Appendix Contents

This appendix provides additional technical details and experimental insights. We first describe the backbone representation and our diffusion processes on relevant geometric manifolds. We then outline the model architecture and training setup, including datasets, baselines, and hyperparameters. Further sections present additional experiments, followed by a discussion of limitations and directions for future work.

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 (Yim et al., 2023) and (Yang et al., 2023), we set the local coordinates as:

$$\begin{aligned} \mathrm{N}^{\star} &= (-0.525, 1.363, 0.0) \\ \mathrm{C}_{\alpha}^{\star} &= (0.0, 0.0, 0.0) \\ \mathrm{C}^{\star} &= (1.526, 0.0, 0.0) \\ \mathrm{O}^{\star} &= (0.627, 1.062, 0.0) \end{aligned} \tag{1}$$

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:

$$\begin{aligned} [\mathrm{N}_n, \mathrm{C}_n, (\mathrm{C}_{\alpha})_n, \mathrm{O}_n] = \big[\, & T_n \cdot \mathrm{N}^{\star},\ T_n \cdot \mathrm{C}^{\star}, \\ & T_n \cdot \mathrm{C}_{\alpha}^{\star},\ T_n \cdot T_{\text{psi}}^{\star}(\psi_n) \cdot \mathrm{O}^{\star}\,\big]. \end{aligned} \tag{2}$$

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:

$$\begin{aligned} &\boldsymbol{\omega}_1 = \mathrm{C}_n - (\mathrm{C}_{\alpha})_n, \qquad \boldsymbol{\omega}_2 = \mathrm{N}_n - (\mathrm{C}_{\alpha})_n \\ &\boldsymbol{e}_1 = \boldsymbol{\omega}_1 / \|\boldsymbol{\omega}_1\|, \qquad \boldsymbol{u}_2 = \boldsymbol{\omega}_2 - \boldsymbol{e}_1(\boldsymbol{e}_1^T \boldsymbol{\omega}_2), \\ &\boldsymbol{e}_2 = \boldsymbol{u}_2 / \|\boldsymbol{u}_2\|, \\ &\boldsymbol{e}_3 = \boldsymbol{e}_1 \times \boldsymbol{e}_2, \\ &\boldsymbol{r}_n = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)\,, \\ &\boldsymbol{m}_n = (\mathrm{C}_{\alpha})_n, \\ &T_n = (\boldsymbol{r}_n, \boldsymbol{m}_n). \end{aligned} \tag{3}$$

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: $T_{\text{psi}}^{\star}(\psi_n) = (\boldsymbol{r}_x(\psi_n), \boldsymbol{m}_{\text{psi}})$, where ψ_n represents a rotation angle along x-axis. The transformation funtions are defined as:

$$\boldsymbol{r}_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}, \quad \boldsymbol{m}_{\text{psi}} = (1.526, 0.0, 0.0). \tag{4}$$

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 $\mathrm{C}_{\alpha}^{\star}$. The combined transformation $T_n \cdot T_{\text{psi}}^{\star}(\psi_n)$ 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 (Yim et al., 2023), 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 $\mathrm{SO}(3)$:

$$f(\omega, t) = \textstyle\sum_{\ell\in\mathbb{N}} (2\ell + 1) e^{-\ell(\ell+1)t/2} \frac{\sin((\ell+1/2)\omega)}{\sin(\omega/2)}. \tag{5}$$

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

$$\nabla \log p_{t|0}(\boldsymbol{r}_t \mid \boldsymbol{r}_0) = \tfrac{\boldsymbol{r}_t}{\omega_t} \log(\boldsymbol{r}_0^\top \boldsymbol{r}_t) \frac{\partial_\omega f(\omega^{(t)}, t)}{f(\omega^{(t)}, t)}, \tag{6}$$

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:

$$p_{t|0}(\boldsymbol{m}_t|\boldsymbol{m}_0) = \mathcal{N}(x_t; e^{-t/2}\boldsymbol{m}_0, (1 - e^{-t})\,\mathrm{I}_3). \tag{7}$$

Then we can get the score as:

$$\nabla \log p_{t|0}(\boldsymbol{m}_t|\boldsymbol{m}_0) = \frac{1}{1 - e^{-t}}(e^{-t/2}\boldsymbol{m}_0 - \boldsymbol{m}_t). \tag{8}$$

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:

$$\begin{aligned} \nabla_{\mathcal{T}_t} \log p_{t|0}(\mathcal{T}_t \mid \hat{\mathcal{T}}_0) &= (s_\theta^{\boldsymbol{r}}(t, \mathcal{T}_t), s_\theta^{\boldsymbol{m}}(t, \mathcal{T}_t)), \\ s_\theta^{\boldsymbol{r}}(t, \mathcal{T}_t) &= \nabla_{\boldsymbol{r}_t} \log p_{t|0}(\boldsymbol{r}_t|\hat{\boldsymbol{r}}_0), \\ s_\theta^{\boldsymbol{m}}(t, \mathcal{T}_t) &= \nabla_{\boldsymbol{m}_t} \log p_{t|0}(\boldsymbol{m}_t|\hat{\boldsymbol{m}}_0). \end{aligned} \tag{9}$$

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:

$$\begin{aligned} h_i^0 &= \mathrm{MLP}([\phi(h_{\mathcal{B}_i}), \phi(t)]) \in \mathbb{R}^{D_h}, \\ z_{ij}^0 &= \mathrm{MLP}([\phi(h_{\mathcal{B}_i}), \phi(h_{\mathcal{B}_i}), \phi(i-j), \phi(dis(i,j)), \phi(ori(i,j)), \phi(t)]) \in \mathbb{R}^{D_z}, \end{aligned} \tag{10}$$

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:

$$h_{\mathrm{surf}} = \mathrm{MLP}([\mathrm{surf}_t, \mathrm{surf}_{\mathrm{hbond}}, \mathrm{surf}_{\mathrm{hp}}]). \tag{11}$$

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 Steinegger & Söding (2017), 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** (Lisanza et al., 2023) is a RoseTTAFold-based diffusion model that jointly generates protein sequences and structures, with flexible conditioning on target sequence and structural attributes.
- **RFDiffusion** (Watson et al., 2023) is a generative model fine-tuned on structure denoising tasks, enabling high-accuracy design of monomers, binders, and symmetric protein assemblies.
- **PPFLOW** (Lin et al., 2024) is a target-aware peptide designer that performs conditional flow matching on torus manifolds to model peptide torsion-angle geometry.
- **PepFlow** (Li et al., 2024) 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** (Kong et al., 2024) 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** (Ingraham et al., 2023) 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** (Krivov et al., 2009) is a widely used side-chain packing tool employing a backbone-dependent rotamer library and a statistical energy function.
- **DLPacker** (Misiura et al., 2022) is a 3D CNN-based model for residue side-chain conformation prediction. We utilize the official implementation along with the model weights.
- **AttnPacker** (McPartlon & Xu, 2022) utilizes equivariant attention mechanisms on backbone 3D geometry to predict all side-chain coordinates simultaneously.
- **DiffPack** (Zhang et al., 2023) 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 Cramér (1999) 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 Song et al. (2024). 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 1.

Table 1: 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 2, 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 2: 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.

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 3. Increasing the number of steps consistently improved all quality metrics. The largest gains occurred up to 1000 steps, with smaller but still

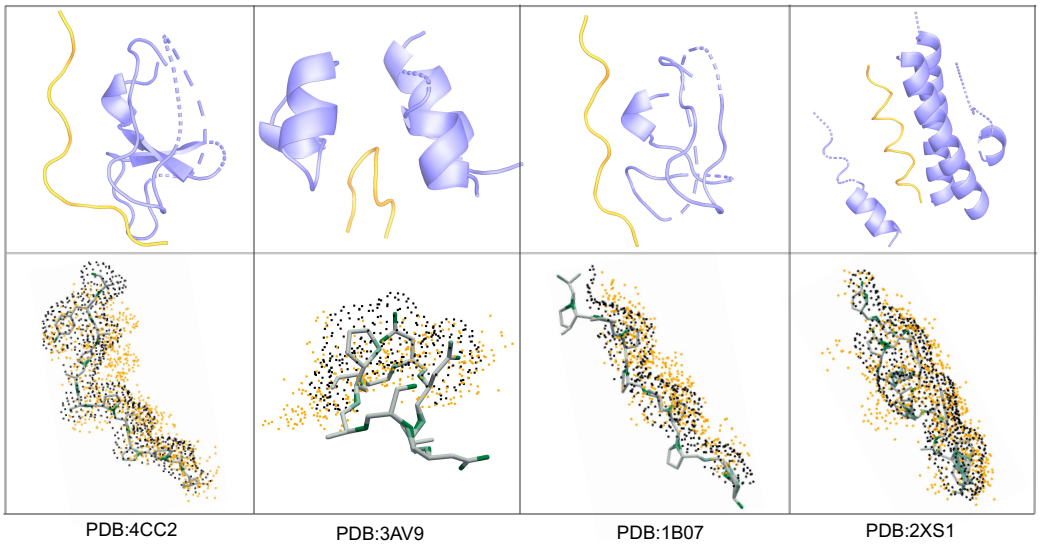

Figure 1: Visualization of generated peptides by PepBridge. **Top:** Generated peptides (in orange) for receptors (in purple). The PDB ID of the receptors are 4CC2, 3AV9, 1B07, and 2XS1. **Bottom:** The generated backbone structure and surface. The ground-truth surface structure (in black) and generated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

Table 3: 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

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 4, 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.

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

Table 4: 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

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.

References

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

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.

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.

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.

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.

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.

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.

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

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.

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

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.

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.

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.

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.

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.