Joint Design of Protein Surface and Structure Using a Diffusion Bridge Model

Guanlue Li
University of Hamburg
Hamburg, Germany
guanlue.li@uni-hamburg.de

Xufeng Zhao
University of Hamburg
Hamburg, Germany
xufeng.zhao@uni-hamburg.com

Fang Wu*
Stanford University
Stanford, CA, USA
fangwu97@stanford.edu

Sören Laue*
University of Hamburg
Hamburg, Germany
soeren.laue@uni-hamburg.de

Abstract

Protein-protein interactions (PPIs) are governed by surface complementarity and hydrophobic interactions at protein interfaces. However, designing diverse and physically realistic protein structure and surfaces that precisely complement target receptors remains a significant challenge in computational protein design. In this work, we introduce PepBridge, a novel framework for the joint design of protein surface and structure that seamlessly integrates receptor surface geometry and biochemical properties. Starting with a receptor surface represented as a 3D point cloud, PepBridge generates complete protein structures through a multi-step process. First, it employs denoising diffusion bridge models (DDBMs) to map receptor surfaces to ligand surfaces. Next, a multi-model diffusion model predicts the corresponding structure, while Shape-Frame Matching Networks ensure alignment between surface geometry and backbone architecture. This integrated approach facilitates surface complementarity, conformational stability, and chemical feasibility. Extensive validation across diverse protein design scenarios demonstrates PepBridge's efficacy in generating structurally viable proteins, representing a significant advancement in the joint design of top-down protein structure. The code can be found at https://github.com/guanlueli/Pepbridge.

1 Introduction

Proteins are fundamental biological macromolecules that perform their functions through intricate interactions with other biomolecules, particularly through protein-protein interactions (PPIs) [22]. PPIs are primarily determined by surface complementarity and hydrophobic interactions at the interface regions, which facilitate specific and stable binding [36]. Understanding and designing PPIs is a central challenge in computational protein design, which seeks to predict sequences, generate structures, and design proteins with tailored properties while adhering to biochemical and geometric constraints [9]. These constraints are crucial for engineering proteins with desired binding characteristics and functional properties. Recent studies underscore that a protein's surface features, such as geometry and biochemical properties, have a more direct influence on its biological function than its sequence or backbone structure alone [20, 44, 53]. This insight is particularly relevant to PPIs, where interacting protein complexes exhibit geometric complementarity in the 3D space.

*Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

The interacting surfaces conform to their ligands' shapes and chemical properties, highlighting the importance of surface characteristics in protein design.

Protein design methods can generally be categorized into three approaches: sequence-based methods [15, 30, 49], structure-based methods [50, 55, 58], and sequence-structure co-design approaches [21, 25]. Sequence-based and structure-based methods focus on isolated aspects, which simplifies modeling but limits their ability to explore interactions at interface regions. Co-design approaches aim to holistically model both sequence and structure to capture their interdependence, yet they still struggle to accurately represent interface interactions. Providing the crucial role of protein surface analysis in predicting interaction sites and inferring PPIs [35, 46, 47], more efforts have considered comprising surface geometry and biochemical properties for protein discovery in parallel. For instance, Gainza et al. [14] built a surface-centric *de novo* design framework to capture the physical and chemical determinants of molecular recognition for new protein binders. Subsequent works [31] extract surface fingerprints from protein-ligand complexes for innovative drug-controlled cell-based therapies. Another line of works [44, 48] incorporates surface point clouds augmented with biochemical properties for protein engineering. Despite these advancements, existing methodologies face several limitations: **(i)** Limited ability to generate diverse yet receptor-compatible surface configurations. **(ii)** Lack of explicit modeling to establish robust correspondences between molecular shapes and backbone structures. **(iii)** Absence of a comprehensive strategy for top-down protein design, where coherent protein structures are generated based on receptor surface features.

To address these challenges, we introduce PepBridge, a novel framework for top-down protein design based on a multi-modal diffusion approach [17, 41–43]. As shown in Figure 1, given a receptor represented as a surface point cloud and structure annotated with geometric and biochemical properties, PepBridge generates a complete protein structure, including both the upper surface and the underlying residue structure. Notably, PepBridge leverages denoising diffusion bridge models (DDBMs) [62, 63], which interpolate between paired distributions, enabling the direct mapping of receptor surfaces to ligand surfaces while preserving physical and biochemical relevance. For structure generation, PepBridge employs an SE(3) diffusion model for backbone prediction, a torus diffusion model for torsion angle generation, and a logit-normal diffusion model for residue identity prediction. To ensure alignment and consistency, we introduce a Shape-Frame Matching Network that learns correspondences between generated ligand surfaces and backbone structures.

Our main contributions are as follows:

- **Unified Protein Design Framework**: We present PepBridge, a novel framework that jointly designs protein surfaces and structures by integrating receptor surface geometry and biochemical properties—tackling core challenges in top-down protein design.
- **Methodological Advances**: PepBridge incorporates DDBMs to generate receptor-compatible ligand surfaces and a multi-modal diffusion model for peptide structure prediction. Additionally, a Shape-Frame Matching Network is introduced to align generated surfaces and backbone structures, improving geometric and biochemical consistency.
- **Effective Validation**: We demonstrate the efficacy of PepBridge through extensive validation on peptide design tasks, showcasing its ability to generate diverse, structurally viable proteins with receptor-specific binding characteristics.

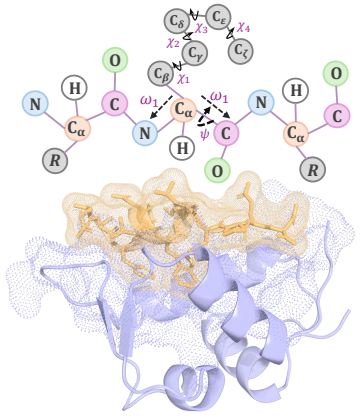

Figure 1: Top-down view of the receptor-peptide complex.

2 Preliminaries and Background

Diffusion Models. Let $q_0(\boldsymbol{x}_0)$ be a d-dimensional data distribution. The forward diffusion process [17, 40, 42] is defined by the following stochastic differential equation (SDE) with an initial condition $\boldsymbol{x}_0 \sim q_0$:

$$\mathrm{d}\boldsymbol{x}_t = f(t)\boldsymbol{x}_t\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{\omega}_t, \tag{1}$$

where $t \in [0, T]$. $f(t)$ and $g(t)$ are scalar-valued drift and diffusion coefficients, respectively. $\boldsymbol{\omega}_t \in \mathbb{R}^d$ is a standard Wiener process. q_0 usually conforms to a random Gaussian noise. The

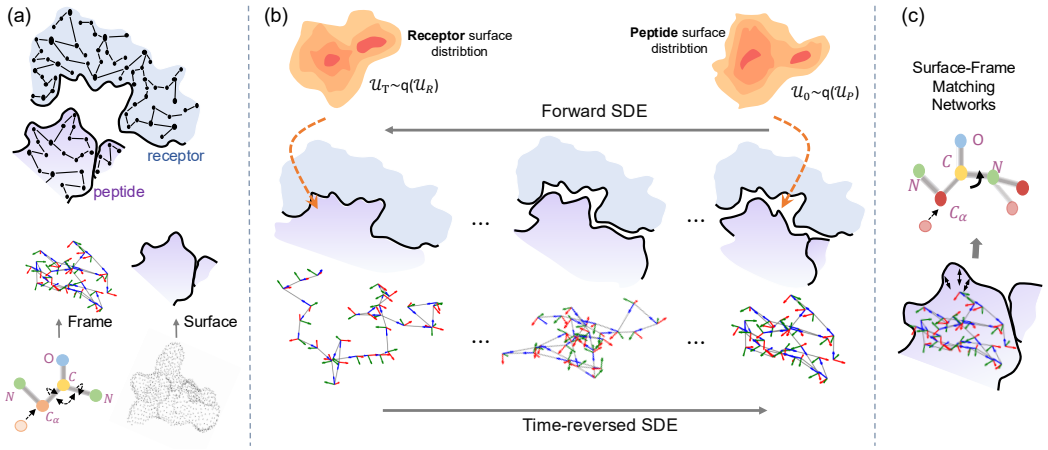

Figure 2: Illustration of the PepBridge architecture for joint surface-structure peptide generation. **(a)** The model processes receptor-ligand pairs through a top-down structure comprising molecular surface and frame components. **(b)** Two specialized diffusion models are employed simultaneously. A diffusion bridge model leverages the receptor surface as the starting point to generate peptide surfaces. An SE(3) diffusion model shoulders the responsibility of frame construction, which incorporates translation and torsion angles. **(c)** A surface-frame matching network facilitates the interaction between creased structures, while multi-modal diffusion reconstructs the complete peptide structure.

corresponding reverse-time SDE for sampling from $q_0(\boldsymbol{x}_0)$ is:

$$\mathrm{d}\boldsymbol{x}_t = [f(t)\boldsymbol{x}_t - g^2(t)\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\hat{\boldsymbol{\omega}}_t, \tag{2}$$

where $\hat{\boldsymbol{\omega}}_t$ denotes the reverse-time Wiener process and $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ is the score function of the marginal density q_t.

Diffusion Bridge Models. Traditional diffusion models assume a Gaussian prior as the starting point for the generative process. However, in many practical scenarios, including protein design, the initial state may not follow a random Gaussian distribution, requiring a more flexible approach. To address this limitation, diffusion bridge models [7, 62, 63] provide a framework for modeling structured data distributions by matching the conditional score of a tractable bridge distribution. These models enable a transport between distributions through either a reverse SDE or a probability flow ordinary differential equation (ODE). For diffusion bridges with an initial condition $\boldsymbol{x}_0 \sim q_0 = p_{\text{data}}$ and a terminal condition $\boldsymbol{x}_T = \boldsymbol{y}$, the forward process is:

$$\mathrm{d}\boldsymbol{x}_t = f(t)\boldsymbol{x}_t\mathrm{d}t + g^2(t)\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_T = \boldsymbol{y}|\boldsymbol{x}_t) + g(t)\mathrm{d}\boldsymbol{\omega}_t, \tag{3}$$

where $\boldsymbol{y}$ is drawn from a prior distribution rather than Gaussian noise. The corresponding reverse SDE is:

$$\mathrm{d}\boldsymbol{x}_t = [f(t)\boldsymbol{x}_t - g^2(t)(\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_T = \boldsymbol{y}) - \nabla_{\boldsymbol{x}_t} \log q_{T|t}(\boldsymbol{x}_T = \boldsymbol{y}|\boldsymbol{x}_t))]\mathrm{d}_t + g(t)\mathrm{d}\hat{\boldsymbol{\omega}}_t, \tag{4}$$

where $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_T = \boldsymbol{y})$ represents the bridge score function and $\hat{\omega}$ is the reverse-time Wiener process.

3 Method

3.1 Problem Statement

The protein-peptide complex pair can be represented as $\mathcal{C} = \mathcal{P} \cup \mathcal{R}$, where $\mathcal{P}$ and $\mathcal{R}$ denote the peptide and receptor, respectively. For both the peptide and the receptor, we build the top-down (Upper-Bottom) structural representation $\mathcal{U} \cup \mathcal{B}$, where $\mathcal{U} = \{u_i\}_{N_\mathcal{U}}$ denotes the spatial surface point cloud and $\mathcal{B} = \{b_i\}_{N_\mathcal{B}}$ denotes the residue-level structure. The bottom structure $\mathcal{B}$ is composed of amino acid residues, where each residue b_i is characterized by its backbone frame, residue type, and side-chain dihedral angles [12], as illustrated in Figure 1. The goal of target-aware peptide generation is to learn a probabilistic model that captures the distribution over peptide top-down structures, $p(\mathcal{P}|\mathcal{R})$, conditioned on the receptor as a structural and biochemical reference.

3.2 Surface Diffusion Bridge

Peptides typically fold into complementary shapes when binding to their receptors [29]. The geometric and biochemical properties of the receptor's binding site impose natural constraints on the conformational space of compatible peptide structures. For the sake of leveraging this relationship, we develop a tailored diffusion bridge model [63] that treats the receptor surface $\mathcal{U}_{\mathcal{R}}$ as a prior distribution to generate energetically favorable peptide conformations with complementary binding surfaces $\mathcal{U}_{\mathcal{P}}$.

Surface Representation. Firstly, we devise a pipeline for molecular surface processing and point cloud extraction. Starting with a protein structure in PDB format, we use PyMol [10] to generate solvent-accessible surface representations. The probe molecular surface approximates both the Solvent-Accessible Surface Area (SASA) and Solvent Excluded Surface (SES). The resulting point cloud consists of surface points $\boldsymbol{u}_i$, each annotated with 3D spatial coordinates and physicochemical features, including hydrophobicity and hydrogen bonding potential [13, 31].

Diffusion Bridge via h-transform. The surfaces of receptor and peptide exhibit close interactions, with their distributions $p_{\mathcal{P}}$ and $p_{\mathcal{R}}$ naturally forming pairs. We model their surface fit by reconstructing a stochastic trajectory between observed positions. Specifically, let $(\mathcal{U}_0, \mathcal{U}_T)$ denote a pair of surface datasets with empirical marginal distributions p_0 and p_T at times $t = 0$ and $t = T$, respectively. Given these endpoint distributions, our objective is to reconstruct the continuous-time stochastic process p_t over $t \in [0, T]$ that interpolates between p_0 and p_T. Using Doob's h-transform [11, 38], we define a surface diffusion process that transitions from the peptide surface $\mathcal{U}_0$ to the receptor surface $\mathcal{U}_T$. The forward SDE is given by:

$$\mathrm{d}\mathcal{U}_t = f(\mathcal{U}_t, t)\mathrm{d}t + g(t)^2 \boldsymbol{h}(\mathcal{U}_t, t, \mathcal{U}_{\mathcal{R}}, T) + g(t)\mathrm{d}\boldsymbol{w}_t, \tag{5}$$

where $\boldsymbol{h}(\mathcal{U}_t, t, \mathcal{U}_{\mathcal{R}}, T) = \nabla_{\mathcal{U}_t} \log p(\mathcal{U}_T|\mathcal{U}_t)|_{\mathcal{U}_t=\mathcal{U}_{\mathcal{P}}, \mathcal{U}_T=\mathcal{U}_{\mathcal{R}}}$ represents the gradient of the logarithmic transition kernel. The corresponding time-reversed SDE is constructed as

$$\mathrm{d}\mathcal{U}_t = [f(\mathcal{U}_t, t) + g(t)^2(\boldsymbol{s}(\mathcal{U}_t, t, \mathcal{U}_{\mathcal{R}}, T) - \boldsymbol{h}(\mathcal{U}_t, t, \mathcal{U}_{\mathcal{R}}, T))]\mathrm{d}t + g(t)\mathrm{d}\hat{\boldsymbol{w}}_t. \tag{6}$$

As shown in Figure 2, we use the receptor surface as an informative prior in place of Gaussian noise, enabling more efficient generation of complementary peptide surfaces tailored to the binding site. Accordingly, we define the forward transition kernel as $q(\mathcal{U}_t|\mathcal{U}_0, \mathcal{U}_T) = \mathcal{N}(\hat{\mu}_t, \hat{\sigma}_t^2 \boldsymbol{I})$, where $\hat{\mu}_t = \frac{\alpha_t}{\alpha_T}\frac{\mathrm{SNR}_T}{\mathrm{SNR}_t}\mathcal{U}_T + \alpha_t\mathcal{U}_0(1 - \frac{\mathrm{SNR}_T}{\mathrm{SNR}_t})$ and $\hat{\sigma}_t^2 = \sigma_t^2(1 - \frac{\mathrm{SNR}_T}{\mathrm{SNR}_t})$. α_t is a fixed signal scaling factor and normally takes the value of 1.0. σ_t defines the noise schedule, and $\mathrm{SNR}_t = \alpha_t^2/\sigma_t^2$ denotes the signal-to-noise ratio at time t. During the sampling process, we start from $p_T \sim p_{\mathcal{U}_{\mathcal{R}}}$ and approximate the score via $\boldsymbol{s}(\mathcal{U}_t, t, \mathcal{U}_{\mathcal{R}}, T) = \nabla_{\mathcal{U}_t} \log q(\mathcal{U}_t|\mathcal{U}_T)|_{\mathcal{U}_t=\mathcal{U}_{\mathcal{P}}, \mathcal{U}_T=\mathcal{U}_{\mathcal{R}}}$, where $q(\mathcal{U}_t|\mathcal{U}_T) = \int_{\mathcal{U}_0} q(\mathcal{U}_t|\mathcal{U}_0, \mathcal{U}_T) q_{\text{data}}(\mathcal{U}_0|\mathcal{U}_T)\mathrm{d}\mathcal{U}_0$.

Surface Generation Loss. We employ denoising score-matching [42] with neural network approximation of the true score $\nabla_{\mathcal{U}_t} \log q(\cdot)$, leading to the surface generation loss $\mathcal{L}_{\mathcal{U}}$ as:

$$\mathcal{L}_{\mathcal{U}} = \mathbb{E}_t \mathbb{E}_{\mathcal{U}_0, \mathcal{U}_R \sim p_{\text{data}}(\mathcal{U}_0, \mathcal{U}_R)} \mathbb{E}_{\mathcal{U}_t \sim q(\mathcal{U}_t|\mathcal{U}_0, \mathcal{U}_T=\mathcal{U}_R)} [w(t) \parallel \boldsymbol{s}_\theta(\mathcal{U}_t, \mathcal{U}_T, t) - \nabla_{\mathcal{U}_t} \log q(\mathcal{U}_t|\mathcal{U}_0, \mathcal{U}_T) \parallel^2],$$

where $q(\mathcal{U}_t|\mathcal{U}_0, \mathcal{U}_T)$ is the previously defined forward transition kernel and $w(t)$ is the time-dependent weighting function. $\boldsymbol{s}_\theta(\cdot)$ represents the parameterized geometric network, with detailed information provided in Appendix D.

3.3 Bottom Structure Diffusion Generation

Bottom Structure Parameterization. Following AlphaFold2 and recent works [26, 57, 58], we parameterize the peptide backbone using four key atoms $\mathrm{N}^\star, \mathrm{C}_\alpha^\star, \mathrm{C}^\star, \mathrm{O}^\star$, which form a rigid body frame. The rigid frame centered at C_α atom, *i.e.*, $\mathrm{C}_\alpha^\star = (0, 0, 0)$. Applying an SE(3) transformation $\mathcal{T}_n$ to the local backbone frame of residue n yields the global atomic coordinates:

$$[\mathrm{N}_n, \mathrm{C}_n, (\mathrm{C}_\alpha)_n] = \mathcal{T}_n \cdot [\mathrm{N}^\star, \mathrm{C}^\star, \mathrm{C}_\alpha^\star], \quad \mathrm{O}_n = \mathcal{T}_n \cdot \mathcal{T}_{\text{psi}}^\star(\psi_n) \cdot \mathrm{O}^\star,$$

where $\mathcal{T}_n = (r_n, m_n)$ consists of a rotation matrix $r_n \in \mathrm{SO}(3)$ and a translation vector $m_n \in \mathbb{R}^3$. The transformation$\mathcal{T}_{\text{psi}}^\star(\psi_n) = (r(\psi_n), m_{psi})$ encodes a rotation of $\mathrm{O}^\star$ around the $\mathrm{C}_\alpha - \mathrm{C}$ bond by torsion angle ψ_n. The amino acid type of the i-th residue $a_i \in \{1..20\}$ is determined by the side-chain R group.The side-chain conformation is governed by torsion angles between side-chain

atoms, represented as $\chi \in [0, 2\pi)^4$. A detailed description is provided in Appendices A and B. Our approach to bottom structure prediction integrates three components: a multi-modal diffusion process on SE(3) for backbone prediction, a torus diffusion model for torsion angle generation, and a logit-normal diffusion model for residue identity prediction.

Frame Structure Generation. Given a sequence of N rigid transformations $\mathcal{T} = [\mathcal{T}_1, ..., \mathcal{T}_N] \in \mathrm{SE}(3)^N$, we model their distribution using Riemannian diffusion on $\mathrm{SE}(3)^N$ [8]. The forward diffusion process on the SE(3)-invariant measure is:

$$\mathrm{d}\mathcal{T}_t = [0, -P\frac{1}{2}\boldsymbol{m}_t]\mathrm{d}t + [\mathrm{d}\boldsymbol{B}_t^{\mathrm{SO}(3)}, \mathrm{d}\boldsymbol{B}_t^{\mathbb{R}^3}], \tag{7}$$

where $\boldsymbol{B}_t^{\mathrm{SE}(3)} = [\boldsymbol{B}_t^{\mathrm{SO}(3)}, \boldsymbol{B}_t^{\mathbb{R}^3}]$ represent Brownian motion on $\mathrm{SO}(3)$ and $\mathbb{R}^3$, and $P \in \mathbb{R}^{3N\times 3N}$ is a projection matrix removing the center of mass $\frac{1}{N}\sum_{n=1}^{N} m_n$. In Appendix C, we show the choice of metric on $\mathrm{SE}(3)$, which allows decomposing the process into independent translational and rotational components. The backward process is given by :

$$\nabla \boldsymbol{r}_t = \nabla_{\boldsymbol{r}} \log p_{T_F - t}(\mathcal{T}_t)\mathrm{d}t + \mathrm{d}\boldsymbol{B}_t^{\mathrm{SO}(3)}, \nabla \boldsymbol{m}_t = P\{\frac{1}{2}\boldsymbol{m}_t + \nabla_{\boldsymbol{m}} \log p_{T_F - t}(\mathcal{T}_t)\}\mathrm{d}t + P\mathrm{d}\boldsymbol{B}_t^{\mathrm{SO}(3)},$$

where T_F denotes the final time step. We show more details about the training and sampling in Appendix C.

Structure Generation Loss. Backbone generation is supervised by a denoising score matching loss $\mathcal{L}_{\mathcal{T}}$, combining rotation and translation components. The loss function for the rotation component is expressed mathematically as:

$$\mathcal{L}_{\boldsymbol{r}}(\theta) = \mathbb{E}_t \mathbb{E}_{\boldsymbol{r}_0, \boldsymbol{r}_T \sim p_{\mathrm{data}}(\boldsymbol{r}_0, \boldsymbol{r}_T)} \mathbb{E}_{\boldsymbol{r}_t \sim p_{\mathrm{data}}(r_t|\boldsymbol{r}_0, \boldsymbol{r}_T)} [\lambda_t^r \parallel \nabla \log p_{t|0}(\boldsymbol{r}_t|\boldsymbol{r}_0) - s_\theta(t, \boldsymbol{r}_t) \parallel^2], \tag{8}$$

where the rotation weighting schedule is formulated as $\lambda_t^r = 1/E[\parallel \nabla \log p_{t|0}(\boldsymbol{r}_t, \boldsymbol{r}_0) \parallel^2_{\mathrm{SO}(3)}]$. Meanwhile, the translation loss component is written as:

$$\mathcal{L}_{\boldsymbol{m}}(\theta) = \mathbb{E}_t \mathbb{E}_{\boldsymbol{m}_0, \boldsymbol{m}_T} \mathbb{E}_{\boldsymbol{m}_t} [\parallel \boldsymbol{m}_0 - s_\theta(t, \boldsymbol{m}_t) \parallel^2]. \tag{9}$$

We observe that directly regressing C_α coordinates improves stability over using score matching for $\boldsymbol{m}_t$. This operation ensures that the generated backbone structures remain physically consistent with the receptor-ligand complex. The translation weight schedule is defined as $\lambda_t^m = (1 - e^{-t}/e^{-t/2})$.

Residue Type Prediction. Residue types a^j are modeled as categorical variables embedded in logit space via a sharp one-hot encoding: $\boldsymbol{v}^j \in \mathbb{R}^{20}$, where $\mathbf{v}^j[i] = K$ if $i = a^j$, otherwise $-K$, with $K > 0$ a fixed constant. Applying a softmax transformation to $\boldsymbol{v}^j$ yields a distribution over the 20-simplex, sharply peaked at the index corresponding to a^j. This effectively embeds the discrete residue type as a concentrated categorical distribution on the simplex. We define a forward diffusion process in logit space: $q(\mathbf{v}_t^j|\mathbf{v}_{t-1}^j) = \mathcal{N}(\mathbf{v}_t^j; \sqrt{\alpha_t}\mathbf{v}_{t-1}^j, (1-\alpha_t)K^2 I)$, with the prior $p(\boldsymbol{v}_T^j) = \mathcal{N}(0, K^2 I)$, corresponding to a logit-normal distribution [3]. The reverse process recover the categorical residue types by sampling from the softmax output:

$$\mathbf{v}_{t-1}^j = \sqrt{\bar{\alpha}_t}\boldsymbol{v}_0^j + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, K^2 I), \tag{10}$$

with final prediction $a^j \sim \mathrm{softmax}(\mathbf{v}_0^j)$. The model is trained to predict $\boldsymbol{\epsilon}$ using the following loss function:

$$\mathcal{L}_{\text{type}}^j = \mathbb{E}_{t, \mathbf{v}_0^j, \boldsymbol{\epsilon}} \parallel \boldsymbol{\epsilon}_\theta^{\text{type}}(\mathbf{v}_t^j, t) - \boldsymbol{\epsilon} \parallel_2^2, \tag{11}$$

Torsion Angle Prediction. The torsion vector $\boldsymbol{\chi}_i \in [0, 2\pi)^5$ consists of four side-chain angles and one backbone torsion angle ψ. To model angular diffusion, we apply the DDPM framework on the torus, using a wrap function to maintain values within $[-\pi, \pi)$: $\mathrm{wrap}(\boldsymbol{\chi}) = (\boldsymbol{\chi} + \pi)\%(2\pi) - \pi$. The forward process perturbs the angles with Gaussian noise: $\boldsymbol{\chi}_t = \mathrm{wrap}(\sqrt{\bar{\alpha}_t}\boldsymbol{\chi}_{t-1} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}), \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ and $\beta_t \in [0, 1]$ is the noise schedule.The reverse process approximates the posterior distribution: $p(\boldsymbol{\chi}^{t-1}|\boldsymbol{\chi}^t) = \mathrm{wrapnormal}\left[\mu_\theta(\boldsymbol{\chi}^t, t), \sigma_t^2 I\right]$, where $\mu_\theta(\boldsymbol{\chi}^t, t) = \mathrm{wrap}\left[\frac{1}{\sqrt{\alpha_t}}(\boldsymbol{\chi}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta^{\text{ang}}(\chi_t, t))\right]$ is the model-predicted mean. The network predicts the noise vector $\boldsymbol{\epsilon}_\theta$ and the training loss is defined as:

$$\mathcal{L}_{\text{ang}} = \mathbb{E}_{t, \boldsymbol{\chi}_0, \boldsymbol{\epsilon}_t} \parallel \mathrm{wrap}(\boldsymbol{\epsilon}_\theta^{\text{ang}}(\boldsymbol{\chi}_t, t) - \boldsymbol{\epsilon}) \parallel^2 . \tag{12}$$

3.4 Shape-Frame Matching Network

Our co-design network implements iterative updates to the top-down structure through a Shape-Frame Matching Network. We denote its l-th layer as $\mathrm{SFMNet}(\{\boldsymbol{h}_{\mathcal{U}}^{(l)}, \boldsymbol{x}_{\mathcal{U}}^{(l)}\}, \{\boldsymbol{h}_{\mathcal{B}}^{(l)}, \boldsymbol{x}_{\mathcal{B}}^{(l)}\})$. Here, $\boldsymbol{x}_{\mathcal{U}}^{(l)} \in \mathbb{R}^{N_{\mathcal{U}} \times 3}$, $\boldsymbol{x}_{\mathcal{B}}^{(l)} \in \mathbb{R}^{N_{\mathcal{B}} \times 3}$ are transformed coordinates of surface and frame, while $\boldsymbol{h}_{\mathcal{U}}^{(l)} \in \mathbb{R}^{N_{\mathcal{U}} \times d_{\mathcal{U}}}$ and $\boldsymbol{h}_{\mathcal{B}}^{(l)} \in \mathbb{R}^{N_{\mathcal{B}} \times d_{\mathcal{B}}}$ are feature embeddings. This architecture jointly transforms both features and 3D coordinates to perform interaction between surface points and backbone frames. By stacking L layers of SFMNet, we ensure equivariant updates to the protein's top-down structure. The single l-th layer is formulated as a variant of the 3D equivariant graph neural network (EGNN) [39, 54]. First of all, we get the attention score $\mathbf{att}_{ij}^{(l)} = \frac{1}{\sqrt{d_{\mathcal{U}}}}(\boldsymbol{h}_{b_i}^{(l)} \boldsymbol{W}_Q)(\boldsymbol{h}_{u_j}^{(l)} \boldsymbol{W}_K + g_{ij})$, where $\boldsymbol{W}_Q \in \mathbb{R}^{d_{\mathcal{B}} \times d_{\mathcal{U}}}$ and $\boldsymbol{W}_K \in \mathbb{R}^{d_{\mathcal{U}} \times d_{\mathcal{U}}}$ are learnable matrices. A geometric structural embedding $g_{ij} \in \mathcal{R}^{d_{\mathcal{U}}}$ is incorporated into the attention computation. It is obtained by feeding the geodesic distance into a multi-layer perceptron (MLP) as $g_{ij} = \mathrm{MLP}(\parallel \boldsymbol{x}_{b_i} - \boldsymbol{x}_{u_j} \parallel)$. Then we aggregate the message from both backbone frames and surface points, and update the backbone node features $\boldsymbol{h}_{\mathcal{B}}^{(l)}$ as

$$\boldsymbol{\nu}_{b_i,u_j} = \phi_\nu(\boldsymbol{h}_{b_i}^{(l)}, \boldsymbol{h}_{u_j}^{(l)}, \mathbf{att}_{ij}^{(l)}, t, \parallel \boldsymbol{x}_{b_i} - \boldsymbol{x}_{u_j} \parallel), \quad \boldsymbol{h}_{b_i}^{(l+1)} = \phi_h(\boldsymbol{h}_{b_i}^{(l)}, \sum_{j \in \mathcal{N}(b_i)} \boldsymbol{\nu}_{b_i,u_j}),$$

where $\phi_\nu(\cdot)$ and $\phi_h(\cdot)$ are two additional MLPs to accumulate the adjacent messages and features. $\mathcal{N}(b_i)$ is the neighborhood set of frame node b_i that contains all surface points $\{u_j | \parallel \boldsymbol{x}_{b_i} - \boldsymbol{x}_{u_j} \parallel \leq \gamma\}$ that interact with this residue, where γ is the distance threshold. After that, we calculate the shift of translation $\Delta \boldsymbol{m}^l$ and rotation $\Delta \boldsymbol{r}^l$, adding them to the original values:

$$\boldsymbol{m}_{b_i}^{(l+1)} = \boldsymbol{m}_{b_i}^{(l)} + \phi_m(\boldsymbol{h}_{b_i}^{(l+1)}), \quad \boldsymbol{r}_{b_i}^{(l+1)} = \boldsymbol{r}_{b_i}^{(l)} + \phi_r(\boldsymbol{h}_{b_i}^{(l+1)}). \tag{13}$$

3.5 Overall Training Loss

The complete loss function combines the surface and bottom structure loss functions:

$$L_{\text{total}} = \boldsymbol{\mu}^T [L_{\mathcal{U}}, L_r, L_m, L_{\text{type}}, L_{\text{ang}}], \tag{14}$$

where $\boldsymbol{\mu}$ denotes the set of weighting hyperparameters balancing each loss term. This formulation enables joint optimization over both surface geometry and residue structure for comprehensive protein structure prediction.

4 Experiments

This section presents comprehensive experimental evaluations to demonstrate the efficacy of our proposed method. Our investigation addresses three fundamental questions: **Q1**: How do the generated samples perform in terms of quality? **Q2**: Does the method generate physically valid samples? **Q3**: What is the impact of key architectural decisions on model performance? We evaluate PepBridge and baseline methods on two tasks: (1) *Surface-Structure Joint-design*. Generation of peptide structure and surface conditional on a specified receptor binding site. This task involves the simultaneous generation of peptide surface characteristics and structural conformations, conditional on a specified receptor binding site. (2) *Side-chain Packing.* Prediction of optimal side-chain angles for peptides when they are bound to a specified receptor site.

4.1 Experimental Setup

Dataset. The evaluation utilized the PepMerge dataset [25], a collection derived from the integration of PepBDB [52] and Q-BioLip [51] databases. Following the protocol established in [25], we implemented rigorous filtering criteria: eliminating redundant entries, excluding structural data with resolution exceeding 4 Å, and selecting peptides with sequence lengths between 3 and 25 residues. We evaluate the generation model's performance using multiple criteria to ensure candidates exhibit high diversity and novelty while maintaining validity and desired distributional properties. For each receptor, we generate 40 candidates. Additional dataset details are provided in Appendix E.1.

Baselines. We compare our approach against two distinct categories of state-of-the-art protein design methods. The first category consists of backbone-centric approaches that do not explicitly model

Table 1: Evaluation on surface-structure joint-design. On each target, 40 candidates are generated for evaluation. Div., Aff. and Stab. are abbreviations for diversity, affinity and stability, respectively.

	Div_{stru} (↑)	Aff. % (↑)	Stab. % (↑)	RMSD Å(↓)	BSR (↑)
ProteinGenerator	0.54	13.47	23.48	4.35	24.62
RFDiffusion	0.38	16.53	26.82	4.17	26.71
Chroma (RIA)	0.59	17.96	16.69	3.97	74.12
PPFLOW	0.53	17.62	17.25	2.94	78.72
PepGLAD	0.32	10.47	20.39	3.83	19.34
PepFlow w/Bb	0.64	18.10	14.04	2.30	82.17
PepFlow w/Bb+Seq	0.50	19.39	19.20	2.21	85.19
PepFlow w/Bb+Seq+Ang	0.42	21.37	18.15	2.07	86.89
PepBridge w/Bb+surf	0.60	20.07	21.75	2.04	84.62
PepBridge w/Bb+Seq+surf	0.62	22.07	20.71	2.18	84.91
PepBridge w/Bb+Seq+Ang+surf	0.59	19.16	25.02	2.19	83.90

side-chain conformations, such as RFDiffusion [50], ProteinGenerator [27] and PPFLOW [26]. RFDiffusion generates backbones via diffusion, followed by sequence prediction with ProteinMPNN [6], while ProteinGenerator jointly models sequence and backbone. PPFLOW [26] conditions on the target receptor and uses conditional flow matching on torus manifolds to generate peptide backbones by modeling torsional geometry. The second category comprises full-atom models like PepGLAD [21], Chroma [19], and PepFlow [25], which explicitly generate side-chain conformations. To evaluate Chroma, we used its conditional generation with receptor interaction area (RIA) as the conditioning input. Further details are provided in Appendix E.2.

4.2 Surface-Structure Joint-Design

Metrics. We evaluate the validity of the generated backbone structures using the following metrics. (1) *Diversity*: The average of one minus the pairwise TM-score [59] between generated peptides. A higher diversity score indicates higher dissimilarity and greater structural variety among the generated peptides. (2) *Affinity*: Binding affinity is evaluated using Rosetta's energy function [2], measured in kcal/mol. We report that the proportion of generated peptides with binding energies is lower than that of the native peptide. (3) *Stability*: The percentage of generated peptide-protein complexes with total energy lower than the native complex. (4) RMSD_{C_α}: The root mean square deviation of C_α atom coordinates between the generated and reference peptide structures, measured in Ångströms (Å). (5) *BSR*: The binding site ratio, which measures the overlap between the binding sites of the generated and native peptides on the target protein.

Results. As shown in Table 1, PepBridge consistently benefits from explicit surface conditioning. Among our variants, PepBridge w/Bb+Seq+surf attains the highest affinity, and PepBridge w/Bb+surf yields the lowest RMSD across all methods, indicating strong geometric fidelity to native-like docked poses. PepBridge w/Bb+Seq+Ang+surf ranks second in stability at 25.02% and maintains competitive structural diversity. Surface-aware generation consistently outperforms backbone-only counterparts. Relative to PepFlow w/Bb, PepBridge w/Bb+surf increases affinity and markedly reduces RMSD. PepBridge w/Bb+Seq+surf continues this trend, further reducing RMSD while raising affinity. PepBridge variants deliver strong RMSD, affinity, and stability, translating into competitive enrichment (BSR). Modeling interface geometry through joint surface–backbone conditioning proves especially effective for enhancing conformational accuracy and binding complementarity.

4.3 Ablation Study

To comprehensively assess the contribution of each component in our proposed model, we conduct an ablation study by systematically removing or modifying key elements. (1) *-bridge/+vanilla diffusion*: The denoising diffusion bridge model is replaced with a standard diffusion model, which diminishes the incorporation of receptor surface geometry as prior information. (2) *-bridge/+CFG diffusion*: The denoising diffusion bridge model is replaced with a classifier-free guidance diffusion model [16], which enables receptor-guided conditional generation. However, this approach still maintains a Gaussian prior distribution, rather than incorporating explicit geometric priors. (3)

Table 2: Ablations on different components of PepBridge, where the best model is in **bold**.

Ablations	Div_{stru} (↑)	Div_{surf} (↑)	Aff. % (↑)	Stab. % (↑)	RMSD Å(↓)	BSR (↑)	Con. (↑)
PepBridge	**0.59**	**0.46**	**19.16**	**25.02**	**2.19**	**83.90**	**0.43**
-bridge/+vanilla diffusion	0.42	0.39	15.97	17.72	3.18	46.21	0.31
-bridge/+CFG diffusion	0.39	0.41	16.38	19.32	3.46	57.39	0.36
-surface context	0.51	–	16.17	15.37	4.21	31.37	–
-surface&frame matching	0.42	0.35	14.82	22.41	3.71	54.71	0.25

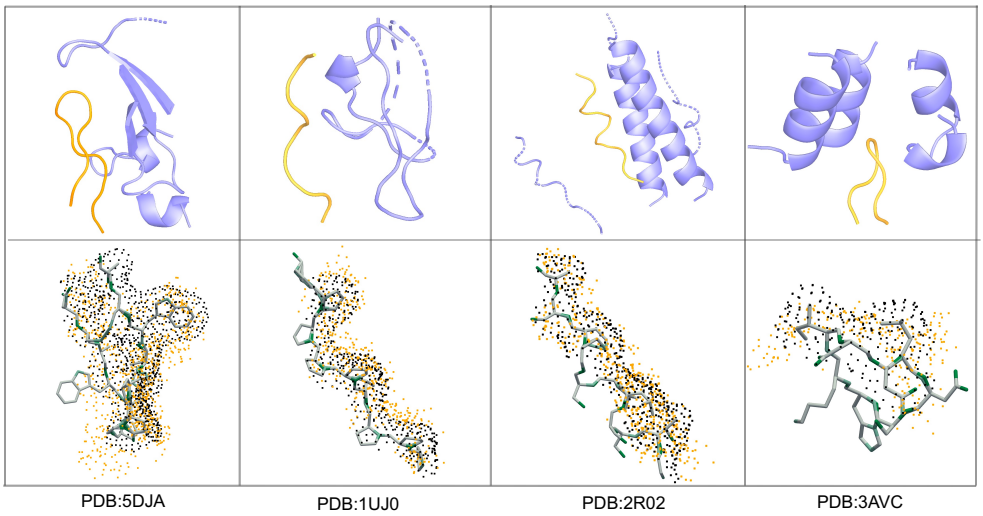

Figure 3: Visualization of generated peptides by our PepBridge. **Top:** Generated peptides (in orange) for receptors (in purple). The PDB ID of the receptors are 5DJA, 1UJ0, 2R02, and 3AVC. **Bottom:** The generated backbone structure and surface. The ground-truth surface structure (in black) and generated surface (in orange) are shown to compare the ability of the interface caption.

-*surface context*: Surface context information is removed, leaving only the backbone generation component. (4) -*surface&frame matching*: The surface-frame matching mechanism is excluded during the training of the denoising model. We also evaluate additional metrics to analyze model performance: (1) *Consistency*: Surface-structure consistency is quantified using Cramér [5], which measures the association between surface and structure clustering labels. Higher consistency values indicate that candidates with similar structures tend to have similar surfaces, suggesting that the model effectively captures the interdependence between backbone structures and surface features. (2) *Surface Diversity*: Div_{surf} is computed based on surface alignment similarities. Detailed information about the metrics can be found in Appendix E.3.

Table 2 records the ablation results. It shows that during surface generation, the basic diffusion model struggles to capture the correct distribution compared to the diffusion bridge model, resulting in instability and low consistency in the generated structures. Replacing the denoising diffusion bridge model with the classifier-free guidance diffusion model still fails to capture the surface distribution accurately. When eliminating the surface context and only generating the backbone, the performance on affinity, RMSD, and BSR drops dramatically, since it can not get enough binding set information. Without surface guidance, the model will generate unrealistic and unstable peptides. As for the component of the surface-frame matching mechanism, through the result, we can know that the network helps to achieve high stability and enhance consistency between the surface and backbone. It also helps to reduce RMSD_{C_α} by providing effective structural patterns.

4.4 Visualization

We further present four examples of generated peptides in Figure 4.3. We observe that PepBridge produces peptides with proper geometries and positions. The generated surface exhibits similar

Table 3: Evaluation of different methods in the side-chain packing task, where the best and the suboptimal approaches are **bolded** and underlined, respectively.

	Angle MAE °(↓)				Angle Accuracy %(↑)		
	χ_1	χ_2	χ_3	χ_4	All residues	Core residues	Surface residues
SCWRL4	29.79	30.12	52.38	62.03	45.93	66.25	34.59
DLPacker	28.35	32.62	54.69	59.60	49.00	68.03	39.56
AttnPacker	29.61	28.83	47.66	53.64	47.53	71.65	38.90
DiffPack	26.29	29.57	47.64	56.85	55.86	**79.62**	41.31
PepFlow w/Bb+Seq+Ang	27.61	**25.60**	48.20	54.02	54.29	70.47	44.06
PepBridge (ours)	**25.96**	26.76	**46.81**	**52.95**	**56.71**	73.79	**46.17**

conformation with the ground-truth surface, which show the ability to interact with the target binding sites and capture the right shape. We provide additional experiments in Appendix F.

4.5 Side-chain Packing

The backbone prior state is initialized using a native peptide structure and subsequently generate the surface and side-chain angles. We compare our approach against established methods including SCWRL4 [23], DLPacker [34], AttnPacker [32], DiffPack [60], and PepFlow [25]. The evaluation metrics include: (1) *Angle MAE*, which quantifies the mean absolute error between predicted and ground-truth angles, and (2) *Angle Accuracy*, which considers torsion angles correct when their deviation falls within 20°. Following the methodology of [60], we present results for core residues, surface residues, and all residues. For each peptide, we generate an ensemble of 64 conformations.

Table 3 presents the comparative results. Our model demonstrates superior performance in predicting χ_1, χ_3, and χ_4 angles. Particularly, PepBridge reduces the prediction error by 2% to 52.95 for the most challenging angle χ_4. This suggests that the integration of surface-level information enhances side-chain angle prediction accuracy. Furthermore, the model exhibits particularly strong performance in surface residue prediction with a high accuracy of 46.17%, indicating that PepBridge effectively captures spatial relationships in interface regions. It also attains the best overall accuracy of 56.71%, which significantly improves the prior state-of-the-art PepFlow by 4.45%.

5 Related Works

Computational Protein Design Sequence-based and structure-based approaches are two main trajectories in computational protein design. The former models amino acid chains by language models [15, 30, 49], whereas the latter models the 3D geometry. Notable structure-based methods include FoldingDiff [55], which represents protein backbones through sequential angles, and RFDiffusion [50], which employs varied diffusion schemes for backbone generation. FrameDiff [58] advanced this field by developing SE(3)-invariant diffusion models for protein modeling. Flow models have also shown promise in backbone design, as demonstrated by FOLDFLOW [4] and PPFLOW [26]. Recently, sequence-structure co-design has gained attention, with models such as PepGLAD [21] and PepFlow [25]. PepHAR [24] further targets peptide binders via hotspot sampling and multifragment autoregressive extension, enforcing geometric validity. Surface-conditioned protein modeling represents the latest frontier in this field. Surface-VQMAE [53] introduced a Transformer-based architecture that integrates surface geometry and captures patch-level relations. Despite those achievements, the joint design of the surface and structure remains an unexplored area, and we position our study as a pioneering effort in this direction.

Surface Context in design ligands Classical methods explicitly model geometric and physicochemical complementarity (e.g., shape matching, lock-and-key). Mesh-based learning like MaSIF [13] encodes surfaces with handcrafted geometric/chemical descriptors, while dMaSIF [47] accelerates this via point-cloud surfaces with atom-level features. Recent generative methods like ShEPhERD [1] and DSR [46] incorporate surface features into diffusion frameworks, and SurfPro [44] generates sequences conditioned on known surfaces. However, these approaches typically assume strict complementarity, require hand-crafted features, or depend on ground-truth surfaces. PepBridge instead uses denoising diffusion bridge models to learn data-driven mappings between receptor and ligand

interfaces, capturing flexible, non-complementary interactions crucial for peptide binding where induced fit dominates.

Diffusion Models Diffusion models have emerged as powerful probabilistic generative models [17, 41–43]. Recent advances have extended these models to handle data with inherent invariances [37, 56], as well as to discrete domains [28, 33]. In parallel, manifold-aware diffusion models have been introduced, including the Riemannian Score-Based Generative Model (RSGM)[8] and the Riemannian Diffusion Model (RDM)[18]. A significant variant, diffusion bridge models [7, 62, 63], enables interpolation between paired endpoint distributions. In this work, we employ a diffusion bridge model to construct a stochastic bridge between receptor and ligand distributions, enabling the generation of complementary, high-quality samples.

6 Conclusion

This work presents PepBridge, a novel framework conditioned on receptor structures for joint protein surface-backbone design. We employ a stochastic bridge process between receptor and ligand surfaces with tractable marginal distributions, where the model learns by matching conditional scores of the bridge distribution. For backbone generation, an SE(3) diffusion model is used to predict frame geometry. The Surface-Frame Matching Network enables bidirectional information flow between surface and backbone levels, facilitating coherent structural development. The superior performance of PepBridge highlighs the advantages of shape-driven generation in target protein design.

References

[1] Adams, K., Abeywardane, K., Fromer, J. C., and Coley, C. W. Shepherd: Diffusing shape, electrostatics, and pharmacophores for bioisosteric drug design. In *ICLR*, 2025.

[2] Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

[3] Atchison, J. and Shen, S. M. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

[4] Bose, A. J., Akhound-Sadegh, T., Huguet, G., Fatras, K., 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 preprint arXiv:2310.02391*, 2023.

[5] Cramér, H. *Mathematical methods of statistics*, volume 26. Princeton university press, 1999.

[6] Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[7] De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[8] De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

[9] Defresne, M., Barbe, S., and Schiex, T. Protein design with deep learning. *International Journal of Molecular Sciences*, 22(21):11741, 2021.

[10] DeLano, W. L. et al. Pymol: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr*, 40(1):82–92, 2002.

[11] Doob, J. L. and Doob, J. *Classical potential theory and its probabilistic counterpart*, volume 262. Springer, 1984.

[12] Fisher, M. Lehninger principles of biochemistry, ; by david l. nelson and michael m. cox. *The Chemical Educator*, 6:69–70, 2001.

[13] Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M. M., and Correia, B. E. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[14] Gainza, P., Wehrle, S., Van Hall-Beauvais, A., Marchand, A., Scheck, A., Harteveld, Z., Buckley, S., Ni, D., Tan, S., Sverrisson, F., et al. De novo design of protein interactions with learned surface fingerprints. *Nature*, 617(7959):176–184, 2023.

[15] Hie, B., Candido, S., Lin, Z., Kabeli, O., Rao, R., Smetanin, N., Sercu, T., and Rives, A. A high-level programming language for generative protein design. *bioRxiv*, pp. 2022–12, 2022.

[16] Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[17] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[18] Huang, C.-W., Aghajohari, M., Bose, J., Panangaden, P., and Courville, A. C. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

[19] Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

[20] Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A., Aflalo, C., and Vakser, I. A. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89(6):2195–2199, 1992.

[21] Kong, X., Jia, Y., Huang, W., and Liu, Y. Full-atom peptide design with geometric latent diffusion. *Advances in Neural Information Processing Systems*, 37:74808–74839, 2024.

[22] Krapp, L. F., Abriata, L. A., Cortés Rodriguez, F., and Dal Peraro, M. Pesto: parameter-free geometric deep learning for accurate prediction of protein binding interfaces. *Nature communications*, 14(1):2175, 2023.

[23] Krivov, G. G., Shapovalov, M. V., and Dunbrack Jr, R. L. Improved prediction of protein side-chain conformations with scwrl4. *Proteins: Structure, Function, and Bioinformatics*, 77 (4):778–795, 2009.

[24] Li, J., Chen, T., Luo, S., Cheng, C., Guan, J., Guo, R., Wang, S., Liu, G., Peng, J., and Ma, J. Hotspot-driven peptide design via multi-fragment autoregressive extension. In *The Thirteenth International Conference on Learning Representations*.

[25] Li, J., Cheng, C., Wu, Z., Guo, R., Luo, S., Ren, Z., Peng, J., and Ma, J. Full-atom peptide design based on multi-modal flow matching. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 27615–27640, 2024.

[26] Lin, H., Zhang, O., Zhao, H., Jiang, D., Wu, L., Liu, Z., Huang, Y., and Li, S. Z. Ppflow: target-aware peptide design with torsional flow matching. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 30510–30528, 2024.

[27] Lisanza, S. L., Gershon, J. M., Tipps, S., Arnoldt, L., Hendel, S., Sims, J. N., Li, X., and Baker, D. Joint generation of protein sequence and structure with rosettafold sequence space diffusion. *bioRxiv*, pp. 2023–05, 2023.

[28] Liu, X. and Wu, L. Learning diffusion bridges on constrained domains. In *international conference on learning representations (ICLR)*, 2023.

[29] London, N., Movshovitz-Attias, D., and Schueler-Furman, O. The structural basis of peptide-protein binding strategies. *Structure*, 18(2):188–199, 2010.

[30] Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, 2023.

[31] Marchand, A., Buckley, S., Schneuing, A., Pacesa, M., Elia, M., Gainza, P., Elizarova, E., Neeser, R. M., Lee, P.-W., Reymond, L., et al. Targeting protein–ligand neosurfaces with a generalizable deep learning tool. *Nature*, pp. 1–10, 2025.

[32] McPartlon, M. and Xu, J. An end-to-end deep learning method for rotamer-free protein side-chain packing. *bioRxiv*, pp. 2022–03, 2022.

[33] Meng, C., Choi, K., Song, J., and Ermon, S. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35:34532–34545, 2022.

[34] Misiura, M., Shroff, R., Thyer, R., and Kolomeisky, A. B. Dlpacker: Deep learning for prediction of amino acid side chain conformations in proteins. *Proteins: Structure, Function, and Bioinformatics*, 90(6):1278–1290, 2022.

[35] Mylonas, S. K., Axenopoulos, A., and Daras, P. Deepsurf: a surface-based deep learning approach for the prediction of ligand binding sites on proteins. *Bioinformatics*, 37(12):1681–1690, 2021.

[36] Nada, H., Choi, Y., Kim, S., Jeong, K. S., Meanwell, N. A., and Lee, K. New insights into protein–protein interaction modulators in drug discovery and therapeutic advance. *Signal Transduction and Targeted Therapy*, 9(1):1–32, 2024.

[37] Niu, C., Song, Y., Song, J., Zhao, S., Grover, A., and Ermon, S. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, pp. 4474–4484. PMLR, 2020.

[38] Rogers, L. C. and Williams, D. *Diffusions, markov processes, and martingales: Volume 1, foundations*. Cambridge university press, 2000.

[39] Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

[40] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

[41] Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[42] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[43] Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

[44] Song, Z., Huang, T., Li, L., and Jin, W. Surfpro: Functional protein design based on continuous surface. *arXiv preprint arXiv:2405.06693*, 2024.

[45] 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.

[46] Sun, D., Huang, H., Li, Y., Gong, X., and Ye, Q. Dsr: dynamical surface representation as implicit neural networks for protein. *Advances in Neural Information Processing Systems*, 36: 13873–13886, 2023.

[47] Sverrisson, F., Feydy, J., Correia, B. E., and Bronstein, M. M. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15272–15281, 2021.

[48] Tang, X., Ye, X., Wu, F., Shao, D., Fang, Y., Chen, S., Xu, D., and Gerstein, M. Bc-design: A biochemistry-aware framework for highly accurate inverse protein folding. *bioRxiv*, pp. 2024–10, 2024.

[49] Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., and Rives, A. Language models generalize beyond natural proteins. *BioRxiv*, pp. 2022–12, 2022.

[50] 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., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[51] Wei, H., Wang, W., Peng, Z., and Yang, J. Q-biolip: A comprehensive resource for quaternary structure-based protein–ligand interactions. *Genomics, Proteomics & Bioinformatics*, 22(1), 2024.

[52] Wen, Z., He, J., Tao, H., and Huang, S.-Y. Pepbdb: a comprehensive structural database of biological peptide–protein interactions. *Bioinformatics*, 35(1):175–177, 2019.

[53] Wu, F. and Li, S. Z. Surface-vqmae: Vector-quantized masked auto-encoders on molecular surfaces. In *International Conference on Machine Learning*, pp. 53619–53634. PMLR, 2024.

[54] Wu, F., Jin, S., Jiang, Y., Jin, X., Tang, B., Niu, Z., Liu, X., Zhang, Q., Zeng, X., and Li, S. Z. Pre-training of equivariant graph matching networks with conformation flexibility for drug binding. *Advanced Science*, 9(33):2203796, 2022.

[55] Wu, K. E., Yang, K. K., van den Berg, R., Alamdari, S., Zou, J. Y., Lu, A. X., and Amini, A. P. Protein structure generation via folding diffusion. *Nature communications*, 15(1):1059, 2024.

[56] Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[57] Yang, Z., Zeng, X., Zhao, Y., and Chen, R. Alphafold2 and its applications in the fields of biology and medicine. *Signal Transduction and Targeted Therapy*, 8(1):115, 2023.

[58] Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

[59] Zhang, Y. and Skolnick, J. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

[60] Zhang, Y., Zhang, Z., Zhong, B., Misra, S., and Tang, J. Diffpack: A torsional diffusion model for autoregressive protein side-chain packing. *Advances in Neural Information Processing Systems*, 36, 2024.

[61] Zhang, Y. et al. Diffpack: A torsional diffusion model for autoregressive protein side-chain packing. In *Advances in Neural Information Processing Systems*, volume 36, pp. 48150–48172, 2023.

[62] Zheng, K., He, G., Chen, J., Bao, F., and Zhu, J. Diffusion bridge implicit models. *arXiv preprint arXiv:2405.15885*, 2024.

[63] Zhou, L., Lou, A., Khanna, S., and Ermon, S. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix H

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Appendix B,C

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide an anonymous link to the code and also include it in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide an anonymous link to the code and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix E

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We followed standard evaluation practices in the field by comparing methods across diverse models, rather than relying on statistical significance tests.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to ethical standards by ensuring responsible data use, transparency, and consideration of potential societal impacts, in line with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Introduction

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not involve models or datasets with a high risk for misuse, so no special safeguards were necessary.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, all creators and original owners of assets used in the paper are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The study does not involve human participants, so these considerations do not apply.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs do not impact the core methodology, so no declaration is required.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Backbone Representation

As introduced in 3.3 section, every frame is composed by four atomic group $\mathrm{N}^\star, \mathrm{C}_\alpha^\star, \mathrm{C}^\star, \mathrm{O}^\star$, which is idealized atom coordinates that assumes chemically idealized bond angles and lengths.

We use the tuple $T = (r, m)$ to denote the Euclidean transformations corresponding to frames, where $\boldsymbol{r} \in \mathrm{SO}(3)$ for the rotation and $\boldsymbol{m} \in \mathbb{R}^3$ for the translation components. We use the dot product operator $(\cdot)$ to denote application of a transformation to the position of frame $\boldsymbol{b} \in \mathbb{R}^3$:

$$\begin{aligned}\hat{\boldsymbol{b}} &= T \cdot \boldsymbol{b} \\ &= (\boldsymbol{r}, \boldsymbol{m}) \cdot \boldsymbol{b} \\ &= \boldsymbol{r}\boldsymbol{b} + \boldsymbol{m}.\end{aligned}$$

The composition of Euclidean transformations denoted as:

$$\begin{aligned}T &= T_1 \cdot T_2 \\ (\boldsymbol{r}, \boldsymbol{m}) &= (\boldsymbol{r}_1, \boldsymbol{m}_1) \cdot (\boldsymbol{r}_2, \boldsymbol{m}_2) \\ &= (\boldsymbol{r}_1\boldsymbol{r}_2,\ \boldsymbol{r}_1\boldsymbol{m}_2 + \boldsymbol{m}_1).\end{aligned}$$

The group inverse of the transform T is denoted as:

$$\begin{aligned}T^{-1} &= (\boldsymbol{r}, \boldsymbol{m})^{-1} \\ &= (\boldsymbol{r}^{-1},\ -\boldsymbol{r}^{-1}\boldsymbol{m})\end{aligned}$$

The tuple transforms a position in local coordinates $\boldsymbol{b}_{\text{local}} \in \mathbb{R}^3$ to a position in global coordinates $\boldsymbol{b}_{\text{global}} \in \mathbb{R}^3$ as

$$\begin{aligned}\boldsymbol{b}_{\text{global}} &= T \cdot \boldsymbol{b}_{\text{local}} \\ &= \boldsymbol{r}\boldsymbol{b}_{\text{local}} + \boldsymbol{m}\,.\end{aligned}$$

In local position of frame, the bond angles and lengths values differ slightly per amino acid type. Follow [58] and [57], we set the local coordinates as:

$$\begin{aligned}\mathrm{N}^\star &= (-0.525, 1.363, 0.0) \\ \mathrm{C}_\alpha^\star &= (0.0, 0.0, 0.0) \\ \mathrm{C}^\star &= (1.526, 0.0, 0.0) \\ \mathrm{O}^\star &= (0.627, 1.062, 0.0)\end{aligned} \tag{15}$$

where $\mathrm{C}_\alpha^\star$ is central in protein backbones, connecting $\mathrm{N}^\star$ and $\mathrm{C}^\star$ groups. Using the transformation T_n, we manipulate idealized coordinates to construct global coordinates of backbone atoms for residue n via:

$$\begin{aligned}[\mathrm{N}_n, \mathrm{C}_n, (\mathrm{C}_\alpha)_n, \mathrm{O}_n] = \big[\, & T_n \cdot \mathrm{N}^\star,\ T_n \cdot \mathrm{C}^\star, \\ & T_n \cdot \mathrm{C}_\alpha^\star,\ T_n \cdot T^\star_{\text{psi}}(\psi_n) \cdot \mathrm{O}^\star \,\big].\end{aligned} \tag{16}$$

Given the coordinates of three atoms $[\mathrm{N}_n, \mathrm{C}_n, (\mathrm{C}_\alpha)_n]$, we construct a local rigid frame using a Gram-Schmidt process:

$$\begin{aligned}&\boldsymbol{\omega}_1 = \mathrm{C}_n - (\mathrm{C}_\alpha)_n, \qquad \boldsymbol{\omega}_2 = \mathrm{N}_n - (\mathrm{C}_\alpha)_n \\ &\boldsymbol{e}_1 = \boldsymbol{\omega}_1 / \|\boldsymbol{\omega}_1\|, \qquad \boldsymbol{u}_2 = \boldsymbol{\omega}_2 - \boldsymbol{e}_1(\boldsymbol{e}_1^T \boldsymbol{\omega}_2), \\ &\boldsymbol{e}_2 = \boldsymbol{u}_2 / \|\boldsymbol{u}_2\|, \\ &\boldsymbol{e}_3 = \boldsymbol{e}_1 \times \boldsymbol{e}_2, \\ &\boldsymbol{r}_n = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)\,, \\ &\boldsymbol{m}_n = (\mathrm{C}_\alpha)_n, \\ &T_n = (\boldsymbol{r}_n, \boldsymbol{m}_n).\end{aligned} \tag{17}$$

In this construction, two directional vectors are first defined: from C_α to C and C_α to N. We normalize the first direction $\boldsymbol{\omega}_1$ to define the local x-axis $\boldsymbol{e}_1$, and orthogonalize and normalize $\boldsymbol{\omega}_2$ to define the y-axis $\boldsymbol{e}_2$. The z-axis $\boldsymbol{e}_3$ is computed as the cross product of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, forming a right-handed orthonormal basis. The resulting frame T_n placing the local frame at the C_α of residue n. To define the local frame for placing the oxygen atom, we begin with the residue's central frame T_n, then apply a secondary transformation: $T^\star_{\text{psi}}(\psi_n) = (\boldsymbol{r}_x(\psi_n), \boldsymbol{m}_{\text{psi}})$, where ψ_n represents a rotation angle along x-axis. The transformation funtions are defined as:

$$\boldsymbol{r}_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}, \quad \boldsymbol{m}_{\text{psi}} = (1.526, 0.0, 0.0). \tag{18}$$

This transformation represents a rotation around the x-axis (aligned with the bond from C_α to C) by an angle ψ_n, followed by a translation to the position of the carbon atom $\mathrm{C}^\star$ in the idealized frame centered at $\mathrm{C}^\star_\alpha$. The combined transformation $T_n \cdot T^\star_{\text{psi}}(\psi_n)$ thus defines the final frame in which the idealized oxygen $\mathrm{O}^\star$ is placed to obtain its global coordinate.

B Diffusion on the Toric Manifold

A torsion vector $\chi \in [0, 2\pi)^d$ naturally resides on a flat d-dimensional torus, which can be represented as the quotient space $\mathbb{R}^d/L$, where $L = (2\pi\mathbb{Z}^d)$ denotes a discrete lattice subgroup of $\mathbb{R}^d$ isomorphic to $\mathbb{Z}^d$. This space models periodic angular data, and inherits a flat metric from its covering Euclidean space. The tangent space of the torus at any point is identified with $\mathbb{R}^d$, and all operations are performed modulo 2π.

C Diffusion on SE(3)

Following previous work [58], we treat the group $\mathrm{SE}(3)$ as the product space $\mathrm{SO}(3) \times \mathbb{R}^3$, and endow it with a product Riemannian metric. Specifically, for tangent vectors $(a, b), (a', b') \in \mathrm{T}_r\,\mathrm{SO}(3) \times \mathbb{R}^3$, the metric is defined as:

$$\langle (a, b), (a', b') \rangle_{\mathrm{SE}(3)} = \langle a, a' \rangle_{\mathrm{SO}(3)} + \langle b, b' \rangle_{\mathbb{R}^3}.$$

This structure allows for a natural decomposition of differential geometric objects on $\mathrm{SE}(3)$ into rotational and translational components. In particular, the gradient of a function $f : \mathrm{SE}(3) \to \mathbb{R}$ at $\mathcal{T} = (\boldsymbol{r}, \boldsymbol{x})$ is given by:

$$\nabla_{\mathcal{T}} f(\mathcal{T}) = [\nabla_r f(\boldsymbol{r}, \boldsymbol{m}), \nabla_m f(\boldsymbol{r}, \boldsymbol{m})],$$

and the Laplace–Beltrami operator decomposes as:

$$\Delta_{\mathrm{SE}(3)} f(T) = \Delta_{\mathrm{SO}(3)} f(\boldsymbol{r}, \boldsymbol{m}) + \Delta_{\mathbb{R}^3} f(\boldsymbol{r}, \boldsymbol{m}).$$

We define Brownian motion on $\mathrm{SE}(3)$ as the product of independent Brownian motions on $\mathrm{SO}(3)$ and $\mathbb{R}^3$:

$$\boldsymbol{B}_t^{\mathrm{SE}(3)} = [\boldsymbol{B}_t^{\mathrm{SO}(3)}, \boldsymbol{B}_t^{\mathbb{R}^3}]$$

where the rotational and translational components evolve independently. This product metric allows us to treat the rotational and translational components of the forward diffusion process independently, leading to the following decomposition of the conditional score:

$$\nabla_{\mathcal{T}_t} \log p_{t|0}(\mathcal{T}_t|\mathcal{T}_0) = [\nabla_{r_t} \log p_{t|0}(\boldsymbol{r}_t|\boldsymbol{r}_0), \nabla_{m_t} \log p_{t|0}(\boldsymbol{m}_t|\boldsymbol{m}_0)]$$

The forward process on $\mathrm{SE}(3)$ is thus described by two independent SDEs. Let $\mathcal{M}$ be a compact Lie group (e.g., $\mathrm{SO}(3)$), and let χ_ℓ denote the character of the ℓ-th irreducible unitary representation of dimension d_ℓ. Then, the heat kernel (transition density of Brownian motion) on $\mathcal{M}$ is given by:

$$p_{t|0}(x_t|x_0) = \textstyle\sum_{\ell \in \mathbb{N}} d_\ell e^{-\lambda_\ell t/2} \chi_\ell((x_0)^{-1} x_t).$$

where λ_ℓ is the eigenvalue of the Laplace–Beltrami operator associated with χ_ℓ. Specializing to $\mathrm{SO}(3)$, the heat kernel becomes the isotropic Gaussian on $\mathrm{SO}(3)$:

$$f(\omega, t) = \textstyle\sum_{\ell \in \mathbb{N}} (2\ell + 1) e^{-\ell(\ell+1)t/2} \frac{\sin((\ell+1/2)\omega)}{\sin(\omega/2)}. \tag{19}$$

where ω is the angle of the relative rotation. The corresponding score function is:

$$\nabla \log p_{t|0}(\boldsymbol{r}_t \mid \boldsymbol{r}_0) = \frac{\boldsymbol{r}_t}{\omega_t} \log(\boldsymbol{r}_0^\top \boldsymbol{r}_t) \frac{\partial_\omega f(\omega^{(t)}, t)}{f(\omega^{(t)}, t)}, \tag{20}$$

where ω_t is the angle of the relative rotation $\boldsymbol{r}_0^\top \boldsymbol{r}_t$, and the matrix logarithm term maps to the tangent space at $\boldsymbol{r}_t$.

For the translational component, we model the forward process using a Variance Preserving SDE (VP-SDE). The transition density is given by:

$$p_{t|0}(\boldsymbol{m}_t|\boldsymbol{m}_0) = \mathcal{N}(x_t; e^{-t/2}\boldsymbol{m}_0, (1 - e^{-t})\,\mathrm{I}_3). \tag{21}$$

Then we can get the score as:

$$\nabla \log p_{t|0}(\boldsymbol{m}_t|\boldsymbol{m}_0) = \frac{1}{1 - e^{-t}}(e^{-t/2}\boldsymbol{m}_0 - \boldsymbol{m}_t). \tag{22}$$

We use a learned denoising network to approximate the conditional score of the full $\mathrm{SE}(3)$ transformation. The score is decomposed into rotational and translational components as follows:

$$\begin{aligned} \nabla_{\mathcal{T}_t} \log p_{t|0}(\mathcal{T}_t \mid \hat{\mathcal{T}}_0) &= (s_\theta^{\boldsymbol{r}}(t, \mathcal{T}_t), s_\theta^{\boldsymbol{m}}(t, \mathcal{T}_t)), \\ s_\theta^{\boldsymbol{r}}(t, \mathcal{T}_t) &= \nabla_{\boldsymbol{r}_t} \log p_{t|0}(\boldsymbol{r}_t|\hat{\boldsymbol{r}}_0), \\ s_\theta^{\boldsymbol{m}}(t, \mathcal{T}_t) &= \nabla_{\boldsymbol{m}_t} \log p_{t|0}(\boldsymbol{m}_t|\hat{\boldsymbol{m}}_0). \end{aligned} \tag{23}$$

D Architecture

Here we provide mathematical detail of PepBridge presented in method section. Let $\mathbf{h}^\ell = [h_1^\ell, \ldots, h_N^\ell] \in \mathbb{R}^{N \times D_h}$ denote the node embeddings at layer ℓ, where h_n^ℓ represents the embedding for residue n. Similarly, let $\mathbf{z}^\ell \in \mathbb{R}^{N \times N \times D_z}$ represent the edge embeddings, where z_{ij}^ℓ encodes the interaction between residues i and j.

Node embeddings are initialized using residue indices, atom coordinates, backbone dihedral angles, side-chain angles $h_{\mathcal{B}}$, and the diffusion timestep t. For edge (residue-pair) embeddings, we incorporate a combination of residue-type pairs, relative sequence positions, pairwise distances, and relative orientations. Each of these features is individually encoded using a dedicated multi-layer perceptron (MLP). The resulting feature vectors are concatenated and passed through another MLP to produce the final embeddings.

The initial layer-0 embeddings for residues i and residue pairs (i, j) are computed using MLPs applied to sinusoidal encodings $\phi(\cdot)$ of the input features:

$$\begin{aligned} h_i^0 &= \mathrm{MLP}([\phi(h_{\mathcal{B}_i}), \phi(t)]) \in \mathbb{R}^{D_h}, \\ z_{ij}^0 &= \mathrm{MLP}([\phi(h_{\mathcal{B}_i}), \phi(h_{\mathcal{B}_i}), \phi(i-j), \phi(dis(i,j)), \phi(ori(i,j)), \phi(t)]) \in \mathbb{R}^{D_z}, \end{aligned} \tag{24}$$

where D_h, D_z denote the dimensions of the node and edge embeddings, respectively. The functions $\phi(\mathrm{dis}(i,j))$ and $\phi(\mathrm{ori}(i,j))$ represent sinusoidal encodings of the distance and relative orientation between residues i and j.

To encode the surface of the receptor protein, we extract node-level features from the surface points and apply an MLP to obtain embeddings. Each surface node is represented by its 3D position (surf_t), hydrogen bonding potential ($\mathrm{surf}_{\mathrm{hbond}}$), and hydrophobicity score ($\mathrm{surf}_{\mathrm{hp}}$). These features are concatenated and passed through an MLP to produce the surface node embeddings:

$$h_{\mathrm{surf}} = \mathrm{MLP}([\mathrm{surf}_t, \mathrm{surf}_{\mathrm{hbond}}, \mathrm{surf}_{\mathrm{hp}}]). \tag{25}$$

For the peptide surface representation, we encode only the 3D positional coordinates using an MLP, omitting auxiliary features such as hydrogen bonding and hydrophobicity. At each diffusion timestep t, the model takes as input the receptor's node and edge embeddings, the noised peptide descriptors, and a sinusoidal embedding of the timestep. It predicts a denoising score that guides the reverse diffusion process toward the clean peptide descriptors at $t = 0$. The model architecture is based on Invariant Point Attention (IPA), which employs SE(3)-invariant attention to capture interactions between the receptor and the peptide. The output of the IPA module is passed through separate MLP decoders to reconstruct various ground-truth peptide descriptors, such as atom coordinates, dihedral angles, and residue types. Notably, certain residue types may be partially inferred from the number of side-chain dihedral angles, due to structural constraints.

E Experimental Details

The experiments were conducted on a computing cluster with 2 NVIDIA RTX A6000, each with 48 GB of memory. The total computation time for training was approximately 21 hours. We trained for 900000 steps with batch size 8. We used the Adam optimizer with a start learning of 5e-4. We also schedule to decay the learning rate exponentially with a factor of 0.6 and a minimum learning rate of 1e-6. The learning rate is decayed if there is no improvement for the validation loss in 10 consecutive evaluations.

E.1 Dataset

The filtered dataset underwent sequence-based clustering using MMseqs2 [45], resulting in 9,816 protein-peptide complexes organized into 292 distinct clusters. For systematic evaluation, we designated 10 clusters encompassing 158 complexes as the test set, with the remaining complexes allocated to training and validation cohorts.

E.2 Baselines

We briefly summarize the baselines and tools used in our study, including generative approaches for protein and peptide design, as well as side-chain packing methods.

- **ProteinGenerator** [27] is a RoseTTAFold-based diffusion model that jointly generates protein sequences and structures, with flexible conditioning on target sequence and structural attributes.
- **RFDiffusion** [50] is a generative model fine-tuned on structure denoising tasks, enabling high-accuracy design of monomers, binders, and symmetric protein assemblies.
- **PPFLOW** [26] is a target-aware peptide designer that performs conditional flow matching on torus manifolds to model peptide torsion-angle geometry.
- **PepFlow** [25] is a multimodal flow-matching model for full-atom peptide design targeting specific protein receptors. It jointly models backbone geometry, side-chain conformations, and residue identities over appropriate geometric manifolds.
- **PepGLAD** [21] combines geometric latent diffusion with receptor-specific transformations to generate full-atom peptides. The model operates in a learned latent space and adapts to diverse binding geometries for improved generalization.
- **Chroma** [19] is a unified generative framework for proteins and protein complexes that integrates a polymer-aware diffusion process with a scalable architecture, supporting constraint-driven design across sequence, structure, and function.
- **SCWRL4** [23] is a widely used side-chain packing tool employing a backbone-dependent rotamer library and a statistical energy function.
- **DLPacker** [34] is a 3D CNN-based model for residue side-chain conformation prediction. We utilize the official implementation along with the model weights.
- **AttnPacker** [32] utilizes equivariant attention mechanisms on backbone 3D geometry to predict all side-chain coordinates simultaneously.
- **DiffPack** [61] is a diffusion-based generative model that autoregressively samples side-chain angles on a toric manifold.

E.3 Training Metrics Details

RMSD. Root-Mean-Square Deviation is a widely used metric for assessing structural similarity between proteins. In our evaluation, we align the generated peptide to the native peptide within the complex using the Kabsch algorithm. considering only the peptide portion for superposition. We then compute the RMSD based on normalized C_α atom distances between the generated and native peptides. Lower RMSD values indicate greater structural similarity.

BSR. Binding Site Recovery measures the similarity of interaction patterns between the generated and native peptide-protein complexes. Specifically, it evaluates whether the generated peptide

engages target protein residues in a manner similar to the native peptide, potentially reflecting similar biological functions. A residue in the protein is considered part of the binding site if its C_β atom lies within 6 Å of any residue in the peptide. BSR is defined as the ratio of overlapping binding site residues between the generated and native complexes. Higher BSR values indicate greater similarity in binding interactions.

Consistency represents the statistical association between the clustering results of surface and structures. This metric quantifies how well a model captures the fundamental consistency between surfaces and their corresponding structures. A model that accurately represents the joint distribution should achieve a high score, while a low score indicates the model generates inconsistent surface-structure pairs. The evaluation process involves clustering both surfaces and structures, assigning discrete labels to each. These clustering labels can be interpreted as nominal variables. Given that similar surfaces should correspond to similar structures. We employ Cramér's V association [5] to measure this correlation, where a value of 1.0 indicates perfect association and 0.0 indicates no association. For surface representation, we first obtain molecular fingerprints using the methodology described in [44]. These fingerprints serve as input for the clustering algorithm, which assigns labels to the generated peptide surfaces.

Diversity. To assess diversity, we compute all pairwise TM-scores among the generated peptides for a given target using the original TM-align program. TM-scores quantify structural similarity between peptide pairs. We define diversity as 1 minus the average TM-score, where higher values indicate greater structural variability among the generated peptides. This metric reflects the breadth of structural exploration achieved during the design process.

E.4 Hyperparameters

Our proposed method incorporates several hyperparameters, including sample step, learning rate, batch size and feature dimensions. To validate these hyperparameters, we conducted a random search. The search space are presented in Table 4.

Table 4: Search space for all PepBridge. The parameters used in validation are marked in **bold**

Parameter	Search Space
Learning rate	0.0009, 0.0007, **0.0005**, 0.00001
Hidden dimension of residue feature	64, **128**, 256
Hidden dimension of edge feature	**64**, 128, 256
Hidden dimension of surface feature	**16**, 24, 32
Number of attention heads	**8**, 16, 24
Loss weight of surface	0.1, **0.5**, 1
Loss weight of backbone position	0.1, 0.5, **1**
Loss weight of backbone rotation	0.1, 0.5, **1**
Sampling steps	500, **1000**, 1500
Training steps	500, **1000**, 1500
Batch size	4, **8**, 16

E.5 Computational Complexity

We compared PepBridge with several baseline methods in terms of training time, inference time per sample, GPU usage, and model size. The time cost is reported as the total time spent divided by the number of designed candidates. As summarized in Table 5, multi-modal processing does introduce additional complexity relative to uni-modal approaches, our analysis shows that PepBridge achieves a good balance between computational efficiency and performance.

Table 5: Computational cost and resource footprint across methods. Training and inference times are normalized per designed sample.

Method	Training time	Inference time (s/sample)	GPU(s) used	Params (M)
RFdiffusion	3 days	80–180	8×A100	∼120
ProteinGenerator	4 weeks	152	64×V100	∼120
PepFlow	20 hours	14–24	2×A6000	∼7
Chroma	10 weeks	185–226	8×V100	∼20
PepGLAD	29 hours	3	1×24 GB GPU	∼2.5
PepBridge	21 hours	16–37	2×A6000	∼10

F Additional Experiments

F.1 Visualization

Figure 1 provides additional examples of peptides generated by PepBridge. These visualizations include both surface and backbone structures of the generated peptides in a top-down view, further illustrating the model's ability to produce geometrically coherent and interface-aware designs.

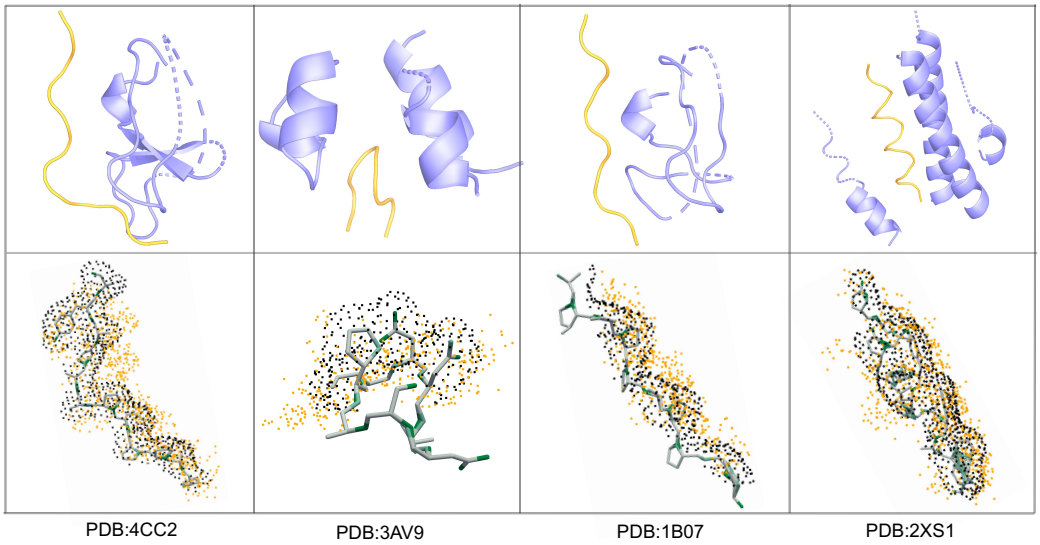

Figure 4: Visualization of generated peptides by PepBridge. **Top:** Generated peptides (in orange) for receptors (in purple). The PDB ID of the receptors are 4CC2, 3AV9, 1B07, and 2XS1. **Bottom:** The generated backbone structure and surface. The ground-truth surface structure (in black) and generated surface (in orange) are shown to compare the ability of interface caption.

F.2 Training and Sampling Time Steps

We ablated the number of diffusion steps used during training while fixing the sampling procedure at 1000 steps. The results are summarized in Table 6. Increasing the number of steps consistently improved all quality metrics. The largest gains occurred up to 1000 steps, with smaller but still measurable improvements between 1000 and 1500 steps. Given these trends, we adopt 1000 training steps as a favorable trade-off between overall accuracy and computational cost.

We further conducted experiments to assess how different inference time steps affect the final performance metrics using the model trained with 1000 steps. As shown in Table 7, the results indicate that longer sampling chains (e.g., 1000 steps) improve the quality of the generated structures. However, performance gains begin to plateau beyond 800 steps, and we observe a slight decrease in structural diversity, suggesting a trade-off between generation determinism and diversity. Based on this analysis, we selected 1000 steps as the default setting to achieve the best overall balance.

Table 6: Effect of training steps on performance (sampling fixed at 1000). Higher is better (↑) except RMSD (↓).

	Div_{stru} (↑)	Aff. % (↑)	Stab. % (↑)	RMSD Å(↓)	BSR (↑)
PepBridge (time step =500)	0.57	18.96	24.79	2.96	82.84
PepBridge (time step =800)	0.61	19.07	26.31	2.36	85.57
PepBridge (time step =1000)	0.59	19.16	25.02	2.19	83.90
PepBridge (time step =1500)	0.62	23.28	26.68	2.11	86.92

Table 7: Effect of sampling steps on performance (model trained with 1000 steps). Higher is better (↑) except RMSD (↓).

	Div_{stru} (↑)	Aff. % (↑)	Stab. % (↑)	RMSD Å(↓)	BSR (↑)
PepBridge (time step =500)	0.61	17.42	23.97	2.85	80.05
PepBridge (time step =800)	0.62	18.77	24.58	2.69	82.33
PepBridge (time step =1000)	0.59	19.16	25.02	2.19	83.90
PepBridge (time step =1500)	0.56	18.46	24.71	2.74	83.78

G Limitations and Future Work

While PepBridge presents a structured approach to joint protein surface and backbone design, several limitations remain that can be addressed in future work. One key limitation lies in the simplification of surface representations. The current model relies on solvent-accessible point clouds with biochemical annotations, which, while effective, may not fully capture finer molecular interactions such as electrostatic potential fields and solvent dynamics. These factors play crucial roles in protein-protein interactions and could enhance the accuracy of designed peptides if incorporated. Another limitation is the model's reliance on receptor geometry. PepBridge assumes that receptor surface features sufficiently dictate the constraints on peptide binding. However, this does not account for receptor flexibility or conformational changes upon ligand binding, which are common in many biological systems. Addressing this aspect could make the model more applicable to highly dynamic binding sites. Computational efficiency also poses a challenge. The diffusion bridge model and SE(3) diffusion backbone generation require computationally intensive sampling. While the hierarchical generation process improves efficiency, generating high-quality peptides remains expensive, particularly for longer sequences. Further optimization is necessary to reduce the computational cost while maintaining or improving accuracy. Additionally, the current evaluation primarily focuses on geometric complementarity and binding affinity predictions. While these provide useful insights, they do not fully capture the functional stability of designed peptides. Wet-lab experiments and molecular dynamics simulations are necessary to assess real-world applicability, ensuring that the generated structures remain stable under physiological conditions.

Future work can address these limitations in several ways. Enhancing surface representations by incorporating higher-order biochemical features such as electrostatic potential fields, solvent effects, or graph-based molecular embeddings could improve the precision of surface-conditioned peptide generation. Furthermore, integrating receptor flexibility into the model by leveraging conformational ensembles or reinforcement learning-based refinement strategies would allow for more realistic modeling of dynamic binding sites. To improve computational efficiency, future research could explore accelerated sampling techniques, such as adaptive noise schedules, score distillation sampling (SDS), or flow-matching approaches. These methods could significantly reduce inference time while preserving or enhancing model accuracy. Finally, validating PepBridge through real-world applications, particularly in therapeutic peptide design, remains a crucial next step. Incorporating experimental validation through biochemical assays and integrating co-evolutionary signals into the design process could further enhance the biological relevance of the generated structures. By addressing these challenges, PepBridge can be refined to enable more accurate, efficient, and versatile protein design.