neural network: $u^\theta : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$, $(x, t) \mapsto u_t^\theta(x)$ with parameters θ

Fixed: $\sigma_t : [0, 1] \rightarrow [0, \infty)$, $t \mapsto \sigma_t$

To obtain samples from our SDE model (i.e. generate objects), the procedure is as follows:

Initialization: $X_0 \sim p_{\text{init}}$	\blacktriangleright Initialize with simple distribution, e.g. a Gaussian
Simulation: $dX_t = u_t^\theta(X_t)dt + \sigma_t dW_t$	\blacktriangleright Simulate SDE from 0 to 1
Goal: $X_1 \sim p_{\text{data}}$	\blacktriangleright Goal is to make X_1 have distribution p_{data}

A diffusion model with $\sigma_t = 0$ is a **flow model**.

3. Constructing the Training Target

训练流模型或扩散模型的核心在于找到一个训练目标 (**Training Target**), 即一个理想的向量场 u_t^{target} , 它能够引导初始分布 p_{init} 精确地演变为数据分布 p_{data} ,

- 模型定义

$$\begin{array}{ll} X_0 \sim p_{\text{init}}, \quad dX_t = u_t^\theta(X_t)dt & (\text{Flow model}) \\ X_0 \sim p_{\text{init}}, \quad dX_t = u_t^\theta(X_t)dt + \sigma_t dW_t & (\text{Diffusion model}) \end{array}$$

- 通用损失函数

$$\mathcal{L}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,1], x \sim p_t} \|u_t^\theta(x) - \underbrace{u_t^{\text{target}}(x)}_{\text{training target}}\|^2$$

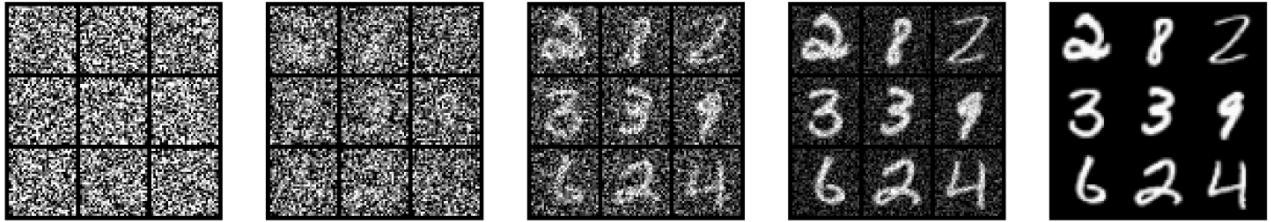
- 条件 (Conditional): 微观视角, 针对单个数据点 z 的动力学。
- 边缘 (Marginal): 宏观视角, 针对整体分布 p_{data} 的动力学。



3.1 Conditional and Marginal Probability Path 条件/边缘概率路径

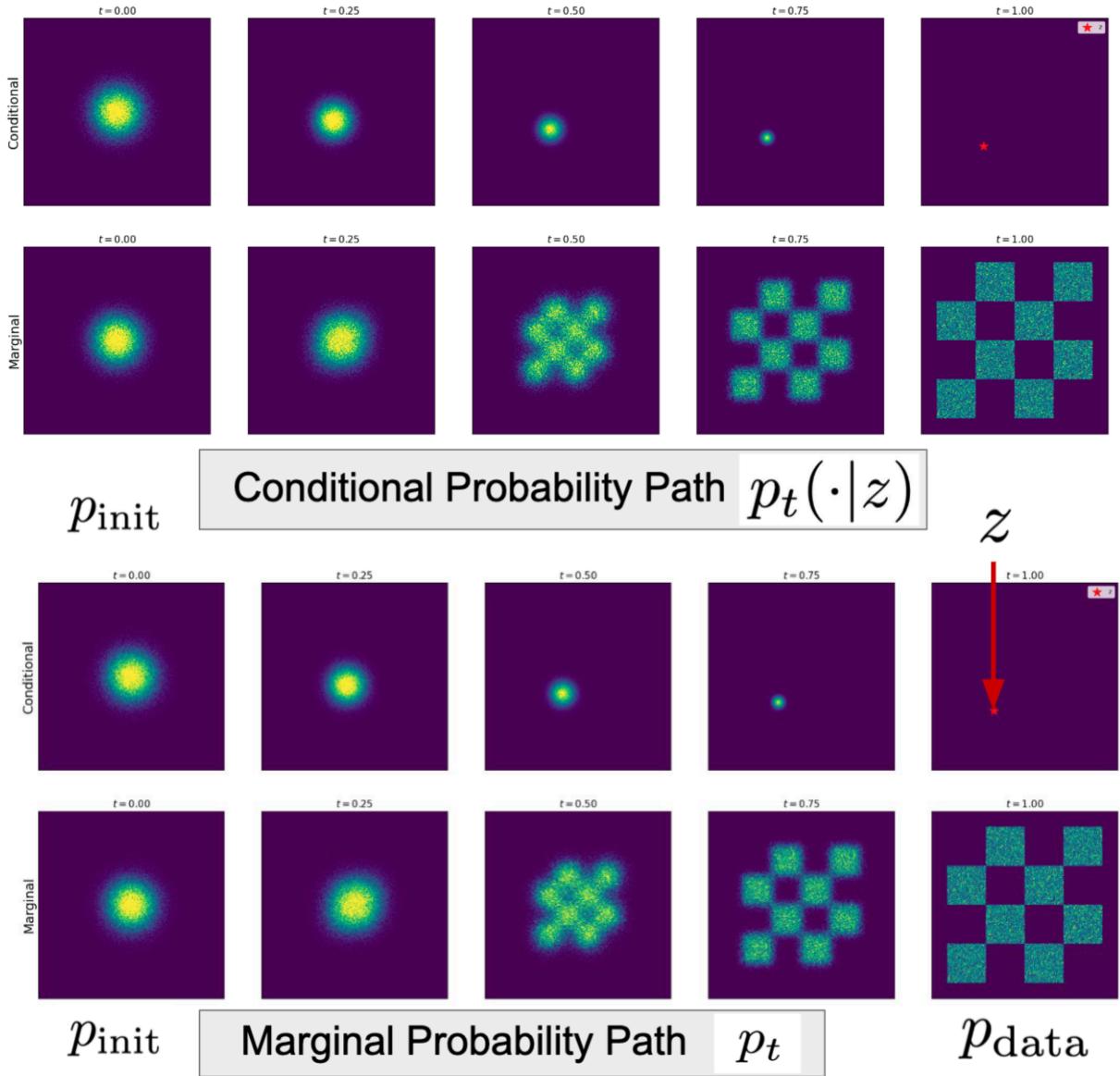
构造训练目标 u_t^{target} 的第一步是指定概率路径

- 概率路径描述了分布空间中从噪声到数据的平滑插值轨迹



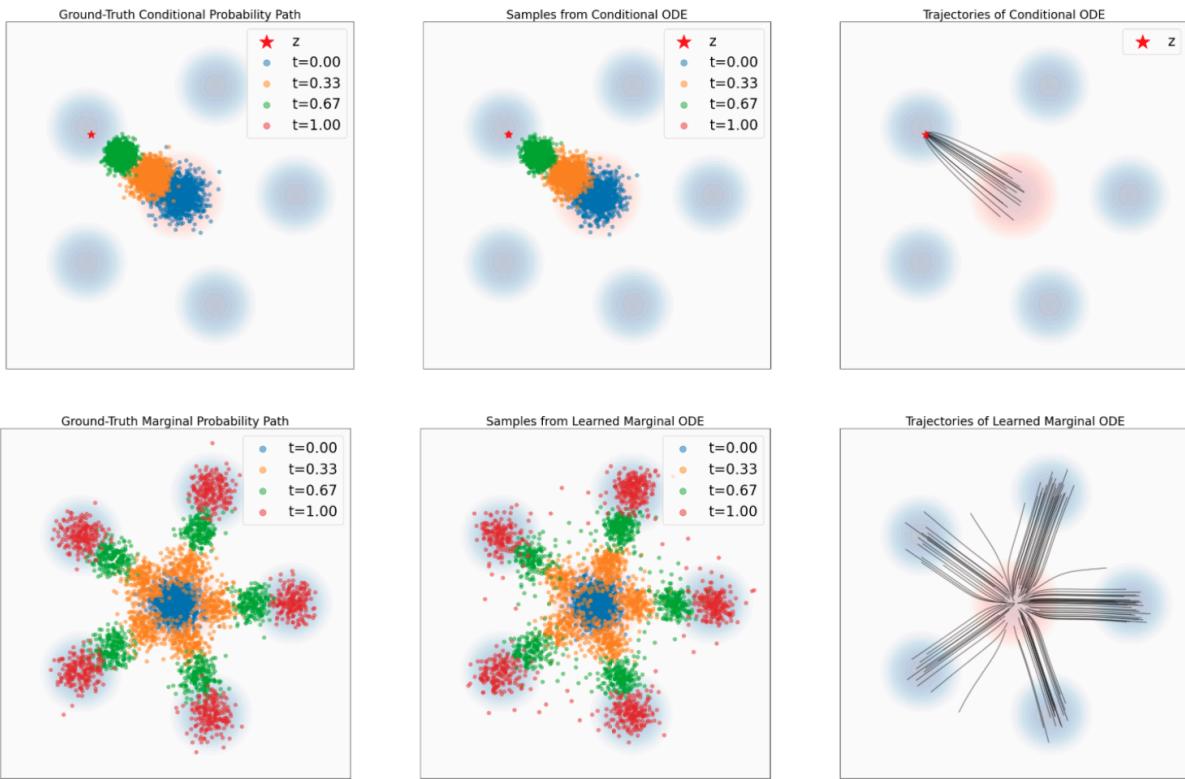
- 条件概率路径 $p_t(x | z)$:
 - 将初始噪声平滑转换为单个数据点 z 的过程
 - 边界条件: $p_0(\cdot | z) = p_{\text{init}}$ 且 $p_1(\cdot | z) = \delta_z$ (Dirac 狄拉克 delta 分布)
- 边缘概率路径 $p_t(x)$:
 - 实现整体分布从 $p_0 = p_{\text{init}}$ 到 $p_1 = p_{\text{data}}$ 的变换
 - 通过对所有数据点 $z \sim p_{\text{data}}$ 积分得到 $p_t(x) = \int p_t(x|z)p_{\text{data}}(z)dz$; 但往往难以计算精确值, 需要采样得到
 - 采样过程: 先从 $z \sim p_{\text{data}}$ 抽样得到样本点 z , 再抽样路径 $x \sim p_t(\cdot | z) \implies x \sim p_t$
 -
- 例如 Gaussian 高斯概率路径 $p_t(\cdot | z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$
 - 最常用的路径形式, 具有解析解。
 - 线性噪声调度器 (Noise Schedulers) $\alpha_t = t, \beta_t = 1 - t$ 满足 $\alpha_0 = 0, \beta_0 = 1$ (噪声) 和 $\alpha_1 = 1, \beta_1 = 0$ (数据)。
 - 实际应用中还有其他选择如余弦调度: $\alpha_t = \cos(\frac{\pi t}{2})$, 多项式调度: $\alpha_t = t^p$
 - 边界条件: $p_0(\cdot | z) = \mathcal{N}(\alpha_0 z, \beta_0^2 I_d) = \mathcal{N}(0, I_d), p_1(\cdot | z) = \mathcal{N}(\alpha_1 z, \beta_1^2 I_d) = \mathcal{N}(z, 0) = \delta_z$

- 从高斯边缘概率路径 p_t 采样即为 $z \sim p_{\text{data}}, \epsilon \sim p_{\text{init}} = \mathcal{N}(0, I_d) \Rightarrow x = \alpha_t z + \beta_t \epsilon \sim p_t$



3.2 Conditional and Marginal Vector Fields 条件/边缘向量场

- 为了训练模型，需要找到驱动上述概率路径的向量场 u_t
 - 直接构建一个能将噪声 p_{init} 变换为复杂数据分布 p_{data} 的全局向量场非常困难，且无法解析写出
 - 需要先针对单个数据点 z 构建简单的条件向量场，再通过积分将它们组合成全局的边缘向量场



条件向量场(Conditional Vector Field)

对于任意单一数据点 $z \in \mathbb{R}^d$, 定义条件向量场 $u_t^{\text{target}}(\cdot | z)$, 使其驱动的 ODE 轨迹符合条件概率路径 $p_t(\cdot | z)$, 即粒子从噪声出发, 最终精确汇聚到 z

$$X_0 \sim p_{\text{init}}, \quad \frac{d}{dt} X_t = u_t^{\text{target}}(X_t | z) \quad \Rightarrow \quad X_t \sim p_t(\cdot | z)$$

例如, 扩散模型中常用的高斯路径有显式的解析解 $u_t^{\text{target}}(x | z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t \alpha_t}{\beta_t} \right) z + \frac{\dot{\beta}_t}{\beta_t} x$

- **路径定义:** $p_t(\cdot | z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$
- **采样公式:** $X_t = \psi_t^{\text{target}}(X_0 | z) = \alpha_t z + \beta_t X_0 \sim \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$ (其中 $X_0 \sim p_{\text{init}} = \mathcal{N}(0, I_d)$ 是初始标准噪声)

$$\frac{d}{dt} X_t = \dot{\alpha}_t z + \dot{\beta}_t X_0, \quad X_0 = \frac{X_t - \alpha_t z}{\beta_t}, \quad \dot{\alpha}_t = \partial_t \alpha_t, \quad \dot{\beta}_t = \partial_t \beta_t$$

$$\begin{aligned} u_t^{\text{target}}(x | z) &= \dot{\alpha}_t z + \dot{\beta}_t \left(\frac{x - \alpha_t z}{\beta_t} \right) \\ &= \left(\dot{\alpha}_t - \frac{\dot{\beta}_t \alpha_t}{\beta_t} \right) z + \frac{\dot{\beta}_t}{\beta_t} x \end{aligned}$$

| 结论: 高斯路径的条件向量场是关于位置 x 的线性函数, 极易计算

边缘向量场 (Marginal Vector Field)

边缘向量场 $u_t^{\text{target}}(x)$ 不再关注某个特定的点 z , 而是指挥整个“概率云”从起始噪声平滑地变形为完整的真实数据分布。

边缘化技巧 (Marginalization Trick)

边缘向量场本质上是所有条件向量场的后验加权平均。如果条件向量场 $u_t^{\text{target}}(x|z)$ 能生成条件路径 $p_t(x|z)$, 那么对应的边缘向量场为

$$u_t^{\text{target}}(x) = \int u_t^{\text{target}}(x|z) \underbrace{\frac{p_t(x|z)p_{\text{data}}(z)}{p_t(x)}}_{p_t(z|x) \text{ 后验概率}} dz$$

$$X_0 \sim p_{\text{init}}, \quad \frac{d}{dt} X_t = u_t^{\text{target}}(X_t) \quad \Rightarrow \quad X_t \sim p_t$$

- 在位置 x 的全局速度, 等于所有可能流经此点的路径速度的平均值 (权重是该路径属于真实数据的概率)。
- 由于涉及对未知数据分布 $p_{\text{data}}(z)$ 和 $p_t(x)$ 的积分, 这个 $u_t^{\text{target}}(x)$ 通常是不可计算的 (Intractable)
- 为了证明上述定义的“边缘化技巧”是正确的 (即它确实能生成边缘概率 $p_t(x)$), 我们需要利用物理学中的连续性方程。

连续性方程 Continuity Equation

一个随时间变化的概率密度 $p_t(x)$ 由向量场 u_t 驱动, 当且仅当满足偏微分方程 PDE:

$$\partial_t p_t(x) = -\text{div}(p_t u_t)(x)$$

- 物理意义: 概率质量守恒。某处的密度变化率 = -(净流出量) (负散度)。
- 散度算子定义

$$\text{div}(v_t)(x) = \nabla_x \cdot v_t = \sum_{i=1}^d \frac{\partial}{\partial x_i} v_t(x)$$

验证 $u_t^{\text{target}}(x)$ 是否满足 $\partial_t p_t = -\text{div}(p_t u_t^{\text{target}})$ 。

1. 展开左边 (时间导数) :

$$\partial_t p_t(x) = \partial_t \int p_t(x|z)p_{\text{data}}(z)dz = \int \partial_t \mathbf{p}_t(\mathbf{x}|z)p_{\text{data}}(z)dz$$

2. 应用条件连续性方程 (将条件导数替换为散度形式) :

$$= \int \left[-\text{div} \left(p_t(x|z)u_t^{\text{target}}(x|z) \right) \right] p_{\text{data}}(z)dz$$

3. 交换运算次序 (积分与散度线性交换) :

$$= -\operatorname{div} \left(\int u_t^{\text{target}}(x|z) p_t(x|z) p_{\text{data}}(z) dz \right)$$

4. 混形式 (Multiply and Divide):

在积分内乘除 $p_t(x)$, 构造出边缘向量场的定义:

$$= -\operatorname{div} \left(p_t(x) \underbrace{\int u_t^{\text{target}}(x|z) \frac{p_t(x|z) p_{\text{data}}(z)}{p_t(x)} dz}_{u_t^{\text{target}}(x) \text{ (Def)}} \right)$$

5. 得证:

$$= -\operatorname{div}(p_t(x) u_t^{\text{target}}(x))$$

这证明了定义的边缘向量场确实驱动了边缘概率分布的演化。

实际上可以使用连续性方程推导的结果定义上述边缘化技巧得到的边缘向量场。

- 要证明边缘向量场 $u_t^{\text{target}}(x)$ 能够生成边缘概率路径 $p_t(x)$, 必须验证它们满足连续性方程:

$$\frac{\partial}{\partial t} p_t(x) = -\operatorname{div}(p_t u_t^{\text{target}})(x) = \frac{\partial}{\partial t} \int p_t(x|z) p_{\text{data}}(z) dz = \int \frac{\partial}{\partial t} p_t(x|z) p_{\text{data}}(z) dz$$

- 条件向量场 $u_t^{\text{target}}(x|z)$ 和对应的条件概率路径 $p_t(x|z)$ 必须满足连续性方程:

$$\frac{\partial}{\partial t} p_t(x|z) = -\operatorname{div}(p_t(x|z) u_t^{\text{target}}(x|z))$$

- 因此

$$\frac{\partial}{\partial t} p_t(x) = \int -\operatorname{div}(p_t(x|z) u_t^{\text{target}}(x|z)) p_{\text{data}}(z) dz = -\operatorname{div} \left(p_t(x) \int u_t^{\text{target}}(x|z) \frac{p_t(x|z) p_{\text{data}}(z)}{p_t(x)} dz \right)$$

- 与连续性方程对比形式, 可以直接定义括号内的积分项为边缘向量场

$$u_t^{\text{target}}(x) = \int u_t^{\text{target}}(x|z) \frac{p_t(x|z) p_{\text{data}}(z)}{p_t(x)} dz$$

Flow Matching Theorem 重参数化引理:

最小化神经网络 v_θ 与“不可计算的”边缘向量场之间的误差, 等价于最小化它与“可计算的”条件向量场之间的误差 (差一个常数)。

$$\underbrace{\mathbb{E}_{t,x \sim p_t} \|v_\theta(x) - u_t^{\text{target}}(x)\|^2}_{\text{不可计算的理想Loss}} = \underbrace{\mathbb{E}_{t,z,x \sim p_t(\cdot|z)} \|v_\theta(x) - \mathbf{u}_t^{\text{target}}(\mathbf{x}|z)\|^2}_{\text{可计算的实际Loss}} + C(\theta \text{无关常数})$$

- 左边: 我们真正想优化的目标 (无法直接算)。

- 右边：我们在代码里实际写的 Loss (可以算！因为 $u_t^{\text{target}}(x|z)$ 有解析解)。

证明：

$$\begin{aligned}
 & \mathbb{E}_{t,x \sim p_t} \|v_\theta(x) - u_t^{\text{target}}(x)\|^2 \\
 &= \mathbb{E}_{t,x} \left[\|v_\theta(x)\|^2 - 2v_\theta(x)^\top u_t^{\text{target}}(x) + \|u_t^{\text{target}}(x)\|^2 \right] \\
 &= \mathbb{E}_{t,x} \|v_\theta(x)\|^2 - 2\mathbb{E}_{t,x} [v_\theta(x)^\top u_t^{\text{target}}(x)] + \underbrace{\mathbb{E}_{t,x} \|u_t^{\text{target}}(x)\|^2}_{C_1(\text{与 } \theta \text{ 无关})}
 \end{aligned}$$

关键是展开 $u_t^{\text{target}}(x)$ 的定义：

$$\begin{aligned}
 \mathbb{E}_{t,x} [v_\theta(x)^\top u_t^{\text{target}}(x)] &= \mathbb{E}_{t,x} \left[v_\theta(x)^\top \int u_t^{\text{target}}(x|z) \frac{p_t(x|z)p_{\text{data}}(z)}{p_t(x)} dz \right] \\
 &= \mathbb{E}_t \int \int v_\theta(x)^\top u_t^{\text{target}}(x|z) p_t(x|z) p_{\text{data}}(z) dx dz \\
 &= \mathbb{E}_{t,z,x \sim p_t(\cdot|z)} [v_\theta(x)^\top u_t^{\text{target}}(x|z)]
 \end{aligned}$$

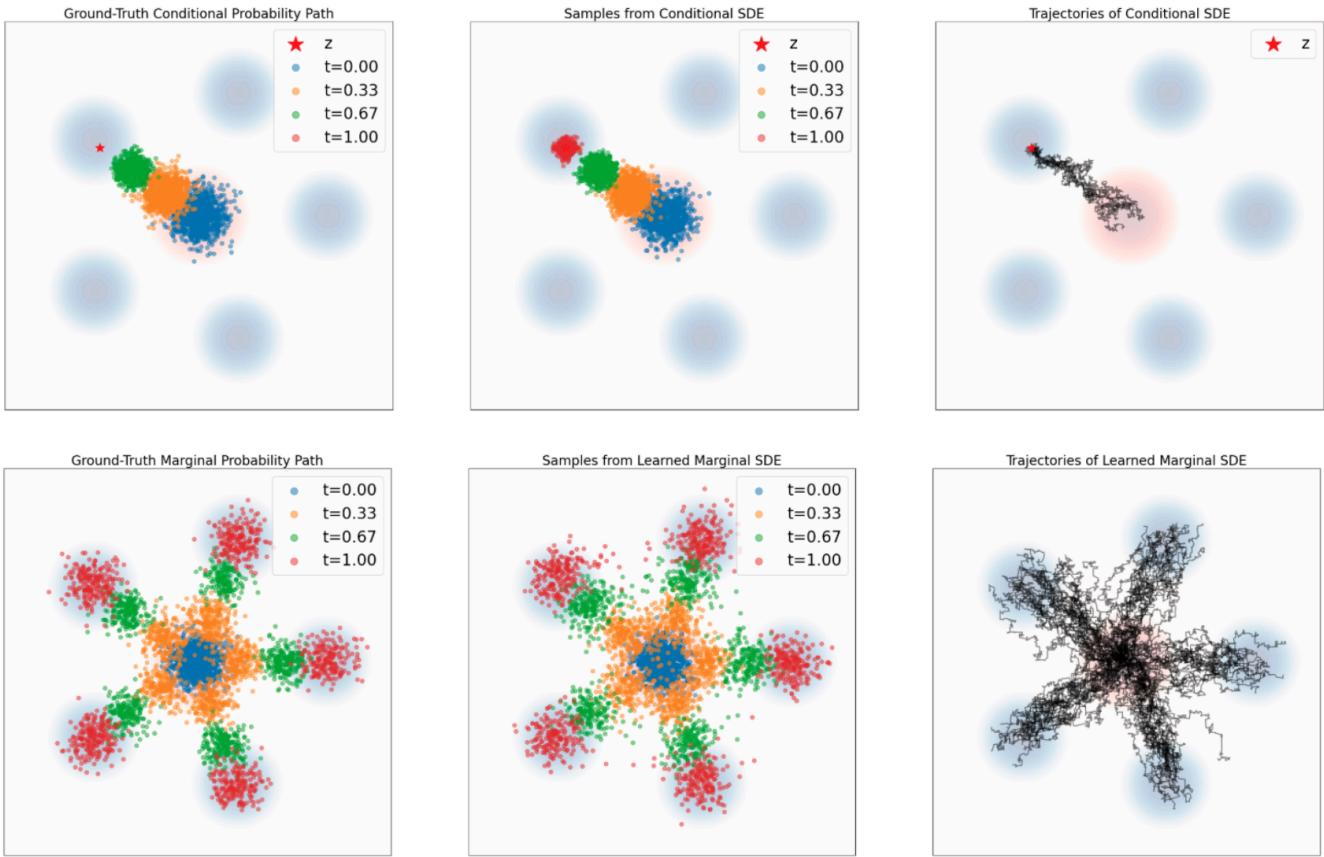
因此：

$$\begin{aligned}
 & \mathbb{E}_{t,x} \|v_\theta(x) - u_t^{\text{target}}(x)\|^2 \\
 &= \mathbb{E}_{t,x} \|v_\theta(x)\|^2 - 2\mathbb{E}_{t,z,x} [v_\theta(x)^\top u_t^{\text{target}}(x|z)] + C_1 \\
 &= \mathbb{E}_{t,z,x} \|v_\theta(x) - u_t^{\text{target}}(x|z)\|^2 + C_2
 \end{aligned}$$

其中 C_2 是与 θ 无关的常数。

3.3 Conditional and Marginal Score Functions 条件/边缘分数函数

为了引入随机性 (SDE)，我们需要**分数函数 (Score Function)**。



- 概率密度对数的梯度，称为**分数函数**，指向概率密度增长最快的方向。

$$s_t(x) = \nabla_x \log p_t(x)$$

- 条件分数函数** $\nabla \log p_t(x|z)$ 通常有解析解（例如高斯分布）

$$\nabla_x \log p_t(x|z) = \nabla_x \log \mathcal{N}(x; \alpha_t z, \beta_t^2 I_d) = -\frac{x - \alpha_t z}{\beta_t^2}$$

- 对于一个 d 维多元高斯分布 $\mathcal{N}(x; \alpha_t z, \beta_t^2 I_d)$ ，其均值 $\mu = \alpha_t z$ ，协方差矩阵 $\Sigma = \beta_t^2 I_d$ ，密度函数定义为：

$$\begin{aligned}
p_t(x|z) &= \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right) \\
\log p_t(x|z) &= \log\left(\frac{1}{\sqrt{(2\pi)^d |\Sigma|}}\right) + \log\left(\exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right)\right) \\
&= \underbrace{-\frac{1}{2}\log((2\pi)^d |\Sigma|)}_{\text{常数项 } C} - \frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu) \\
&= C - \frac{1}{2}(x - \alpha_t z)^\top \left(\frac{1}{\beta_t^2} I_d\right)(x - \alpha_t z) \\
&= C - \frac{1}{2\beta_t^2}(x - \alpha_t z)^\top (x - \alpha_t z) \\
\nabla_x \log p_t(x|z) &= \nabla_x \left[C - \frac{1}{2\beta_t^2}(x - \alpha_t z)^\top (x - \alpha_t z)\right] \\
&= 0 - \frac{1}{2\beta_t^2} \cdot \nabla_x [(x - \alpha_t z)^\top (x - \alpha_t z)] \\
&= -\frac{1}{2\beta_t^2} \cdot 2(x - \alpha_t z) \\
&= -\frac{x - \alpha_t z}{\beta_t^2}
\end{aligned}$$

- 高斯概率路径的条件分数函数 $\nabla_x \log p_t(x | z)$ 是 x 的线性函数
- 与向量场类似，全局的边缘分数函数也是无法直接计算的。但可以利用同样的边缘化技巧，将其表示为条件分数函数的后验加权平均：

$$\underbrace{\nabla \log p_t(x)}_{\text{边缘分数}} = \frac{\nabla p_t(x)}{p_t(x)} = \frac{\nabla \int p_t(x|z)p_{\text{data}}(z)dz}{p_t(x)} = \frac{\int \nabla p_t(x|z)p_{\text{data}}(z)dz}{p_t(x)} = \int \underbrace{\nabla \log p_t(x|z)}_{\text{条件分数}} \frac{p_t(x|z)p_{\text{data}}(z)}{p_t(x)}$$

SDE extension trick SDE 扩展技巧

- 如果已经有了向量场 u_t^{target} 和分数函数 $\nabla \log p_t$ ，可以构造如下 SDE，使其边缘分布 X_t 依然服从 p_t ：

$$X_0 \sim p_{\text{init}}, \quad dX_t = \underbrace{\left[u_t^{\text{target}}(X_t) + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t) \right]}_{\text{修正后的漂移项 (Drift)}} dt + \underbrace{\sigma_t dW_t}_{\text{扩散项 (Diffusion)}} \implies X_t \sim p_t \quad (0 \leq t \leq 1)$$

u_t^{target} : 原本的流场，试图把粒子推向目标。

$\sigma_t dW_t$: 布朗运动，会把粒子向四周随机驱散（导致分布扩散）。

$\frac{\sigma_t^2}{2} \nabla \log p_t$: 这是一个修正项。因为布朗运动会使粒子倾向于从高密度区流向低密度区（扩散效应），我们需要在这个修正项里加一个力，把粒子“拉回”高密度区域（分数函数指向高密度方向），从而抵消随机噪声导致的分布坍塌。

Fokker-Planck Equation 福克-普朗克方程

- 定义Laplacian 拉普拉斯算子：先求梯度（一阶导），再对梯度求散度（二阶导求和）

$$\Delta w_t(x) = \sum_{i=1}^d \frac{\partial^2}{\partial x_i^2} w_t(x) = \operatorname{div}(\nabla w_t)(x)$$

- Fokker-Planck 方程是 SDE 对应的概率密度演化方程。它描述了当微观粒子服从 SDE 运动时，宏观的概率密度 p_t 是如何随时间变化的。
- 当粒子服从SDE

$$X_0 \sim p_{\text{init}}, \quad dX_t = \mu_t(X_t)dt + \sigma_t dW_t$$

- 其概率密度 p_t 的演化满足偏微分方程 PDE：

$$\partial_t p_t(x) = \underbrace{-\operatorname{div}(p_t \mu_t)(x)}_{\text{对流项 (Advection)}} + \underbrace{\frac{\sigma_t^2}{2} \Delta p_t(x)}_{\text{扩散项 (Diffusion)}}$$

- 第一项 $-\operatorname{div}(p_t u_t)$ 是对流项 (**Advection**)，来自 ODE 的确定性移动（即连续性方程）
- 第二项 $\frac{\sigma_t^2}{2} \Delta p_t$ 是扩散项 (**Diffusion**)，由拉普拉斯算子 $\Delta p_t = \operatorname{div}(\nabla p_t)$ 控制，描述热量（概率）如何从高浓度向低浓度自然扩散。类似于热传导方程，由布朗运动 $\sigma_t dW_t$ 引入，会把原本紧凑的分布“推散”
- $\sigma_t = 0$ 时退化为连续性方程 continuity equation，此时只要沿着 $dX_t = u_t dt$ 运动，宏观分布就会完美符合 p_t
- 但直接加上随机噪声 $\sigma_t dW_t$ 时，宏观分布会偏离原本定义的 p_t ，多出一项拉普拉斯项 (**Laplacian term**) $\frac{\sigma_t^2}{2} \Delta p_t$ ，因此必须用**SDE extension trick**在漂移项中加入分数项，来抵消抵消噪声带来的“模糊”效果，以保持概率路径 (Probability Path) 的一致性
- 也就是将 SDE 的漂移项从 u_t^{target} 修正为 $u_t^{\text{target}} + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t)$ 。修正后的对流项会多产生一项 $-\operatorname{div}(p_t \cdot \frac{\sigma_t^2}{2} \nabla \log p_t) = -\frac{\sigma_t^2}{2} \Delta p_t$ ，即与随机噪声产生的扩散项 $\frac{\sigma_t^2}{2} \Delta p_t$ 相互抵消
- 完整证明SDE extension trick，从无噪声的连续性方程出发 $\partial_t p_t = -\operatorname{div}(p_t u_t^{\text{target}})$ ：

$$\begin{aligned} \partial_t p_t(x) &= -\operatorname{div}(p_t u_t^{\text{target}})(x) \\ &= -\operatorname{div}(p_t u_t^{\text{target}})(x) - \frac{\sigma_t^2}{2} \Delta p_t(x) + \frac{\sigma_t^2}{2} \Delta p_t(x) \quad (\text{加减同项}) \\ &= -\operatorname{div}(p_t u_t^{\text{target}})(x) - \operatorname{div}\left(\frac{\sigma_t^2}{2} \nabla p_t\right)(x) + \frac{\sigma_t^2}{2} \Delta p_t(x) \\ &= -\operatorname{div}(p_t u_t^{\text{target}})(x) - \operatorname{div}\left(p_t \left[\frac{\sigma_t^2}{2} \nabla \log p_t\right]\right)(x) + \frac{\sigma_t^2}{2} \Delta p_t(x) \\ &= -\operatorname{div}\left(p_t \left[u_t^{\text{target}} + \frac{\sigma_t^2}{2} \nabla \log p_t\right]\right)(x) + \frac{\sigma_t^2}{2} \Delta p_t(x) \end{aligned}$$

- 对比 Fokker-Planck 方程的标准形式 PDE $\partial_t p_t(x) = \underbrace{-\text{div}(p_t \mu_t)(x)}_{\text{对流项 (Advection)}} + \underbrace{\frac{\sigma_t^2}{2} \Delta p_t(x)}_{\text{扩散项 (Diffusion)}},$ 方

括号内的部分就是 SDE 所需的新漂移项 (Drift) :

$$u_t(x) = u_t^{\text{target}}(x) + \frac{\sigma_t^2}{2} \nabla \log p_t(x)$$

- 对应的 SDE 即为:

$$dX_t = \underbrace{\left[u_t^{\text{target}}(X_t) + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t) \right]}_{\text{修正后的漂移项 (Drift)}} dt + \underbrace{\sigma_t dW_t}_{\text{扩散项 (Diffusion)}}$$

物理现象	微观视角SDE (粒子 X_t)	宏观视角PDE (密度 p_t)
确定性运动	漂移项 (Drift) $u_t dt$	对流项 (Advection) $-\text{div}(pu)$
随机性运动	布朗噪声 $\sigma_t dW_t$	扩散项 (Diffusion) $\frac{\sigma_t^2}{2} \Delta p$

Langevin Dynamics 郎之万动力学

当概率分布不随时间变化 ($p_t = p$ 是静态分布), 且没有基础流场 ($u_t = 0$) 时, SDE 退化为著名的 **Langevin Dynamics 郎之万动力学**:

$$dX_t = \underbrace{\frac{\sigma_t^2}{2} \nabla \log p(X_t) dt}_{\text{Score Function (Drift)}} + \underbrace{\sigma_t dW_t}_{\text{Noise (Diffusion)}}$$

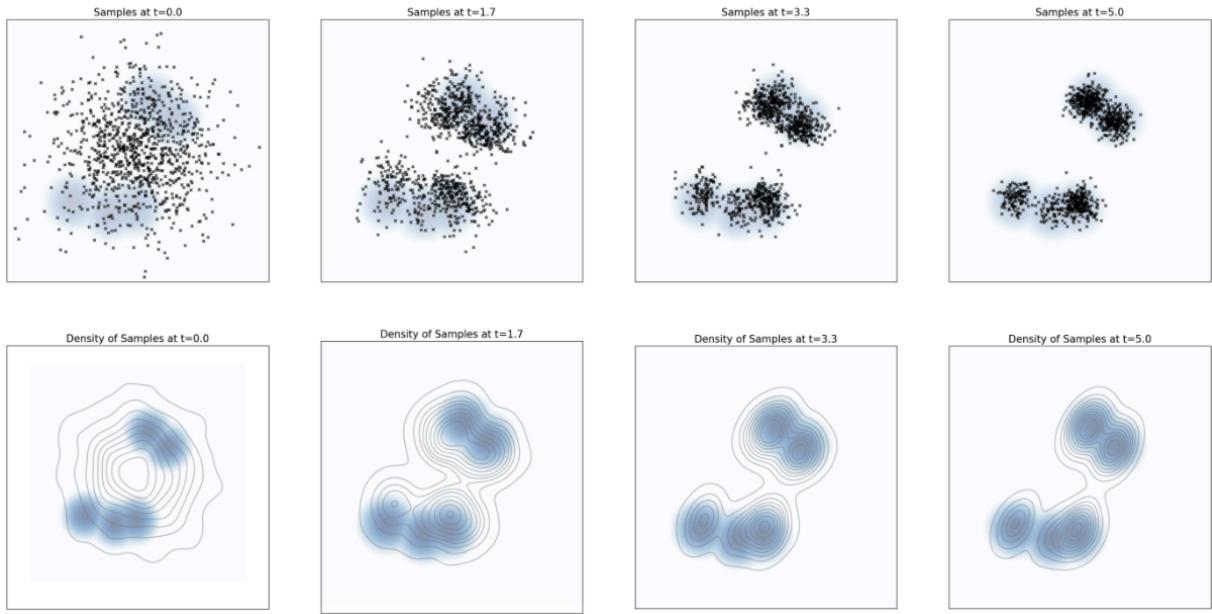
- 代入 Fokker-Planck 方程中:

$$\text{对流项} = -\text{div}(pu_t) = -\text{div}\left(p(x) \cdot \left[\frac{\sigma_t^2}{2} \nabla \log p(x)\right]\right) = -\text{div}\left(\frac{\sigma_t^2}{2} \nabla p(x)\right) = -\frac{\sigma_t^2}{2} \Delta p(x)$$

$$\partial_t p_t(x) = 0 = \underbrace{-\text{div}(pu_t)}_{\text{对流项}} + \underbrace{\frac{\sigma_t^2}{2} \Delta p}_{\text{扩散项}} = \left(-\frac{\sigma_t^2}{2} \Delta p(x)\right) + \left(+\frac{\sigma_t^2}{2} \Delta p(x)\right)$$

- 即 Langevin Dynamics 的 SDE 满足 Fokker-Planck 方程, $p_t = p$ 即是该随机过程的 **平稳分布 (Stationary Distribution)**
 - 如果初始点 $X_0 \sim p$, 那么对于任意 $t \geq 0$, 都有 $X_t \sim p$ 。
 - 即使初始点 X_0 不服从 p (即 $X_0 \sim p' \neq p$), 只要运行这个 SDE 足够长的时间, 不管初始点 X_0 在哪, 分布 p_t 最终都会收敛到 p 。这被广泛用于分子动力学模拟和 MCMC 采样。

样。



Summary

1. 设定 (Setup)

- **边界:** $p_0 = \mathcal{N}(0, I)$ (噪声), $p_1 = \delta_z$ (数据)。
- **路径:** $p_t(\cdot|z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$ 。
- **参数:** α_t, β_t 是时间 t 的平滑函数。

2. 训练目标 (Training Targets)

这是我们需要让神经网络去拟合的“正确答案”：

- **向量场 (Vector Field) / Flow Matching Target:**

$$u_t^{\text{flow}}(x|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) z + \frac{\dot{\beta}_t}{\beta_t} x$$

(注意：是 x 和 z 的线性组合)

- **分数函数 (Score Function) / Diffusion Target:**

$$\nabla \log p_t(x|z) = -\frac{x - \alpha_t z}{\beta_t^2}$$

(注意：这就是高斯分布对数密度的导数，形式非常简单，就是去噪方向)

3. 生成过程 (Inference)

训练好神经网络 $v_\theta \approx u_t$ 和 $s_\theta \approx \nabla \log p_t$ 后，我们可以选择：

- **ODE 采样:** $dX_t = v_\theta(X_t)dt$

- **SDE 采样**: $dX_t = [v_\theta(X_t) + \frac{\sigma_t^2}{2} s_\theta(X_t)]dt + \sigma_t dW_t$

Summary 17 (Derivation of the Training Target)

The flow training target is the marginal vector field u_t^{target} . To construct it, we choose a **conditional probability path** $p_t(x|z)$ that fulfills $p_0(\cdot|z) = p_{\text{init}}$, $p_1(\cdot|z) = \delta_z$. Next, we find a **conditional vector field** $u_t^{\text{flow}}(x|z)$ such that its corresponding flow $\psi_t^{\text{target}}(x|z)$ fulfills

$$X_0 \sim p_{\text{init}} \Rightarrow X_t = \psi_t^{\text{target}}(X_0|z) \sim p_t(\cdot|z),$$

or, equivalently, that u_t^{target} satisfies the continuity equation. Then the **marginal vector field** defined by

$$u_t^{\text{target}}(x) = \int u_t^{\text{target}}(x|z) \frac{p_t(x|z)p_{\text{data}}(z)}{p_t(x)} dz, \quad (32)$$

follows the marginal probability path, i.e.,

$$X_0 \sim p_{\text{init}}, \quad dX_t = u_t^{\text{target}}(X_t)dt \Rightarrow X_t \sim p_t \quad (0 \leq t \leq 1). \quad (33)$$

In particular, $X_1 \sim p_{\text{data}}$ for this ODE, so that u_t^{target} "converts noise into data", as desired.

Extending to SDEs. For a time-dependent diffusion coefficient $\sigma_t \geq 0$, we can extend the above ODE to an SDE with the same marginal probability path:

$$X_0 \sim p_{\text{init}}, \quad dX_t = \left[u_t^{\text{target}}(X_t) + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t) \right] dt + \sigma_t dW_t \quad (34)$$

$$\Rightarrow X_t \sim p_t \quad (0 \leq t \leq 1), \quad (35)$$

where $\nabla \log p_t(x)$ is the **marginal score function**

$$\nabla \log p_t(x) = \int \nabla \log p_t(x|z) \frac{p_t(x|z)p_{\text{data}}(z)}{p_t(x)} dz. \quad (36)$$

In particular, for the trajectories X_t of the above SDE, it holds that $X_1 \sim p_{\text{data}}$, so that the SDE "converts noise into data", as desired. An important example is the **Gaussian probability path**, yielding the formulae:

$$p_t(x|z) = \mathcal{N}(x; \alpha_t z, \beta_t^2 I_d) \quad (37)$$

$$u_t^{\text{flow}}(x|z) = \left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) z + \frac{\dot{\beta}_t}{\beta_t} x \quad (38)$$

$$\nabla \log p_t(x|z) = -\frac{x - \alpha_t z}{\beta_t^2}, \quad (39)$$

for **noise schedulers** $\alpha_t, \beta_t \in \mathbb{R}$: continuously differentiable, monotonic functions such that $\alpha_0 = \beta_1 = 0$, $\alpha_1 = \beta_0 = 1$.

Conditional Prob. Path, Vector Field, and Score

	Notation	Key property	Gaussian example
Conditional Probability Path	$p_t(\cdot z)$	Interpolates p_{init} and a data point z	$\mathcal{N}(\alpha_t z, \beta_t^2 I_d)$
Conditional Vector Field	$u_t^{\text{target}}(x z)$	ODE follows conditional path	$\left(\dot{\alpha}_t - \frac{\dot{\beta}_t}{\beta_t} \alpha_t \right) z + \frac{\dot{\beta}_t}{\beta_t} x$
Conditional Score Function	$\nabla \log p_t(x z)$	Gradient of log-likelihood	$-\frac{x - \alpha_t z}{\beta_t^2}$

Marginal Prob. Path, Vector Field, and Score

	Notation	Key property	Formula
Marginal Probability Path	p_t	Interpolates p_{init} and p_{data}	$\int p_t(x z)p_{\text{data}}(z)dz$
Marginal Vector Field	$u_t^{\text{target}}(x)$	ODE follows marginal path	$\int u_t^{\text{target}}(x z) \frac{p_t(x z)p_{\text{data}}(z)}{p_t(x)} dz$
Marginal Score Function	$\nabla \log p_t(x)$	Can be used to convert ODE target to SDE	$\int \nabla \log p_t(x z) \frac{p_t(x z)p_{\text{data}}(z)}{p_t(x)} dz$