《物理与人工智能》前沿讲座

AI与大规模数值模拟

冯旭（北大理论所）

邮箱：xu.feng@pku.edu.cn

2025年12月15日

大规模数值模拟与蒙特卡洛

- 大规模数值模拟中，蒙特卡洛（Monte Carlo, MC）方法的应用范围非常广
- 研究对象满足以下特点
  - 高维积分 $I = \int f(x_1, \ldots, x_d)\, dx^d.$
  - 多体相互作用
  - 复杂随机系统
  - 稀疏或不规则空间

常规方法：梯形法、Simpson，网格数量~N^d

每维只取10个点，10^{20}就完全不可算

蒙特卡洛：$I \approx \frac{1}{M} \sum_{i=1}^{M} f(x^{(i)}),$

误差几乎与维度无关

大规模数值模拟与蒙特卡洛

- 大规模数值模拟中，蒙特卡洛（Monte Carlo, MC）方法的应用范围非常广
- 研究对象满足以下特点
  - 高维积分
  - 多体相互作用
  - 复杂随机系统
  - 稀疏或不规则空间

N个粒子的量子系统，波函数在3N维空间

多体量子系统的路径积分本质上是高维积分

$$Z = \int \mathcal{D}\phi \, e^{-S[\phi]}.$$

大规模数值模拟与蒙特卡洛

- 大规模数值模拟中，蒙特卡洛（Monte Carlo, MC）方法的应用范围非常广
- 研究对象满足以下特点
  - 高维积分
  - 多体相互作用
  - 复杂随机系统 →
  - 稀疏或不规则空间

很多系统本身就是随机的

- ✓ 粒子shower的级联过程
- ✓ 致密介质中，如恒星内部、中子星表面、超新星爆炸喷发层，光子传播是随机游走
- ✓ 金融中的随机过程
- ✓ 生化反应网络

大规模数值模拟与蒙特卡洛

- 大规模数值模拟中，蒙特卡洛（Monte Carlo, MC）方法的应用范围非常广
- 研究对象满足以下特点
  - 高维积分
  - 多体相互作用
  - 复杂随机系统
  - 稀疏或不规则空间

积分域有奇异点、缺口、高度不规则；或数据来自实验/真实地形结构

传统网格方法需要复杂网格剖分，局部refinement，大量人工处理

蒙卡直接随机采样空间中的点即可

$$I = \mathbb{E}[f(X)].$$

大规模数值模拟与蒙特卡洛

- 大规模数值模拟中，蒙特卡洛（Monte Carlo, MC）方法的应用范围非常广
- 研究对象满足以下特点
  - 高维积分
  - 多体相互作用
  - 复杂随机系统
  - 稀疏或不规则空间

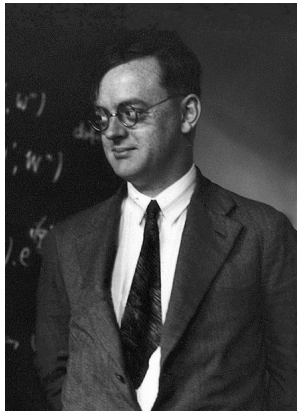
以格点量子场论为例

场与谐振子

- 用谐振子来描述场可追溯到1925-1926

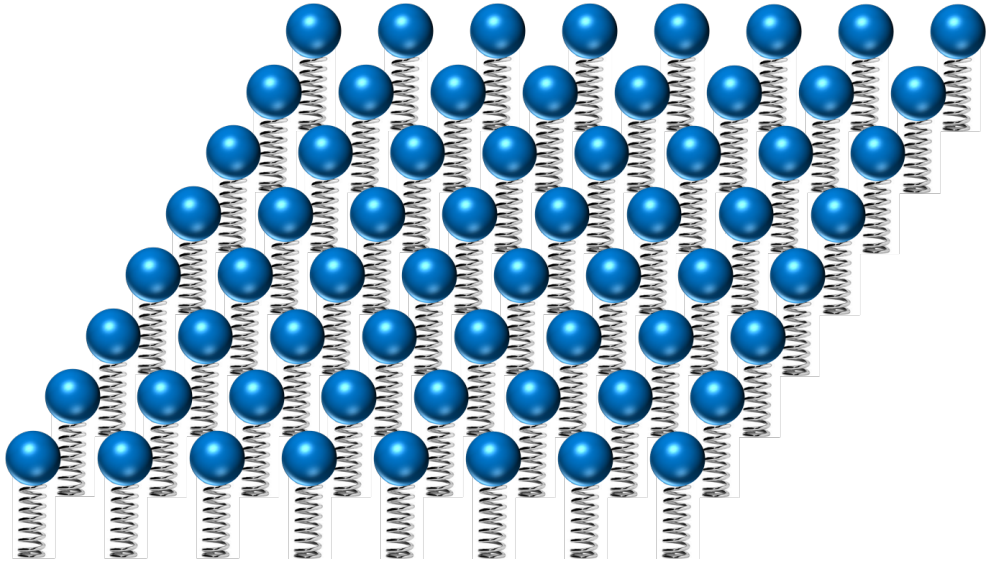

波恩


海森堡

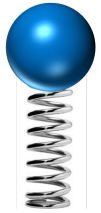

约尔当

用量子谐振子来处理电磁场正则量子化

- 场有无穷多自由度 → 空间每个点都有一个谐振子



- 考虑一个谐振子，只有垂直方向运动



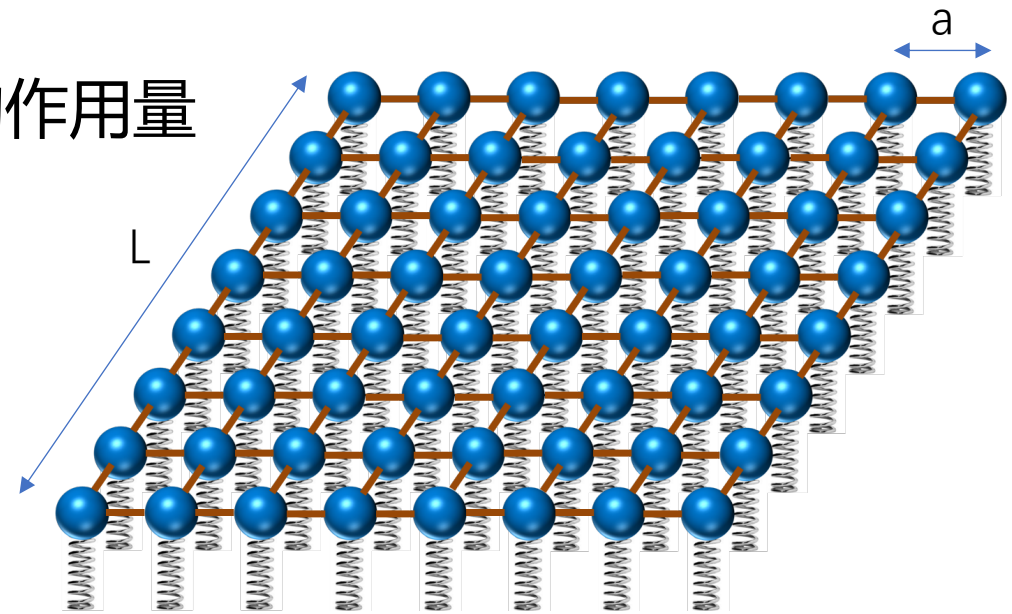
$$L = \frac{1}{2}m\dot{q}^2 - \frac{1}{2}kq^2$$

每个小球的高度 q_i 都可以用一个数来表示 → 标量场

$$L = \frac{1}{2}\sum_i m\dot{q}_i^2 - \frac{1}{2}\sum_i kq_i^2$$

从量子力学到量子场论

- 场的作用量



假设只对尺度远大于格距 a 的现象感兴趣

取连续极限：a → 0

$$i \to \vec{x}$$

$$q_i(t) \to q(t,\vec{x}) \to \phi(t,\vec{x}) \to \phi(x)$$

量子力学等价于 (0+1) 维量子场论

动能项 $\sum_i \frac{1}{2} m \dot{q}_i^2$

势能项 $V(q_1, q_2, \cdots, q_N) = \frac{1}{2} \sum_{ij} k_{ij} (q_i - q_j)^2 + \cdots$

场的路径积分

- 配分函数 $$Z = \int \mathcal{D}\phi\, e^{-S[\phi]}$$

$$\mathcal{D}\phi = \prod_x d\phi_x$$

- 物理量 $$\langle f(\phi)\rangle = \frac{1}{Z}\int \mathcal{D}\phi\, f(\phi) e^{-S[\phi]}$$

- 由于欧氏时空下，$S[\phi]$ 是正定的，积分的大部分区域都被 $e^{-S[\phi]}$ 指数压低了

- 具有Z_2对称性的$\lambda\phi^4$理论的欧氏空间作用量

$$S_{\mathrm{E}}[\phi] = \sum_x \left[-\frac{1}{2}\phi_x(\hat{\partial}_\mu\hat{\partial}^*_\mu)\phi_x + \frac{m_0^2}{2}\phi_x^2 + \frac{\lambda_0}{4!}\phi_x^4\right]$$

- 离散化后，并对场变量进行替换

$\lambda \to \infty$对应Ising模型

$$S_{\mathrm{E}}[\phi] = -\kappa\sum_{x,\mu}\phi_x[\phi_{x+\mu} + \phi_{x-\mu}] + \sum_x\left[\phi_x^2 + \lambda(\phi_x^2 - 1)^2\right]$$

场的路径积分

- 配分函数 $$Z = \int \mathcal{D}\phi\, e^{-S[\phi]}$$

$$\mathcal{D}\phi = \prod_x d\phi_x$$

- 物理量 $$\langle f(\phi)\rangle = \frac{1}{Z}\int \mathcal{D}\phi\, f(\phi) e^{-S[\phi]}$$

- 由于欧氏时空下，$S[\phi]$ 是正定的，积分的大部分区域都被 $e^{-S[\phi]}$ 指数压低了

- 严格计算，对于每个ϕ，有±1两种取值，10^4的格点，需要求和的项数为$2^{10^4} \approx 10^{3\times 10^3}$，这个计算量不是目前地球人的经典计算机能承受的

- 将场ϕ定义在 32^4-64^4 的格子上，则对应百万-千万维的高维积分

重点抽样：提取积分中最重要的部分是完成路径积分的核心

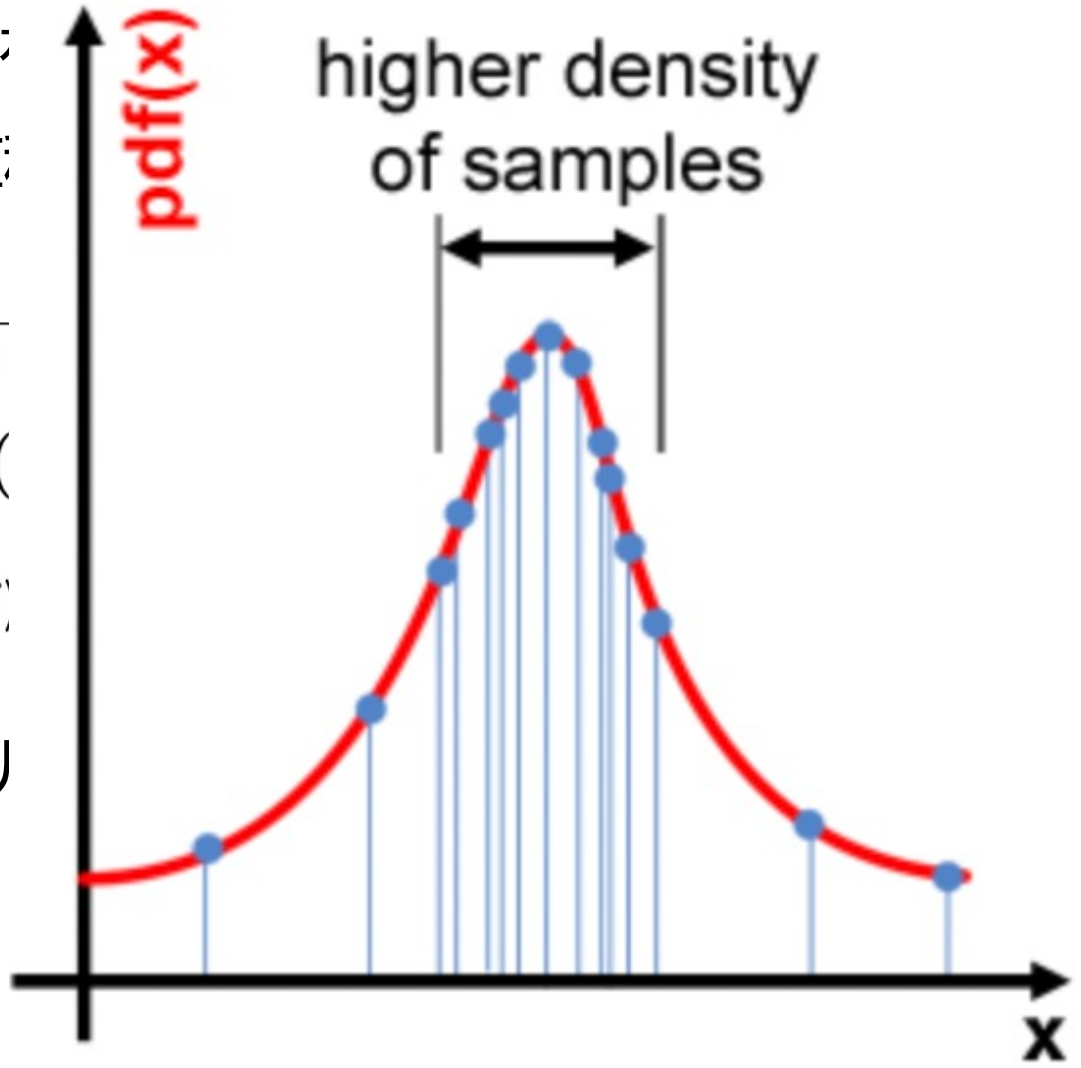
蒙特卡洛方法

- 蒙特卡洛方法归根结底
  - 计算某个积分，可在
  - 如果有个权重因子 $\rho($

$\overline{b}$

$\langle f$

这里 x_n 的取样满足

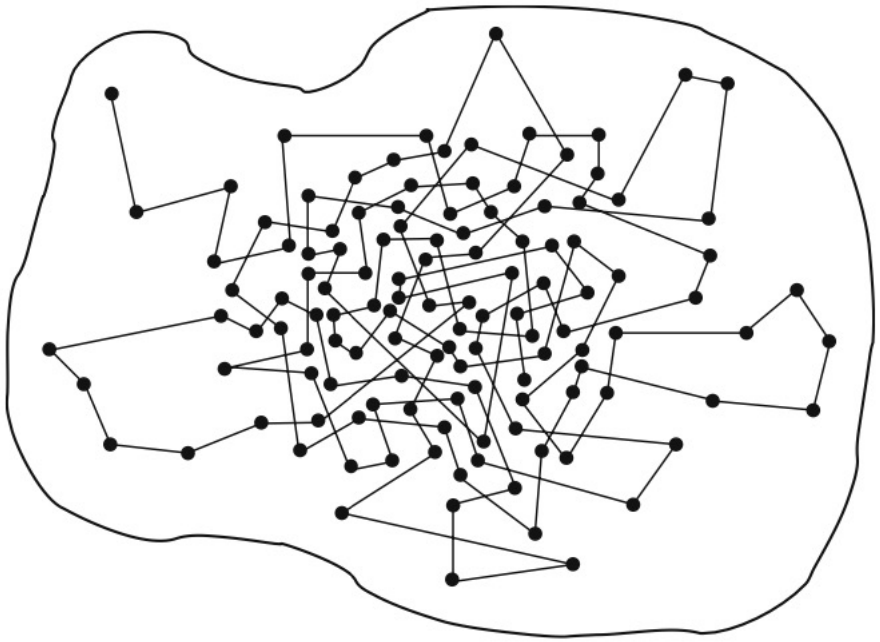


率密度是 $\rho(x_n) = \frac{1}{b-a}$

马尔科夫链演化

- 按照某个几率产生的组态很难直接获得，但可以从另一个组态中被“修改”出来
  - 从一个初始组态出发，按照一定规则+随机演化构造下一个组态
  - 继续这个过程，直到构造出满足分布几率的组态

$$\{\phi\}_0 \to \{\phi\}_1 \to \{\phi\}_2 \to \cdots \{\phi\}_n \to \cdots$$

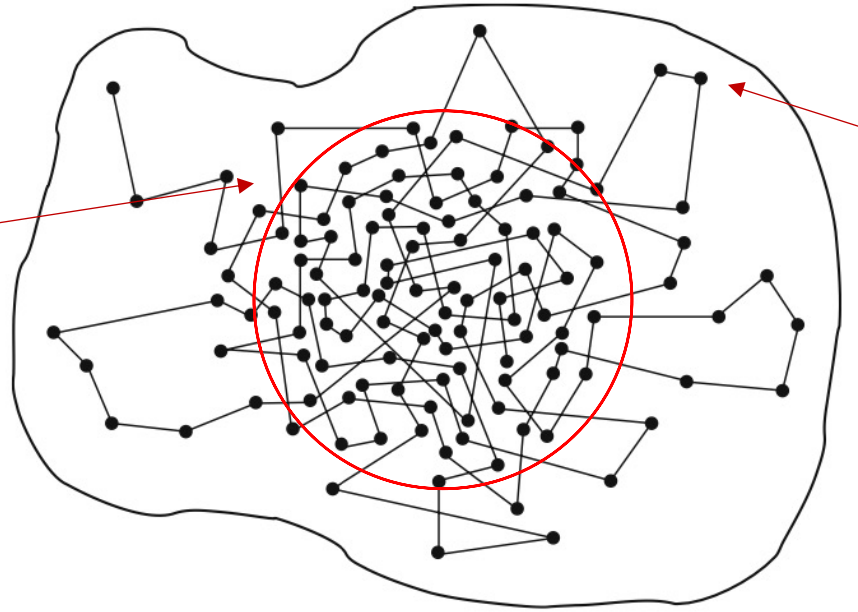


蒙卡模拟产生的组态

- 最简单的一维积分　　区间 [a,b] 内按权重 ρ(x) 产生的一组数 {x}
- 一维量子力学路径积分　等分时间间隔，组态由一系列 N+1 维的矢量给出

$$x = \{x(t_0), x(t_1), \cdots, x(t_N)\} = \{x_0, x_1, \cdots, x_N\}$$

- 标量场路径积分　　将场按空间 L 等分，时间 T 等分，离散为 L^3xT 维的高维矢量



S[Φ] 小的组态，出现可能性更大　$\frac{e^{-S[\phi]}}{Z}$

最小作用量原理的体现！

系综空间每个点对应于一个 L^3xT 维矢量 {Φ}

此即为组态

演化规律

- 在马尔科夫过程中，定义 ϕ 到 ϕ' 跃迁几率密度为 $W(\phi \to \phi')$

  W 定义了跃迁的规则，它满足 $\int \mathcal{D}[\phi']\, W(\phi \to \phi') = 1$

- 把组态 ϕ 出现的几率密度记为 $P[\phi]$

$$P^{(1)}[\phi'] = \int \mathcal{D}\phi\, W(\phi \to \phi') P^{(0)}[\phi]$$

- 通过跃迁规则进行演化，不停地更新可以产生一系列几率密度

$$P^{(0)}[\phi] \xrightarrow{W} P^{(1)}[\phi] \xrightarrow{W} \cdots P^{(n)}[\phi] \xrightarrow{W} \cdots$$

- 跃迁规则满足两个条件：
  - 各态历经：系统空间每个点都被走到，不存在死角
  - 稳定性条件：最后达到的几率分布，在 W 作用下是稳定的

$$e^{-S[\phi']} = \int \mathcal{D}[\phi]\, W(\phi \to \phi') e^{-S[\phi]}$$

Metropolis 算法实现马尔科夫过程

- 第一步：预选一种跃迁几率 $W_0(\phi \to \phi')$，使得组态 ϕ 能够以几率 W_0 的形式跃迁到 ϕ'
- 第二步：设置一定几率决定是否接受 ϕ' 作为新的组态，接收几率为

$$W_A(\phi \to \phi') = \min\left(1, \frac{W_0(\phi \to \phi')\exp(-S[\phi'])}{W_0(\phi' \to \phi)\exp(-S[\phi])}\right)$$

称为Metropolis Hastings → 构造一个以目标分布为稳定分布的马尔可夫链

- 第三步：重新回到第一步
- 结合前两步，从 ϕ 到 ϕ' 的跃迁几率为 $W(\phi \to \phi') = W_A(\phi \to \phi')W_0(\phi \to \phi')$

它同时满足各态历经和比稳定性条件更强的细致平衡条件

$$e^{-S[\phi']}W[\phi' \to \phi] = W[\phi \to \phi']e^{-S[\phi]}$$

Metropolis 算法实现马尔科夫过程

Algorithm 1 Metropolis 算法

Require:初始的组态记为 $\phi^{(0)}$, 选择一个适当的跃迁概率函数 $W_0[\{\phi\} \to \{\phi'\}]$.

1: **for** $k = 1, 2, \cdots$ **do**

2. 将 $\phi^{(k-1)}$ 更新为 $\phi^{(k)}$, 即

$$\{\phi^{(k-1)}\} \overset{W_0}{\Longrightarrow} \{\phi^{(k)}\}.$$

其中更新的规则就定义了 $W_0[\{\phi\} \to \{\phi'\}]$.

3. 以下述概率接受上面的更新:

$$P_A[\{\phi\}, \{\phi'\}] = \min\left(1, \frac{W_0[\{\phi'\} \to \{\phi\}]\mathrm{e}^{-S[\phi']}}{W_0[\{\phi\} \to \{\phi'\}]\mathrm{e}^{-S[\phi]}}\right).$$

特别地, 如果我们选择的 W_0 是对称的, 即 $W_0[\{\phi'\} \to \{\phi\}] = W_0[\{\phi\} \to \{\phi'\}]$, 这时的接受率可以表达为

$$P_A[\{\phi\}, \{\phi'\}] = \min\left(1, \mathrm{e}^{-\Delta S}\right),$$

其中 $\Delta S = S[\phi'] - S[\phi]$ 是更新后作用量的变化.

4. **end for**

Ising模型与Heatbath算法

➢ 考虑Ising模型作用量 $$S_{\mathrm{E}}[\phi] = -\kappa \sum_{x,\mu} \phi_x [\phi_{x+\mu} + \phi_{x-\mu}]$$

- 如果需要更新特定的ϕ_x，作用量中与其相关的项包括

$$S(\phi_x) = -(2\kappa) \sum_{\mu} \phi_x (\phi_{x+\mu} + \phi_{x-\mu})$$

- 之所以是2κ，是因为对x求和跑遍所有格点时$\phi_x \phi_{x+\mu}$会出现两次
- 如果一次仅更新一个固定的ϕ_x，暂时保持其他$\phi_{y\neq x}$不变，则希望产生的概率分布为

$$P(\phi_x) \propto \mathrm{e}^{-\phi_x \cdot J_x} \qquad J_x = -2\kappa \sum_{\mu} (\phi_{x+\mu} + \phi_{x-\mu})$$

- 这个指数分布可以直接产生：先在(0,1)之间产生均匀分布的随机数r，然后令$y = -\ln(r)$，那么得到的y就服从指数分布e^{-y}；如果令$y = -\frac{1}{J_x}\ln(r)$，则满足$e^{-J_x y}$的分布

Ising模型与Heatbath算法

Algorithm 2 Heatbath 算法

Require: 初始的组态记为 $\{\phi_x\}$.

1: **for** $k = 1, 2, \cdots$ **do**

2: **for** 对每个格点 x, **do**

3: 计算

$$J_x = -2\kappa \sum_{\mu} (\phi_{x+\mu} + \phi_{x-\mu})$$

4: 从下面的概率分布中产生新的场 ϕ_x:

$$P(\phi_x) \propto \mathrm{e}^{-J_x \phi_x}.$$

5: **end for**

6: **end for**

- 从第2步到第5步，被称为一“扫”(sweep)，也就是将每个场变量都扫一遍
- 扫一遍、两遍、三遍…，就构成了马尔科夫链；可以证明，算法满足细致平衡

Ising模型与Heatbath算法

- Heatbath 用的是“条件分布”，而不是“联合分布”

$$P(\phi_x|\phi_{\neq x}) = \frac{P(\phi)}{\sum_{\phi_x} P(\phi)} \qquad P(\phi_x|\phi_{\neq x}) = \frac{\frac{1}{Z}e^{-s[\phi]}}{\sum_{\phi_x=\pm 1}\frac{1}{Z}e^{-S[\phi]}}$$

配分函数Z完全抵消

- 我们真正需要的是

$$P(\phi_x = +1) = \frac{e^{-J_x}}{e^{J_x} + e^{-J_x}} \qquad P(\phi_x = -1) = \frac{e^{+J_x}}{e^{J_x} + e^{-J_x}}$$

- 对于马尔科夫链蒙特卡洛（MCMC）

Metropolis 用的是： $\frac{P(\phi_x \to -\phi_x)}{P(\phi_x \to \phi_x)} = e^{-\Delta S}$ 配分函数Z也被抵消掉了

Ising模型与Heatbath算法

➢ Heatbath

- 局部更新，每次只更新一个场变量
- 直接从条件概率分布中抽样，无需Metropolis 接受-拒绝纠正误差
- 属于纯随机马尔科夫链，非常“干净”的随机更新

➢ 缺点

- 完全局部，容易慢
- 每次更新并没有使得组态产生足够多的改变，两个相邻组态之间关联性极强

Hybrid Monte Carlo (HMC) 算法

- HMC是从分子动力学算法衍生出来的蒙特卡洛算法
- 假定我们需要产生概率分布

$$P[\phi] \propto \mathrm{e}^{-S[\phi]}$$

$S[\phi]$代表体系作用量，ϕ标记体系所有的自由度

- 自然界中空气分子数密度的分布就是玻尔兹曼分布$e^{-\beta H} \sim e^{-TH} \sim e^{-S}$

这里的 $\beta = \frac{1}{k_B T_{temp}}$

玻尔兹曼分布描述系统在温度T_{temp}下处于能量为E_i的态的概率为$e^{-\beta E_i}$

自然界中天然实现玻尔兹曼分布的机制就是靠分子的运动和碰撞

Hybrid Monte Carlo (HMC) 算法

- 分子的运动可以按照经典力学来描写，碰撞可以看成不断更新的动能
- 将ϕ视作体系的广义坐标，$S[\phi]$视为仅依赖于坐标的势能，引入与ϕ共轭的广义动量π
- 定义体系的哈密顿量为 $\mathcal{H}[\pi,\phi]=\sum_x \frac{\pi_x^2}{2}+S[\phi]$
- 要求体系在一个虚拟的时间τ做经典的演化，即按照哈密顿正则方程演化

$$\begin{cases}\dot{\pi}_x \equiv \dfrac{\mathrm{d}\pi_x}{\mathrm{d}\tau}=-\dfrac{\partial\mathcal{H}[\pi,\phi]}{\partial\phi_x}=-\dfrac{\partial S[\phi]}{\partial\phi_x}, \\ \dot{\phi}_x \equiv \dfrac{\mathrm{d}\phi_x}{\mathrm{d}\tau}=\dfrac{\partial\mathcal{H}[\pi,\phi]}{\partial\pi_x}=\pi_x. \end{cases}$$

- 随时间演化，利用哈密顿动力学流来生成一个新的组态，或者说ϕ_x在位形空间中形成一条轨迹，轨迹上的每个点对应的总哈密顿量$\mathcal{H}$是守恒的

Hybrid Monte Carlo (HMC) 算法

- 当我们产生分布

$$e^{-\mathcal{H}} = \exp\left(-\sum_x \frac{\pi_x^2}{2}\right) \exp(-S[\phi]),$$

  - 体系关于动能和势能的概率分布是完全独立的
  - 势能部分对应我们需要的玻尔兹曼分布
  - 动能部分是标准的高斯分布（正态分布）

- 实际计算过程中，需要将无穷长的分子的轨迹分成若干段独立的径迹

  比如将运动方程从$\tau = 0$积分到$\tau = \tau_0$；然后重新产生高斯分布的动量π_x，再从$\tau = \tau_0$积分到$\tau = 2\tau_0$；再产生动量，再继续

  每过一段虚拟时间τ_0重新从高斯分布中产生新的动量实际上就是模拟了“分子碰撞”

  ➡ 经过碰撞，分子忘掉原先的速度，获得新的速度，然后在力场下按牛顿方程运动

Hybrid Monte Carlo (HMC) 算法

- 将运动方程积分到$\tau = N\tau_0$，就获得了N个组态，$\{\phi_x^{(i)} : i = 1, 2, \cdots, N\}$

  τ_0的选取依赖于所模拟的系统的具体物理性质

- 真实情况是，场论对应的哈密顿方程几乎不可能获得解析解，只能得到数值近似解

  常用的处理微分方程的方法是将τ离散化，引入积分步长$\delta\tau$

  ➡ 哈密顿量守恒仅对 $\delta\tau \to 0$ 适用

  对于有限步长，哈密顿量会产生改变 $\Delta\mathcal{H} = \mathcal{H}(\tau = \tau_0) - \mathcal{H}(\tau = 0) \neq 0$

  采用Runge-Kuta积分方法，可以使得 $\Delta\mathcal{H} \propto (\delta\tau)^2$

$$H(p,q) = \frac{p^2}{2m} + V(q), \qquad \dot{q} = \frac{\partial H}{\partial p}, \qquad \dot{p} = -\frac{\partial H}{\partial q}$$

- Step 1 - 动量半步更新

$$p\left(t + \frac{\epsilon}{2}\right) = p(t) - \frac{\epsilon}{2}\frac{\partial V}{\partial q}(q(t))$$

- Step 2 – 位置整步更新

$$p\left(t + \frac{\epsilon}{2}\right) = p(t) - \frac{\epsilon}{2}\frac{\partial V}{\partial q}(q(t))$$

- Step 3 - 动量再半步更新

$$p(t + \epsilon) = p\left(t + \frac{\epsilon}{2}\right) - \frac{\epsilon}{2}\frac{\partial V}{\partial q}(q(t + \epsilon))$$

Hybrid Monte Carlo (HMC) 算法

- 对于场的系统，也有类似的蛙跳算法（Leapfrong integration）

$$
\begin{cases}
\pi_x\left(\frac{\delta\tau}{2}\right) = \pi_x(0) - \frac{\partial S[\phi(0)]}{\partial \phi_x}\cdot\frac{\delta\tau}{2}, \\
\phi_x(\delta\tau) = \phi_x(0) + \pi_x\left(\frac{\delta\tau}{2}\right)\cdot\delta\tau, \\
\pi_x(\delta\tau) = \pi_x\left(\frac{\delta\tau}{2}\right) - \frac{\partial S[\phi(\delta\tau)]}{\partial \phi_x}\cdot\frac{\delta\tau}{2}.
\end{cases}
$$

- 在实际计算中，我们将上述步骤重复N_{step}次，得到径迹长度为$\tau_0 = N_{step}\delta\tau$
- 由于 $\Delta\mathcal{H} \propto (\delta\tau)^2$ 还是需要按照Metropolis接受-拒绝条件进行组态取舍

Hybrid Monte Carlo (HMC) 算法

Algorithm 4 杂化 Monte Carlo 算法

Require: 初始的组态记为 $\{\phi_x(0)\}$. 每个径迹长度为 $\tau_0 = N_{\text{step}}\delta\tau$, 其中 $\delta\tau$ 为步长.

1: **for** $k = 1, 2, \cdots$ **do**

2: 对每一个场自由度 ϕ_x, 从高斯分布中产生与它共轭的动量 π_x:

$$P[\pi_x] \propto \exp\left(-\frac{\pi_x^2}{2}\right).$$

3: 利用蛙跳积分法　　将运动方程积分 N_{step} 步.

4: 计算径迹初态和末态哈密顿量的改变 $\Delta\mathcal{H} = \mathcal{H}(\tau_0) - \mathcal{H}(0)$.

5: **if** $\Delta\mathcal{H} < 0$,**then**

6: 　接受新的组态.

7: **else**

8: 　产生一个在 $(0, 1)$ 区间均匀分布的随机数 r.

9: 　**if** $r < e^{-\Delta\mathcal{H}}$, **then**

10: 　　接受新的组态.

11: 　**else**

12: 　　恢复本径迹初的组态.

13: 　**end if**

14: **end if**

15: 回到第 1 步开始新的一条径迹.

16: **end for**

以标量场为例

- 考虑一个D维格点系统，格距为a，长度为L=Na

$$x = a(n_0, \ldots, n_{D-1}) \; ; \quad n_i \in \mathbb{N}_0 \; , \quad 0 \le n_i < N$$

- 取周期性边界条件 $\phi_{x+L\hat{\mu}} = \phi_x$
- 作用量取为

$$S(\phi) = \sum_x \left[-2\kappa \sum_{\mu=0}^{D-1} \phi_x \phi_{x+\hat{\mu}} + \phi_x^2 + \lambda(\phi_x^2 - 1)^2 \right]$$

- 物理期望值

$$\langle A \rangle = \frac{1}{Z} \int \prod_x \mathrm{d}\phi_x \exp(-S(\phi)) A(\phi) \qquad Z = \int \prod_x \mathrm{d}\phi_x \exp(-S(\phi))$$

- 磁化强度 $m = \sum_x \phi_x$
- 磁化率 $\chi = \frac{1}{V}\langle m^2 \rangle$
- 四阶累积量 $U = \frac{\langle m^4 \rangle}{(\langle m^2 \rangle)^2}$
- 物理量对κ的导数 $\frac{\partial}{\partial \kappa}\langle A \rangle = \langle WA \rangle - \langle W \rangle \langle A \rangle \qquad W = 2 \sum_x \sum_{\mu=0}^{D-1} \phi_x \phi_{x+\hat{\mu}}$

以标量场为例

- 第一步：写出作用量$S[\phi]$

```
1  double action(void)
2  {
3      int i;
4      double J;
5
6      S=0;
7      for (i=0;i<V;i++) /* loop over all sites */
8      {
9          /*sum over neighbors in positive direction*/
10         J=0.0;
11         for (mu=0;mu<2*D;mu++) J+=phi[hop[i][mu]];
12
13         phi2=phi[i]*phi[i];
14         S+=-2*kappa*J*phi[i]+phi2+lambda*(phi2-1.0)*(phi2-1.0);
15     }
16     return S;
17 }
```

这里的hop场是个hop[V][2D]维的数组，前D个表示前向耦合，后D个表示反向耦合

以标量场为例

- 第二步：产生高斯型分布 $P(x) \propto e^{-x^2/2}$ ，构造动量场 mom[V]
- 第三步：利用随机数生成场$\phi[V]$（随机初始化或称为热初态）

  保留一份$\phi[V]$，记为$\phi_{old}[V]$，做Metropolis hastings用
- 第四步：利用ϕ和mom计算分子动力学哈密顿量 $\mathcal{H}$

以标量场为例

- 第五步：给出ϕ和mom的更新

```
1  void move_phi(double eps)
2  {
3    int i;
4    for (i=0;i<V;i++) phi[i]+=mom[i]*eps;
5  }
6
7
8  void move_mom(double eps)
9  {
10   int i,mu;
11   double J, force;
12
13   for (i=0;i<V;i++)
14   {
15      J=0;
16      for (mu=0;mu<2*D;mu++) J+=phi[hop[i][mu]];
17
18      force=2*kappa*J-2*phi[i]-lambda*4*(phi[i]*phi[i]-1)*phi[i];
19      mom[i]+=force*eps;
20   }
21 }
```

以标量场为例

- 第六步：检验Leapfrog integration给出的误差是$O(\epsilon^2)$

使用4^4的格子，$\kappa = 0.18169$，$\lambda = 1.3282$，产生1000个组态，检验每一个 $\Delta\mathcal{H}$

比Leapfrog更优化的是Omelyan积分器

$$[\mathcal{I}_1(\epsilon/2)\mathcal{I}_2(\epsilon)\mathcal{I}_1(\epsilon/2)]^{N_s}$$

Leapfrog

$$[\mathcal{I}_1(\xi\epsilon)\mathcal{I}_2(\epsilon/2)\mathcal{I}_1((1-2\xi)\epsilon)\mathcal{I}_2(\epsilon/2)\mathcal{I}_1(\xi\epsilon)]^{N_s}$$

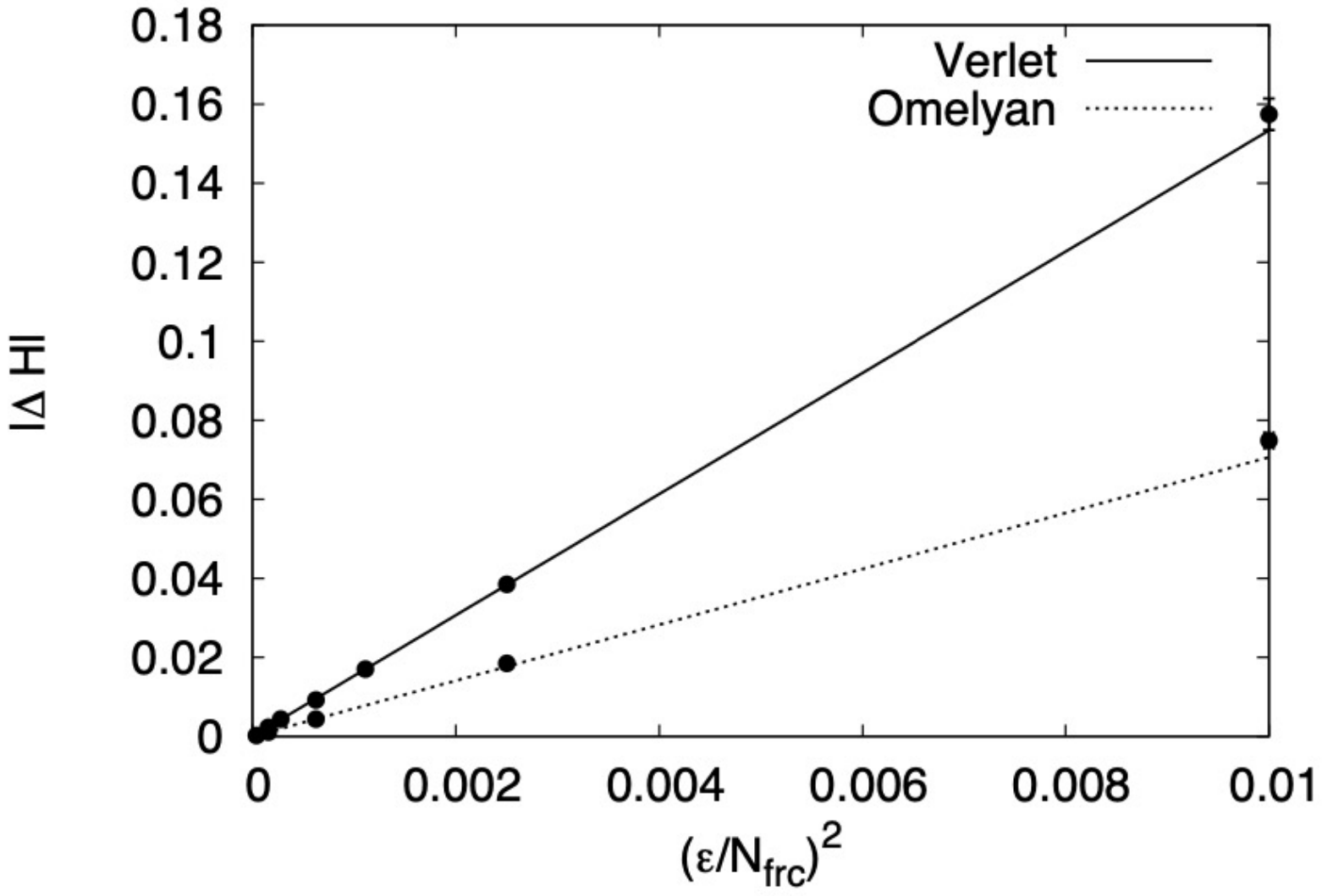
Omelyan

ξ是任意可调的参数，建议值为0.1931833

- 第七步：可以检验积分器的可逆性

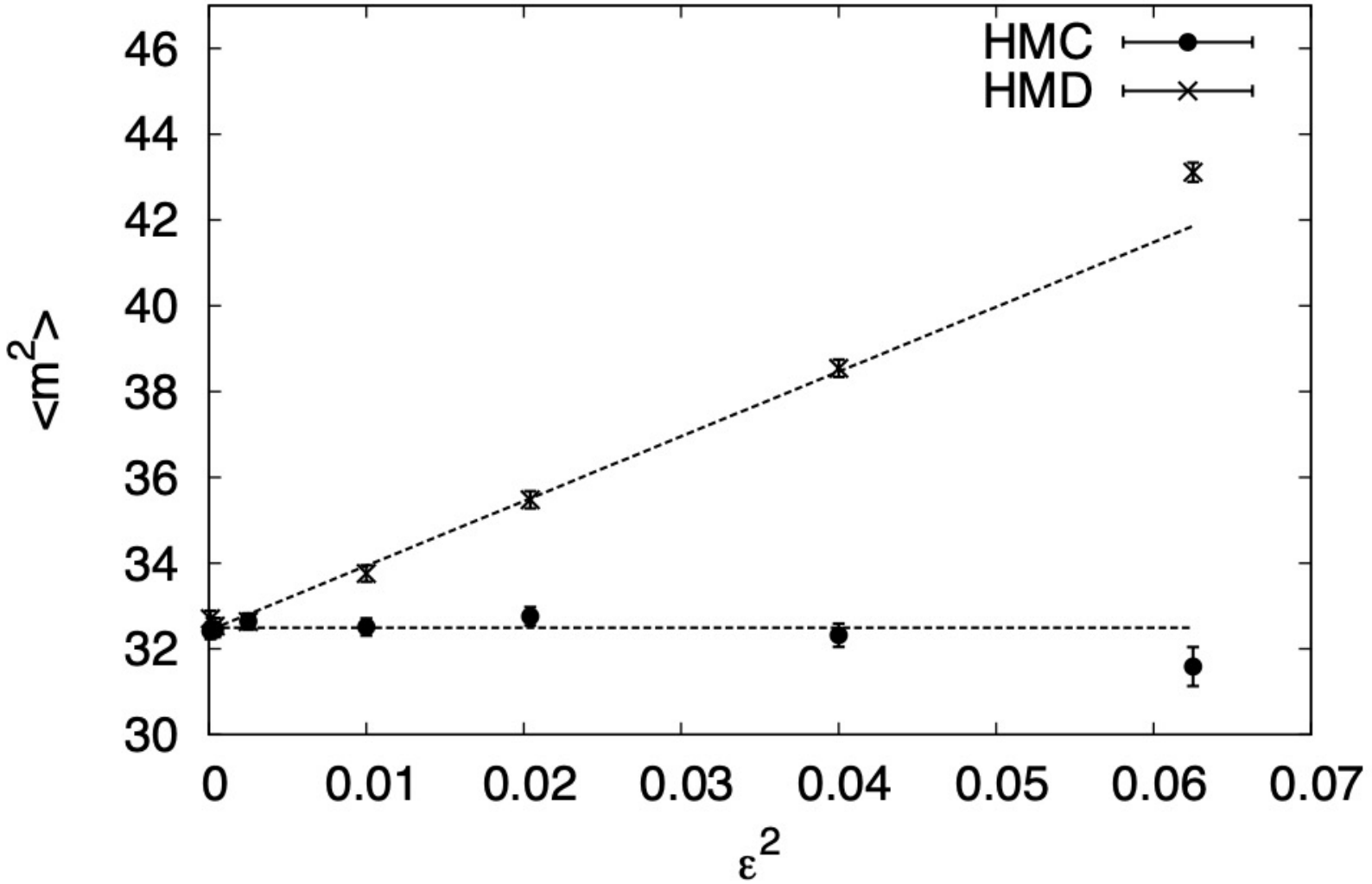
反向执行Leapfrog只需反转操作顺序，令步长$\epsilon \to -\epsilon$

以标量场为例

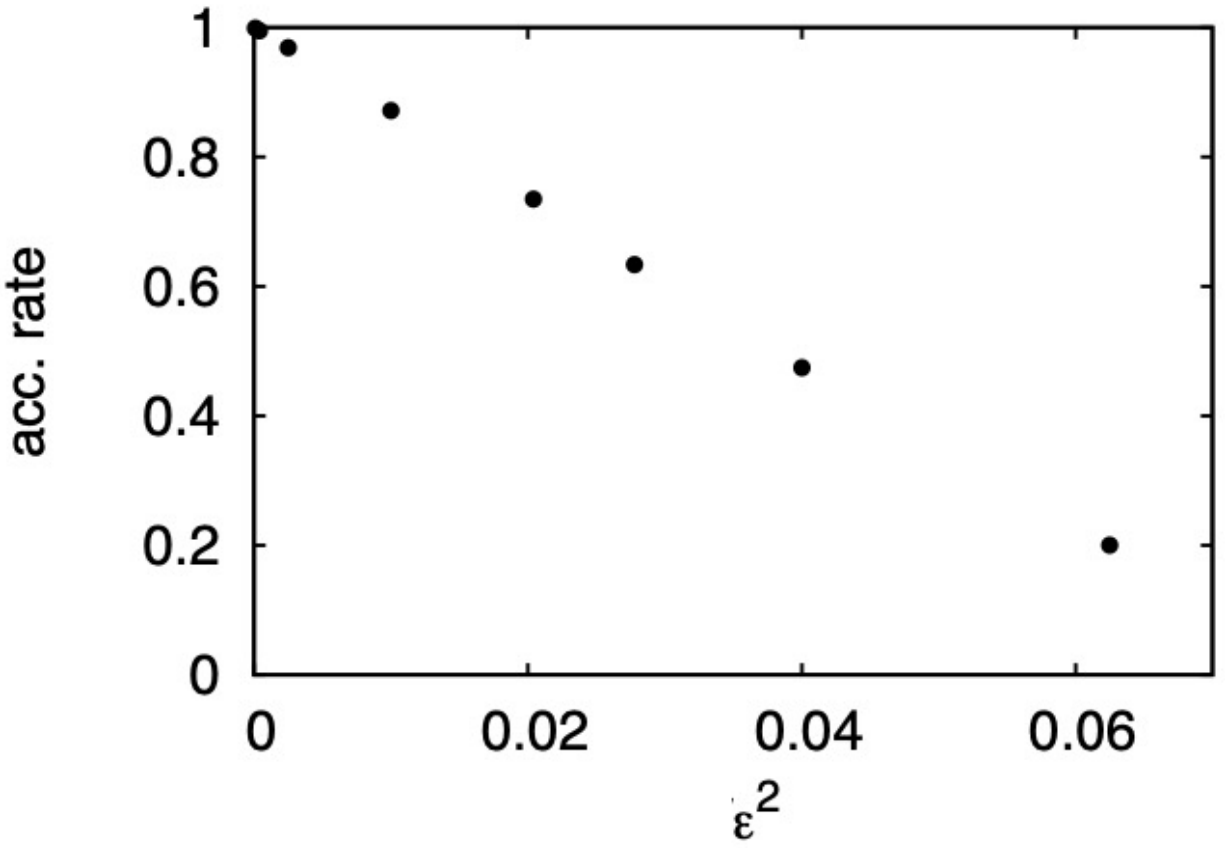

0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
|Δ H|
Verlet
Omelyan
0
0.002
0.004
0.006
0.008
0.01
(ε/N_frc)^2

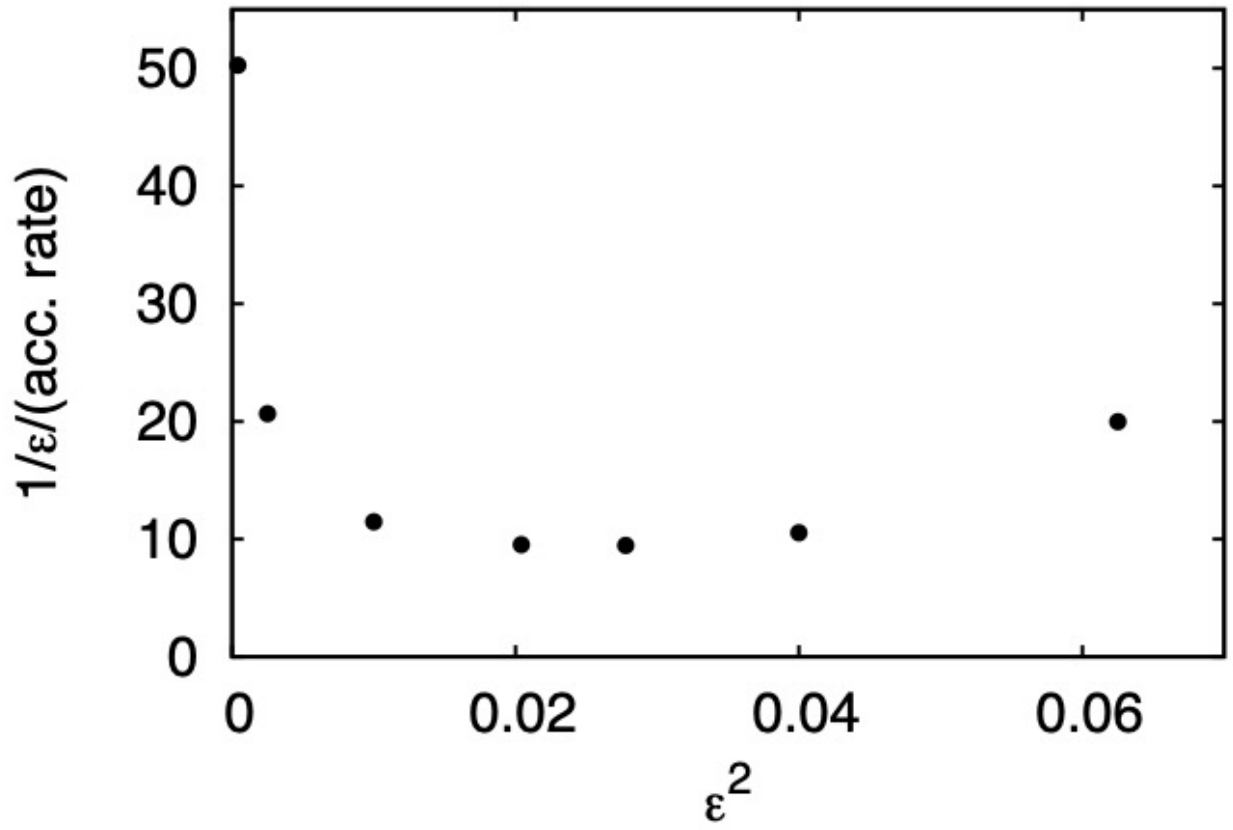
以标量场为例

- 第七步：Metropolis hastings：接收/拒绝



以标量场为例





每产生一个组态的开销对应50%-90%的接收率

以标量场为例

- 热化（Thermalization）

  马尔科夫链可以从任意组态开始，经过足够多次迭代，初始分布中不正确的成分被逐渐消除，模拟达到平衡——获得正确的分布

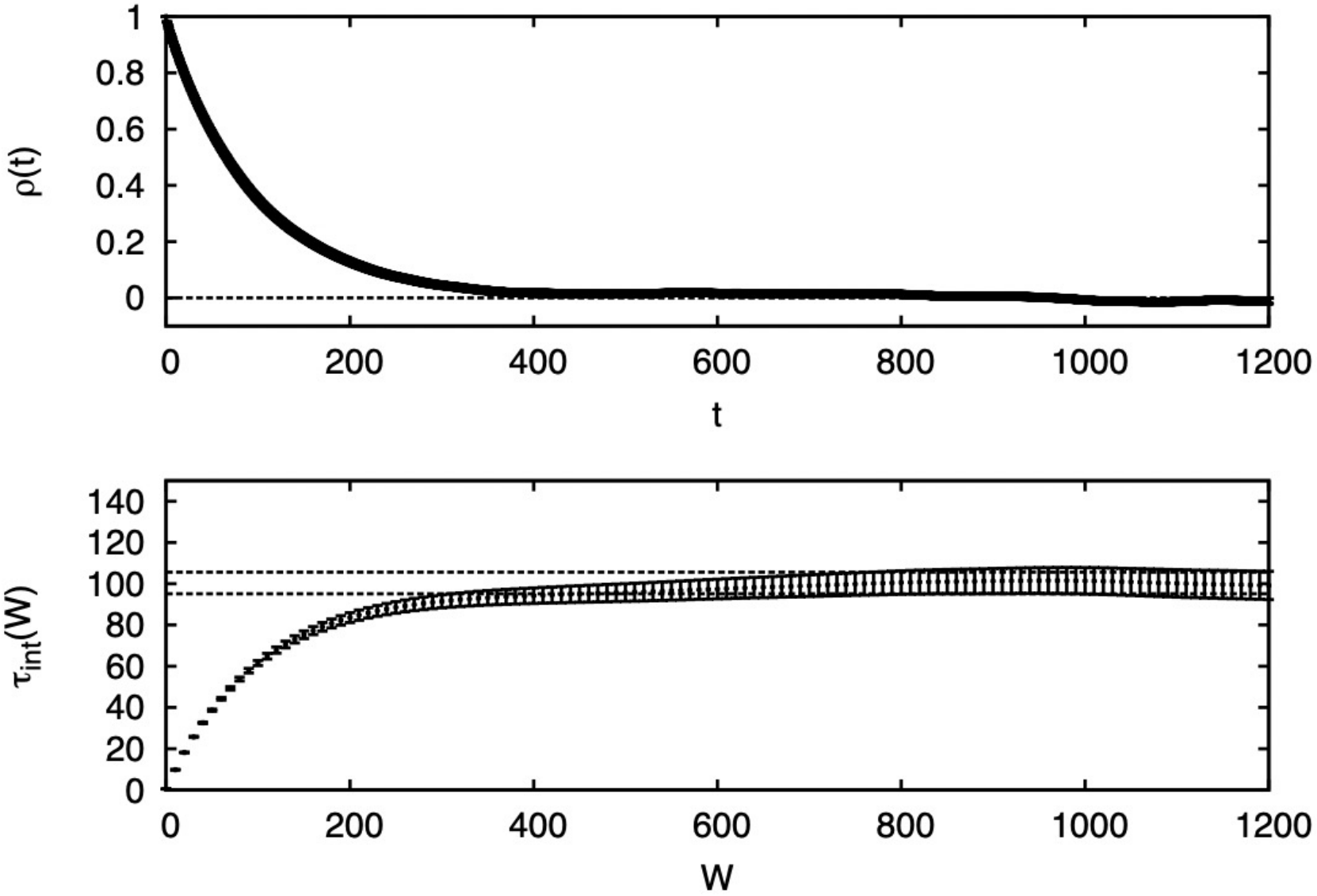
- 自关联（Auto correlation）

  自关联与物理量相关 $F = F(A^1, A^2, \cdots, A^n)$ $f^\alpha = \partial F / \partial A^\alpha$

$$\Gamma_{\alpha\beta}(t) = \langle (A_i^\alpha - \langle A^\alpha \rangle) \left(A_{i+t}^\beta - \langle A^\beta \rangle \right) \rangle$$

$$\tau_{int} = \frac{1}{2} + \frac{1}{v_F} \sum_{t=1}^{\infty} \sum_{\alpha\beta} f_\alpha f_\beta \Gamma_{\alpha\beta}(t) = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_F(t) \qquad v_F = \sum_{\alpha\beta} f_\alpha f_\beta \Gamma_{\alpha\beta}(0)$$

以标量场为例


ρ(t)
1
0.8
0.6
0.4
0.2
0
0
200
400
600
800
1000
1200
t
$\tau_{int}(W)$
140
120
100
80
60
40
20
0
0
200
400
600
800
1000
1200
W

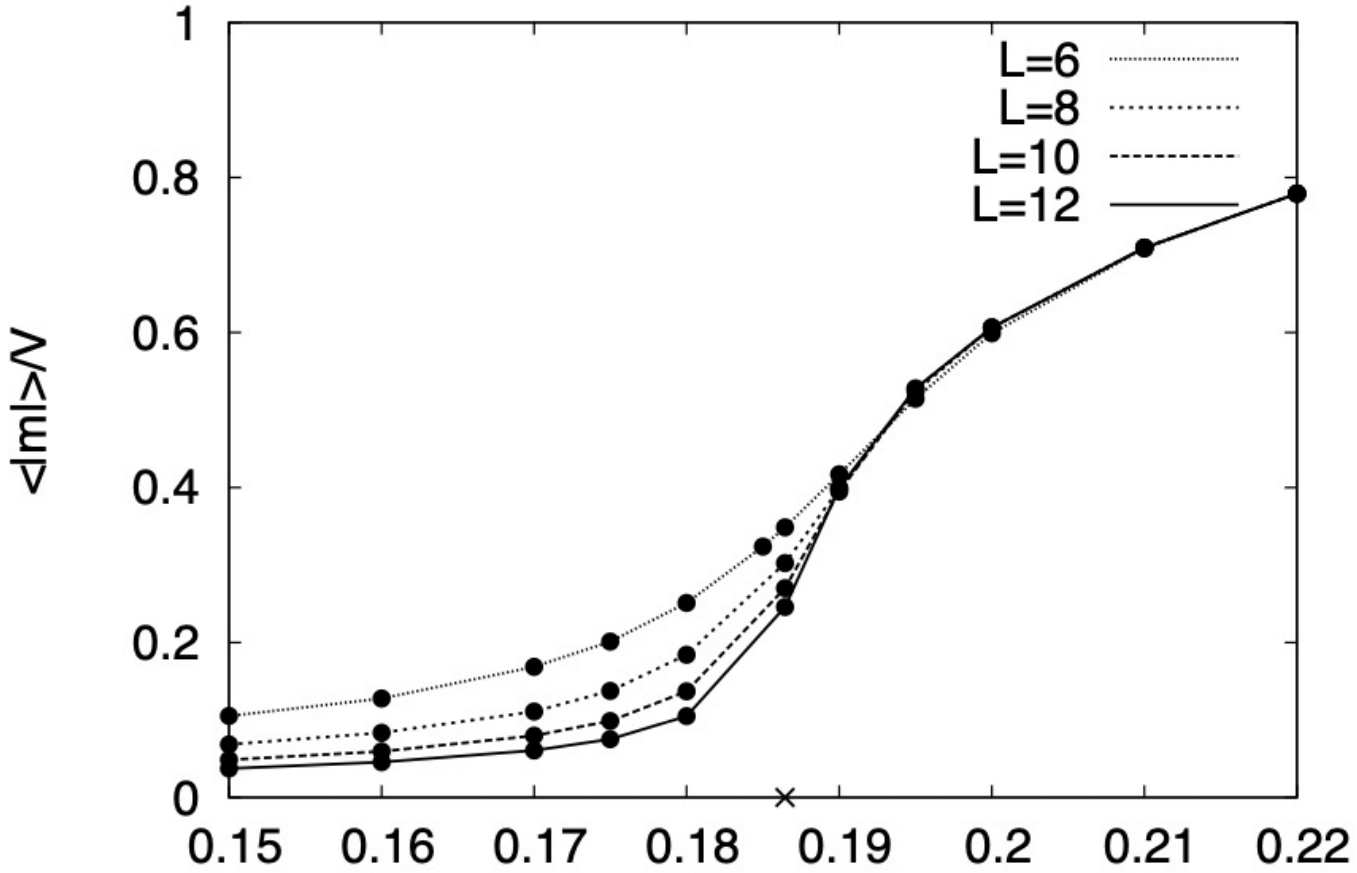
以标量场为例

- 设置$L=6,\kappa=0.185825,\lambda=1.1689$，这个参数很接近临界点。设组态径迹长度为1，步长$\epsilon=0.05$，从随机初始组态出发，生成5000个组态，观察热化现象。与$\kappa=0.1$和$\kappa=0.2$的情况比较，看何时达到平衡

- 现在生成10^5个组态，在达到热平衡之后开始测量。先不考虑自关联效应，对1000个连续的组态求平均，然后再进行误差分析

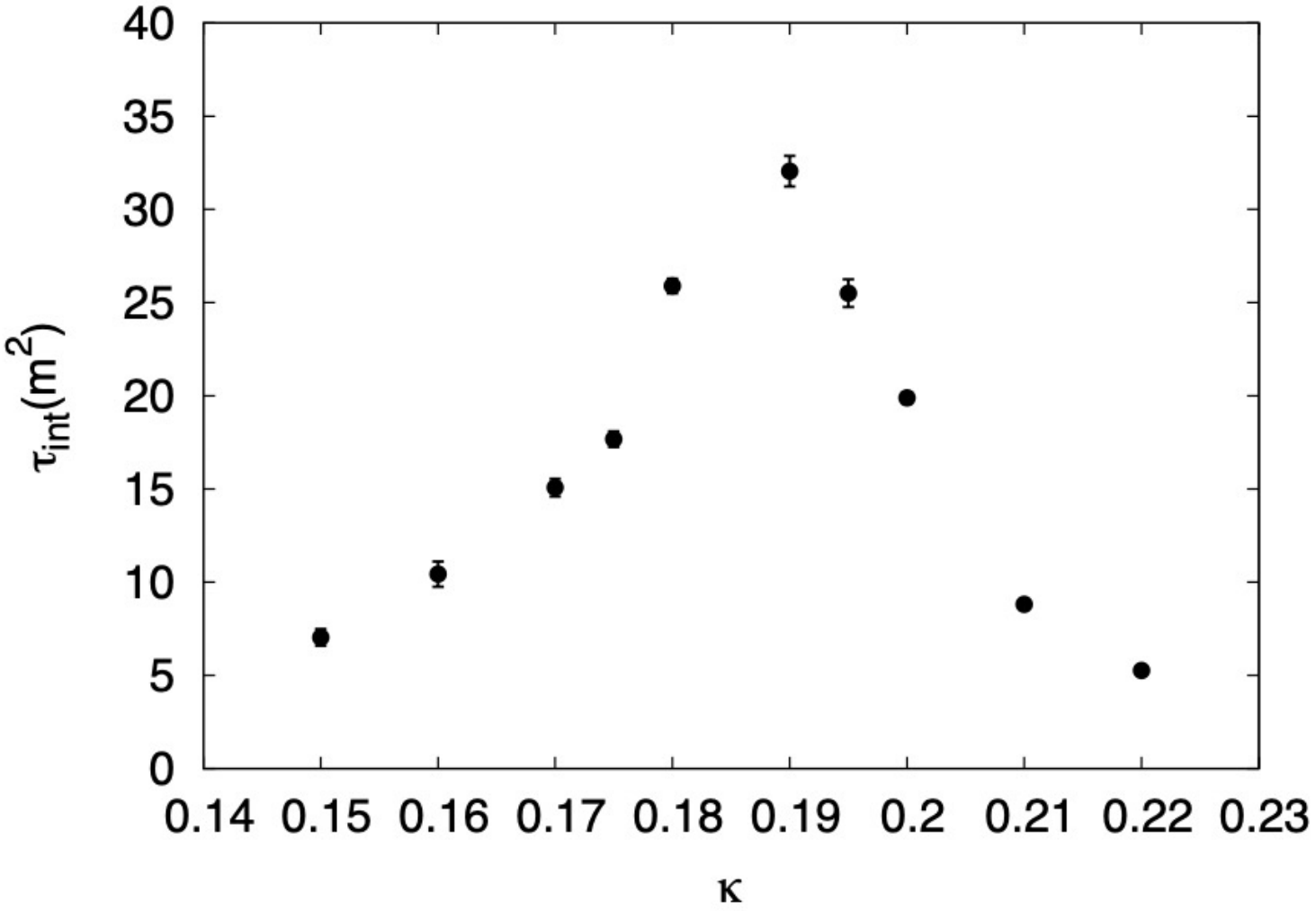
- 确认你的程序是正确的。一个判定要求是，结果应该和步长的选取无关；如果步长变大，只是接收率会降低，但物理量的期望值不应该受到影响

以标量场为例



- 在低温/大κ时，对称性自发破缺，系统有两个对称的平衡态$m = \pm m_0$
- 无穷体积下，系统会 “选择”一个（对称性破缺），另一态几乎不会访问
- 严格热力学极限下，统计力学的平均应在一个平衡态中取平均
- 有限体积下，自由能势垒是有限的，系统会在两个态之间随机翻转

以标量场为例



- 越接近临界点，自关联长度越长

传统蒙卡的本质是局域更新

- Heatbath 算法：严格局域
  - 每次只更新一个ϕ_x场，其变化只依赖临近的J_x
- HMC 算法：弱非局域更新
  - 每次更新所有的ϕ_x场变量，貌似全局更新
  - 更新的动力学方程是一个局域哈密顿动力系统，每一个格点的演化由局部力（local force）决定
  - HMC采用“分子动力学”力 $\dot{\pi}_x \equiv \frac{\mathrm{d}\pi_x}{\mathrm{d}\tau} = -\frac{\partial \mathcal{H}[\pi,\phi]}{\partial \phi_x} = -\frac{\partial S[\phi]}{\partial \phi_x}$

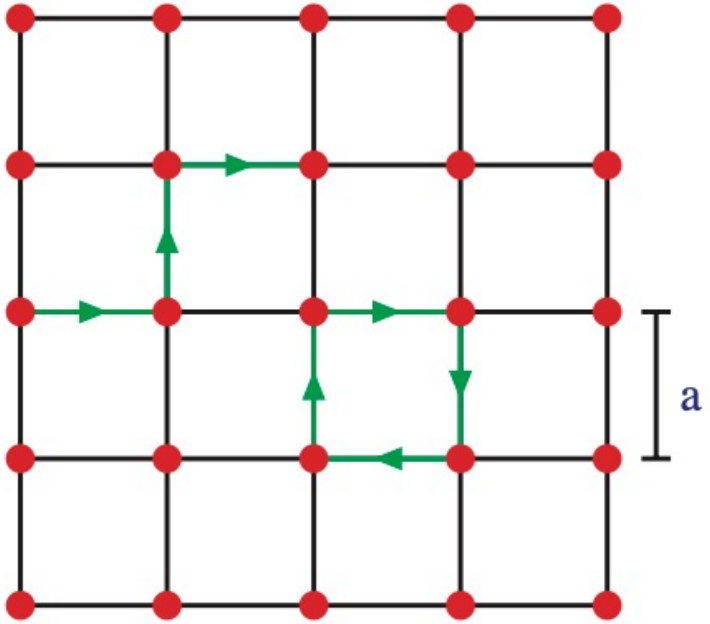
而作用量 S 是局域的 → 因此每个格点独立受到局域作用力，整个系统更新是一个沿着高维相空间的连续路径，是局部动力学“缓慢滑动”，而不是全局随机跳跃

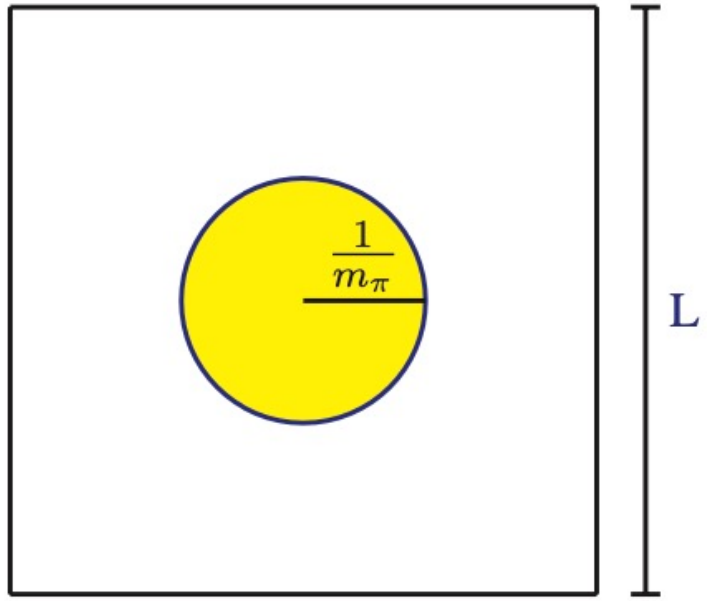
→ 因此导致自关联很长

大规模数值模拟与格点QCD

将夸克场和胶子场都放在网格上

- 夸克场位于格点上，$\psi(x)$，$x_\mu = n_\mu a$
- 胶子场由格点之间的链接来表示 $U_\mu(x) = e^{iagA_\mu(x)}$





- 计算机只能模拟有限的自由度 $\Rightarrow$ 格距a不能是无穷小，格子长度L不能是无穷大
- $N = L/a \sim 32, 48, 64, \cdots$，考虑到 4 维时空，总的格点数就是 N^4

蒙卡模拟产生的组态

- 量子色动力学路径积分 $Z = \int [\mathcal{D}\psi][\mathcal{D}\bar{\psi}][\mathcal{D}U]\, e^{-S_f[\psi,\bar{\psi},U]} e^{-S_g[U]}$

每个时空格点上的费米子场都要进行积分

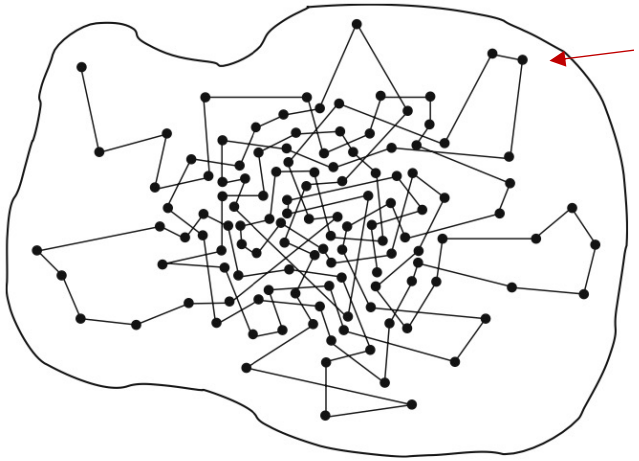
- 利用Grassmann代数将费米子场积掉

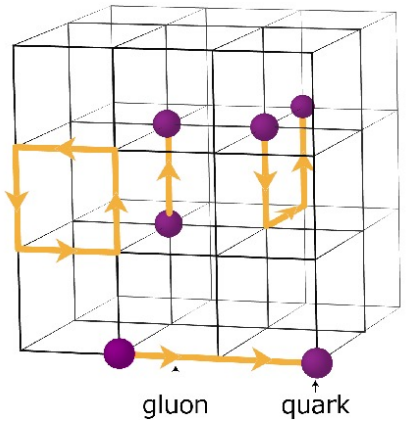
$$Z = \int [\mathcal{D}U] \det(\not{D} + m)[U] e^{-S_g[U]}$$

只留 SU(3) 规范场的积分，U 是 3x3 的幺正矩阵

- 蒙特卡洛重点采样——生成满足分布几率的组态

$$p[U] = \frac{\det(\not{D} + m) e^{-S_g[U]}}{Z}$$





规范场定义在联结格点的链接上

系综每个组态对应于 L^3xTx4 个3x3矩阵的集合

物理量的计算

- 最简单的一维积分

$$\frac{1}{b-a}\int_a^b dx\, f(x) = \langle f \rangle_\rho = \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f(x_n)$$

- 格点QCD中研究的物理量从关联函数中提取

$$\langle O \rangle = \frac{1}{Z}\int [\mathcal{D}\psi][\mathcal{D}\bar{\psi}][\mathcal{D}U]\, e^{-S_f[\psi,\bar{\psi},U]} e^{-S_g[U]}$$

$$= \frac{1}{Z}\int [\mathcal{D}U]\, O[U] \det(\not{D} + m) e^{-S_g[U]}$$

路径积分由组态求平均来近似

$$\frac{1}{Z}\int [\mathcal{D}U] \det(\not{D} + m) e^{-S_g[U]} \quad \to \quad \frac{1}{N}\sum_{\{U\}}$$

$$\langle O \rangle \quad \to \quad \frac{1}{N}\sum_{\{U\}} O[U]$$

组态数 N 越大，结果越精确；统计误差随 N 增加按 $\frac{1}{\sqrt{N}}$ 减小

格点QCD受到计算成本的限制

- 世界级百亿亿次级超级计算机（exascale，10^{18}次运算每秒，相当于五千万台笔记本电脑同时工作）
  - 如果计算范式没有突破，许多重要研究仍将无法实现
  - 如果能大幅降低格点场论的计算成本，粒子物理、核物理中的基础问题将能够得到解答
  - 从头算的计算可以 研究核物理中的精细调控（fine-tuning）

    揭示宇宙中碳的生成（通过三α过程）对理论自由参数的敏感性

    解释质子和中子为何在原子核中聚集

    阐明宇宙早期大爆炸核合成（Big Bang nucleosynthesis）中最轻元素的形成机制

格点QCD受到计算成本的限制

- 目前最先进的格点QCD计算涉及的格点大小可达

$$256^3 \times 512 = 86\text{亿个格点}$$

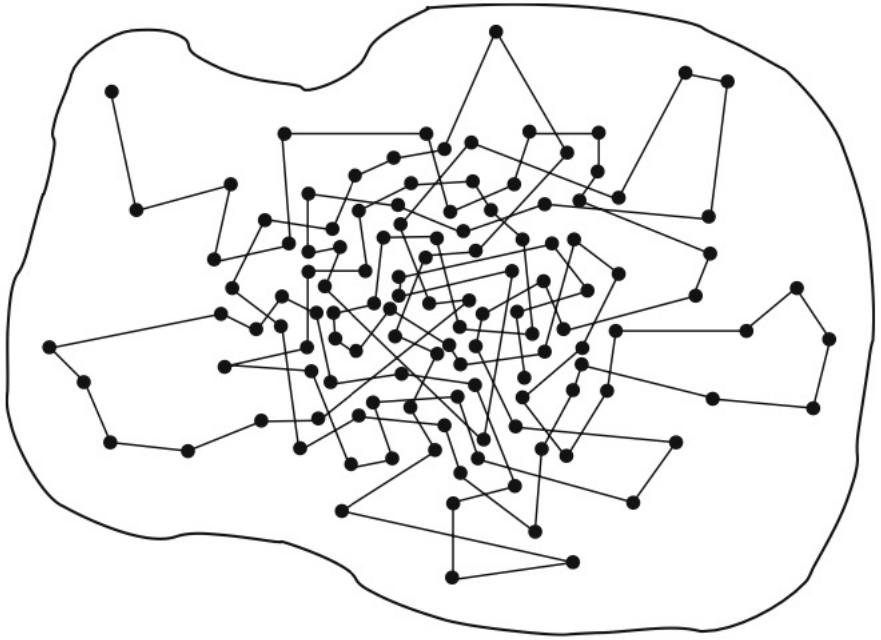
每个格点上量子场大约有 50 个自由度

- 每个格点对应4个SU(3)矩阵，每个矩阵8个生成元，共32个自由度
- 每个格点的复4×3矢量，对应费米子场，用$2\times4\times3 = 24$个自由度

实际上计算涉及对多达 10^{12} 个变量进行蒙特卡洛积分

Generative Models

传统蒙卡面临的几个挑战



- 临界减速(Critical Slowing Down)

  样本高度关联性，随格距变小，急剧增长

- 符号问题

  有限密度，化学势不为零，$S[\phi]$ 非正定

- 反问题

  定义在欧氏时间上的关联函数 $\langle O(t)O(0)\rangle = \int dE\,|\langle 0|O|E\rangle|^2\, e^{-Et} = \int dE\,\rho(E)\, e^{-Et}$

  因为欧氏关联函数不同于闵氏，因此需提取谱权重 $\rho(E)$

替代方法

- 任何用于采样格点场配置的替代方法，要在实际应用中可行，需要满足几个关键要求

① 它必须是可统计改进的（statistically improvable）：在样本数足够大的极限下，必须能够恢复真实的概率分布 ，包括该分布所遵守的各种对称性。

② 该方法必须能够高效扩展到最先进的格点场论研究规模，这些研究涉及的场配置每个可占用几个 TB 内存，总自由度可达 10^{12}

② 该方法必须在某些物理感兴趣的区域内改进 HMC 框架的表现，并缓解临界减速（critical slowing-down）和拓扑冻结（topological freezing）等挑战

生成式模型（Generative Models）

- 希望按照某个分布来产生样本

$$\mathbf{X} = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\} \overset{i.i.d}{\sim} p_{data}(x)$$

- 我们用参数化的模型来逼近目标分布

$$p_\theta(x) \rightarrow p_{data}(x)$$

- 量化模型逼近的好坏——Kullback-Leibler散度（KL散度）

$$\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

- 训练模型时的两种方案
  - 最小化正向KL散度（相当于在做最大似然估计）
  - 最小化反向KL散度

KL散度

$$\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

- KL散度的正定性

对任意正数t，有经典不等式 $\ln t \leq t - 1$；等号仅当$t = 1$时取到

取$t = \frac{q(x)}{p(x)}$，则有

$$\int dx\, p(x) \log \frac{q(x)}{p(x)} \leq \int dx\, p(x) \left(\frac{q(x)}{p(x)} - 1 \right) = \int dx\, q(x) - \int dx\, p(x) = 1 - 1 = 0$$

于是，有 $KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} \geq 0$

KL散度的极小值对应于 $p(x) = q(x)$

前向KL散度

Forward KL： $\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$

- 重点考察 $p(x) \log q(x)$

  如果真实分布$p(x) > 0$，而模型给出$q(x) \approx 0$，则 $\log \frac{1}{q(x)} \to \infty$，这意味着漏掉真实mode会产生无限大惩罚

- Forward KL容忍 q(x) 多覆盖一些地方，但不允许 q 把 p 有的地方漏掉

Forward KL 会画一个很大的山把两座峰都覆盖进去

反向KL散度

Reverse KL：$\mathrm{KL}(q\|p) = \int q(x) \log \frac{q(x)}{p(x)} dx$

- 重点考察 $q(x) \log p(x)$

  如果模型给出概率 $q(x) > 0$，而真实 $p(x) \approx 0$，则 $\log \frac{1}{p(x)} \to \infty$，这意味着模型覆盖到真实没有的区域会产生无限大惩罚

- Reverse KL容忍 q(x) 忽略掉真实的某个模式，但不允许制造虚假模式

Reverse KL 会只画一座山，而忽略掉另一座

Forward KL vs Reverse KL

Forward KL： $\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx$

- ✓ 希望模型覆盖所有真实模式

  例如：语言模型、扩散模型、概率密度估计

- ✓ 希望生成的样本“多样性”更高

  Forward KL 不喜欢把样本集中在一点，它鼓励模型覆盖整个真实分布

- ✓ 数据相对复杂、分布有多个峰时

  Forward KL 宁愿“多给一点”概率，也不愿漏掉

  想要“全覆盖、多样性、高召回率” → Forward KL 更好

Forward KL vs Reverse KL

Reverse KL：$\mathrm{KL}(q\|p) = \int q(x) \log \frac{q(x)}{p(x)} dx$

- ✓ 希望模型生成“典型样本”，而不是“全部覆盖”

  Reserve KL挑选最显著的模式，而忽略小的模式，例如Normalizing flow in LQCD

- ✓ 不能容忍“虚假样本”

  例如某些强化学习场景、风险敏感决策

- ✓ 模型必须精准，而不是全面

  避免生成不存在的危险状态

  想要“典型、稳妥、集中、高精确率”→ Reverse KL 更好

能否兼顾Forward & Reverse KL

KL(p‖q) + KL(q‖p) 行不行？

① 两者的惩罚方向本质相反，会互相抵消、打架

如果线性相加会同时受到要“覆盖所有模式”的压力（Forward）和“不能多覆盖”的压力，最终模型往往倾向于“夹在中间的奇怪状态”，两边不讨好

② 会过度惩罚p=0和q=0的区域，造成数值不稳定

两者相加会让模型几乎“无路可走”，非常难训练

③ 从信息论上看，把Forward KL和Reverse KL相加没有明确意义

最优编码长度类似于熵：$H(p) = \mathbb{E}_p[-\log p(x)]$

Forward KL：用 q 去编码真实来自 p 的数据时，额外要付出的平均编码长度

用错误的 q 去编码，会多付出 $\mathbb{E}_p[-\log q(x)] - \mathbb{E}_p[-\log p(x)]$

能否兼顾Forward & Reverse KL

- 替代方案：JS散度（Jensen-Shannon散度）

$$JS(p,q) = \frac{1}{2}KL\left(p\middle\|\frac{p+q}{2}\right) + \frac{1}{2}KL\left(q\middle\|\frac{p+q}{2}\right)$$

  - 因为KL是非负的，则JS必然也是非负的
  - JS永不发散

- 但JS散度实现起来会困难得多

  - KL可以写成 $KL(p\|q) = -\mathbb{E}_p[\log q(x)] + \text{const}$

    可以直接对模型参数求梯度，且训练非常稳定（最大似然估计就是这么来的）

  - Reverse KL可以利用恒等式 $\nabla_\theta \mathbb{E}_{x\sim q_\theta}[f(x)] = \mathbb{E}_{x\sim q_\theta}[f(x)\nabla_\theta \log q_\theta(x)]$

  - JS定义为 $JS(p,q) = \frac{1}{2}KL(p\|m) + \frac{1}{2}KL(q\|m), \quad m = \frac{p+q}{2}$

    里面出现了混合分布 m，无法直接对模型参数求梯度

Diffusion Model

随机量子化

- 随机量子化（Stochastic Quantization）

Parisi和吴咏时1981年提出，后成为格点量子场论的重要工具

量子场论中很多路径积分无法直接求解，或者求解过程很复杂

通过引入一个虚构时间（Langevin 时间），让量子场在这个虚构时间上做布朗运动，最终达到平衡分布，该平衡分布恰好就是量子场论的路径积分权重

随机量子化

- 原本d维度时空外，再引入虚构时间τ $\phi(x) \longrightarrow \phi(x,\tau)$
- 满足Langevin方程 $\dfrac{\partial\phi(x,\tau)}{\partial\tau} = -\dfrac{\delta S[\phi]}{\delta\phi(x,\tau)} + \eta(x,\tau)$

漂移项：沿作用量负梯度下降，趋于最小作用量

噪声项：保证遍历所有场构型，满足 $\langle\eta(x,\tau)\eta(x',\tau')\rangle = 2\delta(x-x')\delta(\tau-\tau')$

- 当$\tau \to \infty$，Langevin 动力学达到平衡分布 $P[\phi] \propto e^{-S[\phi]}$
- 关键直觉：量子涨落由噪声产生，作用量驱动系统向“量子平衡”收敛

这就是随机量子化等价于传统路径积分的原因

Fokker–Planck方程

- Langevin 随机微分方程（SDE）定义了一种 Markov 随机过程

  ➡ 其概率密度满足相应的 Fokker–Planck方程

- 从最简单的一维 Langevin 说起：先看一个普通变量 $x(\tau)$

$$\frac{dx}{d\tau} = K(x) + \eta(\tau), \qquad \langle \eta(\tau)\eta(\tau') \rangle = 2\delta(\tau - \tau')$$

把 Langevin 方程在一个小步长 $\Delta\tau$上积分

$$x(\tau + \Delta\tau) - x(\tau) = \int_{\tau}^{\tau+\Delta\tau} d\tau'\, K(x(\tau')) + \int_{\tau}^{\tau+\Delta\tau} d\tau'\, \eta(\tau')$$

定义

$$\boxed{\sqrt{2\Delta\tau}\,\xi \equiv \int_{\tau}^{\tau+\Delta\tau} d\tau'\, \eta(\tau')}$$

其统计性质

$$\left\langle \left(\int_{\tau}^{\tau+\Delta\tau} \eta(\tau') d\tau' \right)^2 \right\rangle = \int_{\tau}^{\tau+\Delta\tau} d\tau' \int_{\tau}^{\tau+\Delta\tau} d\tau''\, 2\delta(\tau' - \tau'') = 2\Delta\tau$$

因此 $\langle \xi \rangle = 0, \qquad \langle \xi^2 \rangle = 1$ ξ是标准正态分布变量

Fokker–Planck方程

- 考虑一个很小的时间步$\Delta\tau$ $x(\tau+\Delta\tau)=x(\tau)+K(x)\Delta\tau+\sqrt{2\Delta\tau}\,\xi$

  其中 $\xi\sim\mathcal{N}(0,1)$ （一般定义 $\xi\sim\mathcal{N}(\mu,\sigma^2)$，意思是$\xi$服从均值为$\mu$，误差为$\sigma^2$的正态分布）

  极限情况 $\boxed{\lim_{\sigma\to 0}\mathcal{N}(\mu,\sigma^2)=\delta(x-\mu)}$

- 我们关心的不是单条轨迹，而是：$P(x,\tau)\equiv$ 在时间τ取值为x的概率密度
  - 假如不引入白噪声，则 $x(\tau+\Delta\tau)=x(\tau)+K(x)\Delta\tau$

    概率密度满足 $P(x',\tau+\Delta\tau)=\int dx\;\delta(x'-x-K\Delta\tau)P(x,\tau)$

  - 引入白噪声的情况下，则有

$$P(x',\tau+\Delta\tau)=\int dx\;\underbrace{\mathcal{N}\left(x'-x-K\Delta\tau,\;2\Delta\tau\right)}_{\text{转移概率}}P(x,\tau)$$

Fokker–Planck方程

- 等价于概率论里的Chapman-Kolmogorov公式

  概率密度在小步长后的分布 $P(x',\tau+\Delta\tau)=\int dx\,P(x',\tau+\Delta\tau\,|\,x,\tau)\,P(x,\tau)$

  这里$P(A|B)$表示“在B已发生的条件下，A发生的概率”

$$P(x',\tau+\Delta\tau\,|\,x,\tau)=\frac{1}{\sqrt{4\pi\Delta\tau}}\exp\left[-\frac{(x'-x-K(x)\Delta\tau)^2}{4\Delta\tau}\right]$$

  做变量替换 $y=x'-x$

$$P(x',\tau+\Delta\tau)=\int_{-\infty}^{\infty}dy\,\frac{1}{\sqrt{4\pi\Delta\tau}}\exp\left[-\frac{(y-K(x'-y)\Delta\tau)^2}{4\Delta\tau}\right]P(x'-y,\tau)$$

- 小$\Delta\tau$展开策略：
  - $y\sim O(\sqrt{\Delta\tau})$，因为噪声项为$\sqrt{2\Delta\tau}\xi$
  - 指数只保留到$O(\Delta\tau)$ $\quad(y-K(x'-y)\Delta\tau)^2\approx y^2-2yK(x'-y)\Delta\tau+O(\Delta\tau^2)$

Fokker–Planck方程

- 利用y高斯积分求平均

$$\int dy \frac{1}{\sqrt{4\pi\Delta\tau}} e^{-\left(y-K(x'-y)\Delta\tau\right)^2/(4\Delta\tau)} P(x'-y) = \left\langle P(x'-y)\left(1+\frac{yK(x'-y)}{2}\right)\right\rangle_y$$

- 对于上面的平均，有　$\langle 1\rangle_y = 1$　$\langle y\rangle_y = 0$　$\langle y^2\rangle_y = 2\Delta\tau$
- 对于小y，分别对$P(x'-y)$和$K(x'-y)$做Talyor展开

$$P(x'-y,\tau) \approx P(x',\tau) - y\partial_x P(x',\tau) + \frac{y^2}{2}\partial_x^2 P(x',\tau) + \dots$$

$$K(x'-y) \approx K(x') - yK'(x') + \dots$$

- 整理得到　$P(x',\tau) - \Delta\tau\partial_x(K(x')P(x',\tau)) + \Delta\tau\partial_x^2 P(x',\tau)$

Fokker–Planck方程

- 整理$O(\Delta\tau)$项 $$\frac{P(x',\tau+\Delta\tau)-P(x',\tau)}{\Delta\tau}=-\partial_x\big(K(x')P(x',\tau)\big)+\partial_x^2P(x',\tau)$$

- 取$\Delta\tau\to 0$极限 $$\boxed{\frac{\partial P(x,\tau)}{\partial\tau}=-\frac{\partial}{\partial x}\big(K(x)P(x,\tau)\big)+\frac{\partial^2}{\partial x^2}P(x,\tau)}$$

  此即为一维Fokker-Planck方程

- 推广到场变量，Langevin方程变为 $$\frac{\partial\phi(x,\tau)}{\partial\tau}=-\frac{\delta S}{\delta\phi(x,\tau)}+\eta(x,\tau)$$

  则得到场论版Fokker-Planck方程

$$\boxed{\frac{\partial P[\phi,\tau]}{\partial\tau}=\int d^dx\,\frac{\delta}{\delta\phi(x)}\left[\frac{\delta}{\delta\phi(x)}+\frac{\delta S}{\delta\phi(x)}\right]P[\phi,\tau]}$$

Langevin时间演化规律

- 有一个重要的问题：

  当$\tau \to \infty$时，有没有一个稳定平衡态$P_{eq}[\phi]$，使得$\partial_\tau P_{eq} = 0$

- 物理直觉：这是“热浴+力”的平衡

  把随机量子化类比为
  - ϕ：粒子坐标
  - $S[\phi]$：粒子势能
  - η：温度$T = 1$的热浴
  - τ：真实时间

  则Langevin动力学的长期行为必然是玻尔兹曼分布 $P \sim e^{-S/T} \quad (T = 1)$

- 验证：把 $P_{eq}[\phi] \propto e^{-S[\phi]}$代入Fokker-Planck方程

$$\frac{\delta}{\delta\phi}\left(\frac{\delta}{\delta\phi}+\frac{\delta S}{\delta\phi}\right)e^{-S}=\frac{\delta}{\delta\phi}\left(-\frac{\delta S}{\delta\phi}+\frac{\delta S}{\delta\phi}\right)e^{-S}=0 \quad \Rightarrow \quad \boxed{P[\phi,\tau\to\infty] \longrightarrow \frac{1}{Z}e^{-S[\phi]}}$$

Diffusion Model

随机量子化是真实的动力学微分方程演化

Diffusion Model是学习出来的概率生成器

如何对接？

随机量子化

- 解一个随机微分方程（SDE）
- 依赖$\delta S/\delta\phi$
- 每一步都有物理意义

Diffusion Model

- 学习一个反向生成过程
- 只需要样本
- 中间步骤只是计算工具

➢ Diffusion Model的核心思想

Forward过程（人为定义的）：从真实分布 $p_0(\phi)$ 出发，不断加噪声 $\phi_t = \sqrt{\alpha_t}\phi_0 + \sqrt{1-\alpha_t}\,\xi$

Reverse过程（机器学习的关键）：理论上存在一个反向SDE $d\phi = \left[-\nabla_\phi \log p_t(\phi)\right]dt + d\tilde{W}$

Diffusion Model 用神经网络来学习score函数 $s_\theta(\phi,t) \approx \nabla_\phi \log p_t(\phi)$ 然后用它从高斯噪声生成样本

Diffusion Model

- 核心点：Diffusion Model ≈ 学习到“有效Langevin力”

  对一个概率密度$p(x)$，score函数定义为概率密度对变量的对数梯度

$$\boxed{\text{score}(x) \equiv \nabla_x \log p(x)}$$

这个梯度指向概率更大的方向

- 在随机量子化里 $p(\phi) \propto e^{-S(\phi)}$

$$\log p(\phi) = -S(\phi) + \text{const}$$

$$\boxed{\nabla_\phi \log p(\phi) = -\nabla_\phi S(\phi)}$$

score函数等于“负的作用量梯度”，这正是Langevin方程中的力

- 回顾Langevin SDE

$$d\phi = -\nabla S(\phi)\, d\tau + \sqrt{2}\, dW_\tau$$

$$d\phi = \underbrace{\nabla \log p(\phi)}_{\text{score}}\, d\tau + \sqrt{2}\, dW_\tau$$

Langevin 就是沿着 score 方向漂移，再加噪声

Diffusion Model

- 一个极简单的score 函数

$$p(x) = \mathcal{N}(0,1)$$

$$\log p(x) = -\frac{x^2}{2} + \text{const}$$

$$\boxed{\text{score}(x) = \frac{d}{dx}\log p(x) = -x}$$

- $x > 0$，往左拉
- $x < 0$，往右拉

} 全部往概率峰值$x = 0$处收敛

- score 函数是“概率分布在构型空间中的力场”
  - 在随机量子化中，它是$-\delta S/\delta\phi$
  - 在Diffusion Model中，它是被神经网络学习出来的

Diffusion Model

- 为什么score函数定义成$\log p(x)$的梯度？

  我们知道权重函数$p(x) \propto e^{-S(x)}$，但不知道它在高维构型空间中的几何结构，所以这不是一个直接可用的概率分布

$$p(x) = \frac{1}{Z} e^{-S(x)}, \qquad Z = \int \mathcal{D}x \, e^{-S(x)}$$

- 在实际计算当中，配分函数Z不可得：高维、非线性强耦合、拓扑扇区分离

$$\nabla \log p(x) = -\nabla S(x)$$

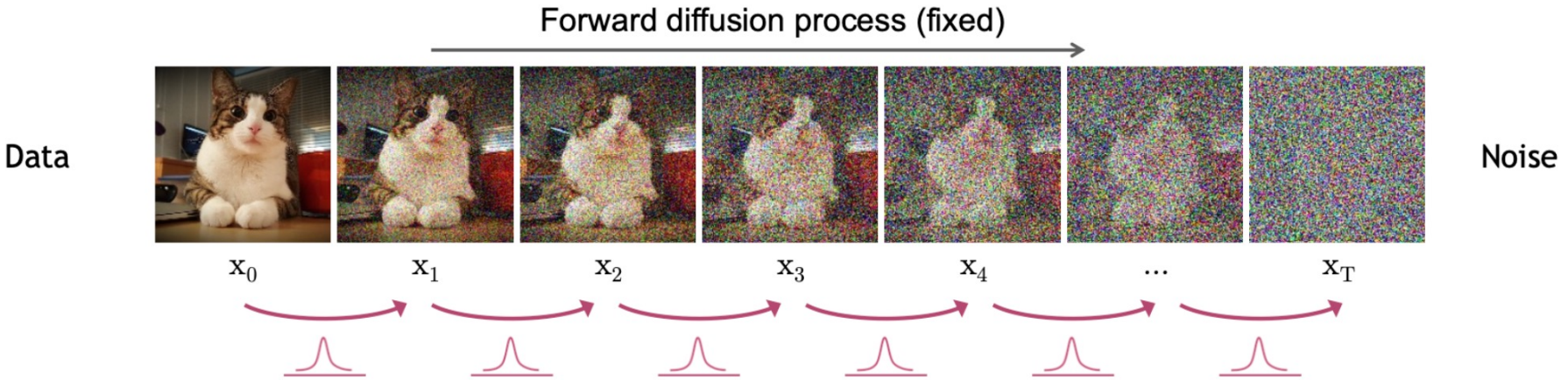
- Langevin 本质上还是在局部推进，马尔科夫链演化，一步只需要当前的∇S

  即使可行，但代价是：自关联时间长、隧穿慢、临界减速

Diffusion Model

- Diffusion Model中，如果获得score函数？

  Forward diffusion：人为加噪声，$p_t(x)$变得越来越平



  Reverse diffusion：生成样本，需要解 $dx = \underbrace{\nabla_x \log p_t(x)}_{\text{score}}\, dt + d\tilde{W}_t$

  但问题是：$p_t(x)$不知道 ➡ 于是用神经网络学习一个

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

Diffusion Model

- 在$t=0$处，如果作用量S可算，目标分布处 $p_0(x)$ 和score 函数已知
- 但在$t \neq 0$处，Forward diffusion重新定义了分布

$$x_t = \sqrt{\alpha_t}\, x_0 + \sqrt{1-\alpha_t}\, \xi, \quad \xi \sim \mathcal{N}(0, I)$$

- 于是

$$p_t(x) = \int dx_0\, p_0(x_0)\, \mathcal{N}(x \mid \sqrt{\alpha_t} x_0, (1-\alpha_t) I)$$

这是一个卷积后的分布，几乎不可能写出$p_t(x)$的解析形式

$p_t(x)$可以看成是一个被“高斯噪声模糊过”的路径积分的概率密度

- score 函数很难计算 $\nabla \log p_t(x) = \dfrac{\nabla p_t(x)}{p_t(x)}$ $p_t(x) = \int dx_0\, p_0(x_0)\, \mathcal{N}(x \mid \sqrt{\alpha_t} x_0, (1-\alpha_t) I)$

分子、分母都是高维积分，每个x要重做一次，比直接采样还难！

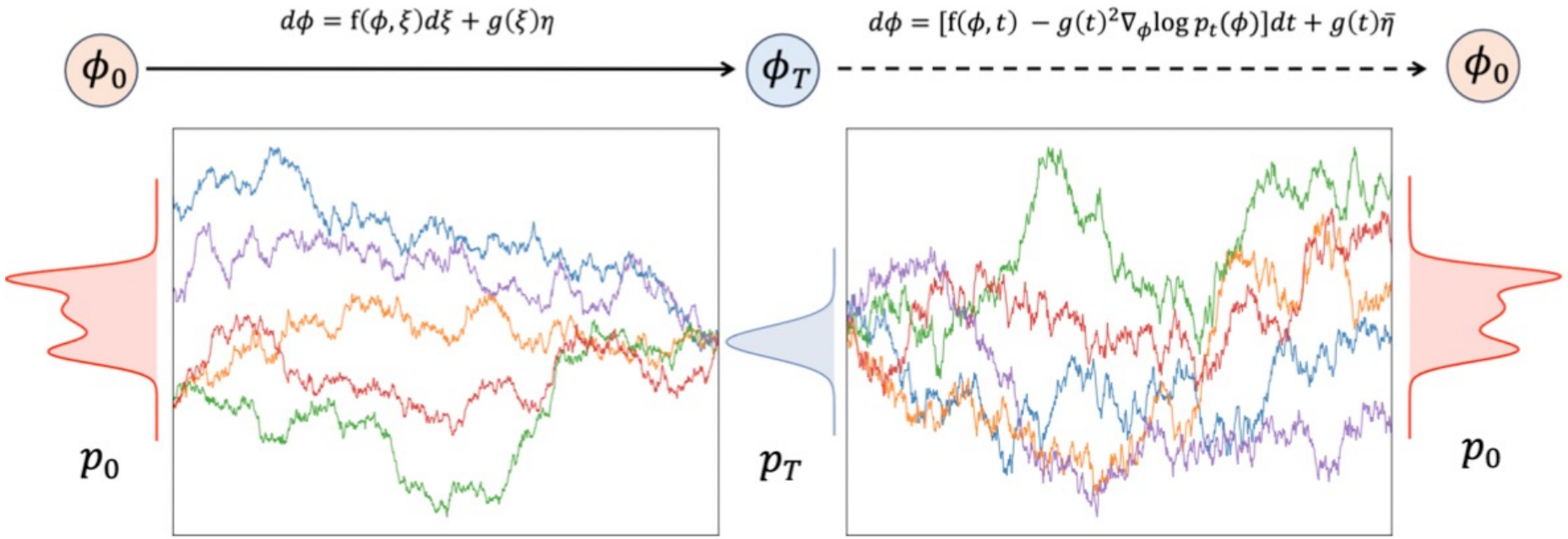
所以需要学习score函数！

正向和反向生成扩散

- Forward Diffusion SDE $\frac{d\phi}{d\tau} = f(\phi,\tau) + g(\tau)\eta(\tau)$

漂移项　　扩散项

- Reverse Generative Diffusion SDE $\frac{d\phi}{d\tau} = \left[f(\phi,\tau) - g^2(\tau)\nabla_\phi \log p_\tau(\phi)\right] + g(\tau)\bar{\eta}(\tau)$



漂移项变得复杂，特别是里面包含了score函数

Anderson定理

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

$$\frac{\partial p}{\partial t} = -\nabla \cdot (\mathbf{f}p) + \frac{1}{2}g^2\Delta p$$

Fokker-Planck Equation

$$d\bar{t} = -t$$

Reverse Time!

$$d\mathbf{x} = \tilde{\mathbf{f}}(\mathbf{x}, t)d\bar{t} + g(t)d\bar{\mathbf{w}}$$

$$-\frac{\partial p}{\partial t} = -\nabla \cdot (\tilde{\mathbf{f}}p) + \frac{1}{2}g^2\Delta p$$

$$\frac{\partial p}{\partial t} = \nabla \cdot (\tilde{\mathbf{f}}p) - \frac{1}{2}g^2\Delta p$$

Reverse Fokker-Planck Equation

Anderson定理

$$\frac{\partial p}{\partial t} = -\nabla \cdot (\mathbf{f}p) + \frac{1}{2}g^2 \Delta p \qquad \frac{\partial p}{\partial t} = \nabla \cdot (\tilde{\mathbf{f}}p) - \frac{1}{2}g^2 \Delta p$$

$$\underbrace{-\nabla \cdot (\mathbf{f}p) + \frac{1}{2}g^2 \Delta p}_{Forward} = \underbrace{\nabla \cdot (\tilde{\mathbf{f}}p) - \frac{1}{2}g^2 \Delta p}_{Backward}$$

$$-\nabla \cdot (\mathbf{f}p) - \nabla \cdot (\tilde{\mathbf{f}}p) = -\frac{1}{2}g^2 \Delta p - \frac{1}{2}g^2 \Delta p$$

$$\nabla \cdot [(\mathbf{f} + \tilde{\mathbf{f}})p] = g^2 \Delta p \qquad \Delta p = \nabla \cdot (\nabla p) = \nabla \cdot (p \nabla \log p)$$

$$\nabla \cdot [(\mathbf{f} + \tilde{\mathbf{f}})p] = \nabla \cdot [g^2 p \nabla \log p]$$

$$\tilde{\mathbf{f}} = -\mathbf{f} + g^2 \nabla \log p$$

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

从Noise出发生成目标分布

$$\frac{d\phi}{dt} = \left[f(\phi, t) - g^2(t)\mathbf{s}_{\hat{\theta}}(\phi, t)\right] + g(t)\bar{\eta}(t)$$

- 简化 $\frac{d\phi}{dt} = -g(t)^2 \nabla_\phi \log p_t(\phi) + g(t)\bar{\eta}$
- 定义 $\tau \equiv T - t (d\tau \equiv -dt)$ $\frac{d\phi}{d\tau} = g_\tau^2 \nabla_\phi \log q_\tau(\phi) + g_\tau \bar{\eta}$

$$\phi(\tau_{n+1}) = \phi(\tau_n) + g_\tau^2 \nabla_\phi \log q_{\tau_n}[\phi(\tau_n)]\Delta\tau + g_\tau \sqrt{\Delta\tau}\bar{\eta}(\tau_n)$$

- Fokker Planck方程

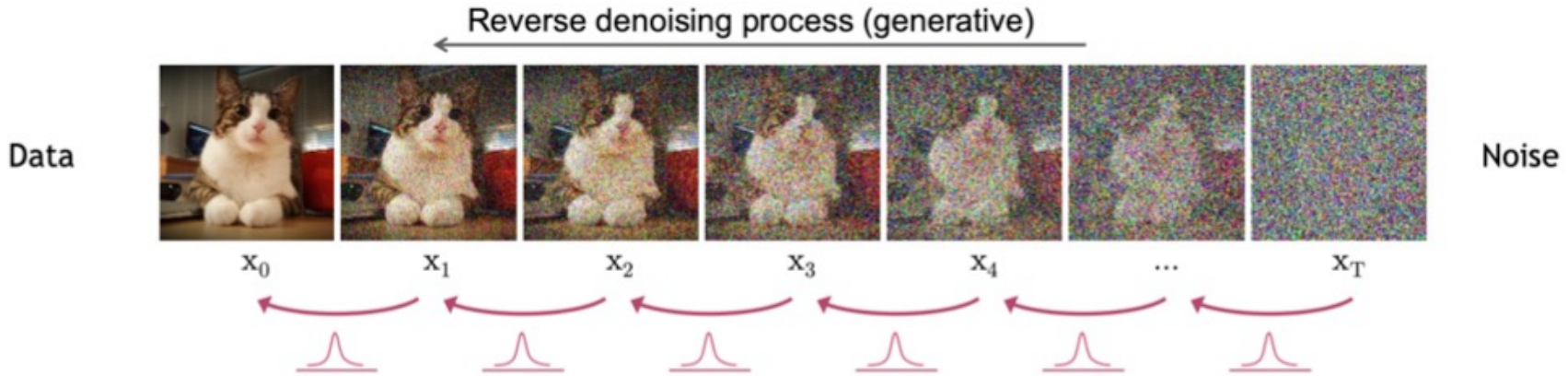
Noise scale: $\langle \bar{\eta}^2 \rangle \equiv 2\bar{\alpha}$, **time scale:** $g_\tau^2 \Delta\tau$

$$\frac{\partial p_\tau(\phi)}{\partial \tau} = \int d^n x \left\{ g_\tau^2 \bar{\alpha} \frac{\delta}{\delta \phi} \left(\frac{\delta}{\delta \phi} + \frac{1}{\bar{\alpha}} \nabla_\phi S_{\mathbf{DM}} \right) \right\} p_\tau(\phi) \qquad \nabla_\phi S_{\mathbf{DM}} \equiv -\nabla_\phi \log q_\tau(\phi)$$

- $\tau \to T$

$$p_{eq}(\phi) \propto e^{-\frac{S_{DM}}{\bar{\alpha}}} \qquad p_{\tau=T}(\phi) \to P[\phi, T]$$

从Noise出发生成目标分布



- 在已知“正向加噪扩散过程”的前提下，通过学习每个中间时间的 score函数，构造一个“从纯噪声逐步去噪、最终生成真实样本”的反向随机过程。

  不是倒着算Forward，而是新的、依赖概率分布的反向SDE

  - 因为Forward diffusion是个无条件、无记忆的过程
  - 倒着走，在单条轨迹层面不可能，但反向扩散存在于“概率分布层面”
  - score 提供了“概率力场”，在每个噪声尺度上，把构型往高概率区域拉回去

从Noise出发生成目标分布

- score函数 学的不是$S(x)$，而是在“典型构型”附近，$\nabla \log p(x)$长什么样
  - Langevin / HMC：有一个完整的势能函数，每一步靠本地坡度走
  - Diffusion / score：不重建整张地图，只在“常被访问的区域”，学一张高速公路网路，快速访问高概率区

- Diffusion Model 学习的是近似、全局的力 $\nabla S(x)$ vs $s_\theta(x)$

  $s_\theta(x) \approx \mathbb{E}[\nabla \log p(x) \mid x \text{ 在典型区域}]$ 在物理上等价于积掉高频自由度后的有效力

  不同时间下的有效作用量在改变，类似于粗粒化重整化群流

- 最后，必须做 Metropolis hastings！

Score matching

- 引入Fisher散度
  - Forward KL 散度 $\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)}\, dx$
  - Fisher 散度 $D_F(p\|q) = \frac{1}{2} \int p(x) \left|\nabla_x \log p(x) - \nabla_x \log q(x)\right|^2 dx$

    Forward型、但比较的是log密度的梯度

- 最小化Fisher散度

Score matching

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

Minimizing the Fisher divergence

此处$s_\theta(x)$应为$\log q_\theta(x)$

1-D

$$\frac{1}{2}\mathbb{E}_{p_{\mathbf{data}}}[(\nabla_x \log p_{\mathbf{data}}(x) - \nabla_x s_\theta(x))^2]$$

$$= \frac{1}{2}\int p_{\mathbf{data}}(x)(\nabla_x \log p_{\mathbf{data}}(x) - \nabla_x s_\theta(x))^2 dx$$

$$= \underbrace{\frac{1}{2}\int p_{\mathbf{data}}(x)(\nabla_x \log p_{\mathbf{data}}(x))^2 dx}_{\textbf{const}} + \frac{1}{2}\int p_{\mathbf{data}}(x)(\nabla_x s_\theta(x))^2 dx$$

$$- \int p_{\mathbf{data}}(x)\nabla_x s_\theta(x)\nabla_x \log p_{\mathbf{data}}(x) dx$$

$$= \mathbb{E}_{p_{\mathbf{data}}}[\nabla_x^2 s_\theta(x)] + \frac{1}{2}\mathbb{E}_{p_{\mathbf{data}}}[(\nabla_x s_\theta(x))^2] + \textbf{const}$$

$$- \int p_{\mathbf{data}}(x)\nabla_x s_\theta(x)\nabla_x \log p_{\mathbf{data}}(x) dx$$

$$= -\int \nabla_x s_\theta(x)\nabla_x p_{\mathbf{data}}(x) dx$$

$$= -p_{\mathbf{data}}(x)\nabla_x s_\theta(x)\Big|_{-\infty}^{\infty} + \int p_{\mathbf{data}}(x)\nabla_x^2 s_\theta(x) dx$$

$$= \mathbb{E}_{p_{\mathbf{data}}}[\nabla_x^2 s_\theta(x)],$$

Integration by parts

Multi-Variable $\rightarrow \mathbb{E}_{p_{\mathbf{data}}}\left[\mathrm{tr}(\nabla_{\mathbf{x}}^2 s_\theta(\mathbf{x})) + \frac{1}{2}\|\nabla_{\mathbf{x}} s_\theta(\mathbf{x})\|_2^2\right] + \textbf{const}$

Score matching

$$\min_\theta D_F(p\|q_\theta) \iff \min_\theta \mathbb{E}_{x\sim p}\left[\frac{1}{2}|\nabla_x \log q_\theta(x)|^2 + \nabla_x^2 \log q_\theta(x)\right]$$

- 对于大多数神经网络，一阶梯度项可以用反向传播直接算，成本是和前向传播同量级
  - 把神经网络比作“工厂流水线”
    1. 原料 x（输入数据）放在传送带上
    2. 每一层 $f_1, f_2, \ldots, f_L$ 是流水线工序（比如加工、检测、包装）
    3. 最终得到产品 $\hat{y} = f_L \circ \cdots \circ f_1(x)$
  - 前向传播：原料加工成成品 $x \xrightarrow{f_1} h_1 \xrightarrow{f_2} h_2 \cdots \xrightarrow{f_L} \hat{y}$
    每一层的输出都被记录下来（类似加工记录），目的是计算损失函数
  - 反向传播：发现损失函数大，信息回流，改进工序 $\frac{\partial \mathcal{L}}{\partial W_L}, \frac{\partial \mathcal{L}}{\partial W_{L-1}}, \ldots, \frac{\partial \mathcal{L}}{\partial W_1}$
    - W_L：神经网络第 L 层的权重（参数）
    - $\frac{\partial \mathcal{L}}{\partial W_L}$：**损失对第 L 层权重的变化率**

Score matching

$$\min_\theta D_F(p\|q_\theta) \quad \Longleftrightarrow \quad \min_\theta \mathbb{E}_{x\sim p}\left[\frac{1}{2}|\nabla_x \log q_\theta(x)|^2 + \nabla_x^2 \log q_\theta(x)\right]$$

- 二阶梯度项，出现了Hessian矩阵的trace $\nabla_x^2 \log q_\theta(x) = \mathrm{tr}(\text{Hessian of log q})$
  - x维度高时（如场论），成本很大
  - Hessian矩阵是个大型矩阵，trace访问对角元
  - 自动微分的反向传播天然是 输出对中间变量和参数的一阶梯度
  - Hessian 元素是 $\dfrac{\partial^2 f}{\partial x_i \partial x_j}$
  - 自动微分框架没有“只算对角线”，前向+反向传播计算梯度的过程会涉及到所有节点

Score matching

- 解决方案：加噪声
  - 对数据加高斯噪声 $\tilde{x} = x + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2 I),\ \ p(\tilde{x}|x) = \mathcal{N}(\tilde{x}; x, \sigma^2 I)$
  - 得到平滑分布 $p_\sigma(\tilde{x}) = \int dx\, p(x)\mathcal{N}(\tilde{x}; x, \sigma^2 I)$
  - 考虑 加噪版本的 Fisher 散度 $D_F^{(\sigma)}(p\|q_\theta) = \frac{1}{2}\int d\tilde{x}\, p_\sigma(\tilde{x}) \left|\nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) - \nabla_{\tilde{x}} \log q_\theta(\tilde{x})\right|^2$
  - 关键恒等式 $\nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) = \frac{\int dx\, p(x)\, \mathcal{N}(\tilde{x}; x, \sigma^2)\, \frac{x-\tilde{x}}{\sigma^2}}{\int dx\, p(x)\, \mathcal{N}(\tilde{x}; x, \sigma^2)}$

    在给定$\tilde{x}$的条件下，score 指向“最可能生成$\tilde{x}$的干净样本x”的方向

    尽管 $D_F(p\|q_\theta) = D_F^{(\sigma)}(p\|q_\theta)$ ❌ **数值上不等**

    如果模型足够强，那么 $\arg\min_\theta D_F(p\|q_\theta) = \arg\min_\theta D_F^{(\sigma)}(p\|q_\theta)$

Score matching

- 原始Fisher vs 加噪Fisher 散度

  想象一条高维概率“山谷”

  - 原始Fisher散度要求在每个点，精确对齐“坡度方向”

    但山谷太尖、太嘈杂→二阶项炸掉

  - 加噪Fisher散度先用高斯噪声把山谷抹平，再对齐平滑后的坡度

    山谷中心线不变，但把小尺度结构平均掉了

- 定义Explicit score matching (ESM)

$$L_{ESM}(\theta) = \mathbb{E}_{p(\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log p(\tilde{x})\|^2\right]$$

定义Denoising score matching (DSM)

$$L_{DSM}(\theta) = \mathbb{E}_{p(x,\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log p(\tilde{x}\,|\,x)\|^2\right]$$

定理：$\nabla_\theta L_{DSM}(\theta) = \nabla_\theta L_{ESM}(\theta)$

Score matching

$$L_{DSM}(\theta) = \mathbb{E}_{p(x,\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log p(\tilde{x}\,|\,x)\|^2\right]$$

$\mathbb{E}_{p(x,\tilde{x})}[\cdot]$ = 对 **联合分布** $p(x,\tilde{x})$ 的期望

可以展开为积分形式：$\int dx\, d\tilde{x}\, p(x,\tilde{x})(\cdot)$

在 DSM 中，x 来自真实数据，$\tilde{x}$ 来自加噪声

$$L_{ESM}(\theta) = \mathbb{E}_{p(\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log p(\tilde{x})\|^2\right]$$

$$= \mathbb{E}_{p(\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x})\|^2 - s_\theta(\tilde{x})^T\nabla_{\tilde{x}}\log p(\tilde{x})\right] + C_1$$

$$L_{DSM}(\theta) = \mathbb{E}_{p(x,\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log p(\tilde{x}\,|\,x)\|^2\right]$$

$$= \mathbb{E}_{p(x,\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x})\|^2 - s_\theta(\tilde{x})^T\nabla_{\tilde{x}}\log p(\tilde{x}\,|\,x)\right] + C_2$$

只要证明第二项相等即可！

Score matching

$$Term_2 = \mathbb{E}_{p(\tilde{x})}\left[s_\theta(\tilde{x})^T \nabla_{\tilde{x}} \log p(\tilde{x})\right]$$

$$Term_2 = \mathbb{E}_{p(x,\tilde{x})}\left[s_\theta(\tilde{x})^T \nabla_{\tilde{x}} \log p(\tilde{x}\,|\,x)\right]$$

$$= \mathbb{E}_{p(\tilde{x})}\left[\mathbb{E}_{p(x|\tilde{x})}\left[s_\theta(\tilde{x})^T \nabla_{\tilde{x}} \log p(\tilde{x}\,|\,x)\right]\right]$$

$$= \mathbb{E}_{p(\tilde{x})}\left[s_\theta(\tilde{x})^T \mathbb{E}_{p(x|\tilde{x})}\left[\nabla_{\tilde{x}} \log p(\tilde{x}\,|\,x)\right]\right]$$

只要证明 $\nabla_{\tilde{x}} \log p(\tilde{x}) = \mathbb{E}_{p(x|\tilde{x})}\left[\nabla_{\tilde{x}} \log p(\tilde{x}\,|\,x)\right]$ 即可！

Score matching

$$\nabla_{\tilde{x}} \log p(\tilde{x}) = \mathbb{E}_{p(x|\tilde{x})} \left[\nabla_{\tilde{x}} \log p(\tilde{x}|x) \right]$$

$$\nabla_{\tilde{x}} \log p(\tilde{x}) = \frac{\nabla_{\tilde{x}} p(\tilde{x})}{p(\tilde{x})} = \frac{\nabla_{\tilde{x}} \int p(\tilde{x}|x) p(x) dx}{p(\tilde{x})} = \frac{\int \nabla_{\tilde{x}} p(\tilde{x}|x) p(x) dx}{p(\tilde{x})} \qquad \nabla p = p \cdot \nabla \log p$$

$$= \frac{\int p(\tilde{x}|x) \nabla_{\tilde{x}} \log p(\tilde{x}|x) p(x) dx}{p(\tilde{x})} = \int \underbrace{\frac{p(\tilde{x}|x) p(x)}{p(\tilde{x})}}_{p(x|\tilde{x})} \nabla_{\tilde{x}} \log p(\tilde{x}|x) dx$$

$$= \mathbb{E}_{p(x|\tilde{x})} \left[\nabla_{\tilde{x}} \log p(\tilde{x}|x) \right]$$

Q.E.D

Score matching

$$\mathcal{N}(\tilde{x}; x, \sigma^2 I)$$

$$L_{DSM}(\theta) = \mathbb{E}_{p(x,\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p(\tilde{x}\,|\,x)\|^2\right]$$

$$\int dx\, d\tilde{x}\, p(x,\tilde{x}) f(x,\tilde{x}) = \int dx\, p(x) \int d\tilde{x}\;\, p(\tilde{x}|x)\, f(x,\tilde{x})$$

$$\int dx\, p(x\,) f(x\,) = \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad x_i \sim p(x)$$

Diffusion Model与传统采样的分界

- 样本≠可操作的生成机制

  Diffusion model 学的不是“已有样本”，而是一个可泛化、可条件化、可加速的生成动力学

- 样本只是被动数据

  不能重新加权、快速生成独立样本、改变概率分布参数

- 样本≠知道概率分布

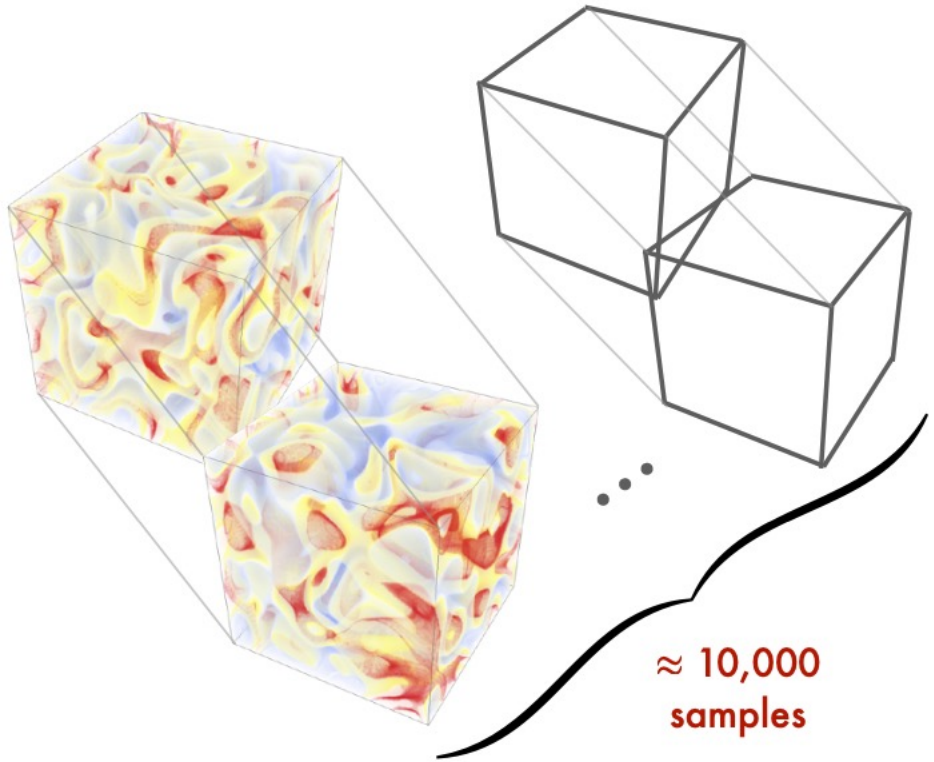
  不知道概率分布的解析形式，而Diffusion Model学习如何从噪声中生成数据

- Diffusion ≠ 复刻已有样本
  - 从噪声生成无限新样本
  - 用作 proposal distribution
  - 生成样本之间相关性可控
  - 替代或加速蒙特卡洛

Normalizing Flow

反向 KL：流映射采样

Quantum field generation



256 x 256 x 256 x 512	Lattice geometry
x 4 x 8	SU(3) link variables

≈ 100,000,000,000 dof

Image generation

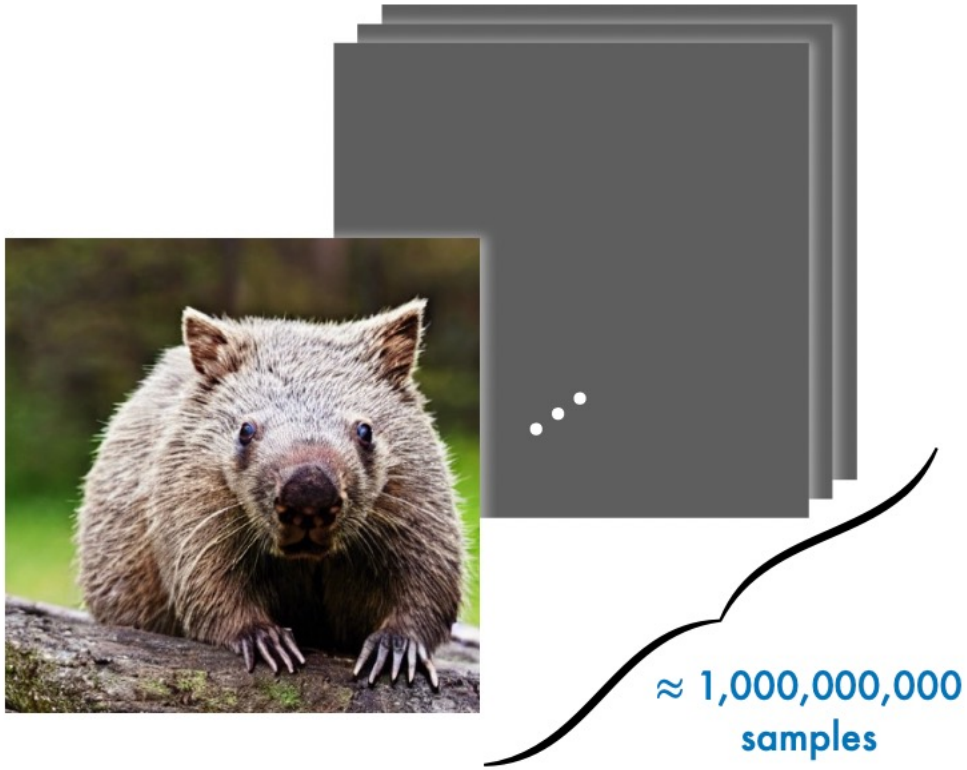


Image geometry	512 x 512
RGB pixel variables	x 3

≈ 1,000,000 dof

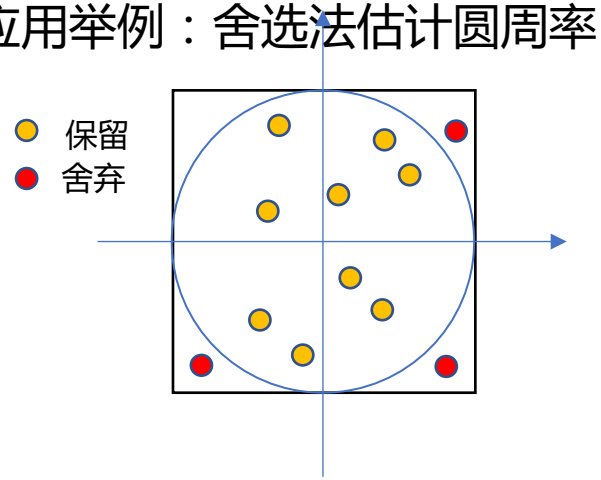
反向 KL：流映射采样

- 利用正则化流映射(normalizing flow)采样
- 流映射采样的初步体验——Box Muller 方法

任务：产生按f(x)分布的随机数

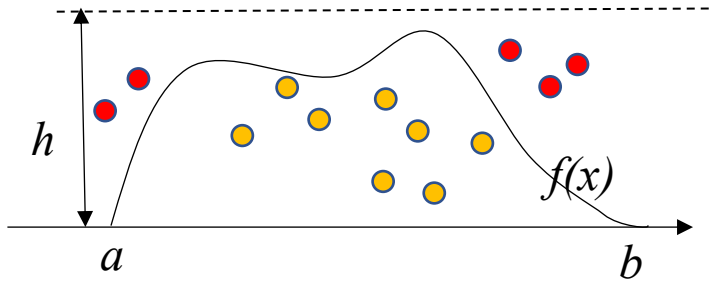
- 舍选法——简单直观

应用举例：舍选法估计圆周率

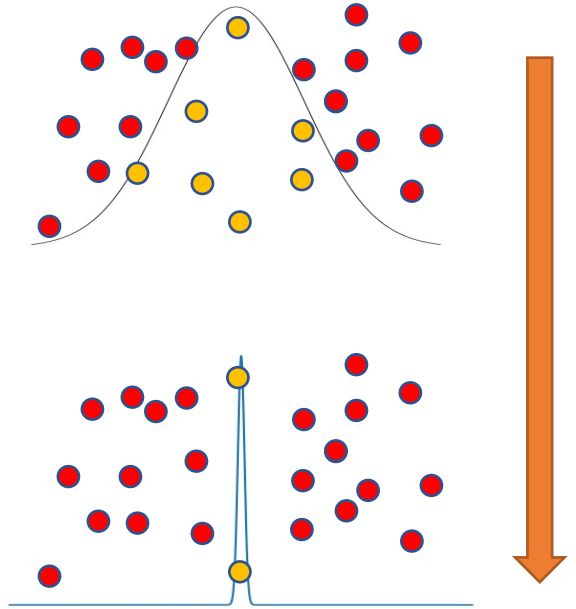


$$P \approx \frac{\pi}{4}$$

在区域$X \in (a, b), Y \in (0, h)$内多次均匀撒点，若$(X, Y)$落在曲线$f(x)$以下，保留$X$作为一个抽样值，则生成的样本按照密度函数 $f(x)$分布



对于正态分布，
若分布的峰很窄，
舍选的效率将比较低。



流映射采样

- Box Muller 方法
  - 回忆 Gaussian 积分中的技巧：$$\int e^{-\frac{x_1^2+x_2^2}{2}} dx_1 dx_2 = \int e^{-\frac{r^2}{2}} r dr d\theta = 2\pi.$$
  - 做变量替换 $$\int \frac{1}{2\pi} e^{-\frac{x_1^2+x_2^2}{2}} dx_1 dx_2 = \int d\xi_1 d\xi_2$$

$$\begin{cases} \xi_1 = e^{-(x_1^2+x_2^2)/2} \\ \xi_2 = \frac{1}{2\pi}\tan^{-1}\frac{x_2}{x_1} \end{cases} \Rightarrow \begin{cases} x_1 = \sqrt{-2\ln\xi_1}\cos(2\pi\xi_2) \\ x_2 = \sqrt{-2\ln\xi_1}\sin(2\pi\xi_2) \end{cases}$$

定义了流映射 f

(ξ_1, ξ_2)

↓

(x_1, x_2)

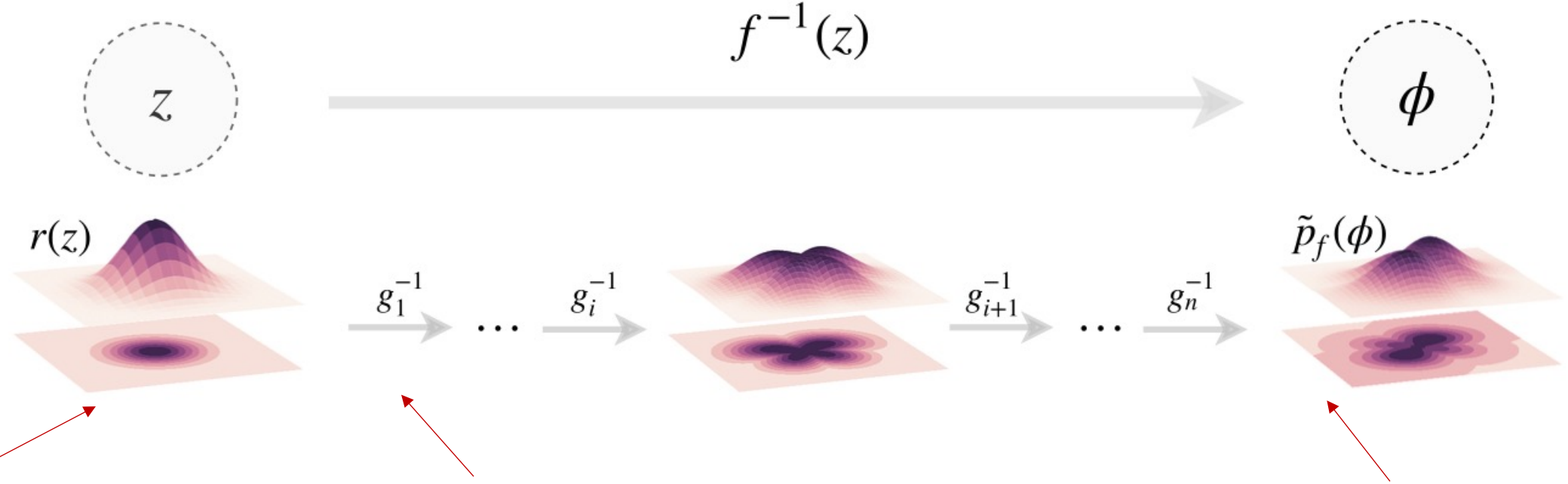
  - 如果 $\xi_1, \xi_2 \sim U(0,1) \quad \Rightarrow \quad x_1, x_2 \sim N(0,1)$

    x_1 与 x_2 服从均值为0，方差为1的高斯分布；效率高，计算过程简单
  - 总结：(x_1, x_2) 的几率密度为 $q(x_1, x_2)$，(ξ_1, ξ_2) 的几率密度为 $r(\xi_1, \xi_2) = 1$

$$q(x_1, x_2) = r(\xi_1, \xi_2)\left|\det J_f\right|^{-1} = \frac{1}{2\pi} e^{-\frac{x_1^2+x_2^2}{2}}$$

流映射采样

M. S. Albergo, G. Kanwar, and P. E. Shanahan Phys. Rev. D 100 no. 3, (2019) 034515



先验分布
(比如正态分布)

- 神经网络位于每个耦合层中，负责对输入数据进行缩放和偏移，以逐步调整分布
- 通过叠加多个耦合层构造流模型

$$f(\phi) = g_1(g_2(\dots g_n(\phi)\dots))$$

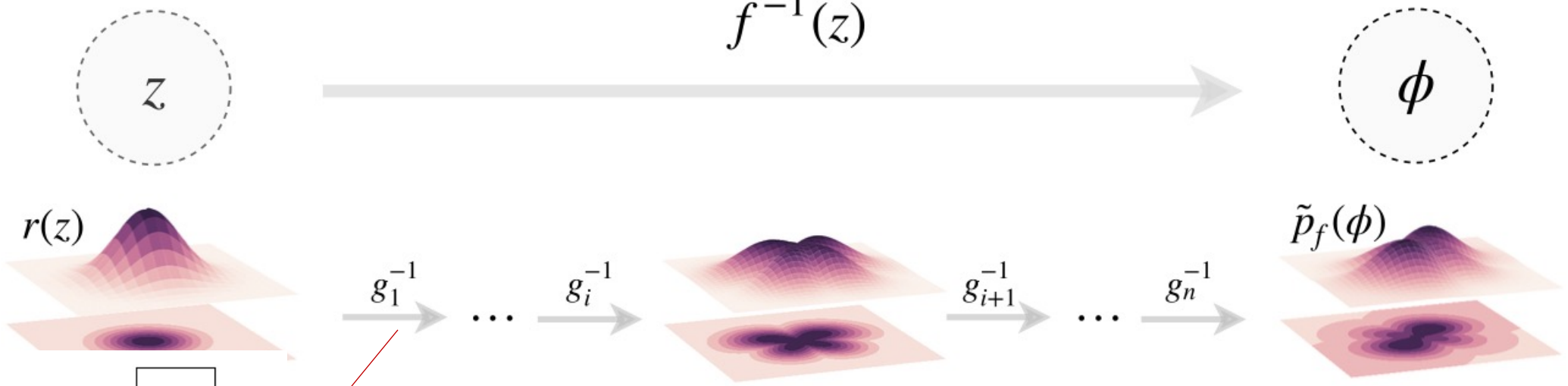
- 雅可比行列式可通过链式法则得到

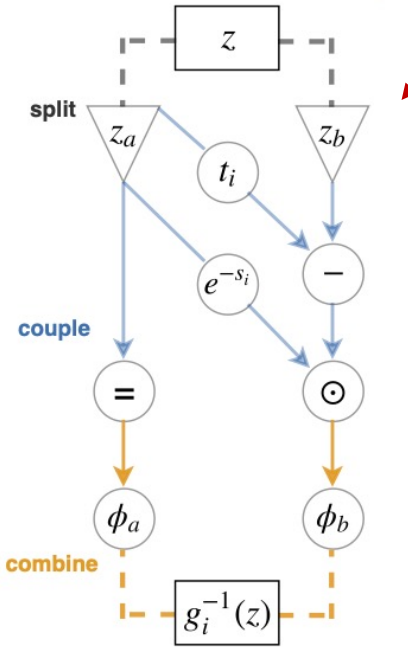
$$\det(\text{Jacobian of } f) = \prod_{i=1}^{n} \det(\text{Jacobian of } g_i)$$

目标是调整神经网络的参数，使得流模型生成的分布 $\tilde{p}_f(\phi)$ 尽可能逼近目标分布 $p(\phi)$

$p(\phi)$ 通常较复杂，难以直接生成

流映射采样





$$g_i(\phi) := \begin{cases} z_a = \phi_a \\ z_b = \phi_b \odot e^{s_i(\phi_a)} + t_i(\phi_a) \end{cases}$$

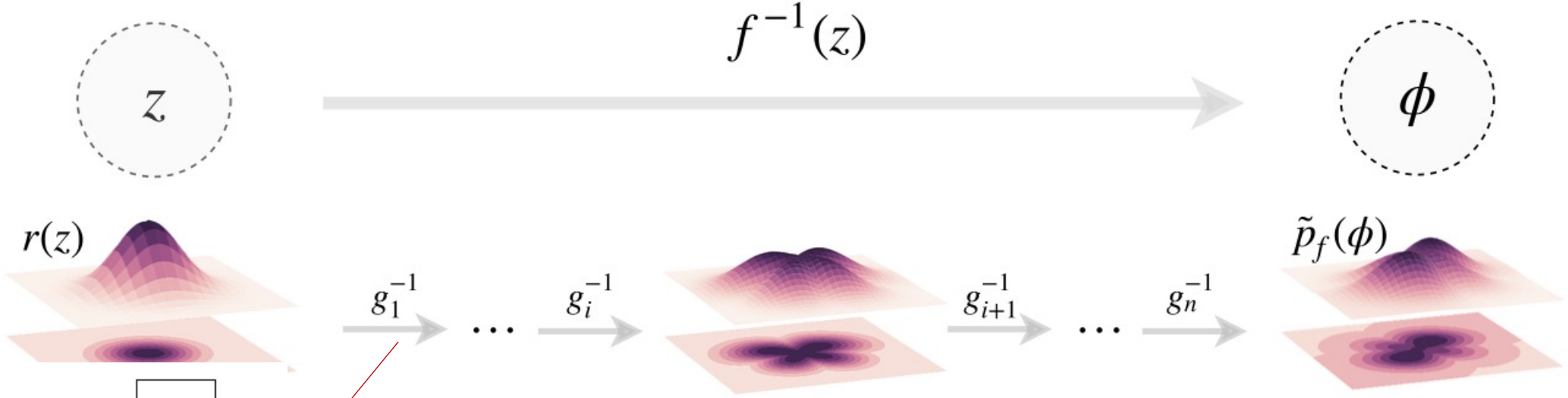
$$g_i^{-1}(z) := \begin{cases} \phi_a = z_a \\ \phi_b = (z_b - t_i(z_a)) \odot e^{-s_i(z_a)} \end{cases}$$

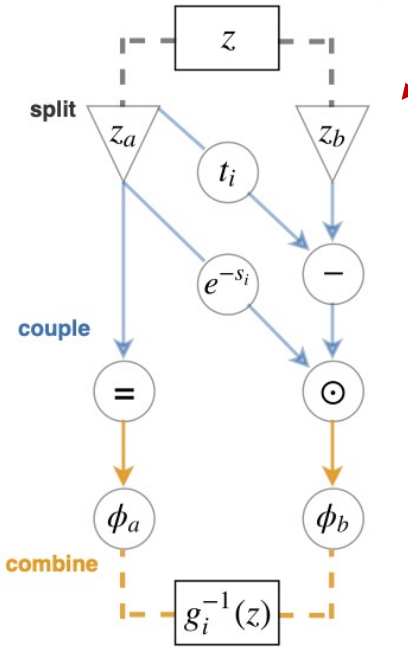
s_i 和 t_i 是神经网络，进行变换操作

$\odot$ 表示元素逐位相乘

变换 g_i 被设计成一种特殊形式，即使 s_i 和 t_i 是复杂的神经网络，整个变换 g_i 依然是可逆的

流映射采样





$$g_i(\phi) := \begin{cases} z_a = \phi_a \\ z_b = \phi_b \odot e^{s_i(\phi_a)} + t_i(\phi_a) \end{cases}$$

$$g_i^{-1}(z) := \begin{cases} \phi_a = z_a \\ \phi_b = (z_b - t_i(z_a)) \odot e^{-s_i(z_a)} \end{cases}$$

雅可比矩阵是个下三角，行列式很容易计算

$$\left| \det \frac{\partial g_i(\phi)}{\partial \phi} \right| = \prod_{j=1}^{D/2} e^{[s_i(\phi_a)]_j}$$

- 可逆性对于流映射至关重要，因为需要根据变换后的样本计算其概率密度
- 如果每一层的变换是显式可逆的，就可以快速地进行采样和密度计算，而不会引入额外的计算负担。

流映射采样

- 目标分布 $p(\phi) = \frac{e^{-S(\phi)}}{Z}$
- 训练方法：最小化损失函数

使用的损失函数是目标分布 $p(\phi)$ 与生成分布 $\tilde{p}_f(\phi)$ 之间的偏移Kullback-Leibler散度

$$
\begin{aligned}
L(\tilde{p}_f) &:= D_{KL}(\tilde{p}_f||p) - \log Z \\
&= \int \prod_j d\phi_j \, \tilde{p}_f(\phi) \left(\log \tilde{p}_f(\phi) - \log p(\phi) - \log Z\right) \\
&= \int \prod_j d\phi_j \, \tilde{p}_f(\phi) \left(\log \tilde{p}_f(\phi) + S(\phi)\right).
\end{aligned}
$$

- KL散度是一种衡量两个概率分布之间差异的指标
- 通过最小化 KL 散度，可以逐步优化神经网络，使得生成分布趋近于目标分布
- 实际应用中，损失值通过随机抽样来估算

  从流映射生成的分布 $\tilde{p}_f(\phi)$ 中抽取一批样本 $\{\phi^i\}$，然后通过计算样本均值来得到损失的近似值

流映射采样

➢ 自训练特性

- 流映射允许从自身分布 $\tilde{p}_f$ 中高效采样，因此训练过程可以完全基于模型生成的样本进行
- 无需依赖目标分布$p(\phi)$的现有样本，这在计算目标分布样本代价高昂时尤其有用

➢ 具体算法流程

① 训练流生成模型以使得输出分布 $\tilde{p}_f(\phi)$ 接近目标分布 $p(\phi)$，损失函数使用方向 KL 散度

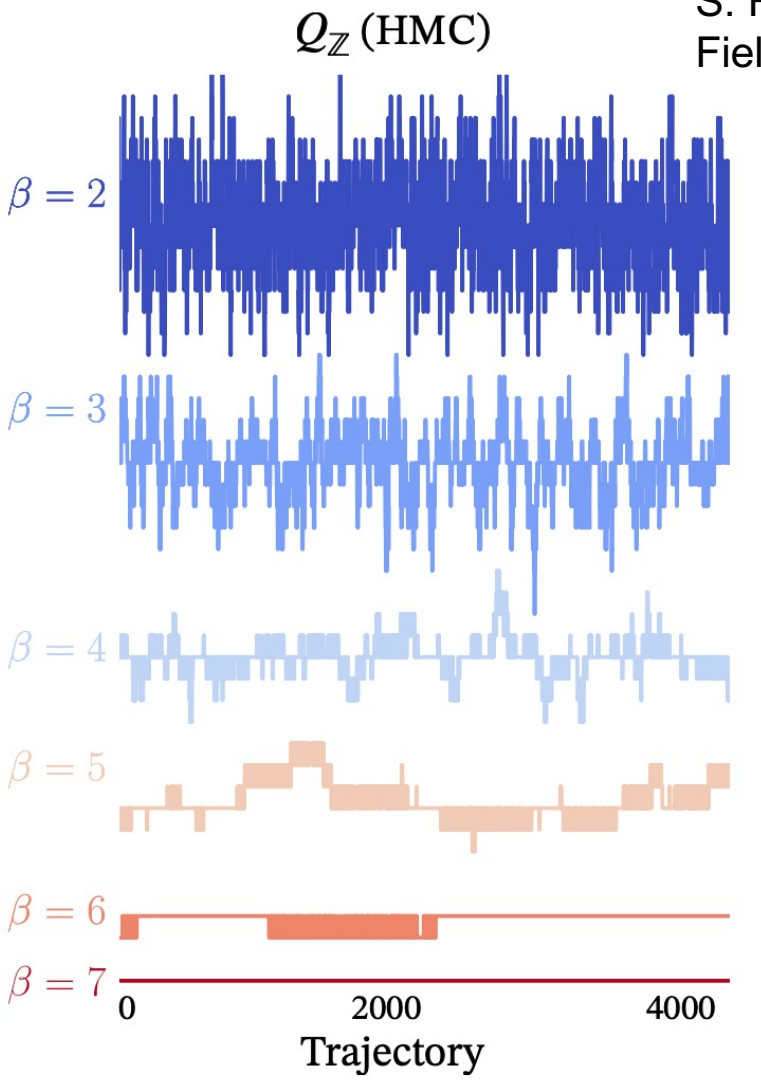
② 从生成模型中采样 N个样本 $\phi'(i) \sim \tilde{p}_f$ (可并行采样)，并计算每个样本的作用量 $S(\phi')$

③ 使用 Metropolis算法，依次接受或拒绝这些提案样本，构建长度为 N的马尔科夫链

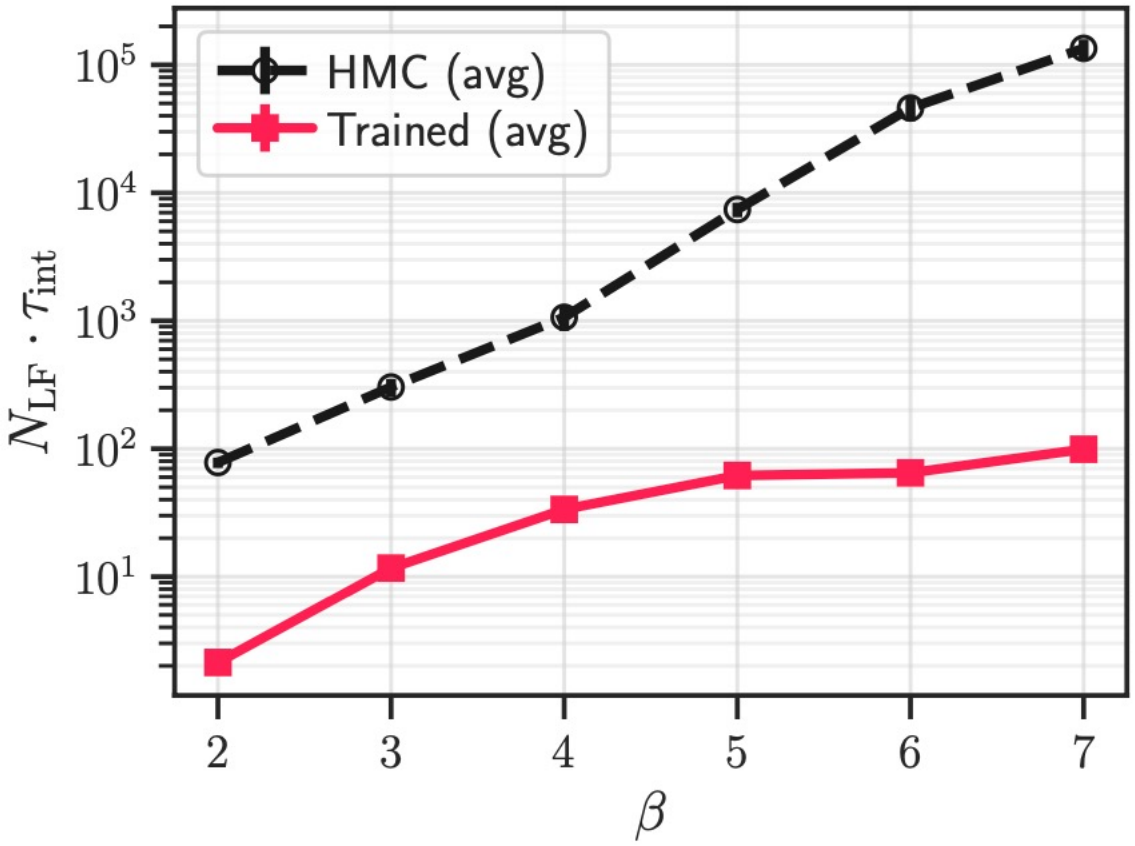
➢ 理论保证

- 分布的正定性：如果先验分布 r(z)严格为正，并且流映射 f 可逆且连续，则生成分布 $\tilde{p}_f(\phi)$ 保证严格为正
- 各态历经：对于所有具有有限作用量的模型，由此生成的马尔科夫链被证明具有遍历性

流映射采样



S. Foreman, X.-Y. Jin, and J. C. Osborn in 38th International Symposium on Lattice Field Theory. 12, 2021. arXiv:2112.01582



- 传统蒙特卡洛模拟，β越大，格距越小，拓扑荷冻结
- 2维U(1)规范场，传统马尔科夫过程与机器学习后的马尔科夫过程对比

流映射采样

- 小结：基于流的马尔科夫链蒙特卡罗方法具有几项优势

① 可系统降低自相关时间

- 通过对模型进行训练，可以逐步减少马尔科夫链中的自相关时间，从而提高采样效率

② 计算简化

- 马尔科夫链的每一步更新仅需要执行一次模型评估和一次作用量计算，计算过程简洁高效

③ 提议独立性和并行化

- 每次更新的提议样本彼此独立，因此可以并行生成提议样本，从而高效地构建马科可夫链

④ 无需先验样本

- 模型的训练过程完全依赖于模型自身生成的样本，而不需要从目标分布中预先获得样本

流映射采样

- 应用
  - 标量场

M. S. Albergo, G. Kanwar, and P. E. Shanahan Phys. Rev. D 100 no. 3, (2019) 034515

D. C. Hackett, C.-C. Hsieh, M. S. Albergo, et al, arXiv:2107.00734

L. Del Debbio, J. M. Rossney, and M. Wilson Phys. Rev. D 104 no. 9, (2021) 094507

M. Caselle, E. Cellini, A. Nada, and M. Panero arXiv:2201.08862

A. G. D. G. Matthews, M. Arbel, D. J. Rezende, and A. Doucet arXiv:2201.13117

  - U(1)、SU(N)规范理论

G. Kanwar, M. S. Albergo, D. Boyda, et al, Phys. Rev. Lett. 125 no. 12, (2020) 121601

D. Boyda, G. Kanwar, S. Racanière, et al, Phys. Rev. D 103 no. 7, (2021) 074504

S. Foreman, X.-Y. Jin, and J. C. Osborn in 9th International Conference on Learning Representations. 5, 2021. arXiv:2105.03418

S. Foreman, X.-Y. Jin, and J. C. Osborn in 38th International Symposium on Lattice Field Theory. 12, 2021. arXiv:2112.01582

S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya in 38th International Symposium on Lattice Field Theory. 12, 2021. arXiv:2112.01586

流映射采样

- 应用
  - 复作用量

    S. Lawrence and Y. Yamauchi Phys. Rev. D 103 no. 11, (2021) 114509

    M. Rodekamp, E. Berkowitz, C. Gäntgen, et al, arXiv:2203.00390
  - 态密度方法

    M. S. Albergo, D. Boyda, D. C. Hackett, G. Kanwar, K. Cranmer, S. Racanière, D. J. Rezende, and P. E. Shanahan arXiv:2101.08176
- 其他机器学习方法的应用
  - 限制玻尔兹曼机
  - 自回归网络
  - 自学习蒙特卡罗方法
  - 对抗学习

格点QCD领域人工智能的发展潜力

- 新兴的人工智能技术 为 LQCD领域带来新的机遇
  - 提升效率：通过人工智能改进算法，可以更有效地利用现有计算资源
  - 创新性框架：人工智能有潜力提出全新的 LQCD 形式化方法，解决传统方法无法处理的问题
- 格点QCD工作流程的三个阶段及人工智能的应用
  - 组态生成——新的采样算法

    需要对高维分布进行高效采样，同时保持与目标分布的精确一致性
  - 可观测量测量——新可观测量的设计、可观测量的高效近似计算方法

    在有限的样本数下提高统计精度，并开发能捕捉复杂物理信息的新方法
  - 分析过程——处理病态反问题的新方法

    需要在复杂数据中提取准确的物理结果，同时对误差保持严格控制

格点QCD与人工智能的计算需求相似性

- 计算需求
  - 人工智能和传统 LQCD 方法都计算密集，且主要的计算开销都集中在数值线性代数上，线性代数可以通过 GPU 实现高效并行化
- 硬件需求差异
  - 当前 LQCD的计算需要比人工智能更紧密连接的节点，但随着非常大模型和快速通信需求的增加，这一差异正在缩小
- 硬件的交叉适用性
  - 计算需求上的相似性意味着，LQCD 研究可能会促进人工专用硬件在传统 LQCD 计算中的应用，推动计算科学领域的整体进展

    比如google设计的TPU (Tensor Processing Unit) 和Graphcore设计的IPU (Intelligence Processing Unit)

格点QCD应用人工智能的挑战

- 组态生成

  需要从由格点作用量定义的玻尔兹曼分布中精确地抽样，或者至少抽样要提供足够的统计信息，以完全修正任何精确性偏差，这对人工智能方法和架构提出了严格的约束

- 观测量测量

  训练数据的模型依赖可能会对渐近性产生额外影响，机器学习生成的观测量必须谨慎地进行特性描述和解释

- 分析过程

  与典型的机器学习应用类似，人工智能的大多数应用会引入模型依赖，必须作为系统误差的来源来控制

- 并行方案

  典型的机器学习应用处理的数据点大小通常在几千字节，LQCD处理的数据至少是几GB到几百GB，甚至是TB量级，应用机器学习方法时需要比常规情况更高程度的并行性