arXiv:2302.01170v2 [stat.ML] 1 Dec 2023

Timewarp：通过学习时间粗粒化动力学加速分子动力学，并实现可迁移性

Leon Klein[*†]
柏林自由大学
leon.klein@fu-berlin.de

Andrew Y.K. Foong[*]微软研究院 AI4Science
andrewfoong@microsoft.com

Tor Erlend Fjelde[*†]剑桥大学 tef30@cam.ac.uk

Bruno Mlodozeniec[*†]剑桥大学 bkm28@cam.ac.uk

Marc Brockschmidt[†]

Sebastian Nowozin †

Frank Noé
微软研究院 AI4Science 柏林自由大学 莱斯大学
franknoe@microsoft.com

Ryota Tomioka
微软 ft 研究 h AI4S
科学 ryoto@microsoft.com

摘要

分子动力学（MD）模拟是一种广泛使用的分子系统模拟技术，最常见的是在全原子分辨率下进行模拟，并以飞秒量级的时间步长（1fs $= 10^{-15}$s）积分运动方程。分子动力学常用于计算平衡性质，这需要从诸如玻尔兹曼分布这样的平衡分布中进行采样。然而，结合和折叠等许多重要过程发生在毫秒甚至更长的时间尺度上，无法通过常规分子动力学高效采样。此外，每研究一个分子系统，都需要进行新的分子动力学模拟。我们提出了*Timewarp*，这是一种增强采样方法，它在以玻尔兹曼分布为目标的马尔可夫链蒙特卡罗方法中，将归一化流用作提议分布。该流在分子动力学轨迹上离线训练，并学习进行大跨度的时间推进，从而模拟时长为$10^5 - 10^6$ 飞秒的分子动力学。至关重要的是，Timewarp 可在不同分子系统之间迁移：我们展示了训练完成后，该方法能够泛化到未见过的小肽（2–4 个氨基酸），并在全原子分辨率下探索其亚稳态；与标准分子动力学相比，它还能加速采样的实际运行时间。我们的方法是迈向通用、可迁移的分子动力学加速算法的重要一步。

1 引言

分子动力学（MD）是模拟原子尺度物理系统的成熟技术。准确执行时，它无需湿实验，便能让我们无与伦比地深入了解分子运动的详细机制。分子动力学模拟已被用于理解分子生物物理学中备受关注的过程，例如蛋白质折叠 [1, 2],蛋白质-配体结合 [3],以及蛋白质-蛋白质缔合 [4]。分子动力学的许多重要应用归根结底都是为了高效地从玻尔兹曼分布中采样，即温度为

[*]贡献相同。
[†]工作完成于微软研究院任职期间。

第 37 届神经信息处理系统大会（NeurIPS 2023）。

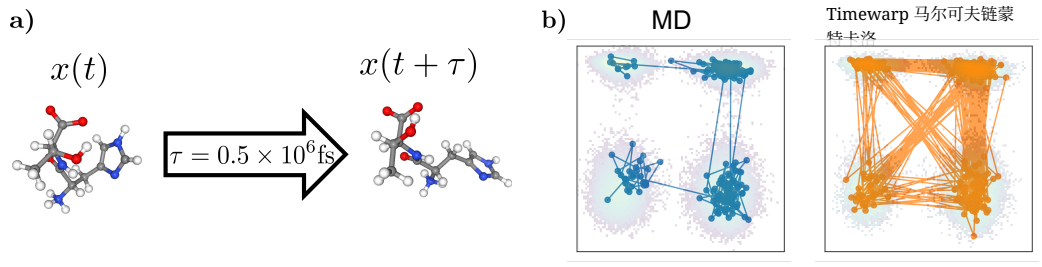

图 1：（a）使用 Timewarp 对二肽 HT（训练期间未见过）进行采样得到的初始状态 $x(t)$（左）和接受的提议状态 $x(t+\tau) \sim p_\theta(x(t+\tau)|x(t))$（右）。（b）模拟轨迹的 TICA投影，展示了亚稳态之间的跃迁；分别为短时分子动力学模拟（左）和 Timewarp 马尔可夫链蒙特卡洛（右），两者的实际运行时间均为 30 分钟。就每秒有效样本量而言，Timewarp 马尔可夫链蒙特卡洛相较于分子动力学实现了 ≈ 33 倍的加速。

下分子系统的平衡分布 T。设 $(x^p(t)\ x^v(t)) := x(t) \in \mathbb{R}^{6N}$ 为分子在时间 t的状态，该状态由笛卡尔坐标系下 N 个原子的位置 $x^p(t) \in \mathbb{R}^{3N}$ 和速度 $x^v(t) \in \mathbb{R}^{3N}$ 组成。玻尔兹曼分布定义如下：

$$\mu(x^p, x^v) \propto \exp\left(-\tfrac{1}{k_B T}(U(x^p) + K(x^v))\right), \quad \mu(x^p) = \int \mu(x^p, x^v)\, \mathrm{d}x^v. \tag{1}$$

其中，$U(x^p)$ 是势能，$K(x^v)$ 是动能，k_B 是玻尔兹曼常数。许多重要的量，例如蛋白质折叠和蛋白质-配体结合的自由能，都可以作为 $\mu(x^p)$ 下的期望值来计算。从 $\mu(x^p)$ 中采样的一种常用分子动力学方法是朗之万动力学 [5]，，它遵循以下随机微分方程（SDE）：

$$m_i \mathrm{d}x_i^v = -\nabla_i U \mathrm{d}t - \gamma m_i x_i^v \mathrm{d}t + \sqrt{2 m_i \gamma k_B T}\mathrm{d}B_t. \tag{2}$$

这里，i 表示原子的索引，m_i 是原子 i的质量，$U(x^p)$是势能，γ 是摩擦参数，$\mathrm{d}B_t$ 是标准布朗运动过程。从初始状态 $x(0)$出发，模拟方程（2），并结合关系式$\mathrm{d}x^p = x^v \mathrm{d}t$，即可得到 $x(t)$ 的取值；当 $t \to \infty$时，这些取值服从玻尔兹曼分布。标准分子动力学库会以时间步长 Δt对该随机微分方程进行离散化。为保证稳定性，时间步长必须选为 $\sim 1\text{fs} = 10^{-15}\text{s}$。不幸的是，许多生物分子都包含被能垒隔开的亚稳态，跨越这些能垒可能需要毫秒级的分子动力学仿真时间（$\sim 10^{12}$ 个连续积分步），使得这种方法难以实施。为解决这一问题，先前的研究提出了多种增强采样方法，例如粗粒化 [6, 7] 和元动力学 [8]。然而，要有效实施这些方法，需要针对每个分子系统具备特定的领域知识。

标准分子动力学模拟不会在不同分子系统之间传递信息：对于每个研究对象，都必须进行新的模拟。这无疑浪费了机会：许多分子系统的动力学高度相似，对一个系统进行模拟应能为类似系统提供相关信息。尤其是，蛋白质由 20 种氨基酸组成的序列构成，是研究这种可迁移性的理想对象。我们提出 *Timewarp*，这是一种通用、可迁移的增强采样方法，它使用归一化流 [9] 作为马尔可夫链蒙特卡罗（MCMC）方法的提议分布，以玻尔兹曼分布为目标。我们的主要贡献如下：

1. 我们提出了首个能够在通用笛卡尔坐标中运行的机器学习算法，并证明其可迁移至训练期间未见过的小型肽系统。2. 我们首次展示了对未见过的肽系统进行玻尔兹曼分布渐近无偏马尔可夫链蒙特卡罗采样时，能够实现实际运行时间加速。3. 我们定义了一种以玻尔兹曼分布为目标的马尔可夫链蒙特卡罗算法，该算法使用条件归一化流作为提议分布，并通过梅特罗波利斯–黑斯廷斯（MH）校正步骤确保细致平衡（第 3.4 节）。4. 我们设计了一种置换等变的、基于 Transformer 的归一化流。5. 我们构建了一个全新的训练数据集，其中包含数千种小肽的分子动力学轨迹。6. 我们表明，即使在不使用 MH 校正的情况下（第3.5节），Timewarp 也能用于探索新肽的亚稳态，速度远超分子动力学。

代码可在此处获取：https://github.com/microsoft/timewarp。数据集可应要求提供[3]。

2 相关工作

近年来，利用深度学习研究分子的热潮不断兴起。玻尔兹曼生成器[10, 11, 12]使用流从玻尔兹曼分布中渐近无偏地采样。生成样本有两种方式：(i) 从流中生成独立同分布样本，并使用统计重加权校正期望值的偏差。(ii) 在马尔可夫链蒙特卡罗框架中使用玻尔兹曼生成器 [13], , 如 Timewarp 中所做的那样。目前，与 Timewarp 不同，玻尔兹曼生成器尚无法在多个分子之间实现泛化。唯一的例外是 [14], , 他们提出了一个扭转空间中的扩散模型，并将其底层常微分方程作为可迁移的玻尔兹曼生成器。然而，与 Timewarp 不同，他们使用内坐标，并不作用于全原子系统。此外， [15, 16]最近提出了一种针对分子的笛卡尔坐标玻尔兹曼生成器，有望实现可迁移训练。最近， [17] 提出了*GeoDiff*，这是一种根据分子图预测分子构象的扩散模型。与 Timewarp 类似，GeoDiff 在笛卡尔坐标中工作，并且能够泛化到未见过的分子。然而，GeoDiff 并未应用于蛋白质，而是应用于小分子，也不以玻尔兹曼分布为目标。与 Timewarp 不同， [18] 针对多个时间分辨率学习转移概率，从而准确捕捉动力学。然而，他们没有展示系统之间的可迁移性。与 Timewarp 最为相似的是， [19] 在近期工作中训练了一个可迁移的机器学习模型，用于模拟聚合物的时间粗粒化动力学。然而，与 Timewarp 不同，他们的模型作用于粗粒化表示。此外，该模型并未应用于蛋白质，也没有 Metropolis–Hastings步骤，这意味着模拟中的误差可能会不断累积而无法得到校正。

马尔可夫状态模型(MSMs) [20, 21, 22] 通过运行许多短时分子动力学模拟来工作，这些模拟用于定义一个离散状态空间，并估计转移概率矩阵。与 Timewarp 类似，MSMs 估计时间 t 的状态与时间 $t+\tau$ 的状态之间的转移概率，其中 $\tau \gg \Delta t$。近期研究将深度学习应用于 MSMs，产生了 *VAMPnets*[23] 和 深度生成式马尔可夫状态模型 [24], , 以深度网络取代 MSM 的数据处理流程。与 Timewarp 不同，这些模型无法迁移，并且是在粗粒化的离散状态空间中对动力学建模，而非使用全原子坐标表示。

此前已有大量关于神经自适应采样器 [25, 26, 27], 的研究，这类采样器使用深度生成模型作为提议分布。*A-NICE-MC* [25] 采用通过无似然对抗方法训练的保体积流。其他方法则使用旨在鼓励探索的目标函数。我们的目标函数中的熵项受到 [28] 的启发。与这些方法不同，Timewarp 致力于在不重新训练的情况下推广 到新的分子系统。

有许多用于分子动力学的增强采样方法，例如并行回火 [29, 30] ，或沿跃迁路径提出集体变量的更新 [8, 31]。Timewarp 能够加速分子动力学，因此通常可以与这些技术结合使用。

3 方法

考虑从 $x(t)$ 出发，对方程（2）进行时长为 $\tau \gg \Delta t$ 的分子动力学模拟所诱导的 $x(t+\tau)$ 分布。我们将此条件分布记为 $\mu(x(t+\tau)|x(t))$。Timewarp 使用深度概率模型近似 $\mu(x(t+\tau)|x(t))$（见图 1）。训练完成后，该模型用于马尔可夫链蒙特卡罗方法，以从玻尔兹曼分布中采样。

3.1 条件归一化流

我们拟合一个条件归一化流， $p_\theta(x(t+\tau)|x(t))$，以 $\mu(x(t+\tau)|x(t))$为目标，其中 θ 是可学习参数。归一化流由一个基分布（通常为标准高斯分布）和一个微分同胚 f定义，即一个具有可微逆映射的可微双射。具体而言，我们将$p_\theta(x(t+\tau)|x(t))$设为由以下生成过程定义的分布的密度：

$$z^p, z^v \sim \mathcal{N}(0, I), \quad x^p(t+\tau), x^v(t+\tau) \coloneqq f_\theta(z^p, z^v; x^p(t), x^v(t)). \tag{3}$$

[3]请联系 ontact andrewfoong@microsoft.com 获取数据集 a ccess。

这里 $z^p \in \mathbb{R}^{3N}$和 $z^v \in \mathbb{R}^{3N}$。对于 θ和 $x(t)$的所有取值， $f_\theta(\cdot; x(t))$都是一个微分同胚，它将潜变量（z^p， z^v） $\in \mathbb{R}^{6N}$映射到（$x^p(t+\tau)$ $x^v(t+\tau)$） $\in \mathbb{R}^{6N}$。条件状态$x(t)$参数化了一个微分同胚族，从而定义了一个条件归一化流 [32]。请注意，从 $x(t)$到输出 $x(t+\tau)$的映射不受可逆性约束，只有从 z 到 $x(t+\tau)$的映射必须可逆。利用变量变换公式，我们可以计算：

$$p_\theta(x(t+\tau)|x(t)) = \mathcal{N}\left(f_\theta^{-1}(x(t+\tau); x(t)); 0, I\right) \left|\det \mathcal{J}_{f_\theta^{-1}(\cdot; x(t))}(x(t+\tau))\right|,$$

其中， $f_\theta^{-1}(\cdot; x(t)): \mathbb{R}^{6N} \to \mathbb{R}^{6N}$是微分同胚 $f_\theta(\cdot; x(t))$ 的逆映射，而$\mathcal{J}_{f_\theta^{-1}(\cdot; x(t))}(x(t+\tau))$ 表示在 $x(t+\tau)$ 处求得的 $f_\theta^{-1}(\cdot; x(t))$ 的雅可比矩阵。

3.2 数据集生成

我们使用 *OpenMM* 库 [33] 对方程（2）进行积分，以生成分子动力学轨迹。我们在隐式水中模拟小型蛋白质（肽），即不显式建模水分子的自由度。具体而言，我们生成一个轨迹数据集 $\mathcal{D} = \{\mathcal{T}_i\}_{i=1}^P$，其中 P 表示肽的数量。每条分子动力学轨迹按 τ 的间隔进行时间下采样，因此 $\mathcal{T}_i = (x(0)\ x(\tau)\ x(2\tau)\ldots)$。训练期间，我们从 $\mathcal{D}$ 中随机采样样本对 $x(t)\ x(t+\tau)$。每对样本都表示来自条件分布 $\mu(x(t+\tau)|x(t))$ 的一个样本。更多细节见附录 E。由于该流是在多个 肽的轨迹数据上训练的，因此我们可以在测试时使用它，对训练数据中未出现过的新 肽进行泛化。

3.3 增广归一化流

我们通常主要关注位置 x^p 的分布，而非速度x^v。因此，公式 (3) 中的 $x^v(t)\ x^v(t+\tau)$ 无须表示原子的实际速度。于是，我们将 x^v 视为增广归一化流框架 [34] 中非物理的辅助变量，从而简化学习问题。对于 $\mathcal{D}$ 中的每个数据点 $x(t) = x^p(t)\ x^v(t)$，我们不再通过记录分子动力学轨迹中的速度来获取 $x^v(t)$，而是丢弃分子动力学速度，并独立地抽取 $x^v(t) \sim \mathcal{N}(0, I)$。由于 $x^v(t)$ 与$x^v(t+\tau)$ 是独立抽取的，辅助变量$x^v(t)$ 不再包含有关未来状态 $x^p(t+\tau)$ $x^v(t+\tau)$ 的信息。因此，我们可以简化 f_θ，使其仅依赖于 $x^p(t)$，其中$x^p(t+\tau)\ x^v(t+\tau) := f_\theta(z^p, z^v; x^p(t))$。我们引入辅助变量有两个原因：第一，它们能提高 x^p 分布的表达能力，而不会大幅增加计算成本 [34, 35]。第二，借助辅助变量，可以更轻松地构建满足置换等变性的条件流——见第 4.1 节。

3.4 使用渐近无偏马尔可夫链蒙特卡罗方法针对玻尔兹曼分布

训练流 $p_\theta(x(t+\tau)|x(t))$ 后，我们将其用作马尔可夫链蒙特卡罗方法中的提议分布，以针对位置 x^p 和辅助变量 x^v的联合分布，该联合分布的密度为：

$$\mu_{\text{aug}}(x^p, x^v) \propto \exp\left(-\tfrac{U(x^p)}{k_B T}\right) \mathcal{N}(x^v; 0, I). \tag{4}$$

从状态 $X_0 = (X_0^p, X_0^v) \in \mathbb{R}^{6N}$（对应状态 $m=0$）开始，我们进行迭代：

$$\tilde{X}_m \sim p_\theta(\cdot|X_m^p), \quad X_{m+1} := \begin{cases} \tilde{X}_m & \text{with probability } \alpha(X_m, \tilde{X}_m) \\ X_m & \text{with probability } 1-\alpha(X_m, \tilde{X}_m) \end{cases} \tag{5}$$

whe ˜重新 $\alpha(X_m, X_m)$是针对*Metropolis-Hastings*（*MH*）接受率 [36]，目标为公式（4）：

$$\alpha(X, \tilde{X}) = \min\left(1, \frac{\mu_{\text{aug}}(\tilde{X}) p_\theta(X \mid \tilde{X}^p)}{\mu_{\text{aug}}(X) p_\theta(\tilde{X} \mid X^p)}\right) \tag{6}$$

用于 p_θ 的流必须支持高效采样和精确的似然评估，这对于快速实现方程 (5) 和 (6) 至关重要。此外，每个 Metropolis–Hastings步骤之后，我们都会使用吉布斯采样更新重新采样辅助变量 X^v：

$$(X_m^p, X_m^v) \leftarrow (X_m^p, \epsilon), \quad \epsilon \sim \mathcal{N}(0, I). \tag{7}$$

迭代这些更新即可得到样本 $X_m^p, X_m^v \sim \mu_{\text{aug}}$，其分布为 $m \to \infty$。要获得位置 $X_m^p \sim \mu$ 的玻尔兹曼分布样本，我们只需丢弃辅助变量 X_m^v。由于发送 $m \to \infty$ 不可行，我们会持续模拟该链，直到达到固定的计算预算。在实践中，我们发现模型的接受率可能较低，约为 1%。不过，我们强调，即使接受率较低，只要提出的变化足够大，我们的模型仍能加快探索速度，正如第 6 节所示。此外，我们引入了一种批量采样流程，可在保持细致平衡的同时显著加快采样。该流程通过一次前向传递对一批提议进行采样，并接受其中第一个满足 MH 接受准则的提议。附录 C 中的算法1给出了马尔可夫链蒙特卡罗算法的伪代码。

3.5 不进行 MH 校正的快速但有偏状态空间探索

我们可以不使用 MH 校正来保证样本渐近无偏，而选择使用 Timewarp 执行一种简单的探索算法。在这种情况下，我们忽略 MH 校正，并接受所有能量变化低于某个截断值的提议。这样可以更快地探索状态空间；在第 6 节中我们将展示，尽管这种方法在技术上有偏，但往往能得到定性上准确的自由能估计。与算法1和标准分子动力学相比，它还能够快几个数量级地发现所有亚稳态，因此可用于为后续的马尔可夫状态模型方法提供初始状态例如。伪代码见附录 D 中的算法 2。

4 模型架构

现在我们将介绍流 $f_\theta(z^p,\ z^v; x^p(t))$ 的架构，如图 2 所示。

RealNVP 耦合流 我们的架构基于 RealNVP [37], ，它由一系列耦合层组成。这些层根据潜变量的其他维度，对其部分维度进行仿射变换。具体而言，我们根据辅助变量变换位置变量，反之亦然。在流的第 ℓ 个耦合层中，实现了以下变换:

$$z^p_{\ell+1} = s^p_{\ell,\theta}(z^v_\ell; x^p(t)) \odot z^p_\ell + t^p_{\ell,\theta}(z^v_\ell; x^p(t)), \tag{8}$$

$$z^v_{\ell+1} = s^v_{\ell,\theta}(z^p_{\ell+1}; x^p(t)) \odot z^v_\ell + t^v_{\ell,\theta}(z^p_{\ell+1}; x^p(t)). \tag{9}$$

接下来，我们省略耦合层索引 ℓ。这里，$\odot$ 是逐元素乘积，而s^p_θ： $\mathbb{R}^{3N} \to \mathbb{R}^{3N}$ 是我们的原子变换器，这是一个基于变换器架构 [38] 的神经网络，它接收辅助潜变量 z^v 和条件状态 $x(t)$，并输出位置潜变量 z^p 的缩放因子。函数 t^p_θ： $\mathbb{R}^{3N} \to \mathbb{R}^{3N}$ 由另一个原子变换器实现，它使用 z^v 和 $x(t)$，输出位置潜变量 z^p的平移。位置变量的仿射变换（见公式（8））与辅助变量的类似仿射变换（见公式（9））交错进行。由于位置的缩放因子和平移因子仅取决于辅助变量，反之亦然，因此该变换的雅可比矩阵为下三角矩阵，从而可以高效计算密度。完整的流 f_θ由 N_{coupling} 个堆叠的耦合层组成，从 $z \sim \mathcal{N}(0,\ I)$开始，以来自 $p_\theta(x(t+\tau)|x(t))$的样本结束。这一点在图 2 左侧有所展示。请注意，这里有一个跳跃连接，因此流的输出预测的是变化 $x(t+\tau) - x(t)$，而不是直接预测 $x(t+\tau)$。

原子变换器 现在我们介绍原子变换器网络。令 $x^p_i(t)$ z^p_i, z^v_i 表示 $\mathbb{R}^3$ 中的所有元素，分别代表条件状态中原子 i 的位置、原子 i 的位置潜变量，以及原子 i 的辅助潜变量。为了实现一个以 z^v 为输入的原子变换器（例如公式（8）中的 $s^p_\theta(z^v, x^p(t))$ 和 $t^p_\theta(z^v, x^p(t))$），我们首先将与原子 i 相关的变量拼接起来。由此得到一个向量 $a_i := [x^p_i(t)\ h_i, z^v_i]^\intercal \in \mathbb{R}^{H+6}$，其中排除了 z^p_i，因为不允许 s^p_θ, t^p_θ 依赖于 z^p。这里，$h_i \in \mathbb{R}^H$ 是一个仅取决于原子类型的学习得到的嵌入向量。向量 a_i 被输入一个多层感知机 ϕ_{in}： $\mathbb{R}^{H+6} \to \mathbb{R}^D$，其中 D 是 Transformer 的特征维度。随后，将向量 $\phi_{\text{in}}(a_1) \ldots, \phi_{\text{in}}(a_N)$ 输入 $N_{\text{transformer}}$ 个堆叠的 Transformer 层。经过这些 Transformer 层后，再将它们输入另一个逐原子多层感知机 ϕ_{out}： $\mathbb{R}^D \to \mathbb{R}^3$。最终输出属于 $\mathbb{R}^{3N}$，符合要求。图 2 中间部分对此进行了示意。在实现公式（9）中的 s^v_θ 和 t^v_θ 时，对向量 $[x^p_i(t)\ h_i, z^p_i]^\intercal$ 采用类似的过程，但此时包含 z^p_i，并排除 z^v_i。原子变换器与 [38] 中的架构有两个关键区别。首先，为了保持置换等变性，

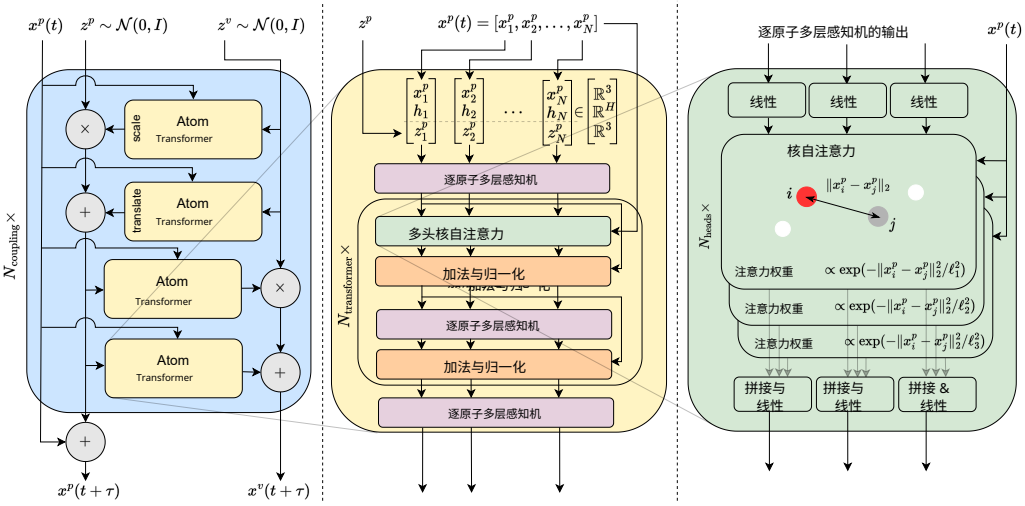

图 2：Timewarp 条件流架构的示意图，详见第4节。左侧：单个 RealNVP 耦合层。中间：单个原子 Transformer 模块。右侧：多头核自注意力模块。

我们不使用位置编码。其次，我们使用简单的核自注意力模块，而不是点积注意力，接下来将对其进行介绍。

核自注意力我们通过以下观察来阐释核自注意力模块：作用于分子中原子的物理力具有局部性：也就是说，它们对邻近原子的作用更强。直观而言，对于不太大的 τ 值，时刻 $t+\tau$ 的位置受时刻 t 附近原子的影响将大于受远处原子的影响。因此，我们将原子 i 关注原子 j 时的注意力权重 w_{ij} 定义如下：

$$w_{ij} = \frac{\exp(-\|x_i^p - x_j^p\|_2^2/\ell^2)}{\sum_{j'=1}^N \exp(-\|x_i^p - x_{j'}^p\|_2^2/\ell^2)}, \tag{10}$$

其中，ℓ 是长度尺度参数。给定输入 $\{r_{\text{in},i}\}_{i=1}^N$，输出 $\{r_{\text{out},i}\}_{i=1}^N$ 如下：

$$r_{\text{out},i} = \sum_{j=1}^N w_{ij} V \cdot r_{\text{in},j}, \tag{11}$$

其中 $V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ 是一个可学习矩阵。这种核自注意力是 [39]中研究的 RBF 核注意力的一种实例。与 [38], 类似，我们引入了多头版本，其中每个注意力头具有不同的长度尺度。这一点在图 2 右侧有所展示。我们发现，核自注意力的计算速度显著快于点积注意力，且性能相近。

4.1 对称性

分子动力学遵循某些物理对称性，将这些对称性纳入模型会带来优势。下面介绍如何在 Timewarp 中纳入这些对称性。

置换等变性设 σ 是 N 个原子的一种置换。由于原子没有内在的排序，对 $x(t)$ 进行置换对未来状态 $x(t+\tau)$ 的唯一影响，就是以相同方式置换原子，即，

$$\mu(\sigma x(t+\tau)|\sigma x(t)) = \mu(x(t+\tau)|x(t)). \tag{12}$$

我们的条件流严格满足置换等变性。为说明这一点，我们采用附录 A.1 中证明的以下命题，该命题是对条件流的扩展：

命题 *4.1*。设 σ 是一个对称作用，且 $f(\cdot;\cdot)$ 是一个等变映射，使得对所有 σ、z、x，$f(\sigma z;\sigma x)=\sigma f(z;x)$。此外，设基分布 $p(z)$ 满足对所有 σ、z，$p(\sigma z)=p(z)$。那么，由 $z\sim p(z)$、$x(t+\tau):=f(z;x(t))$ 定义的条件流满足 $p(\sigma x(t+\tau)|\sigma x(t))=p(x(t+\tau)|x(t))$。

由于 Transformer 具有置换等变性，因此我们的流满足 $f_\theta(\sigma z;\sigma x(t))=\sigma f_\theta(z;x(t))$；同时置换 z 和 $x(t)$ 会一并置换输入。此外，基分布 $p(z)=\mathcal{N}(0,I)$ 具有置换不变性。请注意，辅助变量的存在使我们能够轻松构造置换等变的耦合层。

平移和旋转等变性 考虑作用于 x^p 的变换 $T=(R,a)$：

$$Tx_i^p=Rx_i^p+a,\quad 1\le i\le N, \tag{13}$$

其中，R 是一个旋转矩阵，$a\in\mathbb{R}^3$ 是一个平移向量。我们希望模型满足 $p_\theta(Tx(t+\tau)|Tx(t))=p_\theta(x(t+\tau)|x(t))$。我们通过减去初始状态中原子的平均位置来实现平移等变性（附录 A.2）。旋转等变性并未编码在架构中，而是通过数据增强处理：在每次迭代中，对每个训练样本对 $(x(t)\ x(t+\tau))$ 应用一个随机旋转矩阵 R，以形成 $(Rx(t)\ Rx(t+\tau))$。

5 训练目标

模型分两个阶段进行训练：(i) 似然训练，模型通过对数据集中轨迹里的状态对进行最大似然训练。令 k 为训练样本对的索引，使得 $\{(x^{(k)}(t)\ x^{(k)}(t+\tau))\}_{k=1}^K$ 表示在 $\mathcal{D}$ 中时间相隔 τ 的所有状态对。我们优化：

$$\mathcal{L}_{\text{lik}}(\theta):=\tfrac{1}{K}\textstyle\sum_{k=1}^K\log p_\theta(x^{(k)}(t+\tau)|x^{(k)}(t)). \tag{14}$$

(ii) 接受率训练，对模型进行微调，以最大化 MH 接受概率。令 $x^{(k)}(t)$ 从 $\mathcal{D}$ 中均匀采样。然后，使用模型通过公式 (3) 对 $\tilde{x}_\theta^{(k)}(t+\tau)\sim p_\theta(\cdot|x^{(k)}(t))$ 进行采样。我们据此优化公式 (6) 中关于 $\mathcal{J}_\theta$ 的接受概率。令 $r_\theta(X,X)$ 表示公式 (6) 中依赖于模型的接受率项：

$$r_\theta(X,\tilde{X}):=\frac{\mu_{\text{aug}}(\tilde{X})p_\theta(X\mid\tilde{X}^p)}{\mu_{\text{aug}}(X)p_\theta(\tilde{X}\mid X^p)}. \tag{15}$$

接受目标为：

$$\mathcal{L}_{\text{acc}}(\theta):=\tfrac{1}{K}\textstyle\sum_{k=1}^K\log r_\theta(x^{(k)}(t),\tilde{x}_\theta^{(k)}(t+\tau)). \tag{16}$$

以最大化接受概率为目标进行训练，可能导致模型提出的变化过小：如果 $\tilde{x}_\theta^{(k)}(t+\tau)=x^{(k)}(t)$，那么所有提议都会被接受。为缓解这一问题，在接受率训练期间，我们使用一个加权平均目标，该目标结合了 $\mathcal{L}_{\text{acc}}(\theta)$、$\mathcal{L}_{\text{lik}}(\theta)$ 和平均微分熵的蒙特卡洛估计，

$$\mathcal{L}_{\text{ent}}(\theta):=-\tfrac{1}{K}\textstyle\sum_{k=1}^K\log p_\theta(\tilde{x}_\theta^{(k)}(t+\tau)|x^{(k)}(t)). \tag{17}$$

6 实验

我们在小型肽系统上评估 Timewarp。为与 MD 进行比较，我们重点关注亚稳态之间最慢的跃迁，因为这些跃迁最难跨越。为找到这些跃迁，我们使用时间滞后独立成分分析（TICA）[42], 这是一种线性降维技术，可最大化变换后坐标的自相关。最慢的分量 TIC 0 和 TIC 1 尤为值得关注。为衡量 Timewarp 实现的加速效果，我们计算 TICA 分量每秒实际运行时间的有效样本量（ESS/秒）。ESS/秒的计算公式为

$$\text{ESS/s}=\frac{M_{\text{eff}}}{t_{\text{sampling}}}=\frac{M}{t_{\text{sampling}}\left(1+2\sum_{\tau=1}^\infty\rho_\tau\right)}, \tag{18}$$

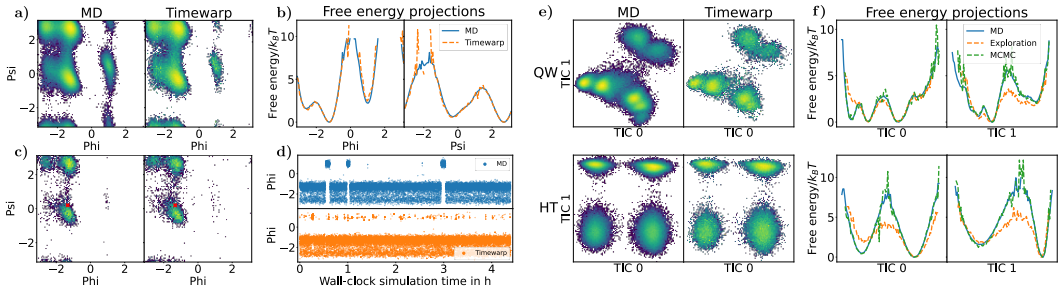

图 3：左半部分：**丙氨酸二肽实验**。(a) 根据算法1生成的分子动力学和 Timewarp 样本的拉曼查德兰图。(b) 两个二面角φ 和 ψ的自由能对比。(c) 分子动力学条件分布与 Timewarp 模型的拉曼查德兰图对比。红色叉号表示初始状态。(d) 分子动力学中 φ 二面角随时间的变化，以及由 Timewarp 模型生成的马尔可夫链中该角度随时间的变化。右半部分：**针对 2AA 测试二肽 QW（上排）和 HT（下排）的实验**。(e) 长分子动力学链与使用 Timewarp MCMC算法（算法1）生成的样本的 TICA 图。(f) 分子动力学轨迹、Timewarp 马尔可夫链蒙特卡罗（算法1）和 Timewarp 探索（算法 2）的自由能对比。

其中， M 是链长， M_{eff} 是有效样本数， t_{sampling} 是采样实际运行时间， ρ_τ 是滞后时间 τ [43] 对应的自相关。加速倍数定义为 Timewarp 达到的 ESS/秒除以 MD 达到的 ESS/秒。更多实验和结果见附录B。我们在三个数据集上训练三个流模型：(i) 丙氨酸二肽（AD），由丙氨酸二肽的模拟组成；(ii)*2AA*，包含由 2 个氨基酸组成的肽；以及 (iii)*4AA*，包含由 4 个氨基酸组成的肽。所有数据集均通过使用相同参数进行的分子动力学模拟创建（见附录 E）。对于 2AA 和 4AA，我们在随机选取的短轨迹训练集上进行训练（50ns $= 10^8$ 步），并在未见过的测试肽上进行评估。2AA 和 4AA 中氨基酸的相对频率在各个划分中相近。对于 4AA，训练集约占所有可能四肽总数的 1%（20^4），因此其泛化任务明显比 2AA 更困难。更多详情请参见附录 E 中的表 2。

丙氨酸二肽（丙氨酸二肽）我们首先研究丙氨酸二肽，这是一种小型单肽分子（含 22 个原子）。我们按照第5节所述，在丙氨酸二肽上训练 Timewarp，并使用算法1对链长为 1000 万的链进行采样，提案接受率约为 2%。在图 3a 中，我们使用拉曼查德兰图 [44], 展示样本，该图显示了主链二面角 φ 和 ψ的分布。图中的每个峰都代表一种亚稳态。可以看到，Timewarp 样本与分子动力学结果高度吻合，能够以正确的相对权重访问所有亚稳态。在图 3b 中，我们绘制了 φ 和 ψ 角的自由能（即相对对数概率），再次显示出高度吻合。图中的粗糙起伏源于状态空间中的某些区域样本极少。在图 3c 中，对于初始状态 $x\left(t\right)$，我们将通过积分方程（2）得到的分子动力学条件分布 $\mu\left(x\left(t+\tau\right)|x\left(t\right)\right)$ 与模型 $p_\theta\left(x\left(t+\tau\right)|x\left(t\right)\right)$ 进行比较，结果显示两者高度吻合。最后，图 3d 展示了分子动力学和 Timewarp 中 φ 角随时间的演变。与分子动力学相比，Timewarp 在亚稳态之间的跃迁显著更多。因此，按实际运行时间计算，沿 φ 角的自相关衰减快得多，与分子动力学相比，ESS/秒 提升了 $\approx 7\times$ 倍（见附录B.4）。

二肽（2AA）接下来，我们展示 Timewarp 在 2AA 二肽上的迁移能力。在训练二肽上完成训练后，我们使用算法1将 Timewarp 部署到测试二肽上，链长为 2000 万步。Timewarp 的接受概率介于 0.03% 和 2% 之间，并且能够探索所有亚稳态（附录B.1）。图3ef展示了二肽 QW 和 HT 的结果，表明 Timewarp 与长分子动力学链（1 μs $= 2 \times 10^9$ 步）高度吻合。对于这些二肽，Timewarp 相较于分子动力学的 ESS/秒加速比分别为 5 和 33（附录B.4）。图4左侧展示了 Timewarp 相较于分子动力学在 100 种测试二肽上的加速比。Timewarp 在这些肽上的加速比中位数约为 5。此外，我们还按照第3.5节所述，使用 Timewarp 模型生成了不进行MH 校正的样本。我们从相同的起点开始，仅运行 10000 步，采样 100 条并行链

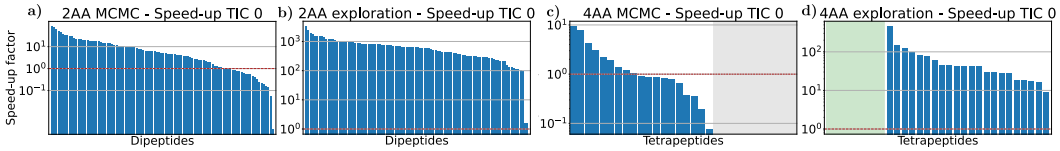

图4：Timewarp 马尔可夫链蒙特卡罗算法和探索算法相较于分子动力学的加速比，以最慢的 TICA 分量的 ESS/秒比值表示。红色虚线表示加速比为 1。灰色区域表示 Timewarp 在 2000 万步内未能探索所有亚稳态、而分子动力学能够做到的肽。绿色区域表示分子动力学未能找到所有亚稳态、而 Timewarp 能够做到的肽。(a)、(c) 分别表示 Timewarp MCMC算法（算法1）在测试二肽（2AA）和四肽（4AA）上的加速比。(b)、(d) 分别表示 Timewarp 探索算法（算法 2）在测试二肽（2AA）和四肽（4AA）上的加速比。

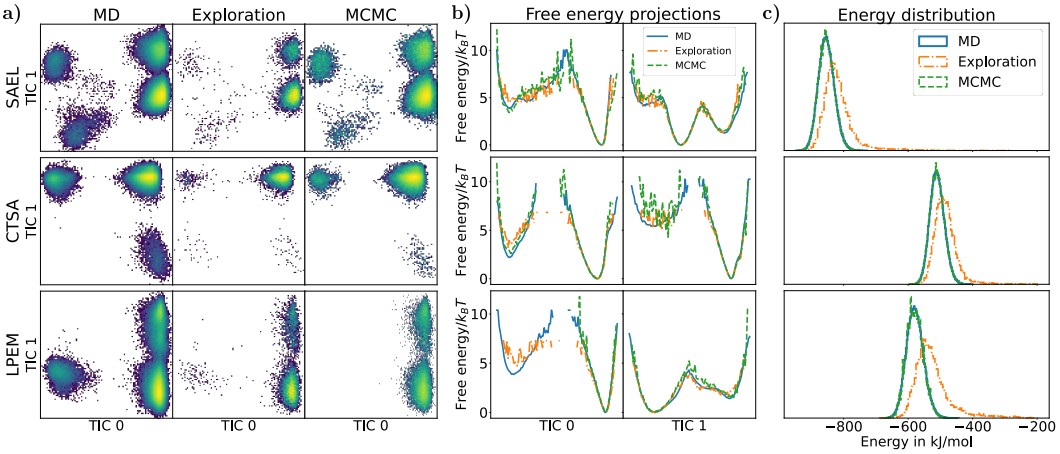

图5：在 4AA 测试四肽 SAEL、CTSA 和 LPEM 上进行的实验（分别对应顶部、中部和底部各行）。样本通过分子动力学、Timewarp 探索（算法 2）和 Timewarp 马尔可夫链蒙特卡洛（算法1）生成。(a) 样本的 TICA 图。(b) 沿前两个 TICA 分量的自由能。(c) 势能分布。

每种测试肽的初始状态。对于每种肽，我们仅选择一条 能够找到所有亚稳态的链进行评估。与前文一样，我们计算 ESS/秒以便与分子动力学比较，得出中位加速比为 ≈ 600 （图4c）。请注意，若使用所有并行采样的链，实际加速比会大得多。Timewarp 探索得到的自由能估计在定性上与分子动力学相似，但不如 Timewarp 马尔可夫链蒙特卡洛准确（图3f）。

四肽（4AA）最后，我们研究了更具挑战性的4AA数据集。在训练集上完成训练后，我们使用算法1为每个测试四肽采样2000万个马尔可夫链状态，并将其与长分子动力学轨迹（1μs）进行比较。与较简单的二肽不同，Timewarp 马尔可夫链蒙特卡洛和长分子动力学轨迹都遗漏了一些亚稳态。不过，Timewarp 的探索模式（算法 2）可用作验证工具，快速确认是否已探索整个状态空间。图5a显示，分子动力学和 Timewarp 马尔可夫链蒙特卡洛未探索到的亚稳态，可以通过 Timewarp 探索算法找到。我们通过在这些状态附近运行较短的分子动力学轨迹（见附录 B.5），仔细确认所发现状态的物理有效性，以确保它们并非模型凭空产生的伪影。与2AA一样，我们再次在图4b、d中报告 Timewarp 相对于分子动力学的加速比。尽管 Timewarp 马尔可夫链蒙特卡洛未能加速大多数四肽的采样，Timewarp 探索的加速比中位数为 ≈ 50。对于8个测试四肽，分子动力学未能探索所有亚稳态，而 Timewarp 做到了——这些四肽以绿色标出。对于10个四肽，Timewarp 马尔可夫链蒙特卡洛未能找到分子动力学发现的所有亚稳态——这些四肽以灰色标出。图5b显示，当 Timewarp 马尔可夫链蒙特卡洛发现了所有亚稳态时，其自由能估计与分子动力学的结果非常吻合。然而，它有时会

遗漏亚稳态，导致这些区域的自由能估计较差。图5c显示，Timewarp 马尔可夫链蒙特卡洛得到的势能分布也与分子动力学非常接近。相比之下，Timewarp 探索能够发现所有亚稳态（包括分子动力学遗漏的亚稳态），但其自由能图的准确性较低。此外，与分子动力学和 Timewarp 马尔可夫链蒙特卡洛相比，它得到的势能分布略偏高。

7 局限性/未来工作

对于大多数肽，Timewarp MCMC算法产生的接受概率较低（$< 1\%$）（见附录B.1）。然而，这本身并不是一种局限。一般来说，提议时间步长 τ 越大，预测问题就越困难，接受率也越低。不过，借助算法1，我们可以并行评估多个样本，几乎不会产生额外成本。因此，当较低的接受率与较大的时间步长 τ 相结合时，往往是值得的权衡。使用 MH 校正时，我们仅能加速大约三分之一的四氨基酸肽；但对于此前未见过的肽，在全原子表示下以实际运行时间击败分子动力学，这是一项颇具挑战的任务，而机器学习方法此前尚未实现这一点。此外，还可以考虑将目标系统改为采用半经验力场，而非经典力场。鉴于 Timewarp 所需的能量计算次数远少于分子动力学模拟，因此可以预期，在这种情况下加速效果会更加显著。

尽管分子动力学和 Timewarp MCMC 未能找到 Timewarp 探索发现的某些亚稳态，但由于计算成本高昂（附录F），我们没有进一步延长分子动力学和 Timewarp MCMC 的运行时间。在相同时间内，与传统分子动力学模拟相比，Timewarp 生成的样本更少。因此，在过渡态中，样本不足的问题会更加突出，这使得 Timewarp 难以应用于化学反应。Timewarp 可以与其他增强采样方法结合使用，例如并行回火或过渡路径采样。对于并行回火，要有效地结合使用，就需要在多个温度下训练 Timewarp 模型，从而能够用 Timewarp 而非分子动力学对所有副本进行采样。我们还可以将 Timewarp 提议与集体变量（例如二面角）的学习更新交替进行。这些组合步骤仍能确保从目标分布中进行无偏采样 [31]。

此外，本研究只考察了小型肽系统。如何将 Timewarp 扩展到更大的系统仍是未来研究的课题，有几个值得探索的方向。一种方法是研究不同的网络架构，或许能够捕捉系统固有的所有对称性。另一种选择是研究粗粒化结构，而非全原子表示，从而降低问题的维数。

8 结论

我们介绍了 Timewarp，这是一种可迁移的增强采样方法，利用深度网络在模拟分子系统时提出较大的构象变化。我们证明，结合 MH 校正使用的 Timewarp 能够加速对许多未见过的二肽进行渐近无偏采样，从而更快地计算平衡期望值。虽然这种加速仅适用于我们考察的少数四肽，但我们证明，不使用*MH* 校正 的 Timewarp 探索二肽和四肽亚稳态的速度远快于标准分子动力学，并验证了所发现的亚稳态具有物理意义。这为快速验证分子动力学模拟是否访问了所有亚稳态提供了一种有前景的方法。尽管要将 Timewarp 扩展到规模更大、也更有研究意义的生物分子，还需要进一步工作，但本研究清楚地展示了深度学习算法利用可迁移性加速分子动力学采样问题的能力。

致谢

感谢 Bas Veeling、Claudio Zeni、Andrew Fowler、Lixin Sun、Chris Bishop、Rianne van den Berg、Hannes Schulz、Max Welling 以及整个 Microsoft AI4Science 团队提供富有见地的讨论和计算支持。

参考文献

[1] Frank Noé、Christof Schütte、Eric Vanden-Eijnden、Lothar Reich 和 Thomas R Weikl。通过短时非平衡模拟构建折叠路径的平衡系综。《美国国家科学院院刊》，106(45):19011–19016，2009.[2] Kresten Lindorff-Larsen、Stefano Piana、Ron O Dror 和 David E Shaw。快速折叠蛋白质如何折叠。《科学》，334(6055):517–520，2011.[3] Ignasi Buch、Toni Giorgino 和 Gianni De Fabritiis。通过分子动力学模拟完整重建酶-抑制剂结合过程。《美国国家科学院院刊》，108(25):10184–10189，2011.[4] Nuria Plattner、Stefan Doerr、Gianni De Fabritiis 和 Frank Noé。通过分子动力学模拟和马尔可夫建模，以原子级细节揭示完整的蛋白质-蛋白质缔合动力学。《自然·化学》，9(10):1005–1011，2017.[5] Paul Langevin。关于布朗运动理论。*Compt.Rendus*，146:530–533，1908.[6] Cecilia Clementi。蛋白质折叠的粗粒化模型：玩具模型还是预测工具？结构生物学当前观点，18(1):10–15，2008.[7] Sebastian Kmiecik、Dominik Gront、Michal Kolinski、Lukasz Wieteska、Aleksandra Elzbieta Dawid 和 Andrzej Kolinski。粗粒化蛋白质模型及其应用。《化学评论》，116(14):7898–7936，2016.[8] Alessandro Laio 和 Michele Parrinello。逃离自由能极小值。《美国国家科学院院刊》，99(20):12562–12566，2002.[9] Danilo Rezende 和 Shakir Mohamed。使用归一化流进行变分推断。载于国际机器学习会议，第1530–1538页。机器学习研究论文集，2015年。[10] Frank Noé、Simon Olsson、Jonas Köhler 和 Hao Wu。玻尔兹曼生成器——利用深度学习采样多体系统的平衡态。《科学》，365:eaaw1147，2019.[11] Jonas Köhler、Andreas Krämer 和 Frank Noé。平滑归一化流。载于 M. Ranzato、A. Beygelzimer、Y. Dauphin、P.S. Liang 和 J. Wortman Vaughan 编，《style id='32'>神经信息处理系统进展》，第34卷，第2796–2809页。Curran Associates, Inc.，2021年。[12] Jonas Köhler、Michele Invernizzi、Pim de Haan 和 Frank Noé。用于分子晶体结构采样的刚体流。载于国际机器学习会议，*ICML 2023*，第202卷《style id='37'>机器学习研究论文集，第17301–17326页。机器学习研究论文集，2023年。[13] Manuel Dibak、Leon Klein、Andreas Krämer 和 Frank Noé。温度可控流与玻尔兹曼生成器。《物理评论研究》，4:L042005，2022年10月。[14] Bowen Jing、Gabriele Corso、Jeffrey Chang、Regina Barzilay 和 Tommi S. Jaakkola。用于分子构象生成的扭转扩散。载于 Alice H. Oh、Alekh Agarwal、Danielle Belgrave 和 Kyunghyun Cho 编，《style id='43'>神经信息处理系统进展，2022年。[15] Leon Klein、Andreas Krämer 和 Frank Noe。等变流匹配。载于第三十七届神经信息处理系统大会，2023年。[16] Laurence Illing Midgley、Vincent Stimper、Javier Antoran、Emile Mathieu、Bernhard Schölkopf 和 José Miguel Hernández-Lobato。SE(3)等变增强耦合流。载于第三十七届神经信息处理系统大会，2023年。[17] Minkai Xu、Lantao Yu、Yang Song、Chence Shi、Stefano Ermon 和 Jian Tang。GeoDiff：用于分子构象生成的几何扩散模型。载于国际学习表征会议，2022年。[18] Mathias Schreiner、Ole Winther 和 Simon Olsson。隐式转移算子学习：用于分子动力学的多种时间分辨率模型。载于第三十七届神经信息处理系统大会，2023年。

[19] Xiang Fu、Tian Xie、Nathan J. Rebello、Bradley Olsen 和 Tommi S. Jaakkola。使用多尺度图网络模拟时间积分的粗粒化分子动力学。机器学习研究汇刊，2023。[20] Jan-Hendrik Prinz、Hao Wu、Marco Sarich、Bettina Keller、Martin Senne、Martin Held、John D Chodera、Christof Schütte 和 Frank Noé。分子动力学的马尔可夫模型：生成与验证。《化学物理学杂志》，134(17):174105，2011。[21] William C Swope、Jed W Pitera 和 Frank Suits。通过分子动力学模拟描述蛋白质折叠动力学。1. 理论。《物理化学杂志*B*》，108(21):6571–6581，2004。[22] Brooke E Husic 和 Vijay S Pande。马尔可夫状态模型：从艺术走向科学。《美国化学会志》，140(7):2386–2396，2018。[23] Andreas Mardt、Luca Pasquali、Hao Wu 和 Frank Noé。用于分子动力学深度学习的 VAMPnets。《自然·通讯》，9(1):1–11，2018。[24] Hao Wu、Andreas Mardt、Luca Pasquali 和 Frank Noé。深度生成式马尔可夫状态模型。神经信息处理系统进展，31，2018。[25] Jiaming Song、Shengjia Zhao 和 Stefano Ermon。A-NICE-MC：用于马尔可夫链蒙特卡罗的对抗训练。神经信息处理系统进展，30，2017。[26] Daniel Levy、Matt D. Hoffman 和 Jascha Sohl-Dickstein。利用神经网络推广哈密顿蒙特卡罗。载于国际学习表征会议，2018。[27] Zengyi Li、Yubei Chen 和 Friedrich T. Sommer。最大化提议熵的神经网络马尔可夫链蒙特卡罗采样器。《熵》，23(3)，2021。[28] Michalis Titsias 和 Petros Dellaportas。基于梯度的自适应马尔可夫链蒙特卡罗。神经信息处理系统进展，32，2019。[29] Robert H Swendsen 和 Jian-Sheng Wang。自旋玻璃的副本蒙特卡罗模拟。《物理评论快报》，57(21):2607，1986。[30] David J. Earl 和 Michael W. Deem。并行回火：理论、应用与新视角。*Phys. Chem. Chem. Phys.*，7:3910–3916，2005。[31] Jerome P Nilmeier、Gavin E Crooks、David DL Minh 和 John D Chodera。非平衡候选蒙特卡罗是进行平衡态模拟的高效工具。《美国国家科学院院刊》，108(45):E1009–E1018，2011。[32] Christina Winkler、Daniel Worrall、Emiel Hoogeboom 和 Max Welling。利用条件归一化流学习似然。*arXiv* 预印本 *arXiv:1912.00042*，2019。[33] Peter Eastman、Jason Swails、John D Chodera、Robert T McGibbon、Yutong Zhao、Kyle A Beauchamp、Lee-Ping Wang、Andrew C Simmonett、Matthew P Harrigan、Chaya D Stern 等。OpenMM 7：快速开发用于分子动力学的高性能算法。《公共科学图书馆·计算生物学》，13(7):e1005659，2017。[34] C. Huang、Laurent Dinh 和 Aaron C. Courville。增强归一化流：弥合生成流与潜变量模型之间的差距。*arXiv*，abs/2002.07101，2020。[35] Jianfei Chen、Cheng Lu、Biqi Chenli、Jun Zhu 和 Tian Tian。VFlow：通过变分数据增强实现更具表达力的生成流。载于国际机器学习会议，第1660–1669页。机器学习研究论文集，2020。[36] Nicholas Metropolis、Arianna W Rosenbluth、Marshall N Rosenbluth、Augusta H Teller 和 Edward Teller。利用快速计算机进行状态方程计算。《化学物理学杂志》，21(6):1087–1092，1953。[37] Laurent Dinh、Jascha Sohl-Dickstein 和 Samy Bengio。使用 real NVP 进行密度估计。载于国际会议

*f*关于学习的会议 *g Rep*论文集，2017年。

[38] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。Attention is all you need。神经信息处理系统进展，30，2017年。[39] Yao-Hung Hubert Tsai、Shaojie Bai、Makoto Yamada、Louis-Philippe Morency 和 Ruslan Salakhutdinov。Transformer 解剖：通过核的视角统一理解 Transformer 注意力。载于*2019*年自然语言处理实证方法会议和第*9*届国际自然语言处理联合会议论文集（*EMNLP-IJCNLP*），第4344–4353页，中国香港，2019年11月。计算语言学协会。[40] Jonas Köhler、Leon Klein 和 Frank Noé。等变流：用于对称密度的精确似然生成式学习。载于国际机器学习会议，第5361–5370页。机器学习研究论文集，2020年。[41] Danilo Jimenez Rezende、Sébastien Racanière、Irina Higgins 和 Peter Toth。等变哈密顿流。*arXiv* 预印本 *arXiv:1909.13739*，2019年。[42] Guillermo Pérez-Hernández、Fabian Paul、Toni Giorgino、Gianni De Fabritiis 和 Frank Noé。用于马尔可夫模型构建的慢分子序参量识别。《化学物理学杂志》，139(1):07B604_1，2013年。[43] Radford M Neal。使用马尔可夫链蒙特卡罗方法进行概率推断。多伦多大学计算机科学系，加拿大安大略省多伦多，1993年。[44] G N Ramachandran、C Ramakrishnan 和 V Sasisekharan。多肽链构象的立体化学。《分子生物学杂志》，第95–99页，1963年。[45] William Thomson Baron Kelvin。晶体的分子策略。克拉伦登出版社，1894年。[46] Jeff Rasley、Samyam Rajbhandari、Olatunji Ruwase 和 Yuxiong He。DeepSpeed：系统优化使训练参数超过1000亿的深度学习模型成为可能。载于第*26*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集，KDD ’20，第3505–3506页，美国纽约州纽约，2020年。计算机协会。

A 架构的对称性

A.1 命题 4.1 的证明

在本附录中，我们将更详细地介绍 Timewarp 架构的等变性。首先，我们证明正文中的命题 4.1：

证明。 令 $X(t+\tau)^{x(t)}$ 表示通过对 $Z \sim p(z)$ 进行采样并计算 $X(t+\tau) := f(Z;\, x(t))$ 所得到的随机变量。这里，我们用 $x(t)$ 作为 $X(t+\tau)^{x(t)}$ 的下标，以强调这是在以 $x(t)$ 为条件时，对流进行条件化后得到的随机变量。我们首先指出，密度 $p(\sigma x(t+\tau)|\sigma x(t)) = p(x(t+\tau)|x(t))$ 的等变性条件，等价于对随机变量施加以下约束：

$$X(t+\tau)_{\sigma x(t)} \overset{d}{=} \sigma X(t+\tau)_{x(t)}, \tag{19}$$

其中，$\overset{d}{=}$ 表示分布相等。为说明这一点，令 p_X 表示随机变量 X 的密度。那么，就密度而言，方程 (19) 等价于对所有 x 均有：

$$p_{X(t+\tau)_{\sigma x(t)}}(x) = p_{\sigma X(t+\tau)_{x(t)}}(x) \tag{20}$$

$$= p_{X(t+\tau)_{x(t)}}(\sigma^{-1}x), \tag{21}$$

其中，在公式 (21) 中，我们使用了变量替换公式，并利用了我们所考虑的群作用（旋转、平移、置换）的雅可比行列式绝对值均为 1 这一事实。重新定义 $x \leftarrow \sigma x$，可得对于所有 x，

$$p_{X(t+\tau)_{\sigma x(t)}}(\sigma x) = p_{X(t+\tau)_{x(t)}}(x). \tag{22}$$

回顾一下记号：$X(t+\tau)^{x(t)}$ 表示通过以 $x(t)$ 为条件对流进行条件化而得到的随机变量，因此可以将其写为

$$p(\sigma x|\sigma x(t)) = p(x|x(t)) \tag{23}$$

这正是上文以密度形式表述的等变性条件。将公式 (19) 中的等变性条件改写为随机变量形式后，命题 4.1 的证明就很直接了。

$$X(t+\tau)_{\sigma x(t)} := f(Z, \sigma x(t)) \tag{24}$$

$$\overset{d}{=} f(\sigma Z, \sigma x(t)) \tag{25}$$

$$= \sigma f(Z, x(t)) \tag{26}$$

$$:= \sigma X(t+\tau)_{x(t)}, \tag{27}$$

其中，在公式 (25) 中，我们利用了基分布 $p(z)$ 具有 σ-不变性这一事实。 □

A.2 通过规范化实现平移等变性

现在我们介绍用于使模型具有平移等变性的规范化技术。设 $q(x^p(t+\tau)\ x^v(t+\tau)|x^p(t))$ 是任意一个条件密度模型，它不一定具有平移等变性。我们可以通过以下方式使其具有平移等变性。令 x^p 表示原子的平均位置，

$$\overline{x^p} := \frac{1}{N}\sum_{i=1}^{N} x_i^p. \tag{28}$$

然后我们定义

$$p(x^p(t+\tau), x^v(t+\tau)|x^p(t)) := q(x^p(t+\tau) - \overline{x^p(t)}, x^v(t+\tau)|x^p(t) - \overline{x^p(t)}) \tag{29}$$

其中，$x^p(t)$ 的减法会广播到所有原子。现在，我们考虑将 $x^p(t)$ 和 $x^p(t+\tau)$ 同时平移相同的距离所产生的影响。设 a 是 $\mathbb{R}^3$ 中的一个平移向量。那么

$$p(x^p(t+\tau)+a, x^v(t+\tau)|x^p(t)+a) \tag{30}$$

$$= q(x^p(t+\tau)+a-(\overline{x^p(t)+a}), x^v(t+\tau)|x^p(t)+a-(\overline{x^p(t)+a})) \tag{31}$$

$$= q(x^p(t+\tau)+a-\overline{x^p(t)}-a, x^v(t+\tau)|x^p(t)+a-\overline{x^p(t)}-a) \tag{32}$$

$$= q(x^p(t+\tau)-\overline{x^p(t)}, x^v(t+\tau)|x^p(t)-\overline{x^p(t)}) \tag{33}$$

$$= p(x^p(t+\tau), x^v(t+\tau)|x^p(t)). \tag{34}$$

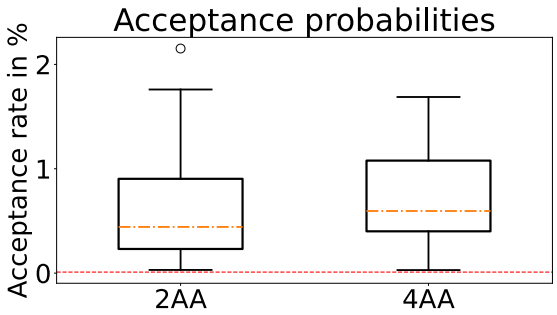

图6：使用 Timewarp 马尔可夫链蒙特卡罗算法对未见测试肽进行采样时的接受概率。红线位于 0.01%；低于该值后，高效采样将变得困难。

因此，即使 q 是平移等变的，p 也具有平移等变性 not.

A.3 手性

除第 4.1 节所述的对称性外，分子构型的势能 $U(x^p)$ 在镜像变换下也保持不变。然而，存在手性中心时，镜像构型无法与其原始构型 [45] 重合。氨基酸中的手性中心例如是一个连接着四个不同基团的碳原子，如 C_α 原子。在自然界中，大多数氨基酸只有一种形式，即 L-氨基酸。因此，我们的所有数据集都由仅含 L-氨基酸的肽组成。在少数情况下，当模型提出较大的步长时，某一步可能会将肽中的一个 L-氨基酸变为 D-氨基酸，且所得构型的能量较低，因而该步会被接受。我们通过在每一步检查所有手性中心是否发生变化，并拒绝发生此类变化的样本来避免这种情况。这样做不会显著增加计算开销。

B 附加结果

本节展示了更多结果，例如条件分布，以及第 6 节所讨论实验中的更多肽示例。

B.1 2AA 附加结果

图 7 和图 8 展示了更多来自 2AA 测试集的示例。对于二肽 GP（图 7 最后一行），我们获得的加速效果最差，因为它没有表现出任何缓慢跃迁。图 6 展示了 Timewarp MCMC算法的接受概率分布。

B.2 4AA 附加结果

Mo更多来自 4AA 测试集的示例见图 9 和图 11。其分布 e
^{acc}Timewarp MCMC算法的接受概率如图6所示。

B.3 条件分布

模型经过训练，可以从条件玻尔兹曼分布 $\mu(x(t+\tau)|x(t))$中生成样本。此处我们展示一些示例，将 Timewarp 模型生成的条件分布与分子动力学所诱导的条件分布进行比较。模型可以并行生成 5,000 个条件分布样本，而要通过分子动力学模拟对长度为 τ 的条件分布进行采样，则需要 5,000 条不同的分子动力学轨迹。因此，使用模型生成条件分布样本的速度快了好几个数量级。图10至图12展示了丙氨酸二肽以及来自 2AA 和 4AA数据集的肽的条件分布示例。对于所有肽，模型都学到了与分子动力学条件分布相近的条件分布。此外，TICA投影中的相对权重以及键长分布也非常吻合。只有模型样本的能量较高，

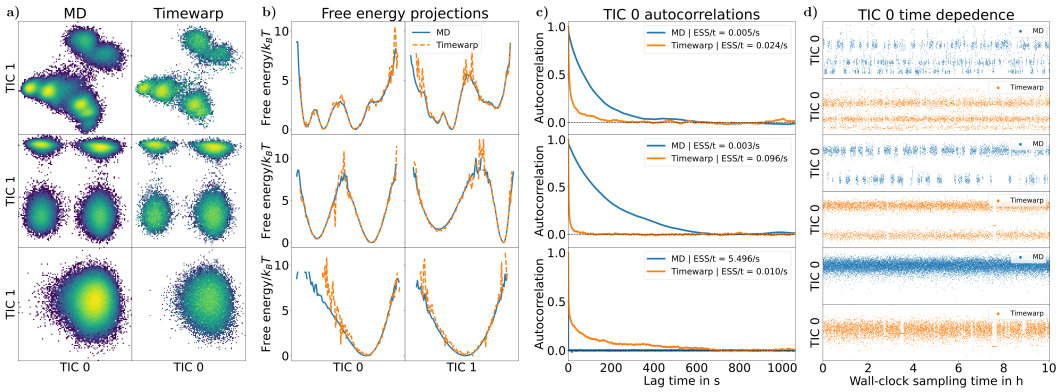

图7：对 2AA 测试二肽 QW（第一行）、HT（第二行）和 GP（第三行）的实验。比较长分子动力学轨迹与 Timewarp 马尔可夫链蒙特卡洛（算法1）。(a) TICA 图。(b) 前两个 TICA 分量的自由能比较。(c) TIC 0 分量的自相关。(d) TIC 0 分量随时间的变化。

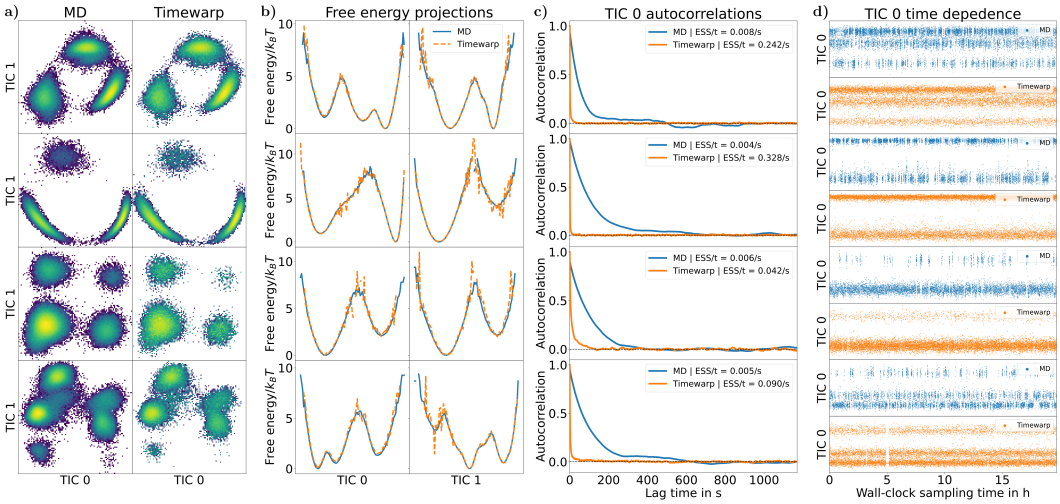

图8：对 2AA 测试二肽 DH（第一行）、GT（第二行）、TK（第三行）和 CW（最后一行）的实验。比较长分子动力学轨迹与 Timewarp 马尔可夫链蒙特卡洛（算法1）。(a) TICA 图。(b) 前两个 TICA 分量的自由能比较。(c) TIC 0 分量的自相关。(d) TIC 0 分量随时间的变化。

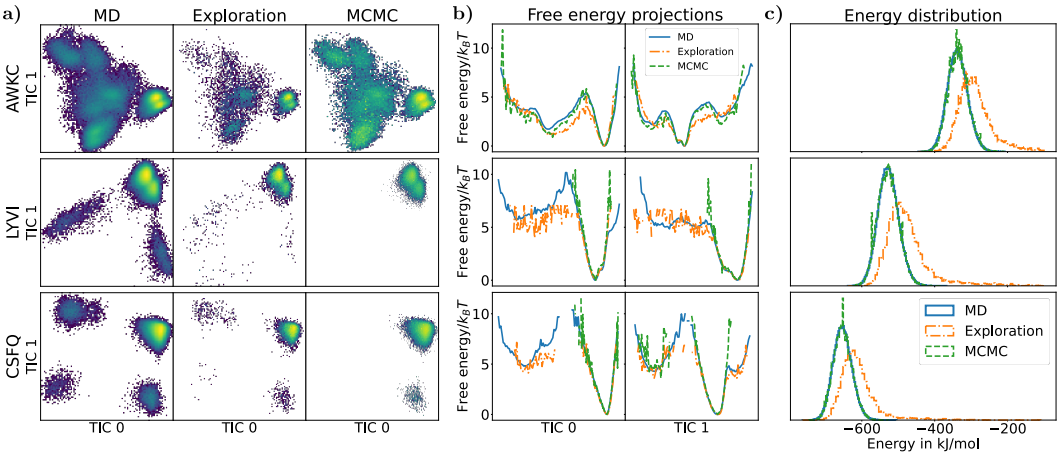

图 9：对 4AA 测试四肽 AWCK、LYVI 和 CSFQ 的实验（分别对应顶行、中行和底行）。样本通过分子动力学、Timewarp 探索（算法 2）和 Timewarp 马尔可夫链蒙特卡罗（算法1）生成。(a) 样本的 TICA 图。(b) 沿前两个 TICA 分量的自由能。(c) 势能分布。对于 AWCK，所有方法都找到了全部亚稳态；对于 LYVI，分子动力学轨迹漏掉了一个状态；对于 CSFQ，Timewarp 马尔可夫链蒙特卡罗漏掉了最慢的跃迁。在所有情况下，Timewarp 探索都发现了全部亚稳态。

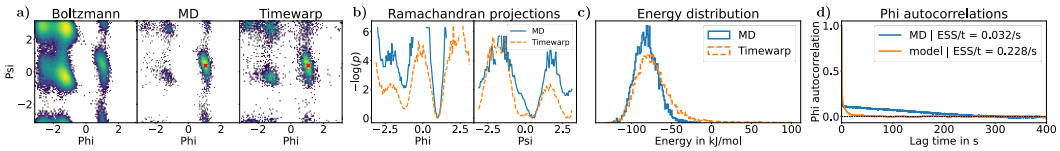

图 10：比较由丙氨酸二肽的分子动力学轨迹和 Timewarp 模型生成的条件玻尔兹曼分布。(a) 条件分布的拉曼查德兰图，并与平衡玻尔兹曼分布进行比较。红色叉号表示条件状态。这与图 3c 所示的图类似，但这里展示的是另一个条件状态。这里条件分布之间的吻合程度不如图 3c 中那么高，这可能是因为此处选择的条件状态位于概率较低的亚稳态中。(b) 条件分布在前两个二面角上的投影。(c) 条件分布的势能。(d) 根据马尔可夫链蒙特卡罗算法（算法1）生成的样本的自相关，并与一条较长的分子动力学轨迹进行比较。请注意，这张自相关图并非针对条件分布，而是对应于图 3 中展示的结果。

强调了使用 Metropolis–Hastings 校正，以通过 Timewarp 模型从玻尔兹曼分布中获得无偏样本的重要性。

B.4 自相关

在第 6 节中，我们通过将最慢跃迁的每秒有效样本量（公式（18））与分子动力学进行比较，计算 Timewarp 模型的加速比。由于有效样本量取决于自相关，因此从实际运行时间的角度考察自相关衰减也很有意义。我们在图 7、图 8 和图 10 中展示了所研究肽的一些自相关示例。请注意，自相关曲线下的面积与有效样本量成反比。

B.5 新亚稳态的探索

对于测试集中的一些四肽，即使是较长的分子动力学轨迹（1μs）也会遗漏某些亚稳态，例如图12a所示的 LYVI 和 CSTA。不过，我们可以轻松地通过 Timewarp 探索算法（算法 2）探索这些状态。为了确认这些额外的亚稳态确实

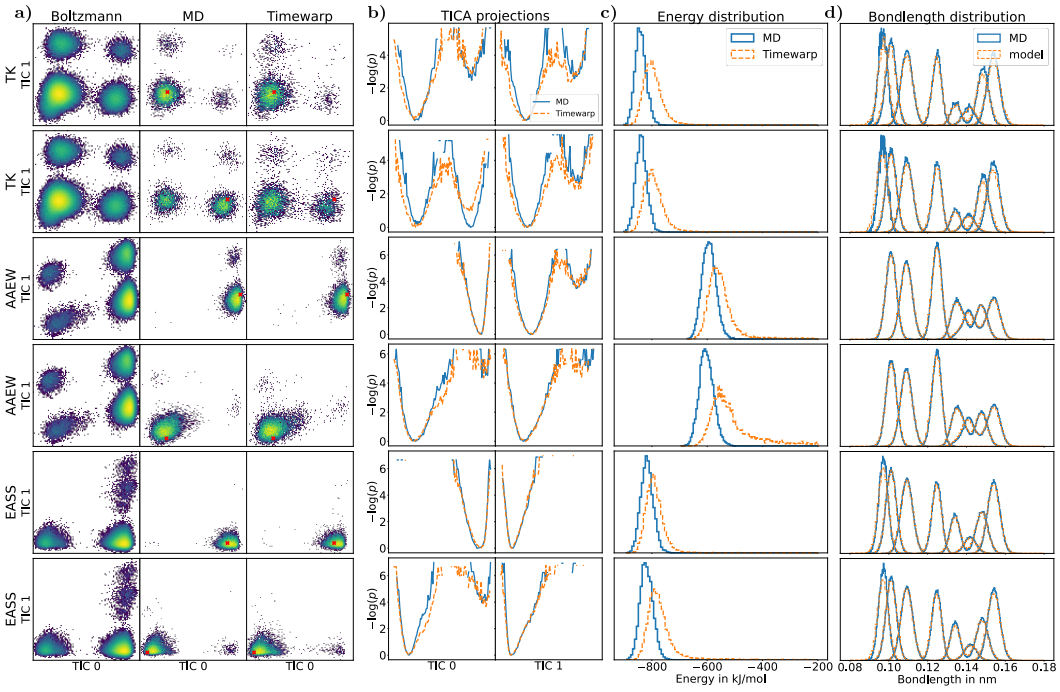

图 11：比较 Timewarp 模型的条件分布 $p_\theta(x(t+\tau)|x(t))$ 与通过分子动力学生成的条件分布 $\mu(x(t+\tau)|x(t))$。各行依次对应肽 TK、AAEW 和 EASS；对于每种肽，我们展示两种不同的条件状态。(a) TICA 图。第一列：通过分子动力学生成的玻尔兹曼分布 $\mu(x)$ 样本。第二列：通过分子动力学生成的 $\mu(x(t+\tau)|x(t))$ 样本。条件状态 $x(t)$ 用红色叉号标出。第三列：通过 Timewarp 条件流生成的 $p_\theta(x(t+\tau)|x(t))$ 样本，未进行 MH 校正。条件状态 $x(t)$ 用红色叉号标出。(b) 将 (a) 中的条件分布投影到前两个 TICA 分量上。(c) 条件分布的势能分布。(d) 条件键长分布；对于这些 τ 值，该分布将接近平衡分布。图中的每种模式代表一种不同的键类型，例如 C-H。

稳定，我们按照附录B.3中的相同方法，从较长分子动力学轨迹已发现的一个邻近亚稳态出发，运行多条较短的分子动力学轨迹（0.5×10^6fs）（图12b）。其中一条轨迹一旦到达这个此前尚未发现的新状态，我们也从该状态出发运行新的短MD轨迹（0.5×10^6fs）（图12c）。这些新的短MD轨迹要么在这个此前未发现的状态内采样，要么跃迁到其他亚稳态。这表明，Timewarp 探索发现的这个亚稳态确实有效，只是在较长的分子动力学轨迹中未被发现。此外，我们在图12中比较了条件分子动力学分布与 Timewarp 的分布，结果再次显示二者高度吻合。

C 使用 MH 校正的 Timewarp 批量采样算法

第 3.4 节所述算法的伪代码见算法1。

D 使用未经 MH 校正的 Timewarp 进行亚稳态探索

我们介绍 Timewarp 的探索算法，该算法接受所有提出的状态，除非其能量高于某个截断值。由于没有 MH 校正，生成的样本不会渐近地服从玻尔兹曼分布，但与分子动力学或算法 1 相比，该算法探索状态空间的速度要快得多。伪代码见算法 2：

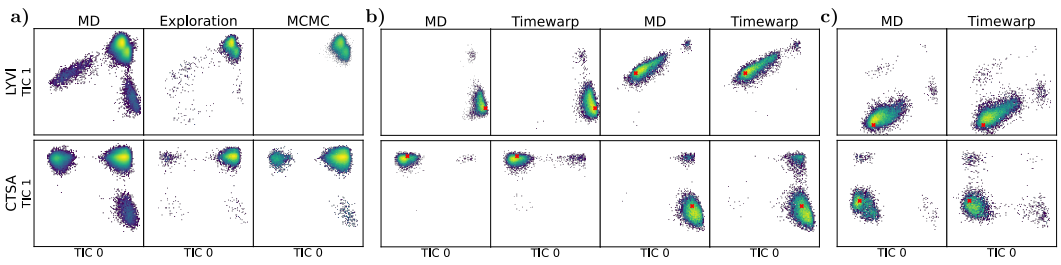

图12：验证通过 Timewarp 探索发现的新亚稳态。红色十字表示不同的条件状态。(a) 通过长分子动力学轨迹、Timewarp 探索（算法 2）和 Timewarp 马尔可夫链蒙特卡罗（算法1）生成的（非条件）样本的 TICA 图。Timewarp 探索发现了一些在长分子动力学轨迹中未曾出现的亚稳态（LYVI图底部、CTSA图左下角）。(b) 从 (a) 所示长分子动力学轨迹访问过的状态出发，通过分子动力学或 Timewarp 模型生成的条件分布。一些短MD轨迹现在发现了这些新亚稳态，验证了它们确实是有效状态。这里只记录每条短MD轨迹的最终状态。(c) 从 (b) 所示某条短MD轨迹发现的新亚稳态出发，通过分子动力学或 Timewarp 模型生成的条件分布。

算法1 Timewarp MH校正MCMC，采用批量提议

要求： 初始状态 $X_0 = (X_0^p, X_0^v)$，链长 M，提议批量大小 B。

$m \leftarrow 0$
当 $m < M$ **时执行**
  采样 $\tilde{X}_1, \ldots, \tilde{X}_B \sim p_\theta(\cdot | X_m^p)$ {批次样本}
  对于 $b = 1, \ldots, B$ **do**
    $\epsilon \sim \mathcal{N}(0, I)$ {重新采样辅助变量}
    $X_b \leftarrow (X_m^p, \epsilon)$
    采样 $I_b \sim$ 伯努利$(\alpha(X_b, \tilde{X}_b))$
  结束循环
  如果 $S := \{b : I_b = 1, 1 \leq b \leq B\} \neq \emptyset$ **则**
    $a =$ 最小值(S) {首个接受的样本}
    $(X_{m+1}^p \ldots, X_{m+a-1}^p) \leftarrow (X_m^p, \ldots, X_m^p)$
    $X_{m+a}^p \leftarrow \tilde{X}_a^p$
    $m \leftarrow m + a$
  else
    $(X_{m+1}^p \ldots, X_{m+B}^p) \leftarrow (X_m^p, \ldots, X_m^p)$
    $m \leftarrow m + B$
  结束如果
结束循环
输出 $X_0^p, \ldots X_M^p$

算法 2使用 Timewarp 快速、有偏地探索状态空间**要求：** 初始状态 X_0^p，步数 M，允许的最大能量增加 ΔU_{max}**对于** $m = 0, \ldots, M$**执行**~采样 $X_m^p \sim p_\theta(\cdot \mid X_m^p)$ {从条件流中采样}~**如果** $U(X_m^p) - U(X_m^p) < \Delta U_{\text{max}}$**则**$X_{m+1}^p \leftarrow \tilde{X}_m^p$**否则**$X_{m+1}^p \leftarrow X_m^p$ {若能量变化过大，则拒绝}**结束如果 结束循环 输出** $X_0^p, \ldots X_M^p$

请注意，与算法1不同，这里不需要辅助变量，因为条件流仅取决于位置，而且这里不会计算 MH 接受率。此处的势能U 还包括一项较大的惩罚：如果手性中心的顺序发生变化，就会施加该惩罚，具体说明见附录 A.3。由于可以对来自 Timewarp 的采样提议进行批处理，因此我们可以并行生成 B 条链，它们都从同一个初始状态开始。这种批量采样过程还能带来更高的加速比。对于所有探索实验，我们使用批量大小 100，并运行 $M = 10000$个探索步数。允许的最大能量变化截断值设为 $\Delta U_{\text{max}} = 300$kJ/mol。

E 数据集详情

我们在三个不同的数据集（丙氨酸二肽、2AA 和 4AA）上评估模型，这些数据集已在第 6 节中介绍。所有数据集均使用 OpenMM 库 [33] 在隐式溶剂中进行模拟。对于所有分子动力学模拟，我们均使用表 1 所示的参数。

表 1：OpenMM 分子动力学模拟参数

力场	amber-14
时间步长	0.5飞秒
摩擦系数	0.3 $\frac{1}{\text{ps}}$
温度	310K
积分器	LangevinMiddleIntegrator

我们在表 2 中提供了更多数据集详情，例如仿真时间和肽的数量。

F 超参数

根据数据集的不同，使用了不同规模的 Timewarp 模型，如表 3 所示。对于所有数据集，多头核自注意力层均由 6 个注意力头组成，其长度尺度为 $\ell_i = \{0.1, 0.2, 0.5, 0.7, 1.0, 1.2\}$，单位为纳米。

表 2：数据集详情

数据集名称	AD	2AA	4AA
训练集模拟时间	100 纳秒	50 纳秒	50 纳秒
测试集仿真时间	100 纳秒	1 μs	1 μs
分子动力学积分步长 Δt	0.5 飞秒	0.5 飞秒	0.5 飞秒
Timewarp 预测时间 τ	0.5×10^6飞秒	0.5×10^6飞秒	0.5×10^5飞秒
训练肽数量	1	200	1400
每条肽的训练对数量	2×10^5	1×10^4	1×10^4
测试肽数量	1	100	30

表 3：Timewarp 模型超参数

数据集	RealNVP 层	Transformer 层	参数	原子嵌入维度 H
AD	12	6	1×10^8	64
2AA	12	6	1×10^8	64
4AA	16	16	4×10^8	128

ϕ_{in} 和 ϕ_{out} 多层感知机使用 SiLU 作为激活函数，而 Transformer 多层感知机使用 ReLU。请注意，Transformer 多层感知机指的是图 2 中间部分、位于 Transformer 块内部的逐原子多层感知机。不同数据集所用的这些多层感知机的形状各不相同，如表 4 所示。

表 4：Timewarp 多层感知机层大小

数据集	ϕ_{in} 多层感知机	ϕ_{out} 多层感知机	Transformer 多层感知机
AD	$[70, 256, 128]$	$[128, 256, 3]$	$[128, 256, 128]$
2AA	$[70, 256, 128]$	$[128, 256, 3]$	$[128, 256, 128]$
4AA	$[134, 2048, 128]$	$[128, 2048, 3]$	$[128, 2048, 128]$

核自注意力模块中的第一个线性层始终具有形状 $[128, 768]$（在第4节中记为 V），第二个线性层（拼接注意力头 head 的输出后）具有形状 $[768, 128]$。对于所有数据集，Transformer 特征维度 D 均为 128。

完成似然训练后，我们结合第5节中讨论的全部三种损失函数，对模型在AD和2AA数据集上进行微调。对于在4AA数据集上训练的模型，我们未进行微调。我们采用损失函数的加权和，权重详见表5。我们使用

表5：Timewarp 损失加权因子

数据集	$\mathcal{L}_{\text{lik}}(\theta)$	$\mathcal{L}_{\text{acc}}(\theta)$	$\mathcal{L}_{\text{ent}}(\theta)$
AD	0.99	0.01	0.1
2AA	0.9	0.1	0.1
4AA	1	0	0

所有实验均采用 *FusedLamb* 优化器和 *DeepSpeed* 库 [46] 。批量大小和训练时间列于表 6。所有模拟均以 5×10^{-4} 的学习率启动，

表 6：Timewarp 训练参数

数据集 +训练方法	批量大小	A-100 数量	训练时间
丙氨酸二肽 — 似然	256	1	1 周
丙氨酸二肽 — 接受率	64	1	2 天
2AA — 似然	256	4	2 周
2AA — 接受率	256	4	4 天
4AA — 似然	256	4	3 周

每当训练损失进入平台期，学习率就连续减半。

G 计算基础设施

2AA 和 4AA 数据集的训练使用了 4 块 NVIDIA A-100 GPU，AD 数据集的训练则使用了 1 块 NVIDIA A-100 GPU。模型推理以及所有分子动力学模拟中，AD 和 2AA 使用单块 NVIDIA V-100 GPU，4AA 则使用单块 NVIDIA A-100 GPU。