arXiv:2305.04120v2 [q-bio.BM] 6 Dec 2023

用于蛋白质结构生成的潜在扩散模型

Cong Fu[1]*, Keqiang Yan[1]*, Limei Wang[1], Wing Yee Au[2], Michael McThrow[2], Tao Komikado[3], , Koji Maruhashi[3], Kanji Uchino[2], Xiaoning Qian[1], Shuiwang Ji[1] 德克萨斯农工大学，美国德克萨斯州大学城 [2] 富士通美国研究院，美国加利福尼亚州桑尼维尔[3] 富士通研究所，日本神奈川 {congfu, keqiangyan, limei, xqian, sji}@tamu.edu {wau, mmcthrow, komikado.tao, maruhashi.koji, kanji}@fujitsu.com

摘要

蛋白质是复杂的生物分子，在生物体内发挥着各种关键作用。设计和生成新型蛋白质能够为未来众多合成生物学应用铺平道路，包括药物发现。然而，由于蛋白质结构的建模空间十分庞大，这仍然是一项具有挑战性的计算任务。在本研究中，我们提出了一种潜在扩散模型，能够在压缩的潜在空间中灵活捕获天然蛋白质结构的分布，同时降低蛋白质建模的复杂性。具体而言，我们提出了一种等变蛋白质自编码器，将蛋白质嵌入潜在空间，然后使用等变扩散模型学习潜在蛋白质表示的分布。实验结果表明，我们的方法能够以较高的可设计性和效率，有效生成新型蛋白质主链结构。代码将在以下网址公开：
https://github.com/divelab/AIRS/tree/main/OpenProt/LatentDiff。

1 引言

人工智能已成为一种前景广阔的方法，显著推动了各领域的科学研究 [1], ，例如物理模拟 [2, 3], 、量子力学 [4, 5],、材料 [6, 7], 和生物学 [8–13]。发现新型蛋白质 [14–19] 对生物医学至关重要。近年来，研究者不再先生成新型蛋白质序列 [20–27] ，再预测其对应结构，而是鉴于扩散模型在图像和小分子生成方面出色的建模能力与生成质量 [30–34] ，Trippe 等人 [28] 和 Wu 等人 [29] 提出直接使用扩散模型生成蛋白质结构。然而，生成三维蛋白质结构是一项更具挑战性的任务，因为其几何结构复杂，探索空间广阔。此外，随着建模空间增大，训练扩散模型以及从中采样所需的时间和计算资源成本也会显著增加。

已有研究尝试在图像和小分子领域缩小扩散模型的建模空间。稳定扩散 [34] 结合预训练图像自编码器和潜在扩散模型，以缩小大型图像的建模空间。然而，目前还没有适用于三维蛋白质结构的稳健且强大的三维图自编码器和潜在扩散模型。扭转扩散 [33]仅关注扭转角，并使用 RDKit [35] 预测键长和键角，因为小分子中键角和键长的分布范围受到严格限制。但这一假设并不适用于蛋白质结构。

本文通过结合三维图自编码器和潜在三维扩散模型，缩小复杂三维蛋白质结构的扩散建模空间。为此，我们解决了以下挑战：(1) 确保自编码器设计具有旋转等变性；(2) 在解码过程中准确重建三维图中复杂的连接信息；(3) 针对三维蛋白质潜在表示（包括位置和节点潜在表示）开发专门的潜在扩散过程。在接下来的章节中，我们首先回顾以下内容的背景和相关工作：

*贡献相同

C. Fu 等，《用于蛋白质结构生成的 LatentDiffusion 模型》。第二届图学习会议论文集（*LoG 2023*），PMLR 231，线上活动，2023年11月27日至30日。

第2节介绍蛋白质主链结构生成和扩散模型，随后我们将在第 3节详细说明如何应对上述挑战。我们将在第 4节展示所提方法的效率及其生成新型蛋白质主链结构的能力。

2 背景与相关工作

2.1 蛋白质主链结构生成

蛋白质主链生成旨在通过学习真实数据分布来生成新颖的蛋白质主链结构。为此，需要在已知分布（如高斯分布）与高维且稀疏的真实数据分布之间构建映射。由于蛋白质全局几何结构主要由主链决定，蛋白质结构的生成可以简化为生成由氨基酸序列及其对应位置组成的主链。遵循 ProtDiff [28]，我们使用 α碳的位置来表示氨基酸的位置。蛋白质骨架结构由此表示为

$$\mathcal{S} = \{(\boldsymbol{x}_i, a_i)\}_{i=1}^{n}, \tag{1}$$

其中， $\boldsymbol{x}_i \in \mathbb{R}^3$ 表示第 i个氨基酸中 α 碳的三维位置， $a_i \in \{k|1 \leq k \leq 20,\ k \in \mathbb{Z}\}$ 表示对应的氨基酸类型。

以往研究 [28] 表明，与同时建模氨基酸类型和 α 碳位置不同，将整个生成过程分解为两个阶段会更好，即 $p\ \boldsymbol{a}|\boldsymbol{X}\ p\ \boldsymbol{X}$ $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$ $\boldsymbol{a} = [a_1, a_2, \cdots, a_n]^T$ p $\boldsymbol{a}$ $(\boldsymbol{X},\ )$

( ) ( )，其中 ，以及 。具体而言，首先生成 α 碳的位置，再使用预训练的逆折叠模型（如 ProteinMPNN）预测对应的氨基酸类型 [36]。

2.2 去噪扩散概率模型

作为一类强大的生成模型 [37–39]，，去噪扩散概率模型（DDPM） [30] 通过在给定的先验分布与 $p_{\text{data}}(\boldsymbol{z})$ 之间建立双射映射，解决了推导底层数据分布 $p_{\text{data}}(\boldsymbol{z})$ 的贝叶斯逆问题。我们遵循 ScoreSDE [31]采用的惯例，在此回顾 DDPM 的背景知识。为了通过对更简单的先验分布进行采样，实现基于 $p_{\text{data}}(\boldsymbol{z})$ 的忠实生成，模型采用离散马尔可夫链，逐步将输入扩散为随机噪声，即将给定的训练数据映射为随机噪声，例如遵循多元正态（高斯分布）。对于每个训练样本 $\boldsymbol{z}_0 \sim p_{\text{data}}(\boldsymbol{z})$，DDPM 会考虑一系列方差值 $0 < \beta_1, \beta_2, \ldots, \beta_N < 1$，并构建离散马尔可夫链 $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$，其中 $p(\boldsymbol{z}_i|\boldsymbol{z}_{i-1}) = \mathcal{N}(\boldsymbol{z}_i;\ \sqrt{1-\beta_i}\boldsymbol{z}_{i-1}, \beta_i\mathbf{I})$。据此，我们得到 $p(\boldsymbol{z}_i|\boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{z}_i;\ \sqrt{\alpha_i}\boldsymbol{z}_0, (1-\alpha_i)\mathbf{I})$，其中 $\overline{\alpha_i} = \prod_{t=0}^{i}(1-\beta_t)$。因此，可以预先设定一系列噪声尺度，使得 $\alpha_N \to 0$，且 $\boldsymbol{z}_N$ 近似服从 $\mathcal{N}(0, \mathbf{I})$。对于从 $\mathcal{N}(0, \mathbf{I})$ 到 $p_{\text{data}}(\boldsymbol{z})$ 的反向映射，反向马尔可夫链被参数化为 $p_\theta(\boldsymbol{z}_{i-1}|\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i;\ \mu_\theta(\boldsymbol{z}_i, i)\ \beta_i\mathbf{I})$，其中 $\mu_\theta(\boldsymbol{z}_i, i) = \frac{1}{\sqrt{1-\beta_i}}(\boldsymbol{z}_i - \frac{\beta_i}{\sqrt{1-\alpha_i}}\mathrm{s}_\theta(\boldsymbol{z}_i, i))$。反向扩散模型 s_θ 使用重新加权的证据下界（ELBO）进行训练，如下所示

$$\boldsymbol{\theta}^\star = \text{argmin}_{\boldsymbol{\theta}}\mathbb{E}_{t,\boldsymbol{z}_0,\boldsymbol{\sigma}}[\|\boldsymbol{\sigma} - \boldsymbol{s}_{\boldsymbol{\theta}}(\sqrt{\alpha_t}\boldsymbol{z}_0 + \sqrt{1-\alpha_t}\boldsymbol{\sigma}, t)\|^2], \tag{2}$$

其中 $\boldsymbol{\sigma} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$。在训练好 s_θ 后，反向采样过程首先从 $\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 中采样，然后根据估计的反向马尔可夫链，从时间 N 逐步更新至时间 0

$$\boldsymbol{z}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(\boldsymbol{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t)) + \sqrt{\beta_t}\boldsymbol{\sigma}. \tag{3}$$

2.3 相关工作

用于蛋白质结构生成的扩散模型。近期研究 [28, 29, 40–45] 一直在探索使用扩散模型生成新型蛋白质结构，这一探索建立在扩散模型在图像 [30, 31] 和小分子 [32, 33, 46]等其他领域取得成功的基础上。其中，ProtDiff [28] 通过确定α碳的位置来生成蛋白质主链结构，而 FoldingDiff [29] 使用键角和扭转角表示蛋白质主链结构，并采用序列扩散模型生成新的主链结构。Anand 和 Achim [40] 尝试使用三个独立的扩散模型生成完整的蛋白质结构。

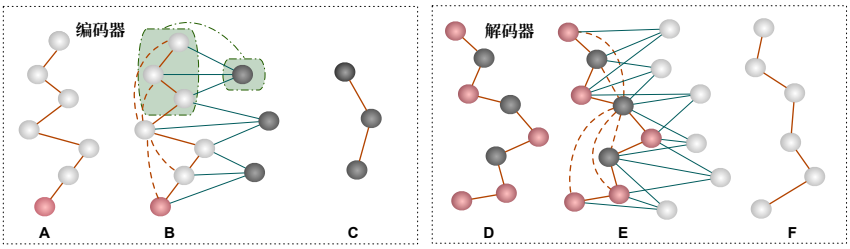


图 1： 蛋白质的自编码器网络结构。步骤 A、B 和 C 表示编码器网络。A. 增广后的输入蛋白质结构（白色）及填充部分（红色节点），类似于图像填充。B. (1) 边构建：在填充后的结构上创建全连接图（为简洁起见，仅显示部分边）；(2) 图扩展：按照 1D-CNN 约定引入具有特定连接关系的新节点（黑色）。C. 压缩后的结构（位于潜在空间中）。步骤 D、E 和 F 表示解码器网络。D. 对潜在结构进行填充以进行上采样（类似于图像转置卷积中的填充操作）。E. 边构建和图扩展与 B 类似。F. 重建的蛋白质链。

依次生成 α 碳位置、氨基酸类型和侧链旋转角度，但联合建模的性能相对较低。此外，Lee 和 Kim [41] 提出对氨基酸残基的二维成对距离矩阵和角度矩阵进行扩散，但仍需使用 Rosetta 最小化进行进一步优化 [47]。

值得注意的是，在我们开发该方法的同时，也涌现出了其他几项能够生成高质量蛋白质的研究。RFdiffusion [42]利用强大的蛋白质结构预测模型 RoseTTAFold [48], , 在许多生成任务上取得了显著成果。RFdiffusion 在蛋白质结构预测任务上对 RoseTTAFold 进行预训练，然后在生成任务上进行微调。但 RFdiffusion 仅展示了使用预训练权重生成蛋白质的有效性。Chroma [43] 使用相关扩散过程，将蛋白质结构转换为随机塌缩聚合物，并通过设计的协方差模型对链约束和回转半径约束进行编码。通过这种方式，Chroma 能够保留蛋白质中的一些基本结构，从而更高效地对目标分布建模。Genie [44] 和 FrameDiff [45] 采用定向参考系对残基建模。Genie 仅考虑 α 碳原子，因此扩散只需作用于原子位置。FrameDiff 会生成完整的主链原子，因此需要同时考虑参考系位置和方向上的扩散。

尽管蛋白质主链结构生成 [28, 29, 40–45], 已取得成功，扩散模型的建模空间仍然十分广阔，因此从扩散模型的训练到采样都需要投入大量时间和计算资源。

缩小蛋白质结构的建模空间。蛋白质结构生成的建模空间可通过多种方式缩小。ProtDiff [28] 仅考虑α碳的位置，而 FoldingDiff [29] 使用键角和扭转角表示蛋白质主链结构，并省略键长，以缩小建模空间。Torsional Diffusion [33] 使用 RDKit 生成的键长和键角，仅对扭转角进行扩散，以生成小分子的构象，但不适用于蛋白质结构。

最近，稳定扩散 [34]在图像领域展现出的出色生成能力引起了广泛关注。通过将预训练的图像自编码器与潜在扩散模型相结合，稳定扩散缩小了大型图像的建模空间，并提升了图像扩散模型的生成能力。然而，蛋白质结构的三维几何图与图像不同，目前尚未提出适用于蛋白质结构的稳健三维等变蛋白质自编码器和三维潜在扩散模型。

3 方法

本节将介绍我们用于生成蛋白质主链结构的 LatentDiff。我们将在第 3.1 节介绍等变蛋白质自编码器的设计，随后在第 3.2 节介绍潜在空间扩散模型。

3.1 等变蛋白质自编码器

我们首先介绍有助于缩小蛋白质设计空间的等变自编码器。为了设计这样的自编码器，我们总结了蛋白质骨架的一些约束条件和独特之处。首先，由于氨基酸序列具有顺序性，蛋白质骨架中的 C_α原子具有固定的顺序。通常，可以通过一维卷积神经网络（CNN）对序列数据进行下采样或上采样。此外，由于 C_α 原子构成的链状结构可以在上采样过程中保留，因此我们无需像传统图自编码器那样重建边连接。其次，尽管蛋白质骨架采用序列表示，但也具有三维几何结构，因此在下采样和上采样阶段需要满足等变性要求。传统 CNN 无法满足这一等变性要求，而图神经网络（GNN）则能够应对这一挑战。基于这些观察，我们提出了一种新颖的等变蛋白质自编码器，同时考虑蛋白质骨架的氨基酸序列和三维图信息。

概述。 在等变蛋白质自编码器中，我们首先对蛋白质进行下采样以缩小其尺寸，再对潜在图进行上采样，以重建原始蛋白质。每个下采样层和上采样层都包含四个步骤，分别是**结构填充**、**边构建**、**图扩展**和**等变消息传递**。前三个步骤用于构建一个图，其中包含输入节点以及在当前层初始化的下采样或上采样节点。消息传递完成后，只保留更新后的下采样或上采样节点，作为下一层的输入，以便进一步执行下采样或上采样操作。接下来，我们将介绍网络输入以及一个下采样层的具体细节。上采样层采用完全相同的步骤，只有结构填充不同；我们也会在结构填充部分对此进行介绍。

网络输入。对于蛋白质骨架结构 $\mathcal{S}$，我们将结构移至零质心，以避免模型捕获平移等变性。随后，我们将蛋白质增强至固定长度 m，以简化网络中的后续操作。因此， m 是我们能够生成的最大蛋白质长度；在本工作中，我们将其设为 128。增强后的蛋白质在图 1.A 中以白色部分表示。具体而言，我们在蛋白质结构末端附加 $m-n$ 个额外节点。每个额外节点的位置均设为零，且节点类型相同。我们将增强蛋白质结构记为 $\mathcal{S}_{\text{aug}} = (\boldsymbol{X}, \boldsymbol{H})$，其中 $\boldsymbol{X} \in \mathbb{R}^{3\times m}$ 和 $\boldsymbol{H} \in \mathbb{R}^{d\times m}$ 分别表示节点位置和节点特征向量。对于 $\boldsymbol{X}$，前 n 列 $\{\boldsymbol{x}_i\}_{i=1}^{n}$ 表示原始蛋白质中所有 C_α 原子的节点位置，最后 $m-n$ 列 $\{\boldsymbol{x}_i\}_{i=n+1}^{m}$ 表示额外节点的零位置。 $\boldsymbol{H}$ 中的每个节点特征向量 $\boldsymbol{h}_i \in \mathbb{R}^d$ 都是一个 d 维的类型嵌入，用于指示相应的节点类型。随后，预处理后的 $\mathcal{S}_{\text{aug}}$ 将作为第一个下采样层的输入。

结构填充。 与图像卷积中的填充类似，在每一层中，我们首先需要对增强蛋白质结构进行填充 $\mathcal{S}_{\text{aug}}$，然后再对结构进行下采样或上采样，以获得所需尺寸的输出。假设我们有 k 个节点经过结构填充。将填充后的结构记为 $\mathcal{S}_{\text{pad}} = (\boldsymbol{X}_{\text{pad}}, \boldsymbol{H}_{\text{pad}})$，其中 $\boldsymbol{X}_{\text{pad}} \in \mathbb{R}^{3\times k}$ 且 $\boldsymbol{H}_{\text{pad}} \in \mathbb{R}^{d\times k}$。如图 1.A 和 D 所示，红色节点是填充节点。对于下采样，我们通过添加与边界节点具有相同节点位置和节点特征的节点，在输入结构的边界处进行填充。例如，在图 1.A 中，红色节点是最后一个白色节点的副本。对于上采样，我们需要进行边界填充和内部填充，这与转置卷积中的图像填充类似。边界填充方式与下采样相同。对于内部填充节点，例如图 1.D 中第二个红色节点，其位置和节点特征初始化为它两侧最近两个节点对应值的平均值。

边构建。结构填充后，我们执行边构建步骤，从填充后的蛋白质结构构建一个图 $\mathcal{S}_{\text{pad}}$。我们可以采用全连接图，以捕捉所有原子对之间的相互作用。如图1.B 所示，构建的完全图中的边以红色表示。为简单起见，我们只展示一个节点的边连接。请注意，此步骤中的边连接方式可以灵活选择。根据经验，我们发现，在下采样过程中仅对非填充结构构建完全图，能获得更好的重建性能。

图扩展。 接下来，在图扩展步骤中，我们首先需要初始化下采样节点，并将它们连接到边构建步骤中构造的图。我们将扩展图表示为$\mathcal{G}_{\text{exp}} = (\boldsymbol{X}_{\text{exp}}, \boldsymbol{H}_{\text{exp}}, \boldsymbol{A}_{\text{exp}})$，其中 $\boldsymbol{X}_{\text{exp}} = [\boldsymbol{X}_{\text{pad}}, \boldsymbol{X}_{\text{down}}] \in \mathbb{R}^{3\times(k+\frac{m}{2})}$、 $\boldsymbol{H}_{\text{exp}} = [\boldsymbol{H}_{\text{pad}}, \boldsymbol{H}_{\text{down}}] \in \mathbb{R}^{d\times(k+\frac{m}{2})}$ 和 $\boldsymbol{A}_{\text{exp}} \in \mathbb{R}^{(k+\frac{m}{2})\times(k+\frac{m}{2})}$。具体而言，我们创建一组具有如下位置的新节点

$\boldsymbol{X}_{\text{down}} \in \mathbb{R}^{3\times\frac{m}{2}}$和节点特征向量 $\boldsymbol{H}_{\text{down}} \in \mathbb{R}^{d\times\frac{m}{2}}$，它们表示下采样结构。下采样结构与增强蛋白质结构之间的边连接按照一维卷积神经网络的惯例创建。具体来说，只有卷积核大小的窗口内的节点才会与新节点相连。例如，如图 1.B 所示，绿色区域表示大小为 3 的卷积核，第一个黑色节点与绿色区域内的前三个白色节点相连。每个新节点的位置和节点特征均初始化为其所连接节点的平均值。

SE(3)等变消息传递。 蛋白质只含有右手性α螺旋，因此网络不应对反射保持等变。于是，我们使用SE(3)等变图神经网络在扩展图上执行消息传递 $\mathcal{G}_{\text{exp}}$，以更新下采样节点。我们改编了Schneuing等人 [49], 提出的网络架构，他们通过在坐标更新步骤中添加一个额外的叉积项，对E(n)等变图神经网络（EGNN） [50] 进行了改进。这样一来，网络便能感知反射。形式上，

$$\hat{\boldsymbol{X}}_{\text{exp}}, \hat{\boldsymbol{H}}_{\text{exp}} = \text{EGNN}_{SE(3)}[\boldsymbol{X}_{\text{exp}}, \boldsymbol{H}_{\text{exp}}], \tag{4}$$

其中，$\hat{}\,\boldsymbol{X}_{\text{exp}} = [\,\hat{}\boldsymbol{X},\ \hat{}\boldsymbol{X}_{\text{down}}]$ 和$\hat{}\,\boldsymbol{H}_{\text{exp}} = [\,\hat{}\boldsymbol{H},\ \hat{}\boldsymbol{H}_{\text{down}}]$。$\text{EGNN}^{SE(3)}$ 包含 L 个等变卷积层（EGCL）。每一层都会更新位置和特征，使得 $\boldsymbol{x}_i^{l+1}$， $\boldsymbol{h}_i^{l+1}$ =EGCL$[\boldsymbol{x}_i^l, \boldsymbol{h}_i^l]$，其定义如下:

$$\boldsymbol{m}_{ij} = \phi_e(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, d_{ij}^2, a_{ij}), \tag{5}$$

$$\boldsymbol{h}_i^{l+1} = \phi_h(\boldsymbol{h}_i^l, \sum_{j\neq i} \tilde{e}_{ij}\boldsymbol{m}_{ij}), \tag{6}$$

$$\boldsymbol{x}_i^{l+1} = \boldsymbol{x}_i^l + \sum_{j\neq i} \frac{\boldsymbol{x}_i^l - \boldsymbol{x}_j^l}{d_{ij}+1}\phi_x(\boldsymbol{m}_{ij}) + \frac{\left(\boldsymbol{x}_i^l - \overline{\boldsymbol{x}}^l\right)\times\left(\boldsymbol{x}_j^l - \overline{\boldsymbol{x}}^l\right)}{\left\|\left(\boldsymbol{x}_i^l - \overline{\boldsymbol{x}}^l\right)\times\left(\boldsymbol{x}_j^l - \overline{\boldsymbol{x}}^l\right)\right\|+1}\phi_x^{\times}\left(\boldsymbol{m}_{ij}\right), \tag{7}$$

其中，$d_{ij} = \left\|\boldsymbol{x}_i^l - \boldsymbol{x}_j^l\right\|_2$ 表示节点 i 与 j之间的欧几里得距离，a_{ij} =MLP($[\boldsymbol{h}_i^l, \boldsymbol{h}_j^l]$) 是边 $(i,\ j)$ 的边特征。 $\boldsymbol{x}$ 表示所有节点的质心。¯为提高数值稳定性，可以选择使用 $d_{ij}+1$ 对节点距离进行归一化。遵循 Hooge- boom 等人的做法 [46], ，我们使用注意力机制 $\tilde{e}_{ij} = \phi_{inf}(\boldsymbol{m}_{ij})$ 来推断边的软估计。

消息传递后，我们只保留更新后的下采样结构($\hat{}\,\boldsymbol{X}_{\text{down}}$，$\hat{}\,\boldsymbol{H}_{\text{down}}$)作为下一层的输入，如图1C 所示。在解码器的上采样阶段，我们执行上述相同的四个步骤。将输入的增强蛋白质上采样至原始大小后，我们得到一个重建结构，其中每个节点都具有位置和节点嵌入。接着，我们使用一个多层感知机处理最终的节点嵌入，并预测重建节点是否属于增强节点类型，具体内容将在下文的训练损失部分介绍。然后，我们使用另一个多层感知机预测每个节点的氨基酸类型。

训练损失。 自编码器的重建损失由六部分组成。首先，我们在二元分类任务上使用交叉熵损失 $\mathcal{L}_{\text{aug}}$，以确定每个重建节点是否为不属于原始蛋白质的增强节点。接下来，我们使用另一项交叉熵损失 $\mathcal{L}_{\text{aa}}$，预测每个节点的氨基酸类型。然后，我们计算重建蛋白质与真实值之间每个未增强节点的位置的平均绝对误差（MAE），并将其记为 $\mathcal{L}_{\text{pos}}$。除了这三项损失外，为了进一步考虑蛋白质的二级结构重建，我们还纳入了在未增强节点上计算的边距离损失 $\mathcal{L}_{\text{dist}}$ 和扭转角损失 $\mathcal{L}_{\text{tor}}$。具体而言，边距离是每两个连续的 C_α 原子之间的欧几里得距离，而扭转角是由四个连续的 C_α 原子形成的两个平面之间的夹角。为避免潜在节点嵌入的方差任意增大，我们使用较小的 KL 散度损失 $\mathcal{L}_{\text{reg}}$ 对潜在节点嵌入进行正则化，这与变分自编码器类似。因此，总损失是这些单项损失的加权和。形式化地，

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{aug}} + \mathcal{L}_{\text{aa}} + \mathcal{L}_{\text{pos}} + w_1 * \mathcal{L}_{\text{dist}} + w_2 * \mathcal{L}_{\text{tor}} + w_3 * \mathcal{L}_{\text{reg}}, \tag{8}$$

其中，w_1、w_2和 w_3 分别是用于控制边距离损失、扭转角损失和正则化损失的相对权重。我们希望网络首先优化每个节点的绝对位置，然后再调整边距离和扭转角，因此将 w_1 和 w_2 设为 0.5。此外，我们希望自编码器具有良好的重建性能，因此仅使用非常小的正则化项，并将 w_3 设为 $1e^{-4}$。

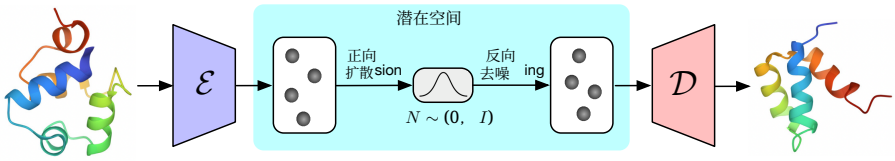


图 2： LatentDiff 的流程。编码器 $\mathcal{E}$ 和解码器 $\mathcal{D}$ 通过第 3.1 节介绍的等变蛋白质自编码器进行预训练，并在训练潜在扩散模型期间固定其参数。蛋白质结构通过编码器 $\mathcal{E}$编码为潜在表示。随后，潜在表示逐渐受到扰动，变为高斯噪声。在生成过程中，我们首先对高斯噪声进行采样，并使用学习得到的去噪网络在潜在空间中生成蛋白质表示。然后，解码器 $\mathcal{D}$ 将潜在表示解码为蛋白质结构。

3.2 潜在扩散

由于提取出的蛋白质主链结构的潜在表示（$\boldsymbol{X}_{\text{down}}$、$\boldsymbol{H}_{\text{down}}$）由三维欧几里得位置组成，与图像和文本有所不同，因此对其进行建模面临独特的挑战。在本节中，我们首先解释所需分布的 SE(3) 不变性，然后详细描述满足该性质的潜在扩散过程，以用于蛋白质主链生成任务。在本节中，p_{data}、p_{model} 和 p_θ 分别表示底层数据分布、整个模型框架的输出分布以及潜在扩散模型中的潜在分布。

分布的 SE(3) 不变性。对于给定的蛋白质骨架结构（$\boldsymbol{X}, \boldsymbol{H}$），我们希望学习到的数据分布具有 SE(3) 不变性：$p_{\text{data}}(\boldsymbol{X}, \boldsymbol{H}) = p_{\text{data}}(R\boldsymbol{X}+b, \boldsymbol{H})$，因为经过 SE(3) 变换后，几何三维结构保持不变；其中，$R \in \mathbb{R}^{3\times3}, |R| = 1$ 仅描述旋转变换，$b \in \mathbb{R}^3$ 表示三维空间中的平移。由于我们的蛋白质自编码器具有平移不变性（如第 3.1 节所述），$p_{\text{model}}(\boldsymbol{X}, \boldsymbol{H}) = p_{\text{model}}(\boldsymbol{X}+b, \boldsymbol{H})$ 自然成立。因此，对于潜在扩散过程，需要满足分布的旋转不变性 $p_{\text{model}}(\boldsymbol{X}, \boldsymbol{H}) = p_{\text{model}}(R\boldsymbol{X}, \boldsymbol{H})$。

在我们的方法中，我们提出将蛋白质主链结构的生成分解为两个阶段，包括（1）蛋白质潜在表示生成和（2）潜在表示解码。模型分布可定义为 $p_{\text{model}}(\boldsymbol{X},\ \boldsymbol{H}) = p_{\text{decoder}}(\boldsymbol{X},\ \boldsymbol{H}|\boldsymbol{X}_{\text{down}},\ \boldsymbol{H}_{\text{down}})p_\theta(\boldsymbol{X}_{\text{down}},\ \boldsymbol{H}_{\text{down}})$。鉴于解码过程具有 SE(3) 等变性且为确定性过程，若潜在扩散模型的$_\theta$ 满足 $p_\theta(\boldsymbol{X}_{\text{down}},\ \boldsymbol{H}_{\text{down}}) = p_\theta(R\boldsymbol{X}_{\text{down}}+b,\ \boldsymbol{H}_{\text{down}})$，则分布的 SE(3) 不变性 $p_{\text{model}}(\boldsymbol{X},\ \boldsymbol{H}) = p_{\text{model}}(R\boldsymbol{X}+b,\ \boldsymbol{H})$也能得到保证。

$p_\theta(\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}}) = p_\theta(R\boldsymbol{X}_{\text{down}}+b, \boldsymbol{H}_{\text{down}})$这一挑战可通过以下方式解决：（1）为 $\boldsymbol{X}$建模零均值几何分布；（2）使用高维高斯分布作为先验分布；（3）采用旋转等变反向扩散过程 [32, 46]。具体而言，通过将 $\boldsymbol{X}$的中心位置归零，忽略三维空间中平移变换的影响。此外，借助各向同性高维高斯先验，我们有$p_\theta(\boldsymbol{X}_T, \boldsymbol{H}_T) = p_\theta(R\boldsymbol{X}_T, \boldsymbol{H}_T)$。旋转等变反向扩散过程进一步保证，对于任意时间 t，$p_\theta(\boldsymbol{X}_t, \boldsymbol{H}_t) = p_\theta(R\boldsymbol{X}_t, \boldsymbol{H}_t)$，证明见附录 A.1。

旋转分布不变潜在扩散。 基于上述考虑，我们针对提取的蛋白质主链潜在特征，提出了旋转分布不变的潜在正向和反向扩散过程（$\boldsymbol{X}_{\text{down}}$，$\boldsymbol{H}_{\text{down}}$）。该方法基于 EDM [46] 实现，并进行了调整以支持潜在扩散过程。具体而言，我们生成带有位置和潜在节点特征的潜在三维点，因此无需在反向扩散的最后一步解码节点类型。此外，由于蛋白质结构具有天然的顺序，我们加入正弦位置编码特征，以提供序列顺序信息。最重要的是，与第 3.1 节类似，我们还修改了 EDM 中的消息传递，使其具备 SE(3) 等变性。蛋白质潜在扩散的流程如图 2 所示。在正向过程中，输入潜在表示（$\boldsymbol{X}_{\text{down}}$，$\boldsymbol{H}_{\text{down}}$）通过一系列噪声尺度 $0 < \beta_1$，β_2、……、$\beta_N < 1$ 逐渐扩散为随机噪声，过程如下

$$\boldsymbol{X}_i = \sqrt{1-\beta_i}\boldsymbol{X}_{i-1} + \sqrt{\beta_i}\boldsymbol{\sigma}_{\boldsymbol{X}}, \qquad (9) \qquad \boldsymbol{H}_i = \sqrt{1-\beta_i}\boldsymbol{H}_{i-1} + \sqrt{\beta_i}\boldsymbol{\sigma}_{\boldsymbol{H}}, \qquad (10)$$

其中 $\boldsymbol{\sigma_H} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，且 $\boldsymbol{\sigma_X}$ 首先从 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 中采样，然后遵循 Hoogeboom等人 [46]的方法，根据相应的中心位置进行约简。闭式正向过程可写为

$$\boldsymbol{X}_t = \sqrt{\alpha_t}\boldsymbol{X}_{\text{down}} + \sqrt{1-\alpha_t}\boldsymbol{\sigma_X}, \tag{11}$$

$$\boldsymbol{H}_t = \sqrt{\alpha_t}\boldsymbol{H}_{\text{down}} + \sqrt{1-\alpha_t}\boldsymbol{\sigma_H}, \tag{12}$$

其中 $\alpha_t = \prod_{i=0}^{t}(1-\beta_i)$。由于 α_t 是一个标量值，因此有 $p_t(\boldsymbol{X}_t, \boldsymbol{H}_t) = p(\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}})p(\boldsymbol{\sigma_X}, \boldsymbol{\sigma_H})$，其中 p_t 是时间 t 下的数据分布，而 $p(\boldsymbol{\sigma_X}, \boldsymbol{\sigma_H}) = p(\boldsymbol{\sigma_X})p(\boldsymbol{\sigma_H})$ 表示相应的多元高斯分布。可以看出，$p_t(\boldsymbol{X}_t, \boldsymbol{H}_t) = p_t(R\boldsymbol{X}_t, \boldsymbol{H}_t)$，因为 $p(\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}})p(\boldsymbol{\sigma_X}, \boldsymbol{\sigma_H}) = p(R\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}})p(\boldsymbol{\sigma_X}, \boldsymbol{\sigma_H})$。因此，前向扩散过程满足旋转分布不变性。

对于反向扩散过程，构造如下反向马尔可夫链

$$(\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}) = \frac{1}{\sqrt{1-\beta_t}}\boldsymbol{\mu_t} + \sqrt{\beta_t}(\boldsymbol{\sigma_X}, \boldsymbol{\sigma_H}), \tag{13}$$

$$\boldsymbol{\mu_t} = (\boldsymbol{X}_t, \boldsymbol{H}_t) - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{s_\theta}(\boldsymbol{X}_t, \boldsymbol{H}_t, t), \tag{14}$$

其中 $\boldsymbol{s_\theta}$ 是一个基于等变图神经网络（EGNN）的 SE(3) 版本实现的旋转等变网络 [49, 50]。

训练损失。反向扩散模型 $\boldsymbol{s_\theta}$ 按照 ProtDiff [28] 和 DDPM [30]的方法，使用重新加权的证据下界（ELBO）进行训练，如下所示

$$\boldsymbol{\theta}^\star = \text{argmin}_{\boldsymbol{\theta}}\mathbb{E}_{t,(\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}}),\boldsymbol{\sigma}}[\|\boldsymbol{\delta}\|^2], \tag{15}$$

$$\boldsymbol{\delta} = \boldsymbol{\sigma} - \boldsymbol{s_\theta}(\sqrt{\alpha_t}(\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}}) + \sqrt{1-\alpha_t}\boldsymbol{\sigma}, t), \tag{16}$$

其中 $\boldsymbol{\sigma} = (\boldsymbol{\sigma_X}, \boldsymbol{\sigma_H})$。

4 实验

我们通过实验展示了所提出的方法在生成蛋白质主链结构方面的有效性和效率。完整的生成过程见附录 A.4。在第 4.1 节中，我们首先介绍从现有蛋白质数据库中整理的数据集以及基线模型。在第 4.2–第 4.4 节中，我们展示预训练自编码器的重建性能、生成的蛋白质的可设计性，以及 LatentDiff 的并行采样效率。此外，我们还在附录 A.5、A.6、A.9 和附录 A.8 中分别补充了有关二级结构、多样性、生成的蛋白质的结构分布以及结构—序列协同设计的实验。在附录 A.3 中，我们介绍自编码器和潜在扩散模型的训练细节。

4.1 实验设置

数据集。我们从 AlphaFold 蛋白质结构数据库（AlphaFold DB）中的蛋白质数据库（PDB）和 Swiss-Prot 数据中整理数据集 [51, 52]。有关数据集的详细信息，请参见附录 A.2。

基线方法。为了评估我们提出的方法，我们将其与三种蛋白质生成方法进行比较：ProtDiff [28], 、FoldingDiff [29], 和 FrameDiff [45]。前两种方法发表于我们开始开发自己方法之前，而 FrameDiff 是一种较新的蛋白质主链生成方法。

4.2 自编码器重建

本节展示蛋白质自编码器的重建性能。我们比较了采用不同下采样因子 $f = \{2, 4, 8\}$ 的自编码器，并将其记为 $auto-f$。

指标。首先，我们评估增强节点和未增强节点的分类准确率（增强准确率），以及氨基酸类型分类的准确率（残基准确率）。此外，我们采用以下三项几何评估。我们使用均方根偏差（RMSD）进行比较。

表 1：采用不同下采样因子的自编码器性能。↑（↓）表示数值越高（越低），性能越好。

因子	RMSD（Å）↓	增强准确率（%）↑	残基准确率 (%)↑	边稳定性 (%) ↑	扭转角平均绝对误差 (rad) ↓
2	0.5280	100	99	95.29	0.4361
4	1.2755	100	98	70.99	0.8951
8	2.2772	100	45	59.97	1.1903

重建的 C_α 原子与真实值之间的绝对位置误差。此外，我们还测量边稳定性，即统计处于 [3.65Å 至 3.95Å] 范围内的 $C_\alpha - C_\alpha$ 距离所占的比例。选择此范围的原因是，真实值中 99 % 的 $C_\alpha - C_\alpha$ 距离都在此范围内。我们还计算扭转角的平均绝对误差（MAE）。请注意，所有几何评估均在原始蛋白质骨架上进行，不考虑增强节点。

在表 1中，我们总结了采用不同下采样因子的蛋白质自编码器在这五项指标上的结果。为了缩小蛋白质的建模空间，让扩散模型更容易学习潜在分布，我们倾向于使用较大的下采样因子；但与此同时，要取得良好的重建结果也会更加困难。我们可以看到$auto-8$ 的重建性能最差，因为自编码器压缩的信息过多。尽管 $auto-2$ 在三种设置中表现最佳，但潜在空间中的节点数量仍然相对较多。因此，为了平衡计算开销与重建性能，我们最终选择 $auto-4$ 作为预训练模型，用于生成潜在空间数据并解码蛋白质骨架。

4.3 计算机模拟评估

对于生成的蛋白质结构，我们需要评估其可设计性，即能否构建出可折叠成目标主链结构的氨基酸序列。最可靠且理想的评估方式是通过湿实验进行检验，但这通常需要大量资源，且难以实现。因此，我们采用计算机评估作为替代方案。

具体而言，对于一个生成的主链结构，我们首先使用逆折叠模型 ProteinMPNN [36], 预测八条可能折叠成该主链结构的氨基酸序列。随后，使用 OmegaFold [53]预测每条氨基酸序列的折叠结构。接下来，我们采用 TMalign [54] 计算生成的主链结构与每个由 OmegaFold 预测的主链结构之间的相似度，并计算 TM评分以量化相似度。

表 2： scTM 分数为 > 0.5 的生成蛋白质所占百分比。与 FoldingDiff 和 ProtD- iff 一致，结果按短链（50–70）和长链（70–128）类别展示。

方法	50–70	70–128	50–128
ProtDiff	17.1%	8.9%	11.8%
FoldingDiff	27.1%	9.4%	14.2%
FrameDiff	86.6%	87.7%	87.4%
LatentDiff	64.7%	82.8%	66.9%

这八个分数中的最大 TM-score 称为自一致性 TM-score（scTM）。如果 scTM 分数大于 0.5，则认为两个主链结构具有相同的折叠，且该生成的主链结构可用于设计。

与以往工作类似， [28, 29], 我们生成了 780 个长度各异、介于 50 至 128 之间的主链结构，并使用 scTM 分数对其进行评估，其中 ProteinMPNN 的采样温度为 0.1。与 FoldingDiff、ProtDiff 和 FrameDiff 的比较结果见表 2。按照 ProtDiff 的做法，我们将生成的蛋白质进一步分为短链（50–70）和长链（70–128）两类。对于我们的 LatentDiff，$66。9\%$ 生成结构的 scTM 分数 > 0.5，显著优于 FoldingDiff（$14。2\%$）和 ProtDiff（$11。8\%$）。与 FrameDiff 等较新的工作相比，尽管 LatentDiff 的可设计性表现较差，但其采样效率仍具优势，如表 3所示。效率比较的详情见第 4.4节。我们还使用 PyMOL 可视化了一些示例主链及 OmegaFold 预测的主链结构，见图 3。

4.4 并行采样效率比较

在本节中，我们展示了我们方法的并行采样效率。扩散模型通常需要执行数千个反向步骤才能生成单个数据点，而购彩平台

表3： 潜在空间与蛋白质空间中扩散模型的采样效率比较。[1]LatentDiff-P 表示不使用蛋白质自编码器，而是直接在蛋白质空间中进行扩散。采样 1000 个蛋白质以计算采样时间。

方法	参数	蛋白质长度	潜在节点	扩散步数	时间（小时）	速度（秒/样本）
ProtDiff	1.9M	128	N/A	1000	1.9	6.85
FrameDiff	17.4M	128	N/A	500	16.6	60
RFdiffusion	59.8M	128	N/A	200	46.6	168
LatentDiff-P	2.9M	128	N/A	1000	3.9	14.15
LatentDiff	2.9M	128	32	1000	0.25	0.93
LatentDiff	2.9M	128	32	2000	0.51	1.84
LatentDiff	2.9M	256	64	1000	0.95	3.42

每个反向步骤中的数据大小必须相同。因此，生成过程非常耗时且计算成本高昂，尤其是在扩散模型的建模空间较大时。这限制了在计算资源有限的情况下进行高效并行采样。

在潜在空间中进行生成可以降低内存使用量和计算复杂度，因为潜在空间远小于蛋白质空间，从而提高生成吞吐量。我们之所以从并行采样的角度比较效率，是因为筛选过程中需要对大量蛋白质进行采样，因此需要高吞吐量的采样。从这个意义上说，在潜在空间中采样显著提高了效率。在效率比较中，我们使用单块 NVIDIA 2080Ti GPU 对 1000 个蛋白质进行采样，并在表3中汇总结果。为排除建模空间差异以外的因素，我们还与未进行下采样的 LatentDiff（称为 LatentDiff-P）进行比较。对于我们的模型，解码器的处理时间比潜在扩散模型的处理时间少几个数量级，因此不计入解码器时间。从结果可以看出，在蛋白质空间中生成 1000 个蛋白质结构所需的时间为

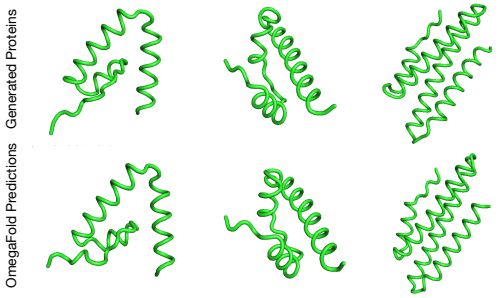


图3： 部分 scTM > 0.5 生成结构样例。第一行展示我们生成的主链，第二行展示 OmegaFold 根据预测的氨基酸序列预测出的主链结构。

约需 3.9 小时，而在潜在空间中生成后再映射到蛋白质空间，仅需约 15 分钟。此外，我们还将效率与 FrameDiff 和 RFdiffusion 进行了比较，生成速度分别快约 $64\times$ 和 $180\times$ 。[1] 尽管与近期最先进的方法相比，性能仍有待进一步提升，但在降维后的建模空间中进行扩散这一思路，已展现出实际应用潜力。LatentDiff 的采样时间随扩散步数线性增长，因为扩散步数是按顺序执行的。此外，由于我们在扩散模型中使用了全连接图，增加潜在节点会使内存消耗和计算复杂度呈二次增长。因此，采样吞吐量会降低，并取决于 GPU内存和计算能力；吞吐量受限于最先达到上限的资源。

5 结论

在这项工作中，我们提出了 LatentDiff，这是一种用于蛋白质主链结构生成的三维潜在扩散框架。为缩小蛋白质结构的建模空间，LatentDiff 使用预训练等变三维自编码器将蛋白质骨架转换到更紧凑的潜在空间，并通过等变潜在扩散模型对潜在分布进行建模。全面的实验结果表明，LatentDiff 能够有效且高效地生成可设计的蛋白质主链结构。

[1]请注意，FrameDiff 和 RFdiffusion 会生成完整的骨架原子，而 ProtDiff 和 LatentDiff 会生成 C_α 原子。

参考文献

[1] Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, Keir Adams, Maurice Weiler, Xiner Li, Tianfan Fu, Yucheng Wang, Haiyang Yu, YuQing Xie, Xiang Fu, Alex Strasser, Shenglong Xu, Yi Liu, Yuanqi Du, Alexandra Saxton, Hongyi Ling, Hannah Lawrence, Hannes Stärk, Shurui Gui, Carl Edwards, Nicholas Gao, Adriana Ladera, Tailin Wu, Elyssa F. Hofgard, Aria Mansouri Tehrani, Rui Wang, Ameya Daigavane, Montgomery Bohde, Jerry Kurtin, Qian Huang, Tuong Phung, Minkai Xu, Chaitanya K. Joshi, Simon V. Mathis, Kamyar Azizzadenesheli, Ada Fang, Alán Aspuru-Guzik, Erik Bekkers, Michael Bronstein, Marinka Zitnik, Anima Anandkumar, Stefano Ermon, Pietro Liò, Rose Yu, Stephan Günnemann, Jure Leskovec, Heng Ji, Jimeng Sun, Regina Barzilay, Tommi Jaakkola, Connor W. Coley, Xiaoning Qian, Xiaofeng Qian, Tess Smidt, and Shuiwang Ji。《量子、原子尺度和连续介质系统的科学人工智能》，2023年。1[2] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia。利用图网络学习模拟复杂物理。在国际机器学习会议论文集，第8459–8468页。PMLR，2020年。1[3] Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji。用于偏微分方程的群等变傅里叶神经算子。arXiv 预印本 arXiv:2306.05697，2023年。1[4] Dmitrii Kochkov, Tobias Pfaff, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Bryan K Clark。利用图网络学习量子哈密顿量的基态。arXiv 预印本 arXiv:2110.06390，2021年。1[5] Cong Fu, Xuan Zhang, Huixin Zhang, Hongyi Ling, Shenglong Xu, and Shuiwang Ji。用于学习量子多体系统基态的晶格卷积网络。arXiv 预印本 arXiv:2206.07370，2022年。1[6] Janet R McMillan, Oliver G Hayes, Peter H Winegar, and Chad A Mirkin。利用DNA进行蛋白质材料工程。《化学研究述评》，52(7):1939–1948，2019年。1[7] Keqiang Yan, Yi Liu, Yuchao Lin, and Shuiwang Ji。用于晶体材料性质预测的周期图变换器。在第36届神经信息处理系统年会，2022年。1[8] Meng Liu, Cong Fu, Xuan Zhang, Limei Wang, Yaochen Xie, Hao Yuan, Youzhi Luo, Zhao Xu, Shenglong Xu, and Shuiwang Ji。通过更深层的二维和三维图网络快速预测量子性质。arXiv 预印本 arXiv:2106.08551，2021年。1[9] Meng Liu, Youzhi Luo, Limei Wang, Yaochen Xie, Hao Yuan, Shurui Gui, Haiyang Yu, Zhao Xu, Jingtun Zhang, Yi Liu, Keqiang Yan, Haoran Liu, Cong Fu, Bora M Oztekin, Xuan Zhang, and Shuiwang Ji。DIG：用于探索图深度学习研究的即用型库。《机器学习研究杂志》，22(240):1–9，2021年。

[10] Meng Liu、Youzhi Luo、Kanji Uchino、Koji Maruhashi 和 Shuiwang Ji。为靶蛋白结合生成三维分子。载于第*39*届国际机器学习会议论文集，第13912–13924页，2022年。[11] Yi Liu、Limei Wang、Meng Liu、Yuchao Lin、Xuan Zhang、Bora Oztekin 和 Shuiwang Ji。用于三维分子图的球面消息传递。载于国际学习表征会议，2022年。[12] Limei Wang、Yi Liu、Yuchao Lin、Haoran Liu 和 Shuiwang Ji。ComENet：面向三维分子图的完整且高效的消息传递。载于第*36*届神经信息处理系统年会，2022年。

[13] Zhengyang Wang、Meng Liu、Youzhi Luo、Zhao Xu、Yaochen Xie、Limei Wang、Lei Cai、Qi Qi、Zhuoning Yuan、Tianbao Yang 和 Shuiwang Ji。用于分子性质预测和药物发现的高级图神经网络与序列神经网络。《生物信息学》，38(9):2579–2586，2022年。1

[14] Namrata Anand 和 Possu Huang。蛋白质结构的生成建模。载于 S. Bengio、H. Wallach、H. Larochelle、K. Grauman、N. Cesa-Bianchi 和 R. Garnett 编，神经信息处理系统进展，第 31 卷。Curran Associates, Inc.，2018年。1

[15] Raphael R Eguchi、Christian A Choe 和 Po-Ssu Huang。Ig-VAE：通过直接生成三维坐标对蛋白质结构进行生成建模。PLoS计算生物学，18(6)：e1010271，2022年。[16] Namrata Anand、Raphael Eguchi 和 Po-Ssu Huang。完全可微的全原子蛋白质骨架生成，2019年。[17] Sari Sabban 和 Mikhail Markovsky。RamaNet：利用长短期记忆生成神经网络进行计算式从头螺旋蛋白质骨架设计。bioRxiv，第 671552 页，2020年。[18] Shitong Luo、Yufeng Su、Xingang Peng、Sheng Wang、Jian Peng 和 Jianzhu Ma。利用基于扩散的生成模型进行抗原特异性抗体设计与优化。bioRxiv，2022年。[19] Chence Shi、Chuanrui Wang、Jiarui Lu、Bozitao Zhong 和 Jian Tang。利用等变平移进行蛋白质序列与结构协同设计。arXiv预印本 arXiv:2210.08761，2022年。 1[20] Zachary Wu、Kadina E Johnston、Frances H Arnold 和 Kevin K Yang。利用深度生成模型进行蛋白质序列设计。《化学生物学当前观点》，65：18–27，2021年。 1[21] Ivan Anishchenko、Samuel J Pellock、Tamuka M Chidyausiku、Theresa A Ramelot、Sergey Ovchinnikov、Jingzhou Hao、Khushboo Bafna、Christoffer Norn、Alex Kang、Asim K Bera 等。从头蛋白质设计中的深度网络幻觉。《自然》，600(7889)：547–552，2021年。[22] Noelia Ferruz、Steffen Schmidt 和 Birte Höcker。ProtGPT2：用于蛋白质设计的深度无监督语言模型。《自然通讯》，13(1)：4348，2022年。[23] Donatas Repecka、Vykintas Jauniskis、Laurynas Karpus、Elzbieta Rembeza、Irmantas Rokaitis、Jan Zrimec、Simona Poviloniene、Audrius Laurynenas、Sandra Viknander、Wissam Abuajwa 等。利用生成对抗网络扩展功能性蛋白质序列空间。《自然·机器智能》，3(4)：324–333，2021年。[24] Alex Hawkins-Hooker、Florence Depardieu、Sebastien Baur、Guillaume Couairon、Arthur Chen 和 David Bikard。利用变分自编码器生成具有功能的蛋白质变体。PLoS计算生物学，17(2)：e1008736，2021年。[25] Ali Madani、Bryan McCann、Nikhil Naik、Nitish Shirish Keskar、Namrata Anand、Raphael R Eguchi、Po-Ssu Huang 和 Richard Socher。ProGen：用于蛋白质生成的语言建模。arXiv预印本 arXiv:2004.03497，2020年。[26] Erik Nijkamp、Jeffrey Ruffolo、Eli N Weinstein、Nikhil Naik 和 Ali Madani。ProGen2：探索蛋白质语言模型的边界。arXiv预印本 arXiv:2206.13517，2022年。[27] Mostafa Karimi、Shaowen Zhu、Yue Cao 和 Yang Shen。利用引导式条件Wasserstein生成对抗网络为新型折叠进行从头蛋白质设计。《化学信息与建模杂志》，60(12)：5667–5681，2020年。 1[28] Brian L Trippe、Jason Yim、Doug Tischer、Tamara Broderick、David Baker、Regina Barzilay 和 Tommi Jaakkola。针对基序支架构建问题的三维蛋白质骨架扩散概率建模。arXiv预印本 arXiv:2206.04119，2022年。 1, 2, 3, 7, 8[29] Kevin E Wu、Kevin K Yang、Rianne van den Berg、James Y Zou、Alex X Lu 和 Ava P Amini。通过折叠扩散生成蛋白质结构。arXiv预印本 arXiv:2209.15611，2022年。 1, 2, 3, 7, 8[30] Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型。《神经信息处理系统进展》，33：6840–6851，2020年。 1, 2, 7[31] Yang Song、Jascha Sohl-Dickstein、Diederik P Kingma、Abhishek Kumar、Stefano Ermon 和 Ben Poole。通过随机微分方程进行基于得分的生成建模。载于《国际学习表征会议》，2020年。 2[32] 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》，2021年。 2, 6, 14[33] Bowen Jing、Gabriele Corso、Regina Barzilay 和 Tommi S Jaakkola。用于分子构象生成的扭转扩散。载于 ICLR2022 药物发现机器学习会议，2022年。 1, 2, 3

[34] Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser 和 Björn Ommer。利用潜在扩散模型进行高分辨率图像合成。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第10684–10695页，2022年。1, 3[35] Greg Landrum 等。RDKit：用于化学信息学、计算化学和预测建模的软件套件。1[36] Justas Dauparas、Ivan Anishchenko、Nathaniel Bennett、Hua Bai、Robert J Ragotte、Lukas F Milles、Basile IM Wicky、Alexis Courbet、Rob J de Haas、Neville Bethel 等。使用 ProteinMPNN 进行基于稳健深度学习的蛋白质序列设计。《科学》，378(6615):49–56，2022年。2, 8, 15[37] Youzhi Luo、Keqiang Yan 和 Shuiwang Ji。GraphDF：用于分子图生成的离散流模型。载于第38届国际机器学习会议论文集，第7192–7203页，2021年。2[38] Meng Liu、Keqiang Yan、Bora Oztekin 和 Shuiwang Ji。GraphEBM：利用基于能量的模型生成分子图。载于基于能量的模型研讨会—ICLR 2021，2021年。[39] Youzhi Luo 和 Shuiwang Ji。用于从头生成三维分子几何结构的自回归流模型。载于国际学习表征会议，2022年。2[40] Namrata Anand 和 Tudor Achim。利用等变去噪扩散概率模型生成蛋白质结构和序列。arXiv预印本 arXiv:2205.15019，2022年。2, 3[41] Jin Sub Lee 和 Philip M Kim。ProteinSGM：用于从头蛋白质设计的基于得分的生成建模。bioRxiv，2022年。3[42] Joseph L Watson、David Juergens、Nathaniel R Bennett、Brian L Trippe、Jason Yim、Helen E Eisenach、Woody Ahern、Andrew J Borst、Robert J Ragotte、Lukas F Milles 等。整合结构预测网络和扩散生成模型，实现广泛适用且准确的蛋白质设计。bioRxiv，第2022–12页，2022年。3[43] John Ingraham、Max Baranov、Zak Costello、Vincent Frappier、Ahmed Ismail、Shan Tie、Wujie Wang、Vincent Xue、Fritz Obermeyer、Andrew Beam 等。利用可编程生成模型探索蛋白质空间。bioRxiv，第2022–12页，2022年。3, 15[44] Lin Yeqing 和 AlQuraishi Mohammed。通过对定向残基云进行等变扩散，生成新颖、可设计且多样的蛋白质结构。2023年。doi: 10.48550。arXiv预印本 arXiv.2301.12485。3[45] Jason Yim、Brian L Trippe、Valentin De Bortoli、Emile Mathieu、Arnaud Doucet、Regina Barzilay 和 Tommi Jaakkola。用于蛋白质主链生成的 Se (3) 扩散模型。arXiv预印本 arXiv:2302.02277，2023年。2, 3, 7[46] Emiel Hoogeboom、Vıctor Garcia Satorras、Clément Vignac 和 Max Welling。用于三维分子生成的等变扩散。载于国际机器学习会议论文集，第8867–8887页。PMLR，2022年。2, 5, 6, 7, 14, 15[47] Jianyi Yang、Ivan Anishchenko、Hahnbeom Park、Zhenling Peng、Sergey Ovchinnikov 和 David Baker。利用预测的残基间取向改进蛋白质结构预测。《美国国家科学院院刊》，117(3):1496–1503，2020年。3[48] Minkyung Baek、Frank DiMaio、Ivan Anishchenko、Justas Dauparas、Sergey Ovchinnikov、Gyu Rie Lee、Jue Wang、Qian Cong、Lisa N Kinch、R Dustin Schaeffer 等。利用三轨神经网络准确预测蛋白质结构和相互作用。《科学》，373(6557):871–876，2021年。3[49] Arne Schneuing、Yuanqi Du、Charles Harris、Arian Jamasb、Ilia Igashov、Weitao Du、Tom Blundell、Pietro Lió、Carla Gomes、Max Welling 等。利用等变扩散模型进行基于结构的药物设计。arXiv预印本 arXiv:2210.13695，2022年。5, 7, 14[50] Vıctor Garcia Satorras、Emiel Hoogeboom 和 Max Welling。E(n)等变图神经网络。载于国际机器学习会议论文集，第9323–9332页。PMLR，2021年。5, 7[51] John Jumper、Richard Evans、Alexander Pritzel、Tim Green、Michael Figurnov、Olaf Ron- neberger、Kathryn Tunyasuvunakool、Russ Bates、Augustin Žídek、Anna Potapenko 等。利用 AlphaFold 高精度预测蛋白质结构。《自然》，596(7873):583–589，2021年。7, 14

[52] Mihaly Varadi、Stephen Anyango、Mandar Deshpande、Sreenath Nair、Cindy Natassia、Galabina Yordanova、David Yuan、Oana Stroe、Gemma Wood、Agata Laydon 等。AlphaFold蛋白质结构数据库：通过高精度模型大幅扩展蛋白质序列空间的结构覆盖范围。《核酸研究》，50(D1):D439–D444，2022年。7, 14[53] Ruidong Wu、Fan Ding、Rui Wang、Rui Shen、Xiwen Zhang、Shitong Luo、Chenpeng Su、Zuofan Wu、Qi Xie、Bonnie Berger 等。从一级序列进行高分辨率从头结构预测。BioRxiv，2022年。8[54] Yang Zhang 和 Jeffrey Skolnick。TM-align：一种基于 TM-score 的蛋白质结构比对算法。《核酸研究》，33(7):2302–2309，2005年。8[55] Warren L DeLano。PyMOL分子图形系统。http://www. pymol. org，2002年。8[56] Diederik P Kingma 和 Jimmy Ba。Adam：一种随机优化方法。载于 ICLR（海报），2015年。15[57] Sashank J Reddi、Satyen Kale 和 Sanjiv Kumar。论 Adam 的收敛性及其他。载于国际学习表征会议，2018年。15[58] Gilles Labesse、N Colloc'h、Joël Pothier 和 J-P Mornon。P-SEA：一种新的高效方法，用于从蛋白质 cα trace 中指派二级结构。《生物信息学》，13(3):291–295，1997年。15

A 附录

A.1 分布旋转不变反向扩散过程

在本节中，我们证明，通过 (1) 使用高维高斯分布作为先验分布，以及 (2) 采用旋转等变逆向扩散模型 s_{θ} [32, 46, 49]，可以解决 $p_\theta(\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}}) = p_\theta(R\boldsymbol{X}_{\text{down}}, \boldsymbol{H}_{\text{down}})$ 所面临的挑战。证明过程借鉴了 Xu 等人 [32] 和 Hoogeboom等人 [46]的思路。

首先，由于 p_θ $R\boldsymbol{X}_T$ $\boldsymbol{H}_T$ p_θ $R \in \mathbb{R}^{3\times3}$ $|R| = 1$ p_θ 3 $= (\boldsymbol{X}_T, \boldsymbol{H}_T)$ $= \mathcal{N}(\mathbf{0}, \mathbf{I})$，且 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 是各向同性的，因此我们有 $(\boldsymbol{X}_T, \boldsymbol{H}_T)$ (,)，其中，描述了 D 维空间中的旋转变换。

其次，由于 s_{θ} 对于 $\boldsymbol{X}_t$ 具有旋转等变性、对于 $\boldsymbol{H}_t$具有旋转不变性，并且

$$\boldsymbol{X}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(\boldsymbol{X}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{H}_t, t)_{\boldsymbol{X}}) + \sqrt{\beta_t}\boldsymbol{\sigma}_{\boldsymbol{X}}, \tag{17}$$

$$\boldsymbol{H}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(\boldsymbol{H}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{H}_t, t)_{\boldsymbol{H}}) + \sqrt{\beta_t}\boldsymbol{\sigma}_{\boldsymbol{H}}, \tag{18}$$

其中，$s_{\theta}(\boldsymbol{X}_t,\ \boldsymbol{H}_t,\ t)_{\boldsymbol{X}}$ 和 $s_{\theta}(\boldsymbol{X}_t,\ \boldsymbol{H}_t,\ t)_{\boldsymbol{H}}$ 分别表示用于更新 $\boldsymbol{X}$ 和 $\boldsymbol{H}$的网络预测。当我们将变换 $R \in \mathbb{R}^{3\times3}$, $|R| = 1$ 应用于 $\boldsymbol{X}_{t-1}$时，我们将得到

$$R\boldsymbol{X}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}R(\boldsymbol{X}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{H}_t, t)_{\boldsymbol{X}}) + \sqrt{\beta_t}R\boldsymbol{\sigma}_{\boldsymbol{X}} \tag{19}$$

$$= \frac{1}{\sqrt{1-\beta_t}}(R\boldsymbol{X}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}R\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{X}_t, \boldsymbol{H}_t, t)_{\boldsymbol{X}}) + \sqrt{\beta_t}R\boldsymbol{\sigma}_{\boldsymbol{X}} \tag{20}$$

$$= \frac{1}{\sqrt{1-\beta_t}}(R\boldsymbol{X}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(R\boldsymbol{X}_t, \boldsymbol{H}_t, t)_{\boldsymbol{X}}) + \sqrt{\beta_t}R\boldsymbol{\sigma}_{\boldsymbol{X}}, \tag{21}$$

并且我们可以得到如下结果。

$$\begin{aligned} p_\theta(\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}|\boldsymbol{X}_t, \boldsymbol{H}_t) = p_\theta(\boldsymbol{X}_t, \boldsymbol{H}_t)p(\boldsymbol{\sigma}_{\boldsymbol{X}}, \boldsymbol{\sigma}_{\boldsymbol{H}}) = p_\theta(R\boldsymbol{X}_t, \boldsymbol{H}_t)p(R\boldsymbol{\sigma}_{\boldsymbol{X}}, \boldsymbol{\sigma}_{\boldsymbol{H}}) \\ = p_\theta(R\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}|R\boldsymbol{X}_t, \boldsymbol{H}_t). \end{aligned} \tag{22}$$

除此之外，对于反向扩散时间 $t \in \{T, T-1, \cdots, 1\}$，假设 $p_\theta(\boldsymbol{X}_t, \boldsymbol{H}_t)$ 满足 $p_\theta(\boldsymbol{X}_t, \boldsymbol{H}_t)$ $= p_\theta(R\boldsymbol{X}_t, \boldsymbol{H}_t)$，其中 $R \in \mathbb{R}^{3\times3}$, $|R| = 1$ 描述三维空间中的旋转变换。于是我们有:

$$\begin{aligned} p_\theta(R\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}) &= \int_{(\boldsymbol{X}_t, \boldsymbol{H}_t)} p_\theta(R\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}|\boldsymbol{X}_t, \boldsymbol{H}_t)p_\theta(\boldsymbol{X}_t, \boldsymbol{H}_t) \\ &= \int_{(\boldsymbol{X}_t, \boldsymbol{H}_t)} p_\theta(R\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}|RR^{-1}\boldsymbol{X}_t, \boldsymbol{H}_t)p_\theta(RR^{-1}\boldsymbol{X}_t, \boldsymbol{H}_t) \\ &= \int_{(\boldsymbol{X}_t, \boldsymbol{H}_t)} p_\theta(\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}|R^{-1}\boldsymbol{X}_t, \boldsymbol{H}_t)p_\theta(R^{-1}\boldsymbol{X}_t, \boldsymbol{H}_t), \end{aligned}$$

令 $\boldsymbol{X}' = R^{-1}\boldsymbol{X}_t$，我们有 det $R = 1$且

$$p_\theta(R\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}) == \int_{(\boldsymbol{X}', \boldsymbol{H}_t)} p_\theta(\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}|\boldsymbol{X}', \boldsymbol{H}_t)p_\theta(\boldsymbol{X}', \boldsymbol{H}_t) * \det R = p_\theta(\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1}), \tag{23}$$

且 $p_\theta(\boldsymbol{X}_{t-1}, \boldsymbol{H}_{t-1})$ 是不变的。通过数学归纳法，$p_\theta(\boldsymbol{X}_{T-1}, \boldsymbol{H}_{T-1}) \ldots, p_\theta(\boldsymbol{X}_0, \boldsymbol{H}_0)$ 均为不变的，证明完毕。

A.2 数据集

我们从 Protein Data Bank (PDB) 和 AlphaFold 蛋白质结构数据库 (AlphaFold DB) 中的 Swiss-Prot 数据整理数据集 [51, 52]。我们筛选 PDB 中所有单链蛋白质数据，要求 $C_\alpha - C_\alpha$ 距离小于 5Å，且序列长度介于 40 至 128 个残基之间，最终得到 4460 条蛋白质序列。我们按照 80/10/10 的训练集/验证集/测试集比例随机划分数据。为纳入更多训练数据，我们进一步从两个来源整理蛋白质数据，并将其加入当前训练集。增强训练数据的第一部分来自 AlphaFold DB。

具体而言，我们筛选 Swiss-Prot 中长度介于 40 至 128 之间的单链蛋白质，并将这些蛋白质加入训练数据。增强训练数据的第二部分来自 PDB，我们从中整理出 $C_\alpha - C_\alpha$ 距离大于 5Å 且序列长度大于 40 的单链蛋白质数据。具体而言，我们在 $C_\alpha - C_\alpha$距离大于 5Å 的位置切分这些蛋白质，以获得蛋白质片段。随后，我们将长度介于 50 至 128 之间的片段加入训练数据。对于长度大于 256 的片段，我们将其均匀切分为长度介于 50 至 128 之间的片段，并加入训练数据。经过这一数据增强流程，我们最终获得约 $100k$ 训练数据。

A.3 实验详情

训练自编码器时，我们使用了所有可用的训练数据。随后，我们使用训练好的编码器嵌入所有训练用蛋白质数据，并使用其潜在表示训练潜在扩散模型。我们使用Adam优化器训练自编码器200个epoch，批量大小为128 [56]，学习率为 $1e^{-3}$，$\beta_1 = 0.9$，$\beta_2 = 0.999$，权重衰减为 $2e^{-4}$。潜在扩散模型的训练轮数为 $16k$ 个epoch，批量大小为2048，使用Amsgrad优化器 [57]，学习率为 $5e^{-5}$，$\beta_1 = 0.9$，$\beta_2 = 0.999$，权重衰减为 $1e^{-12}$。我们使用1000个扩散步数，并采用Hoogeboom等人使用的相同噪声调度器。 [46]所有模型均使用PyTorch实现。蛋白质自编码器在单张NVIDIA A100 GPU上训练了6天。潜在扩散模型在四张NVIDIA A100 GPU上训练了7天。

A.4 总体生成过程

为了生成新颖的蛋白质骨架结构，我们首先对多元高斯噪声进行采样，并使用学习到的潜在扩散模型在潜在空间中生成三维位置和节点嵌入。我们在扩散模型的逆向过程中使用低温采样 [43] 。随后，我们使用预训练解码器在蛋白质空间中生成骨架结构。需要注意的是，解码器的输出具有预先定义的固定大小。为了生成不同长度的蛋白质，我们预测解码器输出中的每个节点是否为增强节点。我们只需找到第一个被分类为增强节点的节点，并丢弃生成的蛋白质骨架结构中的其余节点。需要注意的是，对于相应节点，我们不使用重建的氨基酸类型。相反，我们使用逆折叠模型 ProteinMPNN [36] ，根据生成的骨架结构预测蛋白质氨基酸序列。

A.5 二级结构

我们使用 P-SEA [58] 来统计生成的蛋白质中两种二级结构的数量。具体而言，我们分别计算仅包含 α-螺旋、仅包含 β-折叠以及同时包含 α-螺旋和 β-折叠的生成蛋白质所占的百分比。结果如表 4所示。可以看到，超过一半的生成蛋白质包含 α-螺旋，并且相当一部分生成蛋白质包含β-折叠。这证明我们的方法能够成功生成天然蛋白质中的各种二级结构。

表 4： 仅含 α-螺旋、仅含 β-折叠以及同时含有 α-螺旋和 β-折叠的生成蛋白质所占百分比分别为。

仅含 α-螺旋	仅含 β-折叠	α-螺旋 + β-折叠
73.4%	2.2%	23.8%

A.6 多样性

我们还使用 scTM 评估生成的蛋白质的多样性 > 0.5（可设计蛋白质），如表 5 所示。具体而言，我们为每个可设计蛋白质计算其与所有其他可设计蛋白质的 TM 分数，并选择最大的 TM 分数，以衡量其与生成的蛋白质的相似性。然后，我们计算所有可设计蛋白质的最大 TM 分数的平均值，以评估生成的蛋白质的多样性（越低越好）。

表 5：生成的可设计蛋白质的多样性（scTM >0.5）。$\downarrow$ 表示值越低，性能越好。

方法	多样性$^{\downarrow}$
ProtDiff	0.836±0.1648
FoldingDiff	0.585±0.1276
FrameDiff	0.611±0.1544
LatentDiff	0.634±0.0919

A.7 潜在空间插值

通常，可视化潜在空间并进行潜在编码插值，以测试潜在空间的结构是否良好，这是一种自然的做法。然而，在我们的潜在空间中，蛋白质并非由单个潜在特征向量表示，而是由一组关联着三维坐标和节点特征的节点表示。因此，很难使用 t-SNE 等降维技术来可视化潜在空间。此外，我们没有对坐标添加 KL 散度损失，因为这会破坏等变性。即使对于不变节点特征，我们也只添加了最小限度的 KL 散度惩罚项来控制潜在空间的方差，因为我们的目标是保持自编码器的高重建准确率。因此，在我们的情形下，潜在空间不一定需要具有良好的结构，而且任意插值在解码后也未必能保证得到有效的蛋白质结构。

为展示这一点，我们选取了两个 scTM>0.5（可设计的）生成的蛋白质，其对应的潜在空间数据分别为（X^s_{emb}, H^s_{emb}）和（X^t_{emb}, H^t_{emb}）。然后，我们将这两个潜在空间数据进行插值，得到（$X^{interp}_{emb}, H^{interp}_{emb}$）$=$（$X^s_{emb} *$（$1-\lambda$）$+X^t_{emb} * \lambda$，$H^s_{emb} *$（$1-\lambda$）$+H^t_{emb} * \lambda$）。我们选取 λ 的不同取值，将插值后的潜在空间数据解码为蛋白质，并计算 scTM 分数，如表6所示。可以看到，当 λ 接近 0 或 1 时，生成的蛋白质仍然是可设计的。然而，当 λ 接近 0.5 时，生成的蛋白质无效，正如我们上文所分析的那样。

表6： 由两个潜在蛋白质表示的插值解码得到的蛋白质的 scTM 分数。λ 是插值权重。TM-left 表示与起始蛋白质的 TM评分，TM-right 表示与终止蛋白质的 TM评分。

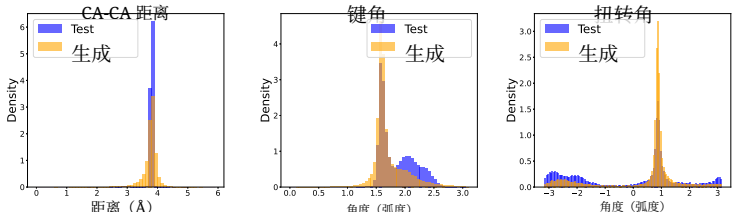
λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
scTM	0.86	0.61	0.49	0.48	0.32	0.33	0.27	0.30	0.35	0.62	0.78
TM-left	1.0	0.74	0.57	0.48	0.36	0.29	0.31	0.35	0.40	0.43	0.48
TM-right	0.49	0.51	0.46	0.41	0.37	0.36	0.32	0.39	0.56	0.75	1.0

A.8 结构与序列协同设计

由于蛋白质自编码器的解码器能够预测氨基酸类型，LatentDiff 也具备执行结构与序列协同设计的能力，这是其与其他蛋白质生成方法的一个关键区别。具体而言，我们可以直接使用解码后的序列作为生成的蛋白质序列，而无需采用逆折叠方法从解码结构中预测序列。可设计性结果如表 7 所示。我们可以看到，与生成序列相比，逆折叠预测的序列与生成结构具有更好的对齐效果。这是因为有条件地预测序列比联合生成结构和序列更为容易。然而，即使使用生成的序列，LatentDiff 仍能取得与 ProtDiff 等早期蛋白质生成方法相近的结果。

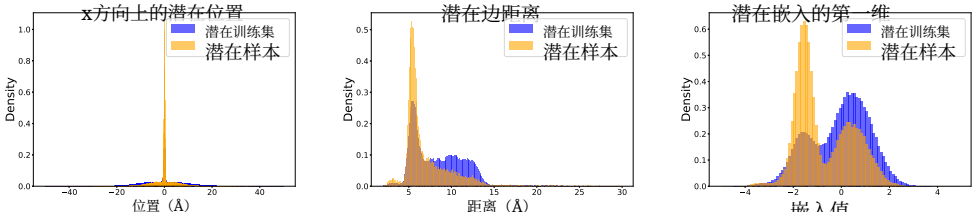
表7： scTM 分数为 >0.5 的生成蛋白质百分比。结果按短链（50–70）和长链（70–128）类别展示。

方法	50–70	70–128	50–128
ProtDiff（使用逆折叠序列）	17.1%	8.9%	11.8%
LatentDiff（使用生成的序列）	14.7%	9.7%	14.1%
LatentDiff（使用逆折叠序列）	64.7%	82.8%	66.9%



(a) **(b)** **(c)**

图 4： 生成的主链结构与测试集蛋白质骨架的分布比较。 (a) 蛋白质链上任意两个连续 C_α 原子之间的边距离。 (b) 蛋白质链上任意三个连续 C_α 原子形成的键角。 (c)蛋白质链上任意四个连续 C_α 原子形成的扭转角。



(a) **(b)** **(c)**

图 5： 训练数据与潜在空间中生成样本的分布比较。(a) 潜在节点在 x 方向上的位置。 (b) 潜在空间中任意两个连续节点之间的边距离。 (c)潜在节点嵌入的第一维。

A.9 结构分布分析

除了展示 计算机模拟 测试的成功之外，我们还展示了生成样本在原始蛋白质空间和潜在空间中的分布。首先，我们展示生成的主链和测试集主链的边距离、键角和扭转角分布。如图 4所示，生成样本的分布与测试集的分布相似。我们进一步研究了潜在空间中的分布。具体而言，我们展示了潜在空间中节点位置、边距离和节点嵌入的分布。为简化起见，我们仅展示潜在节点位置的 x 坐标以及潜在节点嵌入的第一维。如图 5所示，这些生成的潜在样本分布几乎恢复了潜在训练数据的分布。