PPFLOW: Target-Aware Peptide Design with Torsional Flow Matching

Haitao Lin [1 2] **Odin Zhang** [1] **Huifeng Zhao** [1] **Dejun Jiang** [1]
Lirong Wu [2 1] **Zicheng Liu** [2 1] **Yufei Huang** [2 1] **Stan Z. Li** [2]

Abstract

Therapeutic peptides have proven to have great pharmaceutical value and potential in recent decades. However, methods of AI-assisted peptide drug discovery are not fully explored. To fill the gap, we propose a target-aware peptide design method called PPFLOW, based on conditional flow matching on torus manifolds, to model the internal geometries of torsion angles for the peptide structure design. Besides, we establish a protein-peptide binding dataset named `PPBench2024` to fill the void of massive data for the task of structure-based peptide drug design and to allow the training of deep learning methods. Extensive experiments show that PPFLOW reaches state-of-the-art performance in tasks of peptide drug generation and optimization in comparison with baseline models, and can be generalized to other tasks including docking and side-chain packing.

1. Introduction

Therapeutic peptides are a unique class of pharmaceutical agents composed of a series of well-ordered amino acids. The development of peptide drug design and discovery is accelerated by fast advances in structural biology, recombinant biologics, and synthetic and analytic technologies (Wang et al., 2022). Peptide drugs play an important role in pharmacology because they usually bind to cell surface receptors and trigger intracellular effects with high affinity and specificity, showing less immunogenicity and taking lower production costs (Muttenthaler et al., 2021). Therefore, they have raised great research interests as new peptide therapeutics are continuously developed with more than 150 peptides in clinical tests and another 400–600 peptides undergoing preclinical studies.

Deep learning has revolutionized fields like drug discovery and protein design (Makhatadze, 2021; Watson et al., 2023; Dauparas et al., 2022), which proves to be effective tools to assist the development of small and large molecule drugs. Peptide drugs occupy a unique chemical and pharmacological space between small and large molecules, but the AI-assisted peptide drug discovery methods remain limited compared with those established for small molecules and proteins. Unbound peptide chains are usually at high free energy and entropy values, thus showing unstable conformations, while they trigger pharmacological effects when binding to specific receptors, forming a complex with equilibrium structures that are composed of a pair of receptor and ligand. (Marullo et al., 2013; Seebach et al., 2006). Therefore, we focus on the designation of peptide drugs that can bind to specific receptors (Todaro et al., 2023). Recently, structure-based drug design (SBDD) methods are developed for target-aware small molecule generation (Peng et al., 2022a; Guan et al., 2023; Lin et al., 2022), while these methods cannot be simply transferred to peptide design tasks for the following reasons: (i) Peptide drugs are usually of larger molecular weights ($500-5000$ Da) compared with small molecule drugs (<1000 Da) (Francoeur et al., 2020); (ii) Topologies of atom connectivity in peptides are close to proteins rather than molecules; (iii) Internal geometries in peptides are of different patterns from small molecules.

Hence, we identify four challenges for the target-aware peptide drug design. *First*, how to extract sufficient information from the conditional receptor contexts. This is indispensable for the model to generalize to new receptors. *Second*, how to generate valid peptides that satisfy physicochemical rules. If the generated peptides are not chemically valid, other drug properties are unnecessary for further consideration. *Third*, searching for natural peptides and replacing them with animal homologs, such as the discovery of insulin, GLP-1 and somatostatin, were the important strategies used for peptide drug discovery (Kelly et al., 2022). Therefore, instead of de novo design, the model should apply to another scenario: optimizing natural peptides to drugs with higher binding affinities. *Besides*, there are few available benchmark datasets large enough to support the training of deep learning models for structure-based peptide drug design tasks, and it is urged to collect and organize a high-quality dataset that satisfies the demand for massive data.

(i) To address data shortage, we first construct a dataset consisting of a large number of protein-peptide complexes through a series of systematic steps. *(ii) To give a solution to the task*, we propose a generative model based on flow matching, to learn the structure distribution of peptides from their torsion geometry and the distributions of global translation and orientation. Instead of directly learning the explicit

probability, PPFLOW fits the gradient fields of the variables' evolutionary process from a prior distribution to the peptides' sequence-structure distributions and uses the learned gradient field to iteratively push the sequences and structures toward data distribution by an ordinary differential equation. *(iii) To sufficiently extract representation from the protein*, we employ neural networks as the approximation to the gradient fields, parameterized with attention-based architectures to encode contextual proteins' structure and sequence. *(iv) To confirm the validity of the model*, extensive experiments on multi-tasks are conducted. PPFLOW reaches state-of-the-art performance compared with diffusion-based methods we transferred in generating peptides with higher affinity, stability, validity, and novelty. In optimization tasks, PPFLOW also shows great superiority. Besides, experiments on flexible re-docking and side-chain packing demonstrate the models' potential in structure modeling.

To sum up, our contributions include: (i) **New Task**. We consider the conditional generation of peptide drugs targeted at protein receptors as contexts, which is an area that has rarely been explored by deep learning. Other tasks including peptide optimization, flexible re-docking, and side-chain packing are also studied and evaluated. (ii) **Novel Model**. To fulfill the task, we establish a new model called PPFLOW as a complete solution to model the internal geometries of torsion angles in peptides, achieving competitive performance in the discussed tasks. In addition, to our best knowledge, we are the first to establish flow-matching models for torsional angles lying on torus manifolds and to propose a deep-learning model for target-specific peptide generation. (iii) **Dataset Establishment**. For training the deep learning model, we collect a benchmark dataset called `PPBench2024` consisting of 15593 protein-peptide pairs with high quality. Compared with previous protein-peptide datasets (Agrawal et al., 2019; Martins et al., 2023; Wen et al., 2018), `PPBench2024` augments and expands available data, and filters large amounts of low-quality data according to strict criteria, thus allowing deep-learning-based models to be trained on the datasets and to fulfill the peptide drug discovery tasks.

2. Background

2.1. Problem Statement

For a binding system composed of a protein-ligand pair (*i.e.* protein-peptide pair) as $\mathcal{C}$, which contains N_{pt} amino acids of the protein and N_{pp} amino acids of the peptide, we represent the index set of the amino acids in the peptide as $\mathcal{I}_{\text{pp}} = \{1, \ldots, N_{\text{pp}}\}$, and the protein's amino acids as $\mathcal{I}_{\text{pt}} = \{N_{\text{pp}}+1, \ldots, N_{\text{pp}}+N_{\text{pt}}\}$. The amino acids can be represented by its type $s^{(i)}$, atom coordinates $X^{(i)} = (\mathbf{x}^{(i,1)}, \ldots, \mathbf{x}^{(i,M^{(i)})})$, where $s^{(i)} \in \{1, \ldots, 20\}$, $\mathbf{x}^{(i,j)} \in \mathbb{R}^3$, and $M^{(i)} = M(s^{(i)})$, meaning that the number of atoms is decided by the amino acid types. The sets of $\{s^{(i)}\}_{i=1}^{N_{\text{pp}}}$ refers to a peptide sequence, and $\{X^{(i)}\}_{i=1}^{N_{\text{pp}}}$ refers to its structure, which is the same for protein.

Therefore, $\mathcal{C} = \{(s^{(i)}, X^{(i)})\}_{i=1}^{N_{\text{pt}}+N_{\text{pp}}}$ can be split into two sets as $\mathcal{C} = \mathcal{P} \cup \mathcal{L}$, where $\mathcal{P} = \{(s^{(i)}, X^{(i)}) : i \in \mathcal{I}_{\text{pt}}\}$ and $\mathcal{L} = \{(s^{(i)}, X^{(i)}) : i \in \mathcal{I}_{\text{pp}}\}$. For *protein-specific peptide generation*, our goal is to establish a probabilistic model to learn the distribution of molecules conditioned on the target proteins, *i.e.* $p(\mathcal{L}|\mathcal{P})$.

2.2. Riemanian Flow Matching

Let $\mathcal{P} = \mathcal{P}(\mathcal{M})$ be the probability function space over a manifold $\mathcal{M}$ with Riemanian metric g. q is probability distribution of data $x \in \mathcal{M}$, and p is the prior distribution. The probability path on $\mathcal{M}$ as an interpolation in probability space written as $p_t : [0,1] \to \mathcal{P}$ satisfies $p_0 = p$ and $p_1 = q$. $u_t(x) \in \mathcal{T}_x\mathcal{M}$ is the corresponding gradient vector of the path on x at time t. A Flow Matching (FM) tangent vector field $v_t : [0,1] \times \mathcal{M} \to \mathcal{M}$ is used to approximate $u_t(x)$ with the objective $L_{\text{RFM}}(\theta) = \mathbb{E}_{t,p_t(x)}\|v_t(x) - u_t(x)\|_g^2$, in which θ are the parameters in v_t. u_t is intractable, and an alternative is to construct conditional density path $p_t(x|x_1)$ whose gradient reads $u_t(x|x_1)$, and use Conditional Flow Matching (CFM) objective as

$$L_{\text{CRFM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),p_1(x_1),p_t(x|x_1)}\|v_t(x) - u_t(x|x_1)\|_g^2,$$

since FM and CFM objectives have the same gradients as shown in (Lipman et al., 2022; Chen & Lipman, 2023). Once the gradient field v is learned, one can use ordinary differential equations $\frac{d}{dt}\varphi_t(x) = v_t(\varphi_t(x))$ with $\varphi_t(x) = x \sim p_0$ and $\varphi_t(x) = x_t$ to push the random variable $x_t \in \mathcal{M}$ from prior distribution p_0 to the data distribution p_1.

3. Proposed Method

3.1. Peptide Structure Parameterization

There are several parameterizations for peptide structures. Here, we firstly focus on the four backbone atoms, *i.e.* {N, Cα, C, O}. One effective parameterization follows ALPHAFOLD2 (Jumper et al., 2021), in which the four atom positions $X^{(i)*} = \{\mathbf{x}^{(i,1)}, \ldots, \mathbf{x}^{(i,4)}\}$ form a rigid body, and represent $X^{(i)*} \triangleq (\mathbf{x}^{(i,2)}, O^{(i)})$ as variables on SE(3), in which $\mathbf{x}^{(i,2)} \in \mathbb{R}^3$ as Cα's coordinate is the translation vector and $O_i \in \text{SO}(3)$ is an rotation matrix obtained by the ideal frame constructed by $X^{(i)*}$, leading $\{(\mathbf{x}^{(i,2)}, O^{(i)})\}_{i=1}^{N_{\text{pp}}}$ to represent the backbone structure. Hence, the structure representation lies on a manifold with $6N_{\text{pp}}$-degree of freedom.

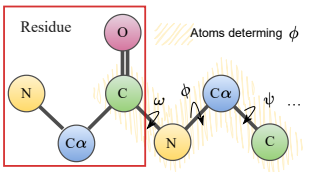


Figure 1: Backbone atom parameterization.

However, while the intra-residues' internal geometries are fixed, *e.g.* bond lengths of 'N-Cα' and 'Cα-C', and bond angles of 'N-Cα-C', the other inflexible inter-residues' geometries of bond lengths such as 'C-N' and angles like 'C-N-Cα' are not constrained in the parameterization.

In comparison, we focus on the torsion angles of ϕ, ψ, and ω (shown in Figure. 1) as redundant geometries according to the physicochemical conclusions (Padmanabhan, 2014) and our observations (See Appendix. B.2). While the local structures can be reconstructed with torsion angles and ideal bond lengths and angles with NERF (Parsons et al., 2005), the global translation and orientation have to be determined to obtain the peptides' poses relative to the target proteins (Corso et al., 2023). By this means, the backbone peptide structure can be represented as $\{X^{(i)*}\}_{i=1}^{N_{\text{PP}}} \triangleq \{(\phi^{(i)}, \psi^{(i)}, \omega^{(i)})\}_{i=2}^{N_{\text{PP}}} \cup \{\mathbf{x}^{(\text{C})}, O^{(\text{C})}\}$. Note that i start from 2 because four consecutive atoms form a torsion angle, $\mathbf{x}^{(\text{C})}$ is the global translation of the peptide centroid, and $O^{(\text{C})}$ is the rotation matrix of the global frame constructed by $\{\mathbf{x}^{(1,1)}, \mathbf{x}^{(\text{C})}, \mathbf{x}^{(N_{\text{PP}},3)}\}$. Hence, the backbone structure is parameterized by variables on $\mathbb{T}^{3N_{\text{PP}}-3} \times \text{SE}(3)$. The advantages of the parametrization include *(i)* the inflexible bond lengths and angles are set to be constants, avoiding broken or overlapped bonds as unrealistic structures and *(ii)* fewer degrees of freedom to reduce the task difficulties ($3N_{\text{bb}} + 3 < 6N_{\text{bb}}$).

In the following, the CFM on the discussed manifolds will be described in Sec. 3.2 and Sec. 3.3, and for the sequence type CFM, it will be discussed in Sec. 3.4. Besides, for the other atoms in side-chains, we use the rotamer χ-angles as the parameterization, discussed in Sec. 3.8.

3.2. PPFLOW on Torus

For each torsion angle lies in $[-\pi, \pi)$, N torsion angles of a structure define a hypertorus $\mathbb{T}^N$, with $N = 3N_{\text{pp}} - 3$. To construct a conditional flow on $\mathbb{T}^N$, we follow RCFM, to obtain $\boldsymbol{\tau}_t$ using the geodesic connecting $\boldsymbol{\tau}_0$ and $\boldsymbol{\tau}_1$ with

$$\boldsymbol{\tau}_t = \exp_{\boldsymbol{\tau}_0}(t \log_{\boldsymbol{\tau}_0}(\boldsymbol{\tau}_1)), \tag{1}$$

where $\boldsymbol{\tau}_0 \sim p_0$, $\boldsymbol{\tau}_1 \sim p_1$ and $\boldsymbol{\tau}_t \in \mathbb{T}^N$, and the conditional vector field $u_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1) = \frac{d}{dt}\boldsymbol{\tau}_t = \dot{\boldsymbol{\tau}}_t$. $\log_{\boldsymbol{\tau}_0} : \mathbb{T}^N \to \mathcal{T}_{\boldsymbol{\tau}_0}\mathbb{T}^N$ is the logarithm map, projecting points on $\mathbb{T}^N$ to the tangent space of $\boldsymbol{\tau}_0$; $\exp_{\boldsymbol{\tau}_0} : \mathcal{T}_{\boldsymbol{\tau}_0}\mathbb{T}^N \to \mathbb{T}^N$ is the exponential map, projecting points back to torus (Chen & Lipman, 2023).

For the torus, we parameterize the manifold as the quotient space $\mathbb{R}^N/2\pi\mathbb{Z}^N$, leading to the equivalence relations $\boldsymbol{\tau} = (\tau^{(1)}, \ldots, \tau^{(N)}) \cong (\tau^{(1)} + 2\pi, \ldots, \tau^{(N)}) \cong (\tau^{(1)}, \ldots, \tau^{(N)} + 2\pi)$ (Jing et al., 2023). By this means the prior distribution p_0 is chosen as a product of standard wrapped normal distribution on $\mathbb{R}^N$ as

$$p_0(\boldsymbol{\tau}) = \mathcal{WN}(\boldsymbol{\tau}) \propto \prod_{i=1}^{N} \sum_{d \in \mathbb{Z}} \exp\left(-\frac{\|\tau^{(i)} + 2\pi d\|^2}{2}\right). \tag{2}$$

To construct the conditional path, considering the flow

$$\boldsymbol{\tau}_t = \sigma_t(\boldsymbol{\tau}_1)\boldsymbol{\epsilon} + \boldsymbol{\mu}_t(\boldsymbol{\tau}_1), \tag{3}$$

where $\boldsymbol{\epsilon} \sim \mathcal{WN}(\boldsymbol{\epsilon})$. To build the gradient vector, we propose that **Theorem 3.** in (Lipman et al., 2022) still holds for the defined path, as the following proposition:

Proposition 3.1. *Let $p_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1)$ be probability path as in Equation. 3. Its vector field has the form*

$$u_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1) = \frac{\dot{\sigma}_t(\boldsymbol{\tau}_1)}{\sigma_t(\boldsymbol{\tau}_1)}(\boldsymbol{\tau} - \boldsymbol{\mu}_t(\boldsymbol{\tau}_1)) + \dot{\boldsymbol{\mu}}_t(\boldsymbol{\tau}_1). \tag{4}$$

Hence, $u_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1)$ generates the probability path $p_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1)$.

The proof is given in Appendix. A.1. Besides, the following proposition ensures the correctness of the learning objective in Conditional Torus Flow Matching with proof in Appendix. A.2:

Proposition 3.2. *Given $p_t(\boldsymbol{\tau})$, $\boldsymbol{\tau} \in \mathbb{T}^N$, the conditional and unconditional flow matching losses have equal gradients* w.r.t. *θ: $\nabla_\theta L_{\text{UTFM}}(\theta) = \nabla_\theta L_{\text{CTFM}}(\theta)$.*

Following (Tong et al., 2023), we in practice construct our flow-matching objective in an Independent Coupling way:

$$\begin{aligned} \boldsymbol{\tau}_t(\boldsymbol{\tau}_0, \boldsymbol{\tau}_1) &= \exp_{\boldsymbol{\tau}_0}(t \log_{\boldsymbol{\tau}_0}(\boldsymbol{\tau}_1)) + \sigma\boldsymbol{\epsilon}; \\ u_t(\boldsymbol{\tau}|\boldsymbol{\tau}_0, \boldsymbol{\tau}_1) &= \frac{d}{dt}\exp_{\boldsymbol{\tau}_0}(t \log_{\boldsymbol{\tau}_0}(\boldsymbol{\tau}_1)), \end{aligned} \tag{5}$$

which is a specific instance of the above proposition. The explicit equations for geodesics on a hypertorus $\mathbb{T}^N$ can be complex and depend on the specific parametrization chosen for the hypertorus (Jantzen, 2012). Instead, we regard the torsion angles as mutually orthogonal and their interpolation paths on the torus are linear *w.r.t.* t in each direction. Therefore, we employ a fast implementation of calculating $\boldsymbol{\mu}_t \approx \exp_{\boldsymbol{\tau}_0}(t \log_{\boldsymbol{\tau}_0}(\boldsymbol{\tau}_1))$ by

$$\begin{aligned} \boldsymbol{\mu}_t(\boldsymbol{\tau}|\boldsymbol{\tau}_0, \boldsymbol{\tau}_1) &= t\boldsymbol{\tau}_1' + (1-t)\boldsymbol{\tau}_0; \\ u_t(\boldsymbol{\tau}|\boldsymbol{\tau}_0, \boldsymbol{\tau}_1) &= \boldsymbol{\tau}_1' - \boldsymbol{\tau}_0, \end{aligned} \tag{6}$$

where $\boldsymbol{\tau} \cong \boldsymbol{\tau}_1' = (\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0 + \pi) \bmod (2\pi) - \pi$. The geodesic distance is approximated with the euclidean ones, as $\|\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0\|_{\mathbb{T}^N}^2 \approx \|\boldsymbol{\tau}_1' - \boldsymbol{\tau}_0\|_2^2$, which satisfies definitions of premetric in Sec. 3.2 in (Chen & Lipman, 2023). This leads the closed-from expression of the loss to train the conditional Torus Flow Matching to

$$L_{\text{TFM}}(\theta) = \mathbb{E}_{\substack{t \sim \mathcal{U}(0,1), p_1(\boldsymbol{\tau}_1), \\ p_0(\boldsymbol{\tau}_0), p_t(\boldsymbol{\tau}|\boldsymbol{\tau}_0, \boldsymbol{\tau}_1)}} \|v_t(\boldsymbol{\tau}) - \boldsymbol{\tau}_1' + \boldsymbol{\tau}_0\|_2^2. \tag{7}$$

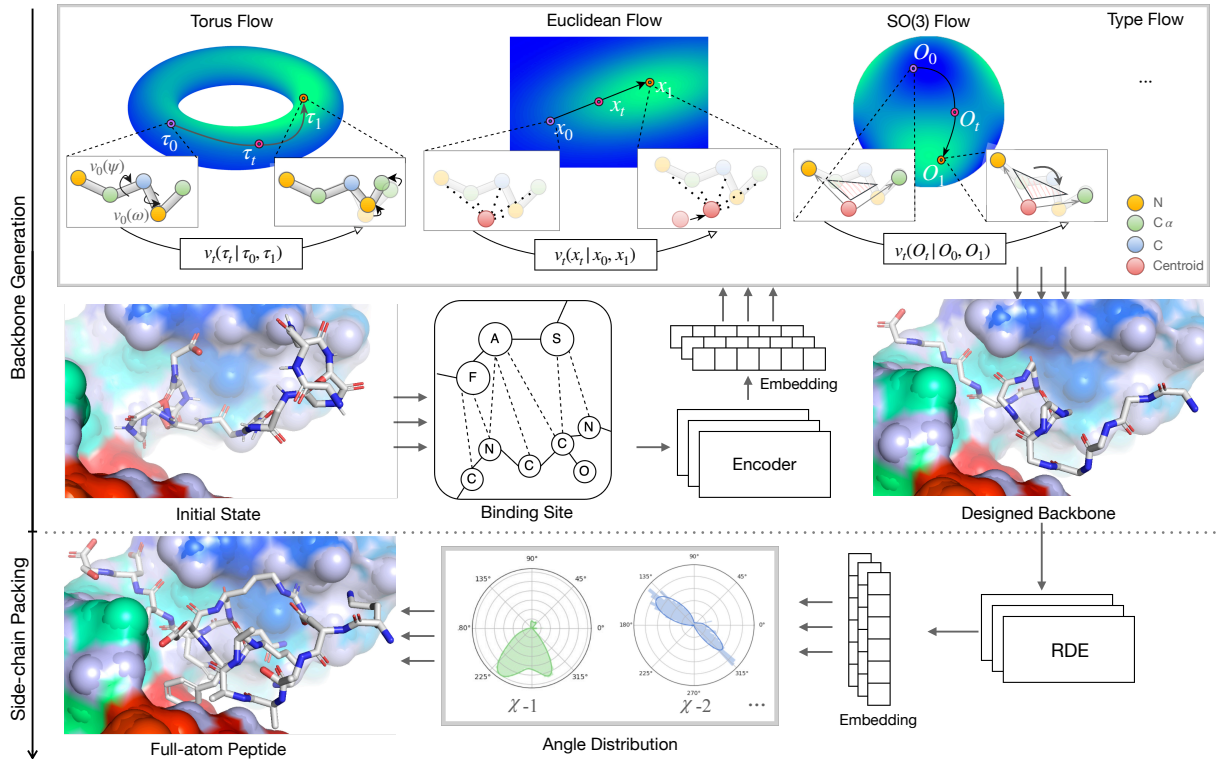


Figure 2: Workflows of PPFLOW in target-aware peptide drug generation task.

3.3. PPFLOW on SE(3)

The torsion angles can reconstruct the local structures of the peptides, while in the global coordinate system, to represent the positions of residues, we need to determine their global translations and rotations.

As discussed in Sec. 3.1, the pose representation on SE(3) can be decomposed into global translation as $\mathbf{x}^{(\mathrm{C})} \in \mathbb{R}^3$ and rotation $O^{(\mathrm{C})} \in \mathrm{SO}(3)$. For notation simplicity, we omit the superscript $^{(\mathrm{C})}$ in this part. To model the probability path of $\mathbf{x}_t$, we employ vanilla Gaussian CFM on Euclidean manifolds, with Independent Coupling techniques:

$$\begin{aligned} \boldsymbol{\mu}_t(\mathbf{x}|\mathbf{x}_0,\mathbf{x}_1) &= t\mathbf{x}_1 + (1-t)(\mathbf{x}_1 - \mathbf{x}_0); \\ \sigma_t &= \sigma, \end{aligned} \tag{8}$$

where $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}|\mathbf{0}, I)$, thus leading to the Gaussian probability path of $p_t(\mathbf{x}|\mathbf{x}_0,\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_t, \sigma_t)$, and the loss to train the conditional Euclidean Flow Matching as

$$L_{\mathrm{EFM}}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1), p_1(\mathbf{x}_1), \\ p_0(\mathbf{x}_0), p_t(\mathbf{x}|\mathbf{x}_0,\mathbf{x}_1)}} \|v_t(\mathbf{x}) - \mathbf{x}_1 + \mathbf{x}_0\|_2^2. \tag{9}$$

To model the probability path of rotation matrix $O_t \in \mathrm{SO}(3)$, we employ SO(3)-CFM (Yim et al., 2023a), in which $O_t = \exp_{O_0}(t \log_{O_0}(O_1))$. In the implementation, since the SO(3) is a simple manifold with closed-form geodesics, the exponential map can be computed using Rodrigues' formula and the logarithmic map is similarly easy to compute with its Lie algebra $\mathfrak{so}(3)$ (Yim et al., 2023b). The prior distribution to sample O_0 is defined as isotropic Gaussian distribution, by first parameterizing O_0 in axis-angle, where the axis of rotation is sampled uniformly and the density function of rotation angle ϑ reads $\mathcal{IG}(\vartheta) = \frac{1-cos\vartheta}{\pi}\sum_{l=0}^{\infty}(2l+1)e^{-l(l+1)\epsilon}\frac{\sin((l+\frac{1}{2})\vartheta)}{\sin(\vartheta/2)}$ (Creasey & Lang, 2017). For the conditional gradient field, we employ fast numerical tricks (Bose et al., 2023) to calculate the SO(3) component of the global Orientation conditional Flow Matching objectives:

$$L_{\mathrm{OFM}}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1), p_1(O_1), \\ p_0(O_0), p_t(O|O_0,O_1)}} \left\| v_t(O) - \frac{\log_{O_t}(O_0)}{t} \right\|_{\mathrm{SO}(3)}^2. \tag{10}$$

The correctness of the learning objective of Equation. 9 and 10 is proposed and proven in (Lipman et al., 2022) and (Bose et al., 2023), respectively.

3.4. PPFLOW on Amino Acid Types

For the amino acid sequence $\boldsymbol{s} = \{s^{(i)}\}_{i=1}^{N_{\mathrm{PP}}}$, we directly model the probability vector of each type, where $\boldsymbol{c}_t^{(i)}$ is defined as the probability vector of multinomial distribution with $s_t^{(i)} \sim P(\boldsymbol{c}_t^{(i)})$. We omit $^{(i)}$ for notation simplicity. To build a path, we define $\boldsymbol{c}_1 = \mathrm{onehot}(s_i)$, and $\boldsymbol{c}_0 = (\frac{1}{20}, \ldots, \frac{1}{20})$. Analogously to the Euclidean flow, we define $\boldsymbol{c}_t = t\boldsymbol{c}_1 + (1-t)\boldsymbol{c}_0$, and $u_t(\boldsymbol{c}|\boldsymbol{c}_0,\boldsymbol{c}_1) = \boldsymbol{c}_1 - \boldsymbol{c}_0$. It is

easy to prove $\boldsymbol{c}_t$ is a probability vector since its summation equals 1 across all types. Instead of an MSE training loss in $\mathbb{R}^{20}$, we, inspired by multinomial diffusion (Hoogeboom et al., 2021), propose a multinomial flow matching objective:

$$L_{\text{SFM}}^{(i)}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1), p_1(\boldsymbol{c}_1),\\ p_0(\boldsymbol{c}_0), p_t(\boldsymbol{c}|\boldsymbol{c}_0,\boldsymbol{c}_1)}} \text{CE}\left(\boldsymbol{c} + (1-t)v_t(\boldsymbol{c}), \boldsymbol{c}_1\right),$$

where $\text{CE}(\cdot,\cdot)$ is cross-entropy, and the loss directly measures the difference between the true probability and the inferred one $\hat{\boldsymbol{c}}_1 = \boldsymbol{c} + (1-t)v_t(\boldsymbol{c})$. For the whole sequence, the loss function is a summation of $L_{\text{SFM}}^{(i)}(\theta)$ over i, as

$$L_{\text{SFM}}(\theta) = \sum_{i=1}^{N_{\text{pp}}} L_{\text{SFM}}^{(i)}(\theta). \tag{11}$$

3.5. Overall Training Loss

The overall loss function is the summation of the four loss functions in Equation. 7, 9, 10 and 11. To fully utilize the sufficient contextual information as conditional inputs, for each flow-matching vector field $v_t(\cdot)$, we instead use all the structure-sequence contexts as input. In this way, we will first sample $\boldsymbol{\tau}_t$, $\mathbf{x}_t^{(\text{C})}$, $O_t^{(\text{C})}$ and $\boldsymbol{s} = \{s_t^{(i)}\}_{i=1}^{N_{\text{pp}}}$ through the defined conditional probability paths, and then employ NERF (Parsons et al., 2005) to reconstruct the sequence and backbone structure in local frame, and translate and rotate the structures such that its centroid coordinate and orientation equals to $\mathbf{x}_t^{(\text{C})}$ and $O_t^{(\text{C})}$. Thus, we can obtain $\mathcal{L}_t^* = \{(s_t^{(i)}, X_t^{(i)*}) : i \in \mathcal{I}_{\text{pp}}\}$. Finally, the whole binding complex at t is used as inputs of $v_t(\cdot)$, as $v_t(\cdot|\mathcal{L}_t^* \cup \mathcal{R}) = v_t(\cdot|\mathcal{C}_t^*)$. Therefore, our overall training loss is written as

$$L_{\text{PPF}} = \mathbb{E}_{\mathcal{C}^*}(L_{\text{TFM}} + L_{\text{EFM}} + L_{\text{OFM}} + L_{\text{SFM}}), \tag{12}$$

in which $L_{\text{TFM}} = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1), p_1(\boldsymbol{\tau}_1),\\ p_0(\boldsymbol{\tau}_0), p_t(\boldsymbol{\tau}|\boldsymbol{\tau}_0,\boldsymbol{\tau}_1)}} \|v_t(\boldsymbol{\tau};\mathcal{C}^*) - \boldsymbol{\tau}_1' + \boldsymbol{\tau}_0\|_2^2$, and the other three can be written in a similar way. We write $\mathcal{C}^* = \text{rec}(\boldsymbol{\tau}, \mathbf{x}^{(\text{C})}, O^{(\text{C})}, \boldsymbol{s})$, meaning the input binding complexes are reconstructed from the four variables.

3.6. Sampling with ODE

For the trained flow-matching vector fields, we sampling process is the solution of the following ordinary the differential equation as $\frac{d}{dt}\mathcal{C}^* = v_t(\mathcal{C}^*)$, in which the defined method is employed as the numerical solution:

$$\begin{aligned}
\text{Step 1:}\quad & \boldsymbol{\tau}_{t+\Delta t} = \text{reg}(\boldsymbol{\tau}_t + v_t(\boldsymbol{\tau}_t;\mathcal{C}_t^*)\Delta t);\\
& \mathbf{x}_{t+\Delta t}^{(C)} = \mathbf{x}_t + v_t(\mathbf{x}_t^{(C)};\mathcal{C}_t^*\Delta t);\\
& O_{t+\Delta t}^{(\text{C})} = O_t^{(\text{C})} \exp_{O_t^{(\text{C})}}\left(v_t(O_t^{(\text{C})};\mathcal{C}_t^*)\Delta t\right);\\
& \boldsymbol{c}_{t+\Delta t}^{(i)} = \text{norm}(\boldsymbol{c}_t^{(i)} + v_t(\boldsymbol{c}_t^{(i)};\mathcal{C}_t^*)\Delta t)\\
& s_t^{(i)} \sim \boldsymbol{c}_t^{(i)} \qquad 1 \le i \le N_{\text{pp}};
\end{aligned} \tag{13}$$

$$\text{Step 2:}\quad \mathcal{C}_{t+\Delta t}^* = \text{rec}(\boldsymbol{\tau}_{t+\Delta t}, \mathbf{x}_{t+\Delta t}^{(\text{C})}, O_{t+\Delta t}^{(\text{C})}, \boldsymbol{s}_{t+\Delta t}), \tag{14}$$

where $\text{norm}(\cdot)$ means normalizing the vector to a probability vector such that the summation is 1, and $\text{reg}(\cdot)$ means regularize the angles by $\text{reg}(\boldsymbol{\tau}) = (\boldsymbol{\tau} + \pi) \bmod (2\pi) - \pi$.

3.7. Parameterization with Neural Networks

Encoder. We first adopt two multi-layer perceptrons (MLPs). The embedding of single residues is obtained by one MLP, which encodes the residue type, backbone dihedral angles, and local atom coordinates. The other MLP for residue pairs encodes the distance, the dihedral angles between the i and j residues, and their relative positions. Then, we stack 6-layer transformers with self-attention mechanisms used to update the inter-residues' embedding and obtain residues' embedding for each amino acid by $\boldsymbol{h}_i \in \mathbb{R}^D$.

Equivariance. The conditional distribution of $p(\mathcal{L}|\mathcal{R})$ have to be roto-translational equivariant to ensure the generalization, *i.e.* $p(O\mathcal{L} + \mathbf{x}|O\mathcal{R} + \mathbf{x}) = p(\mathcal{L}|\mathcal{R})$, for $\mathbf{x} \in \mathbb{R}^3$ and $O \in \text{SO}(3)$, where $O\mathcal{L} + \mathbf{x} = \{(s^{(i)}, OX^{(i)} + \mathbf{x})\}$. To avoid the problem of non-existence of such a probability due to translation, we here adopt the zero-mass-center techniques (Rudolph et al., 2020; Yim et al., 2023b), by subtracting the mass center of the conditional receptor from all inputs' coordinates to the neural network (Guan et al., 2023; Lin et al., 2022; Satorras et al., 2021), which also helps to improve the training stability. Further, in terms of CFM, the following proposition indicates the roto-translational equivariance of each flow-matching vector field.

Proposition 3.3. *Let $p_0(\boldsymbol{\tau})$, $p_0(\mathbf{x}^{(\text{C})})$, $p_0(O^{(\text{C})})$, and $p_t(\boldsymbol{c})$ be SE(3)-invariant distribution, and the flow-matching vector field $v_t(\mathbf{x}^{(\text{C})}|\mathcal{C}^*)$ be SE(3)-equivariant, $v_t(O^{(\text{C})}|\mathcal{C}^*)$ be T(3)-invariant and SO(3)-equivariant, and $v_t(\boldsymbol{c}|\mathcal{C}^*)$ and $v_t(\boldsymbol{\tau}|\mathcal{C}^*)$ be SE(3)-invariant, then the density $p(\mathcal{L}^*|\mathcal{R})$ generated by the ODE sampling process SE(3)-equivariant.*

We employ ITA (Jumper et al., 2021) with LOCS (Kofinas et al., 2022) (For details see Appendix. A.4), to ensure the equivariance and the invariance of the vector fields in 3.3.

3.8. Side-Chain Packing

For the complete solution to full-atom design, after the backbone atoms are generated, we employ a Rotamer Density Estimator (RDE) (Luo et al., 2023) as the probabilistic model to model the side-chain rotamers $\{\boldsymbol{\chi}_i\}_{i=1}^{N_{\text{pp}}}$. It uses a Conditional Flow on $\mathbb{T}^{N_{\text{rt}}}$ based on the rational quadratic spline flow (Durkan et al., 2019; Rezende et al., 2020), where N_{rt} is the total rotamer number. In practice, we use the pre-trained version of RDE capable of perceiving the side-chain conformations by training the model on large datasets of PDB-REDO (Joosten et al., 2014). Further, we conduct fine-tuning on our protein-peptide complex datasets.

The effectiveness of the side-chain packing model named RDE-PP, transferred from protein-protein complexes to the protein-peptide ones, is empirically shown in Sec. 5.6.

The overall workflow of PPFLOW including the backbone generation and side-chain packing is shown in Figure. 2.

4. Related Work

Structure-based drug design. Success in 3D molecule generation and increasing available structural data raises scientific interest in structure-based drug design (SBDD). Grid-based methods regard the Euclidean space as discrete and predict the molecules' structures on the grids (Luo et al., 2021; Masuda et al., 2020). Equivariant neural networks have advanced the structure predictions and helped to improve the tasks greatly (Peng et al., 2022b; Liu et al., 2022). Then, diffusion methods (Guan et al., 2023; Schneuing et al., 2022; Lin et al., 2022) attempt to generate the ordered atoms' positions and types at full atom levels. Further, a series of works of fragment-based drug design are proposed, by mimicking classical drug design procedures to generate both atoms and functional groups that form molecule drugs (Zhang et al., 2023b; Lin et al., 2023).

Protein generation. Peptides can be regarded as fragments making up a protein. Techniques on protein backbone generation assist realistic peptide structure design. For example, FoldingDiff (Wu et al., 2022) is also based on redundant angle generation, which employs diffusion models on torus to achieve backbone design. RFDiffusion and Chroma (Watson et al., 2023; Ingraham et al., 2022) use different diffusion schemes, and achieve state-of-the-art generation performance on protein generations. Recently, protein backbone generation methods employ flow matching techniques, to explore the applicability and effectiveness (Yim et al., 2023a; Bose et al., 2023). For protein side chains, methods usually focus on protein-protein complexes, such as RED-PPI (Luo et al., 2023) and DiffPack (Zhang et al., 2023a). Our side-chain packing methods follow RDE-PPI since it is pre-trained on a large amount of protein data, and achieves a good generalization performance on peptides.

5. Experiment

5.1. Dataset

Training set. To satisfy the need for massive data to train deep learning models, we construct `PPBench2024`, through a series of systematic steps: *First*, we source complexes from the RCSB database (Zardecki et al., 2016), specifically selecting those containing more than two chains and excluding any with nucleic acid structures, and defining interactions between a pair as a minimum intermolecular distance of 5.0Å or less. *Subsequently*, only those complexes featuring peptide chains that do not exceed 30 amino acids in length are included, to better mimic existing peptide drug sizes. *Then*, the water molecules and heteroatoms are eliminated. *Finally*, we select only peptide molecules composed entirely of amino acids, remove the modified peptides with functional groups other than amino acids, and filter peptides with broken bonds according to the ideal bond length, *i.e.* the bond is unbroken if the observed bond length is between the ideal length plus or minus 0.5Å. The number of final screened complex instances of peptide-protein pairs is 9070. Further, we screen the existing datasets of `PropediaV2.3` (Martins et al., 2023) and `PepBDB` (Wen et al., 2018) with the same criterion, leading to additional 6523 instances to expand it. Appendix. B.1 gives details. We split the `PPBench2024` into training and validation sets according to the clustering of the proteins that are closest to the peptide ligand via MMSEQS2 (Steinegger & Söding, 2017) with the ratio of $9:1$.

Test set. To evaluate the model performance, we use an existing benchmark dataset called `PPDBench` (Agrawal et al., 2019) consisting of 133 pairs with peptides' lengths ranging from 9 to 15 as the test set. We eliminate all complexes from `PPBench2024` whose PDB-ID are the same as those in `PPDBench` to avoid potential data leakage.

5.2. Baseline Models

We extend two models for the following tasks, including

(i) DIFFPP, as a variant of DIFFAB, which is a diffusion model for generating CDRs in antibodies targeted at antigens. DIFFPP parametrizes the protein backbones in the same way as ALPHAFOLD2, in which the positions of atoms in a residue are determined by Cα's translation vector $\mathbf{x}_{i,2}$ and the rotation matrix $O_i \in \mathrm{SO}(3)$ of the frame constructed by positions of $\{\mathrm{N}, \mathrm{C}\alpha, \mathrm{C}\}$. The diffusion and reverse process on translation and orientation variables are modeled by DDPM (Ho et al., 2020) and SO(3)-DPM (Leach et al., 2022). For amino acid types, it uses a multinomial diffusion (Hoogeboom et al., 2021).

(ii) DIFFBP-PP, as a variant of DIFFBP or TARGETDIFF, as an atom-level diffusion model for generating molecules that bind to specific proteins. It decomposes the residues into atoms, and uses DDPM to model the distribution of atoms' positions and D3PM (Austin et al., 2021) to predict amino acid types through mask language modeling techniques.

These two models are the state-of-the-art models for target-specific protein and molecule generation, respectively. For other classical models, we describe them in different tasks.

5.3. Peptide Generation

Metrics. We choose 6 metrics for evaluating the quality of generated peptides. $\boldsymbol{\Delta G}$ is binding energies calculated by ADCP re-docking (Zhang & Sanner, 2019), since ADCP has shown the best performance in estimating the binding

Table 1: Comparison for target-aware peptide generation. Values in **bold** are the best.

	$\Delta G(\downarrow)$	IMP%-B(↑)	IMP%-S(↑)	Validity(↑)	Novelty(↑)	Diversity
DIFFPP	-319.54	28.75%	7.56%	0.94	0.79	0.28
DIFFBP-PP	-221.06	19.34%	4.09%	0.42	0.76	0.41
PPFLOW-BB	-349.59	**36.02%**	10.34%	**1.00**	**0.84**	**0.76**
PPFLOW-FA	**-351.27**	35.43%	**12.88%**	**1.00**	**0.84**	**0.76**

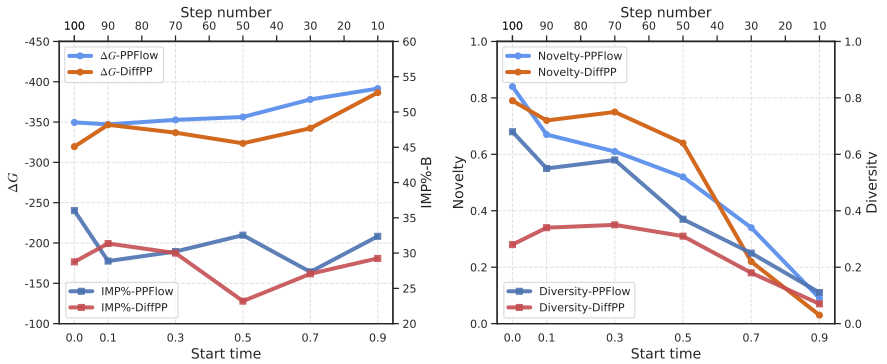


Figure 3: Metrics for generated peptides of methods in different optimization steps.

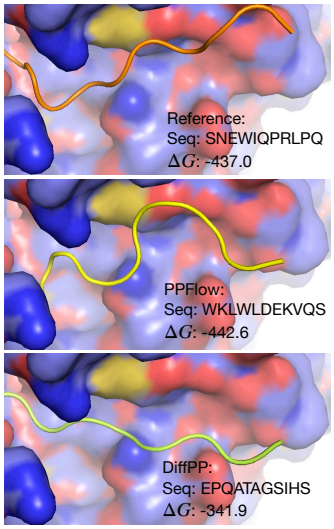


Figure 4: Peptides designed by different methods and reference.

energies and poses of protein-peptide complexes (Weng et al., 2020). ΔG reflects the potential of a peptide being pharmaceutically active towards the targets, and **IMP%-B** gives the percentages of the designed peptides with lower (better) ΔG than the reference peptides. The mean of reference ΔG is -427.72kcal/mol. Stability evaluates binding scores of the original pose of the generated peptides, which directly reflects the quality of generated peptides without re-docking. The stability scores are calculated by FOLDX (Schymkowitz et al., 2005) since it performs fast and accurate stability calculation in protein binding tasks compared with other energy-based methods (Luo et al., 2023). We give **IMP%-S** as the percentages of the designed ones with better stability than reference rather than average, because some extremely unstable structures will make the comparison on average values meaningless. **Validity** is the ratio of the designed peptides that is chemically valid, through the criterion of whether the bonds of the atoms that should be bonded are broken. We follow our filtering process of datasets, and define the bond is not broken if its length is within 0.5Å above and below the ideal value. **Novelty** is measured considering two aspects of structure and sequence: (i) the fraction of peptides with TM-score < 0.5 as used in (Lin & AlQuraishi, 2023) and (ii) the sequence overlap (SeqOL) less than 0.5. The given score is the mean ratio of novel peptides *v.s.* references. **Diversity** is the product of pairwise $(1-\text{TM-score})$ and $(1-\text{SeqOL})$ of the generated samples averaged across their target proteins. The higher the score, the more diverse peptides a model generates. Note that all the metrics except Validity are measured with chemically valid peptides.

Setup. For each model for comparison, we generate 20 peptides for each conditioned protein because the evaluation process requires excessive time. For example, ΔG obtained by ADCP requires more than 10 minutes for one pair of protein and peptides on a server with 128 CPU threads. Besides, since the greater the number of atoms, the more likely it is that binding interactions will occur, we eliminate the effects of peptide sizes (lengths) on the binding affinity by setting the generated sequence lengths the same as references.

Results. Table. 1 gives the comparison on peptide generation task with generated examples shown in Figure. 4. For PPFLOW, we test two variants of it. PPFLOW-BB (PPFlow-BackBone) uses the generated backbone structures with ROSETTA side-chain packing (Alford et al., 2017) before re-docking scoring and stability calculation, and side-chain atoms in PPFLOW-FA (PPFlow-FullAtom) is predicted by RDE. Note that the backbone structures before side-chain packing are the same for the two variants, leading to metrics except IMP%-S being of little difference. Besides, for the others, the side-chain atoms are constructed by ROSETTA. It can be concluded that (i) PPFLOW has the greatest potential to generate peptide drugs with high binding affinity towards the target protein according to ΔG and IMP%-B metrics. (ii) The complexes of peptides of original poses generated by PPFLOW and the target proteins are the most stable, with the highest IMP%-S metrics. (iii) PPFLOW and DIFFPP generate a comparably high ratio of novel peptides, while peptides generated by PPFLOW are more diverse. (iv) The rotamers estimated by RDE-PP are more stable in the binding site than ROSETTA according to IMP%-S metrics of PPFLOW-BB and PPFLOW-FA. (v) With the bond lengths and angles fixed, the structures generated by PPFLOW are chemically valid, while DIFFBP-PP shows the lowest Validity due to the highest degree of freedom it models, *i.e.* $4 \times 3N_{\text{pp}}$, for atom-level backbone structure, which also

Table 2: Comparison for flexible peptide re-docking.

Methods	L-RMSD(↓)			C-RMSD(↓)			Success%(↑)			Time (s)
	10%	30%	50%	10%	30%	50%	2	4	8	
DIFFPP	**8.01**	**15.26**	28.47	6.35	12.68	30.25	**9.24**	**20.21**	**34.45**	27
PPFLOW	12.44	17.82	31.24	**2.60**	**3.28**	**4.26**	6.13	11.24	22.47	46
HDOCK	9.95	18.65	**23.07**	7.29	10.65	18.84	6.40	10.72	25.50	186
VINA	11.38	19.03	27.69	6.27	9.56	16.59	5.45	9.96	21.12	1325

Table 3: Comparison for side-chain packing.

	Methods	χ_1	χ_2	χ_3	χ_4
MAE	ROSETTA	38.58	**39.41**	68.75	64.54
	RDE-PP(w/o pt)	44.28	51.22	70.21	71.85
	RDE-PP	**37.24**	47.67	**66.88**	**62.86**
NLL	RDE-PP(w/o pt)	0.75	1.26	3.37	4.10
	RDE-PP	**0.62**	**0.78**	**3.05**	**3.85**

empirically shows the infeasibility of SBDD methods to generalize to peptide drug design. Therefore, in the following parts, we will not compare DIFFBP-PP as baselines.

5.4. Peptide Optimization

Setup. We apply our model to another common pharmaceutical application: optimization of the existing peptides. To optimize a peptide, we first choose a start time $\text{start_time} \in [0, 1]$, and use the probability paths constructed in Sec. 3.2, 3.3 and 3.4, to sample a perturbed peptide $\mathcal{L}_{t_s} = \{X_{t_s}^{(i)*}, s_{t_s}^{(i)}\}_{i=1}^{N_{\text{PP}}}$. By using the ODE sampling methods in Sec. 3.6, the trained PPFLOW iteratively updates the sequences and structures to recover the peptides as a set of optimized ones. For DIFFPP, we use the same setting as (Luo et al., 2022). The 'num_step' is the number of how many optimization steps used in the generative diffusion process, where the start step of noised peptides is $(\text{total_step} - \text{time_step})$. The conversion between them is $\text{total_step} - \text{num_step} = \text{total_step} \times \text{start_time}$. We sample 20 new peptides for each protein receptor in PPDBench and evaluate them in the same metrics as discussed in Sec. 5.3. We set $\text{total_step} = 100$ in this part.

Results. Figure. 3 gives the comparison on peptide optimization results. It can be concluded that (i) The larger number of optimization steps contributes little to the improvement in the binding affinity, but PPFLOW usually generates more peptides with higher binding affinities; (ii) Optimized peptides are usually similar to the original one in small optimization steps since the novelty are smaller, which is desired in many practical applications; (iii) The diversity is more controllable in PPFLOW by changing the number of optimization steps.

5.5. Protein-Peptide Docking

Besides the two tasks, in this part, we generalize our model to the peptide flexible re-docking task to figure out if the model can learn the binding poses of the ligands. The task can be regarded as establishing a probabilistic model of $p(\{X^{(i)}\}_{i=0}^{N_{\text{PP}}} | \{s^{(i)}\}_{i=0}^{N_{\text{PP}}}, \mathcal{R}\})$. For DIFFPP and PPFLOW, $\{X_0^{(i)}\}$, as the initialized peptide structures, are obtained by first moving them to the pocket center, and then adding Gaussian noise to each atom. We retrain DIFFPP and PPFLOW with the sequences given as additional conditions. For classical docking methods, we choose HDOCK (Yan et al., 2020) which is a re-docking method for protein-protein interactions, and VINADOCK (Eberhardt et al., 2021) proposed for molecule docking. We choose the best 10%/30%/50% poses for comparison on ligand-RMSD (**L-RMSD**) between Cα atoms and centroid-RMSD (**C-RMSD**), and the success rate (**Success%**) of the docked ligands' RMSD smaller than 2/4/8Å. It can be concluded in Table. 2 that (i) DIFFPP shows the best docking performance on L-RMSD and Success% because it directly optimized the RMSD between Cα atoms, and PPFLOW models the ligands' centroid most accurately. (ii) The deep-learning-based generative models can achieve competitive performance with the classical energy-based ones, while the latter requires enormous rounds of iteration, costing dozens of times longer for computation.

5.6. Side-Chain Packing

To figure out the effectiveness of our fine-tuned model of RDE-PP, We compare it with two baseline methods Rosetta(fixbb) (Leman et al., 2019) and the unpretrained version RDE-PP(w/o pt). For each reference peptide, we initialize the rotamers randomly with uniform angle distribution, and sample 20 conformations on side chains. We evaluate the mean absolute error (**MAE**) of the predicted sidechain torsional angles as $\text{MAE} = |(\hat{\chi} - \chi) \bmod (2\pi)|$ where $\hat{\chi}$ is the estimated ones and χ is the ground truth, and negative likelihood (**NLL**). In Table. 3, the results demonstrate that the RDE-PP with pretraining outperforms the baselines on three of the four torsional angles in terms of the evaluated metrics.

6. Conclusion

In this paper, we focus on the target-specific peptide design tasks. To fulfill it, we first establish a dataset called PPBench2024, and then propose a flow-matching generative model on torus manifolds called PPFLOW, which attempts to directly learn the distribution of torsion angles of peptide backbones. Experiments are conducted to evaluate its performance on several tasks, indicating its superior performance on target-aware peptide design and optimization.

However, since the classical docking methods for binding pose calculation are extremely slow, we extend our model to docking tasks. Still, it is not as competitive as the baseline models, which will be our future focus. Besides, the performance gaps between the extended side-chain packing model and the classical ones are still small, urging us to develop a side-chain packing model with high prediction accuracy.

Impact Statements

With the long-term continuation of this COVID-19, more and more AI scientists are beginning to turn their research interests to drug design. The focus of this paper is on one of them – peptide drugs. There is very limited work aimed at using AI algorithms to design and optimize peptide drugs. However, peptide drugs are now raising as an effective therapeutics, with hundreds of them proves to be successful, such as semaglutide for obesity and diabetes. Here, this work presents a considerable contribution to establishing a large dataset and deep-learning model for protein-specific peptide design, in which the effectiveness is validated empirically. As far as we know, it is the first AI-assisted peptide drug design solution, so we here emphasize the social significance of the work and hope to get the attention of the reviewers and the review committee.

References

Agrawal, P., Singh, H., Srivastava, H. K., Singh, S., Kishore, G., and Raghava, G. P. Benchmarking of different molecular docking methods for protein-peptide docking. *BMC Bioinformatics*, 19, 2019.

Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., Labonte, J. W., Pacella, M. S., Bonneau, R., Bradley, P., Dunbrack, R. L., Das, R., Baker, D., Kuhlman, B., Kortemme, T., and Gray, J. J. The rosetta all-atom energy function for macromolecular modeling and design. *bioRxiv*, 2017.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Bose, A. J., Akhound-Sadegh, T., Fatras, K., Huguet, G., Rector-Brooks, J., Liu, C.-H., Nica, A. C., Korablyov, M., Bronstein, M., and Tong, A. Se(3)-stochastic flow matching for protein backbone generation. *ArXiv*, abs/2310.02391, 2023.

Chen, R. T. Q. and Lipman, Y. Riemannian flow matching on general geometries. *ArXiv*, abs/2302.03660, 2023.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. S. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kKF8_K-mBbS.

Creasey, P. E. and Lang, A. Fast generation of isotropic gaussian random fields on the sphere. *Monte Carlo Methods and Applications*, 24:1 – 11, 2017.

Dauparas, J., Anishchenko, I. V., Bennett, N. R., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N. P., Leung, P. J. Y., Huddy, T. F., Pellock, S. J., Tischer, D. K., Chan, F., Koepnick, B., Nguyen, H. A., Kang, A., Sankaran, B., Bera, A. K., King, N. P., and Baker, D. Robust deep learning based protein sequence design using proteinmpnn. *Science (New York, N.Y.)*, 378:49 – 56, 2022.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows, 2019.

Eberhardt, J., Santos-Martins, D., Tillack, A. F., and Forli, S. Autodock vina 1.2.0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 2021. URL https://api.semanticscholar.org/CorpusID:236092162.

Francoeur, P. G., Masuda, T., Sunseri, J., Jia, A., Iovanisci, R. B., Snyder, I., and Koes, D. R. 3d convolutional neural networks and a crossdocked dataset for structure-based drug design. *Journal of chemical information and modeling*, 2020.

Guan, J., Qian, W. W., Peng, X., Su, Y., Peng, J., and Ma, J. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kJqXEPXMsE0.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions, 2021.

Ingraham, J., Baranov, M., Costello, Z., Frappier, V., Ismail, A., Tie, S., Wang, W., Xue, V., Obermeyer, F., Beam, A., and Grigoryan, G. Illuminating protein space with a programmable generative model. *Nature*, 623:1070 – 1078, 2022.

Jantzen, R. T. Geodesics on the torus and other surfaces of revolution clarified using undergraduate physics tricks with bonus: Nonrelativistic and relativistic kepler problems, 2012.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation, 2023.

Joosten, R. P., Long, F., Murshudov, G. N., and Perrakis, A. The pdb_redo server for macromolecular structure model optimization. *IUCrJ*, 1:213 – 220, 2014.

Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D. A., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589, 2021.

Kelly, M., Lewis, J., Rao, H., Carter, J., Portillo, I. B., and Beuttler, R. Effects of glp-1 receptor agonists on cardiovascular outcomes in patients with type 2 diabetes and chronic kidney disease: A systematic review and meta-analysis. *Pharmacotherapy*, 42:921 – 928, 2022.

Kofinas, M., Nagaraja, N. S., and Gavves, E. Roto-translated local coordinate frames for interacting dynamical systems, 2022.

Köhler, J., Klein, L., and Noe, F. Equivariant flows: Exact likelihood generative learning for symmetric densities. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5361–5370. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kohler20a.html.

Leach, A., Schmon, S. M., Degiacomi, M. T., and Willcocks, C. G. Denoising diffusion probabilistic models on SO(3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. URL https://openreview.net/forum?id=BY88eBbkpe5.

Leman, J. K., Weitzner, B. D., Lewis, S. M., Consortium, R., and Bonneau, R. Macromolecular modeling and design in rosetta: New methods and frameworks. 2019. URL https://api.semanticscholar.org/CorpusID:241255608.

Lin, H., Huang, Y., Liu, M., Li, X. C., Ji, S., and Li, S. Z. Diffbp: Generative diffusion of 3d molecules for target protein binding. *ArXiv*, abs/2211.11214, 2022.

Lin, H., Huang, Y., Zhang, O., Wu, L., Li, S., Chen, Z., and Li, S. Z. Functional-group-based diffusion for pocket-specific molecule generation and elaboration, 2023.

Lin, Y. and AlQuraishi, M. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds, 2023.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022.

Liu, M., Luo, Y., Uchino, K., Maruhashi, K., and Ji, S. Generating 3d molecules for target protein binding. In *International Conference on Machine Learning*, 2022.

Luo, S., Guan, J., Ma, J., and Peng, J. A 3D generative model for structure-based drug design. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=jSorGn2Tjg.

Luo, S., Su, Y., Wu, Z., Su, C., Peng, J., and Ma, J. Rotamer density estimator is an unsupervised learner of the effect of mutations on protein-protein interaction. *bioRxiv*, 2023.

Makhatadze, G. I. Faculty opinions recommendation of accurate prediction of protein structures and interactions using a three-track neural network. *Faculty Opinions – Post-Publication Peer Review of the Biomedical Literature*, 2021.

Martins, P. M., Mariano, D. C. B., Carvalho, F. C., Bastos, L. L., Moraes, L., Paixão, V. M., and de Melo Minardi, R. C. Propedia v2.3: A novel representation approach for the peptide-protein interaction database using graph-based structural signatures. *Frontiers in Bioinformatics*, 3, 2023. URL https://api.semanticscholar.org/CorpusID:256978014.

Marullo, R., Kastantin, M., Drews, L. B., and Tirrell, M. Peptide contour length determines equilibrium secondary structure in protein-analogous micelles. *Biopolymers: Original Research on Biomolecules*, 99(9):573–581, 2013.

Masuda, T., Ragoza, M., and Koes, D. R. Generating 3d molecular structures conditional on a receptor binding site with deep generative models. *arXiv preprint arXiv:2010.14442*, 2020.

Muttenthaler, M., King, G. F., Adams, D. J., and Alewood, P. F. Trends in peptide drug discovery. *Nature Reviews Drug Discovery*, 20:309 – 325, 2021.

Padmanabhan, S. Handbook of pharmacogenomics and stratified medicine. 2014.

Parsons, J., Holmes, J., Rojas, J., Tsai, J., and Strauss, C. Practical conversion from torsion space to cartesian space for in silico protein synthesis. *Journal of computational chemistry*, 26:1063–8, 07 2005. doi: 10.1002/jcc.20237.

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International Conference on Machine Learning*, 2022a.

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International Conference on Machine Learning*, 2022b.

Rezende, D. J., Papamakarios, G., Racanière, S., Albergo, M. S., Kanwar, G., Shanahan, P. E., and Cranmer, K. Normalizing flows on tori and spheres, 2020.

Rudolph, M., Wandt, B., and Rosenhahn, B. Same same but differnet: Semi-supervised defect detection with normalizing flows. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1906–1915, 2020.

Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I., and Welling, M. E(n) equivariant normalizing flows. In *Neural Information Processing Systems*, 2021.

Schneuing, A., Du, Y., Harris, C., Jamasb, A., Igashov, I., Du, W., Blundell, T., Lió, P., Gomes, C., Welling, M., Bronstein, M., and Correia, B. Structure-based drug design with equivariant diffusion models, 2022.

Schymkowitz, J., Ferkinghoff-Borg, J., Stricher, F., Nys, R., Rousseau, F., and Serrano, L. The foldx web server: an online force field. *Nucleic Acids Research*, 33:W382 – W388, 2005.

Seebach, D., Hook, D. F., and Glättli, A. Helices and other secondary structures of β-and γ-peptides. *Peptide Science: Original Research on Biomolecules*, 84(1):23–37, 2006.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Todaro, B., Ottalagana, E., Luin, S., and Santi, M. Targeting peptides: The new generation of targeted drug delivery systems. *Pharmaceutics*, 15, 2023.

Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport, 2023.

Wang, L., Wang, N., Zhang, W., Cheng, X., Yan, Z., Shao, G., Wang, X., Wang, R., and Fu, C. Therapeutic peptides: current applications and future directions. *Signal Transduction and Targeted Therapy*, 7, 2022.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., Bortoli, V. D., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T., DiMaio, F., Baek, M., and Baker, D. De novo design of protein structure and function with rfdiffusion. *Nature*, 620:1089 – 1100, 2023.

Wen, Z., He, J., Tao, H., and Huang, S. Pepbdb: a comprehensive structural database of biological peptide-protein interactions. *Bioinformatics*, 35 1:175–177, 2018. URL https://api.semanticscholar.org/CorpusID:51600791.

Weng, G., Gao, J., Wang, Z., Wang, E., Hu, X., Yao, X., Cao, D., and Hou, T. Comprehensive evaluation of fourteen docking programs on protein-peptide complexes and a new strategy for improving the performance of global docking. *Journal of chemical theory and computation*, 2020.

Wu, K. E., Yang, K. K., van den Berg, R., Zou, J., Lu, A. X., and Amini, A. P. Protein structure generation via folding diffusion. *ArXiv*, abs/2209.15611, 2022.

Yan, Y., Tao, H., He, J., and Huang, S. The hdock server for integrated protein–protein docking. *Nature Protocols*, 15:1829 – 1852, 2020. URL https://api.semanticscholar.org/CorpusID:215411844.

Yim, J., Campbell, A., Foong, A. Y. K., Gastegger, M., Jiménez-Luna, J., Lewis, S., Satorras, V. G., Veeling, B. S., Barzilay, R., Jaakkola, T., and Noé, F. Fast protein backbone generation with se(3) flow matching, 2023a.

Yim, J., Trippe, B. L., Bortoli, V. D., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se(3) diffusion model with application to protein backbone generation, 2023b.

Zardecki, C., Dutta, S., Goodsell, D. S., Voigt, M., and Burley, S. K. Rcsb protein data bank: A resource for chemical, biochemical, and structural explorations of large and small biomolecules. *Journal of Chemical Education*, 93: 569–575, 2016.

Zhang, Y. and Sanner, M. F. Autodock crankpep: combining folding and docking to predict protein-peptide complexes. *Bioinformatics*, 2019.

Zhang, Y., Zhang, Z., Zhong, B., Misra, S., and Tang, J. Diffpack: A torsional diffusion model for autoregressive protein side-chain packing, 2023a.

Zhang, Z., Zheng, S., Min, Y., and Liu, Q. Molecule generation for target protein binding with structural motifs. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=Rq13idF0F73.

A. Method

A.1. Proof of Proposition. 3.1

By Equation. 3, we can obtain $\boldsymbol{\epsilon} = \frac{\boldsymbol{\tau}_t - \boldsymbol{\mu}_t}{\sigma_t}$. Besides, from the definition of $u_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1) = d\dot{\boldsymbol{\tau}}/dt$, it can be written as

$$\begin{aligned} u_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1) =& \dot{\sigma}_t \boldsymbol{\epsilon} + \dot{\boldsymbol{\mu}}_t \\ =& \dot{\sigma}_t (\frac{\boldsymbol{\tau}_t - \boldsymbol{\mu}_t}{\sigma_t}) + \dot{\boldsymbol{\mu}}_t \\ =& \frac{\dot{\sigma}_t(\boldsymbol{\tau}_1)}{\sigma_t(\boldsymbol{\tau}_1)} (\boldsymbol{\tau} - \boldsymbol{\mu}_t(\boldsymbol{\tau}_1)) + \dot{\boldsymbol{\mu}}_t(\boldsymbol{\tau}_1). \end{aligned} \tag{15}$$

A.2. Proof of Proposition. 3.2

Firstly, we claim the disintegration of measures of $p_t(\boldsymbol{\tau})$, as $p_t(\boldsymbol{\tau}) = \prod_i p(\tau^{(i)})$. For $p_0(\boldsymbol{\tau})$, the disintegration satisfies. For $p_1(\boldsymbol{\tau})$, our parametrization assumes the torsion angles are orthogonal and the distribution is independent, so it also satisfies disintegration. In this way, it is easy to obtain an intermediate probability p_t satisfies disintegration, and similar for the conditional probability $p_t(\boldsymbol{\tau}|\boldsymbol{\tau}_1)$.

Then we can factorizes the metric on $\mathbb{T}^N$ into $\mathbb{S} \times \cdots \times \mathbb{S}$, and $p(\tau) \in \mathcal{P}(\mathbb{S})$.

Let $u_t = \mathbb{E}_{\tau_0 \sim p_0, \tau_1 \sim p_1} \left[\frac{p_t(\tau|\tau_0,\tau_1)}{p_t(\tau)} u_t(\tau|\tau_0,\tau_1) \right]$, and

$$\begin{aligned} &\nabla_\theta (\mathbb{E}_{\tau_0 \sim p_0, \tau_1 \sim p_1, \tau \sim p_t(\tau|\tau_0,\tau_1)} [\|v_t(\tau) - u_t(\tau|\tau_0,\tau_1)\|^2] - \mathbb{E}_{\tau \sim p_t} [\|v_t(\tau) - u_t(\tau)\|^2]) \\ =& -2\nabla_\theta (\mathbb{E}_{\tau_0 \sim p_0, \tau_1 \sim p_1, \tau \sim p_t(\tau|\tau_0,\tau_1)} \langle v_t(\tau), u_t(\tau|\tau_0,\tau_1)\rangle - \mathbb{E}_{\tau \sim p_t} \langle v_t(\tau), u_t(\tau)\rangle). \end{aligned} \tag{16}$$

Then,

$$\begin{aligned} &\mathbb{E}_{\tau \sim p_t} \langle v_t(\tau), u_t(\tau)\rangle \\ =& \int \langle v_t(\tau), u_t(\tau)\rangle p_t(\tau) d\tau \\ =& \int \langle v_t(\tau), \mathbb{E}_{p_0,p_1} \left[\frac{p_t(\tau|\tau_0,\tau_1)}{p_t(\tau)} u_t(\tau|\tau_0,\tau_1) \right] \rangle p_t(\tau) d\tau \\ =& \int \langle v_t(\tau), \int \frac{p_t(\tau|\tau_0,\tau_1)}{p_t(\tau)} u_t(\tau|\tau_0,\tau_1) p_0(\tau_0) p_1(\tau_1) d\tau_0 \tau_1 \rangle p_t(\tau) d\tau \\ =& \int \int \langle v_t(\tau), u_t(\tau|\tau_0,\tau_1)\rangle p_t(\tau|\tau_0,\tau_1) p_0(\tau_0) p_1(\tau_1) d\tau d\tau_0 d\tau_1 \\ =& \mathbb{E}_{\tau_0 \sim p_0, \tau_1 \sim p_1, \tau \sim p_t(\tau|\tau_0,\tau_1)} \langle v_t(\tau), u_t(\tau)\rangle, \end{aligned} \tag{17}$$

where in the last equality we change the order of integration. It proves Equation. 16 equals 0.

A.3. Proof of Proposition. 3.3

The proof is inspired by (Köhler et al., 2020; Lin et al., 2023), as follows:

Lemma A1. Let $\mathrm{T}_g(\cdot)$ be the operation in $\mathrm{SE}(3)$, if the following update function in the ODE sampler (Sec. 3.6) for the atom level's positions are defined as

$$v_t(\{X^{(i)*}\}|\mathcal{C}_t^*) = (\{X_{t+\Delta t}^{(i)*}\} - \{X_t^{(i)*}\})/\Delta t, \tag{18}$$

in which $\{X_t^{(i)*}\} = O_t \mathrm{nerf}(\boldsymbol{\tau}_t) + \mathbf{x}^{(\mathrm{C})}$. The invariance and equivariance of the following functions in the updating process are listed as

$$\begin{aligned} v\left(\{X_t^{(i)}\}|\mathrm{T}_g(\mathcal{C}^*)\right) &= \mathrm{T}_g\left(v(\{X_t^{(i)}\}|\mathcal{C}^*)\right); \\ v\left((\{s_t^{(i)}\})|\mathrm{T}_g(\mathcal{C}^*)\right) &= v\left((\{s_t^{(i)}\})|\mathcal{C}^*)\right), \end{aligned} \tag{19}$$

and the prior distribution as

$$p_0(\{X_0^{(i)*}\}|\mathrm{T}_g(\mathcal{R})) = p_0(\{X_0^{(i)*}\}|\mathcal{R}) \tag{20}$$

$$p_0(\{s_0^{(i)*}\}|\mathrm{T}_g(\mathcal{R})) = p_0(\{s_0^{(i)*}\}|\mathcal{R}). \tag{21}$$

Then the distribution which the final structures and sequences are sampled from as $p\left(\mathcal{L}|\mathcal{R}\right)$ is SE(3)-equivariant.

Proof: In the following, we write $\boldsymbol{X} = \{X^{(i)*}\}_{i=1}^{N_{\mathrm{PP}}}$, and $\boldsymbol{s} = \{s^{(i)}\}_{i=1}^{N_{\mathrm{PP}}}$ for notation simplicity. For the defined v_t, we first obtain that the update process is equivariant, as

$$\begin{aligned}
&v_t(\boldsymbol{X}|\mathrm{T}_g(\mathcal{C}_t^*))\Delta t + \mathrm{T}_g(\boldsymbol{X}_t) \\
=&\mathrm{T}_g\left(v_t(\boldsymbol{X}|\mathcal{C}_t^*)\right)\Delta t + \mathrm{T}_g(\boldsymbol{X}_t) \\
=&\mathrm{T}_g\left(v_t(\boldsymbol{X}|\mathcal{C}_t^*)\Delta t + \boldsymbol{X}_t\right) \\
=&\mathrm{T}_g\left(\boldsymbol{X}_{t+\Delta t}\right)
\end{aligned} \tag{22}$$

and for each transition kernel $p(\boldsymbol{X}_{t+\Delta t}|\mathcal{C}_t^*)$, it is SE(3)-equivariant, since

$$p(\mathrm{T}_g(\boldsymbol{X}_{t+\Delta t})|\mathrm{T}_g(\mathcal{C}_t^*)) \tag{23}$$

$$=p(\mathrm{T}_g(\boldsymbol{X}_t + v_t(\boldsymbol{X}|\mathrm{T}_g(\mathcal{C}_t^*))\Delta t)|\mathrm{T}_g(\mathcal{C}_t^*)) \tag{24}$$

$$= \int p(\mathrm{T}_g(\boldsymbol{X}_{t+\Delta t}) - \mathrm{T}_g(\boldsymbol{X}_t), \mathrm{T}_g(\boldsymbol{X}_t))|\mathrm{T}_g(\mathcal{C}_t^*))d\mathrm{T}_g(\boldsymbol{X}_t) \tag{25}$$

$$= \int p(\mathrm{T}_g(\boldsymbol{X}_{t+\Delta t}) - \mathrm{T}_g(\boldsymbol{X}_t)|\mathrm{T}_g(\mathcal{C}_t^*))p(\mathrm{T}_g(\boldsymbol{X}_t)|\mathrm{T}_g(\mathcal{C}_t^*))d\boldsymbol{X}_t \tag{26}$$

$$= \int p(v_t(\boldsymbol{X}|\mathrm{T}_g(\mathcal{C}_t^*))\Delta t|\mathrm{T}_g(\mathcal{C}_t^*))p(\boldsymbol{X}_t|\mathcal{C}_t^*)d\boldsymbol{X}_t \tag{27}$$

$$= \int p(v_t(\boldsymbol{X}|\mathcal{C}_t^*)\Delta t|\mathcal{C}_t^*)p(\boldsymbol{X}_t|\mathcal{C}_t^*)d\boldsymbol{X}_t \tag{28}$$

$$= \int p(\boldsymbol{X}_{t+\Delta t} - \boldsymbol{X}_t\}|\mathcal{C}_t^*)p(\boldsymbol{X}_t|\mathcal{C}_t^*)d\boldsymbol{X}_t \tag{29}$$

$$=p(\boldsymbol{X}_{t+\Delta t}|\mathcal{C}_t^*) \tag{30}$$

And it is easy to obtained that the transition kernel $p(\boldsymbol{s}_{t+\Delta t}|\mathcal{C}_t^*)$ is SE(3)-invariant, since $v_t(\boldsymbol{s}|\mathcal{C}_t^*)$ is invariant. Besides, for $p_0(\{X_0^{(i)*}\})$ and $p_0(\{s_0^{(i)*}\})$, it is a SE(3)-invariant distribution. Therefore,

$$p\left(\mathrm{T}_g(\boldsymbol{X}_1), \boldsymbol{s}_1|\mathrm{T}_g(\mathcal{R})\right) \tag{31}$$

$$= \int p\left(\mathrm{T}_g(\boldsymbol{X}_0), \boldsymbol{s}_0|\mathrm{T}_g(\mathcal{R})\right) \prod_{k=1}^{K} p\left(\mathrm{T}_g(\boldsymbol{X}_{k\Delta t}), \boldsymbol{s}_{k\Delta t}|\mathrm{T}_g(\mathcal{C}_{(k-1)\Delta t}^*)\right) d\mathcal{L}_{0:(K-1)\Delta t} \tag{32}$$

$$= \int p\left(\boldsymbol{X}_0, \boldsymbol{s}_0|\mathcal{R}\right) \prod_{k=1}^{K} p\left(\mathrm{T}_g(\boldsymbol{X}_{k\Delta t})|\mathrm{T}_g(\mathcal{C}_{(k-1)\Delta t}^*)\right) p\left(\boldsymbol{s}_{k\Delta t}|\mathrm{T}_g(\mathcal{C}_{(k-1)\Delta t}^*)\right) d\mathcal{L}_{0:(K-1)\Delta t} \tag{33}$$

$$= \int p\left(\boldsymbol{X}_0, \boldsymbol{s}_0|\mathcal{R}\right) \prod_{k=1}^{K} p\left(\boldsymbol{X}_{k\Delta t}|\mathcal{C}_{(k-1)\Delta t}^*\right) p\left(\boldsymbol{s}_{k\Delta t}|\mathcal{C}_{(k-1)\Delta t}^*\right) d\mathcal{L}_{0:(K-1)\Delta t} \tag{34}$$

$$= \int p\left(\boldsymbol{X}_0, \boldsymbol{s}_0|\mathcal{R}\right) \prod_{k=1}^{K} p\left(\boldsymbol{X}_{k\Delta t}, \boldsymbol{s}_{k\Delta t}|\mathcal{C}_{(k-1)\Delta t}^*\right) d\mathcal{L}_{0:(K-1)\Delta t} \tag{35}$$

$$=p\left(\boldsymbol{X}_1, \boldsymbol{s}_1|\mathcal{R}\right), \tag{36}$$

where K is the step number of the ODE sampler, and $\Delta t = 1/K$. Because $p\left(\mathrm{T}_g(\boldsymbol{X}_1), \boldsymbol{s}_1|\mathrm{T}_g(\mathcal{R})\right) = p\left(\boldsymbol{X}_1, \boldsymbol{s}_1|\mathcal{R}\right)$, which is equivalent to $p\left(\mathrm{T}_g(\mathcal{L}_1^*)|\mathrm{T}_g(\mathcal{R})\right) = p\left(\mathcal{L}_1^*|\mathcal{R}\right)$, the Lemma is proved.

Then we write $\{X_t^{(i)*}\} = O_t\mathrm{nerf}(\boldsymbol{\tau}_t) + \mathbf{x}_t^{(\mathrm{C})}$, and the following proposition can give the conditions that the transition kernels of $p(O_{t+\Delta t}|\mathcal{C}_t^*)$, $p(\mathbf{x}_{t+\Delta t}^{\mathrm{C}}|\mathcal{C}_t^*)$ and $p(\boldsymbol{\tau}_{t+\Delta t}|\mathcal{C}_t^*)$ and $p(\boldsymbol{s}_{t+\Delta t}|\mathcal{C}_t^*)$ should satisfiy.

Lemma A2. If $p(O_0^{(\mathrm{C})})$, $p(\mathbf{x}_0^{(\mathrm{C})})$, $p(\boldsymbol{\tau}_0)$ and $p(\boldsymbol{s}_0)$ are SE(3)-invariant, and $p(O_{t+\Delta t}^{(\mathrm{C})}|\mathcal{C}_t)$ is SO(3)-equivariant and T(3)-invariant, $p(\mathbf{x}_{t+\Delta t}^{\mathrm{C}}|\mathcal{C}_t^*)$ is SE(3)-equivariant, $p(\boldsymbol{\tau}_{t+\Delta t}|\mathcal{C}_t^*)$ and $p(\boldsymbol{s}_{t+\Delta t}|\mathcal{C}_t^*)$ is SE(3)-invariant, then $p\left(\mathrm{T}_g(\mathcal{L}_1^*)|\mathrm{T}_g(\mathcal{R})\right) = p\left(\mathcal{L}_1^*|\mathcal{R}\right)$, where $\mathcal{L}_1^* = (O_1^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_1) + \mathbf{x}_1^{(\mathrm{C})}, \boldsymbol{s}_1)$.

Proof: Here for T_g, we can decompose it as $\mathrm{T}_g = \mathrm{T}_r \circ \mathrm{T}_t$, meaning the rotation and translation opeartions as $\mathrm{SE}(3) \cong \mathrm{SO}(3) + \mathrm{T}(3)$. By this mean,

$$
\begin{aligned}
& p\left(\mathrm{T}_g(\mathcal{L}_1^*)|\mathrm{T}_g(\mathcal{R})\right) && (37)\\
=& p\left(\mathrm{T}_g(O_1^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_1) + \mathbf{x}_1^{(\mathrm{C})}), \boldsymbol{s}_1)|\mathrm{T}_g(\mathcal{R})\right) && (38)\\
=& p\left(\mathrm{T}_g(O_1^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_1)) + \mathrm{T}_g(\mathbf{x}_1^{(\mathrm{C})})), \boldsymbol{s}_1|\mathrm{T}_g(\mathcal{R})\right) && (39)\\
=& p\left(\mathrm{T}_g(O_1^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_1))|\mathrm{T}_g(\mathcal{R})\right) p\left(\mathrm{T}_g(\mathbf{x}_1^{(\mathrm{C})}))|\mathrm{T}_g(\mathcal{R})\right) p\left(\boldsymbol{s}_1|\mathrm{T}_g(\mathcal{R})\right) && (40)\\
=& p\left(\mathrm{T}_g(O_1^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_1))|\mathrm{T}_g(\mathcal{R})\right) p\left(\mathrm{T}_g(\mathbf{x}_1^{(\mathrm{C})}))|\mathrm{T}_g(\mathcal{R})\right) p\left(\boldsymbol{s}_1|\mathrm{T}_g(\mathcal{R})\right). && (41)
\end{aligned}
$$

Because $\mathrm{nerf}(\boldsymbol{\tau}_1)$ is always reconstructed with unit rotation $\mathrm{diag}(1,1,1)$ and zero-mass-centered, therefore,

$$
\begin{aligned}
& p\left(\mathrm{T}_g(O_1^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_1))|\mathrm{T}_g(\mathcal{R})\right) && (42)\\
=& p\left(\mathrm{T}_g(O_1^{(\mathrm{C})})\mathrm{nerf}(\boldsymbol{\tau}_1)|\mathrm{T}_g(\mathcal{R})\right) && (43)\\
=& p\left(\mathrm{T}_r \circ \mathrm{T}_t(O_1^{(\mathrm{C})})\mathrm{nerf}(\boldsymbol{\tau}_1)|\mathrm{T}_g(\mathcal{R})\right) && (44)\\
=& p\left(\mathrm{T}_r(O_1^{(\mathrm{C})})\mathrm{nerf}(\boldsymbol{\tau}_1)|\mathrm{T}_g(\mathcal{R})\right) && (45)\\
=& \int p\left(\mathrm{T}_r(O_0^{(\mathrm{C})})\mathrm{nerf}(\boldsymbol{\tau}_0)|\mathrm{T}_g(\mathcal{R})\right) \prod_{k=1}^{K} p\left(\mathrm{T}_r(O_{k\Delta t}^{(\mathrm{C})})\mathrm{nerf}(\boldsymbol{\tau}_{k\Delta t})|\mathrm{T}_g(\mathcal{C}_{(k-1)\Delta t}^*)\right) dM_{0:(K-1)\Delta t} && (46)\\
=& \int p\left(O_0^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_0)|\mathrm{T}_g(\mathcal{R})\right) \prod_{k=1}^{K} p\left(O_{k\Delta t}^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_{k\Delta t})|\mathcal{C}_{(k-1)\Delta t}^*\right) dM_{0:(K-1)\Delta t} && (47)\\
=& p\left(O_1^{(\mathrm{C})}\mathrm{nerf}(\boldsymbol{\tau}_1)|\mathcal{R}\right) && (48)
\end{aligned}
$$

where $dM_{0:(K-1)\Delta t} = dO_{0:(K-1)\Delta t}^{(\mathrm{C})} d\boldsymbol{\tau}_{0:(K-1)\Delta t}$. For $p\left(\mathrm{T}_g(\mathbf{x}_1^{(\mathrm{C})}))\right)$ and $p\left(\boldsymbol{s}_1|\mathrm{T}_g(\mathcal{R})\right)$, the equivariance and invariance are also easy to obtain. Therefore, the Lemma A2. is proved.

Finally, **PROOF OF PROPOSITION 3.3**:

For the transition kernels which are generated by the following equations as

$$
\begin{aligned}
p(\boldsymbol{\tau}_{t+\Delta t}|\mathcal{C}_t^*) &= p(\boldsymbol{\tau}_t + v_t(\boldsymbol{\tau}|\mathcal{C}_t^*)\Delta t|\mathcal{C}_t^*);\\
p(\mathbf{x}_{t+\Delta t}^{(\mathrm{C})}|\mathcal{C}_t^*) &= p(\mathbf{x}_t^{(\mathrm{C})} + v_t(\mathbf{x}^{(\mathrm{C})}|\mathcal{C}_t^*)\Delta t|\mathcal{C}_t^*);\\
p(O_{t+\Delta t}^{(\mathrm{C})}|\mathcal{C}_t^*) &= p(O_t^{(\mathrm{C})}\exp_{O_t^{(\mathrm{C})}}(v_t(\boldsymbol{\tau}|\mathcal{C}_t^*)\Delta t)|\mathcal{C}_t^*);\\
p(\boldsymbol{s}_{t+\Delta t}|\mathcal{C}_t^*) &= p(\boldsymbol{s}_t + v_t(\boldsymbol{s}|\mathcal{C}_t^*)\Delta t|\mathcal{C}_t^*),
\end{aligned} \tag{49}
$$

if the conditions of flow-matching vector fields in **Propostion 3.3** are all satisfied, then $p(\boldsymbol{\tau}_{t+\Delta t}|\mathcal{C}_t^*)$, $p(\mathbf{x}_{t+\Delta t}^{(\mathrm{C})}|\mathcal{C}_t^*)$, $p(O_{t+\Delta t}|\mathcal{C}_t^*)$ and $p(\boldsymbol{s}_{t+\Delta t}|\mathcal{C}_t^*)$ satisfies the conditions in **Lemma A.2** resepectively, and $v(\boldsymbol{x}|\mathcal{C}_t^*)$ satisfies **Lemma A.1**. From both perspectives, the roto-translational equivariance can be proved.

We demonstarte the first proof path, to shown that the conditions of **Lemma A.2** holds.

For $\boldsymbol{\tau}$:

$$
\begin{aligned}
p(\mathrm{T}_g(\boldsymbol{\tau}_{t+\Delta t})|\mathrm{T}_g(\mathcal{C}_t^*)) &= p(\mathrm{T}_g(\boldsymbol{\tau}_t + v_t(\boldsymbol{\tau}|\mathcal{C}_t^*)\Delta t)|\mathrm{T}_g(\mathcal{C}_t^*)) \\
&= p(\mathrm{T}_g(\boldsymbol{\tau}_t) + \mathrm{T}_g(v_t(\boldsymbol{\tau}|\mathcal{C}_t^*)\Delta t)|\mathrm{T}_g(\mathcal{C}_t^*)) \\
&= \int p(\mathrm{T}_g(\boldsymbol{\tau}_{t+\Delta t}) - \mathrm{T}_g(\boldsymbol{\tau}_t)|\mathrm{T}_g(\mathcal{C}_t^*))p(\mathrm{T}_g(\boldsymbol{\tau}_t)|\mathrm{T}_g(\mathcal{C}_t^*))d\mathrm{T}_g(\boldsymbol{\tau}_t) \\
&= \int p(\mathrm{T}_g(v_t(\boldsymbol{\tau}|\mathcal{C}_t^*)\Delta t)|\mathrm{T}_g(\mathcal{C}_t^*))p(\mathrm{T}_g(\boldsymbol{\tau}_t)|\mathrm{T}_g(\mathcal{C}_t^*))d\mathrm{T}_g(\boldsymbol{\tau}_t) \\
&= \int p(v_t(\boldsymbol{\tau}|\mathcal{C}_t^*)\Delta t|\mathcal{C}_t^*)p(\boldsymbol{\tau}_t|\mathcal{C}_t^*)d\boldsymbol{\tau}_t \\
&= \int p(\boldsymbol{\tau}_{t+\Delta t} - \boldsymbol{\tau}_t|\mathcal{C}_t^*)p(\boldsymbol{\tau}_t|\mathcal{C}_t^*)d\boldsymbol{\tau}_t \\
&= p(\boldsymbol{\tau}_{t+\Delta t}|\mathcal{C}_t^*)
\end{aligned}
\tag{50}
$$

Beside, for the other three variables, the equations can be deduced similarly.

A.4. LoCS Updating

Because the outputs should include the gradient vectors of rotation angle, global translational vector, rotation matrix, and type probability, we employ the LoCS to output the vector fields by

$$
\begin{aligned}
v_t(\boldsymbol{c}^{(i)}|\mathcal{C}^*) &= \mathrm{MLP}_s(\boldsymbol{h}_i); \\
v_t(\{\phi^{(i)}, \psi^{(i)}, \omega^{(i)}\}|\mathcal{C}^*) &= \mathrm{MLP}_\tau(\boldsymbol{h}_i); \\
v_t(\mathbf{x}^{(\mathrm{C})}|\mathcal{C}^*) &= \mathrm{MLP}_{\mathrm{x}}(\boldsymbol{h}_{\mathrm{G}})O_t; \\
v_t(O^{(\mathrm{C})}|\mathcal{C}^*) &= \mathrm{tran}_{O_t}(\mathrm{MLP}_O(\boldsymbol{h}_{\mathrm{G}})),
\end{aligned}
\tag{51}
$$

in which $\boldsymbol{h}_{\mathrm{G}} = \sum_i \boldsymbol{h}_i$ is the global representation obtained by summation, $\mathrm{MLP}_s : \mathbb{R}^D \to \mathbb{R}^{20}$, $\mathrm{MLP}_\tau : \mathbb{R}^D \to \mathbb{T}^3$, $\mathrm{MLP}_{\mathrm{x}} : \mathbb{R}^D \to \mathbb{R}^3$ and $\mathrm{MLP}_{\mathrm{x}} : \mathbb{R}^D \to \mathbb{R}^3$. MLP_O predicts a vector in Lie group $\mathfrak{so}(3)$, and translate it to O_t's tangent space by $\mathrm{tran}_{O_t}(\cdot)$. The output vector fields satisfy the equivariance and invariance conditions in Proposition. 3.3.

B. Dataset Statistics

B.1. Data preprocess

The construction of raw `PPBench2024` is given in the Sec. 5.1. For the additional complexes from `PropediaV2.3` and `PepBDB`, we give the detailed screening process in the Figure. 5. Because in PropediaV2.3, one peptide can be paired with several chains in a protein complex, another re-matching step should be conducted first.

Then, we give an empirical distribution on the peptide lengths of `PPBench2024`, in Figure. 6(a).

B.2. Analysis on Geometry

As the PPFLOW models the internal redundant geometry of the peptides, here we give a statistical illustration to show the flexible geometries that need to be generated. NeRF can use the following geometries in Figure. 6(b), 6(c) and 6(d) to reconstruct the full backbone structures. For these internal geometries, it can be concluded that the torsion angles of 'N-Cα-C-N' and 'C-N-Cα-C' are the two most flexible geometries. Besides, 'Cα-C-N-Cα' is theoretically inflexible since the constraints on peptide bond. However, in the observation, we find that it will deviate from the ideal value (π) a lot (about plus and minus 7°). In this way, we include it as another flexible geometries that the model needs to generate. These three torsion angles are named ϕ, ψ and ω, respectively in formal definition.

B.3. Experiment

Here we give the hyper-parameters and other training details. The learning rate lr is $5e-5$. In all training, the max training iteration is 200000. `LambdaLR` schedule is used, with `lr_lambda` is set as $0.95 \times lr$. The batch size is set 16 or 32, because it affects the performance little. In the neural networks, we set the MLP for extracting pair relations as 2 layers with

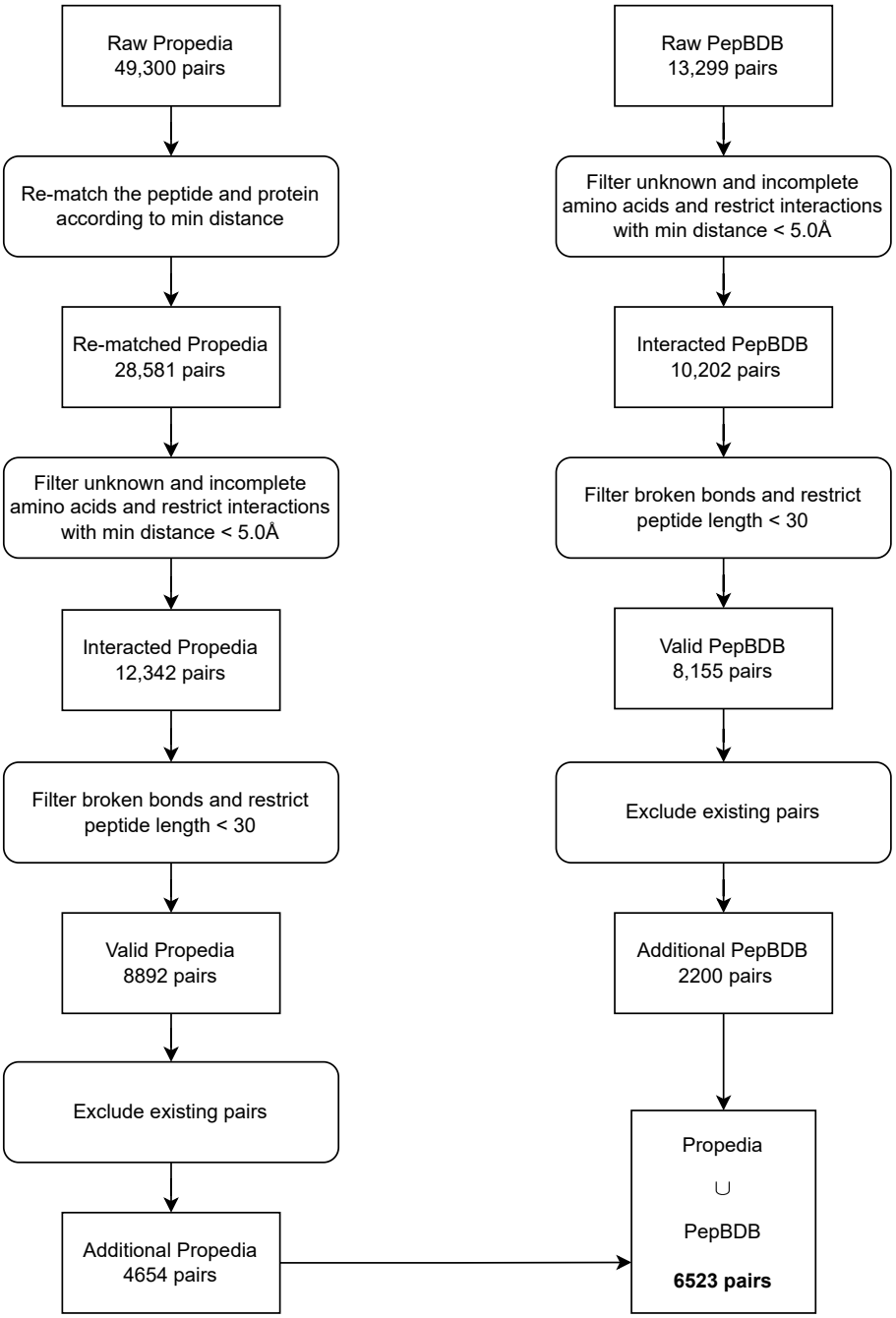
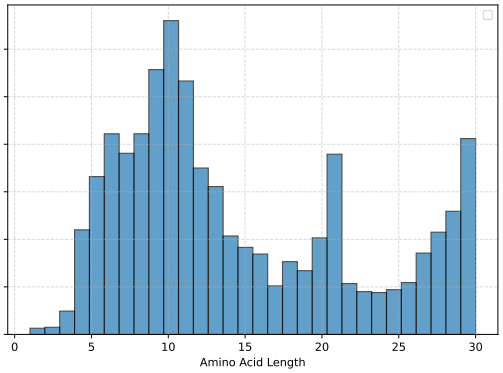
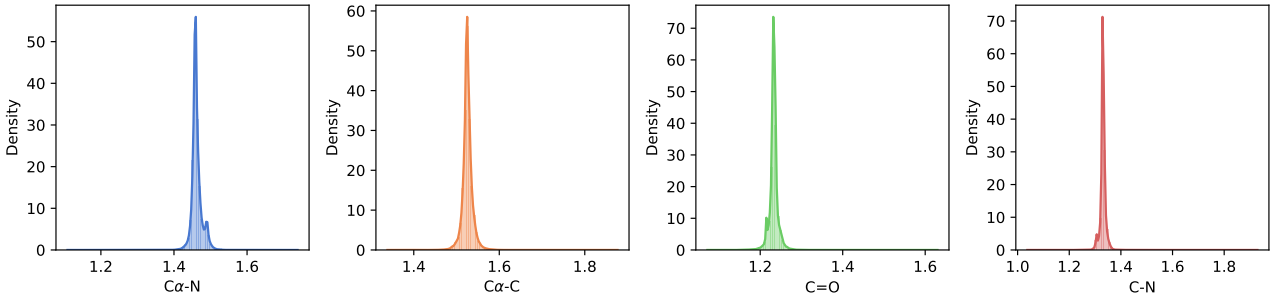


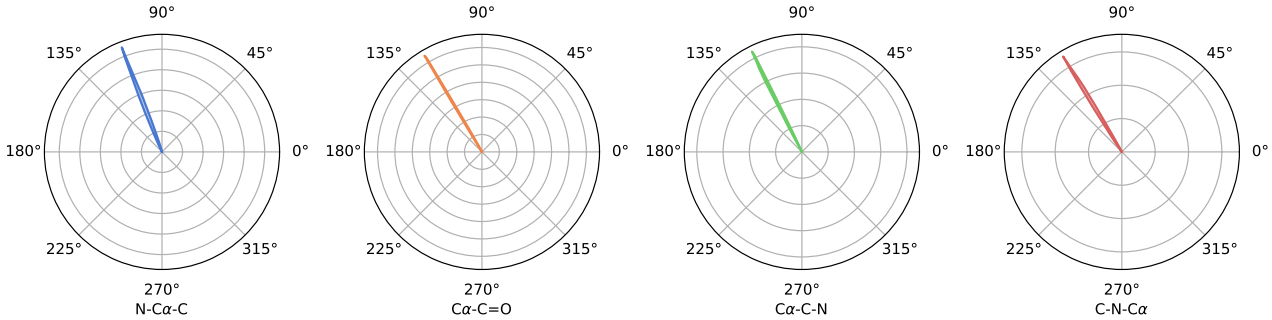
Figure 5: The process of screening the two datasets to expand the raw PPBench.



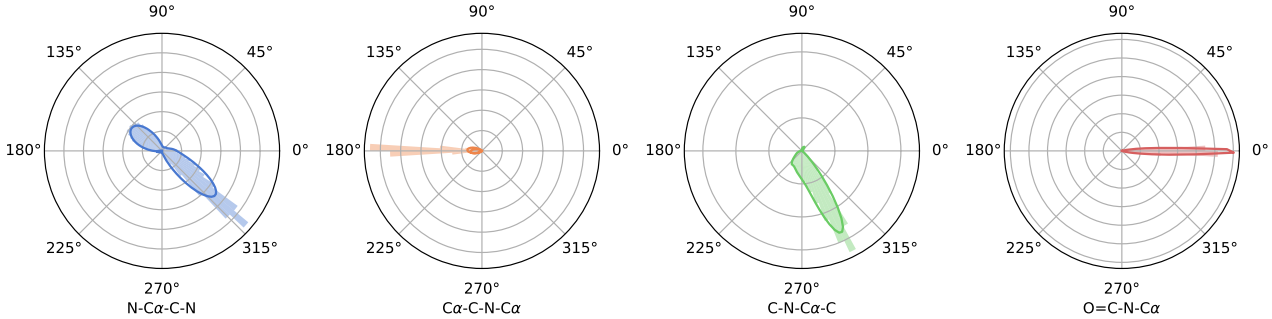
(a) Histogram of peptide lengths



(b) Empirical and estimated bond length distribution.



(c) Empirical and estimated bond angle distribution.



(d) Empirical and estimated torsion angle distribution.

Figure 6: Distributions of flexible and inflexible geometries obtained by peptides in `PPBench2024` datasets.

hidden dimension as 64, and the MLP for single amino acid as 2 layers with hidden dimension as 128. Following, 6 layers of transformer are stacked behind, and the final layer is the LOCS which has been discussed before.